

A black and white photograph showing the back of a person wearing a dark jacket and jeans, standing against a wall made of vertical wooden planks. The wall shows significant weathering and discoloration.

irontec

Introducción a

**pfSense®**

# Características

# Firewall

- Filtrado por IP de origen y destino
- Limitar el número de conexiones basado en reglas
- pfSense usa **p0f**, sistema pasivo de detección del sistema operativo que inicializa la conexión
- Posibilidad de loguear el tráfico, por cada regla
- Posibilidad de elegir rutas específicas por regla, indicando el gateway (para balanceo de tráfico, multiples WAN...)
- Posibilidad de crear "alias" para agrupar IPs, redes y puertos
- Deshabilitar filtrado. Por si queremos convertir pfSense en router
- ...

# Tabla de estados

- Nos muestra las conexiones abiertas
  - en pfSense todas las reglas son "stateful", **sólo los paquetes que coincidan con una conexión activa conocida serán permitidos por el firewall; los otros serán rechazados.**
- Tamaño ajustable al vuelo
  - Cada estado ocupa ~1KB de RAM
    - Hay que tenerlo en cuenta a la hora de modificarla
- Posibilidad de ajustarlo por reglas
  - Limitar el número de conexiones simultaneas por cliente
  - Por cliente
  - Nuevas conexiones por segundo

## Tabla de estados (II)

- Tipos de estado
  - **keep**: Funciona con todos los protocolos. Por defecto para todas las reglas
  - **sloppy**: Funciona con todos los protocolos. Estado menos estricto de "trackeo".
  - **synproxy**: Hace de proxy en las conexiones TCP entrantes, que ayudan a proteger de ataques de tipo **TCP SYN floods**.
  - **none**: No se mantiene ningún estado para este tráfico. Es raro que se use, pero la posibilidad existe

## Tabla de estados (III)

- Optimizaciones de la tabla de estados
  - **Normal**: El algoritmo por defecto
  - **High latency**: Utilizado para enlaces de alta latencia. Expira las conexiones en estado *idle* más tarde de lo habitual
  - **Aggressive**: Expira las conexiones más rápidamente. Más eficiente con los recursos hardware, pero puede *dropear* conexiones legítimas.
  - **Conservative**: Intenta evitar *dropear* conexiones legítimas, pero incrementando el uso de memoria y CPU.

# NAT

- Port forwarding (rango de puertos)
- NAT 1:1, para IPs individuales o subredes
- NAT saliente
  - Por defecto todo el trafico saliente hace NAT por la ip WAN.
  - Posibilidades de NAT saliente avanzado
- NAT *reflection*
  - También conocido como *NAT hairping* o *NAT loopback*
  - Permite conexiones a la IP pública desde las redes internas

# Alta disponibilidad (HA)

- Combinación de CARP, pfsync y sincronización de configuración para permitir entornos HA
  - El sistema es *primary-secondary*
- Posibilidad de tener dos o más pfSense configurados como grupo failover
- La configuración se realiza en el servidor primario, y se sincroniza con los secundarios
  - Existe la posibilidad de que ciertos paquetes no tengan esta opción
- La tabla de estados es replicada a todos los servidores failover.
  - Las conexiones establecidas se mantendrán en caso de fallo

## Otras opciones

- Multi-WAN
  - Posibilidad de tener más de una conexión a internet, con balanceo de carga y/o failover
- VPN
  - IpSec
  - OpenVPN
- Monitorización
  - Posibilidad de ver el estado, paquetes, tráfico por interfaz...
  - Generación de gráficas
- DNS Dinámico

## Otras opciones (II)

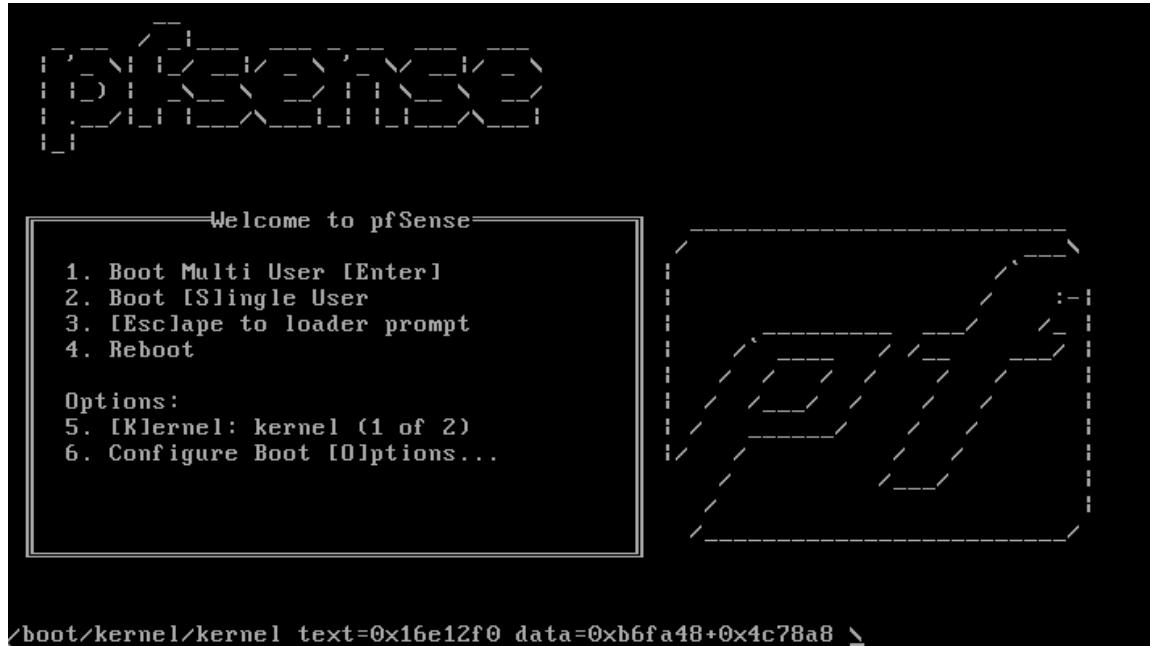
- DHCP Server y relay
- Portal cautivo
- Balanceo de carga
  - Haproxy
- Filtrado web
  - Squidguard
- ...
  - Multiples paquetes a instalar.

# Software Libre

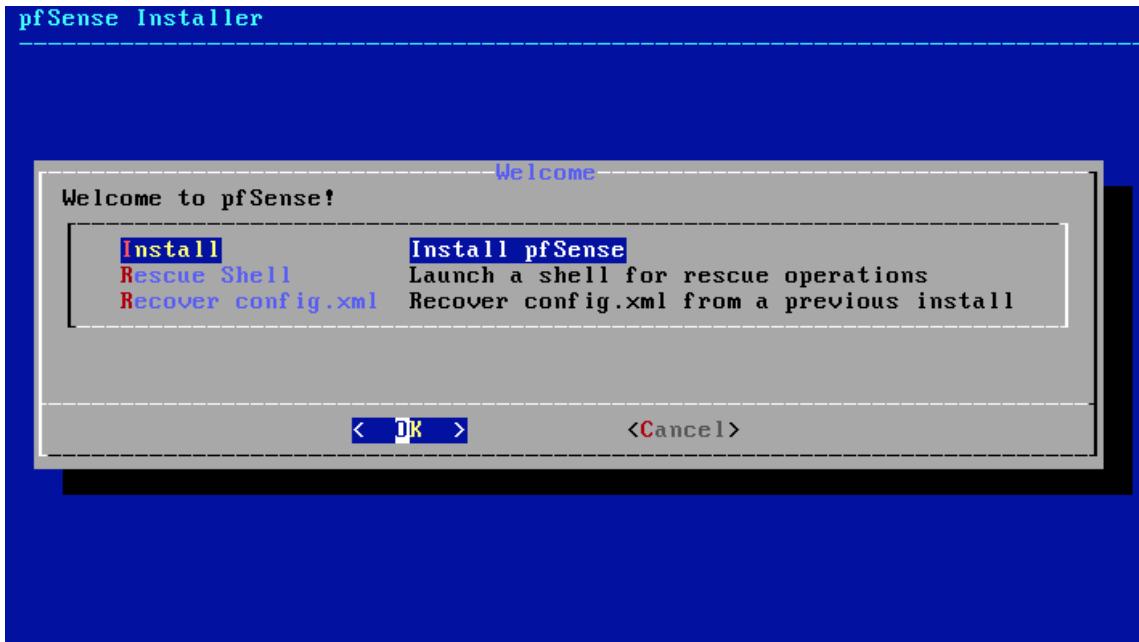
- Basado en [FreeBSD 11.1](#)
- Última versión [2.4.3](#)
- Originalmente un fork de [m0n0wall](#)
  - proyecto abandonado y "renacido" en [OPNsense](#)
    - fork a su vez de pfSense
- Hasta la versión 2.3 (2016) el interfaz era bastante feo
  - Versiones antiguas de FreeBSD
- [Appliance propio](#)
- [Soporte directo](#)

# Instalación y asistente de configuración

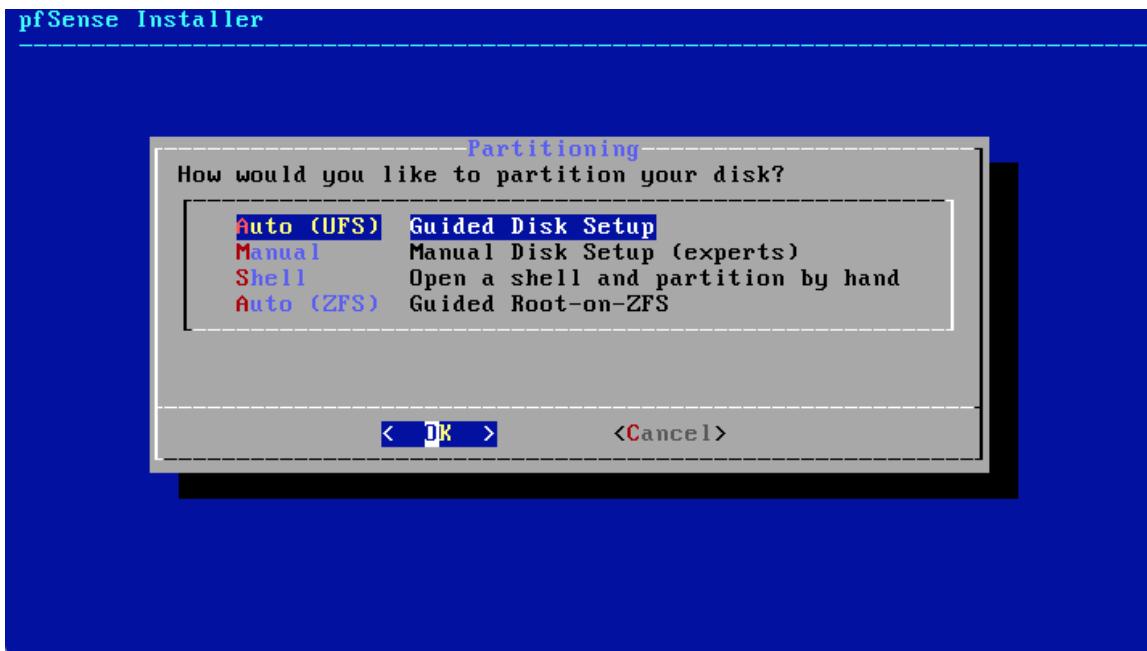
# Menú de instalación



# Menú de instalación (II)



# Menú de instalación (III)



# Configuración

- Tras un primer reinicio
  - Configurar las posibles VLAN e interfaces de red

```
Features2=0x80202001<SSE3,CX16,x2APIC,HU>
AMD Features=0x20100800<SYSCALL,NX,LM>
AMD Features2=0x1<LAHF>
Hypervisor: Origin = "KUMKUMKUM"
Done.
.... done.
Initializing..... done.
Starting device manager (devd)...done.
Loading configuration.....done.
Updating configuration...done.

Default interfaces not found -- Running interface assignment option.
vtnet0: link state changed to UP

Valid interfaces are:

vtnet0 66:03:e8:0f:9a:76 (down) VirtIO Networking Adapter
em0      a6:31:35:02:6c:f1 (down)

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y/n]? 
```

# Configuración (II)

- Por defecto
  - WAN: IP por DHCP
  - LAN: 192.168.1.1/24
  - user / pass: root / pfsense

```
Starting syslog...done.
Starting CRON... done.
pfSense 2.4.3-RELEASE amd64 Mon Mar 26 18:02:04 CDT 2018
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

pfSense - Netgate Device ID: ca41b36788616fbe06c3

*** Welcome to pfSense 2.4.3-RELEASE (amd64) on pfSense ***

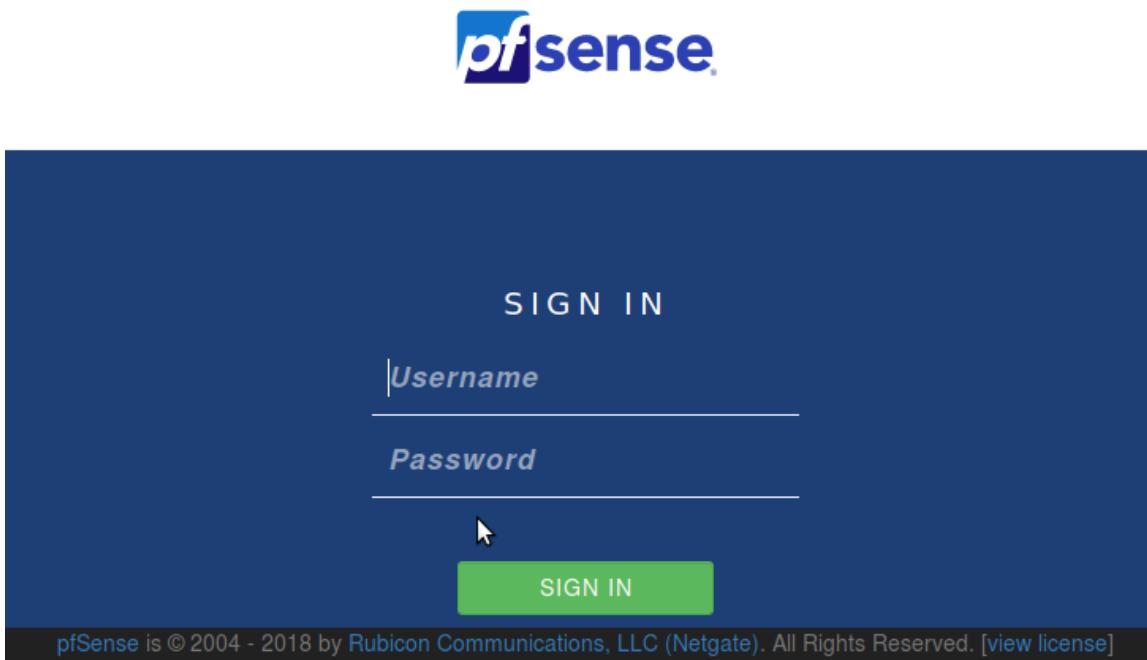
WAN (wan)      -> vtnet0      -> v4/DHCP4: 10.10.0.161/24
LAN (lan)      -> vtnet1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █
```

# Configuración (III)

- Accediendo desde la LAN



# Asistente de configuración

- Al entrar por primera vez, aparece un asistente de configuración (wizard)
  - Nos realiza una serie de preguntas para configurar por primera vez pfSense
  - Son 9 pasos para realizar configuración básica
  - Esta configuración puede ser modificada posteriormente en cualquier momento

# Asistente de configuración (II)

Wizard / pfSense Setup / General Information

Step 2 of 9

## General Information

On this screen the general pfSense parameters will be set.

**Hostname** pfSense  
EXAMPLE: myserver

**Domain** irontec.com  
EXAMPLE: mydomain.com

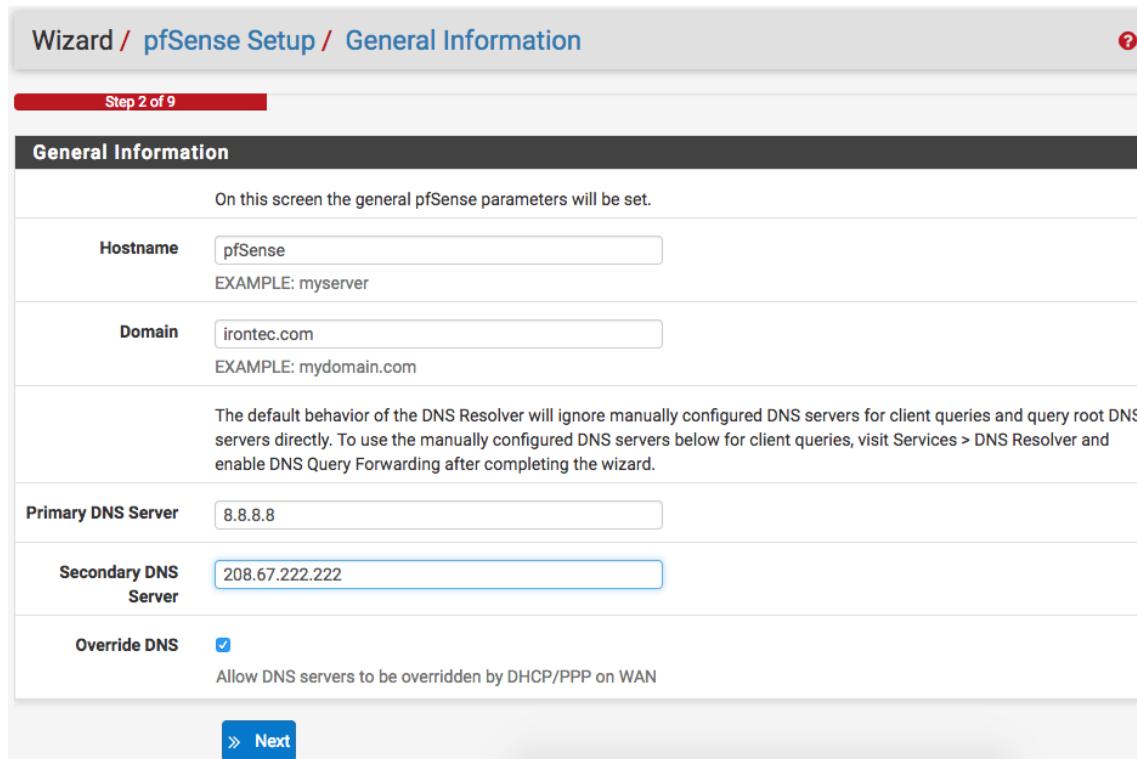
The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit Services > DNS Resolver and enable DNS Query Forwarding after completing the wizard.

**Primary DNS Server** 8.8.8.8

**Secondary DNS Server** 208.67.222.222

**Override DNS**  Allow DNS servers to be overridden by DHCP/PPP on WAN

» Next



# Asistente de configuración (III)

Wizard / pfSense Setup / Time Server Information

Step 3 of 9

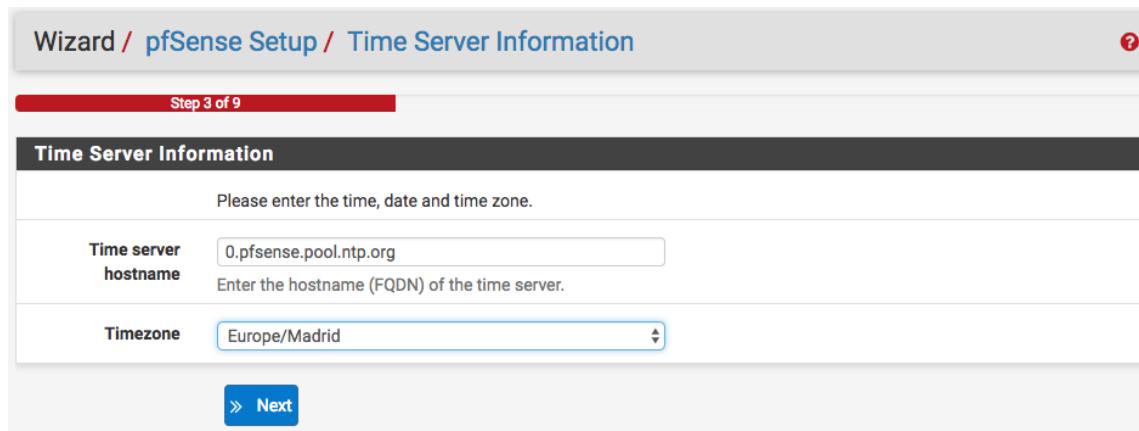
## Time Server Information

Please enter the time, date and time zone.

Time server hostname: 0.pfsense.pool.ntp.org  
Enter the hostname (FQDN) of the time server.

Timezone: Europe/Madrid

» Next



# Asistente de configuración (IV)

Wizard / pfSense Setup / Configure WAN Interface

Step 4 of 9

## Configure WAN Interface

On this screen the Wide Area Network information will be configured.

SelectedType

### General configuration

MAC Address

This field can be used to modify ("spoof") the MAC address of the WAN interface (may be required with some cable connections). Enter a MAC address in the following format: xxxx:xxxx:xxxx or leave blank.

MTU

Set the MTU of the WAN interface. If this field is left blank, an MTU of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.

MSS

If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect. If this field is left blank, an MSS of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed. This should match the above MTU value in most all cases.

# Asistente de configuración (V)

Wizard / pfSense Setup / Configure LAN Interface

Step 5 of 9

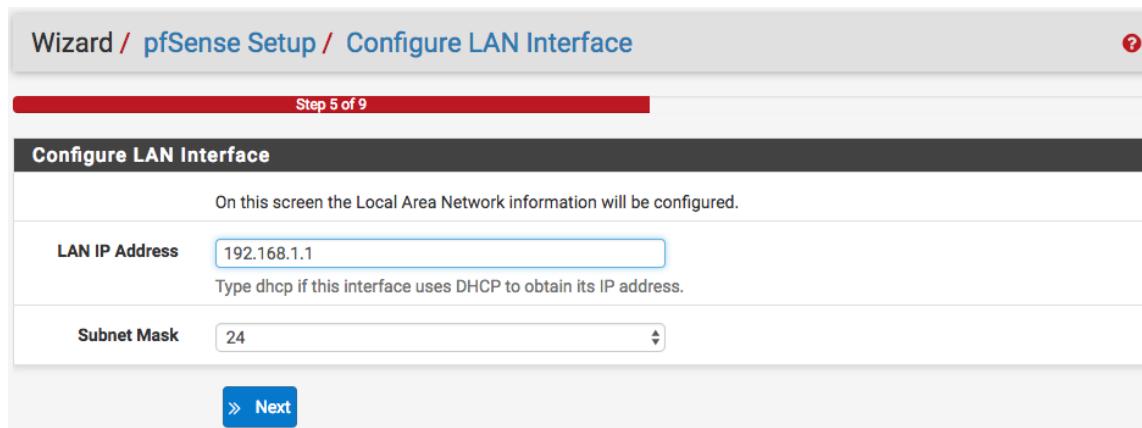
## Configure LAN Interface

On this screen the Local Area Network information will be configured.

LAN IP Address: 192.168.1.1  
Type dhcp if this interface uses DHCP to obtain its IP address.

Subnet Mask: 24

» Next



# Asistente de configuración (VI)

Wizard / pfSense Setup / Set Admin WebGUI Password

Step 6 of 9

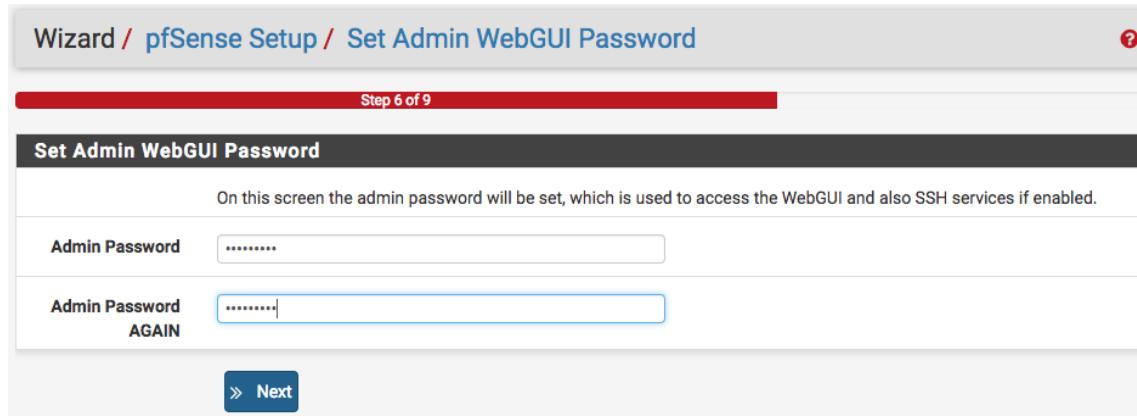
### Set Admin WebGUI Password

On this screen the admin password will be set, which is used to access the WebGUI and also SSH services if enabled.

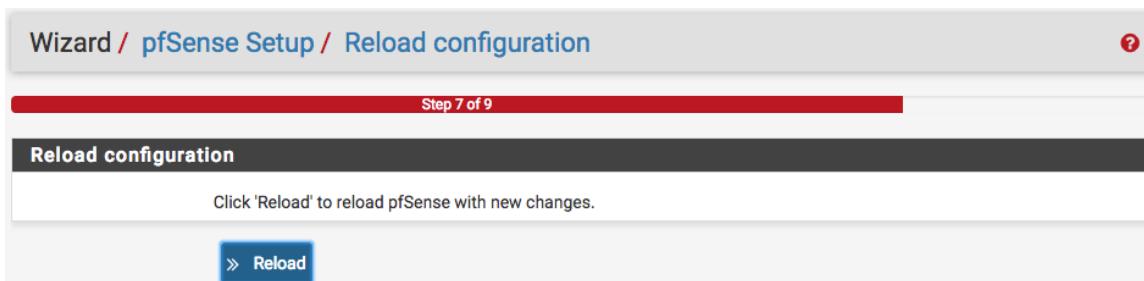
Admin Password: .....

Admin Password AGAIN: .....

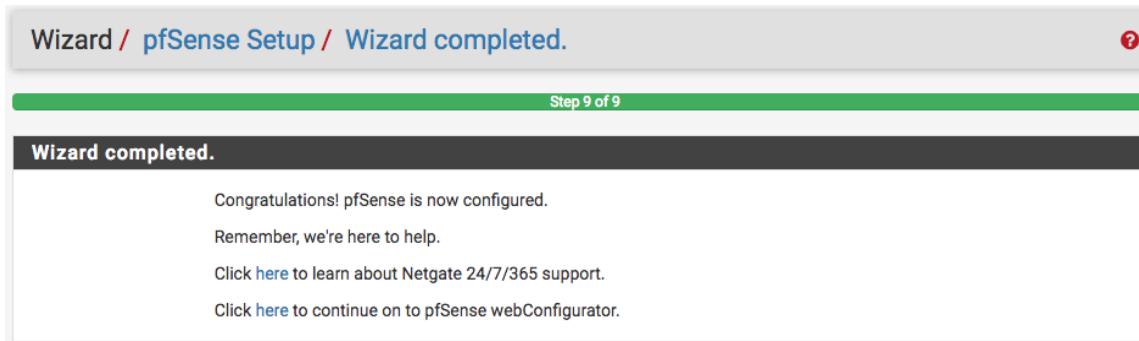
**» Next**



# Asistente de configuración (VII)



# Asistente de configuración (VIII)



# Asistente de configuración

- Configuración base terminada
  - Tenemos un firewall básico configurado
  - Todo el acceso bloqueado hacia el interior
  - DHCP en la LAN y configuración básica
  - Un buen punto de partida para cualquier oficina pequeña

# Configuración avanzada del sistema

# **Configuración avanzada**

**System → Advanced**

# Admin Access

- Preferencias para el acceso web
  - **Protocolo:** HTTP/HTTPS
  - **SSL Certificate:** Certificado del interfaz web
  - **TCP port:** puerto de acceso web
  - **WebGUI redirect:** Hacer redirect HTTP → HTTPS
  - **HSTS:** Para que el navegador pueda o no entrar al HTTP
  - **Anti-lockout:** Permite acceso desde LAN generando una regla propia. Si se desmarca, habría que crear regla para el acceso web! PELIGRO!
  - **DNS Rebind Check:** Protege de un acceso a través de un DNS no permitido  
**DNS Rebinding attacks**

## Admin Access (II)

- SSH
  - Importante para poder conectarnos desde una consola
  - **Puerto:** Por si queremos cambiar el puerto por defecto
- Console Options
  - **Console menu:** Para proteger el acceso al menú principal conectado al propio servidor

# Firewall & NAT

- **Firewall Advanced:**
  - Opciones avanzadas de Firewall
  - **Disable Firewall:** Opción para deshabilitar el filtrado de paquetes
    - Convierte pfSense en un mero router.
  - Modificaciones de la tabla de estados
    - Posibilidad de incrementarla

# Networking

- **IPv6**
  - Si no vamos a hacer uso de ello, mejor deshabilitarlo
- **Network Interfaces**
  - **Hardware Checksum Offloading:** Existe incompatibilidades con algunas tarjetas. Suele ser aconsejable marcar la casilla para deshabilitarlo.
  - **Reset All States:** No suele ser habitual cambiar la WAN, pero igual nos puede interesar.

## Otros

- Miscellaneous
  - RAM disk
  - Hardware criptográfico
- System Tunables
  - **Cuidado!!!** Opciones muy avanzadas. Tocar con cuidado
- Notifications
  - Configurar notificaciones por mail y/o growl

# Configuración avanzada (II)

# **Administración de certificados**

## System → Certificate Manager

- Lugar donde se guardan los certificados que posteriormente se pueden utilizar en pfSense
  - Interfaz web
  - Haproxy
  - OpenVPN
    - Cliente y/o Servidor

# Configuración General

# System → General Setup

- Sistema
  - **Hostname:** Nombre del firewall
  - **Domain:** Nombre del dominio. Debería ser FQDN
- DNS
  - Configuración utilizada por pfSense
- Localization
  - Fecha e idioma
- webConfigurator
  - Configuración específica para el interfaz web

# Gestor de paquetes

## System → Package Manager

- pfSense cuenta con la posibilidad de instalar paquetes externos al sistema base
- Estos paquetes amplian las funcionalidades del sistema añadiendo servicios nuevos
- Son paquetes conocidos en el mundo del software libre, con integración al interfaz web de pfSense
  - El interfaz web intenta evitar que tengamos que modificar el fichero de configuración

# Paquetes habituales

- **acme**: Sistema automático para desplegar certificados Let's Encrypt
- **cron**: Utilidad para ejecutar comandos en un horario específico
- **frr**: Demonio de enrutado dinámico para BGP, OSPF.
- **haproxy**: Balanceador de carga TCP/HTTP(S).
- **nmap**: explorador de red y auditador de red
- **nrpe**: servidor NRPE para poder monitorizar desde Nagios

## Paquetes habituales (II)

- **openvpn-client-export**: exportador para la configuración de clientes OpenVPN
- **squid**: Proxy caché para contenido web.
- **squidGuard**: filtrador de URLs

# Gestor de actualización

## System → Update

- Lugar donde poder actualizar el sistema completamente
  - **NOTA:** Deberías hacer previamente un backup de la configuración por si acaso
    - Diagnostics → Backup & Restore
- También podemos elegir si queremos la versión estable o la de desarrollo

# Gestión de Usuarios

**System → User Manager**

# Usuarios

- Entidad única de autenticación
  - Pueden ser usados para acceder al pfSense o conectarse por OpenVPN
- Pueden tener fecha de expiración
  - momento en el cual dejan de ser válidos
- Deben de tener contraseña
- Pueden tener certificado propio
- Pueden pertenecer a grupos.
- Pueden tener acceso limitado al interfaz web

# Grupos

- Sirve para agrupar usuarios
- Limitar el acceso al interfaz web
  - Permitir sólo a ciertas partes

# Servidores de autenticación

- pfSense usa su propia base de datos para los usuarios
- Podemos configurar servidor externo de autenticación
- Los usuarios serán autenticados contra los servidores externos
- Podemos hacer uso de estos servidores en configuración como OpenVPN
- Tipos
  - LDAP / Active Directory
  - Radius

# Configuración contra Active Directory

- **Type:** LDAP
- **IP address:** 192.168.x.y
- **Protocol Version:** 3
- **Search scope:** Entire Subree
- **Base DN:** DC=corp,DC=irontec,DC=com
- **Authentication containers:** OU=Usuarios,DC=corp,DC=irontec,DC=com
- **Bind credentials:** user@corp.irontec.com , \*\* (password)
- **User naming attribute:** samAccountName
- **Group namgin attribute:** cn
- **Group member attribute:** memberOf
- **Group Object Class:** posixGroup

# Ejercicio

- Crear usuario
  - Que sólo pueda entrar por SSH
  - Intentar loguearnos al interfaz web
- Crear grupo
  - Permitir acceso a toda la web de "Firewall"
    - Añadir a este grupo el usuario creado anteriormente

# **Asignación de Interfaces, VLAN, LAGG...**

# Interfaces

- WAN
  - pfSense por defecto espera al menos una interfaz WAN
    - No debería estar configurada por DHCP
  - IP pública
    - Estática
    - PPPoE
  - Si no tiene IP pública:
    - Deshabilitar "*Block private networks and loopback addresses*".
- LAN
  - Red LAN, con IP estática
  - Se podrá acceder al pfSense desde esta red normalmente

# Grupo de interfaces

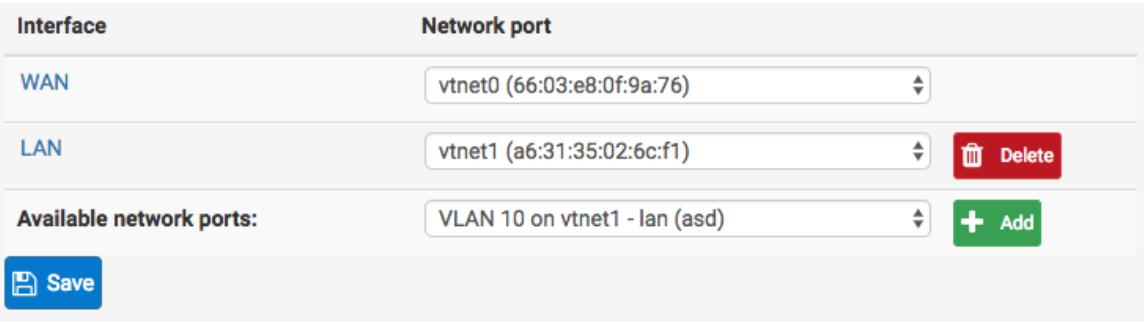
- Sirve para agrupar interfaces creadas
  - Captain obvious!
- Su función sirve para poder realizar reglas de filtrado sobre el grupo, sin tener que duplicarlas por interfaces
- Nos facilita la administración en caso de tener distintas redes LAN que puedan compartir reglas

# VLANs

- Creación de VLANs asociadas a un interfaz físico concreto
- Una vez creada la VLAN, nos crea un interfaz "virtual" donde podemos asignar IP para dicha VLAN

Interface	Network port
WAN	vtnet0 (66:03:e8:0f:9a:76)
LAN	vtnet1 (a6:31:35:02:6c:f1) <span style="color: red;">Delete</span>
Available network ports:	VLAN 10 on vtnet1 - lan (asd) <span style="color: green;">+ Add</span>

Save



- A todos los interfaces se les puede cambiar el nombre, para que sean más descriptivos

## PPPs

- Creación de interfaces de tipo PPP (Point-to-Point Protocol)
  - PPPoE
  - PPTP
  - L2TP
- Posibilidad de obtener IP pública de manera directa con algunos proveedores de internet

# LAGGs

- Creación de agregación de enlaces
- Posibilidad de tener varios interfaces físicos como si fueran uno sólo
- Tipos:
  - **lacp (IEEE 802.3ad)**
  - **failover**
  - **fec**
  - **loadbalance**
  - **roundrobin**

# Ejercicios

- Crear un par de VLANs
- Crear interfaces específicos para cada VLAN
- Crear grupo de interfaces con las dos interfaces creadas
- Crear otro grupo de interfaces sólo con la LAN

# Firewall

# Aliases

- Permite crear alias por
  - IP (host/red)
  - Puertos
  - URLs
- Pueden ser usados en las reglas
  - Es más fácil recordar nombres que IPs
  - En un único alias podemos tener varias entradas
    - **HOSTS\_PERMITIDOS:** 192.168.100.1, 10.200.20.10, 172.17.22.22...
    - **REDES\_PERMITIDAS:** 192.168.200.0/25, 10.11.22.0/22...

## NAT (Port Forward)

- Desvío de puertos desde un interfaz a un equipo de la red
  - Opción de elegir un rango entero de puertos
  - Posibilidad de elegir origen permitido
  - Tendremos que aceptar conexiones al puerto en dicho interfaz

# NAT (Outbound)

- Podemos realizar NAT de salida
  - Si queremos que cierto tráfico se enmascare con una IP que no sea la de por defecto
  - Lo tendremos que usar si hacemos uso de HA.

# Reglas de filtrado (rules)

- pfSense controla el tráfico entre redes
- Posibilidad de crear reglas de filtrado por interfaz
  - Se ejecutan al **entrar** al interfaz (no en salida)
  - Son evaluadas en base a "primer acierto"
    - La acción que se va a ejecutar va a ser la primera en encontrarse
    - Es importante tener en cuenta el orden (descendente)
    - Cualquier regla que no sea explícitamente **pass**, será bloqueada por defecto

## Reglas de filtrado (rules) (II)

- Acciones principales de las Reglas
  - **Pass:** Permite el tráfico al destino
  - **Reject:** Rechaza el paquete y avisa al emisor
  - **Block:** Rechaza el paquete de manera *silenciosa*
- Acciones secundarias
  - No alteran las acciones anteriores
  - **Marcar:** Para usar con otros servicios, como **traffic shaper**
  - **Log:** Para loggear el tráfico, y hacer diagnósticos

## Reglas de filtrado (rules) (III)

- Diferencias entre **block** y **reject**
  - En caso de duda, mejor usar **block**
    - No damos pistas de que el paquete ha sido bloqueado por un firewall
  - **Reject** responde con **TCP RST** o "port unreachable"
    - Posibilidad de recibir un **ataque**
    - **Nunca usar reject en WAN**
    - En redes privadas es útil, porque avisa a los programas que intentan realizar conexiones: más rápido, no espera timeout

## Ciclo de vida de un paquete

- El paquete, como parte de una nueva conexión, llega a un interfaz
- Se comprueban las reglas de filtrado en orden contra el paquete
- Cuando existe coincidencia, se ejecuta la acción de la regla de filtrado
- Se para las comprobaciones de las reglas
- Si no ha existido una coincidencia tras comprobar todas las reglas, **el paquete es bloqueado por defecto**

# Orden de las reglas de filtrado

- Reglas NAT
  - primero se aplica todo el NAT
- Reglas de servicios
  - reglas añadidas por programas
- Floating rules
- Reglas de **grupos de interfaces**
- Reglas de interfaz específico

# Floating rules

- Son reglas avanzadas que pueden ser aplicadas a más de un interfaz y en cualquier dirección
- No suelen ser necesarias para la mayoría de las redes
  - Salvo que uses **traffic shaper**
- Estas reglas pueden ser sobreescritas por otras reglas
  - Las **floating rules** se comprueban antes, pero se ejecutan al final
    - Una regla de interfaz específico puede sobreescribir esta regla
  - Este comportamiento puede ser modificado si activamos la opción **quick**

# Consejos

- El orden es importante!
  - Asegurar el orden de las reglas, interfaz de entrada, que no haya otra regla en un *grupo de interfaz* o una floating rule con *quick* activado
- Las reglas deben ir de **más específico a más general**
  - Cuanto más criterio/especificaciones tenga la regla, más arriba tendrás que estar
- Las reglas se aplican a la conexión, **no al paquete**
  - Una nueva regla no aplicará a una conexión existente
  - Puedes hacer reset de la tabla de estados para corregirlo

## Consejos (II)

- Una interfaz no tiene reglas por defecto
  - Todo el tráfico será bloqueado
  - Hay que añadir reglas que permitan tráfico en cualquier interfaz opcional
- Por defecto, todo el tráfico que no coincide con reglas, es bloqueado
  - Seguridad por defecto (*whitelist approach*)
  - Lo mejor para las conexiones WAN
- Se puede invertir el comportamiento añadiendo un **permitido todo** como regla wildcard al final de las reglas del interfaz
  - Lo habitual en interfaces LAN (*blacklist approach*)

# Ejercicios

- Crear alias con IP origen
- Poner a la escucha un puerto en el pfSense
  - `nc -l 9999`
- Crear regla en WAN que acepte tráfico a ese puerto
- Crear *floating rule* que bloquee ese puerto
  - ¿qué pasa?
    - Hacer uso de la opción **quick**
    - ¿Se bloquea?

# Servicios

# Captive Portal

- Permite securizar una red a través de una página del portal
- Tipos de autenticación:
  - **Nada:** Simplemente te lleva a una web a modo de aviso
  - **Local:** Hace uso de los usuarios de pfSense con la opción *Captive portal login*
    - **Cupones/Vouchers:** Permite crear cupones de un único uso para un tiempo limitado
  - **Radius:** Hace uso de un servidor externo
- Su uso habitual: redes wifi

## DHCP server

- Permite gestionar un servidor DHCP por cada red
- Creación de distintos pools de rangos
- Creación de mapeo estático MAC ↔ IP
- Control por dirección MAC
- ...

## DNS resolver

- Basado en **unbound**
- Posibilidad de elegir:
  - El interfaz de escucha
  - Habilitar "forwarding mode"
  - Registrar subdominios
- ...

# Load Balancer

- Sistema básico para balanceo de carga
  - Basado en **relayd**
- Permite la distribución de carga de una web sobre distintos servidores
- Se puede reconocer si un servidor se ha caído
- Es útil para desarrollos simples, pero si necesitas más features recomendamos **HAproxy**

# SNMP

- Simple Network Management Protocol
- Sirve para poder monitorizar ciertos estados de pfSense
  - Gráficas de networking
- Es recomendable:
  - Cambiar "Read Community String"
  - Interfaz de escucha sólo en LAN

# Comandos y ficheros importantes

# pfctl

- Controla a **packet filter**
- Tiene muchos parámetros, pero los interesantes
  - **-d**: deshabilita el filtrado de paquetes
  - **-e**: habilita el filtrado de paquetes
  - **-s rules**: muestra las reglas de filtrado
  - **-s nat -i vtnet0**: muestra las reglas NAT de un interfaz
  - **-vv**: muestra información verbose sobre lo anterior

# clog

- pfSense usa el formato **Circular Log**, conocido como **clog**
  - así mantiene un tamaño consistente de los logs
- Los logs no crecen libremente (no nos dejan sin espacio)
  - El tamaño por defecto puede ser pequeño y no permitir mucho log (500KB. 2000-4000 registros)
    - Status → System Logs → Settings
- Por contra, no podemos ver los logs con cat, grep...
  - clog -f /var/log/filter.log

# LOGs

- En el sistema de ficheros
  - Podemos verlos con **clog**
    - en **/var/log**
  - Algunos paquetes no hacen uso de clog
    - Pueden estar en otra ruta, depende del mantenedor del paquete
- Vía web
  - Status → System Logs
    - Nos permite realizar filtros

# Backup

- Web
  - Diagnostics → Backup & Restore
    - Nos permite elegir opciones de backup
    - Podemos realizar una restauración de una config previa
      - Descargada por nosotros
      - Del histórico existente
- Sistema de ficheros
  - **/cf/conf/config.xml**
    - En **/cf/conf/backup/** existe un histórico de configuraciones anteriores

# CRON

# CRON

- Sistema que se encarga de lanzar procesos según un intervalo definido
  - Habitual en los sistemas operativos UNIX
- pfSense cuenta con la instalación de CRON.
  - Fichero de tareas a ejecutar: **/etc/crontab**
- Existe un paquete para realizar configuración vía web
  - System → Package Manager → Available Packages → CRON

# Configuración

- Services → Cron
  - Viene con la configuración que ya aparece en /etc/crontab
  - Podemos añadir nuevos lanzadores
    - **Minute:** minuto de la ejecución
    - **Hour:** hora de la ejecución
    - **Day of month\***: día de la semana concreto para ejecutarlo
    - **Month of the year:** mes concreto para ejecutarlo
    - **day of week:** día de la semana
    - **User:** usuario con el que se va a ejecutar
    - **Command:** ruta completa al comando, con parámetros

# Consejos

- Si un proceso puede tardar mucho, es conveniente que el propio script compruebe si ya se está ejecutando una versión previa
  - La duración de ejecución puede variar, y para no tener que tocar el *scheduling* de ejecución, el propio script debería comprobar si se está ejecutando
- Lanzar el comando con **nice**
  - permite alterar la prioridad del comando a ejecutar, para que sea más o menos prioritario
    - nice -n20 ls
      - lanza *ls* con la prioridad más baja

# Proxy web



# Squid

- Sistema proxy-cache para HTTP, HTTPS, FTP, ...
- Permite modo reverse proxy
- Reduce el ancho de banda
- Permite la gestión de ACLs (control de accesos)
- Modo transparente
  - Evitamos tener que realizar configuración en los clientes web
- Podemos filtrar URLs
  - squidGuard

# Instalación y configuración

- System → Package Manager → Available Packages → Squid
- Configuración:
  - Services → Squid Proxy Server
  - **Enable Squid Proxy**
    - Previamente hay que configurar la caché
  - Habilitar **Transparent HTTP Proxy**
  - **Enable Access Logging**
    - **Cuidado!** puede llenar el disco
    - **/var/squid/logs**

# ACLs

- **Allowed Subnets:** Redes permitidas
- **Unrestricted IPs:** IPs que no se van a restringir
- **Whitelist / Blacklist:** Dominios que son permitidos o restringidos
  - Es un poco pesado tener que ir metiendo a mano dominios
    - Para eso está **SquidGuard**

# SquidGuard

- Sistema proxy de filtrado de URLs para Squid
- Posibilidad de gestionar URLs por categorías
  - Podemos descargar *blacklists* generadas
  - Podemos crear nuestras propias categorías
- System → Package Manager → Available Packages → squidGuard

# Configuración

- Services → SquidGuard Proxy Filter
  - Habilitar **Enable**
  - Habilitar **Blacklist**
    - Descargar lista gratuita (**para uso no comercial y privado**)
    - <http://www.shallalist.de/Downloads/shallalist.tar.gz>
      - Path: **/var/db/squidGuard**
  - **Log**
    - Cuidado al habilitar! Sólo para debugging

# Configuración (II)

- **Common ACL**
  - Comportamiento por defecto para squidGuard
    - **Allow:** Permitimos todo por defecto, y restringimos el acceso a categorías
    - **Deny:** Bloqueamos todo por defecto, y permitimos el acceso a categorías
  - **Redirect Mode**
    - Al entrar a una URL bloqueada, podemos hacer que se redirija la petición
- **IMPORTANTE!** al hacer cualquier cambio, hay que ir a **General settings** y darle a **Apply**

# Log

- Habilitar los logs puede ser peligroso por espacio
  - Sólo para debugging o en caso de bloqueos erróneos
    - En el log podemos ver por qué categoría se ha bloqueado

The screenshot shows a web-based interface for managing a SquidGuard configuration. The top navigation bar includes links for Package, SquidGuard, and Logs. Below the navigation is a horizontal menu with tabs: General settings, Common ACL, Groups ACL, Target categories, Times, Rewrites, Blacklist, Log (which is highlighted in red), and XMLRPC Sync. The main content area is titled "Blacklist Update" and displays a table of log entries. At the top of the table are several buttons: Blocked (highlighted in green), Filter GUI log, Filter log, Proxy config, and Filter config. The table has a header row with columns for timestamp, source IP, destination URL, and action. Below the header, a message says "Show 50 entries starting at << 0 >>". Four log entries are listed:

15.04.2018 22:32:59	192.168.2.111/192.168.2.111	http://playboy.com/	Request(default/blk_BL_porn/-) - GET REDIRECT
15.04.2018 22:32:10	192.168.2.111/192.168.2.111	playboy.com:443	Request(default/blk_BL_porn/-) - CONNECT REDIRECT
15.04.2018 22:31:01	192.168.2.111/192.168.2.111	playboy.com:443	Request(default/blk_BL_porn/-) - CONNECT REDIRECT
15.04.2018 22:31:01	192.168.2.111/192.168.2.111	playboy.com:443	Request(default/blk_BL_porn/-) - CONNECT REDIRECT

# Groups ACLs

- Podemos generar ACLs más específicas, pudiendo seleccionar:
  - **Origen del cliente:** Para poder aplicar reglas por IPs, redes,...
    - **Ejemplo:** En un colegio la red de profesores tiene acceso a todo, pero la de alumnos sólo se permite wikipedia.
  - **Time:** aplicar reglas dependiendo de la hora o el día
  - **Target Rules:** Permitimos o denegamos las categorías que tenemos
  - **Redirect:** Redirección al ser bloqueado

# Target categories

- Podemos crear categorías propias donde añadir:
  - **Domain List:** dominios o IPs
  - **URL List:** URLs completas
  - **Regular Expression:** Expresiones regulares de URLs
    - Documentación en <http://www.squidguard.org/Doc/expressionlist.html>

# Ejercicio

- Descargar desde el interfaz web una blacklist
  - <http://www.shallalist.de/Downloads/shallalist.tar.gz>
- Configurar la **Common ACL**
- Configurar una **Groups ACL** teniendo en cuenta
  - Crear una categoría propia
    - Que bloquee la web <https://blog.irontec.com>
    - Crear una regla de tiempo
    - Habilitar logs y ver el resultado al navegar

# HAproxy

balanceo de carga HTTP/TCP



# Presentación e Instalación

- Sistema HA de balanceo de carga
- Servidor proxy para aplicaciones basados en aplicaciones TCP y HTTP
- Permite controlar el estado de los backends
  - **Health checks**, que si fallan, no se mandan peticiones
- **SSL offloading**
  - Permite quitar carga de procesamiento a los backends
- System → Package Manager → Available Packages → haproxy

# Frontend

- Acepta las conexiones entrantes
  - Tenemos que especificar la IP y el puerto en el que escuchamos
  - Si es el 443, tenemos que cambiar el puerto de escucha de pfSense, o tener más IPs
- Protocolo (TCP o HTTP)
- ACL
  - Dependiendo del host, path, IP...
- Acción
  - Dependiendo del ACL, se pueden ejecutar distintas acciones
  - Normalmente, enviar la petición a un **backend**

# Backend

- Contiene la lista de servidores que responderán la petición enviada desde un *frontend* y la estrategia de balanceo
- **Server List:** Listado de servidores a los que poder enviar la petición
- Tipo de balanceo:
  - **None**
  - **Round Robin**
  - **Static Round Robin**
  - **Least Connections**

# Health checking

- Sistema para monitorizar los backends
  - Podemos crear nuestro propio sistema
- Dependiendo de la respuesta, el backend será accesible o se sacará del listado
  - **200 / 300**: el backend está respondiendo y es accesible
  - **cualquier otro**: el backend no está accesible y será quitado del pool, no recibirá peticiones

# Avanzado

- Redirect HTTP → HTTPS
  - Custom Action:
    - redirect scheme https if !{ ssl\_fc }
- Documentación oficial

# Estadísticas

HAProxy version 1.7.10-a7dcc3b, released 2018/01/02

## Statistics Report for pid 51956

### > General process information

```
pid = 51956 (process #1, nbproc = 1)
uptime = 0d 0h2m35s
system load average: unlimited; ultiminh = 2034
maxsock = 2034; maxconn = 1000; maxpipes = 0
current conn = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/17; idle = 100 %
```

active UP  
 active UP, going down  
 active DOWN, going up  
 active or backup DOWN  
 active or backup DOWN for maintenance (MAINT)  
 active or backup SOFT STOPPED for maintenance  
 Note: "NOLB"|"DRAIN" = UP with load-balancing disabled.

Display option:  
 External resources:  
 • Scope :  
 • Primary site  
 • Hide DOWN servers  
 • Refresh now  
 • CSV export

HAProxy.localState																			Server													
	Queue				Session rate				Sessions				Bytes				Denied				Errors				Warnings				Server			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtie		
Frontend	1	1	-	1	1	2 000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPEN										
Backend	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2m35s UP		0	0	0		0				

pruse																			Server													
	Queue				Session rate				Sessions				Bytes				Denied				Errors				Warnings				Server			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtie		
Frontend	0	8	-	0	7	2 000	12	3 288	21 395	0	0	9	0	0	0	0	0	0	0	0	0	OPEN										

pruse_http_ipANY																			Server													
	Queue				Session rate				Sessions				Bytes				Denied				Errors				Warnings				Server			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtie		
	apache	0	0	-	0	3	0	1	-	6	6	2m16s	2 772	19 514	0	0	0	0	0	0	0	2m35s UP	L7OK200 in 2ms	1	Y	-	0	0	0s	-		
	Backend	0	0	0	3	0	1	200	6	6	2m16s	2 772	19 514	0	0	0	0	0	0	0	0	2m35s UP		1	1	0		0	0s			

Choose the action to perform on the checked servers :  Apply

# Ejercicio

- Crear un backend con al menos dos servidores
  - Uno real, otro ficticio
- Crear un frontend
  - Que escuche en 80 y 443
    - Haga redirect 80 → 443
- Mirar estadísticas
- Parar ambos backends
  - Mirar estadísticas

# Let's Encrypt



## ¿Qué es?

- Let's Encrypt es una **autoridad de certificación**
- Desarrollado y mantenido por **Internet Security Research Group**
  - Mozilla
  - Cisco
  - Akamai
  - ...
  - Sin ánimo de lucro

## ¿Para qué?

- Permite generar certificados X.509 de manera automatizada y sencilla
  - Evitamos los procesos manuales actuales
- Sirve para cifrar las conexiones desde nuestro navegador al servidor

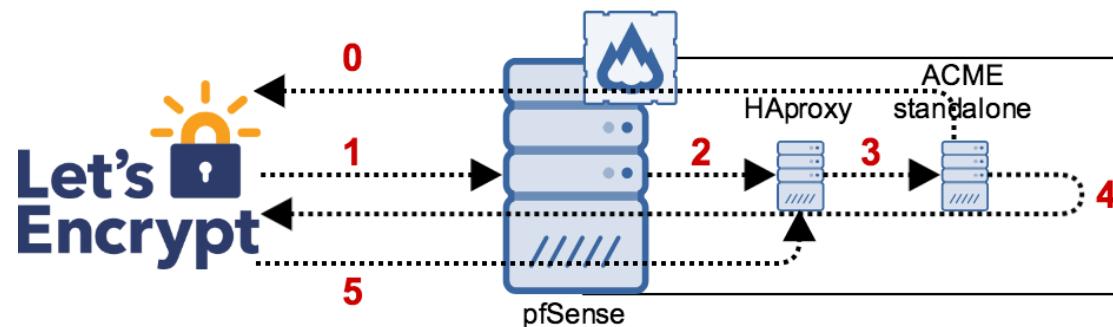
# Configuración

- Generamos **Account keys**
- Activamos el CRON
- Creamos la configuración del certificado
  - El certificado se debe pedir desde el servidor donde se va a instalar
  - Distintos modos de validación
    - DNS
      - Si tenemos esta posibilidad, es la mejor
    - Directorio local (donde se almacena temporalmente un hash)
    - Servidor HTTP temporal

# Integración con HAProxy

- HAproxy
  - Generamos un backend **ACME**, cuyo host sea 127.0.0.1 y puerto 8888
  - Frontend:
    - ACL prioritaria (envíamos al backend si)
      - **Path starts with:** /.well-known/acme-challenge/
- En la configuración del certificado ACME
  - **Domain SAN list:** standalone HTTP server en puerto 8888
  - **Actions list:** reiniciar el haproxy al renovar el certificado
    - /usr/local/etc/rc.d/haproxy.sh restart como **shell command**

# Diagrama de renovación de certificado simplificado



# Flujo de renovación de certificado simplificado

1. El CRON de ACME pide renovar el certificado
2. Let's Encrypt valida la petición del dominio
  - GET `/.well-known/acme-challenge/` al puerto 80
3. El HAProxy está escuchando en el puerto 80, entra al frontend
4. Recibe la petición y se la manda al backend 127.0.0.1:8888
  - es un server standalone levantado por ACME, que manda el CSR
5. Let's Encrypt recibe el CSR, lo firma y nos devuelve el certificado
  - Lo instalamos en el HAProxy y lo reiniciamos

# pfBlockerNG

## ¿Qué es?

- Versión mejorada de pfblocker
- Permite manejar listas de IPs (IPv4/v6)
- Base de datos GeolP
- Bloqueo por DNSBL
  - Domain Name System-based Blackhole List
  - Puede subir la carga del servidor

# Configuración

- **GeolP**
  - Podemos seleccionar por continente o por el top20 de países más spammers
  - Seleccionar a los puertos destino
    - Útil si tenemos port-forwarding
- **General**
  - Podemos elegir en qué orden se meten las nuevas reglas generadas
    - Por defecto se añaden al principio
  - Matar estados cuando se realiza modificación en la configuración

# Configuración (II)

- **Cuidado!!** al habilitar logging
  - Lo mejor es deshabilitarlo en cuanto se ponga en producción
- Al realizar modificaciones:
  - Update → "Select 'Force' option"

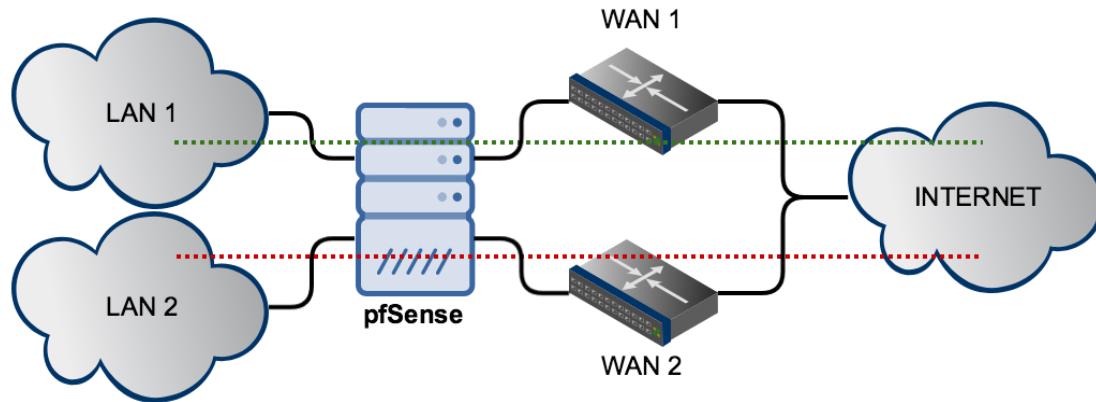
The screenshot shows a network configuration interface with a top navigation bar featuring tabs for 'Floating', 'WAN', and 'LAN'. The 'WAN' tab is currently selected, indicated by a red underline. Below the tabs is a section titled 'Rules (Drag to Change Order)'. This section contains a table with the following data:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/> ✘ 0 / 0 B	IPv4 *	pfB_Africa_v4	*	*	*	*	none		pfB_Africa_v4 auto rule	
<input type="checkbox"/> ✘ 0 / 0 B	IPv4 *	pfB_Asia_v4	*	*	*	*	none		pfB_Asia_v4 auto rule	
<input type="checkbox"/> ✘ 0 / 0 B	IPv4 *	pfB_Oceania_v4	*	*	*	*	none		pfB_Oceania_v4 auto rule	

# Ejercicio

- Seleccionar distintas listas de GeolP
- Comprobar los **alias** generados
- Modificar el orden de las reglas:
  - Añadir como Floating Rules
  - Cambiar orden
  - Comprobar las reglas generadas

# Multi-WAN



# Configuración

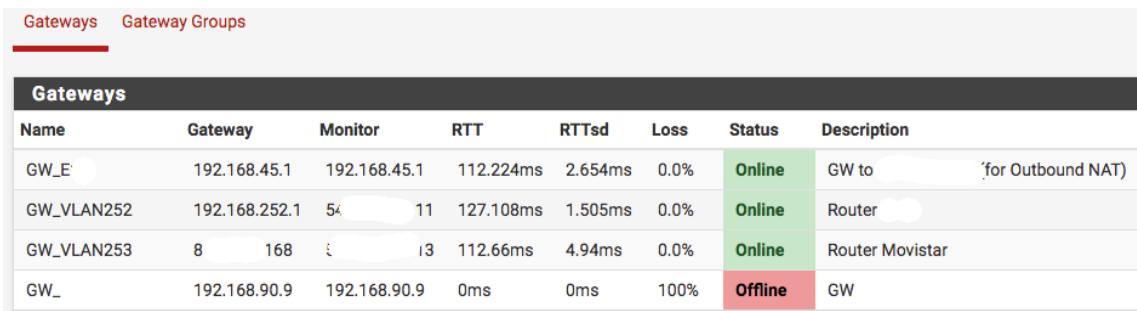
- **Interfaces:** Tantos como WANs tengamos
  - Cada WAN tendrá su propio gateway
- **Crear gateway groups**
  - System → Routing
  - Añadir los WAN correspondientes, con el **tier** (peso) estimado
  - **Trigger Level**
    - **Member Down:** la IP monitorizada tiene 100% de perdidas
    - **Packet Loss:** Cuando hay perdida de paquetes
    - **High Latency:** Cuando hay latencia al gateway
    - **Packet Loss or High Latency:** Cualquiera de las condiciones.

## Configuración (II)

- Firewall Rules → LAN
  - Modificar el **gateway** por un grupo
- System → General Setup
  - Añadir un servidor DNS por cada WAN gateway
- **Outbound NAT**
  - En caso de tenerlo en *manual*, hay que crear una nueva regla para la nueva WAN

# Monitorización

- Status → Gateways



The screenshot shows a user interface for monitoring network gateways. At the top, there are two tabs: "Gateways" (which is selected) and "Gateway Groups". Below the tabs is a table with the following data:

Name	Gateway	Monitor	RTT	RTTsd	Loss	Status	Description	
GW_E	192.168.45.1	192.168.45.1	112.224ms	2.654ms	0.0%	Online	GW to [redacted] (for Outbound NAT)	
GW_VLAN252	192.168.252.1	54	11	127.108ms	1.505ms	0.0%	Online	Router [redacted]
GW_VLAN253	8.168.253.1	13	112.66ms	4.94ms	0.0%	Online	Router Movistar	
GW_	192.168.90.9	192.168.90.9	0ms	0ms	100%	Offline	GW [redacted]	

# Alta disponibilidad (HA)

# HA

- Permite tener dos servidores
  - Uno será *master* y otro *failover*
- Es necesario un interfaz para sincronización entre nodos
  - A través de él, se sincronizará la información
  - Lo ideal suele ser cable directo
- Por cada red, se necesita una IP balanceada

# Interfaz de sincronización

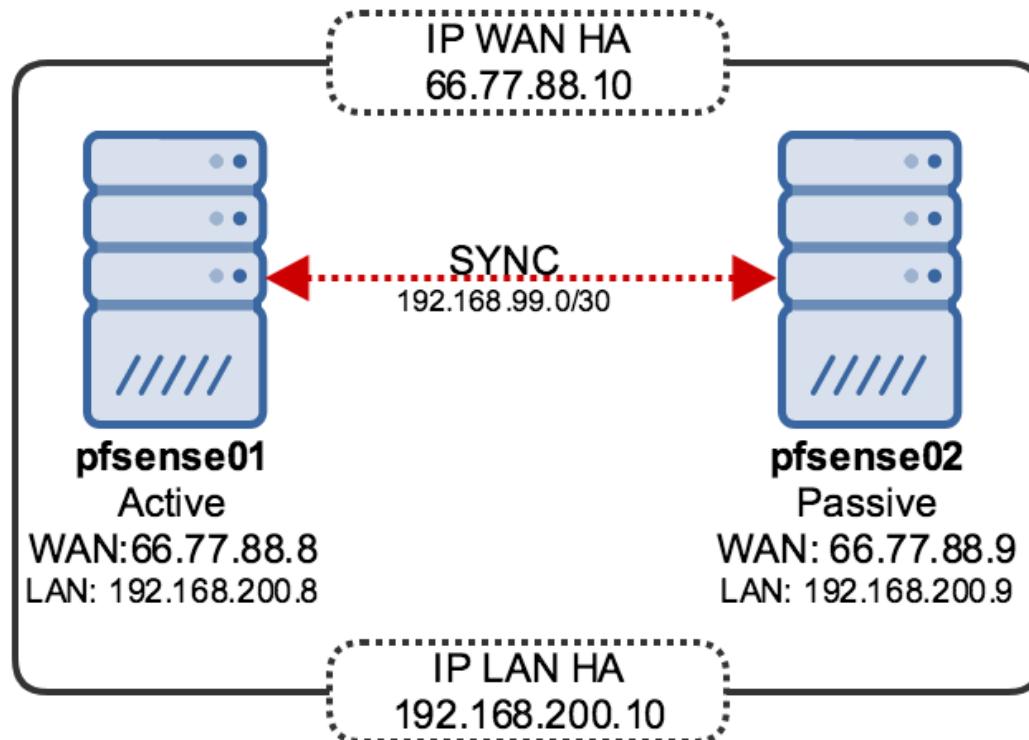
- Cada nodo tiene que tener un interfaz dedicado
  - un /30 de cualquier red que no conflicte
  - Le damos un nombre tipo SYNC
- **Importante!!** crear regla para aceptar todo el tráfico en este nuevo interfaz
  - necesario para la sincronización de datos y estados

# IPs virtuales (CARP)

- Cada nodo tiene su propia IP
  - Tanto públicas como privadas
- El HA necesita IPs balanceadas
  - Son virtuales (CARP)
    - Firewall → Virtual IPs
    - **Importante!!** Tienen que tener un VHID único por cada IP en el mismo interfaz!
  - Balancearían del master al esclavo en caso de necesidad
  - **Importante!!** Se crean en el nodo **master**

# HA

- Esquema de red de IPs CARP



# Habilitar sincronización

- System → High Avail. Sync
- State Synchronization Settings (pfsync)
  - Habilitamos **Synchronize states**
  - Seleccionamos en **Synchrone Interface** el dedicado
  - En **pfsync Synchronize Peer IP** la IP del otro servidor
- Configuration Synchronization Settings (XMLRPC Sync)
  - **Importante!!:** sólo en el master!
  - **Synchronize Config to IP:** IP del failover
  - Credenciales de acceso
  - Seleccionamos todas las opciones a sincronizar

# NAT: outbound

- Tenemos que hacer que todo el tráfico saliente salga con la IP balanceada (CARP)
  - Firewall → NAT → Outbound:
    - Seleccionamos **Hybrid Outbound NAT rule generation**
    - Creamos regla:
      - **Interfaz:** WAN
      - **Protocol:** any
      - **Source:** nuestra red LAN
      - **Destination:** any
      - **NAT Address:** ip WAN\_CARP

## DHCP en red LAN

- En la LAN suele ser habitual tener DHCP
- Tenemos que asegurarnos que el gateway entregado es la IP balanceada
  - si no, podríamos dejar tener acceso hacia el exterior!

# Comprobar status

- Status → CARP (failover)

Status / [CARP](#)

[Temporarily Disable CARP](#) [Enter Persistent CARP Maintenance Mode](#)

CARP Interfaces		
CARP Interface	Virtual IP	Status
WAN@1	10.10.0.110/24	<span>▶</span> MASTER
LAN@1	192.168.1.10/24	<span>▶</span> MASTER

pfSync Nodes		
pfSync nodes:	206509a2 8a4e29a7 bcc6bb0b cfb9c91a f18c7a22	

# Traffic Shaper

## ¿Qué es?

- Control del tráfico de red para optimizar o garantizar el rendimiento, la baja latencia y/o el incremento del ancho de banda a usar, retrasando los paquetes que coincidan con ciertos criterios
  - Usado para Quality of Service (QoS)
  - También para simular condiciones de red

# Control de latencia

- La **latencia** puede ser añadida usando una cola FIFO
  - Los paquetes son almacenados en una cola, y se envían a su destino un tiempo después
  - Lo que queremos es minimizar ese tiempo de encolado
    - La red ideal es aquella que no tiene colas, pero no es posible en la práctica
  - Podemos usar las colas para
    - Retrasar paquetes para cuando la red esté menos congestionada
    - Simular una ruta más larga al destino

## Control de ancho de banda

- ¿Cómo controlarmos el ancho de banda?
  - Normalmente usando un ratio
    - bits/unidad de tiempo
    - paquetes/unidad de tiempo
  - En realidad es mucho más complejo
    - **1 paquete/segundo != 60 paquetes/minuto**

## Control de ancho de banda (II)

- Queremos tener en consideración:
  - **velocidad media de datos**: en un promedio largo de tiempo
  - **burst rate**: ratio máximo de datos en un momento dado
  - **burst frequency**: frecuencia con la que los datos pueden alcanzar dicho ratio máximo

# Limitadores

- La forma más básica de traffic shaping: **Limiters**
- Se definen una o más *tuberías*
  - Cada una tiene un ancho de banda **maximo** configurado
  - No pueden usar más ancho de banda que el configurado, aunque la conexión lo permita
    - Puede llevar a perder recursos/infrautilización de la red
- El tráfico es clasificado (en las reglas del firewall) y asignado a una tubería

# ALTQ schedulers

- ALTQ es un framework que proporciona distintos modos/algoritmos de encolar los paquetes salientes de un interfaz.
  - Algunos modos son más complejos que otros
    - Un scheduler complejo mal configurado puede dar peor rendimiento que uno más simple bien configurado
  - Cada algoritmo tiene sus propias peculiaridades
- `man altq`

## ALTQ schedulers (II)

### Controlled Delay (CoDelQ)

- Realmente no es un "moldeador" de tráfico
- No hay que crear reglas en el firewall
  - No diferencia tipo de tráfico
- Sólo 1 opción: ancho de banda
- Se puede habilitar en cualquier interfaz
- CoDel está diseñado para evitar que los buffers de los interfaces estén llenos
  - esta congestión genera latencia

## ALTQ schedulers (III)

### Fair Queuing (FairQ)

- Diseñado para dar a todas las conexiones un acceso justo al ancho de banda de la cola
- No funciona bien con aplicaciones que generan muchas conexiones
  - ejemplo: BitTorrent
- Se crea una cola o más por interfaz
- Las reglas del firewall agrupan el tráfico en esas colas
- Resumen:
  - Hace una especie de round-robin de las conexiones

## ALTQ schedulers (IV)

### Priority Queuing (PriQ)

- Se separa el tráfico por prioridades
- 2 niveles de colas FIFO anidadas
  - cola **root**: se le pone el ancho de banda total del enlace
  - 2 o más colas hijas
    - Cada cola tiene una prioridad de 0 a 7 (**7 es lo más prioritario**)
    - Los paquetes son entregados de la cola más prioritaria a la menos
      - Si dos colas tienen la misma prioridad, entonces se usa round-robin
- Puede haber problemas al entregar tráfico poco prioritario

## ALTQ schedulers (V)

### Class Based Queuing (CBQ)

- 2 o más niveles de colas FIFO anidadas
  - cola **root**: se le pone el ancho de banda total del enlace
  - 2 o más colas hijas
    - Cada cola tiene una prioridad de 0 a 7 (**7 es lo más prioritario**)
    - Cada cola tiene un ancho de banda (sumados todos, iguala al de la cola padre)
    - Los paquetes son entregados de la cola más prioritaria, *hasta que se llega al límite del ancho de banda*
    - Las colas pueden *robar* ancho de banda de otras colas si no se están usando

## ALTQ schedulers (V)

### Hierarchical Fair Packet Scheduler (HFSC)

- Se construye sobre CBQ
  - Lo extiende añadiendo *curvas no-lineales de servicio*
- Ejemplo: árbol de 2 niveles HFSC
  - Cola **root**: [50mbps]
    - SSH [S(45mbps,0.1mbps,50ms)]
    - BitTorrent [S(0mbps,5mbps,50ms)]
    - HTTP [S(30mbps,44.9mbps,50ms)]
      - el segundo número debería sumar el ancho de banda total -

# Diseñando estrategias

- Debes de conocer tu red
- Capturar tráfico te ayudará
  - Tráfico de producción
- Habrá que tomar asunciones
  - Sobre éstas, construir las reglas

## Limitadores (II)

- Firewall → Traffic Shapper → Limiters
- Se puede asignar ancho de banda en base a un scheduler
- Podemos crear limiters anidados
  - A los hijos de un limiter, no se les puede asignar ancho de banda (cogen el del padre)
  - Los limiters hijos pueden tener prioridad/peso
- Si elegimos una mascara, se creará un limiter dinámico por cada IP destino/origen teniendo en cuenta la mascara elegida.
- Diagnostics → Limiter info

## Limitadores (III)

- Los limiters se añaden en las reglas del Firewall
- Normalmente, queremos asignar a todo el interfaz
  - Si sólo tenemos una regla de salida, mejor aquí
  - Si tenemos más reglas, tendríamos que añadirlo a una *floating rule*.
    - **Action:** Match
    - Cuidado al añadir el limiter en el interfaz como inbound/outbound.

# ALTQ wizard

- Firewall → Traffic Shapper → Wizards → Multiple Lan/Wan
- Elegimos:
  - Número de WAN y LAN que tenemos
  - Priorizar VoIP
  - Penalizar ciertas IPs
  - Penalizar P2P
  - Penalizar juegos
  - Otros protocolos
    - VNC, HTTP, Git, DNS, Crashplan...

# ALTQ wizard (II)

- Todas las reglas que genera el wizard están en Floating Rules

Rules (Drag to Change Order)											Actions
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	  0 / 0 B	IPv4 *	192.168.2.5	*	*	*	*	qOthersLow		Penalty Box	 
<input type="checkbox"/>	  0 / 0 B	IPv4 UDP	*	*	*	5060 - 5069	*	qVoIP		m_voip Asterisk outbound	 
<input type="checkbox"/>	  0 / 10 KIB	IPv4 UDP	*	*	*	10000 - 20000	*	qVoIP		m_voip Asterisk outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 TCP	*	*	*	6881 - 6999	*	qP2P		m_P2P BitTorrent outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 UDP	*	*	*	6881 - 6999	*	qP2P		m_P2P BitTorrent outbound	 
<input checked="" type="checkbox"/>	  0 / 0 B	IPv4 TCP	*	*	*	412	*	qP2P		m_P2P DirectConnect outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 TCP	*	*	*	4661 - 4665	*	qP2P		m_P2P eDonkey2000 outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 TCP	*	*	*	6699	*	qP2P		m_P2P WinMX outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 TCP	*	*	*	6112 - 6119	*	qACK/qGames		m_Game Battle.NET-game1-tcp outbound	 
<input type="checkbox"/>	  0 / 0 B	IPv4 UDP	*	*	*	6112 - 6119	*	qGames		m_Game Battle.NET-game1-udp outbound	 

# Documentación

# Enlaces de interés

- Wiki oficial
- Foros oficiales
- Vídeos explicativos
  - Traffic Shaper
- Documentación Squid
- Documentación HAProxy
- Let's Encrypt

# Gracias!

Foto portada [Tanel Teemusk](#)