Implementing Firewalls with Firewalld

Andrew Mallett
Linux Author and Trainer

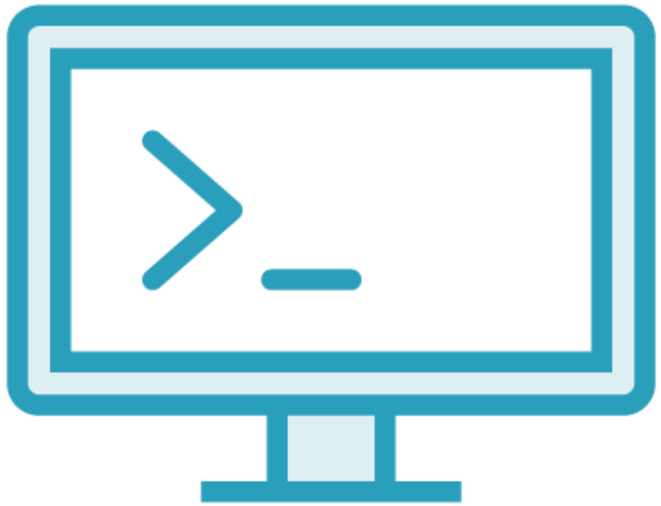
@theurbanpenguin www.theurbanpenguin.com

Overview

Securing your System via Host Based Firewall

- firewalld
  - firewall-cmd
  - Firewall zones
  - Configuring rules

Lab Systems



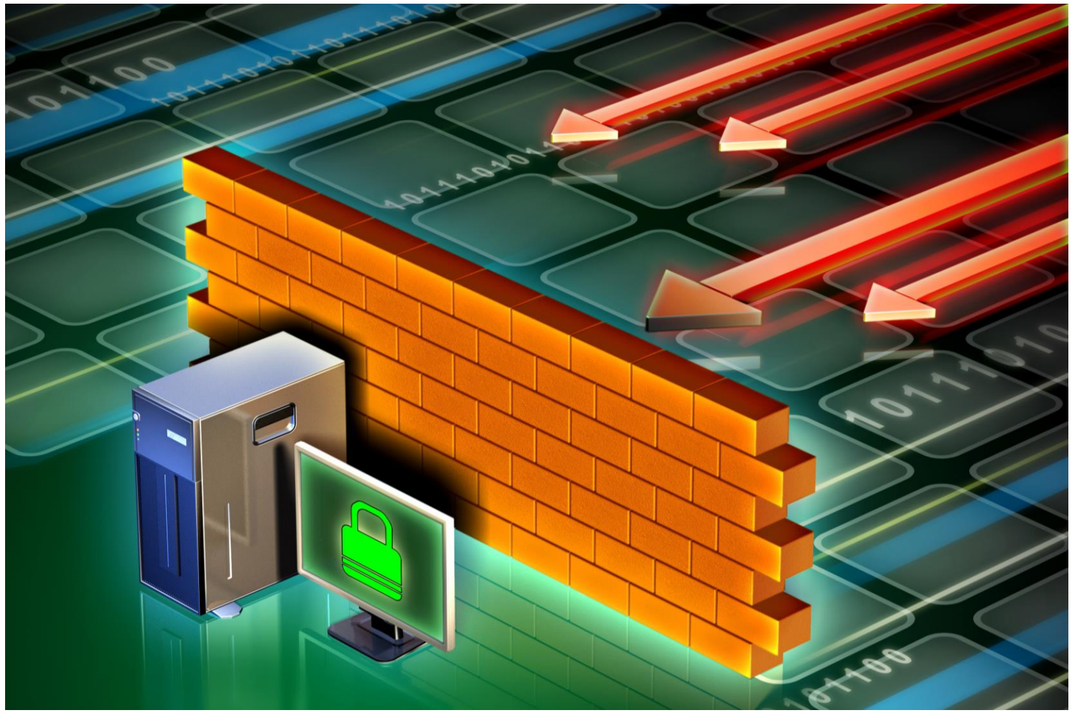
AlmaLinux 8

- firewalld

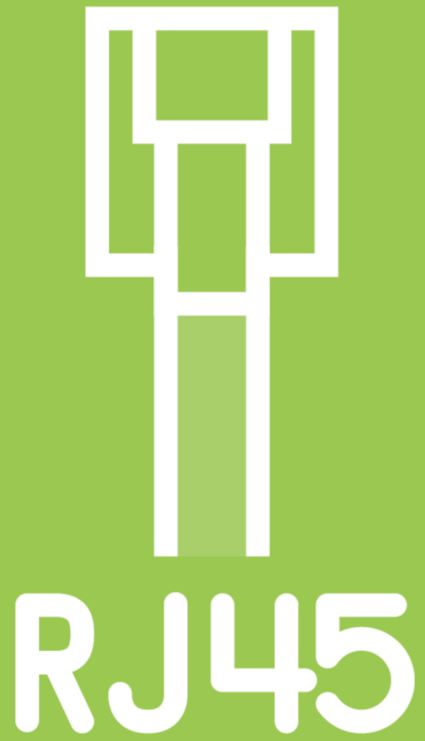
Ubuntu 20.04

- client

Host-Based Firewalls



Incoming traffic to your system may be safe, you may have border routers that control incoming traffic; but is this reason enough not to further secure your system with a host-based firewall?



RHEL Firewalls

The default firewall in RHEL and variants is managed via FirewallD. In RHEL 7 and variants the backend was IPTables; in RHEL 8 and variants the backend is NFTables

```
$ sudo -i

# firewall-cmd --state
running

# firewall-cmd --list-all
...
```

Managing Firewalld

To manage the backend firewalld firewall we use the command firewall-cmd as the root user.

```
# firewall-cmd --list-all --permanent
...
# ls /usr/lib/firewalld
# ls /etc/firewalld
```

Permanent and Runtime

The default operations target the runtime configuration, we have persistent storage files to persist settings that we need to continue. The default settings come from /usr/lib/firewalld, edit we make are stored in /etc/firewalld

Demo

Let's begin by examining the firewall configuration:

- Configuration files
- firewall-cmd

```
# yum install httpd

# systemctl enable --now httpd

# (from remote) curl 192.168.33.11

# firewall-cmd --add-service=http

# (from remote) curl 192.168.33.11

# firewall-cmd --list-all ; firewall-cmd --list-all --permanent

# firewall-cmd --runtime-to-permanent
```

Adding Services

Many common services will have an XML file representing their needs, we add these files to the configuration using --add-service. When tested, we can persist the settings with --runtime-to-permanent

Demo

Let's begin by securing the firewall to our needs:

- firewalld.conf
- removing services
- adding services

```
# firewall-cmd --remove-service=http

# (from remote) curl 192.168.33.11

# firewall-cmd --add-service=http --zone=internal

# firewall-cmd --list-all --zone=internal

# firewall-cmd --add-source=192.168.33.0/24 --zone=internal

# (from remote) curl 192.168.33.11
```

Sources and Zones

Sources represent inbound connections. We can add a source to a zone to trust or block connections

Demo

We now modify our rule

- Looking at zones
- We allow access from a source address range

```
# firewall-cmd --info-service=http

# firewall-cmd --add-port=443/tcp --timeout=5

# cp /usr/lib/firewalld/services/http.xml /etc/firewalld/services/

# vim /etc/firewalld/services/http.xml
...
<port protocol=”tcp” port=”443”/>
...

# firewall-cmd --reload

# firewall-cmd --info-service=http
```

Services and Ports

We can use ports in place of services. It may also be more convenient to add ports to an existing service. Timeouts can also be used with firewall rules, the units, default to seconds.

Demo

We now investigate:

- Ports
- Timeouts
- Service XML files

Summary

In this module we have introduced host-based firewalls and firewalld:

- firewall-cmd
- runtime and permanent settings
- Services, Zones, and Ports

Implement Firewalls with UFW