Práctica 1 Análisis y Etiquetado de Tráfico de Red

Pablo Chantada Saborido

1. Captura y Análisis de Tráfico (Wireshark)

Utilizando Wireshark para la captura de paquetes (`archivo .pcap`), podemos analizar el flujo de datos de nuestra red (WiFi en este caso). A continuación, se muestran diferentes capturas del tráfico. Durante el proceso de captura estaban abiertas diversas aplicaciones como: Steam, Firefox, y otras en segundo plano como los controladores de ASUS.

1.1. Tráfico hacia ASUS (ARP)

Se observó tráfico ARP (Figura **??**) (proceso de conectar una dirección IP dinámica a la dirección MAC de una máquina física) generado por el hardware ASUS, en el que se puede apreciar la resolución de direcciones MAC a direcciones IP locales. De esta forma no solo sabemos la dirección IP de nuestro dispositivo, sino también que nuestra placa base u otros componentes son de ASUS. Esto podría permitir *exploits* de vulnerabilidades de la marca, similar a lo ocurrido con los chips Intel.

Figura 1: Captura de tráfico ARP ASUS

1.2. Tráfico hacia Google (DNS)

Se capturó una respuesta DNS (Figura **??**) para el dominio `clients3.google.com`, que incluye un registro CNAME (sirve para crear un alias de un dominio hacia otro dominio) y la dirección IP resuelta (`216.58.215.174`). En este caso vemos que el CNAME pasa del dominio `clients3.google.com` a `clients.l.google.com`, para terminar la resolución a la IP real.

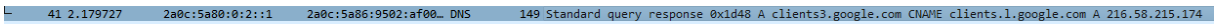

Figura 2: Captura de tráfico DNS Google

1.3. Tráfico hacia Steam (DNS)

Se capturó una consulta DNS (Figura **??**) de tipo AAAA (apunta un dominio o subdominio a una dirección IPv6, igual que el A lo hace a una IPv4) para el dominio `store.steampowered.com`, correspondiente a la plataforma Steam. Este tráfico, además, se realizó usando la aplicación de Steam y no la página web. Es posible que ciertos paquetes estén presentes en la app y no en la web, pero el funcionamiento (debería) ser el mismo.

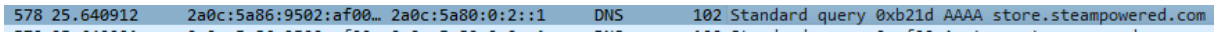

Figura 3: Captura de tráfico DNS Steam

1.4. Creación del dataset

Con la captura actual, no se debería crear un dataset público. Como se menciona anteriormente, podemos discernir componentes del sistema, direcciones IP, aplicaciones [1], etc. Si quisiéramos generar un dataset público con la captura, deberíamos ocultar las IPs privadas, componentes, aplicaciones que puedan presentar vulnerabilidades, etc.

1.5. Intercepción de Credenciales en Texto Plano

La web utilizada para demostrar las vulnerabilidades de **HTTP** contra **HTTPS** fue testphp.vulnweb.com/signup.php, ya que los datos enviados por el usuario no viajan cifrados (Figura **??**).

Tras completar el registro, el servidor devuelve una página de confirmación que muestra en claro todos los datos introducidos por el usuario (Figura **??**), incluyendo contraseña y número de tarjeta de crédito:

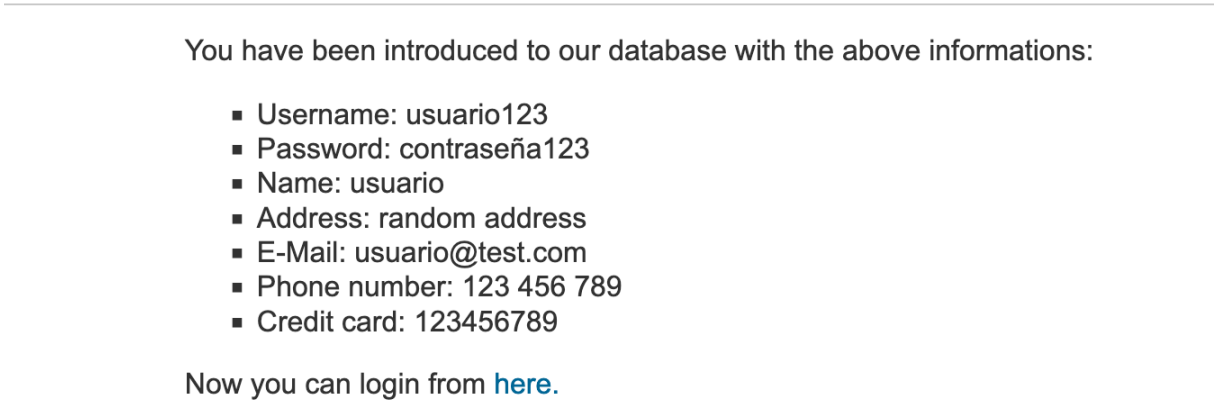

Figura 4: Página de confirmación de registro en VulnWeb

Al analizar la captura con Wireshark, inspeccionamos el paquete `HTTP POST` y podemos observar directamente las credenciales de registro. En el panel resaltado (Figura **??**) se puede ver el formulario con todos los campos enviados (`uname`, `upass`, `ucc`, `uemail`, `uphone`, `uaddress`):

Esto demuestra el grave riesgo de seguridad que supone no implementar protocolos de cifrado (TLS) en formularios de autenticación, permitiendo a cualquier atacante en la red local (ataques *Man-in-the-Middle*) obtener las cuentas de los usuarios, siempre y cuando obtengan el paquete en la conversación entre el servidor y el usuario.

[1] En el caso de Steam, en 2025 se filtraron 90 millones de cuentas. Permitiendo a un atacante exprimir vulnerabilidades similares

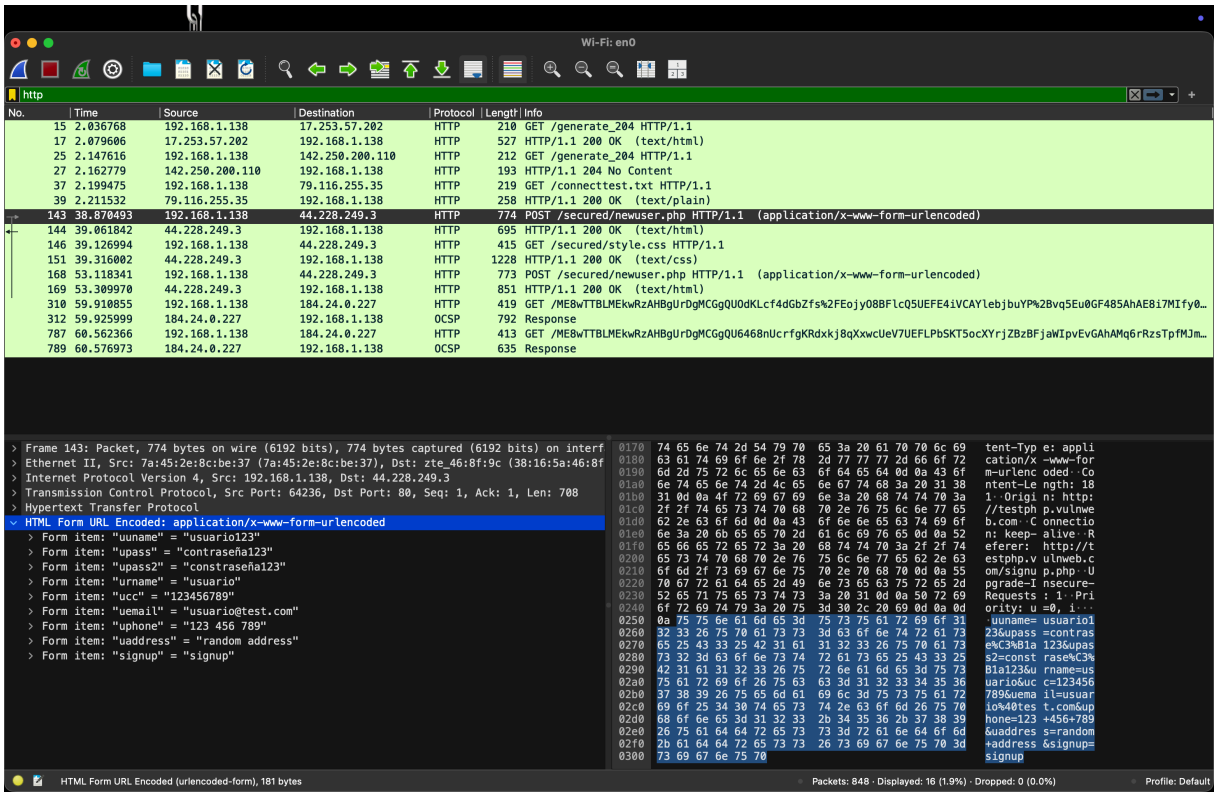

Figura 5: Wireshark Intercepción del POST HTTP con credenciales en texto plano

2. Extracción y Generación de Flujos de Tráfico

El script `traffic.py` agrupa los paquetes individuales en **flujos de tráfico**.
Un flujo se define mediante una lista de 5 elementos:

`[IP Origen, IP Destino, Puerto Origen, Puerto Destino, Protocolo]`

Iterando sobre la captura podemos, usando el puerto de origen y destino, identificar a qué tipo de tráfico corresponde (HTTP: puerto 80, HTTPS: puerto 443, DNS: puerto 53). Los paquetes restantes se identifican como ICMP [2] o como otros.

3. Análisis Cuantitativo de Flujos (`flows.csv`)

Los flujos se mantuvieron abiertos acumulando paquetes y solo se cerraron al detectar banderas de finalización o reseteo (**flags FIN o RST**). Esta lógica asegura que, si dos máquinas se comunican por los mismos puertos de forma repetida pero en momentos distintos (tras un cierre de conexión), se contabilicen como flujos separados.

El análisis estadístico de esta captura permite extraer los siguientes datos globales:

- **Total de flujos registrados:** 56
- **Volumen total de tráfico:** 6.428 paquetes procesados, sumando un total de **8.284.303 bytes** transferidos.

[2] Los mensajes del Internet Control Message Protocol no precisan de puertos, al ser un protocolo de capa 3 va directamente en la IP, sin necesidad de multiplexar servicios.

- **Estado de las conexiones:** De los 56 flujos, solo **13** se cerraron correctamente (capturando la finalización de la conexión), mientras que **43** flujos permanecían abiertos en el momento en que se detuvo la captura de tráfico.

Al analizar los flujos con mayor volumen de datos (Top 3), observamos que todos corresponden (posiblemente) a descargas o recepciones de contenido pesado desde la misma IP remota (`151.101.135.52`, puerto `443` HTTPS) hacia puertos dinámicos de nuestra máquina local (`192.168.1.138`):

Ranking	Puerto local	Paquetes	Bytes totales	Tamaño medio por paquete
1ž	65095	3.526	5.132.277 bytes	1.455,55 bytes
2ž	65094	1.676	2.424.014 bytes	1.446,31 bytes
3ž	65093	117	138.192 bytes	1.181,13 bytes

Cuadro 1: Top 3 flujos con mayor volumen de datos

El hecho de que el tamaño medio de los paquetes en los flujos principales esté tan cercano a los **1500 bytes** (MTU estándar de Ethernet) indica que la red estaba aprovechando al máximo la capacidad del canal para transferir archivos o flujos multimedia pesados, seguramente correspondientes a una descarga.

4. Conclusión y Aplicación en Ciberseguridad

La práctica demuestra cómo los datos sin cifrar son interceptables y la facilidad para generar conjuntos de datos orientados a Machine Learning. A diferencia del análisis de un solo paquete aislado (que apenas nos revela IPs, puertos u otros datos simples), evaluar una conversación completa extrayendo métricas de flujo (como la duración, el recuento total de paquetes, la tasa de bytes transferidos o el comportamiento de los *flags* TCP) nos proporciona características fundamentales sobre el comportamiento de la red.

Son precisamente estas métricas las que ayudarían a un modelo a identificar anomalías, clasificar el tipo de tráfico (por ejemplo, diferenciar un flujo de streaming de una descarga directa) o detectar ciberataques (como escaneos de puertos o denegaciones de servicio).