

Laboratorio 2.2 - Análisis de protocolos de la capa de aplicación utilizando Wireshark

Juan Alegría – 202011282

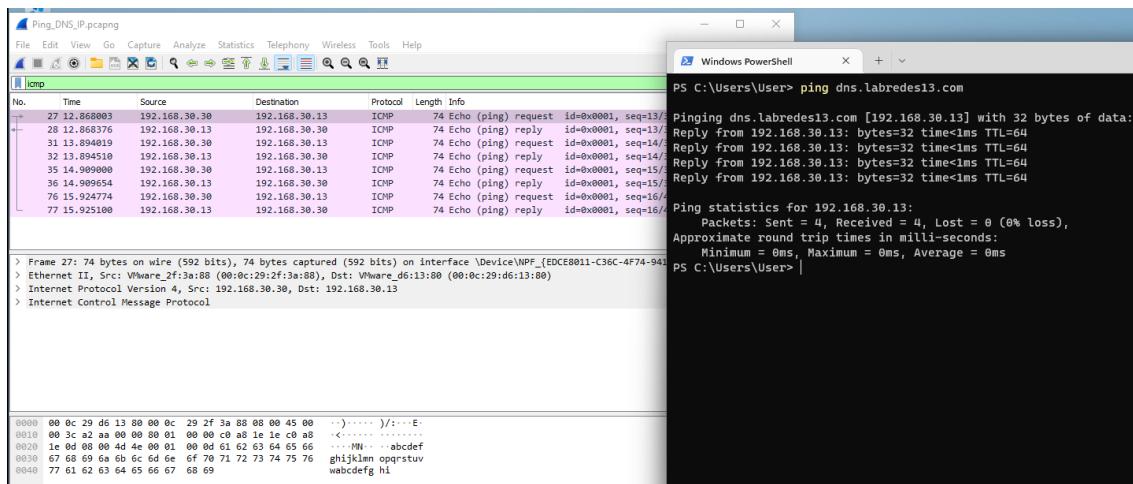
Juan Andrés Romero – 202013449

Evidencia de realización de actividades de laboratorio

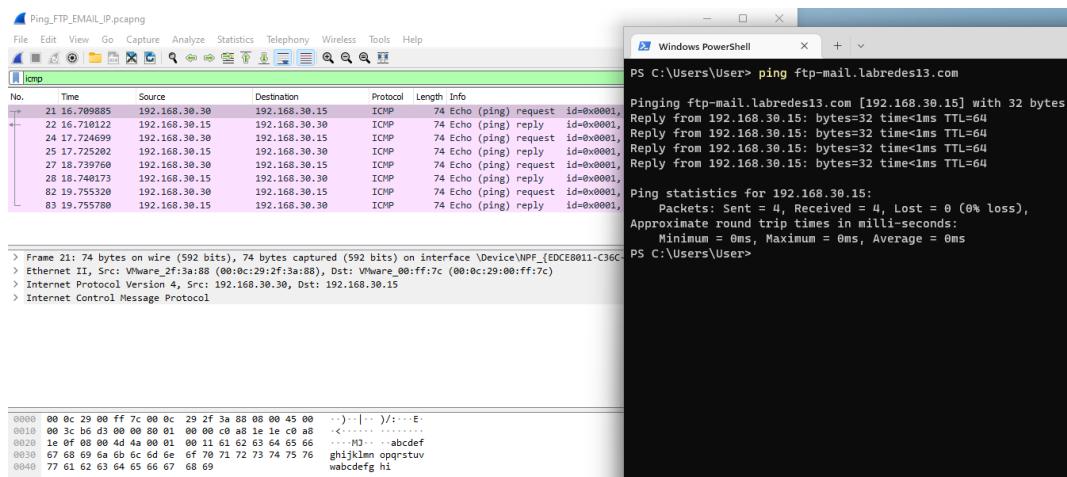
Para visualizar los archivos de captura de tráfico ingrese al siguiente enlace:

4.1 Pruebas de Conectividad

4.1.1 Pruebas de Ping DNS



4.1.2 Pruebas de Ping FTP-Mail



4.1.3 Identificación de Información de los archivos

Archivo DNS

1. Comando Request:

- IP Source: 192.168.30.30
- IP Destination: 192.168.30.13

2. Comando Reply:

- IP Source: 192.168.30.13
- IP Destination: 192.168.30.30

3. Direcciones MAC:

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
+	27 12.868003	192.168.30.30	192.168.30.13	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 28)
-	28 12.868376	192.168.30.13	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=64 (request in 27)
31	13.894019	192.168.30.30	192.168.30.13	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 32)
32	13.894510	192.168.30.13	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=64 (request in 31)
35	14.899000	192.168.30.30	192.168.30.13	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 36)
36	14.909654	192.168.30.13	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=64 (request in 35)
76	15.924774	192.168.30.30	192.168.30.13	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 77)
-	77 15.925100	192.168.30.13	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=64 (request in 76)


```
> Frame 27: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
< Ethernet II, Src: VMware_2f:3a:88 (00:0c:29:2f:3a:88), Dst: VMware_d6:13:80 (00:0c:29:d6:13:80)
  > Destination: VMware_d6:13:80 (00:0c:29:d6:13:80)
  > Source: VMware_2f:3a:88 (00:0c:29:2f:3a:88)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.13
  > Internet Control Message Protocol
```

- MAC Cliente: 00:0C:29:2F:3A:88
- MAC Servidor: 00:0C:29:D6:13:80

Archivo FTP Email

1. Comando Request:

- IP Source: 192.168.30.30
- IP Destination: 192.168.30.15

2. Comando Reply:

- IP Source: 192.168.30.15
- IP Destination: 192.168.30.30

3. Direcciones MAC:

No.	Time	Source	Destination	Protocol	Length	Info
21	16.709885	192.168.30.30	192.168.30.15	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (reply in 22)
22	16.710122	192.168.30.15	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=17/4352, ttl=64 (request in 21)
24	17.724699	192.168.30.30	192.168.30.15	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (reply in 25)
25	17.725202	192.168.30.15	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=18/4608, ttl=64 (request in 24)
27	18.739760	192.168.30.30	192.168.30.15	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (reply in 28)
28	18.740173	192.168.30.15	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=19/4864, ttl=64 (request in 27)
82	19.755320	192.168.30.30	192.168.30.15	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 83)
83	19.755780	192.168.30.15	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=64 (request in 82)

```
> Frame 21: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
✓ Ethernet II, Src: VMware_2f:3a:88 (00:0c:29:2f:3a:88), Dst: VMware_00:ff:7c (00:0c:29:00:ff:7c)
  > Destination: VMware_00:ff:7c (00:0c:29:00:ff:7c)
  > Source: VMware_2f:3a:88 (00:0c:29:2f:3a:88)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.15
> Internet Control Message Protocol
```

0000	00 0c 29 00 ff 7c 00 0c 29 2f 3a 88 08 00 45 00	...)- ...):/...E-
0010	00 3c b6 d3 00 00 80 01 00 00 c0 a8 1e 1e c0 a8	<-----
0020	1e 0f 08 00 4d 4a 00 01 00 11 61 62 63 64 65 66	...MJ... abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuvwxyz
0040	77 61 62 63 64 65 66 67 68 69	wabdefghi

- MAC Cliente: 00:0C:29:2F:3A:88
- MAC Servidor: 00:0C:29:00:FF:7C

4.2 Análisis del Tráfico DNS

4.2.1

The screenshot shows NetworkMiner capturing ICMP traffic between two hosts. The traffic consists of several Echo (ping) requests and replies. A PowerShell window is open, showing the command `ping web.labredes13.com` being run, which returns successful results with a TTL of 64. Below the traffic capture, the raw hex and ASCII data for one of the ICMP frames is displayed.

No.	Time	Source	Destination	Protocol	Length	Info
22	4.281648	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=1/25
25	4.282367	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=1/25
35	5.303072	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=2/25
31	5.303520	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=2/25
41	6.318894	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=3/25
42	6.319234	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=3/25
47	7.334305	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=4/25
48	7.334718	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=4/25

```
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d43 [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 24 (0x0018)
Sequence Number (LE): 6144 (0x1800)
[Response frame: 25]
Data (32 bytes)
Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
[Length: 32]

0000 00 0c 29 e5 4c 04 00 0c 29 2f 3a 88 08 00 45 00  ...)-L... ):/...E-
0010 00 3c d1 46 00 00 80 01 00 00 c0 a8 1e 1e c0 a8  <----- .....
0020 1e 0e 08 00 4d 43 00 01 00 18 61 62 63 64 65 66  ...-MC... abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstuvwxyz
0040 77 61 62 63 64 65 66 67 68 69  wabdefghi
```

4.2.2

```
PS C:\Users\User> ipconfig /displaydns

Windows IP Configuration

dns.labredes13.com
-----
Record Name . . . . . : dns.labredes13.com
Record Type . . . . . : 1
Time To Live . . . . . : 84283
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . . : 192.168.30.13

web.labredes13.com
-----
Record Name . . . . . : web.labredes13.com
Record Type . . . . . : 1
Time To Live . . . . . : 86237
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . . : 192.168.30.14

ftp-mail.labredes13.com
-----
Record Name . . . . . : ftp-mail.labredes13.com
Record Type . . . . . : 1
Time To Live . . . . . : 84769
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . . : 192.168.30.15

PS C:\Users\User>
```

4.2.3

```
PS C:\Users\User> ipconfig /flushdns

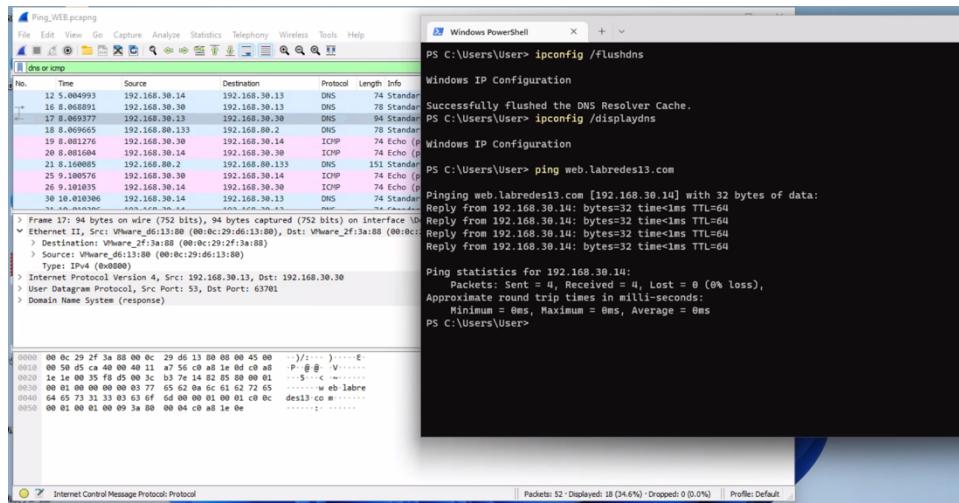
Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
PS C:\Users\User> ipconfig /displaydns

Windows IP Configuration

PS C:\Users\User>
```

4.2.4



4.2.5 Análisis de archivos .pcap

a) Información de la capa de aplicación:

Standard query 0x1482 A web.labredes.com

Standardquery response 0x1482 A web.labredes13.com A 192.168.30.14

```
✓ Domain Name System (response)
  Transaction ID: 0xa313
  > Flags: 0x8183 Standard query response, No such name
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ✓ Queries
    ✓ web.labredes13.com: type A, class IN
      Name: web.labredes13.com
      [Name Length: 18]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  ✓ Authoritative nameservers
    ✓ com: type SOA, class IN, mname a.gtld-servers.net
      Name: com
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 5 (5 seconds)
      Data length: 61
      Primary name server: a.gtld-servers.net
      Responsible authority's mailbox: ns1.verisign-grs.com
      Serial Number: 1662834437
      Refresh Interval: 1800 (30 minutes)
      Retry Interval: 900 (15 minutes)
      Expire limit: 604800 (7 days)
      Minimum TTL: 86400 (1 day)
      [Request In: 18]
      [Time: 0.090420000 seconds]
```

b) Protocolo de la capa de transporte utilizado: UDP

c) Puertos utilizados:

- Cliente: 63701

- Servidor: 53

```
> Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.13
✓ User Datagram Protocol, Src Port: 63701, Dst Port: 53
  Source Port: 63701
  Destination Port: 53
  Length: 44
  Checksum: 0xbdbb9 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 2]
```

d) Tabla del servidor WEB

Comando Request:

- IP Source: 192.168.30.30
- IP Destination: 192.168.30.14

Comando Reply:

- IP Source: 192.168.30.14
- IP Destination: 192.168.30.30

3. Direcciones MAC:

No.	Time	Source	Destination	Protocol	Length	Info
19	8.081276	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=28/7168, ttl=128 (reply in 20)
20	8.081604	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=28/7168, ttl=64 (request in 19)
25	9.100576	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=29/7424, ttl=128 (reply in 26)
26	9.101035	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=29/7424, ttl=64 (request in 25)
32	10.116282	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=30/7680, ttl=128 (reply in 33)
33	10.116969	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=30/7680, ttl=64 (request in 32)
37	11.147811	192.168.30.30	192.168.30.14	ICMP	74	Echo (ping) request id=0x0001, seq=31/7936, ttl=128 (reply in 38)
38	11.148240	192.168.30.14	192.168.30.30	ICMP	74	Echo (ping) reply id=0x0001, seq=31/7936, ttl=64 (request in 37)

```

> Frame 19: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
  Ethernet II, Src: VMware_2f:3a:88 (00:0c:29:e5:4c:04), Dst: VMware_e5:4c:04 (00:0c:29:e5:4c:04)
    > Destination: VMware_e5:4c:04 (00:0c:29:e5:4c:04)
    > Source: VMware_2f:3a:88 (00:0c:29:f2:3a:88)
      Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.14
  > Internet Control Message Protocol

```

- MAC Cliente: 00:0C:29:2F:3A:88
- MAC Servidor: 00:0C:29:E5:4C:04

4.3 Análisis del Tráfico FTP

FTP_upload.pcapng						
No.	Time	Source	Destination	Protocol	Length	Info
34	10.838896	192.168.30.15	192.168.30.30	FTP	106	Response: 220 ProFTPD Server (Debian) [::ffff:192.168.30.15]
35	10.839592	192.168.30.30	192.168.30.15	FTP	64	Request: AUTH TLS
37	10.840351	192.168.30.15	192.168.30.30	FTP	79	Response: 500 AUTH not understood
38	10.840576	192.168.30.30	192.168.30.15	FTP	64	Request: AUTH SSL
40	10.840987	192.168.30.15	192.168.30.30	FTP	79	Response: 500 AUTH not understood
41	10.842484	192.168.30.30	192.168.30.15	FTP	69	Request: USER usuario1
43	10.842984	192.168.30.15	192.168.30.30	FTP	90	Response: 331 Password required for usuario1
44	10.843164	192.168.30.30	192.168.30.15	FTP	69	Request: PASS usuario1
46	10.854511	192.168.30.15	192.168.30.30	FTP	83	Response: 230 User usuario1 logged in
47	10.857775	192.168.30.30	192.168.30.15	FTP	59	Request: PWD

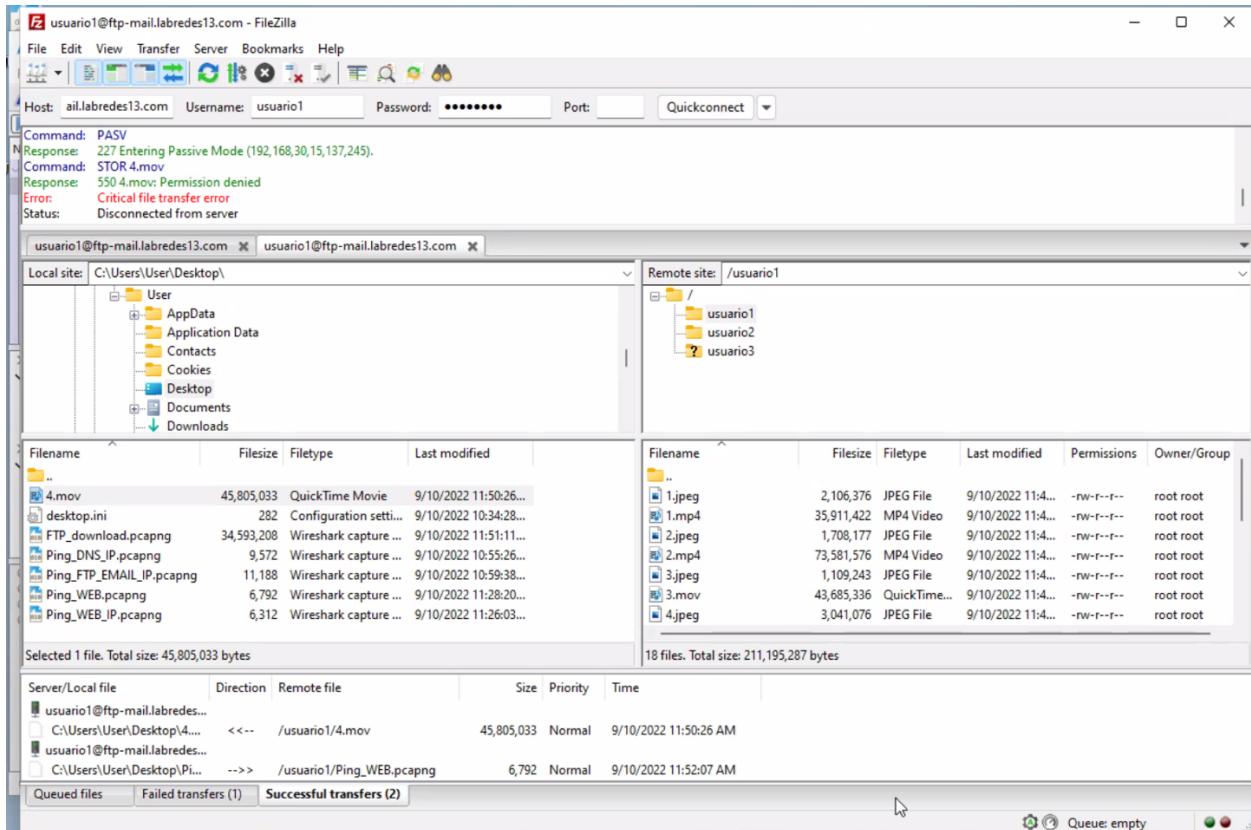
```

> Frame 35: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
  Ethernet II, Src: VMware_2f:3a:88 (00:0c:29:2f:3a:88), Dst: VMware_00:ff:7c (00:0c:29:00:ff:7c)
    > Destination: VMware_00:ff:7c (00:0c:29:00:ff:7c)
    > Source: VMware_2f:3a:88 (00:0c:29:2f:3a:88)
      Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.15
  > Transmission Control Protocol, Src Port: 60583, Dst Port: 21, Seq: 1, Ack: 53, Len: 10
    Source Port: 60583
    Destination Port: 21
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 10]
0000  00 0c 29 00 ff 7c 00 0c 29 2f 3a 88 00 45 00  ..|..|.. ):/---E
0010  00 32 c4 54 40 00 80 06 00 00 c0 a8 1e c0 a8  ..2 T@..... 
0020  1e 0f ec a7 00 15 49 4b e6 cf 11 5f 9e 70 50 18  ....IK ..._pP...
0030  04 02 bd a2 00 41 55 54 48 20 54 4c 53 0d 0a  ....AU TH TLS..

```

File Transfer Protocol (FTP): Protocol

Packets: 265 · Displayed: 49 (18.5%) · Dropped: 0 (0.0%) || Profile: Default



4.3.5 Análisis de archivos .pcap

a) Información de la capa de aplicación:

- Request AUTH

```
File Transfer Protocol (FTP)
> AUTH TLS\r\n
[Current working directory: ]
```

- Request USER

```
File Transfer Protocol (FTP)
> USER usuario1\r\n
[Current working directory: ]
```

- Request PASS

```
File Transfer Protocol (FTP)
> PASS usuario1\r\n
[Current working directory: ]
```

- Request LIST

```
✓ File Transfer Protocol (FTP)
  > LIST\r\n
[Current working directory: /]
[Command response frames: 1]
[Command response bytes: 195]
[Command response first frame: 89]
[Command response last frame: 89]
[Setup frame: 82]
```

- Request RETR

```
✓ File Transfer Protocol (FTP)
  > RETR 4.mov\r\n
[Current working directory: /usuario1]
[Command response frames: 22005]
[Command response bytes: 32127300]
[Command response first frame: 297]
[Command response last frame: 25343]
[Response duration: 581ms]
[Response bitrate: 442372Kbps]
[Setup frame: 291]
```

Resumen de la captura de paquetes:

Time	Source IP	Destination IP	Protocol	Request/Response
26 12.718217	192.168.30.15	192.168.30.30	FTP	106 Response: 220 ProFTPD Server (Debian) [::ffff:192.168.30.15]
27 12.718606	192.168.30.30	192.168.30.15	FTP	64 Request: AUTH TLS
29 12.719463	192.168.30.15	192.168.30.30	FTP	79 Response: 500 AUTH not understood
30 12.719570	192.168.30.30	192.168.30.15	FTP	64 Request: AUTH SSL
32 12.720092	192.168.30.15	192.168.30.30	FTP	79 Response: 500 AUTH not understood
33 12.721880	192.168.30.30	192.168.30.15	FTP	69 Request: USER usuario1
35 12.722808	192.168.30.15	192.168.30.30	FTP	99 Response: 331 Password required for usuario1
36 12.723082	192.168.30.30	192.168.30.15	FTP	69 Request: PASS usuario1
38 12.723108	192.168.30.15	192.168.30.30	FTP	83 Response: 230 User usuario1 logged in
40 12.734628	192.168.30.30	192.168.30.15	FTP	60 Request: SYST
41 12.735185	192.168.30.15	192.168.30.30	FTP	73 Response: 215 UNIX Type: L8
42 12.735400	192.168.30.30	192.168.30.15	FTP	60 Request: FEAT
74 12.739077	192.168.30.30	192.168.30.15	FTP	59 Request: PWD
76 12.739573	192.168.30.15	192.168.30.30	FTP	88 Response: 257 "/" is the current directory
77 12.740255	192.168.30.30	192.168.30.15	FTP	62 Request: TYPE I
79 12.740838	192.168.30.15	192.168.30.30	FTP	73 Response: 200 Type set to I
80 12.741083	192.168.30.30	192.168.30.15	FTP	60 Request: PASV
82 12.741725	192.168.30.15	192.168.30.30	FTP	102 Response: 227 Entering Passive Mode (192,168,30,15,138,117).
83 12.742200	192.168.30.30	192.168.30.15	FTP	60 Request: LIST
88 12.743339	192.168.30.15	192.168.30.30	FTP	108 Response: 150 Opening BINARY mode data connection for file list
93 12.744273	192.168.30.15	192.168.30.30	FTP	77 Response: 226 Transfer complete
98 14.561459	192.168.30.30	192.168.30.15	FTP	68 Request: CWD usuario1
108 14.562060	192.168.30.15	192.168.30.30	FTP	82 Response: 250 CWD command successful
108 14.562217	192.168.30.30	192.168.30.15	FTP	59 Request: PWD
102 14.562604	192.168.30.15	192.168.30.30	FTP	99 Response: 257 "/usuario1" is the current directory
103 14.562767	192.168.30.30	192.168.30.15	FTP	60 Request: PASV
104 14.563182	192.168.30.15	192.168.30.30	FTP	108 Response: 227 Entering Passive Mode (192,168,30,15,161,119).
105 14.563468	192.168.30.30	192.168.30.15	FTP	60 Request: LIST
109 14.564288	192.168.30.15	192.168.30.30	FTP	109 Response: 150 Opening BINARY mode data connection for file list
114 14.565787	192.168.30.15	192.168.30.30	FTP	77 Response: 226 Transfer complete
268 33.059806	192.168.30.15	192.168.30.30	FTP	108 Response: 220 ProFTPD Server (Debian) [::ffff:192.168.30.15]
269 33.060078	192.168.30.30	192.168.30.15	FTP	64 Request: AUTH TLS
273 33.061767	192.168.30.15	192.168.30.30	FTP	79 Response: 500 AUTH not understood
273 33.062002	192.168.30.30	192.168.30.15	FTP	60 Request: AUTH SSL
273 33.062567	192.168.30.15	192.168.30.30	FTP	79 Response: 500 AUTH not understood
275 33.064631	192.168.30.30	192.168.30.15	FTP	69 Request: USER usuario1
277 33.065270	192.168.30.15	192.168.30.30	FTP	99 Response: 331 Password required for usuario1
278 33.065465	192.168.30.30	192.168.30.15	FTP	69 Request: PASS usuario1
288 33.075783	192.168.30.15	192.168.30.30	FTP	83 Response: 230 User usuario1 logged in
288 33.078366	192.168.30.30	192.168.30.15	FTP	69 Request: CWD /usuario1
288 33.079338	192.168.30.15	192.168.30.30	FTP	82 Response: 250 CWD command successful
284 33.079489	192.168.30.30	192.168.30.15	FTP	59 Request: PWD
286 33.080073	192.168.30.15	192.168.30.30	FTP	96 Response: 257 "/usuario1" is the current directory
287 33.099507	192.168.30.30	192.168.30.15	FTP	62 Request: TYPE I
289 33.100110	192.168.30.15	192.168.30.30	FTP	73 Response: 200 Type set to I
293 33.100313	192.168.30.30	192.168.30.15	FTP	60 Request: PASV
293 33.100886	192.168.30.15	192.168.30.30	FTP	108 Response: 227 Entering Passive Mode (192,168,30,15,140,15).
292 33.101405	192.168.30.30	192.168.30.15	FTP	60 Request: RETR 4.mov
296 33.102776	192.168.30.15	192.168.30.30	FTP	122 Response: 150 Opening BINARY mode data connection for 4.mov (45805033 bytes)
25346 33.729293	192.168.30.15	192.168.30.30	FTP	77 Response: 226 Transfer complete

b) Protocolo de la capa de transporte utilizado: TCP

c) Puertos utilizados por el programa FTP

- Cliente: 60583

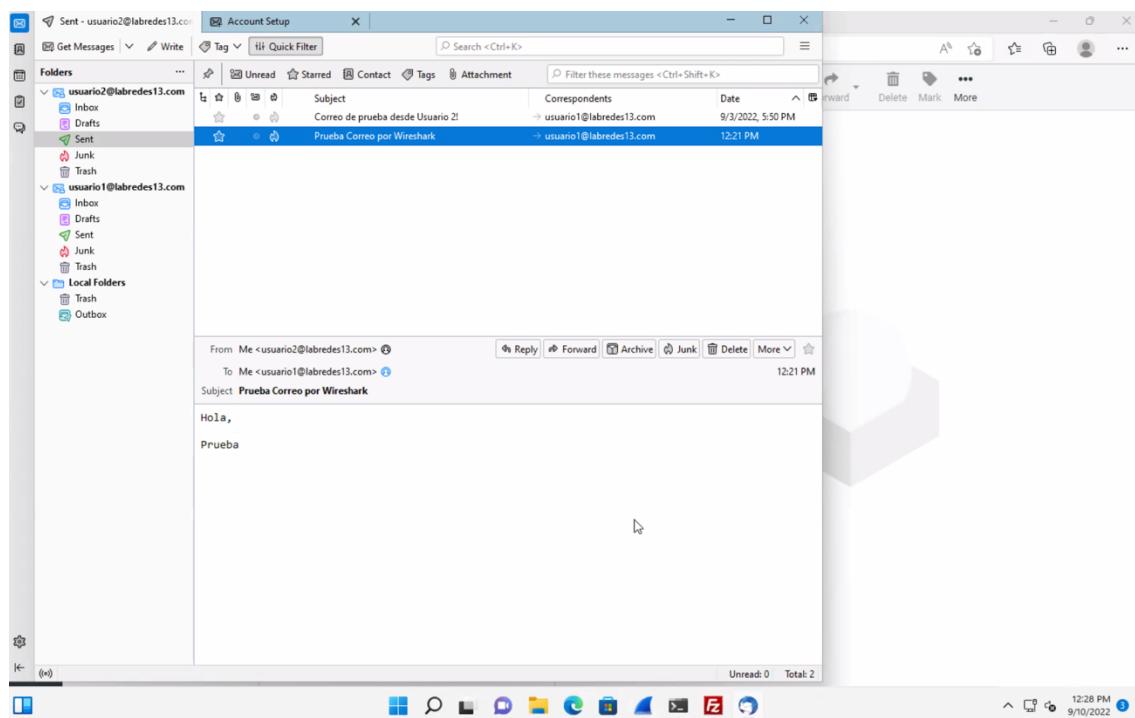
- Servidor: 21

No.	Time	Source	Destination	Protocol	Length	Info
202	21.144922	192.168.30.30	192.168.30.15	FTP	62	Request: TYPE I
203	21.145457	192.168.30.15	192.168.30.30	FTP	73	Response: 200 Type set to I
204	21.145625	192.168.30.30	192.168.30.15	FTP	60	Request: PASV
205	21.146202	192.168.30.15	192.168.30.30	FTP	105	Response: 227 Entering Passive Mode (192,168,30,15,143,31).
206	21.146534	192.168.30.30	192.168.30.15	FTP	76	Request: STOR Ping_WEB.pcapng
212	21.147483	192.168.30.15	192.168.30.30	FTP	115	Response: 150 Opening BINARY mode data connection for Ping_WEB.pcapng
216	21.148053	192.168.30.15	192.168.30.30	FTP	77	Response: 226 Transfer complete
218	21.155337	192.168.30.30	192.168.30.15	FTP	60	Request: PASV
219	21.155996	192.168.30.15	192.168.30.30	FTP	106	Response: 227 Entering Passive Mode (192,168,30,15,151,221).
220	21.156282	192.168.30.30	192.168.30.15	FTP	60	Request: LIST

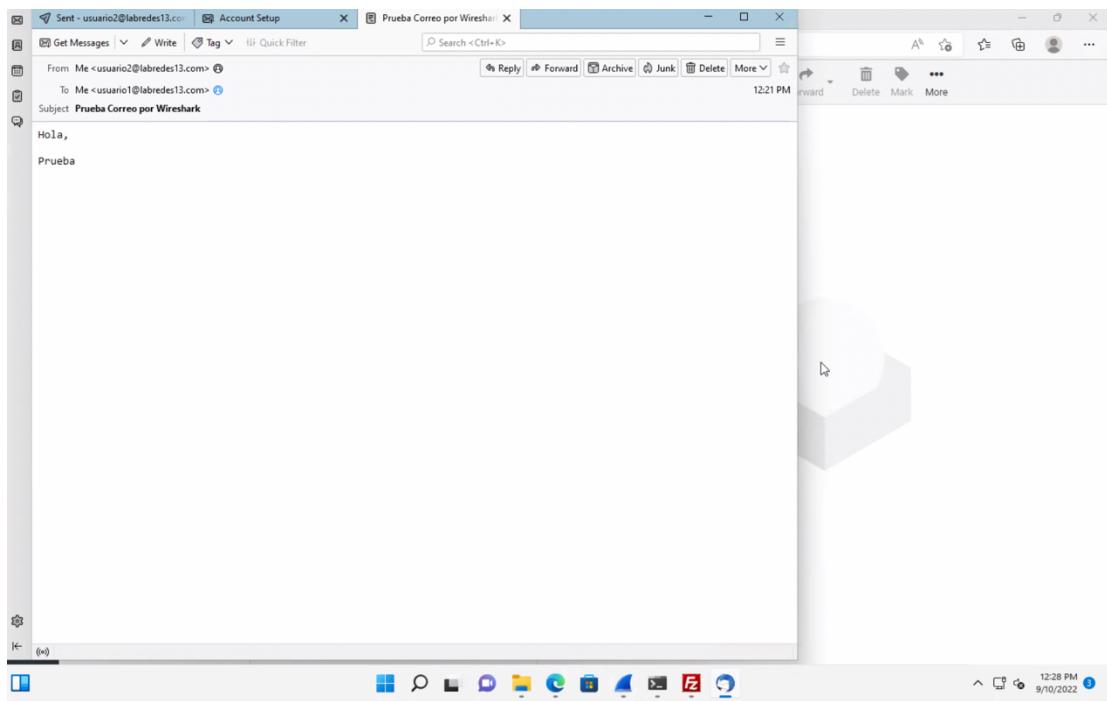
> Frame 206: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
Ethernet II, Src: VMware_2f:3a:88 (00:0c:29:2f:3a:88), Dst: VMware_00:ff:7c (00:0c:29:00:ff:7c)
> Destination: VMware_00:ff:7c (00:0c:29:00:ff:7c)
> Source: VMware_2f:3a:88 (00:0c:29:2f:3a:88)
Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.30.30, Dst: 192.168.30.15
Transmission Control Protocol, Src Port: 60585, Dst Port: 21, Seq: 80, Ack: 266, Len: 22
Source Port: 60585
Destination Port: 21
[Stream index: 2]
[Conversation completeness: Incomplete, DATA (15)]
[TTO Comment len: 22]

4.4 Análisis del Tráfico Correo

Envío de Correo por Thunderbird



Correo Recibido!



4.4.5 Análisis de archivos .pcap

a) Información de la capa de aplicación:

Comandos notorios encontrados

- ECHLO

```
✗ Simple Mail Transfer Protocol
  ✓ Command Line: EHLO [192.168.30.30]\r\n
    Command: EHLO
    Request parameter: [192.168.30.30]
```

- PIPELINING

```
✗ Simple Mail Transfer Protocol
  ✓ Response: 250-ftp-mail.labredes13.com\r\n
    Response code: Requested mail action okay, completed (250)
    Response parameter: ftp-mail.labredes13.com
    Response parameter: PIPELINING
    Response parameter: SIZE 15728640
    Response parameter: ETRN
    Response parameter: STARTTLS
    Response parameter: ENHANCEDSTATUSCODES
    Response parameter: 8BITMIME
    Response parameter: DSN
    Response parameter: SMTPUTF8
    Response parameter: CHUNKING
```

- STARTTLS

- ✓ Simple Mail Transfer Protocol
 - ▼ Command Line: STARTTLS\r\n
 - Command: STAR
 - Request parameter: TLS

No.	Time	Source	Destination	Protocol	Length	Info
154	34.012594	192.168.30.15	192.168.30.30	SMTP	97 S:	220 ftp-mail.labredes13.com ESMTP Postfix
155	34.034012	192.168.30.30	192.168.30.15	SMTP	76 C:	EHLO [192.168.30.30]
157	34.034596	192.168.30.15	192.168.30.30	SMTP	218 S:	250-ftp-mail.labredes13.com PIPELINING SIZE 15728640 ETRN STARTTLS ENHANCEDSTATUSCODES 8BITMIME DSN SMTP-
160	34.082824	192.168.30.30	192.168.30.15	SMTP	64 C:	STARTTLS
162	34.083319	192.168.30.15	192.168.30.30	SMTP	84 S:	220 2.0.0 Ready to start TLS
214	42.283492	192.168.30.30	192.168.30.30	SMTP	97 S:	220 ftp-mail.labredes13.com ESMTP Postfix
216	42.345294	192.168.30.30	192.168.30.15	SMTP	76 C:	EHLO [192.168.30.30]
218	42.345887	192.168.30.15	192.168.30.30	SMTP	218 S:	250-ftp-mail.labredes13.com PIPELINING SIZE 15728640 ETRN STARTTLS ENHANCEDSTATUSCODES 8BITMIME DSN SMTP-
219	42.348462	192.168.30.30	192.168.30.15	SMTP	64 C:	STARTTLS
221	42.348892	192.168.30.15	192.168.30.30	SMTP	84 S:	220 2.0.0 Ready to start TLS

B) Protocolo de Transporte usado: TCP

c) Puertos usados por el correo:

- Puerto Cliente: 60956

- Puerto Servidor: 587

No.	Time	Source	Destination	Protocol	Length	Info
218	42.345887	192.168.30.30	192.168.30.15	SMTP	76 C:	EHLO [192.168.30.30]
219	42.348462	192.168.30.30	192.168.30.15	SMTP	64 C:	STARTTLS
221	42.348892	192.168.30.15	192.168.30.30	SMTP	84 S:	220 2.0.0 Ready to start TLS


```
> Frame 218: 218 bytes on wire (1744 bits), 218 bytes captured (1744 bits) on interface \Device\NPF_{EDCE8011-C36C-4F74-941D-EE6CD5B042C4}, id 0
> Ethernet II, Src: VMware_c8:9b:4f (00:0c:29:c8:9b:4f), Dst: VMware_2f:3a:88 (00:0c:29:2f:3a:88)
> Internet Protocol Version 4, Src: 192.168.30.15, Dst: 192.168.30.30
▼ Transmission Control Protocol, Src Port: 587, Dst Port: 60956, Seq: 44, Ack: 23, Len: 164
  Source Port: 587
  Destination Port: 60956
  [Stream index: 9]
  [Conversation completeness: Complete, WITH_DATA (63)]
```

4.4.6 Traza TCP de paquete ICMP

Wireshark · Follow TCP Stream (tcp.stream eq 9) · Email_sent.pcapng

```

220 ftp-mail.labredes13.com ESMTP Postfix
EHLO [192.168.30.30]
250-ftp-mail.labredes13.com
250-PIPELINING
250-SIZE 15728640
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
STARTTLS
220 2.0.0 Ready to start TLS
.....\RZ.....J.jO.PR..Y....., .U.....|.x4...|(z|=.("....Zn.,.".....+./.....,.0.
. ......./.5.....ftp-mail.labredes13.com.....
.....#. .....
.....3.k.i... .
31.d.s;G....Q... 6b...h. i...A...m....YV\....._m|I .....L..%0.6TP.....? .z.....(./...
+.....
.....@.
.....z...v...@KT...1*.v.I...=>@....."....U.....|(z|=.(
.....Zn.,....+....3.$... 7....R.U...x...)...L.. S.!....?....9..7....A.Nayy...?....iw..6... u....n..!@..M...
36.zy..../.t..i.M.....He.k...g...z&b?...b...qQW...1d....T.B...K.jq..K.
...!...].p.
.B.g...*Nj...a.9F..0.K.n....{|g..?Y.=.0~W..i...e<.....C..55...~.6.....-5P.En.].....UN....])A.R.....%.
3...\.pz}...N4..g..d.
b..D.>D...\.#.M..fN.i..o..V....>f0..~^..G&.R..#.T.b}J...=h*..;
.Pr`...
]...q.!J.)...[E.kx\!QYG...9@. !*..C.gr.=.521..HD..)\.H"...sT/.A....*;..9.H.2..3GB..n...2u)...z....../...#.1P.#..
[36\.../...E.]L.;7.~..o6...^..kEq..!./.../2P...P...[K..]W.>.m....D[...`..0F..jD..vx>...T,f.(k..$.*...c..`y..h...q...
...0.r%..
....3.sR..ry.._4* u.'9.d.0$...a/FM.V..l..F~J.{p.B..(.J...{`l.v.._....^i.b...y...EX.pJ...bsh.m).j...."....XQ.{.
.ZZ..Cz..8.6<..BG..tF.....D.....cq..^}@."....y.1.....n;..@..k.9..4..+...../.%.HX[n..n...<mT..Ci..N!s...Fh.....#.
...q.. "....e.p.W...NA|.....n..`4XQp..-Q..&.....1.T.C.._.<....{6D....L>..Z5>.i.rm.').
3..R..Le.....z1'd...../2+.cvF.....xs..@.
@...
"....#..}....m.....]..r...../...0...'\\...2cb0.....)|.....
....U.....jf....6.)Z"....O.z..P..b.G....m^...b..<...:<...2L.t..t.....Q..e.....4d8.,...!.>.... @..C..Po...r..d.B..r.D
4
.HX...*4..bAq...7....(.D.(Bm1mg...U.<...`....y..... u.....^6...+`...x.c.4....@.W....?....N2..
.R..3i.x.....#...
..1.....4..h.....
....>dv
;Dd..m...F.....0.ktV//...?....C....}....3....al.:~/..I.h.....M...$.W.b.k.D....T....'... j;..e.....'...M^.....
8BA2Q...=...q..@

```

13 client pkts, 14 server pkts, 21 turns.

Entire conversation (4868 bytes) Show data as ASCII Stream 9

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

Notas acerca del ejercicio de correo:

Logramos identificar que, en Wireshark, al filtrar por ICMP y POP, no aparecían casi paquetes, pero había muchos de tipo TCP e IMAP. Al analizar los ICMP, no pudimos encontrar muchos comandos de envío o recepción de mensajes, mientras que al mirar los POP, no había ninguno registrado. De igual manera, intentamos usar el correo via RoundCube, sin embargo, la totalidad de paquetes eran TCP y no encontramos ninguno ICMP o POP. De igual manera, no encontramos paquetes ICMP o POP que utilizaran comandos de autenticación o similares.

4.5 Análisis del Tráfico de Servicio Web

4.5.3

a) Información de la capa de aplicación:

Se puede ver un GET con la información de cada paquete y hay un get por cada elemento de la página web

```
✓ Hypertext Transfer Protocol
> GET / HTTP/1.1\r\n
Host: web.labredes13.com\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.81 Safari/537.36 Edg/104.0.1293.47\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://web.labredes13.com/]
[HTTP request 1/6]
[Response in frame: 233]
[Next request in frame: 235]
```

Resumen:

No.	Time	Source	Destination	Protocol	Length	Info
215	8.663802	192.168.30.30	192.168.30.14	HTTP	498	GET / HTTP/1.1
233	8.739218	192.168.30.14	192.168.30.30	HTTP	1440	HTTP/1.1 200 OK (text/html)
235	8.757521	192.168.30.14	192.168.30.14	HTTP	416	GET /bootstrap.min.css HTTP/1.1
236	8.756647	192.168.30.30	192.168.30.14	HTTP	445	GET /jromero.jpg HTTP/1.1
249	8.769161	192.168.30.30	192.168.30.14	HTTP	441	GET /bot.gif HTTP/1.1
252	8.769381	192.168.30.30	192.168.30.14	HTTP	447	GET /jalegria.jpeg HTTP/1.1
254	8.769388	192.168.30.30	192.168.30.14	HTTP	442	GET /rock.gif HTTP/1.1
255	8.769517	192.168.30.30	192.168.30.14	HTTP	440	GET /1.jpeg HTTP/1.1
381	8.788847	192.168.30.14	192.168.30.30	HTTP	1201	HTTP/1.1 200 OK (JPEG JFIF image)
383	8.792070	192.168.30.30	192.168.30.14	HTTP	440	GET /2.jpeg HTTP/1.1
446	8.961306	192.168.30.14	192.168.30.30	HTTP	845	HTTP/1.1 200 OK (JPEG JFIF image)
448	8.964222	192.168.30.30	192.168.30.14	HTTP	440	GET /3.jpeg HTTP/1.1
1017	9.061260	192.168.30.14	192.168.30.30	HTTP	130	HTTP/1.1 200 OK (text/css)
1059	9.064146	192.168.30.30	192.168.30.14	HTTP	440	GET /4.jpeg HTTP/1.1
1976	9.088151	192.168.30.14	192.168.30.30	HTTP	750	HTTP/1.1 200 OK (GIF89a)
2010	9.092831	192.168.30.30	192.168.30.14	HTTP	440	GET /5.jpeg HTTP/1.1
2993	9.158813	192.168.30.14	192.168.30.30	HTTP	624	HTTP/1.1 200 OK (GIF89a)
3336	9.164257	192.168.30.30	192.168.30.14	HTTP	448	GET /6.jpeg HTTP/1.1
3852	9.171456	192.168.30.14	192.168.30.30	HTTP	1449	HTTP/1.1 200 OK (JPEG JFIF image)
4134	9.181829	192.168.30.30	192.168.30.14	HTTP	412	GET /1.m4 HTTP/1.1

b) Protocolo de transporte usado: TCP

c) Puertos utilizados:

- Cliente: 64352

- Servidor: 80

```
✓ Transmission Control Protocol, Src Port: 64352, Dst Port: 80, Seq: 1, Ack: 1, Len: 444
  Source Port: 64352
  Destination Port: 80
  [Stream index: 4]
  [Conversation completeness: Complete, WITH_DATA (63)]
  [TCP Segment Len: 444]
```

4.6 Análisis del Protocolo HTTPS realizando navegación en YouTube y otros:

Análisis de los archivos .pcap

a) Información de la capa de aplicación:

```
> Frame 23309: 1294 bytes on wire (10352 bits), 1294 bytes captured (10352 bits) on interface \Device\NPF_{38600972-26D3-45B6-ADDE-4DF20FC17ED6}, id 0
> Ethernet II, Src: HefeiRad_00:00:20 (14:82:5b:00:00:20), Dst: Giga-Byt_a1:ec:5f (b4:2e:99:a1:ec:5f)
> Internet Protocol Version 6, Src: 2800:3f0:4005:1::7, Dst: 2800:484:4e83:e520:79cb:30c7:258f:9d7e
> Transmission Control Protocol, Src Port: 443, Dst Port: 63229, Seq: 9458258, Ack: 31458, Len: 1220
> [14 Reassembled TCP Segments (16406 bytes): #23295(548), #23296(1220), #23297(1220), #23298(1220), #23299(1220), #23300(1220), #23301(1220), #23302(1220),
  < Transport Layer Security
    < TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 16401
      Encrypted Application Data: 6d22a35ec9d3b77276d1f979a90404e1ca5632fd20b80317feb8a27a7c5462712e6a49c4...
      [Application Data Protocol: http-over-tls]
```

Encontramos que todos los paquetes estaban encriptados de forma segura mediante HTTPS, por lo tanto, no se puede identificar información específica de cada aplicación.

b) Protocolos de transporte utilizados: TCP y UDP

c) Puertos utilizados:

- Cliente: 52556 TCP, 63464 UDP
- Servidor: 443 TCP. 8801 UDP

```
< User Datagram Protocol, Src Port: 63464, Dst Port: 8801
  Source Port: 63464
  Destination Port: 8801
  Length: 468
  Checksum: 0x0dea [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  > [Timestamps]
    UDP payload (460 bytes)

< Transmission Control Protocol, Src Port: 52556, Dst Port: 443, Seq: 26213, Ack: 16806, Len: 233
  Source Port: 52556
  Destination Port: 443
  [Stream index: 0]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 233]
  Sequence Number: 26213      (relative sequence number)
  Sequence Number (raw): 4237841486
  [Next Sequence Number: 26446      (relative sequence number)]
  Acknowledgment Number: 16806      (relative ack number)
  Acknowledgment number (raw): 2503811414
  0101 .... = Header Length: 20 bytes (5)
```

Respuesta a preguntas de profundización de conocimientos

1. ¿Qué información le muestra en pantalla el comando ipconfig /displaydns?

R// Este comando ipconfig se encarga de mostrar los contenidos del caché de resolución de DNS, aquel encargado de convertir un nombre de dominio en una dirección IP. Este caché ayuda en el rendimiento de la velocidad de consulta de un sitio, ya que la próxima vez que se ingrese no se tiene que hacer esta conversión. Al utilizar la opción /displaydns con ipconfig se muestra lo que está actualmente guardado en la máquina.

2. Explique qué ocurre si desde un PC cliente se intenta hacer ping a la URL de uno de los servidores, pero dicho cliente tiene configurada la IP de forma estática y no le fue definida la dirección IP de servidor DNS.

R// Si un PC cliente intenta hacer ping a la URL de un servidor pero este no tiene configurada la dirección IP del servidor DNS, le es imposible resolver la IP vinculada a la URL, por lo que posiblemente se obtendría un código de error similar a este “DNS_PROBE_FINISHED_NXDOMAIN”. Incluso, algunos sistemas operativos como Windows no permiten configurar la IP estática sin antes definir la IP del DNS.

3. Es posible ver durante la autenticación en la captura de tráfico de Wireshark, el nombre de usuario y contraseña de un usuario en el servidor FTP, ¿A qué se debe? Encuentre evidencia que sustenta el enunciado.

R// Sí, se puede ver el usuario y contraseña debido a que se está usando un protocolo de FTP no asegurado entonces ese contenido se manda en texto plano sin ningún tipo de encriptación que evite a terceros visualizar esta información. A continuación, se puede

visualizar un ejemplo de captura de usuario y contraseña en el protocolo FTP.

The screenshot shows a NetworkMiner capture of an FTP session. The traffic is between two hosts: 192.168.30.15 and 192.168.30.30. The session includes the following sequence of events:

- Frame 28: AUTH SSL
- Frame 29: Response: 500 AUTH not understood
- Frame 30: Request: AUTH SSL
- Frame 31: Response: 500 AUTH not understood
- Frame 32: Request: USER usuario1
- Frame 33: Response: 500 USER not understood
- Frame 34: Request: PASS usuario1
- Frame 35: Response: 331 Password required for usuario1
- Frame 36: Request: File Transfer Protocol (FTP)
- Frame 37: Response: 230 User usuario1 logged in
- Frame 38: Request: SYST
- Frame 39: Response: 215 UNIX Type: L8

Below the frames, the raw hex and ASCII data is shown for frame 36:

```
0000  00 0c 29 00 ff 7c 00 0c 29 2f 3a 88 08 00 45 00  .)...) /:...E.
0010  00 37 b7 6f 40 00 80 06 00 00 c0 a8 1e 1e c0 a8  .7@.....I-F-P.
0020  1e 0f ec a2 00 15 04 c2 9c fb 49 99 46 a7 50 18  .....US ER usar
0030  04 02 bd a7 00 00 55 53 45 52 20 75 73 75 61 72  io1...
0040  69 6f 31 0d 0a
```

4. ¿Identifica tráfico HTTP y HTTPS generado al ingresar a los sitios web? ¿El tráfico

HTTP es significativo frente al tráfico HTTPS? ¿Identifica que componentes o información de su navegación en el portal web generó este tráfico HTTP?

R// Sí se identifica ese tipo de tráfico, pero HTTPS es más significativo frente al HTTP en cuanto al volumen de tráfico. Los componentes que fueron identificados en el uso del HTTP fueron peticiones GET para la obtención de contenido como imágenes, información del DNS, redirecciones de IP, y otros recursos sin cifrar.

5. Para el protocolo HTTPS: Explique gráficamente y con sus palabras el proceso de handshake. Encontrar e inspeccionar los detalles del intercambio de certificados, incluyendo la expansión del bloque de protocolo de enlace dentro de la TLS Record. Al igual que los mensajes “Hello”, el contenido del mensaje de certificado es visible, ¿A qué se debe esto?

R// El proceso de three way handshaking consiste en un intercambio de mensajes entre un cliente y un servidor para establecer una conexión entre ellos. En el caso del HTTPS se puede usar SSL o TLS, que consiste en:

- El cliente le envía un mensaje al servidor que contiene un HELLO para iniciar el handshaking. Este mensaje incluye la versión de TLS/SSL, el tipo de cifrado soportado, y un string aleatorio de bytes conocido como el client random.
- Cuando el servidor recibe el HELLO del cliente, este le manda una respuesta contenido su certificado SSL, el tipo de cifrado elegido por el servidor, y otro string aleatorio de bytes conocido como el server random.
- Posteriormente, cuando el cliente recibe el certificado del servidor este procede a comprobar su validez con la autoridad de certificados que lo expidió. En este paso, el cliente confirma si el servidor es quién dice que es y si está interactuando con el verdadero propietario del dominio.
- Después de confirmar la identidad del servidor, el cliente envía otro string aleatorio de bytes (llamado premaster secret) el cual está encriptado con la llave pública del servidor (el cliente obtiene la llave pública del certificado SSL).
- Una vez que el servidor recibe el premaster secret, este lo desencripta con su llave privada. Luego, en este mismo paso tanto cliente como servidor crean unas llaves de sesión apartir del client random, server random, y premaster secret. Estas llaves deberían generar los mismos resultados.
- Cuando las llaves de sesión están creadas, el cliente le manda al servidor un mensaje de que la conexión está lista, el cual está encriptado con la llave de sesión anterior. Cuando el servidor recibe este mensaje, lo desencripta y confirma la sesión.
- En este paso, la encriptación simétrica está asegurada, el handshake completado y la conexión establecida. Después de esto, el resto de la comunicación sigue siendo cifrada usando las llaves de sesión.

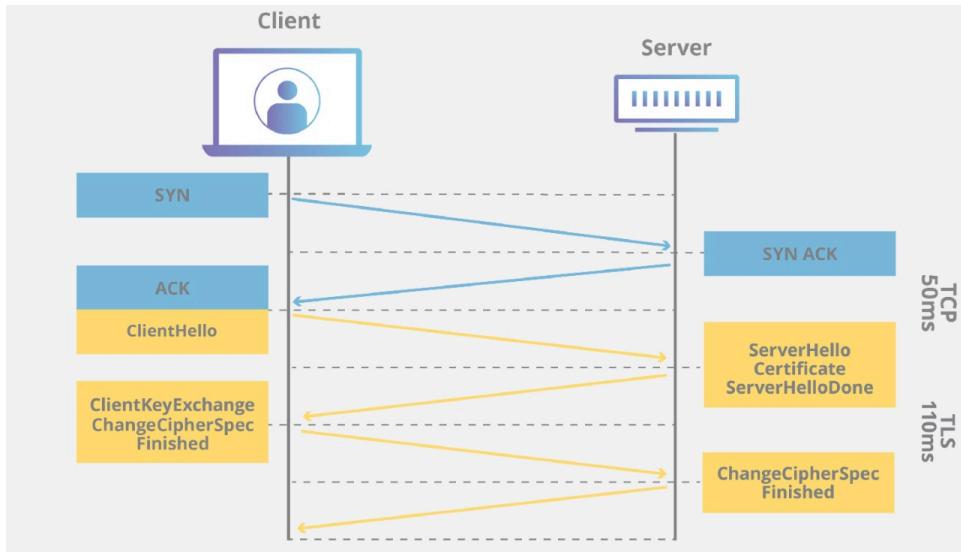


Figura 1. Proceso del Three Hand Shaking

6. ¿Quién envía el certificado, el cliente, el servidor, o ambos? Un certificado es enviado por una de las partes para que la otra parte autentique que es quien dice ser. Sobre la base de este uso, usted debería ser capaz de reconocer con Wireshark quién envía el certificado y comprobar los mensajes en su rastro.

R// El certificado es enviado por el servidor como se puede apreciar en la respuesta a la pregunta anterior y en la captura de tráfico realizada con Wireshark.

7. ¿Puede encontrarse información adicional sobre los servicios prestados en la topología usando la captura de paquetes? Por ejemplo, ¿sería posible encontrar la ubicación de los recursos de contenido? Justifique su respuesta.

R// Sí, es posible encontrar la ubicación de los recursos analizando la IP origen de los paquetes que recibe el host. Principalmente en los paquetes enviados por el DNS, ya que así identificamos a las fuentes de los recursos de contenido.

Enlace de descarga a capturas de tráfico realizadas durante la práctica

- [https://drive.google.com/drive/folders/1zuLUE2fT-yIEImkvU3dFV2k5rR5OMGZs?usp=sharing](https://drive.google.com/drive/folders/1zuLUE2fTyIEImkvU3dFV2k5rR5OMGZs?usp=sharing)

Listado de referencias

- <https://superuser.com/questions/230308/explain-output-of-ipconfig-displaydns>
- <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/>