Packet Sniffing

Transcribed on August 1, 2025 at 5:27 PM by Minutes AI

Speaker 1 (00:02)

Bienvenidos a esta nueva sesión.

En esta sesión vamos a tratar el tema del packet sniffing o interceptación de paquetes y veremos cómo utilizarlo de una forma práctica con TCP Dump y T Shark.

El packet sniffing es una técnica utilizada para capturar y analizar el tráfico de datos que fluye a través de una red, permitiendo a los administradores de red o a los atacantes examinar el contenido de los paquetes de datos, incluyendo información como direcciones IP, protocolos utilizados y también los datos transmitidos.

Esto se puede utilizar con fines legítimos, como la depuración de redes o la monitorización de tráfico, pero también puede ser aprovechado para propósitos maliciosos como la interceptación de contraseñas o información confidencial.

El modo promiscuo en tarjetas de red es fundamental para el paquete sniffing, ya que permite que la tarjeta capture todos los paquetes de datos que pasan por la red, independientemente de si están destinados a la tarjeta receptora o no.

Esto es crucial para realizar un análisis exhaustivo del tráfico de red y para detectar posibles amenazas o anomalías.

Sin el modo promiscuo, la tarjeta de red solo capturaría paquetes dirigidos específicamente a ella, limitando significativamente la capacidad de monitoreo y el análisis de tráfico de red.

Por lo tanto, el modo promiscuo permite a los administradores de red y a los investigadores de seguridad obtener una visión completa y precisa de la actividad de red, lo que es esencial para mantener la seguridad y el rendimiento óptimo de la infraestructura.

Este esquema ilustra de forma genérica cómo funciona el packet sniffing.

Podemos ver diferentes protagonistas de todo este proceso, como pueden ser los routers, los switches, pero fijaros también que tenemos lo que es el traffic mirror.

El traffic mirror es la técnica de hacer espejo o duplicar el tráfico de red, lo que hace que tengamos una copia exacta de los paquetes de datos que se envíen a un segundo dispositivo.

Esto es muy útil tanto para la administración de redes como para el análisis forense.

Después, el paquete sniffer es un software o un dispositivo que captura los paquetes de datos que pasan a través de la red.

Estos analizadores de paquetes pueden ser utilizados por administradores de red para monitorizar y diagnosticar problemas de la red o por atacantes para recopilar información sensible.

Malicious eavesdropping es la acción de interceptar secretamente el tráfico de la red con intenciones maliciosas.

Este tráfico, que se ha aplicado a la técnica de mirroring de espejo, se dirige al PC del atacante donde el sniffer de paquetes está activo.

Es vital para packet sniffing entender perfectamente las tramas de red o los paquetes.

No voy a entrar en profundidad en cómo funcionan los paquetes, pero aquí os dejo de ejemplo uno de ellos que es el TCP Header, que es la cabecera del TCP en la que vemos los diferentes componentes que lleva, como puede ser el source port, que es el puerto de origen del paquete, el destination port que es el puerto destino, el número de secuencia, etcétera.

También vemos puntos clave como son los flags, que son los diferentes parámetros como CWR, EC, URG, ACK, etc.

Que son los que nos pueden dar una mejor, digamos, versatilidad para configurar algún tipo de detección o entender bien lo que está ocurriendo en esta acción o en esta configuración.

El checksum también es un campo muy utilizado ya que nos certifica que no ha habido ningún error durante la transmisión y muchos más.

Pero insisto, es importante conocer bien cómo funcionan las tramas de los paquetes para entender el packet sniffing porque si no, no vamos a saber interpretar la información.

No podemos hablar de packet sniffing sin nombrar a Wiresh.

Wireshark es una herramienta de análisis de red de código abierto que además es gratuita y se utiliza muchísimo para la captura y el análisis detallado de tráfico.

Permite a los usuarios inspeccionar datos de red a nivel de paquete, lo que permite que se desglose de una forma sencilla para visualizarlos y diagnosticar problemas.

Wireshark soporta cientos de protocolos y tipos de medios, ofreciendo filtros potentes y la capacidad de inspeccionar y visualizar la jerarquía de paquetes y sesiones.

These notes were taken with Minutes AI (https://myminutes.ai)

Además tiene una interfaz gráfica y esto posibilita trabajar con archivos de captura que se pueden visualizar, lo que lo hace una herramienta totalmente fundamental para cualquier tipo de trabajo relacionado con el package sniffing.

TCP Dump y TestShark son dos herramientas de análisis de paquetes muy sencillas, fáciles de instalar, pero que ofrecen una gran y potente cantidad de comandos y opciones relacionadas con el paquete Sniffing CCP Dump es un potente y muy utilizado analizador de paquetes en sistemas operativos similares a Unix.

Se usa muchísimo en la depuración de tráfico, pero también se puede utilizar para monitorizar la seguridad de la red, ya que permite guardar paquetes capturados en un archivo para luego analizarlo.

Y T shark es esencialmente Wireshark, lo que pasa que esta es la versión en línea de comandos.

Ofrece muchas de las mismas características que Wireshark para capturar y analizar los paquetes de red, pero no lleva la interfaz gráfica.

Por ese motivo, T shark es útil para tareas automatizadas o entornos donde no está disponible una interfaz gráfica, como por ejemplo servidores remotos.

Bien, pues lo mejor para entender cómo funciona el paquete sniffing es hacer una prueba práctica, y para ello utilizaremos otra vez las dos máquinas virtuales con las que hemos estado trabajando en estos últimos ejercicios.

De nuevo tengo la máquina 1 que acaba en la IP y tengo la máquina 2 que acaba en la IP de 10211.55.

Como siempre, lo primero es asegurarnos que tenemos TCP Dump instalados, con lo cual hacemos un sudo apt install tcp dump y ya lo tenemos.

Y haremos lo mismo con instalamos esta opción, de momento no nos interesa y ya continuamos.

Ya tenemos C y TCP.

Bien, para probarlo nos vamos a generar un poco de tráfico.

Entonces lo primero que voy a hacer es un pin 55.17 para comprobar que todo está correcto y que se puede hacer.

Como tengo comunicación me voy a ir a la máquina 2, que va a ser la que va a interceptar todo, toda la información.

Bien, pues aquí en la máquina 2, que también tengo instalado TCP TAM y T Shark y que tiene la dirección IP que acaba en 17, vamos a lanzar TCP TAM para que empiece a capturar toda la información.

These notes were taken with Minutes AI (https://myminutes.ai)

Con lo cual haremos un sudo tcp dump i Y aquí ponemos de dónde a dónde queremos capturar todo el tráfico, sin importarnos ni protocolo ni nada, queremos todos los paquetes, pues le diremos any, le pondremos host 10211 55.5 que es la otra máquina y también nosotros que es el 10211.55.17.

Cerramos, le ponemos W que será el fichero donde se va a guardar y llamamos captura PK que es el formato típico de las capturas de package sniffing.

Bien, lo ejecuto y lo dejamos aquí trabajando.

Está escuchando cualquier tráfico que hay en la red entre esos dos hostias.

Bien, pues me voy a la otra máquina para generar un poco de tráfico y que TCP vaya almacenando información.

Nos vamos a dar la otra máquina y lo primero que voy a hacer es otra vez otro ping, mi preping de antes, pero esta vez este ping se va a quedar registrado en la otra máquina que tiene el TCP, con unos cuantos ya no sirve.

Lo siguiente que haremos será un netcat, que es para la transferencia de archivos.

Pues bien, he vuelto a la máquina número 2 porque ahora voy a poner a netcad escuchando en un puerto para recibir un fichero, porque también quiero que TCP dump capture esa información.

Con lo cual lo que voy a hacer es abrir otra ventana del terminal y aquí voy a hacer.

Vamos a ponernos aquí para que no moleste.

Haremos un netcat que será ebt y pondremos el puerto en el que va, por ejemplo este puerto y le decimos que aquí reciba un fichero que se va a llamar archivo recibido txt y ya lo dejamos en escucha.

Bien, tenemos dos cosas aquí trabajando ahora mismo.

TCP por un lado, capturando toda la información y netcat esperando un fichero desde la otra máquina al PC número 1.

Y desde aquí vamos a enviar el fichero.

En este caso voy a enviar a la 102.1155.17, que es la IP de la otra máquina, al puerto, hemos dicho, al puerto 1, 2, 3, 4 y 5.

Y le voy a enviar el fichero.

Por ejemplo tengo uno que se llama planets, que es un CSV.

Pues lo hacemos.

These notes were taken with Minutes AI (https://myminutes.ai)

Bien, pues estamos ya enviando el fichero.

El problema con netcat es que depende de la versión y también un poco de los parámetros.

Es posible que no separe y se quede aquí siempre escuchando aunque haya enviado el fichero.

En este caso el fichero lo ha enviado seguro.

Vamos a ir a ver el segundo servidor.

Bien, aquí lo tenemos, pero si yo cancelo ahora y le hacemos un cat al archivo recibido, está completo, ¿Veis?

Es lo que tiene contenido.

Este contenido es el mismo que podemos ver en el otro servidor en el mismo fichero.

Vamos a verlo.

Completo, ¿Veis?

Es lo que tiene contenido.

Este contenido es el mismo que podemos ver en el otro servidor en el mismo fichero, vamos a verlo, hacemos un cut a planets, es exactamente igual, se ha transmitido bien, lo que queríamos era simplemente hacer una transferencia de ficheros no cifrada, porque netcat no cifra, por eso lo hemos hecho, para ver que podemos ver la trama de paquetes.

Así que bueno, en este momento hemos enviado un fichero, hemos hecho un ping, hemos creado algo de tráfico para ahora analizarlo.

Bien, pues vamos a volver otra vez a la máquina que está escuchando ahora mismo, vamos a cerrar esta ventana que ya no nos hace falta y vamos a parar el TCP da, fijaros, ya nos pone que ha habido paquetes capturados, ahora quedaría analizarlo, recordad que tenemos un fichero que se llama captura cap, que es donde hemos almacenado todos esos paquetes que he ido capturando durante toda la sesión.

Bien, para poder verlo podemos utilizar el siguiente comando con T shark, pues podemos utilizar tsark y ponerle fichado que es captura cap, decimos que yes y ahora utilizamos este comando, y ahora os lo explicaré addr y le ponemos igual igual a la 10 211 55 5 and ip adr la otra, bien, 211 55 17 vale, ¿Que hemos hecho ahí?

El guión es recaptura, lo que le indica a t shark que lea los datos del archivo que hemos especificado, en este caso el captura pk y con toda esa cadena de ipadr con las dos ips, es un filtro de visualización que le está diciendo a T shark que muestre solo los paquetes que involucran a ambas direcciones IP, o sea, IP alberte se refiere a cualquier dirección IP, ya sea fuente o destino, pero tiene que coincidir con esos valores.

These notes were taken with Minutes AI (https://myminutes.ai)

Bien, pues al pulsar enter ya podemos ver aquí todos los paquetes que he ido capturando.

Bien, aquí se pueden visualizar varias cosas.

Primero podemos ver que los paquetes del 1 al 14, si no me equivoco hasta aquí, los paquetes del 1 al 14 son los del ping, porque podemos ver la dirección de IP de origen y el tipo de mensaje ICMP, como podemos ver aquí.

Después tenemos la ID de secuencia, lo típico, y el TTL.

El TTL es el time to live, TTL time to live nos indica cuántos saltos puede hacer un paquete antes de ser descartado.

Del 17 al 18 lo que podemos ver aquí es un inicio de sesión y una transmisión de datos.

Este es el NETCAT.

Fijaos que hay un inicio de sesión TCP con un SYNC, que es un inicio de conexión, seguido de un SYN ACK, que es un reconocimiento de conexión.

Entonces aquí la información relevante incluye números de secuencia, opciones TCP como MSS, que es el Maximum Segment Size, el SAC PERM, que es el Selective o acknowledgement Permitted y el WS que es el Windows Scale.

Son todos flags asociados a ese paquete y ese tipo de conexión.

El paquete 19, si os fijáis, es el ACK, es decir, ACK nos está diciendo que se ha completado el proceso de three way handshake de tres vías, que es vital en una conexión de TCP.

Bien, los paquetes del 20 al 27 que están aquí, a ver si los marco bien hasta aquí, y también del 42 al 47, aquí también se admite la secuencia, son los datos que se han enviado, porque tiene la bandera push PSH que podéis ver aquí.

Aquí también está aquí abajo tienen PUSH y ACK, lo que indica la transmisión de datos dentro de la sesión TCP que hemos establecido con NETCAT.

Bien, pues los números de secuencia y el acuse de recibo, que es el AFK, indica el orden y la confirmación de los datos transmitidos.

Los paquetes del 32 al 34 que están aquí, son los que indican la finalización de la sesión TCP con un intercambio de fin de conexión.

Hay después un ACK y un ACK final, con lo que hace cierra todo el proceso.

Bien, pero aún podemos hacer más cosas con C para que nos muestre más información.

Con esto ya tenemos muchísima, hemos visto bastante información de lo que ha ocurr, el ping, la transmisión de datos, etc.

These notes were taken with Minutes AI (https://myminutes.ai)

Pero podemos todavía afinar más y mostrar más información.

Así que vamos a probarlo.

Vamos a probarlo.

Bien, podemos todavía afinar más.

Podemos, por ejemplo, conseguir toda la información que se ha transmitido.

Eso también está almacenado en el PCAP.

Por ejemplo, podemos ver lo que hemos capturado utilizando el siguiente comando con T sharp, por ejemplo, podemos ver toda la información que hemos capturado en formato hexadecimal con T sharp R le ponemos el fichero que da captura, pcad y y decimos que utilice el puerto TCP port y le ponemos el puerto que hemos utilizado para la captura, para que lo filtre dentro del BK.

Después le ponemos T y le ponemos field.

Ahora os cuento lo que es cada cosa.

Data hacemos un pipe y le ponemos xxd R p.

El XXD que tenemos aquí simplemente lo hemos utilizado para que lo convierta el código a ASCII y no nos devuelva todo en hexadecimal.

A ver qué ocurre con esto.

Es exactamente lo que hemos capturado durante la transmisión.

De hecho, si le quitamos el xxxd veréis que aparece todo en código hexadecimal, con lo cual de esta forma tenemos toda la información que se ha transmitido por el PCAP que era el fichero planet CSV.

Bien, como podéis observar, TCPDAM junto a T Sharp son una combinación muy potente para el paquete sniffing.

Y fijaros, todo desde la línea de comandos sin interfaz gráfica.

Quería hacerlo así porque lo normal es que utilicéis servidores y no equipos con una interfaz gráfica para utilizar este tipo de información o este tipo de actividades de paquete sniffing.

El paquete sniffing desempeña un papel fundamental en la seguridad de la red al proporcionar a los administradores y equipos de seguridad una visibilidad muy grande sobre el tráfico de datos en la red.

These notes were taken with Minutes AI (https://myminutes.ai)

Esta técnica nos permite detectar actividades maliciosas, identificar vulnerabilidades de seguridad y realizar análisis forenses en tiempo real.

Al capturar y analizar paquetes de datos, el packet sniffing ayuda a las empresas y organizaciones a prevenir intrusiones, mitigar amenazas y responder eficazmente a incidentes de seguridad.

En resumen, el pack sniffing es fundamental para fortalecer la seguridad de la red y mantener un entorno de IT seguro y protegido contra las crecientes amenazas que tenemos en Internet.

Llegamos al final de la sesión.

Os esperamos en el siguiente vídeo.

These notes were taken with Minutes AI (https://myminutes.ai)