RETO 7

TÚNELES Y REDES PRIVADAS VIRTUALES

Pablo Ramiro Foronda

Marianela Estévez Bosso

Ignacio García García

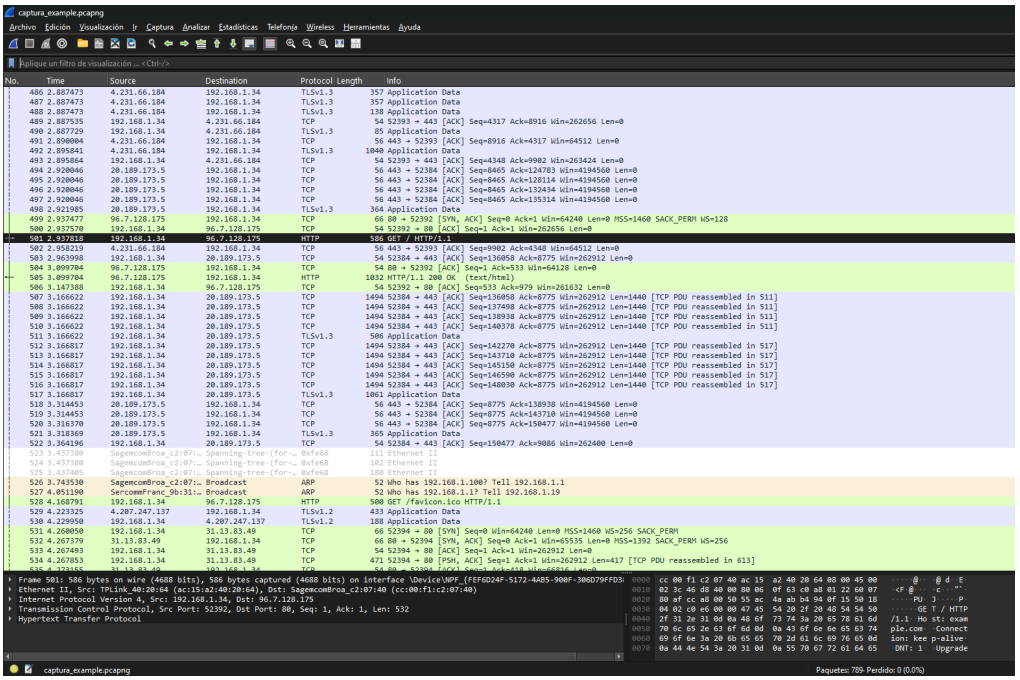
Guillermo Rojo Martín

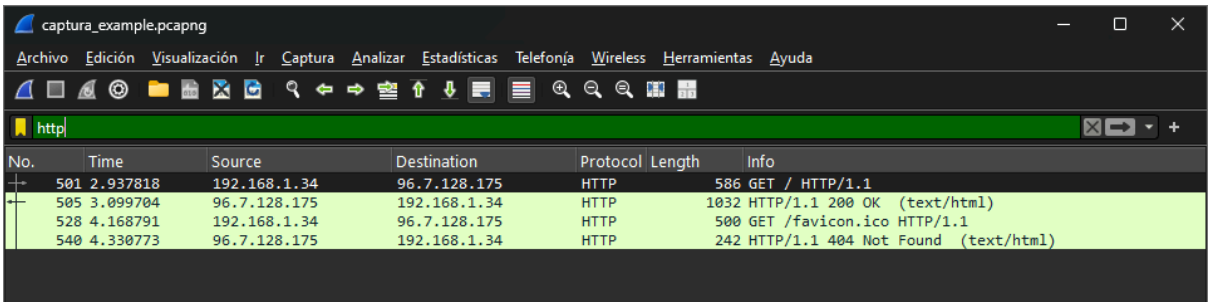
Realización de capturas de tráfico:

1. **Primer escenario: Navegación web tradicional**

   - HTTP. Ejemplo: http://example.com/

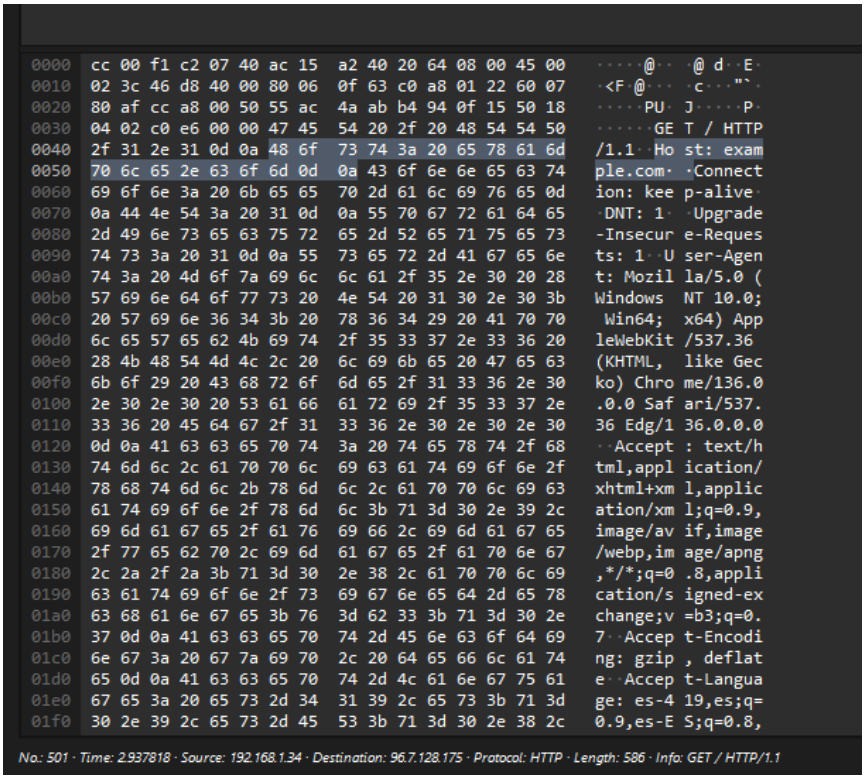
   La dirección IP de nuestro pc es 192.168.1.34 y la de la página es 96.7.128.198. Es una IP diferente pero muy cercana en el rango a la que capturó Wireshark (96.7.128.175). Esto es normal para sitios que pueden usar múltiples IPs o balanceadores de carga. Lo importante es a qué IP se conectó mi navegador en el momento de la captura, y Wireshark nos lo dice.





- Paquete 501: Mi petición GET

  - Este es nuestro navegador pidiendo la página principal de example.com. Si seleccionamos este paquete en el panel de detalles muestra claramente el host.

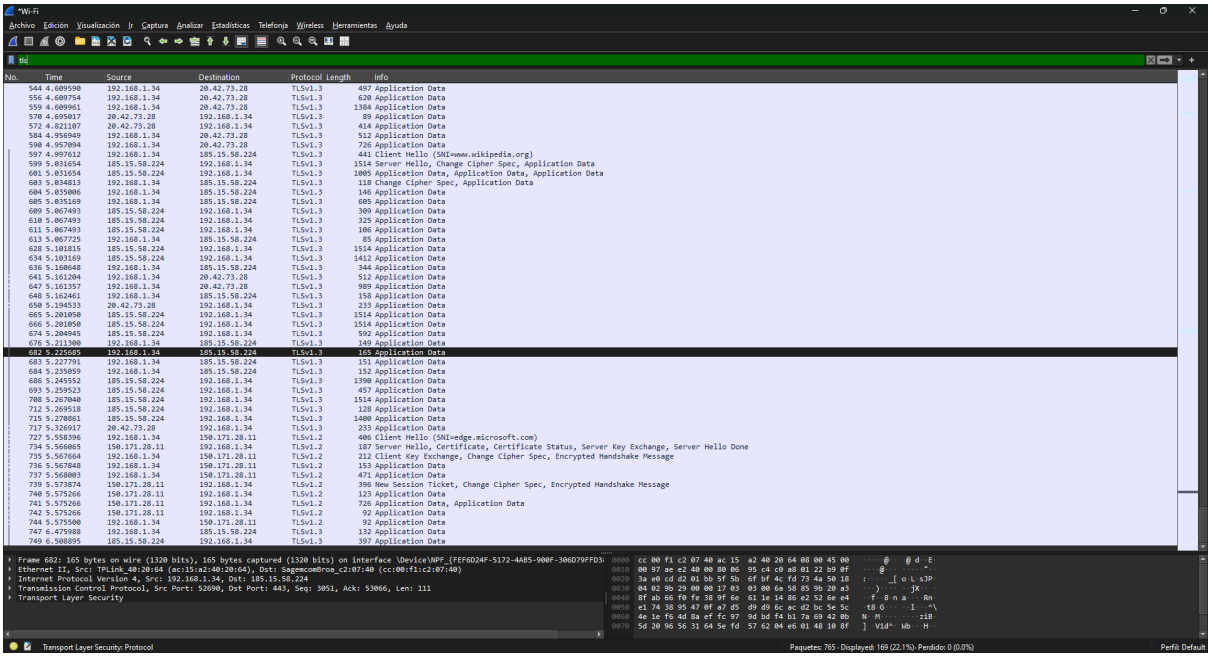


- Paquete 505: Respuesta del Servidor 200 OK.
    - El servidor nos responde que todo está bien y nos envía la página, que es de tipo text/html.
- Paquete 528: Petición GET para favicon.ico
    - Después de cargar la página principal, los navegadores casi siempre intentan pedir un archivo llamado favicon.ico. Este es el pequeño icono que aparece en la pestaña del navegador. Es una petición HTTP normal.
- Paquete 540 (Respuesta del Servidor 404 Not Found para favicon.ico):
    - El servidor de example.com responde que no tiene un archivo llamado favicon.ico (404 Not Found). Esto es muy común para páginas simples.

El análisis de la captura para http://example.com/ demuestra claramente cómo las peticiones y respuestas HTTP viajan en texto plano. Se pudo identificar el host solicitado (example.com) a través de la cabecera Host en la petición GET, y tanto la solicitud como el contenido de la respuesta son visibles para cualquiera que capture el tráfico.

- HTTPS: Navego a un sitio HTTPS (WEB SEGURA)

La conversación entre nuestro ordenador y el servidor web estará cifrada, como si la "postal" fuera dentro de un sobre cerrado con un código secreto. Wireshark seguirá viendo los sobres, pero no podrá leer fácilmente lo que hay dentro después de que se establezca la seguridad. Voy a usar la pagina HTTPs de prueba: https://www.wikipedia.org

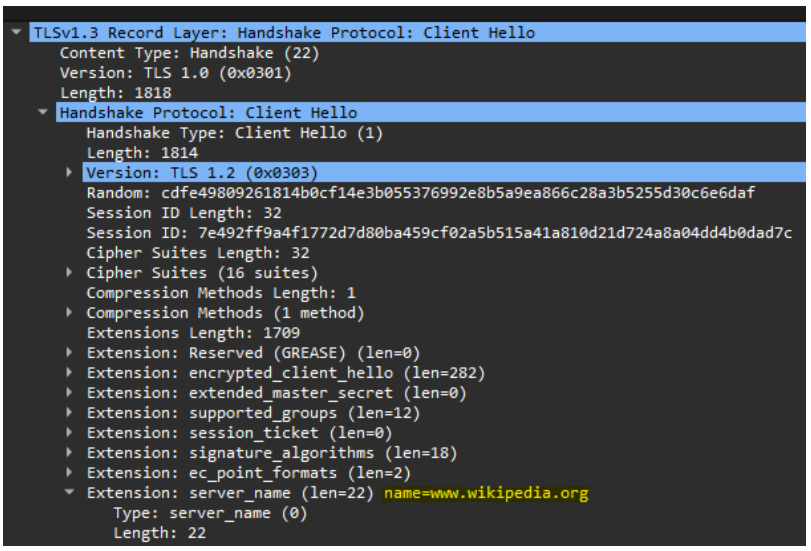


- Muchos paquetes Application Data: Filas 544-598...

    - El navegador y el servidor de Wikipedia están intercambiando la información de la página. Todo este contenido está cifrado, no se puede leer directamente en Wireshark. Estos paquetes demuestran que la comunicación principal está protegida

- El Handshake TLS

    - Paquete 597. Este es el Client Hello que envía nuestro navegador. La parte (SNI=www.wikipedia.org) es crucial. Antes de que se establezca el cifrado completo, el navegador le dice al servidor que quiere establecer una conexión segura con la página. Esto es visible en texto plano dentro de este paquete inicial del handshake. Este es el punto donde se puede identificar el sitio web al que intentas acceder, incluso antes de que la comunicación esté totalmente cifrada.



- Paquete 599

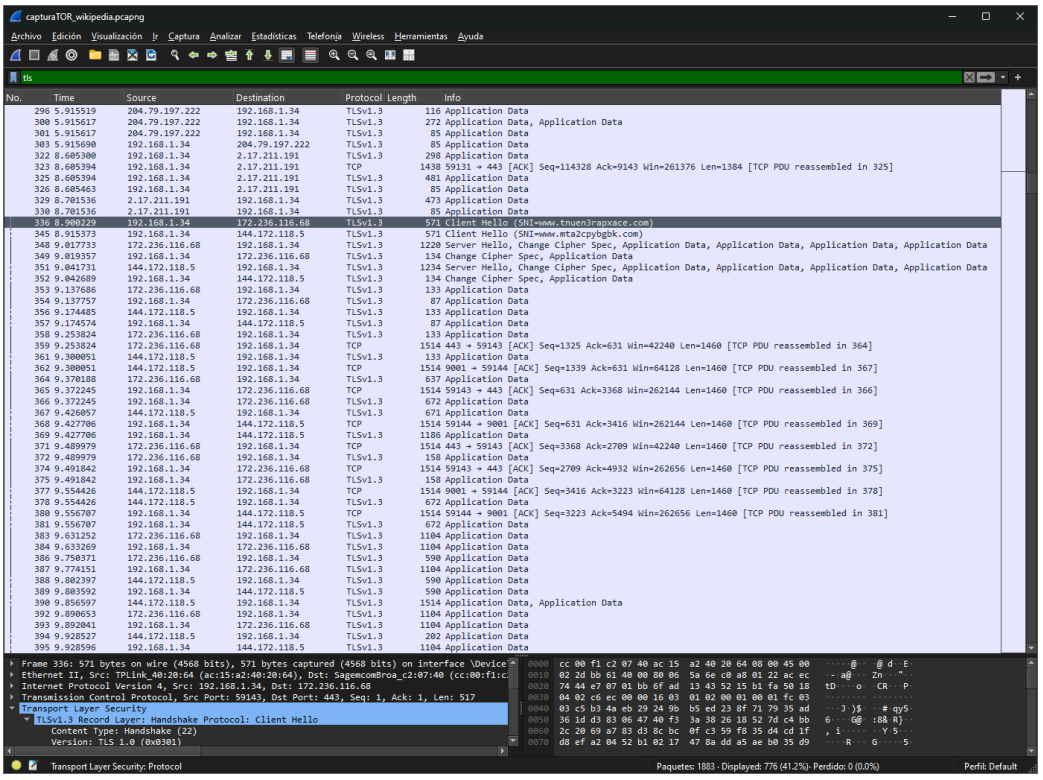
    - El servidor responde al Client Hello. El Server Hello contiene los parámetros de cifrado elegidos por el servidor. Change Cipher Spec indica que el servidor va a empezar a usar cifrado. El Application Data aquí probablemente contiene el resto de los mensajes del handshake del servidor, ya cifrados con las claves temporales del handshake.

- Paquete 601:

    - Nuestro navegador también dice "Ok, yo también empiezo a cifrar (Change Cipher Spec)" y el Application Data es el final del handshake desde nuestro lado, ya cifrado.

2. Segundo Escenario: Navegación Web en TOR hacia Clear Web

Abrimos TOR Browser. Esperamos a que nos indique que está conectado a la red TOR. Luego, en la barra de direcciones, escribimos la dirección de un sitio web normal (de la "clear web"), por ejemplo, https://www.wikipedia.org.



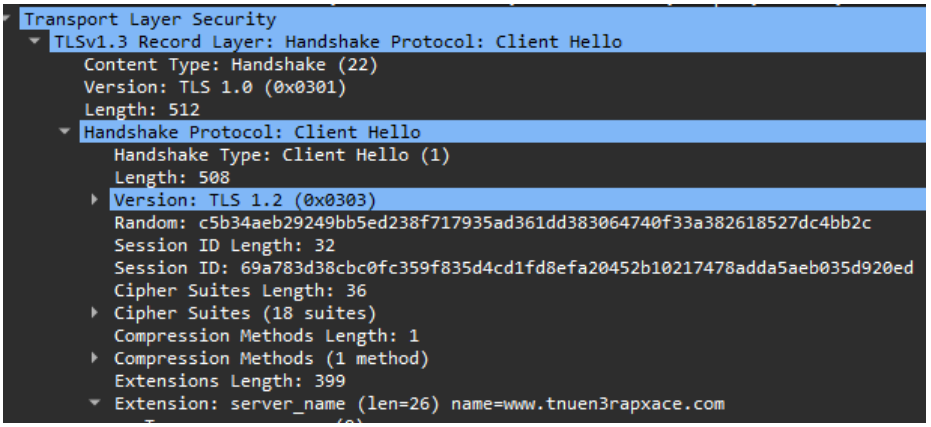
No vemos peticiones DNS directas desde mi IP para wikipedia, ni conexiones TCP/TLS directas desde mi IP a las direcciones IP conocidas de wikipedia (como las que vimos en el escenario HTTPS directo).

Lo que si vemos ahora:

- Conexión inicial a la Red TOR

    - Al principio de la captura, poco después de abrir TOR, vemos tráfico desde mi IP a varias IP desconocidas. El protocolo para estas conexiones es TLSv1.2 o TLSv1.3. Los puertos de destino en estos

servidores remotos suelen ser el 443 o el 9001 (puertos comunes para los nodos TOR). Estas IPs son los nodos de entrada (entry relays o guard relays) de la red TOR. TOR Browser se conecta a uno de estos para entrar a la red.



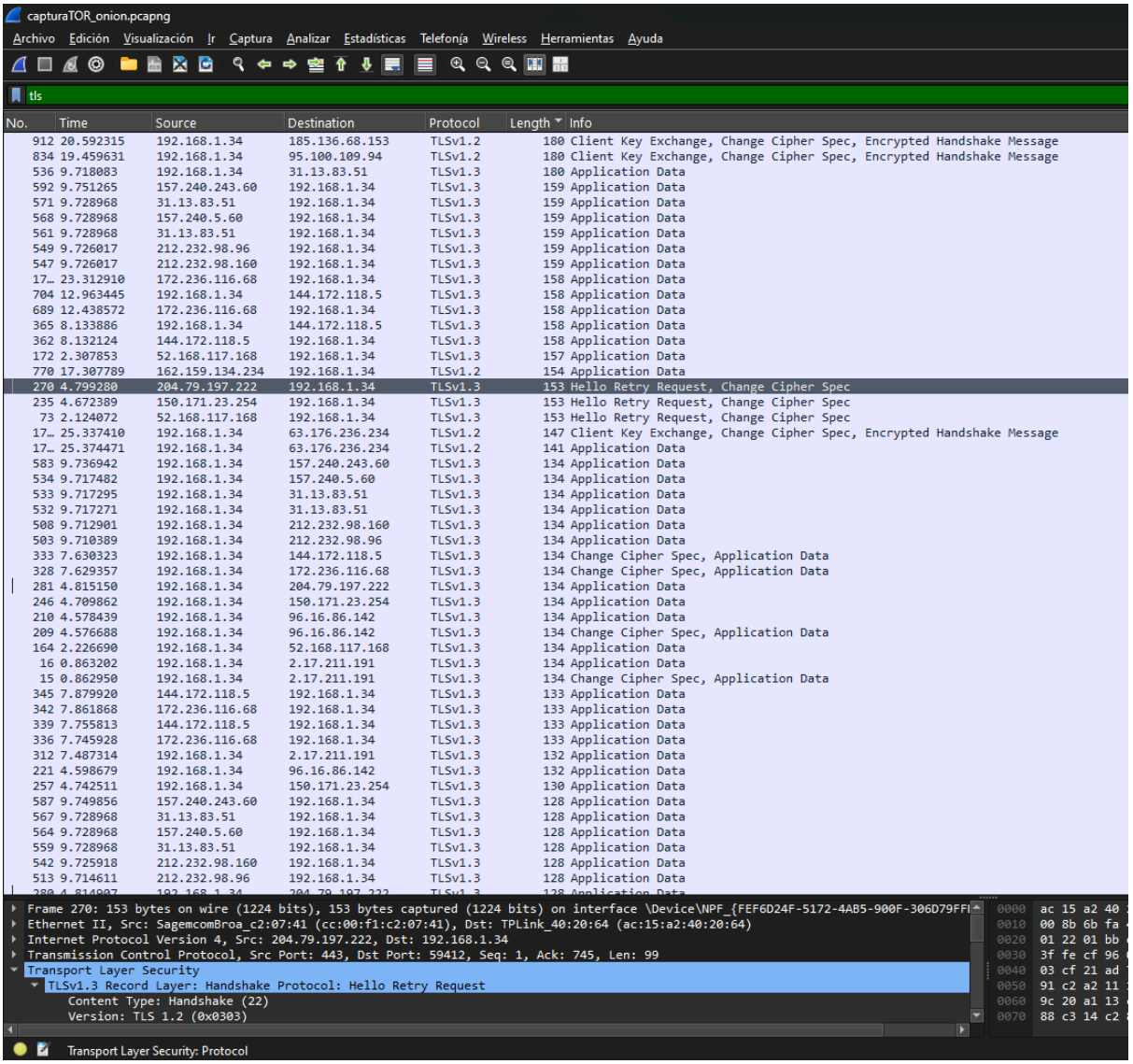
El SNI aquí no es wikipedia.org, es tnuen3rapxace.com. Cuando un TOR Browser establece la conexión cifrada con el primer nodo TOR, el SNI que se usa es para el propio nodo TOR o un dominio genérico que usa TOR para el handshake, no para el sitio web final que vamos a visitar. Wikipedia se solicitará dentro de este túnel cifrado. Toda la navegación hacia wikipedia está ocurriendo dentro de flujos de Application Data cifrados con los nodos TOR. No podemos ver wikipedia directamente.

- Tráfico de Navegación Encapsulado y Cifrado
    - Una vez que TOR Browser está conectado y empezamos a navegar hacia www.wikipedia.org, todo ese tráfico se enviará primero al nodo de entrada TOR al que te conectaste. En Wireshark, seguimos viendo paquetes TLS y luego Application Data dentro de TLS, entre nuestra IP (192.168.1.34) y la IP de ese nodo de entrada TOR.
    - No se ve ninguna comunicación directa con las IPs de Wikipedia. El contenido de nuestra petición a Wikipedia y la respuesta de Wikipedia están cifrados y encapsulados dentro de la comunicación que tenemos con el nodo TOR.

Se demuestra cómo la red TOR anonimiza el tráfico. En lugar de conexiones directas al servidor de destino, se observan conexiones cifradas a nodos de la red TOR. La identidad del sitio web final y el contenido de la comunicación están protegidos de la observación local, cumpliendo el objetivo de privacidad de TOR.

3. Tercer Escenario: Navegación Web en TOR hacia Dominios .onion

Ahora visitamos una dirección de un sitio .onion. El ejemploque usamos es el de DuckDuckGo: https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/.

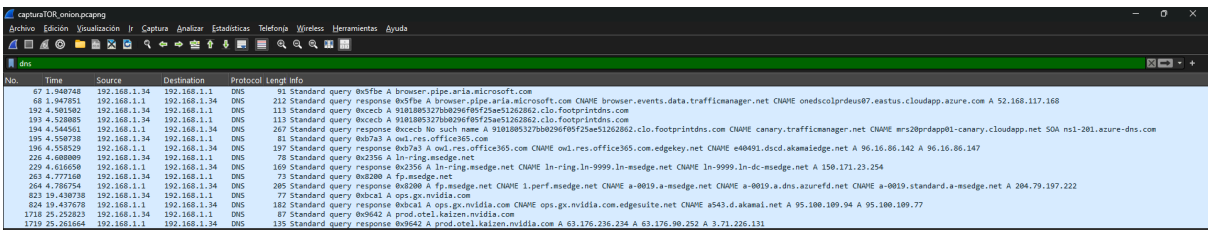


Es similar al escenario anterior hacia clear web en cuanto a la conexión iniciar a la red TOR, pero hay algunas diferencias sutiles.

No vemos:

- Peticiones DNS tradicionales: Los dominios .onion no se resuelven mediante el sistema DNS público. Su resolución ocurre dentro de la red TOR utilizando un sistema de directorios distribuidos(DIstributed Hash Table - DHT) y servicios de introducción

- Conexiones directas a una IP pública del servicio .onion.



Hay varias consultas DNS con nombres de dominio variados de clear web, pero hay ausencia de consulta para el dominio .onion. Esto confirma que no se resuelven a través del sistema DNS público tradicional. Se gestiona internamente por la red TOR

Lo que seguimos viendo similar al escenario anterior son las conexiones iniciales a la red TOR y el tráfico de navegación encapsulado y cifrado hacia nodos TOR.