NetBSD PPPoE Router and Firewall

Date: 2025-01-17
Updated: 2025-01-29
OS: NetBSD 10.1
Written By: Pete

I recently switched my broadband provider to **Zen Internet**, who use **City Fibre's** infrastructure. The type of connection Zen use (at least in my case) is **PPPoE** (point to point protocol over ethernet) which also requires a **VLAN**.

The whole process was completely satisfactory and I now have some serious bandwidth at home.

Although Zen provide a decent router, I wanted to use my own which is a **Draytek 2962** router/firewall. This device worked very well but after a while I wanted to see if I could get more control by utilising a Unix-like operating system on an older piece of hardware.

I have an old **Mac Mini i5 (late 2014)** which is a sturdy little beast. I equipped it with a **Thunderbolt 2 to ethernet adapter (mini display port connector)** so it then had two network interfaces. It has 4 CPU cores, 4 GB of RAM and a 512 GB non-SSD disk.

After some experimentation, I decided to give **NetBSD 10.1** a go to see if it could turn this little machine in to a respectable router/firewall. Spoiler alert - I did.

Configuring Network Interfaces

First of all I read the excellent documentation here: https://www.netbsd.org/docs/guide/en/chap-net-practice.html

I only had to make a few changes to suit my particular situation.

As the aforementioned URL does such a good job in providing the correct information, I'll just show you the relevant parts of the network configuration on the NetBSD Mac Mini.

I have two physical network interfaces, **bge0** (LAN) and **bge1** (WAN). In order to bring these interfaces up each time the machine is booted, I created the following files.

/etc/ifconfig.bge0

```
up
media autoselect
192.168.0.5 netmask 255.255.255.0
```

/etc/ifconfig.bge1

```
up
media autoselect
```

/etc/ifconfig.pppoe0

```
create
! /sbin/pppoectl -e vlan0 $int

! /sbin/pppoectl $int  myauthproto=chap myauthname=something@zen myauthsecret=myP8ssword hisauthproto=none

# Configure the PPPoE interface itself. These addresses are magic
# meaning we don't care about either address and let the remote
# ppp choose them.

inet 0.0.0.0 0.0.0.1 netmask 0xffffffff up
```

/etc/ifconfig.vlan0

```
up
create
vlan 911 vlanif bge1
```

We now have the following connection sequence (on the Mac Mini itself).

`bge0 -> pppoe0 -> vlan0 -> bge1 -> Internet`

/etc/sysctl.conf

In this file, I added the following entries to help with packet forwarding and connection stability.

```
kern.tty.qsize=32768
net.inet.tcp.mss_ifmtu=1
net.inet.ip.forwarding = 1
```

Boot time Configuration

/etc/rc.conf

In this file I added the following lines to start the interfaces in the correct order, set up a route to the Internet and activate the **NetBSD Packet Filter (NPF)** with a logging interface.

```
auto_ifconfig=NO
net_interfaces="bge1 vlan0 pppoe0 bge0"
npf=YES
npd=YES
ifwatchd=YES
ifwatchd_flags="-u /etc/ppp/ip-up -d /etc/ppp/ip-down pppoe0"
```

You will need to create the directory and files shown in the last line above. Make sure the files are executable.

/etc/ppp/ip-down

```
#!/bin/sh
/sbin/route delete default $5
```

/etc/ppp/ip-up

```
#!/bin/sh
/sbin/route add default $5
```

NPF (NetBSD Packet Filter)

NetBSD uses **npf** to control data flowing through network interfaces. Here is my configuration which (amongst other things) enables **NAT** (Network Address Translation) and **port-forwarding** for a website using the HTTPS protocol. Use **npfctl** to administer NPF.

/etc/npf.conf

```
$int_if = "bge0"
$ext_if = "pppoe0"
$localnet = { 192.168.0.0/24 }
# $wifi_if = "bge2"
# $wifinet = { 10.0.0.0/8 }


set bpf.jit off


alg "icmp"


procedure "log" {
        log: npflog0
  }


procedure "norm4" {
        normalize: "random-id", "max-mss" 1440
}


# Populate this table with single IP addresses and/or networks (use CIDR)
# table <badnets> type lpm file "/etc/bad_nets"


# NAT
map inet4($ext_if) dynamic $localnet -> inet4($ext_if)
# map inet4($ext_if) dynamic $wifinet -> inet4($ext_if)


# Port-forwarding (see as well the rule in the external group)
map $ext_if dynamic proto tcp 192.168.0.10 port 443 <- $ext_if port 443
# map $ext_if dynamic proto tcp 192.168.0.10 port 21 <- $ext_if port 21


# Port-forwarding a range of ports (see as well the rules in the external group)
# map $ext_if dynamic proto tcp 192.168.0.10 <- $ext_if port 3000-4000


group "external" on $ext_if {

        # block in final from <badnets>
        pass in final proto icmp icmp-type 11 all
        pass in final proto icmp icmp-type 3 all
        pass in final proto icmp icmp-type 0 all
        pass in final proto icmp icmp-type 4 all
        pass in final proto icmp icmp-type 12 all
        pass in final proto icmp icmp-type 8 all


        pass stateful out final family inet4 all apply "norm4"


        pass stateful in final family inet4 proto tcp to $ext_if \
                port 443 apply "log"


        # pass stateful in final family inet4 proto tcp to $ext_if port 21
        # pass stateful in final family inet4 proto tcp to $ext_if port 3000-4000

}


# group "wifi" on $wifi_if {
        # uncomment the following line to segregate the wifi and internal networks
        # block in final from any to $localnet
#       pass in all
#       pass out all
#}


group "internal" on $int_if {

        # uncomment the following line to segregate the wifi and internal networks
        # block in final from any to $wifinet
        pass in all
        pass out all

}


group default {
        pass final on lo0 all
        block all

}
```

Rebooting The Router/Firewall

Hopefully the information above will enable you to configure your own router/firewall using **NetBSD 10.1**.

Please note that you may run in to issues with the router rebooting as **NetBSD 10.1** appears to require a connected monitor to boot correctly. This is obviously a problem if you want to run the router in headless mode (no monitor).

Here's some **dmesg** output which might be relevant.

```
[     9.870300] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[     9.885832] [drm] Driver supports precise vblank timestamp query.
[     9.892915] i915drmkms0: interrupting at msi6 vec 0 (i915drmkms0)
[     9.900301] [drm] Initialized i915 1.6.0 20200114 for i915drmkms0 on minor 0
[    10.350300] intelfb0 at i915drmkms0
[    10.680300] [drm] DRM_I915_DEBUG enabled
[    11.020300] [drm] DRM_I915_DEBUG_GEM enabled
[    11.350300] intelfb0: framebuffer at 0x90009000, size 1920x1080, depth 32, stride 7680
[    11.390301] (drm:netbsd:pipe_config_infoframe_mismatch+0x40) *ERROR* mismatch in hdmi infoframe
[    11.720300] (drm:netbsd:pipe_config_infoframe_mismatch+0x4e) *ERROR* expected:
[    12.050299] i915drmkms0: 68 bytes @ 0xffffe55066778be8
[    12.390299] 81 00 00 00 01 00 00 00  03 0c 00 00 00 00 00 00 | ................
[    12.720299] ff ff ff ff 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    13.050299] 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    13.380299] 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    13.720299] 00 00 00 00                                     | ....
[    14.050298] (drm:netbsd:pipe_config_infoframe_mismatch+0x6f) *ERROR* found:
[    14.380298] i915drmkms0: 68 bytes @ 0xffffe55066778be8
[    14.720298] 01 00 00 00 01 04 00 00  00 00 00 00 00 00 00 00 | ................
[    15.050298] 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    15.380298] 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    15.720297] 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
[    16.050298] 00 00 00 00                                     | ....
[    16.380297] warning: /usr/src/sys/external/bsd/drm2/dist/drm/i915/display/intel_display.c:14031: pipe state doesn't match!
```

I've managed to overcome this issue by following some sage advice on T'Internet and have purchased an **EVanlak Dummy HDMI plug** which allows NetBSD to boot correctly without a physical display attached to my Mac Mini i5 (late 2014) computer.

If you use tables and see an error when reloading the `/etc/npf.conf` configuration file that states **File Exists**, it may be down to overlapping network ranges in those table files.

I hope you found this information useful. Thanks for visiting.

Pete.