



guía de uso

Versión 1.1



Índice

Introducción.....	6
Acerca de Tuna.....	7
Bienvenido.....	7
¿Qué es Tuna?.....	7
Para configurar Tuna.....	8
Uso de la interfaz de línea de comandos CLI.....	9
Acceso.....	9
Modos de comandos.....	9
Ayuda.....	10
Autocompletar.....	11
Abreviaciones.....	12
Histórico.....	12
Guía de comandos.....	13
Convenciones de notación.....	14
Sintaxis.....	14
Terminología.....	14
Comandos básicos	16
exit	16
end	16
Comandos en modo config	17
hostname	17
enable secret	17
login password	18
ip name-server	18
Comandos en modo priv	19
configure terminal (cambio de modo).....	19
write memory	19
show configuration	20
show startup-configuration	21
erase startup-configuration	21
disable (cambio de modo).....	21
logout	21
Comandos en modo exec	22
enable (cambio de modo).....	22
Módulo ifether	23
Comandos en modo config-if	23
description	23
ip address	23
ip address secondary	24
shutdown	24
uplink	25
Comandos en modo config	25
interface ethernet (cambio de modo).....	25

ip route	26
Módulo aaa	27
Comandos en modo config	27
aaa authentication server	27
aaa accounting server	27
aaa radius authentication.....	27
aaa radius secondary-authentication.....	28
aaa radius accounting	29
aaa radius secondary-accounting	29
aaa idle-timeout	30
aaa default-bandwidth	30
Módulo dhcp	32
Comandos en modo config-dhcp-pool	32
range	32
router	32
domain-name	33
name-server	33
lease	33
max-lease	34
fixed-ip	34
exclude	35
Comandos en modo config-if	35
ip dhcp allow-unknown-clients	36
ip dhcp pool (cambio de modo).....	36
Comandos en modo config	36
ip dhcp domain-name	36
ip dhcp lease	37
ip dhcp max-lease	37
service dhcp	38
Módulo dns-cache	39
Comandos en modo config	39
service dns-cache	39
dns forwarder	39
Módulo enable-aaa	41
Comandos en modo config	41
service aaa	41
Módulo filters	42
Comandos en modo config-filter-group	42
description	42
filter	42
Comandos en modo config-if	44
filter	44
Comandos en modo config	45
filter group (cambio de modo).....	45
Módulo shaping	47
Comandos en modo config-shaping-class	47

description	47
match	47
Comandos en modo config-if	49
shaping class	49
bandwidth	50
Comandos en modo config	51
shaping class (cambio de modo).....	51
Módulo graph	52
Comandos en modo config	52
graph	52
Módulo http	53
Comandos en modo config	53
service http	53
Módulo nat	54
Comandos en modo config-if	54
nat-rule	54
Comandos en modo config	54
nat-rule	54
Módulo snmp	56
Comandos en modo config	56
service snmp	56
snmp community	56
Módulo sshd	57
Comandos en modo config	57
service sshd	57
ssh 2 authorized_key	57
ssh authorized_key	58
ssh port	58
ssh protocol	58
ssh rsa_auth	59
ssh password_auth	59
ssh pubkey_auth	59
Módulo telnet	61
Comandos en modo config	61
service telnet	61
telnet no-password	61
telnet secret	61
Apéndices.....	63
Preguntas frecuentes.....	64
General.....	64
¿Por qué me puede interesar Tuna?.....	64
¿Qué es QoS?.....	64
¿Qué significa “gestionar acceso a servicios”?.....	64
¿Qué significa “tarificar” el uso de Internet?.....	64
Aplicabilidad.....	65
¿En qué es útil Tuna para accesos inalámbricos a Internet?.....	65

¿Por qué Tuna resulta útil para núcleos empresariales, como ser polígonos industriales, parques tecnológicos, centros comerciales o poblaciones alejadas?.....	65
¿Por qué Tuna resulta útil para empresas que utilizan VoIP?.....	65
¿Por qué Tuna resulta útil para ASPs?.....	65
¿En qué mercados se ha detectado potencialidad de usar Tuna?.....	65
¿De qué formas permite Tuna cobrar por el acceso a Internet?.....	66
¿Cómo pueden los usuarios cambiar dinámicamente su QoS?.....	66
Generalidades técnicas.....	66
¿Es posible integrar Tuna a otros sistemas de gestión?.....	66
¿En qué lugar de la red se coloca?.....	66
¿De qué maneras puede clasificar el tráfico Tuna?.....	67
¿Cómo autentica los usuarios Tuna?.....	67
¿Necesito un servidor RADIUS para poder manejar usuarios?.....	67
¿Qué requerimientos debe cumplir una red y sus equipos para conectarse a Tuna?.....	67
Información comercial.....	68
¿De qué formas se comercializa Tuna?.....	68
¿Cómo se licencia?.....	68
¿Cuál es la visión a futuro de la tecnología?.....	68
Especificaciones técnicas.....	69
Clasificación de Tráfico por Flujo.....	69
Implementación de Calidad de Servicio (QoS).....	69
Seguridad en el Tráfico.....	69
Autenticación, Tarificación y Monitoreo.....	69
Autorización.....	69
Miscelaneo.....	70
Índice alfabético.....	71

Introducción



Acerca de Tuna

Bienvenido

Bienvenido a la guía de uso de **Tuna**. Este documento está dirigido a los administradores y operadores de red que deberán instalar, configurar y mantener el sistema. El objetivo es asistir a estas personas, detallando las principales técnicas de configuración y mantenimiento.

¿Qué es Tuna?

Tuna es una tecnología diseñada por **netlabs** para poder hacer un uso más eficiente de las redes IP y en particular de los enlaces a Internet. La misma se aplica en algunas soluciones a medida y otras estándar, que componen la línea de **Productos Tuna**. En todos los casos, se instala en un equipo al cual nos referiremos de aquí en adelante como “equipo **Tuna**”, o simplemente “**Tuna**”, que se incorpora a la red objetivo.

Sus principales aplicaciones incluyen:

Control de usuarios (AAA). El sistema incorpora un servidor RADIUS, que permite realizar AAA (autorización, autenticación y contabilización) de usuarios. Gracias a esta utilidad, es posible gestionar los usuarios que acceden a servicios de la red y obtener información acerca de su uso.

Gestión de calidad de servicio (QoS). Quality of Service, calidad de servicio, refiere a la calidad con la que se accede a Internet. Los principales factores involucrados a esta calidad son el máximo ancho de banda utilizable (MIR) y el mínimo garantizado (CIR). Tuna permite definir estos límites para usuarios, aplicaciones y/o equipos.

Firewall. Barrera de seguridad configurable, fundamental para cualquier red conectada a Internet.

Funcionalidades de uso general. **Tuna** incluye funcionalidades de DHCP, caché de DNS, NAT y SNMP, entre otros.

Al final de esta guía de uso, se incluyen apéndices con las preguntas más frecuentemente realizadas acerca de **Tuna**, tanto a nivel de usabilidad como comercial, así como un resumen de sus funcionalidades a nivel técnico.

Para configurar Tuna

El sistema cuenta con una interfaz de línea de comandos (CLI), a través de la cual se realiza la instalación y configuración inicial del sistema y, por otra parte, una interfaz web de administración de usuarios, las cuales se describirán en detalle en los próximos capítulos.

Uso de la interfaz de línea de comandos CLI

Acceso

La Interfaz de línea de comandos, o CLI por su sigla en inglés, es accesible a través de telnet o ssh desde cualquier terminal con acceso al **Tuna**.

```
[foo home]$telnet 192.168.100.71
Connected to 192.168.100.71.
Escape character is '^]'.
tuna>_
```

El prompt indica al inicio el nombre del equipo, que por defecto es "tuna", y luego el símbolo ">", que indica que se encuentra en modo usuario.

Modos de comandos

Los comandos accesibles dependen del modo en que el CLI se encuentre. Al iniciar una sesión en **Tuna**, se ingresa en modo usuario (también conocido como modo EXEC).

Para poder realizar cambios sobre la configuración de Tuna, es necesario ingresar al modo privilegiado, lo cual logra con el comando **enable**.

```
tuna>enable
tuna# _
```

El símbolo "#" indica que se encuentra en modo privilegiado. Los comandos disponibles varían al cambiar de modo.

Cuando se ingresa a un nuevo modo, éste se mostrará entre paréntesis – "(" y ")" – a continuación del nombre del equipo.

```
tuna#configure terminal
tuna(config)# _
```

Los modos de configuración soportados por **Tuna** se muestran a continuación.

Modo	Ejemplo de acceso
Privilegiado (priv)	tuna> enable tuna# _
Configuración (config)	tuna# configure terminal tuna (config) # _
Configuración de filtros (config-flt)	tuna (config) # filter group 1 tuna (config-flt) # _
Configuración de interfaz (config-if)	tuna (config) # interface ethernet 1 tuna (config-if) # _
Configuración de pool DHCP (config-dhcp-pool)	tuna (config-if) # ip dhcp pool pool1 tuna (config-dhcp-pool) # _
Configuración de grupo de filtros (config-filter-group)	tuna (config) # filter group foobar tuna (config-flt) # _
Configuración de clase de calidad de servicio (config-shaping-class)	tuna (config) # shaping class 1 tuna (config-sh) # _

Ayuda

Presionando el símbolo de interrogación (?) se listan los comandos accesibles en el momento.

```
tuna# ?
configure  Entra a modos de configuración
*disable   Sale del modo privilegiado
*logout    Sale del modo de ejecución y termina la sesión
show       Muestra la configuración del sistema
write      Guarda la configuración del sistema
```

La lista de comandos puede variar de este ejemplo, dependiendo de la versión de Tuna que se esté ejecutando.

En la ayuda, aparecen con asterisco (*) las expresiones utilizadas para finalizar un comando.

Es posible obtener ayuda ya iniciado un comando, como se muestra en el ejemplo:

```
tuna (config) #default ?
aaa        Establece valores por defecto de AAA del módulo RADIUS
```

```
*hostname  Asigna nombre por defecto del equipo
ip         Asigna valores de interfaces TCP/IP y otros parámetros
ntop      Asigna configuraciones por defecto
ssh       Asigna configuraciones por defecto del servidor ssh
tuna(config)#default _
```

En la ayuda, las expresiones utilizan las siguientes notaciones:

- Entre aspas ("<" y ">") representan variables que deben ser sustituidas por un valor. Ejemplo:

```
tuna(config)#hostname ?
*<name> Nombre del equipo
```

- Entre paréntesis ("(" y ")") las expresiones que deben ser seleccionadas de una lista. Ejemplo:

```
tuna(config)#ip dhcp lease 35 ?
*(time_unit) Unidad de tiempo
```

Esto significa que el siguiente parámetro debe ser un literal de una lista de valores predeterminada (en este caso **days**, **hours**, **mins** o **secs**).

En la sección [Autocompletar](#) se describirá cómo hacer para saber los valores válidos al momento de ingresar el parámetro.

- Sin símbolos alrededor aparecen los literales.

Es posible también, obtener la ayuda de todos los comandos que comienzan con una expresión parcial, presionando ? luego de ella.

```
tuna(config)#i?
interface  Configura interfaces de red
ip         Configura rutas de interfaces TCP/IP y otros parámetros
tuna(config)#i_
```

Autocompletar

El CLI de Tuna incluye la funcionalidad de autocompletado de expresiones. En caso de no recordar el nombre completo de un comando, se puede presionar <TAB> para completar una entrada parcial.

```
tuna#con<TAB>
tuna#configure _
```

En caso de existir más de un comando con la entrada parcial tecleada, el sistema hará un *bip*. Luego, si se presiona nuevamente <TAB>, el sistema mostrará la lista de posibles comandos que completan la expresión.

```
tuna (config) #e<TAB><TAB>
enable end exit
tuna (config) #e_
```

Este método de presionar dos veces la tecla <TAB> se puede utilizar también para saber la lista de todos los comandos válidos en un momento dado.

```
tuna (config) #<TAB><TAB>
aaa enable filter ip ntop snmp
default end hostname nat-rule service ssh
dns exit interface no shaping telnet
tuna (config) # _
```

Finalmente, cuando un comando incluye una lista de selección, también es utilizable la tecla <TAB> para saber las opciones válidas.

```
tuna (config) #ip dhcp lease 35 <TAB><TAB>
days hours mins secs
tuna (config) #ip dhcp lease 35 _
```

Abreviaciones

Al teclear comandos, es posible realizar abreviaciones de los literales. Esto significa que en caso que no exista ningún otro comando en el modo actual que comience con las mismas letras, no es necesario tipear el resto del comando.

Por ejemplo, escribir:

```
tuna#c t
```

Es equivalente a escribir:

```
tuna#configure terminal
```

Histórico

En todo momento es posible acceder a un histórico de los comandos que se han entrado. Esta funcionalidad resulta muy práctica para acceder a comandos complejos que se hayan teclado recientemente. Es accesible con el uso de las flechitas (<ARRIBA>, <ABAJO>).

Guía de comandos



Convenciones de notación

Sintaxis

Para describir las convenciones de notación para la sintaxis utilizadas en esta guía de uso, nos basaremos en el siguiente ejemplo, del comand **nat-rule**:

```

nat-rule snat <name> <prefix> <snat-src>
nat-rule dnat <name> <dnat-src> <dnat-dst>
nat-rule dnat <name> ( tcp | udp ) <dnat-src> <sport> <dnat-dst>
    [ <dnatport> ]

no nat-rule %nat-rules
no nat-rule

```

- Cada renglón representa una posible expresión válida del comando.
- Los ítems en **negritas**, como **nat-rule**, **dnat**, o **tcp**, representan literales que deben ser tecleados exactamente como aparecen en la sintaxis, a menos que se pueda utilizar una abreviación, como se explicó en el capítulo anterior.
- Los ítems ente aspas (“<” y “>”), como <name>, representan valores que deben ser ingresados por el usuario.
- Cuando se muestra una lista de ítems entre paréntesis (“(“ y “)”), separados por barras (“|”), significa que el usuario debe elegir una de estas opciones.
- Los ítems opcionales aparecen entre corchetes (“[“ y “]”).
- Los ítems que deben ser seleccionados de una lista dinámica de valores válidos, aparecen precedidos por el símbolo de porcentaje (“%”). En el ejemplo, %nat-rule indica que el usuario debe elegir una regla antes establecida. Para saber las reglas disponibles para ingresar en cualquier momento, alcanza con presionar 2 veces la tecla <TAB>.

Terminología

A lo largo de toda la guía de uso, se utilizarán algunos términos que vale la pena precisar:

- **prefijo**: Sub-red, expresada en formato IP/bits. Por ejemplo, 192.168.0.0/24 refiere a la subred con la IP 192.168.0.0, máscara 255.255.255.0.

- **cidr**: Dirección IP y subred a la que pertenece, en formato IP/bits. Por ejemplo, 192.168.0.2/24 refiere a la IP 192.168.0.2, y subred con IP subred IP 192.168.0.0 y máscara 255.255.255.0.

Comandos básicos

exit

Sintaxis

```
exit
```

Descripción

Sale del modo de configuración actual, regresando al modo padre.

Ejemplo

En el siguiente ejemplo, se define una clase de servicio y se regresa al modo de configuración:

```
tuna#configure terminal
tuna(config)#shaping class 1
tuna(config-sh)#description correo
tuna(config-sh)#exit
tuna(config)#
```

end

Sintaxis

```
end
```

Descripción

Sale del modo de comandos actual, regresando al modo de ejecución. En terminales unix, es equivalente a presionar las teclas <Ctrl>Z.

Ejemplo

En el siguiente ejemplo, se define una clase de servicio y se regresa al modo de ejecución :

```
tuna#configure terminal
tuna(config)#shaping class 1
tuna(config-sh)#description correo
tuna(config-sh)#end
tuna# _
```

Comandos en modo config

hostname

Sintaxis

```
hostname <name>  
default hostname
```

Parámetros

<name> Nombre del equipo (Ej: foo.com)

Valor por defecto

tuna

Descripción

Establece el nombre del equipo Tuna.

Modifica el nombre del equipo, que aparece en el prompt. Con el comando **default hostname** es posible regresar el nombre del equipo a su valor por defecto.

Ejemplo

Cambia el nombre del equipo a “tuna2”:

```
tuna (config) #hostname tuna2  
tuna2 (config) # _
```

enable secret

Sintaxis

```
enable secret ( 5 <encrypted-secret> | <secret> )  
no enable secret
```

Parámetros

5 Indica que se ingresa una contraseña ya encriptada.
<encrypted-secret> Contraseña ya encriptada.
<secret> Contraseña sin encriptar. Acepta todo tipo de caracteres.

Descripción

El comando **enable secret** permite establecer la contraseña a ser solicitada al intentar ingresar al modo privilegiado (comando **enable**, estando en modo usuario). Esta contraseña puede ingresarse ya encriptada, tipeando el literal **5** antes de la misma.

En caso de desear que Tuna no solicite más contraseña para ingresar al modo privilegiado, se debe utilizar el comando **no enable secret**.

Ejemplo

Para establecer que Tuna solicite contraseña para ingresar a modo privilegiado, y que esta sea *foobar*:

```
tuna(config)#enable secret foobar
```

Para establecer que Tuna solicite contraseña para ingresar a modo privilegiado, y tipearla en modo encriptado:

```
tuna(config)#enable secret 5 a1.*X!y
```

Para establecer que Tuna no solicite contraseña para ingresar a modo privilegiado:

```
tuna(config)#no enable secret
```

login password

Sintaxis

```
login password ( 5 <encrypted-secret> | <secret> )
```

Parámetros

- | | |
|--------------------|---|
| 5 | Indica que se ingresa una contraseña ya encriptada. |
| <encrypted-secret> | Contraseña ya encriptada. |
| <secret> | Contraseña sin encriptar. Acepta todo tipo de caracteres. |

Descripción

Establece la contraseña de autenticación del usuario administrador. Esta es la contraseña que se deberá utilizar para loguearse en **Tuna** como administrador vía ssh. Esta contraseña puede ingresarse ya encriptada, tipeando el literal **5** antes de la misma.

Ejemplo

Para establecer como contraseña de acceso a **Tuna** “foobar.*X!y”:

```
tuna(config)#login password foobar.*X!y
```

ip name-server

Sintaxis

```
ip name-server <ip>
no ip name-server %nameserver
```

Parámetros

- | | |
|------|--|
| <ip> | Número de IP del servidor dns a agregar. |
|------|--|

Descripción

El comando **ip name-server** permite establecer los servidores de dns a utilizar, a través de sus direcciones IP. Estas direcciones serán utilizadas por **Tuna** en el orden que se han establecido.

Con el comando **no ip name-server** es posible eliminar una dirección de servidor dns previamente establecida.

Ejemplo

Pasa establecer como dns primario 207.3.121.254 y como secundario 207.3.121.253:

```
tuna(config)#ip name-server 207.3.121.254  
tuna(config)#ip name-server 207.3.121.253
```

Comandos en modo priv

configure terminal (cambio de modo)

Sintaxis

```
configure terminal
```

Descripción

Entra al modo de configuración (config). Los comandos en este modo afectan la configuración global del sistema.

Ejemplo

Para ingresar al modo config y ver los comandos disponibles:

```
tuna>enable  
tuna#configure terminal  
tuna(config)# ?
```

write memory

Sintaxis

```
write memory
```

Descripción

Guarda la configuración del sistema en memoria permanente. Una vez ejecutado este comando, la configuración actual del sistema será la utilizada cada vez que se inicie el equipo **Tuna**.

Para ver la configuración inicial establecida, ver comando **show startup-configuration**.

show configuration

Sintaxis

```
show configuration
```

Descripción

Muestra la configuración completa del sistema. El formato en que se muestra, corresponde a los comandos necesarios para llegar a dicha configuración. De ese modo, es posible copiar desde la terminal la salida de este comando, para reproducir esta configuración en otro momento u otro **Tuna**.

Ejemplo

Para ver la configuración actual del sistema:

```
tuna>enable
tuna#show configuration
ip name-server 207.3.121.254
ip name-server 207.3.121.253
nat-rule snat salida 192.168.200.0/24 192.168.100.91
shaping class 1
    match include ip 192.168.200.10 any
no service aaa
aaa authentication server radius
aaa accounting server radius
aaa radius authentication ip 127.0.0.1 port 1812 secret secret
aaa radius accounting ip 127.0.0.1 port 1813 secret secret
aaa idle-timeout 1800
filter group bloqueo_telnet
login password 5 $1$GnTCZsIT$FRP9/8ksOaft805MjHDEd/
interface ethernet 0
    description Internet
    ip address 192.168.100.91/24
    uplink
    filter bloqueo_telnet in
    bandwidth 100 mbps
    shaping class 1 64/256 kbps
    nat-rule salida
interface ethernet 1
    description Red Interna de Servidores
    ip address 192.168.200.1/24
    bandwidth 100 mbps
    shaping class 1 reversed 64/256 kbps
ip route 0.0.0.0/0 192.168.100.1
end
tuna#
```

show startup-configuration

Sintaxis

```
show startup-configuration
```

Descripción

Muestra la configuración del sistema, en su estado inicial. Esta será la configuración que utilizará el equipo **Tuna** la próxima vez que se inicie.

erase startup-configuration

Sintaxis

```
erase startup-configuration
```

Descripción

Elimina la configuración inicial del sistema. Ésta puede ser re-establecida a partir de la configuración actual, con el comando **write-memory**.

Ejemplo

Elimino la configuración inicial del sistema:

```
tuna#erase startup-configuration
tuna#show startup-configuration
tuna#
```

disable (cambio de modo)

Sintaxis

```
disable
```

Descripción

Sal del modo privilegiado, previamente accedido con el comando **enable**.

Ejemplo

Ingreso y salgo del modo privilegiado:

```
tuna>enable
tuna#disable
tuna>
```

logout

Sintaxis

logout**Descripción**

Sale del modo de ejecución y termina la sesión.

Comandos en modo exec

enable (cambio de modo)

Sintaxis**enable****Descripción**

Entra al modo privilegiado. Desde este modo, es posible realizar tareas de mantenimiento del sistema, así como acceder al modo **config**.

Debido a que muchos comandos de este modo establecen parámetros de operación, se aconseja proteger el acceso a este modo con contraseña, utilizando el comando **enable password**. Cuando **Tuna** solicita esta contraseña, no es mostrada en pantalla por motivos de seguridad. Esta contraseña, como todas las del sistema, es sensible a las mayúsculas.

Ejemplo

Ingreso al modo privilegiado, con contraseña:

```
tuna>enable  
Password:  
tuna#
```

Módulo ifether

Comandos en modo config-if

description

Sintaxis

```
description <desc>  
no description
```

Parámetros

<desc> Nombre descriptivo de la interfaz.

Descripción

El comando **description** asigna una descripción a la interfaz que se está configurando. Con el comando **no description** se elimina ésta.

Ejemplo

Para describir la interfaz ethernet 0 como “Salida a Internet”:

```
tuna(config)#interface ethernet 0  
tuna(config-if)#description Salida a Internet
```

ip address

Sintaxis

```
ip address <prefix>  
no ip address
```

Parámetros

<prefix> Dirección ip y subred correspondiente a la interfaz, en formato cidr.

Descripción

Establece parámetros IP.

El comando **ip address** es utilizado para asignar una dirección IP primaria y su correspondiente subred a la interfaz. Con el comando **no ip address** se elimina este dato.

Ejemplo

Para asignar a la interfaz 1 la IP 192.168.0.2 en la subred con máscara 255.255.255.0:

```
tuna(config)#interface ethernet 1  
tuna(config-if)#ip address 192.168.0.2/24
```

ip address secondary

Sintaxis

```
ip address <prefix> secondary  
no ip address %ip_sec secondary  
no ip address secondary
```

Parámetros

<prefix> Dirección ip y subred secundarias correspondientes a la interfaz, en formato cidr.

Descripción

El comando **ip address secondary** es utilizado para asignar una o varias direcciones IP secundarias con sus correspondientes subredes a la interfaz. Con el comando **no ip address** se eliminan estos datos.

Ejemplo

Para asignar a la interfaz 1 varias direcciones cidr:

```
tuna(config)#interface ethernet 1  
tuna(config-if)#ip address 192.168.0.2/24  
tuna(config-if)#ip address 192.168.100.2/24 secondary  
tuna(config-if)#ip address 192.168.200.2/24 secondary
```

Para eliminar todas las direcciones cidr secundarias:

```
tuna(config-if)#no ip address secondary
```

shutdown

Sintaxis

```
shutdown  
no shutdown
```

Descripción

El comando **shutdown** desactiva la interfaz en configuración, mientras que el comando **no shutdown** la re-activa.

Ejemplo

Desactivo la interfaz ethernet 1:

```
tuna(config)#interface ethernet 1
tuna(config-if)#shutdown
```

uplink

Sintaxis

```
uplink
no uplink
```

Descripción

El comando **uplink** activa la interfaz como externa, mientras que el comando **no uplink** la desactiva como tal.

Ejemplo

Activo la interfaz ethernet 1 como externa:

```
tuna(config)#interface ethernet 1
tuna(config-if)#uplink
```

Comandos en modo config

interface ethernet (cambio de modo)

Sintaxis

```
interface ethernet <ifnum>
no interface ethernet %ifether
```

Parámetros

<ifnum> Número de interfaz ethernet a configurar

Descripción

Establece parámetros de la interfaz de red ethernet especificada como parámetro.

Ejemplo

Para configurar la interfaz ethernet 0:

```
tuna(config)#interface ethernet 0
tuna(config-if)# _
```

Para deshabilitar los parámetros de esta interfaz:

```
tuna(config)#no interface ethernet 0
```

ip route

Sintaxis

```
ip route ( <prefix> <ip> | <prefix> ethernet <ifnum> )  
no ip route %iproute
```

Parámetros

<prefix>	Prefijo de origen (Ej: 192.168.100.2/24)
<ip>	Número de IP destino del ruteo
ethernet	Indica que se representará el ruteo por la interfaz correspondiente
<ifnum>	Número de interfaz ethernet destino del ruteo

Descripción

El comando **ip route** agrega rutas de interfaces para protocolo TCP/IP. El ruteo se establece indicando, para cada sub-red, expresada por su prefijo en formato cidr, la puerta de enlace a utilizar. Esta última se puede indicar como una dirección IP o como una interfaz ethernet previamente configurada (a partir del comando **interface ethernet**).

Con el comando **no ip route** es posible eliminar rutas antes establecidas.

Ejemplo

Para establecer que la puerta de enlace por defecto sea 192.168.100.1:

```
tuna#ip route 0.0.0.0/0 192.168.100.1
```

Para establecer que la puerta de enlace por defecto sea la interfaz ethernet 2:

```
tuna#ip route 0.0.0.0/0 ethernet 2
```

Módulo aaa

Comandos en modo config

aaa authentication server

Sintaxis

```
aaa authentication server ( radius | pam | ad )
no aaa authentication server
```

Descripción

Con el comando **aaa authentication server** se establece qué servidor de autenticación usuarios se utilizará. Los protocolos de autenticación soportados por **Tuna** son **radius**, **pam**, y Active Directory (**ad**).

Con el comando **no aaa authentication server** se indica que **Tuna** no realizará autenticación de usuarios.

Ejemplo

Para establecer que **Tuna** autentique usuarios por RADIUS:

```
tuna(config)#aaa authentication server radius
```

aaa accounting server

Sintaxis

```
aaa accounting server radius
no aaa accounting server
```

Descripción

Con el comando **aaa accounting server radius** se establece que se realizará contabilidad (accounting) de usuarios por radius. Con el comando **no aaa accounting server** se indica que el sistema no realice contabilidad de usuarios.

aaa radius authentication

Sintaxis

```
aaa radius authentication ip <auth_ip> port <auth_port> secret
    <auth_secret>
no aaa radius authentication
```

Parámetros

<auth_ip> Número de IP del servidor de autenticación radius.
 <auth_port> Número de puerto (udp) del servidor de autenticación.
 <auth_secret> Contraseña del servidor de autenticación. Acepta todo tipo de caracteres.

Descripción

El comando **aaa radius authentication** permite establecer los parámetros de conexión al servidor de autenticación (authentication). Con el comando **no aaa radius authentication** se deshabilitan estos parámetros.

Ejemplo

Establezco parámetros del servidor de autenticación:

```
tuna(config)#aaa radius authentication ip 192.168.100.77 port 50
secret a1.-*X!y
```

aaa radius secondary-authentication

Sintaxis

```
aaa radius secondary-authentication ip <secauth_ip> port
    <secauth_port> secret <secauth_secret>
no aaa radius secondary-authentication
```

Parámetros

<auth_ip> Número de IP del servidor de autenticación radius.
 <auth_port> Número de puerto (udp) del servidor de autenticación.
 <auth_secret> Contraseña del servidor de autenticación. Acepta todo tipo de caracteres.

Descripción

El comando **aaa radius secondary-authentication** permite habilitar el servidor de autenticación de respaldo, y establecer sus parámetros. Con el comando **no aaa radius secondary-authentication** se elimina el servidor de respaldo establecido.

Ejemplo

Para habilitar y establecer los parámetros del servidor de autenticación de respaldo:

```
tuna(config)#aaa radius secondary-authentication ip 192.168.100.77
port 50 secret a1.-*X!y
```

Para deshabilitar el servidor de autenticación de respaldo:

```
tuna(config)#no aaa radius secondary-authentication
```

aaa radius accounting

Sintaxis

```
aaa radius accounting ip <acct_ip> port <acct_port> secret
    <acct_secret>
no aaa radius accounting
```

Parámetros

<auth_ip> Número de IP del servidor de contabilidad radius.

<auth_port> Número de puerto (udp) del servidor de contabilidad.

<auth_secret> Contraseña del servidor de contabilidad. Acepta todo tipo de caracteres.

Descripción

El comando **aaa radius accounting** permite establecer los parámetros de conexión al servidor de contabilidad (accounting). Con el comando **no aaa radius accounting** se deshabilitan estos parámetros.

Ejemplo

Establezco parámetros del servidor de contabilidad:

```
tuna(config)#aaa radius accounting ip 192.168.100.77 port 50 secret
a1.-*X!y
```

Ejemplo

```
desc
```

```
code
```

aaa radius secondary-accounting

Sintaxis

```
aaa radius secondary-accounting ip <secacct_ip> port <secacct_port>
    secret <secacct_secret>
no aaa radius secondary-accounting
```

Parámetros

<auth_ip> Número de IP del servidor de contabilidad radius.

<auth_port> Número de puerto (udp) del servidor de contabilidad.

<auth_secret> Contraseña del servidor de contabilidad. Acepta todo tipo de caracteres.

Descripción

El comando **aaa radius secondary-accounting** permite habilitar el servidor de contabilidad (accounting) de respaldo, y establecer sus parámetros. Con el comando **no aaa radius secondary-accounting** se elimina el servidor de respaldo establecido.

Ejemplo

Para habilitar y establecer los parámetros del servidor de contabilidad de respaldo:

```
tuna(config)#aaa radius secondary-accounting ip 192.168.100.77 port  
50 secret al.-*X!y
```

Para deshabilitar el servidor de contabilidad de respaldo:

```
tuna(config)#no aaa radius secondary-accounting
```

aaa idle-timeout

Sintaxis

```
aaa idle-timeout <number>  
default aaa idle-timeout
```

Parámetros

<number> Tiempo de inactividad máximo, expresado en segundos

Valor por defecto

60 segundos.

Descripción

El comando **aaa idle-timeout** permite definir el tiempo de inactividad máximo antes que un usuario sea deslogueado automáticamente.

Ejemplo

Para establecer que Tuna desconecte a los usuarios luego de 30 segundos de inactividad:

```
tuna(config)#aaa idle-timeout 30
```

aaa default-bandwidth

Sintaxis

```
aaa default-bandwidth <cir/mir>  
default aaa default-bandwidth
```

Parámetros

<cir/mir> cir/mir a establecer por defecto, expresado en kbits

Valor por defecto

Por defecto el cir es 64 kbits y el mir 128 kbits.

Descripción

El comando **aaa default-bandwidth** permite definir el cir (mínimo ancho de banda garantizado) y mir (máximo ancho de banda permitido) por defecto de las clases de servicio.

Ejemplo

Para establecer por defecto 32 kbits de cir y 64 kbits de mir:

```
tuna(config)#aaa default-bandwidth 32/64
```

Para volver a los valores por defecto:

```
tuna(config)#default aaa default-bandwidth
```

Módulo dhcp

Comandos en modo config-dhcp-pool

range

Sintaxis

```
range <prefix>  
no range
```

Parámetros

<prefix> Subred en que se define el pool de direcciones, en formato cidr

Descripción

El comando **range** establece la subred en la que se basa el pool de direcciones. Con el comando **no range** se pueden eliminar este dato.

Ejemplo

Establezco el pool de direcciones en la subred 192.168.100.0/24:

```
tuna(config-dhcp-pool)#range 192.168.100.0/24
```

router

Sintaxis

```
router <ip>  
no router [ %dhcp-router ]
```

Parámetros

<ip> Dirección IP de la puerta de enlace por defecto a asignar.

Valor por defecto

La puerta de enlace por defecto asignada por dhcp es la correspondiente a la IP de la interfaz que se encuentra dentro del pool que se está definiendo.

Descripción

El comando **router** establecen las puertas de enlace por defecto que asigna dhcp. Con el comando **no router** se pueden eliminar estos datos en forma individual o conjunta.

Ejemplo

Establezco el pool de direcciones en la subred 192.168.100.2/24:

```
tuna (config-dhcp-pool) #range 192.168.100.2/24
```

domain-name

Sintaxis

```
domain-name <name>  
no domain-name
```

Parámetros

<name> Nombre de dominio

Descripción

Establece el nombre de dominio a establecer por dhcp. En caso de no estar definido, utiliza el definido por el comando **ip dhcp domain-name**.

Ejemplo

Para establecer como nombre del dominio foo.com:

```
tuna (config-dhcp-pool) #domain-name foo.com
```

name-server

Sintaxis

```
name-server <ip>  
no name-server [ %dhcp-name-server ]
```

Parámetros

<ip> Dirección IP del dns

Descripción

El comando **name-server** permite establecer la lista de servidores dns que puede asignar el protocolo dhcp. Con el comando **no name-server** se pueden eliminar estas asociaciones en forma individual o conjunta.

Ejemplo

Para que dhcp establezca en el pool de IPs actual que se asigne 2 servidores dns:

```
tuna (config-dhcp-pool) #name-server 206.99.44.254  
tuna (config-dhcp-pool) #name-server 206.99.44.253
```

lease

Sintaxis

```
lease <number> <time_unit>  
no lease
```

Parámetros

<number> Tiempo de renta.
<time_unit> Unidad de tiempo. Los valores válidos son: **days, hours, mins, secs**

Valor por defecto

Por defecto asigna el tiempo establecido con el comando **ip dhcp lease**.

Descripción

El comando **lease** establece el tiempo de renta que el protocolo dhcp asigna direcciones IP a clientes. Pasado este tiempo, la IP queda disponible para otro cliente. Con el comando **no lease** establece este límite como el configurado con el comando **ip dhcp lease**.

Ejemplo

Limito el tiempo de renta de IPs para el pool a 4 horas:

```
tuna(config-dhcp-pool)#lease 4 hours
```

max-lease

Sintaxis

```
max-lease <number> <time_unit>  
no max-lease
```

Parámetros

<number> Tiempo de renta
<time_unit> Unidad de tiempo. Los valores válidos son: **days, hours, mins, secs**

Descripción

El comando **max-lease** establece el tiempo de renta (lease) máximo que el protocolo dhcp puede otorgar direcciones IP a clientes. Con el comando **no max-lease** establece este límite como el configurado con el comando **ip dhcp max-lease**.

Ejemplo

Limito el tiempo de renta máximo de IPs para el pool a 8 horas:

```
tuna(config-dhcp-pool)#max-lease 8 hours
```

fixed-ip

Sintaxis

```
fixed-ip <mac> <ip>
no fixed-ip [ %dhcp-fixed-ip ]
```

Parámetros

<mac> Dirección MAC del cliente
 <ip> Dirección IP a asociar a la MAC

Descripción

El comando **fixed-ip** permite fijar direcciones IP a equipos particulares, para el protocolo dhcp. Esto se logra asociándole una dirección IP a la MAC-Address de la tarjeta de red con que se conecta el equipo. Con el comando **no fixed-ip** se pueden eliminar estas asociaciones.

Ejemplo

Establezco que al cliente con MAC-Address A1:FD:22:BD:4F:FF se le asigne siempre la IP 192.168.100.71:

```
tuna(config-dhcp-pool)#fixed-ip A1:FD:22:BD:4F:FF 192.168.100.71
```

exclude**Sintaxis**

```
exclude <ip1> [ <ip2> ]
no exclude %dhcp-exclude
no exclude
```

Parámetros

<ip1> Número de IP de inicio del rango, o IP única a excluir.
 <ip2> (opcional) Número de ip de fin del rango.

Descripción

El comando **exclude** define direcciones IP o rangos de direcciones a ser excluidos para el protocolo dhcp. Con el comando **no exclude** se pueden eliminar estas exclusiones, todas juntas o individualmente.

Ejemplo

Para definir un pool de direcciones para dhcp en la red 192.168.0.0/24 excluyendo las IP entre 192.168.0.7 y 192.168.0.21:

```
tuna(config-dhcp-pool)#range 192.168.0.0/24
tuna(config-dhcp-pool)#exclude 192.168.0.7 192.168.0.21
```

Comandos en modo config-if

ip dhcp allow-unknown-clients

Sintaxis

```
ip dhcp allow-unknown-clients
no ip dhcp allow-unknown-clients
```

Descripción

El comando **ip dhcp allow-unknown-clients** habilita la conexión de clientes desconocidos para el protocolo dhcp, mientras que **no ip dhcp allow-unknown-clients** la deshabilita.

ip dhcp pool (cambio de modo)

Sintaxis

```
ip dhcp pool <name>
no ip dhcp pool %dhcp-pool
```

Parámetros

<name> Nombre de pool de direcciones IP a asignar al protocolo dhcp

Descripción

El comando **ip dhcp pool** es utilizado para asignar uno o varios pools de direcciones IP al protocolo dhcp correspondiente a la interfaz. Al teclear este comando, se ingresa a un modo en que se definen estos pools. Con el comando **no ip dhcp pool** se eliminan estos datos.

Ejemplo

desc

```
code
```

Comandos en modo config

ip dhcp domain-name

Sintaxis

```
ip dhcp domain-name <name>
no ip dhcp domain-name
```

Parámetros

<name> Nombre de dominio del servidor dhcp.

Descripción

El comand **ip dhcp domain-name** establece nombre de dominio por defecto del servidor de dhcp. Este valor puede ser cambiado para un pool de direcciones particular con el comando **domain-name** en el modo **config-dhcp-pool**.

Con el comando **no ip dhcp domain-name** se desestablece este valor.

Ejemplo

Establezco que por defecto el servidor dhcp establezca el dominio foo.com

```
tuna(config)#ip dhcp domain-name foo.com
```

ip dhcp lease

Sintaxis

```
ip dhcp lease <number> <time_unit>  
no ip dhcp lease  
default ip dhcp lease
```

Parámetros

<number> Tiempo de renta a establecer por defecto.

<time_unit> Unidad en que se expresa el tiempo. Valores posibles: **days, hours, mins, secs**.

Valor por defecto

6 horas.

Descripción

El comando **ip dhcp lease** establece el tiempo de renta (lease) por defecto que el servidor dhcp establecerá a los clientes. Este valor puede ser cambiado para un pool de IPs particular, con el comando **lease**, en modo **config-dhcp-pool**.

Con el comando **no ip dhcp lease** se establece que por defecto no se limite el tiempo de renta.

Ejemplo

Establece que por defecto el tiempo de renta otorgado por dhcp sea de 45 minutos:

```
tuna(config)#ip dhcp lease 45 mins
```

ip dhcp max-lease

Sintaxis

```
ip dhcp max-lease <number> <time_unit>  
no ip dhcp max-lease  
default ip dhcp max-lease
```

Parámetros

<code><number></code>	Tiempo de renta máximo a establecer por defecto.
<code><time_unit></code>	Unidad en que se expresa el tiempo. Valores posibles: days, hours, mins, secs.

Valor por defecto

12 horas.

Descripción

El comando **ip dhcp max-lease** establece el tiempo de renta (lease) máximo por defecto que el servidor dhcp establecerá a los clientes. Este valor puede ser cambiado para un pool de IPs particular, con el comando **max-lease**, en modo **config-dhcp-pool**.

Con el comando **no ip dhcp max-lease** se establece que por defecto no se limite el tiempo de renta.

Ejemplo

Establece que por defecto el tiempo máximo de renta otorgado por dhcp sea de 1 día:

```
tuna(config)#ip max-dhcp lease 1 days
```

service dhcp

Sintaxis

```
service dhcp  
no service dhcp
```

Descripción

El comando **service dhcp** inicia el servicio dhcp del equipo, mientras que el comando **no service dhcp** lo finaliza.

Módulo dns-cache

Comandos en modo config

service dns-cache

Sintaxis

```
service dns-cache
no service dns-cache
```

Descripción

El comando **service dns-cache** inicia el servicio de cache de dns del equipo, mientras que el comando **no service dns-cache** lo finaliza.

dns forwarder

Sintaxis

```
dns forwarder <ip>
no dns forwarder [ %dns-forwarders ]
```

Parámetros

<ip> Dirección IP a agregar como redirector de dns.

Descripción

Agrega un redirector de dns.

El comando **dns forwarder** permite definir a qué direcciones IP debe redirigirse Tuna para resolver nombres de dominios (dns).

Utilizando **no dns forwarder**, se puede eliminar una de las direcciones IP definidas para redirección (especificándola como parámetro), o todas juntas (sin agregarle parámetros).

Ejemplo

Para establecer que Tuna tenga 3 direcciones IP para redireccionar dns:

```
tuna(config)#dns forwarder 206.99.44.252
tuna(config)#dns forwarder 206.99.44.253
tuna(config)#dns forwarder 206.99.44.254
```

Para deshabilitar como redirección de dns una de estas IP:

```
tuna(config)#no dns forwarder 206.99.44.253
```

Para deshabilitar todas las redirecciones de dns establecidas:

```
tuna(config)#no dns forwarder
```

Módulo enable-aaa

Comandos en modo config

service aaa

Sintaxis

```
service aaa  
no service aaa
```

Descripción

El comando **service aaa** inicia el servicio aaa (autenticación, autorización y contabilidad) del equipo, mientras que el comando **no service aaa** lo finaliza.

Módulo filters

Comandos en modo config-filter-group

description

Sintaxis

```
description <desc>
no description
```

Parámetros

<desc> Descripción del grupo de filtros de paquetes.

Descripción

Asigna una descripción al grupo de filtros de paquetes.

El comando **description** es utilizado para asignarle una descripción a un grupo de filtros de paquetes.

El comando **no description** permite eliminar la descripción, si ésta ha sido definida previamente.

Ejemplo

En el siguiente ejemplo, se asigna una descripción a un nuevo grupo de filtro de paquetes:

```
tuna(config)#filter group g1
tuna(config-flt)#description Filtro de ftp
tuna(config-flt)# _
```

filter

Sintaxis

```
filter ( deny | permit | reject ) ( ip | icmp [ icmp-type <icmp> ] |
protocol <prot> ) ( <src-ip> | <src-prefix> | any ) ( <dst-
ip> | <dst-prefix> | any )
```

```
filter ( deny | permit | reject ) ( tcp | udp ) ( <src-ip> | <src-
prefix> | any ) [ <src-port> | <src-port-range> ] ( <dst-ip>
| <dst-prefix> | any ) [ <dst-port> | <dst-port-range> ]
```

```
no filter %matches
no filter
```

Parámetros

deny	Indica que se desea descartar los paquetes que cumplan la regla.
permit	Indica que se desea permitir el paso de los paquetes que cumplan la regla.
reject	Indica que se desea rechazar los paquetes que cumplan la regla.
ip	Indica que se creará un filtro para protocolo IP
icmp	Indica que se creará un filtro para protocolo de control de mensajes de Internet (icmp, Internet-Control Message Protocol)
icmp-type	(opcional) Indica que se especificará un tipo de paquete icmp particular. Tipo de paquete icmp a filtrar. Los tipos válidos son: address-mask-reply, port-unreachable, address-mask-request, precedence-cutoff, communication-prohibited, protocol-unreachable, destination-unreachable, redirect, echo-reply, required-option-missing, echo-request, router-advertisement, fragmentation-needed, router-solicitation, host-precedence-violation, source-quench, host-prohibited, source-route-failed, host-redirect, time-exceeded, host-unknown, timestamp-reply, host-unreachable, timestamp-request, ip-header-bad, tos-host-redirect, network-prohibited, tos-host-unreachable, network-redirect, tos-network-redirect, network-unknown, tos-network-unreachable, network-unreachable, ttl-zero-during-reassembly, parameter-problem, ttl-zero-during-transit.
<icmp>	
protocol	Indica que se desea filtrar paquetes de un protocolo particular. Protocolo a filtrar. Los protocolos válidos son: 3pc, egp, ipcomp, mobile, sat-expak, tcf, a/n, eigrp, ipcv, mtp, sat-mon, tlsp, argus, emcon, ipencap, mux, scc-sp, tp++, aris, encap, ipip, narp, scps, trunk-1, ax.25, etherip, iplt, netblt, sctp, trunk-2, bbn-rcc, fc, ippc, nsfnet-igp, sdrp, ttp, bna, fire, ipx-in-ip, nvp, secure-vmtp, uti, br-sat-mon, ggp, irtp, ospf, sep, vines, cbt, gmtp, isis, pgm, skip, visa, cftp, gre, iso-ip, pim, sm, vmtp, chaos, hmp, iso-tp4, pipe, smp, vrrp, compaq-peer, inlsp, kryptolan, pnni, snp, wb-expak, cphb, iatp, l2tp, prm, sprite-rpc, wb-mon, cpnx, idpr, larp, ptp, sps, wsn, crdup, idpr-cmtp, leaf-1, pup, srp, xnet, crtp, idrp, leaf-2, pvp, sscopmce, xns-idp, dcn, ifmp, merit-inp, qnx, st, xtp, ddp, igmp, mfe-nsp, rdp, stp, ddx, igp, mhrp, rsvp, sun-nd, dgp, il, micp, rvd, swipe.
<prot>	
tcp	Indica que se creará un filtro para protocolo TCP

udp	Indica que se creará un filtro para protocolo UDP
<src-ip>	Dirección IP de origen de paquetes a filtrar.
<src-prefix>	Prefijo origen del paquete.
<src-port>	(opcional) Puerto de origen a filtrar.
<src-port-range>	<i>(opcional) Rango de puertos de origen a filtrar, en formato xx/yy, siendo xx el primero e yy el último del rango.</i>
any	Indica que se desea filtrar cualquier IP de origen/destino.
<dst-ip>	Dirección IP de destino de paquetes a filtrar.
<dst-prefix>	Prefijo destino del paquete.
<dst-port>	(opcional) Puerto de destino a filtrar.
<dst-port-range>	<i>(opcional) Rango de puertos de destino a filtrar, en formato xx/yy, siendo xx el primero e yy el último del rango.</i>

Descripción

El comando **filter** es utilizado para definir paquetes a ser filtrados, y el tipo de filtro a aplicarles. Estos filtros se establecen en base al protocolo al que pertenecen los paquetes, la dirección IP de origen y la de destino, pudiendo también filtrar por puerto de origen y/o destino (protocolos tcp y udp).

Las acciones posibles con los paquetes filtrados, son rechazarlos (reject) o simplemente descartarlos (drop). Es posible permitir el paso de algunos paquetes que se encuentran dentro del ámbito de otros filtros, con la opción **permit**.

El comando **no filter** permite eliminar filtros establecidos previamente, en forma individual o en su conjunto.

Ejemplo

Para establecer que se descarten todos los paquetes provenientes de la ip 66.218.71.198:

```
tuna(config-flt)#filter deny ip 66.218.71.198 any
```

Para establecer que se rechacen todos los paquetes provenientes de la red 66.218.71.198/24, del puerto 80, hacia el puerto 80:

```
tuna(config-flt)#filter reject tcp 66.218.71.198/24 80 any 80
```

Para establecer que se rechacen todos los paquetes de protocolo ospf:

```
tuna(config-flt)#filter reject protocol ospf any any
```

Para eliminar esta última regla de filtrado:

```
tuna(config-flt)#no filter reject protocol ospf any any
```

Comandos en modo config-if

filter

Sintaxis

```
filter <filter-group> ( in | out )  
no filter [ %filter ]
```

Parámetros

<filter-group> Nombre de grupo de filtros a activar para la interfaz
in Indica que el filtro es en la entrada de la interfaz
out Indica que el filtro es en la salida de la interfaz

Descripción

Activa reglas de filtros de paquetes.

El comando **filter** asigna un grupo de filtros previamente definido (con el comando **filter** en modo **config**) a la interfaz, indicando si estos se aplicarán a la entrada o a la salida de la misma. Con el comando **no filter** se deshabilitan estos filtros.

Ejemplo

Para crear un grupo de filtros para protocolo ospf y asignarlo a la entrada de la interfaz 1:

```
tuna(config)#filter group g1  
tuna(config-flt)#description Filtro de ospf  
tuna(config-flt)#filter protocol ospf any any drop  
tuna(config-flt)#exit  
tuna(config)#interface ethernet 1  
tuna(config-if)#filter g1 in
```

Comandos en modo config

filter group (cambio de modo)

Sintaxis

```
filter group <filter-group>  
no filter group %filter-group
```

Parámetros

<filter-group> Nombre de grupo. Acepta todo tipo de caracteres.

Descripción

Ingresa al modo de configuración de grupos de filtros de paquetes.

El comando **filter gorup** es utilizado para crear un nuevo grupo de filtro de paquetes, o modificar uno existente. Utilizando los filtros de paquetes, se puede restringir el pasaje de paquetes de distintos protocolos a través de Tuna.

El comando **no filter gorup** permite eliminar un filtro de paquetes previamente definido.

Módulo **shaping**

Comandos en modo **config-shaping-class**

description

Sintaxis

```
description <desc>
no description
```

Parámetros

<desc> Descripción de la clase de calidad de servicio.

Descripción

Con el comando **description** se asigna una descripción a la clase de calidad de servicio. Con el comando **no description** se elimina esta definición.

Ejemplo

Para establecer que la clase de calidad de servicio 1 se llame “correo electronico”

```
tuna(config)#shaping class 1
tuna(config-sh)#description correo electronico
```

match

Sintaxis

```
match ( include | exclude ) ( ip | icmp [ icmp-type <icmp> ] |
    protocol <prot> ) ( <src-ip> | <src-prefix> | any ) ( <dst-
    ip> | <dst-prefix> | any )

match ( include | exclude ) ( tcp | udp ) ( <src-ip> | <src-prefix> |
    any ) [ <src-port> | <src-port-range> ] ( <dst-ip> | <dst-
    prefix> | any ) [ <dst-port> | <dst-port-range> ]

no match %matches
no match
```

Parámetros

include Indica que se desea incluir la regla en la clase de calidad de servicio.

exclude	Indica que se desea excluir la regla en la clase de calidad de servicio
ip	Indica que se creará una regla para protocolo IP
icmp	Indica que se creará una regla para protocolo de control de mensajes de Internet (icmp, Internet-Control Message Protocol)
icmp-type	(opcional) Indica que se especificará un tipo de paquete icmp particular.
<i><icmp></i>	Tipo de paquete icmp. Los tipos válidos son: address-mask-reply, port-unreachable, address-mask-request, precedence-cutoff, communication-prohibited, protocol-unreachable, destination-unreachable, redirect, echo-reply, required-option-missing, echo-request, router-advertisement, fragmentation-needed, router-solicitation, host-precedence-violation, source-quench, host-prohibited, source-route-failed, host-redirect, time-exceeded, host-unknown, timestamp-reply, host-unreachable, timestamp-request, ip-header-bad, tos-host-redirect, network-prohibited, tos-host-unreachable, network-redirect, tos-network-redirect, network-unknown, tos-network-unreachable, network-unreachable, ttl-zero-during-reassembly, parameter-problem, ttl-zero-during-transit.
protocol	Indica que se creará la regla para paquetes de un protocolo particular.
<i><prot></i>	Los protocolos válidos son: 3pc, egp, ipcomp, mobile, sat-expak, tcf, a/n, eigrp, ipcv, mtp, sat-mon, tisp, argus, emcon, ipencap, mux, scc-sp, tp++, aris, encap, ipip, narp, scps, trunk-1, ax.25, etherip, iplt, netblt, sctp, trunk-2, bbn-rcc, fc, ippc, nsfnet-igp, sdrp, ttp, bna, fire, ipx-in-ip, nvp, secure-vmtp, uti, br-sat-mon, ggp, irtp, ospf, sep, vines, cbt, gmtp, isis, pgm, skip, visa, cftp, gre, iso-ip, pim, sm, vmtp, chaos, hmp, iso-tp4, pipe, smp, vrrp, compaq-peer, i-nlsp, kryptolan, pnni, snp, wb-expak, cphb, iatp, l2tp, prm, sprite-rpc, wb-mon, cpnx, idpr, larp, ptp, sps, wsn, crdup, idpr-cmtp, leaf-1, pup, srp, xnet, crtp, idrp, leaf-2, pvp, sscopmce, xns-idp, dcn, ifmp, merit-inp, qnx, st, xtp, ddp, igmp, mfe-nsp, rdp, stp, ddx, igp, mhrp, rsvp, sun-nd, dgp, il, micp, rvd, swipe.
tcp	Indica que se creará una regla para protocolo TCP
udp	Indica que se creará una regla para protocolo UDP
<i><src-ip></i>	Dirección IP de origen de paquetes.
<i><src-prefix></i>	Prefijo origen del paquete.
<i><src-port></i>	(opcional) Puerto de origen.

<code><src-port-range></code>	<i>(opcional) Rango de puertos de origen, en formato xx/yy, siendo xx el primero e yy el último del rango.</i>
any	Indica que la regla aplica a cualquier IP de origen/destino.
<code><dst-ip></code>	Dirección IP de destino de paquetes.
<code><dst-prefix></code>	Prefijo destino del paquete.
<code><dst-port></code>	<i>(opcional) Puerto de destino.</i>
<code><dst-port-range></code>	<i>(opcional) Rango de puertos de destino, en formato xx/yy, siendo xx el primero e yy el último del rango.</i>

Descripción

El comando **match** agrega una regla a la clase de calidad de servicio. Estas reglas se establecen en base al protocolo al que pertenecen los paquetes, la dirección IP de origen y la de destino, pudiendo también filtrar por puerto de origen y/o destino (protocolos tcp y udp). Es posible excluir algunos paquetes que se encuentran dentro del ámbito de otras reglas, con la opción **exclude**.

Estas reglas pueden luego ser habilitadas con el comando **shaping class** (modo **config-if**).

El comando **no match** permite eliminar reglas establecidas previamente, en forma individual o en su conjunto.

Ejemplo

Para incluir en la regla todos los paquetes provenientes de la IP 66.218.71.198:

```
tuna(config-sh)#match include ip 66.218.71.198 any
```

Para establecer que se incluyan todos los paquetes provenientes de la red 66.218.71.198/24, del puerto 80, hacia el puerto 80:

```
tuna(config-sh)#match include tcp 66.218.71.198/24 80 any 80
```

Para excluir de la regla todos los paquetes de protocolo ospf:

```
tuna(config-sh)#match exclude protocol ospf any any
```

Para eliminar esta última regla:

```
tuna(config-sh)#no match exclude protocol ospf any any
```

Comandos en modo config-if

shaping class

Sintaxis

```
shaping class <shaping-class> [ reversed ] <cirr> <unit>
no shaping class %shaping-class
no shaping class
```

Parámetros

<code><shaping-class></code>	Número de clase de calidad de servicio.
reversed	(opcional) Indica que se desea revertir las reglas de la clase.
<code><cir-mir></code>	CIR/MIR para la clase de servicio en la interfaz.
<code><unit></code>	Unidad de ancho de banda. Los valores aceptados son kbps y mbps .

Descripción

El comando **shaping class** activa, para la interfaz que se está configurando, reglas de calidad de servicio previamente definidas en el modo **config-sh**. El CIR establece el ancho de banda garantizado, mientras que el MIR indica el máximo utilizable para la interfaz en la clase de servicio. Para evitar definir clases de servicio simétricas, es posible utilizar el parámetro **reversed**, que indica que se debe aplicar la regla reversa a la definida.

Con el comando **no shaping class** es posible eliminar las asignaciones de clases de servicio a la interfaz, en forma individual o conjunta.

Ejemplo

Para crear una regla de calidad de servicio para web, teniendo la interfaz ethernet 1 como salida a Internet y la interfaz 2 hacia la red interna:

```
tuna(config)#shaping class 2
tuna(config-sh)#description web
tuna(config-sh)#match include tcp any any 80
tuna(config-sh)#exit
tuna(config)#interface ethernet 1
tuna(config-if)#shaping class 2 32/64 kbps
tuna(config-if)#exit
tuna(config)#interface ethernet 2
tuna(config-if)#shaping class 2 reversed 32/64 kbps
```

bandwidth

Sintaxis

```
bandwidth <number> <unit>
no bandwidth
```

Parámetros

<code><number></code>	Ancho de banda de la interfaz.
<code><unit></code>	Unidad en que se encuentra expresado el ancho de banda. Los valores posibles son: kbit , kbps , mbit .

Descripción

El comando **bandwidth** establece el ancho de banda que opera la interfaz de red que se está configurando. Especificar bien este dato es importante para que **Tuna** pueda manejar bien los CIR.

Con el comando **no bandwidth** se deshabilita este valor.

Ejemplo

Para configurar que la interfaz ethernet 0, que corresponde a la salida a Internet, maneje como ancho de banda 256 kbits:

```
tuna(config)#interface ethernet 0  
tuna(config-if)#bandwidth 256 kbit
```

Comandos en modo config

shaping class (cambio de modo)

Sintaxis

```
shaping class <shaping-class>  
no shaping class %shaping-class
```

Parámetros

<shaping-class> Número de clase

Descripción

Ingresa al modo **config-sh**, en que se establecen parámetros de calidad de servicio.

Módulo graph

Comandos en modo config

graph

Sintaxis

```
graph <name> shaping-class ( auto <shp-cls> [ reversed ] | in <shp-  
  cls-in> out <shp-cls-out> )  
no graph %graph  
no graph
```

Parámetros

<name> Nombre de gráfica.

auto Indica que se establezca la regla en forma automática.

<shp-cls> Número de clase de calidad de servicio.

reversed Indica que se utilice la regla reversa.

<shp-cls-in> Número de clase de calidad de servicio.

<shp-cls-out> Número de clase de calidad de servicio.

Descripción

Establece gráficas de clases de calidad de servicio.

Módulo http

Comandos en modo config

service http

Sintaxis

```
service http  
no service http
```

Descripción

El comando **service http** inicia el servicio web (http) del equipo, mientras que el comando **no service http** lo finaliza.

Módulo nat

Comandos en modo config-if

nat-rule

Sintaxis

```
nat-rule <name>
no nat-rule %nat-enabled
```

Parámetros

<name> Nombre de regla de NAT

Descripción

El comando **nat-rule** activa, para la interfaz, reglas de traducción de direcciones NAT previamente definidas con el comando **nat-rule** en modo **config**. Para poder establecer esta regla correctamente, la regla de traducción de direcciones NAT debe adjudicar una IP asignada a la interfaz (a través el comando **ip address**).

El comando **no nat-rule** permite deshabilitar estas reglas de la interfaz.

Ejemplo

Para asociar la regla de NAT regla.nat a la interfaz ethernet 1:

```
tuna(config)#nat-rule snat regla.nat 192.168.100.2/24 206.99.44.254
tuna(config)#interface ethernet 1
tuna(config-if)#ip address 206.99.44.254/24
tuna(config-if)#nat-rule regla.nat
```

Comandos en modo config

nat-rule

Sintaxis

```
nat-rule snat <name> <prefix> <snat-src>
nat-rule dnat <name> <dnat-src> <dnat-dst>
nat-rule dnat <name> ( tcp | udp ) <dnat-src> <sport> <dnat-dst>
    [ <dnatport> ]

no nat-rule %nat-rules
no nat-rule
```

Parámetros

snat	Indica que se desea crear una regla de traducción de direcciones de origen (SNAT).
dnat	Indica que se desea crear una regla de traducción de direcciones de destino (DNAT).
<name>	Nombre de la nueva regla.
<prefix>	Prefijo de direcciones origen, para SNAT.
<snat-src>	Número de ip de la traducción para SNAT.
<dnat-src>	Número de ip destino a traducir en regla DNAT.
<dnat-dst>	Número de ip destino traducida.
tcp	Indica que se seleccionen paquetes del protocolo tcp para aplicar regla DNAT.
udp	Indica que se seleccionen paquetes del protocolo udp para aplicar regla DNAT.
<sport>	Puerto destino a traducir.
<dnatport>	Puerto destino traducido.

Descripción

El comando **nat-rule** es usado para establecer reglas de traducción de direcciones NAT. Estas reglas pueden luego ser aplicadas a las interfaces con el comando **nat-rule** en modo **config-if**.

Con el comando **no nat-rule** se pueden eliminar estas reglas, en forma individual o conjunta, siempre y cuando no se encuentren asociadas a ninguna interfaz.

Ejemplo

Creo una regla para que cualquier IP de la red interna salga por la 206.99.44.254:

```
tuna(config)#nat-rule snat regla.snat 192.168.100.2/24 206.99.44.254
```

Creo una regla para que todo el tráfico dirigido tcp al puerto 81 de la IP 206.99.44.254 se dirija al puerto 80 de la IP 192.168.100.2:

```
tuna(config)#nat-rule dnat regla.dnat tcp 206.99.44.254 81  
192.168.100.2 80
```

Módulo snmp

Comandos en modo config

service snmp

Sintaxis

```
service snmp  
no service snmp
```

Descripción

Inicia servicios del equipo

Ejemplo

```
desc  
code
```

snmp community

Sintaxis

```
snmp community <snmp-community>  
no snmp community %snmp-communities  
no snmp community
```

Parámetros

<snmp-community> Nombre de la comunidad (Ej: community-1)

Descripción

Establece parámetros del servicio snmp

Ejemplo

```
desc  
code
```

Módulo sshd

Comandos en modo config

service sshd

Sintaxis

```
service sshd
no service sshd
```

Descripción

El comando **service sshd** inicia el servicio ssh del equipo, mientras que el comando **no service sshd** lo finaliza.

ssh 2 authorized_key

Sintaxis

```
ssh 2 authorized_key ( ssh-rsa | ssh-dss ) <pkey> <user>
no ssh 2 authorized_key %sshd-authorized-keys2
no ssh 2 authorized_key
```

Parámetros

ssh-rsa	Indica que se ingresará una clave RSA.
ssh-dss	Indica que se ingresará una clave DSS.
<pkey>	Clave. Acepta todo tipo de caracteres.
<user>	Usuario.

Descripción

El comando **ssh 2 authorized_key** establece claves públicas autorizadas del servicio ssh versión 2. Con el comando **no ssh 2 authorized_key** se pueden eliminar éstas, en forma individual o conjunta.

Ejemplo

Establezco autorización para el usuario foo@bar.com por ssh con su clave pública RSA:

```
tuna(config)#ssh 2 authorized_key ssh-rsa XPIB8o..wE82m== foo@bar.com
```

ssh authorized_key

Sintaxis

```
ssh authorized_key <bits> <exp> <mod> <user>  
no ssh authorized_key %sshd-authorized-keys  
no ssh authorized_key
```

Parámetros

<bits>	Bits de la clave RSA.
<exp>	Exponente de la clave RSA.
<mod>	Módulo de la clave RSA.
<user>	Usuario.

Descripción

El comando **ssh authorized_key** establece claves públicas autorizadas del servicio ssh. Con el comando **no ssh authorized_key** se pueden eliminar éstas, en forma individual o conjunta.

Ejemplo

Establezco autorización para el usuario foo@bar.com por ssh:

```
tuna(config)#ssh authorized_key 1024 35 234561 foo@bar.com
```

ssh port

Sintaxis

```
ssh port <ssh-port>  
default ssh port
```

Parámetros

<ssh-port>	Número de puerto
------------	------------------

Valor por defecto

22.

Descripción

Establece el puerto del servicio ssh.

ssh protocol

Sintaxis

```
ssh protocol <ssh-protocol>  
default ssh protocol
```

Parámetros

`<ssh-protocol>` Número de versión del protocolo ssh.

Valor por defecto

2.

Descripción

Establece la versión del protocolo ssh.

ssh rsa_auth

Sintaxis

```
ssh rsa_auth
no ssh rsa_auth
default ssh rsa_auth
```

Valor por defecto

Habilitada.

Descripción

El comando **ssh rsa_auth** habilita la autenticación vía RSA para el protocolo ssh versión 1. El comando **no ssh rsa_auth** la deshabilita.

ssh password_auth

Sintaxis

```
ssh password_auth
no ssh password_auth
default ssh password_auth
```

Valor por defecto

Habilitada.

Descripción

El comando **ssh password_auth** habilita la autenticación por contraseña para ssh. El comando **no ssh password_auth** la deshabilita.

ssh pubkey_auth

Sintaxis

```
ssh pubkey_auth
```

```
no ssh pubkey_auth  
default ssh pubkey_auth
```

Valor por defecto

Habilitada.

Descripción

El comando **ssh pubkey_auth** habilita la autenticación por clave pública para el protocolo ssh versión 2. El comando **no ssh pubkey_auth** la deshabilita.

Módulo telnet

Comandos en modo config

service telnet

Sintaxis

```
service telnet
no service telnet
```

Descripción

El comando **service telnet** inicia el servicio telnet del equipo, mientras que el comando **no service telnet** lo finaliza.

telnet no-password

Sintaxis

```
telnet no-password
no telnet no-password
```

Descripción

El comando **telnet no-password** desactiva la autenticación del servicio telnet, mientras que el comando **no telnet no-password** la re-habilita.

telnet secret

Sintaxis

```
telnet secret ( 5 <encrypted-secret> | <secret> )
no telnet secret %telnetsecret
no telnet secret
```

Parámetros

- | | |
|--------------------|---|
| 5 | Indica que se ingresa una contraseña ya encriptada. |
| <encrypted-secret> | Contraseña ya encriptada. |
| <secret> | Contraseña sin encriptar. Acepta todo tipo de caracteres. |

Descripción

El comando **telnet secret** permite establecer la contraseña a ser solicitada para acceder vía telnet a **Tuna**. Esta contraseña puede ingresarse ya encriptada, tipeando el literal **5** antes de la misma.

En caso de desear que **Tuna** no solicite contraseña para ingresar vía telnet, se debe utilizar el comando **no telnet secret**.

Ejemplo

Para establecer que **Tuna** solicite contraseña para ingresar vía telnet, y que esta sea *foobar*:

```
tuna(config)#telnet secret foobar
```

Para establecer que **Tuna** solicite contraseña para ingresar vía telnet, y tipearla en modo encriptado:

```
tuna(config)#telnet secret 5 a1.-*X!y
```

Para establecer que **Tuna** no solicite contraseña para ingresar vía telnet:

```
tuna(config)#no telnet secret
```

Apéndices



Preguntas frecuentes

General

¿Por qué me puede interesar Tuna?

La tecnología, junto con los planes comerciales de **netlabs**, están concebidos para viabilizar negocios. En particular, una empresa puede encontrar tres tipos de beneficios en el uso de la tecnología: 1) Mejorar la productividad y reducir costes, 2) Crear nuevos negocios, 3) Brindar valor añadido y fidelizar a sus Clientes.

¿Qué es QoS?

Quality of Service, calidad de servicio, refiere a la calidad con la que se accede a Internet. Los principales factores involucrados a esta calidad son el máximo ancho de banda utilizable (MIR) y el mínimo garantizado (CIR). **Tuna** permite definir estos límites para usuarios, aplicaciones y/o equipos.

¿Qué significa “gestionar acceso a servicios”?

Se trata de un conjunto de reglas que determinan los permisos de equipos y usuarios para acceder a servicios de Internet, como la web, el correo, aplicaciones internas de la empresa, etc.

¿Qué significa “tarificar” el uso de Internet?

Consiste en tener un registro de los usuarios que utilizan el enlace, momentos de conexión y desconexión, ancho de banda utilizado, cantidad de bytes subidos y bajados, y todo tipo de información que permita cobrar de acuerdo a criterios determinados.

Aplicabilidad

¿En qué es útil Tuna para accesos inalámbricos a Internet?

Los accesos inalámbricos a Internet, como ser la tecnología Wi-Fi, tienen varios requerimientos a los que **Tuna** ofrece respuesta, entre los que se destacan: Poder autenticar usuarios, definir QoS de cada uno, cuidar la seguridad de los usuarios, y mantener un registro de los accesos. Todo esto resulta fundamental para controlar la accesibilidad a la red inalámbrica, y eventualmente poder cobrar por ella.

¿Por qué Tuna resulta útil para núcleos empresariales, como ser polígonos industriales, parques tecnológicos, centros comerciales o poblaciones alejadas?

En estos lugares, suele ser conveniente contratar un único enlace a Internet, el cual debe ser distribuido entre varias empresas. La tecnología **Tuna** permite gerenciar el modo en que se reparte el ancho de banda contratado, pudiendo asegurar máximos y mínimos.

¿Por qué Tuna resulta útil para empresas que utilizan VoIP?

Por su naturaleza, las tecnologías de Voz IP generan un tráfico IP que necesita tener asegurada una porción del enlace a Internet. Este requisito no se cumple generalmente en redes que comparten un único enlace entre varios usuarios. En estos casos, la incorporación de **Tuna** asegura buena calidad de comunicación.

¿Por qué Tuna resulta útil para ASPs?

Tuna ofrece a un ASP opciones de cobro por sus servicios. Dado que puede identificar los usuarios que acceden a las aplicaciones, regular su velocidad y tiempo de acceso y mantener un registro del uso de las mismas, genera interesantes modelos de negocio para ASPs y proveedores de hosting.

¿En qué mercados se ha detectado potencialidad de usar Tuna?

- En todo tipo de espacios en que sea útil el acceso inalámbrico: Universidades, bibliotecas, centros comerciales, centros empresariales, hoteles, centros de convenciones, barrios privados, puertos náuticos, estaciones de pasajeros, salas de espera, restaurantes... Y donde resulte útil disponer de conectividad a Internet.

- En empresas que quieran mejorar su productividad, empresas con VoIP, empresas con VPNs.
- En ISPs, ASPs, proveedores de hosting y housing, etc.
- En lugares donde se comparte un enlace a Internet, como edificios empresariales, polígonos industriales, parques tecnológicos o poblaciones alejadas.

¿De qué formas permite Tuna cobrar por el acceso a Internet?

Es posible cobrar de las siguientes formas:

- Acceso prepago
- Pago según uso (pudiendo medirse según tiempo, volumen de tráfico o ancho de banda reservado)
- Tarifas planas.

Por otra parte, es posible diseñar mecanismos especiales de cobro, e implementarlos en base a la tecnología **Tuna**.

¿Cómo pueden los usuarios cambiar dinámicamente su QoS?

Los usuarios que tienen autorizado el cambio dinámico de ancho de banda, pueden hacerlo accediendo a una página web especialmente dedicada a este fin.

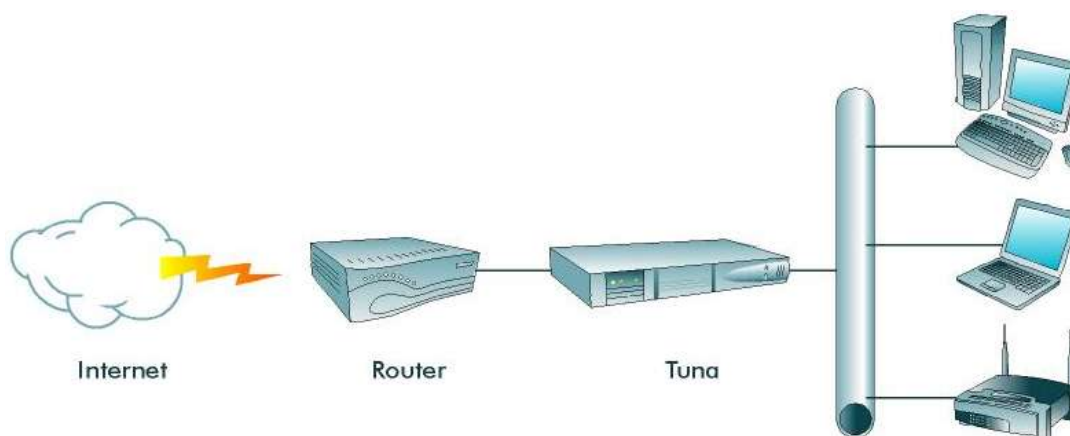
Generalidades técnicas

¿Es posible integrar Tuna a otros sistemas de gestión?

Si. Por un lado, la tecnología **Tuna** genera registros de tarificación, que indican el uso que le dieron a Internet los usuarios de la red. Esta información es exportable en formatos estándar. Por otra parte, uno de los principales valores diferenciales de **Tuna** es su integrabilidad con distintos sistemas, en el marco de proyectos específicos.

¿En qué lugar de la red se coloca?

Se coloca de lado del router de salida a Internet. En algunos casos, se une la tecnología **Tuna** al punto de enlace, como puede ser en equipos con modem ADSL.



¿De qué maneras puede clasificar el tráfico Tuna?

Tuna puede gerenciar QoS según distintos criterios:

- Por usuario
- Por equipo (MAC Address)
- Por dirección IP
- Por aplicación (IP + puerto)
- Por combinaciones de estos factores. Por ejemplo, en primer lugar por usuario y luego por aplicación.

¿Cómo autentica los usuarios Tuna?

Desde el punto de vista del usuario, la primera vez que éste intenta acceder a una página desde su navegador, se le intercepta el tráfico y se le dirige a una página de login. Una vez autenticado, se le permite navegar de acuerdo a su perfil de usuario (QoS, tiempo autorizado de acceso, etc.).

Internamente, **Tuna** realiza AAA con un servidor RADIUS incorporado o externo.

¿Necesito un servidor RADIUS para poder manejar usuarios?

No. **Tuna** incluye un servidor RADIUS propio, aunque puede también actuar como cliente, conectándose a un RADIUS estándar remoto.

¿Qué requerimientos debe cumplir una red y sus equipos para conectarse a Tuna?

No es necesaria una reconfiguración de la red para instalar **Tuna**. Dado que se instala en el borde, todo equipo con acceso a la red local, podrá ser gerenciado por **Tuna**.

Información comercial

¿De qué formas se comercializa Tuna?

Existen cinco formas estándar de comercialización de productos basados en tecnología **Tuna**, las cuales difieren en la interfaz de uso y en los precios:

- **Tuna Corporativo:** Tiene todas las prestaciones y permite manejo centralizado de usuarios de una red.
- **Tuna Conectividad:** Funcionalidades de acceso por usuario y tarificación.
- **Tuna ASP:** Manejo de aplicaciones especialmente pensados para ASPs.
- **Tuna Shaper:** No gestiona usuarios, permite gestionar QoS por equipo (MAC Address o dirección IP).
- **Tuna VoIP:** Asegura ancho de banda para VoIP

Finalmente, la tecnología se puede utilizar en todo tipo de proyectos específicos, en cuyos casos se adaptan los planes comerciales al modelo de negocio puntual.

¿Cómo se licencia?

Cada **Tuna** estándar tiene un costo de licencia que depende del ancho de banda que maneje. Es independiente de la cantidad de usuarios que registre, o de los access points que se le conecten.

¿Cuál es la visión a futuro de la tecnología?

En **netlabs** estamos convencidos del gran futuro que tienen las comunicaciones inalámbricas, y la necesidad cada vez más real de la gente de tener conectividad a Internet en todas partes. La tecnología **Tuna** busca cubrir estas necesidades ofreciendo soluciones a quienes quieran proveer todo tipo acceso a Internet.

Especificaciones técnicas

Clasificación de Tráfico por Flujo

- IP/MAC address (direcciones IP o sub redes)
- Protocolos IP, puertos TCP/UDP, Aplicaciones
- Número de VLAN
- ToS byte / DiffServ / IP Precedence bits

Implementación de Calidad de Servicio (QoS)

- Se pueden definir varias jerarquías de clases de servicio
- Manejo de mínimos y máximos (CIR/MIR) por flujo
- Implementa hasta 10 niveles de prioridad por flujo
- Ancho de banda garantizado, burst rate y CBR por flujo
- Fairness entre flujos de igual prioridad

Seguridad en el Tráfico

- Manejo de ACLs (firewall a nivel de packet filter)
- Controla el número y frecuencia de las conexiones

Autenticación, Tarificación y Monitoreo

- Autenticación vía WEB. No es necesario ningún protocolo de tunelización (PPPoE) para loguearse a la red. La autenticación se hace transparentemente vía WEB cuando el usuario accede a cualquier sitio en internet.
- Tarificación por usuario, por sesión. La tarificación se registra en un servidor RADIUS.
- Acceso a estadísticas vía SNMP

Autorización

- TUNA implementa firewalling, ancho de banda dinámico y tiempo máximo de conexión por usuario. Utiliza un servidor de RADIUS para la autorización.

Miscelaneo

- DHCP Relay
- DNS Caching
- ProxyARP

Índice alfabético



A

aaa accounting server 27
aaa authentication server 27
aaa default-bandwidth 30
aaa idle-timeout 30
aaa radius accounting 29
aaa radius authentication 27
aaa radius secondary-accounting 29
aaa radius secondary-authentication 28

B

bandwidth 50

C

cidr 15
configure terminal 19

D

description 23, 42, 47
disable 21
dns forwarder 39
domain-name 33

E

enable 22
enable secret 17
end 16
erase startup-configuration 21
exclude 35
exit 2, 16

F

filter 42, 44
filter group 45
fixed-ip 34

G

graph 52

H

hostname 17

I

interface ethernet 25
ip address 23
ip address secondary 24
ip dhcp allow-unknown-clients 36
ip dhcp domain-name 36
ip dhcp lease 37
ip dhcp max-lease 37
ip dhcp pool 36
ip name-server 18
ip route 26

L

lease 33
login password 18
logout 21
M
match 47
max-lease 34
N
name-server 33
nat-rule 54
P
prefijo 14
R
range 32
router 32
S
service aaa 41
service dhcp 38
service dns-cache 39
service http 53
service snmp 56
service sshd 57
service telnet 61
shaping class 49, 51
show configuration 20
show startup-configuration 21
shutdown 24
snmp community 56
ssh 2 authorized_key 57
ssh authorized_key 58
ssh password_auth 59
ssh port 58
ssh protocol 58
ssh pubkey_auth 59
ssh rsa_auth 59
T
telnet no-password 61
telnet secret 61
U
uplink 25
W
write memory 19

