Capítulo 5

Monitorización de tráfico

Índice del capítulo

5.1. La retransmisión del tráfico

Como se ha indicado en la sección 4.2.3, "*La sonda y el protocolo NetFlow*", el uso de sistemas de monitorización remota estadística de tráfico, como el uso del protocolo *NetFlow*, llevan al problema de no disponer al completo de la información que está siendo transportada en otro segmento de la red. Como se ha explicado a lo largo de la sección 4.2 de este documento, con *NetFlow* podremos configurar una sonda para que intercepte el tráfico comportándose como un *sniffer* en modo promiscuo y se enviará información de los niveles de Red y Transporte hacia un recolector debidamente configurado. Ahora bien, ¿y si necesitamos poder monitorizar al 100 % el contenido del tráfico que circula por nuestro sistema remoto?

Si queremos poder estudiar con todo nivel de detalle, sin suprimir los campos de datos de los paquetes, el tráfico que circula por un sistema remoto y así poder acceder a la información del nivel de Aplicación tenemos que hacer de alguna forma que dicho tráfico llegue completamente a nuestro sistema remoto. Una vez que disponemos de manera local de dicho tráfico podemos pasar a su estudio utilizando las distintas herramientas que existen, como por ejemplo *ethereal*, tal como se muestra en la Figura 5.1.

Podríamos pensar en soluciones *hardware*, como *switchs* o concentradores comerciales que disponen de un puerto especial por el que replican o retransmiten una copia de todo el tráfico que atraviesa sus puertos. Dicho puerto se podría conectar a nuestro sistema de monitorización para que analizase el tráfico, pero esta forma de solventar el problema tiene sus inconvenientes:

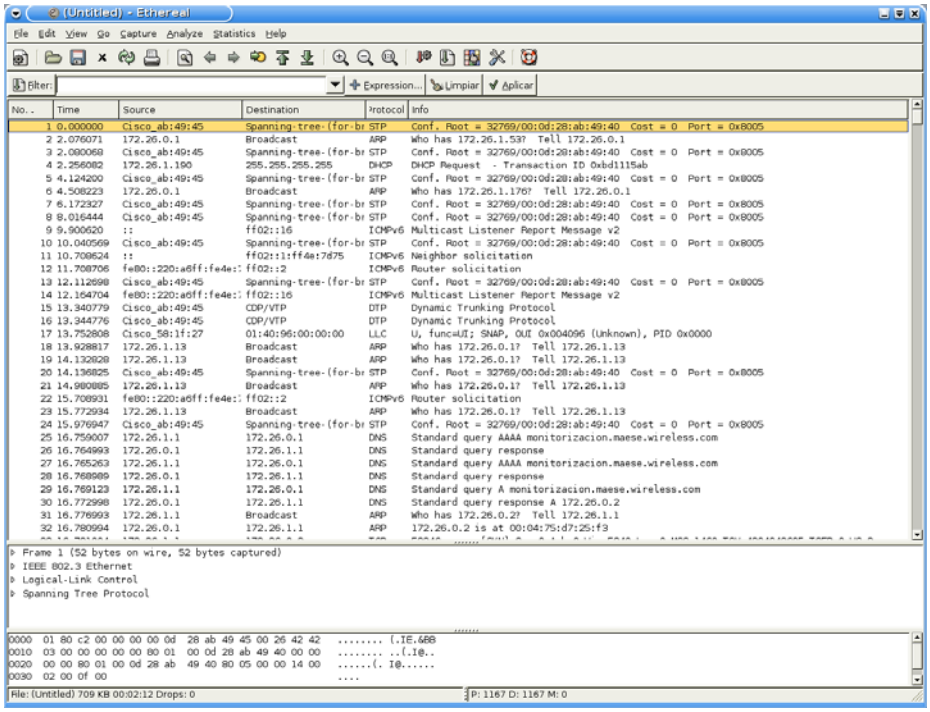

Figura 5.1: Tráfico retransmitido capturado con *ethereal*

- Impone la necesidad de comprar equipos *hardware* adicionales, con el consecuente gasto económico.

- En redes muy cargadas, un ordenador normal que podría bastarnos para nuestro sistema de monitorización podría no ser suficientemente potente para realizar adecuadamente la captura de paquetes, pudiendo producirse pérdidas indeseables. Desear evitar estar pérdidas podría llevar a cambiar dicho equipo, aumentando el coste de la operación.

Por tanto surge la necesidad de buscar formas más elegantes y selectivas, y sobre todo que sean soluciones *software*, para provocar la retransmisión del tráfico.

En este capítulo vamos a ver una posibilidad que existe para obtener una copia del tráfico que circula por un sistema remoto. Primero hablaremos de las capacidades que *iptables* ofrece como retransmisor de paquetes, capacidad que puede ser usada para realizar una copia del tráfico y enviarla hacia otro sistema. En segundo lugar haremos una breve reseña a cómo se pueden efectuar operaciones sencillas y de manera cómoda sobre una máquina remota. Por último, en el apéndice B, "*Monitorización de tráfico*", veremos los detalles de configuraciones e instalaciones de las distintas herramientas comentadas en este capítulo.

5.2. Retransmisión del tráfico: *iptables*

5.2.1. Introducción a *iptables*

iptables es el conocido cortafuegos utilizado en los sistemas Linux a partir de su versión de núcleo 2.4 [25, 26].

iptables consiste en una serie de tablas por las que deben transcurrir los paquetes que queremos enviar, recibir, o que han de atravesar nuestro sistema (paquetes de los cuales no somos ni origen ni destino). En dichas tablas podemos introducir reglas particulares que permiten configurar el cortafuegos, como impedir las comunicaciones con un *host* concreto o prohibir el tráfico de un servicio específico que atraviesa el sistema [27].

iptables dispone de unas capacidades y posibilidades sorprendentes si se profundiza en su funcionamiento. *iptables* puede marcar paquetes de forma interna al núcleo de la máquina para que a los paquetes marcados se les aplique un procesado posterior, puede modificar campos de los paquetes como el campo *ToS* o incluso las direcciones IP de origen o destino de una comunicación, o es capaz de realizar funciones de supervisión de ciertos protocolos. Estas funcionalidades son posibles gracias no sólo al núcleo del propio *iptables* sino a su interacción con el núcleo del sistema (o ***kernel*** de aquí en adelante) y a la modularidad del propio *iptables* que ha permitido a múltiples desarrolladores añadir funcionalidades a este cortafuegos.

iptables suele necesitar de su correspondiente módulo en el *kernel* del sistema para poder funcionar correctamente, por lo que la tarea de utilizar funcionalidades algo especiales de *iptables* suele estar asociada a la tarea de parchear el *kernel* del sistema para que la nueva funcionalidad sea reconocida. Para facilitar esta labor existe la herramienta ***patch-o-matic-ng*** [31] que es un *script* que realiza de forma cómoda la labor de introducir en el código fuente del *kernel* y de *iptables* el código de los módulos de los que queremos hacer uso. Detalles sobre esta herramienta y su uso se verán en el apéndice B, "*Monitorización de tráfico*".

5.2.2. *iptables* y el módulo *ROUTE*

La capacidad de retransmisión de paquetes de *iptables* la hemos conseguido gracias al módulo ***ROUTE*** de *iptables* [29, 30]. Dicho módulo, en la versión que nosotros hemos necesitado usar, está disponible en el *paquete Debian* de la versión actualmente en estado *testing*, concretamente la versión 1.3.3, de *iptables* mientras que la versión en estado *stable* se encuentra en la 1.2.2.

Además de tener que utilizar una versión *testing* de *iptables* es necesario un *kernel* compilado adecuadamente con soporte para el módulo *ROUTE*, y ninguna de las dos versiones del *kernel* que se distribuyen para la versión *stable* de **Debian Sarge** en el momento de realización de este Proyecto, las versiones 2.4.27 y 2.6.8, lo soportan.

Por ello, tras probar a compilar tanto el *kernel* como *iptables* de las versiones *stable* o *testing* para hacer funcionar el módulo *ROUTE* y no haber sido posible, hemos utilizado las versiones de *kernel*, *iptables* y *patch-o-matic-ng* que se detallan en los Cuadros 5.1, 5.2 y 5.3, respectivamente.

Linux Kernel	
Versión	2.6.15.1
Lenguaje	C
Web	http://www.kernel.org

Cuadro 5.1: Ficha de *Linux Kernel*

Utilidad	***iptables***
Autor	Netfilter Team
Versión	1.3.5
Lenguaje	C
Web	http://www.netfilter.org

Cuadro 5.2: Ficha de *iptables*

Utilidad	***patch-o-matic-ng***
Autor	Netfilter Team
Versión	20060206 (snapshot)
Lenguaje	C
Web	http://www.netfilter.org

Cuadro 5.3: Ficha de *patch-o-matic-ng*

Con el módulo *ROUTE*, *iptables* es capaz de realizar funciones de encaminamiento poco habituales, como por ejemplo desviar el tráfico proveniente de un primer sistema destinado a un segundo sistema dirigiéndolo hacia una tercera máquina distinta. En versiones más modernas de este módulo este desvío de tráfico se puede realizar no sólo desviando el tráfico sino dejando el tráfico original sin modificaciones y permitiendo que siga su flujo habitual dentro de las tablas de *iptables*, y realizando una copia de los paquetes se puede encaminar hacia un tercer sistema, que será nuestro sistema de monitorización. A esto es a lo que denominamos *retransmisión* del tráfico.

Gracias a esta funcionalidad, la retransmisión del tráfico es posible con *iptables*. Para ello será necesario la instalación de un núcleo compatible con el módulo *ROUTE* y la instalación de un paquete de *iptables* configurado a tal efecto en cada una de las máquinas donde queramos proceder a la retransmisión de los paquetes. Este proceso de instalación se detalla en el apéndice B.1, "*Instalación del módulo ROUTE*", aunque sería más útil y conveniente disponer de *paquetes Debian* de estas versiones compatibles con *ROUTE* [37, 38, 39]. En los

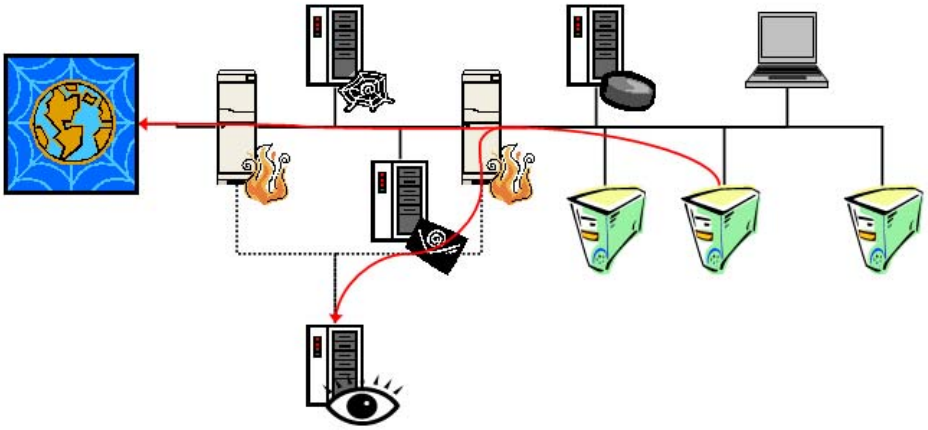

Figura 5.2: Esquema de funcionamiento de la retransmisión de tráfico

apéndices B.1.3.1, "*Creando un paquete Debian del kernel*", y B.1.4.1, "*Creando un paquete Debian de iptables*", se detalla cómo fabricar esos *paquetes Debian*. Tras la instalación bastará proceder a la activación de la retransmisión que consistirá en activar la regla de *iptables* adecuada, algo que veremos con más detalle en el apéndice B.1.5, "*Activación del módulo ROUTE*". Cabe aquí destacar que dado que se basa en el uso de *iptables* cada regla de retransmisión es altamente configurable, de manera que podemos ser muy selectivos con el tráfico que deseemos retransmitir: exclusivamente un puerto, exclusivamente un *host* concreto, exclusivamente un interfaz, etc.

5.3. Ejecución remota de operaciones

Como se ha comentado en la sección 5.2.2, para efectuar la retransmisión de paquetes en las máquinas remotas hacia la máquina que alberga el sistema de monitorización es necesario introducir reglas adicionales en las tablas de *iptables* de los sistemas remotos. Esto implica, evidentemente, una acción de control de forma remota desde la máquina de monitorización sobre las máquinas que retransmitirán el tráfico.

Una manera simple y segura de efectuar esta acción de control es basarse en el uso del paquete ***ssh*** [40, 41] que viene incluido por defecto dentro de nuestro sistema **Debian Sarge**. *ssh* es un reemplazo seguro para los viejos sistemas de ejecución remota de operaciones y de consola remota como *rlogin*, *rcp* o *telnet*. *ssh* se utilizará desde el sistema de monitorización conectándose a servidores activos en las máquinas que habrán de retransmitir el tráfico donde se estará ejecutando el demonio del servidor *ssh*, ***sshd***, también incluido por defecto en la distribución **Debian Sarge**. Tras ello, sobre una conexión encriptada y segura, podremos pasar a la ejecución de comandos en el sistema remoto para activar la retransmisión de paquetes.

Paquete	***ssh***
Autor	Tatu Ylonen < ylo@cs.hut.fi >
Versión	3.8.1
Mantenedor	Matthew Vernon < matthew@debian.org >
Versión del *paquete Debian*	3.8.1p1-8.sarge.4
Lenguaje	C
Web	http://www.openssh.net

Cuadro 5.4: Ficha de *ssh*

En el apéndice B.2, "*Instalación y configuración del paquete ssh*", veremos los detalles de instalación y configuración de ***ssh*** para la creación de pares de llaves cliente-servidor [42] para realizar de forma cómoda y segura la autenticación entre el sistema de monitorización y los sistemas remotos.