Dual-stack Firewalling with husk

Phil Smith

linux.conf.au – Perth 2014
Phil Smith

- SysAdmin from Melbourne
- Personal Care Manufacturer
  - Implemented complete Dual-stack
- Previous role in managed security
- 4WD'ing & Fire-fighting
Google IPv6 Statistics

Per-Country IPv6 adoption

Google IPv6 Statistics

Per-Country IPv6 adoption

Are you running IPv6?

Am I running IPv6 in my network?

Have you configured IPv6 on any of your layer 3 aware network components?

- Yes
  - Have you disabled the IPv6 stack on all these computers?
    - Yes
      - You are running IPv6
    - No
      - You may not be running IPv6
  - No
    - Are you running a modern OS on any server or user device? (Windows Vista, MAC OS 10.3, Linux kernel 2.6.x and up)
      - Yes
        - Have you disabled the IPv6 stack on all these computers?
          - Yes
            - You are running IPv6
          - No
            - You may not be running IPv6
      - No
        - You may not be running IPv6
IPv6 Challenge for Firewalls

Who has the attention span for this everytime?

```bash
# vi /etc/sysconfig/iptables
  *tap tap tap tap*
# vi /etc/sysconfig/ip6tables
  *tap tap tap tap*
# iptables-restore < /etc/sysconfig/iptables
# ip6tables-restore < /etc/sysconfig/ip6tables
# kill $(pidof me)
```
IPv6 Challenge for Firewalls

- That's hard.
- If it's hard, it won't get done.

😃

- That's repetitive.
- If it's repetitive, script it!!
Husk

Sweet shell around the juicy core; like a coconut!

- Wrapper around netfilter (iptables / ip6tables)
- IPv6 Support
- Perl (mostly)
- Only 2 Dependencies
- Custom DSL
- Fails Safe (LOG and DROP by default)
- Hooks (eg, fail2ban)
- Helpers

Sweet shell around the juicy core; like a coconut!
Husk is not...

- Complete abstraction
- Automatic
- Incremental
Fun Fact

Production Dual-stack Firewall

- 2,540 rules managed in 796 lines of configuration 😊

```
fw1 ~ # (iptables-save && ip6tables-save) | grep -c -- -A
2540
fw1 ~ # egrep -cv "^#|^$" /etc/husk/rules.conf
796
```
Custom DSL

• Flexible, human-readable and case-insensitive

  \(<\text{action}>\) <\text{match criteria}>\)

• Examples:

  \text{accept in NET protocol tcp port http}
  \text{vs}
  \text{-A INPUT -i eth0 -p tcp -dport 80 -j ACCEPT}

  \text{drop in NET source address microsoft.com}
  \text{vs}
  \text{-A INPUT -i eth0 -s microsoft.com -j DROP}
Custom DSL

• Multiport Example:

\[
\text{accept in NET protocol tcp ports http,https}
\]

vs

\[
-A \text{ INPUT} -i \text{ eth0} -p \text{ tcp} -m \text{ multiport --dports 80,443 -j ACCEPT}
\]

• NAT* Example:

\[
\text{map in NET protocol tcp port http to 192.0.2.100}
\]

vs

\[
-t \text{ nat} -A \text{ PREROUTING} -i \text{ eth0} -p \text{ tcp -dport 80 -j DNAT -to 192.0.2.100}
\]

* Please don't do NAT. It kills unicorns.
Custom DSL

- Also raw iptables rules

```
iptables -t nat -A POSTROUTING -s 150.101.140.197 -j SNAT -to 1.2.3.4

ip6tables -A INPUT -m physdev --physdev-in eth0 -j ACCEPT
```

* Please don't do NAT. It kills unicorns.
Zones

• Give interfaces nice names

• Example:

  ppp0 → NET
  eth1 → LAN
  eth2 → DMZ
  lo → ME
Helpers - Built-in

- **NAT**
  - Apply NAT to outbound traffic in zone.
  - Only applied to RFC1918 source addresses

- **BOGON**
  - Drop common IPv4 + IPv6 Bogon Traffic (RFC1918, CGN, LL etc)

- **PORTSCAN**
  - Common port scanning patterns

- **XMAS**
  - Christmas Tree Packets
Helpers - Custom

• Various helpers distributed with Husk
  - Active Directory
  - GoToMeeting
  - DNS
  - Email
  - ICMP rate-limiting
  - More...
Simple Example – Standalone Host

```plaintext
define rules SSH_OK
    • accept source address example.com
• end define

•

define rules INPUT
    SSH_OK protocol tcp port ssh
    • accept protocol tcp ports http,https
• end define

•

define rules OUTPUT
    • accept all
• end define
```
Simple Example – Router

define rules LAN to NET
reject protocol tcp port smtp
accept protocol tcp ports http,https
end define

define rules NET to DMZ
accept protocol tcp ports smtp, pop3 destination address mail.example.com
DNS destination address ns1.example.com
end define

define rules LAN to DMZ
accept all
end define
Simple Example – Adding IPv6

```plaintext
define rules SSH_OK
  accept ip both source address example.com
  accept ip 4 source address 192.0.2.123
  accept ip 6 source address 2001:db8::beef
end define

define rules INPUT
  SSH_OK ip both protocol tcp port ssh
  accept ip both protocol tcp ports http,https
end define

define rules OUTPUT
  accept ip both all
end define
```
Simple Example – Adding IPv6

define rules LAN to NET
reject ip both protocol tcp port smtp
accept ip both protocol tcp ports http,https
end define

define rules NET to DMZ
accept ip 4 protocol tcp ports smtp, pop3 dest address mail.example.com
DNS ip both destination address ns1.example.com
end define

define rules LAN to DMZ
accept ip both all
end define
Applying Changes

- Atomic Loads using `iptables-restore` and `ip6tables-restore`
- Logged to syslog

```
~ # fire
Compiling rulesets...
  => IPv4
  => IPv6
Saving current rulesets...
  => IPv4
  => IPv6
Running pre-hooks...
Applying new rulesets...
  => IPv4
  => IPv6
Running post-hooks...
Can you establish NEW connections to the machine? (y/N) y
Thank-you, come again!
IPv4: Loaded 470 rules in 47 chains.
IPv6: Loaded 419 rules in 46 chains.
~ #
```
Husk Firewall

Questions?

http://huskfw.info

github.com/fukawi2/husk

@fukawi2