Phishing: Fraude informático y como evitarlo

Mamani Castillo Carla María
Universidad Mayor De San Andrés
Carrera De Informática
Simulación de Sistemas
krlis_t@hotmail.com

RESUMEN

En este artículo nos vamos a enfocar en el análisis del tráfico de red, junto con su contenido, que resulta esencial para conocer qué uso se le ha dado a la red en un momento determinado.

Nos centraremos en la reconstrucción de ciertos tipos de datos capturados por la red y en última instancia se darán algunas recomendaciones para la automatización de este tipo de tareas.

Palabras clave

Análisis de red, Wireshark

1. INTRODUCCIÓN

El análisis de reconstrucción del tráfico de red nos ayuda por ejemplo en un servidor infectado mediante un exploit lanzado de forma remota o el conocer a ciencia cierta que están robando información confidencial de una empresa. Gracias a la reconstrucción del tráfico de red, es posible que se puedan dar respuestas a un buen número de cuestiones.

Para este análisis se procedió a realizar pruebas en el laboratorio para capturar tráfico de dos puntos en una LAN y guardarlos en un archivo PCAP. Las siglas PCAP (Packet Capture) corresponden a una API desarrollada para capturar tráfico de una red. En el laboratorio de pruebas, se utilizo el port para Windows WinPcaP y Posteriormente la herramienta Wireshark para la extracción de posibles datos.

2. ANÁLISIS TRÁFICO DE RED

Algo que puede que nos interese, en lo referente al análisis de la red, es la reconstrucción de la navegación web, guardado en un determinado fichero de captura PCAP.

Para intentar reconstruir ese tráfico, Wireshark implementa varios filtros que podemos utilizar. En un principio puede que nos interese todo el tráfico, pero gracias a los filtros, podremos centrarnos en ciertas partes del tráfico que más nos interese.

Para captura web, Wireshark implementa el . Gracias a este filtro, podremos extraer prácticamente todos los objetos enviados y recibidos mediante este protocolo.

¿Y cómo podemos extraer datos una vez aplicado este filtro? Wireshark implementa varias formas de realizarlo con el protocolo HTTP.

Una de ellas es la capacidad de poder exportar un conjunto de bytes, utilizando únicamente para ello el botón derecho del ratón.

Otra forma de realizar esta exportación de datos, es utilizando el filtro http.content_type, el cual nos mostrará sólo los paquetes que contengan algún tipo de dato y hayan viajado a través de ese protocolo.

Como se puede observar en la imagen, el tipo de dato o recurso que viaja por HTTP no siempre es texto. De ahí que se deba analizar con cuidado, por si nos encontramos con el lanzamiento de un exploit, un ataque de denegación de servicio, aplicaciones, objetos comprimidos, etc…

Desde hace algún tiempo, Wireshark implementa una forma rápida de agrupar todos los objetos HTTP en una sola ventana, para que así nos resulte más fácil “ver” qué viaja por este protocolo. Para disfrutar de esta característica, debemos ir a File –> Export –> Objects –> HTTP

Otro filtro importante que tenemos a nivel de HTTP, es el filtro HTTP.date, este filtro, nos situará exactamente en la cabecera de HTTP que contenga el campo fecha. Esta acción nos puede ser muy útil para realizar un timeline de peticiones y poder reconstruir una sesión de navegación, por ejemplo.

Visto lo anterior, Wireshark puede reconstruir todo el tráfico de cualquier protocolo? La respuesta es que NO. Wireshark permite capturar todo tipo de paquetes pero eso no quiere decir que pueda diseccionarlos todos. Existen protocolos que están documentados o se basan en una RFC, y protocolos para los que no hay apenas documentación pública. Para este tipo de protocolos es en donde entra en juego la ingeniería inversa.

3. NO MUY DOCUMENTADOS

La gran mayoría de analizadores de red ofrecen soporte a protocolos que están bien documentados y dan parte de soporte o ninguno a protocolos que no están documentados. Este tipo de problemas se pueden dar en algún tipo de investigación forense.

En nuestro caso, vamos a centrarnos en un protocolo que no está muy documentado, pero que es ampliamente utilizado. Es el caso de la mensajería instantánea con Windows Live Messenger y derivados.

Los primeros atisbos de información que nos encontramos, los tenemos en la Web del Internet Engineering Task Force.

En dicha Web, mantienen un borrador de un protocolo llamado . Leyendo este documento, tendremos una idea básica de cómo el protocolo de Windows Live Messenger trabaja a la hora de enviar y recibir información

Wireshark implementa un único filtro para filtrar la comunicación a través de Messenger . Gracias a este filtro, podremos trabajar sólo con este protocolo.

Al ser un protocolo que envía los mensajes en texto plano, es fácil extraer una conversación de Messenger aplicando como búsqueda la cadena "Text/Plain" en el filtro de búsqueda por paquetes que dispone Wireshark.

Herramientas comerciales como , pueden capturar este tráfico y parsearlo directamente a una salida más "humana".

Lo malo de todo esto, es que no toda la información enviada a través de este cliente de mensajería se envía a través del protocolo MSNMS. El envío de mensajes de Voz, ficheros o vídeo se realiza a través de protocolos comunes como TCP o UDP. Y, en este caso, ni MSN-Sniffer ni ninguna otra herramienta es capaz de detectarlo y extraerlo.

En el caso del envío de ficheros a través de Messenger, y leyendo la valiosa información de la Web , nos encontramos con que primeramente se envía un número de identificador único (GUID), el cual se especifica para determinar que lo que se va a transmitir es un fichero. En el mismo paquete, se añade un identificador llamado APPID.

Si se observa con atención este paquete, en el contexto de este campo (AppID), aparecen una serie de caracteres codificados en Base64. Decodificando este texto nos daría el nombre del fichero que se está enviando.

Una vez recibida esta información se empieza con el envío de datos. La transmisión de ficheros a través de Messenger es algo complicada, ya que se pueden enviar a través de TCP o UDP, y en este caso, no se conoce ninguna herramienta que, a través de un archivo PCAP y en modo Offline, pueda reconstruir los ficheros enviados a través de este cliente de mensajería instantánea, como extraer ficheros enviados a través de Messenger. La idea básica es ir recopilando cada paquete (ya sea TCP o UDP), para después unir todas las tramas y formar el fichero extraído.

Como esto se complica cuando se transmiten varios ficheros simultáneamente, se creó una herramienta muy interesante hace tiempo que realizaba esto mismo. Capturar todos los paquetes transmitidos a través de messenger, para después reconstruir los ficheros realizando un fingerprinting de los mismos.

4. FORENSE DE TRAMAS DE RED

Para el análisis forense de tramas de Red, existen herramientas que son capaces de automatizar el proceso de extracción, siendo válidas para extraer "de una tirada" conjuntos de datos que viajan a través de un protocolo.

Una herramienta bastante potente para realizar este tipo de extracciones es . Esta herramienta, de momento sólo existe para sistemas operativos Linux. Hasta la fecha es la herramienta más interesante, ya que de momento, está realizando una muy buena labor de soporte para .

Para Windows, la única herramienta que vale la pena, se llama . Es una herramienta bastante interesante, ya que, además de la extracción de datos, es capaz de realizar tareas de fingerprinting sobre los paquetes de datos para averiguar, por ejemplo, el sistema operativo que envía o recibe paquetes. Su funcionamiento se basa en las bases de datos que utilizan diversas herramientas de auditoría, como por ejemplo, NMap.

Lamentablemente, y utilizando esta herramienta con un fichero PCAP, en cuyo interior se encontraban datos relativos a conversaciones de vídeo, audio y datos, transmitidos todos ellos a través de mensajería instantánea con Windows Live Messenger, ésta no ofrece ningún avance positivo.

Después de la prueba fallida, lo siguiente es analizar en detalle cada paquete de la captura, para encontrar una huella, o pista que nos de esperanza para continuar.

Analizando la captura de datos, me encuentro con un paquete algo especial, con el formato siguiente:

Este paquete es utilizado por Windows Live Messenger para establecer la conexión. Una vez que la conexión está establecida, envía regularmente un paquete de las mismas

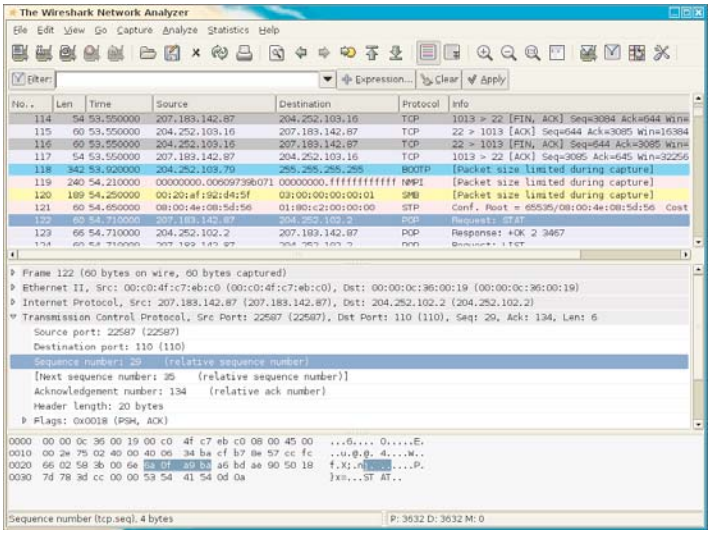

Figura 1. Interfaz de WireShark

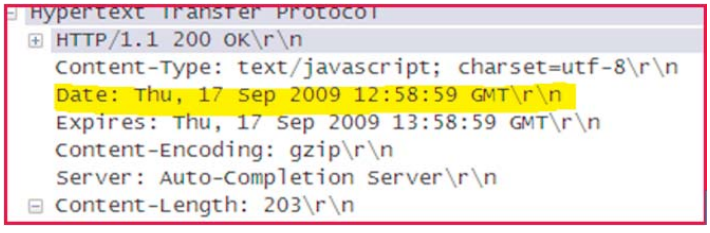

Figura 2. Filtro HTTP.date

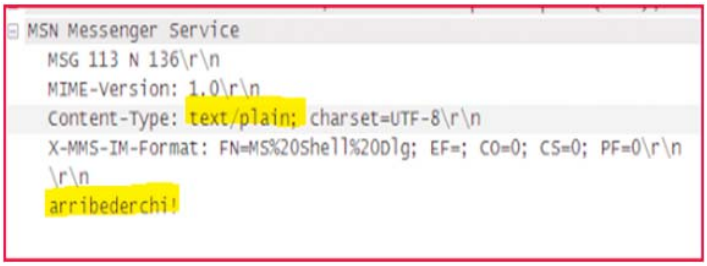

Figura 3. Extracción de conversación en Messenger via WireShark

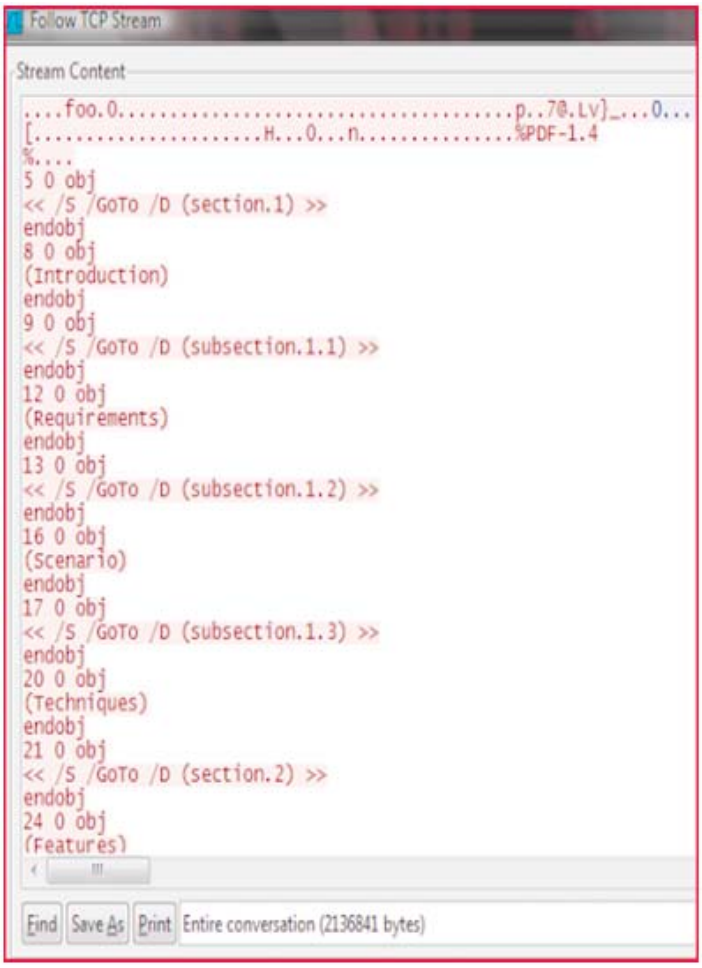

Figura 4. Follow TCP transmisión de paquetes en Messenger

características para asegurarse de que la conexión sigue permanente.

Wireshark tiene, como añadido a la herramienta, opciones de traducción de paquetes. Esta capacidad, dota a la herramienta con un decodificador de paquetes. Con este tipo de opciones, podemos escoger un paquete, y decirle a Wireshark que lo decodifique según un protocolo establecido.

Nuevo intento, fallido, de decodificar todo protocolo UDP como si fuese RTP. Resultados cero.

Inspeccionando el tráfico UDP, me encuentro con que existen patrones que se repiten a la hora del envío de paquetes a través de Windows Live Messenger. En cada envío de paquetes, me encuentro con unos mismos valores para distintos paquete. Este tipo de patrón da a pensar que podemos estar ante algún tipo de . Para ello, voy copiando en una tabla, todos los valores repetidos que me voy encontrando. La lista que obtengo es la siguiente:

44, 48, 66, 4A

Los payloads anteriormente descritos, se refieren a qué tipo de acción va a realizar Windows Live Messenger. Es decir, si va a realizar una conexión, enviar una trama de audio, enviar una trama de vídeo, etc…

Hablando de la herramienta, cabe decir que la mayor característica de la misma, es la capacidad de trabar "offline" con capturas de datos. Una vez que se carga la captura de datos en la herramienta, se procede a la extracción de paquetes de audio y vídeo.

Una vez que la herramienta ha realizado su trabajo, se puede proceder a exportar los datos, pudiéndose utilizar compresores de audio y vídeo, para que la captura final sea más pequeña.

5. CONCLUSIONES

Wireshark utilizado como una herramienta de control de tráfico en la red es en cierto grado de gran potencialidad, debido a que su fortaleza está en la detección de intromisiones externas, apoyándose en protocolos de conocimiento público como lo son el TCP y UDP, este software se implementa para el control de tráfico de aplicaciones de mensajería instantánea.

6. REFERENCIAS

[1] http://en.wikipedia.org/wiki/Pcap

[2] http://www.wireshark.org/docs/dfref/h/http.html

[3]http://www.wireshark.org/docs/wsug_html_chunked/ChIOExportSection.html

[4] http://SoyForense.htm