Envenenamiento ARP

Jaime Pérez Crespo <japecre@gsyc.escet.urjc.es>

29 de Junio del 2005

1. Introducción

Actualmente la forma más habitual de construcción de redes de área local consiste en el uso de conmutadores o *switches* para conectar entre sí los equipos informáticos que las componen.

Un *switch* trabaja a nivel de enlace, pero de forma diferente a como lo hace un concentrador o *hub*, o un medio compartido clásico. Al disponer de *buffers* de almacenamiento, un *switch* es capaz de analizar una trama *ethernet* antes de replicarla por sus bocas. De esta manera, puede comprobar a dónde va dirigida y replicarla **exclusivamente** por la boca correspondiente al destinatario. Esto tiene dos ventajas inmediatas. Por un lado, es posible que distintos equipos utilicen la red de forma simultánea sin interferirse entre ellos, lo cual aumenta considerablemente el rendimiento efectivo de la misma. Adicionalmente, al no replicar las tramas por todas las bocas, evita en un principio la posibilidad de *sniffing* o captura de paquetes de forma fraudulenta.

La técnica del *sniffing* se aprovecha de una característica de todas las tarjetas de red conocida como *modo promíscuo*. Básicamente, una tarjeta de red descarta todas las tramas que le llegan pero no van destinadas a la dirección *MAC* que le corresponde. De esta manera sólo se captura el tráfico legítimo que le corresponde a la tarjeta. El modo promíscuo elimina esta limitación, de forma que una aplicación tiene acceso a todo el tráfico que circula por la red. Por lo general esto tiene aplicación dentro de la gestión de la propia red y la resolución de problemas, pero nada impide que se utilice de forma maliciosa para obtener información sensible que circule por la misma.

2. Protocolo ARP

En las redes actuales que hacen uso de la pila de protocolos *TCP/IP*, como Internet y la mayoría de redes de área local, el encargado de establecer una correspondencia entre las direcciones del nivel de enlace y del nivel de red es el protocolo *ARP*. Se trata de un protocolo sencillo que permite averiguar la dirección hardware de nivel de enlace, en *ethernet* conocida como la dirección *MAC*, partiendo de una dirección *IP* de nivel de red. De la resolución inversa se encarga el protocolo *RARP* (*Reverse ARP*).

El protocolo se desarrolla en apenas el intercambio de dos mensajes. El nodo que desea realizar la traducción compone una petición *ARP* que encapsula en una trama *ethernet* con la dirección *broadcast* del segmento, FF:FF:FF:FF:FF:FF, solicitando que el nodo con la IP buscada responda a dicha petición. Esta trama es recogida por todos los *hosts* de la red, al ir dirigida a la dirección de *broadcast*. Aquellos nodos cuya dirección *IP* no se corresponda con la indicada en la petición *ARP* la ignorarán sin más. Si un nodo encuentra una coincidencia en la petición con su propia *IP*, enviará una trama *ethernet* dirigida al emisor de la petición identificándose a sí mismo. El nodo que originó la petición recibirá la respuesta *ARP* con la dirección *MAC* que deseaba averiguar como origen de la trama.

De esta forma es posible obtener correspondencias de forma dinámica entre direcciones *IP* de 32 bits y direcciones *MAC* de 48 bits, necesarias para que el núcleo del sistema pueda enviar tramas a otras máquinas, pero de las que no se sabe nada a priori. Esto facilita sobremanera la contrucción y administración de una red, ya que ésta es capaz de afrontar los cambios que surjan en tiempo real.

Adicionalmente, y como forma de evitar tráfico innecesario en la red, se optimiza el protocolo con la introducción de cachés de *ARP*. En estas cachés

se almacena la correspondencia entre direcciones *MAC* del segmento de red y las direcciones *IP* asociadas, de forma que antes de enviar una petición *ARP* se trata de resolver buscando en las propias entradas de la caché. Dado que todos los nodos de la red reciben tanto las peticiones como las respuestas de *ARP*, un nodo aprovechará generalmente las respuestas para introducirlas en su propia caché, pese a que no vayan dirigidas a él mismo. Esto hace que una petición *ARP* actualice las cachés no sólo del que la hace, sino del resto de *hosts* de la misma red.

Es posible servirse del protocolo para otros fines. Gracias a la técnica *ARP gratuito*, consistente en el envío de peticiones *ARP* preguntando por la dirección *IP* propia, una máquina puede detectar conflictos con otros nodos con su misma dirección *IP* (en caso de recibir respuesta), e informar al resto de su presencia durante el arranque, induciendo la actualización de sus cachés.

3. Técnica de envenenamiento ARP

El protocolo *ARP* tiene ciertas carencias que facilitan el uso ilegítimo del mismo para recibir tráfico ajeno. En particular, en el caso que nos ocupa, resultan clave las siguientes características:

- Ausencia absoluta de autenticación en el protocolo. Una máquina modificará su comportamiento acorde con los paquetes *ARP* recibidos, sin poder determinar de ningún modo la autenticidad de los mismos.
- Cachés sujetas a alteraciones externas. Es posible modificar los contenidos de una caché *ARP* tan sólo con construir y enviar una petición o respuesta adecuada.
- Actualización de las cachés a iniciativa externa. Con la técnica de *ARP gratuito*, una máquina puede actualizar las cachés *ARP* del resto en cualquier momento.

Precisamente de estas características nos aprovecharemos en la técnica del *envenenamiento AR*P o *ARP spoofing* para recibir tráfico ajeno en una red construida con conmutadores. Se basa en "envenenar" la caché *ARP* de los dos nodos cuya comunicación queremos intervenir con información falsa, haciéndoles creer que su interlocutor es la máquina atacante. De esta forma, el tráfico generado entre ambas máquinas tiene como destino nuestra propia máquina, y desde ésta las tramas son reenviadas al destino real, evitando así la detección del ataque. Más en detalle, un ataque de envenenamiento *ARP* se produce en las siguientes condiciones:

1. La máquina atacante, conociendo las direcciones IP de los dos nodos cuyas comunicaciones se quieren intervenir, resuelve mediante *ARP*, si es necesario, las direcciones *MAC* que les corresponden.

2. Bien mediante respuestas *ARP* o mediante la técnica de *ARP gratuito*, el atacante modifica el contenido de las cachés de las víctimas de forma que para la dirección IP de su interlocutor se corresponda la dirección *MAC* real del atacante.

3. Cada vez que alguno de los nodos quiera enviar información al otro, resolverá la dirección *MAC* del mismo mediante su caché de *ARP* previamente envenenada, enviando así el tráfico al atacante en vez de al destinatario real.

4. El *switch* enviará las tramas por la boca del destinatario, que en este caso es el atacante. Éste las recibirá y las pasará a la aplicación adecuada, que puede ser un *sniffer* que capture todo el tráfico. Al estar todas las tramas destinadas a su dirección *MAC*, **no** es necesario que la tarjeta de red se encuentre en modo promíscuo.

5. El atacante reenviará el contenido de las tramas al destinatario real. La única diferencia entre la trama original y la modificada es, en un principio, la dirección *ethernet* del destinatario, que varía de la del atacante a la de una de las víctimas.

6. El nodo correspondiente recibirá el tráfico como si nada hubiese ocurrido. El atacante, haciendo uso del *envenenamiento ARP* y la técnica del *hombre en el medio* o *man in the middle* ha interceptado el tráfico sin que ninguno de los interlocutores se percate.

4. Herramientas disponibles

En la actualidad existen múltiples herramientas que permiten, de forma más o menos sofisticada, realizar ataques de *envenenamiento ARP* para espiar el tráfico en una red de área local o incluso modificarlo a voluntad.

Cain&Abel es un programa para sistemas operativos de Microsoft que permite recuperar contraseñas. Aunque no es su principal función, utiliza técnicas de *ARP spoofing* para capturar el tráfico de la red y recuperar o incluso modificar contraseñas. Es capaz de interpretar múltiples protocolos, incluso *VoIP* para realizar escuchas telefónicas de forma sencilla. Toda la información relacionada en [4].

dsniff es un conjunto de herramientas para la auditoría de redes de cualquier tipo, incluyendo aquellas construidas con conmutadores. Algunas de sus características más llamativas incluyen la intercepción de tráfico cifrado con los protocolos *HTTPS* (*HTTP Secure*) y *SSHv1* (*Secure SHell*), aprovechando la debilidad de los mismos a la hora de establecer contextos de seguridad basados en una infraestructura de clave pública (*PKI*).

El uso es bastante sencillo. En primer lugar habilitamos el encaminamiento en el núcleo del sistema:

```
# echo 1 > \
> /proc/sys/net/ipv4/ip_forward
```

A continuación modificamos las tablas *ARP* de los dos nodos que deseamos intervenir:

```
# arpspoof -i eth0 192.168.0.1 \
> -t 192.168.0.35
# arpspoof -i eth0 192.168.0.35 \
> -t 192.168.0.1
```

Por último arrancamos *dsniff* indicándole que recomponga el tráfico de red en caso de estar interviniendo una comunicación con la puerta de enlace del segmento de red:

```
# dsniff -c -m -i eth0
```

dsniff capturará todos los paquetes intercambiados por los nodos, detectando automáticamente los protocolos utilizados, y extrayendo contraseñas siempre que sea posible. Si consultamos las tablas *ARP* de cada una de las víctimas mientras se está produciendo el ataque, observaremos que la dirección *MAC* que contienen para el otro nodo ha cambiado a la de la máquina atacante. En [5] se puede ampliar información sobre este potente conjunto de programas.

arp-sk se define por sus propios autores como una "navaja suiza" para *ARP*. Se trata de un generador de paquetes *ARP* que permite casi cualquier uso posible del protocolo para realizar numerosos ataques, no sólo aquellos basados en el envenenamiento de cachés. Desde suplantaciones más elaboradas, hasta denegaciones de servicio a una máquina o incluso toda la subred. En [6] se incluye una guía básica de los usos posibles de *arp-sk*.

arp-tool es un sencillo programa para plataformas *UNIX* que permite realizar envenenamientos de cachés *ARP* y otras manipulaciones construyendo paquetes personalizados. Se ha hecho muy famoso al ser incluido junto a un exploit que permite la obtención de contraseñas de los usuarios de los servicios Hotmail y Messenger de Microsoft, así como la denegación de servicio de los clientes o la instalación en los mismos de software no deseado. Más información en [7].

arpoison es otra herramienta para construir paquetes *ARP* personalizados que permitan realizar ataques de envenenamiento de cachés o similares. Su principal ventaja reside en la posibilidad de ajustar el número de paquetes y el lapso de tiempo entre los mismos a los parámetros que mejor se ajusten a la red que estamos analizando.

ettercap es la herramienta por excelencia para realizar ataques de tipo *man in the middle*, utilizando *ARP spoofing* y otras sofisticadas técnicas. Permite la monitorización y filtrado del tráfico de una red o entre dos nodos de la misma, incluyendo la disección tanto activa como pasiva de múltiples protocolos. Al

igual que *dsniff* o *Cain & Abel*, es capaz de extraer información sensible del tráfico analizado. Está disponible para múltiples plataformas (tanto *UNIX* como Windows) y provee además varias interfaces de uso diferentes, para adaptarse a las circunstancias de la mejor forma posible.

ettercap es capaz de intervenir comunicaciones cifradas bajo *SSL* en protocolos como *HTTPS* o *SSHv1*, lo que lo convierte en una herramienta muy polivalente y ampliamente utilizada. Incluye la posibilidad de realizar ataques de envenenamiento de *DHCP*, robo de puertos o redirecciones *ICMP*, así como el descubrimiento preciso de una red de ordenadores. Por si todas estas características no fuesen suficientes, implementa una arquitectura escalable por medio de *plugins* que permite ampliar su funcionalidad. En [8] se encuentra toda la información relativa a este excelente programa de auditoría y análisis de redes informáticas.

Referencias

[1] W. Richard Stevens (1994), "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley Professional.

[2] David C. Plummer (1982), "Ethernet Address Resolution Protocol", http://www.faqs.org/rfcs/rfc826.html

[3] Sean Whalen (2001), "An introduction to ARP Spoofing", http://packetstormsecurity.com/papers/protocols/intro_to_arp_spoofing.pdf

[4] Massimiliano Montoro, *Cain & Abel*, http://www.oxid.it/cain.html

[5] Dug Song, *dsniff*, http://www.monkey.org/~dugsong/dsniff/

[6] Frédéric Raynal, Éric Detoisien, Cédric Blancher, *arp-sk*, http://www.arp-sk.org/

[7] Gregory Duchemin (2001), *Messenger and Hotmail MITM Exploit (Arptool and Neaky)*, http://www.securiteam.com/exploits/5IP0E2A4UE.html

[8] Alberto Ornaghi, Marco Valleri, *ettercap*, http://ettercap.sourceforge.net/