Network Sniffing

Transcribed on July 14, 2025 at 4:06 PM by Minutes AI

Speaker 1 (00:08)

Hola a todos y bienvenidos a esta última sesión donde vamos a estar hablando de qué es el sniffing y de algunas herramientas para realizarlo.

Hablaremos de Wireshark, de TCPDAN, de T Shar.

Vamos a ello.

Lo primero que haremos en esta sesión será presentaros el entorno de laboratorio con el que vamos a trabajar al final de la misma.

Luego veremos qué es el sniffing, los tipos de sniffing que existen y terminaremos viendo las herramientas para hacer sniffing que comentá Wireshark, TCPDAN y TEESAR.

Por último, iremos a este laboratorio que hemos presentado anteriormente y realizaremos algunas pruebas con las diferentes herramientas de pnifin.

Comenzamos entonces presentando el entorno de laboratorio.

Por un lado tengo una máquina Kali.

Esta máquina está en la dirección IP diez cero punto uno doce y y la red es la uno cero punto cero punto uno cero dos cuatro.

En esta misma red tengo otras dos máquinas, la máquina 1 y la máquina 2.

La dirección IP de la máquina 1 es la diez cero uno tres y la dirección IP de la mÁquina 2 es la diez cero unO catorce.

Por tanto, estas tres máquinas están en la misma red y con ella realizaremos tráfico para posteriormente detectarlo utilizando una herramienta de sniffing.

Vamos entonces a ver qué es el sniffing.

Se denomina sniffing al proceso de capturar y monitorear una red para posteriormente analizar los datos de los paquetes que están viajando entre los dispositivos de dicha red.

El sniffing no únicamente tiene connotaciones negativas como el espionaje, sino que también permite a los administradores de redes la búsqueda de problemas o realizar análisis de seguridad.

Lo que estamos capturando al hacer sniffing son los paquetes de datos que se transmiten en la red.

Estos paquetes, como ya hemos ido viendo, contienen el payload, los datos de ese paquete y las múltiples cabeceras que nos indican de dónde y hacia dónde va ese paquete.

Por tanto, existen dos tipos de sniffing.

Tenemos el passive sniffing y el Active sniffing.

En el sniffing pasivo, el atacante o la persona que realiza la captura de los paquetes de la red no interactúa en ningún momento con el sistema o red, sino que pasivamente lo que tratará será de escuchar el tráfico sin generar ningún tipo de paquete adicional.

El caso contrario es el active sniffing.

En este tipo de capturas el usuario tiene que realizar ciertas interacciones con la red para empezar a capturar de alguna forma.

¿Vale, y de qué forma hacemos esto?

Existen varias técnicas.

La primera de ellas de la que vamos a hablar es el modo promiscuo.

Esta se refiere a la capacidad de una interfaz de red para capturar todos los paquetes de datos que pasan a través de ella, independientemente si están destinados o no a esa interfaz en particular.

Esta capacidad es utilizada tanto en el Active Sniffing como en el Passive Sniffing.

Luego tenemos otras dos técnicas más, DNS spoofing y RPSpoofing.

Estas dos técnicas sí que están más centradas para el Active Sniffing.

Vamos a comenzar por la primera, DNS Spoofing.

Aquí lo que hacemos es confundir al host objetivo para que en lugar de enviar la petición al destino, nos la envíe a nosotros.

Nosotros veremos de qué se trata y ya la redireccionaremos a su destino y retornamos la respuesta.

Luego tenemos el ARP Spoofing.

Aquí el atacante lo que hace es enviar mensajes de tipo ARP falsificados a la red, asociando su dirección Mac con la dirección IP del otro dispositivo, como la del router.

De esa forma todos los paquetes se redirigen hacia el atacante.

En la infografía podéis ver cómo es la interacción en una red de tipo ARP, donde un juego se presenta y Oye, yo soy el juego, 10 puntos de enlace de esta red y y mi Mac es la los demás host lo conocen, se enteran y guardan la puerta de enlace del host equivocado, cuando realmente esa no es la puerta de enlace hacia Internet y por tanto todo el tráfico va a pasar por dicho host.

Por último tenemos el DHCP Sniffing.

Aquí lo que hace un atacante es monitorear el tráfico de HCP en la red para obtener información sobre direcciones IP disponibles, direcciones Mac de dispositivos y la configuración de la red.

Esto al final lo que puede permitir al atacante es realizar otros ataques basados en la información obtenida a partir de este sniffing.

Una vez visto esto, vamos a hablar ya de las diferentes herramientas que tenemos.

Comenzamos por Wireshark.

Wireshark es un sniffer de paquetes de red de código abierto y con una interfaz gráfica super completa.

Su uso es muy diverso, desde comprobar paquetes caídos, problemas de latencia o chequear si hay actividad maliciosa en la red, por ejemplo, viendo que ips consulta una determinada aplicación.

Al arrancarse la aplicación es muy completa, podemos ver todo tipo de falletes, tanto la petición como la respuesta y el antes y después de la comunicación.

Nos permite aplicar filtros para ver únicamente el tráfico deseado y nos crea estadísticas del escáner realizado.

Podemos exportar al final la captura de red que hemos realizado en diferentes formatos, siendo el más común la captura de tipo PCAP y PCAPNG.

Por supuesto, esta aplicación también nos sirve para abrir capturas de red pasadas y volver a analizar e investigar las mismas.

Imagínate que te has descargado una captura de tipo PCAP y la quieres abrir para realizarla, pues podrías utilizar Wireshark para esto.

Pasamos a hablar de TSAR.

Wireshark cuenta con una herramienta de líneas de comando para ejecutar funcionalidades llamada TSAR.

Es similar al terminal de Linux.

Entre los comandos más destacados podemos mencionar rowsart, editcap, mercap, testpcap.

En general hay muchos que iremos viendo después.

Por supuesto, al igual que un Wireshark, en el caso de C se puede realizar un análisis en tiempo real o investigar una captura guardada previamente.

Estos son los conocidos ficheros pcap y pkpng que comentábamos.

Además la instalación de tear si no lo tienes en tu máquina es muy sencilla y basta con hacer únicamente un sudo aptstall t shark Por último vamos a ver esta herramienta llamada TCP DUN que es muy similar a T Sharp.

Esta herramienta es muy utilizada en entornos Linux que no disponen de entorno gráfico para trabajar y también es open source.

Es muy buena opción si la quieres utilizar por ejemplo en un servidor en el que no tienes una interfaz gráfica para poder utilizar Wireshark.

Una vez hecha esta introducción vamos a pasar al laboratorio y vamos a estar trabajando con cada una de las herramientas que hemos comentado.

Pues ya estamos en el entorno de laboratorio, fijaros por aquí y Wireshark viene instalado por defecto en Kali.

Voy a abrirlo y por aquí lo tengo ya listo para usarlo.

Antes de usarlo lo que voy a hacer es generar por aquí un servidor rápidamente voy a hacer un python m HTTP server y finalmente le indico el puerto que va a ser el 80.

Escribo por aquí kali y ya estamos listos y a la escucha en ese servidor para qué es esto para posteriormente, con las máquinas 1 y 2 que tenemos por aquí listas, vamos a realizar algunas peticiones a este servidor y así generar algo de tráfico.

Volvemos a la máquina Kali y vamos a empezar ya a capturar ese tráfico.

Voy a capturar el tráfico de la interfaz de red ETH, clico por aquí y ya estamos escuchando.

Voy a hacer una petición a esta máquina, que es la uno cero cero uno doce desde mi máquina 1.

Para ello hago por aquí un diez cero uno doce y fijaros por ahí que ya estamos capturando el tráfico.

La petición que he hecho ha sido de tipo ICMP, es decir, un echo Request, lo podéis ver por aquí y mi máquina ha respondido con un echo reply.

Además, fijaros que la máquina no sabía qué dirección Mac tenía la dirección diez cero uno doce, por tanto necesita hacer una RP Request para conocer dicha dirección.

Voy a borrar por aquí la captura y vamos ahora desde el navegador de Kali a realizar una petición a por ejemplo.

Fijaros que nada más abrir el navegador ya estamos realizando un montón de peticiones.

Nosotros vamos a hacer una a Sample por aquí, Run, damos a Enter y fijaros que por aquí se ha cargado.

Voy a parar el escáner y nosotros lo que queremos es el tráfico de tipo HTTP.

Filtro por aquí y fijaros aquí al final tengo la petición con hipertexto y aquí tengo el host EXAMPLE COM que acabo de solicitar la respuesta de dicho host, que si unificáramos esto, pues podríamos ver la respuesta de la página en código HTML que nos ha devuelto.

Y finalmente el navegador por defecto siempre lanza un getFabicon.

Ico para obtener el favicon de esa página.

En este caso me ha dicho que NOT FOUND, por tanto aquí en Firefox fijaros que no hemos conseguido encontrar ningún favicon para dicha página.

Vamos ahora a intentar utilizar TCPEDAN, Fijaros que la tenemos por defecto instalada en nuestra máquina.

Si doy a Enter me dice que no tengo permiso para realizar una captura en el dispositivo, por tanto entendemos que necesitamos lanzarla con permisos de administrador.

Escribo para tisudo dcpdan, pero antes voy a hacer un H.

Fijaros que tenemos las diferentes Account Interface, podemos indicarle en la interfaz queremos escuchar diferentes tipos para filtrar el fichero donde queremos guardar en general diferentes opciones que nos permite TCPDAMP.

Nosotros lo vamos a lanzar sin ninguna opción y por defecto se va a poner a escuchar en la interfaz eth.

Voy a limpiar por aquí, limpio la terminal y ejecuto.

Ya estamos escuchando y como podéis ver estamos capturando ciertas peticiones.

Voy a lanzar por aquí para recargar la página y fijaros que ya se ha cargado y por aquí podremos ver que se ha lanzado una petición a Example Run y por aquí la hemos capturado.

Al final estamos utilizando una interfaz, una interfaz por línea de comandos y no es tan bonita o tan visual como era la de Wireshark y por tanto puede costar aquí más ver algunas cositas.

Lo que sí es útil es que esta aplicación, esta salida lo podríamos capturar y guardar en algún formato para posteriormente analizar con algunas herramientas o con algún script.

Por último vamos a ver rápidamente T Shar, vamos a ver si lo tenemos instalado en nuestra máquina y fijaros si ya lo tenemos y ya estamos capturando rápidamente en la interfaz de eth.

Voy a recargar la página de Sample Chrome y fijaros como aquí me ha capturado y me ha dicho que acabamos de solicitar la RAID de HTTP y me ha respondido que no se ha modificado la página y por tanto no me ha devuelto nada.

Ahora voy a hacer una petición con la máquina Ubuntu Server de nuevo a esta máquina diez cero uno doce a la máquina Kali y fijaros por aquí cómo la estamos capturando.

Lo que estoy haciendo con la máquina es simplemente un pin y por aquí vemos el eco pin request, el eco pin reply y las diferentes peticiones que ha hecho.

Fijaros que lo he hecho con la máquina diez cero uno tres.

Voy a hacerlo ahora con la máquina diez cero uno catorce, lo mismo lo lanzó por aquí y fijaros ya lo estamos volviendo a capturar esta máquina, fijaros por aquí, la diez cero un catorce, no tenía ni idea antes de hacer ese corequest de cuál era la dirección Mac de la máquina diez.

Cero uno punto uno doce.

Es por ello que ha dicho por aquí oye, ¿Quién tiene la dirección IP?

Y nuestra máquina ha respondido y ha dicho, oye, la 10.0.1.1 12 tiene la siguiente dirección Mac.

Una vez que tenía esta dirección Mac, pues ya mi máquina a las diez cero uno catorce estaba lista para solicitar ese echo request y comprobar si la máquina Kali en este caso estaba levantada y activa y funcionando.

Y como podemos ver, nuestra máquina Kali está respondiendo por aquí, lo que indica que sí que está activa y es trazable dentro de la red.

Y con esto llegamos.