



freja.nordsiek@gwdg.de

Freja Nordsiek

Firewalls

Table of contents

1 Firewall Concepts

2 Routing

3 Firewalld

What is a Firewall?

Chooses which incoming and outgoing packets to accept, drop, forward, and/or modify.

- Reduce attack surface
 - ▶ Prevent exposing services to the wrong machines
 - e.g. don't let the internet use your squid proxy
 - Some services don't have builtin ways to restrict allowed IP addresses
 - ▶ Limit the rate of packets or connections (preven DoS)
 - ▶ Restrict outgoing packets
- Notify admins of suspicious packets or high rates
- Redirect network traffic
 - ▶ Routing, possibly with rewriting (NAT)
- Ensure that responses from the outside reach their clients inside

General Architecture

Firewall

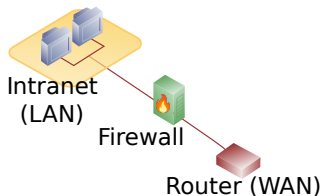


Figure: Source: [1] (heavily modified)

- Network packets pass through the firewall
- A firewall can be local to the computer system
- Packets can be accepted, rejected, forwarded
- Packets can be modified and redirected...

DMZ = Demilitarized Zone

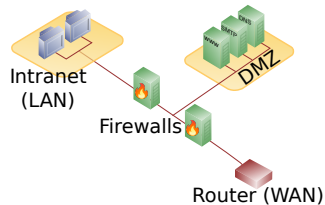


Figure: Source: [1] (modified)

- Typically employs two firewalls
- Exposes externally facing services to untrusted network
- Protects the local network by isolating Internet and private network

[1] [https://en.wikipedia.org/wiki/DMZ_\(computing\)](https://en.wikipedia.org/wiki/DMZ_(computing))

Common Packet Operations

- Accept
- Drop (delete and don't notify sender)
- Reject (delete but notify sender)
- Count
- Log
- Rewrite (e.g. NAT via masquerade)

Directly Reach

Devices can only directly reach addresses in their networks (subnets)

Example

A machine with the following networks:

- 192.168.0.3/24

- 10.3.39.10/18

Can only directly reach

- 192.168.0. [0-255]

- 10.3. [0-63]. [0-255]

Remember the caching proxy (Squid) you setup Monday so the compute nodes could reach the internet?

Gateways

Device that receives packets on one network meant for a different network and sends them to that network or the next gateway.

■ Example

▶ A gateway with the following networks:

- 192.168.0.3/24
- 10.3.39.10/18

▶ Receives packet from 192.168.0.10 directed to 10.3.38.4

▶ Resends it on the 10.3.39.10/18 network

■ Possible because physical layer has different addressing (MAC vs. IP)

■ 192.168.0.10 would have a route to 10.3.0.0/18 like

```
10.3.0.0/18 via 192.168.0.3 dev lan0 proto static metric 400
```

Routing

■ Source routing

- ▶ Gateway forwards packet from one network to another unmodified
- ▶ Simple and fast
- ▶ Responses require the target to have a route back to the source
- ▶ Great between networks you control

■ NAT (Network Address Translation) routing

- ▶ Gateway rewrites packet to make it look like gateway is the sender
- ▶ Gateway has to keep track of connections to redirect and rewrite responses
- ▶ Hides network addresses of the source network
- ▶ Requires port forwarding to expose services in the network being NAT-ed
- ▶ Not needed for IPv6 unless you need to hide network addresses
- ▶ Good for connections to networks you don't control

Making Firewalls on Linux

- Low level (nftables, successor to iptables)
 - ▶ Full control and no limitations, but unwieldy
 - ▶ Often use high level tools over the top
- firewalld (<https://firewalld.org>)
 - ▶ High level firewall
 - ▶ Easy to use with XML config files
 - ▶ Has pre-made configurations for many services
 - ▶ Can work with other tools like Network Manager
- Common alternatives we won't be working with today:
 - ▶ UFW (<https://wiki.ubuntu.com/UncomplicatedFirewall>)
 - ▶ shorewall (<https://shorewall.org>)

Structure

- Zones – `/etc/firewalld/zones/*.xml`
 - ▶ Split IP address range into one or more zones
 - ▶ Each zone file controls how incoming packets are handled
- Policies – `/etc/firewalld/policies/*.xml`
 - ▶ Rules for packets going from one zone to another, or outgoing packets
- Services – `/etc/firewalld/services/*.xml`
 - ▶ Aliases for the ports, protocols, etc. used by a service
 - ▶ Allow referring to services by name in Zones and Policies

Zone

```
<?xml version="1.0" encoding="utf-8"?>
<zone target="%%REJECT%%">
  <short>Cluster</short>
  <description>Cluster network</description>

  <interface name="lan0"/>
  <source address="192.168.0.0/16"/>

  <service name="ssh"/>
  <port port="631" protocol="tcp"/>
  <port port="2030-2038" protocol="udp"/>

</zone>
```

- `zone target=""` defines the default rule
 - ▶ "ACCEPT", "DROP", or "%%REJECT%%" (default)
- `interface` and `source` determine what addresses are part of zone
- `service` and `port` determine exceptions to the default rule
- Supports much more

Service

`foo.xml` can be referred to as "`foo`" in Zones and Policies.

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>foo awesome</short>
  <description>Foo's awesome service</description>

  <port protocol="tcp" port="6800"/>
  <port protocol="udp" port="7200-7299"/>

  <include service="bar"/>
</service>
```

- `port` specifies the ports it listens on
- `include` specifies other services that this depends on and should be accepted, dropped, or rejected together
- There are other options for services that need helpers like FTP

Policy

```
<?xml version="1.0" encoding="utf-8"?>
<policy target="REJECT">
  <short>internet</short>
  <description>Outgoing traffic to the internet</description>
  <ingress-zone name="HOST"/>
  <egress-zone name="public"/>
  <service name="http"/>
  <port port="391" protocol="tcp"/>
</policy>
```

- `zone target=""` defines the default rule
 - ▶ "ACCEPT", "DROP", "REJECT", or "CONTINUE" (default)
- `ingress-zone` and `egress-zone` define source and destination zones
 - ▶ "HOST" means machine itself
 - ▶ "ANY" means all zones, but not the machine itself
- `service` and `port` define exceptions to the default rule

Routing

- In Zone file for between networks in same Zone
 - ▶ Add `<forward/>`
 - ▶ For NAT, additionally add `<masquerade/>`
- In Policy file for between different non-HOST Zones
 - ▶ forwarding is implied
 - ▶ For NAT, additionally add `<masquerade/>`