Adam Sandberg

ITAS 281

2022-09-12

ITAS 281 - Linux Server Management II

Assignment One - Linux Security aspects

Table of Contents

Introduction:

This document will go over the step-by-step procedures of using SSH, firewalld and nmap scanning. The document will cover how to properly secure your Linux server using the aforementioned applications and features. To begin SSH will be setup using a key-pair for an extra level of security. This keypair will also require a passphrase to be authenticated. Secondly the firewall of our Linux server will be configured using the firewalld application. The configuration of the firewall will provide a higher level of protection and security to our Linux server. And finally, nmap scans will be run on the Linux servers for the purpose of penetration testing. This will verify that the actions and steps we took to secure our servers was successful.

Part 1 - Secure SSH and firewalld configuration

Objective: Setup SSH keypair authentication and a firewall that only allows ssh connections on the host-only private subnet NIC and denies ssh connections on the public NIC.

1. Using an application called PuTTYgen, we will create the SSH keypair.
2. In PuTTYgen make sure SSH-2 RSA key is enabled in the key dropdown menu.
3. Click the generate button to begin the creation of your keys. You will need to move your cursor around in the white box to progress the loading bar.

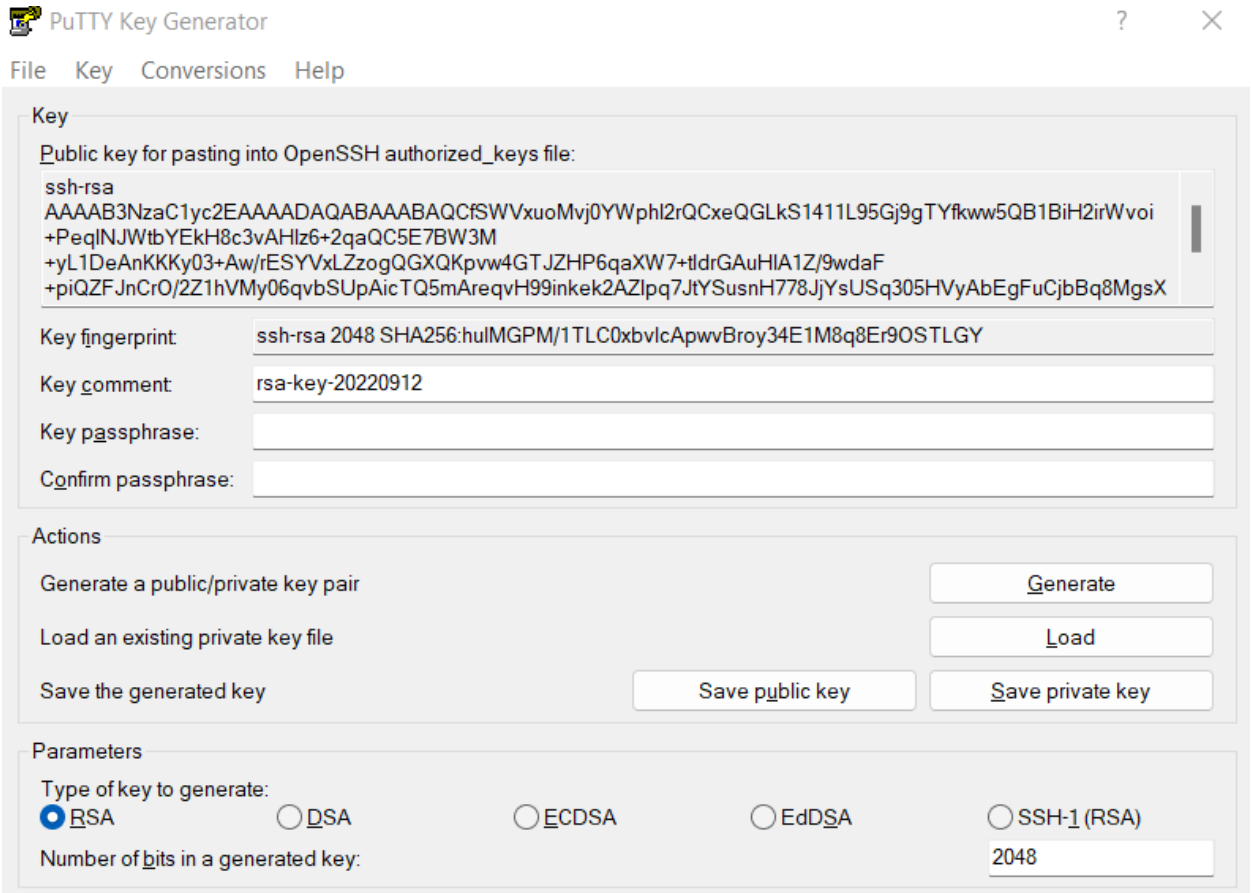

Figure 1 Generating Key in PuTTYgen

4. You can now enter a key passphrase. This will be used for further authentication when using the key to login through SSH.
5. After entering the passphrase change the number of bits at the lower right of the window from 2048 to 4096. Then press Save Public and Private key. After saving the keys to your desired location rename then to your initials and the type of key it is.
6. Now that the keys have been generated, we need to transfer the public key into our Linux machine.

Install Public Key on Linux Server

1. Using MobaXterm, SSH into your Linux Server.
2. We need to create a directory for our SSH key to live in.
3. CD to your user in the home directory then use this command 'mkdir ~/.ssh'
4. You will then need to assign permissions to the folder. Use chmod 700 ~/.ssh
5. Now create the authorized keys file. touch ~/.ssh/authorized_keys.
6. Now give that file these permissions. chmod 600 ~/.ssh/authorized_keys. Your directory should look like this.

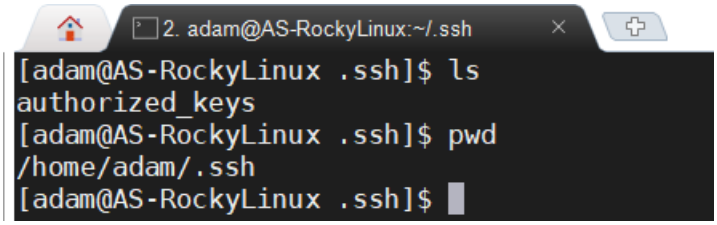

Figure 2 LS and PWD of newly created Directory

7. Now we need to use the Vi editor to paste or public key into our authorized_keys file. Use the vi ~/.ssh/authorized command. Then press "o" and right click to paste your key. Save and close the document by typing ":" followed by wq then pressing enter. It will look something like this.

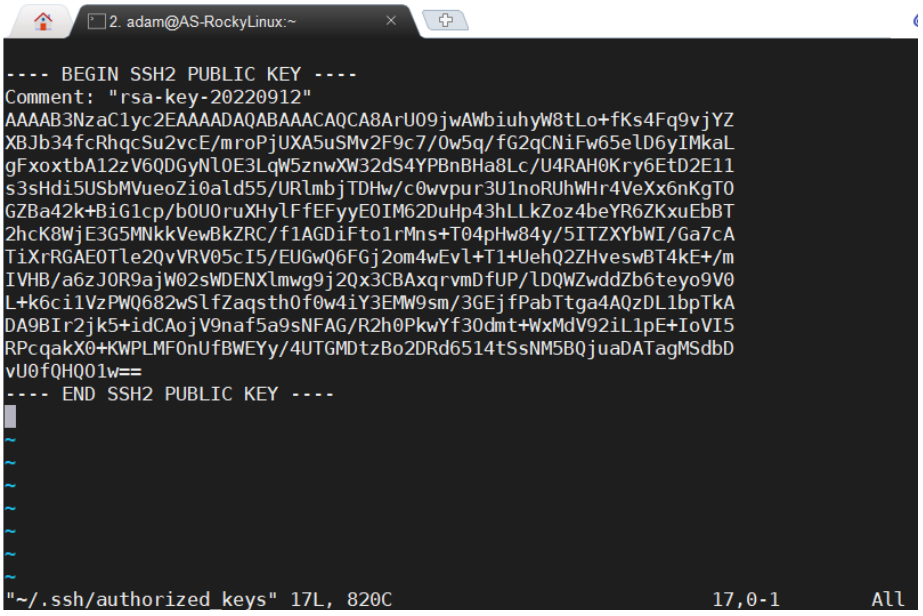

Figure 3 Contents of authorized_keys File

Use PuTTY and your Private key for SSH

1. Now that the public key has been installed onto our Linux server, we can now use PuTTY to connect via SSH and keypair to the Server.
2. Launch PuTTY, then in the left category area under connection open the SSH dropdown menu. Select Auth.
3. At the bottom of the Auth page press browse and locate your private key and press okay. See figure 4.

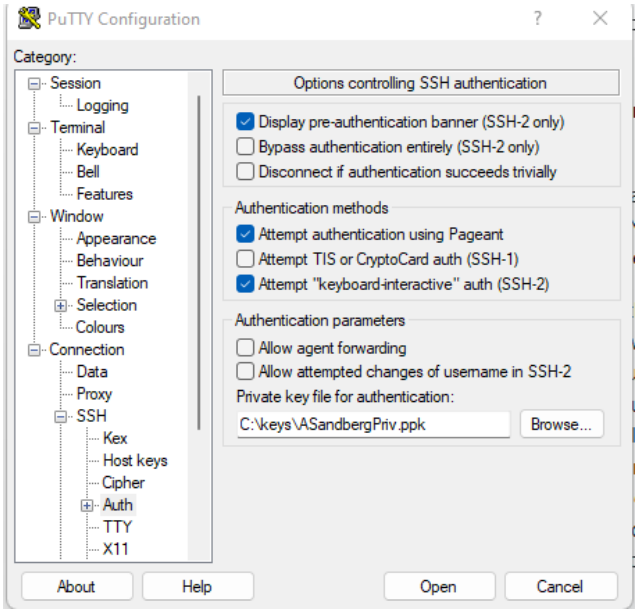

Figure 4 PuTTY Configuration Page

4. Now go back up to session. Inside the session window is where you will enter your Linux servers IP address or hostname. To use hostname ensure you have added the entry into your host file on you host machine. My session window looks like this. Note that I have saved this a profile for easy connection. This will not cache the key either so do not worry about that aspect.

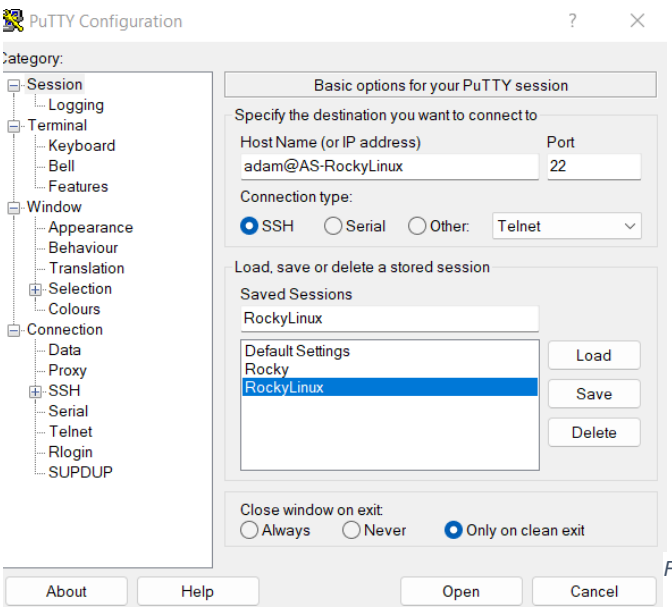

Figure 5 PuTTY Session Page

5. No press open, you will be prompted to enter your keys passphrase, then you will be logged in via SSH to your Linux Server. See figure 6.

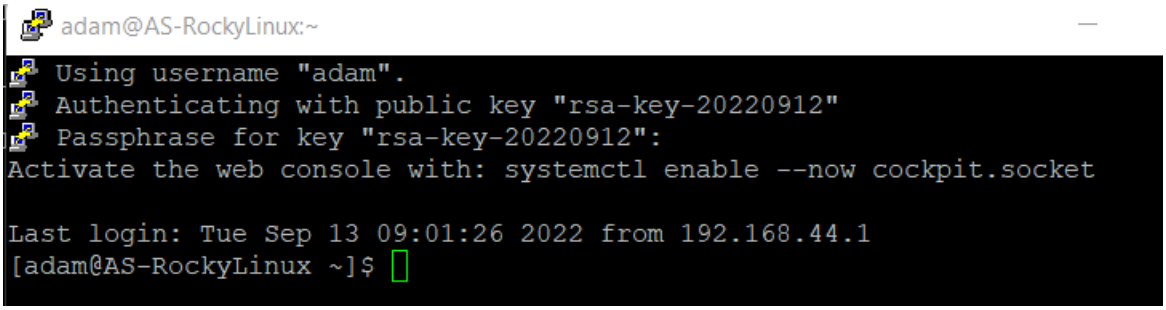

Figure 6 Successful Login Through PuTTY

Part 2 - Configuring Firewalld

Objective: A firewall structure will be created that only allows SSH connections through our host only NIC and denies SSH attempts to our bridged connection NIC. The rules must be boot safe.

1. Power on your Linux machine then SSH into it with PuTTY using the profile you saved from the previous part.
2. First let us look at what our active zones are. Type firewall-cmd –get-active-zones. This should show what adapters are in what zones.

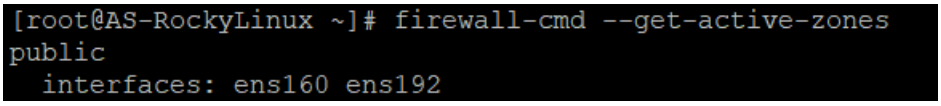

Figure 7 Active Zones

3. As you can see in figure 7, both adapters are in the public zone. We need to move the ens192 (Host-Only) to the internal zone and the ens160 will stay in the public zone as it is our world facing NIC.
4. To move ens192 to the internal zone use this command ‘firewall-cmd –zone=internal –change-interface=ens192’
5. To verify this has worked use the command from step 2.

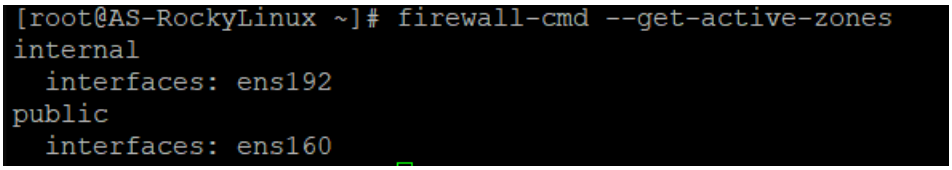

Figure 8 Active Zones After Moving ens192

6. Now that the adapters are in the correct zones, we need to configure the zones with there own individual rules. We need to disable ssh in the public zone. To do this we need to remove SSH from the public zone. Use this command ‘# firewall-cmd --permanent --zone=public --remove-service=ssh’ then firewall-cmd –permanent –zone=public –remove-port=22/tcp. Once finished, reload the firewall with firewall-cmd –reload.

```
[root@AS-RockyLinux ~]# # firewall-cmd --permanent --zone=public --remove-service=ssh
[root@AS-RockyLinux ~]# firewall-cmd --reload
success
[root@AS-RockyLinux ~]# firewall-cmd -permanent -zone=public -remove-port=22/tcp
usage: see firewall-cmd man page
firewall-cmd: error: unrecognized arguments: -permanent -zone=public -remove-port=22/tcp
[root@AS-RockyLinux ~]# firewall-cmd --permanent --zone=public --remove-port=22/tcp
Warning: NOT_ENABLED: 22:tcp
success
[root@AS-RockyLinux ~]# firewall-cmd --reload
success
```

Figure 9 Firewalld configuration Commands

7. Now that the firewall zones have been configured, we need to make a failsafe in case the firewall goes down. To do this we need to tell the ssh daemon to only listen to ssh requests from our host only adapter. This will be done in the SSH config file. Type vi /etc/ssh/sshd_config and press enter. Locate the ListenAddress field. Uncomment it and change the address to that of your host only network adapter.

```
#Port 22
#AddressFamily any
ListenAddress 192.168.44.2
#ListenAddress ::
```

Figure 10 ListenAddress Changed in sshd_config

8. That completes the setup of our firewall.

Using TCPdump to show SSH requests

1. Now we can look at our SSH connections in real time using TCPdump. To begin power on your Linux server and login to your account.
2. To monitor the SSH connections we will use this command 'tcpdump -I ens160 port 22 -n' this will show SSH attempts on the external NIC.

```
root@AS-RockyLinux:~
Passphrase for key "rsa-key-20220912":
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Sep 13 18:14:13 2022 from 192.168.44.1
[adam@AS-RockyLinux ~]$ su -
Password:
Last login: Tue Sep 13 18:31:08 PDT 2022 on tty1
[root@AS-RockyLinux ~]# tcpdump -I ens160 port 22 -n
tcpdump: ens160: That device doesn't support monitor mode
[root@AS-RockyLinux ~]# tcpdump -i ens160 port 22 -n
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), capture size 262144 bytes
18:38:04.686010 IP 192.168.1.74.49874 > 192.168.1.72.ssh: Flags [S], seq 7331566
24, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
18:38:05.688669 IP 192.168.1.74.49874 > 192.168.1.72.ssh: Flags [S], seq 7331566
24, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
18:38:07.690152 IP 192.168.1.74.49874 > 192.168.1.72.ssh: Flags [S], seq 7331566
24, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
18:38:11.702396 IP 192.168.1.74.49874 > 192.168.1.72.ssh: Flags [S], seq 7331566
24, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
18:38:19.708661 IP 192.168.1.74.49874 > 192.168.1.72.ssh: Flags [S], seq 7331566
```

Figure 11 SSH tcpdump on external NIC

As you can see in figure 11, the SSH request was never accepted and eventually timed out.

3. Now we need to do the same but on the other network adapter. Use this command ‘tcpdump -I ens192 port 22 -n’
4. You will be assaulted by roughly a million messages per second but that’s okay, SSH is very chatty.

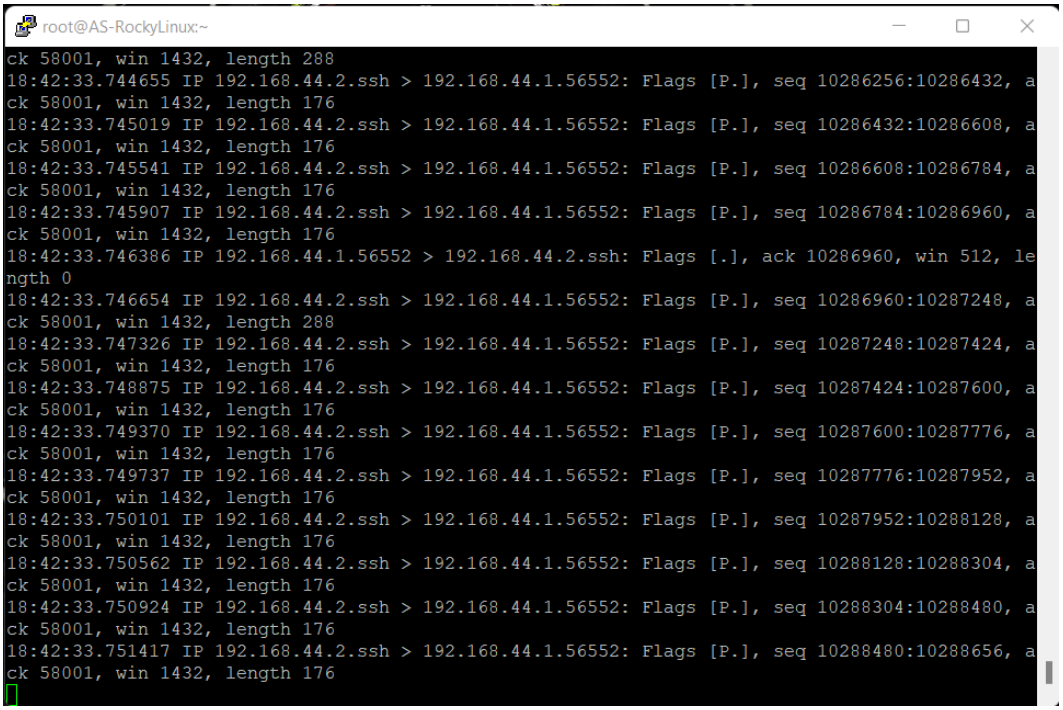

Figure 12 Activity on Host Only NIC

This concludes the tcpdump section, this can be a very helpful tool in diagnosing network issues on your Linux server as well as give you the ability to see what devices are using/logging into to SSH.

Part 3 - Nmap Scans

Objective: Use Nmap to perform various scans on our networks and adapters. The commands used will be documented as well as the flags. The results will then be presented along with the commands. A target server was created and the nfs-utils package was installed on it to add more ports that are actively listening. This target server will be the “target” of the nmap scans. Scans will be done on each adapter with the firewall and off for one of the scans.

TCP SYN Scan

Commands Used: nmap -sS -p22,25,110 192.168.44.5 and nmap -sS -p22,25,110 192.168.1.71

Host-Only Adapter:

```
[root@AS-RockyLinux ~]# nmap -sS -p22,25,110 192.168.44.5
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 15:49 PDT
Nmap scan report for 192.168.44.5
Host is up (-0.13s latency).

PORT    STATE    SERVICE
22/tcp  open     ssh
25/tcp  filtered smtp
110/tcp filtered pop3
MAC Address: 00:0C:29:8E:8F:2D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.76 seconds
```

Figure 13 TCP SYN Scan Host-Only

Public Adapter:

```
[root@AS-RockyLinux ~]# nmap -sS -p22,25,110 192.168.1.71
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 15:50 PDT
Nmap scan report for 192.168.1.71
Host is up (-0.13s latency).

PORT    STATE    SERVICE
22/tcp  open     ssh
25/tcp  filtered smtp
110/tcp filtered pop3
MAC Address: 00:0C:29:8E:8F:23 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
```

Figure 14 TCP SYN Scan Internal Firewall on

TCP SYN Scan Results:

For the TCP SYN scan both the Host-Only and the Public NIC produced similar results. The SSH port state was open and 998 other ports were filtered. I used the -p flag to filter it down to more relevant ports.

UDP Scan

Command Used nmap -sU -v --min-rate 5000 'IP address'

-sU is the program option for the UDP scan and -v is verbose mode. The min rate flag is important as it makes the scan complete very quickly. "When the --min-rate option is given Nmap will do its best to send packets as fast as or faster than the given rate."

UDP Scan Host-Only:

```
[root@AS-RockyLinux ~]# nmap -sU -v --min-rate 5000 192.168.44.5
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 16:02 PDT
Initiating ARP Ping Scan at 16:02
Scanning 192.168.44.5 [1 port]
Completed ARP Ping Scan at 16:02, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:02
Completed Parallel DNS resolution of 1 host. at 16:02, 0.03s elapsed
Initiating UDP Scan at 16:02
Scanning 192.168.44.5 [1000 ports]
Completed UDP Scan at 16:02, 0.58s elapsed (1000 total ports)
Nmap scan report for 192.168.44.5
Host is up (0.00072s latency).
Not shown: 994 open|filtered ports
PORT      STATE    SERVICE
112/udp   filtered mcidas
139/udp   filtered netbios-ssn
682/udp   filtered xfr
7000/udp  filtered afs3-fileserver
19374/udp filtered unknown
28493/udp filtered unknown
MAC Address: 00:0C:29:8E:8F:2D (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.92 seconds
           Raw packets sent: 1995 (57.714KB) | Rcvd: 7 (364B)
```

Figure 15 UDP Scan Host-Only

UDP Scan Public:

```
[root@AS-RockyLinux ~]# nmap -sU -v --min-rate 5000 192.168.1.71
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 16:07 PDT
Initiating ARP Ping Scan at 16:07
Scanning 192.168.1.71 [1 port]
Completed ARP Ping Scan at 16:07, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:07
Completed Parallel DNS resolution of 1 host. at 16:07, 0.18s elapsed
Initiating UDP Scan at 16:07
Scanning 192.168.1.71 [1000 ports]
Completed UDP Scan at 16:07, 0.54s elapsed (1000 total ports)
Nmap scan report for 192.168.1.71
Host is up (-0.090s latency).
Not shown: 994 open|filtered ports
PORT      STATE    SERVICE
120/udp   filtered cfdptkt
217/udp   filtered dbase
1058/udp  filtered nim
3456/udp  filtered IISrpc-or-vat
27482/udp filtered unknown
39888/udp filtered unknown
MAC Address: 00:0C:29:8E:8F:23 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.02 seconds
           Raw packets sent: 2002 (57.910KB) | Rcvd: 7 (364B)
```

Figure 16 UDP Scan Public

UDP Port Scan Results:

The Host-Only and the Public adapter share no relation in terms of what ports were shown after the scan. The one similarity is that every single port produced a filtered status. Seeing all of the filtered states I thought this scan would be a good one to show with firewall disabled. The scans below were done with firewalld stopped.

UDP Scan Host-Only Firewall Off:

```
[root@AS-RockyLinux ~]# nmap -sU -v --min-rate 5000 192.168.44.5
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 17:11 PDT
Initiating ARP Ping Scan at 17:11
Scanning 192.168.44.5 [1 port]
Completed ARP Ping Scan at 17:11, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:11
Completed Parallel DNS resolution of 1 host. at 17:11, 0.03s elapsed
Initiating UDP Scan at 17:11
Scanning 192.168.44.5 [1000 ports]
Discovered open port 111/udp on 192.168.44.5
Completed UDP Scan at 17:11, 0.50s elapsed (1000 total ports)
Nmap scan report for 192.168.44.5
Host is up (-0.078s latency).
Not shown: 993 open|filtered ports
PORT      STATE  SERVICE
111/udp   open   rpcbind
997/udp   closed maitrd
18605/udp closed unknown
21405/udp closed unknown
28973/udp closed unknown
32768/udp closed omad
62699/udp closed unknown
MAC Address: 00:0C:29:8E:8F:2D (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.83 seconds
           Raw packets sent: 1995 (57.674KB) | Rcvd: 8 (424B)
```

Figure 17 UDP Scan Firewall Off

In the scan done in figure 17 it now shows that all but ports are closed and not filtered now that the firewall is off. The 111-port used for rpcbind (from nfs-utils) was the only scanned port that was open. This is not what I was really expecting to happen, but it appears even with the firewall disabled, Rocky Linux does an excellent job of locking these ports down.

UDP Scan Public Firewall Off:

```
[root@AS-RockyLinux ~]# nmap -sU -v --min-rate 5000 192.168.1.71
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 17:27 PDT
Initiating ARP Ping Scan at 17:27
Scanning 192.168.1.71 [1 port]
Completed ARP Ping Scan at 17:27, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:27
Completed Parallel DNS resolution of 1 host. at 17:27, 0.22s elapsed
Initiating UDP Scan at 17:27
Scanning 192.168.1.71 [1000 ports]
Discovered open port 111/udp on 192.168.1.71
Completed UDP Scan at 17:27, 0.81s elapsed (1000 total ports)
Nmap scan report for 192.168.1.71
Host is up (-0.068s latency).
Not shown: 993 open|filtered ports
PORT      STATE  SERVICE
111/udp   open   rpcbind
998/udp   closed puparp
1028/udp  closed ms-lsa
1050/udp  closed cma
9020/udp  closed tambora
23531/udp closed unknown
64481/udp closed unknown
MAC Address: 00:0C:29:8E:8F:23 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds
           Raw packets sent: 3001 (86.769KB) | Rcvd: 9 (484B)
```

The UDP scan on the public adapter showed related results to that of the Host-Only. The difference being that on the public Nic, the rpcbind port 111 was opened and the rest were in a close state. I believe these ports are open because they are installed/enabled software.

TCP Scan Range 20-80

Command used: nmap -p20-80 'IP Address'

TCP Scan Range 20-80 - Host Only:

```
[root@AS-RockyLinux ~]# nmap -p20-80 192.168.44.5
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 17:37 PDT
Nmap scan report for 192.168.44.5
Host is up (-0.068s latency).
Not shown: 60 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:8E:8F:2D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.00 seconds
```

Figure 18 TCP Scan Host-Only

TCP Scan Range 20-80 - Public:

```
[root@AS-RockyLinux ~]# nmap -p20-80 192.168.1.71
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 17:40 PDT
Nmap scan report for 192.168.1.71
Host is up (-0.068s latency).
Not shown: 60 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:8E:8F:23 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.10 seconds
```

Figure 19 TCP Scan Public

TCP Scan Range 20-80 Results:

The two scans were basically identical in results. The scan says sixty ports were filtered and not shown. This means the sixty ports that were scanned all dropped the packet with zero response. This method of filtering gives attackers little info so its good to see most of my ports are in that state.

Info Gathering Scan:

Using nmaps remote OS detection using TCP/IP stack fingerprinting we can gather some info on the targets. This will tell us nmaps best guess or exact version of the OS of the target. The guesses will be made with a percentage of certainty next to it as well as some extra info on uptime, network distance and a TCP sequence prediction.

Command Used: nmap -O -v 'IP Address'

Including the -O -v options caused Nmap to include more OS based info.

Info Scan - Linux Target:

```
[root@AS-RockyLinux ~]# nmap -O -v 192.168.44.5
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 18:00 PDT
Initiating ARP Ping Scan at 18:00
Scanning 192.168.44.5 [1 port]
Completed ARP Ping Scan at 18:00, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:00
Completed Parallel DNS resolution of 1 host. at 18:00, 0.14s elapsed
Initiating SYN Stealth Scan at 18:00
Scanning 192.168.44.5 [1000 ports]
Discovered open port 22/tcp on 192.168.44.5
Completed SYN Stealth Scan at 18:00, 8.71s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.44.5
Retrying OS detection (try #2) against 192.168.44.5
adjust_timeouts2: packet supposedly had rtt of -180951 microseconds.  Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -180951 microseconds.  Ignoring time.
Nmap scan report for 192.168.44.5
Host is up (-0.0027s latency).
Not shown: 998 filtered ports
PORT     STATE  SERVICE
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:0C:29:8E:8F:2D (VMware)
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.2 - 4.9 (96%), Linux 3.16 - 4.6 (95%), Linux 2.6.32 - 3.13 (95%), Linux 4.10 (93%), Linux 2.6.22 - 2.6.3
6 (93%), Linux 3.10 (93%), Linux 2.6.39 (93%), Linux 4.4 (92%), Linux 2.6.32 (92%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 2.914 days (since Sun Sep 11 20:04:40 2022)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/bin/../share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.17 seconds
           Raw packets sent: 3052 (138.614KB) | Rcvd: 63 (5.886KB)
```

Figure 20 Info Scan - Linux Target

Info Scan – Windows Host:

```
[root@AS-RockyLinux ~]# nmap -O -v 192.168.1.74
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-14 18:06 PDT
Initiating ARP Ping Scan at 18:06
Scanning 192.168.1.74 [1 port]
Completed ARP Ping Scan at 18:06, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:06
Completed Parallel DNS resolution of 1 host. at 18:06, 0.16s elapsed
Initiating SYN Stealth Scan at 18:06
Scanning 192.168.1.74 [1000 ports]
Completed SYN Stealth Scan at 18:07, 21.27s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.1.74
Retrying OS detection (try #2) against 192.168.1.74
Nmap scan report for 192.168.1.74
Host is up (-0.20s latency).
All 1000 scanned ports on 192.168.1.74 are filtered
MAC Address: C4:03:A8:36:B6:40 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Read data files from: /usr/bin/../share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.60 seconds
           Raw packets sent: 2050 (94.728KB) | Rcvd: 1 (28B)
```

Figure 21 Info Scan - Windows Host

Info Scan Results:

After completing the two info scans on my Linux guest OS and my Windows host OS it was observed that the Linux scan and the Windows scan were inconclusive. Nmap determined that it was with 97% certainty running a Linux kernel version from 3.10 to 4.11. When in reality the kernel version of Rocky Linux used was 4.18 as pictured below using the uname -r command. Nmap never made a guess for kernel version 4.18.

```
[root@localhost ~]# uname -r
4.18.0-372.9.1.el8.x86_64
[root@localhost ~]# _
```

Figure 22 Linux Kernel Version

As for the Windows host machine, the scan was unable to determine the OS details. The reason for this was “Too many fingerprints match this host”. This means that out of all the tests nmap does to check for the OS it was unable to find any matching options. I wonder if this possibly could be because my laptop is now on Windows 11, or possibly it is part of my Windows Defender blocking nmap from getting any data on it.

Ping Scan (Subnet Scan)

Use a ping scan to determine what devices are on the ITAS year 2 network.

Command Used: nmap -sn 172.16.102.0/24 | more

```
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 12:55 PDT
Nmap scan report for Assignment1R (172.16.102.17)
Host is up (0.0021s latency).
MAC Address: 00:0C:29:A8:0A:77 (VMware)
Nmap scan report for 172.16.102.23
Host is up (0.0013s latency).
MAC Address: E4:54:E8:2D:E5:4E (Unknown)
Nmap scan report for Rocky (172.16.102.32)
Host is up (0.0020s latency).
MAC Address: 00:0C:29:46:83:F5 (VMware)
Nmap scan report for 172.16.102.40
Host is up (-0.10s latency).
MAC Address: 42:58:60:59:79:31 (Unknown)
Nmap scan report for AS-XPS (172.16.102.56)
Host is up (-0.10s latency).
MAC Address: 3C:18:A0:99:5E:58 (Luxshare Precision Industry Company Limited)
Nmap scan report for 172.16.102.68
Host is up (-0.10s latency).
MAC Address: FC:34:97:4B:B8:23 (Unknown)
Nmap scan report for eh11 (172.16.102.77)
Host is up (-0.10s latency).
MAC Address: 48:4D:7E:E3:FA:20 (Dell)
Nmap scan report for 172.16.102.78
Host is up (-0.093s latency).
MAC Address: 94:65:9C:61:57:BA (Intel Corporate)
Nmap scan report for kobi (172.16.102.80)
Host is up (-0.086s latency).
MAC Address: 04:56:E5:AE:25:31 (Unknown)
Nmap scan report for 172.16.102.90
Host is up (0.0013s latency).
MAC Address: 7C:C2:C6:1D:9B:5F (Unknown)
Nmap scan report for KenechukwuObi-PC (172.16.102.92)
Host is up (0.0048s latency).
MAC Address: 04:56:E5:AE:25:31 (Unknown)
Nmap scan report for 172.16.102.93
Host is up (0.0038s latency).
MAC Address: 70:66:55:AF:E1:A9 (Unknown)
Nmap scan report for 172.16.102.96
Host is up (0.0015s latency).
MAC Address: 00:E0:4C:36:07:B3 (Realtek Semiconductor)
Nmap scan report for 172.16.102.252
Host is up (0.012s latency).
MAC Address: 00:01:E6:80:9A:0C (Hewlett Packard)
Nmap scan report for 172.16.102.253
Host is up (0.0042s latency).
MAC Address: B4:99:BA:C9:A4:80 (Hewlett Packard)
Nmap scan report for 172.16.102.254
```

Figure 23 nmap Ping Scan

Ping Scan Results:

Looking at the results of our ping scan we can see a few devices. An obvious one would be the 172.16.102.254 which is labeled as a Hewlett Packard device. This is our ITAS year 2 default gateway/router. 172.16.102.253 is out ITAS year 2 switch, this device is also an HP and was located with the nmap command from this step.

NMAP Scan and TCPdump Capture:

Objective: Get a partner to do a nmap scan on your Linux machine while capturing the activity using TCPdump. Save the TCPdump to a .cap file and open it in wireshark. Using certain wireshark settings, narrow down and search for specific sections of the capture.

My partner scanned my device with nmap, using this TCPdump command 'tcpdump -i <interface> -s 65535 -w <file>' I was able to save the scan to a file compatible for wireshark. Now that the file was made from TCPdump we need to get the file onto the Windows host. On the Host open PowerShell and use the sftp command.

```
PS C:\Users\Adam> sftp root@192.168.44.5:capture.cap C:\
```

Figure 24 sftp to Get TCPdump

This will place the file on your Windows C drive.

Launch Wireshark and open the file.

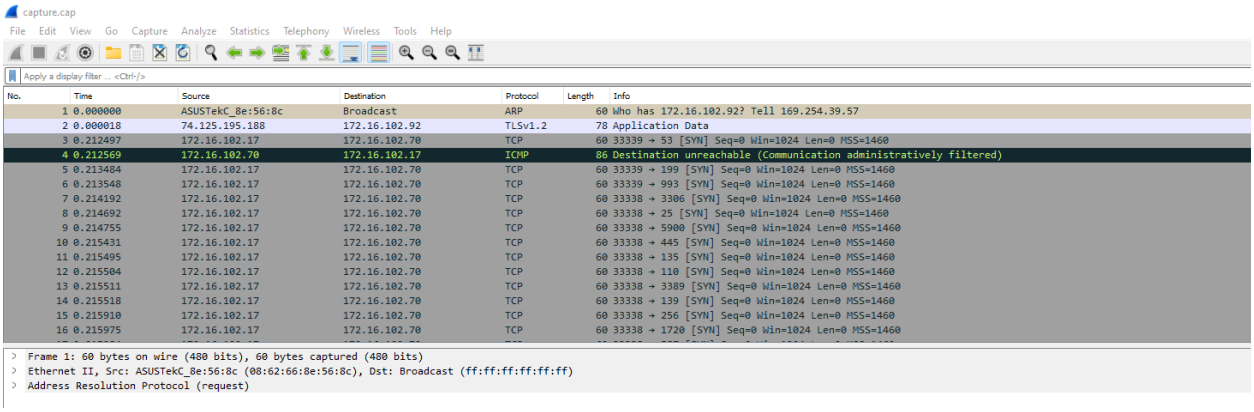

Figure 25 Wireshark with TCPdump File Loaded

This completes the TCPdump and Wireshark section.

Conclusion:

After the completion of this document, we now have the ability to create SSH keypairs and log in securely using this new method. I learned that using keypairs is a great way to increase the security of your SSH routine. After configuring the keypairs firewalld was explored and configured to offer secure settings when using a Linux machine and its SSH features. Finally, nmap and TCPdump were used to probe some of our Linux servers. This showed us how firewalls and Linux operating systems oversee internet ports, port states and incoming connections. Understanding these quirks and features to Linux is a major step to being able to secure a server properly. I really enjoyed this project and I think working with the firewall on any machine is an extremely important skill to have.

YouTube Link: https://youtu.be/nEVr0BBoHPU

References

Admin. “Filtering SSH Packets with Tcpdump Port 22.” Howtouselinux, 3 Feb. 2021, https://www.howtouselinux.com/post/debugging-ssh-packets-with-tcpdump.

Arj. “Difference between NMAP TCP SYN Scan and TCP CONNECT SCAN.” Medium, Medium, 11 Aug. 2017, https://medium.com/@avirj/nmap-tcp-syn-scan-50106f818bf1.

Bajrami, Valentin. “Running a Quick Nmap Scan to Inventory My Network.” Enable Sysadmin, Red Hat, Inc., 27 July 2022, https://www.redhat.com/sysadmin/quick-nmap-inventory.

“Capturing TCPdump for Wireshark Viewing.” D.3. Tcpdump: Capturing with "Tcpdump" for Viewing with Wireshark, https://www.wireshark.org/docs/wsug_html_chunked/AppToolstcpdump.html.

cyberciti. How to Find Which Linux Kernel Version Is Installed on My System. https://www.cyberciti.biz/faq/find-print-linux-unix-kernel-version/.

“Os Detection: Nmap Network Scanning.” OS Detection | Nmap Network Scanning, https://nmap.org/book/man-os-detection.html.

Stackexchange. “Increase Speed in Nmap UDP Scan?” Information Security Stack Exchange, 1 July 1961, https://security.stackexchange.com/questions/52566/increase-speed-in-nmap-udp-scan.

“Technical Tip: Nmap Scan Shows Ports as Filtered.” Technical Tip: NMAP Scan Shows Ports as Filtered, 9 June 2021, https://community.fortinet.com/t5/FortiGate/Technical-Tip-NMAP-scan-shows-ports-as-filtered/ta-p/194519?externalID=FD52501.

“UDP Scan (-SU): Nmap Network Scanning.” UDP Scan (-SU) | Nmap Network Scanning, https://nmap.org/book/scan-methods-udp-scan.html.

“Usage and Examples: NMAP Network Scanning.” Usage and Examples | Nmap Network Scanning, https://nmap.org/book/osdetect-usage.html.

Vance, Nathan. “Sysadmin.” Home, https://www.linuxjournal.com/content/understanding-firewalld-multi-zone-configurations.

Yasin, Aqsa. CentOS 8 Disable Firewall, 1 Jan. 1968, https://linuxhint.com/disable-firewall-centos-8/.