

Introduction

This penetration testing environment is to be used to generate and gather data from simulated attacks on computers, and later, IoT (Internet of Things) devices deployed on protected and unprotected networks. With such a dataset, automated systems such as IDS (Intrusion Detection System) and IPS (Intrusion Prevention System) could be better trained and updated to recognize and act on various types of attacks.

This guide aims to help whoever may wish to replicate this project or take it over in the future. This guide will contain step-by-step instructions outlining each step taken to create this pen-testing environment. Since this is an ongoing project, this document will be dynamic and continuously changing, therefore, **everything in this document is subject to change**.

Before we begin, one should make sure that they have the required hardware to recreate this penetration testing environment:

- a USB drive with **at least 8GB** of space
- 3 physical computers with a **minimum of 12GB of RAM** and an **6 core CPU**
- 6 Ethernet cables
- a **managed switch** such as [this one here](#)
- a **router** such as [this one here](#)

Physical Device Specifications

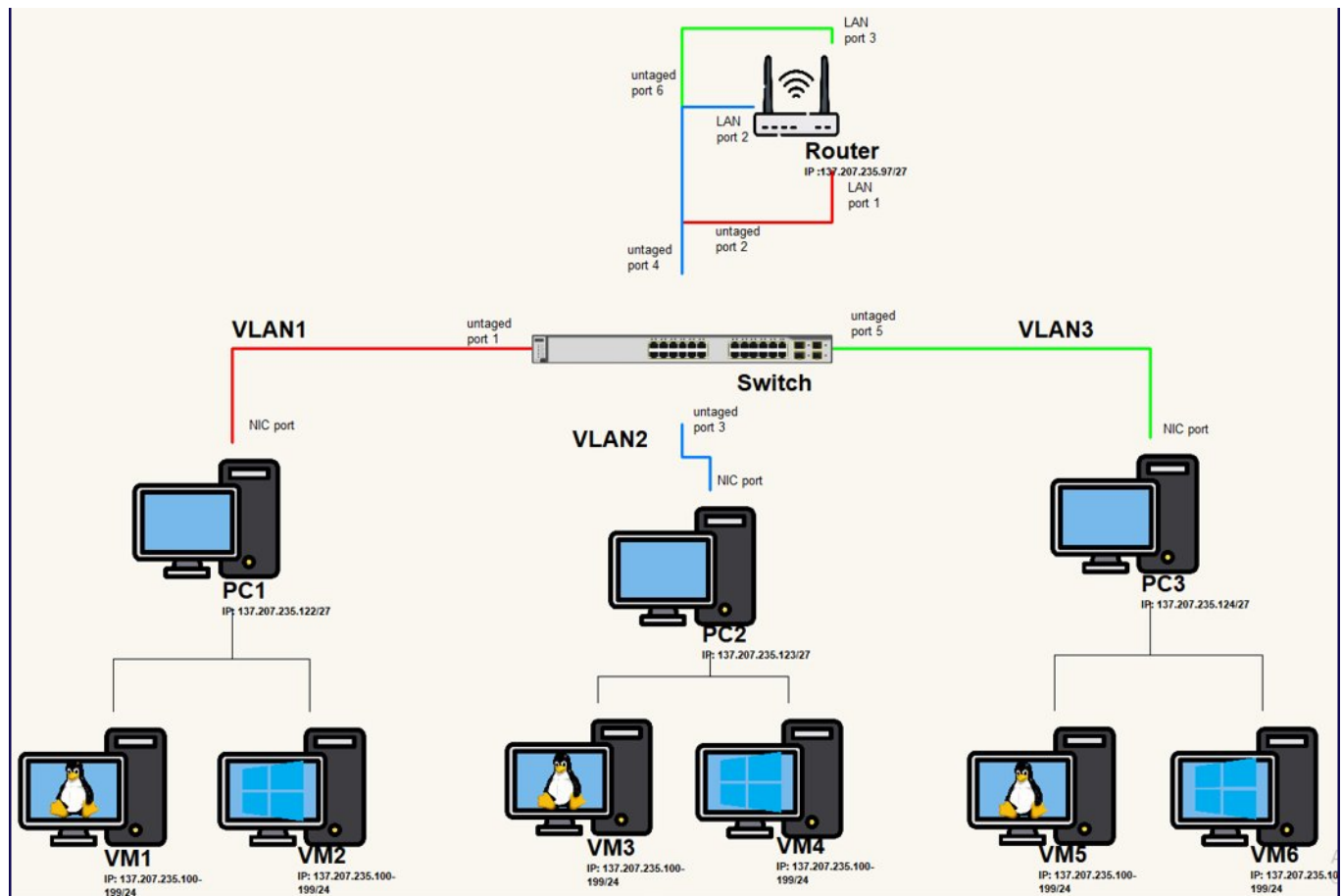
Below are the hardware specifications of the PCs that we used to setup this lab environment. These specifications would be considered as the minimum requirements for such a setup.

- **Memory:** 12 GB DDR3 RAM @ 2600 MHz
- **Processor:** Intel i5-8400 6-core/6-thread CPU @ 2.8 GHz
- **Hard Disk Drive:** 1000 GB 7200 RPM SATA HDD

Project Overview

In short, this pen-testing environment will consist of a total of six virtual machines divided over three physical host devices. Each VM (Virtual Machine) will have its network card configured in a way that allows all VMs to communicate with one another as well as with each physical host (including the physical that a VM is deployed on). In order to simulate the VMs being on different networks, there is a managed switch which is used

to create the three VLANs (Virtual Local Area Networks); one VLAN per physical host (or per two VMs). Finally, there is a physical routing device that is in charge of routing network traffic between the different VLANs. Below is a simplified diagram of the current network topology.



1: Installing Linux

Before we can create the virtual machines to simulate our network, we must first install a flavour of Linux onto each physical machine. We have chosen [Ubuntu 22.04.1 LTS](#) as our host operating system and as one of the guest operating systems for the virtual machines (more on that later). Installing the operating system on the physical systems can be done by following the steps below.

1. Download the Image file (.iso extension) from the [official Ubuntu download page](#)
2. Use an image burning tool such as [Balena Etcher](#) (works on Windows, Mac and Linux) or [Rufus](#) (only Windows) to burn the image file onto a bootable piece of media such as a USB stick with at least 8GB of space
3. Insert the newly created bootable media into the computer you wish to install the operating system on and follow the instruction on the screen to proceed with the installation

Warning

It's important to make sure that you use an **empty** USB drive when burning the image file onto it as the burning process will wipe all existing data on the USB drive.

2: Installing VirtualBox

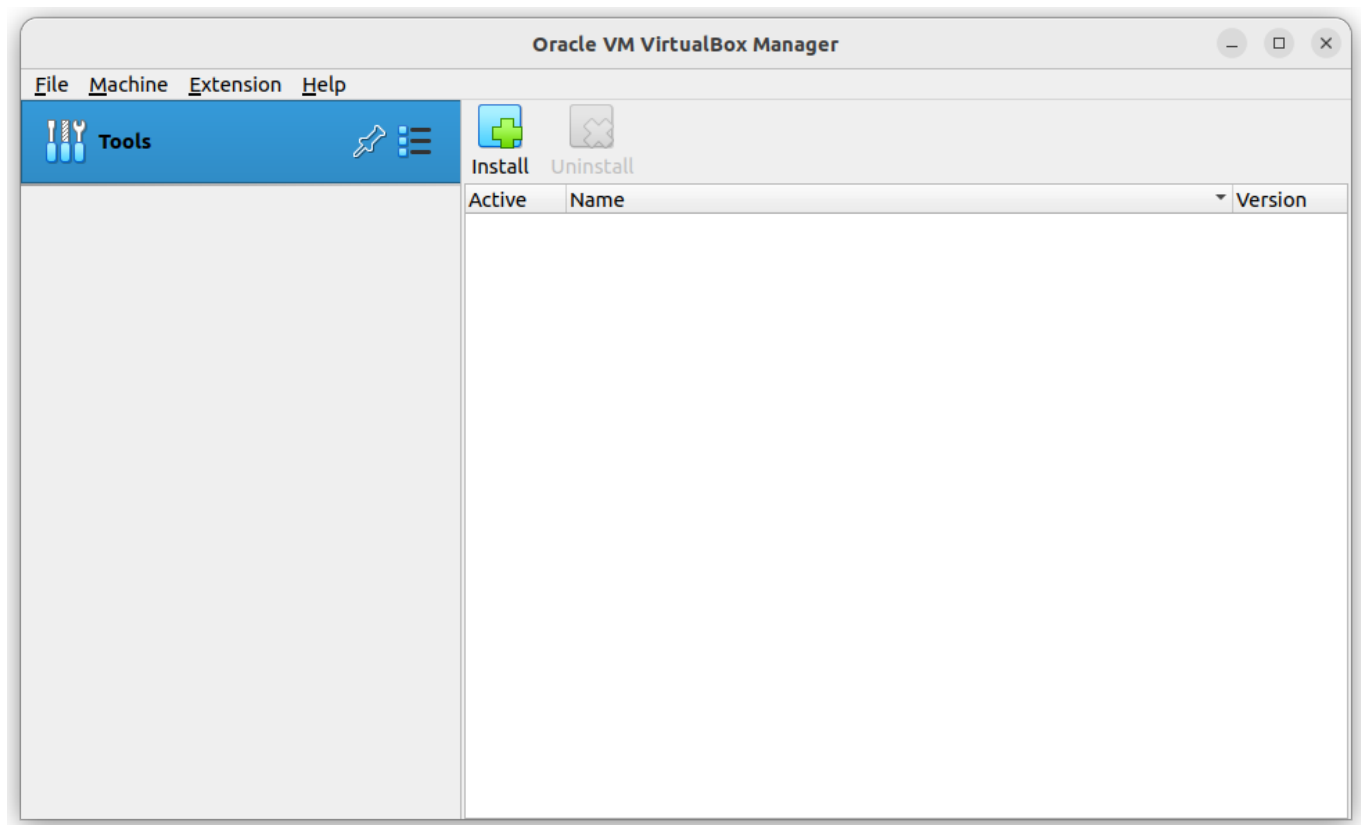
Once the host operating system has been installed, it is time to install [VirtualBox](#) which will be our way to create and manage our virtual machines. To install VirtualBox head over to the [official download page](#) and download version **7.0.4 for Ubuntu 22.04**. Using another version of VirtualBox is not prohibited but certain features may vary slightly from our version or be buggy. You may also download the SHA256 or MD5 checksums to verify your downloads if you wish. Once you have downloaded the **.deb** file for VirtualBox, locate it in its directory and install it with the following command:

```
sudo dpkg -i virtualbox-7.0_7.0.4-154605~Ubuntu-jammy_amd64.deb
```

Note

Don't forget to change the filename in the command above if you downloaded a different version!

Finally, the last thing we will need is the **VirtualBox Extension Pack** which can be [found here](#). The extension pack is not strictly necessary, but it does provide VirtualBox some extra functionalities such as disc encryption, support for USB 3.X devices and much more. To install this extension pack, click on the **menu icon** (three dots with three horizontal lines) and then click on **Extension**. This should open a window on the right side of the VirtualBox window with a **Install** button at the top of it, click this button and locate the extension pack that you downloaded. The extension pack should now be installed. Refer to the picture below for a reference.



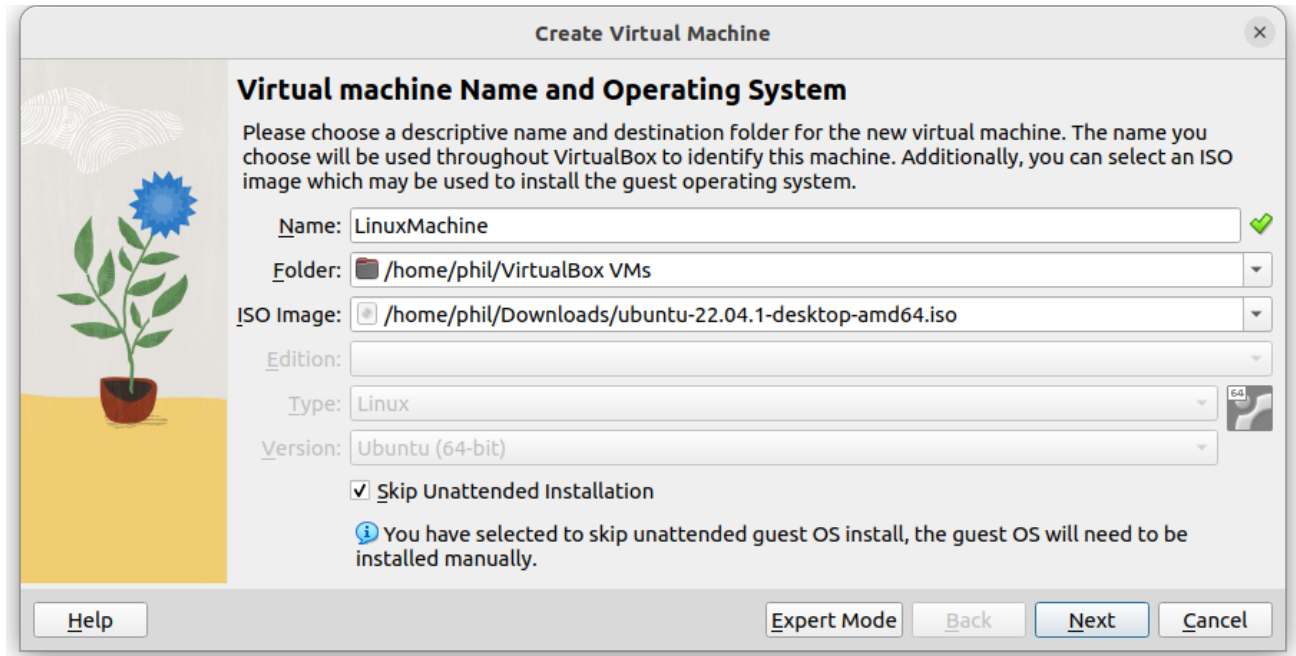
3: Creating the Virtual Machines

Now we can finally start creating our virtual machines! Since it is much easier to follow along with pictures this is what we will do here.

3.1: Initial Configurations

1. On the home screen of VirtualBox, click on **Machine** and select **New**
2. Give your machine a name, it can be anything
3. VMs are like files, chose a directory to store this file on your local machine, the default is usually fine
4. Select the Ubuntu 22.04 ISO image file you downloaded from earlier
5. Check the **Skip Unattended Installation** box below

6. Hit Next



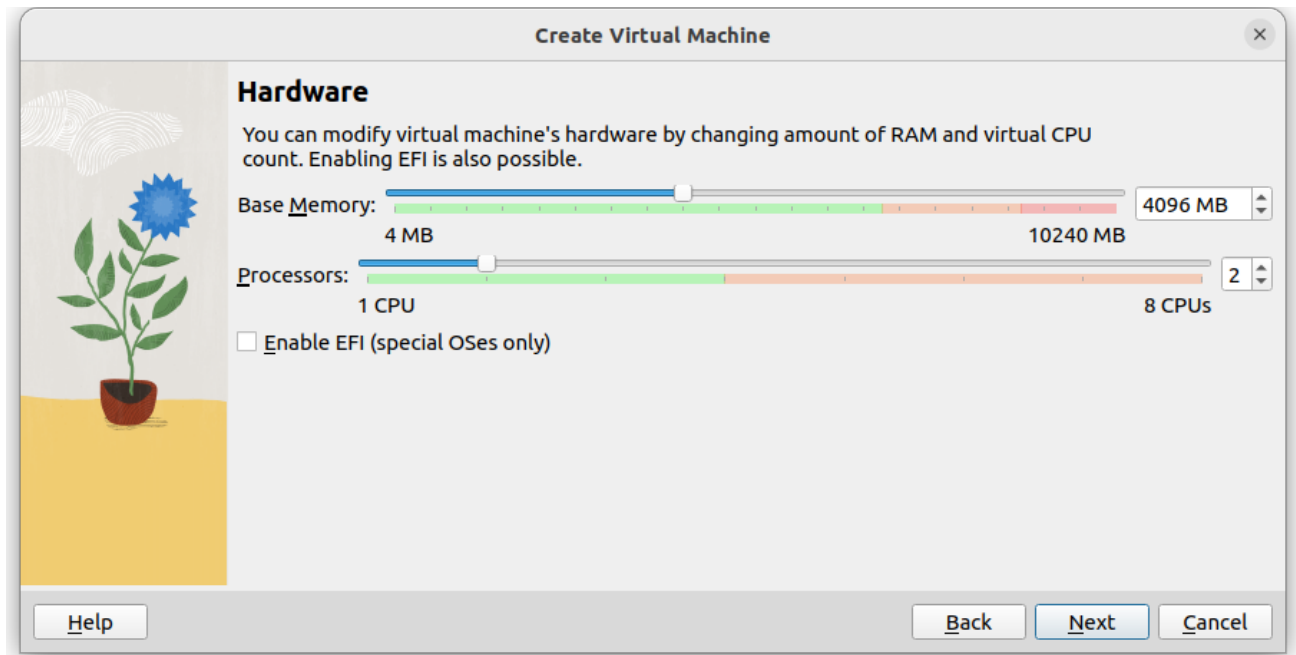
The screenshot shows the 'Create Virtual Machine' window with the 'Virtual machine Name and Operating System' tab selected. On the left is a decorative illustration of a blue flower in a red pot on a yellow surface. The main area contains the following fields and options:

- Name:** LinuxMachine (with a green checkmark icon)
- Folder:** /home/phil/VirtualBox VMs
- ISO Image:** /home/phil/Downloads/ubuntu-22.04.1-desktop-amd64.iso
- Edition:** (empty dropdown)
- Type:** Linux
- Version:** Ubuntu (64-bit)
- ☒ **Skip Unattended Installation**
- Information text: You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.

At the bottom, there are buttons for 'Help', 'Expert Mode', 'Back', 'Next' (highlighted in blue), and 'Cancel'.

3.2: Allocating Resources

1. Allocate at least **4GB of RAM** to the machine
2. Allocate at least **2 CPUs** to the machine
3. Hit **Next**



The screenshot shows the 'Create Virtual Machine' window with the 'Hardware' tab selected. On the left is the same decorative illustration of a blue flower in a red pot. The main area contains the following settings:

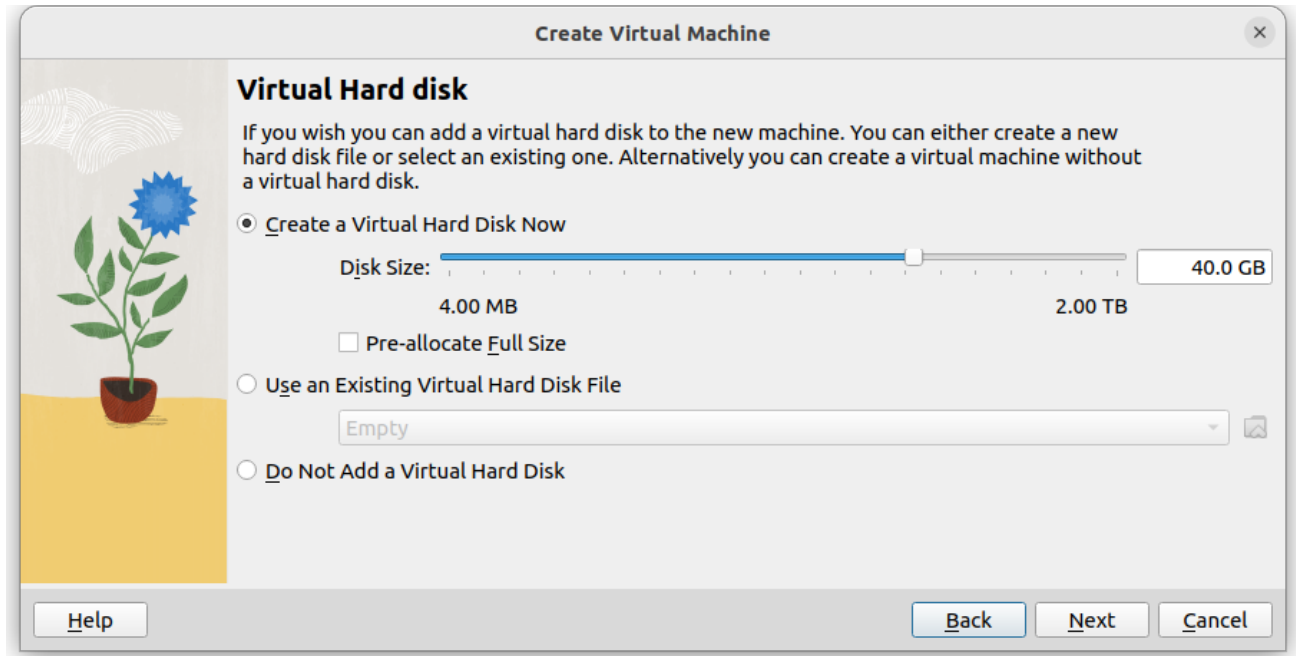
- Base Memory:** A slider set to 4096 MB, with a range from 4 MB to 10240 MB.
- Processors:** A slider set to 2, with a range from 1 CPU to 8 CPUs.
- ☐ **Enable EFI (special OSes only)**

At the bottom, there are buttons for 'Help', 'Back', 'Next' (highlighted in blue), and 'Cancel'.

3.3: Allocating Storage

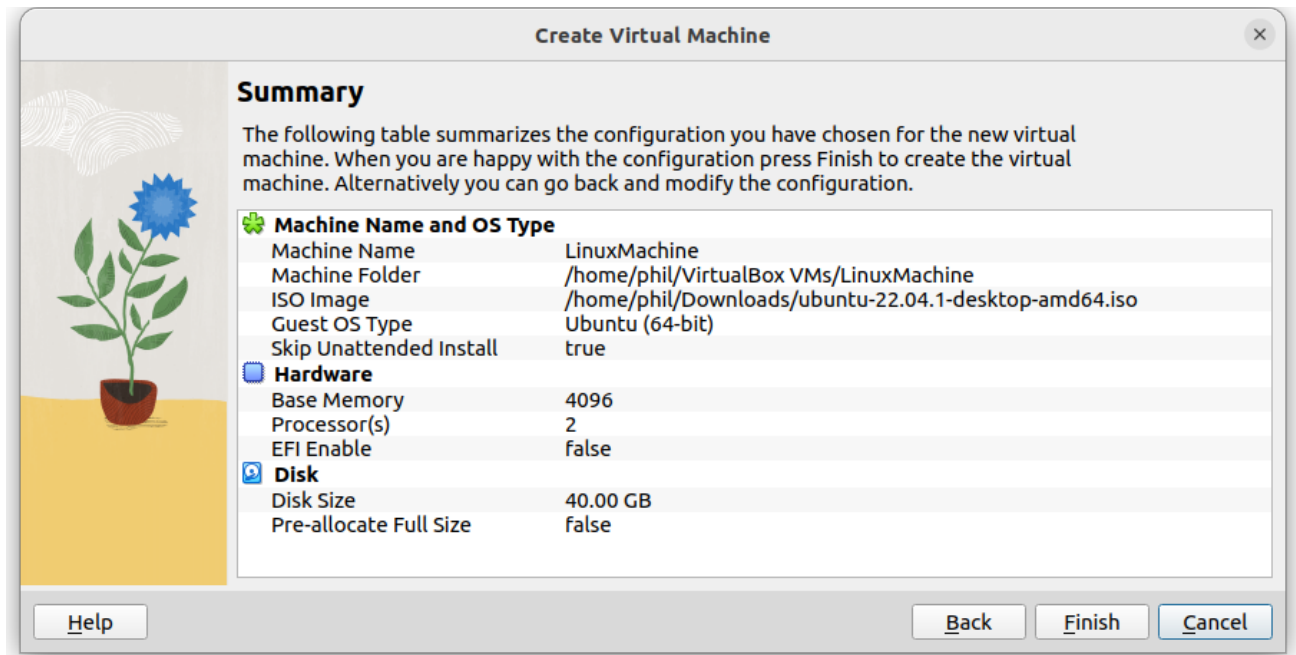
1. Allocate at least **40GB** of storage space (we may need the extra space later on)

2. Hit Next



3.4: Summary

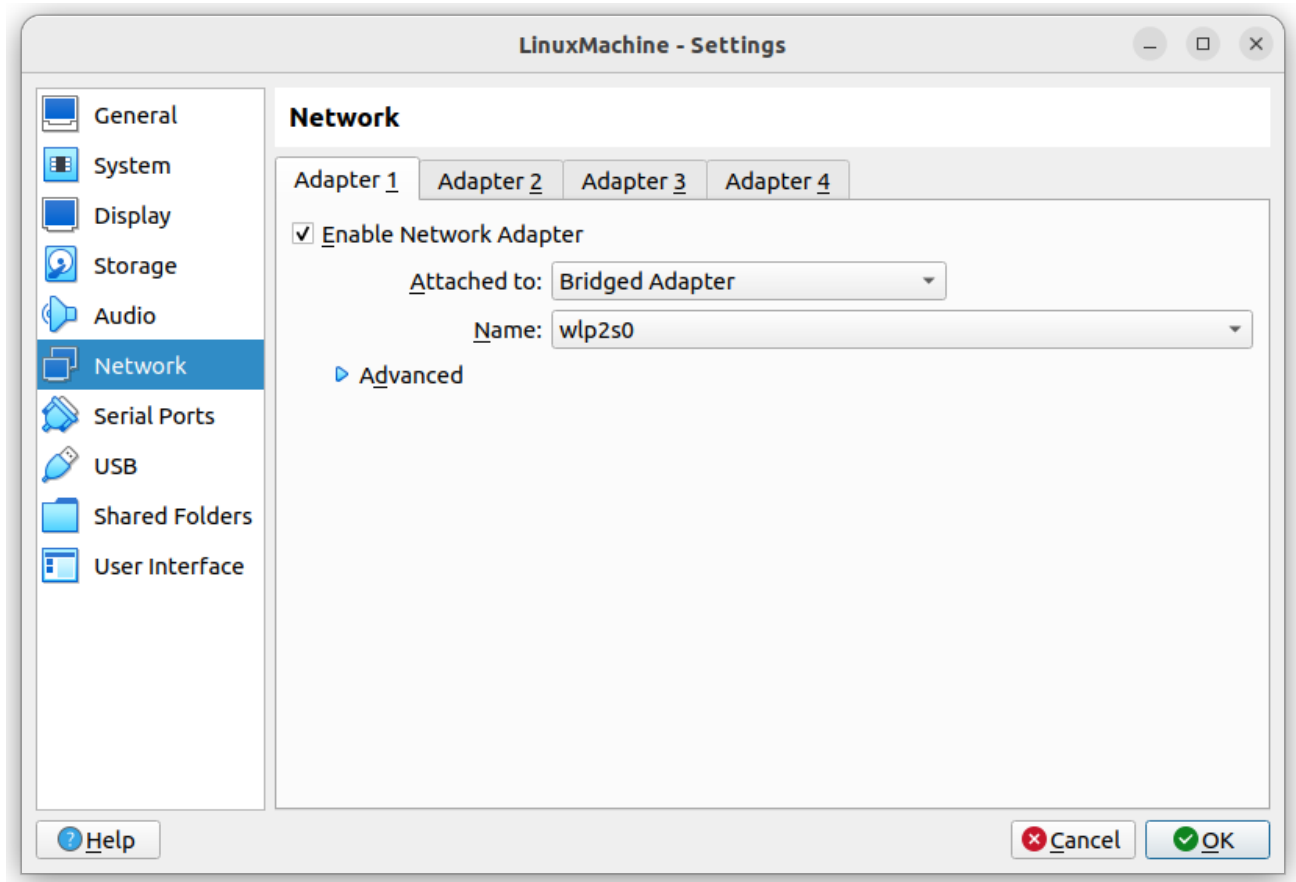
1. Make sure that your summary looks similar to the one in the image below, it's ok for the file paths to the VM and the ISO file to be different.
2. Hit **Finish**



3.5: Configuring the Network Card

1. Hit **Settings** followed by **Network**
2. In the **Attached to** drop down menu select **Bridged Adaptor**

3. Hit **OK**



The short reason why we choose the **Bridged Adaptor** is because this will allow this VM-to-VM communication as well as host-to-VM and VM-to-host, regardless of which host the VMs are on. For a more detailed description of the network options available in VirtualBox, please refer to [this website which covers them all in great detail](#) .

Once the VM has been created, you can simply hit the **Start** button with the green arrow and your VM will boot up just like a normal computer would. All that is left to do now is to follow the installation instruction provided by the Ubuntu operating system once the VM boots up. If you find the VM to be very slow and you still have plenty of system resources to spend, simply allocate more RAM and/or CPUs by clicking on **Settings > System** and adding more of whatever is needed.

4: Installing Windows

We will be using the latest version of **Windows 10** on the Windows VMs. The steps taken here will be very similar to the ones taken in step 1 where we installed Ubuntu except that we will not need our ISO burning tool since we will only use this ISO file to create VMs. To download the Windows 10 ISO file, simply head over to the [official Microsoft download page](#) and select which edition you wish to download (there is currently only one edition to download).

Once the ISO file is downloaded, you can refer to steps 3.X to create the Windows VMs as the process will be very similar, if not identical. Again, if you find the Windows VM to be very slow, feel free to allocate more system resources to it as need. Lastly, in order to avoid any networking issues with the Windows VMs, follow these **Windows only** steps once the VM is created and booted up.

add screen shot of disabling windows defender firewall with steps

Note

Some key points to consider when installing Windows 10:

- During the installation, choosing the **Home** or **Pro** version makes no difference although we chose the **Pro** version
- When asked for a **Product Key**, simply select the **I Don't Have a Product Key** option in the bottom right corner of that window and proceed with the installation as normal

5: Configuring the Managed Switch

We decided to use a managed switch for the sole purpose of being able to divide our network into three separate networks. In order to configure it, enter the switch's **default IP address** which is located on the **bottom side** on the switch itself. Once logged in navigate to the **VLAN** tab on the top of the screen and then select **Advanced > VLAN Configuration** and set the **Advanced 802.1Q VLAN** option to **Enable**.

5.1: Creating VLANs

1. Type the VLAN ID (an integer) that you want to use into the **VLAN ID** field and then click **Add**

5.2: Adding ports to the VLANs

1. Select **VLAN > 802.1Q > Advanced > VLAN Membership**
2. Select the VLAN ID that you want to add ports to by using the **VLAN ID drop down menu**
3. Continue clicking on the ports until a **U** appears in the port itself ("U" stands for "Untagged")

Note

Make sure to not assign more than **two ports per VLAN**; one will connect to the router and the other will go directly to one of the physical computers.

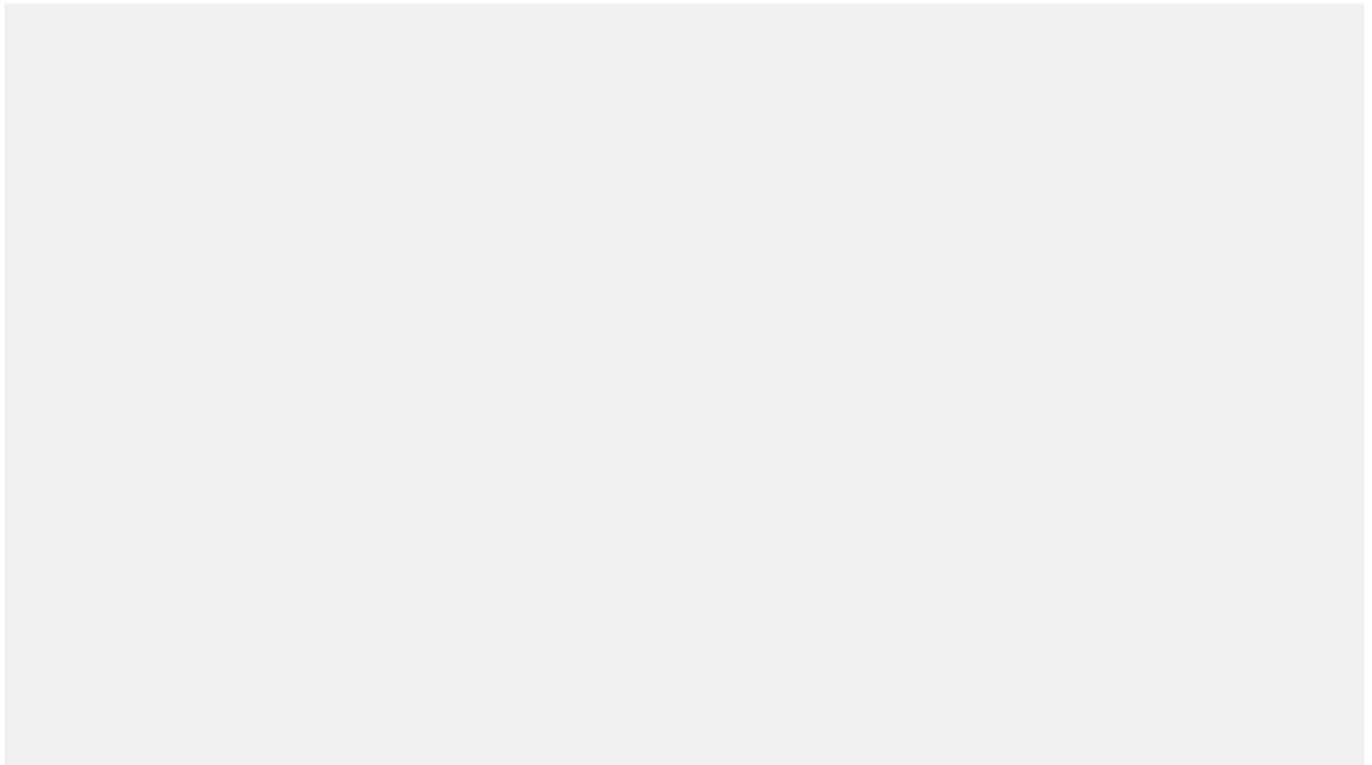
5.3 Configure port PVID settings for untagged ports

1. Select **VLAN > 802.1Q > Advanced > Port PVID**
2. Assign every port that was configured as **untagged** to its proper VLAN ID (example: if ports 1 and 2 were assigned to VLAN 1, set the PVID of ports 1 and 2 to 1)

The steps above can be repeated as many times as needed to create more VLANS if needed.

6: Configuring the Router

Configuring the router is a fairly straight forward process which only requires us to change a handful of things from the default setup. Firstly, we need to log into the router's configuration page using default access page which can be found on the **bottom side of the router**. Once logged in, follow the first-time setup instructions on screen to configure the router and ignore any errors related to Internet access or the WAN port not having anything plugged in. Finally, after the initial setup phase is completed, head over to the **Advanced** tab at the top of the screen and then select **Network > DHCP Server** from the left-hand side menu. You should be greeted with a similar screen:



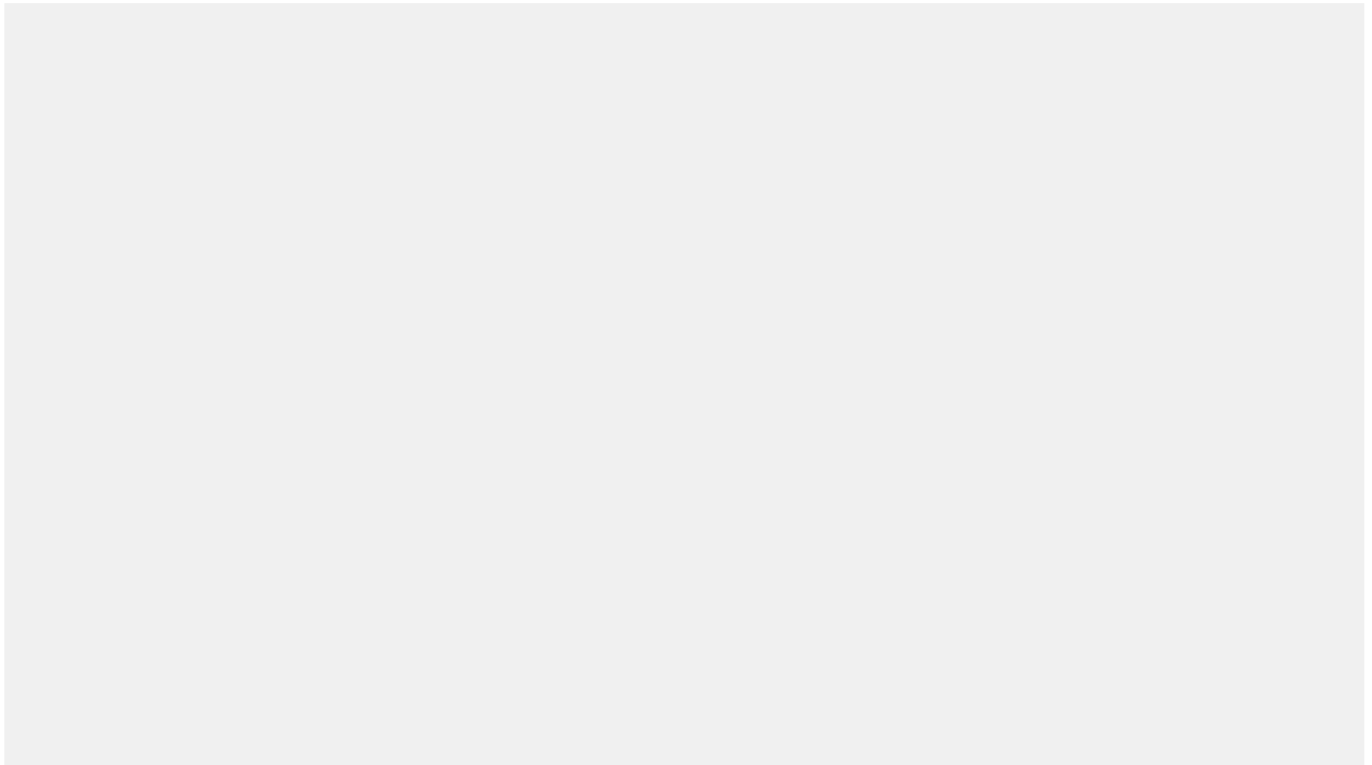
Almost everything that we need to modify for our specific setup is on this screen. Here is a list of all the changes that must be made to configure the network:

- Configure the **address pool** to start from [] all the way through []
- Leave the default address lease time to 120
- Change the **default gateway** of the router to []
- Set the **primary DNS** address to []
- Set the **secondary DNS** address to be []

Once all of those changes have been made, your screen should be identical to the image above. Now for the final configurations, head over to the **Network > LAN** page using the left-hand side menu and change the following:

- Change the **IP Address** to [] (if it hasn't automatically been changed from the previous steps)
- Set the **Subnet Mask** to **255.255.255.0** if it is not already selected

Once complete, your screen should look identical to the one below.



7: Final Test

As a final and precautionary test, the connectivity of each physical devices as well as their VMs can be tested by sending ICMP ping requests to each of them (physical

computer and VMs) from all the other machines (physical computers and VMs). Once connectivity is confirmed, the penetration-testing environment is ready for use.