SERVIDORES WEB DE

ALTAS PRESTACIONES

ANÁLISIS DEL TRÁFICO CON WIRESHARK

ugr | Universidad de Granada

José David Torres de las Morenas

Contenido

1. Wireshark 3
2. Conociendo las distintas partes de Wireshark 3
3. Formas de obtener los datos según los escenarios (modos de captura) 4
3.1. Utilizando un Hub 4
3.2. Modo Bridge 5
3.3. Port Mirroring o VACL 6
3.4. ARP Spoofing 6
3.5. Remote Packet Capture 7
4. Empezando a utilizar Wireshark 7
5. Ataques en redes de área local 10
5.1. ARP Spoofing 10
5.2. Port Flooding 10
5.3. DDOS Attacks 11
5.4. DHCP Spoofing 11
5.5. VLAN Hopping 12
5.5.1. Suplantación del switch 12
5.5.2. Ataque de etiquetado doble 12
6. Bibliografía 12

1. Wireshark

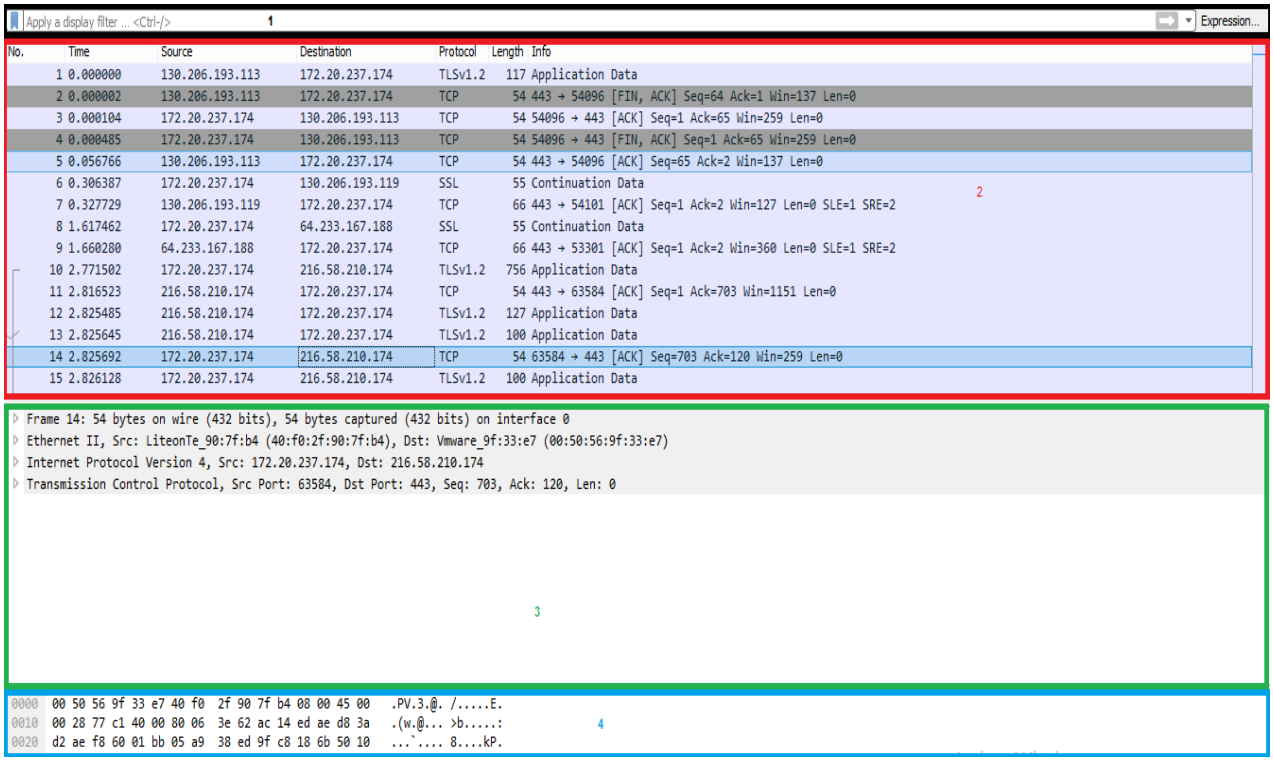
Wireshark es un analizador de protocolos open-source que se utiliza para realizar análisis y estudiar redes de comunicaciones y resolución de problemas de red, aunque se utiliza principalmente para el análisis de tráfico. Wireshark está disponible para su uso tanto en plataformas Windows como Unix.

Wireshark implementa filtros que facilitan la definición de criterios de búsqueda a través de una sencilla interfaz donde podemos ver los datos de cada una de las capas de los paquetes capturados, esto nos proporciona muchas posibilidades a la hora de analizar el tráfico.

Wireshark dispone de una versión en línea de comandos denominada Tshark, aunque nos centraremos en su versión gráfica. Existen situaciones en las que Wireshark no es capaz de estudiar ciertos protocolos por la documentación de los mismos, donde el mejor método para esta situación es utilizar la ingeniería inversa.

2. Conociendo las distintas partes de Wireshark

Vamos a presentar la interfaz de Wireshark para poder interpretar mejor cada una de las zonas de datos que se nos muestra en el programa:



- Zona 1 (negro): podemos definir los distintos filtros de búsqueda para realizar una búsqueda con unos parámetros concretos, para realizar una vista de los paquetes más concreta.

- Zona 2 (rojo): vista general de los paquetes que están en transición en tiempo real. Por cada línea como ya veremos se representa el número, tiempo, origen, destino, protocolo utilizado e información adicional. A partir de estas líneas de transiciones es donde vamos a realizar gran parte del estudio de los problemas
  .
- Zona 3 (verde): Por cada línea de transición de la zona que hemos visto anteriormente, podremos desglosar en capas las cabeceras de cada uno de los paquetes y podemos ver en esta zona los distintos campos del paquete en cuestión

- Zona 4 (azul): representa el paquete en formato hexadecimal, que es tal y como s envían los datos por la red, y tal y como los captura nuestra tarjeta de red.

3. Formas de obtener los datos según los escenarios (modos de captura)

Imaginemos por ejemplo un escenario en un entorno conmutado formado por varios switches, varios equipos y un servidor de ficheros. Supongamos que el rendimiento de la red ha disminuido en los últimos días y que carecemos de un Sistema de Intrusos (IDS) que pueda avisarnos sobre algún ataque en la red. Conocemos que el servidor de ficheros es suficiente en cuanto a abastecimiento de los equipos de nuestra área local. Dado que no disponemos de nada en los equipos para poder analizar el tráfico de la red, debemos utilizar Wireshark, pero ¿dónde debemos instalarlo para poder registrar la transferencia de paquetes que deseamos?

Nos encontramos situaciones en las cuales no podemos instalar Wireshark en el servidor de ficheros ya que no tenemos por ejemplos acceso físico al servidor o simplemente por motivos de seguridad. En estas situaciones tenemos alternativas que permiten capturar el tráfico de la red sin necesidad de portar Wireshark al servidor.

3.1. Utilizando un Hub

Si conectamos un equipo con Wireshark instalado a uno de los puertos del switch, solo veíamos los paquetes transferidos entre el switch y nuestra máquina, debido a que el switch divide en segmentos la red, de forma que crea dominios de colisión separados eliminando así el deber de que cada estación compita por el medio. Solamente se hacen envío de paquetes a todos los puertos cuando se trata de difusiones broadcast.
Para conseguir nuestro objetivo podemos utilizar un hub conectándolo en el mismo segmento de red donde se encuentra el servidor. Lo que se ha hecho así es convertirlo en un medio compartido y de esta forma el tráfico entre el servidor y el switch podrá analizarse en nuestro equipo.

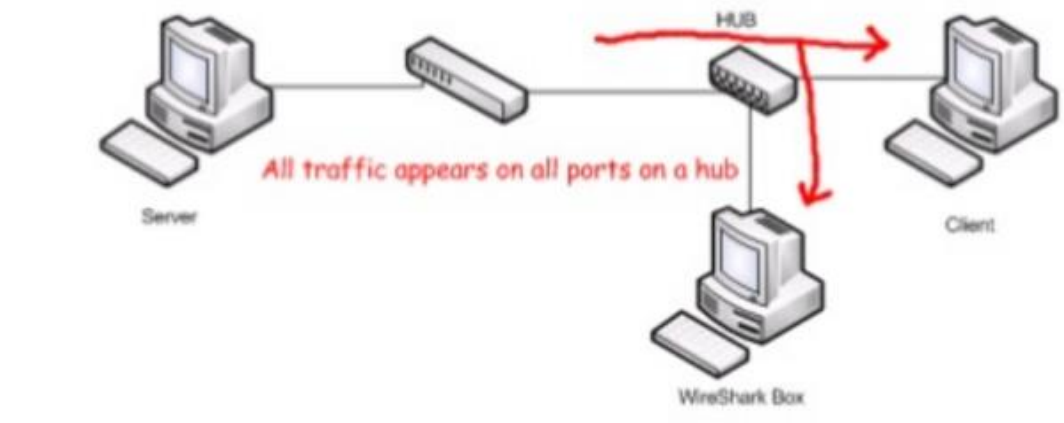


Ilustración 1: Modos de captura. Hub

3.2. Modo Bridge

Puede que no tengamos acceso al switch, por lo que podríamos también utilizar un equipo con dos tarjetas de red de modo que nos situaos entre el switch y el servidor. Esto consiste en un MitM (Man in the Middle) a nivel físico donde podremos analizar el tráfico de paquetes. Para obtener este modo de funcionamiento en Linux es bastante fácil ya que tras su instalación, debemos crear una interfaz de tipo bridge y tras esto las interfaces que forman parte del puente. A continuación levantamos la interfaz y ejecutamos nuestro programa Wireshark. Todo esto lo hacemos con las siguientes instrucciones:

```
brctl addbr mybridge
brctl addif mybridge eth0
brctl addif mybridge eth1
ifconfig mybridge up
```

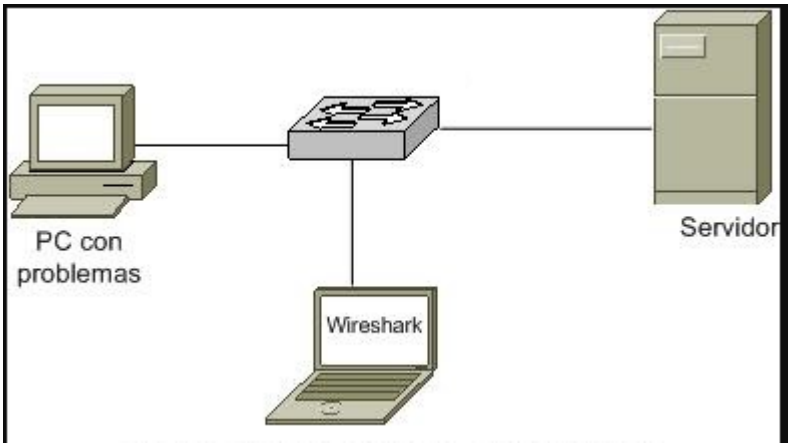


Ilustración 2: Modos de captura. Modo Bridge

3.3. Port Mirroring o VACL

Es la mejor manera que podemos estudiar el tráfico de la red siempre que tengamos acceso a switch. Consiste en duplicar el tráfico de varios puertos del switch a un puerto que queramos.

Una ventaja que presenta VACL (VLAN-BASED ACLS) es que nos permite mayor granularidad a la hora de analizar el tráfico por unos parámetros concretos, es decir, podemos elegir el tipo de tráfico que queremos analizar.

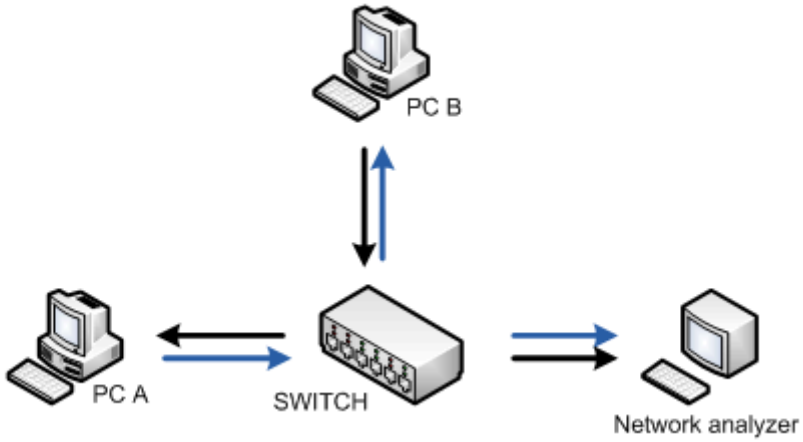


Ilustración 3: Modos de captura. Port Mirroring

3.4. ARP Spoofing

Es un método que se debería de llevar a cabo solo si los anteriores no se pueden llevar a cabo, debido a que es bastante ofensivo ya que se trata de un MitM (Man in the middle). Solo será útil en entornos no críticos. De esta forma, conseguiremos con Wireshark monitorizar todos los paquetes transferidos en la red. Este proceso es llevado a cabo de forma que se vulnera la cache de los equipos con una asociación IP/MAC falsa.

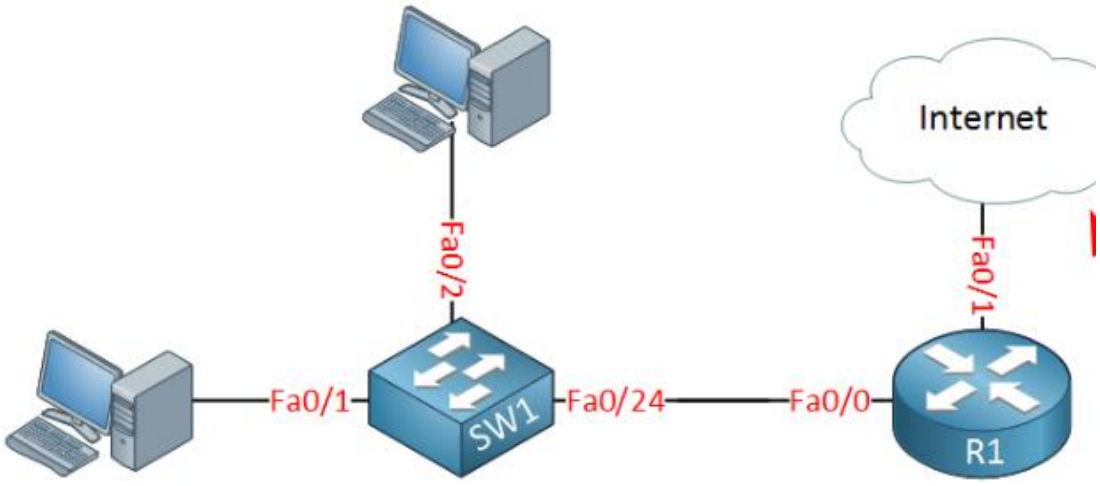


Ilustración 4. Modos de captura. ARP Spoofing

3.5. Remote Packet Capture

Este método solo se puede utilizar en caso de que podamos tener acceso remoto al servidor ya que debemos ejecutar un programa servidor junto con las librerías y un programa cliente desde donde podremos visualizar los datos.
Para poner en marcha este método debemos de configurar el servidor, ejecutando el archivo rpcapd.exe que se incluye en la instalación de WinPcap. Podemos tener dos métodos de escucha, activo y pasivo. En el activo el demonio establecerá conexión con el cliente para que envíe los comandos adecuados al servidor. En el pasivo el cliente inicia la conexión con el servidor para comenzar a monitorizar los datos:

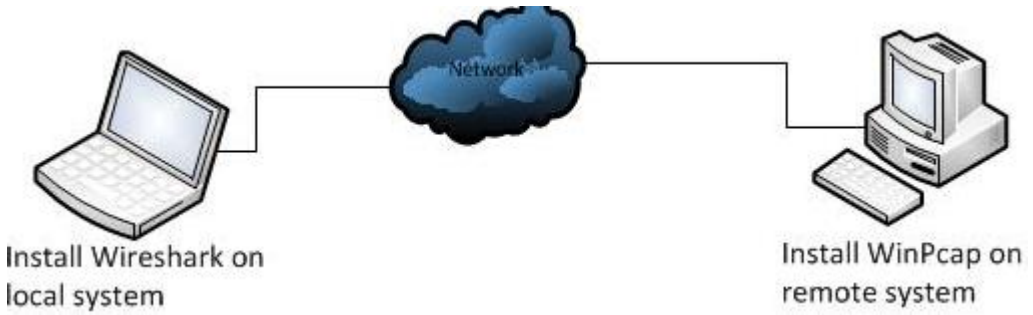


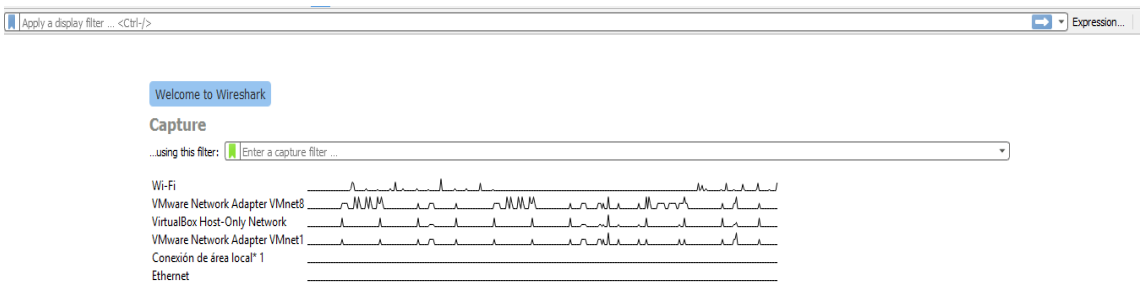
Ilustración 5. Modos de captura. Remote Packet Capture

4. Empezando a utilizar Wireshark

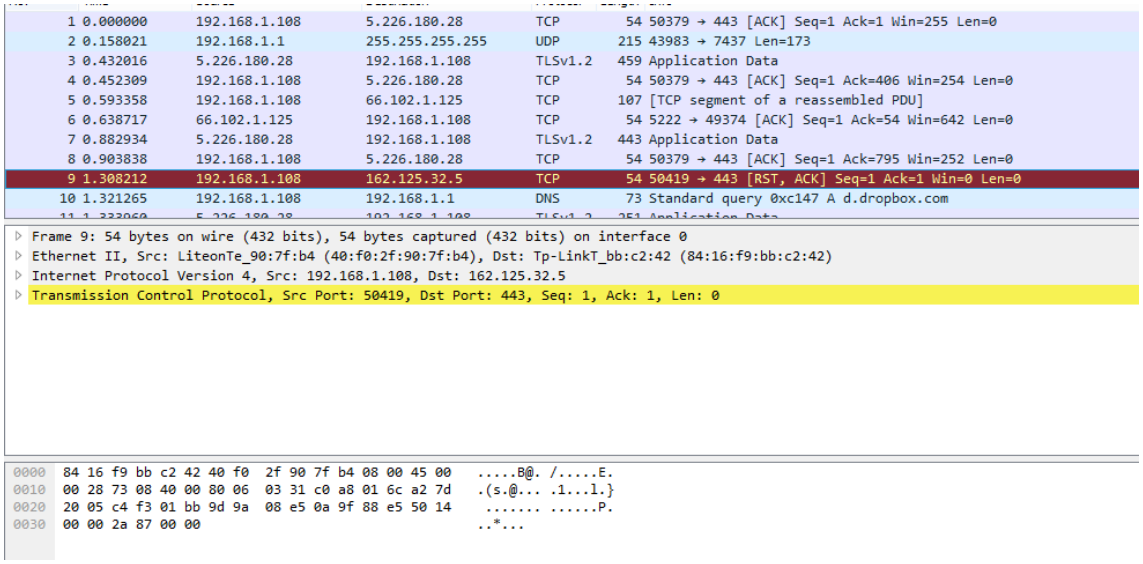
Lo primero que haremos será instalar Wireshark, lo podemos descargar de la página oficial, a través del siguiente enlace: https://www.wireshark.org/#download e instalarlo.

Vamos a ver el tráfico que se produce en nuestra web, para empezar a familiarizarnos con Wireshark:

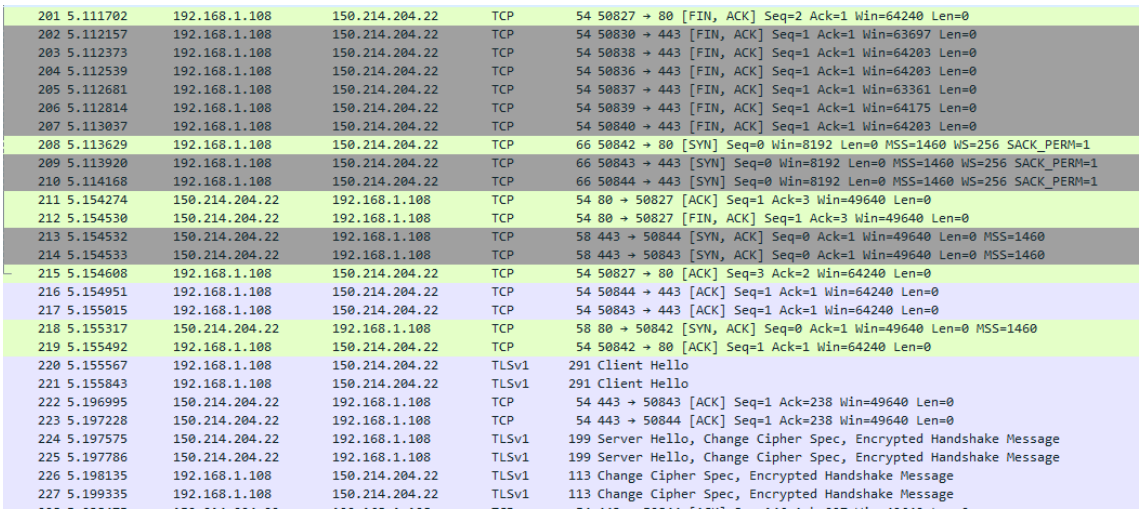
- Lo primero que vemos al abrir Wireshark es la lista de interfaces, donde podemos ver el tráfico de paquetes que se transfieren por nuestra red.



Podemos ver en más detalle los paquetes de cualquiera de estas interfaces, simplemente haciendo doble click sobre él, y nos aparecerá la información de paquetes sobre la interfaz elegida. En nuestro caso, elegimos Wifi para ver la transmisión de paquetes:



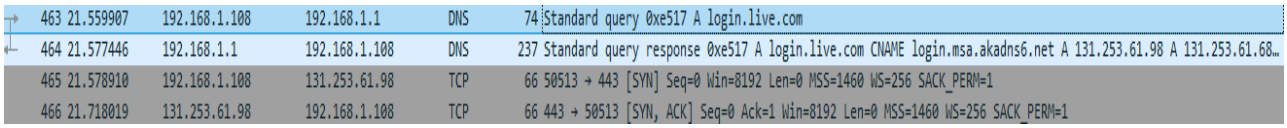
Vamos a realizar un ejemplo de prueba para demostrar la captura de paquetes de Wireshark. Lo primero que haremos será pausar la captura de paquetes de Wireshark y volver a ejecutarlo, tras esto, entraremos en una web. Simplemente vamos a ver los paquetes que se envían y reciben solo con entrar a la web, en este caso hemos entrado al correo institucional de la Universidad de Granada (150.214.204.22):



Lo que podemos ver en la imagen es en orden de izquierda a derecha:

- Número, tiempo, ip fuente, ip destino, protocolo, medida e información.

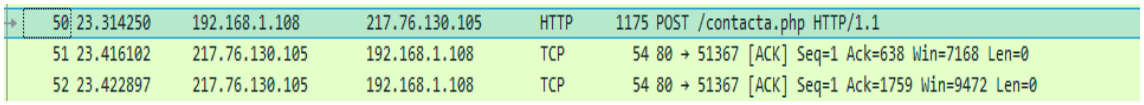
Estos son solamente algunos de los paquetes que se han transferido al entrar al correo de la ugr. Vamos a realizar una segunda, prueba, vamos a entrar a una página, y vamos a hacer un loggin.



Como podemos ver Wireshark ha registrado la transmisión de paquetes de nuestro loggin, y si no tuviese un protocolo de transmisión de datos seguro, podríamos ver los datos que hemos introducido al hacer el log tal como lo hemos introducido, pero tal como se nos muestra está encriptado.

Otra prueba que podemos hacer es la siguiente, vamos a entrar en una página con protocolo HTTP donde vamos a rellenar un formulario de contacto, y vamos a obtener mediante Wireshark:

1. Una vez enviado el formulario de contacto en la página con Wireshark iniciado para Wifi, buscamos si queremos simplificar con búsqueda al protocolo http, una línea de transmisión de paquetes donde se hace POST en este caso de contacta.php:



2. A continuación, vamos a ver con más detalle los datos del paquete transmitido. Pulsamos botón derecho sobre la línea y clickamos en Follow -> TCP Stream. Y aquí podemos observar todos el contenido enviado en el formulario:

```
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="nombre"

ssdjfjodsj
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="direccion"

dsjfdsoj
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="mail"

sdkfsdk@hotmail.es
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="telefono"

666666666
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="destinatario"

Dept. Comercial
------WebKitFormBoundaryESlAoZ32CpqbmQHl
Content-Disposition: form-data; name="asunto"
```

5. Ataques en redes de área local

5.1. ARP Spoofing

ARP Spoofing además de servirnos como método para poder capturar paquetes que se transfieren en la red, también se utiliza por atacantes para interceptar, modificar o capturar paquetes. En la siguiente imagen podemos analizar un ataque:

4	9.028195	10.0.0.109	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
5	9.678865	IntelCor_6e:a2:69	Broadcast	ARP	Who has 10.0.0.1? Tell 10.0.0.101
6	9.681088	Cisco-Li_2b:72:04	IntelCor_6e:a2:69	ARP	10.0.0.1 is at 00:18:39:2b:72:04
7	9.692034	IntelCor_6e:a2:69	Broadcast	ARP	Who has 10.0.0.100? Tell 10.0.0.101
8	9.696736	IntelCor_49:bd:93	IntelCor_6e:a2:69	ARP	10.0.0.100 is at 00:12:f0:49:bd:93
9	10.768172	10.0.0.100	10.0.0.1	ICMP	Echo (ping) request
10	10.800072	10.0.0.1	10.0.0.100	ICMP	Echo (ping) request
11	10.800176	IntelCor_6e:a2:69	Cisco-Li_2b:72:04	ARP	10.0.0.100 is at 00:13:ce:6e:a2:69
12	10.800245	IntelCor_6e:a2:69	IntelCor_49:bd:93	ARP	10.0.0.1 is at 00:13:ce:6e:a2:69
13	11.810451	IntelCor_6e:a2:69	Cisco-Li_2b:72:04	ARP	10.0.0.100 is at 00:13:ce:6e:a2:69
14	11.833724	10.0.0.100		TCP	1390 > www [SYN] Seq=0 Len=0 MSS=1460
15	11.857257	IntelCor_6e:a2:69	IntelCor_49:bd:93	ARP	10.0.0.1 is at 00:13:ce:6e:a2:69
16	11.859246	IntelCor_6e:a2:69	Broadcast	ARP	Who has 10.0.0.1? Tell 10.0.0.101

En la línea 5 vemos como la maquina con IP 10.0.0.101 ha lanzado un ARP request a dirección broadcast preguntando por la MAC de la IP 10.0.0.1 (Gateway de la red). Se contesta con un ARP reply indicando la dirección MAC. Tras esto se repite el paso de la línea 5 enviando otra difusión broadcast y se vuelve a contestar. A partir de aquí tenemos una máquina de nuestra LAN que dispone de la MAC del servidor y del router, por tanto puede compartir tráfico.
En la línea número 11, la máquina envía varias veces ARP reply falsos hacía el servidor y el router, de forma que se consigue asociar la IP de ambos con su propia MAC de forma que el tráfico que transita pasará por la máquina atacante.
Nada nos impide poder modificar un mensaje ARP reply, y modificarlo de tal forma que donde se recibe este ARP reply, tenga una información no verídica en esta trama. Cuando tenemos la trama modificada se puede enviar directamente por la interfaz conectada en nuestra LAN, mediante la siguiente instrucción:

```
file2cable –ii eth0 –f arpreply
```

Podemos mantener el ataque mediante un script que lo ejecute de forma constante en la caché, de esta manera conseguimos tener la caché contaminada en todo momento y conseguir así que todos los paquetes dirigidos fuera de la LAN pasen por nuestro equipo atacante.

5.2. Port Flooding

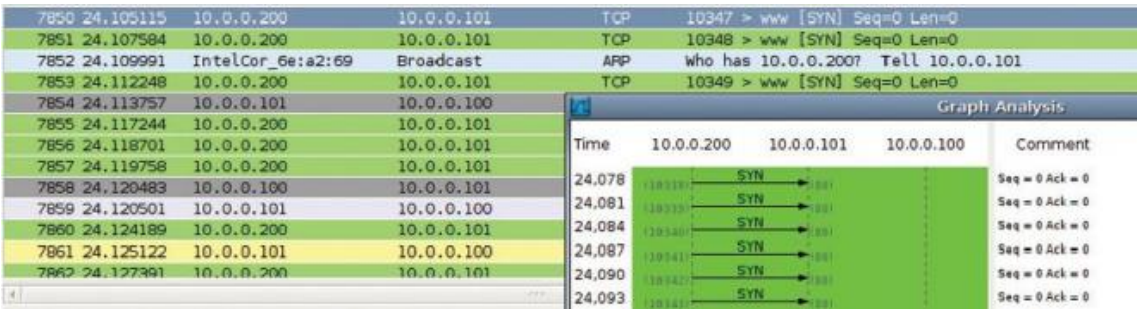
Este ataque se realiza mediante múltiples envíos de tramas falsificadas a través de un puerto con el fin de llenar la tabla de asignaciones del switch. Esta tabla de asignaciones funciona de forma que cuando se recibe un paquee en un determinado puerto se añade en la CAM (Content-Addressable Memory, memoria interna del switch) una entrada especificando la MAC del equipo que envió la trama, de forma que cuando se recibe en el switch una trama de este equipo, sabrá por cual puerto debe enviarla.
Si se desconoce el destino, enviará una copia de la trama y la envía a todos los puertos de la misma VLAN, de forma que todos los equipos conectados reciben la trama y en el cual coincida la MAC será el que contestará.
En caso de que se envíen múltiples tramas falsificando MAC, y alterando la tabla MAC de una forma falsificada, en los switches de gama baja que no disponen de tablas CAM virtualizadas se llenará la tabla y todas las VLAN se verán afectadas, sin embargo en los switches de gama alta, que disponen de un espacio de direcciones independiente para cada VLAN, solo se verían

afectados los equipos de la propia VLAN.
Este tipo de ataques se pueden detectar mirando el tráfico que se genera en un tramo de la red, donde veríamos gran cantidad de tramas con valores aleatorios, y en estas tramas en la descripción aparecería [Malformed Packet], por la forma en la que Macof construye paquetes TCP sin tener en cuenta especificaciones del protocolo. Podemos ver un ejemplo en la siguiente imagen:

348	13.302264	82.8.242.103	225.173.109.6	TCP	[Malformed Packet]
349	13.303184	88.125.244.10	81.219.96.39	TCP	[Malformed Packet]
350	13.305176	92.236.234.36	103.223.24.56	TCP	[Malformed Packet]

5.3. DDOS Attacks

Este tipo de ataque consiste en intentar establecer conexión por parte de un equipo a un mismo puerto, el servidor trata de resolver la MAC de la máquina haciendo broadcast, pero al no recibir respuesta y por tanto no disponer de la dirección del host no puede enviar un ACK-SYN para establecer la conexión. Esto supone que por cada intento de establecer conexión se tiene que esperar un tiempo determinado, en el cual siguen llegando paquetes intentado establecer otras conexiones. Cada intento genera en memoria una estructura TCB (Transmission Control Block) que se usa para identificar las conexiones, y que al generar un número muy elevado puede terminar cayendo el sistema, y por tanto dejar de contestar a las solicitudes de conexión.
Vemos como se manifiesta de forma gráfica este tipo de ataque. La secuencia de paquetes la podemos ver haciendo click en Statistics >> Flow Graph:



Este tipo de ataques se han registrado en empresas como Amazon y Paypal.

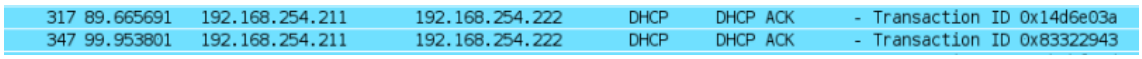
5.4. DHCP Spoofing

Este tipo de ataque consiste en falsificar paquetes DHCP, mediante un software que emule las funciones del mismo respondiendo las peticiones DHCPDISCOVER que envían los clientes. Cuando un cliente se conecta a la red, solicita una dirección IP enviando un DHCPDISCOVER a la dirección broadcast esperando respuesta de algún servidor que contestará a la petición enviando un paquete unicast (DHCPOFFER) y que contiene parámetros de configuración.
DHCP no tiene mecanismos de autenticación que verifiquen origen de las tramas de forma que nada impide la falsificación de paquetes DHCPOFFER con información falsa. De esta forma es posible realizar un ataque proporcionando como puerta de enlace al cliente la IP del atacante y recibiendo tramas fuera de la LAN, de forma que esto es transparente al usuario.

Este tipo de ataques en Wireshark se muestran con un uso anormal del protocolo DHCP, y con posibles errores en las máquinas debido a IPs duplicadas. Además podemos realizar una búsqueda rápida de respuestas CK con un DNS o Gateway diferentes al configurado en el servidor con la siguiente línea en la barra de búsqueda:

> bootp.option.value == 05 && (frame[309:6] ¡= 03:04:c0:a8:fe:fe || frame[315:6] == 06:04:c0:a8:fe:d3)

Obtendríamos una salida como la siguiente en caso de que haya tramas con estos requisitos:



5.5. VLAN Hopping

Este ataque consiste en atacar los recursos de la red que soportan una VLAN, de forma que logre el acceso al tráfico de otras VLAN diferentes a donde se encuentra el dispositivo atacante. Este tipo de ataque se puede hacer mediante suplantación del switch y por doble etiquetado de los paquetes.

5.5.1. Suplantación del switch

Consiste en configurar el equipo de forma que maneje los protocolos de etiquetado y concentración de enlaces utilizados entre switches de la red. En este momento en el que consigue esto, lograría el acceso al tráfico de las tramas del resto de las VLAN ya que se volvería miembro de todas.

5.5.2. Ataque de etiquetado doble

Los paquetes que se transmiten son antepuestos a dos etiquetas VLAN, de forma que al realizar los switch el desencapsulado, ya que solo hacen un nivel, que es el encabezado correspondiente a la VLAN del que el atacante es miembro, y queda la segunda etiqueta con VLAN falso que se destina a la máquina del cliente víctima.

6. Bibliografía

Wireshark, https://www.wireshark.org/

Oficina de Seguridad del Internauta, https://www.osi.es/es/herramientas-gratuitas/wireshark

Wikipedia, https://es.wikipedia.org/wiki/Wireshark

ITSM, http://cs.mty.itesm.mx/lab/redes1/Practicas/Redes1_Pr5_OSI_App.pdf

Instituto Nacional de CiberSeguridad de España, https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf

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

Youtube, https://www.youtube.com/watch?v=Y5rZlmmqVQk