9ª Sesión práctica ASRC - 2017/18. Guión de trabajo:

ESCUCHANDO LA RED

Dentro del proceso de securización de una red, es fundamental comenzar por la recolección y análisis del tráfico circulante por el escenario en el que nos estemos manejando.

En este sentido, tenemos una serie de herramientas que nos proporcionan, de una forma u otra, la capacidad de observar aquello que circula por nuestra red.

Tcpdump

Retomamos la utilidad para la captura y análisis de paquetes, tanto de tipo ICMP, UDP como TCP.

Básicamente su uso elemental era algo así:

Para capturar tráfico y enviarlo a la pantalla:

sudo tcpdump –n –i <interface> -s <longitud>

-n o –nn: evita resoluciones DNS sobre nombres y puertos, agilizando la captura de tráfico. Si usamos sólo –n, no traduce el nombre, pero sí muestra el puerto, si no queremos que se muestre, se usa -nn

-i: indica la interfaz a observar.

-s <longitud>: indica qué parte del paquete debe registrar. Habitualmente un valor de 1515 (1.515 bits) suele ser suficiente

Podemos filtrar la captura de tráfico para sólo observar que contenga como destino u origen, un equipo concreto, por ejemplo mediante *tcpdump –n –i ethX –s 1515 | grep 192.168.0.X*

Podemos enviar el resultado de dicho comando a un fichero (opción -w), en lugar de la salida estándar, en tal caso tendría formato libcap (librería estándar de captura de paquetes usada por diversas herramientas de captura).

Este tipo de fichero luego puede ser analizado desde la herramienta Wireshark, que veremos a continuación.

Nmap y Zenmap: como ya se estudió en la sesión 2, podemos usarlas también para explorar redes y auditar la seguridad de las mismas, posibilitando determinar qué clientes hay disponibles en la red, así como los servicios (nombre de la aplicación y versión), sistema operativo, si existe cortafuegos, y otras características.

http://nmap.org/
http://nmap.org/nmap_doc.html

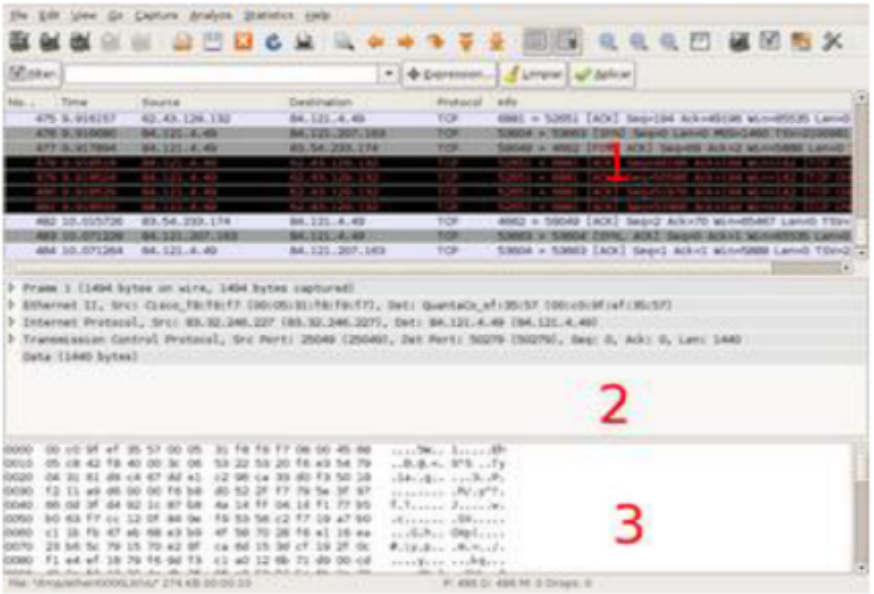
Wireshark

Analizador de tráfico de protocolos de red (llamado habitualmente sniffer), con ciertas funcionalidades añadidas que la convierte en una herramienta más polivalente, analizando y solucionando problemas de red. De ayuda también en el desarrollo de software y protocolos, así como herramienta didáctica para educación.
Licenciado bajo GPL, está disponible para GNU/Linux, Unix-like, Windows y Mac OS X.

Se aconseja leer el documento pdf del INTECO (Instituto Nacional de Tecnologías de Comunicación), poniendo especial interés en la sección en la que se habla sobre la ubicación desde donde se realizaría el análisis del tráfico (apartado 3).

Wireshark es de libre distribución y lo podéis descargar de su página Web:
http://www.wireshark.org/download.html

El proceso de instalación, en caso de Windows, es muy sencillo, debéis marcar la opción de instalar Winpcap, aplicación necesaria para que Wireshark pueda capturar los paquetes.
Al **ejecutar** Wireshark, es imprescindible hacerlo **como administrador**, para poder tener acceso a las interfaces de red.
En la ventana principal de la aplicación se mostrarán los paquetes capturados. Wireshark muestra la información capturada en tres secciones principales.

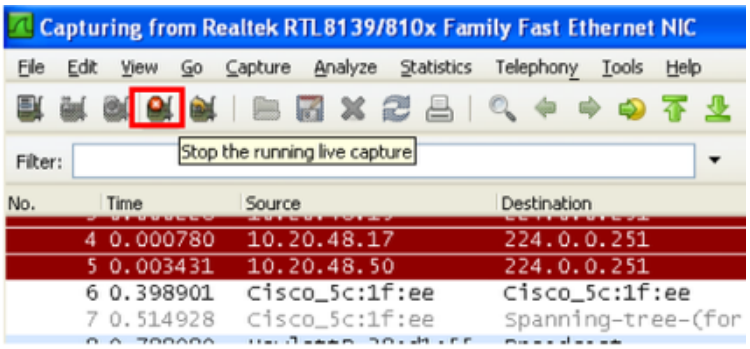


En la primera sección aparece un listado de los paquetes capturados con su información más relevante.
En la segunda sección podemos observar los detalles del mensaje seleccionado en la sección 1.
En la última sección se muestran los paquetes en bruto, es decir, tal y como fueron capturados por la tarjeta de red.

Para comenzar la captura:

Menú Capture->interfaces, y en la ventana con las tarjetas de red disponibles, seleccionar sobre la tarjeta con la que se realice la captura y pulsar Start. En el ejemplo de la figura, se selecciona el cable de red Ethernet.

Es necesario pulsar el botón *Stop the running live capture* para detener la captura de paquetes, y a continuación se pasa a examinar los paquetes capturados.



Al igual que sucede con tcpdump, esta herramienta dispone de dos tipos de filtros para examinar con mayor exactitud el tráfico que nos interesa. Por una parte tenemos los filtros de capturas (que hacen que se tomen sólo los paquetes que cumplan los criterios fijados) y por otra parte, los filtros de pantalla, que se aplican sobre el tráfico ya capturado.

Acerca de esta herramienta, disponemos de diversos recursos:

Ficheros demos con ejemplos de capturas para cargarlos con Wireshark:
http://wiki.wireshark.org/SampleCaptures

Ejemplo uso para detectar posibles ataques:

http://www.welivesecurity.com/la-es/2013/01/28/uso-filtros-wireshark-para-detectar-actividad-maliciosa/

Observad también el documento que se acompaña en el campus, acerca del uso de filtros, llamado: trabajando_con_Filtros_wireshark.pdf

Ejercicio 1 :

Desde una consola ejecuta la orden ping a una url externa y captura los paquetes que lleguen a tu tarjeta de red durante 10 segundos. Observa los paquetes que has capturado. Comenta qué ves sobre los paquetes generados a partir de la orden ping.

Comenta brevemente algunos de los filtros (de captura o de pantalla) que considere más interesante. Para ello, puedes tomar los datos de los ficheros “demos” o bien, aquellos capturados por tí mismo, ya sea por el apartado anterior u otro que estime interesante (por ej. tráfico derivado de conexiones web, etc)

A partir de Windows 7, podemos usar un comando para capturar el tráfico sin tener que instalar herramientas adicionales, se trata de ***netsh trace***. Use la ayuda para hacer alguna prueba de

dicho comando (netsh trace help ; para ver opciones de comienzo de escaneo: netsh trace start ?)

Una vez obtenido el tráfico, puede ser analizado a través de la herramienta gratuita *Microsoft Message Analyzer*.

Ejercicio 2 (opcional):

Realice una captura similar a la del ejercicio 1, pero usando esta otra aplicación comentada.

Cabe mencionar estas otras herramientas para análisis de red también:

a) SteelCentral Packet Analyzer

Es la única herramienta de análisis de red con plena integración con Wireshark, permitiendo a los que la utilizan, combinar funciones de captura de paquetes con las capacidades de generación de informes gráficos de SteelCentral Packet Analyzer, recibiendo además triggers y diagnosticando problemas en el tráfico de manera más rápida que usando solo Wireshark.

b) Acunetix

Puede decirse que Acunetix es uno de los escaneadores web de vulnerabilidades de lo más completo del mercado.

Cuenta con una gran batería de pruebas actualizables o diseñadas por el propio usuario, así como con gran cantidad de herramientas de análisis.

ARP SPOOFING

Arp es el protocolo de resolución de direcciones (Address Resolution Protocol) mediante el cual en una red interna se obtiene una dirección física a partir de una dirección IP.

La mayoría de redes domésticas y corporativas utilizan asignación dinámica de Ip's, por lo cual esta puede variar, provocando que la dirección IP no sirva como identificador inequívoco del equipo.
Sin embargo, la dirección de la tarjeta de red sí que es única. Y es lo que se utiliza para mandar un paquete a un equipo.
Arp trabaja de la siguiente forma.

1) Un equipo de la red pregunta, mediante Broadcast: ¿Qué dirección física tiene la dirección IP 192.168.1.BB?

2) El equipo con dicha IP responde:
A la dirección IP 192.168.1.BB le corresponde la dirección MAC BB:BB:BB:BB:BB:BB

3) El equipo inicial, registra en su tabla ARP que la dirección IP 192.168.1.BB le corresponde la

dirección MAC BB:BB:BB:BB:BB:BB

Si se realiza la técnica de ARP Spoofing, de manera bidireccional, conseguiremos hacer el ataque Man In The Middle (MITM), de tal manera que, los paquetes que procedan del equipo inicial o lo tengan como destinatario, pasarán por el equipo que hace de Hacker (el que ha ejecutado el arp spoofing).

Un procedimiento simple para realizar este Arp Spoofing, podría ser el siguiente:

- Preparar el equipo MITM para la redirección de tráfico:

  $ cat /proc/sys/net/ipv4/ip_forward

Este fichero contendrá el valor 0, por defecto. Tendremos que habilitarlo, sustituyendo dicho valor por el 1.

  $ echo '1' > /proc/sys/net/ipv4/ip_forward

O bien, usando un editor de textos para escribir en el fichero.

A partir de aquí estamos en predisposición de comenzar el ataque.

Abrimos 2 terminales para realizar el arp spoofing en las 2 direcciones(desde la víctima hacia la puerta_enlace y desde puerta_enlace hacia víctima), usaremos los siguientes comandos (cada uno en una terminal diferente):

  $ arpspoof -i interfaz -t 192.168.1.1 192.168.1.14

  $ arpspoof -i interfaz -t 192.168.1.14 192.168.1.1

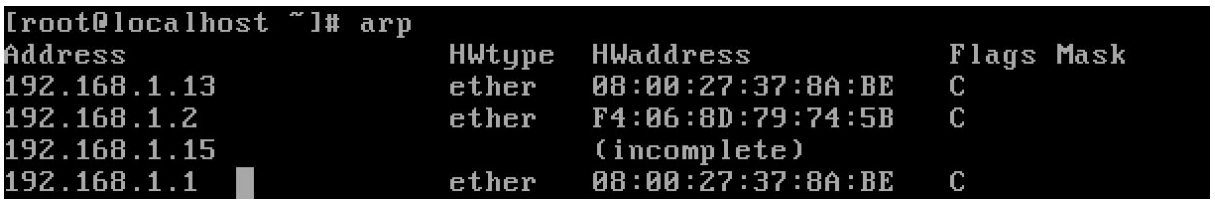
Suponiendo que:
192.168.1.1 es la puerta de enlace
192.168.1.14 es la víctima

En algunos casos, puede ser necesario hacer algún tipo de redirección en el cortafuegos del atacante (aunque, por ahora, en un primer momento, no realizamos este paso):

- Redirigir el tráfico http al puerto 8080 para poder realizar hacer el sniffing:

  $ iptables -t nat -A PREROUTING -p tcp –destination-port 80 -j REDIRECT--to-port 8080

Si observamos las tablas Arp de la víctima (mediante comando arp), vemos que, para dos ip's diferentes (la del atacante: 192.168.1.13 y la de la puerta de enlace: 192.168.1.1) hay una misma dirección MAC, que corresponde, precisamente, al del atacante, que está haciendo la suplantación.



Si detenemos el comando (CTRL + C), la respuesta del comando arp, en la víctima, muestra ahora la MAC auténtica de la puerta de enlace:

A partir de ahora, nuestro equipo está haciendo de MITM entre la víctima y el router o puerta de enlace, y el tráfico intercambiado, pasa por nosotros.

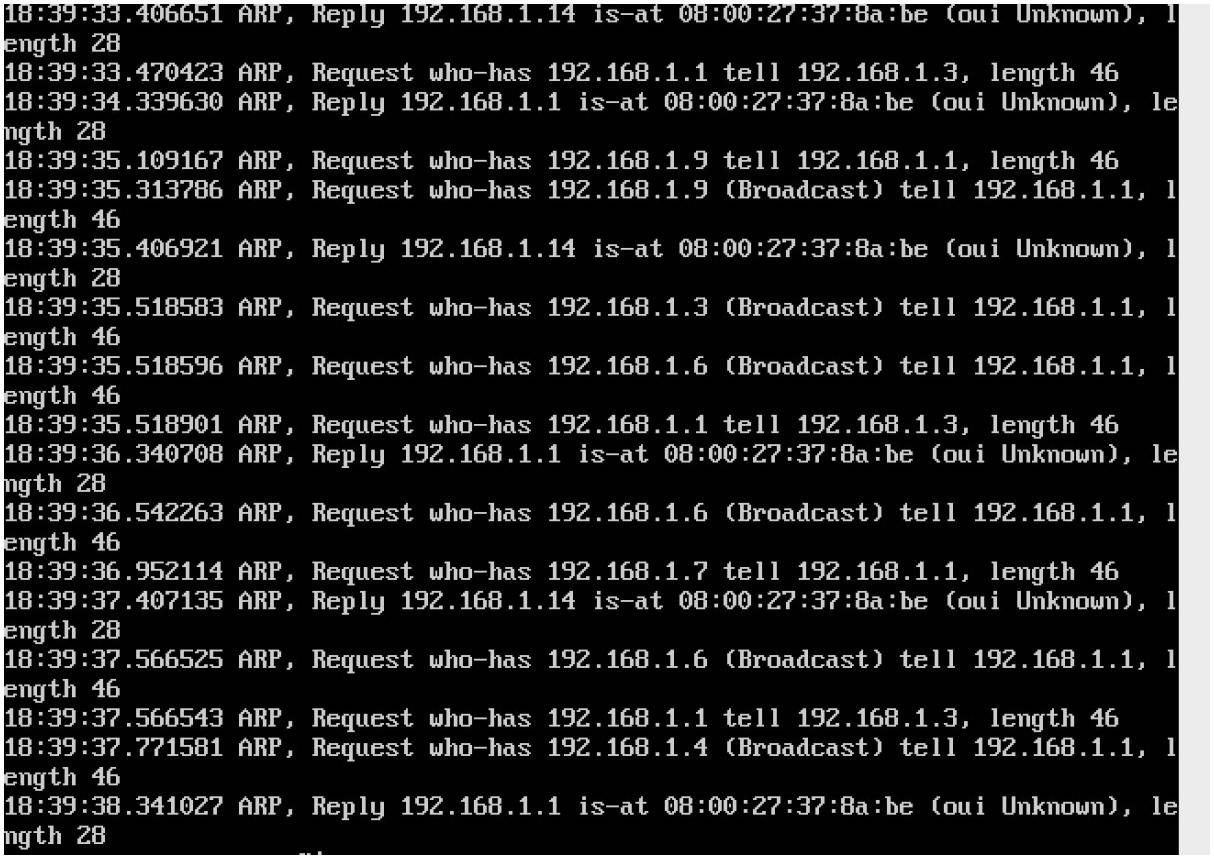
Si lanzamos un tcpdumpo o mediante Wireshark, podemos ver esta suplantación.

Lanzamos captura de tráfico desde el equipo que está haciendo la suplantación de IP (en este ejemplo se ha realizado desde la 192.168.1.13). Se indica que capture el tráfico donde aparezca la ip del Pc suplantado (192.168.1.14) y lo escriba en un fichero.

$ tcpdump –n -i eth0 -host 192.168.1.14 –s 0 –w fichero_recogida

Posteriormente, observamos el contenido del fichero donde se ha recogido el tráfico.

Un ejemplo de lo que nos podemos encontrar sería algo así:

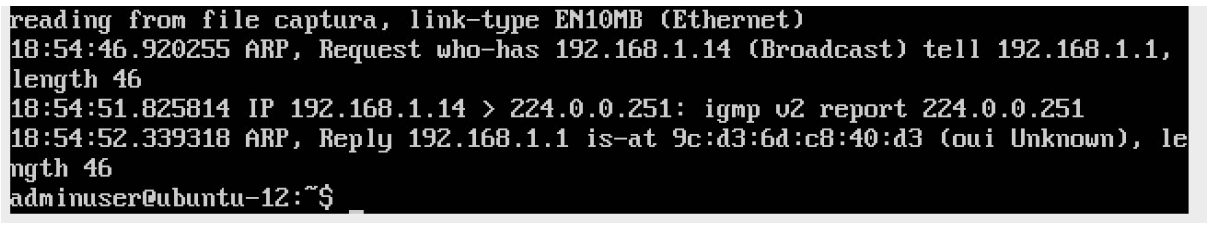


Observa las peticiones Arp y las respuestas a dichas peticiones.

Ejercicio 2.

a) ¿Cómo diferencias lo que es una petición ARP de su respuesta?
b) ¿Cuál es la MAC de la puerta de enlace?
c) ¿Quién la está suministrando?
d) ¿Qué herramienta o método emplearías para obtener la IP de alguna víctima?
e) ¿Qué comando usarías para ver el contenido del fichero_recogida? ¿De qué otra forma podrías ver ese mismo contenido.

Una vez detenida la suplantación, si hacemos otra captura, podemos ver lo siguiente:



La MAC de la puerta de enlace (192.168.1.1) vuelve a ser la real.

Otras observaciones:

Existen otras herramientas de interés usadas para capturar tráfico, como pueden ser: sslstrip, urlsnarf y driftnet. Puede ser interesante echar un vistazo para ver sus utilidades.

Mitigación de riesgos de un Arp Spoofing

Algunas soluciones pasan por el uso de tablas arp estáticas, de tal manera que, sin caché arp, no hay nada que envenenar. El inconveniente viene dado por el coste de mantenimiento que tiene el mantenimiento de dichas tablas.

Otra solución es usar la herramienta Arpwatch, para linux (en Windows es WinARP Watch), avisando a través de correo, de los cambios en las tablas arp que se van produciendo.

También se puede utilizar RARP, en lugar de ARP, esto es Reverse Address Resolution Protocol, en el que a partir de una dirección MAC nos devuelve su correspondiente dirección IP. En caso de devolver más de una, implicaría que esa IP ha sido clonada.

Una recomendación, adicional es que, a la hora de navegar, se use tráfico https en vez de http, ya que el tráfico va cifrado.

Se puede observar un ataque completo y guiado de MITM en el siguente interesante post : https://thacid.wordpress.com/2012/03/02/man-in-the-middle-con-arpspoof-sslstrip-wireshark-y-mucho-tiempo-libre