

MUM Madrid 2015

Tu socio tecnológico

nunsys®

SOFTWARE · COMUNICACIONES · SISTEMAS · CONSULTORÍA · FORMACIÓN · MARKETING

Scripting en RouterOS

Scripting en RouterOS

MUM Madrid 2015

- Félix Calvo
- felix@nunsys.com
- RouterOS desde versión 3.20
- MikroTik Certified Consultant
- Certificaciones:
 - MTCNA
 - MTCRE
 - MTCWE
 - MTCTCE
- Miembro reconocido en foros de habla hispana con el seudónimo **fcmontes**

Centro de Formación Permanente de la
Universidad Politécnica de Valencia
<http://www.formacionmikrotik.com>



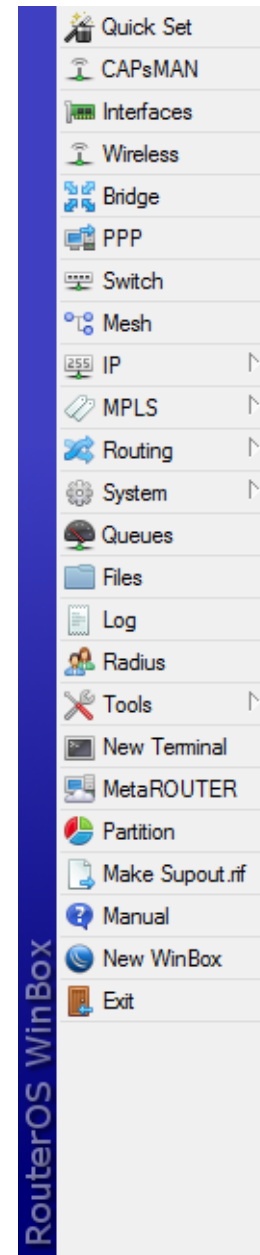
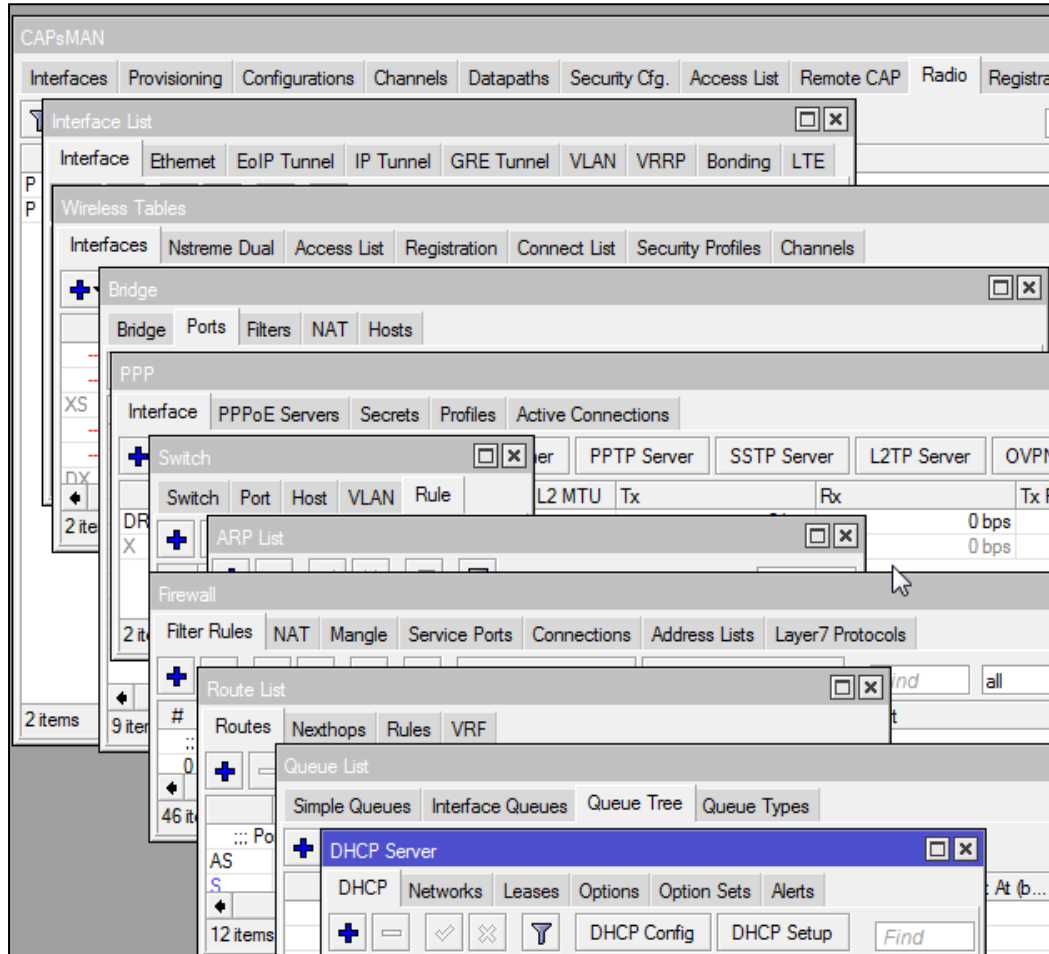
Índice

- 1. ¿Qué esperamos de RouterOS?
- 2. Scripting, el siguiente nivel.
- 3. Scripting en RouterOS
- 4. Ejemplos prácticos de aplicación
- 5. Ruegos y preguntas



1. ¿Qué esperamos de RouterOS?

- ¿Qué esperamos de RouterOS?
- ¿Podemos hacerlo todo con Winbox?



- ¿Qué esperamos de RouterOS?
 - Casi todo, sin duda.
 - Pero... ¿qué hacer cuando no llegamos con Winbox?
 - Necesito ejecutar una acción si ocurre...
 - Necesito que me avise enviando un correo si acontece...
 - Necesito desactivar una reglas si el tráfico supera...



2. Scripting, el siguiente nivel.

- **Scripting, el siguiente nivel**
 - **En RouterOS con un script podemos...**
 - ✓ Automatizar tareas de mantenimiento del router.
 - ✓ Gestión interna y alertas.
 - ✓ Usos definidos por el usuario...





3. Scripting en RouterOS

- **Scripting**

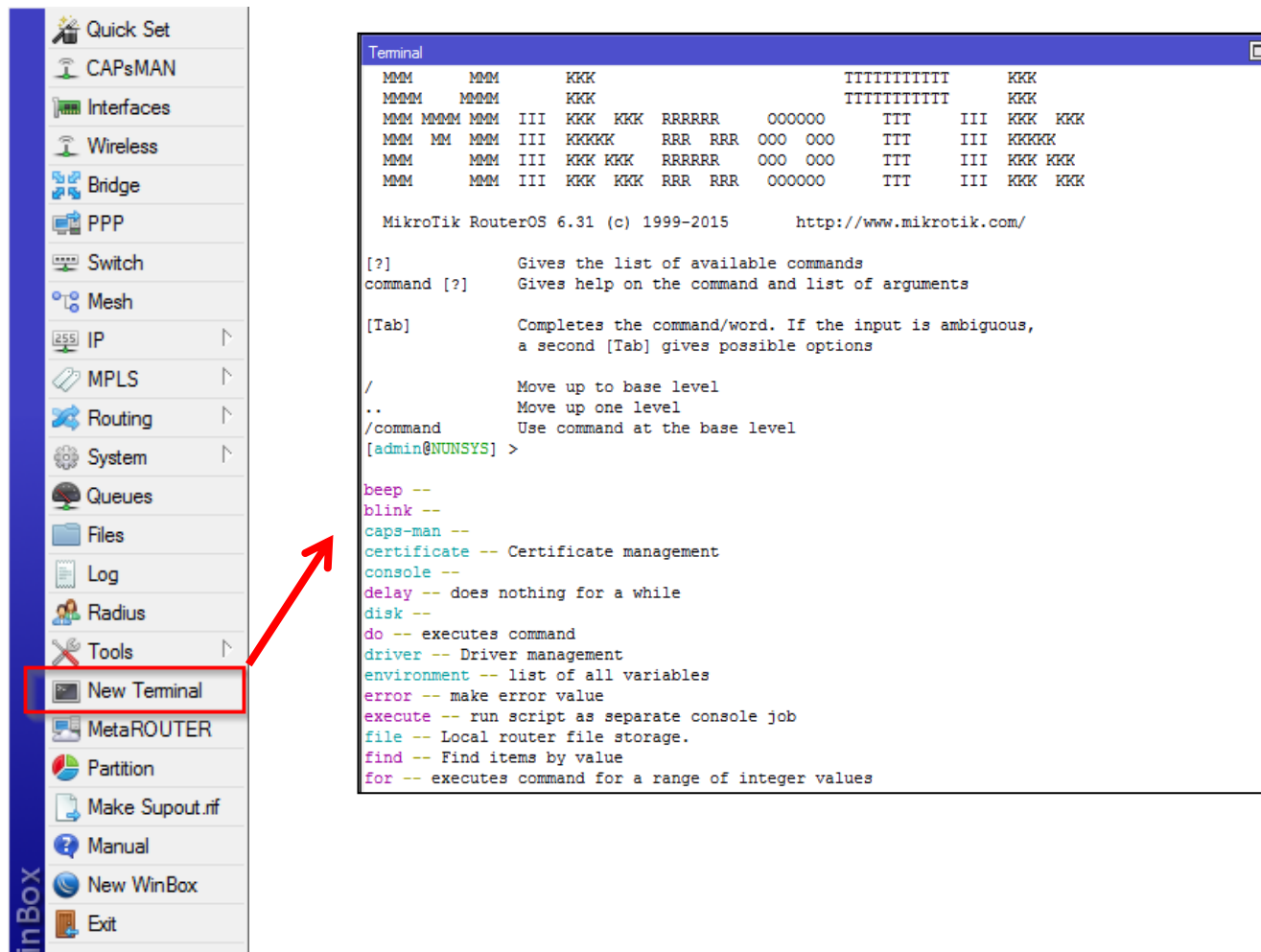
10

- **Los primeros pasos.**



• Scripting

- ✓ Un script, se puede desarrollar y ejecutar desde la consola.



• Scripting

- Aprovechemos las ayudas dentro de cada nivel

✓ Utilizando la tecla del tabulador lateral...

✓ **F1, ? y Tab**, son las teclas de ayuda que nos permiten identificar los distintos niveles dentro de cada menú y su descripción.

✓ La guía de ayuda para componer la línea de forma correcta.



```
Terminal
[admin@NUNSYS] /ip>
.. -- go up to root
accounting -- Traffic accounting
address -- Address management
arp -- ARP entries management
cloud --
dhcp-client -- DHCP client settings
dhcp-relay -- DHCP relay settings
dhcp-server -- DHCP server settings
dns -- DNS settings
firewall -- Firewall management
hotspot -- HotSpot servers management
ipsec -- IP security
neighbor -- Neighbors
packing -- Packet packing settings
pool -- IP address pool
proxy --
route -- Route management
service -- IP services
settings --
smb --
socks -- SOCKS version 4 proxy
ssh --
tftp -- TFTP
traffic-flow --
upnp -- Universal Plug and Play
export -- Print or save an export script that can be used to restore configuration
```

```
[admin@NUNSYS] > ip
accounting  dhcp-client  firewall  packing  service  ssh  export
address     dhcp-relay   hotspot  pool     settings tftp
arp          dhcp-server  ipsec    proxy    smb      traffic-flow
cloud        dns          neighbor route     socks    upnp
```

• Scripting

- Aprovechemos las ayudas dentro de cada nivel
- ✓ Autocompletar palabras o comandos, con distintos colores, azul, fucsia, verde, correspondiente a menú, comando y parámetro, respectivamente.

```
[admin@NUNSYS] > ip address find comment [ ]
```

- ✓ En cada nivel, nos mostrara además de las distintas opciones, varias formas de completar la línea de acción, consiguiendo el mismo resultado.

```
[admin@NUNSYS] > put [/ip address get [find
.dead .nextid          address  comment  dynamic  invalid
.id   actual-interface broadcast disabled interface netmask
[admin@NUNSYS] > put [/ip address get [find comment=LAN] address ]
192.168.110.254/24
```

```
[admin@NUNSYS] > put [/ip address get
value-name  number
[admin@NUNSYS] > put [/ip address get number=0 address ]
192.168.110.254/24
```

- **Scripting**

- **Aprovechemos las ayudas dentro de cada nivel**
- ✓ Y nos indicara de forma visual, si existe algún error de composición.

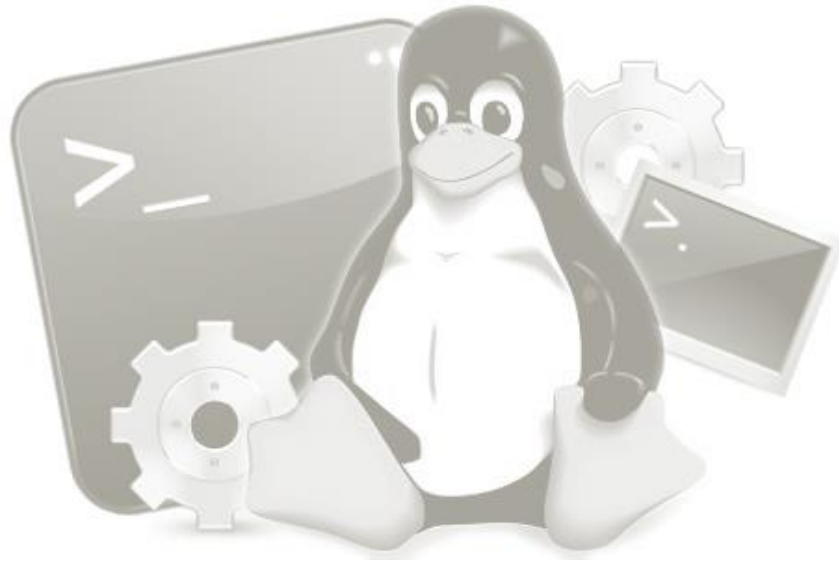
```
[admin@NUNSYS] > ip address [find comment
```

- ✓ Dejando de funcionar el autocompletado de comandos y ayudas.

```
/          Move up to base level
..         Move up one level
/command   Use command at the base level
[admin@NUNSYS] > ip address [find comment ghfhfhfghfhfghghgf
```

- **Scripting**

- ✓ **Algunos comandos importantes**



• Scripting

✓ Algunos comandos importantes

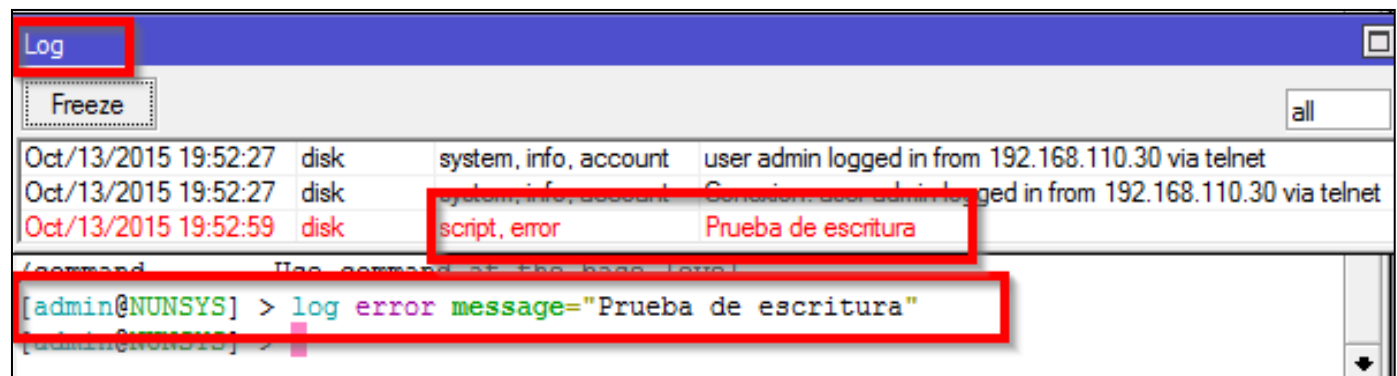
✓ put

- Muestra en consola el valor especificado

```
/command Use command at the base level
[admin@NUNSYS] > put "Distintas opciones"
Distintas opciones
[admin@NUNSYS] > put (5+5)
10
[admin@NUNSYS] >
```

✓ log

- Escribe sobre el registro de eventos del sistema



• Scripting

✓ Algunos comandos importantes

✓ get

- Obtiene el valor especificado

```
[admin@NUNSYS] > put [ip address get number=0 address ]
192.168.110.254/24
[admin@NUNSYS] > █
```

✓ set

- Modifica el valor seleccionado

```
/command          Use command at the base level
[admin@NUNSYS] > global IP [/ip address get number=0 address ]
[admin@NUNSYS] > put $IP
192.168.110.254/24
[admin@NUNSYS] > set $IP 10.10.10.10/30
[admin@NUNSYS] > put $IP
10.10.10.10/30
[admin@NUNSYS] > █
```

✓ find

- Busca dentro de la cadena definida

```
/command          Use command at the base level
[admin@NUNSYS] > :put [:find "abc.def" "."]
3
[admin@NUNSYS] > █
```

• Scripting

✓ Algunos comandos importantes

✓ len

- Especifica la cantidad de caracteres de una cadena

```
/command Use command at the base level
[admin@NUNSYS] > global Texto "MUM MIKROTIK SPAIN 16 DE OCTUBRE DE 2015"
[admin@NUNSYS] > put [:len $Texto]
40
[admin@NUNSYS] >
```

✓ pick

- Obtiene el/los caracteres seleccionados

```
/command Use command at the base level
[admin@NUNSYS] > global Texto "MUM MIKROTIK SPAIN 16 DE OCTUBRE DE 2015"
[admin@NUNSYS] > put [:len $Texto]
40
[admin@NUNSYS] >
[admin@NUNSYS] > put [:pick $Texto 4 12]
MIKROTIK
[admin@NUNSYS] >
```

✓ resolve

- Resuelve un dominio

```
/command Use command at the base level
[admin@NUNSYS] > put [resolve www.nunsys.com]
185.5.236.135
[admin@NUNSYS] >
```

- **Scripting**

- ✓ **Algunos comandos importantes**

- ✓ **Print**

- Podemos imprimir los datos y cantidades de una lista.

```
/command Use command at the base level
[admin@NUNSYS] > ip arp print
Flags: X - disabled, I - invalid, H - DHCP, D - dynamic, P - published
# ADDRESS MAC-ADDRESS INTERFACE
0 D 192.168.1.10 94:DE:80:B4:DD:8D bridge1
1 D 192.168.1.39 00:13:72:4C:92:F8 bridge1
2 D 192.168.1.20 8C:E7:48:27:3A:D3 bridge1
3 D 192.168.1.236 B8:6C:E8:90:38:2D bridge1
4 D 192.168.1.105 00:0B:AB:82:FC:DC bridge1
5 D 172.22.10.1 F8:3D:FF:EA:32:95 ether1
6 D 172.22.20.1 64:70:02:D8:24:C8 ether2
[admin@NUNSYS] > ip arp print count-only
7
[admin@NUNSYS] >
```

• Scripting

✓ Algunos comandos importantes

✓ Print

- Hay otros comandos que consiguen un efecto similar y nos permiten exportar información a un archivo .txt

```
Terminal
/command Use command at the base level
[admin@NUNSYS] > interface wireless monitor wlan1 once
status: running-ap
band: 2ghz-g
frequency: 2437MHz
wireless-protocol: 802.11
noise-floor: -102dBm
overall-tx-ccq: 63%
registered-clients: 2
authenticated-clients: 2
current-distance: 40
current-tx-powers: 1Mbps:27 (27/30), 2Mbps:25 (25/28), 5.5Mbps:25 (25/28),
11Mbps:25 (25/28), 6Mbps:27 (27/30), 9Mbps:27 (27/30),
12Mbps:27 (27/30), 18Mbps:27 (27/30), 24Mbps:27 (27/30),
36Mbps:25 (25/28), 48Mbps:24 (24/27), 54Mbps:21 (21/24)
notify-external-fdb: no
[admin@NUNSYS] >

[admin@NUNSYS] > interface wireless monitor wlan1
append as-value do file interval once without-paging
[admin@NUNSYS] > interface wireless monitor wlan1 file=
```

- Scripting

- Los datos



• Scripting

- ✓ Podemos definir tipos de datos/variables.

- ✓ Numéricas

```
/command Use command at the base level
[admin@NUNSYS] > put (2+2)
4
[admin@NUNSYS] >
```

- ✓ Alfanuméricas

```
/command Use command at the base level
[admin@NUNSYS] > put ("Alfanumerica+1")
Alfanumerica+1
[admin@NUNSYS] >
```

- ✓ Lógicas

```
/command Use command at the base level
[admin@NUNSYS] > put (2+2=4)
true
[admin@NUNSYS] >
```

- **Scripting**
 - **Variables**

✓ Números

```
/ Command
[admin@NUNSYS] > global Numero 5
[admin@NUNSYS] > put $Numero
5
[admin@NUNSYS] >
```

✓ Valores booleanos

```
[admin@NUNSYS] > global Pregunta (5+6=10)
[admin@NUNSYS] > put $Pregunta
false
[admin@NUNSYS] >
```

✓ Cadenas

```
/ Command
[admin@NUNSYS] > global Secuencia (1,2,3,4,6)
[admin@NUNSYS] > put $Secuencia
1;2;3;4;6
[admin@NUNSYS] >
```

• Scripting

○ Variables

✓ IPV4 y 6

```
2.168.110.0
[admin@NUNSYS] > global IP [/ip address get number=0 address ]
[admin@NUNSYS] > put $IP
192.168.110.254/24
[admin@NUNSYS] >
```

✓ Identificadores

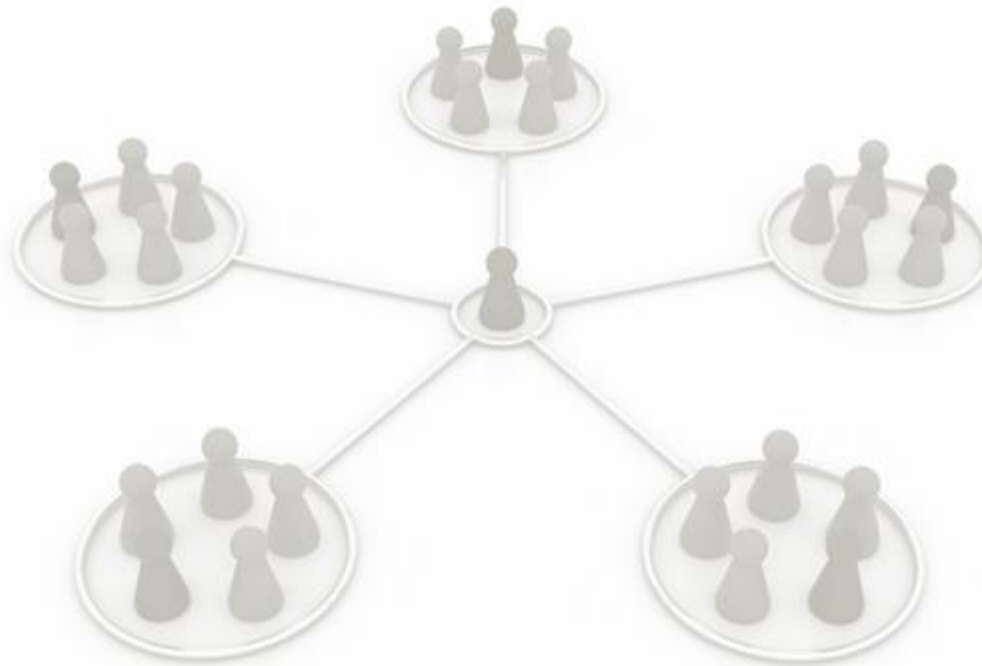
```
[admin@NUNSYS] >
[admin@NUNSYS] > global Vecino [/ip neighbor get number=0]
[admin@NUNSYS] > put $Vecino
.id=*3;address=192.168.110.253;address4=192.168.110.253;age=0
```

✓ Fecha y hora

```
/command Use command at the base level
[admin@NUNSYS] > global Fecha [/system clock get date]
[admin@NUNSYS] > global Hora [/system clock get time]
[admin@NUNSYS] > global Resta ($Hora-01:00:00)
[admin@NUNSYS] > put "Fecha $Fecha"
Fecha oct/10/2015
[admin@NUNSYS] > put "Hora Peninsular $Hora"
Hora Peninsular 02:50:32
[admin@NUNSYS] > put "Hora en Canarias $Resta"
Hora en Canarias 01:50:32
[admin@NUNSYS] >
```


- Scripting

➤ El Entorno.



• Scripting

○ Ámbitos de aplicación (“Scopes”)

➤ Local y Global.

- **Local**, se restringe al entorno en el que se desarrolla dentro del Script, fuera de el, no existe.

```

/Command (Use Command at the base level)
[admin@NUNSYS] > :local uptime [/system resource get uptime];
[admin@NUNSYS] > put $uptime
[admin@NUNSYS] >

```

```

MMM      MMM      KKK
MMM      MMM      KKK
MMM MMM MMM III KKK KKK RRRRRR
MMM MM  MMM III KKKKK RRR RRR
MMM  MMM III KKK KKK RRRRRR
MMM    MMM III KKK KKK RRR RRR

MikroTik RouterOS 6.32.2 (c) 1999-2015

[?]          Gives the list of available
command [?]  Gives help on the command

[Tab]        Completes the command/word
              a second [Tab] gives possible
              completions

/            Move up to base level
..          Move up one level
/command     Use Command at the base level
[admin@NUNSYS] > sys script run prueba
00:15:50
[admin@NUNSYS] >

```

Script <prueba>

Name: prueba
Owner: admin
Policy:
☒ ftp
☒ read
☒ policy
☒ password
☒ sensitive
☒ reboot
☒ write
☒ test
☒ sniff
Last Time Started: Oct/09/2015 18:13:52
Run Count: 1
Source:

```

local uptime [/system resource get uptime]
put $uptime;

```

OK
Cancel
Apply
Comment
Copy
Remove
Run Script

- **Scripting**

- **Ámbitos (“Scopes”)**

- **Global**, se almacena dentro del apartado “environment” en la memoria volátil del mkt y puede ser modificado y usado por otros scripts.

The screenshot displays the MikroTik WinBox interface. The top section shows a terminal window with the following commands and output:

```
/command Use command at the base level
[admin@NUNSYS] > global uptime [/system resource get uptime]
;
[admin@NUNSYS] > put $uptime
2w2d17:12:09
[admin@NUNSYS] >
```

The bottom section shows the 'Script List' tab. The 'Environment' sub-tab is selected. A table displays the environment variables:

User	Name	Value
	uptime	2w2d17:12:09

Red boxes highlight the command `global uptime`, the output `2w2d17:12:09`, the 'Environment' tab, and the 'uptime' variable entry in the table.

- Scripting

➤ Operaciones

$$\begin{array}{r}
 8x^3 - 4x^2 + 2x + 7 \\
 -8x^3 - 4x^2 + 4x \\
 \hline
 -8x^2 + 6x + 7 \\
 + 8x^2 + 4x - 4 \\
 \hline
 10x + 3
 \end{array}
 \quad
 \begin{array}{r}
 2x^2 + x - 1 \\
 \hline
 4x - 4
 \end{array}$$

• Scripting

○ Operaciones

✓ Aritméticas

```
[admin@NUNSYS] > global Suma (5+7)
[admin@NUNSYS] > put $Suma
12
```

✓ Relacionales

```
[admin@NUNSYS] > :put (3<4);
true
[admin@NUNSYS] > :put (4<3)
false
```

✓ Lógicas

```
[admin@NUNSYS] > put (192.168.10.20 in 192.168.10.0/24)
true
[admin@NUNSYS] > put (192.168.10.20 in 192.168.2.0/24)
false
```

✓ Sobre IPv4 y 6

```
/command Use command at the base level
[admin@NUNSYS] > :put (~255.255.255.248)
0.0.0.7
[admin@NUNSYS] >
```

✓ Concatenaciones

```
/command Use command at the base level
[admin@NUNSYS] > global Texto0 MIKROTIK
[admin@NUNSYS] > global Texto1 MUM
[admin@NUNSYS] > global Texto2 SPAIN
[admin@NUNSYS] > global Texto3 2015
[admin@NUNSYS] > put ("{$Texto0}." " {$Texto1}." " {$Texto2}." " {$Texto3}")
MIKROTIK MUM SPAIN 2015
[admin@NUNSYS] >
```

- Scripting

- Bucles.



- Scripting

- Bucles

➤ (while) Ejecuta acción hasta cumplir condición.

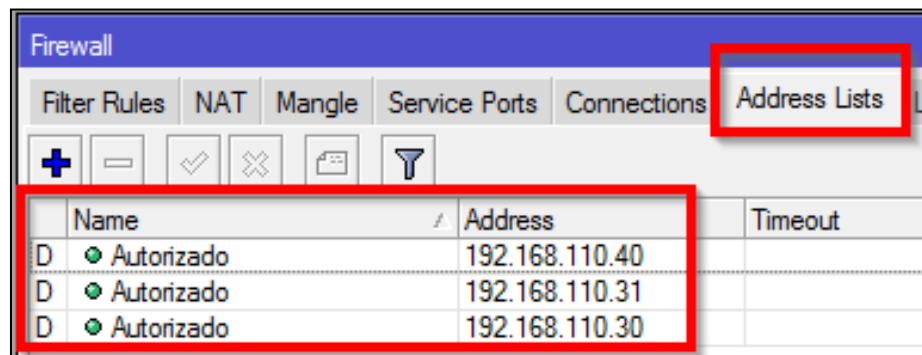
```
[admin@NUNSYS] > global valor 0
[admin@NUNSYS] > :while ($valor<=5) do={:set $valor ($valor+1); :put $valor;}
1
2
3
4
5
6
[admin@NUNSYS] > 
```

➤ (for) Ejecuta acción un numero limitado de veces.

```
.. Move up one level
/command Use command at the base level
[admin@NUNSYS] > global valor 0
[admin@NUNSYS] > :for a from 1 to 5 do={:set $valor ($a); :put $valor;}
1
2
3
4
5
[admin@NUNSYS] > 
```

- **Scripting**
 - **Bucles**

- **(foreach)** Ejecuta acción en todos los elementos que encuentra en una lista.



	Name	Address	Timeout
D	Autorizado	192.168.110.40	
D	Autorizado	192.168.110.31	
D	Autorizado	192.168.110.30	

```
/command      Use command at the base level
[admin@NUNSYS] > global busqueda [/ip firewall address-list find dynamic=yes list="Autorizado"];
[admin@NUNSYS] > :foreach a in=[$busqueda] do={\
{\... :global IP [/ip firewall address-list get $a address];
{\... :put ("IP Autorizada="."."$IP")}
IP Autorizada=192.168.110.40
IP Autorizada=192.168.110.31
IP Autorizada=192.168.110.30
[admin@NUNSYS] >
```


- Scripting

- Condiciones.

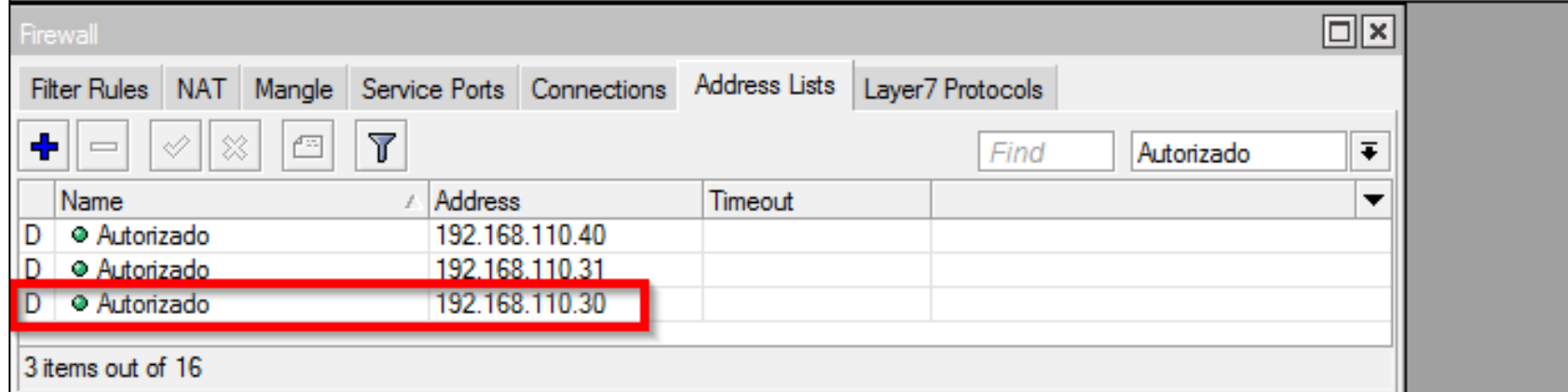


- Scripting

- Condiciones

- (if) Ejecuta acción si se cumple la condición o condiciones definidas.

```
/command Use command at the base level
[admin@NUNSYS] > global Pruebas (192.168.110.30)
[admin@NUNSYS] > global busqueda [/ip firewall address-list find dynamic=yes list="Autorizado"];
[admin@NUNSYS] > :foreach a in=[$busqueda] do={\
{\... :global IP [/ip firewall address-list get $a address];
{\... :delay (10ms);
{\... :if ($IP=$Pruebas) do={\
{{\... :put ("ATENCION: IP de pruebas detectada="."."$IP");}}
ATENCION: IP de pruebas detectada=192.168.110.30
[admin@NUNSYS] >
```



Name	Address	Timeout
Autorizado	192.168.110.40	
Autorizado	192.168.110.31	
Autorizado	192.168.110.30	

3 items out of 16

• Scripting

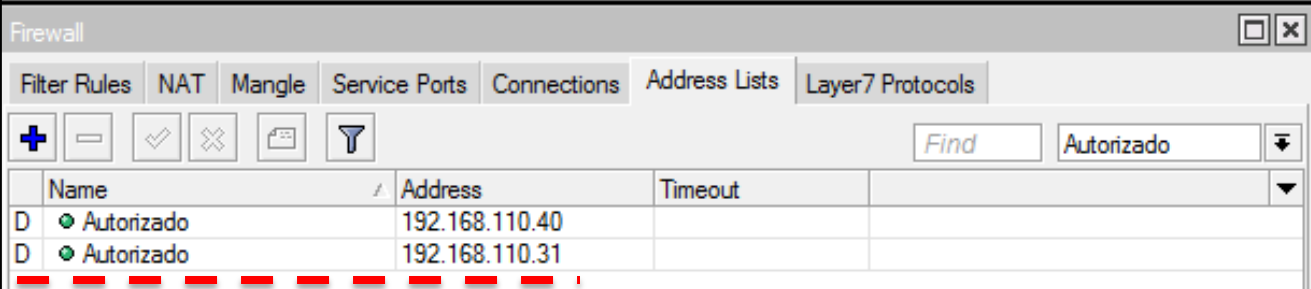
○ Condiciones

- (else) A continuación de la condición afirmativa, podemos definir también, la negativa.

```

/command          Use command at the base level
[admin@NUNSYS] > global Pruebas (192.168.110.30)
[admin@NUNSYS] > global busqueda [/ip firewall address-list find dynamic=yes list="Autorizado"];
[admin@NUNSYS] > :foreach a in=[$busqueda] do={\
{\... :global IP [/ip firewall address-list get $a address];
{\... :delay (10ms);
{\... :if ($IP=$Pruebas) do={\
{{\... :put ("ATENCION: IP de pruebas detectada="."."$IP");} else={\
{{\... :put ("No detectada IP buscada");
{{\... }}
No detectada IP buscada
No detectada IP buscada
[admin@NUNSYS] >
[admin@NUNSYS] >

```



Name	Address	Timeout
Autorizado	192.168.110.40	
Autorizado	192.168.110.31	

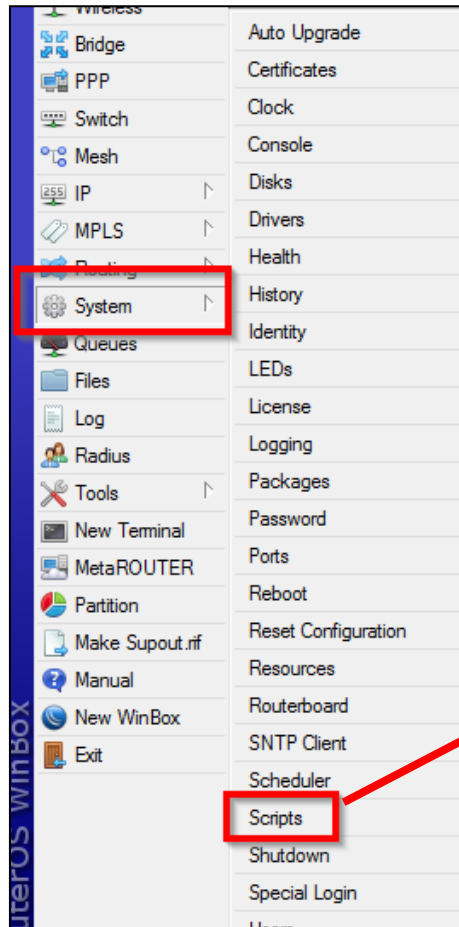
- **Scripting**

- **Scripts ¿dónde están?**



• Scripting

- ✓ Se agrupan en un apartado específico, dentro del nivel de sistema, en el que encontramos 3 pestañas.



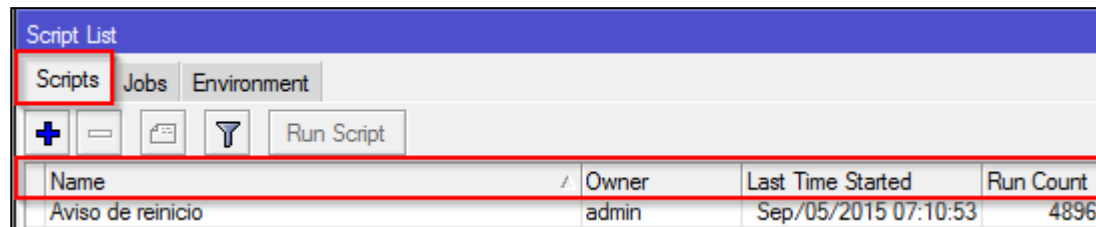
```
[admin@NUNSYS] > system script
environment job add comment edit export find print remove run set
[admin@NUNSYS] > system script
```

Name	Owner	Last Time Started	Run Count
Alama	admin		0
Aviso de reinicio	admin	Sep/05/2015 07:01:53	4894
Bloqueo_Periodicos_Deportivos_L7	admin	Sep/05/2015 01:00:00	16
Bloqueo_Periodicos_Digitales2_L7	admin	Sep/05/2015 01:10:00	16
Bloqueo_Periodicos_Digitales_L7	admin	Sep/05/2015 01:15:00	16
Bloqueo_Reddes_Sociales_L7	admin	Sep/05/2015 01:20:00	16
Cargar Address List	admin		0
Cargar tabla ARP	admin	Jun/18/2014 20:27:54	3
Cloud	admin	Sep/05/2015 07:01:16	442
Comprobar Tunel VPN	admin	Sep/05/2015 06:54:26	1468
Comprobar email	admin	Sep/04/2015 16:31:30	15
Conexion_MKT	admin	Sep/04/2015 16:04:36	20
Crear colas simples	admin		0
Descargar Categorías filtrado L7	admin	Sep/05/2015 00:30:00	85
Dydnns_3G	admin	Sep/05/2015 07:04:14	735
Dydnns	admin	Sep/05/2015 07:04:37	2488
Envio mensual backup	admin	Sep/03/2014 12:47:07	4
Liberar IPs DHCP	admin		0
Limpiar tabla ARP	admin		0
Mario Bros	admin		0
Modem_3G	admin		0
OpenDNS-1 Down	admin		0
OpenDNS-1 Up	admin		0
OpenDNS-2 Down	admin		0
OpenDNS-2 Up	admin		0
Reinicio equipo por carga CPU	admin	Sep/05/2015 06:56:12	2202
Reinicio_mensual	admin		0
Servidor_IP	admin	Sep/05/2015 07:02:46	4412
Upgrade Firmware	admin	Mar/03/2015 20:57:21	1
ether1Down	admin		0
ether1Up	admin		0
facebook	admin		0
queues	admin	Sep/05/2015 05:41:00	597
script1	admin		0
script2	admin		0
script3	admin		0

- **Scripting**

- **Scripts.**

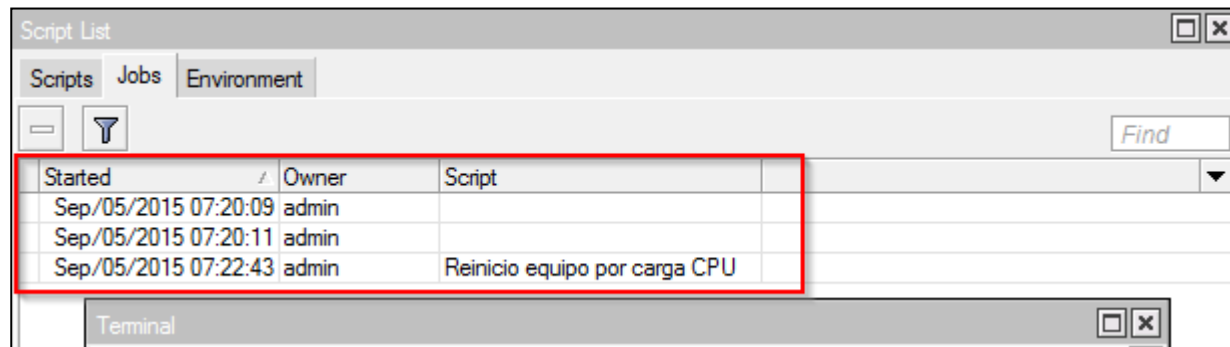
- ✓ Es la raíz donde se agrupan.



Name	Owner	Last Time Started	Run Count
Aviso de reinicio	admin	Sep/05/2015 07:10:53	4896

- **Jobs.**

- ✓ Podemos ver los que se están ejecutando



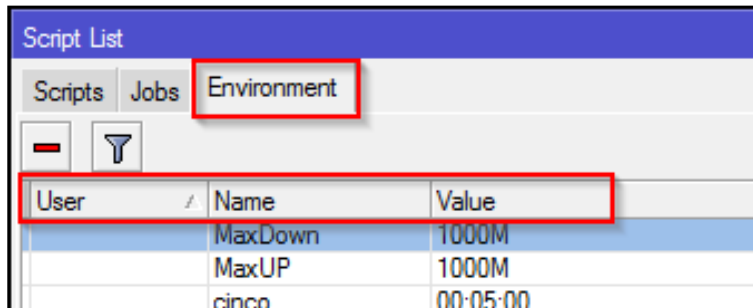
Started	Owner	Script
Sep/05/2015 07:20:09	admin	
Sep/05/2015 07:20:11	admin	
Sep/05/2015 07:22:43	admin	Reinicio equipo por carga CPU

- ✓ Cada ventana del terminal, es identificada como una tarea individual.
 - ✓ Desde esta pestaña podemos interrumpir la ejecución de un script.

• Scripting

• Y Enviroment.

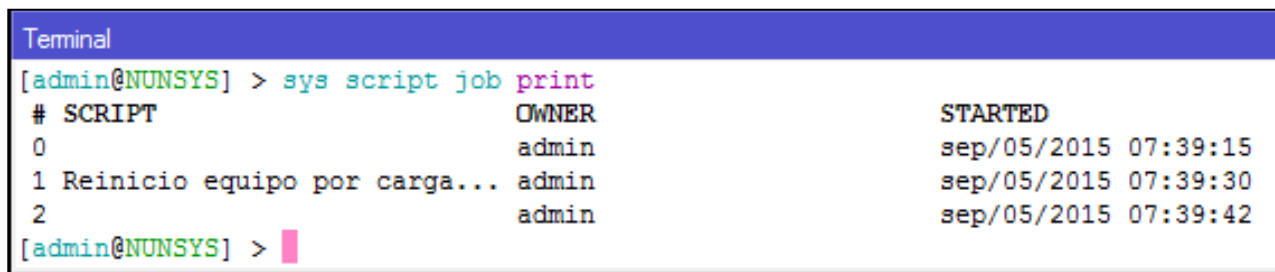
✓ Podemos ver las variables denominadas.



User	Name	Value
admin	MaxDown	1000M
admin	MaxUP	1000M
admin	cinco	00:05:00

✓ Todas las pestañas aquí descritas, también son accesibles desde consola, para posibles usos.

```
[admin@NUNSYS] > system script print
append brief count-only detail file follow follow-only from interval where without-paging
```



```
Terminal
[admin@NUNSYS] > sys script job print
# SCRIPT          OWNER          STARTED
0                admin          sep/05/2015 07:39:15
1 Reinicio equipo por carga... admin          sep/05/2015 07:39:30
2                admin          sep/05/2015 07:39:42
[admin@NUNSYS] >
```

- **Scripting**

- ¿Cómo se ejecutan?



- **Scripting**

- Pueden ser llamados (ejecutados), desde distintos apartados de RouterOS

✓ VRRP, DHCP, HOTSPOT, Netwatch, Traffic Monitor...

The screenshot shows the 'Traffic Monitor' configuration window. The title bar reads 'Traffic Monitor <ether1-UP-porencima>'. The configuration fields are: Name: ether1-UP-porencima, Interface: ether1, Traffic: transmitted, Trigger: above, Threshold: 800000, and On Event: ether1-porencima. The 'On Event' field is highlighted with a red box. The window includes buttons for OK, Cancel, Apply, Enable, Comment, Copy, and Remove.

The screenshot shows the 'Netwatch Host' configuration window. The title bar reads 'Netwatch Host <212.166.210.3>'. The configuration fields are: Host: 212.166.210.3, Interval: 00:01:00, Timeout: 1000 ms, Status: up, and Since: Sep/05/2015 08:29:09. The 'Status' field is highlighted with a red box. The window includes buttons for OK, Cancel, Apply, Disable, Comment, and Copy.

The screenshot shows the 'Netwatch Host' configuration window. The title bar reads 'Netwatch Host <212.166.210.3>'. The configuration fields are: Host: 212.166.210.3, Interval: 00:01:00, Timeout: 1000 ms, Status: up, and Since: Sep/05/2015 08:29:09. The 'Status' field is highlighted with a red box. The window includes buttons for OK, Cancel, Apply, Disable, Comment, and Copy.

• Scripting

- ✓ Se puede programar su ejecución, un día y hora determinado y/o con un intervalo definido.



• Scheduler

- ✓ Podemos automatizar, definiendo:
 - Día y hora de ejecución.
 - En cada reinicio.
 - Intervalos, horas, minutos y segundos.
 - Políticas.
- ✓ Tiene un contador de ejecuciones.
- ✓ Un aviso de próxima ejecución y estado.
- ✓ Podemos añadir comentarios.

Scheduler							
Name	Start Date	Start Time	Interval	On Event	Owner	Run Count	
::: Aviso de reinicio							
Aviso de reinicio	Sep/05/2015	startup	00:04:30	Aviso de reini...	admin	4917	
Bloqueo_Periodicos_Deportivos_L7	May/11/2014	01:00:00	1d 00:00:00	Bloqueo_Peri...	admin	16	
Bloqueo_Periodicos_Digitales2_L7	May/11/2014	01:10:00	1d 00:00:00	Bloqueo_Peri...	admin	16	
Bloqueo_Periodicos_Digitales_L7	May/11/2014	01:15:00	1d 00:00:00	Bloqueo_Peri...	admin	16	
Bloqueo_Redес_Sociales_L7	May/11/2014	01:20:00	1d 00:00:00	Bloqueo_Red...	admin	16	

Schedule <Aviso de reinicio>

Name:

Start Date:

Start Time:

Interval:

On Event:

Owner:

Policy: ☒ ftp ☒ reboot
☒ read ☒ write
☒ policy ☒ test
☒ password ☒ sniff
☒ sensitive

Run Count:

Next Run:

☒ enabled

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove

- **Scripting**

- Recomendaciones y buenas practicas.



• Scripting

○ Recomendaciones y buenas practicas.

✓ Introducir comentarios.

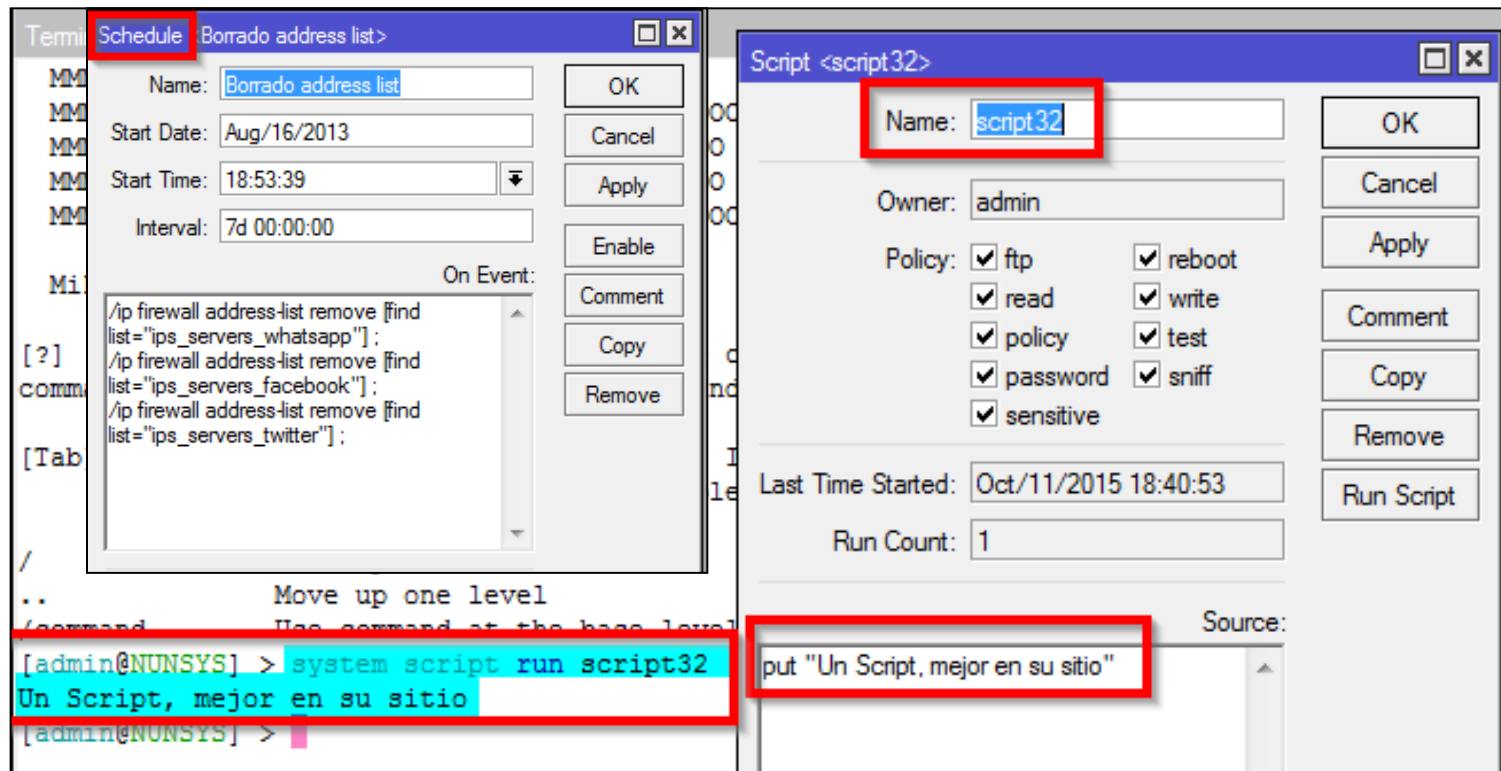
✓ Utilizaremos “#” para introducir comentarios de seguimiento.

```
# Asignar Queue simple desde DHCP Leases
# Definimos los limites de ancho de banda que hay que asignar.
global MaxUP 8M
global MaxDown 8M
# Definimos los limites de ancho de banda para los moviles.
global MaxMUP 1M
global MaxMDown 1M
#-----
# Definimos variables desde DHCP Leases
/ip dhcp-server lease
:foreach x in=[find] do={
:local leaseaddr ([get $x address]."/32")
:local leasemacaddr [get $x mac-address]
:local leasehostname [get $x host-name]
:local leasename [get $x comment]
:local queuecomment
:local leaseinqueue false
```

• Scripting

○ Recomendaciones y buenas practicas.

- ✓ No introducir código fuera de un script.
- ✓ No se puede probar/ejecutar desde consola.



• Scripting

○ Recomendaciones y buenas practicas.

- ✓ Mantener la línea de lectura.

```
#Cogemos la IP asignada, la mac y creamos una cola simple de seguimiento.
#indicamos tambien cuando debe de parar de escribir.
:while ($numero < $hosts) do={ \
:global IPARP [ip arp get value-name=address number=$numero address];
:global Interfaz [ip arp get value-name=inteface number=$numero interface];
:if ($Interfaz="bridge1") do={ \
:put $IPARP;
:put $Interfaz;
#Definimos el ancho de banda de cada cola, por si queremos limitar o simplemente
controlar.
/queue simple add max-limit=1000M/1000M name=$IPARP target=$IPARP total-queue=default
delay (10ms)
}
:global numero ($numero + 1);
```

• Scripting

○ Recomendaciones y buenas practicas.

- ✓ Introducir puntos de control visibles.

```
#Definimos el limite de transferencia
```

```
global limite (50);
```

```
#50% de la tasa de transferencia consumida.
```

```
global limite1 ($limite/2);
```

```
#80% de la tasa de transferencia consumida.
```

```
global limite2 ($limite-($limite/5));
```

```
put "Control 1 limites definidos-----"
```

```
put $limite0;
```

```
put $limite1;
```

```
put $limite2;
```

```
local RxEther1 ( 25
```

```
local TxEther1 ( 40
```

```
local TotEther1 ($RxEther1+$TxEther1);
```

```
#put "$TotEther1 GB";
```

```
/ Command Use Command at the base level
```

```
[admin@NUNSYS] > sys script run script32
```

```
Control 1 limites definidos-----
```

```
[admin@NUNSYS] >
```

*102

24*1

- **Scripting**

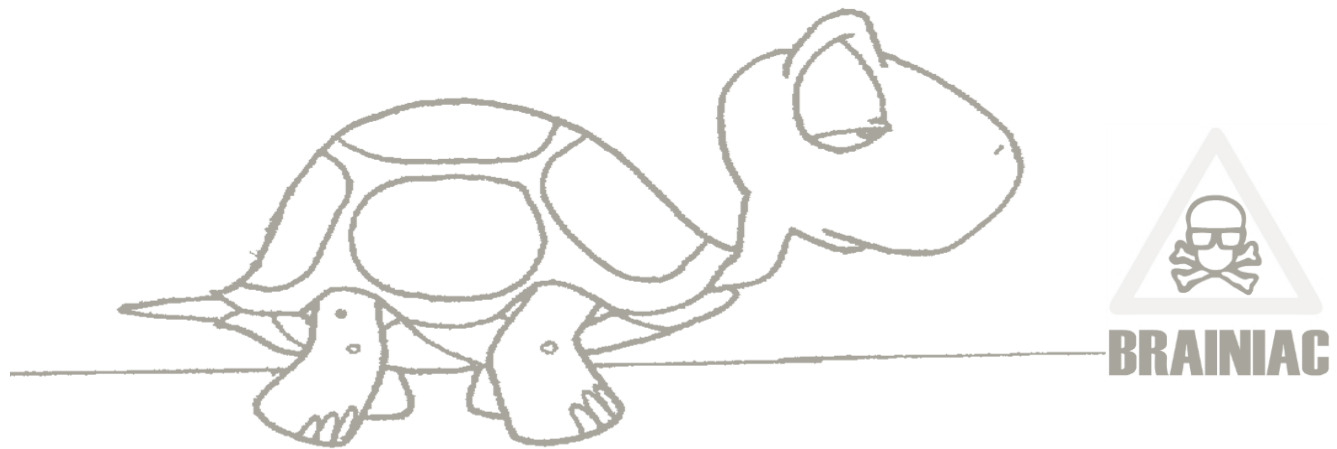
- **Recomendaciones y buenas practicas.**

✓ Paso a paso, un “delay” no esta de mas.

```
#Generamos un archivo txt, por cada muestra. Se toman 6 muestras.
#Muestra 1
/interface monitor-traffic ether1 file=MonitorTX;
delay (30ms)
local resultTX [/file get MonitorTX.txt contents]
# Analizar el resultado del archivo
local TotCarTX [:len $resultTX]
local startLocTX [:find $resultTX "tx-bits-per-second:"]
delay (30ms)
:set startLocTX ($startLocTX + 19)
delay (30ms)
local endLocTX ($startLocTX+11)
delay (30ms)
local TXX [:pick $resultTX $startLocTX $endLocTX]
delay (30ms)
local starKbpsTX [:find $TXX "kbps"]
delay (10ms)
local endKbpsTX ($starKbpsTX+4)
delay (10ms)
```


- Scripting

➤ “Very slowly”



- **Scripting**

- **“Very slowly”**

- ✓ Veamos paso a paso la ejecución de un **Script**.
 - ✓ **Escenario:**
 - ✓ Tenemos 2 WANs y realizamos balanceo y control de tráfico.
 - ✓ Nuestro ISP, nos asigna las IPs por DHCP y cambia la IP publica y la puerta de enlace, de forma aleatoria.
 - ✓ **Problema:**
 - ✓ Al cambiar las GWs, perdemos la conexión y dejan de funcionar las marcas de enrutado.
 - ✓ **Solución:**
 - ✓ Un Script que revise el DHCP-CLIENT y modifique las GWs si detecta cambios.

• Scripting

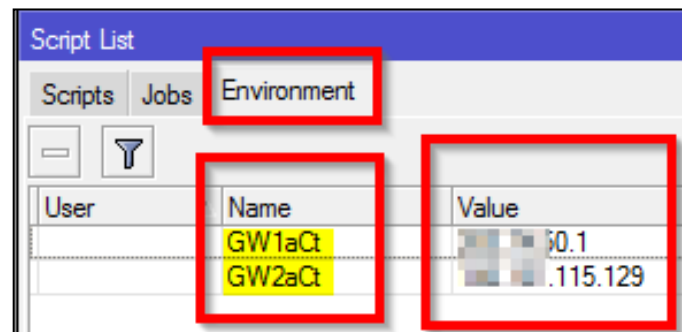
○ “Very slowly”

✓ Definimos variables del GW actual del DHCP-CLIENT.

global **GW1aCt** [ip dhcp-client get number=1 gateway]
delay (10ms);

global **GW2aCt** [ip dhcp-client get number=0 gateway]

```
[admin@NUNSYS] > ip dhcp-client print
Flags: X - disabled, I - invalid
# INTERFACE USE-PEER-DNS ADD-DEFAULT-ROUTE STATUS ADDRESS
0 ;;; Wan2 ether5 yes yes bound [redacted].186/25
1 ;;; Wan1 ether1 yes yes bound [redacted].17/25
[admin@NUNSYS] >
```



• Scripting

○ “Very slowly”

- ✓ Definimos variables del GW que hay en la tabla de enrutado.

global **GW1aNt** [/ip route get [find comment="Enruta Wan1"] gateway]
delay (10ms);

global **GW2aNt** [/ip route get [find comment="Enruta Wan2"] gateway]

The screenshot displays two windows from the Mikrotik WinBox interface. The 'Route List' window on the left shows a table of routes. The 'Script List' window on the right shows the 'Environment' tab with two variables defined: GW1aNt and GW2aNt.

Routes	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	.1 reachable ether1	1	Enruta_Wan1	
AS	0.0.0.0/0	.129 reachable ether5	1	Enruta_Wan2	
S	0.0.0.0/0	.129 reachable ether5	2	Enruta_Wan1	
S	0.0.0.0/0	.1 reachable ether1	2	Enruta_Wan2	
DS	0.0.0.0/0	5.129 reachable ether5	1		
DAS	0.0.0.0/0	.1 reachable ether1	0		

Name	Value
GW1aNt	.1
GW2aNt	15.129

• Scripting

○ “Very slowly”

- ✓ Comparamos **GW1** actual con **GW1** de tabla de enrutamiento.
- ✓ Si **no** coinciden, cambia las rutas y apunta resultado en log.
- ✓ Si coinciden, aplica el contrario (**else**) y no hagas nada.

```
if ($GW1aCt != $GW1aNt) do={\
/ip route set [find static=yes gateway=$GW1aNt] gateway=$GW1aCt;
log error message="Cambio rutas de salida WAN1 IP anterior= $GW1aNt \
IP actual= $GW1aCt";
} else={\
put "GW1 coinciden, no hago nada"}
```

- ✓ En este caso **GW1aCt** y **GW1aNt**, coinciden, por lo que no hace nada.

• Scripting

○ “Very slowly”

- ✓ Comparamos **GW2** actual con **GW2** de tabla de enrutamiento.
- ✓ Si **no** coinciden, cambia las rutas y apunta resultado en log.
- ✓ Si coinciden, aplica el contrario (**else**) y no hagas nada.

```
if ($GW2aCt != $GW2aNt) do={\
/ip route set [find static=yes gateway=$GW2aNt] gateway=$GW2aCt;
log error message="Cambio rutas de salida WAN2 IP anterior= $GW2aNt \
IP actual= $GW2aCt";
} else={\
put "GW2 coinciden, no hago nada"}
```

memory	system, info	route changed by admin
memory	system, info	route changed by admin
memory	script, error	Cambio rutas de salida WAN2 IP anterior= 115.12 IP actual= .115.129
memory	script, info	#####



4. Casos prácticos de aplicación

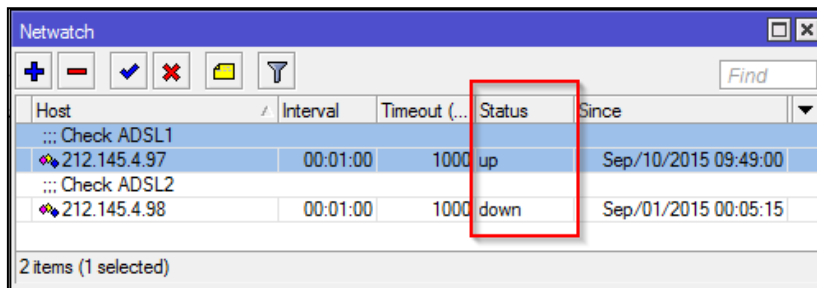
- **Casos prácticos.**

- ✓ Fail-over en balanceo de carga.
 - ✓ Definir un día del mes para ejecutar una acción.
 - ✓ Monitorizar una interfaz WAN
 - ✓ Modificar colas simples.
 - ✓ Comprobar túneles VPN
 - ✓ Reiniciar un equipo por carga de CPU.
 - ✓ Filtro de dominios por L7 con actualización centralizada.
 - ✓ Cambio de frecuencia por bajo SNR
-
- Scripts de ejemplo disponibles en:
 - ✓ <http://tecnologia.nunsys.com/>

• Casos prácticos.

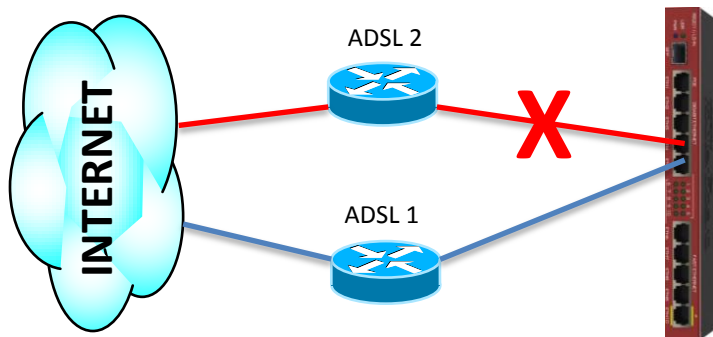
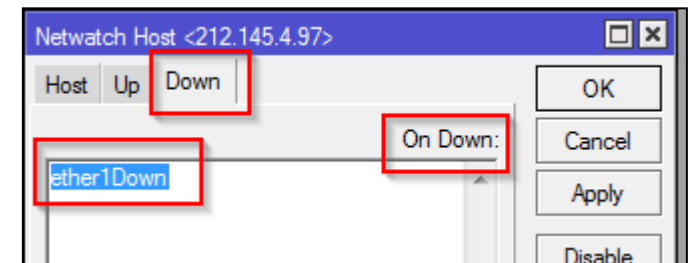
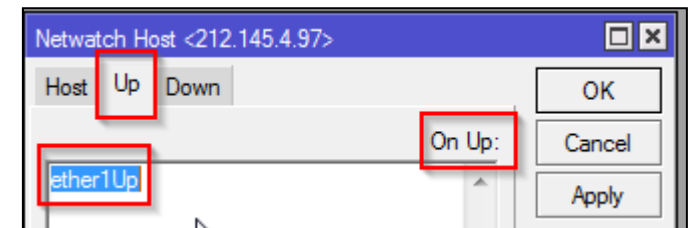
• Netwatch y fail-over en balanceo

- ✓ Es uno de los ejemplos más conocidos.
- ✓ Lanzado desde el netwatch, comprueba si el host definido tiene conectividad. De no ser así, desactiva las rutas de salida de ese interfaz, por lo que se activa el fail-over para las marcas de conexión.
- ✓ Cuando de nuevo hay conectividad, se realiza la comprobación a la inversa y se reactiva la ruta de salida.



Host	Interval	Timeout (...)	Status	Since
... Check ADSL1				
212.145.4.97	00:01:00	1000	up	Sep/10/2015 09:49:00
... Check ADSL2				
212.145.4.98	00:01:00	1000	down	Sep/01/2015 00:05:15

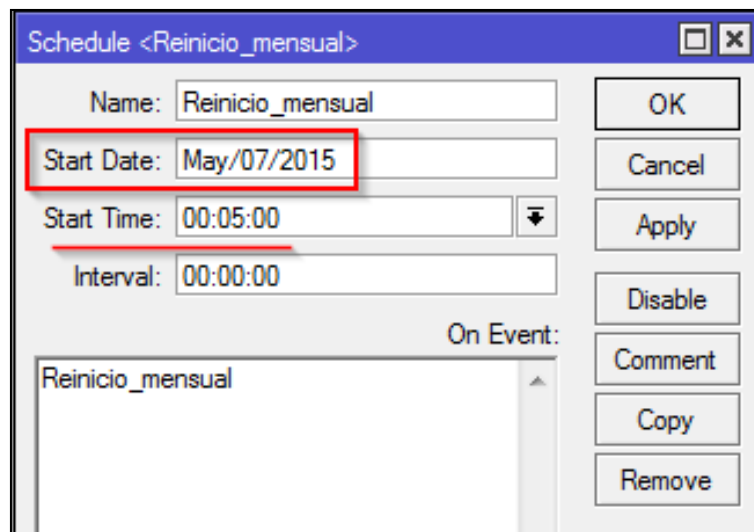
2 items (1 selected)



• Casos prácticos.

• Programar un día en concreto para realizar una acción

- ✓ Necesitamos reiniciar contadores el día 15 de cada mes.
- ✓ Este **Script** modifica el Scheduler, rectificando el día en el que debe volver a producirse el evento.
- ✓ Observar que no existe intervalo de repetición, es el Script el que modifica la fecha según los parámetros configurados.



Schedule <Reinicio_mensual>

Name: Reinicio_mensual

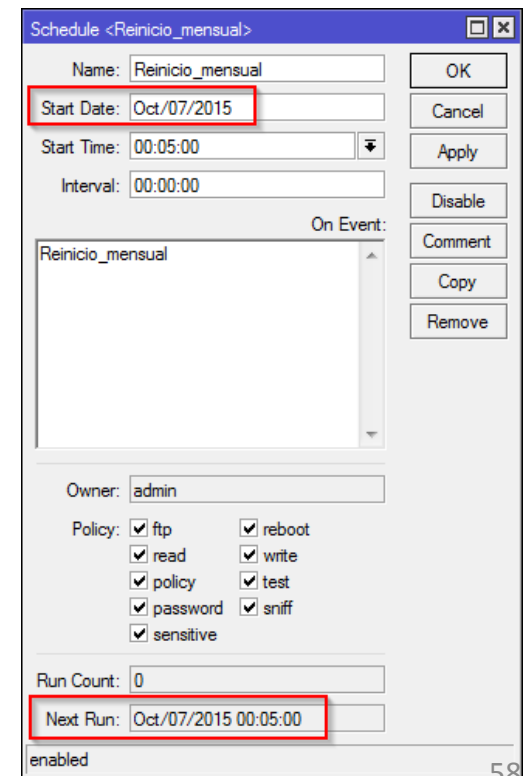
Start Date: May/07/2015

Start Time: 00:05:00

Interval: 00:00:00

On Event: Reinicio_mensual

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove



Schedule <Reinicio_mensual>

Name: Reinicio_mensual

Start Date: Oct/07/2015

Start Time: 00:05:00

Interval: 00:00:00

On Event: Reinicio_mensual

Owner: admin

Policy: ☒ ftp ☒ reboot
☒ read ☒ write
☒ policy ☒ test
☒ password ☒ sniff
☒ sensitive

Run Count: 0

Next Run: Oct/07/2015 00:05:00

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove

enabled

- ✓ Tenemos problemas con nuestra VoIP y necesitamos saber si existen microcortes, por lo que necesitamos monitorizar nuestra WAN.
- ✓ Una forma seria lanzando un ping continuo y esperar a ver si se pierde alguna traza ICMP.
- ✓ Este script, lanza un ping contra una IP (un DNS o servidor de VoIP por ejemplo) y si pierde mas paquetes que el limite definido, nos envía un correo con los resultados, en caso contrario, no hace nada.

Porcentaje de perdida: 10%
Porcentaje aceptable, no hago nada

Porcentaje de perdida: 40%

El porcentaje supera los limites, envio correo de aviso

Resultado monitorizacion Ether1 contra IP: 212.166.210.3 Pings lanzados: 10, Pings devueltos OK= 6, Pings perdidos= 4, Porcentaje de perdida= 40%

- **Casos prácticos.**

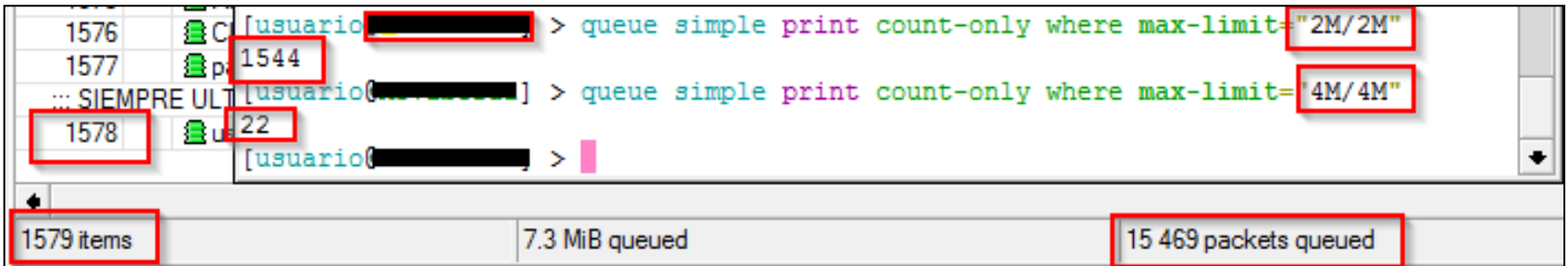
- **Modificar colas simples**

- ✓ Imaginemos este escenario:

- ✓ 1.579 colas simples

- ✓ 3 Perfiles 10Mb, 4Mb y 2Mb

- ✓ Queremos cambiar la velocidad de los que tienen 2Mb, pero tenemos nada menos que 1.544 colas



```
[usuario@Mikrotik] > queue simple print count-only where max-limit=2M/2M
1544
[usuario@Mikrotik] > queue simple print count-only where max-limit=4M/4M
22
[usuario@Mikrotik] >
```

1579 items 7.3 MiB queued 15 469 packets queued

- **Casos prácticos.**

- **Modificar colas simples**

- ✓ Cambiar las colas una a una, seria una tarea imposible, pero en un script, con una simple línea, lo tendríamos resuelto:

/queue simple set max-limit="9500k/9500k" burst-limit=9900k/9900k burst-threshold=9200k/9200k burst-time=200/300 [find max-limit="2M/2M"]

- ✓ Si queremos restaurar la velocidad anterior, la operación seria inversa:

/queue simple set max-limit="2M/2M" burst-limit=2200k/2200k burst-threshold=1900k/1900k burst-time=200/300 [find max-limit="9500k/9500k"]

- ✓ Programándolo desde el scheduler, podemos modificar el ancho de banda de los clientes, aun cuando las colas sean dinámicas generadas por el host-spot.

• Casos prácticos.

• Comprobar túneles VPN

- ✓ Tenemos una VPN, en el que el túnel no cursa tráfico, pero pese a ello, aparece el flag R de “Runing”, por lo que no se restaura la conexión.
- ✓ Este script, comprueba si existe conectividad realizando un ping al otro extremo del túnel y lo restaura si hay 100% de perdida

PPP

Interface PPPoE Servers Secrets Profiles Active Connections

Name	Service	Caller ID	Encoding	Address	Uptime
OVPN	ovpn	83.56.228.43	AES-256-CBC/SHA1	2.2.2.2	1d 20:33:59

Terminal

Tunel activo, no hago nada

```
[admin@Comedor] > system script run "Comprobar Tunel VPN"
```

Flags: R - radius

#	NAME	SERVICE	CALLER-ID	ADDRESS	UPTIME	ENCODING
0	OVPN	ovpn	83.56.228.43	2.2.2.2	1d20h...	AES-256-CBC/SHA1

Numero 0
IP 2.2.2.2

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	2.2.2.2	56	64	111ms	
1	2.2.2.2	56	64	110ms	
2	2.2.2.2	56	64	110ms	
3	2.2.2.2	56	64	111ms	
4	2.2.2.2	56	64	111ms	
5	2.2.2.2	56	64	109ms	
6	2.2.2.2	56	64	110ms	
7	2.2.2.2	56	64	111ms	
8	2.2.2.2	56	64	110ms	
9	2.2.2.2	56	64	111ms	

sent=10 received=10 packet-loss=0% min-rtt=109ms avg-rtt=110ms max-rtt=111ms

Tunel activo, no hago nada

```
[admin@Nunsys] >
```

- **Casos prácticos.**

- **Reiniciar el equipo por sobrecarga de CPU.**

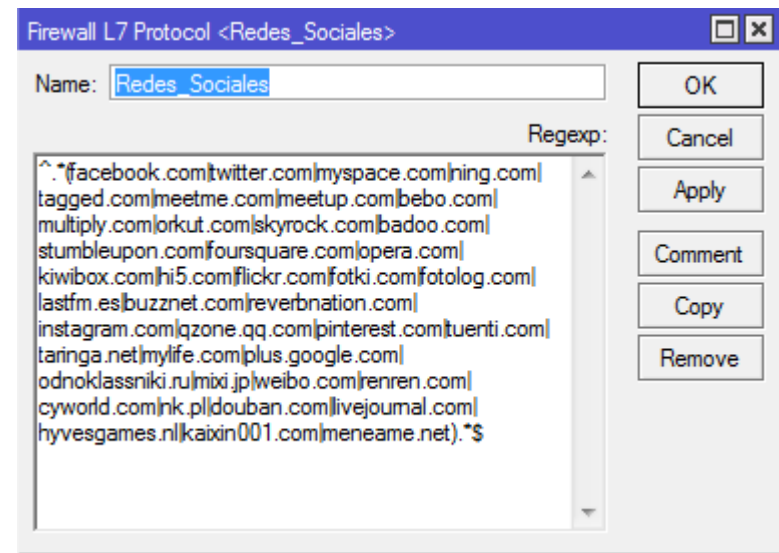
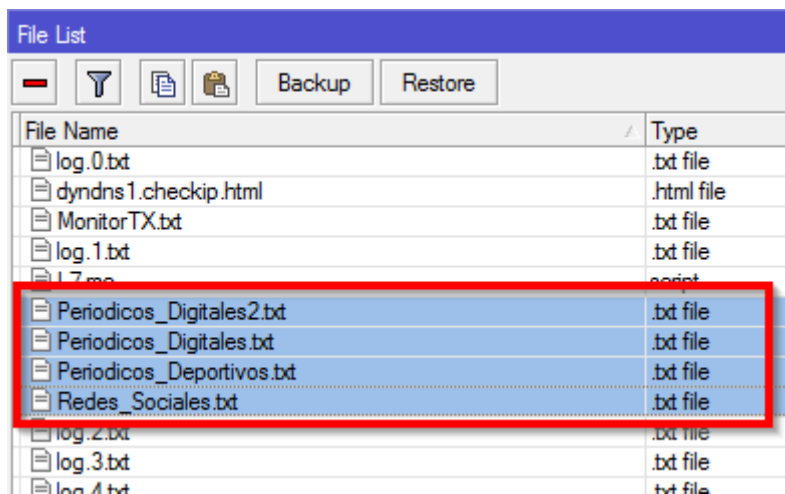
- ✓ En ocasiones, por diversos motivos, hay procesos que provocan inestabilidad y generan consumo incontrolado de CPU, ralentizando los procesos e impidiendo incluso que tengamos acceso al equipo.
- ✓ Este script, toma muestras del porcentaje de proceso de CPU y si durante 1 minuto, es superior al 97%, reinicia el equipo y genera un log.

```
/ Command: Use Command at the base level  
[admin@Nunsys] > system script run "Reinicio equipo por carga CPU"  
Muestra 1, CPU al 5%  
Muestra 2, CPU al 4%  
Muestra 3, CPU al 6%  
Muestra 4, CPU al 9%  
Media de uso de la CPU 6%  
[admin@Nunsys] > █
```

• Casos prácticos.

• Filtro de dominios por L7 con actualización centralizada.

- ✓ Necesitamos filtrar dominios por L7, pero queremos actualizar muchos equipos a la vez.
- ✓ Con este script, el equipo descarga de otro equipo por FTP, varios archivos de texto, con los dominios actualizados, y actualiza el Regexp del archivo L7 correspondiente.



• Casos prácticos.

• Cambio de frecuencia por bajo SNR.

- ✓ Tenemos valores bajos de SNR en nuestros clientes.
- ✓ Con este script, se calcula la media de SNR de los clientes conectados.
- ✓ Si valor por debajo del umbral, cambia la frecuencia.
- ✓ Pasados 3 minutos vuelve a calcular el SNR.
- ✓ Si mejora, se mantiene, sino, vuelve a cambiar de frecuencia.
- ✓ Marcamos un SNR mínimo de 30dB

ARP	Mode	Band	Channel	Frequency (MHz)
34:C0	enabled	ap bri...	2GHz-B/...	20MHz

AP Client <04:E6:76:21:EB:26>	
General	802.1x
Last Activity: 0.060 s	
Tx/Rx Signal Strength: -70 dBm	
Tx/Rx Signal Strength Ch0: -81 dBm	
Tx/Rx Signal Strength Ch1: -70 dBm	
Tx/Rx Signal Strength Ch2:	
Signal To Noise: 25 dB	
Tx/Rx CCQ: 42 %	

ARP	Mode	Band	Channel	Frequency (MHz)
34:C0	enabled	ap bri...	2GHz-B/...	20MHz

AP Client <04:E6:76:21:EB:26>	
General	802.1x
Last Activity: 2.450 s	
Tx/Rx Signal Strength: -64 dBm	
Tx/Rx Signal Strength Ch0: -82 dBm	
Tx/Rx Signal Strength Ch1: -64 dBm	
Tx/Rx Signal Strength Ch2:	
Signal To Noise: 31 dB	
Tx/Rx CCQ: 51 %	



5. Preguntas

Tu socio tecnológico

nunsys®

Gracias por su atención!

902 88 16 26
nunsys.com

VALENCIA · ALICANTE · CASTELLÓN · MURCIA · PALMA DE MALLORCA · VALLADOLID · SEVILLA