IPFire 3: Networking

Basics of the next generation of networking in IPFire

Michael Tremer

IPFire Project

October 10th, 2015

What is this all about?

The new `network` command

What is this all about?

The new `network` command

One to rule them all

Introduction

Rewritten from scratch

Introduction

Rewritten from scratch

Shell code (because it is fast and easily extensible)

Introduction

Rewritten from scratch

Shell code (because it is fast and easily extensible)

Built around new concepts without any patchwork

We can do fancy stuff...

Initialises everything in parallel for fast bootup

Everything is hotpluggable

Auto-completion on command line

Easily extensible =
common language + modular design + huge library of functions for everything

Good Alpha state, testers welcome

Zones

Unlimited Zones

Zones

Zones represent a *logical* segment of the network

There are two types...

Uplink Zones (`upl0`, `upl1`, `...`)

There are two types...

Uplink Zones (`upl0`, `upl1`, `...`)

Network Zones (`net0`, `net1`, `...`)

Zone Hooks

Every supported protocol is realised as a 'hook'

Zone Hooks

Every supported protocol is realised as a 'hook'

Hooks are designed as a modular thing

Zone Hooks

Every supported protocol is realised as a 'hook'

Hooks are designed as a modular thing

Their only requirement is to create an Ethernet-like device that can have an IP address assigned

The 'bridge' hook

- The most important hook (i.e. everything local is a bridge)

The 'bridge' hook

- The most important hook (i.e. everything local is a bridge)
- Can have multiple ports
  - Packets are forwarded between ports
  - Redundancy!
  - Performance!

The 'bridge' hook

- The most important hook (i.e. everything local is a bridge)
- Can have multiple ports
  - Packets are forwarded between ports
  - Redundancy!
  - Performance!
- Entire network segments can be added at runtime

The 'bridge' hook

- The most important hook (i.e. everything local is a bridge)
- Can have multiple ports
  - Packets are forwarded between ports
  - Redundancy!
  - Performance!
- Entire network segments can be added at runtime
- (True) Layer 2 VPNs

The 'bridge' hook

- The most important hook (i.e. everything local is a bridge)
- Can have multiple ports
  - Packets are forwarded between ports
  - Redundancy!
  - Performance!
- Entire network segments can be added at runtime
- (True) Layer 2 VPNs

```
Zone up10 (enabled, bridge)
  Status                            UP
  MTU                               1500

  Statistics
    Received      1946547 packets   170M (0 errors)
    Sent           891860 packets   354M (0 errors)

  Spanning Tree Protocol information
    Version                         Spanning Tree Protocol
    ID                              0200.8e0aa9032f5f
    Priority                        512
    This bridge is root.

    Topology changing               no
    Topology change time            0s
    Topology change count           0

  Ports
    p1            FORWARDING  - DSR: 8e:0a:a9:03:2f:5f - Cost: 4
    p2            FORWARDING  - DSR: 8e:0a:a9:03:2f:5e - Cost: 4

  Configurations
    ipv4-dhcp          UP
      IPv4 address                  192.168.160.126/24
      Gateway                       192.168.160.253

      DNS-Servers                   192.168.160.253
```

The 'pppoe' hook

The PPPoE protocol is used for many dial-up connections like DSL, VDSL, some aerial connections or satellite links

Uses a port to connect to a physical network

Modular design - Takes a single (physical) port

The ’modem’ hook

Used from 56k modems over UMTS/3G to LTE/4G

Simply creates a PPP session over a (serial) link

The ‘wireless’ hook

Connects to an (encrypted) wireless network

Other hooks (only IPv6 transition protocols)

6to4-tunnel	Hurricane Electric tunnels
6rd	The 6rd transitioning protocol used by some ISPs
aiccu	AYIYA (anything-in-anything) protocol used by SIXXS.net

What have we seen?

We support everything that is supported in IPFire 2:
Ethernet (Cable + Fibre), WiFi & PPP (PPPoE, mobile connections)

Except: PPTP

Ports

Ports most often represent a *physical* segment of the network

Ports

Ports most often represent a *physical* segment of the network

They can also be virtual

The 'ethernet' hook

For Ethernet adapters in the system (physical and virtual)

- Get automatically created when a new network device is plugged in

```
[root@prime ~]# network port p1 status
Port p1 (ethernet)
  Status                              UP
  Address                             00:c0:08:87:ef:4a
  MTU                                 1500
  Promisc                             true

  Statistics
    Received      1952204 packets     197M (0 errors)
    Sent          2461555 packets     432M (0 errors)

[root@prime ~]# network device p1 status
Device status: p1
  Name                                p1
  Status                              UP
  Type                                ethernet
  Ethernet-compatible                 true
  Address                             00:c0:08:87:ef:4a

  Link                                1000 MBit/s full duplex
  MTU                                 1500

  Statistics
    Received      1952262 packets     197M (0 errors)
    Sent          2461604 packets     432M (0 errors)

  Has carrier?                        true
  Promisc                             true
```

The ‘vlan’ hook

Plain 802.1q

```
[root@prime ~]# network port new vlan --parent-device=d0 --tag=10
Configuration check succeeded.
Writing settings file /etc/network/ports/d0v10.
[root@prime ~]# network port d0v10 create
vlan device 'd0v10' has been created
[root@prime ~]# network port d0v10 up
Setting up device 'd0v10'
[root@prime ~]# network port d0v10 status
Port d0v10 (vlan)
  Status                          UP
  Address                         b6:2a:2d:f2:ce:13
  MTU                             1500
  Promisc                         false

  Statistics
    Received           0 packets     0B (0 errors)
    Sent               0 packets     0B (0 errors)

  VLAN
    Parent                        d0
    VID                           10
```

The 'bonding' hook

Combines several ports to one

- Also called "Trunk", "LAG" or "Bond"
- Adds layer 2 failover & redundancy
- Can add the throughput of multiple links (20G, 40G, ...)

```
[root@prime ~]# network port new bonding b0 --slave=d0 --slave=d1
Configuration check succeeded.
Writing settings file /etc/network/ports/b0.
[root@prime ~]# network port b0 create
Loading module 'bonding'.
Successfully created bonding device 'b0'
Set mode of bond 'b0' to '802.3ad'
Setting address of 'b0' from 'ee:2f:65:02:2e:40' to '9e:db:22:40:
[root@prime ~]# network port b0 up
Setting up device 'b0'
[root@prime ~]# network port b0 status
Port b0 (bonding)
  Status                          NO-CARRIER
  Address                         9e:db:22:40:46:b0
  MTU                             1500
  Promisc                         false

  Statistics
    Received           0 packets     0B (0 errors)
    Sent               0 packets     0B (0 errors)

  Bonding information
    Mode                          802.3ad
    LACP rate                     slow
```

The ’wireless-ap’ hook

- For Ethernet adapters in the system (physical and virtual)

```
[root@prime ~]# network port new bonding b0 --slave=d0 --slave=d1
Configuration check succeeded.
Writing settings file /etc/network/ports/b0.
[root@prime ~]# network port b0 status
Port b0 (unknown)
  Status                                DOWN
  Address
[root@prime ~]# network port b0 create
Loading module 'bonding'.
Successfully created bonding device 'b0'
Set mode of bond 'b0' to '802.3ad'
Setting address of 'b0' from 'ee:2f:65:02:2e:40' to '9e:db:22:40:
[root@prime ~]# network port b0 up
Setting up device 'b0'
[root@prime ~]# network port b0 status
Port b0 (bonding)
  Status                            NO-CARRIER
  Address                           9e:db:22:40:46:b0
  MTU                               1500
  Promisc                           false

  Statistics
    Received           0 packets      0B (0 errors)
    Sent               0 packets      0B (0 errors)

  Bonding information
    Mode                            802.3ad
    LACP rate                       slow
```

Other hooks

wireless-adhoc	Wireless Ad-hoc networks
dummy	Dummy devices
batman-adv	For wireless mesh networks with B.A.T.M.A.N

Putting it all together

Ports get attached to
and detached from zones

Putting it all together

Ports get attached to
and detached from zones

```
[root@prime ~]# network zone new net9 bridge
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/settings.
Auto-start enabled for zone net9
Started service 'network@net9.service', code=0.
[root@prime ~]# network zone net9 port attach b0
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/ports/b0.
b0 has been attached to net9
bridge: device 'b0' has been attached to bridge 'net9'
[root@prime ~]# network zone net9 port attach d0v10
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/ports/d0v10.
d0v10 has been attached to net9
bridge: device 'd0v10' has been attached to bridge 'net9'
```

Putting it all together

Ports get attached to
and detached from zones

```
[root@prime ~]# network zone new net9 bridge
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/settings.
Auto-start enabled for zone net9
Started service 'network@net9.service', code=0.
[root@prime ~]# network zone net9 port attach b0
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/ports/b0.
b0 has been attached to net9
bridge: device 'b0' has been attached to bridge 'net9'
[root@prime ~]# network zone net9 port attach d0v10
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/ports/d0v10.
d0v10 has been attached to net9
bridge: device 'd0v10' has been attached to bridge 'net9'
```

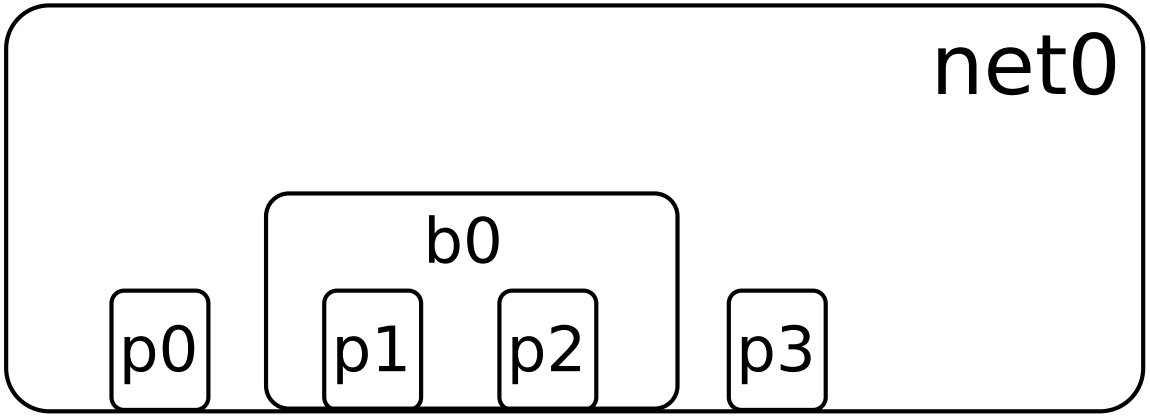
```
[root@prime ~]# network  status net9
Zone net9 (enabled, bridge)
  Status                              UP
  MTU                                 1500

  Statistics
    Received            0 packets     0B (0 errors)
    Sent                7 packets   738B (0 errors)

  Spanning Tree Protocol information
    Version                           Rapid Spanning Tree Protocol
    ID                                0200.caddf06a9f59
    Priority                          512
    This bridge is root.

    Topology changing                 no
    Topology change time              5m
    Topology change count             0

  Ports
    b0              NO-CARRIER
    d0v10           FORWARDING  - DSR: ca:dd:f0:6a:9f:59
                                - Cost: 2000000
```


net0
b0
p0
p1
p2
p3

Configs

Everything has been only layer 2 so far...
Configs add layer 3

The 'ipv{6,4}-static' hook

Static IP address assignment
for IPv6 and IPv4

```
[root@prime ~]# network zone net9 config new ipv6-static \
          --address=2001:db8:: --prefix=32
Configuration check succeeded.
Writing settings file /etc/network/zones/net9/configs/ipv6-static
IP address '2001:db8::' (ipv6) was successfully configured on dev
[root@prime ~]# ip addr show dev net9
22: net9@NONE: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500
    link/ether ca:dd:f0:6a:9f:59 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8::/32 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::c8dd:f0ff:fe6a:9f59/64 scope link
       valid_lft forever preferred_lft forever
```

The 'ipv{6,4}-dhcp' hook

Dynamic Host Configuration Protocol

```
[root@prime ~]# network zone net9 config new ipv4-dhcp
Started service 'dhclient4@net9.service', code=0.
```

Other hooks

pppoe-server	Runs a PPPoE server on a zone with IPv6 & IPv4
ipv6-auto	IPv6 auto-configuration (which is pretty much useless for us)

Static Routes

DNS

Imports DNS servers from dynamically configured connections

DHCP Server

PC Engines

Lightning Wire Labs

We bring future technology into your network

TX-TEAM

- Code: `git://git.ipfire.org/network.git`
- GitWeb: `http://git.ipfire.org/?p=network.git;a=summary`