

DOCUMENTACIÓN TÉCNICA

WIRESHARK_SAD_ACTIVIDAD2_UD1

Thomas Van Vliet Aupetit

19/09/2024

Contenido

Introducción	4
Conceptos básicos	4
HTTP	4
HTTPS:	5
SSL:	5
¿Cómo funciona este método?	6
1. (Handshake)	6
2. Autenticación del servidor	6
3. Intercambio de claves	6
4. Cifrado de la comunicación	6
5. Cierre de la conexión	6
Wireshark:	7
Welcome to wireshark	8
Columnas de Captura	9
• No:	9
• Time:	9
• Source:	9
• Destination:	9
• Protocol	9
• Length:	9
• Info	9
Packet Details (Detalles del paquete)	9
• Frame 14: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits):	9
• Ethernet II, Src: AskeyCompute_b4:52:23, Dst: Intel_le:77:51	10
• Internet Protocol Version 4 (IPv4), Src: 80.58.61.250, Dst: 192.168.1.35:..	10
• User Datagram Protocol (UDP) Src Port: 53, Dst Port: 54168	10
• Domain Name System (DNS):	10
Empezamos demostración	10
Explicación Navegadores http	11
Iniciamos Sesión	11

Capturamos tráfico con wireshark.....	12
Usuario y contraseña Encontrado.....	13
Wireshark VPN funcionamiento.....	14
¿Por qué no se ve nada cuando filtras por hhttp?	16
Conclusión	16
Bibliografía:	17

Introducción

En esta documentación técnica vamos a explorar el uso de Wireshark, para así analizar el tráfico de una página web http sin cifrado SSL. Las páginas web que no utilizan protocolos de cifrado, como el SSL/TLS, transmiten la información en texto plano, esto expone los datos sensibles a posibles interceptaciones. Una de las herramientas que permiten estas acciones, es Wireshark, esta herramienta captura y realiza un análisis de tráfico de red, permitiendo visualizar ese tráfico no cifrado que aparece en texto plano, revelando información crítica que puede ser utilizado por terceros, como (contraseñas, usuarios, datos bancarios, etc.)

A lo largo de este documento, explicaremos como capturar y analizar el tráfico generado al navegar por una página web HTTP sin cifrado, mostrando ejemplos prácticos y visuales de solicitudes y respuestas en texto plano. El objetivo de este trabajo es que mediante el conocimiento de cómo se realizan los procedimientos de captura de tráfico red, podamos y puedan aprender a protegerse en la red.

Conceptos básicos

En este apartado, se pretende hacer un breve apartado sobre conceptos básicos, explicando de la forma más correcta posible los siguientes conceptos:

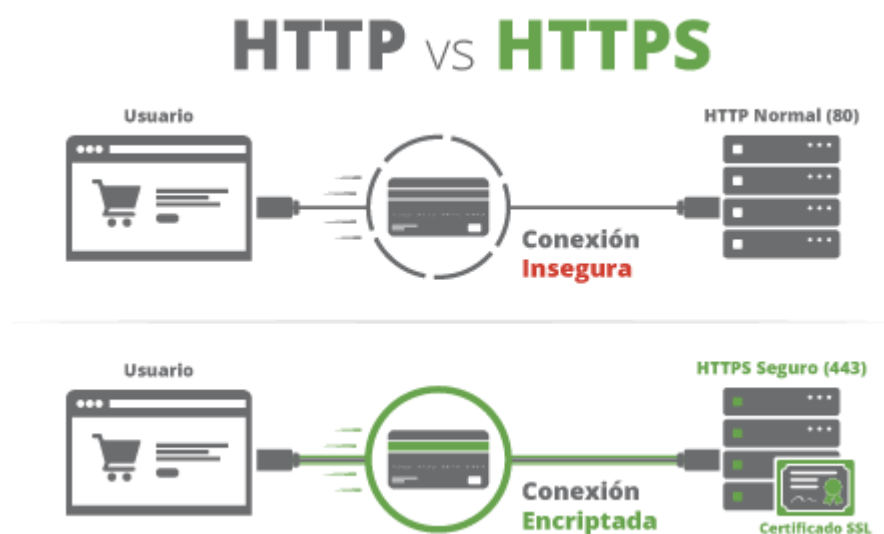
HTTP: De sus siglas, (“Hypertext Transfer Protocol”) se ubica en la capa de aplicación del modelo OSI, siguiendo el protocolo de cliente – servidor, el cliente envía una petición al servidor, el servidor la recibe y seguido le manda la respuesta al cliente. Este Protocolo, hoy en día en cuanto a tema de seguridad se refiere está obsoleto.

¿Por qué? El problema es que cuando el cliente envía la petición al servidor y este la respuesta, ese tráfico de red, aparece totalmente en texto plano (sin cifrado), esto supone un gran peligro para la mayoría de los usuarios medios, que no poseen de conocimientos previos y pueden ser interceptados por terceros con el objetivo de robo de sus datos.

Para que esto no pasará o los usuarios tuvieran más privacidad y por tanto, mas seguridad o viceversa, se creó el (“**HTTPS**”)

HTTPS: Secure Hypertext Transfer Protocol, esto es básicamente el protocolo anterior con más medidas de seguridad, este protocolo que hoy en día usamos globalmente cifra todos los datos que están siendo transferidos en internet entre ordenadores y servidores, convirtiendo los datos totalmente ilegibles e imposibles de leer, mediante el uso de algoritmos de cifrado los cuales se ocupan de codificar los datos.

Por ejemplo, si tu como usuario entras a una pagina web https y esa página web requiere información personal, como contraseñas, datos bancarios etc. La persona que trate de interceptar ese tráfico se va a encontrar con esos datos totalmente encriptados y no va a poder hacer uso de ellos de ningún modo.



Por último, como concepto básico final, vamos a explicar ¿Qué es el SSL? y cómo funciona.

SSL: (“Secure sockets Layer”) este protocolo, es un protocolo criptográfico que esta diseñado para proporcionar una capa más de seguridad, en la comunicación a través de internet. Hoy en día ssl ya tiene un sucesor que es el **TLS** (“Transport layer Security”) una versión más moderna y segura. Aun así, vamos a explicar el ssl ya que forma una parte fundamental para entender cómo funcionan las conexiones seguras en la web.

Básicamente SSL usa un método en base a certificados digitales y cifrado para garantizar que la comunicación entre un cliente y un servidor web sea lo más seguro y privado posible.

¿Cómo funciona este método?

1. (Handshake)

El cliente cuando intenta conectarse a un servidor seguro como https, además de conectarse a una red con el tráfico cifrado, solicita una conexión segura utilizando SSL, entonces el servidor como respuesta le responde con su respectivo certificado SSL que contiene su clave pública y más detalles de identificación.

2. Autenticación del servidor

Una vez recibe el cliente el certificado, lo verifica y comprueba que ese certificado esta aprobado por una autoridad oficial y sin expirar, el cliente lo valida y continua la conexión, y si resulta que lo verifica y no es válido, el usuario recibe una advertencia.

3. Intercambio de claves

Tras la validación, el cliente y el servidor utilizan el denominado (“cifrado asimétrico”) donde establecen una clave de sesión que utilizaran para cifrar el resto de la comunicación. Esta clave de sesión es simétrica, por tanto, tanto como el cliente como el servidor la usan para cifrar y descifrar los datos.

4. Cifrado de la comunicación

A continuación, todos los datos que se intercambien de ahora en adelante entre el cliente y el servidor están totalmente encriptados utilizando la clave de sesión.

5. Cierre de la conexión

Cuando el cliente o el servidor ya no se necesitan más entre sí, se termina la sesión enviando un mensaje de cierre que indica el fin de la conexión segura y la clave quedando inutilizada.

Una vez explicado los conceptos básicos de la mejor forma posible vamos a proceder a documentar el trabajo principal, con la herramienta wireshark, herramienta utilizada en todo tipo de sistemas Linux, Windows, MacOS, como curiosidad si quieres instalar wireshark en un Linux sin GUI puedes descargarte tshark, que es lo mismo, pero en una línea de comandos, (funciona en Ubuntu live server).



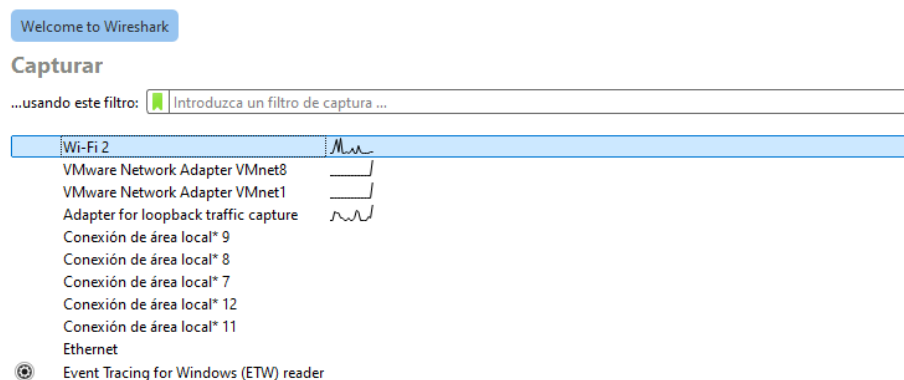
Wireshark: Una vez descargado el ejecutable de wireshark, durante el proceso de instalación nos va a salir una casilla que debemos seleccionar para que wireshark pueda capturar el tráfico de red en Windows.

Se Trata del NPCAP, es un driver de captura de paquetes y una librería de acceso a la red desarrollada para Windows. Actúa como intermediario entre la tarjeta de red de nuestro sistema y aplicaciones como wireshark, (captura los paquetes de red y se los entrega a wireshark para que los vea y analice)

Wireshark = Aplicación que permite analizar el tráfico de red

NPCAP = El driver que permite a wireshark capturar los paquetes desde la red en sistemas Windows.

Tras haber completado con la instalación nos encontramos con esta pantalla



Welcome to wireshark

En la primera pantalla que nos aparece al abrir wireshark se muestran todas las interfaces de red disponibles en el sistema, tanto físicas (wifi- Ethenet) como virtuales (VMware, loopback)

En este caso vamos a hacer doble click en la interfaz Wifi2, esto se debe a que estamos conectados a internet a través de wifi y por tanto, es la que está gestionando todo el tráfico de red que pasa por nuestra conexión inalámbrica y por tanto es la que debemos monitorear.

Tras clicar nos encontramos con:

***Wi-Fi 2**

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
163	14.707657	104.16.103.112	192.168.1.35	TLSv1.2	92	Application Data
164	14.754469	192.168.1.35	104.16.103.112	TCP	54	52383 → 443 [ACK] Seq=43 Ack=39 Win=251 Len=0
165	14.754471	192.168.1.35	104.16.102.112	TCP	54	50035 → 443 [ACK] Seq=85 Ack=77 Win=510 Len=0
166	15.156032	192.168.1.35	3.67.140.45	TCP	54	52690 → 443 [FIN, ACK] Seq=2 Ack=1 Win=516 Len=0
167	15.156234	192.168.1.35	172.64.155.209	TCP	54	52607 → 443 [FIN, ACK] Seq=2 Ack=1 Win=514 Len=0
168	15.156326	192.168.1.35	172.64.155.209	TCP	54	52608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
169	15.171670	172.64.155.209	192.168.1.35	TCP	60	443 → 52607 [FIN, ACK] Seq=1 Ack=3 Win=53 Len=0
170	15.171718	192.168.1.35	172.64.155.209	TCP	54	52607 → 443 [ACK] Seq=3 Ack=2 Win=514 Len=0
171	15.173654	172.64.155.209	192.168.1.35	TCP	60	443 → 52608 [FIN, ACK] Seq=1 Ack=2 Win=13 Len=0
172	15.173693	192.168.1.35	172.64.155.209	TCP	54	52608 → 443 [ACK] Seq=2 Ack=2 Win=511 Len=0
173	15.189305	3.67.140.45	192.168.1.35	TLSv1.2	85	Encrypted Alert
174	15.189305	3.67.140.45	192.168.1.35	TCP	60	443 → 52690 [FIN, ACK] Seq=32 Ack=3 Win=140 Len=0
175	15.189364	192.168.1.35	3.67.140.45	TCP	54	52690 → 443 [RST, ACK] Seq=3 Ack=32 Win=0 Len=0
176	17.615606	192.168.1.35	142.250.184.174	QUIC	1288	Protected Payload (KP0), DCID=ffff3c7ee02c9c637
177	17.615712	192.168.1.35	142.250.184.174	QUIC	106	Protected Payload (KP0), DCID=ffff3c7ee02c9c637
178	17.623214	142.250.184.174	192.168.1.35	QUIC	69	Protected Payload (KP0)
179	17.648993	142.250.184.174	192.168.1.35	QUIC	109	Protected Payload (KP0)
180	17.648993	142.250.184.174	192.168.1.35	QUIC	63	Protected Payload (KP0)
181	17.649126	192.168.1.35	142.250.184.174	QUIC	77	Protected Payload (KP0), DCID=ffff3c7ee02c9c637
182	17.652601	104.16.103.112	192.168.1.35	TLSv1.2	92	Application Data
183	17.652946	192.168.1.35	104.16.103.112	TLSv1.2	96	Application Data
184	17.658589	104.16.103.112	192.168.1.35	TCP	60	443 → 52383 [ACK] Seq=77 Ack=85 Win=11 Len=0
185	17.679201	142.250.184.174	192.168.1.35	QUIC	66	Protected Payload (KP0)
186	17.679287	192.168.1.35	142.250.184.174	QUIC	73	Protected Payload (KP0), DCID=ffff3c7ee02c9c637
187	18.099692	192.168.1.35	52.111.231.21	TLSv1.2	82	Application Data

```
> Frame 1: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{0171CDA-E000-42D3-A6E5-8CFF7DD0A94D}, 1 Ethernet II, Src: Intel_E:77:51 (a0:b3:39:1e:77:51), Dst: AskeyCompute_b4:52:23 (78:29:ed:b4:52:23)
> Internet Protocol Version 4, Src: 192.168.1.35, Dst: 3.67.140.45
> Transmission Control Protocol, Src Port: 52690, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

```
0000  78 29 ed b4 52 23 a0 b3 39 1e 77 51 08 00 45 00 x) : R... 9 wQ : E
0010  00 29 74 f7 40 00 80 06 00 00 c0 a8 01 23 03 43 t@g@... ..# C
0020  8c 2d cd d2 01 bb 38 cd cf 3d 01 5d fe 01 50 10 .....8 . =.] P
0030  02 04 51 57 00 00 00 00 ..Qi...
```

Filtro de visualización

Nada más entrar, podemos observar en esta captura varios detalles sobre el tráfico capturado en nuestra interfaz wifi2, vamos a explicar la interfaz de wireshark y sus funciones, y luego nos centraremos en el objetivo principal de este trabajo que es capturar el tráfico http de la página en cuestión.

No.	Time	Source	Destination	Protocol	Length	Info
-----	------	--------	-------------	----------	--------	------

Lo primero que vemos, nada más ejecutar wireshark es el filtro de visualización, es como una barra e búsqueda donde podemos filtrar lo que queramos en base a lo que

queramos encontrar, es decir, su función es ayudar al usuario a enfocar tu análisis en los paquetes que realmente te interesan, eliminando el ruido de otros protocolos o paquetes que en el momento te parezcan irrelevantes. Aquí mismo, podemos especificar cosas como (“http, DNS, TCP, UDP, filtrar por direcciones IP, filtrar por puerto, por contenido etc)

Columnas de Captura

Justo debajo de la “Barra de búsqueda” vemos las columnas de captura donde nos va a salir todo el tráfico de red. Las desglosaremos a continuación:

- **No:** Es el número de la trama o paquete capturado.
- **Time:** El tiempo transcurrido desde el inicio de la captura hasta que se captura el paquete.
- **Source:** La dirección IP de la máquina que envió el paquete.
- **Destination:** La Dirección de la máquina que recibe el paquete.
- **Protocol:** El protocolo utilizado para este paquete.
- **Lenght:** El tamaño del paquete en bytes.
- **Info:** Información adicional sobre el paquete, como el número de secuencia.

Packet Details (Detalles del paquete): En la parte inferior izquierda de la interfaz gráfica de wireshark podemos observar 5 líneas, que son las denominadas (“Packet Details”) Esta sección muestra gran cantidad de información detallada sobre cada capa del modelo OSI en el paquete seleccionado, desglosando el contenido y los protocolos necesarios.

```
> Frame 14: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface \Device\NPF_{01717CDA-E000-42D3-A6E5-8CFF7DD0494}
> Ethernet II, Src: AskeyCompute_b4:52:23 (78:29:ed:b4:52:23), Dst: Intel_1e:77:51 (a0:b3:39:1e:77:51)
> Internet Protocol Version 4, Src: 80.58.61.250, Dst: 192.168.1.35
> User Datagram Protocol, Src Port: 53, Dst Port: 54168
> Domain Name System (response)
```

- **Frame 14: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits):** Este es el marco o paquete completo capturado, este detalle de paquete te permite saber el tamaño total del paquete que ha viajado por la red (126 bytes) es una visión general del tamaño y el número del paquete capturado.

- **Ethernet II, Src: AskeyCompute_b4:52:23, Dst: Intel_le:77:51:** Esta línea muestra la información de la capa Ethernet, su propósito es indicar las direcciones MAC del dispositivo que envió el paquete, (AskeyCompute, probablemente mi router) y el que lo recibe (dst, mi ordenador con una tarjeta de red Intel).
- **Internet Protocol Version 4 (IPv4), Src: 80.58.61.250, Dst: 192.168.1.35:** Esta es la información de la capa de red (ip) mostrando las direcciones Ip del dispositivo que envió los datos, de un servidor a mi ordenador (SRC a DST)
- **User Datagram Protocol (UDP) Src Port: 53, Dst Port: 54168:** Esto es básicamente la información del protocolo UDP, es un protocolo que se usa para enviar datos rápidamente sin preocuparse de que lleguen correctamente a diferencia del TCP, muestra el puerto 53 que generalmente se usa para DNS y el puerto de destino 54168.
- **Domain Name System (DNS):** Lo que se muestra aquí es que mi ordenador ha pedido la ip de un sitio web o servicio y este paquete contiene la respuesta con esa Ip.

En conclusión, este conjunto de líneas nos muestra un paquete de respuesta DNS que proviene de internet hacia mi ordenador. La red Ethernet lo movió dentro de mi red local, usando ip para identificar los dispositivos, y el protocolo UDP se ha usado para transportar los datos de DNS.

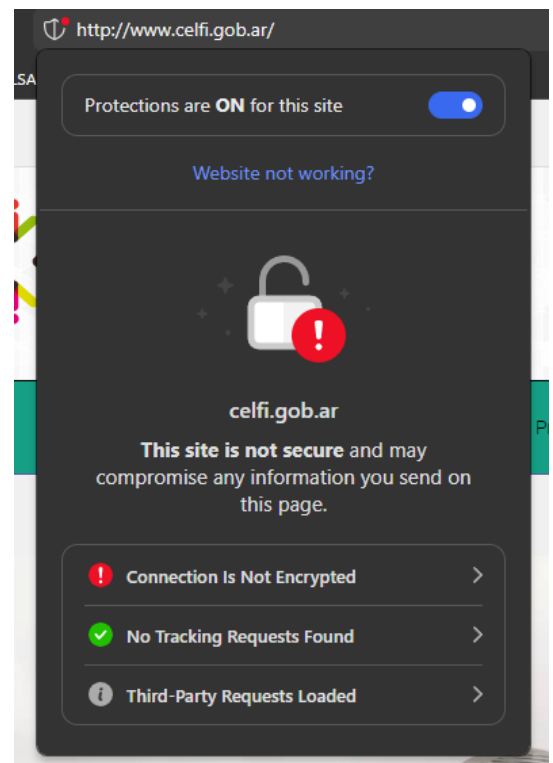
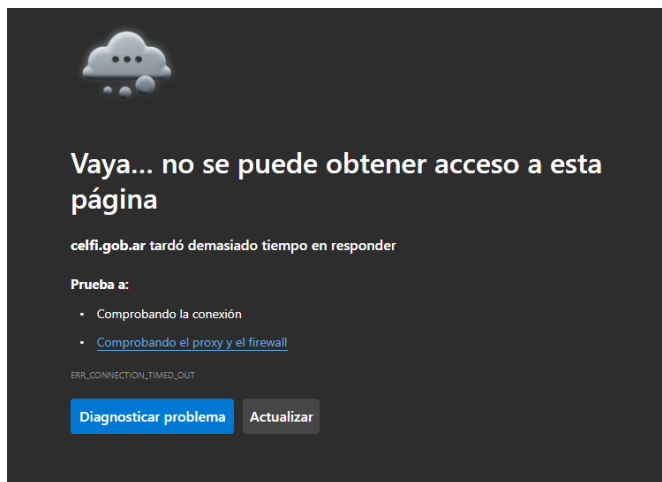
Una vez explicado un ejemplo bien completo de cómo funciona wireshark y su tráfico, desde mi vivienda, vamos a proceder a capturar el tráfico en una página http y realizar algunos filtros.

Empezamos demostración

El nombre de la página a la que debemos acceder es: (<http://celfi.gob.ar/>) como bien vemos esta página no tiene la conexión cifrada.

Explicación Navegadores http

Algunos navegadores web, por defecto tienen prohibido el acceso a estas paginas http, si quieres tener acceso, hazlo o bien desde la configuración, permitiendo el uso de paginas http y bajando la seguridad web de estricta a mínimo, o en este caso engañando al navegador escribiendo “celgi gob ar” ya que si apretamos en mi caso con el navegador duckduckgo directamente al link nos saldría lo siguiente:

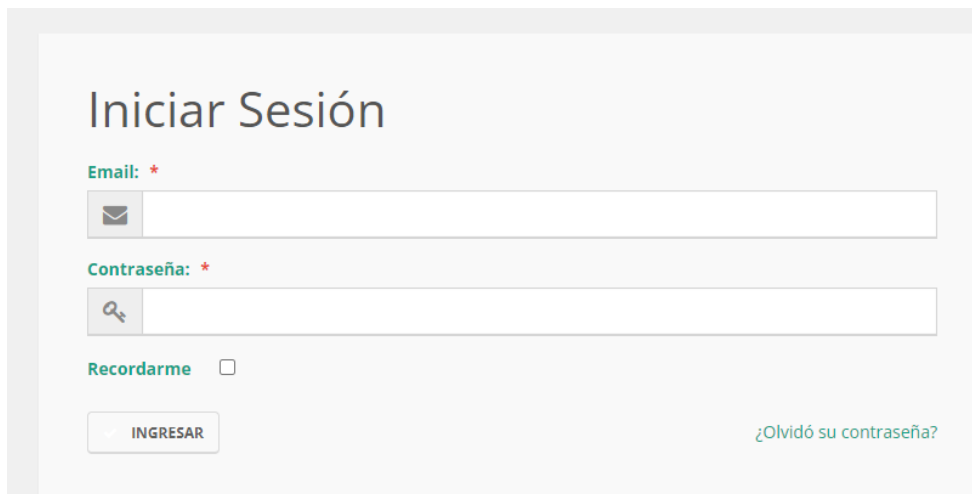


Iniciamos Sesión

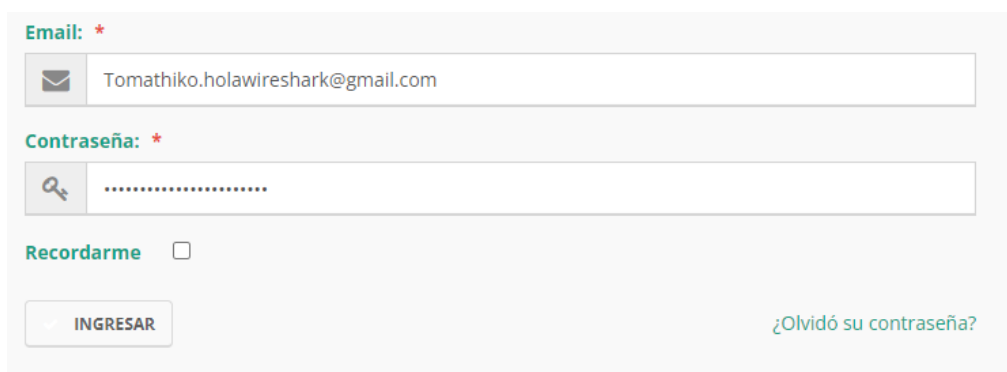
Una vez dentro de la pagina del centro latinoamericano de formación interdisciplinaria, curiosamente pagina perteneciente al Ministerio de Ciencia Tecnología e innovación de Argentina, observamos en la parte superior derecha el logotipo de iniciar sesión.



Hacemos click y se nos abre la siguiente pantalla:



Como hemos comentado anteriormente, esta pagina es http por tanto cualquier usuario que ponga su email y contraseña se va a poder ver en texto plano. Rellenamos los datos.



Capturamos tráfico con wireshark

http						
No.	Time	Source	Destination	Protocol	Length	Info
21	5.118636	192.168.1.35	168.83.5.2	HTTP	55	Continuation
31	6.862910	192.168.1.35	168.83.5.2	HTTP	196	GET /favicon.ico HTTP/1.1
35	6.863841	192.168.1.35	168.83.5.2	HTTP	220	GET /celfi/apple-touch-icon.png HTTP/1.1
39	6.867992	192.168.1.35	168.83.5.2	HTTP	1251	POST /login HTTP/1.1 (application/x-www-form-urlencoded)
68	7.100603	168.83.5.2	192.168.1.35	HTTP	233	HTTP/1.1 404 Not Found (text/html)
70	7.101681	192.168.1.35	168.83.5.2	HTTP	226	GET /celfi/apple-touch-icon-72x72.png HTTP/1.1

Como podemos observar al empezar a capturar tráfico he iniciado sesión en la página web, para tener un listado de búsquedas con menos ruido, filtramos por el protocolo http.

Usuario y contraseña Encontrado

39 6.867992 192.168.1.35 168.83.5.2 HTTP 1251 POST /login
HTTP/1.1 (application/x-www-form-urlencoded)

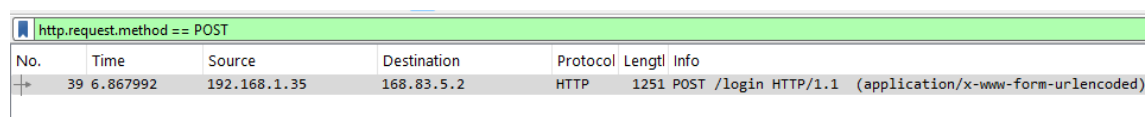
Solo Desglosando esta línea tenemos gran cantidad de información:

El 39 indica que es el paquete numero 39 en la secuencia de todos los paquetes capturados. 6.867... es el tiempo en segundos que ha pasado desde el inicio de la captura hasta que se capturó (6 segundos).

192.168.1.35 es la dirección Ip de origen, es decir, el dispositivo que envió el paquete (yo) y 168.83.5.2 es la ip de destino. http es el protocolo. A continuación, tenemos el tamaño en bytes 1251.

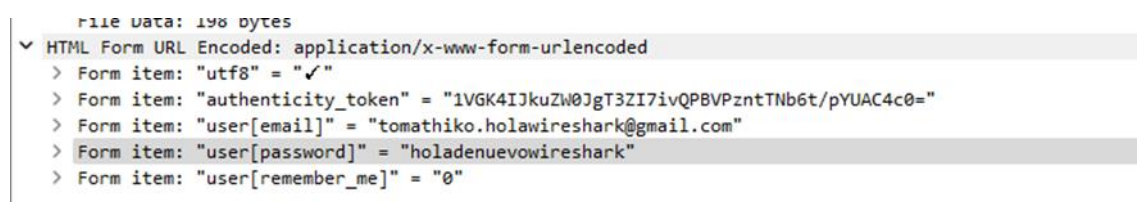
POST /login HTTP/1.1, post es el método http que se esta utilizando para enviar datos, de hecho, en la próxima captura os voy a enseñar como empezar a capturar con el filtro post. Login es el recurso solicitado en el servidor (hemos iniciado sesión) y http1 es la versión del protocolo que estamos usando.

Finalmente (application/x-www-form-urlencoded) es el formato estándar cuando se envían formularios HTML através de POST, significa que los datos del formulario (nombre de usuario, contraseña) están siendo enviados en el cuerpo de la solicitud y se verán en formato ("username=usuario&password=clave)



No.	Time	Source	Destination	Protocol	Length	Info
39	6.867992	192.168.1.35	168.83.5.2	HTTP	1251	POST /login HTTP/1.1 (application/x-www-form-urlencoded)

Si en la barra de búsqueda filtramos con este método vamos a ver todas las solicitudes con Post de manera muy sencilla sin tener que estar gastando mucho tiempo buscando lo que queremos.



```
File data: 198 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "utf8" = "✓"
  > Form item: "authenticity_token" = "1VGK4IJkuZW0JgT3ZI7ivQPBPVzntTNb6t/pYUAC4c0="
  > Form item: "user[email]" = "tomathiko.holawirehawk@gmail.com"
  > Form item: "user[password]" = "holadenuevowireshark"
  > Form item: "user[remember_me]" = "0"
```

En esta imagen vemos en formato HTML nuestro usuario y nuestra contraseña, hasta nuestra disposición del teclado.

```

Transmission Control Protocol, Src Port: 54511, Dst Port: 80, Seq: 2, Ack: 1, Len: 1197
  Source Port: 54511
  Destination Port: 80
  [Stream index: 7]
  [Stream Packet Number: 3]
  > [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 1197]
  Sequence Number: 2      (relative sequence number)
  Sequence Number (raw): 3788457149
  [Next Sequence Number: 1199      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 3442718688
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 516
  [Calculated window size: 516]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0x73e8 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
  TCP payload (1197 bytes)

```

En esta imagen observamos información importante como:

- Src Port: 54511: Es el puerto de origen, que en este caso ha sido asignado por mi sistema operativo de mi dispositivo de forma aleatoria.
- Dst Port 80: Este es el puerto de destino, que en este caso es el 80, es el puerto estándar para http sin cifrar, esto confirma que estamos realizando una conexión http no segura.

Como bien hemos observado, están son las consecuencias de introducirnos en una pagina http sin cifrar, imaginarnos el peligro si un usuario va a una pagina web de compra y venta online, hecha para estafar y un usuario rellena los campos con datos bancarios y contraseñas.

Wireshark VPN funcionamiento

Dejándome llevar por la curiosidad, me he parado a pensar, como sería el funcionamiento de wireshark si me conecto a través de una vpn en una página web http.


Pues lo vais a ver a continuación:

En primer lugar, nos conectamos a una VPN, en mi caso es la vpn del antivirus Norton y al abrir wireshark, capturar los paquetes y haciendo el mismo procedimiento de hacer login en la página de ceelfi nos encontramos con lo siguiente:

No.	Time	Source	Destination	Protocol	Length	Info
25	12.732371	192.168.1.35	18.100.125.213	ESP	170	ESP (SPI=0x938af1bc)
26	12.782402	18.100.125.213	192.168.1.35	ESP	118	ESP (SPI=0xcacad07a)
27	12.886190	18.100.125.213	192.168.1.35	ESP	150	ESP (SPI=0xcacad07a)
28	12.930356	192.168.1.35	18.100.125.213	ESP	118	ESP (SPI=0x938af1bc)
29	12.992353	192.168.1.35	18.100.125.213	ESP	118	ESP (SPI=0x938af1bc)
30	13.036620	18.100.125.213	192.168.1.35	ESP	130	ESP (SPI=0xcacad07a)
31	13.903924	192.168.1.1	192.168.1.35	ICMP	98	Echo (ping) request id=0x4626, seq=0/0, ttl=64 (no response found!)
32	15.802542	192.168.1.35	18.100.125.213	ESP	150	ESP (SPI=0x938af1bc)
33	15.825371	18.100.125.213	192.168.1.35	ESP	250	ESP (SPI=0xcacad07a)
34	18.348008	18.100.125.213	192.168.1.35	UDPENC...	60	NAT-keepalive
35	18.921107	AskeyCompute_b4:52:...	Intel_1e:77:51	ARP	60	Who has 192.168.1.35? Tell 192.168.1.1
36	18.921139	Intel_1e:77:51	AskeyCompute_b4:52:...	ARP	42	192.168.1.35 is at a0:b3:39:1e:77:51
37	19.002119	192.168.1.35	18.100.125.213	UDPENC...	43	NAT-keepalive
38	20.139315	192.168.1.35	18.100.125.213	ESP	150	ESP (SPI=0x938af1bc)
39	20.151548	18.100.125.213	192.168.1.35	ESP	250	ESP (SPI=0xcacad07a)
40	20.310993	fe80::7a29:edff:feb...	ff02::1	ICMPv6	78	Router Advertisement from 78:29:ed:b4:52:23
41	21.030730	192.168.1.35	18.100.125.213	ESP	106	ESP (SPI=0x938af1bc)
42	21.041783	18.100.125.213	192.168.1.35	ESP	134	ESP (SPI=0xcacad07a)
43	21.569308	192.168.1.35	18.100.125.213	ESP	106	ESP (SPI=0x938af1bc)
44	21.581596	18.100.125.213	192.168.1.35	ESP	134	ESP (SPI=0xcacad07a)
45	22.938536	192.168.1.35	18.100.125.213	ESP	146	ESP (SPI=0x938af1bc)
46	23.025086	18.100.125.213	192.168.1.35	ESP	118	ESP (SPI=0xcacad07a)
47	24.597979	192.168.1.1	224.0.0.1	IGMPv2	60	Membership Query, general
48	24.597979	fe80::7a29:edff:feb...	ff02::1	ICMPv6	90	Multicast Listener Query
49	24.610301	18.100.125.213	192.168.1.35	ESP	158	ESP (SPI=0xcacad07a)
50	24.610301	18.100.125.213	192.168.1.35	ESP	118	ESP (SPI=0xcacad07a)

Lo que vemos a continuación, es mayoritariamente paquetes del protocolo ESP (“Encapsulating Security Payload”) esto es un protocolo utilizado por Ipv6, que es una tecnología que usan muchas conexiones VPN para asegurar los datos mediante cifrado. Básicamente cifra el contenido de los paquetes que viajan a través de la VPN. Todos los paquetes etiquetados como ESP están cifrados.

Ahora vamos a proceder a buscar en el visualizador por http:

 http						
No.	Time	Source	Destination	Protocol	Length	Info

Como podéis ver, no sale absolutamente nada.

¿Por qué no se ve nada cuando filtras por http?

El principal motivo por el cual no se ve absolutamente nada al filtrar por http, es por qué todo mi tráfico de red esta cifrado a través de la VPN, todo el tráfico que viaja entre mi dispositivo y el servidor VPN está encapsulado y cifrado. Wireshark únicamente puede ver los paquetes cifrados que viajan entre mi dispositivo y el servidor VPN. Justo por esta razón, el uso de las VPN es muy importantes, cuando quieras navegar en conexiones no seguras, ya sea en lugares públicos o paginas web.

Conclusión

A lo largo de este trabajo, hemos demostrado como utilizar wireshark a la hora de capturar y analizar el tráfico de una página web HTTP no cifrada, revelando vulnerabilidades cruciales relacionadas con este tipo de conexión inseguras. Mediante el uso de ejemplos prácticos hemos podido observar como las credenciales de login y otros datos sensibles son transmitidos en texto plano, lo que puede exponer a muchos usuarios a interceptaciones del trafico de la red. Además, la comparativa del trafico http sin cifrar con el trafico cuando se usa una VPN, demuestra que el uso de tecnologías de cifrado como Ipsec protege de forma efectiva nuestros datos, impidiendo que wireshark pueda mostrar el contenido real de las comunicaciones.

Bibliografía:

Google Workspace. (2023, julio 14). SRE to DevOps: Best practices for cloud-native reliability [Video]. YouTube. <https://youtu.be/hExRDVZHhig>

García Sánchez, Á., González Sotillo, Á., Enamorado Sarmiento, L., & Sanz Rodríguez, J. (2015). Servicios de red e internet (2ª ed.). Ibergaceta Publicaciones S.L.

Wireshark. (n.d.). Wireshark documentation. Wireshark Foundation. <https://www.wireshark.org/docs>

Wireshark. (n.d.). *Wireshark user's guide*. Wireshark Foundation. https://www.wireshark.org/docs/wsug_html_chunked/