eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Ciberseguridad

Carla Carbonell Canseco - ccarbonell001@ikasle.ehu.eus / a00573951@tec.mx

Wireshark
Parte 2
Práctica de Wireshark: Descifrado y Análisis de Tráfico HTTPS

Profesora:

Goizalde Badiola Zabala

8 de Diciembre de 2025

Práctica de Wireshark: Descifrado y Análisis de Tráfico HTTPS

Introducción

Esta práctica tiene como finalidad aprender a interpretar tráfico HTTPS descifrado dentro de Wireshark. Para conseguirlo, se utiliza un archivo de captura junto con un fichero de claves TLS (Pre-Master Secret Log), lo que permite reconstruir el contenido real de las comunicaciones cifradas.

Una vez habilitado el descifrado, se examinan las peticiones y respuestas HTTP recuperadas, identificando comportamientos sospechosos relacionados con actividad maliciosa, como la descarga de archivos o intentos fallidos de conexión con servidores remotos.

Los archivos empleados son el paquete de captura decrypting_HTTPS_TLS_traffic.pcap y el fichero de claves KeysLogFile.txt.

Configuración de Wireshark para descifrar TLS

Para que Wireshark sea capaz de interpretar correctamente el tráfico HTTPS del archivo de captura, se realizó la siguiente configuración:

Abrir Wireshark.

Acceder al menú: Edit → Preferences → Protocols → TLS.

En el apartado destinado al archivo de claves, seleccionar KeysLogFile.txt.

Comprobar que las opciones relacionadas con la reconstrucción de registros TLS estén activadas:

Reassemble TLS records spanning multiple TCP segments

Reassemble TLS Application Data spanning multiple TLS records

Guardar los cambios y reiniciar Wireshark si fuera necesario.

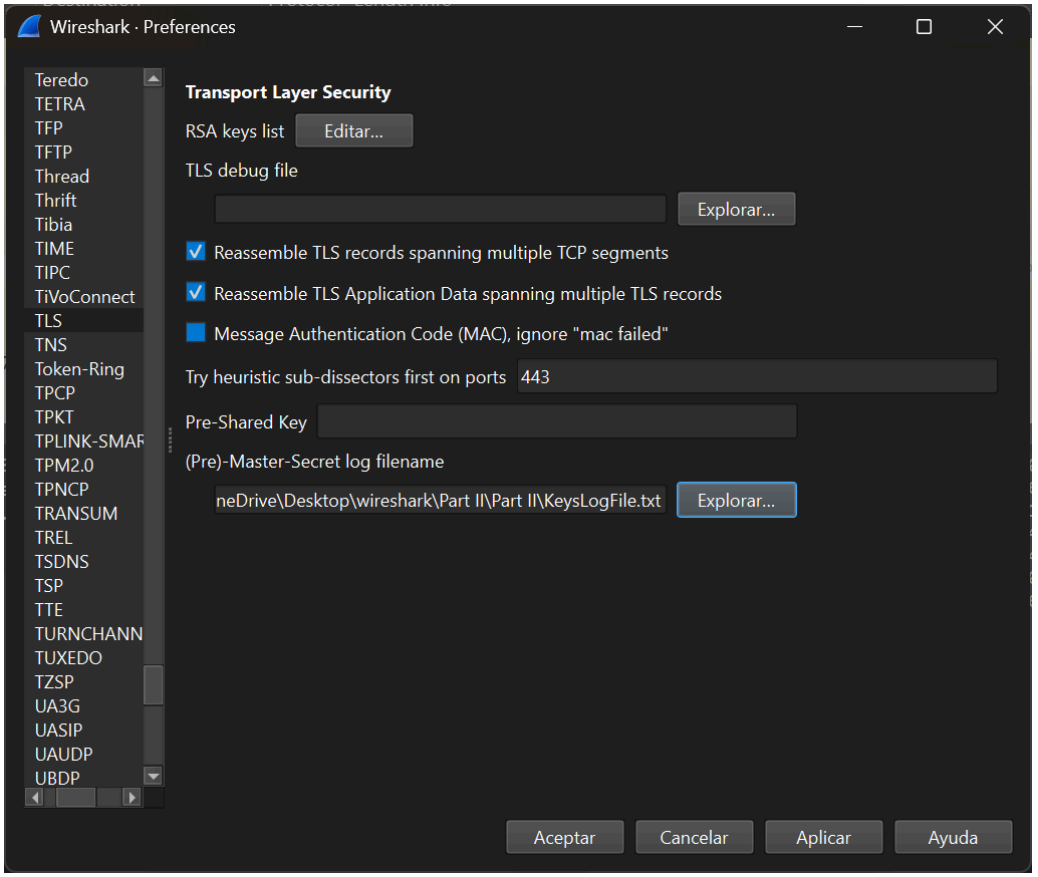

Imagen 1

Visualización del contenido HTTPS una vez descifrado

Después de cargar el archivo de captura y configurar el fichero de claves, Wireshark muestra el tráfico desencriptado como si fuese tráfico HTTP. Para facilitar la visualización se aplicó el filtro:

http

Esto permite ver todas las solicitudes y respuestas generadas durante la sesión original.

Entre los dominios visibles en las comunicaciones destacan varios asociados a servicios legítimos, junto con otros claramente sospechosos, lo cual es un indicio de posible actividad maliciosa. Entre los que aparecen se encuentran:

- config.edge.skype.com
- self.events.data.microsoft.com
- foodsgoodforliver.com
- 105711.com

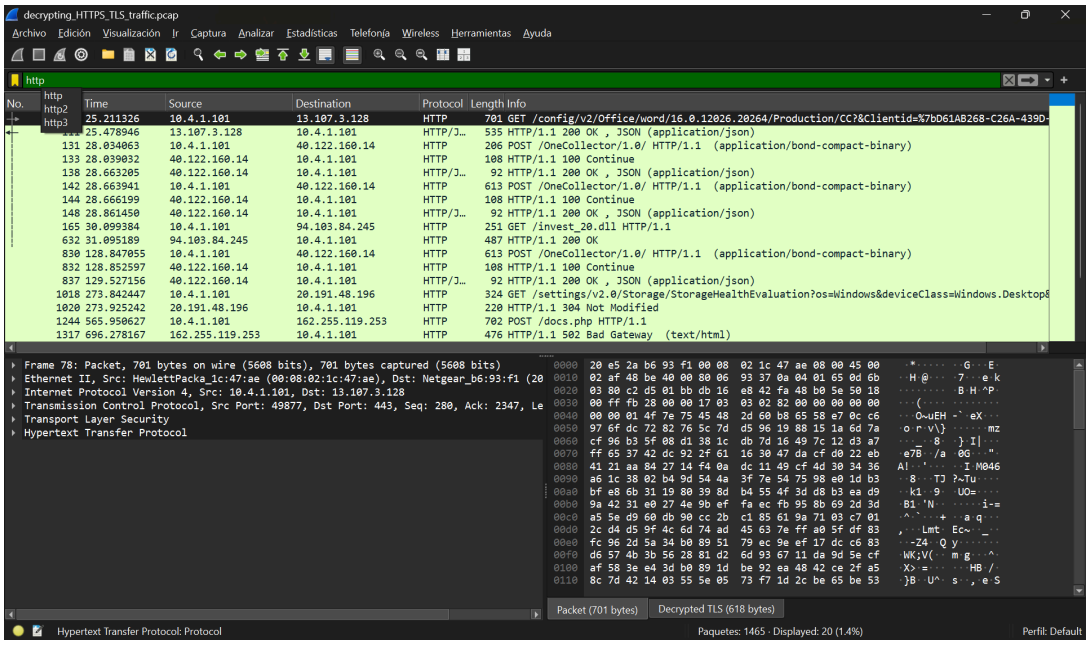

Imagen 2

Exportación de Objetos HTTP

Para examinar los archivos transmitidos durante la captura, Wireshark permite extraerlos directamente.

La opción utilizada fue:

File → Export Objects → HTTP

Desde esta ventana se muestran todos los objetos que el cliente descargó o subió durante la sesión. Cada uno puede guardarse de forma individual para su análisis posterior.

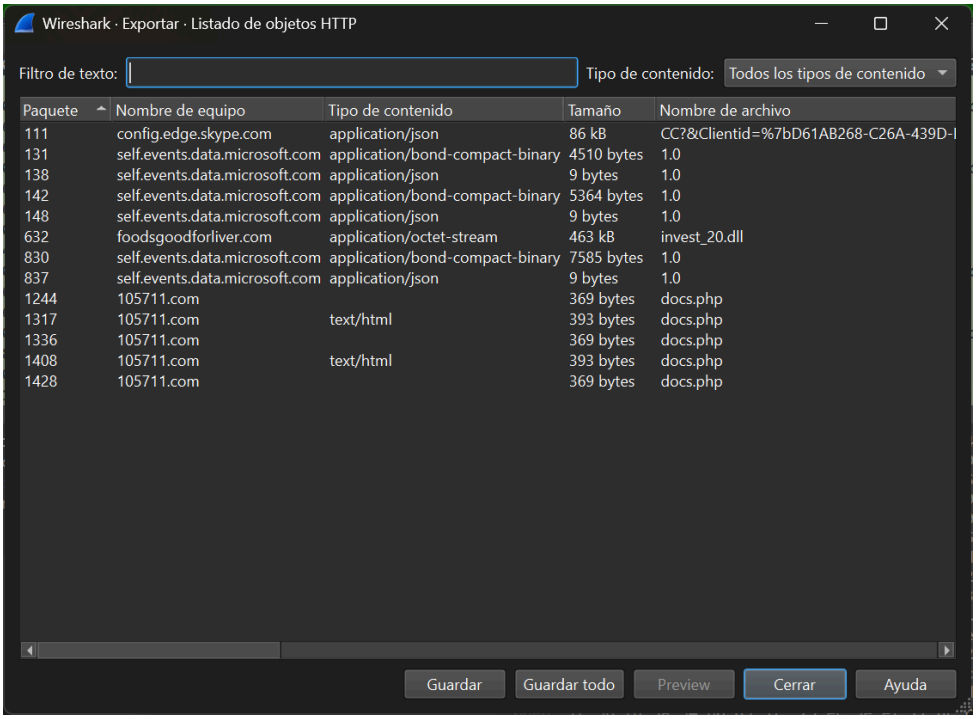

Imagen 3

Análisis del archivo docs.php

Uno de los objetos capturados corresponde al archivo docs.php. Para revisar su contenido se utilizó:

Follow → HTTP Stream

El flujo resultante revela una respuesta HTML indicando un error 502 Bad Gateway. El mensaje hace referencia a un fallo al intentar establecer una conexión TLS con la dirección 162.255.119.253:443.

Este comportamiento es típico de malware que intenta comunicarse con su servidor de comando y control (C2), pero no consigue establecer la conexión. El dominio asociado, 105711.com, refuerza esta interpretación al no tratarse de un servicio legítimo.

Análisis del archivo invest_20.dll

Otro elemento destacado de la captura es el archivo invest_20.dll, que se muestra en forma de datos binarios dentro de un flujo TCP.

Para visualizarlo se utilizó:

Follow → TCP Stream → Raw

El contenido aparece como una secuencia continua de valores hexadecimales, lo que corresponde al cuerpo de un archivo binario. Es habitual que familias de malware descarguen un DLL adicional que actúa como carga útil.

La práctica no requiere ejecutar el archivo, únicamente identificarlo como parte de las actividades sospechosas observadas en la captura.

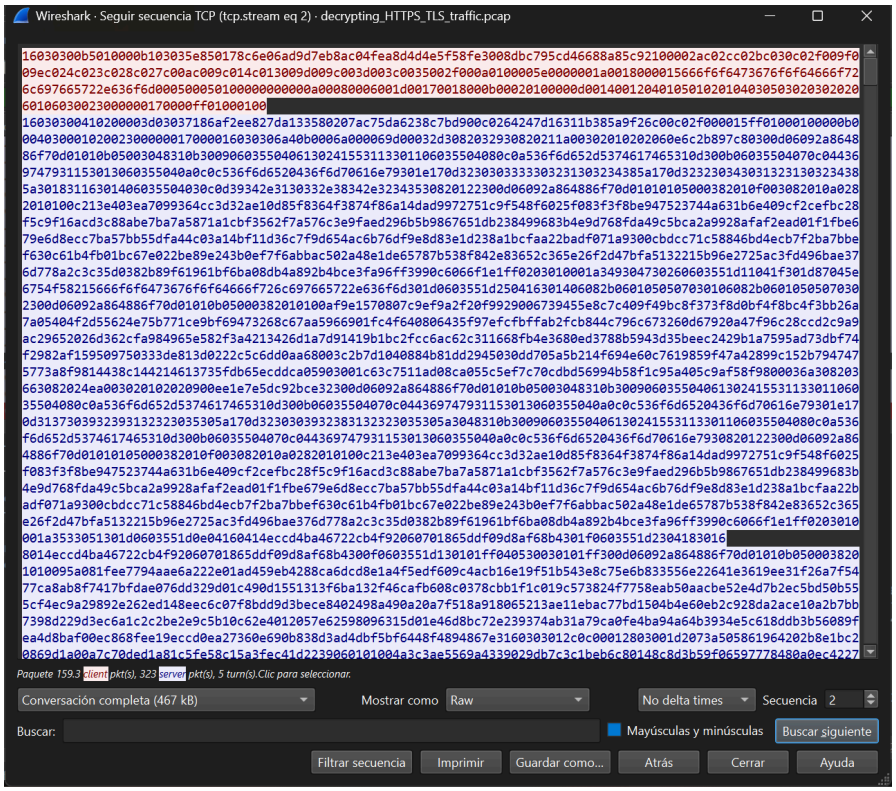

Imagen 4

Conclusión

La práctica permitió comprender cómo se puede descifrar tráfico HTTPS utilizando un archivo de claves TLS y analizar posteriormente el contenido real de la comunicación. Esto facilitó observar peticiones generadas por software potencialmente malicioso, incluyendo intentos de contactar con un servidor remoto y la descarga de un archivo DLL identificado como parte de una posible infección.

El análisis del tráfico descifrado es una herramienta eficaz para estudiar comportamientos maliciosos sin necesidad de ejecutar directamente archivos peligrosos, ofreciendo una forma segura de investigar amenazas dentro de redes o sistemas comprometidos.