# Hacking en redes conmutadas

# INDICE

# 0.Introducción.

Una de las razones por las cuales se usan conmutadores LAN es por la seguridad que estos ofrecen frente a sniffers y capturas de tráfico. Como sabemos el tráfico en una red conmutada es transmitido sólo por el interfaz que contiene en su tabla MAC la dirección destino de la trama, a excepción del tráfico broadcast/multicast. Pero existen mecanismos que permiten capturar el tráfico unicast que no tiene como dirección origen una MAC dada, es decir se puede capturar el tráfico de los interfaces vecinos.
A lo largo de éste documento se expondrá un caso práctico de captura de tráfico en una red conmutada así como otros mecanismos de hacking en redes con switch. También se explicarán las posibles soluciones y posibles prevenciones ante ataques de ésta índole.
Como ejemplo práctico se usará un conocido sniffer llamado, ettercap, y se indicará sniffers similares para conmutadores de nivel 2.

La mayoría de la bibliografía ha sido obtenida de la red, aunque para la consulta del protocolo ARP he usado como referencia el libro Redes de Computadores de Andrew S. Tanenbaum en su tercera edición y el libro Network Intrusion Detection también en su tercera edición.

# 1. Caso práctico: ¡¡Tenemos visita!!

Para situarnos, tenemos una red en la casa de cultura de un pueblo que ésta formada por:

- · PC para la directora de la casa de cultura (windows 2000).
- · PC para la oficina de turismo (windows 2000)
- · 8 PCs en la biblioteca de libre acceso (windows ME)
- · 2 PCs en la biblioteca para la consulta de libros, videoteca, CDteca y DVD's.
- · 1 PC del bibliotecario para la alta de usuarios y los prestamos de libros.
- · 2 Puestos libres para conexión con portátil.
- · Conmutador Ovislink modelo Ether-FSH24RS con 24 interfaces 10/100.
- · Router ADSL 3Com 812, con una velocidad de conexión de 2Mbits/s.
- · Servidor Linux (Debian woody 3.0) que corre los siguientes servicios:
  - -DNS, Bind9. Se tiene un dominio registrado.
  - -DHCP. Para asignar las IPs de todos los ordenadores
  - -EXIM. Gestiona el correo. SMTP, POP.
  - -APACHE. Para el acceso a las web.
  - -MySQL. Contiene la base de datos de los libros y los usuarios. (100Megabytes)
  - -Telnet. Administración del servidor.
  - -Núcleo 2.4.18-bf4.
  - -Iptables para filtrar el tráfico y hacer NAT. Se filtra redes P2P.

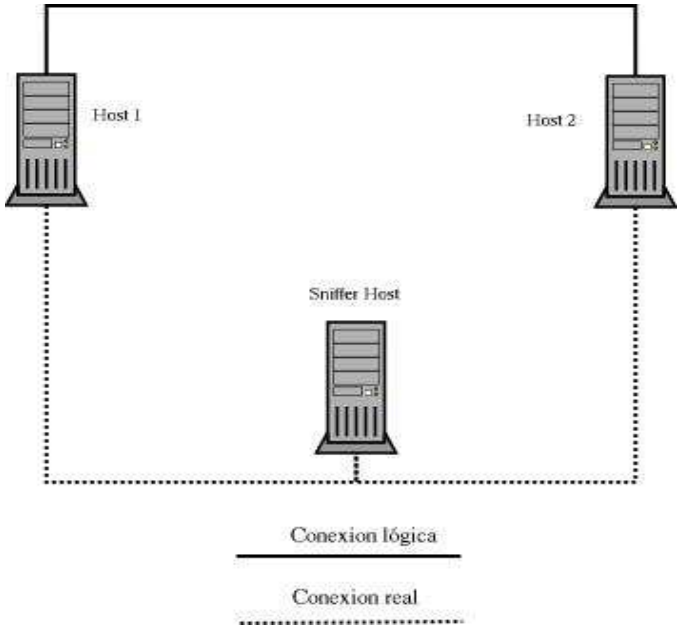
Todos los ordenadores están conectados al conmutador, y el servidor Linux es el que da acceso a Internet a los ordenadores de la red mediante traducción de direcciones (NAT). El servidor tiene 2 tarjetas de red, una conectada al switch y la otra conectada al router que es quien da salida a Internet. La razón de esta configuración es para controlar el tráfico saliente de la red y el entrante mediante iptables, para denegar acceso a redes P2P que saturen el ancho de banda. Los servicios básicos a los que se tiene acceso de Internet son, FTP, http, https, SMTP/POP. Cerrando todo los demá s puertos, Messenger, IRC (6667), etc. No se permite la instalación de software en ninguno de los ordenadores de la biblioteca.
Al servidor telnet sólo se puede acceder desde la red local para evitar accesos no deseados desde el exterior.

La red local nunca ha dado problemas, sólo las veces que nos hemos quedado sin Internet por la caída del servicio ADSL. Pero en momentos determinados se observó que el acceso a la base de datos de los libros y usuarios (OPAC) era excesivamente lenta e incluso no iba. Como primera medida se comprobó que la autonegociación era correcta e incluso se forzó probando todas las combinaciones con mii-tool (man mii-tol). Se probó con 100Half, 100Full y las mismas combinaciones para 10Mbits. El problema no radicaba en la autonegociación, porque la mayoría de la veces con la misma configuración no daba problemas (por defecto 100Full).
Pero cuando se usaba la conexión de portátiles el problema aparecía de nuevo, no pudiendo acceder a la base de datos del servidor. Comprobamos que cuando un usuario concreto conectaba, la red bajaba su rendimiento.
Después de un seguimiento exahustivo al usuario causante de la ralentización de la red, nos fijamos que usaba un programa llamado ettercap.

## 1.1 Ettercap

Ettercap es un sniffer que funciona con un ataque denominado ARP-poison (envenenamiento ARP).
Adress Resolution Protocol es el protocolo que se encarga de averiguar cual es la dirección hardware de una tarjeta de red que tiene asignada una IP, es decir nos devuelve el par MAC/IP de todos los ordenadores de la red local. Lo que hace ARP es enviar un paquete a la dirección broadcast FF:FF:FF:FF:FF:FF de ese segmento preguntando por una IP para que el ordenador con esa IP responda con su dirección MAC. Para evitar tráfico innecesario lo que se hace es guardar en una caché las tablas con las correspondientes MAC/IP, éste mecanismo es lo que permite el ataque que ahora detallaremos. Es imporante tener en cuenta que aunque nosotros no hayamos hecho una petición ARP-Request los paquetes de éste estilo que nos llegan son almacenados en nuestra caché.
Como se deduce del funcionamiento de ARP, el ataque consiste en hacer creer a los ordenadores a espiar que una determinada IP tiene como MAC la dirección hardware del ordenador que hace de espía, de tal manera que todo el tráfico que vaya destinado a la direccion de la vícitima pasará previamente por el ordenador espía.

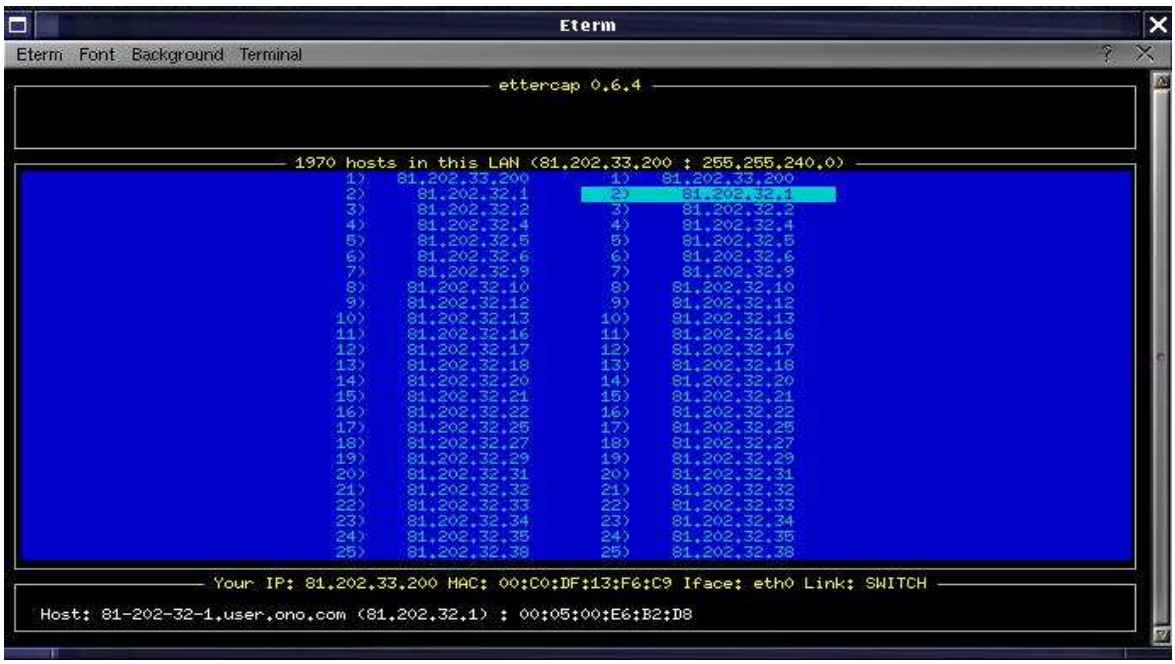
Supongamos que tenemos 3 ordenadores H1, H2 y S (Espía), y queremos ver el tráfico entre H1 y H2. El método que usamos es el siguiene: mandamos paquetes ARP-Reply a los ordenadores H1 y H2, haciendo creer que la direccion MAC de H1 la tiene S y la MAC de H1 la tiene S, y estos guardan en sus cachés esta información. Entonces tanto H1 como H2 al enviar los paquetes lo harán con dirección MAC la de S, por lo que S recibirá todo el tráfico tanto de H1 como de H2.




Es imporante que se manden continuamente paquetes ARP-reply con la MAC de S para que mantengan la caché actualizada y no la refresquen con la información verdadera. Hemos de evitar que los ordenadores destino noten algo raro, para ello se reenvían los paquetes a cada maquina, es decir S actúa como un proxy. Ahora sólo es cuestión de usar un sniffer y capturar los logins o ver el tráfico de un servicio que se tenga abierto.

Nuestro usuario en cuestión lo que hacía era mandar paquetes ARP-reply con la MAC de la tarjeta del servidor que era el router por defecto de todos los ordenadores, de tal manera que todo el tráfico saliente hacia Internet pasaba por él. Evidentemente el tener que procesar tantas

tramas de tantos interfaces (alrededor de 12) hacia que la red se ralentizara mucho. Cada trama enviada hacia Internet, debía pasar por el ordenador espía, mirar si llevaba algún dato importante , guardarlo y reenviar la trama al interfaz correspondiente, en éste caso a la tarjeta del servidor




Éste método permite también que un puente en forma de conmutador pase del estado bridging, o sea transmitiendo por el interfaz que toca, a modo repetición, por todos los interfaces (broadcast), para ello lo que se hace es bombardear al conmutador enviando tramas con MACs falsas hasta llenar sus tablas y como sabemos cuando un conmutador no encuentra en sus tablas la dirección MAC lo que hace es enviarla por flooding.

Ya hemos visto como nuestro usuario mal intencionado lograba capturar el tráfico de todos los ordenadores. Uno de los logins y passwords que capturó fue el login de acceso al servidor para administrarlo, aunque no accediamos mediante root, sino con un usuario normal y luego setuidabamos a root (man su) para actualizar el servidor o hacer cualquier gestion como administradores, el hacker accedio a la máquina con nuestro usuario y aplico un exploit para el kernel, ptrace y consiguió shell de root. (Más info del bug http://lists.debian.org/debian-security/2003/debian-security-200304/msg00118.htm)

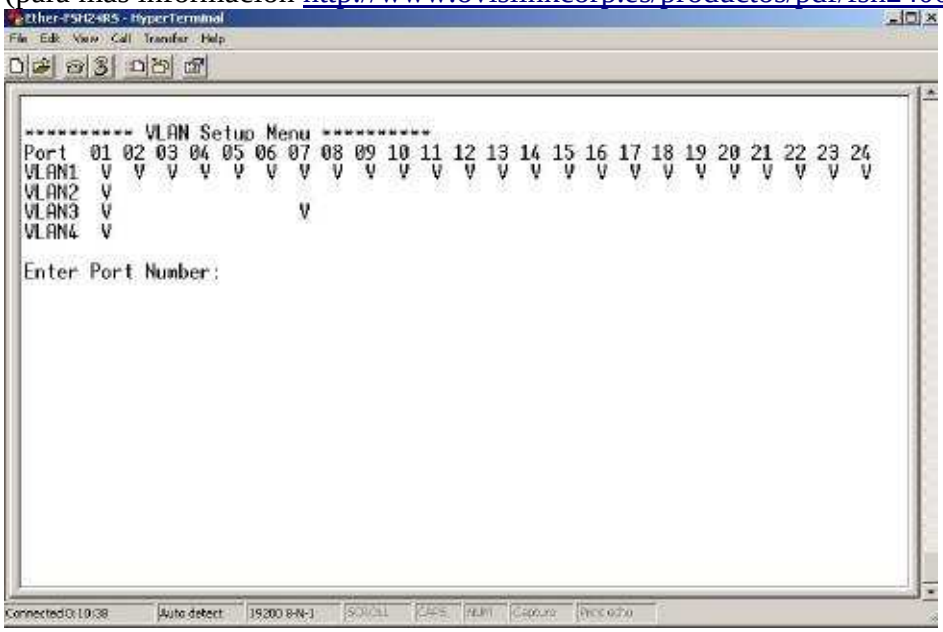
Uno de los problemas con los que cuenta éste ataque es que cualquier sesión SSL (443) puede ser capturada y logeada como si de una sesión sin certificación se tratara. Además también permite capturar sesiones ssh1 que también funciona bajo encriptación. Imaginemos la situación anterior con los ordenadores A (cliente) , B (servidor) y E (espia), capturamos el establecimiento de conexión de A, a continuación evnvíamos a B un certificado identico al de A pero con ip la de la máquina E, al mismo tiempo enviamos un certificado falso al cliente A, por lo que el tráfico sin encriptar pasa por E sin ninún problema.

# 1.2 Evitando el Envenenamiento ARP.

La solución para éste tipo de ataques no es trivial y de hecho en muchos casos no se puede hacer por configuración del conmutador LAN. Buscando en foros de seguridad como el de argo.es, se informa de que existen ciertos conmutadores, como por ejemplo los cisco, que permiten asociar una MAC a un puerto determinado y descartar todo el tráfico que no lleve ésta MAC. (Ver http://mailman.argo.es/pipermail/hacking/2001-September/000677.html ). Pero en otros casos no.

Consultando el manual del conmutador http://www.ovislinkcorp.es/productos/pdf/fsh2400.zip observamos que no se permite la asignación de una o varias MAC a un puerto, de tal manera que por un interfaz sólo puedan pasar tramas con una dirección MAC y no con otras (a modo de filtro). Los conmutadores cisco, si permiten éstas configuraciones. Por ejemplo permite hacer un puerto seguro con el comando "Console> (enable) **set port security 2/1 enable"** y añadir MAC a un puerto en concreto "Console> (enable) **set port security 2/1 enable 00-90-2b-03-34-00**".
Pero como nuestro conmutador no permite éstas opciones y tan sólo permite asignar velocidad y modo (half/full) a cada puerto, crear VLANS y configurar el enlace Trunk, había que buscar una alternativa.
Las posibilidades son dos, bien no permitir la conexión de portátiles (último recurso) o bien filtrar el tráfico que viene de las conexiones portátiles mediante alguna otra manera.
Como solución decidimos crear una VLAN independiente que conectará a otra tarjeta (nueva) del servidor de tal manera que tenemos dos redes independientes, aunque físicamente tuvieran el mismo switch en común. Una vez aislado el tráfico de ambas redes, mediante iptables en el servidor se filtraría el tráfico y ser haría NAT para ambas redes.

Para crear las VLANS hicimos los siguiente:
Accedimos mediante hyperterminal (RS232) al conmutador y en el menu de incio elegimos la opción 2 , settings VLANS. Asignamos a la VLAN2 el puerto 21, 22 , 23, y en la VLAN1 todos los demás.
El puerto 23 es el que va conectado a eth2 del servidor. Éste conmutador no permite la configuración mediante línea de comandos como por ejemplos los cisco, y ha de hacerse todo mediante menús. Como observación decir que éste conmutador en concreto sólo permite la creación de 4 VLANS (para más información http://www.ovislinkcorp.es/productos/pdf/fsh2400.zip )



La red nos querdaría algo como esto

Servidor eth0 ->Conectada al router3com 192.168.5.1
eth1 ->Conectada a VLAN1 192.168.0.1
eth2 ->Conectada a VLAN2 192.168.1.2

VLAN1: Todas la conexiones de la biblioteca y casa de cultura excepto los latiguillos RJ45 para portátiles y el cable hacia eth1.
VLAN2: Los dos latiguillos de los portátiles y el cable hacia eth2.

VLAN1 y VLAN2 no están conectadas de ninguna manera, y de hecho no se ven entre si los ordenadores.
Usamos el programa que maneja iptables, shorewall, para configurar NAT y para filtrar el tráfico. Para ello le indicamos al fichero correspondiente `/etc/shorewall/masq` que debe hacer NAT para las dos interfaces de red.

```
eth0    eth1
eth0    eth2
```

o bien

```
eth0 192.168.0.0/24
eth0 192.168.1.0/24
```

Si lo quisieramos hacer a mano con iptables deberíamos hacer algo del estilo::

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables --flush
iptables --table nat --flush
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
iptables --append FORWARD --in-interface eth1 -j  ACCEPT
iptables --append FORWARD --in-interface eth2 -j  ACCEPT
```

Ahora tenemos dos redes ethernet 192.168.1.0 y 192.168.0.0 totalmente independientes y aisladas de tal manera que el tráfico de una no la ve la otra.

Como medida de seguridad adicional se acutalizó la versión del núcleo a 2.4.21, y se cambio el acceso mediante shell al servidor a ssh2, y se deshabilitó el servicio telnet. Aunque si la red es segura no es necesario encriptar los datos preferimos optar por éste cambio para evitar, por si acaso, captura de los logins y passwords de los usuarios del servidor. Así desahabilitamos todas las cuentas (/bin/false) que no eran propiamente de usuario pero si tenían acceso al servidor, como por ejemplo el usuario opac1 y opac2 que se encargaban de acceder a la base de datos para la busqueda de libros y usuarios.

# 2. Otros métodos de hacking en redes conmutadas.

MAC Flooding: Básicamente se ha explicado anteriormente, consiste en desbordar con direcciones falsas las tablas del conmutador de tal manera que para reenviar las tramas se envían por broadcasting.

MAC Duplicating: Se puede en Linux poner la MAC de otro tarjeta para que el tráfico sea enviado también a éste interfaz. Por ejemplo esto se puede hacer en linux así:

```
ifconfig eth0 down;
ifconfig eth0 hw ether 000000000055;
ifconfig eth0 up;
```

Ataques de Spanning tree: Un hacker envía mensajes BPDU desde su máquina con el objetivo de recalcular el spanning tree. Consecuencia: Negación de servicio (DOS). Durante el recálculo del spanning tree los puentes no transmiten tramas.
También se puede enviar BDPDU con prioridad máxima para que el usuario atacante quede como puente raíz por lo que el hacker puede ver paquetes que antes no podía ver. Para ello el hacker debería estar conectado a dos conmutadores a la vez. Estas técnicas son posibles con un software apropiado en cada caso. (Ver página http://usuarios.lycos.es/elvenbyte/index.php?pagina=stp )

## 2.1 Programas alternativos.

Existen multitud de programas para capturar el tráfico en redes conmutadas, algunos de ellos son ARP-FUN, ARP-tool y Dniff. Las características de dniff son:

Soporta más de 30 protocolos, algunos de ellos estandar y otros propietarios:

FTP, Telnet, SMTP, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP MS-CHAP, NFS, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*Net, Sybase et Microsoft SQL

La mayoría de las sesiones SSL pueden interceptarse, y pueden presentarse certificados ilegítimos pero creíbles.

# 3. Otros usos de ARP-Spoofing

El mecanismo ARP-spoofing no sólo sirve para capturar tráfico, sino también es usado como sistemas de backup o de tolerancia a fallos. Lo que hacemos mediante IP alias y ARP-Sspoofing es permitir que un servidor adopte la identidad de otro servidor que por ejemplo haya caído.

# 4. Detección de ARP-spoofing.

Evitar el ARPspoofing sino se tienen los conmutadores adecuados es casi imposible, pero si es posible la detección de éste tipo de ataques. Hay muchos programas que permiten avisar al administrador de una red o servidor de ataques ARP, básicamente todos hacen lo mismo.
Un programa que hace esto es ARPwatch. Esté programa básicamente monitoriza todos los pares de direcciones ARP/IP y alerta cuando se detecta un cambio. Para realizar esta función se debe de escuchar toda la red a modo de sniffers y compara todo el tráfico con los de una base de datos. Otros programas toman una secuencia de IP/MAC y periódicamente preguntan las actualizaciones en la red. Estos métodos resultan un poco en farragosos cuando se dispone por ejemplo de un servidor DHCP que asigna las IPs dinámicamente. La única solución óptima (a parte de comprar un conmutador que sea “100% against sniffing”) es hacer que todos los datos en la red vayan

encriptados, aunque ésta medida sobrecarga el procesamiento de los datos y añade complejidad en su configuración.

# 5. Conclusiones

Como hemos visto éste tipo de ataques no requieren unas conocimientos muy específicos del funcionamiento de una red ethernet para poderlos efectuar. De hecho existe multitud de software en la red para hacer éste tipo de ataques muy sencillos de manejar.
Es importante en ciertas situaciones con muchos usuarios como por ejemplo redes académicas el controlar éste tipo de ataques porque permiten ver el tráfico de todo un segmento de red.
Tampoco está demás controlar mediante algún programa o viendo las tablas MAC/IP si existe algun tipo de duplicidad o de cambio raro que pueda inducir a un ataque de éste tipo. Siempre se pueden automatizar logs, scripts, etc. para que periódicamente se nos envíe, por ejemplo al correo, unas estadísticas sobre los paquetes ARP y las Ips/MAC.
También se puede aislar tramas en última instancia de tarjetas con una MAC determinada mediante iptables.
Y lo que es más importante, miras que modos de configuración permite un conmutador LAN antes de comprarlo.

# Bibliografía


http://www.seg.inf.uc3m.es/spi/sniffers/EnvenenamientoARP.html Arp Poison

http://bulmalug.net/body.phtml?nIdNoticia=1193 Hacking en redes conmutadas

http://www.governmentsecurity.org/articles/TheIngredientstoARPPoison.php The ingredents to ARP-Poison

http://packetstormsecurity.nl/papers/general/Altering_ARP_Tables_v_1.00.htm Altering ARP tables

http://www.cyruxnet.com.ar/ettercap.htm usando Ettercap

http://www.cyruxnet.com.ar/cuaderno3.htm Man in the middle.

http://www.monkey.org/~dugsong/dsniff/ Dniff. Alternativa a Ettercap.

http://archive.infoworld.com/articles/op/xml/00/05/29/000529opswatch.xml Switched networks lose their security advantage due to packet-capture

http://slashdot.org/articles/00/12/18/0759236.shtml Attack against SSH1 and SSL

http://secinf.net/info/misc/sniffingfaq.html (punto 3.8) How can I sniff a switched network?

http://cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_4/config/sec_port.pdf Conmutadores cisco y configuración óptima.

http://www.ovislinkcorp.es/productos/producto.php?categoria=switch&id=11 Caracetrísticas conmutador Ovislink

http://www.ovislinkcorp.es/productos/pdf/fsh2400.zip Manual conmutador ovislink.

http://usuarios.lycos.es/elvenbyte/index.php?pagina=stp Ataques de spanning tree.

http://packetstorm.securify.com/NT/snarp.zip Sniffer mediante ARP-poison para windows NT

Redes de computadores. Andrew S. Tanenbaum. Tercera edición. **ISBN: 968-880-958-6** (pag 423-433)

Network Intrusion Detection. Stephen Northcutt & Judy Nobak. 3ª edición. **ISBN:0-7357-1265-4**