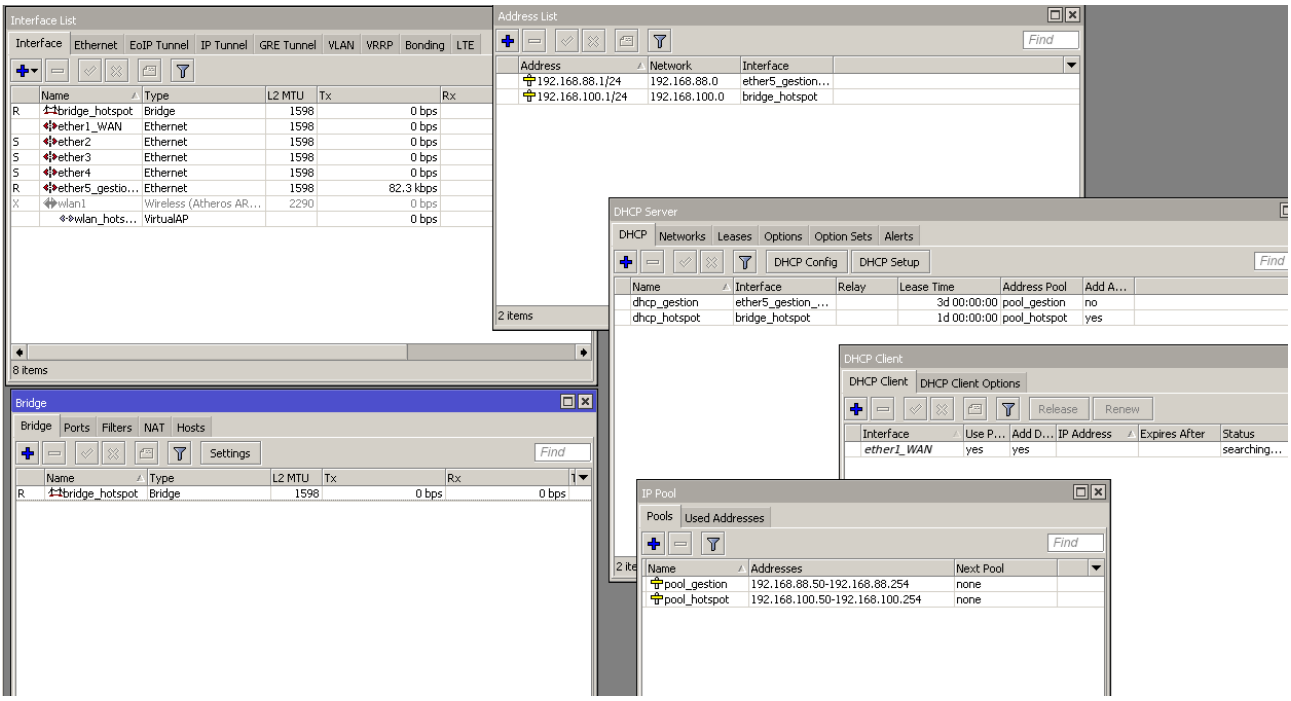
HOTSPOT MIKROTIK

Básico, para poner un hotspot en nuestra conexión a internet y así darte a conocer para compartir gastos de un proyecto de nodo guifi.net

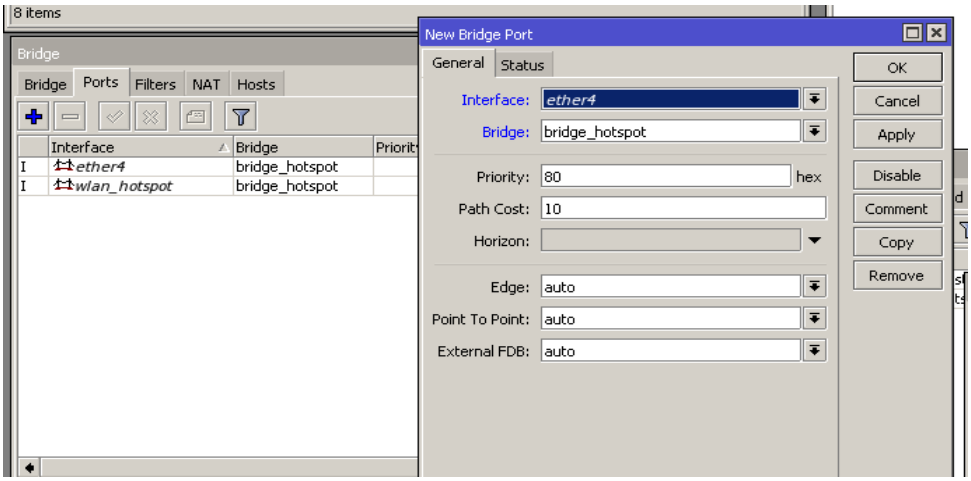
Elegir Mikrotik y licencia, recordar que para IPV6 hace falta una licencia superior que no viene de serie. Se puede instalar el sistema operativo RouterOS (Linux) en un ordenador básico y sacarle aún más partido.
Mikrotik se usa habitualmente por que la relación calidad/precio es muy buena, ofrece funciones de enrutadores de gama alta. A medida que nuestra red se haga grande podremos darle distintos usos. Podemos usarlo directamente como router en nuestra conexión de fibra, ADSL, Cable, como balanceador, como proxy/cache, de firewall. Se pueden usar para interconectar redes como la red guifi.net.
Hay mucha documentación disponible y los distintos modos de configuración que vienen muy bien para aprender más de las redes.
Usaremos entorno grafico, el WINBOX, con una consola muy fácil de usar.

Actualizar la última versión WiBox disponible , tendremos nuestro router con el firmware más reciente.

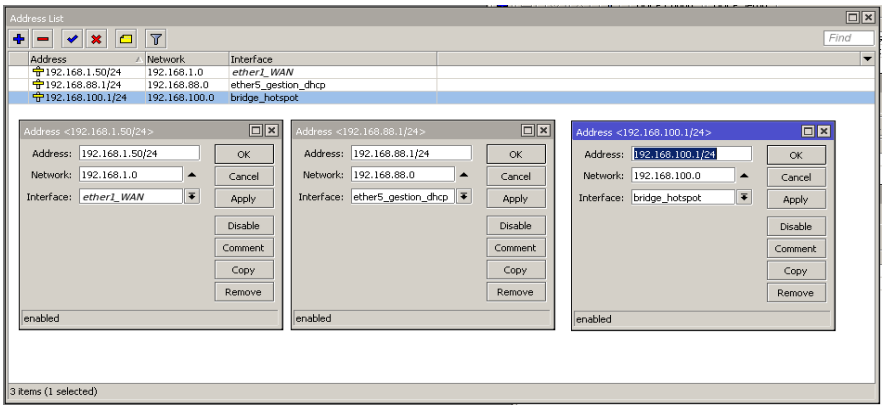
Para empezar hacemos un reset configuration desde SYSTEM lo ponemos a 0 y a jugar con el winbox. En el primer arranque aparece un dialogo para cargar una configuración por defecto que es un router neutro y también nos da opción a cargárnoslo todo. Lo dejamos en blanco y hacemos una conexión por winbox contra la MAC
Cada toma de ethernet tiene una dirección MAC



INTERFACE LIST
Para tener contolado el trafico y más cosas que de momento no usaremos
BRIDGE
Vamos a pestaña PORTS y metemos en un brige las interfaces que van a captar estaciones al HS, el bridge une interfaces si vamos a captar por solo 1 interface no es necesario trabajar con bridge.

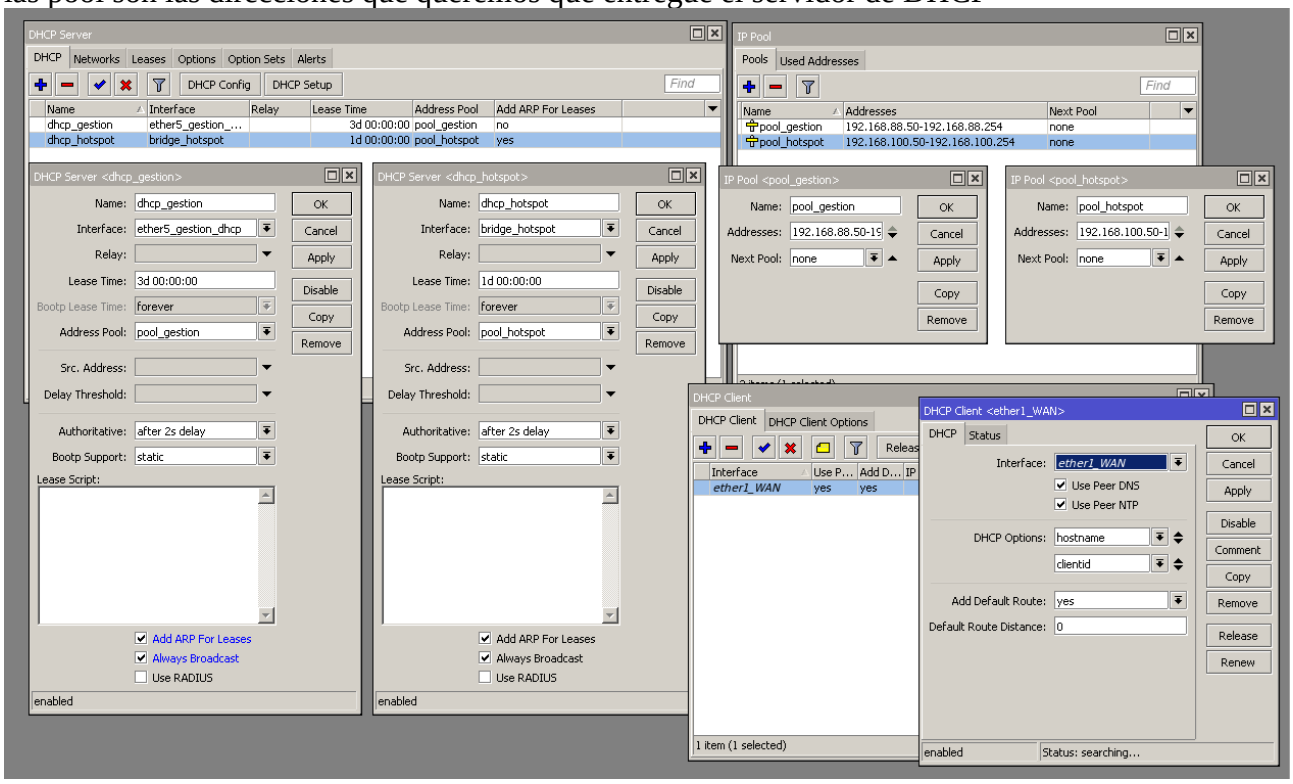


ADDRESS LIST
Vamos a crear direcciones, en este caso,
Para que el HS tenga salida a internet.
Para captar estaciones.
Para gestion.

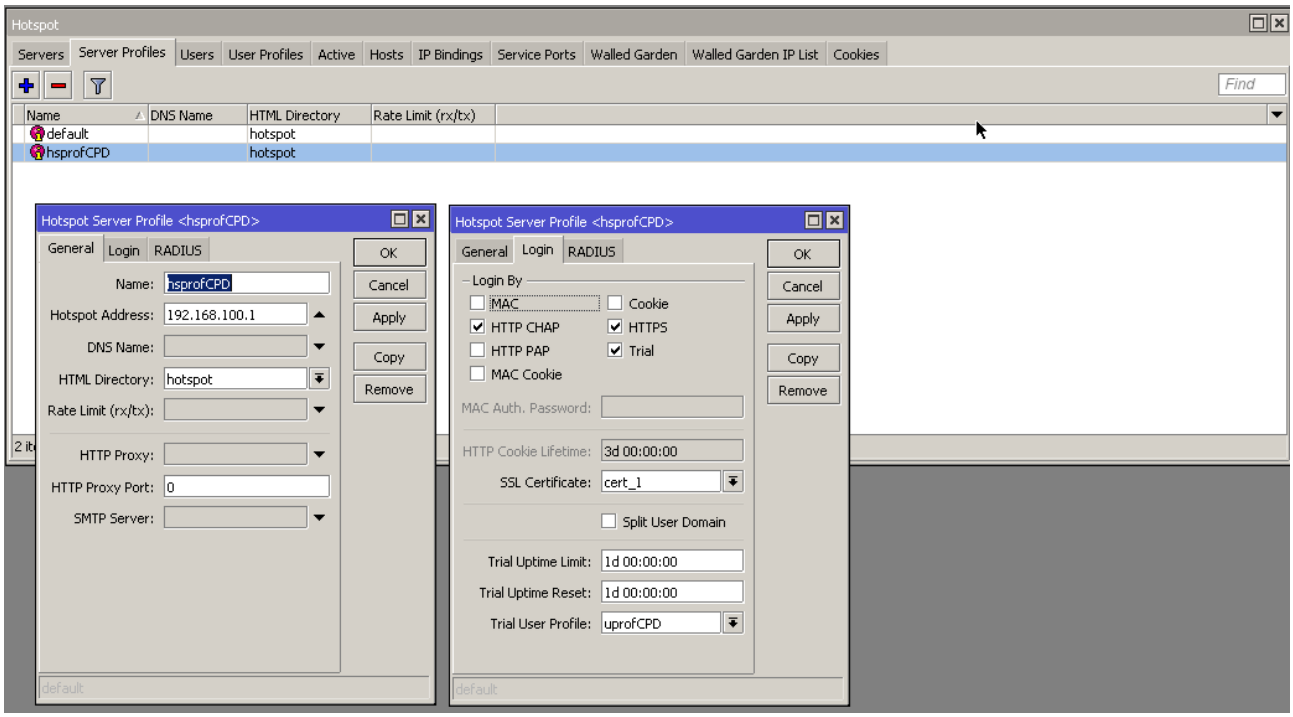


Son tres redes con mascara /24, 255 maquinas por red. De distinto rango. Las redes son independientes y las podemos manejar con otras opciones del router para que trabajen juntas o de forma independiente o compartiendo servicios, lo que necesitemos.

Servicios de DHCP.
Aquí se puede ver un ejemplo de configuración:
DHCP Server, que entrega IP y el
DHCP client que adquiere ip
las pool son las direcciones que queremos que entregue el servidor de DHCP

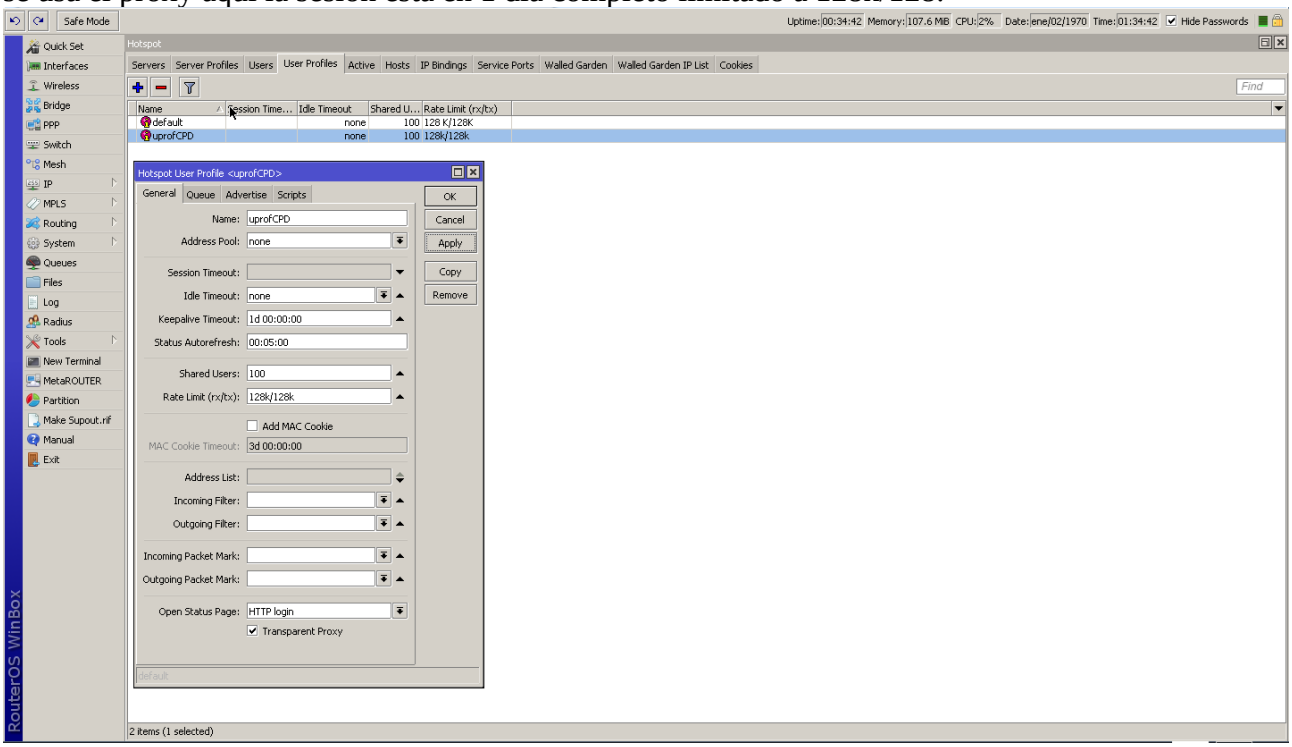


PESTAÑA HOTSPOT
Ejecutamos el SETUP y vamos contestando el dialogo de configuración.
Seleccionamos la interface o bridge para captar clientes, ponemos un nombre y luego next hasta terminar.
Con esto se crea un nuevo perfil en la pestaña SERVER PROFILES que se deja en este caso así:



El Login By HTTPS si no tenemos un certificado instalado no funcionará.
La estación no se contecta al Hotspot hasta que no se valida en el portal cautivo o página de bienvenida en /hotspot/login.html
Para validarse en el hotspot,
1 el cliente se conecta a la red del hotspot y obtiene ip
2 el cliente abre cualquier pagina web con su navegador y se redirecciona a la página de bienvenida, donde se valida.
Se pueden crear usuarios y usar un sistema de usuario contraseña o tener un usuario Trial, de prueba=”Para navegar al hacer click aqui”.
Si el cliente (la estación), tiene como pagina de inicio el buscador google con https, esta redireccion no funcionara y se mostrara en el navegador “Pagina no encontrada”
Por esta razón. Es interesante trabajar con un certificado aunque sea auto firmado.

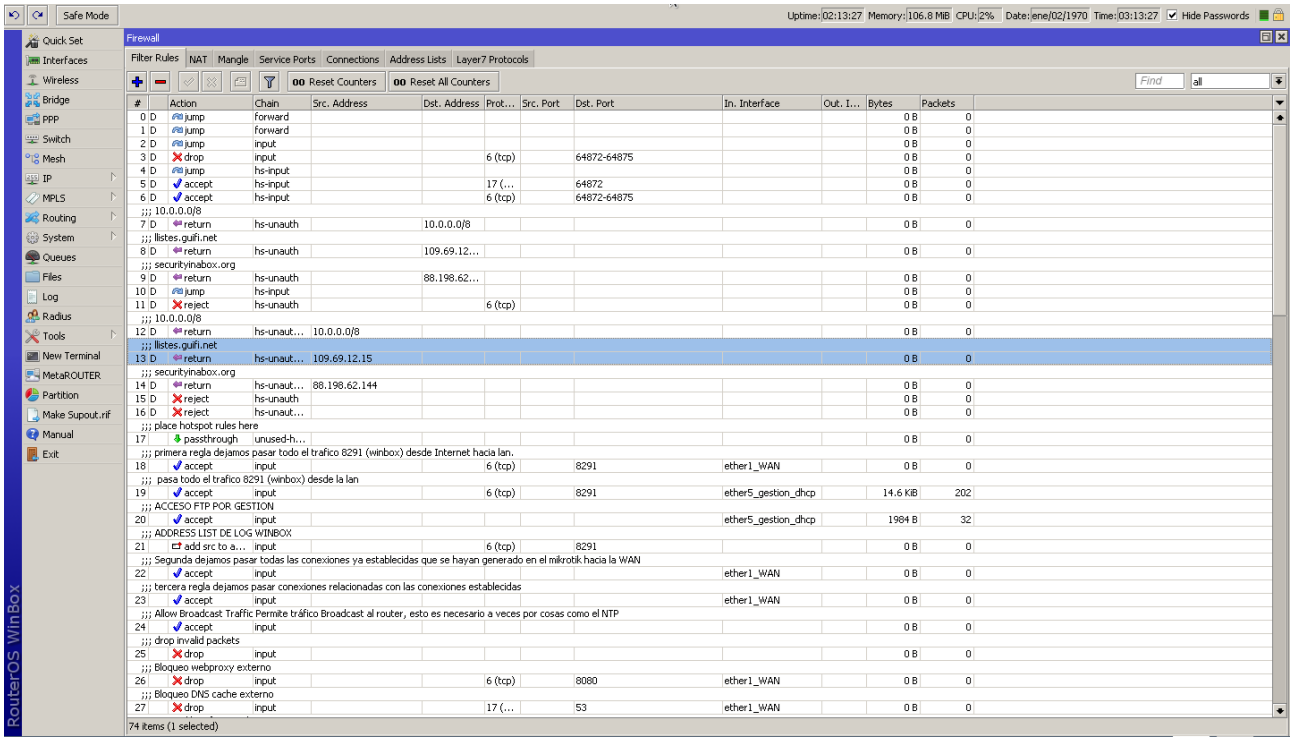
En User Profiles crear un perfil, aquí es donde limitamos el trafico y el tiempo de sesion, marcar si se usa el proxy aquí la sesion está en 1 dia completo limitado a 128k/128.



Active Host Ip Bindings nos van a dar información de clientes conectados, los Garden son para navegacion libre, aunque no pueda validarse la sesión en el portal cautivo.

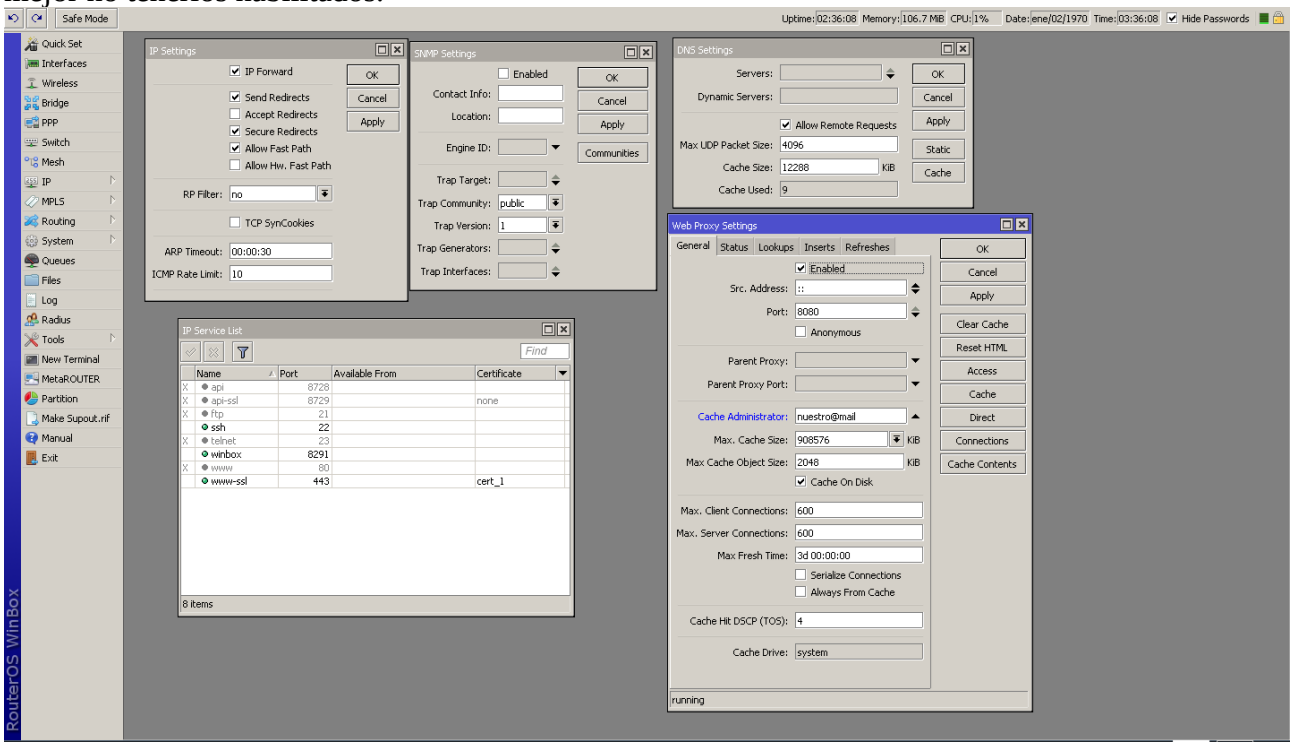
/ip hotspot walled-garden
add comment="place hotspot rules here" disabled=yes
add comment="place hotspot rules here" disabled=yes
add dst-host=guifi.net
add dst-host=*.guifi.net*/*
add dst-host=gg.google.com
add dst-host=maps.google.com
add dst-host=khm0.google.com
add dst-host=khm1.google.com
add dst-host=khm2.google.com
add dst-host=khm3.google.com
add dst-host=mt0.google.com
add dst-host=mt1.google.com
add dst-host=mt2.google.com
add dst-host=mt3.google.com
add dst-host=maps.gstatic.com
add dst-host=gg.google.com
add dst-host=id.google.com
add dst-host=www.google-analytics.com
add dst-host=www.heywhatsthat.com
add dst-host=ocsp.thawte.com
add dst-host=https://securityinabox.org/es*/*
add dst-host=http://www.eltiempo.es/*
add dst-host=http://elrollodecepeda.wordpress.com/*
add dst-host=http://cepedadelamora.blogspot.com/*
add dst-host=https://llistes.guifi.net/sympa/info/guifi-gredos
add dst-host=https://llistes.guifi.net/*
add dst-host=http*://securityinabox.org/es*/*
add dst-host=https://*llistes.guifi.net*/*
add dst-host=https://llistes.guifi.net/sympa/subscribe/guifi-gredos
add dst-host=https://llistes.guifi.net/sympa*/*
add dst-host=https://llistes.guifi.net/sympa/*
add dst-host=http://bin-short.whatsapp.net/
add dst-host=http://cepedadelamora.blogspot.com.es/
/ip hotspot walled-garden ip
add action=accept comment=10.0.0.0/8 disabled=no dst-address=10.0.0.0/8 (red guifi.net)
add action=accept comment=llistes.guifi.net disabled=no dst-address=109.69.12.15
add action=accept comment=securityinabox.org disabled=no dst-address=88.198.62.144

FIREWALL



Para el firewall el windox ayuda pero la consola es más facil.
Para trabajar con el firewall está bien habilitar el SAFE MODE para no perder la gestión del cacharro al poner alguna regla, y hacer backups.

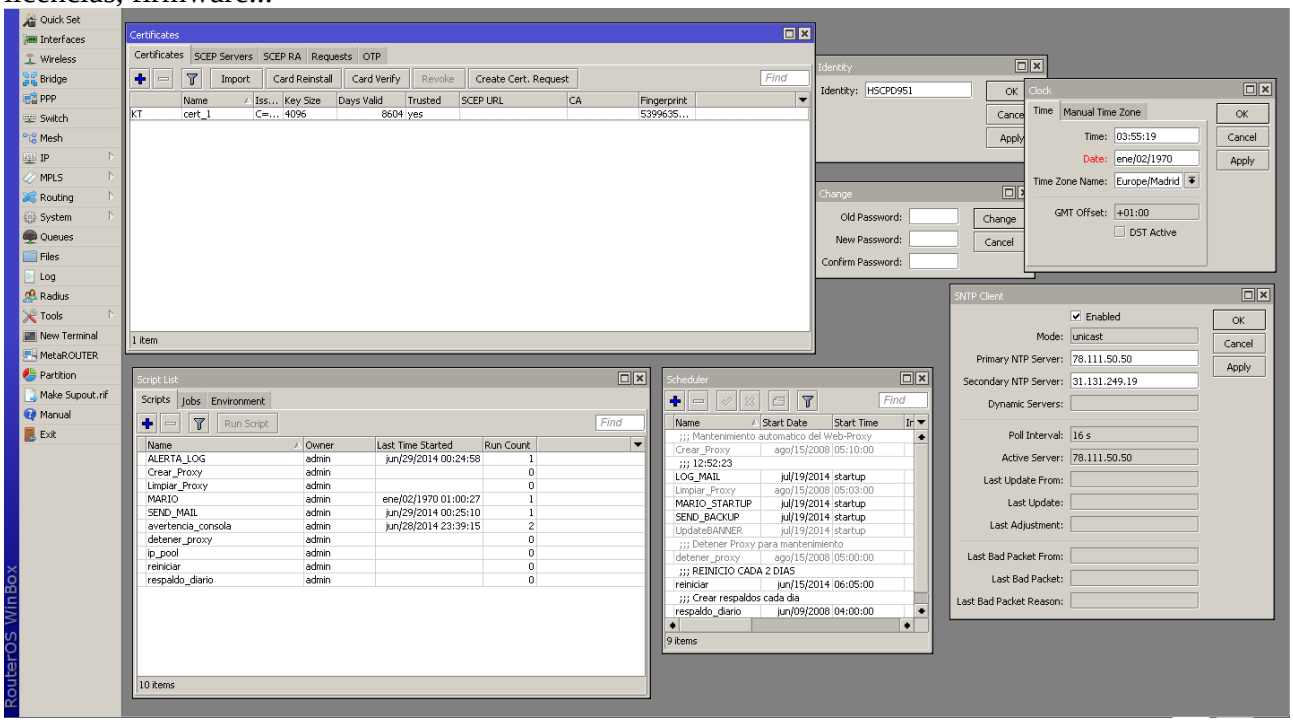
Ajuestes IP
Vamos a dejarlo como está, en DNS se puede especificar un servidor concreto como OpenDNS para control infantil o usar las de otro proveedor o no poner no poner nada y las toma automaticamente.
Web Proxy si queremos usar una cache transparente para que carguen más rapido las páginas es necesario usar almacenamiento externo.
IP Service List son los puertos que tiene habilitado el router para la gestión, si no se van a usar es mejor no tenerlos habilitados.



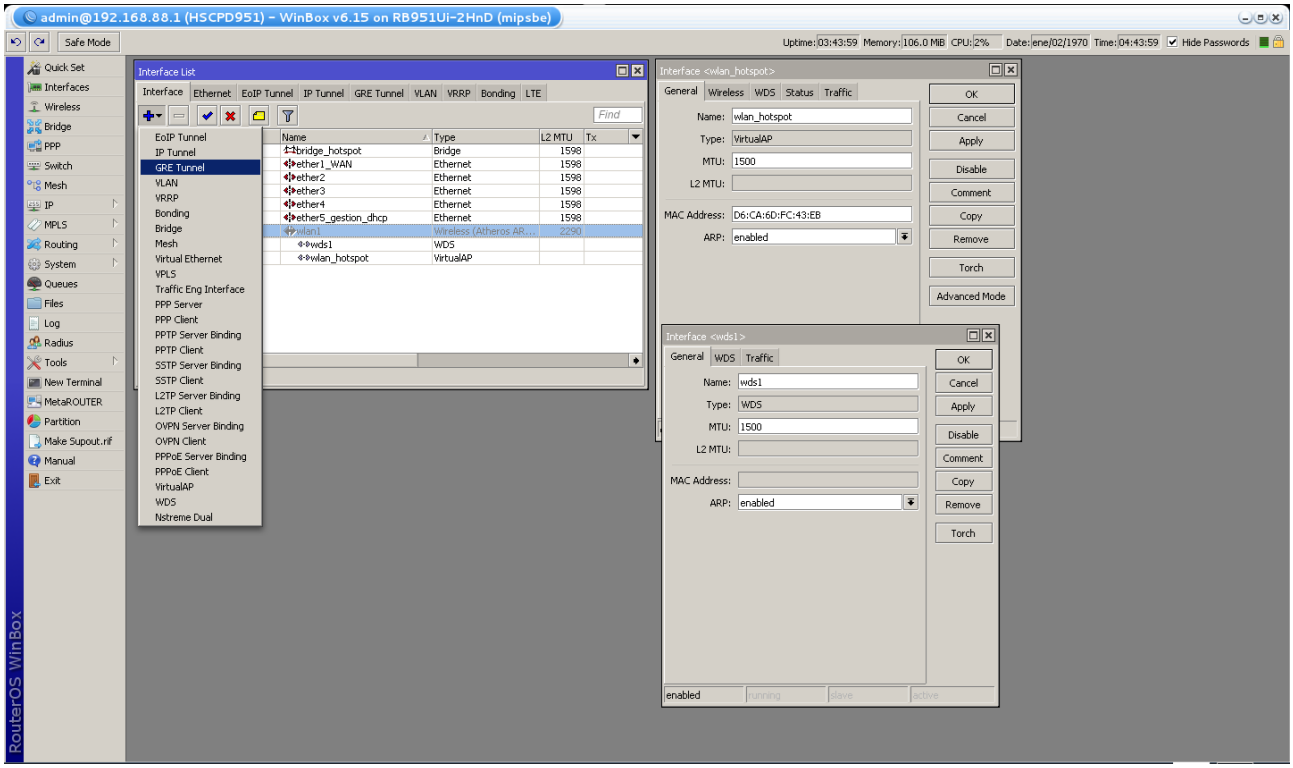
System
Aquí cargamos los certificados
El hostname en Identity
Password
Configurar SNTP nos pone el reloj en hora en el momento que nos conectemos a internet.
SCRIPT , podemos encadenar sentencias o programas para ejecutar cuando pase algo. Con el Scheduler programamos tareas, lanzar Script o comandos en el tiempo.

Pendiente: Hacer un sript que almacene informacion de Log, systemhealt, time date %CPU clientes conectados, intentos de conexión ftp ssh winbox, en un solo archivo txt y lo envie por mail cada 6 horas.

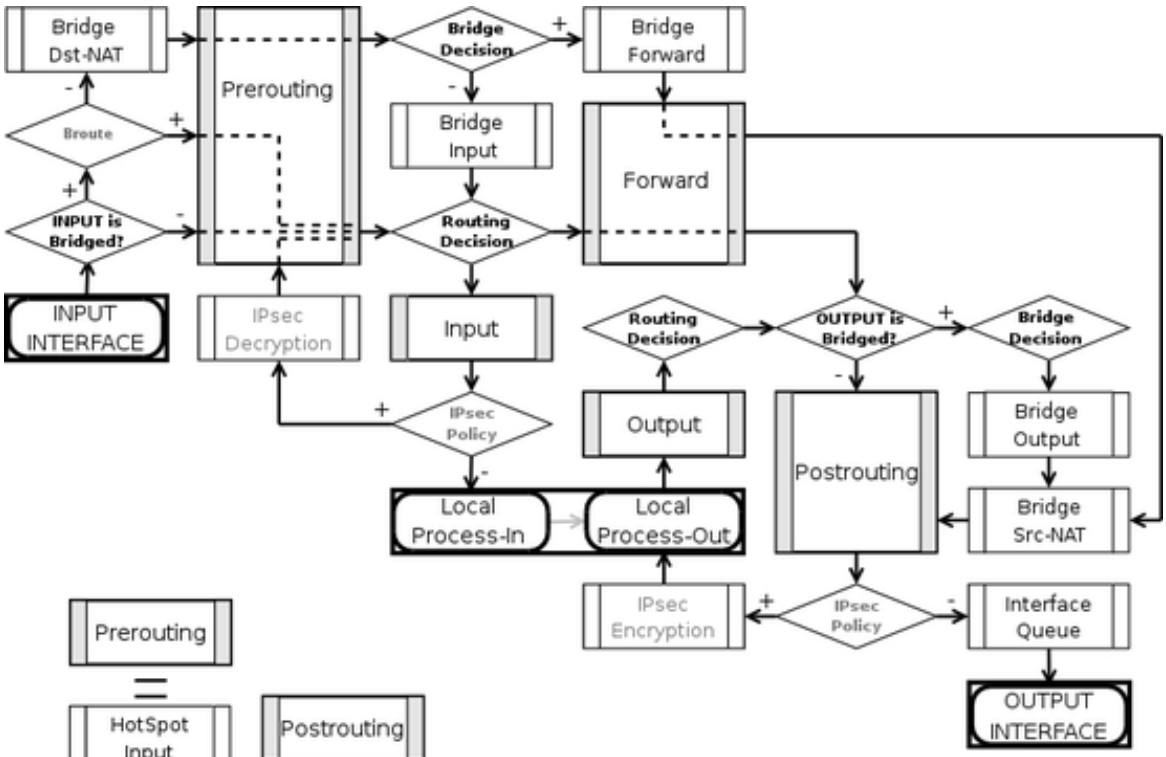
Hay mucho más en System que conviene tener supervisado, está toda la informacion del trasto, licencias, firmware...

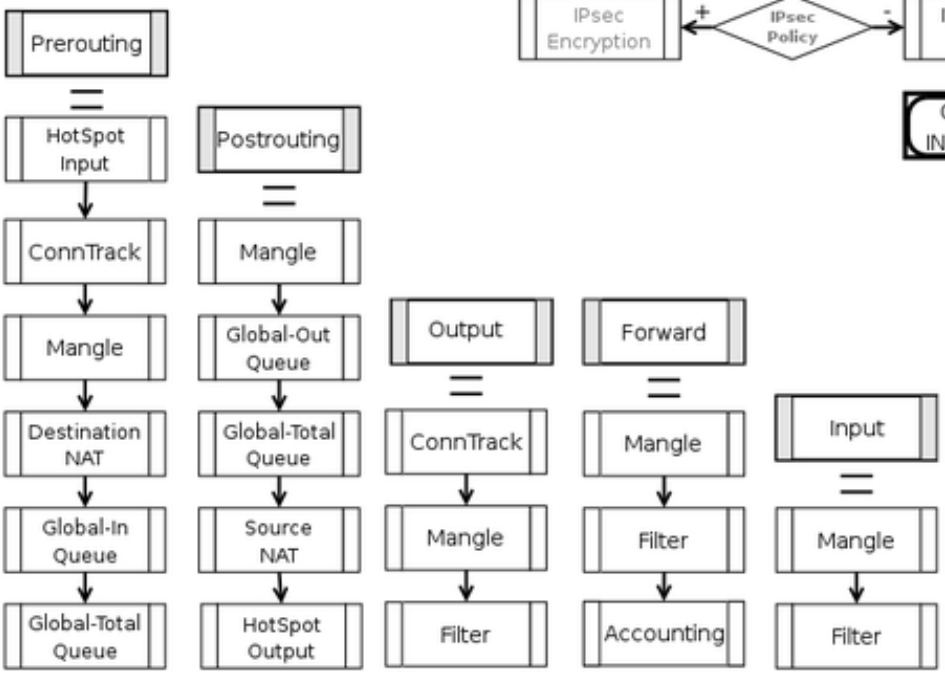


Radio
Para captar estaciones podemos usar un router con wifi en modo AP bridge si se puede y conectarlo a un interface que esté en el bridge del hotspot, (hay que deshabilitar el DHCP del router).
Al captar clientes si estamos limitando el tiempo de sesión, el router identifica las estaciones por MAC, si no captamos las estaciones en modo bridge, a nivel MAC, capa 1, el router va a ver la mac del AP que capta la estación, haciendo imposible la gestión del limite de tiempo por sesión.



Flujo de los paquetes dentro del firewall.
El firewall por defecto lee las reglas de arriba hacia abajo y sale con la primera que matchea. Se usa la orden passthrough para obligar a que, luego de cumplirse una regla, se sigue con las demás. En su naturaleza y funcionamiento es muy similar a iptables.

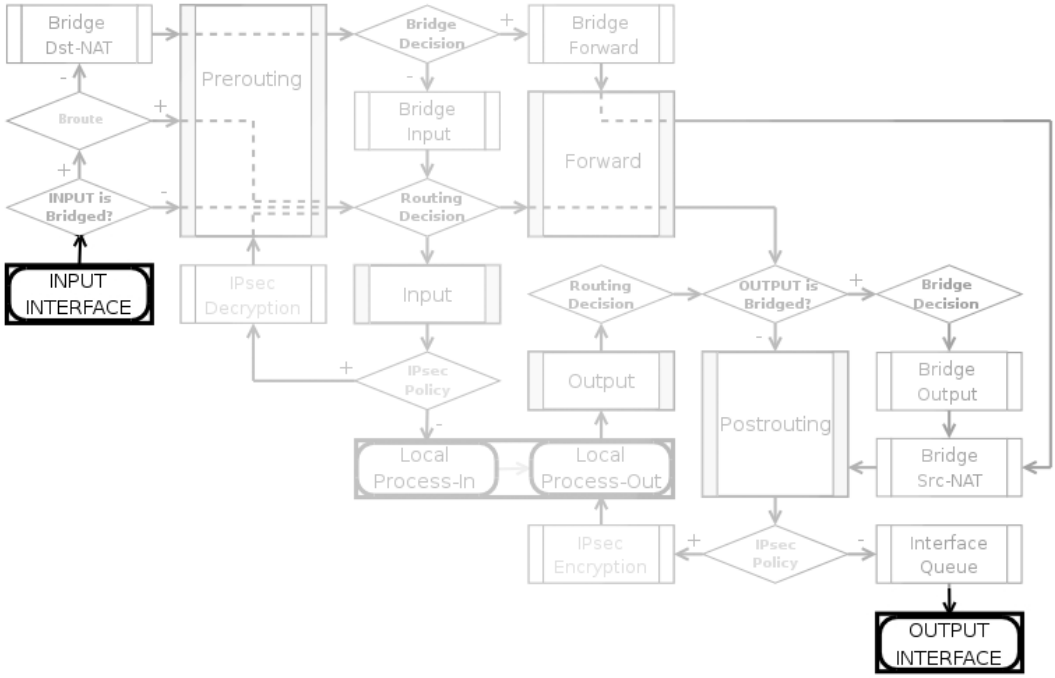




Marca el Inicio y el Fin del diagrama de flujo

Expresa una condición

Llama a un procedimiento



[INPUT INTERFACE]: Normalmente son llamadas interfaces consumidoras, debido a que es en esta **interface donde se inician las peticiones**. Son enviadas desde fuera del router apuntando a Mikrotik, por ello van antes del routing (PRE-ROUNTING). Punto de partida de los paquetes, no importa que interface sea (física o virtual).

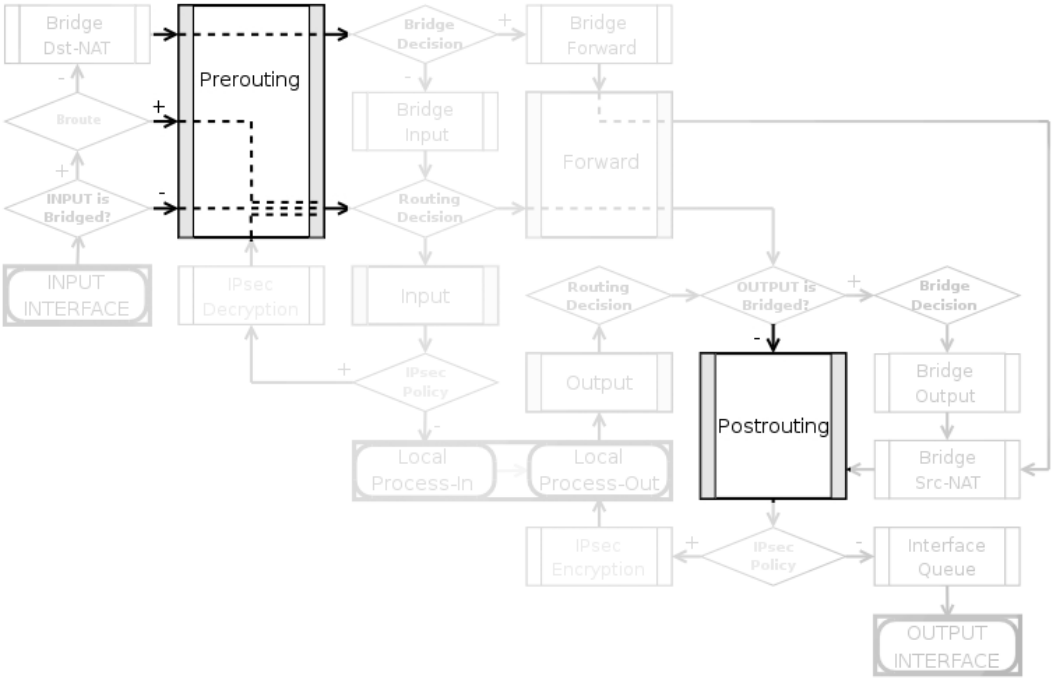
Ejemplo:

La red LAN puede considerarse INPUT INTERFACE cuando se hacen las peticiones de una pagina Web o acceso a una base de datos, todas estas peticiones tienen con dirección al router Mikrotik para que esta pueda resolverlas.

[OUTPUT INTERFACE]: Normalmente son llamadas interfaces productoras debido a que es en **esta interface donde responde a la solicitud de una interface input consumidora**. Este es el ultimo camino que tiene el paquete antes que sea enviado afuera de la red.

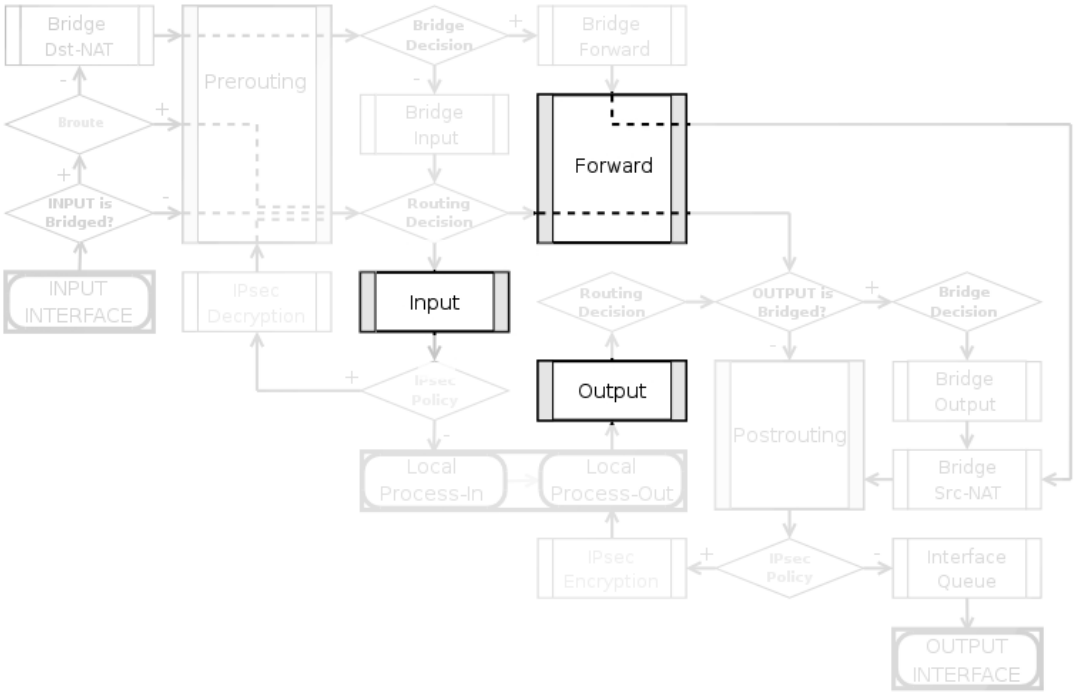
Ejemplo

Un ejemplo de ello es la interface WAN, ya que produce la informacion que la red LAN (consumidora) solicita. Y es en esta Interface donde los paquetes toman el ultimo camino antes que sean enviados a internet



[PRE-ROUTING]: Este es el sitio mas usado para los "**mangles rules**" (o llamados reglas de mangle) De este lugar se procesará la data que se dirigirá a través del router, pero lo mas importante es que procesará antes de que haya una decisión de ruteo. Así que tu puedes **aplicar las marcas** (mark connection) antes de que el router determine que ruteo va hacer. El **99% de tus reglas de mangle estarán aqui**.

[POST-ROUTING]: **En esta ubicación está el paquete que abandona el router**, buen uso para nuestro sistema de mangles aquí es cuando tu estas cambiando el tamaño de tu TCP o MMS, o haciendo otro cambio de de packets. Otra posibilidad es si tú estas cambiando el TOS bit de un paquete.



CADENAS - CHAIN -

[INPUT]: Este lugar tiene la misma característica que tiene la cadena INPUT de las reglas de FIREWALL, todo trafico de **datos que va como destino al ROUTER**.

[OUTPUT]: Este lugar tiene la misma característica que tiene la cadena OUTPUT de las reglas de FIREWALL. La cadena OUTPUT es para cuando haya alguna data que haya sido **generada desde nuestro ROUTER**.

[FORWARD]: Este lugar tiene la misma característica que tiene la cadena FORWARD de las reglas de FIREWALL. La cadena FORWARD es usada para procesar **paquetes y datos que viajan a través del ROUTER**, es decir que NO estan dirigidos hacía el Mikrotik (cadena INPUT) NI TIENEN origen en el Mikrotik (cadena OUTPUT), estos datos pueden estar dirigidos a un servidor de correos, servidor DNS, etc. Es decir tienen dirección distinta a la del **ROUTER** pero que NECESARIAMENTE requieren pasar por él.

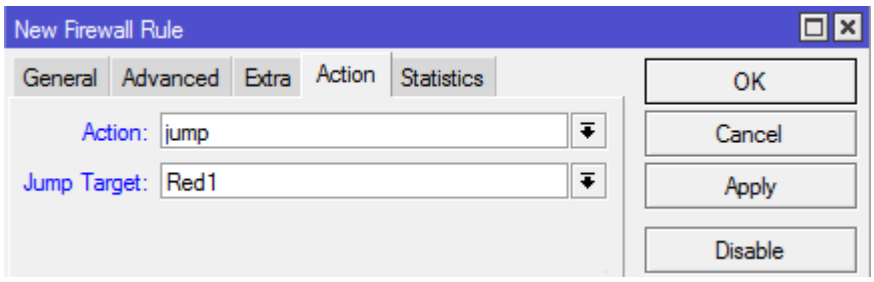
3 por defecto: in, out, fordward. Se pueden crear más…

[JUMP CHAIN] - Creando Nuevas Cadenas

Se permite poder crear tus propias cadenas, esto se consigue con la cadena JUMP. Puedes construirlas y darles el nombre que quieras.

Las cadenas creadas por el usuario sirven para ordenar el firewall. Por ejemplo, creo una cadena que se llame virus donde cargo los virus conocidos. Necesito luego hacer un jump desde la cadena input.

```
[admin@MikroTik] > ip firewall filter add dst-port=135-139,445 protocol=tcp
action=drop chain=virus comment="VIRUS DE WINDOWS"
[admin@MikroTik] > ip firewall filter add chain=input action=jump jump-
target=virus
```



Con estas dos reglas creamos dos nuevas cadenas llamadas Red1 y Red2

```
/ip firewall filter
add chain=forward dst-address=192.168.0.0/24 action=jump jump-target=Red1
add chain=forward dst-address=10.10.10.0/24 action=jump jump-target=Red2
```

Así se aíslan las redes, cualquier conexión que este fuera de la red a la cual pertenece NO podrá tener acceso.

```
/ip firewall filter
add chain=Red1 action=drop
add chain=Red2 action=drop
```

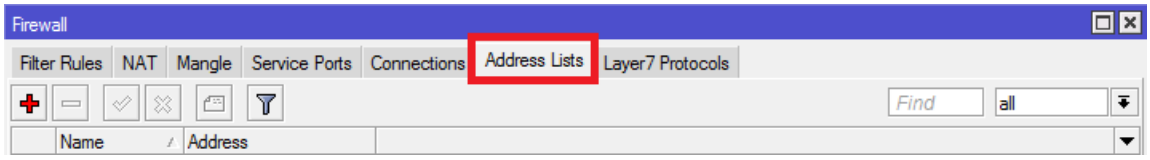
La primera cadena nos dice que SOLO la red 10.10.10.0/24 será bloqueada si es que quiere ingresar a la RED1 (esta es la red 192.168.0.0/24) de igual manera la segunda nos dice que SOLO la red 192.168.0.0/24 será bloqueada para ingresar a la RED2 (que es la red 10.10.10.0/24).
Como pueden observar podemos hacer múltiples opciones, que tal si necesitamos que ENTRE LAS REDES SE PUEDA PINEAR solo incluiremos esta regla por encima de todas y después drop para bloquear otro tipo de acceso.

```
/ip firewall filter
add chain=Red1 src-address=10.10.10.0/24 action=drop
add chain=Red2 src-address=192.168.0.0/24 action=drop
```

Si necesitamos que ENTRE LAS REDES SE PUEDA PINEAR solo incluiremos esta regla por encima de todas y después drop para bloquear otro tipo de acceso.

```
/ip firewall filter
add chain=Red1 src-address=10.10.10.0/24 protocol=icmp action=accept
add chain=Red2 src-address=192.168.0.0/24 protocol=icmp action=accept
add chain=Red1 src-address=10.10.10.0/24 action=drop
add chain=Red2 src-address=192.168.0.0/24 action=drop
```

[Firewall - Filter Rules] **ADDRESS LIST** ***- Creando grupos de IP's -***



Address List es una herramienta poderosa que caracteriza a Mikrotik, ésta nos da la habilidad de proveer una lista de direcciones, ya sea una sola o de un grupo de IP's (el cual puede ser un subred también).
Esto nos ayuda a poder aplicar reglas a un cualquier conjunto de IP's que querramos, inclusive a las IP's externas que no pertenecen a nuestra red, ya sea para bloquearlas, marcarlas, agregarlas a una cadena, etc.

Ejem1: ¿cuantos dispositivos ingresan a la red?

```
/ip firewall filter
add chain=forward src-address=192.168.88.0/24 \
action=add-src-to-address-list address-list="IP's de 192.168.88.0/24"
```

Agregamos una cadena "forward" a la red 192.168.88.0/24, esto representa toda la red, y la acción que vamos a tomar es la de "add-src-to-address-list" traducido al español es agregar al address-list llamado " IP's de 192.168.88.0/24"

Ejem2.1 ¿Que computadoras en la red 192.168.1.0/24 están usando programas P2P?

```
/ip firewall filter
add chain=forward src-address=192.168.1.0/24 p2p=all-p2p \
action=add-src-to-address-list address-list="USAN P2P"
```

Ejem2.2 Bloquear **todo** el tráfico de las ip que han usado P2P:

```
/ip firewall filter add action=drop chain=forward src-address-list="USAN P2P"
```

Las listas contienen direcciones IP para las que podemos tomar determinadas acciones. De esta manera, mantenemos una única lista de direcciones y la invocamos en el firewall:

Ejem3 Crear una lista especificando desde dónde permitimos conexiones SSH

```
[admin@MikroTik] > ip firewall address-list add list=ssh-permitido
address=192.168.1.2/32 comment="MAQUINA DEL ADMINISTRADOR"
[admin@MikroTik] > ip firewall filter add src-address-list=!ssh-permitido
protocol=tcp dst-port=22 action=drop chain=input comment="ACEPTO SSH DESDE LAS
MAQUINAS EN LA LISTA ssh-permitido"
```

Ejem4 **Agregar IPs a una address-list de forma dinámica: si quiero por ejemplo guardar todas las IPs que intentaron acceder por WinBox a mi equipo:**

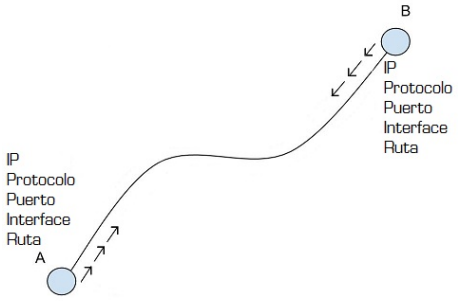
```
[admin@MikroTik] > ip firewall filter add chain=input action=add-src-to-address-
list protocol=tcp address-list=acceso-winbox dst-port=8291
[admin@MikroTik] > ip firewall address-list print
Flags: X - disabled, D - dynamic
```

[Firewall - Mangle] Marcar Conexion vs Marcar Paquetes [Mark Connection vs Packet]

Esta característica se usa para marcar datos e identificarlos para después usarlos en otras reglas (se puede usar para dar prioridades junto con el queues).

Mark Connection

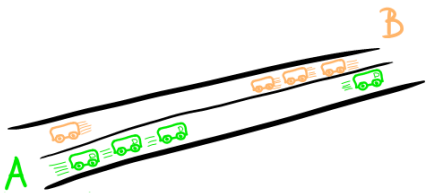
Se marca todo el trazado punto a punto, la dirección Ip, el protocolo (TCP/UDP ICMP etc.), el puerto por el cual se esta entablando la comunicación, la Interface por la cual esta saliendo, etc. Esta identificación es en ambos sentidos.



Esta conexión que se establece se "marca" es decir se le etiqueta con un nombre.
Podemos imaginar que la carretera que une LIMA - TACNA es una conexión, y podemos marcarlo con el nombre de panamericana norte. Obviamente lo que se marca con la etiqueta "panamericana norte" seria toda la infraestructura, el asfalto, las señales de transito, etc. OJO pero lo que NO SE MARCA son los autos.

Mark Packets

El marcado de paquetes se refiere al tráfico de paquetes, y estos se identifican y se marcan.
Utilizando la misma analogía, es decir de la carretera de LIMA a ICA, el marcado de paquetes se refiere a los autos que transitan por la panamericana. Cada una con una matricula distinta y un coche distinto.

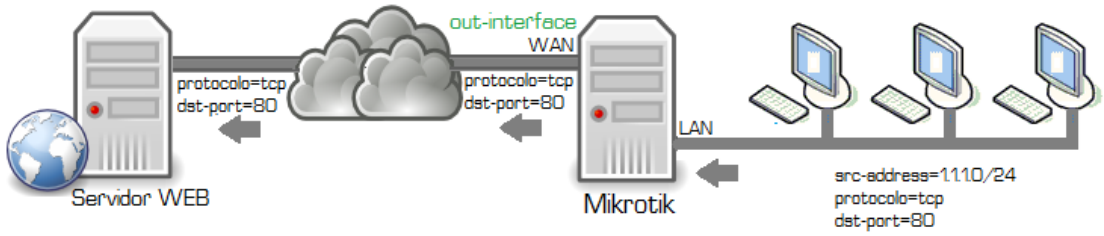


¿Cual es la diferencia entre el marcado de paquetes y marcado de conexiones?
Si marcamos los paquetes el Router trabajara mucho más ya que comprobara x veces por cada paquete, esto provocara que el procesador sea saturado, en cambio si usamos marcado de conexiones y marcado de paquetes solo usará dos comprobaciones (ya establecidas) por lo que se hará un uso muy eficiente del procesador.
NO SE DEBE USAR SOLO MARCADO DE PAQUETES, SINO DEBEMOS AYUDARNOS USANDO MARCADO DE CONEXIONES para poder hacer un uso eficiente de nuestro router.

Ejemplo: Marcar el trafico HTTP en LAN **1.1.1.1/24** por MARCADO de PAQUETES.

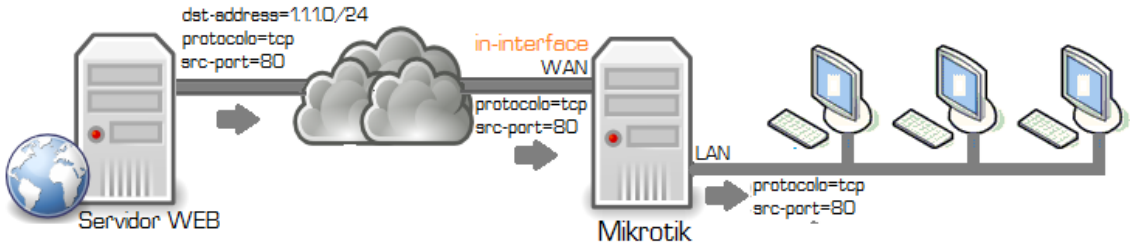
Como los paquetes viajan en dos sentidos, tendremos dos partes que trabajaran en el mangle:
1. Un sentido es cuando la red interna baja información del servidor Web.
2. Otro sentido es cuando la red interna sube información del servidor Web.

Entonces se requiere marcar el trafico HTTP vía MARCADO DE PAQUETES, entonces en donde vamos a marcar será en la tarjeta WAN. En la figura se puede apreciar que la tarjeta WAN esta actuando como OUT-INTERFACE, como ya se ha dicho anteriormente, OUT-INTERFACE hace referencia a las tarjetas productoras, para el caso la tarjeta WAN "PRODUCE" información para la red interna LAN (esto sucede cuando accedemos a una pagina alojada en un servidor Web que esta en la nube, la tarjeta WAN recopila información de esos servidores Web para después producir las paginas Web que la red LAN requiere).




```
/ip firewall mangle
add chain=forward src-address=1.1.1.0/24 protocol=tcp dst-port=80 out-
interface=WAN\
action=mark-packet new-packet-mark=test
```

En la figura se puede apreciar que la tarjeta WAN esta actuando como IN-INTERFACE, hace referencia a las tarjetas consumidoras, para el caso la tarjeta WAN "CONSUMEN" (visto desde fuera de la red interna LAN) información para la red LAN (esto sucede cuando accedemos a una pagina alojada en un servidor Web que esta en la nube, la tarjeta WAN hace peticiones de información de esos servidores Web).



Lo que haríamos con estas reglas es marcar todos los paquetes que son trafico HTTP entre la red 1.1.1.1/24 y un servidor Web con la marca llamada TEST.
Sin embargo para cada paquete tú deberías tener que hacer un montón de trabajo.
El proceso es el siguiente para la primera parte (El primer grafico):

```
add chain=forward src-address=1.1.1.0/24 protocol=tcp dst-port=80 out-
interface=WAN\
action=mark-packet new-packet-mark=test

1)¿El paquete esta saliendo (source address) de la red 1.1.1.1/24? Respuesta
"SI"
2)¿Es el paquete, un paquete TCP? Respuesta "SI"
3)¿El paquete tiene como puerto de destino el 80? Respuesta "SI"
4)¿El paquete esta saliendo por la interface productora (out-interface) WAN?
Respuesta "SI"
```

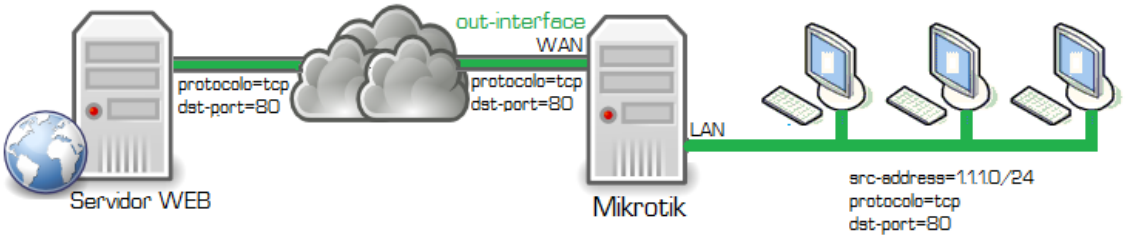
Para la segunda parte (El segundo grafico) el proceso seria el siguiente:

```
add chain=forward dst-address=1.1.1.0/24 protocol=tcp src-port=80 in-
interface=WAN\
action=mark-packet new-packet-mark=test
1)¿El paquete tiene como destino (destine address) la red 1.1.1.1/24? Respuesta
"SI"
2)¿Es el paquete, un paquete TCP? Respuesta "SI"
3)¿El paquete tiene como puerto de salida el 80? Respuesta "SI"
4)¿El paquete esta dirigiéndose a la interface consumidora (in-interface) WAN?
Respuesta "SI"
```

Marcando solo paquetes cada paquete HTTP requiere 8 comprobaciones.

Marcar vía MARCADO de CONEXIONES y PAQUETES

Lo que se busca cuando se marca una conexión es solo marcar una conexión (a diferencia del marcado de paquetes que se tiene que especificar la ida y vuelta), no se necesita marcar las dos direcciones, ya que si establecemos un camino se marcara todo lo que se transite por esta vía, ya sea de ida o vuelta.




```
/ip firewall mangle
add chain=forward src-address=1.1.1.0/24 protocol=tcp dst-port=80 out-
interface=WAN\
connection-state=new action=mark-connection new-connection-mark=test
passthrough=yes

/ip firewall mangle
add chain=forward connection-mark=test action=mark-packet new-packet-mark=test
```

Ahora, SOLO POR UNICA VEZ (es decir para el primer paquete) se comprobara 5 veces, despues que ello ocurra se hara en solo dos conexiones:
Código:

```
1)¿El paquete de la conexion esta saliendo (source address) de la red
1.1.1.1/24? Respuesta "SI"
2)¿El paquete de la conexion, un paquete TCP? Respuesta "SI"
3)¿El paquete de la conexion tiene como puerto de destino el 80? Respuesta "SI"
4)¿El paquete de la conexion esta saliendo por la interface productora (out-
interface) WAN? Respuesta "SI"
5)Marcar todos los pquetes de la conexión establecida.
```

Cada paquete siguiente, es decir cada paquete que sigue al primer paquete solo será comprobado 2 veces:
Código:

```
1)¿Es esta una nueva conexion? Respuesta: "NO"
2)¿Tiene la marca de conexión? Respuesta: "SI"

*hay excepciones donde si debe ir ser marcado por paquetes como este ejemplo
Para este ejemplo no podemos marcar la conexion por que solo priorizan las
banderas TCP o LLamese Flags - syn - ack entonces concluimos tambien que existen
casos especificos donde no se puede usar la marca.
*el QoS para balanceo se hace para cada WAN por que la LAN es una misma asi es
que no podria diferenciar.
```

Explicando las reglas.

Connection tracking: permite visualizar las conexiones en las que interviene nuestro equipo.

```
[admin@MikroTik] > ip firewall connection print
```

Mover una regla: por ejemplo, acabo de agregar una regla y por defecto lo hace al final. Necesito

que dicha regla que esta en el lugar 13 pase al 3:

```
[admin@MikroTik] > ip firewall filter move 13 3
```

Las **reglas en el FIREWALL FILTER se ejecutan por orden, se ejecutan las que se sitúan arriba y después la de abajo.**
Esta primera regla nos indica que toda la red va a poder mandar pin entre ellos 192.168.1.X

```
/ip firewall filter
add chain=input src-address=192.168.1.0/24 protocol=icmp action=accept
```

Esta segunda nos dice que TODOS los demás pin de otras redes son bloqueados.

```
add chain=input protocol=icmp action=drop
```

PERMITIR el FTP del Mikrotik en toda la RED PRIVADA, es decir 192.168.1.0/24 y a la vez deberán DENEGAR el FTP del Mikrotik a toda RED PÚBLICA, es decir que denegar a todos los que estén queriendo entrar desde el Internet al FTP de Mikrotik.

```
/ip firewall filter
add chain=input src-address=192.168.1.0/24 protocol=tcp dst-port=21
action=accept
add chain=input protocol=tcp dst-port=21 action=drop
```

Cuando uses Input chain usas el dst port debido a que INPUT es cuando hay alguna petición desde afuera con dirección de destino al router. Por ello usamos destination-port, que en abreviaturas es dst-port

Notas sobre rendimiento

- Tener en cuenta que siempre conviene que si hay dos reglas que pueden resumirse en una se haga, pues es una regla menos para procesar.
- Si no se utiliza el equipo como router conviene deshabilitar el connection tracking, pues así nos estaríamos ahorrando tiempo de procesamiento y memoria RAM.
- Siempre conviene empezar con las reglas de estado, para ahorrar procesamiento y acelerar las conexiones ya establecidas y las relativas.

```
[admin@MikroTik] > ip firewall filter add connection-state=established
action=accept chain=input
[admin@MikroTik] > ip firewall filter add connection-state=related action=accept
chain=input
[admin@MikroTik] > ip firewall filter add connection-state=invalid  action=drop
chain=input
[admin@MikroTik] > ip firewall filter add protocol=tcp src-port=8291 in-
interface=!wlan1 action=accept chain=input comment="DENIEGA WINBOX DESDE LA
WIRELESS"
```

 Firewall **Protegiendo al Router**

Ejem1 Digamos que nuestra red privada es 192.168.0.0/24 y la WAN es laEther1. Vamos a configurar el firewall para permitir conexiones hacia el mismo router sólo de nuestra red local y dropear el resto. También vamos a permitir el protocolo ICMP en cualquier interfaz para que cualquier persona pueda hacer ping al router desde Internet.

/ip firewall filteradd chain=input connection-state=invalid action=drop comment="DropInvalid connections"add chain=input connection-state=established action=accept comment="Allow Established connections"add chain=input protocol=icmp action=accept comment="Allow ICMP"add chain=input src-address=192.168.0.0/24 action=accept in-interface=!ether1add

chain=input action=drop comment="Drop everything else"

Ejem2 Bloquearemos todo el tráfico Generado desde Internet hacia la LAN.

```
/ip firewall filter
#esto permite trafico de winbox, se pone encima de las prohibiciones.
add action=accept chain=input comment="" disabled=no dst-port=8291 in-
interface=wan protocol=tcp
#dejamos pasar todas las conexiones ya establecidas que se hayan generado en el
mikrotik hacia la publica o wan en este caso practicamente seria el DNS o el
Webproxy si es que lo activan.
add action=accept chain=input comment="" connection-
state=established disabled=no in-interface=wan
#dejamos pasar conexiones relacionadas, existen aplicaciones como puede ser el
caso FTP donde la autenticación la hacen en un puerto y el trafico en otro
básicamente para ello agregamos esta regla.
add action=accept chain=input comment="" connection-
state=related disabled=no in-interface=wan
#cerramos todo para que todo lo que llegue al mikrotik desde WAN - LAN lo
descarte regla en último lugar.
add action=drop chain=input comment="" disabled=no in-interface=wan
```

 Firewall – Protegiendo a los Clientes

Deberíamos considerar el tráfico que pasa a través del router hacia los clientes y bloquear lo no deseado. Para el tráfico icmp, tcp, udp crearemos reglas para dropear paquetes no deseados:

/ip firewall filteradd chain=forward protocol=tcp connection-state=invalid action=drop comment="dropinvalid connections"add chain=forward connection-state=established action=accept comment="allow alreadyestablished connections"add chain=forward connection-state=related action=accept comment="allow relatedconnections"Creamos reglas TCP y denegamos algunos puertos ellas:add chain=forward protocol=tcp dst-port=69 action=drop comment="deny TFTP"add chain=forward protocol=tcp dst-port=111 action=drop comment="deny RPC portmapper"add chain=forward protocol=tcp dst-port=135 action=drop comment="deny RPC portmapper"add chain=forward protocol=tcp dst-port=137-139 action=drop comment="deny NBT"add chain=forward protocol=tcp dst-port=445 action=drop comment="deny cifs"add chain=forward protocol=tcp dst-port=2049 action=drop comment="deny NFS"add chain=forward protocol=tcp dst-port=12345-12346 action=drop comment="deny NetBus"add chain=forward protocol=tcp dst-port=20034 action=drop comment="deny NetBus"add chain=forward protocol=tcp dst-port=3133 action=drop comment="deny BackOriffice"add chain=forward protocol=tcp dst-port=67-68 action=drop comment="deny DHCP"

 Firewall – NAT

Source NAT –

Se utiliza para paquetes originados en la red nateada. El router reemplaza la ip privada origen del paquete IP con una ip pública mientras pasa por el router. La operación contraria sucede con los paquetes que viajan en dirección inversa.

Destination NAT –

Se utiliza para paquetes que van destinados a la red nateada. Comunmente se utiliza para que los host de la red interna sean accesibles desde internet. El router reemplazala IP destino del paquete IP mientras pasa por el mismo hacia lared privada.

 Firewall – Source NAT

Como esconder la red privada tras una IP pública

/ip firewall nat add chain=srcnat action=masquerade out-interface=Public Destination NATComo acceder desde afuera a un Host de la red Privada/ip firewall nat add chain=dstnat dst-address=10.5.8.200 action=dst-nat to-addresses=192.168.0.109

Agrego esta regla si quiero que el Host se comunique con otras redes mantiendo como IP origen la IP pública

/ip firewall nat add chain=srcnat src-address=192.168.0.109 action=src-nat to-addresses=10.5.8.200

Port knocking **Port knocking** consiste en utilizar un preámbulo específico para luego lograr el acceso a donde necesitamos. Por ejemplo, si queremos acceder por

SSH al equipo podemos definir que desde la IP que deseamos ingresar hagamos un intento de acceso al puerto 33, luego al 55 y finalmente al 77. Al cumplir lo anterior entonces abriremos el puerto 22 para esa IP por un tiempo limitado.

La forma de implementar el port knocking del ejemplo con Mikrotik es la siguiente:

```
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=77 action=add-src-to-address-list address-list=ssh-permit-
temp address-list-timeout=1h src-address-list=step2
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=55 action=add-src-to-address-list address-list=step2
address-list-timeout=1m src-address-list=step1
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=33 action=add-src-to-address-list address-list=step1
address-list-timeout=1m
[admin@MikroTik] > ip firewall filter add chain=input protocol=tcp dst-port=22
src-address-list=!ssh-permit-temp action=drop
```

Fuerza bruta

Para impedir que, por ejemplo, nos descubran la password del SSH utilizando fuerza bruta podemos implementar un mecanismo que habilite sólo tres intentos de acceso y luego bloquee la IP por 10 días:

```
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=22 action=add-src-to-address-list address-list=ssh-
blacklist address-list-timeout=10d src-address-list=ssh3
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=22 action=add-src-to-address-list address-list=ssh3
address-list-timeout=1m src-address-list=ssh2
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=22 action=add-src-to-address-list address-list=ssh2
address-list-timeout=1m src-address-list=ssh1
[admin@MikroTik] > ip firewall filter add chain=input connection-state=new
protocol=tcp dst-port=22 action=add-src-to-address-list address-list=ssh1
address-list-timeout=1m
[admin@MikroTik] > ip firewall filter add chain=input protocol=tcp dst-port=22
action=drop address-list=ssh-blacklist
```

Otro ejemplo de protección contra fuerza bruta en el puerto 22

1. Negar cualquiera que esté en la ssh_blacklist una nueva sesión sobre cualquier protocolo.
2. Permitir que cualquier persona que estaba en la "lista gris ssh_Dark" para conectar una nueva sesión en el puerto 22 y añadir la dirección a la "ssh_Blacklist" con un tiempo de 1 hora
3. Permitir que cualquier persona que estaba en la "lista ssh_grey" para conectar una nueva sesión en el puerto 22 y añadir la dirección a la "lista gris ssh_dark" con un tiempo de 1 minuto
4. Permitir que cualquier persona que estaba en el "light_grey_list" para conectar una nueva sesión en el puerto 22 y añadir la dirección a la "lista ssh_grey" con un tiempo de 1 minuto
5. Permitir que cualquier persona que crea una primera sesión en el puerto 22 y añadir la dirección a la "ssh_lightgreylist" con un tiempo de 1 minuto

```
/ip firewall filter

add chain=input src-address-list=sshblacklist action=drop \
```

```
comment="drop all traffic brute force attack sources" disabled=no

add chain=input protocol=tcp dst-port=22 connection-state=new \
src-address-list=sshdarkgreylist action=add-src-to-address-list \
address-list=sshblacklist address-list-timeout=1h \
comment="add new failed sshdarkgreylist to sshblacklist" \
disabled=no

add chain=input protocol=tcp dst-port=22 connection-state=new \
src-address-list=sshgreylist action=add-src-to-address-list \
address-list=sshdarkgreylist address-list-timeout=1m \
comment="add new failed sshgreylist to sshdarkgreylist" \
disabled=no

add chain=input protocol=tcp dst-port=22 connection-state=new \
src-address-list=sshlightgreylist action=add-src-to-address-list \
address-list=sshgreylist address-list-timeout=1m \
comment="add new failed sshlightgreylist to sshgreylist" \
disabled=no

add chain=input protocol=tcp dst-port=22 connection-state=new \
action=add-src-to-address-list \
address-list=sshlightgreylist address-list-timeout=1m \
comment="new connections to sshlightgreylist" \
disabled=no
```

DoS

Los ataques de DoS se llevan a cabo consumiendo y agotando los recursos del equipo/red atacado. Existen algunas formas de mitigarlos.

Una estrategia es utilizar tarpit. Esto baja la ventana TCP a 0, impidiendo que haya transferencia de datos pero dejando que se generen las conexiones. El siguiente ejemplo muestra cómo permitir 19 conexiones simultáneas por IP con destino al servidor web y aplicar tarpit a partir de la conexión 20.

```
[admin@MikroTik] > ip firewall filter add chain=forward dst-address=163.10.0.84
protocol=tcp dst-port=80 action=tarpit connection-limit=20,32
```

Otra estrategia que puede realizarse es permitir un máximo de conexiones nuevas simultáneas. Por ejemplo, la siguiente regla permite 5 conexiones nuevas al mismo tiempo y las restantes las dropea:

```
[admin@MikroTik] > ip firewall filter add chain=forward connection-state=new
dst-address=163.10.0.84 protocol=tcp dst-port=80 action=drop connection-
limit=5,32
```

Aceptar conexiones VPN:

```
[admin@MikroTik] > ip firewall filter add protocol=gre action=accept chain=input
[admin@MikroTik] > ip firewall filter add protocol=tcp dst-port=1723
action=accept chain=input comment="ACEPTO CONEXIONES VPN"
```

Denegar ping y loguear los intentos de ping:

```
[admin@MikroTik] > ip firewall filter add protocol=icmp chain=input action=log
log-prefix="PING DENEGADO"
[admin@MikroTik] > ip firewall filter add protocol=icmp action=accept
chain=input comment="DENIEGO ICMP"
[admin@MikroTik] > log print
23:34:12 firewall,info PING DENEGADO input: in:ether1 out:(none), src-mac
00:21:70:fd:e3:25, proto ICMP (type 8, code 0), 192.168.4.254->192.168.4.1, len
64
```

Mikrotik Bridge : Configuracion de Mikrotik Bridge y Router

Bloqueo de Paginas Pornograficas o Violentas - OpenDNS Family Shield

DNS OpenDNS: 208.67.222.123 y **208.67.220.123**

Instalar Configurar Manual Mikrotik RouterBoard - Basico Avanzado

MUY Completo inkalinux

Recolopilatorio de reglas de virus

Firewall rules

Basic Firewall & Security klseet

Mikrotik Firewall configuration to make the Internet safe

Diapositivas para securizar mKtik

Otra de **Mikrotik Firewall Basic Settings**

Basic universal firewall script