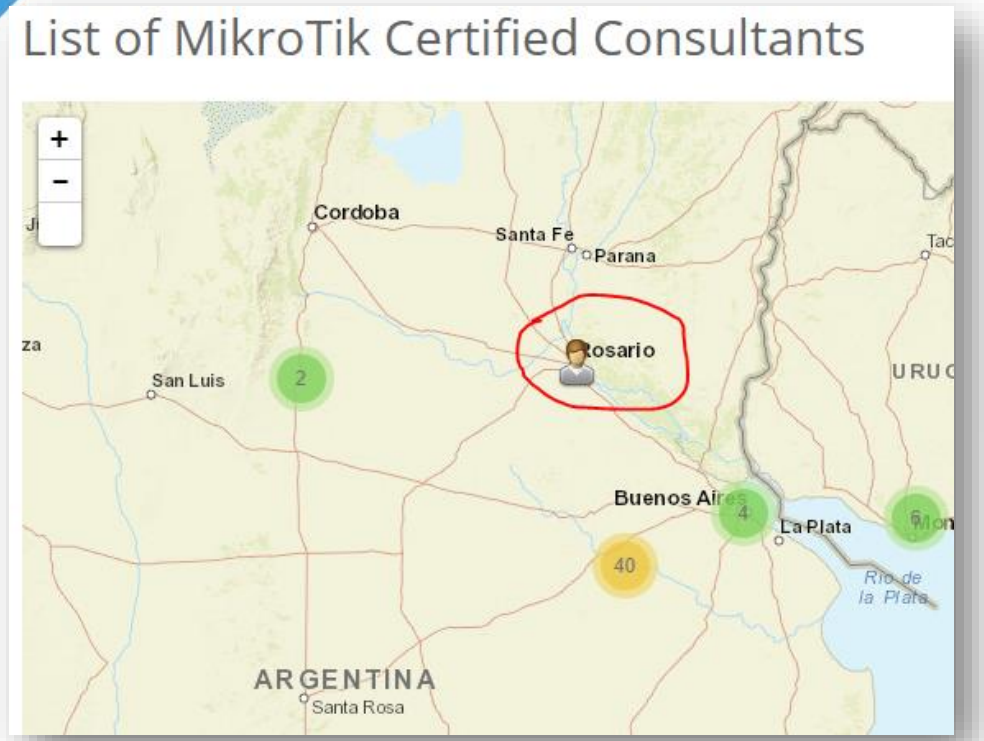
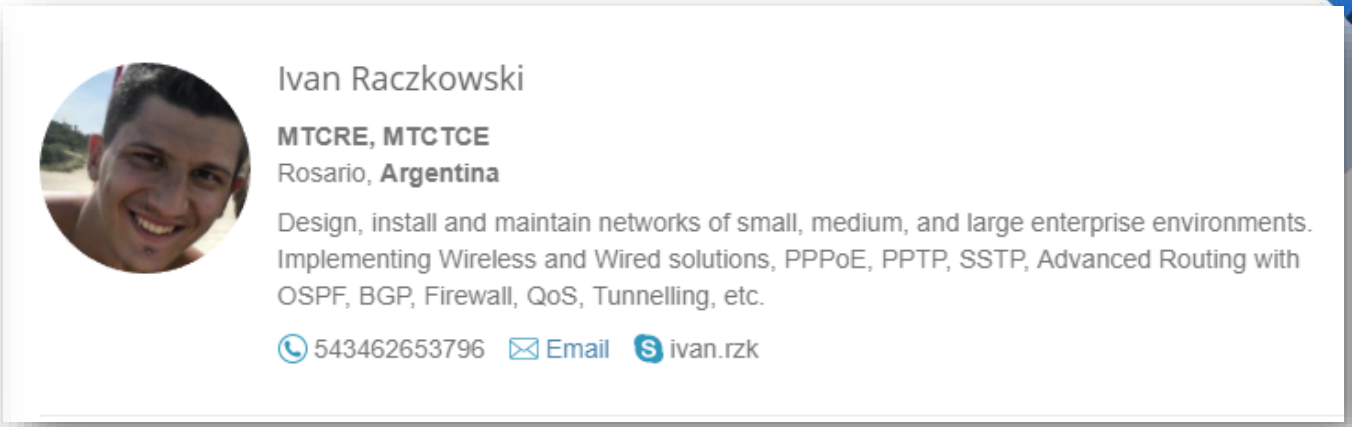
# NeuralSoft SRL

Caso Práctico de Filtrado Web por categorías usando el servicio de DNS de OpenDNS.

# Consultor Certificado




https://mikrotik.com/consultants/latinamerica/argentina

**Otras Certificaciones:**

- CCNA (Cisco)
- NSE (Fortinet)
- IPv6 Certified (HE)

Actualmente me desempeño como Consultor de Infraestructura (Nivel2) en NeuralSoft.

# NeuralSoft

- Empresa de Software de Argentina.
- Con Oficinas en Rosario, Buenos Aires y Córdoba.

**NeuralSoft** está avalada por la certificación del Sistema de Gestión de calidad bajo la norma **ISO 9001:2015**, en el "Desarrollo, Implementación y Post venta de Presea ERP, Rubiro, Deonics y servicios de "outsourcing" de aplicaciones en modalidad ASP desde las ciudades de Rosario, CA de Buenos Aires y Córdoba".

IRAM
GESTIÓN DE LA CALIDAD
RI-9000-1269

CERTIFIED IQNet MANAGEMENT SYSTEM

# Productos que ofrece la Empresa:

## **PRESEA** es el software ERP más completo del mercado

Más de 2700 funcionalidades para la gestión integrada y el gerenciamiento de todas las áreas de su empresa, incluyendo en un sistema único COMPRAS, VENTAS, PUNTOS DE VENTA, STOCK Y LOGISTICA, CONTABILIDAD, FINANZAS, PRODUCCION, RRHH, CRM, WORKFLOW y BUSINESS INTELLIGENCE.

## **RUBIRO** es el software de gestión que pone la tecnología de la nube al alcance de su PYME

Una herramienta sencilla y ágil, que permite gerenciar su empresa con procesos simples, pudiendo acceder en cualquier momento y desde cualquier lugar a la información de las ventas, pedidos, cuentas corrientes, compras, bancos, impuestos, estadísticas y demás procesos de su compañía.

## **DEONICS** Gestión inteligente de transporte, logística, depósito y flotas.

**Deonics** es un software de gestión integral, hecho en Argentina. Ha sido diseñado para cubrir las necesidades especificas de empresas de transporte y logística, grandes almacenes y organizaciones que operen con flotas de vehiculos propios.


NEURALSOFT EN NÚMEROS
3 sedes
Más de 150 profesionales.
Más de 16 años en la Nube
Más de 1500 sucursales de clientes online.
Neuralsoft
Más de 9 terabytes de Ram en nuestros servers.
Más de 750 terabytes de almacenamiento en Fibra óptica.
3 décadas de trayectoria
Más de 400 clientes de los rubros más variados
Certificaciones ISO 9001:2015. Modelo CMMI.

# Esquema de Filtrado Web por DNS usando OpenDNS

**Características de OpenDNS:**

Las características más destacadas del filtro de contenidos ofrecido por OpenDNS son:

- Es totalmente gratuito.
- No se necesita instalar ningún programa en las PC Clientes. Por tanto es imposible eludir el filtrado desactivando el correspondiente programa.
- Funciona con independencia del navegador web utilizado, incluso si el usuario instala un nuevo navegador.
- Permite activar/desactivar y configurar el filtro de una sola vez y aplicar un filtro para toda una Red.
- Posee listas de “always block” y “never block”, además de estadísticas de sitios solicitados.
- La lista de páginas bloqueadas es actualizada y revisada constantemente.
- El bloqueo se hace con gastos mínimos de recursos, ya que se filtra por consulta DNS y no hay que procesar por Paquetes.

## ¿Cómo funciona OpenDNS?

Cada vez que introducimos la dirección de una página web en la barra de direcciones del navegador, por ejemplo, http://www.google.com, el ordenador se dirige a un servidor de nombres DNS para saber la dirección IP del servidor que contiene la página: 172.217.162.4. Este es un paso previo para obtener esta página y descargarla al equipo para su visualización a través del navegador.

Si utilizamos los servidores DNS de OpenDNS entonces navegaremos usando el acceso proporcionado por nuestros proveedores de Internet pero aprovechándonos de las prestaciones de filtrado que nos ofrece de forma gratuita OpenDNS.

Los servidores DNS de OpenDNS son: **208.67.222.222 y 208.67.220.220**.

## Crear una cuenta en OpenDNS

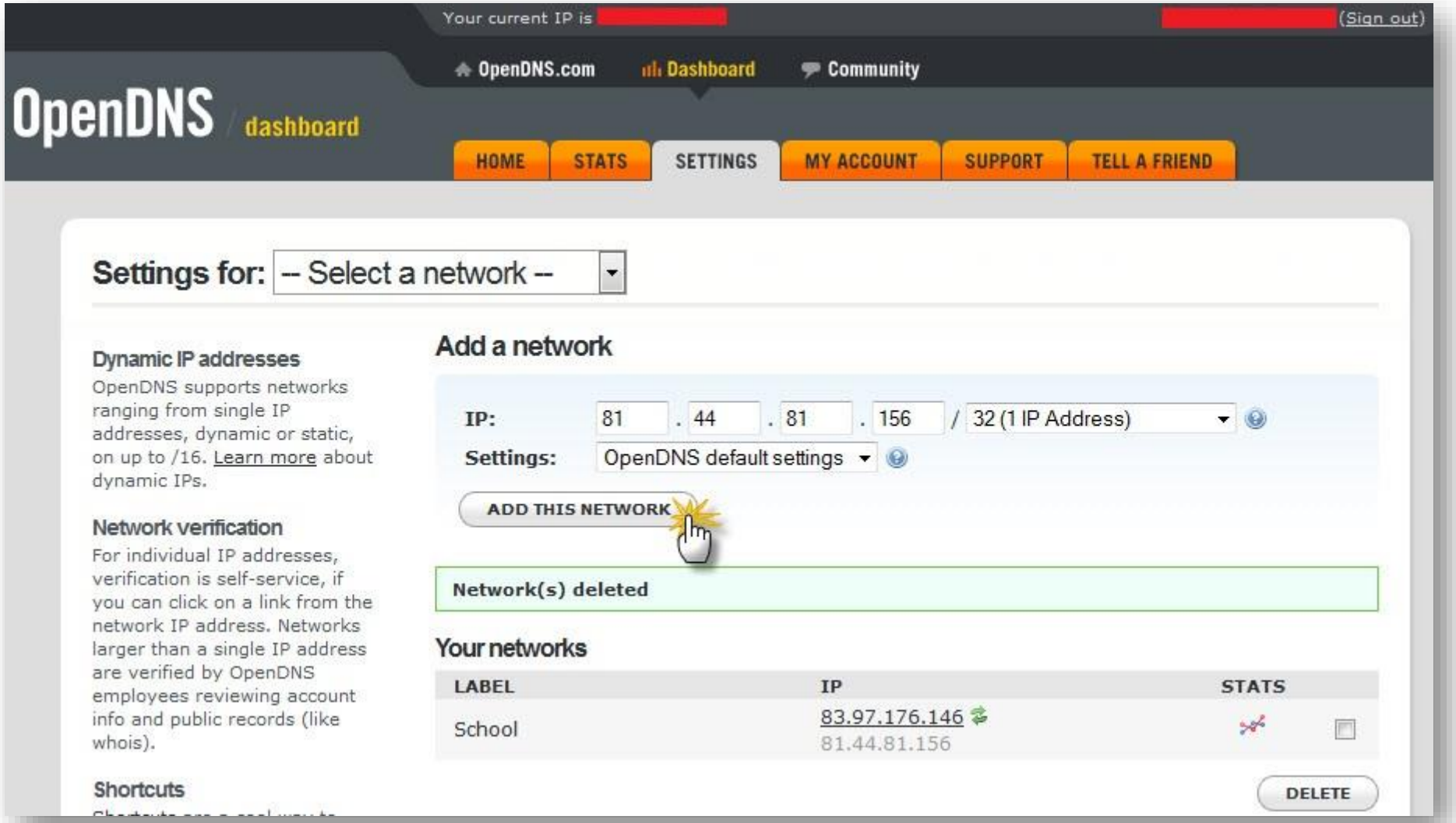
Aunque el servicio es gratuito es necesario registrarse para poder personalizar el filtrado de contenidos de acuerdo con nuestras preferencias.
Para ello sigue estos pasos:

1. Visita la web de OpenDNS: https://www.opendns.com/home-internet-security/
2. Se ofrecen las tres modalidades de servicio. En este caso haz click en el botón Sign up del paquete OpenDNS Home.
3. En el cuadro Create an account (Crear una cuenta) introduce los datos solicitados.
4. Click en el botón Continuar.

## Registrar la IP de acceso a internet de tu red local

Es necesario registrar la IP Pública de salida a Internet para que OpenDNS filtre todas las peticiones que procedan de la misma.

1. Accede a OpenDNS introduciendo tus credenciales.
2. Observa que al conectarte desde un equipo de la red local, en la parte superior de la página ya se muestra la IP externa del router: Your current IP is ... (Tu IP actual es ...)
3. Click en el enlace Dashboard (Panel de control) y luego en la pestaña Settings (Configuración).
4. En la sección Add a network clic en el botón Add this network (Añadir esta red).
5. Se solicita un nombre identificativo para esa red, p.e. Home o Office y a continuación click en el botón Aceptar.
6. En el listado Your networks (Tus redes) se mostrará una nueva entrada con la etiqueta Label definida.


Your current IP is [redacted]
[redacted] (Sign out)
OpenDNS.com
Dashboard
Community
OpenDNS dashboard
HOME
STATS
SETTINGS
MY ACCOUNT
SUPPORT
TELL A FRIEND
Settings for: -- Select a network --
Dynamic IP addresses
OpenDNS supports networks ranging from single IP addresses, dynamic or static, on up to /16. Learn more about dynamic IPs.
Network verification
For individual IP addresses, verification is self-service, if you can click on a link from the network IP address. Networks larger than a single IP address are verified by OpenDNS employees reviewing account info and public records (like whois).
Shortcuts
Add a network
IP: 81 . 44 . 81 . 156 / 32 (1 IP Address)
Settings: OpenDNS default settings
ADD THIS NETWORK
Network(s) deleted
Your networks
LABEL
IP
STATS
School
83.97.176.146
81.44.81.156
DELETE

Si nuestro proveedor de acceso a internet nos proporciona una IP dinámica (la IP externa del router cambiará cada cierto tiempo) entonces será necesario actualizar la IP.

Podemos descargar el programa OpenDNS Updater. Este programa se instala en una PC de la red local y se configura utilizando el usuario, contraseña y nombre de identificación que le dimos a nuestra red. Este programa se encargará de actualizar de forma automática el registro de OpenDNS cada vez que la IP externa del router se modifique.

Desde Mikrotik podemos realizarlo mediante Scripts, podemos correr un Script por Scheduler cada X tiempo que chequee nuestra IP Pública y la actualice al Sitio de OpenDNS.

# Script de Update:

Tomé un Script de la Página Oficial de OpenDNS, un desarrollo de la comunidad del foro de Mikrotik.

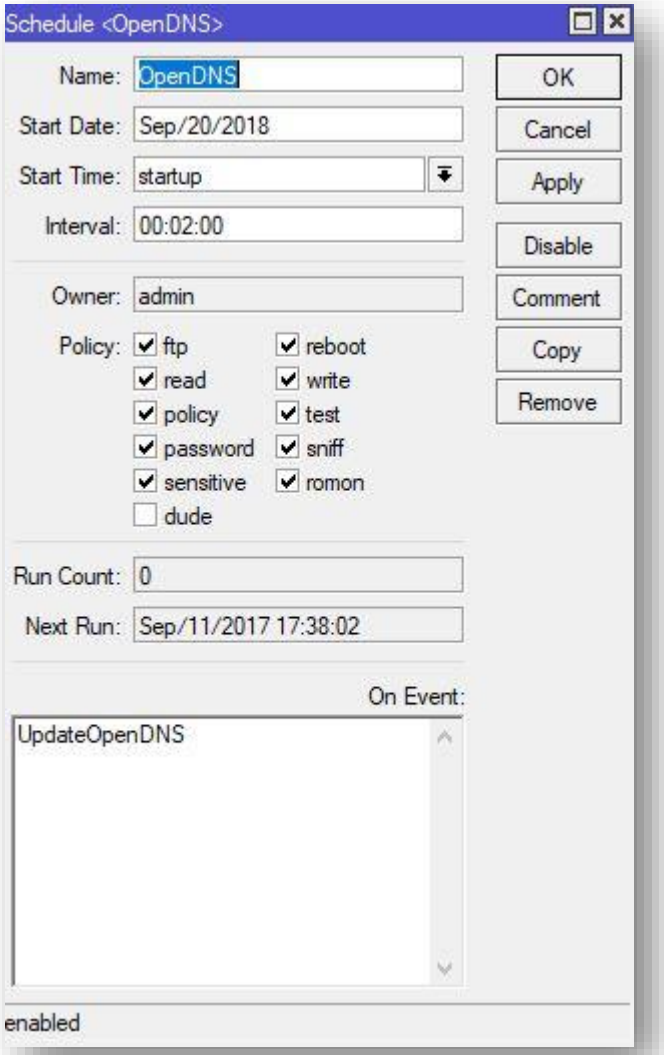
Corregí algunos parámetros que quedaron desactualizados e introduje una forma de obtener la IP Pública usando la funcionalidad Cloud de Mikrotik.

El Script es bastante sencillo.

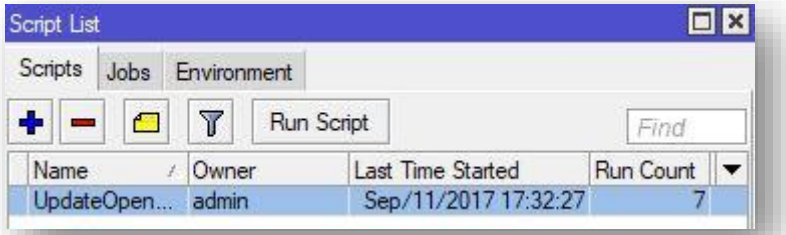
# Script de Update OpenDNS

```
#--------------- Change Values in this section to match your setup ------------------
# User account info of OpenDNS
# Update-only password (obtained from OpenDNS Support).
:local odnsuser User
:local odnspass Pass
# Set the hostname or label of network to be updated. This is the name of your OpenDNS network on the Dashboard.
# Hostnames with spaces are unsupported. Replace the value in the quotations below with your host name.
# Only one host is supported
# Note, you must have admin or edit (Read/Write/Grant in the OpenDNS Dashboard) to update IP addresses.
:local odnshost NombreDeSitioenOpenDNS
# Change to the name of interface that gets the changing IP address
:local inetinterface ether4
#-------------------------------------------------------------------------------

# No more changes needed, one optional change

:global previousIP;

:log info "Fetching current IP"

# Get the current public IP using DNS-O-Matic service.
#/tool fetch url="http://myip.dnsomatic.com/" mode=http dst-path=mypublicip.txt
# Read the current public IP (Using Cloud function in Mikrotik) into the currentIP variable.
/ip cloud force-update
:delay 5s

# Read the current public IP into the currentIP variable.
#:local currentIP [/file get mypublicip.txt contents]
# Read the current public IP (Using Cloud function in Mikrotik) into the currentIP variable.
:local currentIP [ip cloud get value-name=public-address]

:log info "Fetched current IP as $currentIP"

# ---------- Optional check to only run if the IP has changed (one line: :if)

# to disable, set line below to: ":if ($currentIP != 1) do={"

:if ($currentIP != $previousIP) do={
:log info "OpenDNS: Update needed"
:set previousIP $currentIP

# Some older editions of the MicroTik/WinBox OS require the following line instead (http) whereas newer versions require https.

# :local url "http://updates.opendns.com/nic/update\3Fhostname=$odnshost"

:local url "https://updates.opendns.com/nic/update\3Fhostname=$odnshost"
:log info "OpenDNS: Sending update for $odnshost"

/tool fetch url="$url" user=$odnsuser password=$odnspass mode=http dst-path=("/net_odns.txt")

:delay 2;

:local odnsReply [/file get net_odns.txt contents];

:log info "OpenDNS update complete."

:log info "OpenDNS reply was $odnsReply";

} else={

:log info "OpenDNS: Previous IP $previousIP and current IP equal, no update need"
}
```

# Script de Update OpenDNS




Cargamos el Script y creamos un Scheduler que corra cada X tiempo actualizando de ser necesarios los datos de la IP Pública en los Sitios de OpenDNS

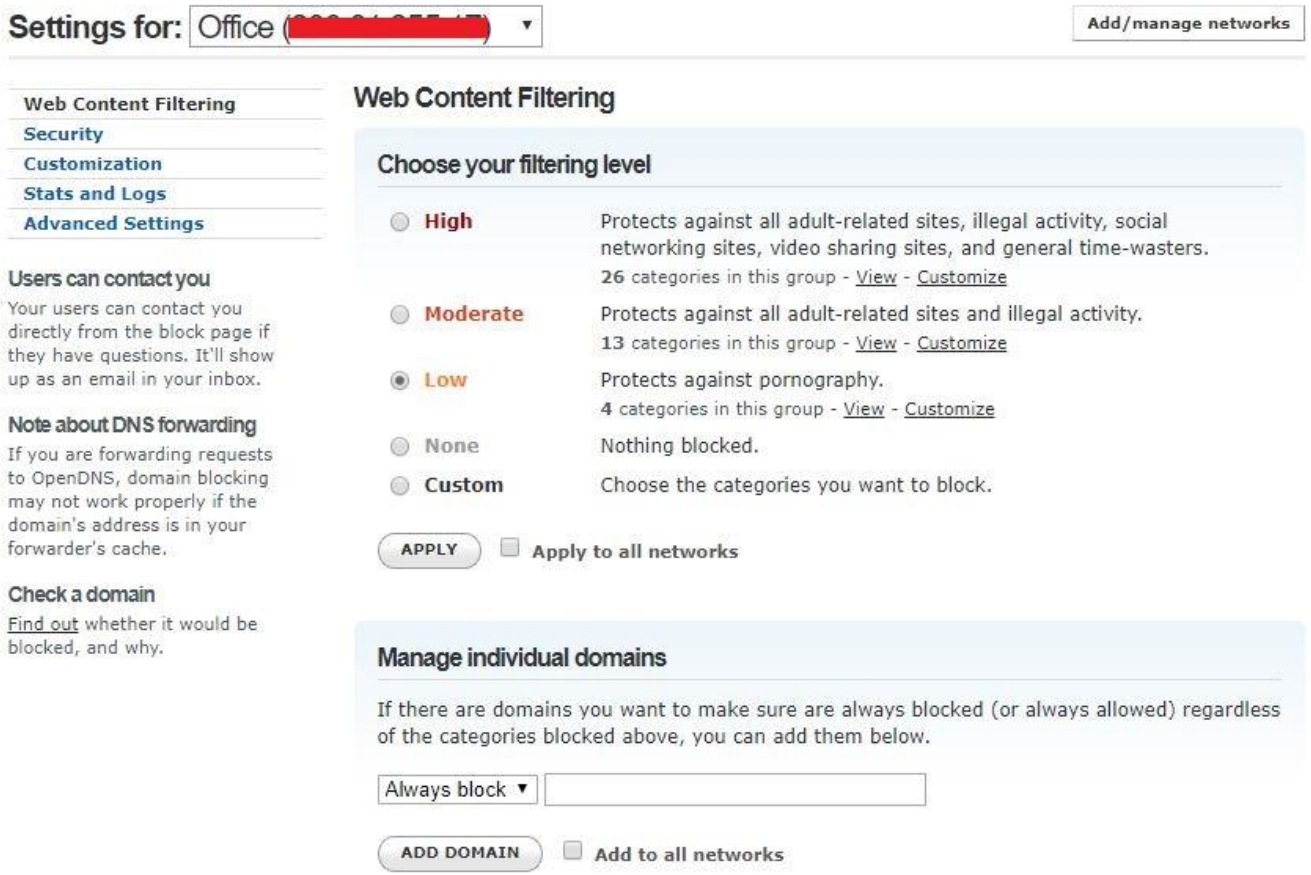



Podemos ver el resultado de una ejecución:

| | | | |
|---|---|---|---|
| Sep/11/2017 17:32:18 | memory | script, info | Fetching current IP |
| Sep/11/2017 17:32:23 | memory | script, info | Fetched current IP as [redacted] |
| Sep/11/2017 17:32:23 | memory | script, info | OpenDNS: Previous IP [redacted] and current IP equal, no update need |
| Sep/11/2017 17:32:27 | memory | script, info | Fetching current IP |
| Sep/11/2017 17:32:32 | memory | script, info | Fetched current IP as [redacted] |
| Sep/11/2017 17:32:32 | memory | script, info | OpenDNS: Update needed |
| Sep/11/2017 17:32:32 | memory | script, info | OpenDNS: Sending update for Office |
| Sep/11/2017 17:32:33 | memory | info | fetch: file "net_odns.txt" downloaded |
| Sep/11/2017 17:32:36 | memory | script, info | OpenDNS update complete. |
| Sep/11/2017 17:32:36 | memory | script, info | OpenDNS reply was good [redacted] |

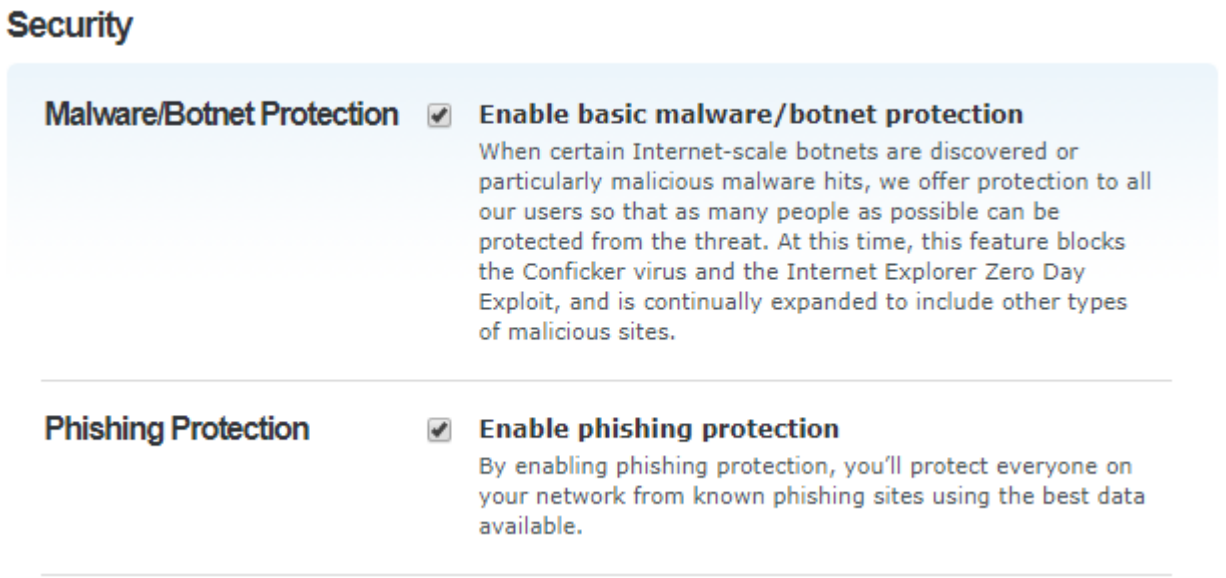
# Configuración de perfil en OpenDNS

**Podemos crear un Perfil de navegación y definir las categorías que queremos bloquear, así como también “always block”, “never block” y definir URLs.**



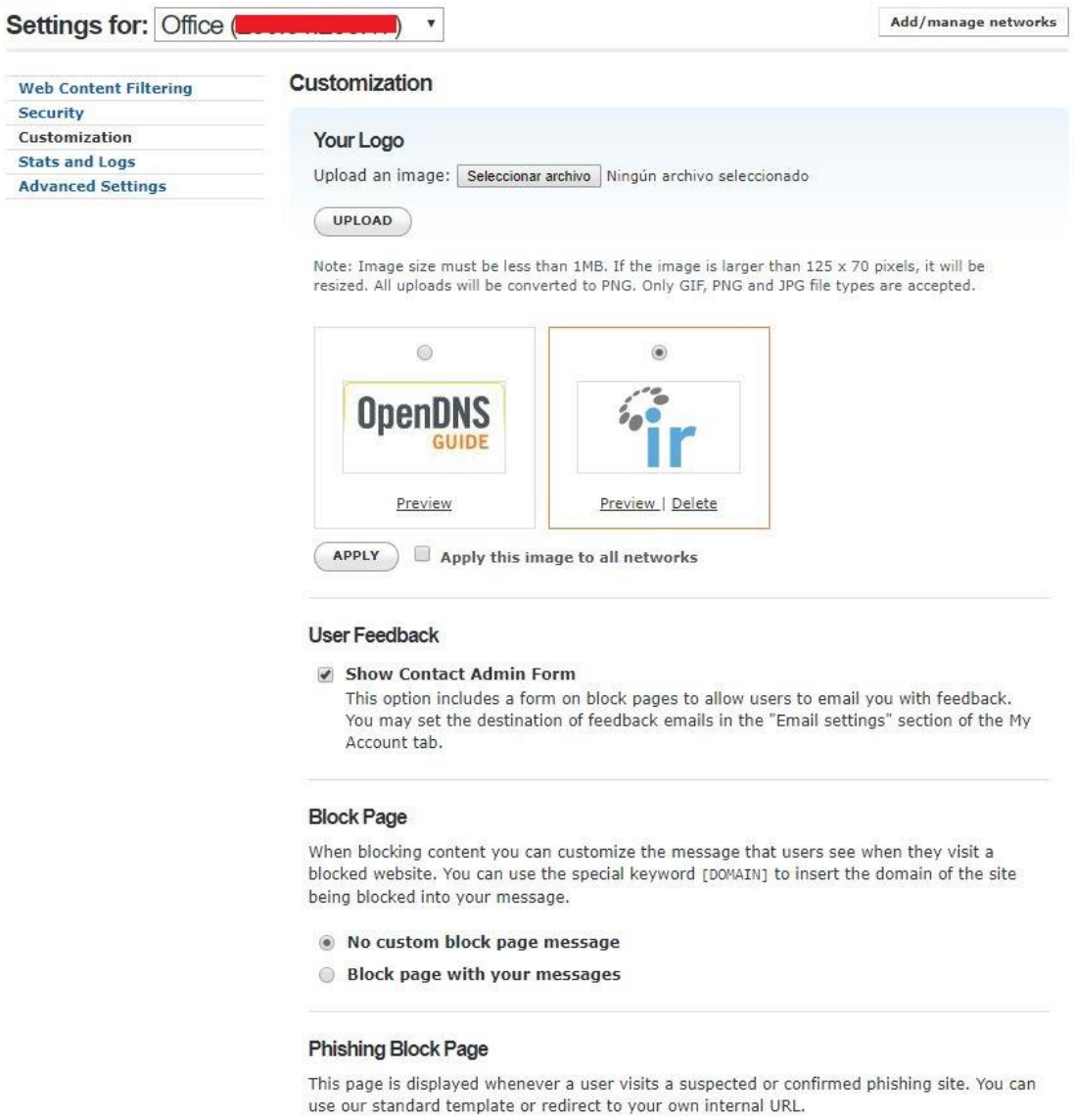
# Configuración de perfil en OpenDNS

**Tiene opciones, no solo de filtrado por categorías, sino también de Sitios considerados como Botnet/Malware o Pishing.**



# Configuración de perfil en OpenDNS

Se puede customizar la respuesta a las Páginas de Bloqueo incluyendo el Logo de nuestra Empresa y un mensaje.



# Configuración de perfil en OpenDNS

### Block Page

When blocking content you can customize the message that users see when they visit a blocked website. You can use the special keyword `[DOMAIN]` to insert the domain of the site being blocked into your message.

- (•) **No custom block page message**
- ( ) **Block page with your messages**

### Phishing Block Page

This page is displayed whenever a user visits a suspected or confirmed phishing site. You can use our standard template or redirect to your own internal URL.

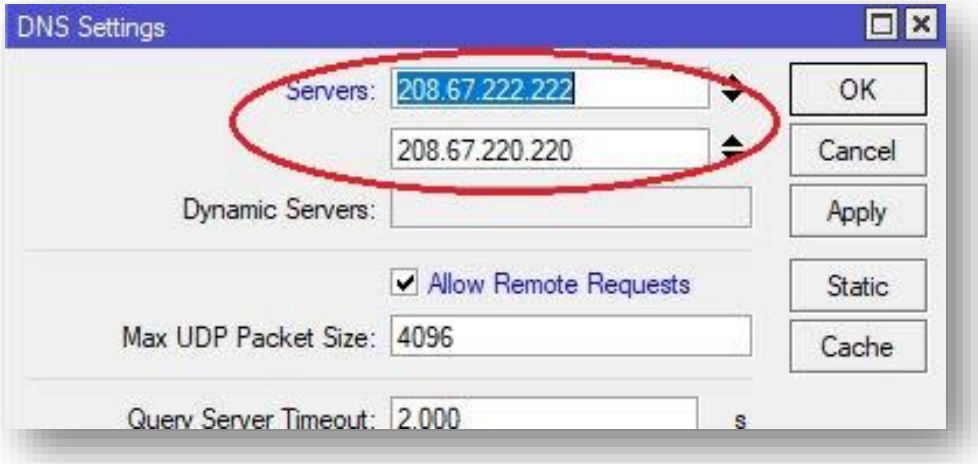
- (•) **No custom phishing block page message**
- ( ) **Phishing block page message**

APPLY [ ] **Apply to all networks**

# Configuración de OpenDNS dentro de RouterOS

Tenemos dos opciones:

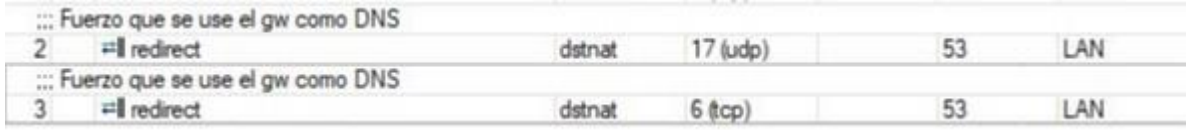
1. Configuramos los DNS de nuestro Router con los DNS de OpenDNS y lo configuramos como DNS Server de nuestra Red Lan.

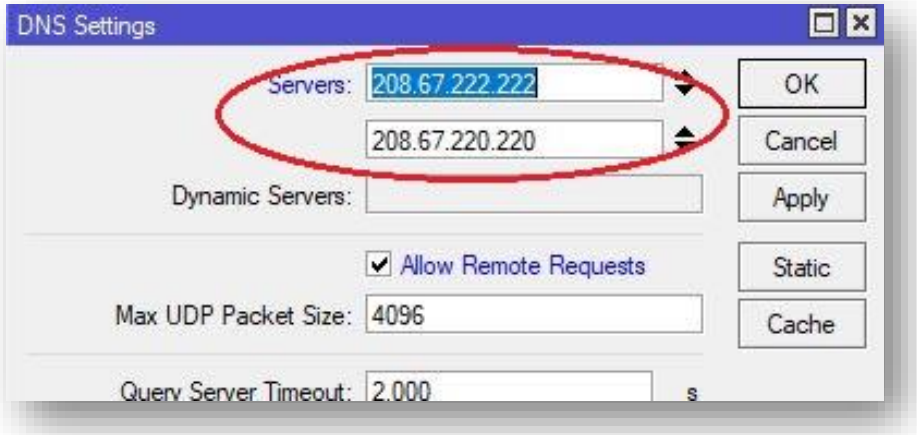


# Configuración de OpenDNS dentro de RouterOS

- No necesitamos mayor configuración.
- Podemos solamente crear una regla de NAT de redirect para forzar que todas las consultas DNS las atienda nuestro Router, de esta manera si alguien cambia sus DNS, las consultas las respondería igualmente el propio Router.

Recordar no solo incluir el puerto 53 UDP, sino también 53 TCP, ya que las consultas demasiado grandes usan TCP.




```
/ip firewall nat
add action=redirect chain=dstnat comment="Fuerzo que se use el gw como DNS" dst-port=53 in-interface-list=LAN protocol=udp
add action=redirect chain=dstnat comment="Fuerzo que se use el gw como DNS" dst-port=53 in-interface-list=LAN protocol=tcp
```

# Configuración de OpenDNS dentro de RouterOS

Desventaja:

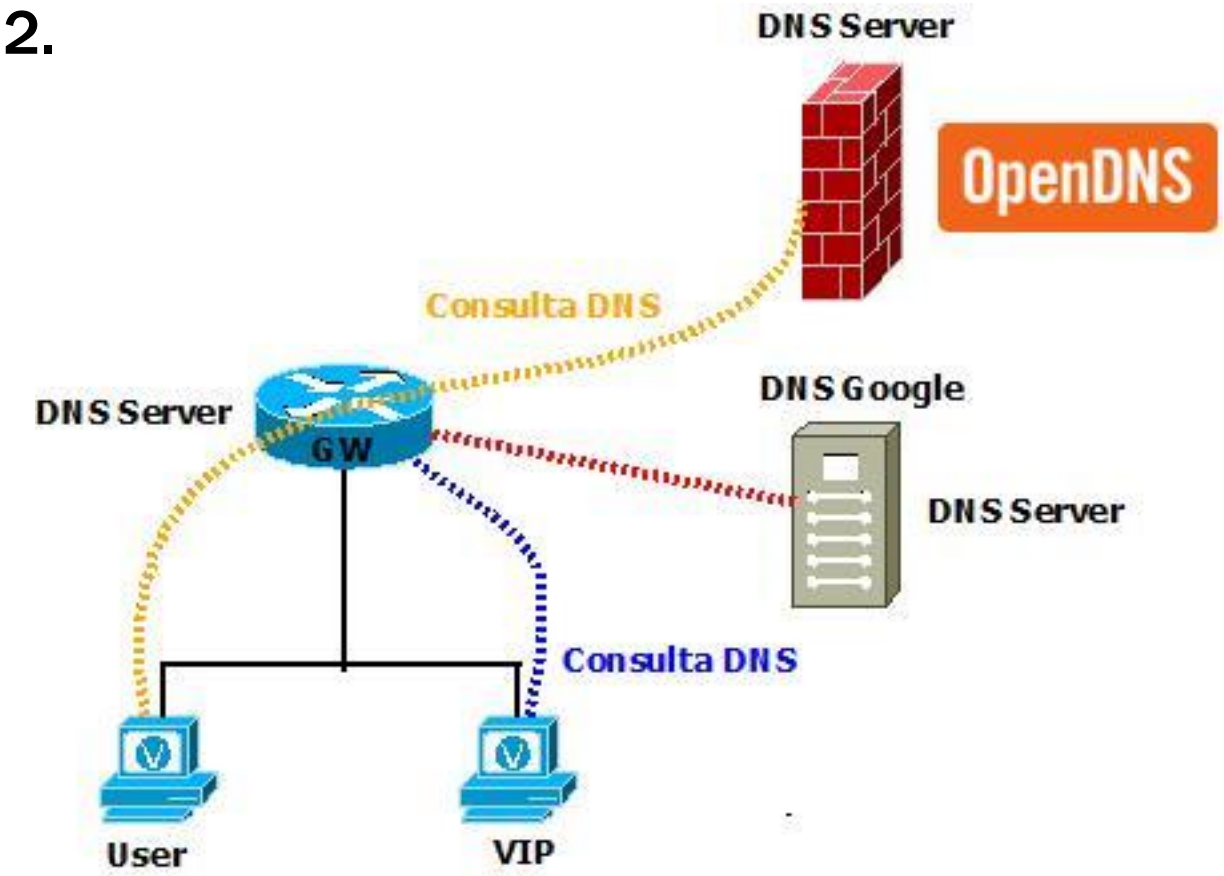
- Todos los Equipos de nuestra Red Local estarían usando la misma Política de Filtrado Web.

¿Qué pasa si tenemos personal VIP que requieren navegar sin restricciones?

Para eso pasamos a la Segunda opción.

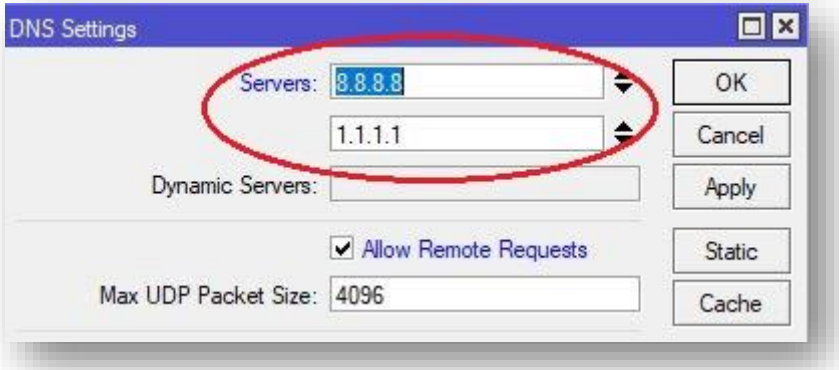
# Configuración de OpenDNS dentro de RouterOS

Opción 2.

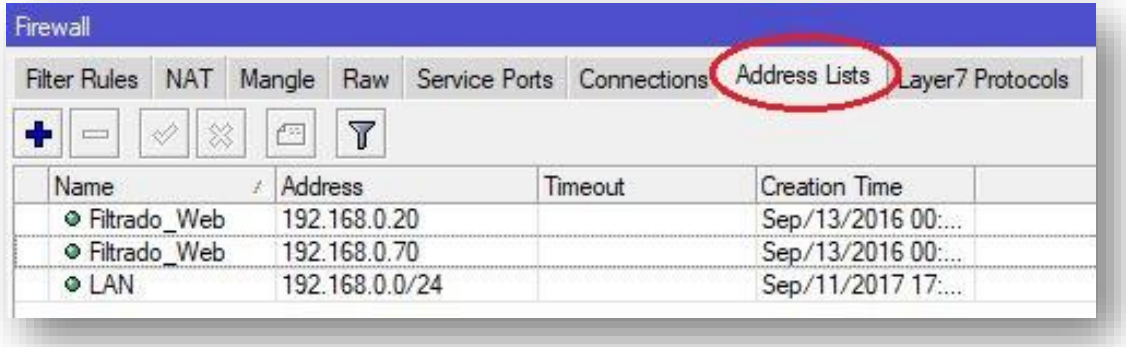


# Configuración de OpenDNS dentro de RouterOS

Configuramos otros DNS que no sean los de OpenDNS en nuestro Router y lo configuramos como DNS Server de nuestra Red.
Por ejemplos los DNS de Google o CloudFlare.




Luego, generar un address-list y dentro de esa lista poner las IP que queremos usar el Filtrado Web.

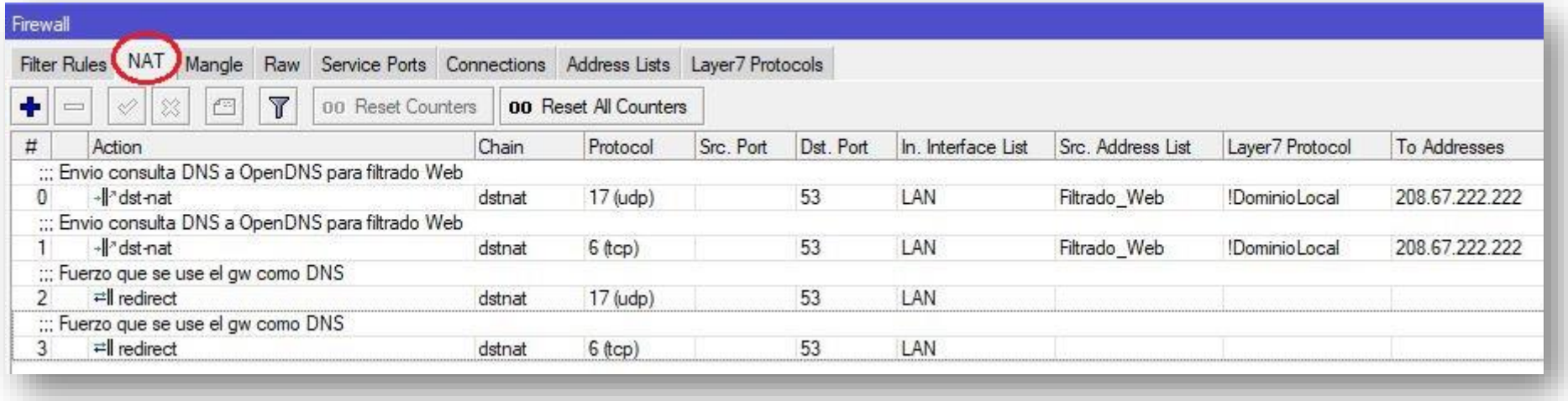


# Configuración de OpenDNS dentro de RouterOS

- Ya con el address-list con las IP que queremos usar el Filtrado Web, creamos reglas de Destination NAT para redirigir las consultas DNS de esos orígenes al DNS de OpenDNS.

- Debajo hacemos la misma regla de redirect para que las demás consultas las siga atendiendo nuestro Router, pero sin restricciones de navegación.

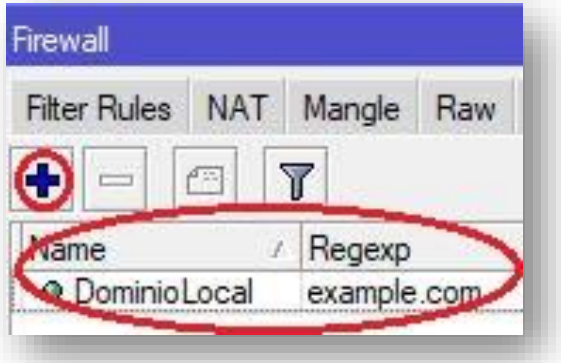
- Recordar en ambos casos las reglas para UDP 53 y TCP 53.




| # | Action | Chain | Protocol | Src. Port | Dst. Port | In. Interface List | Src. Address List | Layer7 Protocol | To Addresses |
|---|---|---|---|---|---|---|---|---|---|
| ;;; Envio consulta DNS a OpenDNS para filtrado Web | | | | | | | | | |
| 0 | dst-nat | dstnat | 17 (udp) | | 53 | LAN | Filtrado_Web | !DominioLocal | 208.67.222.222 |
| ;;; Envio consulta DNS a OpenDNS para filtrado Web | | | | | | | | | |
| 1 | dst-nat | dstnat | 6 (tcp) | | 53 | LAN | Filtrado_Web | !DominioLocal | 208.67.222.222 |
| ;;; Fuerzo que se use el gw como DNS | | | | | | | | | |
| 2 | redirect | dstnat | 17 (udp) | | 53 | LAN | | | |
| ;;; Fuerzo que se use el gw como DNS | | | | | | | | | |
| 3 | redirect | dstnat | 6 (tcp) | | 53 | LAN | | | |

# Configuración de OpenDNS dentro de RouterOS

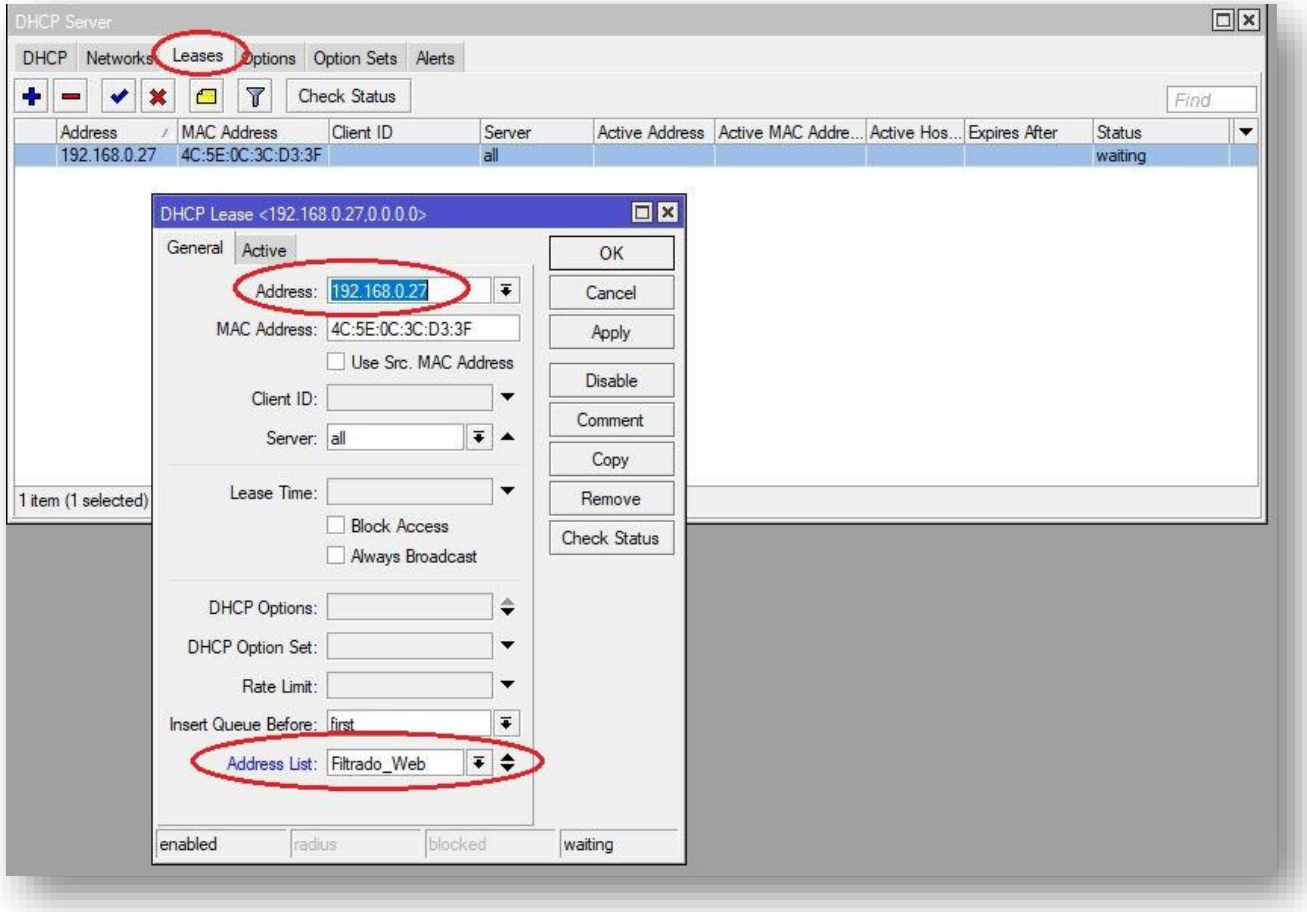
Adicionalmente, si generamos una regla de Layer7 para exceptuar enviar las consultas DNS de nuestro dominio local a OpenDNS, seguimos usando el Servicio de Filtrado Web con respuestas locales sin filtrar.




```
/ip firewall address-list
add address=192.168.0.0/24 list=LAN
add address=192.168.0.20 list=Filtrado_Web
add address=192.168.0.70 list=Filtrado_Web
/ip firewall nat
add action=dst-nat chain=dstnat comment="Envio consulta DNS a OpenDNS para filtrado Web" dst-port=53 in-interface-list=LAN layer7-protocol=!DominioLocal protocol=udp src-address-list=Filtrado_Web to-addresses=208.67.222.222
add action=dst-nat chain=dstnat comment="Envio consulta DNS a OpenDNS para filtrado Web" dst-port=53 in-interface-list=LAN layer7-protocol=!DominioLocal protocol=tcp src-address-list=Filtrado_Web to-addresses=208.67.222.222
add action=redirect chain=dstnat comment="Fuerzo que se use el gw como DNS" dst-port=53 in-interface-list=LAN protocol=udp
add action=redirect chain=dstnat comment="Fuerzo que se use el gw como DNS" dst-port=53 in-interface-list=LAN protocol=tcp
```

# Configuración de OpenDNS dentro de RouterOS

También se puede agregar IPs dinámicamente a nuestra lista de orígenes filtrados usando DHCP Server, haciendo entradas estáticas para los Equipos agregados que cuando se conecten, sean agregados a la lista que definimos como “Filtrado_Web”



# ¿Preguntas?

# Muchas Gracias.