Brooklyn Dressel
Assignment 5

Abstract

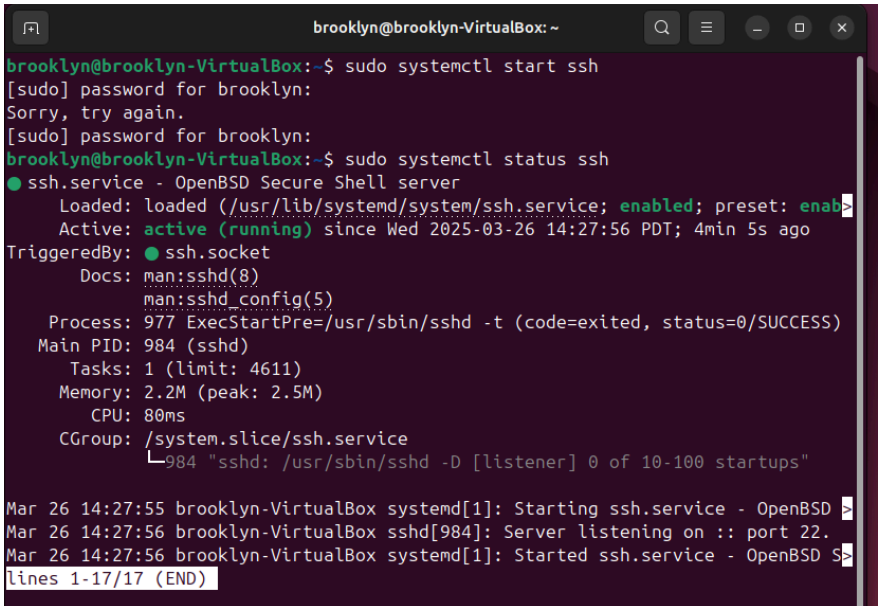
This assignment demonstrates the capabilities of tools such as Nmap, Zenmap, and Wireshark, to scan and analyze network configurations as well as perform security audits by capturing packets sent between a Kali Linux virtual machine and an Ubuntu virtual machine. After enabling network services on Ubuntu, scanning and packet analysis were conducted to explore how these tools operate, collect data, and reveal network service details. The activity demonstrates the availability of network information available to attackers as well as helps to locate the insecurities of the Ubuntu machine.

Introduction

For this assignment inside an Ubuntu virtual machine, python is used to locally host a webserver, SSH is used to host a server, and a NetCat listener is set up, all to test the network connectivity from Kali. Additionally, inside a Kali Linux virtual machine Firefox is used to connect to the Ubuntu web server, SSH is used to connect to the Ubuntu SSH server, and NetCat was used to connect to the NetCat listener on the Ubuntu machine. NMAP was used to complete four different scans of the Ubuntu system. WireShark was used to capture all of these packets. Finally, Zenmap was used to complete three different scans of the Ubuntu machine. The two virtual machines were run on a Bridged Network. Additionally, all commands used during this assignment are attached in a separate text file, *Assign5 - Commands Used*.

Summary of Results

The first step of this project was to start up my Ubuntu virtual machine, making sure that it was set up to be on a Bridged Network. This would allow it to access the internet as well as communicate with any other virtual machines run on my host computer, as it would have its own unique ip address. So, once Ubuntu was running I ran `command 1` in a terminal in order to start an SSH server. To ensure this had started properly I ran `command 2` and the response told me the server was active and listening on port 22.

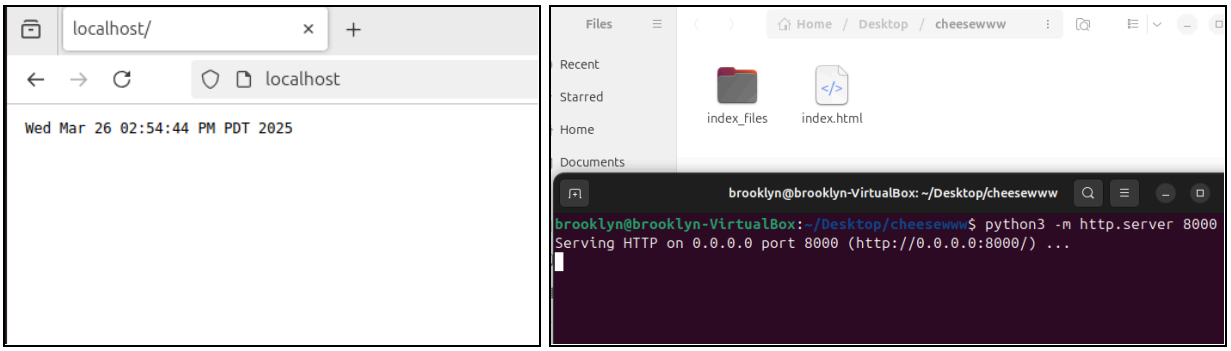


Next I ran `command 3` to find the IP address of my Ubuntu machine, with this information I was able to quickly open a NetCat listener using `command 4`. However, in order to keep this listener going in the background I ran `command 5` and `command 6` to create a looping NetCat listener that would run in the background so I could continue to do other things in the terminal.

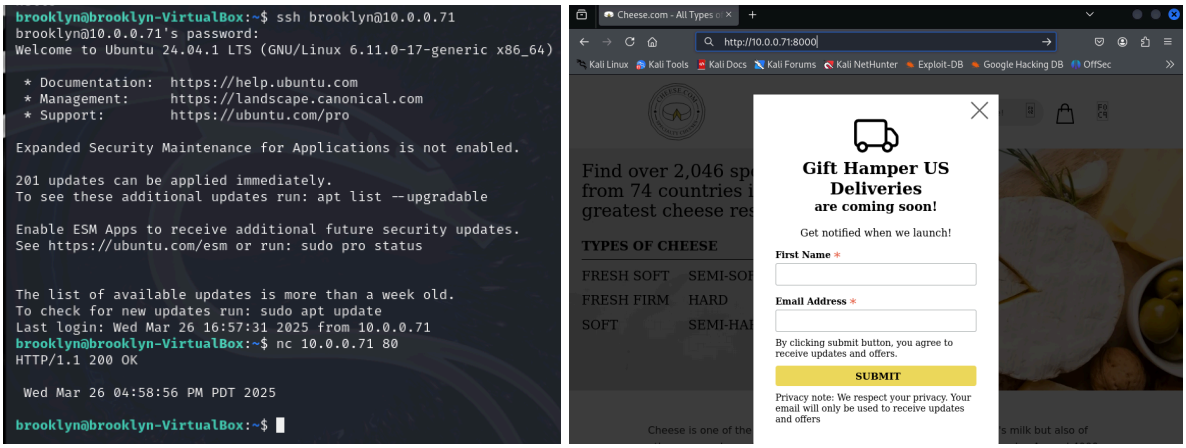
```
brooklyn@brooklyn-VirtualBox:~$ nc -l 10.0.0.71 -p 31337
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
```

```
brooklyn@brooklyn-VirtualBox:~$ sudo bash
[sudo] password for brooklyn:
root@brooklyn-VirtualBox:/home/brooklyn# while true; do echo -e "HTTP/1.1 200 OK
\n\n $(date)" | nc -l -p 80 -q 1; done
```

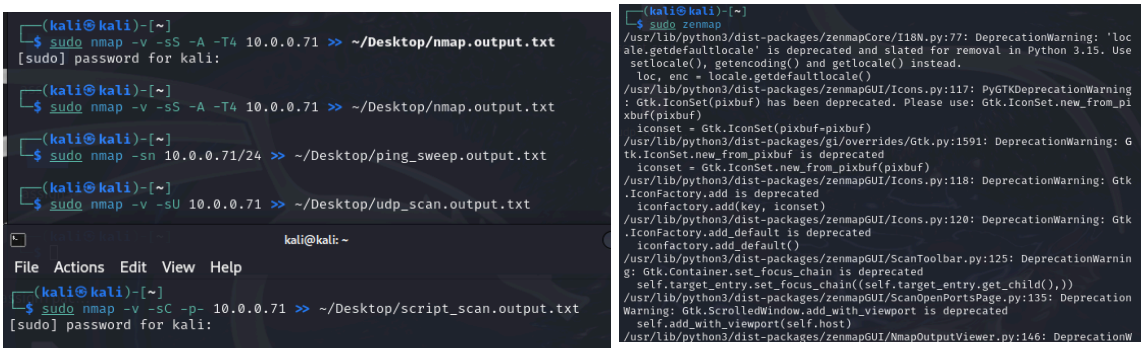
The final service to set up in Ubuntu was a webserver. To do this I opened Firefox and navigated to `cheese.com` once on this page I saved the page to a folder on my desktop. After opening a terminal inside the folder I ran `command 7` to start locally hosting the website. The leftmost screenshot below shows when no port is specified, it defaults to port 80, the while loop used below. The rightmost screenshot below shows the output of `command 7`.



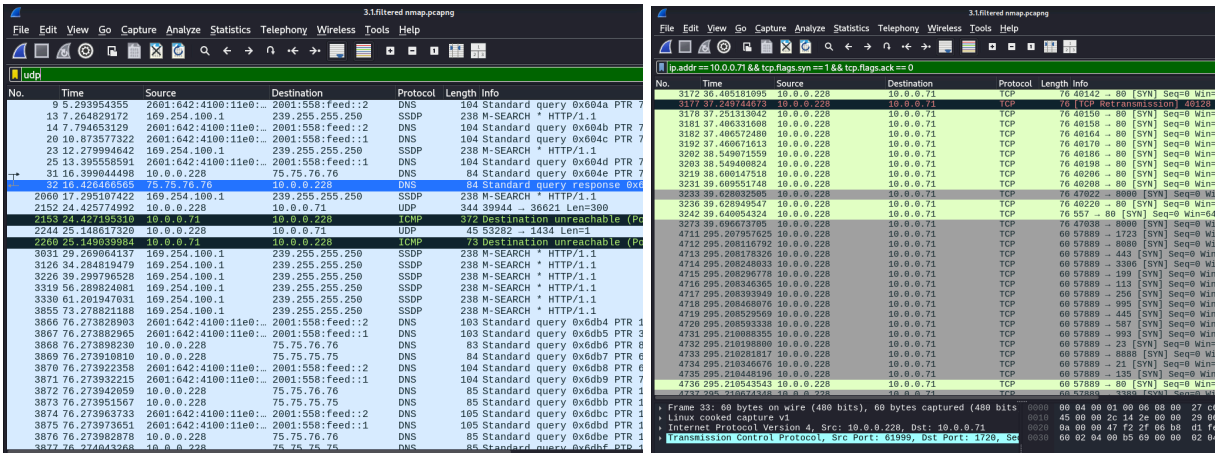
Now that all services were up and running on the Ubuntu virtual machine, I opened Kali and began the process to discover these services through various scanning attempts. The first task once inside Kali was to update it and ensure that Zenmap was downloaded, this was accomplished using `command 8` and `command 9`. Now that this was out of the way, I started WireShark and ran `command 10` to connect to the Ubuntu SSH server, `command 11` to connect to the NetCat listener on the Ubuntu machine, as well as navigated to the Ubuntu web server at `http://10.0.0.71:8000`. After this I stopped the packet capture in WireShark. All of these packets showed a 'normal' way two different machines communicate.



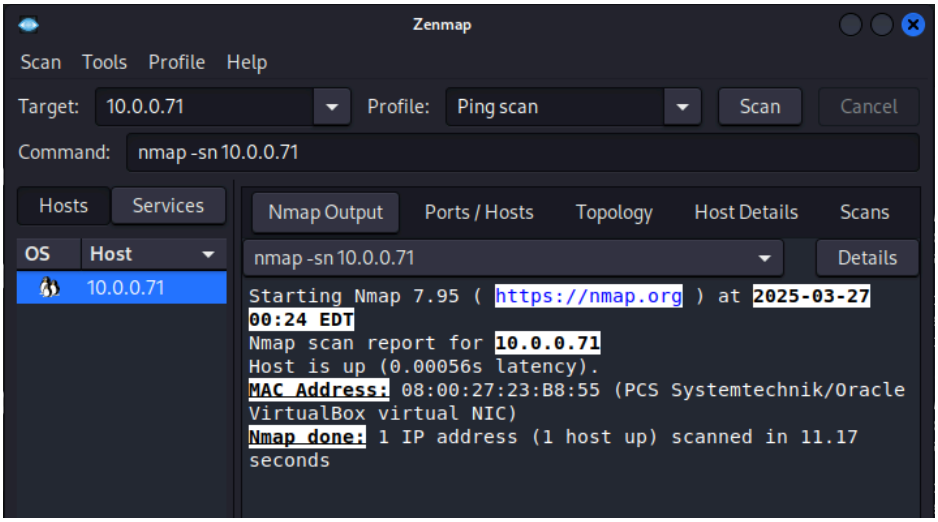
Next I start up WireShark again and start scanning Ubuntu from Kali. The first scan I ran was `command 12`. Breaking this down, "-sS" meaning its sending SYN packets to determine the TCP port states (open/closed), while "-A" enables the OS and version detection to identify more information about the target system. Next, I did a ping sweep by running `command 13`, this determined all active devices on the network (including the subnet "/24"). I continued to scan for UDP ports and then scan for vulnerabilities across *all* ports by running `commands 14` and `15`, respectively. All of the results of these scans are saved as text files on the Kali system's desktop. The names of these files are specified in the command line.



The screenshots below are of the packets captured by WireShark during these NMAP scans. I filtered these packets two separate ways, the first, shown on the left, by UDP, these packets were sent during the third scan. The second way, shown on the right, filtered the TCP data, specified the Ubuntu's IP address as well as by SYN and ACK flags. This specific filter is meant to find the initial moment of establishing a TCP connection between Kali and a specific port on the server (the Ubuntu machine). Analysis of these packet captures will be discussed in the conclusion.



Finally, I used `command 16` to open Zenmap. This tool is essentially the same as NMAP but with a GUI. So I ran three more scans, intense, to find information about running services, operating systems, and network paths. Next, intense + UDP, to discover services running on both TCP and UDP ports. And finally, a ping scan which identifies devices, or hosts, on the network.



Conclusion

Network mapping, using NMAP and Zenmap can provide much information about a target's network. Including finding open ports, running services, as well as connected devices or hosts. The mapping tools are able to do this by performing scans, which send various packets to the target in order to determine information about it, either by the response given by the target, or the lack of response.

Some of the main differences between the packets generated from the scans and the 'normal' packets captured first is that there are so many more packets generated by the scans. It is very apparent that numerous packets are both generated and sent much faster than during a 'normal' connection. However, these scan packets have very little data involved. This is because the scan is attempting to find a variety of information all at once, as opposed to trying to connect to a single service in the 'normal' scan. For this same reason, many of the packets from the scans have random, high-numbered source ports as well as a wide range of destination ports, as the scans attempt to probe the entirety of the target network.

Overall, network mapping tools can be used as a means to detect weaknesses in a network either as a means to improve a weak network or to prepare for an attack on the target network. While this network mapping alone is not considered an attack, the information can be very helpful to an attacker deciding how to do something malicious to the target network. However, it can also be used by the target network to learn where they are weak and know how and where to improve their security.

A good way to prevent network mapping attacks would be to control and monitor unauthorized network scans. For example firewalls could identify unusual scanning activity such as the rapid port sweeps and block the attacker.