ENVENENAMIENTO ARP

Seguridad en redes conmutadas

Sergio Valín Cabrera. Telemática. 78621271-A
correocid@hotmail.com

INDICE

INTRODUCCIÓN

Unos de los puntos principales que han sido siempre motivo de quebraderos de cabeza para administradores y gestores de redes es el tema de la seguridad. La seguridad en la manera de mantener íntegra y salvo toda la información confidencial que viaja através de la misma red, manteniendo cifrados seguros, evitando escuchas ajenas , accesos seguros y un largo etc... En este breve , pero conciso trabajo, nos centraremos en un tipo de vulnerabilidad en concreto en redes de arquitectura IEEE 802 o las antiguas DIX Ethernet que es posible hoy día explotar de manera relativamente fácil con algunas herramientas potentes que existen en Internet. Hablamos del envenenamieto de ARP o ARP-poissoning.

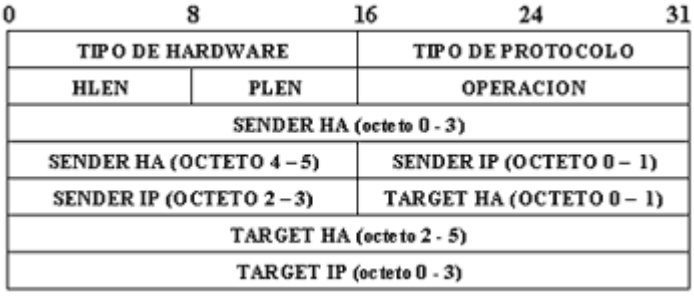
Como su propio nombre indica, el objetivo es envenenar la comunicacion que se produce en el protocolo de comunicacion de paquetes ARP, que es el protocolo de resolución de direcciones responsable de convertir las direcciones de protocolo de alto nivel (direcciones IP) a direcciones de red físicas (MAC). Así pues, este breve trabajo explicaremos básicamente el funcionamiento del protocolo ARP, para centrarnos en donde puede afectar a la seguridad de nuestra red, y una vez hallamos definido el problema y el punto débil, veremos la potencia de la herramienta Ettercap que es capaz de explotar satisfactoriamente la vulnerabilidad a la que nos referimos. Finalmente , conociendo las posibilidades que la vulnerabilidad brinda a posibles atacantes, mostraremos algunas posibles soluciones que evitarán que nuestra red sea vulnerable a hosts maliciosos.

2 .PROTOCOLO ARP

En una red tipo IEEE 802/Ethernet los hosts se comunican conociendo sus direcciones MAC, y no siempre sabremos ni dispondremos a mano de todas las direcciones MAC de todos los ordenadores que conforman la red. Como hemos dicho, el protocolo ARP será el encargado de obtener las direcciones físicas a partir de direcciones IP. En este tipo de redes es necesario conocer la MAC del destino con el fin de que solo el paquete llegue al interfaz de red correspondiente y no a otro, así que cuando se quiere producir una comunicación entre un host1 y un host2 en este tipo de redes, se tendrán que realizar una serie de pasos para llevar a cabo una comunicación correcta. Como observamos, el concepto de ARP solo cobra sentido si nos encontramos en el ámbito de redes conmutadas, pues de lo contrario no sería necesario conocer la MAC del destino dentro de la red, pues la información se enviaría en modo multidifusión a todos los equipos de la red. Imaginemos los siguientes datos y que el host1 quiere enviar paquetes al host2:

HOST1 -> Dirección IP: 10.11.11.1 ; MAC: 11:11:11:11:11:11
HOST2 -> Dirección IP: 10.11.11.2 ; MAC: 22:22:22:22:22:22

Los primero que debemos preguntarnos es: ¿ conoce host1 la dirección MAC del host2 ?. En caso de no conocerla es cuando el protocolo ARP actuará para averiguar a partir de la IP del destino (host2 , que sí la conocemos) su correspondiente MAC. Esta comunicación se encapsula en tramas tipo Ethernet de formato como el que sigue:



TIPO DE PROTOCOLO -> Indica a que tipo de protocolo pertenece la trama (0x0806 indica ARP).
HLEN -> Longitud dirección MAC.
PLEN -> Longitud dirección IP.
OPERACION -> Código de operación (ARP-request o ARP-reply).
SENDER HA -> Dirección de origen hardware. (MAC origen).
SENDER IP -> Dirección IP de origen. (IP de host1).
TARGET HA -> Dirección MAC del destino. (En un arp-request irá vacío).
TARGET IP -> Dirección IP del destino (IP de host2).

En base a este formato, se envían dos tipos de mensajes: Petición ARP (ARP-request) y Respuesta ARP (ARP-reply). El primero es una trama con el código de operación de un

ARP Request (campo operación igual a 1), y es lanzado a la dirección de multidifusión (broadcast FF:FF:FF:FF:FF:FF) del segmento de red con el fin de que el host destino responda a tal mensaje con el respectivo ARP-reply. De esta forma, el host1 que mencionábamos anteriormente tendría que enviar un arp-request a la red con sus campos de la siguiente manera:

OPERACION -> 0x1 (ARP request)
SENDER HA -> 11:11:11:11:11:11 (Ejemplo de MAC del host1)
SEND IP -> 10.11.11.1 (Dirección IP de host1)
TARGET HA -> 00:00:00:00:00:00 (Dirección MAC del host2. Campo vacío, pues es el dato que pretendemos averiguar)
TARGET IP -> 10.11.11.2 (Dirección IP del destino host2)

Esta trama es enviada a todos los ordenadores de la red y únicamente el host destino responderá con un ARP-reply al comparar la dirección contenida en el campo TARGET IP con su propia dirección IP, y en caso de ser iguales intercambiará los valores de los campos de la siguiente manera:

OPERACION -> 0x2 (ARP reply)
SENDER HA -> 22:22:22:22:22:22 (MAC del host2)
SEND IP -> 10.11.11.2 (Dirección IP del destino host2)
TARGET HA -> 22:22:22:22:22:22 (Dirección MAC del host2)
TARGET IP -> 10.11.11.1 (IP del host1)

Con este mecanismo, el host1 ya conoce la dirección MAC del host2 y puede comunicarse con él. A continuación el sistema operativo se encarga de almacenar el par IP/MAC del host2 en una tabla en memoria caché, de donde se reutilizarán por un tiempo las direcciones MAC con el fin de que cada vez que se quiera enviar paquetes IP no se inunde la red con peticiones ARP. Haremos uso de esta tabla para insertar entradas falsas en ella y hacer creer al host de origen que la dirección MAC de una determinada IP le corresponde otra MAC modificada interesadamente.

3. VULNERABILIDAD

Como hemos visto, siempre que un host desea enviar cualquier tipo de información IP a otro host, deberá conocer la MAC del destino para poder transmitir. Es necesario enviar una petición ARP a la red , por medio de la cual sólo responderá el host destino diciendo al origen su dirección hardware. Es posible engañar a un host dentro de la red diciéndole que la dirección MAC con quien se quiere comunicar sea un tercer host, produciéndose así una redirección del tráfico del host origen al host destino, en donde toda la información que salga del host origen pase por un tercer host, y éste a su vez vuelva a redireccionar el tráfico hacia el host destino verdadero con el fin de funcionar transparentemente en la comunicación de ambos y observando todo lo que se dicen. Este tipo de ataque se conoce como “Hombre en medio” o “Man In The Middle”.

Una vez conseguimos “colocar” este tercer host entre dos hosts que se están

comunicando, podremos monitorizar absolutamente todo el tráfico que circule entre origen y destino. En consecuencia podremos extraer cualquier tipo de información valiosa aplicando filtros determinados a los paquetes que el host malicioso va recibiendo del host origen. Claramente se observa aquí como se pone en tela de jucio la seguridad de toda la red en el momento en que un host ajeno es capaz de entrometerse en el tráfico de una comunicación supuestamente segura en una red conmutada.

Como se sabe, en redes conmutadas se asegura que el tráfico llega a un único destino y no a la totalidad de la red como se hacen con concentradores o Hubs, en donde toda la información es enviada a todos los hosts conectados a la red, y son estos los que rechazan todos aquellos paquetes que no van destinados a ellos. En este tipo de redes, es posible configurar la interfaz de red en modo promiscuo y así leer la totalidad de los paquetes que nos llegan. La seguridad en este tipo de redes está claro que no es su punto fuerte. Así, al disponer de un elemento de red como es un switch o encaminador nos aseguramos que nadie más sino el destino recibe nuestros paquetes. Pero, ¿qué ocurriría si un host atacante fuera capaz de hacer creer a un host víctima que la MAC del host-destino no es la MAC de host-destino sino la MAC del host atacante?, pues que claramente para el host víctima , el host atacante estaría suplantando la identidad del host-destino, y en consecuencia recibiendo los paquetes que deberían ir destinados al mismo. Ésto se lleva a cabo en la práctica enviando continuamente arp-replys al host víctima con los campos a los siguientes valores:

OPERACION -> 0x2 (ARP reply)
SEND IP -> 10.11.11.3 (Dirección IP del host víctima)
SEND HA-> 33:33:33:33:33:33
TARGET HA -> 33:33:33:33:33:33 (Dirección MAC del supuesto host-destino, aquí va la MAC del host atacante)
TARGET IP -> 10.11.11.2 (Dirección IP del host víctima)

De esta manera, el host atacante conseguirá modificar la tabla ARP en caché del host victima introduciendo una entrada falsa, de la forma :

10.11.11.2 (IP host destino) is at 33:33:33:33:33:33 (MAC del host atacante)

En consecuencia, cuando el host victima quiera enviar informacion a 10.11.11.2 la encapsulará en una trama ethernet pero con la dirección MAC del host atacante, haciendo que cuando llegue la trama al switch, éste la encamine a la boca de salida que corresponde con la dirección MAC especificada, es decir, hacia el host atacante.

En esencia , hasta aquí queda explicado el concepto de envenenamiento ARP. A continuación pasaremos a contemplar algunas herramientas existentes que se encargan de explotar esta vulnerabilidad, para a posteriori , comentar cómo podríamos evitar este tipo de ataques en nuestra red.

4.- Ettercap

Una de las herramientas más potentes que hacen uso del envenenamiento ARP en entornos GNU/Linux es Ettercap (http://ettercap.sourceforge.net). Ésta es una herramienta

con licencia GPL diseñada con el fin de analizar , filtrar, logear y escuchar determinado tráfico circulando por la red. Soporta disección activa y pasiva de muchos protocolos (incluso los cifrados). Es capaz de realizar ataques MITM (Man in the Middle) entre diferentes hosts de la red , con el fin de entrometerse en su comunicación y obtener información valiosa tipo contraseñas POP, SSH, Telnet, FTP, Https, etc...

Decir que la mayoría de los sistemas operativos (excepto linux 2.4 y solaris 8) no implementan estados en el protocolo arp (aceptan arp-replys sin haber enviado antes un request), por lo que facilmente aceptan respuestas ARP y en consecuencia modifican su tabla ARP en caché con cada arp-reply recibido.

Pasemos pues a explicar el funcionamiento en la práctica de este potente software.

Ettercap tiene dos modos de sniffar:

UNIFIED -> Esnifa todos los paquetes que pasan por el cable. Es posible poner la interfaz de red en modo promiscuo y permitir que los paquetes que no vayan dirigidos a él, los redireccione usando la capa 3 de enrutado. Así es posible lanzar un ataque MITM con otra herramienta y permitir que ettercap redireccione los paquetes a su destino verdadero.

BRIDGED -> Utiliza dos interfaces de red y direcciona el trafico de una a la otra mientras se esnifa y se filtra si interesa. Es un método absolutamente anónimo pues es imposible detectar un ataque MITM por parte de los otros hosts de la red. Se puede decir que se trata de un MITM a nivel de la capa física. Estarás en medio del cable entre dos entidades. No es recomendado en gateways pues transformará tu gateway en un bridge.

Las características mas relevantes de Ettercap son:

*__Soporte SSH1__ -> Capacidad para snifar usuario y contraseña de conexiones ssh1. Ettercap es el primer software capaz de snifar este tipo de tráfico en modo Full-duplex.
*__Soporte SSL__ -> Capacidad de snifar datos cifrados con SSL. Un certificado falso es presentado al host victima y la sesión es desencriptada.
*__Inyección de caracteres__ en una conexión establecida.
*__Filtrado de paquetes__ -> Por medio de scripts es posible buscar por cadenas específicas en paquetes TCP y UDP, con el fin de modificarla por una que nosotros deseemos, deshechar paquetes, etc..
*__Mata conexiones__ -> Capacidad para finalizar conexiones TCP.
*__Soporte de Plug-ins__ -> Capacidad de creación de tus propios plugins utilizando las API de Ettercap.
*__Capturador de contraseñas__: TELNET, FTP, POP, RLOGIN, SSH1, ICQ, SMB, MySQL, HTTP, NNTP, X11, NAPSTER, IRC, RIP, BGP, SOCKS 5, IMAP 4, VNC, LDAP, NFS, SNMP, HALF LIFE, QUAKE 3, MSN, YMSG.
*__Fingerprint pasivo__ -> Posibilidad de escanear la red local en modo pasivo (sin enviar ningun paquete) y obtener información detallada sobre los hosts, como el sistema operativo que utilizan, servicios en ejecución, puertos abiertos, IP, direcciones MAC y vendedor del adaptador de red.

Modo de utilización:

root@localhost # ettercap OPCIONES OBJETIVO1 OBJETIVO2

El trafico será snifado de objetivo1 a objetivo2 y viceversa (no existe el concepto de Origen y Destino).

La sintaxis de OBJETIVO1 y OBJETIVO2 es de la forma MAC/IPs/Puertos. Existe la posibilidad de omitir algun campo , indicando en consecuencia que la condición será siempre aceptada para cualquier valor en el campo omitido. Ejemplo:

//110 significa cualquier dirección MAC e IP, pero sólo el puerto 110.
00:23:64:07:BD:AC//80 -> Cualquier IP, pero sólo la MAC indicada y el puerto 80

Es posible indicar un rango de Ips o varias Ips en el objetivo. Esto se hace utilizando el carácter '-' para indicar rango, coma (',') para especificar extension de la Ip y punto y coma (';') para separar las direcciones IP. Ejemplo:

11:22:33:44:55:66/**10.13.12.4-7;10.13.12.78,79**/80 -> Sólo la MAC 11:22:33:44:55:66, con puerto 80 y las direcciones IP: 10.13.12.4, 10.13.12.5 , 10.13.12.6 , 10.13.12.7 , 10.13.12.78 o 10.13.12.79

Lo mismo ocurre con los rangos de puertos (indicamos la separación con ',' y el rango con '-':

//80-110,143,6667 -> Cualquier MAC e IP, pero solo los puertos del 80 al 110, el 143 o el 6667.

Es posible también indicar el modo inverso al indicar un objetivo, de forma que snifaremos todo el tráfico MENOS el objetivo que hemos definido. Se utiliza con la opción -R (de reverse). Ejemplo:

ettercap **-T -R /10.13.12.7/** -> Snifaremos todo el trafico proveniente de cualquier direccion MAC, cualquier PUERTO y cualquier IP excepto 10.13.12.7. El parámetro -T indica que ejecutaremos ettercap en modo texto, al contrario que -G , que lo ejecutará en modo gráfico haciendo uso de GTK+ y -C haciendo uso de las librerías Curses.

Hasta aquí todo respecto a la sintaxis de los OBJETIVOS. Una vez estemos ejecutando ettercap, implicará que éste direccionará de manera automática todos los paquetes que nos lleguen a sus destinos correspondientes. Esto implica además que el parámetro ip_forwarding (*/proc/sys/net/ipv4/ip_forward*) del kernel deberá estar desactivado con el fin de que la redirección de paquetes la lleve a cabo Ettercap y no el kernel, así se evita que los paquetes aparezcan por duplicado en la red. Por suerte, ettercap en la versión NG 0.7.3 ya modifica este parámetro automáticamente. Es posible ver ayuda más detallada haciendo uso de las páginas de Man que vienen en la instalación del ettercap o consultando el foro de ettercap (http://ettercap.sourceforge.net/forum). Aquí pasaremos a hacer algunas pruebas de cómo llevaríamos a cabo un ataque de MITM con envenenamiento ARP, aunque ettercap posibilita también llevar a cabo otros tipos de ataques (dhcp, port stealing, icmp...), pero que no se profundizarán en ellas aquí.

Para llevar a cabo un ataque en toda la red utilizando envenenamiento ARP escribimos:

 # ettercap -Tq -M arp:remote //

T = Modo texto
q = quiet – No visualizamos el contenido del paquete. Compatible con modo consola solo.
M = Ataque Man in the middle
arp: = envenenamiento ARP
remote = permite snifar direcciones ip remotas , envenenando así al gateway.
// = especificación del objetivo: todos los hosts de la LAN.

Si por ejemplo queremos ejecutar Ettercap como herramienta que efectúa un ataque ARP contra una determinada IP y a continuación monitorizar el tráfico con otras herramientas, como por ejemplo Ethereal, lo haremos por medio del siguiente comando:

 # ettercap -T -o -M arp:remote /10.45.34.1/

Esto indica que ettercap envenenará la tabla ARP de 10.45.34.1 y que estaremos en medio de su tráfico dirigido hacia él, y el que él mismo produzca. Sólo deberemos agregar el flag -z (-Tz) si queremos que solo capturemos paquetes procedentes de la IP, y no tráfico diriguido a él.

Ettercap efectúa un ARP request a los hosts que le pasemos como objetivos con el fin de obtener su verdadera MAC. Esto supondrá que cuando efectuemos un ataque a toda la red (//), ettercap producirá una tormenta de peticiones arp a cada host de la red, con sus respectivas respuestas, lo que no sería deseable pues saturaría de tráfico ARP la red de manera innecesaria. Por ello, para que esto no ocurra cada vez que lanzemos ettercap, es posible guardar la lista de hosts en un fichero, para cargarla en futuros usos con el fin de no producir el torrente enorme de peticiones arp a toda la red. Sería de la siguiente forma:

 # ettercap -T // -k lista_de_hosts.txt

Cuando volvamos a hacer uso de ettercap, solo tendremos que cargar el archivo de la forma:

 # ettercap -T // -j lista_de_hosts.txt

Así mismo, también podemos hacer uso de filtros a partir de la herramienta etterfilter ($ **man etterfilter** para más información), que crea un fichero compilado para que ettercap pueda cargarlo y utilizarlo. Algun ejemplo de creación de un filtro para luego ponerlo en funcionamiento podría ser algo así:

Archivo filtro.txt

```
if (tcp.src == 80 && search(DATA.data,"google.com")) {
        replace("google.com", "yahoo.com");
        msg("Redireccionando a yahoo\n");
}
```

fin de archivo.

En la sintaxis anterior estamos haciendo un filtrado de los paquetes que vayan dirigidos al puerto 80, es decir, el puerto para la navegación y además contengan la palabra “google”. Por lo tanto lo que pretendemos aquí es ver como podemos direccionar a otra dirección (en este caso a yahoo).

 # etterfilter filtro.txt -o filtro // Para compilar el filtro.
 # ettercap -Tq -F filtro -M arp:remote /10.45.76.5/ //

Con esta ultima instrucción lanzamos ettercap para que filtre y redireccione los paquetes origen destino con ip 10.45.76.5. El resultado es que 10.45.76.5 al pretender visitar www.google.com, será direccionado a www.yahoo.com.

Observando los simples ejemplos anteriores, podemos imaginarnos la cantidad de posibilidades que nos ofrece este software, un software muy potente, pero que en manos de gente con objetivos malintencionados puede suponer un severo riesgo para la integridad de nuestra red.Por eso vayamos a mencionar alguna de las posibles soluciones que encontramos ante este tipo de ataques.

4.- Soluciones

4.1 ARPwatch

En sistemas Linux la herramienta **ARPWatch** (http://www-nrg.ee.lbl.gov/) nos puede servir para detectar el uso del envenenamiento ARP en nuestro sistema. Con **ARPWatch** podemos comprobar la correspondencia entre pares IP-MAC (Ethernet). En caso de que un cambio en un par se produzca (esto es, se escuche en el interfaz de red del sistema), **ARPWatch** envía un correo de notificación del suceso a la cuenta root o administrador del sistema con un mensaje tipo "FLIP FLOP o Change ethernet address" . También podemos monitorizar la existencia de nuevos host (aparición de una nueva MAC en la red).

4.2 ARPGuard (ISL y SECUDOS)

Sistema para formar una protección activa contra los ataques internos a la red. Arp-guard no interfiere con las aplicaciones internas de la red, y permite además reconocer posibles amenazas participando como observador, y con potestad de intervención en casos de un ataque eventual. Arp-guard analiza constantemente todos los paquetes ARP y envía las alertas oportunas a los administradores de red indicando el origen del ataque.

En adición, ARP-guard ofrece un sistema denominado arp-guard box que actúa como un analizador/observador del tráfico de paquetes ARP en la red local. Éste nació como la asociación estratégica entre ISL y SECUDOS, en donde la primera suministra el software y la segunda suministra el hardware.

Más información en www.secudos.de

4.3 Asignación estática en la tabla de entradas ARP

Como hemos visto, el ataque se basa en la expiración y posibilidad de modificación de las entradas en la tabla ARP. Si algo es vulnerable porque puede moverse, entonces taladrémoslo y fijémoslo al suelo.

Existen dos formas de añadir una entrada en la tabla ARP, de forma automática, esto es lo que hace nuestro sistema sin que nosotros nos percatemos, y manualmente lo cual nos añade la posibilidad de dejar inmutable y hasta el apagado del sistema dicha entrada en la tabla ARP.

La forma de añadir una entrada ARP en un sistema Win32 es la siguiente, desde el indicador de comandos ejecutamos:

arp -s < Dirección IP del Host> < Dirección MAC del Host>

De esta forma, las direcciones MAC habrían quedado grabadas a fuego y por mucho que el sistema B hubiese intentado perpetrar un ataque de ARP Poisoning este hubiese quedado en un intento. Además si hubiésemos tenido funcionando el software ARPWatch hubiésemos detectado el ataque y posiblemente al atacante.

Dado que la tabla ARP se elimina al apagar el PC y se encuentra vacía al encenderlo, debemos de crear un proceso que al encender el PC cree la tabla ARP como hemos comentado arriba, host por host crítico. Esto es factible en redes reducidas.

No obstante, en redes locales donde se disponga de sistemas Unix/Linux o cualquier otro multiusuario, y siempre y cuando los usuarios de la red no dispongan de la contraseña de administrador, será imposible (siempre hay métodos de obtención de cuenta administrador por otros medios) que alguien con nivel raso pueda lanzar ataques de este tipo, pues es necesario el nivel de administrador para acceder a servicios de la capa 2 de enlace. Aún así, hoy día existen algunos switches que son inmunes a este tipo de ataques, pero son los menos.

6. Conclusión

Hemos visto cómo un ataque tan peligroso puede ser llevado a cabo con significativa facilidad haciendo uso de una herramienta que está a disposición de todo el mundo. Si cualquier aficionado es capaz de logar escuchas protegidas y privadas, ¿qué no podrá hacer un experto en redes?. Muchos son los administradores de redes que adjudican tiempo y dinero a soluciones a nivel de red como pueden ser firewalls, firewalls-bridge, sistemas de detección y prevención de intrusiones (IDS, IPS respectivamente), etc... pero que son inútiles en este tipo de ataques, en donde el nivel más básico y bajo el cual se sustentan todos los demás niveles queda desprotegido ante cualquier aficionado. Así que no deje descuidada la red si desea mantener la privacidad de sus datos. En caso de pertenecer a una red en donde usted contemple que la seguridad queda en entredicho, no dude consultar al administrador y hacerle saber el problema. Por lo demás, tomemos siempre precauciones, es mejor ser desconfiado.

Sergio Valín Cabrera

7.- Bibliografía

www.google.com

http://es.wikipedia.org

www.monografias.com

http://www.rfc-editor.org/rfc/rfc826.txt

http://bulma.net/body.phtml?nIdNoticia=1193

www.secudos.de

http://ettercap.sourceforge.net

http://ditec.um.es/laso/docs/tut-tcpip/3376c28.html#figethfram

www.robota.net

www.irongeek.com

http://www.maestrosdelweb.com/editorial/sniffers/

http://www.xombra.com/go_articulo.php?articulo=47