CIBERSEGURIDAD

PRÁCTICA 5- WIRESHARK

JULEN BERBETOROS

2025-2026

Grado en Ingeniería Informática de Gestión y Sistemas de Información

ÍNDICE

Tabla de contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 Introducción General del Laboratorio

En este informe recojo todo el proceso que he seguido para realizar los ejercicios del laboratorio de Wireshark y la parte de desencriptación de tráfico TLS. La práctica me ha servido para entender mejor cómo funciona el análisis de red, cómo se pueden extraer archivos de capturas, cómo se identifican credenciales expuestas y, sobre todo, cómo es posible analizar tráfico HTTPS cuando se dispone del Key Log File. El objetivo final ha sido obtener una visión práctica del análisis forense de red y del funcionamiento interno de los protocolos.

2 Laboratorio Wireshark (PARTE I)

2.1 set1.pcap – Introducción a Wireshark

Esta primera captura fue una toma de contacto con Wireshark. Revisé los paquetes base, y utilicé filtros sencillos para entender cómo funciona la herramienta. También aprendí a seguir flujos TCP y a interpretar cabeceras. Aunque es el ejercicio más simple, me ayudó bastante a familiarizarme con la interfaz y las funciones principales.

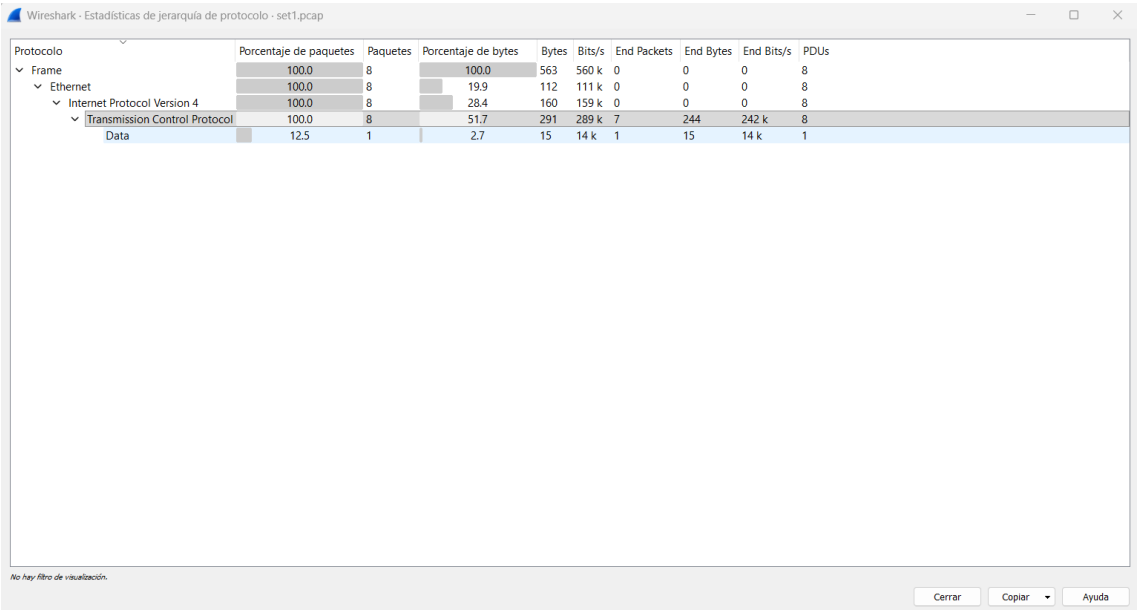

Figura 1 -Estadísticas de jerarquía de protocolo

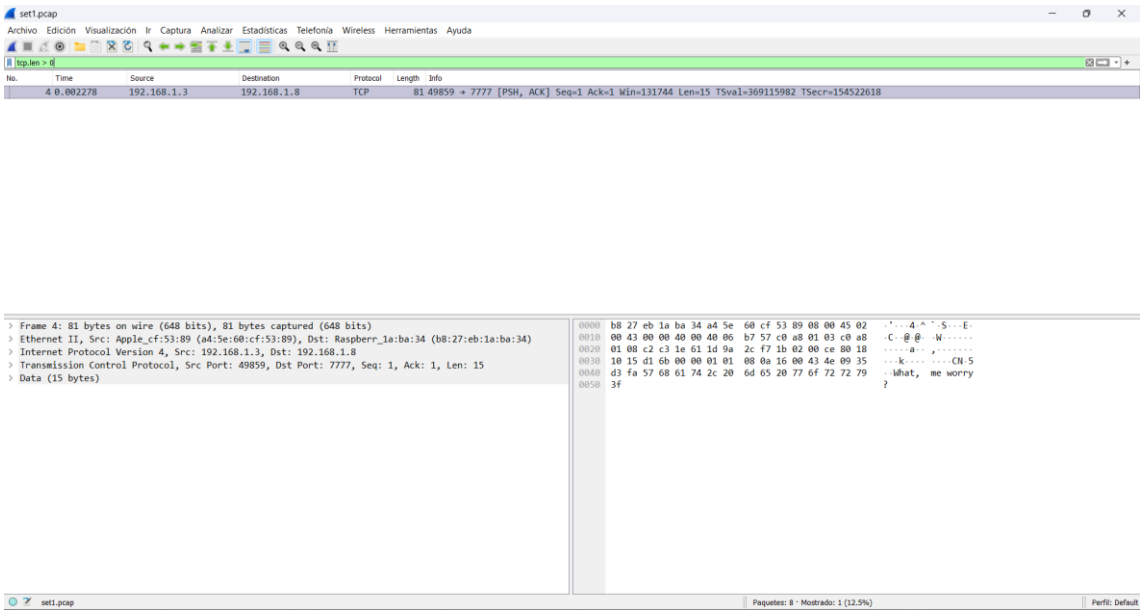

Figura 2 - Filtro básico en captura de paquetes

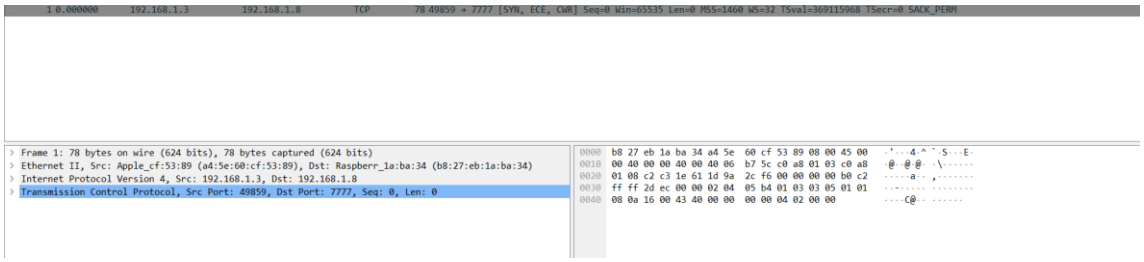

Figura 3 - Observación TCP

2.2 2. set2.pcap – Extracción de Imágenes y Archivos

En esta captura ya pude meterme más en materia. Utilizando 'Export Objects → HTTP' extraje varias imágenes y archivos que estaban siendo transferidos. Fue interesante ver cómo Wireshark permite reconstruir archivos literalmente a partir del tráfico.

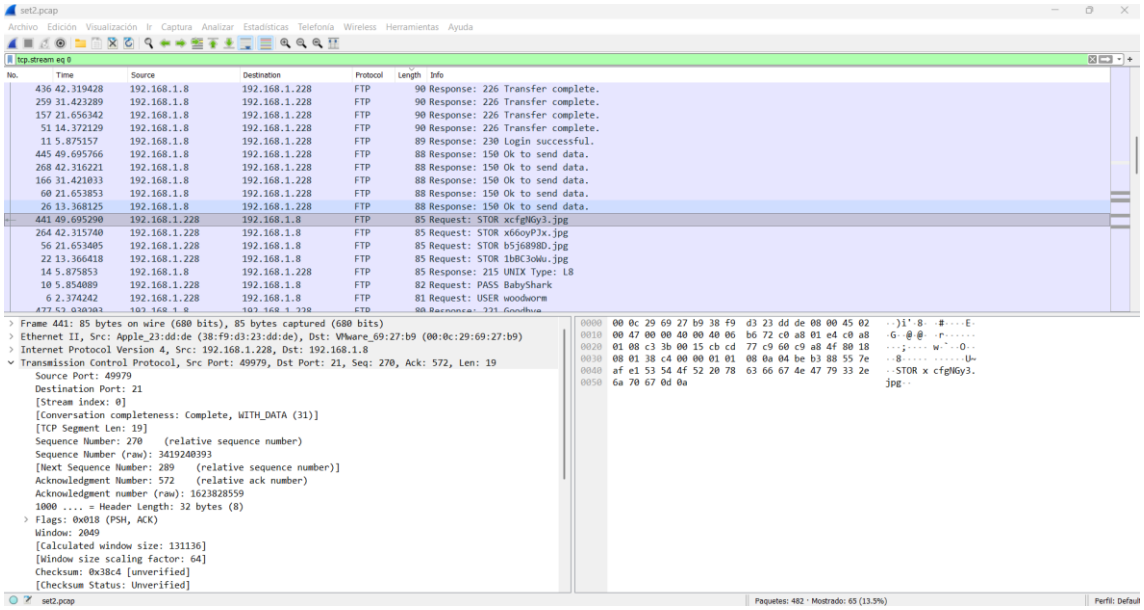

Figura 4 -Filtro tcp.stream eq 0 set2

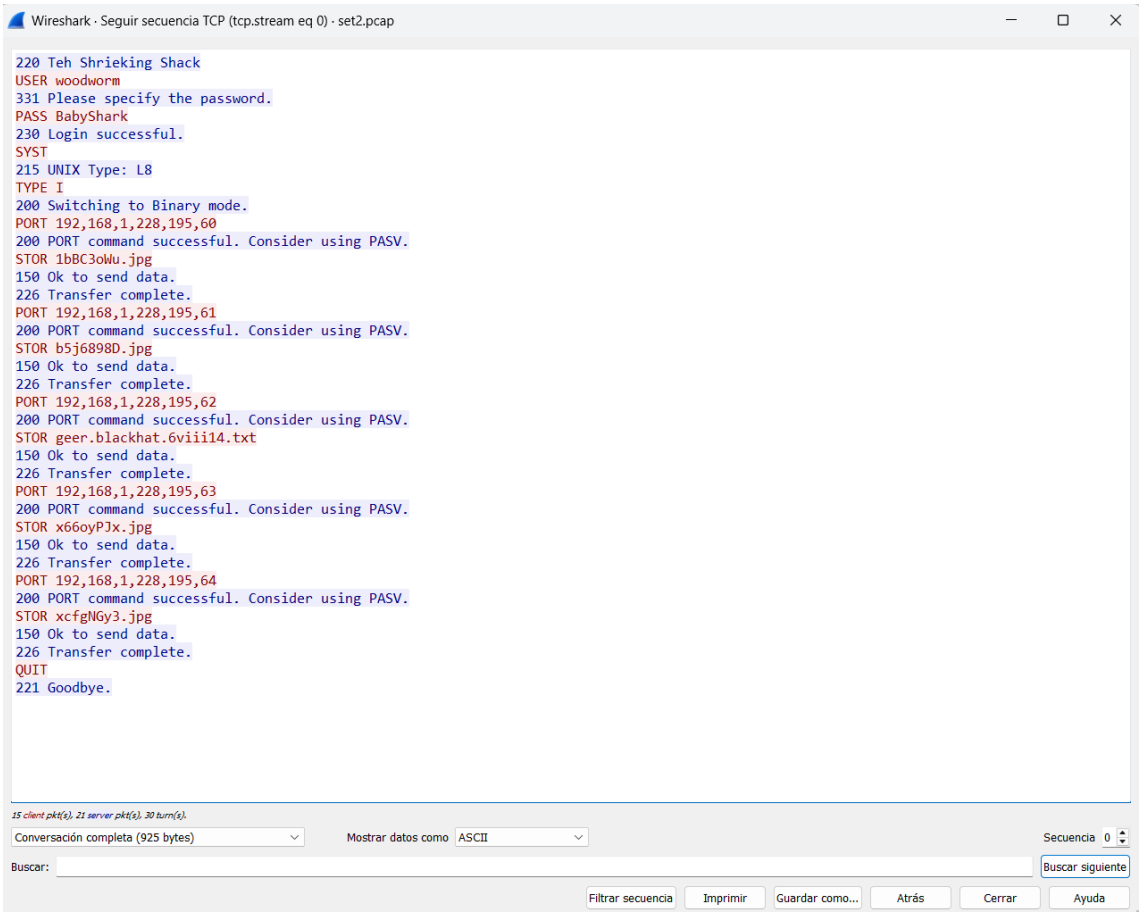

Figura 5 - Seguimiento de secuencia TCP set2

Imágenes obtenidas

Figura 6 -Imagen 1

Figura 7 - Imagen 2

Figura 8 - Imagen 3

Figura 9 - Imagen 4

2.3 3. set3.pcap – Reconstrucción de Archivo Multimedia

Aquí reconstruí un archivo multimedia que había sido fragmentado en varios segmentos TCP. Seguí el flujo con 'Follow TCP Stream', exporté el contenido y comprobé que se podía reproducir correctamente. Este ejercicio me hizo entender realmente cómo viajan los datos en la red y cómo un archivo puede llegar dividido en muchas partes.

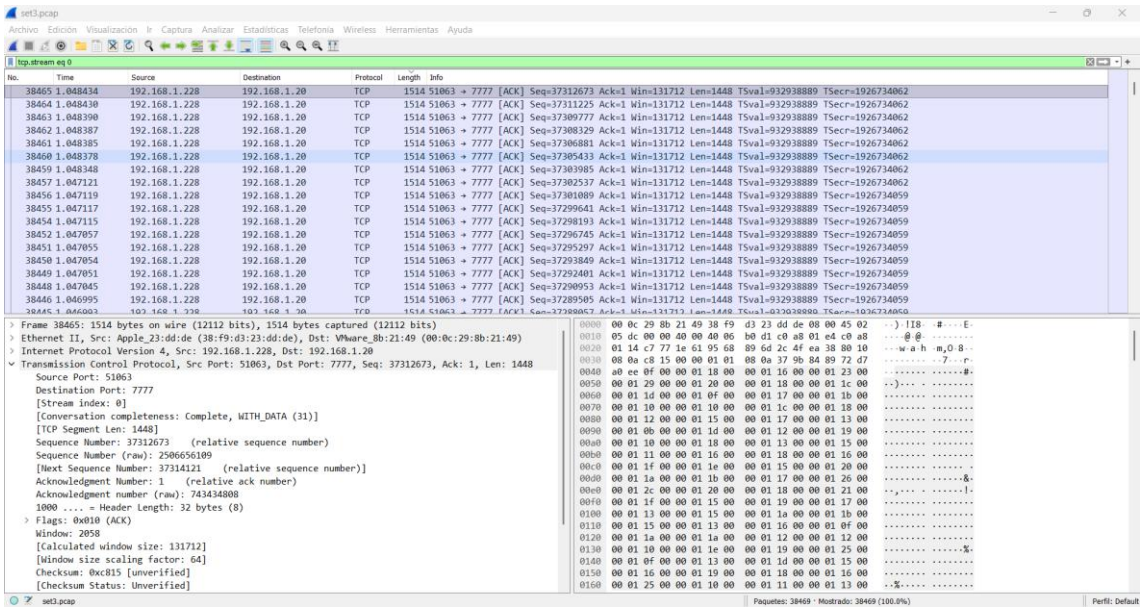

Figura 10 - Filtro tcp.stream eq 0 set3

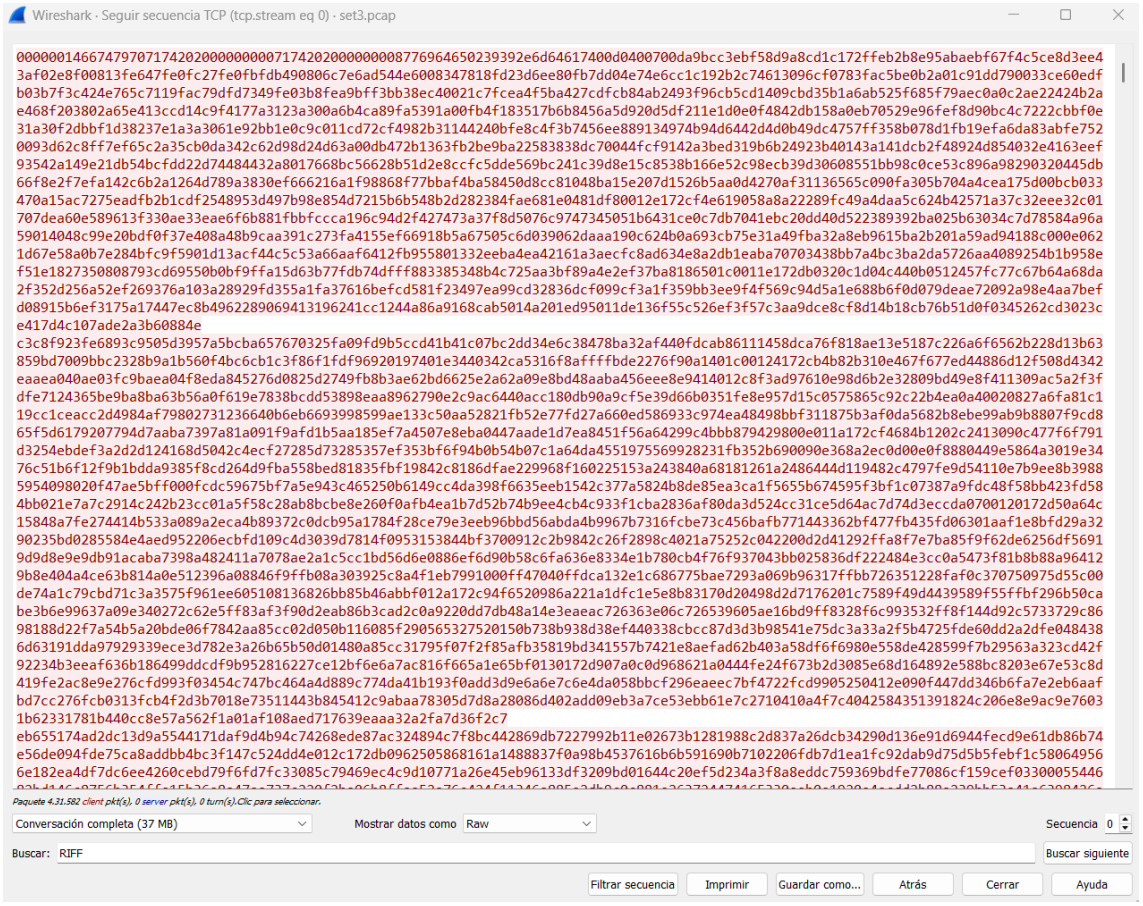

Figura 11- Seguimiento de secuencia TCP set3

Figura 12 - Archivo multimedia

Archivo multimedia obtenido

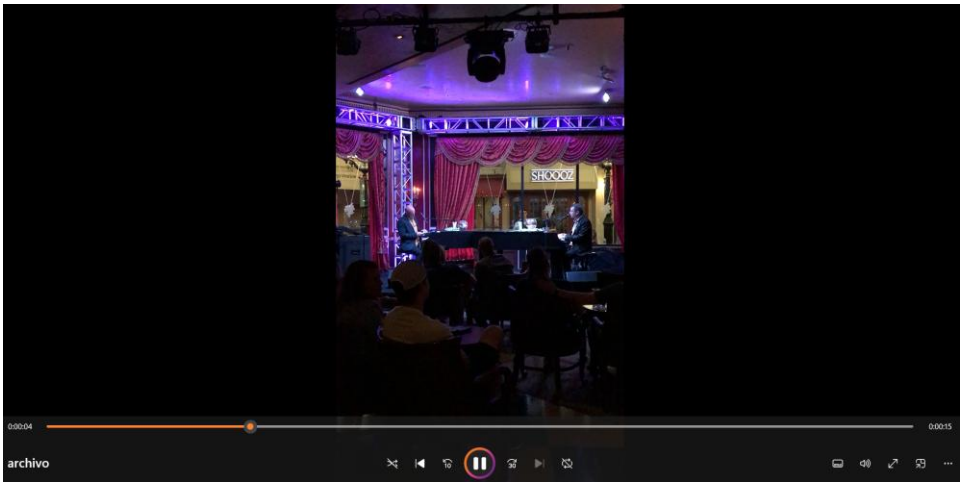

Figura 13 - Reconstrucción del archivo

2.4 4. set4.pcap – Credenciales en Texto Plano

En esta captura busqué credenciales enviadas sin cifrado.

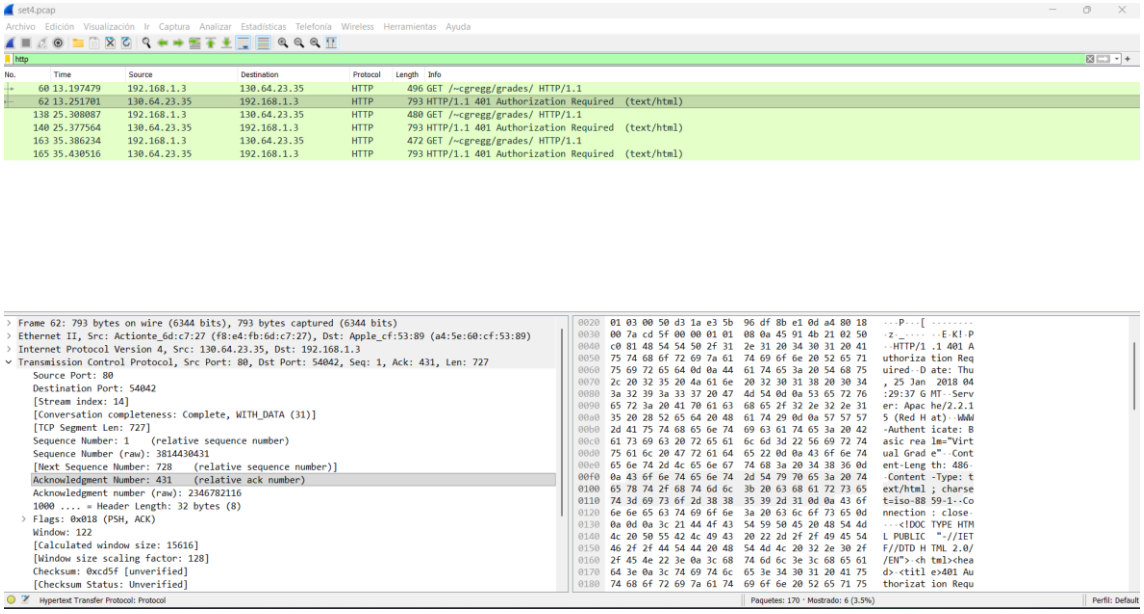

Figura 14 - Filtro http

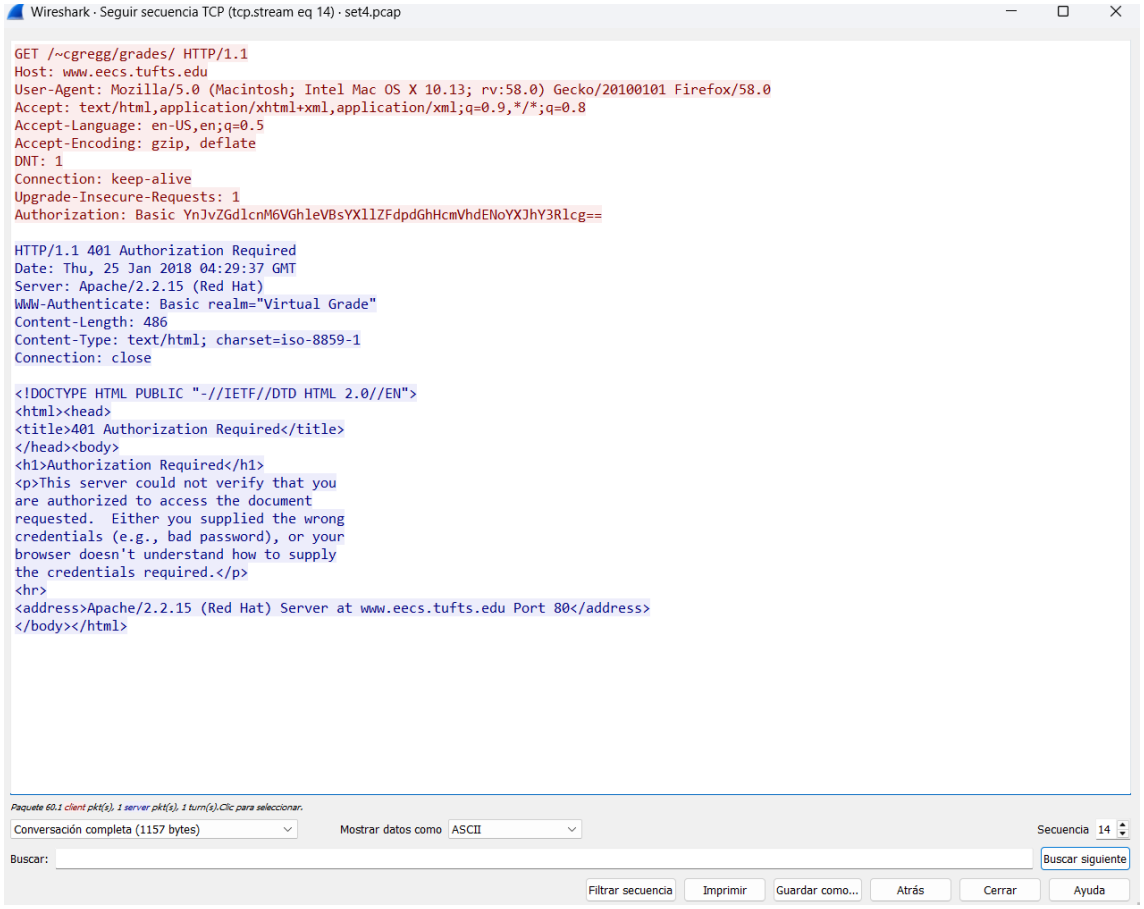

Figura 15 - Seguimiento de secuencia TCP set4

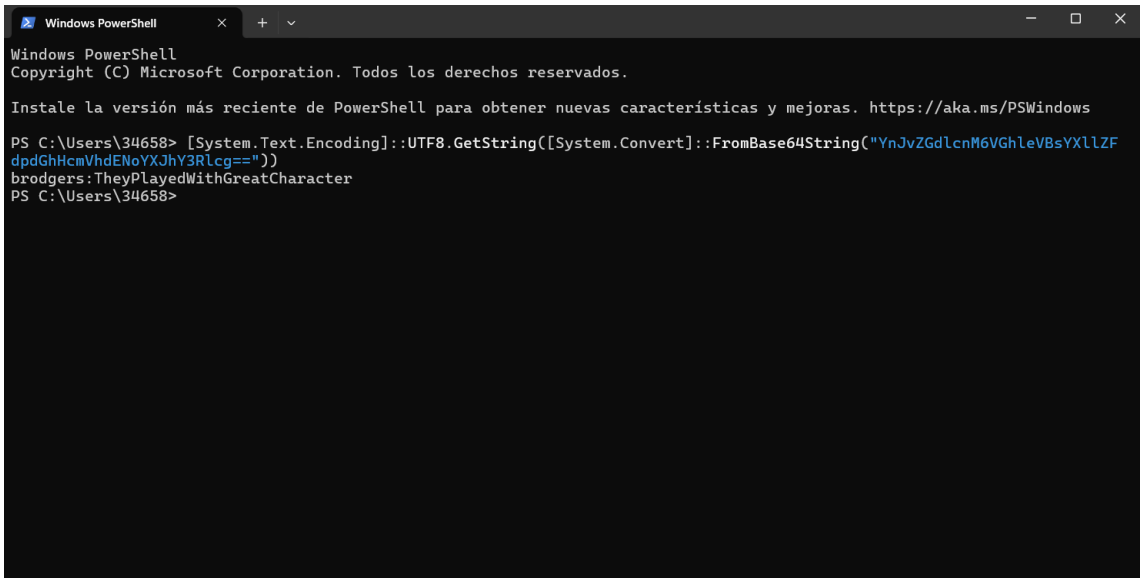

Figura 16 - Conversión de Base64 set4

Credenciales obtenidas

User: brodgers

Password: TheyPlayedWithGreatCharacter

2.5 5. set5.pcap – Credenciales en PCAP Grande

En este ejercicio tuve que revisar una captura mucho más grande. Aun así, gracias a los filtros y a buscar patrones comunes (username=, password=...), pude localizar otro envío de credenciales. Fue más laborioso, pero demuestra que si no hay cifrado, siempre es posible recuperar este tipo de información.

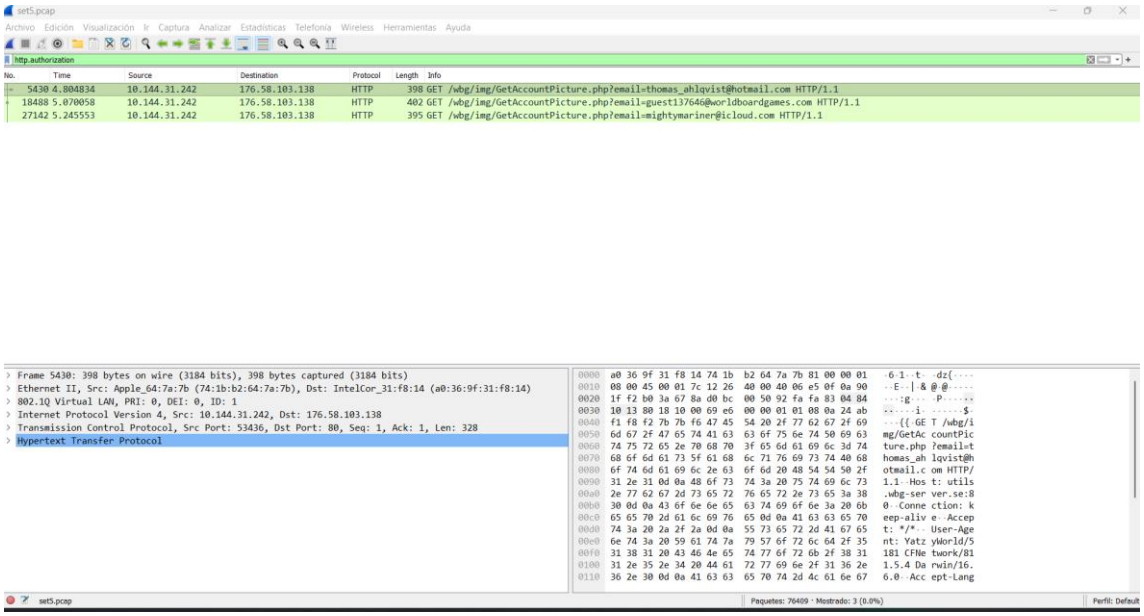

Figura 17 - Filtro http.authorization

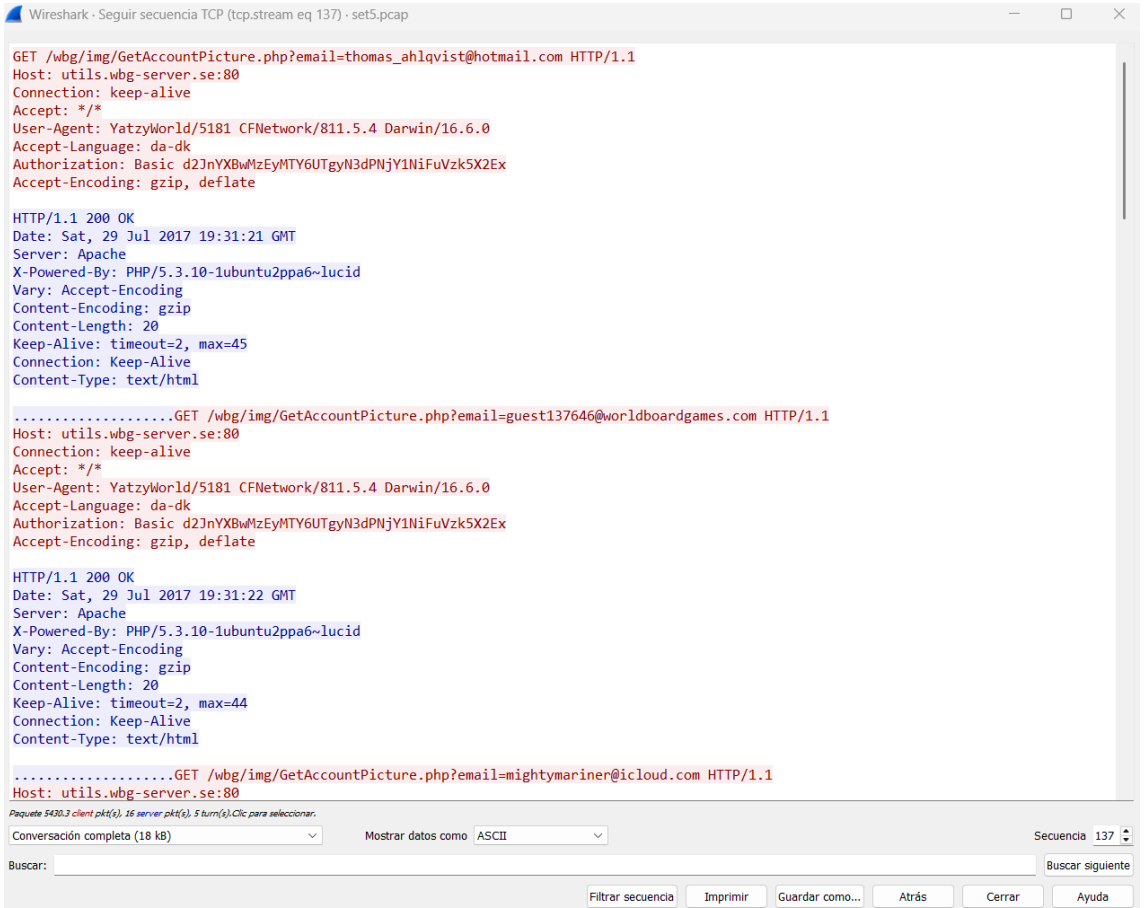

Figura 18 - Seguimiento de secuencia TCP set5

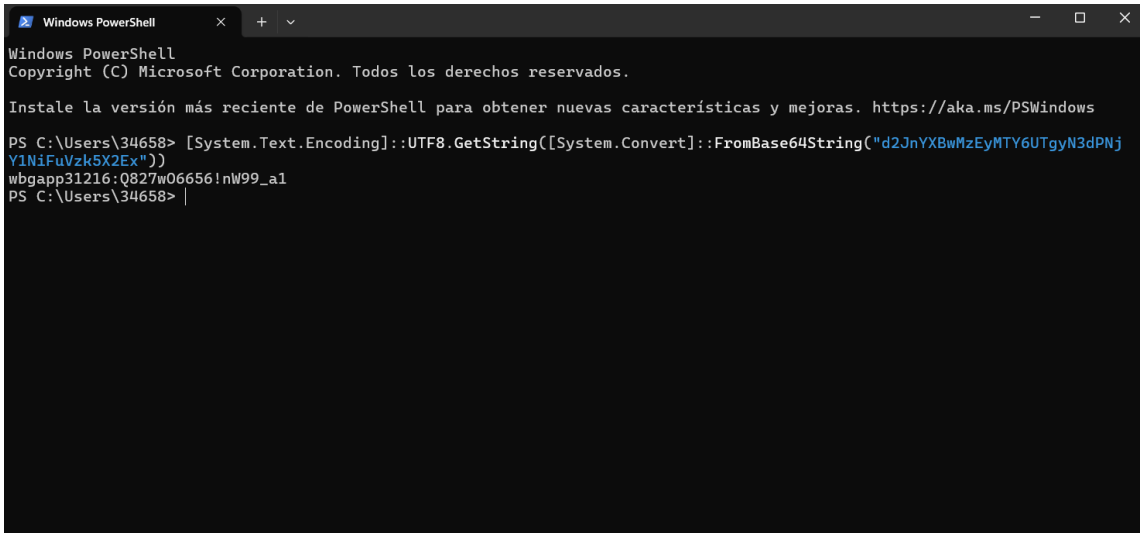

Figura 19 - Conversión Base64 set5

Credenciales obtenidas

User: wbgapp31216

Password: Q827w06656!Nw99_al

2.6 6. set6.pcap – Tráfico con Malware / Malspam

Este ejercicio fue bastante interesante. En la captura aparecían solicitudes HTTP que descargaban archivos ejecutables desde dominios claramente sospechosos. Analicé los patrones del tráfico y vi comportamientos típicos de campañas de malspam. También aparecían conexiones repetitivas y descargas de .exe, algo que en un entorno real sería una alerta inmediata.

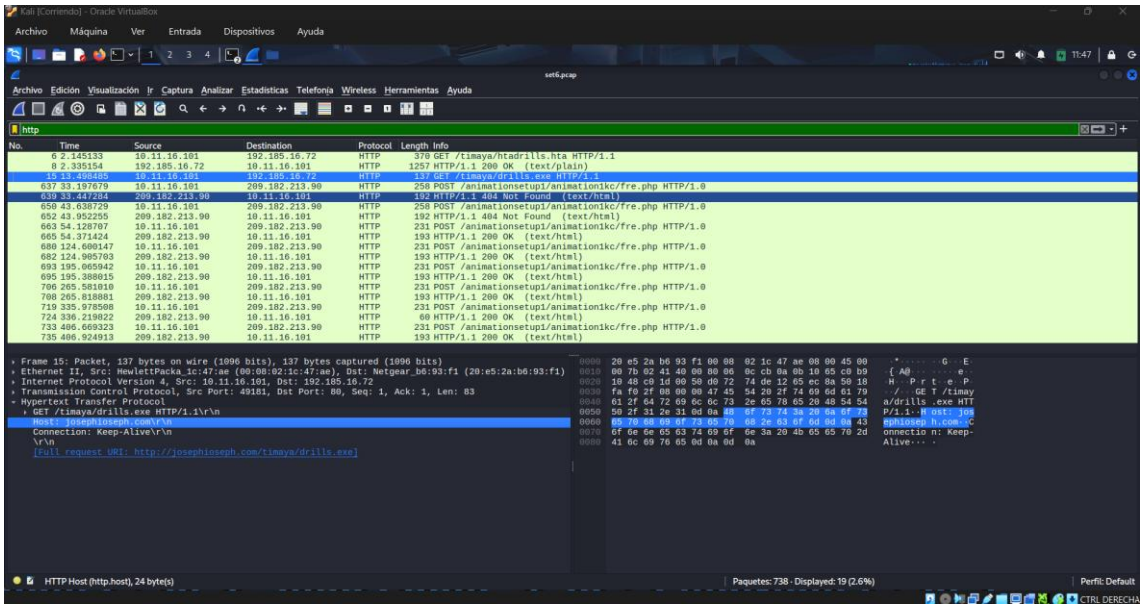

Figura 20 - Filtro http set6

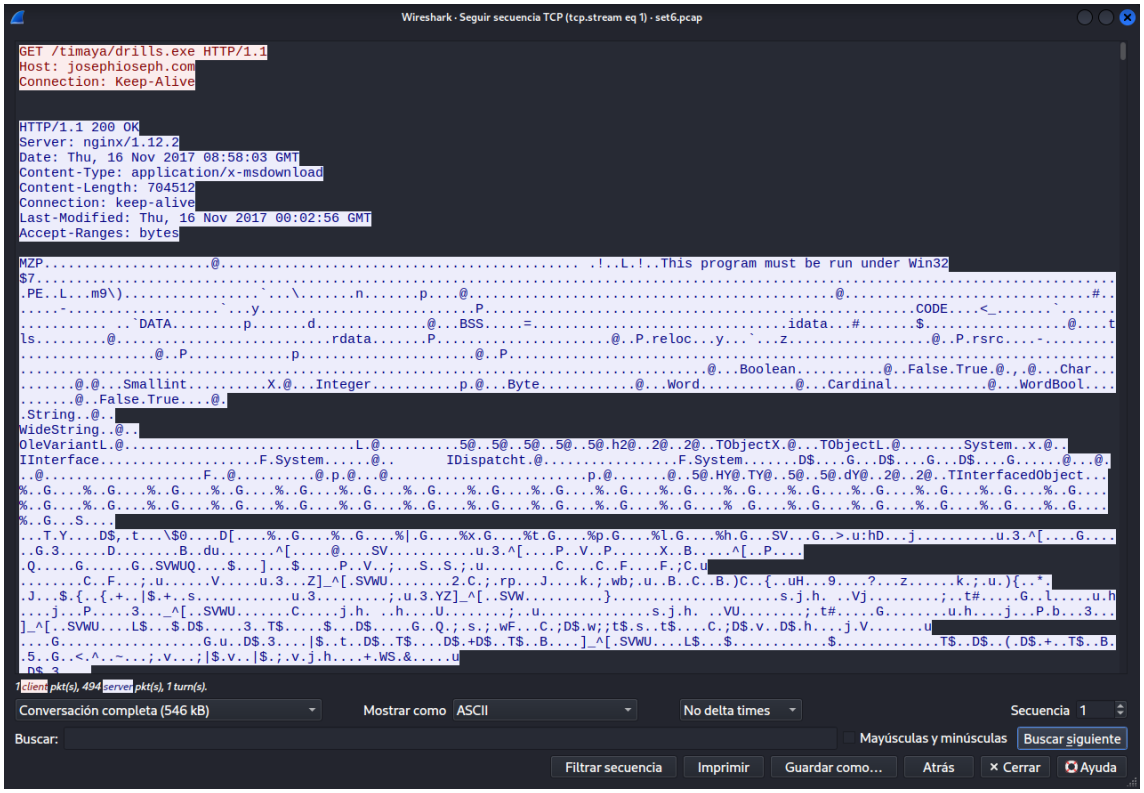

Figura 21 - Seguimiento de secuencia TCP set6

Figura 22 - Obtención .exe

3 Desencriptación TLS/HTTPS (PARTE II)

3.1 Problema del Tráfico Cifrado

En esta parte aprendí que analizar tráfico HTTPS no es tan simple como aplicar un filtro. Como está cifrado con TLS, lo único que suele verse es 'Application Data'. Para poder ver el contenido real es necesario tener el Key Log File generado por el navegador en el momento de la captura.

3.2 Carga del Key Log File en Wireshark

Tras abrir la captura y cargar el archivo KeysLogFile.txt en la configuración del protocolo TLS, la magia ocurrió: todo el tráfico HTTPS pasó a mostrarse descifrado. Pude ver peticiones GET, POST, encabezados y contenido completo. Sin el Key Log File esto sería imposible.

3.3 Análisis del Tráfico HTTP Desencriptado

Una vez descifrado, filtré las peticiones buscando extensiones sospechosas como .exe o .dll. Localicé una descarga directa de un archivo ejecutable que claramente no era legítimo. Este tráfico, si estuviera en una red real, sería un indicador claro de infección o actividad maliciosa.

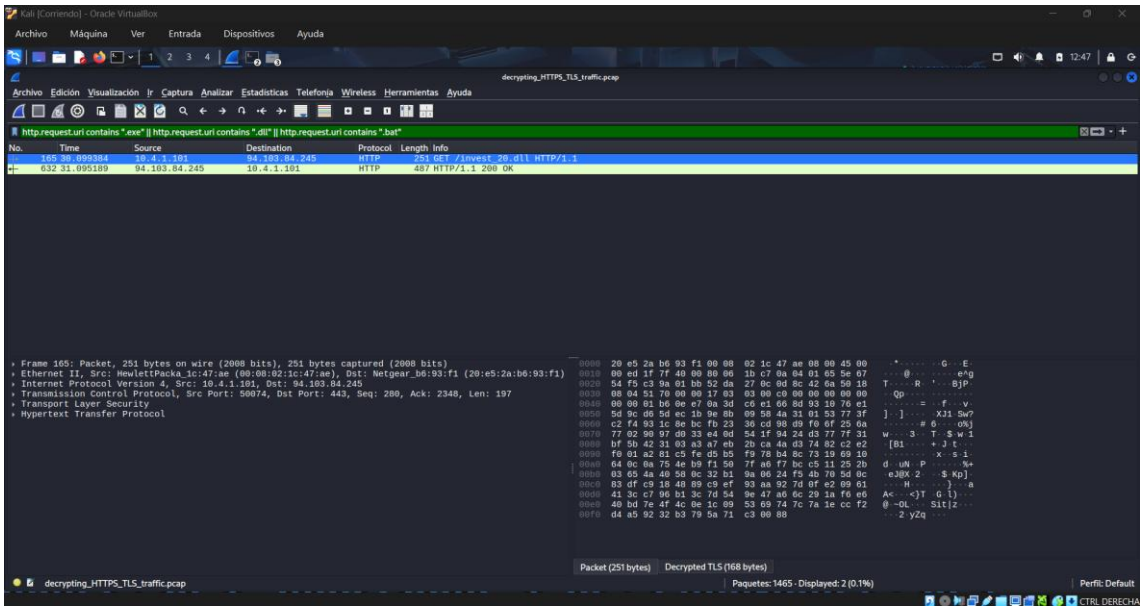

Figura 23 - Filtro http.request.uri contains (.exe, .dll, .bat)

3.4 Extracción, Hash y Archivo Malicioso

Exporté el archivo malicioso desde 'Export Objects → HTTP' y calculé su hash SHA-256 para poder investigarlo en plataformas de análisis. El hash sirvió como identificador único para consultar si ya estaba catalogado como malware, lo cual facilita enormemente la labor de análisis.

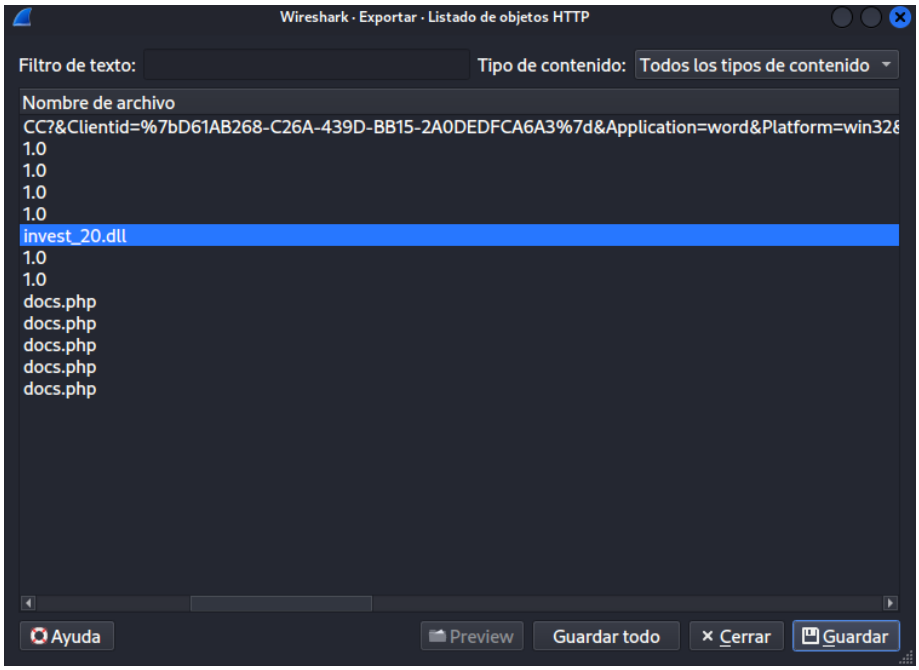

Figura 24 - Exportación de objetos (.dll)

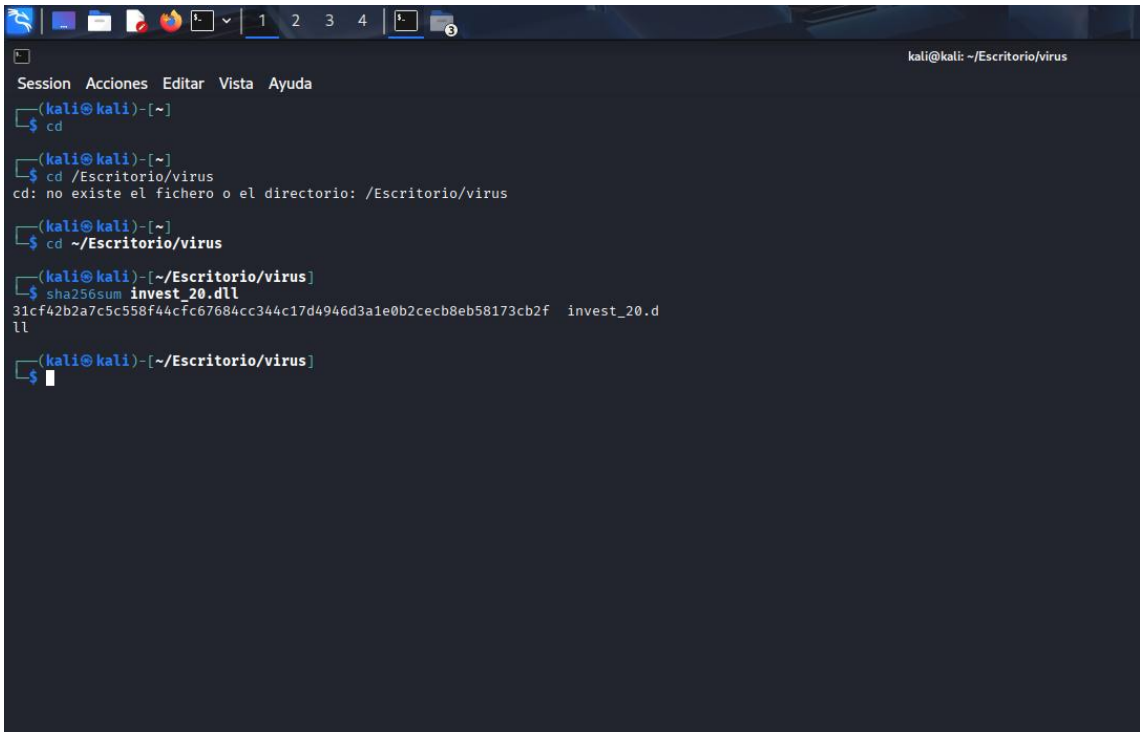

Figura 25 - Obtención del hash del archivo

4 Reporte de Inteligencia de Amenazas (TI)

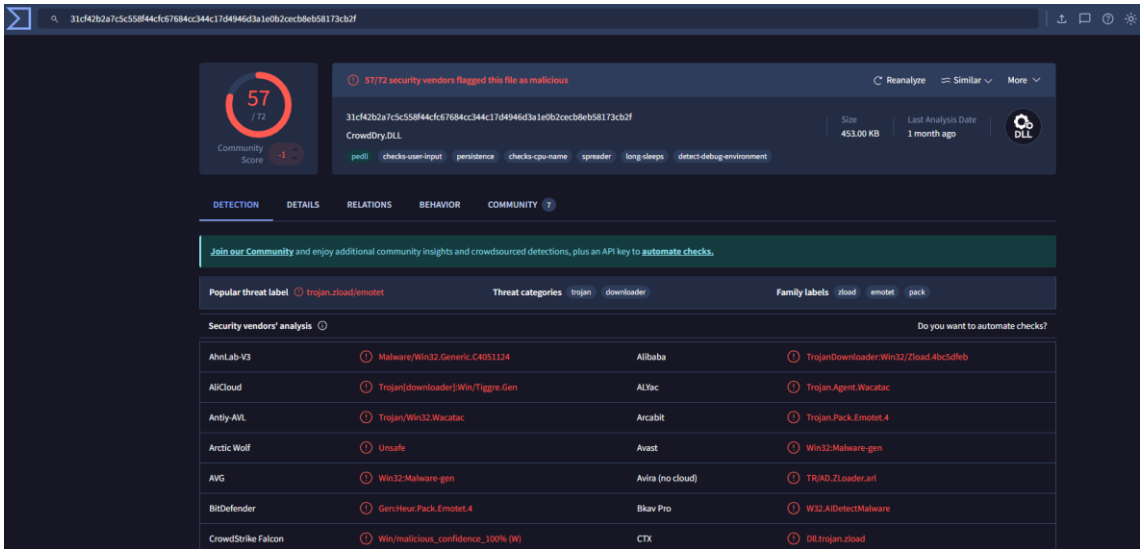

Figura 26 - Reporte del hash

4.1 Resumen Ejecutivo

Durante el análisis de tráfico de red capturado en la sesión HTTPS, se identificó la descarga de un archivo malicioso cifrado mediante TLS. Gracias al uso del archivo de registro de claves TLS, se pudo desencriptar el tráfico y analizar la sesión HTTP subyacente. El archivo descargado, identificado como un troyano / downloader, tiene la capacidad de comprometer sistemas mediante la ejecución de código malicioso. Este hallazgo indica un riesgo potencial de infección y exfiltración de datos en la infraestructura analizada.

4.2 Contexto del Incidente

El análisis se realizó sobre la captura de tráfico decrypting_HTTPS_TLS_traffic.pcap, proveniente de una sesión de navegación segura pero comprometida. Para poder examinar el contenido cifrado, se utilizó el archivo KeysLogFile.txt, que permitió a Wireshark descifrar la capa TLS y revelar las transacciones HTTP. Esto permitió identificar la descarga del archivo malicioso invest_20.dll desde el host sospechoso.

4.3 Indicadores de Compromiso (IOC)

•Nombre del archivo extraído: invest_20.dll

•URL de descarga completa: https://foodsgoodforliver.com/invest_20.dll

•Hash SHA 256:
31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f

Este hash es el identificador único y el IOC principal para detección y bloqueo en sistemas de seguridad.

4.4 Análisis e Inteligencia de Amenazas (TI)

•Plataforma de análisis utilizada: VirusTotal

•Resultados: 57/72 motores detectan el archivo como malicioso

•Categoría de amenaza: Trojan / Downloader

•Descripción: El archivo se clasifica como un troyano capaz de descargar componentes adicionales, lo que lo convierte en un riesgo importante para la seguridad de la red y los sistemas involucrados.

4.5 Recomendaciones de Mitigación

1. Bloquear el hash SHA 256 31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f en los sistemas de seguridad y antivirus corporativos.

2. Bloquear el dominio o la IP del host de descarga en firewalls y proxies de la red.

3. Aislar y eliminar cualquier archivo similar encontrado en los sistemas de la infraestructura.

4. Monitorear posibles comunicaciones de salida a hosts sospechosos.

5. Educar a los usuarios sobre la no ejecución de archivos descargados desde fuentes desconocidas o no confiables.

Elemento	Detalle
Nombre del archivo	invest_20.dll
URL de descarga completa	https:// foodsgoodforliver.com /invest_20.dll
Hash SHA-256	31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f
Categoría de amenaza	Trojan / Downloader
Motores de detección (VirusTotal)	57/72 detectan como malicioso
Protocolo comprometido	TLS (HTTPS)
Método de análisis	Wireshark + Key Log File para desencriptación TLS
Acción recomendada	Bloqueo del hash en sistemas, bloqueo de dominio/IP, aislamiento de archivos

Tabla 1 – Resumen del análisis