

ITAS 281: Linux Security Aspects

By Ethan Holmes

Table of Contents

Introduction.....	3
Part 1 – Secure SSH.....	3
Part 2 – Firewall.....	8
Part 3 – Nmap Scanning.....	10
Conclusion	16
References.....	16
Link to Video Submission.....	17

Introduction

In this assignment, I will be showing how I generated key pairings to be used in SSH authentication and demonstrating how I can SSH from my client application puTTY, into my Linux machines on a private host only network, and the firewall configurations that I use to make it secure.

Part 1 – Secure SSH

After we have created our necessary virtual machines and have done the needed configuration of them, we can start with the creation of our SSH authentication key pair, we will require 2 keys, a public and a private. The public key will be attached to our server, while our public key will be attached to our windows machine as we will be using puTTY to SSH into our machine.

To start the creation of these keys, we will open puTTYgen, which comes with the install of puTTY, puTTYgen will allow us to generate the public and private keys for our machines.

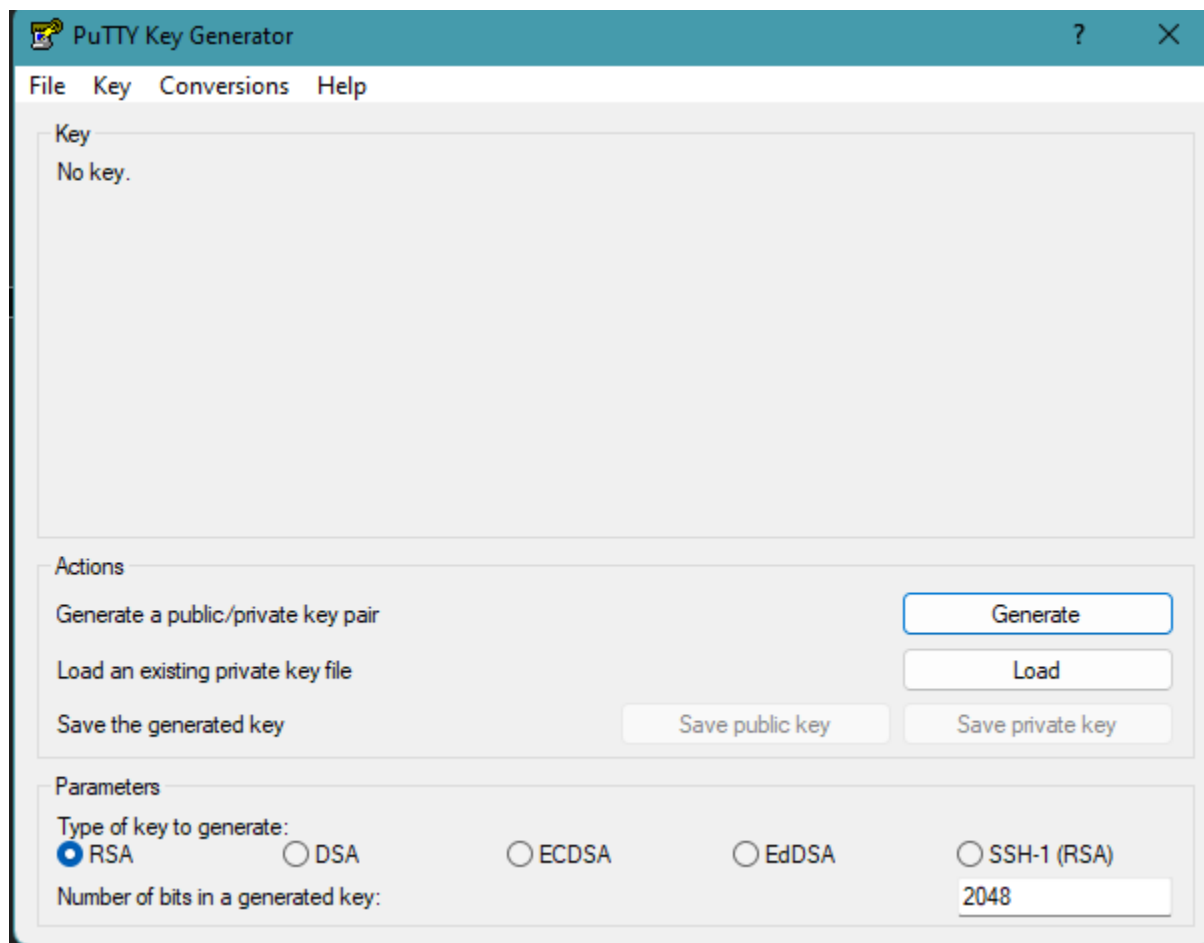


Figure 1: puTTY Key Generator screen

Once opened, set the “Number of bits in a generated key” to 4096 and click “Generate” which will give you the prompt to shake your mouse around to create the hash for the keys.

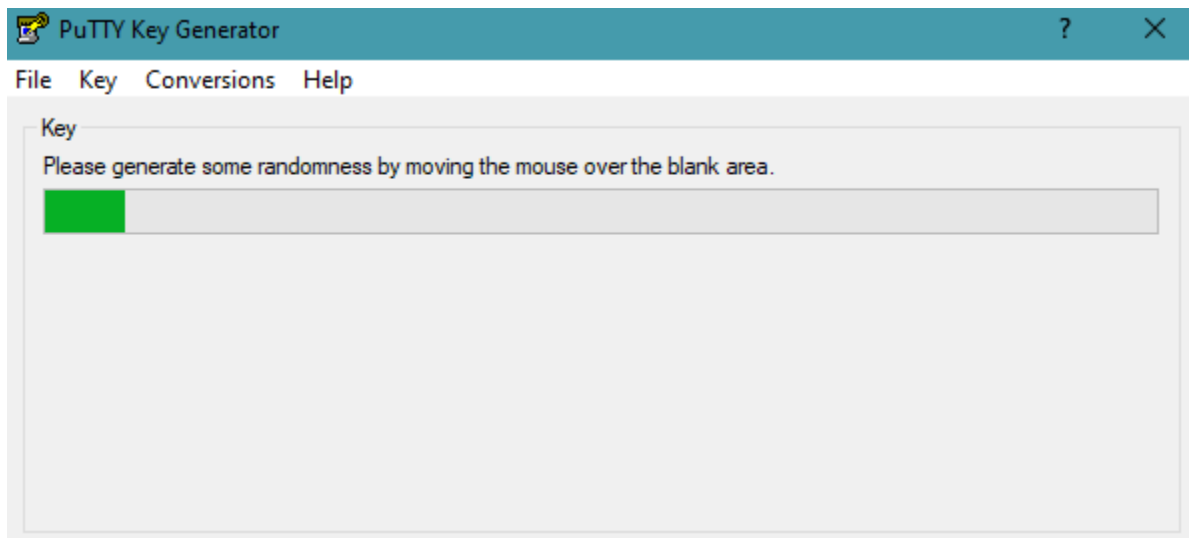


Figure 2: Generating Hash for the public and private keys in puTTY

Once this is done, you will be given a long string that will be your private key for your machine. You will also be given the ability to save your public and private keys, do that as they will be needed in the future.

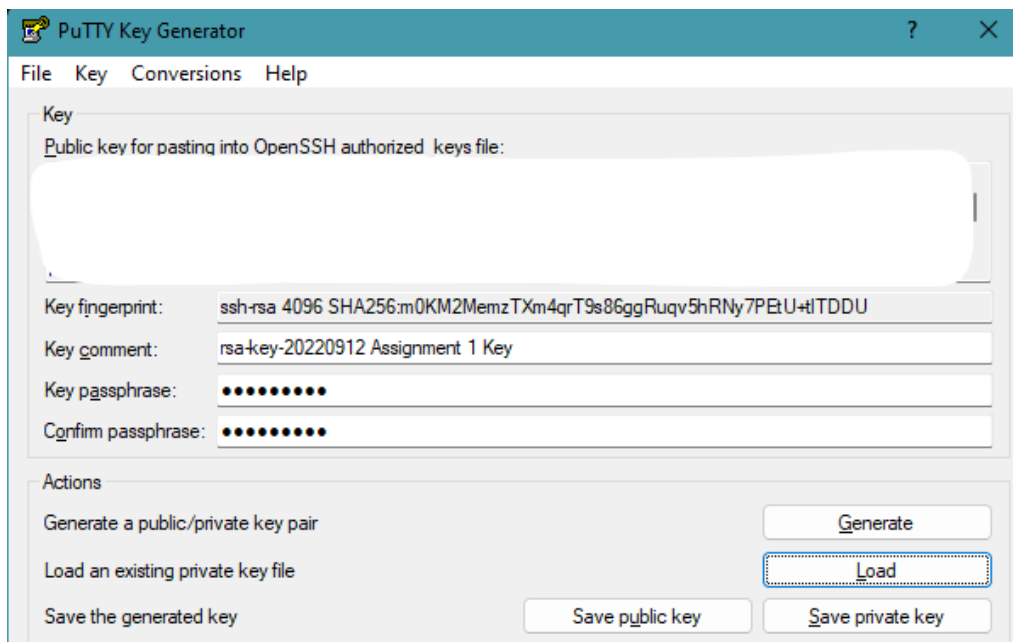


Figure 3: Generated Key along with the Save Public and Private Key options

Below the comment, we will see the “Key Passphrase” section, **THIS IS A MUST HAVE FOR SECURITY REASONS**. Without the passphrase you can SSH into the server that requires the key without the need to

enter a password, it is an extra layer of security for the system. Once this has all been setup and the keys have been saved, we can now move to our puTTY client and our VM to move our keys.

Inside our Virtual Machine, we will first need to prepare the machine to accept the key and allow connections through key authentication. We can do that by changing directory to our /etc/ssh as root and then doing “vi sshd_config” inside this configuration file we will need to find and uncomment and type yes to “PubkeyAuthentication”

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
```

Figure 4: Changing the sshd_config file

After this, we can finally import our private key into our machine. For this we will need to go to our user folder and create a “.ssh” directory as well as a “authorized_keys” file inside to drop our private key into as this will not be automatically created since we did not create our key pairs within the linux machine.

```
Last login: Tue Sep 13 00:00:16 2022 from 192.168.199.1
[Ethan@Assignment1R ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh
[Ethan@Assignment1R ~]$ cd /home
[Ethan@Assignment1R home]$ ls -a
.  ..  Ethan
[Ethan@Assignment1R home]$ cd Ethan
[Ethan@Assignment1R ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh
[Ethan@Assignment1R ~]$ cd .ssh
[Ethan@Assignment1R .ssh]$ ls -a
.  ..  authorized_keys
[Ethan@Assignment1R .ssh]$ vi authorized_keys
[Ethan@Assignment1R .ssh]$
```

Figure 5: Path to the authorized_keys text file

After adding our public key to the authorized_keys file, we can go back to puTTY and test if our key pairing has worked. To do this, we can open puTTY and underneath the “Connection” drop down on the

left side of the screen, we can drop down the “ssh” menu and click on “Auth” where we will be given the option to add a private key to be used for authentication. From here, browse to the location where you’ve saved your private key and add it into puTTY.

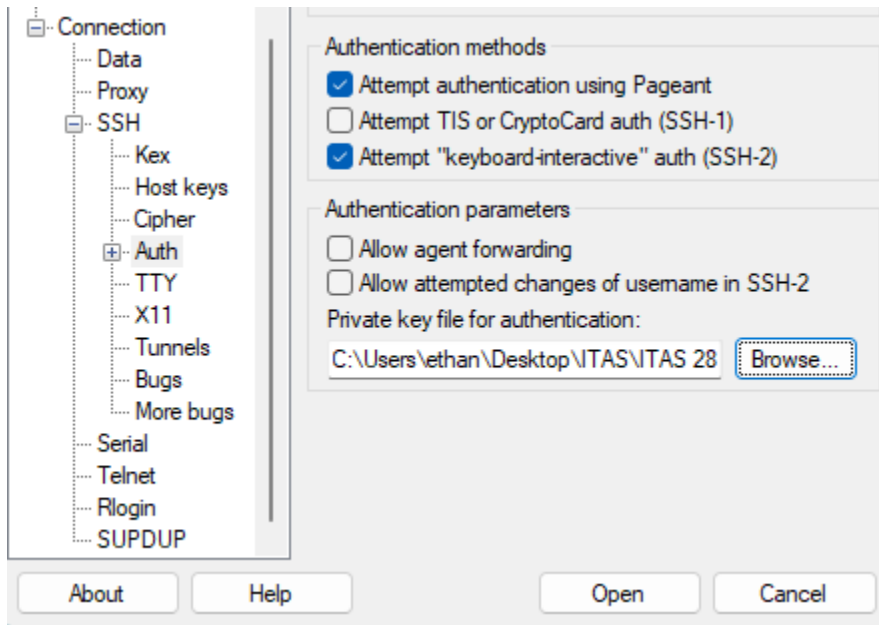


Figure 6: Adding of the Private Key file in puTTY

Once added, you can then click “open” to open the connection to your machine and see if your key pairing has worked.

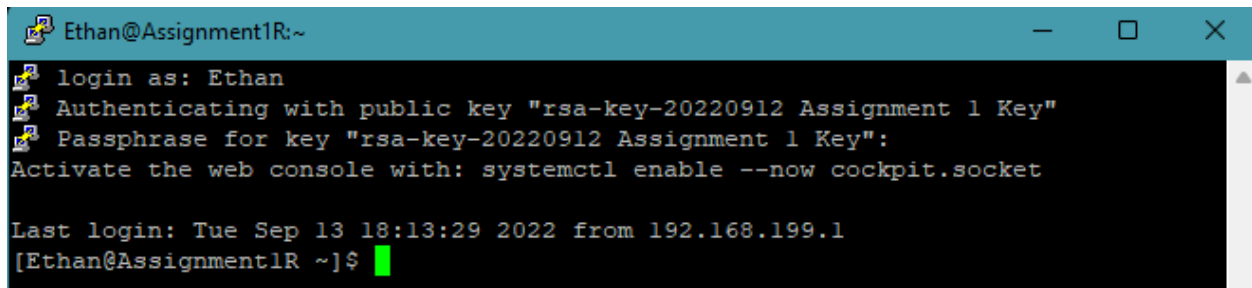


Figure 7: Logging into puTTY using the passphrase of the private key

As demonstrated in figure 7, we have successfully authenticated into the machine using our key pairing and the passphrase we provided to our private key file in figure 3 and can now use the machine from our puTTY terminal.

Part 2 – Firewalld

After configuring our ssh terminal, our next objective towards securing our linux machine is to configure our firewalld, which are our various firewall settings as well as configuring our sshd_config file to allow for certain connections to our machine. The first thing I did first was to change the listening address to our bridged network as seen in figure 8 below, we do this in the case of a firewall failure, we can fall back on the listening port to filter out any ip addresses that we do not want to connect to the machine.

```
#
#Port 22
#AddressFamily any
ListenAddress 192.168.199.14
#ListenAddress ::
```

Changing the port is not necessary unless we want to have a secondary channel open for connections

Figure 8: Assigning our bridged network to be our listening address

After this has been configured, we can then move on to configuring the firewall.

First, we can see what zones we have running currently, to do this we can use “firewall-cmd --get-active-zones” which will bring up our NICs and their current groups ens192 being my host-only network and ens160 being my bridged ITAS network. On an unconfigured machine, these NICs will both be in the “public” group. However, because mine is already configured you will see the groups that the NICs should be in.

```
[root@Assignment1R ssh]# firewall-cmd --get-active-zones
internal
  interfaces: ens192
public
  interfaces: ens160
[root@Assignment1R ssh]#
```

Figure 9: The NICs on their internal and public groups

To separate the NICs into these respective groups, we can run the following command

“firewall-cmd --zone=(zone-name) --change-interface=(interface-name)”

After separating the 2 NICs into the internal and public groups, the next thing we need to do is disable SSH and the SSH port on our bridged network.

To disable the SSH protocol on our bridged network, we can run the following command.

“firewall-cmd --zone=public --remove-service=ssh”

By using the following command, this will disable the service as seen in figure 10 below


```
[root@Assignment1R ssh]# firewall-cmd --zone=public --remove-service=ssh
Warning: NOT_ENABLED: 'ssh' not in 'public'
success
[root@Assignment1R ssh]#
```

Figure 10: Successfully removing SSH from public group

While not necessary, you can also block port 22 from being accessed if you want to be extra safe, however since we have already disabled SSH it is not needed, the command for doing so however is

“firewall-cmd --zone=(zone-name) --remove-port=22/tcp”

That will disable your SSH port

```
[root@Assignment1R ~]# firewall-cmd --zone=public --remove-port=22/tcp
Warning: NOT_ENABLED: '22:tcp' not in 'public'
success
[root@Assignment1R ~]#
```

Figure 11: Disabling the SSH port

After all firewall settings have been configured and tested that they work, our final step is to save all these settings permanently and make them boot safe, to do this we can run the command

“firewall-cmd --runtime-to-permanent”

Now that are settings have been saved, we can reboot our firewall to refresh our settings that we have applied.

Part 3 – Nmap Scanning

For this part, we will be scanning other machines that are on our networks by using nmap (Network Mapping), which is used for network vulnerability testing. After downloading nmap using the “yum install nmap” command, the first thing we will want to do is test on our own machines connected to the ITAS network and has nmap up and running. For my testing purposes, I used the machine that I have created for our lab for this class. For this we will want to run various types of scans for TCP and UDP, mainly TCP SYN port scanning. Below is a collection of screen shots that will have the description of what command was run and what the command searches for.

```
[root@Assignment1R ~]# nmap 172.16.102.32 -sS
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 15:55 EDT
Nmap scan report for Rocky (172.16.102.32)
Host is up (-0.017s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
9090/tcp  closed zeus-admin
MAC Address: 00:0C:29:46:83:F5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 9.73 seconds
[root@Assignment1R ~]# nmap 192.168.199.2 -sS
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 15:56 EDT
Nmap scan report for 192.168.199.2
Host is up (0.0011s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
9090/tcp  closed zeus-admin
MAC Address: 00:0C:29:46:83:FF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 23.02 seconds
[root@Assignment1R ~]# _
```

Figure 12: TCP Syn scan of ITAS network and host only network

```
[root@Assignment1R ~]# nmap 192.168.199.2 -sS -p 20-80
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 16:03 EDT
Nmap scan report for 192.168.199.2
Host is up (-0.068s latency).
Not shown: 60 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:0C:29:46:83:FF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
[root@Assignment1R ~]# nmap 172.16.102.32 -sS -p 20-80
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 16:03 EDT
Nmap scan report for Rocky (172.16.102.32)
Host is up (0.0011s latency).
Not shown: 60 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:0C:29:46:83:F5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
[root@Assignment1R ~]# _
```

Figure 13: port scan of port 20-80 on the Host only and ITAS network, Filters through the selected ports unlike the previous scan

```
Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds
[root@Assignment1R ~]# nmap 172.16.102.32 -A -O
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 16:15 EDT
Nmap scan report for Rocky (172.16.102.32)
Host is up (0.0012s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 48:95:79:f7:bb:56:74:74:06:86:c8:1d:6d:b7:c2:66 (RSA)
|   256  ff:76:a1:73:62:8a:76:5f:54:15:d5:d7:bd:39:11:ff (ECDSA)
|_  256  af:24:01:66:b6:aa:f1:67:a4:cc:c7:e2:ff:2e:a5:39 (ED25519)
9090/tcp  closed zeus-admin
MAC Address: 00:0C:29:46:83:F5 (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).
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   1.16 ms  Rocky (172.16.102.32)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.36 seconds
[root@Assignment1R ~]#
```

Figure 14: OS Detection scan, because of the firewall it can't get an accurate reading of the OS

```

[root@Assignment1R ~]# nmap 172.16.102.77 -o
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 16:32 EDT
Nmap scan report for eh11 (172.16.102.77)
Host is up (0.00058s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
3389/tcp   open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=eh11
| Not valid before: 2022-07-03T22:56:02
|_ Not valid after: 2023-01-02T22:56:02
|_ ssl-date: 2022-09-20T20:34:02+00:00; -1s from scanner time.
MAC Address: 48:4D:7E:E3:FA:20 (Dell)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows 10 (94%), Microsoft Windows Server 2008 SP1 (89%), Microsoft Windows 10 1511 - 1607 (87%), FreeBSD 6.2-RELEASE (87%), M
icrosoft Windows Phone 7.5 or 8.0 (86%), Microsoft Windows 10 1607 (86%), Microsoft Windows 10 1703 (86%), Microsoft Windows Server 2008 R2 or Windows 8.1 (86%)
, Microsoft Windows Server 2016 (86%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ clock-skew: mean: -1s, deviation: 0s, median: -1s
|_ nbstat: NetBIOS name: EH11, NetBIOS user: (unknown), NetBIOS MAC: 48:4d:7e:e3:fa:20 (Dell)
|_ smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
|_ smb2-time:
|   date: 2022-09-20 16:34:02
|_   start_date: N/A

TRACEROUTE
HOP RTT      ADDRESS
1   0.58 ms eh11 (172.16.102.77)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 120.29 seconds
[root@Assignment1R ~]#

```

Figure 15: Scanning my Windows machine on the ITAS network gives different results back

```

[root@Assignment1R ~]# nmap 172.16.102.77 -sU --min-rate 5000
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 17:02 EDT
Nmap scan report for eh11 (172.16.102.77)
Host is up (0.00028s latency).
Not shown: 999 open|filtered ports
PORT      STATE SERVICE
137/udp    open  netbios-ns
MAC Address: 48:4D:7E:E3:FA:20 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 1.00 seconds
[root@Assignment1R ~]# _

```

Figure 16: UDP scan of ports on the ITAS network

```

Nmap done: 1 IP address (1 host up) scanned in 13.04 seconds
[root@Assignment1R ~]# nmap 172.16.102.32 -sU --min-rate 5000
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 17:06 EDT
Nmap scan report for Rocky (172.16.102.32)
Host is up (0.00073s latency).
Not shown: 994 open|filtered ports
PORT      STATE      SERVICE
19130/udp  filtered   unknown
19283/udp  filtered   keysrvr
33459/udp  filtered   unknown
40847/udp  filtered   unknown
49396/udp  filtered   unknown
54807/udp  filtered   unknown
MAC Address: 00:0C:29:46:83:F5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.82 seconds
[root@Assignment1R ~]# nmap 172.16.102.32 -sU --min-rate 5000
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-20 17:07 EDT
Nmap scan report for Rocky (172.16.102.32)
Host is up (0.0058s latency).
Not shown: 993 open|filtered ports
PORT      STATE      SERVICE
111/udp    open       rpcbind
1761/udp   closed     cft-0
3703/udp   closed     adobeserver-3
16947/udp  closed     unknown
19500/udp  closed     unknown
21405/udp  closed     unknown
32771/udp  closed     sometimes-rpc6
MAC Address: 00:0C:29:46:83:F5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.01 seconds
[root@Assignment1R ~]# _

```

Figure 17: Firewalld on vs off on the same machine and what gets scanned

After these scans, our next objective is to run a tcpdump and capture what's going on, on a different machine. For this I ran the following command:

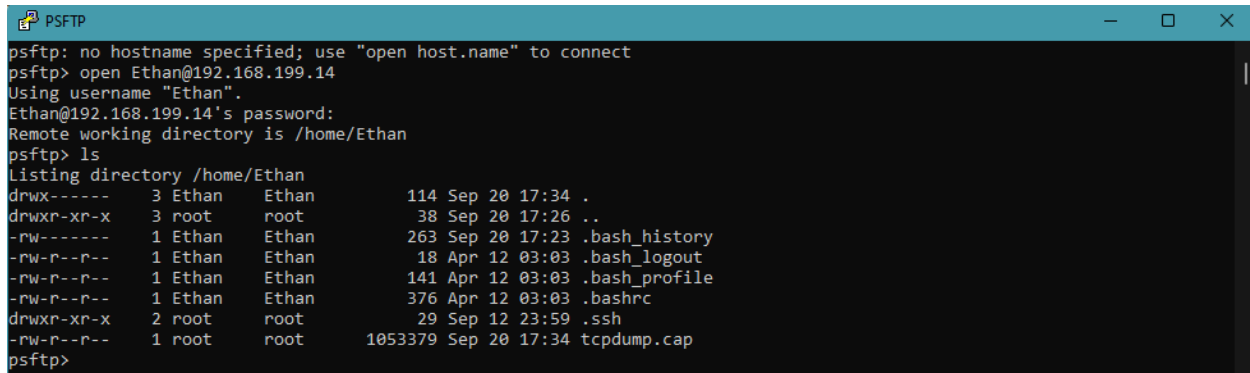
```

[root@Assignment1R Ethan]# tcpdump -i ens160 -s 65535 -w tcpdump.cap_

```

Figure 18: Command ran for tcpdump to work and give a file output

The input above will do a scan of the 172 network and then output the results of that into an extension that Wireshark can read. However, the next challenge is to get this file off my Linux machine and onto my local machine. For this I used a program that comes installed with PuTTY called psftp (PuTTY Simple File Transfer Protocol), we use this application because it is already set up with our public key to get into our remote server. If we used an application like Powershell, it would not work due to the key not being recognized. Once we have launched psftp, we can sign into our machine the same way we would SSH into the machine, where we can see we start in the '/home/Ethan' directory.

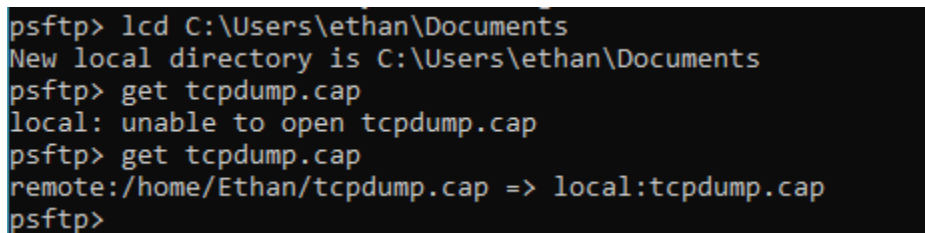


```
PSFTP
psftp: no hostname specified; use "open host.name" to connect
psftp> open Ethan@192.168.199.14
Using username "Ethan".
Ethan@192.168.199.14's password:
Remote working directory is /home/Ethan
psftp> ls
Listing directory /home/Ethan
drwx----- 3 Ethan  Ethan    114 Sep 20 17:34 .
drwxr-xr-x  3 root   root     38 Sep 20 17:26 ..
-rw----- 1 Ethan  Ethan    263 Sep 20 17:23 .bash_history
-rw-r--r-- 1 Ethan  Ethan    18 Apr 12 03:03 .bash_logout
-rw-r--r-- 1 Ethan  Ethan   141 Apr 12 03:03 .bash_profile
-rw-r--r-- 1 Ethan  Ethan   376 Apr 12 03:03 .bashrc
drwxr-xr-x  2 root   root     29 Sep 12 23:59 .ssh
-rw-r--r-- 1 root   root  1053379 Sep 20 17:34 tcpdump.cap
psftp>
```

Figure 19: psftp of my user home directory

Because I have already copied the file into this directory, we can see it's already here, however if it is not in your user directory, go to the directory where you ran your tcpdump, and use a cp command to move it into the directory we are connecting to.

Once in the directory, we can run a 'get tcpdump.cap' which will take the file from our Linux directory and copy it into the local directory that we have chosen, in this case it would be my Documents folder on my local machine.



```
psftp> lcd C:\Users\ethan\Documents
New local directory is C:\Users\ethan\Documents
psftp> get tcpdump.cap
local: unable to open tcpdump.cap
psftp> get tcpdump.cap
remote:/home/Ethan/tcpdump.cap => local:tcpdump.cap
psftp>
```

Figure 20: 'get' command with a lcd to my Documents

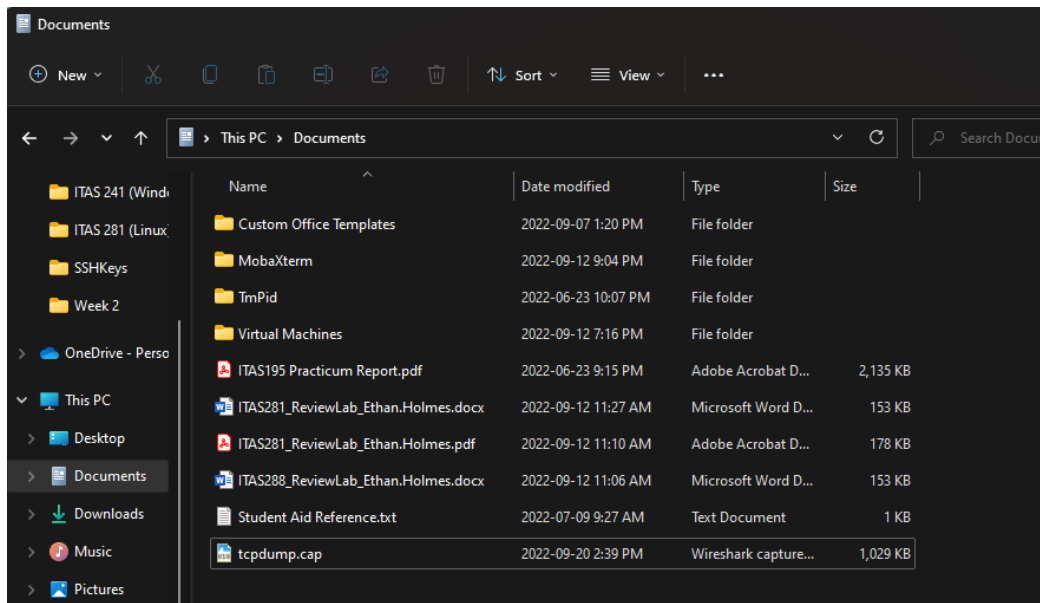


Figure 21: tcpdump.cap inside the local directory

Once on the local directory, we can then open it and see the traffic that we have captured on our interface that is on the 172 network and view it on our local machine.

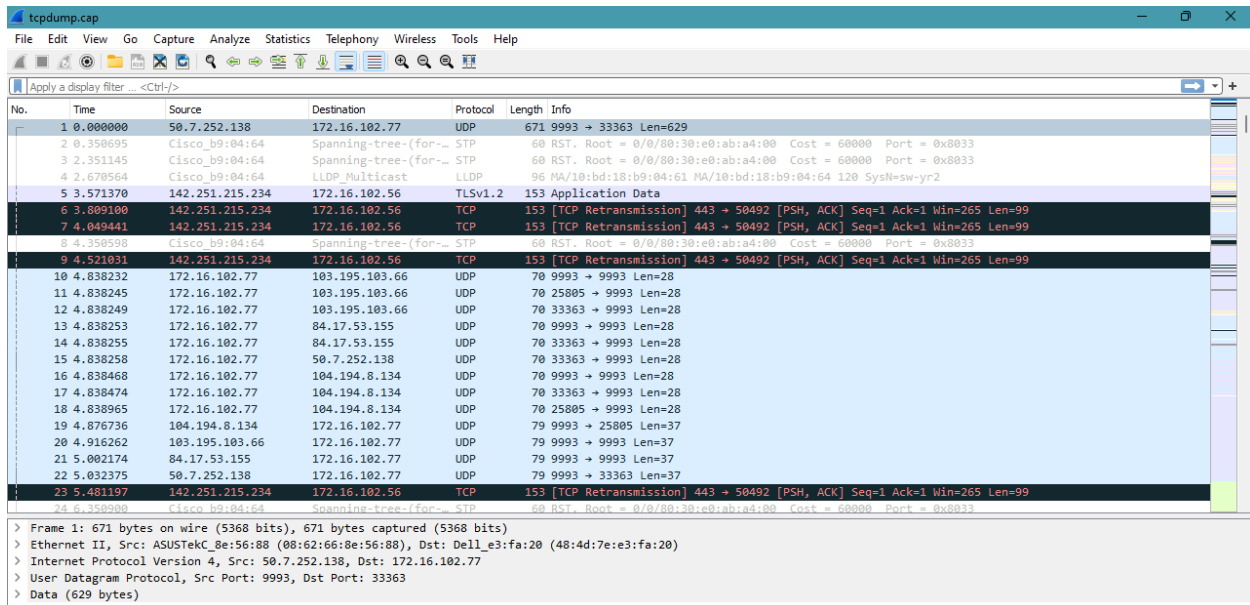


Figure 22: Wireshark results of the tcpdump

Conclusion

At the end of this project, I learned a lot about generating key pairs for SSH use and for secure authenticating, I hadn't done anything in this lab previously so learning it on the go was fun and very useful, and while the VPN was a pain to setup, I think the experience of doing it was valuable and taught me a lot about how VPNs work, on top of that the experience of getting it to work and seeing it function was very cool.

References

- Camisso, J. (2020, April 14). *How to set up and configure an openvpn server on centos 8*. DigitalOcean. Retrieved September 22, 2022, from <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-configure-an-openvpn-server-on-centos-8#optional-point-to-non-default-credentials>
- Camisso, J. (2020, April 6). *How to set up and configure a certificate authority (CA) on centos 8*. DigitalOcean. Retrieved September 22, 2022, from <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-configure-a-certificate-authority-ca-on-centos-8>
- Generating an SSH key pair using PuTTY*. Generating an SSH key pair using putty. (2021, December 14). Retrieved September 22, 2022, from <https://www.ibm.com/docs/en/flashsystem-9x00/8.3.x?topic=host-generating-ssh-key-pair-using-putty>
- Using PSFTP to transfer files securely*. Using PSFTP to transfer files securely - PuTTY Documentation. (n.d.). Retrieved September 22, 2022, from <https://documentation.help/PuTTY/psftp.html>

Link to Video Submission

<https://youtu.be/xEApjaU49rw>