Capítulo 6

"Configuración y Ejecución de un Escenario de Red"

6.1 Introducción

En este capítulo se construirá y configurará un típico escenario de red que plantea un buen número de máquinas y dispositivos de red, implementados mediante instancias UML. El objetivo último es que el lector comprenda como se generan las instancias UML´s y como se interconectan en red (entre sí y con el *host* anfitrión). El escenario escogido para tal fin considera la interconexión de dos redes de área local (LAN) a través de un túnel seguro implementado sobre una red pública o inherentemente insegura.

Es importante señalar que los conceptos vertidos en secciones posteriores se aplican a cualquier escenario de red a abordar y que mediante la herramienta de virtualización elegida sólo es posible replicar el comportamiento de los protocolos asociados con TCP/IP y que de ninguna manera se pueden extraer conclusiones sobre medidas de desempeño ligadas fundamentalmente con tiempos y tasas de transmisión.

6.2 Planteando el Escenario de Red

El escenario de red elegido para implementar mediante UML se ilustra en la Figura 6.1. Está conformado por dos Redes de Área Local (LANs) Ethernet interconectadas a través de un enlace provisto por algún proveedor de servicios de telecomunicaciones. Es importante aclarar que el proveedor ofrece un enlace compartido con otros usuarios (no mostrados en la Figura 6.1) y por ende constituye un enlace potencialmente inseguro para los usuarios de las LAN que se requieren interconectar. A los fines de proveer seguridad a las comunicaciones entre las LAN´s descritas se implementará una VPN o *Virtual Private Network* que utilizará los servicios que brinda el protocolo IPSEC para establecer un túnel seguro a través de la red pública.

Como lo ilustra la Figura 6.1, a las LANs que se busca proteger se les asigna las direcciones IP de subred 10.0.1.0 y 10.0.2.0 (red privada de clase A 10.0.0.0). Cada subred cuenta con un *gateway* que posee dos interfaces de red, uno con la LAN y otro con la red pública. Los *gateway´s* se interconectan a cada LAN a través de un *switch* y a la red pública con un *router* utilizando direcciones de red pertenecientes a la 192.168.122.0 y 192.168.1.0 (direcciones IP privadas de clase C).

El *host* anfitrión posee dos interfaces de red para comunicarse con los correspondientes *gateway´s* y ejecuta ruteamiento estático para encaminar el tráfico en el escenario planteado. De este modo, es preciso definir seis (6) máquinas virtuales interrelacionadas entre sí y con el *host* anfitrión tal como lo indica la Figura 6.1.

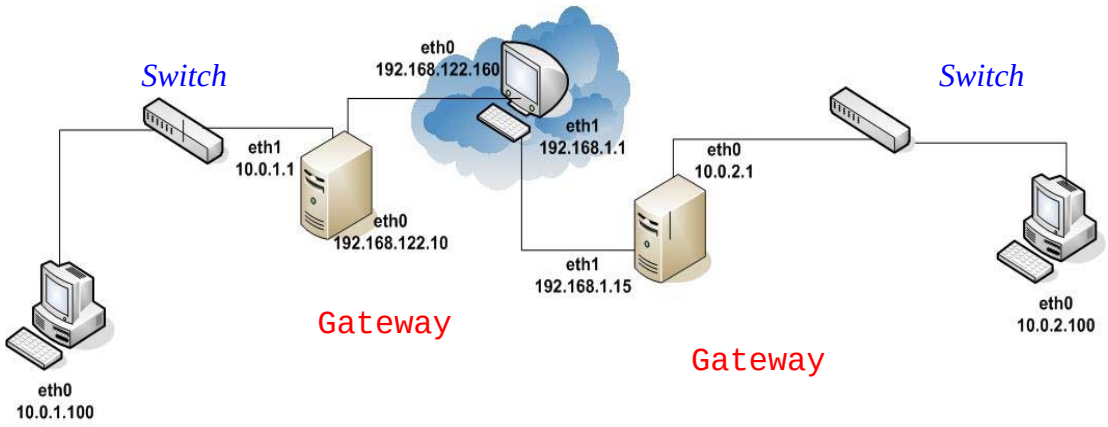

Figura 6. 1: Escenario de red (VPN) a implementar mediante UML

6.3 Ejecutando Instancias UML

Esta sección trata sobre como configurar y ejecutar cada una de las máquinas virtuales necesarias para llevar a cabo el escenario descrito en la Figura 6.1. Antes de ello, es necesario aclarar que este trabajo no persigue como fin, estudiar minuciosamente a cada uno de los comandos ingresados, sino que, el objetivo perseguido es que los dispositivos creados se comporten de manera idéntica a como lo haría un dispositivo real desde el punto de vista de la colección de protocolos TCP/IP.

En primer lugar se utilizarán los comandos adecuados [1] para iniciar cada uno de los *switch*s de la Figura 6.1. Dado a que el escenario planteado es simétrico, en todos los casos y para una fácil comprensión del lector, a los dispositivos se les asignará un sufijo der o izq dependiendo de su posición en la Figura 6.1.

Desde la línea de comando en el host anfitrión y ejecutando la siguiente instrucción se inician ambos *switch*s, o *switch*_izq y *switch*_der:

#uml_*switch* -*Unix* /tmp/izq.ctl –daemon

#uml_*switch* -*Unix* /tmp/der.ctl –daemon

Mediante la ejecución de estos comandos se crean dos *switch*s (Izq y Der) que poseen intefaz de socket por default (puede ser definida por el usuario agregando a *–Unix* el socket deseado) que le permitirá comunicarse con los dispositivos que se conecten a él posteriormente. UML consta de un daemon que cumple las funciones de un *switch* o *hub* (según lo defina el usuario) y que forma parte de las *utilities* insertadas en el proceso de instalación. Este dispositivo provee un mecanismo para crear una red totalmente virtual. Por defecto no provee conexión con el *host* anfitrión (para ello es necesario usar un dispositivo tap).

El siguiente paso en la construcción del escenario virtual dado por la Figura 6.1, es la creación de dos PCs conectadas a cada una de las redes Ethernet en la Figura 6.1. Para ello el lector debe notar que el comando a invocar además de permitir iniciar cada una de las PC´s, debe asegurar la vinculación de ellas a los respectivos *switch*s. De esta forma los comandos para lograr lo anteriormente descrito son:

#./linux umid=”PC_Izq” ubd0=root_fs0.cow,root_fs ubd1=/dev/cdrom ubd2=/dev/fd0 ubd3=/dev/cdrom con=xterm con1=tty con2=tty con3=tty

```
eth0=daemon,10:00:01:02:00:00,,/tmp/izq.ctl

#./linux umid=”PC_Der” ubd0=root_fs1.cow,root_fs ubd1=/dev/cdrom
con=xterm con1=tty con2=tty con3=tty
eth0=daemon,10:00:02:02:00:00,,/tmp/der.ctl
```

Es necesario detener el avance de la explicación por un momento para explicar brevemente la composición de los dos comandos anteriores. En primer lugar es necesario, a fin de evitar confusiones, la identificación de las ventanas que corresponden a las máquinas y dispositivos creados, ello se logra con el argumento umid= seguido por el identificador que aparecerá en la parte superior de la ventana, en este caso PC_Izq y PC_Der.

Todos los dispositivos vistos por el *kernel* Linux UML son virtuales desde el punto de vista del *kernel* anfitrión. Por ejemplo, block devices son dispositivos virtuales que apuntan a archivos dentro del espacio de nombres del *kernel* anfitrión. De esta forma los dispositivos ubd (user bloque device) definidos en ambas líneas de comando (ver más arriba) mapean dispositivos de hardware reales a dispositivos de bloques ubd. Por ejemplo, ubd1=/dev/cdrom mapea el dispositivo de hardware /dev/cdrom a /dev/ubd1 dentro del ambiente user mode *kernel*.

Por último es conveniente explicar el significado de “eth0=daemon, xx:xx:xx:xx:xx,,/tmp/*.ctl”. En primer lugar eth0 es la interfaz de red virtual de PC_Izq o PC_Der, que tiene una dirección MAC dada por xx:xx:xx:xx:xx:xx y que esta ligada al dispositivo *switch* (daemon) a traves del socket *.ctl.

De este modo se han ejecutado las instancias necesarias para construir ambas LAN (Izq y Der). Resta definir los *gateway´*s, cosa que se realizará a continuación:

```
#./linux umid=”GWIzq” ubd0=root_fs2.cow,root_fs con=xterm con1=tty
con2=tty con3=tty eth0=tuntap,,,192.168.122.160
eth1=daemon,10:00:01:01:00:00,,/tmp/izq.ctl

#./linux umid=”DerGW” ubd0=root_fs3.cow,root_fs con=xterm con1=tty
con2=tty con3=tty eth1=tuntap,,,192.168.1.1
eth0=daemon,10:00:02;01:00:00,,/tmp/der.ctl
```

Afortunadamente las explicaciones efectuadas en esta sección son válidas en este caso. De la simple observación en la línea de comandos se desprende que cada *gateway* tendrá una conexión a nivel capa 2 del modelo referencia OSI de la ISO con los *switch*s previamente definidos. Pero ahora hay una nueva interfaz de red (eth1) que necesita de un dispositivo *tuntap* para conectarse al *host* anfitrión. Concretamente una interfaz tap<i> (donde i es un número natural) le permite a una interfaz virtual intercambiar frames ethernet con el *host* anfitrión cuya dirección IP esta incluida en la porción de línea estudiada.

Habiendo finalizado con la etapa de ejecución de las instancias virtuales definidas en el diagrama de la Figura 6.1, sólo resta configurarlas a nivel de capa 3. De ello se tratara la siguiente sección.

6.4 Configuración en Red

Tomando como referencia la Figura 6.1 se procederá a configurar las direcciones de Internet en todos los dispositivos virtuales creados y en el *host* anfitrión (para mayor detalle sobre los comandos de configuración utilizados consultar el Apéndice A). Se debe tener en cuenta que en las máquinas intermedias entre las PC_Izq y Pc_Der (*gateway´s* y *host* anfitrión) es necesario

activar las opciones de reenvió o *ip forwarding*. A continuación se muestran las directivas desde línea de comando para efectuar la configuración:

Configuración del PC_Izq en la sesión UML:

```
# ifconfig eth0 10.0.1.100 netmask 255.255.255.0 up

# route add default gw 10.0.1.1
```

Configuración del PC_Der en la sesión UML:

```
# ifconfig eth0 10.0.2.100 netmask 255.255.255.0 up

# route add default gw 10.0.2.1
```

Configuración del GWIzq (en la instancia UML)

```
# ifconfig eth0 192.168.122.10 up

# route add default gw 192.168.122.160

# ifconfig eth1 10.0.1.1 netmask 255.255.255.0 up

# sysctl -w net.ipv4.ip_forward=1
```

Configuración del GWDer (en la instancia UML)

```
#ifconfig eth1 192.168.1.15 up

#route add default gw 192.168.1.1

#ifconfig eth0 10.0.2.1 netmask 255.255.255.0

#sysctl -w net.ipv4.ip_forward=1
```

Configuración del *host* anfitrión:

```
#route add -net 10.0.1.0 netmask 255.255.255.0 gw 192.168.122.10

#route add -net 10.0.2.0 netmask 255.255.255.0 gw 192.168.1.15
```

Finalmente y a los efectos de verificar la conectividad a nivel de red de la topología dada por la Figura 6.1, es necesario hacer uso del comando ping entre las PC_Izq y PC_Der.

Hasta aquí se han ejecutado todas las instancias virtuales y se ha realizado la configuración a nivel de red de todas las máquinas virtuales y el *host* anfitrión. Resta instalar y configurar el *software* necesario en sendos *gateway´s* para implementar el túnel seguro entre los usuarios de los segmentos de LAN, pero antes de ello, es conveniente realizar un breve resumen sobre IPSec a fin de que el lector comprenda los aspectos que hacen a sus modos de operación como a su funcionamiento.

6.5 IPSEC

IPSec (*Internet Protocol Security*) es un acrónimo de IP *Security*. IPSec provee servicios de seguridad como autentificación, integridad, control de acceso y confidencialidad. Es implementado en la capa de Red o capa 3 del modelo de referencia OSI de la ISO. En el Apéndice B de este trabajo se puede encontrar una descripción detallada de IPSec y de todos los protocolos asociados que hacen a su funcionamiento. A continuación se describirán las opciones obligatorias en cuanto a la compilación del *kernel* según la arquitectura UML, para que la instalación de IPSec sea exitosa. De no tener en cuenta estas opciones de compilación, IPSec no funcionará.

Networking support (NET) [Y/n/?] y

*

* Networking options

*

PF_KEY sockets (NET_KEY) [Y/n/m/?] y

IP: AH transformation (INET_AH) [Y/n/m/?] y

IP: ESP transformation (INET_ESP) [Y/n/m/?] y

IP: IPsec user configuration interface (XFRM_USER) [Y/n/m/?] y

Cryptographic API (CRYPTO) [Y/n/?] y

HMAC support (CRYPTO_HMAC) [Y/n/?] y

Null algorithms (CRYPTO_NULL) [Y/n/m/?] y

MD5 digest algorithm (CRYPTO_MD5) [Y/n/m/?] y

SHA1 digest algorithm (CRYPTO_SHA1) [Y/n/m/?] y

DES and Triple DES EDE cipher algorithms (CRYPTO_DES) [Y/n/m/?] y

AES cipher algorithms (CRYPTO_AES) [Y/n/m/?] y

Al llegar a este punto es necesario recordar que IPSec fue instalado en el sistema de archivos basado en Slackware v 9.1, tal como se mostró en la sección 5.1.1 del capítulo 5. Una vez que el *kernel* a sido configurado e IPSec haya sido instalado se está en condiciones de configurar la VPN.

El comando setkey se utilizará para crear conexiones con difusión manual de claves, y en el escenario de red que se ha planteado se utilizará una conexión en modo túnel, sin embargo es conveniente explicar también el funcionamiento del modo transporte [18]. En el Anexo B el lector encontrará todo lo relativo a IPSec (arquitectura de protocolos y modos de funcionamiento).

6.5.1 Conexiones con difusión manual de claves empleando Setkey

Una conexión con difusión manual de claves significa que todos los parámetros necesarios para el establecimiento de la conexión son proporcionados por el administrador. El protocolo IKE no se emplea para autenticar automáticamente a los comunicantes y negociar estos parámetros. El administrador decide qué protocolo, algoritmo y clave emplear para la creación de las asociaciones de seguridad y rellena la base de datos de asociaciones de seguridad (SAD) de la

manera adecuada.

6.5.1.1 Modo transporte

Esta sección tratará el establecimiento de una conexión en modo transporte con difusión manual de claves. Es la conexión más simple que se puede establecer. Se explicara mediante un ejemplo: dos máquinas con direcciones IP 192.168.1.100 y 192.168.2.100 se comunican empleando IPsec.

Todos los parámetros almacenados en las SAD y SPD pueden modificarse empleando el mandato **setkey**. Indicaremos las opciones necesarias para el establecimiento de una conexión en modo transporte. **Setkey** lee órdenes de un fichero cuando se invoca con **setkey -f /etc/setkey.conf**. A continuación se muestra un archivo /etc/setkey.confTT adecuado:

```
#!/usr/sbin/setkey -f

# Configuración para 192.168.1.100

# Vaciar las SAD y SPD
flush;
spdflush;

# Atención: Emplee estas claves sólo para pruebas
# ¡Debería generar sus propias claves!

# SAs para AH empleando claves largas de 128 bits
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5 \
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5 \
0x96358c90783bbfa3d7b196ceabe0536b;

# SAs para ESP empleando claves largas de 192 bits (168 + 24
paridad)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Políticas de seguridad
spdadd 192.168.1.100 192.168.2.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P in ipsec
           esp/transport//require
           ah/transport//require;
```

El *script* primero limpia la base de datos de asociaciones de seguridad (SAD) y la base de datos de políticas de seguridad (SPD). Después crea las SAs AH y ESP. El mandato **add** añade una asociación de seguridad a la SAD y requiere las direcciones IP de origen y destino, el protocolo IPsec (**ah**), el SPI (**0x200**) y el algoritmo. El algoritmo de autenticación se especifica con **-A.** Tras el algoritmo se especifica la clave. Puede estar formateada en ASCII encerrado entre

comillas dobles, o en hexadecimal con el prefijo **0x**.

Linux da soporte a los siguientes algoritmos para AH y ESP: hmac-md5 y hmac-sha, des-cbc y 3des-cbc.

spdadd añade políticas de seguridad a la SPD. Estas políticas definen qué paquetes se protegerán con IPsec y qué protocolos y claves emplear. El mandato requiere las direcciones IP origen y destino de los paquetes a proteger, el protocolo (y puerto) a proteger (any) y la política a emplear (-P). La política especifica la dirección (in/out), la acción a aplicar (ipsec/discard/none), el protocolo (ah/esp/ipcomp), el modo (transport) y el nivel (use/require).

Este archivo de configuración debe crearse en los dos extremos que formarán parte de la comunicación IPsec. Mientras que el listado mostrado funciona sin ningún cambio en el sistema 192.168.1.100, deberá modificarse ligeramente en 192.168.2.100 para reflejar el cambio de dirección de los paquetes. La manera más sencilla de hacer esto es intercambiar las diercciones en las políticas de seguridad: reemplazar **-P in** por **-P out** y viceversa. El resultado se muestra a continuación:

```
#!/usr/sbin/setkey -f

# Configuración para 192.168.2.100

# Vaciar las SAD y SPD
flush;
spdflush;

# Atención: Emplee estas claves sólo para pruebas
# ¡Debería generar sus propias claves!

# SAs para AH empleando claves largas de 128 bits
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5 \
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5 \
0x96358c90783bbfa3d7b196ceabe0536b;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Políticas de seguridad
spdadd 192.168.1.100 192.168.2.100 any -P in ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;
```

Una vez que el archivo de configuración esté preparado en cada uno de los extremos de la comunicación, puede cargarse empleando **setkey -f /etc/setkey.conf**. Para comprobar el funcionamiento, puede mostrar la SAD y la SPD:

```
# setkey -D
# setkey -DP
```

Si hace ping de una máquina a la otra el tráfico se cifrará y capturando el tráfico con un *snnifer* se mostraran los siguientes paquetes:

```
12:45:39.373005 192.168.1.100 > 192.168.2.100: AH(spi=0x00000200,seq=0x1):
ESP(spi=0x00000201,seq=0x1) (DF)
12:45:39.448636 192.168.2.100 > 192.168.1.100:
AH(spi=0x00000300,seq=0x1):
ESP(spi=0x00000301,seq=0x1)
12:45:40.542430 192.168.1.100 > 192.168.2.100:
AH(spi=0x00000200,seq=0x2):
ESP(spi=0x00000201,seq=0x2) (DF)
12:45:40.569414 192.168.2.100 > 192.168.1.100:
AH(spi=0x00000300,seq=0x2):
ESP(spi=0x00000301,seq=0x2)
```

6.5.1.2 Modo Túnel

El modo túnel se emplea cuando los dos pares que utilizan IPsec funcionan como un *gateway* y protegen el tráfico entre dos redes. Los paquetes IP originales se cifran y encapsulan en un *gateway* y se transmiten al otro extremo del túnel. Allí se desencapsulan y se tratan los paquetes originales sin protección. La configuración de las asociaciones de seguridad y políticas para el modo túnel es similar a la del modo transporte. Se debe crear en el *gateway* derecho el archivo de configuración de *setkey* con los comandos #cd /etc y #cat > setkey.conf.

```
#!/usr/sbin/setkey –f

#Vaciar las SAD y SPD
flush;
spdflush;

#SAs para ESP realizando cifrado con claves largas de 192
bits(168+24paridad)
#y utenticación empleando claves largas de 128 bits
add 192.168.1.15 192.168.122.10 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.122.10 192.168.1.15 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df -A hmac-md5
0x96358cd0783bbfa3d7b196ceabe0536b;

#Politicas de seguridad
spdadd 10.0.2.0/24 10.0.1.0/24 any -P out ipsec
         esp/tunnel/192.168.1.15-192.168.122.10/require;

spdadd 10.0.1.0/24 10.0.2.0/24 any -P in ipsec
         esp/tunnel/192.168.122.10-192.168.1.15/require;
```

Una vez creado se debe cargar la configuración de ipsec ejecutando los siguientes comandos:

```
#cd /usr/sbin
#setkey -f /etc/setkey.conf
```

Ahora se debe crear la configuración correspondiente al *gateway* izquierdo:

```
#!/usr/sbin/setkey -f

#Vaciar las SAD y SPD
flush;
spdflush;

#SAs para ESP realizando cifrado con claves largas de 192
bits(168+24paridad)
#y utenticación empleando claves largas de 128 bits
add 192.168.1.15 192.168.122.10 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.122.10 192.168.1.15 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df -A hmac-md5
0x96358cd0783bbfa3d7b196ceabe0536b;

#Politicas de seguridad
spdadd 10.0.2.0/24 10.0.1.0/24 any -P in ipsec
        esp/tunnel/192.168.1.15-192.168.122.10/require;

spdadd 10.0.1.0/24 10.0.2.0/24 any -P out ipsec
        esp/tunnel/192.168.122.10-192.168.1.15/require;
```

Se deberá cargar dicha configuración con los comandos ya mencionados. Este ejemplo sólo emplea el protocolo ESP. El protocolo ESP asegura integridad y confidencialidad. En este caso el orden de los algoritmos ESP es importante. Primero se necesita definir el algoritmo de cifrado y su clave, y después el algoritmo de autenticación y su clave. Una asociación de seguridad en Linux sólo puede usarse para modo transporte o túnel. El modo transporte es el modo predeterminado, por lo que cuando se desee modo túnel, la asociación de seguridad deberá definirse mediante *-m tunnel*.

Las políticas de seguridad ahora especifican las direcciones IP de las redes protegidas. Los paquetes que empleen estas direcciones IP de origen y destino se cifrarán mediante IPsec. Cuando el modo túnel se usa, la política de seguridad debe especificar *tunnel* y las direcciones IP de los pares que implementan la protección. Esta información es necesaria para localizar las IPsec SA adecuadas.

NOTA: Para mayor información sobre los comandos utilizados en construcción del escenario de red presentado en este capítulo consultar el apéndice A.