Fireside Fridays

Firewalls Hands-on

Thanks to our sponsors!

Lab requirements for this section

- ff-fw-scripts.tar.gz
  - https://random-class.s3.us-east-1.amazonaws.com/ff-fw-scripts.tar.gz
- ncat
- hping3
- We used both of these tools in the packet crafting fireside
- Follow the steps on the next slide

Steps to prepare

```
sudo apt update

sudo apt -y install wget iptables tcpdump ncat hping3

wget https://random-class.s3.us-east-1.amazonaws.com/ff-fw-scripts.tar.gz

tar xvzf ff-fw-scripts.tar.gz

cd fw

ls -al
```

You should see this

```
cbrenton@rita-v5:~/fw$ ls -al
total 36
drwxrwxr-x  2 cbrenton cbrenton 4096 Apr  1  2024 .
drwxr-x--- 12 cbrenton cbrenton 4096 Apr 11 12:30 ..
-rwxrwxr-x  1 cbrenton cbrenton  187 Apr  1  2024 fw-clear
-rwxrwxr-x  1 cbrenton cbrenton  948 Apr  1  2024 fw-inspect
-rwxrwxr-x  1 cbrenton cbrenton   48 Apr  1  2024 fw-rules
-rwxrwxr-x  1 cbrenton cbrenton 1122 Apr  1  2024 fw-static
-rwxrwxr-x  1 cbrenton cbrenton  103 Apr  1  2024 kill-listen
-rwxrwxr-x  1 cbrenton cbrenton  277 Apr  1  2024 listen
-rwxrwxr-x  1 cbrenton cbrenton  543 Apr  1  2024 scan
cbrenton@rita-v5:~/fw$ _
```

I'm going to ass-you-me

- This content builds on the last Fireside Fridays content
- We covered:
  - Static, stateful and stateful inspection firewalls
  - How proxies are a completely different animal than packet filtering
  - Strengths and limitations of each
  - How to test your firewall policy
- You may want to watch that video first
- This video goes hands on with the testing portion

Hands-on walk through

- We will perform the following:
- Verify no firewall rules are in place
- Open a local TCP, then UDP port
- Perform 3 types of scans and review results
- Install static firewall and repeat last step
- Install stateful firewall and repeat again
- Review results looking for variations

My setup

- Three terminals
  - One for setup
    - Start/stop listeners
    - Modify firewall rules
  - One to run the scans and see results
  - One to run tcpdump to watch the packets
- Feel free to duplicate my setup and follow along
- We will be working with the loopback interface
  - Interface name is "lo"
  - 127.0.0.1

Scripts in /fw

- listen = Set ncat to listen on multiple ports
- kill-listen = Stop ncat listeners
- scan = Scan multiple TCP & UDP ports
- fw-rules = Print the current firewall rules
- fw-clear = Clear all firewall rules
- fw-static = Create static filter firewall rules
- fw-inspect = Create stateful firewall rules
- Type "./" before the script name to run it
- May get prompted for sudo password ("network")

Check the firewall

No rules? Use policy.

```
student@snd:~$ sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
student@snd:~$
student@snd:~$ sudo iptables -L OUTPUT
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@snd:~$
```

INPUT - Rules impacting traffic to the firewall
OUTPUT - Rules impacting traffic leaving the firewall
FORWARD - Rules impacting traffic traveling over the firewall
(skipping FORWARD for now)

Work in the ~/fw directory

```
student@snd:~$ cd fw
student@snd:~/fw$ cat fw-clear
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -F INPUT
iptables -F OUTPUT
iptables -L INPUT
iptables -L OUTPUT
student@snd:~/fw$ sudo ./fw-clear
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@snd:~/fw$
```

fw-clear = remove all rules then display

listen will open TCP & UDP test ports

```
student@snd:~/fw$ cat listen
pkill ncat
ncat -lk 127.0.0.1 1234 &
ncat -lk 127.0.0.1 1235 &
ncat -ul -w 1 127.0.0.1 1234 &
ncat -ul -w 1 127.0.0.1 1235 &
echo ncat is now listening on TCP and UDP ports 1234 and 1235 on the loopback interface
echo ncat is using these process IDs
sudo lsof -i | grep ncat

student@snd:~/fw$ ./listen
ncat is now listening on TCP and UDP ports 1234 and 1235 on the loopback interface
ncat is using these process IDs
[sudo] password for student:
ncat      6157        student    3u  IPv4 298677      0t0  TCP localhost:1234 (LISTEN)
ncat      6158        student    3u  IPv4 299922      0t0  TCP localhost:1235 (LISTEN)
ncat      6159        student    3u  IPv4 299921      0t0  UDP localhost:1234
ncat      6160        student    3u  IPv4 298678      0t0  UDP localhost:1235
student@snd:~/fw$
```

Known bug. UDP listeners only work once and then need to be reset.

Script for port scanning

```
student@snd:~/fw$ cat scan
clear
echo SYN scan
sudo hping3 -c 1 -S -p 1234 127.0.0.1
sudo hping3 -c 1 -S -p 1235 127.0.0.1
sudo hping3 -c 1 -S -p 1236 127.0.0.1
read -p "SYN scan complete. Press [ENTER] to continue."
clear
echo FIN scan
sudo hping3 -c 1 -F -p 1234 127.0.0.1
sudo hping3 -c 1 -F -p 1235 127.0.0.1
sudo hping3 -c 1 -F -p 1236 127.0.0.1
read -p "FIN scan complete. Press [ENTER] to continue."
clear
echo UDP scan
sudo hping3 -c 1 -2 -p 1234 127.0.0.1
sudo hping3 -c 1 -2 -p 1235 127.0.0.1
sudo hping3 -c 1 -2 -p 1236 127.0.0.1
echo All scanning complete
student@snd:~/fw$ _
```

TCP SYN scan - no firewall

```
SYN scan of TCP/1234
[sudo] password for student:
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=1234 flags=SA seq=0 win=65495 rtt=7.9 ms

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 7.9/7.9/7.9 ms
SYN scan of TCP/1235
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=1235 flags=SA seq=0 win=65495 rtt=6.9 ms

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 6.9/6.9/6.9 ms
SYN scan of TCP/1236
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1236 flags=RA seq=0 win=0 rtt=3.8 ms

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.8/3.8/3.8 ms
SYN scan complete. Press [ENTER] to continue.
```

Port open

Port closed

TCP FIN scan - no firewall

```
FIN scan of TCP/1234
HPING 127.0.0.1 (lo 127.0.0.1): F set, 40 headers + 0 data bytes

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
FIN scan of TCP/1235
HPING 127.0.0.1 (lo 127.0.0.1): F set, 40 headers + 0 data bytes

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
FIN scan of TCP/1236
HPING 127.0.0.1 (lo 127.0.0.1): F set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1236 flags=RA seq=0 win=0 rtt=4.3 ms

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 4.3/4.3/4.3 ms
FIN scan complete. Press [ENTER] to continue._
```

No response
Port is open

Port closed

UDP scan - no firewall

```
scan of UDP/1234
HPING 127.0.0.1 (lo 127.0.0.1): udp mode set, 28 headers + 0 data bytes

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
scan of UDP/1235
HPING 127.0.0.1 (lo 127.0.0.1): udp mode set, 28 headers + 0 data bytes

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
scan of UDP/1236
HPING 127.0.0.1 (lo 127.0.0.1): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=127.0.0.1 name=localhost
status=0 port=2807 seq=0

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.5/3.5/3.5 ms
UDP scan complete
student@snd:~/fw$ _
```

No response
Port is open

Port closed

Results of scan with no firewall

- TCP scan returned accurate results
  - SYN/ACK for open ports
  - RESET for close port
- TCP FIN scans returned accurate results
  - No response for open ports
  - RESET when port is closed
- UDP scan returned accurate results
  - No response for open ports
  - ICMP port unreachable error when port is closed

Load static rules, reload listeners

```
student@snd:~/fw$ ./fw-static

Current firewall rules
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:1234
ACCEPT     udp  --  anywhere             anywhere             udp dpt:1234
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN
ACCEPT     udp  --  anywhere             anywhere             udp dpts:1024:65535
DROP       all  --  anywhere             anywhere
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
student@snd:~/fw$ ./listen
ncat is now listening on TCP and UDP ports 1234 and 1235 on the loopback interface
ncat is using these process IDs
ncat      8575         student    3u  IPv4 506408      0t0  TCP 127.0.0.1:1234 (LISTEN)
ncat      8576         student    3u  IPv4 505249      0t0  TCP 127.0.0.1:1235 (LISTEN)
ncat      8577         student    3u  IPv4 505248      0t0  UDP 127.0.0.1:1234
ncat      8578         student    3u  IPv4 506425      0t0  UDP 127.0.0.1:1235
student@snd:~/fw$
```

Scan with static rules results

- TCP SYN scan - SYN/ACK from open port
  - Firewall block SYN to non-open ports
  - Correctly identified open port
- TCP FIN scan - RST/ACK from close port
  - Static rules check for SYN=1
  - FIN scan penetrated this rule, correctly identified open port
- UDP scan - ICMP port unreachable from closed port
  - UDP has no state flags, new and established look the same
  - Firewall must let all or nothing through
  - Correctly identified open port

Load stateful rules, reload listeners

```
student@snd:~/fw$ ./fw-inspect

Current firewall rules
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh state NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:1234 state NEW,ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere             udp dpt:1234 state NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             state ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere             state ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere             state ESTABLISHED
DROP       all  --  anywhere             anywhere
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state NEW,ESTABLISHED
student@snd:~/fw$ ./listen
ncat is now listening on TCP and UDP ports 1234 and 1235 on the loopback interface
ncat is using these process IDs
ncat      8810         student    3u  IPv4 510586      0t0  TCP 127.0.0.1:1234 (LISTEN)
ncat      8811         student    3u  IPv4 509285      0t0  TCP 127.0.0.1:1235 (LISTEN)
ncat      8812         student    3u  IPv4 509284      0t0  UDP 127.0.0.1:1234
ncat      8813         student    3u  IPv4 510587      0t0  UDP 127.0.0.1:1235
student@snd:~/fw$ _
```

Scan with stateful rules results

- TCP SYN scan - SYN/ACK from open port
  - Firewall block SYN to non-open ports
  - Correctly identified open port
- TCP FIN scan - RST/ACK from close port
  - Leverage state table to check replies
  - Blocked scan to all open and closed ports
- UDP scan - ICMP port unreachable from closed port
  - Leverage state table to check replies
  - Blocked scan to all open and closed ports

Stateful scan on the wire

```
17:02:12.967607 IP 127.0.0.1.2792 > 127.0.0.1.1234: Flags [S], seq 1487249554, win 512, leng
th 0
17:02:12.967622 IP 127.0.0.1.1234 > 127.0.0.1.2792: Flags [S.], seq 732172888, ack 148724955
5, win 65495, options [mss 65495], length 0
17:02:12.967626 IP 127.0.0.1.2792 > 127.0.0.1.1234: Flags [R], seq 1487249555, win 0, length
 0
17:02:13.031696 IP 127.0.0.1.1925 > 127.0.0.1.1235: Flags [S], seq 1300864157, win 512, leng
th 0
17:02:14.083919 IP 127.0.0.1.2754 > 127.0.0.1.1236: Flags [S], seq 35276343, win 512, length
 0
17:02:26.692319 IP 127.0.0.1.2485 > 127.0.0.1.1234: Flags [F], seq 1027137907, win 512, leng
th 0
17:02:27.720767 IP 127.0.0.1.1447 > 127.0.0.1.1235: Flags [F], seq 1913172986, win 512, leng
th 0
17:02:28.785330 IP 127.0.0.1.2934 > 127.0.0.1.1236: Flags [F], seq 1730661775, win 512, leng
th 0
17:04:29.827988 IP 127.0.0.1.1421 > 127.0.0.1.1234: UDP, length 0
17:04:30.879851 IP 127.0.0.1.1415 > 127.0.0.1.1235: UDP, length 0
17:04:31.936255 IP 127.0.0.1.1546 > 127.0.0.1.1236: UDP, length 0
```

Which firewall "won"?

- Clearly the stateful firewall version
- Only let through SYN to what we exposed
- Blocked FIN to both open and closed
- Handled UDP properly
  - State table compensated for lack of state flags
  - ID established by what appears in the state table

Next week on Fireside Fridays!

- Let's talk about VPN technology!
- No prep needed
- Next week will be just lecture

Wrap up

- Thank you for attending!
- Certs & video will go out by Monday
- If you have any lingering questions, the Discord channel will remain active
  - Also a good chance to socialize with others in the class
  - Have other tips and tricks? Please share with others!
- **Thank you** for sharing your time with us!