Hoy vamos a actualizar la red de casa o pequeña oficina estrenando placa y de paso tirando algo de cable de red de por medio.

Para ello usaremos Opnsense, fork del famoso Pfsense, distro basada en FreeBSD con fines de cortafuegos y enrutador.

La placa SBC (Single Board Computer) o alternativa a Raspberry con “muchos esteroides”, es una Odroid modelo H2+, basada en la arquitectura X86 (nos viene genial para montar servicios en docker próximamente). Tiene múltiples puntos fuertes esta placa, pero en concreto lo que más nos puede resultar útil, para este proyecto en cuestión, es que cuenta con dos interfaces de red 2.5 Gigabit. (Lo que la hacen perfecta para hacer de router).

Imagen de Odroid H2+

Primero procederemos a montar la placa, es muy delicada en cuanto a hardware compatible, al no incluir ni memoria RAM ni disco integrado tendremos que ir buscando que tipos de tarjetas RAM o que discos son compatibles y cuales no.

En mi caso únicamente monto una memoria Ram de 4GB (donada por Ale, GRACIAS de

nuevo) y el sistema correrá en un pendrive USB3*. (Al no detectar los viejos discos SATA que tengo… seguiré investigando).

*Lo ideal sería correr el sistema en una memoria eMMC ya que la velocidad aumentaría considerablemente.

Procedemos a descargar la imagen de Opnsense y comprobar su checksum.

Luego montamos la imagen en el pendrive con dd

```
sudo dd if=opnsense-version.img of=/dev/rutadetupendrive
```

Luego aprovechamos y formateamos otro pendrive USB que tengamos libre en formato FAT32, para cargarle los drivers de las interfaces de red Realtek RTL8125B, ya que no vienen incluidos en la version 12 de FreeBSD.

El binario de los drivers los he encontrado en la siguiente web, de un proyecto similar y procedemos a instalarlos una vez la instalación haya concluido.

FreeNAS on Odroid H2+ – RTL8125B driver installation

Nos logueamos como root y la pass Opnsense y pulsamos 8 + Enter para acceder a la consola FreeBSD.

Montamos el segundo pendrive USB que contiene los drivers Realtek de los puertos Ethernet.

```
mkdir /mnt/usbstick
mount -t msdosfs /dev/da0s1 /mnt/usbstick
```

Copiamos el driver y cambiamos los permisos del archivo.

```
cd /boot/kernel
cp /mnt/usbstick/if_re.ko ./
chown root:wheel if_re.ko
chmod 0555 if_re.ko
```

Para cargar el módulo al arranque debemos añadir la siguiente línea al archivo */boot*/loader.conf

```
vi /boot/loader.conf
add line --> if_re_load="YES"
```

Si no estas familiarizad@ con vi:

i→ Modo de inserción de texto

Esc→ Salir del modo de inserción de texto

:wq→ Guardar fichero y salir

:q!→ Salir sin guardar.

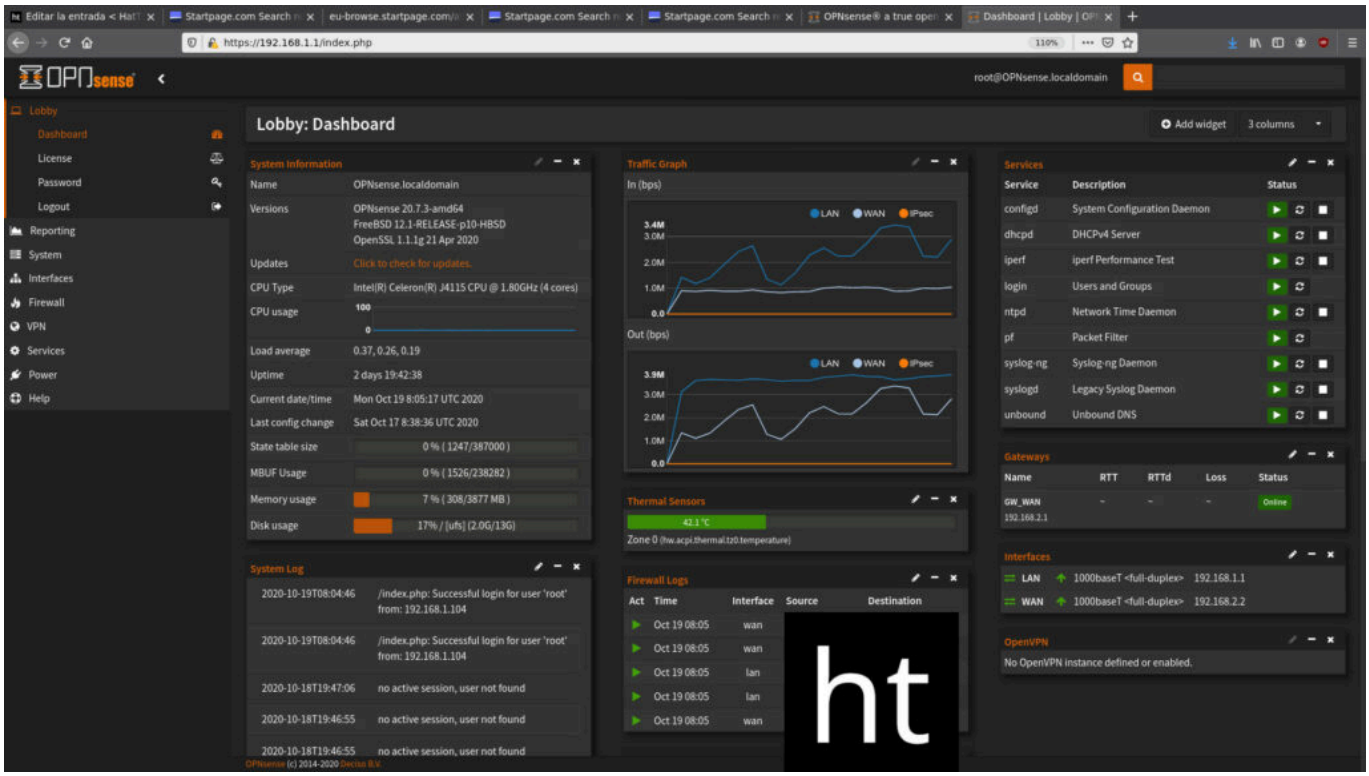
El driver debería de estar cargado al reiniciar.

Lo comprobamos con el comando `kldstat, o ifconfig,`

Deberían aparecer las dos redes re0 y re1. Opnsense asigna la interfaz re0 a la red LAN y la interfaz re1 a la red WAN.

Por defecto el puerto WAN tendra cliente dhcp y espera a ser asignada una ip.

Por defecto el puerto LAN tiene un servidor dhcp y una dirección estática de 192.168.1.1 donde podremos visualizar la interfaz de Opnsense por web https.

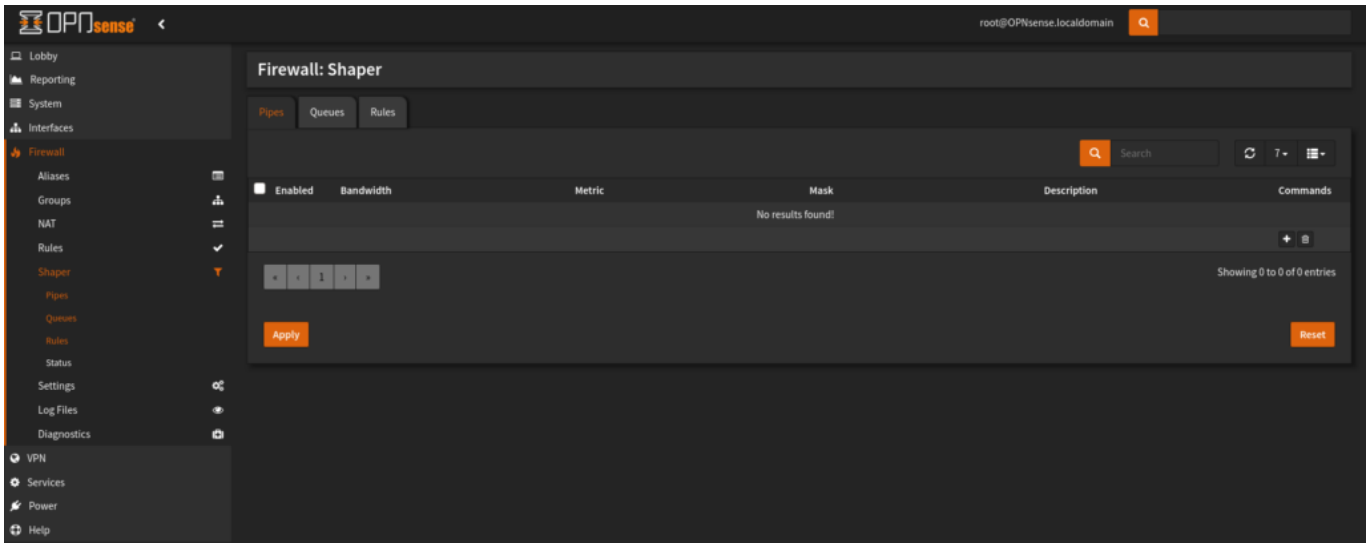


La Lan tendría la ip estática de 192.168.1.1 y gateway de 192.168.2.2. A su vez actuaría como servidor DHCP y DNS.En el modem del proveedor procedemos a cambiar la ip a 192.168.2.1 y añadir como estática la 192.168.2.2 con la MAC del puerto WAN de la placa Odroid H2+, deshabilitamos servidor DHCP al igual que las interfaces Wifi y añadimos la dirección ip 192.168.2.2 como DMZ (zona desmilitarizada).

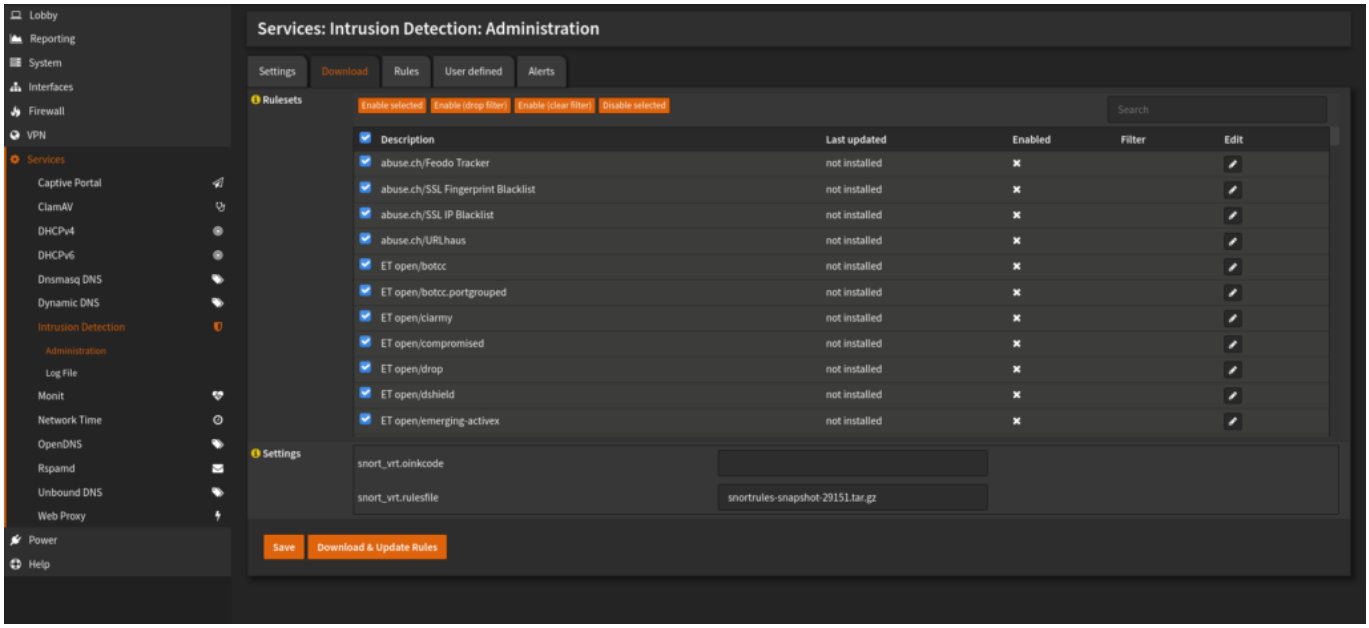
Una vez comprobado que todo funciona con normalidad en nuestro caso procedemos a conectar la interfaz LAN a un switch gigabit de 8 puertos que distribuya la red. Posteriormente procedemos a conectar los routers a este switch, asignamos ip estática del opnsense con cada router con su ip y dirección Mac, desactivando el servidor DHCP en la interfaz LAN de cada router Openwrt.

Todo muy casero .

Luego podemos cacharrear a nuestro gusto , como jugando con las prioridades del tráfico o la prevención de instrusos con la integración de snort y sus reglas.



Aquí priorizamos parte del tráfico.



IDS Reglas por defecto.

Y bueno a partir de ahi ir añadiendo routers secundarios como Puntos de Acceso Wifi, por ejemplo, con sistemas OpenWRT para tener control total sobre la red.

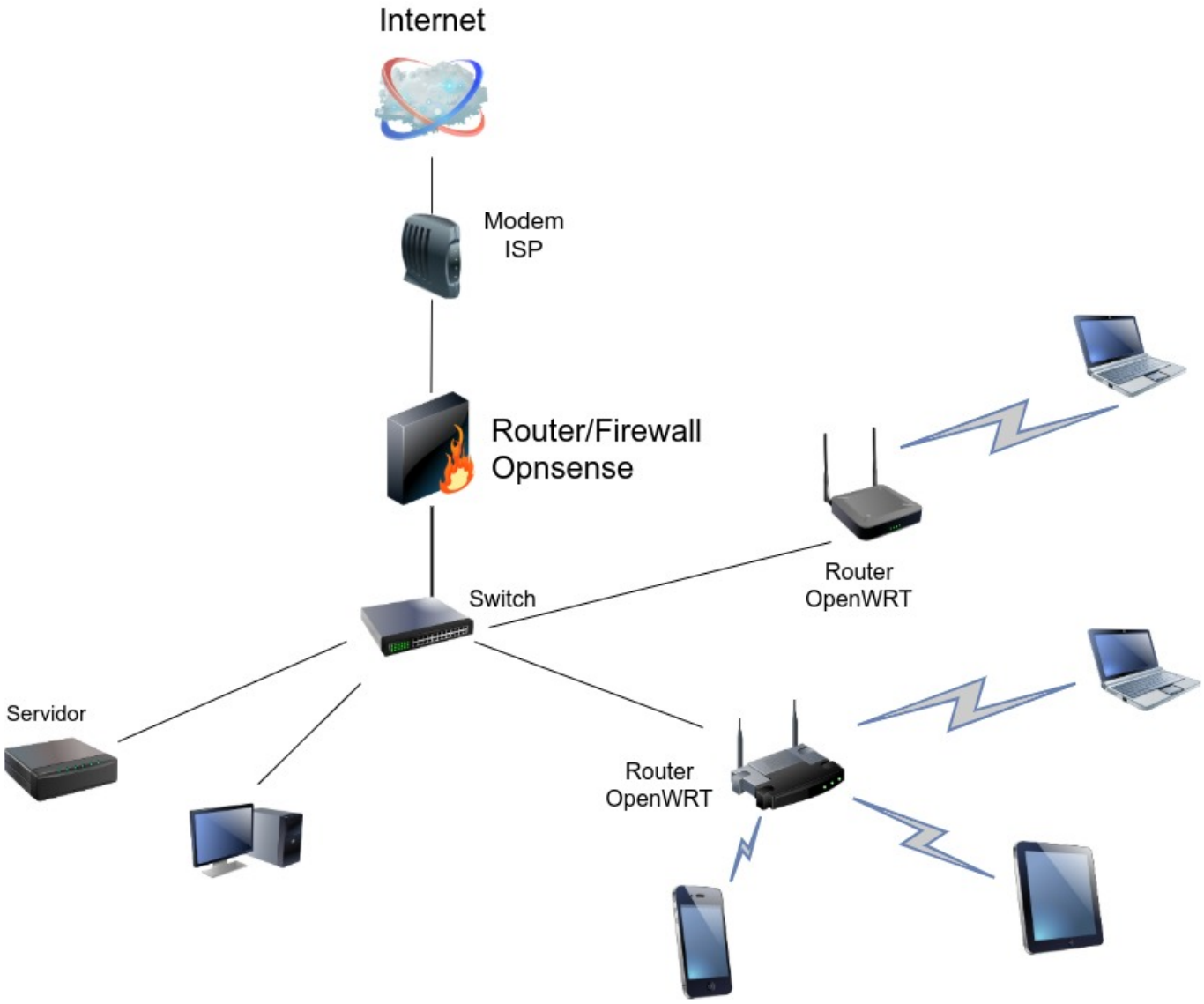


Diagrama de red hecho con draw.io