Fireside Fridays

Firewalls

Thanks to our sponsors!

ACTIVE COUNTERMEASURES

Antisyphon Training

Lab requirements for this section

- Today is just lecture
- No lab setup needed

Firewalls

- Specifically designed to control traffic flow
- Comes in two flavors:
  - Packet filter
  - Proxy
- Typically deployed between security zones
- Host firewalls are useful:
  - Public cloud
  - Remote users

Packet filtering Vs Proxies

- Packet filtering
  - Acts like a traffic cop
  - Packet interaction is similar to a router
  - Pass/Fail packets based on rules
- Proxy
  - Implemented on a per application basis
  - Terminate packet stream similar to a server or a client
  - Payload info passed based on app compliance and rules

Packet filter rule options

- **Permit** - Defined pattern can pass through the firewall
- **Drop** - Block the defined pattern from passing through the firewall. Do not return an error to the transmitting system.
- **Reject** - Block the defined pattern from passing through the firewall. Return an error to the transmitting system.
  - If TCP traffic, return a TCP reset
  - All other transports, return an ICMP administratively prohibited
  - Some firewalls let you customize the reply
  - Example: Netfilter can reject with ICMP host unreachables

Static packet filters

- Decisions based on values in each individual packet
- Can not use earlier packets to make decisions
  - No concept of "state"
- Example:
  - Permit all traffic with a source IP of 10.1.1.10
  - Drop all traffic where TCP SYN=1
  - Drop all ICMP traffic where Type=8 Code=0
- Most rudimentary packet filter
- Functionality built into some routers

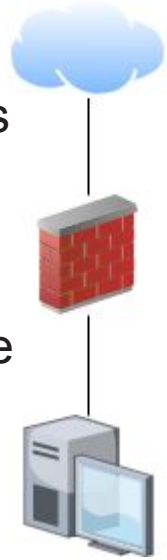
Stateful packet filters

- Functionality of static filters
- Plus the ability to make traffic decisions based on previous packets
  - Implemented via a state table
  - State rules checked after static rules
- Increases RAM and processing requirements but can make more intelligent choices

Static filter example

Policy:
Let all traffic out, let back in replies

Implementation:
Outbound - Permit all packets
Inbound - Permit all packets where
TCP SYN = 0

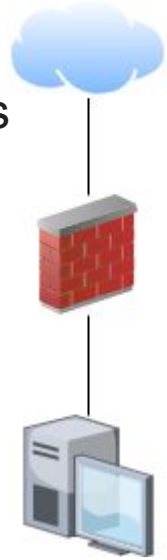


Testing results:
Unsolicited TCP SYN=1 from Internet would be dropped.
Outbound HTTP session would be permitted.
Anything looking like a reply would be let in.
Unsolicited TCP FIN=1 from Internet would be permitted.
Inbound traffic with the source IP of the internal network spoofed may be permitted.

Stateful packet filter example

Policy:
Let all traffic out, let back in replies

Implementation:
Outbound - Permit all packets
Inbound - Drop all traffic
State table - Record header info for passing packets. Permit packets matching state table entries. Source and destination IP and ports can be swapped.



Testing results:
Unsolicited TCP SYN=1 from Internet would be dropped.
Outbound HTTP session would be permitted.
Inbound replies to HTTP session would be permitted.
Inbound traffic with the source IP of the internal network spoofed will be dropped.
Unsolicited TCP FIN=1 from Internet would be dropped.

Stateful packet filter limitations

- Stateful packet filters are limited to working with IP header info only
- This can be a problem for ICMP error packets
- Examples:
  - If a UDP port is unreachable, an ICMP error is returned
  - If TCP/80 traffic is destined for a system that's offline, an ICMP host unreachable error is returned
- This is referred to as "related" traffic

Do I care about related ICMP?

- Two options with static & stateful filtering
  - Permit through all ICMP error traffic
  - Deny all ICMP error traffic
- The first option pokes holes that can be used by C2, probing and other malicious tools
- The second negatively impacts performance
- Stateful inspection addresses this issue

Related traffic example

```
20:29:54.160167 IP (tos 0xc8, ttl 64, id 28828, offset 0, flags [none], proto ICMP (1), leng
th 109)
    167.71.123.148 > 103.145.13.79: ICMP 167.71.123.148 udp port 389 unreachable, length 89
        IP (tos 0x28, ttl 243, id 54321, offset 0, flags [none], proto UDP (17), length 81)
   {103.145.13.79.35883 > 167.71.123.148.389: [no cksum] UDP, length 53}
        0x0000:  45c8 006d 709c 0000 4001 7170 a747 7b94  E..mp...@.qp.G{.
        0x0010:  6791 0d4f 0303 f98d 0000 0000 {4528 0051  g..O........E(.Q
        0x0020:  d431 0000 f311 5b86 6791 0d4f a747 7b94  .1....[.g..O.G{.
        0x0030:  8c2b 0185 003d 0000} 3084 0000 002d 0201  .+...=..0....-..
        0x0040:  0163 8400 0000 2404 000a 0100 0a01 0002  .c....$.........
        0x0050:  0100 0201 0001 0100 870b 6f62 6a65 6374  ..........object
        0x0060:  636c 6173 7330 8400 0000 0000 0a         class0.......
```

Original traffic was UDP going to port 389
Stateful packet filter would permits replies to pass
The decode is an ICMP error because port is close
This is "related" as it's caused by original session but headers info doesn't match
Stateful packet filter would incorrectly drop this traffic (or always pass unchecked)

Stateful inspection

- Stateful packet filtering, plus the ability to pattern match on the payload
- Useful for related traffic
  - For error packets, decode embedded IP headers
  - Match embedded packet against the state table
- Secure complex protocols like FTP
- Usually implemented for functionality
- Can be implemented for security

Packet filter processing

- Stateful inspection implemented on a per application basis
  - Inspect Unreachables, FTP, HTTP, DNS, but no other services
- If the application is not inspected, fall back on transport handling
- Stateful packet filter implemented per transport
  - Stateful for TCP and UDP, but no other transports
- If the transport is not handled statefully, fall back on static filters for traffic control
- Above is usually invisible to the end user

Stateful inspection's dirty secret

- Consider processing order detailed on the previous slide
- Vendor chooses what to implement
- So not all SI firewalls are equal
- Possible that an SI firewall does a large portion of traffic control using only static filters
- You should test to be sure

Netfilter

- Firewall built into Linux
- Management binary is "iptables"
- Supports full stateful inspection
- Arguably the most extensive firewall available
  - Check payloads for patterns
  - Reject with various ICMP type/codes
  - Sooooo much more!
- GUI may dumb it down to stateful packet filtering

Firewall testing

- Always good to verify the firewall policy
- Firewall may pass more traffic than policy
- What you will need:
  - Packet crafting tool - hping3, scapy, etc.
  - A way to open/close TCP/UDP ports on target system
    - ncat, netcat, nc are good choices
  - Generate packets on one side of the firewall and sniff for them on the other side

Network address translation (NAT)

- Permits private addresses to communicate on the Internet
- Provides additional IP privacy
- Works by rewriting IPs and possibly ports in passing packets
- Typically implemented on routers and packet filters

NAT implementations

- Port forwarding
  - All traffic to a specific port is forwarded to a specified IP and port combo
- Many to one NAT
  - Share a single legal IP by rewriting the source port
- One to one NAT
  - Specific mapping between one legal and one private IP address
- NAT address pooling
  - One to one NAT based on a pool of legal addresses
- Most services updated to work well with NAT

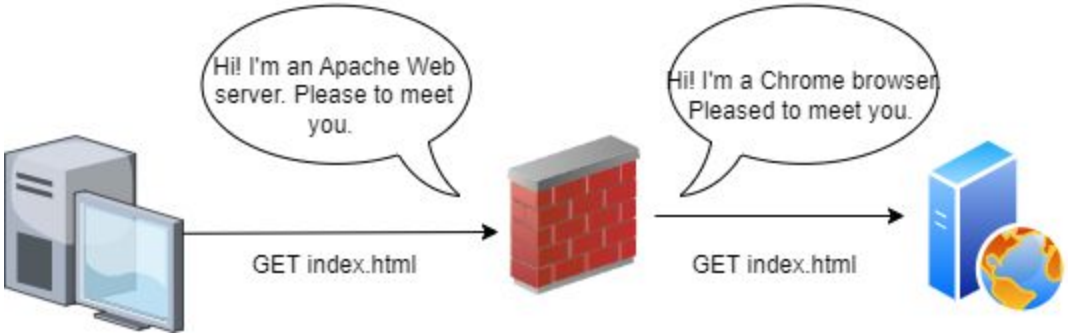
Packet filter firewall logs

- Many vendors label traffic flows based on service associated with well known port
- Examples are HTTP, HTTPS, DNS, etc.
- Most do not inspect for these headers
- So traffic to TCP/80 may be HTTP or it could be anything
  - SSH to TCP/80 will get labeled as HTTP
  - Obfuscated C2 traffic to TCP/443 labeled as HTTPS
- For this reason, many prefer to see raw port numbers

Proxies

- Work at the application/service level
- Stands in for each end of the connection
  - Client thinks the proxy is the server
  - Server thinks the proxy is the client
- Think MITM but "a feature"
- Better able to screen application layer
- Can Improve or negatively impact performance depending on the implementation

A typical implementation



Proxy strengths/weaknesses

- Natural fit for store and forward communications
  - SMTP is a good example
- Can experience compatibility issues
  - Proxy emulates Chrome browser but user running Firefox
- Single point of security implementation
- Single point of security failure & performance choke point
- Greater security & complexity over packet filtering

Reverse proxies

- Use to control inbound sessions
- Typical implementation - front end for web server
- Can provide both better performance and security
- Available as third party services
- Cloudflare is a popular option

Proxy implementations

- Transparent, inline to traffic flow
- Semi-transparent, redirect via DNS
- Port forwarding via packet filtering firewall
- Client configured to send through proxy
- Client agent responsible for forwarding traffic
- Traffic to a non-transparent proxy may be tunneled through another protocol like SOCKS

Learn about networking

- Cisco has some great courses
- Check out NetworkChuck on YouTube
- NIST guidelines on firewalls

https://csrc.nist.gov/pubs/sp/800/41/r1/final

Next week on Fireside Fridays!

- Hands-on walkthrough of testing a packet filtering firewall
- Compare traffic handling of different types
- You will need to prep to follow along
  - Modern Linux distro
  - sudo access to run iptables commands
  - Run the steps shown on the next slide

Steps to prepare

```
sudo apt update

sudo apt -y install wget 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$ _
```

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!