FIREWALLD Structured Ingress and Egress Filtering
29 August 2016, Stephen Satchell, spamfilter@satchell.net
Version 2

PROBLEM

Many people on smaller networks, particularly ones who are not using Cisco, Juniper, or other heavy-duty router equipment, need to practice Safe Internet. As of 28 August 2016, the FirewallD firewall for Red Hat Enterprise and CentOS, among others, is a good start.; what it is lacking is more robust ingress filtering, and a complete lack of egress filtering. As a secondary consideration, it also lacks control of packets forwarded across zones.

This document is IPv4-centric. I believe that most of the practices described here, particularly forged-traffic blocking, can also be appled to IPv6. The reason this is so past-looking is that I'm documenting what I have built with my ADSL/Cablemodem firewall. A casual inspection of some of the recent literature about IPv6 best practices shows that most of the rules for IPv4 also apply to IPv6.

INGRESS FILTERING FOR IPv4, PER INTERFACE (on INPUT chain)

- (default enabled) Block TCP SYN flood
- (default enabled) Block ICMP flood
- (default enabled) Block small services (tcp 0-19, udp 0-19)
- Block forged inbound SOURCE address
  - (default empty) EXCEPTION: allow specific unicast/multicast source addresses and destination port range)
  - (default empty) EXCEPTION: allow specific unicast/multicast source prefixes and destination port range
  - (default disable) EXCEPTION: allow prefixes on local interfaces (needs study as to whether such an exception is needed, as I believe forwarded packets do not traverse the INPUT chain for the interface)
  - (default empty) specific source addresses/prefixes [used for blacklisting]
  - machine interface ip address(es) [spoofing ME]
  - machine interface network address(es), broadcast address(es)
  - (default disabled) routeable network prefixes (nets 1-9, 11-99, 101-126, 128-168, 170, 171, 173-191, 193-198, 191-197, 199-223, and subnets of other ranges not reserved in 100, 169, 172, 192, 198, 203)
  - reserved unicast network prefixes (portions of nets 0, 10, 100, 127, 169, 172, 192, 198, 203)
  - reserved multicast network prefixes (nets 224-255)
- Block unwanted inbound DESTINATION addresses
  - (default empty) EXCEPTION: allow specific unicast/multicast addresses (and destination port range)
  - (default empty) EXCEPTION: allow specific unicast/multicast prefixes (and destination port range)
  - machine interface broadcast addresses
  - reserved multicast network prefixes (nets 224-255)
- Allow services-related ports and protocols
- Block the rest by default

The source address filtering described above is to prevent the local computer from participating in packet amplification attacks or redirection attacks from packets coming in from the Internet, as well as denial-of-service attacks caused by carefully crafted bogus inbound packets. Nothing can be done from this firewall to prevent inband bandwidth overload, but the filters will prevent the system from adversely affecting the outbound bandwidth usage.

Unicast and multicast are treated separately, because many systems don't make use of multicast. This is particularly important in source-address filtering.

Destination address filtering is considerably more sparse, pretty much blocking the subnet and allnet broadcast addreses.

Why does the exception capability allow for port ranges? One example: Dropbox LAN Sync use subnet and allnet broadcasts in UDP to “announce” themselves, on port 17500/UDP. From a wireshark capture, the Dropbox protocol sends to 255.255.255.255 and the subnet broadcast address (as observed, 10.1.1.255/24). The allnet broadcast should't be there at all (and that feeds the egress filtering discussion), but the subnet broadcast could be useful and should be allowed with the smallest pinhole possible.

EGRESS FILTERING FOR IPv4, PER INTERFACE (on OUTPUT chain)

- Block forged SOURCE addresses
  - (default empty) EXCEPTION: allow specific unicast/multicast addresses (and destination port range)
  - (default empty) EXCEPTION: allow specific unicast/multicast prefixes (and destination port range)
  - NOT IN LIST of machine interface ip address(es)
- Block unwanted DESTINATION addresses
  - (default empty) EXCEPTION: allow specific unicast/multicast addresses (and destination port range)
  - (default empty) EXCEPTION: allow specific unicast/multicast prefixes (and destination port range)
  - (default empty) specific addresses/prefixes (used for blacklisting)
  - (default disabled) routeable network prefixes (nets 1-9, 11-99, 101-126, 128-168, 170, 171, 173-191, 193-198, 191-197, 199-223, and subnets of other ranges not reserved in 100, 169, 172, 192, 198, 203)
  - reserved unicast network prefixes (portions of nets 0, 10, 100, 127, 169, 172, 192, 198, 203)
  - reserved multicast network prefixes (nets 224-255)
- Allow service-related ports, protocols
- Allow inherited service-releated ports, protocols from another zone
- (default disabled) Allow all ports, protocols
- Block the rest by default

The egress filtering is slightly different, as the goal is different. It keeps bad packets from being sent to the world. For example, remember that Dropbox sends UDP packets to the Internet broadcast address? If that's allowed to go out, it can trigger alarms in the upstream carrier...or wreak havoc if the upstream is so clueless as to propagate the broadcase!

The option to allow all ports and protocols duplicates the existing behavior of firewalld.

Egress filtering would affect packets forwarded from other zones.

EXAMPLE

For a home or SOHO user wanting to build a Linux netfiler firewall between the Internet and the inside network, s/he would have a two-NIC computer loaded with RHEL7/CentOS7.

- One interface would be designated “Internet”, bound to zone “external”, connected to the Internet upstream
- Another interface is designated “InsideNet”, bound to zone “trusted”, connected to the local network (LAN)
- Both interfaces have fixed IP addresses; the interfaces can have multiple Ips assigned
- Neither interface uses DHCP to obtain an address
- There is a DHCP server running in the firewall for use by the computers connected via “InsideNet” on the “trusted” zone

Public Zone:

```
external (active)
  interfaces: enp0s25 [Internet]
  sources:
  services-inbound: ssh
  services-outbound: ntp !dhcp
  inherit-outbound: trusted (dns ftp http https ipsec mdns ntp pop3 pop3s imap imaps smtp ssh ntp)
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
  ingress-filtering: yes
  egress-filtering: yes
```

The only service enabled inbound from the Internet is SSH. (tcp_wrappers aka tcpd(8) can, and should, be used to limit access to this service.)

All ingress and egress filtering is enabled for this interface, so that this system is a “good neighbor”, and also is immune to some “bad-neighbor” actions; in particular, this system will not participate in any amplification attacks and certain flooding attacks.

On the services list for this zone, note that the sysadmin explicitly disables DHCP. In our example, the Internet interface has fixed IP address(es), and does not use DHCP to obtain the addresses. This prohibition overrides any inheritance of services from the trusted zone. For those firewalls that use DHCP to the upstream, the sysadmin would *not* block dhcp outbound, and in fact would explicitly enable it. In either case, dhcp would be disabled inbound in the “public” zone so that the system doesn't inadvertently supply addresses to rogue DHCP requests.

> Questions: should service inheritance be limited to a sysadmin-selected selection of source zones? Such a limitation would allow a “dmz” zone's services to *not* be inherited by the “public” zone if that is not desired.
>
> Generalizing: how does a sysadmin control forwarding of packets from zone to zone? Would such a express selection of source zones also apply to the packets forwarded from zone to zone, irrespective of masquarade? In other words, if I have multiple inside LANs, should I be able to limit connections from LAN A to LAN B. and have separate limits from LAN B to LAN A?

Trusted Zone:

```
trusted (default, active)
  interfaces: enp3s0 [LocalNet]
  sources:
  services-inbound: dhcp dns ftp http https ipsec mdns ntp pop3 pop3s imap imaps smtp ssh ntp
  services-outbound:
  inherit-outbound:
  masquerade: yes
  forward-ports:
  icmp-blocks:
  rich rules:
  ingress-filtering: no
  egress-filtering: no
```

In the “trusted” zone, only those ports and protocols for services being used by the computers on the local network to make outbound connections to Internet-connected servers/devices are enabled inbound to the firewall through this interface. All other inbound ports and protocols are disabled, and all outbound services are disabled. In this example, ingress and egress filtering are disabled; there is no reason a sysadmin can't enable ingress/egress filtering if s/he wants to, particularly if the sysadmin doesn't control some or all of the “inside” computers..

In this zone, we have enabled masquerade. This creates a bridge between the “inside” connection and a corresponding “outside” connection, with the packet traversing the netfilter FORWARD chains. (This facility is present in the existing implementation of firewalld.) In short, this means that all connections made by the inside computers appear on the external interface to the Internet with the IP address of the Internet interface. This is why another zone needs to be able to inherit the inbound-service connections, so that there is no mismatch between the egress filtering on the external interface and the ingress/service filtering on the trusted interface.