Firewalld, netfilter and nftables

Thomas Woerner
Red Hat, Inc.

NFWS 2015
June 24

firewalld

- Central firewall management service using D-Bus
- Supports
  - IPv4: iptables
  - IPv6: ip6tables
  - Bridges: ebtables
- Sends signals for all actions over D-Bus
- Integration
  - NetworkManager
  - libvirt
  - docker

Configuration

- Completely adaptable, XML config files
- Run-time and persistent configuration separation
- Default and adapted configuration files
  - Default usable as fallbacks
- Services
- Zones
- Direct interface

Services

- Options
  - Port (ranges) with protocol
  - Netfilter helper modules
  - Destination address (range) for IPv4 and/or IPv6
- Nearly 70 built-in services
- Adaptable over D-Bus, config tools and files

Service Examples

dns

```
<service>
  <port protocol="tcp" port="53"/>
  <port protocol="udp" port="53"/>
</service>
```

https

```
<service>
  <port protocol="tcp" port="443"/>
</service>
```

tftp

```
<service>
  <port protocol="udp" port="69"/>
  <module name="nf_conntrack_tftp"/>
</service>
```

dhcpv6-client

```
<service>
  <port protocol="udp" port="546"/>
  <destination ipv6="fe80::/64"/>
</service>
```

Zones I

- Options
  - Services
  - Ports (ranges) with protocols
  - Rich rules
  - Internet Control Message Protocol (ICMP) blocks
  - Masquerading
  - Port/packet forwardings
- Options can be enabled for a limited time frame
- Built-in zones: `block`, `dmz`, `drop`, `external`, `home`, `internal`, `public`, `trusted`, `work`
- Completely adaptable

Zones II

- Zone is similar to a complete firewall
- Initial default: public (FedoraWorkstation, FedoraServer)
- One zone per connection (NM, network service)
  - `ZONE=<name>` in ifcfg file or NM configuration
- One zone per interface or source address (range)
- Internal firewall rule ordering according to rule action
  - log→deny→allow

Zone Examples

public

```
<zone>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
</zone>
```

drop

```
<zone target=”DROP”>
</zone>
```

custom

```
<zone>
  <interface name="em2"/>
  <source address="10.0.1.0/24"/>
  <service name="ssh"/>
  <service name="ipp-client"/>
  <service name="dhcpv6-client"/>
  <rule><protocol value="ah"/><accept/></rule>
</zone>
```

Rich Rules

- Source address (range): optional
- Destination address (range): optional
- One Element
  - Service, port, protocol, icmp-block, masquerade, forward-port
  - Limit: optional
- Logging: optional
  - Log and/or audit
  - Limit: optional
- One Action: accept, reject, drop
  - Limit optional

Rich Rule Examples

Allow new IPv4 and IPv6 connections for service ftp and log 1 per minute using audit

```
rule service name="ftp" log limit value="1/m" audit accept
```

Allow new IPv4 connections from address 192.168.0.0/24 for service tftp, log 1 per minute using syslog

```
rule family="ipv4" source address="192.168.0.0/24" service name="tftp"
log prefix="tftp" level="info" limit value="1/m" accept
```

New IPv6 connections from 1:2:3:4:6:: to service radius are rejected and logged at a rate of 3 per minute. New IPv6 connections from other sources are accepted, saved permanently, reload to activate

```
rule family="ipv6" source address="1:2:3:4:6::" service name="radius" log
prefix="radius" level="info" limit value="3/m" reject
rule family="ipv6" service name="radius" accept
```

Direct Interface

- More complex rules, globally, not in zones
- Config file: `/etc/firewalld/direct.xml`
- Chains
  - For use with rules, same as in netfilter
- Rules
  - ip*tables/ebtables syntax
  - priority for rule ordering
  - added to _direct chains for netfilter built-in chains or own chains
- Passthrough rules (For highly experienced users)
  - Used by libvirt, docker

Direct Interface Examples

Create custom chain blacklist in raw table for IPv4, log and DROP

```
firewall-cmd --direct --add-chain ipv4 raw blacklist
firewall-cmd --direct --add-rule ipv4 raw blacklist 0 -m limit --limit 1/min -j LOG --log-prefix “blacklist: “
firewall-cmd --direct --add-rule ipv4 raw blacklist 1 -j DROP
```

Add black listed IPv4 address to blacklist

```
firewall-cmd --direct --add-rule ipv4 raw PREROUTING 0 -s 192.168.1.0/24 -j blacklist
```

Persistent direct configuration

```
<direct>
  <chain ipv="ipv4" table="raw" chain="blacklist"/>
  <rule ipv="ipv4" table="raw" chain="PREROUTING" priority="0">-s 192.168.1.0/24 -j blacklist</rule>
  <rule ipv="ipv4" table="raw" chain="blacklist" priority="0">-m limit --limit 1/min -j LOG --log-prefix "blacklist: "</rule>
  <rule ipv="ipv4" table="raw" chain="blacklist" priority="1">-j DROP</rule>
</direct>
```

D-Bus Interface

- Full featured
  - Run-time and persistent configuration
  - Zones, services, icmp types
  - Direct interface
  - Lockdown
- Signals for all changes
- Used by
  - Config tools
  - Other projects: NetworkManager, libvirt, docker

Netfilter use in projects

- Parsing of existing rule set complex, adding rules to the first line in the builtin chains very common
- Not using of the wait option initially
- Adding rules or rule sets using ipXtables calls, mostly no cleanup of old rules
- Flushing of rule set before adding own rules
- Using reject rules in the end of own rule set

Netfilter use in firewall managers, issues

- Rule set is mostly cleared on start
- Limitation: Only rule positions, no ids
- Comments usable as a work around for ids, but results in less readable output
- Ordering of rules is important, decides on effect, no way to pin rules to positions
- No signal to user land for changes with in rule set
- Not possible to get rule counters for rules besides parsing whole rule set for statistics

Netfilter use in firwalld I

- iptables, ip6tables and ebtables calls
- Uses set of chains for zones, created only if used
- Orders rules internally in _log, _deny and _allow sub chains
- Possible speedup using -restore calls, but limited

Netfilter use in firwalld II

```
*filter
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -j INPUT_direct
-A INPUT -j INPUT_ZONES_SOURCE
-A INPUT -j INPUT_ZONES
-A INPUT -p icmp -j ACCEPT
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j OUTPUT_direct
-A INPUT_ZONES -i em1 -j IN_block
-A INPUT_ZONES -j IN_block
-A IN_block -j IN_block_log
-A IN_block -j IN_block_deny
-A IN_block -j IN_block_allow
-A IN_block -j REJECT --reject-with icmp-host-prohibited
-A IN_public -j IN_public_log
-A IN_public -j IN_public_deny
-A IN_public -j IN_public_allow
-A IN_public_allow -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
```

(simple use case with block as default zone and public used for the em1 interface, forward chains left out)

nftables I

- Good: Monitor
  - maybe several monitors needed to simplify parsing
- No fixed base chain names, distributions already using different name sets
  - Hard to use for cross-distribution projects
- No fixed order of ip, ip6 and inet filter table handling
  - Creation order important?
  - Different behaviour possible
- Base chain priorities unclear, why different ranges?
- Base chains with different priorities increasing complexity

nftables II

- Only accept and drop as default base chain policy, final reject line required
    - Chains with lower priority not used
- Question: Estimated time frame for use in production

Wish list

- Full features nftables library with same behaviour and checks as the command line tool
  - also for ipXtables compat mode
- Full featured xtables library if nftables release
- Fixed base chain names
- Ids for rules
- Get counters for rules (and chains) without parsing rule set (for statistics mode) at best by id
- Checksums for chains and tables or last modified info
- Write access limitations, unlimited read access
- Way to pin rules to fixed positions

Future Plans

- Statistics and tracing mode
- ipset support
- nftables support (smooth transition for users)
- Security environments (zone interaction)
- Direct rules in zones

More Information

- Web:
  - http://www.firewalld.org/
- Documentation: http://fedoraproject.org/wiki/FirewallD
- Man pages for firewalld, firewalld.zone, firewalld.service, firewalld.direct, firewalld.richlanguage, firewall-cmd, ..
- Source Repository: git://github.com/t-woerner/firewalld
- irc channel: #firewalld on freenode
- Mailing lists:
  - firewalld-users@lists.fedorahosted.org
  - firewalld-devel@lists.fedorahosted.org