Host Defences
- An Intro to Firewalls & Iptables

Greg Horie

Me

… Father & husband

… Tech hobbyist

… Outdoors enthusiast

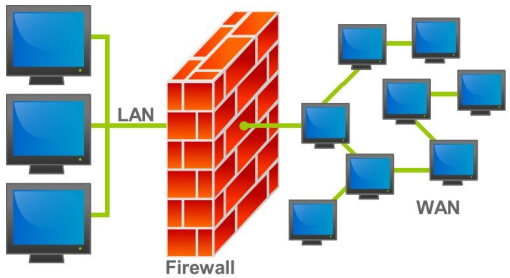
… Ultimate frisbee player

… Board game geek

… Network ops team lead

What is a Firewall?

- A firewall is a security system that filters incoming and outgoing network traffic based on configurable rules.
  - e.g. Filters based on IP address or TCP port.
- Provides protection between a trusted internal network and untrusted external network, such as the Internet.
- Firewalls can be integrated into network routers.
  - e.g. The firewall between your home network and the Internet.
- Firewalls may also reside at the end host-level.
  - Provides protection for servers and PCs.
  - e.g. my laptop from other Starbucks customers



Packet Filters - First Gen Firewalls

- Early firewalls were routers + packet filtering feature.
- Packet filters inspect network packets transferred between computers.
- “Rules” are configured to take action on packets that match given network criteria.
- **Problem** - Packet filtering was not very sophisticated.
  - The packet filters are not network conversation aware.
  - Workarounds employed to enable these conversations.
    - e.g. leave return path open.

Stateful Firewalls - The 2nd Gen

- Performed the same function as a packet filtering firewall, but also maintains a table of network conversations between computers.
  - i.e. The firewall knows the state of the two IP addresses and ports used in a given network conversation.
- **Benefit** - Much better at filtering both inbound and outbound traffic.
  - Increases overall network security.
- **Problem** - Vulnerable to denial-of-service attacks since the connection state table can get overloaded. Prevents legitimate network connections.

Connection State Table

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.9.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.23132.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
219.22.123.32	2112	192.168.1.6	80	Established
210.99.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

Application-Aware Firewalls - The 3rd Gen

- Firewall understands on the behaviour of standard applications and protocols.
- Deeper inspection of traffic based on this application layer understanding.
- Able to detect if an unwanted application or service is attempting to bypass the firewall.
    - e.g. Abusing an allowed port with a disallowed protocol.
- Much harder to perform denial-of-service attacks against a firewall.

Lab Prep Work

- Find a partner - Only one will run iptables.
- Check - Do you have a firewall running?

Try:

```
$ iptables -V
$ ip6tables -V
$ nmap -V
```

CentOS 7

```
$ sudo yum install iptables nmap
```

Ubuntu 18.04

```
$ sudo apt install iptables nmap
```

Exercise 1 - Let’s look at iptables

Try:

```
$ sudo iptables -L
$ sudo ip6tables -L
```

Questions:

- Any difference between IPv4 and IPv6 rules?
- What are the 3 default chains?
- What do the chains do?

Admin Stuff

Save config:

```
$ sudo iptables-save > $HOME/iptables.conf
$ sudo ip6tables-save > $HOME/ip6tables.conf
```

Restore config:

```
$ sudo iptables-restore $HOME/iptables.conf
$ sudo ip6tables-restore $HOME/ip6tables.conf
```

Flush the rules:

```
$ sudo iptables --flush
$ sudo ip6tables --flush
```

Notes:

- Reboot should also restore previous rules

Exercise 2 - Scan your partner

Setup:
One partner will be the **Nmapper** and the other will be the **Firewall**

Nmapper Try:

```
$ nmap -T5 -F <firewall's IPv4>
$ nmap -T5 -F -6 <firewall's IPv6>
```

Firewall Try:

```
$ sudo ncat -v -l -p 80 -k
```

Nmapper Try:

- Retry the nmap scans

Question:

- What did the scan reveal?

Exercise 3 - Block the scans

Firewall - add 2 rules:

```
$ sudo iptables -A INPUT -m conntrack --ctstate \
ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -A INPUT -j REJECT
$ sudo iptables -L
```

Nmapper Try:

```
$ nmap -T5 -F <partner's IPv4>
$ nmap -T5 -F -6 <partner's IPv6>
```

Questions:

- What are the rules doing?
- What did scan show this time? Any behaviour changes vs. Exercise 2?
- Where's the problem? How can this be fixed?
- What generation of firewall is iptables?

Firewall Rules - Basic Components

- **Match Criteria:**
  - Source / destination addresses or prefixes.
  - Source / destination ports (i.e. udp / tcp ports).
- **Actions:**
  - **Drop** - silent (no response to the sender).
  - **Reject** - typically send an ICMP unreachable response.
  - **Allow** - explicitly allow traffic to to pass through the firewall.
- Important Note - These rules are evaluated sequentially.

Exercise 4 - Allow ICMP

Nmapper Try:

```
$ ping <partner's IPv4>
```

Firewall - add 1 rule:

```
$ sudo iptables -I INPUT 1 -p icmp -j ACCEPT
$ sudo iptables -L --line-numbers
```

Nmapper Try:

```
$ ping <partner's IPv4>
```

Questions:

- What did we learn?
- Why `-I 1` rather than `-A`? Why is this necessary?
- Where's the problem? How can it be fixed?
- Homework - Try it with IPv6. Hint - Use `ipv6-icmp`, not `icmp`

Exercise 5 - Allow HTTP

Nmapper Try:

```
$ nmap -T5 -F <partner's IPv4>
```

Firewalll - add 1 rule:

```
$ sudo iptables -I INPUT 3 -p tcp --dport 80 -m state \
 --state NEW -j ACCEPT
$ sudo iptables -L --line-numbers
```

Nmapper Try:

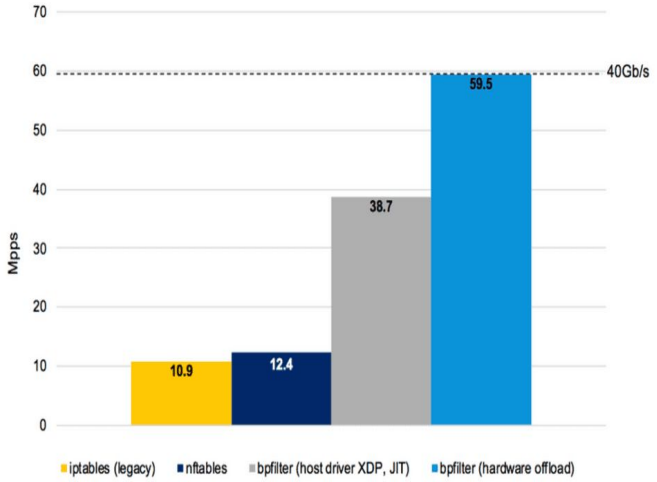
```
$ nmap -T5 -F <partner's IPv4>
```

Questions:

- Why INPUT 3?
- What did we learn?
- Where's the problem? How can it be fixed?
- Homework - Try it with IPv6

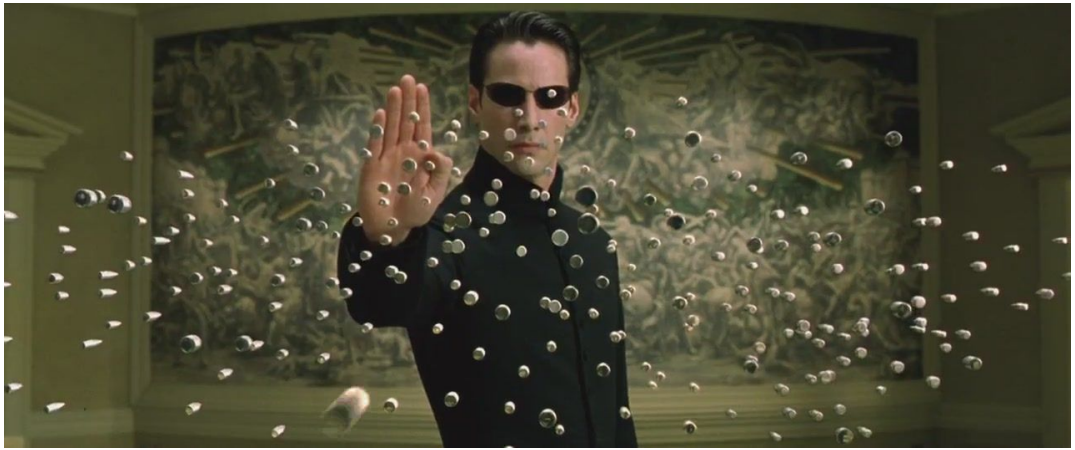
What about nftables & bpfilter?

- Iptables is 20 years old - started in the days of dial-up.
- Iptables has performance problems at scale.
- **nftables** is the current replacement for iptables.
  - Available since Linux kernel 3.13 (2014)
  - Linux community has been slow to adopt
- And now there is a new contender - **bpfilter**.
  - Used by tcpdump and wireshark.
- Proposal to replace kernel part of iptables with bpfilter so iptables will continue to work.
- Bpfilter performance gains are promising:
  - https://cilium.io/blog/2018/04/17/why-is-the-kernel-community-replacing-iptables/
  - https://qmo.fr/docs/talk_20180316_frnog_bpfilter.pdf



Summary

- Firewalls are a useful tool for defending your hosts.
- Iptables is one well-known stateful firewall that has a long history in the Linux community and is still widely used today.
- There are changes coming to the way Linux manages its firewall defences in the future - stayed tuned!

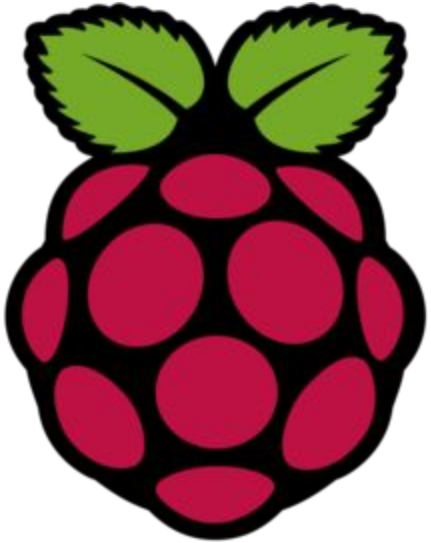


Possible Future Discussions

- Mar 2020 - Wi-Fi
- Apr 2020 - DNS
- More firewalls
  - nftables / bpfilter.
- Vulnerability scanning / Pen-testing
  - e.g. Metasploit, OpenVAS, etc.
- Intrusion detection
  - e.g. Snort, etc.
- Network monitoring
  - e.g. Prometheus, Elastic Stack, Nagios, etc.
- Honey pots
- Container networking
- Other ideas welcome!

VicPiMakers and Others Slack

- Please let us know if you want an invite to this Slack group



Backup Slides

iptables & netfilter

- netfilter is packet filtering framework introduced to Linux kernel 2.4.x.
  - Enables packet filtering, network/port translations and other packet mangling.
- iptables is a generic table structure for the definition of firewall rulesets.
- iptables registers a callback function to a netfilter hook inside the Linux kernel's network stack.
- Each packet that traverses the hook within the network stack will then call this callback function in iptables.