My Home Lab Setup for Cybersecurity Practice

Introduction

As someone keen to develop my cybersecurity skills, I decided to set up a home lab where I could safely experiment with network attacks and defenses. After some research, I opted for **VirtualBox** to create a controlled environment with two virtual machines (VMs): **Ubuntu** for defensive measures and **Kali Linux** for offensive security testing. By connecting these machines over a **NAT network**, I could simulate real-world cyberattacks and practice defending against them.

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

Prerequisites

Before I got started, there were a few things I needed:

- **Basic networking and virtualization knowledge**: Familiarity with IP addresses, subnetting, and how VMs work was helpful.
- **A computer with at least 8GB of RAM**: I wanted to run two VMs smoothly without performance issues.
- **VirtualBox**: I used VirtualBox to manage the virtual machines, which I found easy to set up.

Step-by-Step Setup Guide

Step 1: Installing VirtualBox

The first thing I did was install VirtualBox. I grabbed the installation package from the official VirtualBox website and followed the instructions for my operating system.

Step 2: Creating the Virtual Machines (VMs)

Once VirtualBox was up and running, I moved on to creating my two VMs:

- **Ubuntu VM**:
    - I downloaded the latest version of Ubuntu from Ubuntu Downloads and set it up on VirtualBox.
    - I configured the VM with at least 4GB of RAM and created a virtual hard drive of about 20GB.

- **Kali Linux VM**:
    - For offensive security tasks, I used Kali Linux, which I downloaded from Kali Downloads.
    - Just like the Ubuntu VM, I gave this VM 4GB of RAM and enough disk space for my tools and future experiments.

Step 3: Configuring the Network

To get the VMs communicating with each other, I created a **NAT Network** in VirtualBox. This way, the VMs could interact privately, separate from my host machine's network. Setting this up was straightforward:

- In VirtualBox, I went to File > Preferences > Network and created a new NAT network.
- Both the Ubuntu and Kali VMs were then connected to this network under their **Network Settings** by selecting "NAT Network" as the adapter type.

Step 4: Setting Up the VMs

After I got both VMs installed, I needed to configure them for the tasks ahead.

On Ubuntu:

- I started by updating the system:

Code : sudo apt update && sudo apt upgrade -y

- Then, I installed some essential tools like **UFW** (the firewall) and **Wireshark** (for traffic monitoring):

Code : sudo apt install -y ufw wireshark

- I enabled the **UFW firewall** to block unwanted connections and allowed SSH and communication from the NAT network

Code : sudo ufw enable

sudo ufw allow ssh

sudo ufw allow from 10.0.2.0/24 (Your IP Address)

On Kali Linux:

- I did a system update first:

Code : sudo apt update && sudo apt upgrade -y

- Next, I installed **nmap** for network scanning, which would be my main tool for attacking the Ubuntu VM:

Code : sudo apt install -y nmap

Step 5: Running the Simulations

With both VMs configured, it was time to run some basic tests.

Step 5.1: Network Scanning with Kali Linux

From the Kali Linux VM, I used **nmap** to scan the Ubuntu VM. The idea was to see which ports and services were open and potentially vulnerable. I found the IP address of the Ubuntu VM (assigned by the NAT network) and ran the following command:

Code : nmap 10.0.2.15 (Replace with the actual IP of the Ubuntu VM)

This gave me a list of open ports and services running on the Ubuntu VM.

Step 5.2: Traffic Monitoring with Wireshark

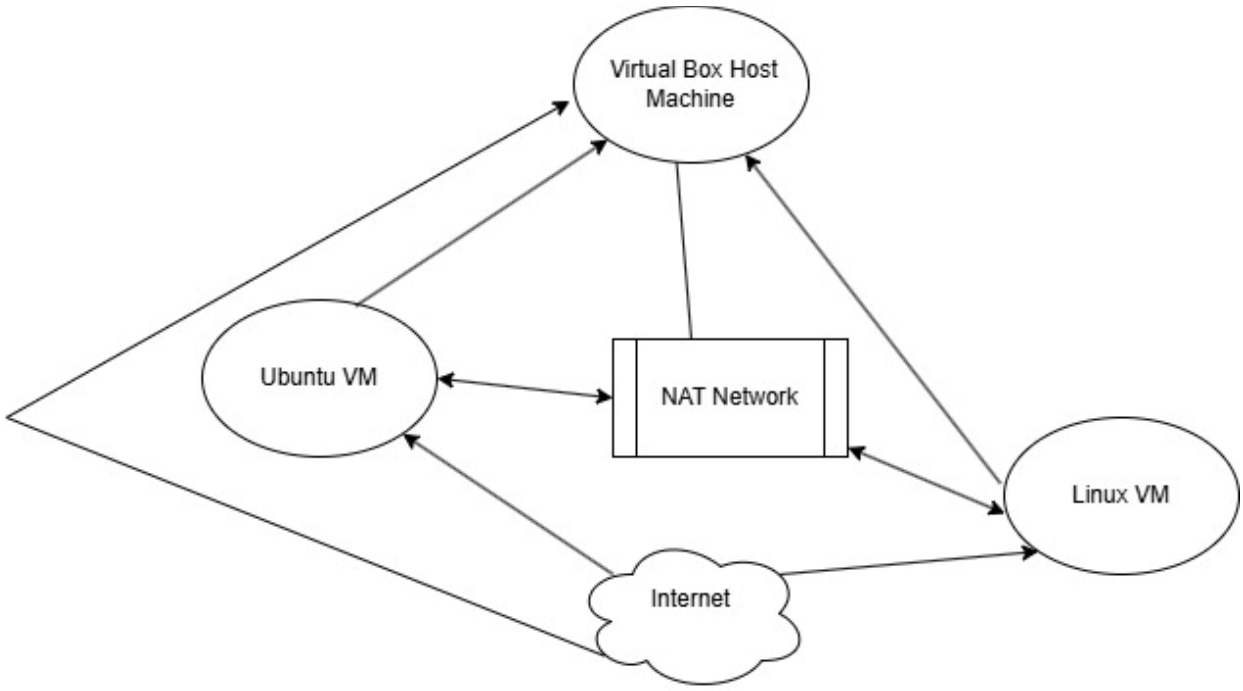
Finally, I used **Wireshark** on the Ubuntu VM to monitor incoming traffic. I set it up to capture packets and analyze them for any unusual or suspicious activity. This was a great way to visualize the network traffic and see the attacks happening in real time.

Code : sudo apt install wireshark

sudo wireshark

Network Diagram

Below is a simplified diagram that represents my home lab setup:



Components:

- **Host Machine**: This is my physical computer running VirtualBox.
- **VirtualBox**: The platform where I manage and run the VMs.
- **NAT Network**: The isolated network that allows the VMs to communicate.
- **Ubuntu VM**: This VM is used to simulate a defensive system.
- **Kali Linux VM**: This VM is used to simulate offensive cybersecurity techniques, such as scanning and attacks.

Conclusion

Setting up this home lab gave me hands-on experience with basic cybersecurity concepts like network scanning, firewall configuration, and traffic monitoring. By isolating the VMs in a NAT network, I could safely simulate real-world scenarios without affecting my host machine or local network.