Los sniffers y las redes Ethernet

6 de marzo, 2003

1 Características de una red Ethernet

Nos enfocamos a las redes Ethernet por ser la tecnología imperante en materia de conectividad. Las redes tipo Ethernet fueron creadas en los 70s, y su tecnología ha demostrado ser eficiente y extensible, al grado de que, partiendo de un estándar a 1Mbps sobre cable coaxial, hoy tenemos estándares de hasta 1Gbps, y las topologías derivadas de Ethernet ahora engloban desde el medio coaxial hasta topologías de estrella e inclusive inalámbricas.

Las redes Ethernet funcionan basadas en el método *CSMA-CD* (Carrier Sense Multiple Access - Collision Detection). Esto significa que cada nodo en una red Ethernet tiene la capacidad de detectar si está conectado a una red o no hay un enlace válido (*Carrier Sense*), y que el mismo medio físico es compartido entre varias computadoras (*Multiple Access*). Al tener un mismo medio compartido, dos computadoras podrían intentar transmitir datos a la vez, lo que llevaría a que ambos flujos de datos se corrompieran, por lo que se hace necesario que haya una detección de colisiones (*Collision Detection*) y un mecanismo de respuesta a las colisiones — En caso de haber una colisión, ambas computadoras esperarán un tiempo aleatorio e intentarán re-enviar sus paquetes.

Las redes Ethernet originalmente estaban conformadas por un sólo cable que conectaba, una a una, a todas las computadoras. Aún hoy, con los cambios topológicos que han sufrido, toda red Ethernet emula este comportamiento: Cualquier paquete que es enviado a la red llega a todos los nodos de la misma (excepto en las redes switcheadas, de las que hablaremos más tarde). Esto significa que cada computadora de la red tiene la capacidad de escuchar el tráfico dirigido a cualquier otra computadora de la red.

Procesar un paquete que llega por la red siempre supone trabajo para el sistema operativo. Es por ello que las tarjetas Ethernet por default no reportan al sistema operativo de paquetes que no estén destinados a esa computadora (dando explícitamente su dirección física o MAC) o a todas las computadoras de la red (enviadas a la dirección física de broadcast, 00:00:00:00:00:00). Para que el sistema operativo reciba todos los paquetes es necesario desactivar este filtro, lo que es conocido como colocar la interfaz en *modo promiscuo*.

Una vez que la tarjeta está en modo promiscuo, ésta entregará al sistema operativo todos los paquetes que pasen por su cable. Utilizando bibliotecas

como libpcap, programas en espacio de usuario pueden solicitar al kernel que les entregue todos estos paquetes para procesarlos y reportar al usuario los datos obtenidos de ellos — Esto es conocido como *sniffing* (*olfateo* en inglés).

Un segmento de red Ethernet que va creciendo en actividad presenta cada vez más colisiones, y su rendimiento cae de manera abrupta. Como las redes medianas y grandes son cada vez más comunes, a fines de los 90s comenzaron a popularizarse los *switches* — equipos de conectividad Ethernet similares a los concentradores que, en vez de enviar cada paquete a todas las computadoras del segmento, los envía únicamente al puerto donde está conectada la computadora destinatario.

Al aparecer los switches, todo parecía indicar que *sniffear* las redes sería ya imposible, a menos que fuera hecho desde el segmento donde estuvieran las computadoras en cuestión. Tristemente, esta ilusión no duró mucho tiempo, gracias al advenimiento del ARP spoofing/poisoning. Para entrar en detalles, veamos rápidamente cómo funciona el pegamento entre Ethernet y TCP/IP: El protocolo ARP.

Cada tarjeta de red Ethernet tiene un identificador de 48 bits único -supuestamente- en el mundo, llamado dirección MAC (Media Access Control). Las direcciones IP son direcciones de 32 bits, y no guardan relación alguna con las direcciones MAC.

Cuando una computadora intenta comunicarse con otra que debe estar -según su dirección/máscara IP- en la misma red que ésta, lanza un paquete ARP (Address Resolution Protocol) de tipo 'who-has', dirigido a todas las computadoras del segmento físico (con la dirección broadcast de Ethernet), con la IP de la máquina destino. A esta solicitud, la computadora dueña de la IP solicitada responde con un nuevo paquete ARP -ya en unicast- a la computadora que originó la solicitud, indicándole su dirección física. Después de esto, ambas conocen ya la relación entre MAC e IP necesaria, y pueden comenzar a enviarse paquetes IP.

Parte del diseño del protocolo ARP estipula que, si una computadora tiene registrada la relación IP <-> MAC de otra en su tabla de ARP y escucha un nuevo paquete ARP anunciando que la IP en cuestión está relacionada con otra ARP, debe olvidar la relación que tenía declarada y registrar la nueva. Por tanto, una computadora cualquiera en la red puede envenenar fácilmente las tablas ARP de las demás, recibiendo los paquetes destinadas a una computadora aún en otro segmento de una red switcheada, e inclusive actuar como proxy, logrando escuchar (e incluso intervenir) de manera completamente transparente la comunicación. Claro está, quien lo esté haciendo tendrá que cuidar el volver a envenenar las tablas ARP cada que haya una solicitud para mantenerse como escucha.

2 Usando los sniffers a nuestro favor

Los sniffers pueden ser una pesadilla para un administrador si son utilizados por usuarios no autorizados. Sin embargo, hay pocas herramientas tan poderosas

como estas para detectar problemas en nuestra red. Es indispensable para un administrador de sistemas el conocer al menos el funcionamiento básico de estas herramientas y utilizarlas como parte de su rutina cotidiana. Entre los más útiles encontramos a:

2.1 tcpdump

Uno de los sniffers más comunes — forma parte del sistema base de OpenBSD, está empaquetado para prácticamente todas las distribuciones de Linux y los otros BSDs, y está disponible para cualquier otro sistema Unix. Nos permite trabajar rápidamente desde línea de comando especificando los patrones que nos interesan, puede examinar una gran cantidad de protocolos, puede guardar el flujo capturado en un archivo o tomar un archivo como fuente para el flujo a analizar.

2.2 darkstat y traffic-vis

Diseñados para funcionar como proceso demonio, recolectando estadísticas de uso de la red. Ambos reportan sus resultados a través de una interfaz Web, un reporte Postscript u otros formatos.

2.3 ngrep

Tiene una filosofía de uso muy similar a la del comando 'grep' de Unix, tomando como entrada el flujo de la red en vez de archivos locales.

2.4 snort

Muy completa herramienta de detección de intrusos en red, toma como entrada el tráfico capturado en una red y lo va comparando cun una serie de reglas, registrando cualquier tráfico sospechoso de llevar un ataque. Snort únicamente lo registra, pero puede trabajar en conjunto con otras herramientas (hogwash, ACID, etc.) para sanear el tráfico, bloquear a la máquina atacante, generar reportes, etc.

2.5 nwatch

Formalmente es un sniffer, pero es más bien una herramienta para realizar lo que sus autores definen como barridos de puertos pasivos: Para detectar puertos que están abiertos por muy cortos periodos de tiempo y para no mostrar actividad sospechosa de barrido, nwatch se queda escuchando la actividad de la red, y manteniendo una lista de qué hosts proveen qué servicios.

2.6 ethereal

Un magnífico sniffer con interfaz gráfica de usuario, nos brinda un análisis completo y detallado de cada paquete a varios niveles, desde nivel Ethernet hasta

detalles de diversos protocolos. Es capaz de convertir en adicción el comprender cómo funcionan determinados protocolos ;-)

2.7 ettercap

Pocas de estas herramientas funcionan adecuadamente en redes switcheadas. Ettercap utiliza técnicas más de sombrero negro, como el ARP spoofing/poisoning, para permitir sniffear redes switheadas. Además de sniffer es interceptor (permite inyectar datos en conexiones existentes o "secuestrar" conexiones).

2.8 kismet

Sniffer específico a Linux para redes inalámbricas. Funciona correctamente con los dos principales tipos de tarjetas inalámbricas.

3 Detectando sniffers en mi red

Los sniffers, claro está, pueden ser un grave problema si tenemos usuarios de nuestra red recolectando información de contraseñas, de archivos personales o, en general, cualquier tráfico no destinado a ellos. Es posible detectar máquinas que estén sniffeando, aunque no es un proceso completamente automático y está a merced de los caprichos del sistema operativo que queramos localizar. En general, las técnicas utilizadas para la detección de sniffers parten de que, para olfatear tráfico en red, una computadora debe poner su interfaz de red en modo promiscuo — Deshabilitar un filtro en hardware diseñado para ahorrar carga al sistema operativo, que descarta todos los paquetes que no estén dirigidos a esa tarjeta en particular o a la dirección MAC de broadcast (00:00:00:00:00:00).

Al recibir el sistema operativo un paquete no destinado a él, antes de entregarlo a libpcap para que lo procese y reporte al sniffer (o el proceso equivalente), lo pasa por su propia pila para ser procesado. Por lo tanto, si enviamos un ping a una dirección IP con la dirección MAC equivocada o si hacemos una solicitud ARP para una dirección IP sin que vaya en un paquete broadcast y recibimos respuesta, es clara indicación de que esa computadora está corriendo un sniffer.

Claro está, esto no siempre es fidedigno — Hay varios métodos más, hay varios programas que nos ayudan a hacer estas búsquedas, y hay varias herramientas para ayudarnos a esconder la presencia de un sniffer (inclusive herramientas físicas — un cable de red con los cables de recepción correctos pero los cables de transmisión cortados es indetectable por definición) el buscar sniffers poco softisticados en nuestra red es también una muy importante obligación de los administradores de sistemas.