Project 1: SET UP A HOME LAB

In this project, I set up a home lab using VirtualBox and two VMs, Ubuntu and Kali-Linux.

My Home Lab Setup for Cybersecurity Practice

Introduction

I decided to create a home lab that would allow me to safely experiment with network attacks and defenses. After some research, I chose to use VirtualBox as my platform and set up two virtual machines (VMs): one running **Ubuntu** for defensive measures, and the other running **Kali Linux** for offensive security testing. By connecting both VMs through a NAT network, I was able to simulate real-world cyberattacks and practice defending against them in a controlled environment.

In this report, I will walk through the steps I took to create the lab, install the required tools, and run some basic attack and defense simulations.

Prerequisites

Before starting, I made sure I had the following in place:

1. Basic networking and virtualization knowledge: Familiarity with IP addressing, subnetting, and how virtual machines (VMs) operate was crucial.

2. A computer with at least 8GB of RAM: This was necessary to run two VMs simultaneously without compromising performance.

3. VirtualBox: I used VirtualBox as my virtualization platform, and I found it to be simple to install and configure.

Step 1: Installing VirtualBox

The first step was to install VirtualBox. I downloaded the installation package directly from the [VirtualBox official website] (https://www.virtualbox.org/). Since I'm using a Windows-based host machine, I followed the windows installation instructions.

Step 2: Creating Virtual Machines (VMs)

Once VirtualBox was set up, I created two virtual machines:

Ubuntu VM:

I downloaded the latest version of Ubuntu from the [Ubuntu Downloads] (https://ubuntu.com/download) page.

Kali Linux VM:

For offensive security tasks, I downloaded Kali Linux from the [Kali Downloads](https://www.kali.org/get-kali/) page.

Step 3: Configuring the Network

To enable communication between the two VMs, I set up a **NAT Network** in VirtualBox. This allowed the VMs to communicate privately with each other while isolating them from my host machine's network for security purposes. Here's how I configured it:

- In VirtualBox, I navigated to File > Preferences > Network and created a new NAT network.

- Both the Ubuntu and Kali Linux VMs were then connected to this network by selecting "NAT Network" as the network adapter type in their respective network settings.

Step 4: Setting Up the VMs

With both VMs installed and connected, I moved on to configuring them for the tasks ahead.

- On the Ubuntu VM:

1. I updated the system by running:

```
sudo apt update && sudo apt upgrade -y
```

2. I installed tools like UFW (the firewall) and Wireshark (for network traffic monitoring):

```
sudo apt install -y ufw wireshark
```

3. To secure the VM, I enabled UFW and allowed only the necessary services (SSH and traffic from the NAT network):

```
sudo ufw enable

sudo ufw allow ssh

sudo ufw allow from 10.0.2.0/24
```

- On the Kali Linux VM:

1. I updated the system as well:

```
sudo apt update && sudo apt upgrade
```

2. I installed nmap, a powerful network scanner used for offensive security tasks:

```
sudo apt install -y nmap
```

Step 5: Running the Simulations

Step 5.1: Network Scanning with Kali Linux

Once the Kali Linux VM was set up, I used nmap to scan the Ubuntu VM to discover open ports and services that could potentially be vulnerable. I first found the IP address of the Ubuntu VM (assigned by the NAT network) and ran:

```
nmap 10.0.2.15
```

This scan revealed which ports and services were open on the Ubuntu VM.

Step 5.2: Defending Ubuntu with UFW

To defend against network scans, I configured UFW on the Ubuntu VM to block certain traffic. I added firewall rules to block nmap scan attempts:

```
sudo ufw enable

sudo ufw allow ssh

sudo ufw allow from 10.0.2.0/24
```

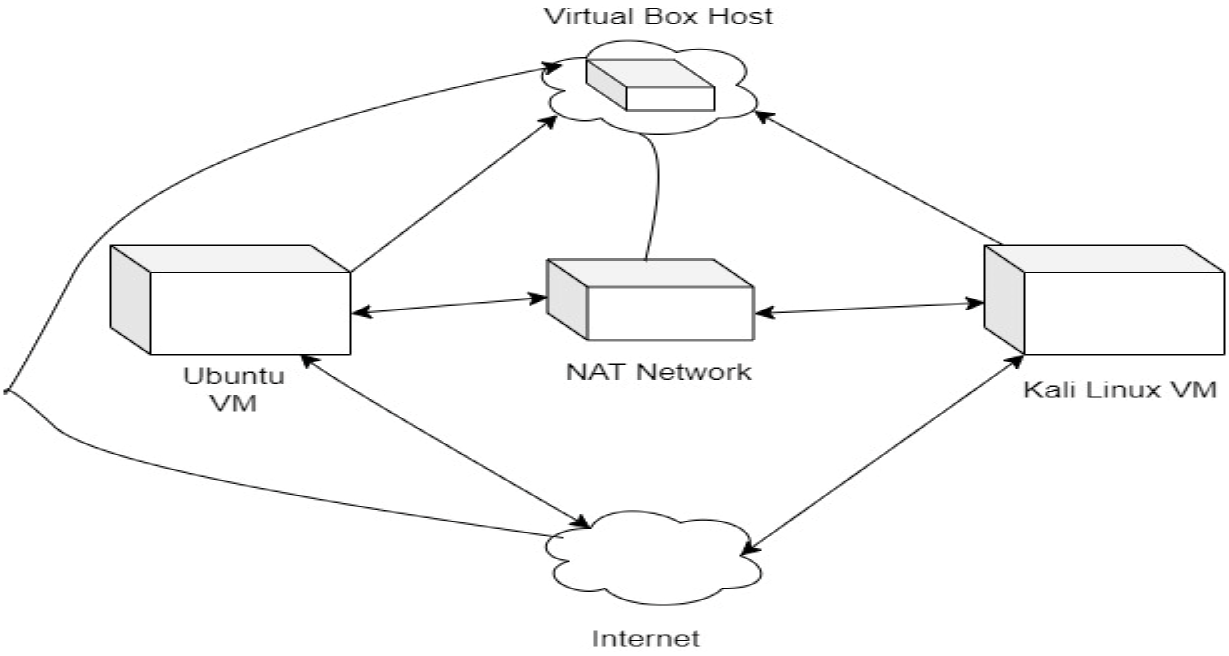
This allowed me to simulate real-world defenses by preventing unauthorized network scans.

Step 5.3: Traffic Monitoring with Wireshark

Finally, I used Wireshark on the Ubuntu VM to monitor incoming traffic and analyze any suspicious activity. By capturing packets, I could visualize the attacks in real-time and see how the Ubuntu VM responded to the scan attempts.

Network Diagram

Below is a simplified diagram representing the setup of my home lab:



Components:

- Host Machine: My physical computer is running VirtualBox.

- VirtualBox: The platform used to manage the VMs.

- NAT Network: The private network that enables the VMs to communicate with each other.

- Ubuntu VM: A virtual machine used to simulate defensive measures.

- Kali Linux VM: A virtual machine used to simulate offensive security techniques, such as network scanning and attacks.

Conclusion

Setting up this home lab gave me invaluable hands-on experience with cybersecurity fundamentals, including network scanning, firewall configuration, and traffic monitoring. Isolating the VMs within a NAT network ensured that I could run experiments safely without affecting my host machine or local network. This lab setup serves as a robust foundation for learning and practicing more advanced cybersecurity concepts.