FIREWALLD Structured Ingress and Egress Filtering

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 is lacking is more robust ingress filtering, and a complete lack of egress filtering.

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.

INGRESS FILTERSING FOR IPv4, PER INTERFACE

- (default enabled) TCP SYN flood abatement
- (default enabled) ICMP flood abatement
- (default enabled) Block small services (tcp 0-19, udp 0-19)
- Block forged inbound SOURCE address
  - (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)
  - machine interface ip address(es) [spoofing ME]
  - machine interface 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
- Block the rest by default

The source address filtering is to prevent the local computer from participating in packet amplification attacks or redirection attacks, as well as denial-of-service attacks caused by carefully crafted packets. 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 prefix broadcasts in UDP to “announce” themselves, on port 17500/UDP. From a wireshark monitor, the Dropbox protocol send 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

- 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

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

EXAMPLE

For a home or SOHO user wanting to build a 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” and connected to the Internet upstream
- Another interface is designated “InsideNet” bound to zone “trusted” connected to the local network
- Both interfaces have fixed IP addresses on them
- 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 services that would be enabled inbound from the Internet is SSH. (tcpwrapper can be used for 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.

The outbound services enabled to the Internet are inherited from the “trusted” zone, except that DHCP is explicitly disabled and overrides the inheritance. This permits the “trusted” zone to forward output connections to the Internet. (We'll talk about masquerade in the next section)

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 services that are being used on the local network to talk to Internet devices are enabled inbound . All other inbound services are disabled, and all outbound services are disabled. In this example, ingress and egress filtering are enabled, to ensure that this system doesn't accept or send malformed packets to the Internet at large.

In the “trusted” zone, only those services that are being used on the local network to talk to Internet devices are enabled inbound . All other inbound services are disabled, and all outbound services are disabled. We don't do ingress or egress filtering on this interface, as we trust all the computers on this network.

In this zone, we have enabled masquerade. This creates a bridge between the “inside” connection and a corresponding “outside” connection, with the outside connection traversing the FORWARD chains. (This 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 with the IP address of the Internet interface. This is why another zone can inherit the inbound-service connections, so that there is no mismatch between the egress filtering on the external interface and the service filtering on the trusted interface. Moreover, the sysadmin won't have to maintain any kind of forwarding list to ensure that the inside can talk to the outside.