Boring Linux IP Routers

A recipe for boring IP routers that can be left in the closet without the Internet going down

znerol

Cosin 2024

Sysadmin Dilemma

Never touch a running system

- Security (Patch now!)
- Availability (Patch never!)

Sysadmin Dilemma

Trustworthy Supply Chain

Linux Distributions: Crowdsourced Maintenance of Software Packages

Sysadmin Dilemma

Trustworthy Supply Chain

Linux Distributions: Trustworthiness through Technology and Procedures

Sysadmin Dilemma

Trustworthy Supply Chain

Linux Distributions:

- Source Code Management
- Reproducible Builds
- Cryptography (signed artifacts)
- Web of Trust
- Project Governance
- ...

Sysadmin Dilemma

Trustworthy Supply Chain

Linux distros help with the sysadmin dilemma:

Running systems with a reasonable level of security and availability is much less effort for admins relying on a distro.

Overview

Software Components (boring)

- Debian Linux 12 (Bookworm)
- systemd-networkd
- nftables
- unattended-upgrades

Overview

Software Components (exciting)

Systemd is too boring for you? Try `https://router7.org/` instead.

Initial connectivity

The management interface

Which interface should we use for the management ip?

Initial connectivity

The management interface

A well-kept secret among network professionals: Use the local loopback, it is always up!

- https://www.oreilly.com/library/view/cisco-ios-in/156592942X/ch05s03.html

Initial connectivity

The ip prefix

2001:db8:1020:ff00::/56

specified in RFC 3849

Initial connectivity

The router ip

2001:db8:1020:ff1d:da69:dfec:a496:363e/128

Derived from `sha256("gw.example.net")`

Initial connectivity

The router ip

`https://znerol.github.io/ipaddrhash-js/`

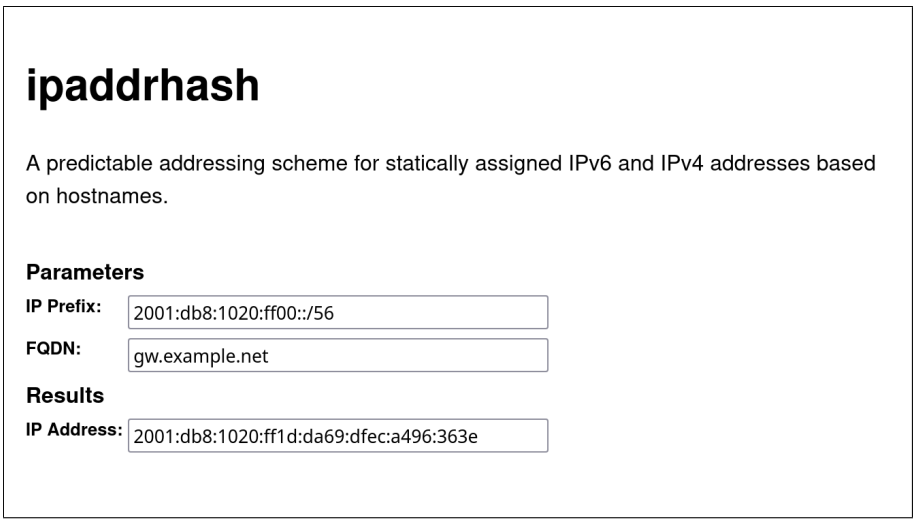

Figure: Screenshot of ipaddrash.js

Initial connectivity

The router ip

/etc/systemd/network/lo.network

```
[Match]
Name=lo

[Network]
KeepConfiguration = static
Address = 2001:db8:1020:ff1d:da69:dfec:a496:363e/128
```

Figure: lo.network

Initial connectivity

Default route for clients

Another well-kept secret among network professionals: Use fe80::1 on every downstream interface!

- `https://blogs.infoblox.com/ipv6-coe/fe80-1-is-a-perfectly-valid-ipv6-default-gatewa`

Initial connectivity

The network interface

/etc/systemd/network/lan.network

```
[Match]
Name=enp1s0

[Network]
Address = fe80::1/64
IPForward = yes
IPv6SendRA = yes

[Route]
Destination = 2001:db8:1020:ff02::/64
```

Initial connectivity

The network interface

/etc/systemd/network/lan.network

```
[Match]
Name=enp1s0

[Network]
Address = fe80::1/64
IPForward = yes
IPv6SendRA = yes

[Route]
Destination = 2001:db8:1020:ff02::/64

[IPv6Prefix]
Prefix = 2001:db8:1020:ff02::/64

[IPv6RoutePrefix]
Route = 2001:db8:1020:ff1d:da69:dfec:a496:363e/128
```

Figure: lan.network

Initial connectivity

Apply the changes

```
sudo networkctl reload
```

Initial connectivity

List interfaces

```
sudo networkctl list
```

Initial connectivity

Interface status

```
sudo networkctl status enp1s0
```

Initial connectivity

DNS Check

```
getent hosts gw.example.com
2001:db8:1020:ff1d:da69:dfec:a496:363e \
  gw.example.com
```

Initial connectivity

SSH

```
ssh gw.example.com
```

Fallback uplink

The network interface

/etc/systemd/network/wwan.network

```
[Match]
Name=enx*

[Network]
DHCP=ipv4
IPMasquerade=ipv4
IPForward=yes

[DHCPv4]
RouteMetric=65536
```

Fallback uplink

Apply the changes

```
sudo networkctl reload
```

Boring Setup

Install additional packages

```
sudo apt update
sudo apt install --yes \
  nftables \
  tcpdump \
  unattended-upgrades \
  zram-tools
```

Boring Setup

Configure zram swap

/etc/default/zramswap

```
ALGO=zstd
PERCENT=50
```

Boring Setup

Activate zram swap

```
sudo systemctl restart \
  zramswap.service
```

Boring Setup

Remove traditional swap

```
sudo swapoff /dev/mmcblk0p5
```

After that, also remove entry from `/etc/fstab`

Boring Setup

Disable journal persistence

```
sudo rm -rf /var/log/journal
sudo systemctl restart systemd-journald
```

(sufficient on Debian systems)

Boring Setup

Enable unattended upgrades

```
sudo dpkg-reconfigure -plow unattended-upgrades
```

Boring Setup

Enable unattended autoreboot

/etc/apt/apt.conf.d/99autoreboot.conf

```
Unattended-Upgrade::Automatic-Reboot "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";
```

Boring Setup

Default firewall config

/etc/nftables.conf

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  chain input {
    type filter hook input priority filter;
  }
  chain forward {
    type filter hook forward priority filter;
  }
  chain output {
    type filter hook output priority filter;
  }
}
```

Boring Setup

Firewall log instructions

Add `log` instructions to `/etc/nftables.conf`

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  chain input {
    type filter hook input priority filter;

    # Log unmatched to NFLOG group 0
    log group 0 prefix "input-filter:drop-default"
  }
  [...]
}
```

Boring Setup

Check and apply

```
sudo nft --check --file \
  /etc/nftables.conf && \
sudo systemctl reload \
  nftables.service
```

Boring Setup

Tail firewall logs

```
sudo tcpdump -qni nflog:0
```

Boring Setup

Firewall connection tracking

Add connection tracking to `/etc/nftables.conf`

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  chain input {
    type filter hook input priority filter;

    # Allow traffic from established and related
    # packets, drop invalid
    ct state established,related accept
    ct state invalid \
      log group 1 prefix "input-ct:drop-invalid" \
      drop

    # Log unmatched to NFLOG group 0
    log group 0 prefix "input-filter:drop-default"
  }
  [...]
}
```

Boring Setup

Firewall NDP support

Add IPv6 NDP support to `/etc/nftables.conf`

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  chain input {
    [...]

    icmpv6 type { \
        nd-neighbor-solicit, \
        nd-neighbor-advert, \
        ind-neighbor-solicit, \
        ind-neighbor-advert \
    } accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "input-filter:drop-default"
  }
  [...]
}
```

Boring Setup

Firewall interface groups

Add interface groups to `/etc/nftables.conf`

```
#!/usr/sbin/nft -f

flush ruleset

define IFACE_UPSTREAM = {
  enx*
}

define IFACE_DOWNSTREAM = {
  enp1s0
}
[...]
```

Boring Setup

Firewall router advertisements

Add RA support to `/etc/nftables.conf`

```
#!/usr/sbin/nft -f

flush ruleset
[...]

table inet filter {
  chain input {
    [...]

    iifname $IFACE_UPSTREAM icmpv6 type \
      nd-router-advert accept
    iifname $IFACE_DOWNSTREAM icmpv6 type \
      nd-router-solicit accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "input-filter:drop-default"
  }
  [...]
}
```

Boring Setup

Firewall accept SSH

```
#!/usr/sbin/nft -f

flush ruleset
[...]

define INET6_INTERNAL = {
  2001:db8:1020:ff00::/56
}

table inet filter {
  chain input {
    [...]

    ip6 saddr $INET6_INTERNAL tcp dport \
      ssh accept
    ip6 saddr $INET6_INTERNAL icmpv6 type \
      echo-request accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "input-filter:drop-default"
  }
  [...]
}
```

Boring Setup

Minimal firewall config with policy drop

/etc/nftables.conf

```
#!/usr/sbin/nft -f

flush ruleset

[...]

table inet filter {
  chain input {
    type filter [...]; policy drop;
    [...]
  }
  chain forward {
    type filter [...]; policy drop;
    [...]
  }
  chain output {
    type filter [...]; policy drop;
    [...]
  }
}
```

DMZ Setup

The network interface

/etc/systemd/network/dmz.network

```
[Match]
Name=enp3s0

[Network]
Address = fe80::1/64
IPForward = yes

[Route]
Destination = 2001:db8:1020:ff03::/64
```

Figure: dmz.network

DMZ Setup

Apply the changes

```
sudo networkctl reload
```

DMZ Setup

Firewall accept ping (out)

```
#!/usr/sbin/nft -f

flush ruleset
[...]

table inet filter {
  [...]
  chain output {
    [...]

    icmpv6 type echo-request accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "output-filter:drop-default"
  }
}
```

DMZ Setup

Firewall accept SSH (forward)

```
#!/usr/sbin/nft -f

flush ruleset

[...]
define IFACE_DMZ = {
  enp3s0
}

table inet filter {
  [...]
  chain forward {
    [...]

    oifname $IFACE_DMZ \
      ip6 saddr $INET6_INTERNAL tcp dport \
      ssh accept
    oifname $IFACE_DMZ \
      ip6 saddr $INET6_INTERNAL icmpv6 type \
      echo-request accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "forward-filter:drop-default"
  }
  [...]
```

DMZ Setup

Firewall accept HTTP (forward)

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  [...]
  chain forward {
    [...]

    oifname $IFACE_DMZ tcp dport http accept

    # Log unmatched to NFLOG group 0
    log group 0 prefix "forward-filter:drop-default"
  }
  [...]
}
```

Documentation

nftables wiki

https://wiki.nftables.org/wiki-nftables/index.php/Netfilter_hooks

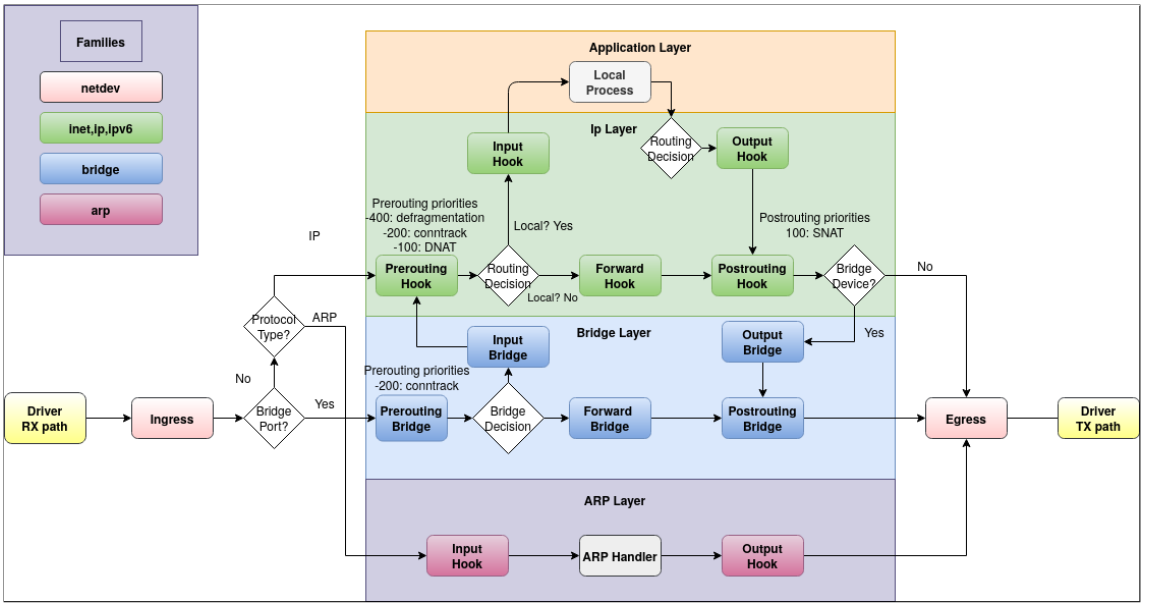

Figure: Packet flow

Documentation

man pages

- `man nft`
- `man systemd.network`
- `man systemd.netdev`

Documentation

Pragmatic IP Networking Guide

- `https://pragmatic-ip-networking-guide.readthedocs.io`
- `https://github.com/znerol/pragmatic-ip-networking-guide` (source)