



REDES DE COMPUTADORA I

Laboratorio II - Investigación con sniffer Wireshark

Autor:

RAMON INSAURRALDE

Resumen

Desarrollaremos un análisis del comportamiento de la red en diferentes situaciones, elaboraremos un resumen de los protocolos, servicios y puertos al utilizar la red gracias a la herramienta tipo sniffer llamada wireshark, para este trabajo usaremos la red wifi para hacer todas las pruebas.

Índice

1. Introducción	2
2. Marco teórico	2
3. Análisis de los resultados	2
3.1. Login HTTP	3
3.1.1. Explicación	3
3.2. Login HTTPS	5
3.2.1. Explicación	5
3.3. Consulta de un dominio DNS	7
3.3.1. Explicación	7
3.4. Envío de Email	9
3.4.1. Explicación	9
3.5. Descarga de archivos P2P	10
3.6. Intercambio de paquetes para conectarse a una red wifi segura . .	10
3.6.1. Explicación	10
3.7. •	11
4. Bibliografía	12

1. Introducción

Wireshark, antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos de forma únicamente hueca. La funcionalidad que provee es similar a la de tcpdump, pero añade una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo. También incluye una versión basada en texto llamada tshark. Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

2. Marco teórico

En este trabajo realizaremos la siguiente actividad, capturar paquetes de la red cuando se da:

- i. Login http.
- ii. Login https.
- iii. Consulta de un dominio a un DNS.
- iv. Envío de un email.
- v. Descarga de archivos P2P.
- vi. Intercambio de paquetes para conectarse a una red WiFi segura.
- vii. Intercambio de paquetes para conexión DHCP.

Obs: Además de capturar los paquetes se debe analizar el contenido del paquete dando explicaciones sobre cada uno de ellos.

3. Análisis de los resultados

En esta sección analizaremos insitu los comportamientos del capturador de paquetes para los casos solicitados por la investigación, cada Item contará con imágenes que apoyaran el proceso explicativo.

3.1. Login HTTP

No.	Time	Source	Destination	Protocol	Length	Info
2	0.41551	192.168.0.4	64.233.199.188	TCP	55	58275 → 443 [ACK] Seq=1 Ack=1 Win=68 Len=1 [TCP segment of a reassembled PDU]
3	0.50827	64.233.199.188	192.168.0.4	TCP	66	443 → 58275 [ACK] Seq=1 Ack=2 Win=181 Len=0 [TCP segment of a reassembled PDU]
5	2.95731	192.168.0.4	200.9.4.8	HTTP	795	POST /login/try HTTP/1.1 (application/x-www-form-urlencoded)
7	2.97259	200.9.4.8	192.168.0.4	TCP	68	80 → 59831 [ACK] Seq=1 Ack=742 Win=178 Len=0
8	3.23596	200.9.4.8	192.168.0.4	HTTP	388	HTTP/1.1 302 Found
9	3.23833	192.168.0.4	200.9.4.8	HTTP	589	GET / HTTP/1.1
10	3.25118	200.9.4.8	192.168.0.4	TCP	68	80 → 59831 [ACK] Seq=327 Ack=1277 Win=181 Len=0
11	3.28123	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=327 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
12	3.38191	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=1787 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
13	3.38204	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=3887 Win=68 Len=0
14	3.38264	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=3887 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
15	3.38360	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=4667 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
16	3.38352	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=5847 Win=68 Len=0
17	3.38373	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=5847 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
18	3.38419	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=1227 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
19	3.38427	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=6087 Win=68 Len=0
20	3.38535	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=6087 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
21	3.38536	200.9.4.8	192.168.0.4	TCP	1482	80 → 59831 [ACK] Seq=9987 Ack=1277 Win=181 Len=1348 [TCP segment of a reassembled PDU]
22	3.38539	200.9.4.8	192.168.0.4	HTTP	60	HTTP/1.1 200 OK (text/html)
23	3.38547	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=11348 Win=68 Len=0

> Frame 5: 795 bytes on wire (6360 bits), 795 bytes captured (6360 bits) on interface 0

> Ethernet II, Src: IntelCor_8b:d5:47 (34:e6:ad:8b:d5:47), Dst: ArrisRo_00:00:03 (00:00:ca:00:00:03)

> Internet Protocol Version 4, Src: 192.168.0.4, Dst: 200.9.4.8

> Transmission Control Protocol, Src Port: 59831, Dst Port: 80, Seq: 1, Ack: 1, Len: 741

Source Port: 59831

Destination Port: 80

[Stream index: 1]

[TCP Segment Len: 741]

Sequence number: 1 (relative sequence number)

[Next sequence number: 742 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

0101 = Header Length: 20 bytes (5)

> Flags: 0x018 (PSH, ACK)

Window size value: 68

[Calculated window size: 68]

[Window size scaling factor: -1 (unknown)]

Checksum: 0xe294 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

> [SEQ/ACK analysis]

> [Timestamps]

TCP payload (741 bytes)

> Hypertext Transfer Protocol

HTML Form URL Encoded: application/x-www-form-urlencoded

Figura 1: Login HTTP-WIRESHARK

3.1.1. Explicación

En la figura 1 podemos observar en verde los paquetes que se intercambian entre el servidor de sapienza (ip 200.9.4.8) y la máquina que utilizamos como prueba (ip 192.168.0.4 local). Al realizar el procedimiento de login, se realizan una serie de intercambios de paquetes, la mayoría de ellos de ack e información emitida y recibida por partes dentro de los envíos. Al inicio se envían usuario y password no encriptado (ver figura 2).

5	2.957591	192.168.0.4	200.9.4.8	HTTP	795 POST /login/try HTTP/1.1 (application/x-www-form-urlencoded)
7	2.972070	200.9.4.8	192.168.0.4	TCP	60 80 → 59831 [ACK] Seq=1342 Win=170 Len=0
8	3.235986	200.9.4.8	192.168.0.4	HTTP	300 HTTP/1.1 302 Found
9	3.238330	192.168.0.4	200.9.4.8	HTTP	589 GET / HTTP/1.1
10	3.251106	200.9.4.8	192.168.0.4	TCP	60 80 → 59831 [ACK] Seq=527 Ack=1277 Win=181 Len=0
11	3.301823	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=327 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
12	3.301991	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=1707 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
13	3.302041	192.168.0.4	200.9.4.8	TCP	54 59831 → 80 [ACK] Seq=1277 Ack=3087 Win=68 Len=0
14	3.302044	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=3087 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
15	3.303000	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=4467 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
16	3.303152	192.168.0.4	200.9.4.8	TCP	54 59831 → 80 [ACK] Seq=1277 Ack=5847 Win=68 Len=0
17	3.303763	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=5847 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
18	3.304169	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=7227 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
19	3.304227	192.168.0.4	200.9.4.8	TCP	54 59831 → 80 [ACK] Seq=1277 Ack=8607 Win=68 Len=0
20	3.305365	200.9.4.8	192.168.0.4	TCP	1434 80 → 59831 [ACK] Seq=8607 Ack=1277 Win=181 Len=1300 [TCP segment of a reassembled PDU]
21	3.305366	200.9.4.8	192.168.0.4	TCP	1402 80 → 59831 [ACK] Seq=9987 Ack=1277 Win=181 Len=1348 [TCP segment of a reassembled PDU]
22	3.305369	200.9.4.8	192.168.0.4	HTTP	60 HTTP/1.1 200 OK (text/html)
23	3.305472	192.168.0.4	200.9.4.8	TCP	54 59831 → 80 [ACK] Seq=1277 Ack=11340 Win=68 Len=0

[Full request URI: http://alumnos.sapiencia.uc.edu.py/login/try]	
[HTTP request 1/2]	
[Response in frame 8]	
[Next request in frame 9]	
[File Data: 83 bytes]	
▼ HTML Form URL Encoded: application/x-www-form-urlencoded	
▼ Form item: "Credenciales[cedula]" = "3238774"	
Key: Credenciales[cedula]	
Value: 3238774	
▼ Form item: "Credenciales[contrasena]" = "██████████"	

0100	33 36 0d 0a 41 63 65 70 74 3a 20 74 65 78 74 36	Accept: text
0101	2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f	/html,application
0102	6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c	n/html+xml,application
0103	69 63 61 74 69 6f 6e 2f 78 6d 6c 2b 71 3d 30 2e	ication/wml;q=0.9,image/webp;img
0104	39 2c 69 6d 61 67 65 2f 77 65 62 70 2c 69 6d 61	ge/png; */*;q=0.8
0105	67 65 2f 61 70 6e 67 2c 2a 2f 2a 3b 71 3d 30 2e	Referer: http
0106	38 0d 0a 52 65 66 65 72 65 72 3a 20 61 74 70 78	://alumnos.sapie
0107	3a 2f 2f 61 6c 75 6d 6e 6f 73 2e 73 61 70 69 65	ntia.uc.edu.py/l
0108	6e 74 69 61 2e 75 63 2e 65 64 75 2e 70 79 2f 6c	ogin/-a-cept-in
0109	6f 6f 69 6e 2f 0d 0a 41 63 65 70 74 2d 65 6e	coding: gzip, de
0110	63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65	flate; Accept-La
0111	66 6c 61 74 65 0d 0a 41 63 65 70 74 2d 4c 61	nguage: es-es,en
0112	6e 67 75 61 67 65 3a 20 65 73 2d 45 53 2c 65 73	;q=0.9,en;q=0.8
0113	3b 71 3d 30 2e 39 2c 65 6e 3b 71 3d 30 2e 38 0d	Cookie: Sapienc
0114	0a 43 6f 6f 69 65 3a 20 53 61 70 69 65 6e 74	iaLalumnos.sapien
0115	69 61 41 6c 75 6d 6e 6f 73 3d 70 69 69 71 75 37	u18fdqsk pjdr3pn5
0116	75 69 38 66 64 71 73 6b 70 6a 64 72 33 70 69 35	utd credenc
0117	75 66 74 30 6d 0a 6d 6e 65 72 61 64 65 6e 63 09	ialalumnos.contra
0118	6e 63 65 69 65 6e 65 64 65 64 65 65 65 65 65	se=3238774&credenc
0119	3d 33 32 33 38 37 37 34 26 43 72 65 64 65 6e 63	iales508 contrase
0120	69 61 6c 65 73 25 35 42 63 6f 6e 74 72 61 73 65	naH5Qd4 81516234
0121	6e 68 61 25 35 44 3d 34 38 11 35 31 30 32 33 34	28reins 540
0122	32 45 72 72 65 69 6e 73 25 34 30	

Figura 2: Login HTTP-WIRESHARK-Login y Password

Descripción Paso a Paso (FIGURA 3):

- 1: Solicitud de conexión del puerto 59831 al puerto 80 de Sapienza, Método: POST comando: login/try protocolo: HTTP/1.1
- 2: Confirmación HTTP/1.1 de protocolo, servidor CENTOS, apache 2.2.15, php-version: 5.5.36,
- 3: Protocolo: HTTP Metodo: GET, Url: / en este caso el servidor da la posibilidad de recibir el url al cual conectarse por parte del navegador,
- 4: El navegador envia un ACK aceptando el GET, del puerto 80 al 59831
- 5: Serie de intercambio de paquetes de información, la fuente envia paquetes con dimensiones iguales y el servidor responde con ACK cada dos envios, al principio supucimos que eran errores de envio vista la periodicidad y dimensión, luego llegamos a la conclusión de que Wireshark ha concatenado una serie de segmentos mostrando en el toda la información completa.No se trata de ningún error
- 6: El servidor aceptó la información enviada, el mismo responde con un codigo HTTP/1.1 numero 200, el cual indica que la información enviada corresponde a un usuario, el mismo envia en su sección text data una página web completa, de tipo text/html con la información del alumno en este caso.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.415551	192.168.0.4	64.233.190.108	TCP	55	58275 → 443 [ACK] Seq=1 Ack=1 Win=60 Len=1 [TCP segment of a reassembled PDU]
3	0.588877	64.233.190.108	192.168.0.4	TCP	66	443 → 58275 [ACK] Seq=1 Ack=2 Win=181 Len=0 SLE=1 SRE=2
5	2.957391	192.168.0.4	200.9.4.8	HTTP	795	POST /login/try HTTP/1.1 (application/x-www-form-urlencoded)
7	2.972670	200.9.4.8	192.168.0.4	TCP	60	80 → 59831 [ACK] Seq=1 Ack=742 Win=170 Len=0 1
8	3.235986	200.9.4.8	192.168.0.4	HTTP	308	HTTP/1.1 302 Found 2
9	3.238330	192.168.0.4	200.9.4.8	HTTP	589	GET / HTTP/1.1 3
10	3.251106	200.9.4.8	192.168.0.4	TCP	60	80 → 59831 [ACK] Seq=327 Ack=1277 Win=181 Len=0 4
11	3.301823	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=327 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
12	3.301991	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=1787 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
13	3.302041	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=3887 Win=68 Len=0
14	3.302604	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=3887 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
15	3.303090	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=4467 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
16	3.303152	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=5847 Win=68 Len=0
17	3.303763	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=5847 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
18	3.304169	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=7227 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
19	3.304227	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=6967 Win=68 Len=0
20	3.305355	200.9.4.8	192.168.0.4	TCP	1434	80 → 59831 [ACK] Seq=6967 Ack=1277 Win=181 Len=1380 [TCP segment of a reassembled PDU]
21	3.305366	200.9.4.8	192.168.0.4	TCP	1402	80 → 59831 [ACK] Seq=9987 Ack=1277 Win=181 Len=1348 [TCP segment of a reassembled PDU]
22	3.305369	200.9.4.8	192.168.0.4	HTTP	60	HTTP/1.1 200 OK (text/html) 6
23	3.305472	192.168.0.4	200.9.4.8	TCP	54	59831 → 80 [ACK] Seq=1277 Ack=11340 Win=68 Len=0 7

```

> Frame 5: 795 bytes on wire (6360 bits), 795 bytes captured (6360 bits) on Interface 0
> Ethernet II, Src: IntelCor_30:05:47:(34:60:ad:30:05:47), Dst: Acr150r_00:00:03 (00:00:ca:00:00:03)
> Internet Protocol Version 4, Src: 192.168.0.4, Dst: 200.9.4.8
> Transmission Control Protocol, Src Port: 59831, Dst Port: 80, Seq: 1, Ack: 1, Len: 741
  Source Port: 59831
  Destination Port: 80
  [Stream index: 1]
  [TCP Segment Len: 741]
  Sequence number: 1 (relative sequence number)
  (Next sequence number: 742 (relative sequence number))
  Acknowledgment number: 1 (relative ack number)
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window size value: 68
  [Calculated window size: 68]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0xae294 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > [Seq/ACK analysis]
  > [Timestamps]
  TCP payload (741 bytes)
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded

```

Figura 3: Login HTTP-WIRESHARK

- 7: ACK de recepción por parte de la computadora de prueba, ambos servidores dejan de enviarse por el momento paquetes ya que el protocolo de login finalizó con éxito.

3.2. Login HTTPS

3.2.1. Explicación

En la figura 4 observamos los paquetes intercambiados por la fuente, mi computadora, con ip (192.168.0.4 local) y el servidor web (31.13.94.38), al realizarse un login bajo el protocolo HTTPS (TLSV1.2), podemos observar a priori que el primer procedimiento que intenta realizar la fuente desde el puerto 61350 al puerto 443 del servidor es tratar de enviar la información encriptada al mismo. Podemos observar que la información que anteriormente vimos estaba pasada directamente con el LOGIN HTTP, ahora esta encriptada y se pasa a traves del protocolo TLS v 1.2 (ver figura 4)

Descripción Paso a Paso (FIGURA 4):

- 1: Intento de envío de información encriptada al servidor, app-data-protocol: http-over-tls; Src Port: 61350; Dst Port: 443;
- 2: En este caso el servidor host, envia un paquete SYN al servidor de Destino, el mismo responde con un paquete SYN nuevamente, en este paso inicia lo que se denomina como HandSHake entre la fuente y el Destino.

No.	Time	Source	Destination	Protocol	Length	Info
12	6.205151	192.168.0.4	31.13.94.38	TLSv1.2	652	Application Data
13	6.205319	192.168.0.4	31.13.94.38	TLSv1.2	404	Application Data
14	6.205428	192.168.0.4	31.13.94.38	TLSv1.2	92	Application Data
15	6.227163	192.168.0.4	31.13.94.38	TLSv1.2	214	Application Data
16	6.227447	192.168.0.4	31.13.94.38	TLSv1.2	769	Application Data
17	6.227664	192.168.0.4	31.13.94.38	TLSv1.2	92	Application Data
18	6.227957	192.168.0.4	31.13.94.24	TCP	66	61422 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
19	6.228576	192.168.0.4	31.13.94.36	TCP	66	61423 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
20	6.241320	31.13.94.38	192.168.0.4	TLSv1.2	96	Application Data
21	6.241388	192.168.0.4	31.13.94.38	TCP	54	61350 → 443 [ACK] Seq=1900 Ack=43 Win=1023 Len=0
22	6.242592	31.13.94.38	192.168.0.4	TCP	60	443 → 61350 [ACK] Seq=43 Ack=987 Win=2043 Len=0
23	6.259708	31.13.94.38	192.168.0.4	TLSv1.2	96	Application Data
24	6.259778	192.168.0.4	31.13.94.38	TCP	54	61350 → 443 [ACK] Seq=1900 Ack=85 Win=1023 Len=0
25	6.264515	31.13.94.36	192.168.0.4	TCP	66	443 → 61423 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1410 SACK_PERM=1 WS=256
26	6.264611	192.168.0.4	31.13.94.36	TLSv1.2	453	Client Hello
27	6.265407	31.13.94.24	192.168.0.4	TCP	66	442 → 61422 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1410 SACK_PERM=1 WS=256
28	6.265576	192.168.0.4	31.13.94.24	TCP	54	61422 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
29	6.265828	192.168.0.4	31.13.94.24	TLSv1.2	456	Client Hello
30	6.296772	31.13.94.36	192.168.0.4	TCP	60	443 → 61423 [ACK] Seq=1 Ack=400 Win=29440 Len=0
31	6.297685	31.13.94.36	192.168.0.4	TLSv1.2	168	Server Hello, Change Cipher Spec, Encrypted Handshake Message
32	6.297747	192.168.0.4	31.13.94.36	TCP	54	61423 → 443 [ACK] Seq=400 Ack=115 Win=261888 Len=0
33	6.298222	192.168.0.4	31.13.94.36	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
34	6.299324	192.168.0.4	31.13.94.36	TLSv1.2	141	Application Data
35	6.301931	31.13.94.24	192.168.0.4	TCP	60	443 → 61422 [ACK] Seq=1 Ack=403 Win=29440 Len=0
36	6.303118	31.13.94.38	192.168.0.4	TCP	60	443 → 61350 [ACK] Seq=85 Ack=1900 Win=2043 Len=0
37	6.306509	31.13.94.24	192.168.0.4	TLSv1.2	168	Server Hello, Change Cipher Spec, Encrypted Handshake Message
38	6.306572	192.168.0.4	31.13.94.24	TCP	54	61422 → 443 [ACK] Seq=403 Ack=115 Win=261888 Len=0
39	6.307102	192.168.0.4	31.13.94.24	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message

```

> Frame 17: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
> Ethernet II, Src: IntelCor_Sb:d5:47 (34:e6:ad:5b:d5:47), Dst: ArrisGro_00:00:03 (00:00:ca:00:00:03)
> Internet Protocol Version 4, Src: 192.168.0.4, Dst: 31.13.94.38
> Transmission Control Protocol, Src Port: 61350, Dst Port: 443, Seq: 1862, Ack: 1, Len: 38
✓ Secure Sockets Layer
  ✓ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 33
    Encrypted Application Data: 0000000000000001c1a64f3edd52f179d7f3be10a3c8bae7e3...

```

Figura 4: Login HTTPS-WIRESHARK

Lo importante es señalar el proceso de conocido para que se realice una conexión socket tcp

El host A envía un paquete TCP SYNchronize al Host B

El anfitrión B recibe el SYN de A El anfitrión B envía un reconocimiento SYNchronize

El anfitrión A recibe el SYN-ACK de B

El anfitrión A envía ACKnowledge

El anfitrión B recibe ACK.

La conexión de socket TCP está ESTABLISHED.

- 3,4 y 5: En estos pasos se explica como funciona el Protocolo de TLS Handshake el cual implica los siguientes pasos:

1- El cliente envía un mensaje (Client hello) al servidor, junto con el valor aleatorio del cliente y las suites de cifrado(cipher suites) admitidas.

2- El servidor responde enviando un mensaje (Server Hello) al cliente, junto con el valor aleatorio del servidor.

3- El servidor envía su certificado al cliente para la autenticación y puede solicitar un certificado del cliente. El servidor envía el mensaje (Servidor hello done).

4- Si el servidor ha solicitado un certificado del cliente, el cliente lo envía.

5- El cliente crea un Pre-Master Secret aleatorio y lo encripta con la clave pública del certificado del servidor, enviando el Pre-Master Secret encriptado al servidor.

6- El servidor recibe el secreto Pre-Master. El servidor y el cliente generan el secreto principal y las claves de sesión basadas en el secreto premaster.

- 7- El cliente envía la notificación (Cambiar especificación de cifrado) al servidor para indicar que el cliente comenzará a utilizar las nuevas claves de sesión para el hash y el cifrado de mensajes. El cliente también envía un mensaje de (Cliente terminado).
- 8- El servidor recibe (Cambiar especificación de cifrado) y cambia su estado de seguridad de la capa de registro a cifrado simétrico utilizando las claves de sesión. El servidor envía el mensaje (Servidor terminado) al cliente.
- 9- El cliente y el servidor ahora pueden intercambiar datos de aplicaciones a través del canal seguro que han establecido. Todos los mensajes enviados de cliente a servidor y de servidor a cliente se cifran mediante la clave de sesión.

3.3. Consulta de un dominio DNS

Para realizar este procedimiento, debemos hacer ping a un dominio cualquiera, como por ejemplo google.com y ver los paquetes asociados al momento de realizar la consulta al dominio.

3.3.1. Explicación

La figura 5 se muestra los paquetes filtrados para la dirección ip de origen, la consulta en cuestión corresponde a un PING al dominio www.google.com

- 1: Consulta desde la ip de origen, a la ip del DNS con la consulta especifica, la consulta origen sale del puerto 60181 al puerto 53 del servidor DNS; la consulta(query) simplemente contiene el nombre de dominio y el tipo de consulta en este caso es A (address), y la clase que es IN(internet), en la segunda linea, el servidor DNS responde(answere) al host origen con lo siguiente: toda la info requerida al inicio mas la dirección IP de la dirección requerida. en este caso es: 216.58.202.4 (ver figura 6)
- 2: A traves del protocolo ICMP(internet control message protocol) el host envia y recibe paquetes del servidor de google en este caso, estos mensajes simplemente contienen un request y un reply en los cuales los numeros de secuencia tanto para request y reply son los mismos, es un ida-verificación-respuesta continua de 3 paquetes.

No.	Time	Source	Destination	Protocol	Length	Info
3	4.097186	192.168.0.4	186.16.16.16	DNS	74	Standard query 0xc74c A www.google.com
4	4.114472	186.16.16.16	192.168.0.4	DNS	98	Standard query response 0xc74c A www.google.com A 216.58.202.4 1
5	4.121790	192.168.0.4	216.58.202.4	ICMP	74	Echo (ping) request id=0x0001, seq=73/18688, ttl=128 (reply in 6)
6	4.175177	216.58.202.4	192.168.0.4	ICMP	74	Echo (ping) reply id=0x0001, seq=73/18688, ttl=51 (request in 5)
7	5.134403	192.168.0.4	216.58.202.4	ICMP	74	Echo (ping) request id=0x0001, seq=74/18944, ttl=128 (reply in 8)
8	5.191609	216.58.202.4	192.168.0.4	ICMP	74	Echo (ping) reply id=0x0001, seq=74/18944, ttl=51 (request in 7)
10	6.150313	192.168.0.4	216.58.202.4	ICMP	74	Echo (ping) request id=0x0001, seq=75/19200, ttl=128 (reply in 11)
11	6.200120	216.58.202.4	192.168.0.4	ICMP	74	Echo (ping) reply id=0x0001, seq=75/19200, ttl=51 (request in 10)
12	7.166087	192.168.0.4	216.58.202.4	ICMP	74	Echo (ping) request id=0x0001, seq=76/19456, ttl=128 (no response found!)


```

> Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: IntelCor_5b:d5:47 (34:e6:ad:5b:d5:47), Dst: Arrisdro_00:00:03 (00:00:ca:00:00:03)
> Internet Protocol Version 4, Src: 192.168.0.4, Dst: 186.16.16.16
  > User Datagram Protocol, Src Port: 60181, Dst Port: 53
    Source Port: 60181
    Destination Port: 53
    Length: 40
    Checksum: 0x3392 [unverified]
    [Checksum Status: Unverified]
    [Stream Index: 0]
  > Domain Name System (query)
    Transaction ID: 0xc74c
    > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    > Queries
      [Response In: 4]

```

Figura 5: Consulta DNS-WIRESHARK

```

Additional RRs: 0
Queries
  > www.google.com: type A, class IN
    Name: www.google.com
    [Name Length: 14]
    [Label Count: 3]
    Type: A (Host Address) (1)
    Class: IN (0x0001)
Answers
  > www.google.com: type A, class IN, addr 216.58.202.4
    Name: www.google.com
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 260
    Data length: 4
    Address: 216.58.202.4
    [Request In: 3]
    [Time: 0.017286000 seconds]

```

Figura 6: Consulta DNS-Query-Answer-WIRESHARK

En la figura 6 Podemos observar como se comporta el servidor DNS al dar una respuesta de localización de la IP del host solicitado

3.4. Envio de Email

En este procedimiento capturaremos los paquetes al realizar un envío de email desde un cliente web de la maquina host, visto que no tenemos un servidor de correo habilitado incitu, veremos como se realiza el proceso de envío desde el cliente de correos de windows.

3.4.1. Explicación

Analizaremos el procedimiento de captura, primero verificando que todos los procedimientos de envío desde servidores web como google, microsoft, etc se realizan a través de protocolos Seguros, osea, utilizan TLS v.1.2 como protocolo de conexión socket con el servidor y así envían la información a través del canal seguro pre-establecido (Ver figura 7 - label intro). Luego Procedimos al análisis de los paquetes tanto enviados y recibidos, y nos dimos cuenta de que:

- 1: La conexión socket con el servidor es segura, para envíos
- 2: Errores de retransmisión entre el servidor host y el servidor de microsoft mail
- 3: Errores en donde el protocolo avisa algunos errores de transmisión en cuanto a ACK duplicados al momento del envío del mail

No	Time	Source	Destination	Protocol	Length	Info
1	0.000000	191.232.98.34	192.168.0.4	TCPv1	127	Application Data, Application Data
2	0.000311	192.168.0.4	191.232.98.34	TCP	60	64081 → 443 [ACK] Seq=224 Wm=1024 Len=0 Sli=188 Sre=224
7	0.006413	216.58.282.17	192.168.0.4	TCPv1	174	Application Data, Application Data
8	0.947295	192.168.0.4	216.58.202.67	TCPv1	100	Application Data
5	1.221892	216.58.282.17	192.168.0.4	TCP	60	443 → 64081 [ACK] Seq=781 ack=Wm=176 Len=0
9	1.964186	192.168.0.4	191.232.98.34	TCPv1	114	64082 → 443 [ACK] Seq=1581 Wm=1038 Len=1 [TCP segment of a reassembled PDU]
9	1.964307	192.168.0.4	191.232.98.34	TCPv1	114	64082 → 443 [ACK] Seq=1588 ACK=1018 Len=1460 [TCP segment of a reassembled PDU]
10	1.212724	192.168.0.4	191.232.98.34	TCPv1	107	Application Data
11	2.458284	192.168.0.4	191.232.98.34	TCP	1514	TCP Retransmission(1514) → 443 [PSH, ACK] Seq=3401 Sli=1818 Len=1460
12	2.458284	192.168.0.4	191.232.98.34	TCP	1514	TCP Retransmission(1514) → 443 [ACK] Seq=1581 Wm=1018 Len=1460
13	2.458284	192.168.0.4	191.232.98.34	TCP	60	443 → 64081 [ACK] Seq=1581 Wm=1018 Len=1460
14	2.458321	192.168.0.4	191.232.98.34	TCPv1	1514	TCP Retransmission(1514) → 443 [ACK] Seq=1588 ACK=1018 Len=1460
15	2.458321	192.168.0.4	191.232.98.34	TCPv1	1514	TCP Retransmission(1514) → 443 [ACK] Seq=1588 ACK=1018 Len=1460
16	2.458321	192.168.0.4	191.232.98.34	TCP	60	443 → 64081 [ACK] Seq=1581 Wm=1018 Len=1460
17	2.444939	191.232.98.34	192.168.0.4	TCP	66	TCP Dup ACK 1681 [ACK] Seq=1680 [ACK] Seq=1681 Wm=5535 Len=0 Sli=2341 Sre=1381
18	2.513126	191.232.98.34	192.168.0.4	TCP	66	TCP Dup ACK 1682 [ACK] Seq=1680 [ACK] Seq=1681 Wm=5535 Len=0 Sli=2341 Sre=1381
19	2.543676	191.232.98.34	192.168.0.4	TCP	66	TCP Dup ACK 1683 [ACK] Seq=1680 [ACK] Seq=1681 Wm=5535 Len=0 Sli=2341 Sre=1381
20	2.513676	191.232.98.34	192.168.0.4	TCP	66	TCP Dup ACK 1684 [ACK] Seq=1680 [ACK] Seq=1681 Wm=5535 Len=0 Sli=2341 Sre=1381
21	4.390182	192.168.0.4	52.165.171.165	TCPv1	127	Application Data
22	4.391409	52.165.171.165	192.168.0.4	TCP	179	Application Data
24	6.939591	192.168.0.4	52.165.171.165	TCPv1	54	83399 → 443 [ACK] Seq=744 Wm=1266 Len=0
25	5.046479	191.232.98.34	192.168.0.4	TCPv1	92	Application Data
26	5.046585	192.168.0.4	191.232.98.34	TCP	54	64083 → 443 [ACK] Seq=1 Wm=1023 Len=0
27	5.523313	191.232.98.34	192.168.0.4	TCPv1	90	Application Data
28	5.523378	192.168.0.4	191.232.98.34	TCP	54	64083 → 443 [ACK] Seq=290 Wm=1023 Len=0
29	5.525275	191.232.98.34	192.168.0.4	TCPv1	90	Application Data
30	5.525276	191.232.98.34	192.168.0.4	TCPv1	90	Application Data

Figura 7: Envio de email

3.5. Descarga de archivos P2P

[illegible]

Figura 8: P2P-DOwnload File

En esta sección analizaremos los paquetes que se reciben y envían en el proceso de descarga de archivos a través de la herramienta bittorrent que usa el protocolo p2p. (ver figura 8)

- 1: Handshake entre el servidor de torrent y el host, a través del protocolo Bittorrent, en esta sección solo se realiza una solicitud unidireccional, se pueden observar que los puertos que utilizan son muy diferentes a los que normalmente conocemos para intercambio de paquetes.
- 2: Envío de paquetes a través de protocolo UDP del servidor luego de recibir el handshake, el host a través del programa responde con un ok por el mismo protocolo
- 3 y 4: a través del protocolo TCP simplemente envío y recepción y finalización de ACK's
- 5: Respuesta de Handshake con el mismo protocolo BitTorrent

3.6. Intercambio de paquetes para conectarse a una red wifi segura

Hemos realizado las capturas al conectarse a una red wifi previamente conocida.

3.6.1. Explicación

- 1:DHCP request, El cliente pide una dirección IP al host, el mismo utiliza el protocolo dhcp, en el ejemplo de la figura 9, el cliente envía como cabecera el mac address correspondiente a sí mismo, recibe una dirección IP.

Time	Source	Destination	Protocol	Length	Info
2.8.00000	192.168.1.1	228.255.255.255	DHCP	300	DHCP Request: 192.168.1.1, 192.168.1.1
2.8.00079	192.168.1.1	Broadcast	ARP	42	who has 192.168.1.1? 192.168.1.1
2.8.00104	fe80::b3da:164d::f82c::18	ff02::1b	DHCP	96	Standard query request v2
2.8.00160	192.168.1.1	228.255.255.255	DHCP	54	Membership Report: Join group 228.255.255.255 for any sources
2.8.00255	fe80::b3da:164d::f82c::18	ff02::1b	LLMNR	68	Standard query 80/80 ask v2
2.8.00260	192.168.1.1	228.255.255.255	LLMNR	68	Standard query 80/80 ask v2
2.8.00301	192.168.1.1	Broadcast	ARP	42	who has 192.168.1.1? 192.168.1.1
2.8.00306	192.168.1.1	228.255.255.255	DHCP	79	Membership Report: Join group 228.255.255.255 for any sources / Join group 228.102.132.141 for any sources / Join group 228.4.8.255 for any sources
2.8.00354	fe80::b3da:164d::f82c::18	ff02::1b	DHCP	79	Neighbor Solicitation request v2
2.8.00359	192.168.1.1	Broadcast	ARP	42	Neighbor Solicitation request v2
2.8.00360	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00361	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00362	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00363	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00364	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00365	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00366	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00367	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00368	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00369	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00370	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00371	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00372	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00373	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00374	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00375	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00376	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00377	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00378	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00379	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00380	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00381	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00382	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00383	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00384	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00385	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00386	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00387	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00388	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00389	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00390	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00391	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00392	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00393	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00394	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00395	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00396	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00397	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00398	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00399	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00400	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00401	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00402	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00403	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00404	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00405	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00406	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00407	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00408	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00409	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00410	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00411	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00412	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00413	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00414	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00415	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00416	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00417	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00418	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00419	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00420	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00421	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00422	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00423	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00424	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00425	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00426	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00427	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00428	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00429	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00430	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00431	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00432	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00433	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00434	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00435	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00436	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00437	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00438	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00439	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00440	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00441	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00442	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00443	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00444	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00445	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00446	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00447	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00448	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00449	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00450	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00451	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00452	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00453	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00454	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00455	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00456	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00457	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00458	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00459	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00460	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00461	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00462	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00463	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00464	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00465	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00466	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00467	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00468	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00469	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00470	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00471	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00472	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00473	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00474	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00475	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00476	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00477	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00478	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00479	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00480	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00481	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00482	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00483	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00484	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00485	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00486	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00487	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00488	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00489	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00490	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00491	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00492	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00493	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00494	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00495	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00496	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00497	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00498	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00499	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00500	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00501	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00502	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00503	192.168.1.1	228.255.255.255	DHCP	118	Standard query request v2
2.8.00504	192.168.1.1	228.255.255.255	DHCP	118	

Figura 9: Login Wifi

- 2: Identificador del router destino.
- 3: El protocolo ICMPv6 es una nueva versión de ICMP y es una parte importante de la arquitectura IPv6 que debe estar completamente soportada por todas las implementaciones y nodos IPv6
- 4: La resolución de nombre de multidifusión local de enlace (LLMNR) es un protocolo basado en el formato de paquete de sistema de nombres de dominio (DNS) que permite que los hosts IPv4 e IPv6 realicen la resolución de nombre para hosts en el mismo enlace local.
- 5: El protocolo de red IGMP se utiliza para intercambiar información acerca del estado de pertenencia entre enrutadores IP que admiten la multidifusión y miembros de grupos de multidifusión. Los hosts miembros individuales informan acerca de la pertenencia de hosts al grupo de multidifusión y los enrutadores de multidifusión sondan periódicamente el estado de la pertenencia.

3.7. Intercambio de paquetes para conexión DHCP

Hemos mostrado en el ítem anterior, al conectarse a la red wifi como el protocolo DHCP funciona y que paquetes se intercambian con el host que requiere la ip.