Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE3501: INFORMATION SECURITY ANALYSIS AND AUDIT

FALL SEMESTER 21-22

CHANDRA MOHAN B

J COMPONENT
FINAL REVIEW DOCUMENTATION

VULNERABILITY SCANNING USING NMAP

TEAM MEMBERS:

Nikhitha Perapola - 19BDS0125

Meghana Dirisala - 19BDS0100

Mutyam Sai Swithika - 19BCE2334

Table of Components

ABSTRACT

With the increasing concern for security in the network, many ways are found out to protect the network from unauthorized access. New methods have been adopted to find the potential discrepancies that may damage the network. The most commonly used approach is the vulnerability assessment. By vulnerability, we mean, the potential flaws in the system that make it prone to the attack. Network administrators, IT managers, and security professionals face a never-ending battle, constantly checking on what exactly is running on their networks and the vulnerabilities that lurk within.

A common issue with internet systems is that they are too complicated for the ordinary person to understand. Even a small home-based system is extremely complex. When it comes to larger companies and agencies that deal with hundreds or even thousands of computers on the network, that complexity grows exponentially. This can be dealt with inefficiently using Nmap. Hence, we have decided to take up NMap for our project for vulnerability scanning and enumeration.

In our project, we've tried to work on: Open Port Identification, Service Name and Version Detection, Host Discovery, OS Detection, Vulnerability Detection & Exploitation using scans, Sniffing a network, and Enumeration of a vulnerable virtual machine using NMAP.

INTRODUCTION

Nmap, short for Network Mapper, is a free, open-source tool for vulnerability scanning and network discovery. Nmap can work on Linux, Unix, BSDs, MacOS X, and Windows.

Network administrators use Nmap to identify what devices are running on their systems, discover hosts that are available and the services they offer, find open ports and detect security risks.

It's a port-scan tool, gathering information by sending raw packets to system ports. It listens for responses and determines whether ports are open, closed, or filtered in some way by, for example, a firewall. Other terms used for port scanning include port discovery or enumeration. During a scan, Nmap sends specially crafted packets to the target host and then analyzes the responses.

Port Status: After scanning, you may see some results with a port status like filtered, open, closed, etc. Let me explain this.

- *Open*: This indicates that an application is listening for connections on this port.
- *Closed*: This indicates that the probes were received but there is no application listening on this port.
- Filtered: This indicates that the probes were not received and the state could not be established. It also indicates that the probes are being dropped by some kind of filtering.
- Unfiltered: This indicates that the probes were received but a state could not be established.
- *Open/Filtered:* This indicates that the port was filtered or open but Nmap couldn't establish the state.

- *Closed/Filtered:* This indicates that the port was filtered or closed but Nmap couldn’t establish the state.

We chose to work on our project on Ubuntu Linux and Kali Linux Virtual machines.

Nmap works by sending IP packets to identify the hosts and services on a computer network and then analyzes the responses to provide information on each host and service, as well as the operating systems that they are running. Nmap reads and interprets the response that comes back and uses the information to create a map of the network.

The map that is created includes detailed information on what each port is doing and who (or what) is using it, how the hosts are connected, what is and is not making it through the firewall, and listing any security issues that come up. All of this is accomplished by utilizing a complex system of scripts that communicate with every part of the network.

The scripts act as communication tools between the network components and their human users. Available hosts scripts that Nmap uses are capable of vulnerability detection, backdoor detection, vulnerability exploitation, and network discovery.

SOFTWARE REQUIREMENTS

KALI LINUX:

Kali Linux is an open-source, Debian-based Linux system developed for information security initiatives involving penetration testing, security research, computer forensics, & reverse engineering.

You can take any Linux and install pen testing tools on it, but you have to set the tools up manually and configure them. Kali is optimized to reduce the amount of work, so a professional can just sit down and go.

The Kali Linux penetration testing platform contains a vast array of tools and utilities. From information gathering to final reporting, Kali Linux enables security and IT professionals to assess the security of their systems.

UBUNTU:

Ubuntu is a comprehensive Linux operating system that is free to use and has not only community but professional support as well. The Ubuntu community is based on the ideas enshrined in the Ubuntu Manifesto: that software should be available for free, that software tools should be usable by people regardless of disability and in their native language, and that people should have the freedom to customize and alter their software in any way they view fit.

NMAP:

Nmap is a fantastic tool for detecting open ports, protocol numbers, Operating System data, firewall details, and so on. Nmap (Network Mapper) is an open-source network exploration and security auditing tool.

Nmap employs new techniques to:

➔ Detect what hosts are available on the network, what services (application name and version) those hosts can provide.
➔ what type of operating systems and their particular versions are running on the system

- ➔ what sort of packet filters/firewalls they are using, and thousands of more details about them.
- ➔ It was developed to scan big networks quickly, although it also works well against single hosts.

Nmap is compatible with the majority of computer operating systems, and official binary packages for Linux, Windows, and Mac OS X are available.

WIRESHARK:

Wireshark is a software application that examines network traffic passing across a network interface. It is currently the most extensively used network monitoring tool. System administrators, network engineers, network hobbyists, network security specialists, and black hat hackers all use Wireshark.

PROCESS

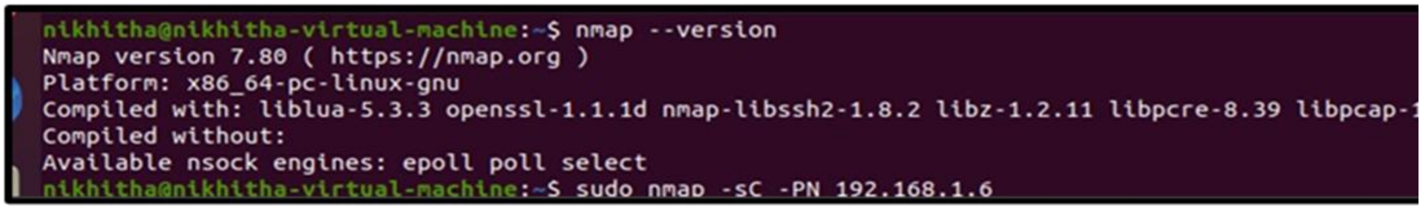
NMAP is a very helpful tool for vulnerability scanning and numerous port scans to understand our own network better. Our paper shows a variety of post scans. This will help her understand our network to determine the host available on the network and what services those hosts are offering and make various conclusions about the network from the port scans. We have also shown how to find out the operating system or systems running and their OS version. Nmap is designed for rapid scans on large networks but works fine against a single host. From this paper, we showed various port scans, OS detection, and also Enumeration and Usage of SSH to connect to a local vulnerable network by finding out its credentials using secure shell and Wireshark.

RESULTS AND DISCUSSION

The first step is to install NMAP in the Linux server if not installed.

NMAP Version installed:

- NMAP Version installed in the Ubuntu system is Nmap version 7.80



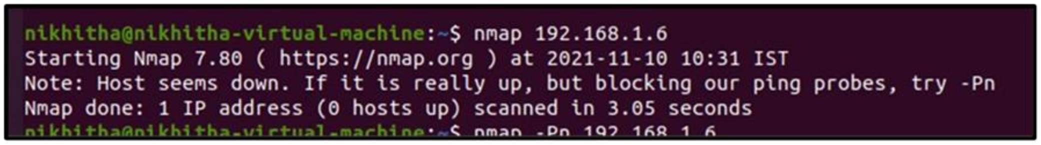
```
nikhitha@nikhitha-virtual-machine:~$ nmap --version
Nmap version 7.80 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.3 openssl-1.1.1d nmap-libssh2-1.8.2 libz-1.2.11 libpcre-8.39 libpcap-1
Compiled without:
Available nsock engines: epoll poll select
nikhitha@nikhitha-virtual-machine:~$ sudo nmap -sC -PN 192.168.1.6
```

Ping only scan:

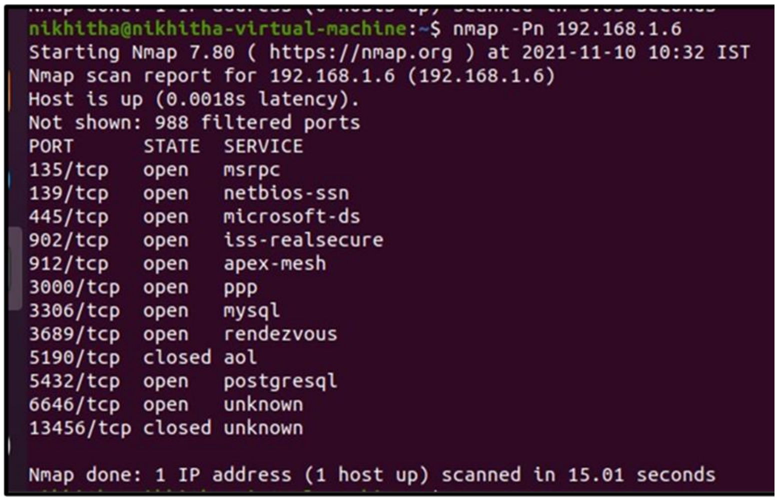
The **-Sp** option is responsible for a ping-only scan.

It sends a UDP packet to the given ports. For most ports, the packet will be empty, though some use a protocol-specific payload that is more likely to elicit a response.

Syntax: nmap –Sp target



```
nikhitha@nikhitha-virtual-machine:~$ nmap 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:31 IST
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 3.05 seconds
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn 192.168.1.6
```

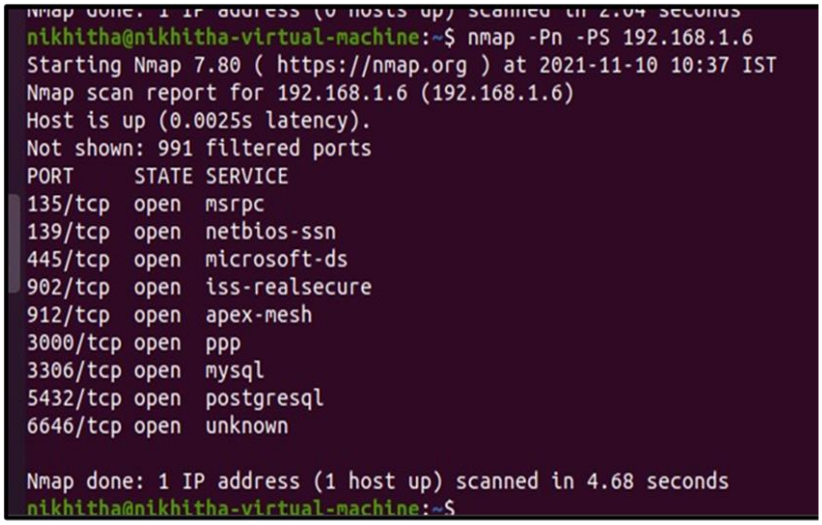


```
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:32 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0018s latency).
Not shown: 988 filtered ports
PORT      STATE  SERVICE
135/tcp   open   msrpc
139/tcp   open   netbios-ssn
445/tcp   open   microsoft-ds
902/tcp   open   iss-realsecure
912/tcp   open   apex-mesh
3000/tcp  open   ppp
3306/tcp  open   mysql
3689/tcp  open   rendezvous
5190/tcp  closed aol
5432/tcp  open   postgresql
6646/tcp  open   unknown
13456/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 15.01 seconds
```

TCP syn scan:

The TCP SYN ping sends a SYN packet to the target system and listens for a response. This alternative discovery method is useful for systems that are configured to block standard ICMP pings.

Syntax: nmap –PS targets



```
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn -PS 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:37 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0025s latency).
Not shown: 991 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
3000/tcp open  ppp
3306/tcp open  mysql
5432/tcp open  postgresql
6646/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds
nikhitha@nikhitha-virtual-machine:~$
```

TCP ack ping scan

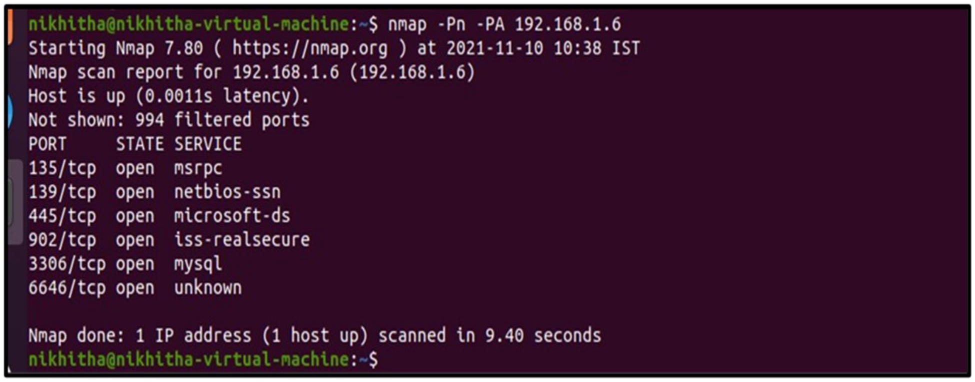
This type of scan will only scan of Acknowledgement(ACK) packet.

The **-PA** performs a TCP ACK ping on the specified target.

The **-PA** option causes Nmap to send TCP ACK packets to the specified hosts.

Syntax: nmap –PA target

This method attempts to discover hosts by responding to TCP connections that are nonexistent in an attempt to solicit a response from the target. Like other ping options, it is useful in situations where standard ICMP pings are blocked.



```
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn -PA 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:38 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0011s latency).
Not shown: 994 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
3306/tcp open  mysql
6646/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 9.40 seconds
nikhitha@nikhitha-virtual-machine:~$
```

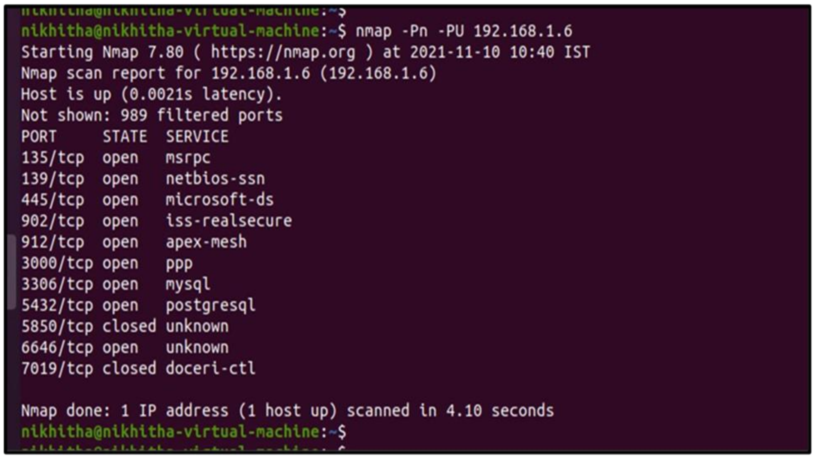
UDP ping scan

The **–PU** scan only on udp ping scans on the target. This type of scan sends udp packets to get a response.

For most ports, the packet will be empty, though some use a protocol-specific payload that is more likely to elicit a response.

The primary advantage of this scan type is that it bypasses firewalls and filters that only screen TCP.

Syntax: nmap –PU target



```
nikhitha@nikhitha-virtual-machine:~$
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn -PU 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:40 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0021s latency).
Not shown: 989 filtered ports
PORT     STATE  SERVICE
135/tcp  open   msrpc
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
902/tcp  open   iss-realsecure
912/tcp  open   apex-mesh
3000/tcp open   ppp
3306/tcp open   mysql
5432/tcp open   postgresql
5850/tcp closed unknown
6646/tcp open   unknown
7019/tcp closed doceri-ctl

Nmap done: 1 IP address (1 host up) scanned in 4.10 seconds
nikhitha@nikhitha-virtual-machine:~$
```

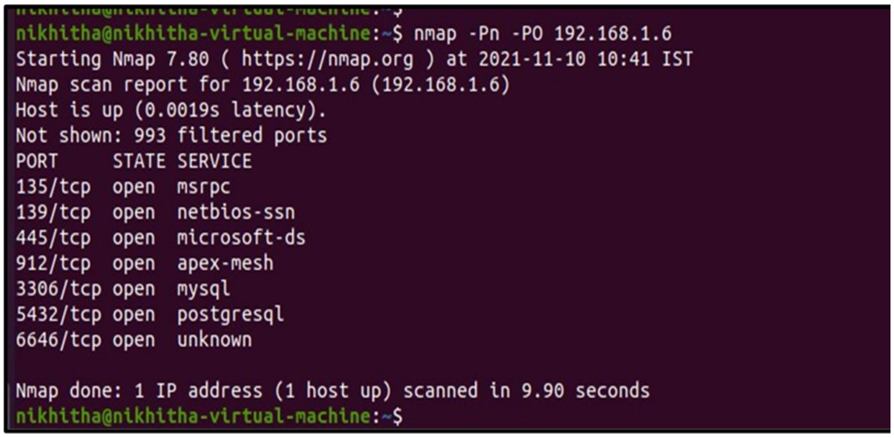
IP protocol ping

The **-PO** option performs an IP protocol ping.

-PO sends IP packets with the specified protocol number set in their IP header. The protocol list takes the same format as do port lists.

This host discovery method looks for either response using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the given protocol isn't supported on the destination host. Either type of response signifies that the target host is alive.

Syntax: nmap –PO protocol target

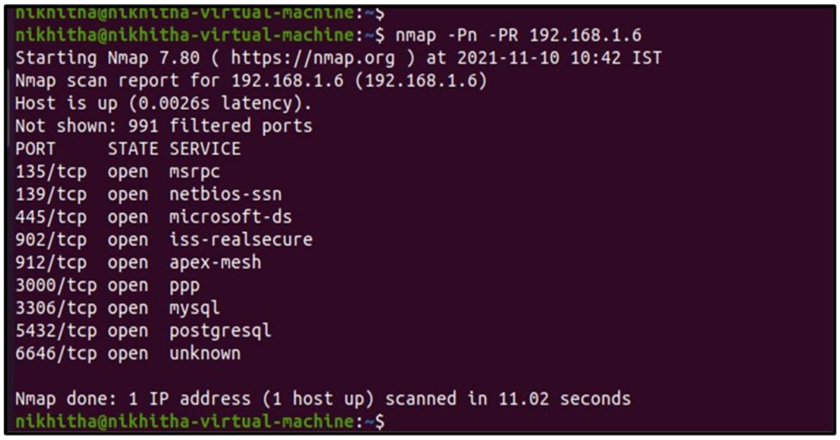


```
nikhitha@nikhitha-virtual-machine:~$
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn -PO 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:41 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0019s latency).
Not shown: 993 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
912/tcp  open  apex-mesh
3306/tcp open  mysql
5432/tcp open  postgresql
6646/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 9.90 seconds
nikhitha@nikhitha-virtual-machine:~$
```

ARP ping

The **–PR** option is used to perform an arp ping scan. The **-PR** option instructs Nmap to perform an ARP (Address Resolution Protocol) ping on the specified target.

Syntax: nmap –PR target

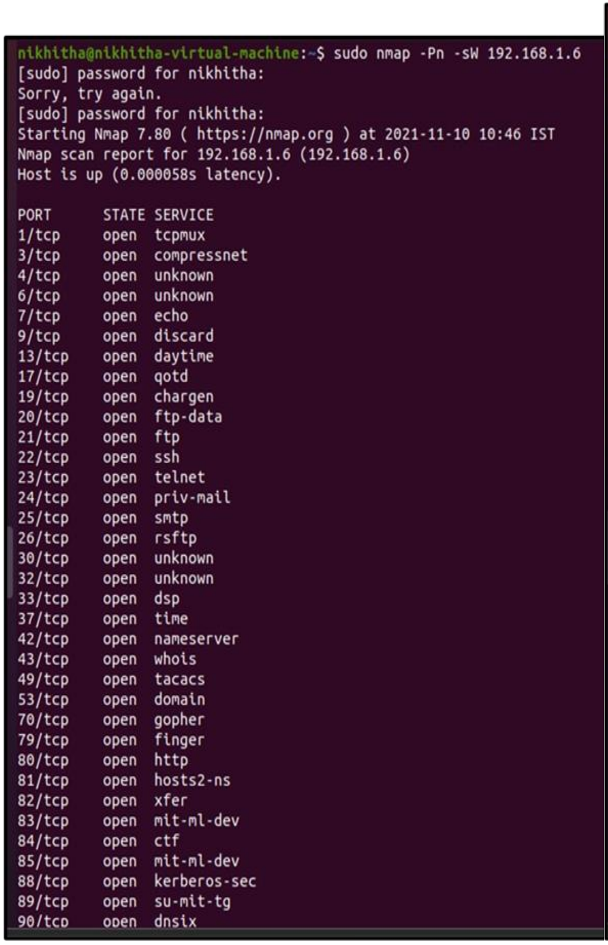


```
nikhitha@nikhitha-virtual-machine:~$
nikhitha@nikhitha-virtual-machine:~$ nmap -Pn -PR 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:42 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0026s latency).
Not shown: 991 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
3000/tcp open  ppp
3306/tcp open  mysql
5432/tcp open  postgresql
6646/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 11.02 seconds
nikhitha@nikhitha-virtual-machine:~$
```

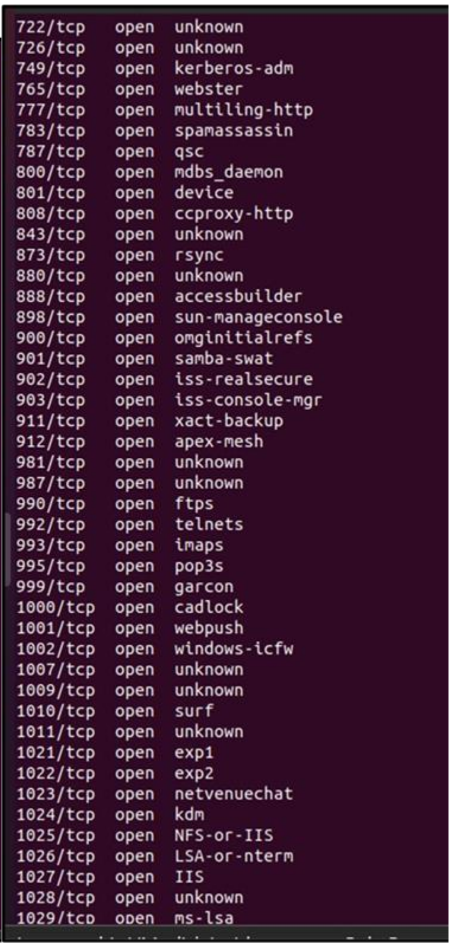
TCP window scan

Window scan is exactly the same as ACK scan, except that it exploits an implementation detail of certain systems to differentiate open ports from closed ones, rather than always printing unfiltered when an RST is returned.

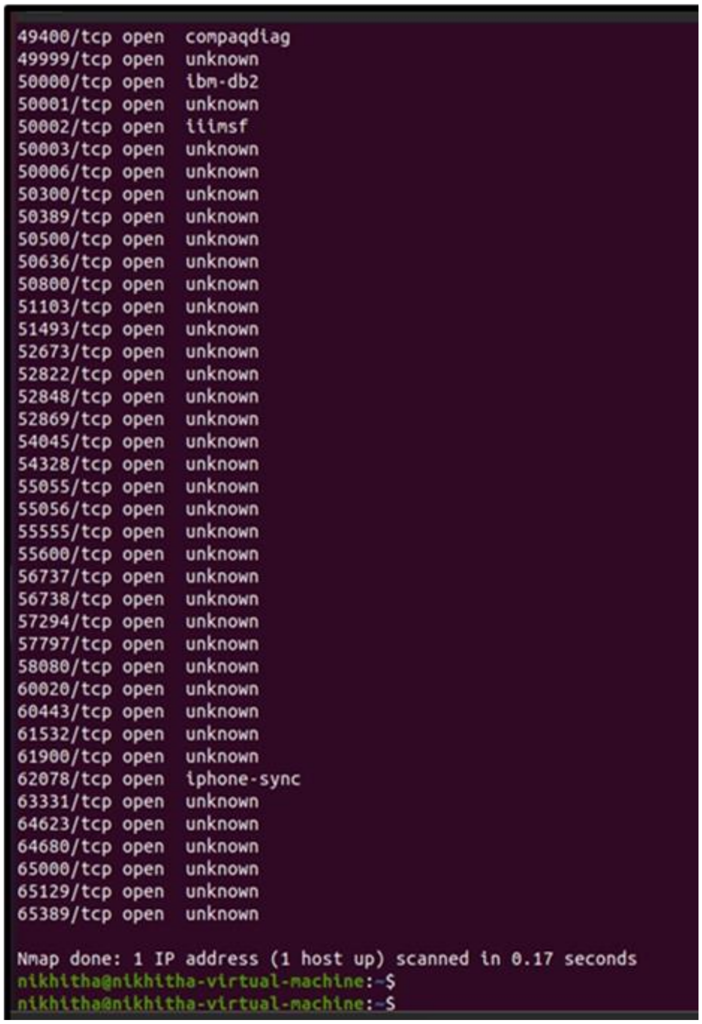
Command: nmap –sW target



```
nikhitha@nikhitha-virtual-machine:~$ sudo nmap -Pn -sW 192.168.1.6
[sudo] password for nikhitha:
Sorry, try again.
[sudo] password for nikhitha:
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 10:46 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.000058s latency).

PORT      STATE SERVICE
1/tcp     open  tcpmux
3/tcp     open  compressnet
4/tcp     open  unknown
6/tcp     open  unknown
7/tcp     open  echo
9/tcp     open  discard
13/tcp    open  daytime
17/tcp    open  qotd
19/tcp    open  chargen
20/tcp    open  ftp-data
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
24/tcp    open  priv-mail
25/tcp    open  smtp
26/tcp    open  rsftp
30/tcp    open  unknown
32/tcp    open  unknown
33/tcp    open  dsp
37/tcp    open  time
42/tcp    open  nameserver
43/tcp    open  whois
49/tcp    open  tacacs
53/tcp    open  domain
70/tcp    open  gopher
79/tcp    open  finger
80/tcp    open  http
81/tcp    open  hosts2-ns
82/tcp    open  xfer
83/tcp    open  mit-ml-dev
84/tcp    open  ctf
85/tcp    open  mit-ml-dev
88/tcp    open  kerberos-sec
89/tcp    open  su-mit-tg
90/tcp    open  dnsix
```



```
722/tcp   open  unknown
726/tcp   open  unknown
749/tcp   open  kerberos-adm
765/tcp   open  webster
777/tcp   open  multiling-http
783/tcp   open  spamassassin
787/tcp   open  qsc
800/tcp   open  mdbs_daemon
801/tcp   open  device
808/tcp   open  ccproxy-http
843/tcp   open  unknown
873/tcp   open  rsync
880/tcp   open  unknown
888/tcp   open  accessbuilder
898/tcp   open  sun-manageconsole
900/tcp   open  omginitialrefs
901/tcp   open  samba-swat
902/tcp   open  iss-realsecure
903/tcp   open  iss-console-mgr
911/tcp   open  xact-backup
912/tcp   open  apex-mesh
981/tcp   open  unknown
987/tcp   open  unknown
990/tcp   open  ftps
992/tcp   open  telnets
993/tcp   open  imaps
995/tcp   open  pop3s
999/tcp   open  garcon
1000/tcp  open  cadlock
1001/tcp  open  webpush
1002/tcp  open  windows-icfw
1007/tcp  open  unknown
1009/tcp  open  unknown
1010/tcp  open  surf
1011/tcp  open  unknown
1021/tcp  open  exp1
1022/tcp  open  exp2
1023/tcp  open  netvenuechat
1024/tcp  open  kdm
1025/tcp  open  NFS-or-IIS
1026/tcp  open  LSA-or-nterm
1027/tcp  open  IIS
1028/tcp  open  unknown
1029/tcp  open  ms-lsa
```



```
49400/tcp open  compaqdiag
49999/tcp open  unknown
50000/tcp open  ibm-db2
50001/tcp open  unknown
50002/tcp open  iiimsf
50003/tcp open  unknown
50006/tcp open  unknown
50300/tcp open  unknown
50389/tcp open  unknown
50500/tcp open  unknown
50636/tcp open  unknown
50800/tcp open  unknown
51103/tcp open  unknown
51493/tcp open  unknown
52673/tcp open  unknown
52822/tcp open  unknown
52848/tcp open  unknown
52869/tcp open  unknown
54045/tcp open  unknown
54328/tcp open  unknown
55055/tcp open  unknown
55056/tcp open  unknown
55555/tcp open  unknown
55600/tcp open  unknown
56737/tcp open  unknown
56738/tcp open  unknown
57294/tcp open  unknown
57797/tcp open  unknown
58080/tcp open  unknown
60020/tcp open  unknown
60443/tcp open  unknown
61532/tcp open  unknown
61900/tcp open  unknown
62078/tcp open  iphone-sync
63331/tcp open  unknown
64623/tcp open  unknown
64680/tcp open  unknown
65000/tcp open  unknown
65129/tcp open  unknown
65389/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
nikhitha@nikhitha-virtual-machine:~$
nikhitha@nikhitha-virtual-machine:~$
```

OS DETECTION USING NMAP:

As a professional security tester, we should go a further step to gain additional information about the network or host which will boost our pen-testing.The –O parameter used to detect the target operating system:

Command: nmap –O target

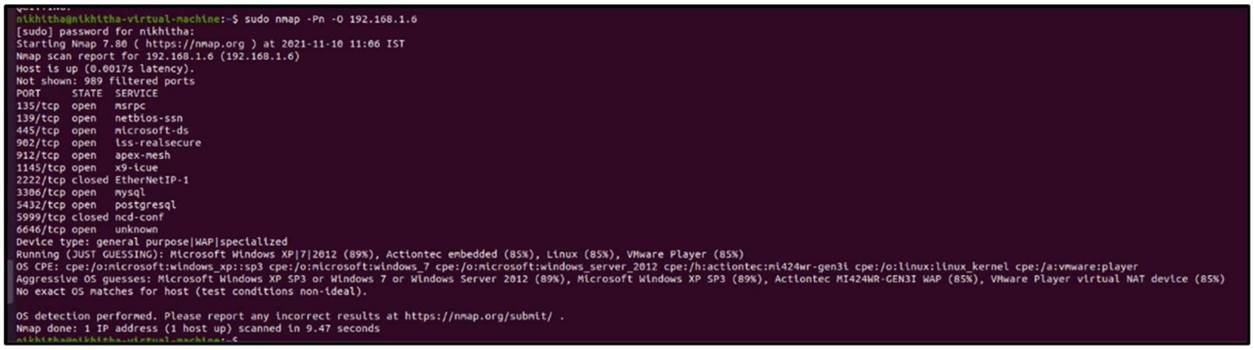
Multiple options for nmap can be used, like –v.

Ex: nmap –v –O <target>

Guessing the Operating System

If Nmap is unable to determine the operating system, we can use the –osscan option to force Nmap into discovering the OS.Note: This option is useful when Nmap is unable to determine the discovered OS

Command: nmap -O –osscan-guess target

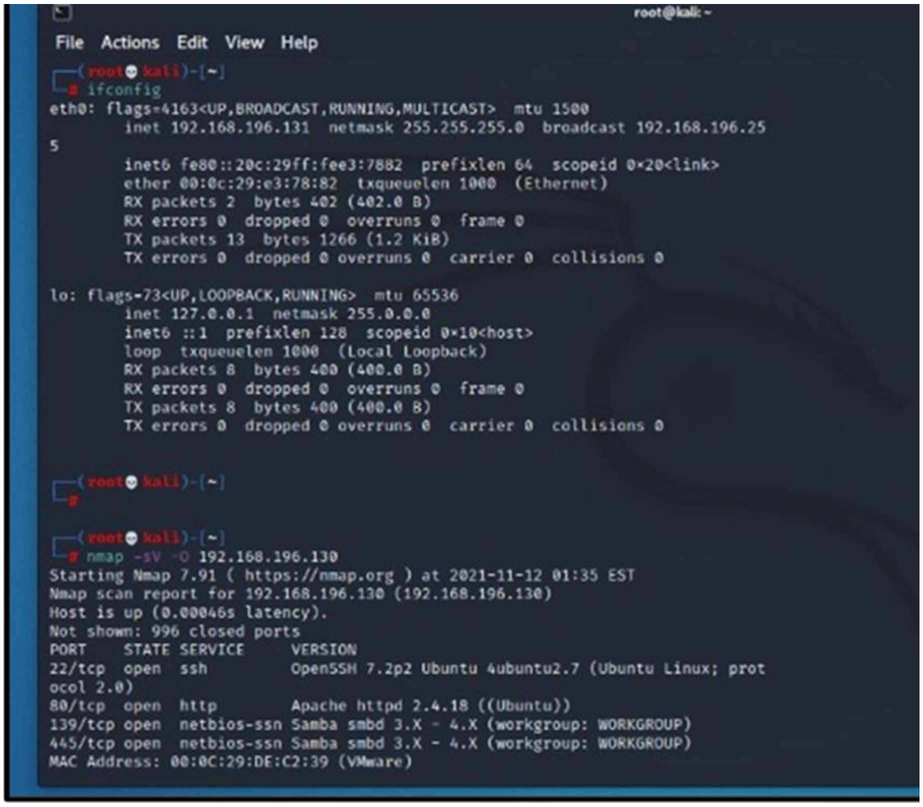


```
nikhitha@nikhitha-virtual-machine:~$ sudo nmap -Pn -O 192.168.1.6
[sudo] password for nikhitha:
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 11:06 IST
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.0017s latency).
Not shown: 989 filtered ports
PORT     STATE  SERVICE
135/tcp  open   msrpc
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
902/tcp  open   iss-realsecure
912/tcp  open   apex-mesh
1145/tcp open   x9-icue
2222/tcp closed EtherNetIP-1
3306/tcp open   mysql
5432/tcp open   postgresql
5999/tcp closed ncd-conf
6646/tcp open   unknown
Device type: general purpose|WAP|specialized
Running (JUST GUESSING): Microsoft Windows XP|7|2012 (89%), Actiontec embedded (85%), Linux (85%), VMware Player (85%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2012 cpe:/h:actiontec:mi424wr-gen3i cpe:/o:linux:linux_kernel cpe:/a:vmware:player
Aggressive OS guesses: Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (89%), Microsoft Windows XP SP3 (89%), Actiontec MI424WR-GEN3I WAP (85%), VMware Player virtual NAT device (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.47 seconds
nikhitha@nikhitha-virtual-machine:~$
```

Recon/Enumeration using NMAP and WIRESHARK:

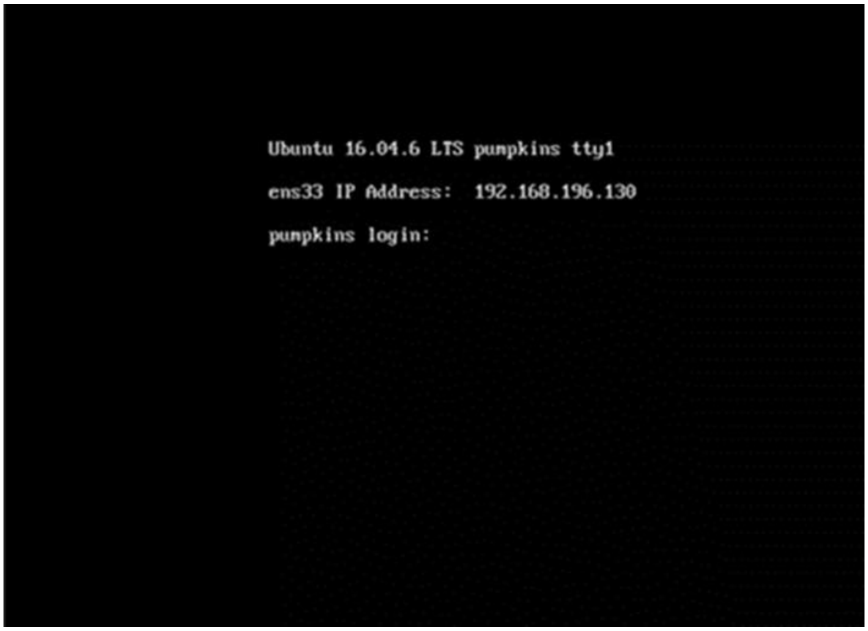
For this part of the project, we used Kali Linux since it has Wireshark already installed in it which we will be using in further steps.

Below is the IP configuration of the Kali VM which is being used. The main IP Address of the Kali Linux is 192.168.192.25.



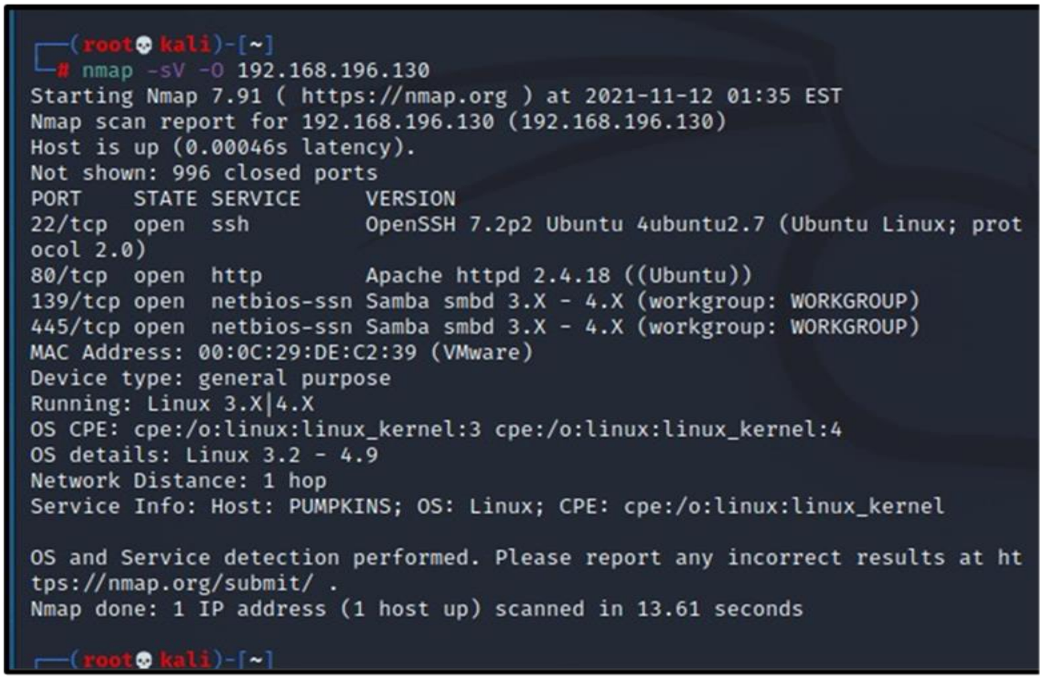
```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.196.131  netmask 255.255.255.0  broadcast 192.168.196.25
5
        inet6 fe80::20c:29ff:fee3:7882  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e3:78:82  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 402 (402.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1266 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(root㉿kali)-[~]
└─#

┌──(root㉿kali)-[~]
└─# nmap -sV -O 192.168.196.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-12 01:35 EST
Nmap scan report for 192.168.196.130 (192.168.196.130)
Host is up (0.00046s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; prot
ocol 2.0)
80/tcp  open  http        Apache httpd 2.4.18 ((Ubuntu))
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:0C:29:DE:C2:39 (VMware)
```

This is a vulnerable Linux system we are trying to enumerate using NMAP and SSH(security shell). Initially, we only know the IP Address of this Virtual Machine here: 192.168.196.130, This has a username and password login authentication., Our main purpose is to sniff on the network to find the username and password of the system and log in from our local machine.



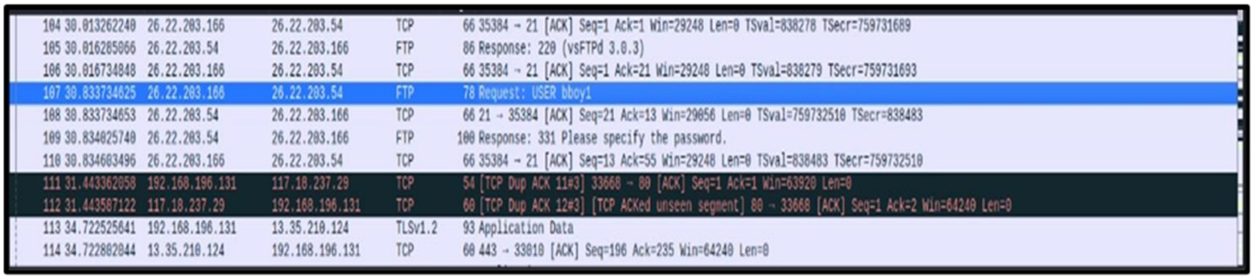
```
Ubuntu 16.04.6 LTS pumpkins tty1

ens33 IP Address:  192.168.196.130

pumpkins login:
```

Since we have the IP address, we can perform Nmap scan from Kali Linux VM to find out the open ports that are available and the services that are running on those ports. The results of the Nmap scan are shown in the following image. We found 4 open ports running ssh, HTTP and smb services on the ports mentioned. The first port has an SSH service running on it which stands for secure shell. This is used to create a secure connection between machine to machine.

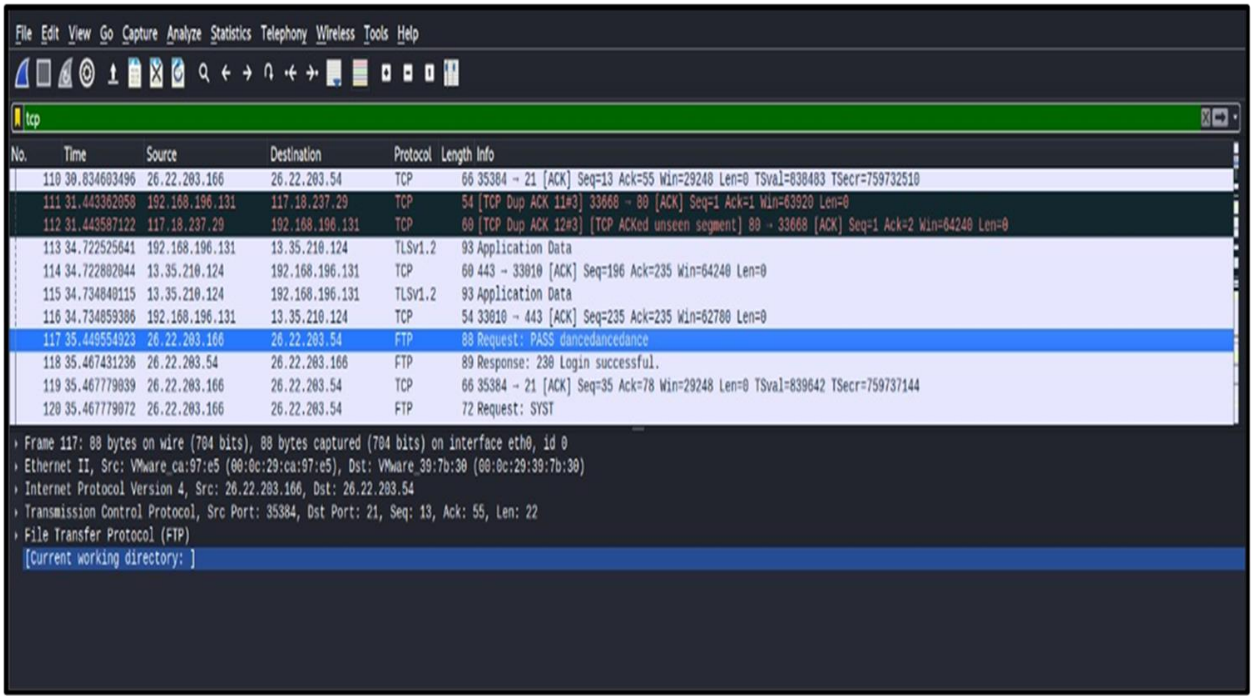


```
┌──(root💀kali)-[~]
└─# nmap -sV -O 192.168.196.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-12 01:35 EST
Nmap scan report for 192.168.196.130 (192.168.196.130)
Host is up (0.00046s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; prot
ocol 2.0)
80/tcp  open  http        Apache httpd 2.4.18 ((Ubuntu))
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:0C:29:DE:C2:39 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: PUMPKINS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.61 seconds

┌──(root💀kali)-[~]
```

SSH (Secure Shell) is a widely used remote access protocol present in almost any network, making it a very important service to enumerate. Nmap ships with multiple ready-to-use SSH enumeration scripts that aid in identifying authentication methods, grabbing SSH host keys, checking if certain public keys are accepted, detecting SSHv1 servers and running brute-force attacks. Next port has a HTTP service running on it. Let us check out the http port. When we open it in the browser by using the IP Address ie./ 192.168.196.130, we find a video as shown in the below screenshot.



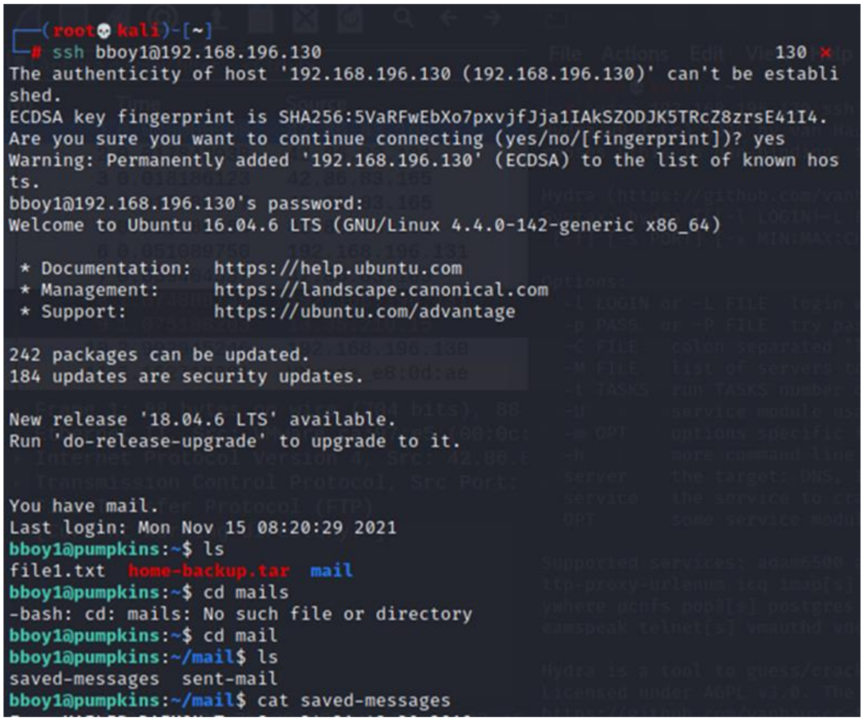
```
104 30.013262240  26.22.203.166    26.22.203.54     TCP      66 35384 → 21 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=838278 TSecr=759731689
105 30.016285066  26.22.203.54     26.22.203.166    FTP      86 Response: 220 (vsFTPd 3.0.3)
106 30.016734848  26.22.203.166    26.22.203.54     TCP      66 35384 → 21 [ACK] Seq=1 Ack=21 Win=29248 Len=0 TSval=838279 TSecr=759731693
107 30.833734625  26.22.203.166    26.22.203.54     FTP      78 Request: USER bboy1
108 30.833734653  26.22.203.54     26.22.203.166    TCP      66 21 → 35384 [ACK] Seq=21 Ack=13 Win=29056 Len=0 TSval=759732510 TSecr=838483
109 30.834025740  26.22.203.54     26.22.203.166    FTP     100 Response: 331 Please specify the password.
110 30.834603496  26.22.203.166    26.22.203.54     TCP      66 35384 → 21 [ACK] Seq=13 Ack=55 Win=29248 Len=0 TSval=838483 TSecr=759732510
111 31.443362058  192.168.196.131  117.18.237.29    TCP      54 [TCP Dup ACK 11#3] 33668 → 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
112 31.443587122  117.18.237.29    192.168.196.131  TCP      60 [TCP Dup ACK 12#3] [TCP ACKed unseen segment] 80 → 33668 [ACK] Seq=1 Ack=2 Win=64240 Len=0
113 34.722525641  192.168.196.131  13.35.210.124    TLSv1.2  93 Application Data
114 34.722802044  13.35.210.124    192.168.196.131  TCP      60 443 → 33010 [ACK] Seq=196 Ack=235 Win=64240 Len=0
```



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.  Time         Source           Destination      Protocol Length Info
110 30.834603496  26.22.203.166    26.22.203.54     TCP      66 35384 → 21 [ACK] Seq=13 Ack=55 Win=29248 Len=0 TSval=838483 TSecr=759732510
111 31.443362058  192.168.196.131  117.18.237.29    TCP      54 [TCP Dup ACK 11#3] 33668 → 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
112 31.443587122  117.18.237.29    192.168.196.131  TCP      60 [TCP Dup ACK 12#3] [TCP ACKed unseen segment] 80 → 33668 [ACK] Seq=1 Ack=2 Win=64240 Len=0
113 34.722525641  192.168.196.131  13.35.210.124    TLSv1.2  93 Application Data
114 34.722802044  13.35.210.124    192.168.196.131  TCP      60 443 → 33010 [ACK] Seq=196 Ack=235 Win=64240 Len=0
115 34.734840115  13.35.210.124    192.168.196.131  TLSv1.2  93 Application Data
116 34.734859386  192.168.196.131  13.35.210.124    TCP      54 33010 → 443 [ACK] Seq=235 Ack=235 Win=62780 Len=0
117 35.449554923  26.22.203.166    26.22.203.54     FTP      88 Request: PASS dancedancedance
118 35.467431236  26.22.203.54     26.22.203.166    FTP      89 Response: 230 Login successful.
119 35.467779039  26.22.203.166    26.22.203.54     TCP      66 35384 → 21 [ACK] Seq=35 Ack=78 Win=29248 Len=0 TSval=839642 TSecr=759737144
120 35.467779072  26.22.203.166    26.22.203.54     FTP      72 Request: SYST

› Frame 117: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface eth0, id 0
› Ethernet II, Src: VMware_ca:97:e5 (00:0c:29:ca:97:e5), Dst: VMware_39:7b:30 (00:0c:29:39:7b:30)
› Internet Protocol Version 4, Src: 26.22.203.166, Dst: 26.22.203.54
› Transmission Control Protocol, Src Port: 35384, Dst Port: 21, Seq: 13, Ack: 55, Len: 22
› File Transfer Protocol (FTP)
  [Current working directory: ]
```

To get further into the machine, we need to find out the username and password of the virtual machine. We have to sniff on the network to see if we can find any user credentials on Wireshark. As we run Wireshark on the network interface of the KALI VM which is pre-installed, we have to wait for 30 seconds and search for packets that have the username and password of the virtual machine running on that IP Address. Once we start sniffing the packets, we can see from the above image that we found a user “bboy1” as an FTP service in port 192 and the password as “dancedancedance” also as FTP service in port 187 as shown in the following figure.



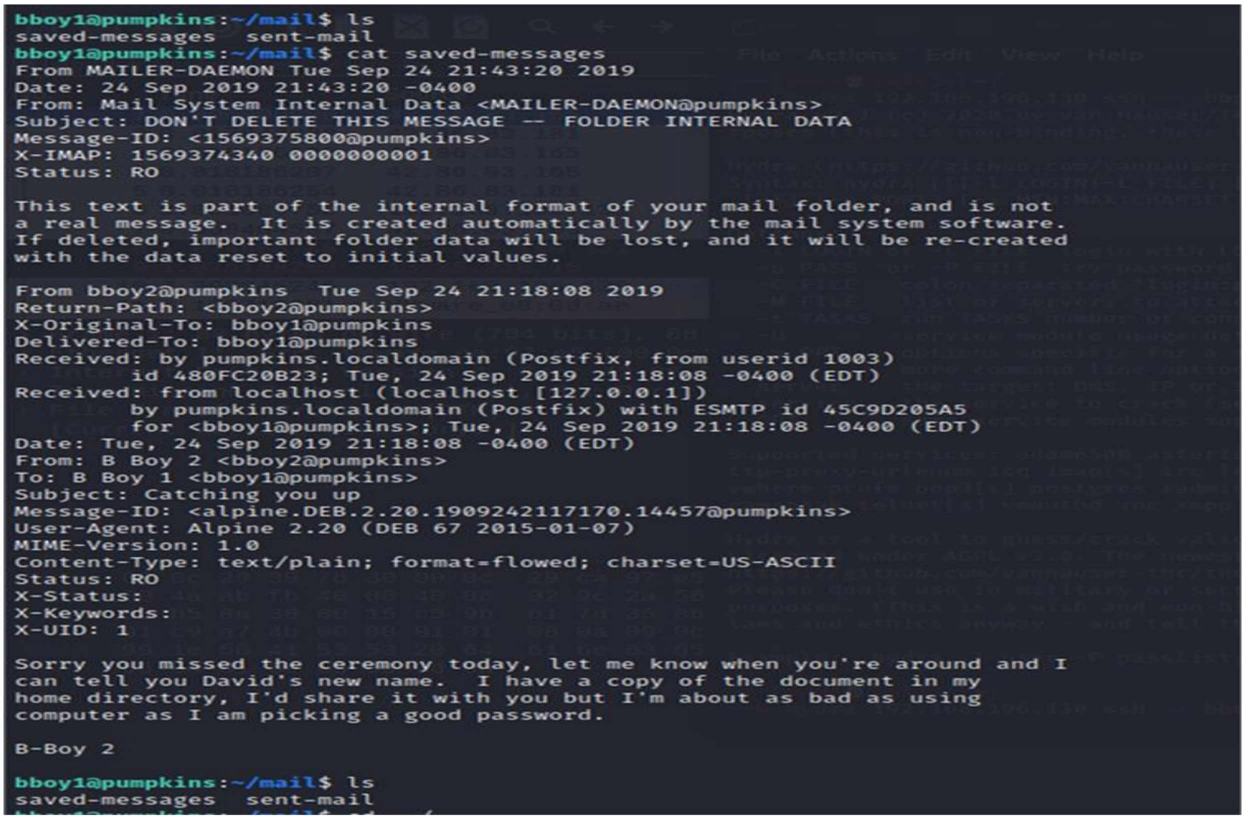
```
┌──(root💀kali)-[~]
└─# ssh bboy1@192.168.196.130                                    130 ×
The authenticity of host '192.168.196.130 (192.168.196.130)' can't be establi
shed.
ECDSA key fingerprint is SHA256:5VaRFwEbXo7pxvjfJja1IAkSZODJK5TRcZ8zrsE41I4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.196.130' (ECDSA) to the list of known hos
ts.
bboy1@192.168.196.130's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

242 packages can be updated.
184 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


You have mail.
Last login: Mon Nov 15 08:20:29 2021
bboy1@pumpkins:~$ ls
file1.txt  home-backup.tar  mail
bboy1@pumpkins:~$ cd mails
-bash: cd: mails: No such file or directory
bboy1@pumpkins:~$ cd mail
bboy1@pumpkins:~/mail$ ls
saved-messages  sent-mail
bboy1@pumpkins:~/mail$ cat saved-messages
```

As a result of packet sniffing from Wireshark, we found out the username and password of the vulnerable virtual machine. Now we login in to the VM for further enumeration from our local machine (kali Linux) by using the below ssh command and entering the password that we found out

Ssh username@ipaddress

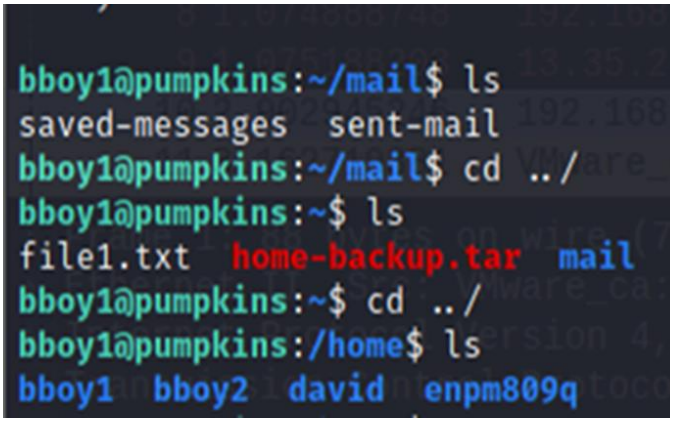
Ssh bboy1@192.168.196.130

Next, it will ask to enter the password ie., ‘dancedancedance’. Once we enter the credentials, we are successfully into the virtual machine. Now we can enter into the directories, view the files, and even open them also as shown in the picture. As we can see below the VM has 2 files and a subdirectory. One .tar file, one mail subdirectory, and a text file. When we change the directory to mail: we can see the directory has two emails saved in it. Below shows the output of when we cat (view the contents) of saved messages. We are able to read a saved message from the mails directory of a virtual machine from our local machine. This shows that this virtual machine is quite vulnerable and easy to break in.



```
bboy1@pumpkins:~/mail$ ls
saved-messages  sent-mail
bboy1@pumpkins:~/mail$ cat saved-messages
From MAILER-DAEMON Tue Sep 24 21:43:20 2019
Date: 24 Sep 2019 21:43:20 -0400
From: Mail System Internal Data <MAILER-DAEMON@pumpkins>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1569375800@pumpkins>
X-IMAP: 1569374340 0000000001
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message.  It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.

From bboy2@pumpkins  Tue Sep 24 21:18:08 2019
Return-Path: <bboy2@pumpkins>
X-Original-To: bboy1@pumpkins
Delivered-To: bboy1@pumpkins
Received: by pumpkins.localdomain (Postfix, from userid 1003)
        id 480FC20B23; Tue, 24 Sep 2019 21:18:08 -0400 (EDT)
Received: from localhost (localhost [127.0.0.1])
        by pumpkins.localdomain (Postfix) with ESMTP id 45C9D205A5
        for <bboy1@pumpkins>; Tue, 24 Sep 2019 21:18:08 -0400 (EDT)
Date: Tue, 24 Sep 2019 21:18:08 -0400 (EDT)
From: B Boy 2 <bboy2@pumpkins>
To: B Boy 1 <bboy1@pumpkins>
Subject: Catching you up
Message-ID: <alpine.DEB.2.20.1909242117170.14457@pumpkins>
User-Agent: Alpine 2.20 (DEB 67 2015-01-07)
MIME-Version: 1.0
Content-Type: text/plain; format=flowed; charset=US-ASCII
Status: RO
X-Status:
X-Keywords:
X-UID: 1

Sorry you missed the ceremony today, let me know when you're around and I
can tell you David's new name.  I have a copy of the document in my
home directory, I'd share it with you but I'm about as bad as using
computer as I am picking a good password.

B-Boy 2

bboy1@pumpkins:~/mail$ ls
saved-messages  sent-mail
```

As part of the enumeration, we got into the mails of the user and we could read the personal mail since it was in read-only (RO) mode. Further, when we navigate to the home directly, we can see the users working on it. The VM has 4 users: bboy1, bboy2, David enpm809q. This process shows us that the VM is quite vulnerable and we could easily enter with the help of NMAP AND WIRESHARK.



```
bboy1@pumpkins:~/mail$ ls
saved-messages  sent-mail
bboy1@pumpkins:~/mail$ cd ../
bboy1@pumpkins:~$ ls
file1.txt  home-backup.tar  mail
bboy1@pumpkins:~$ cd ../
bboy1@pumpkins:/home$ ls
bboy1  bboy2  david  enpm809q
```

CONCLUSION:

As a result of this project, we learned the following: We determined what hosts are available on the network, what services (application name and version) those hosts are offering, We found out the operating systems (and OS versions) they are running. We found out what type of packet filters/firewalls are in use, and dozens of other characteristics. We learned multiple ping messages using NMAP to understand our network better. We also learned that It is very important to keep our network very secure and strong. Our network should have a firewall to protect us from intruders entering our server. Everybody must be aware of the various Nmap scans and detecting the vulnerability of our Machine and NMAP is a very helpful tool.

A secure network not only filters out malware but also provides layers of defence against possible cyber assaults. It accomplishes this by dividing all data flowing into and out of the network into tiny packets. It then encrypts each packet individually and sends it out across numerous channels. Even if someone breaks into your system, they will never have access to all your data in one spot. Therefor, it is very important and is our responsibility as a user to keep our network secure and run regular vulnerability checks.