

RHCSA

BOOT CAMP

Network Security

TCP WRAPPERS

- TCP Wrappers was originally written to provide host based access control for services which did not already include it.
- It was one of the first “firewalls” of a sort. :)
- Before you can set up tcp_wrappers on a service, you have to check if the service supports it...

CHECKING TCP WRAPPER SUPPORT

- Determine which binary the application runs as. Check `init` script or:

```
# which sshd
```

```
/usr/sbin/sshd
```

- Check for `libwrap` support in the binary.
- If you see `libwrap` support in the output, then you can configure access to the service with `tcp_wrappers`.

```
# ldd /usr/sbin/sshd | fgrep wrap
```

```
libwrap.so.0 => /usr/lib/libwrap.so.0 (0x009c5000)
```


TCP WRAPPER OPERATION

- If an application is compiled with support for `tcp_wrappers`, that application will check connection attempts against the `tcp_wrappers` configuration files:
 - `/etc/hosts.allow`
 - `/etc/hosts.deny`

TCP WRAPPER OPERATION

- These files are parsed in the following order:
 - The file `/etc/hosts.allow` is consulted. If the configuration of this file permits the requested connection, the connection is immediately allowed.
 - The file `/etc/hosts.deny` is consulted. If the configuration of this file does not permit the requested connection, the connection is immediately refused.
 - If the connection is not specifically accepted or rejected in either file, the connection is permitted.

TCP WRAPPER CONFIGURATION

- The basic syntax for these files is:
 - `<daemon>: <client>`
- For example, to disable ssh connections from 192.168.2.200, add this line to `/etc/hosts.deny`:
 - `sshd: 192.168.2.200`

IPTABLES

- IPTables works at the kernel level, just above the network drivers, to provide several useful features.
 - Extremely powerful and flexible Layer 2 filtering engine.
 - NAT support
 - Port forwarding
 - And many more

IPTABLES RULE MATCHING

- The IPTables configuration is parsed from top to bottom.
- IPTables will respond based on the first match that it finds.
- If there is no specific match, the chain policy will apply.

IPTABLES TOOLS

- **iptables:** View/modify current firewall rules
- **iptables-save:** Script to save current firewall rules for use with iptables-restore
- **iptables-restore:** Restores iptables-save format firewall rules - useful to set up firewalls at boot
- Consider iptables init script for save/restore. Config file:

`/etc/sysconfig/iptables`

IPTABLES RULES

- When creating a new rule, considerations include:
 - What chain should the rule apply to?
 - What is the traffic pattern to look for?
 - What should happen with the traffic?

IPTABLES CHAINS

- **INPUT**

- This chain is responsible for filtering traffic destined for the local system.

- **OUTPUT**

- This chain is responsible for handling outbound traffic.

- **FORWARD**

- This chain is responsible for controlling traffic routed between different interfaces.

IPTABLES RULES

- Below are a few examples of possible IPTables match criteria:

- incoming interface **-i**
- protocol **-p**
- source ip address **-s**
- destination ip address **-d**
- destination port **--dport**

IPTABLES RULES

- Finally, some examples of what to do with traffic when matched:
 - **DROP** Do not deliver, do not respond
 - **REJECT** Do not deliver, send reject notice
 - **ACCEPT** Deliver
 - **LOG** Just log the packet

IPTABLES RULES

- So to summarize the syntax:
 - `iptables`
 - What chain should the rule apply to?
 - `-A INPUT`
 - What is the traffic pattern to look for?
 - `-s 192.168.2.100`
 - What should happen with the traffic?
 - `-j REJECT`

LAB

1. Using `iptables`, configure your web server to NOT accept connections from the `192.168.1.0/24` network, EXCEPT for the ip address of whomever is sitting next to you. Work together to test the firewall settings, and remember, **web** server. :)
2. Browse through the man page for `iptables`.
3. Use `iptables` to allow `ssh` from the classroom network only.


```
slideshow.end();
```