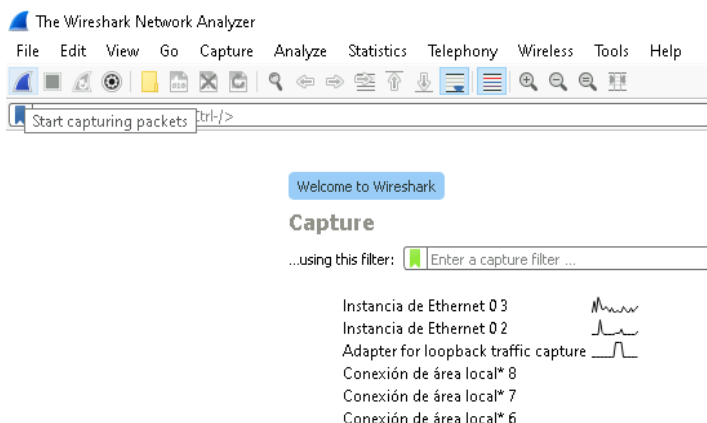
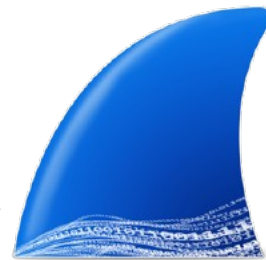


Wireshark

És un analitzador de paquets i protocols utilitzat per a realitzar anàlisi i solucionar problemes en xarxes de comunicacions.

Permet capturar els paquets de la xarxa en mode promiscu, pot monitorar les dades del mateix domini (per Ethernet IEEE 802.11).
Seleccióem una de les interfícies detectades i iniciem la captura de la següent manera:



Què mostren les columnes?

No.	Time	Source	Destination	Protocol	Length	Info
49	1.622884	10.2.0.1	10.2.222.4	TLSv1.2	93	Application Data
50	1.623363	10.2.0.1	10.2.222.4	TCP	60	49470 → 3389 [ACK] Seq=405 Ack=26371 Win=4066 Len=0
51	1.625293	10.2.222.4	10.2.0.1	TLSv1.2	91	Application Data

ID segons l'ordre d'entrada del paquet.

Segons transcorreguts des de l'inici de la monitorització de la xarxa.

Origen del paquet (IP, MAC o hostname).

Destí del paquet.

Protocol del paquet (TCP, UDP, NDS, etc).

Longitud del paquet en bytes.

Informació addicional sobre el paquet.

Filtratge

Tenim una finestra per filtrar els paquets segons les seves característiques, per exemple:

Per protocol UDP:

Per port TCP 3389:

Per IP d'origen 10.2.222.4:

Podem encadenar filtres usant &&:

Capes

A sota tenim la informació del paquet seleccionat.

Per exemple podem veure el port destí.

La informació es mostra desgranada:

```
Total Length: 87
Identification: 0x16d4 (5844)
> 010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.2.222.4
Destination Address: 10.2.0.1
▼ Transmission Control Protocol, Src Port: 3389, Dst Port: 49470, Seq: 4388, Ack: 125, Len: 47
  Source Port: 3389
  Destination Port: 49470
```

I al costat el paquet complet, convertit en hexadecimal (en realitat s'envien en binari) i ASCII (forma llegible):

0000	b6 0e 14 c8 e7 de 52 54	00 23 43 d2 08 00 45 00RT.#C...E-
0010	00 57 16 d4 40 00 80 06	00 00 0a 02 de 04 0a 02	..W.@...
0020	00 01 0d 3d c1 3e 3e e6	e1 33 b9 14 fa 93 50 18	...=>..3...P.
0030	f6 13 f2 52 00 00 17 03	03 00 2a 00 00 00 00 00	...R.... .*.....
0040	00 01 99 96 d0 18 2b cc	04 71 2b a6 c3 0a 3d 17+..q+.....
0050	54 20 d5 8c 68 66 2b 7e	f5 40 ca 6d f4 c6 cc d1	T..hf+~..@.m....
0060	b3 6f 03 e5 22		.o.."

$$49470_{10} = C13E_{16}$$

Estan ordenats per capes segons un model híbrid entre l'OSI i el TCP/IP:

Capa física: Aquesta capa fa referència als components físics de la xarxa, com ara els cables i els dispositius de connexió. Wireshark mostra la informació d'aquesta capa, com ara l'adreça MAC i els detalls de l'enllaç.

Capa d'enllaç de dades: Aquesta capa fa referència als protocols que controlen la transmissió de dades entre dispositius de xarxa propers. Wireshark mostra la informació d'aquesta capa, com ara el protocol Ethernet, ARP, PPP, etc.

Capa de xarxa: Aquesta capa fa referència als protocols que controlen l'encaminament i l'adreça dels paquets de dades a través de la xarxa. Wireshark mostra la informació d'aquesta capa, com ara el protocol IP, ICMP, IGMP, etc.

Capa de transport: Aquesta capa fa referència als protocols que controlen el flux de dades entre dispositius de xarxa. Wireshark mostra la informació d'aquesta capa, com ara el protocol TCP, UDP, etc.

Capa de sessió: Aquesta capa fa referència als protocols que controlen la sessió entre dispositius de xarxa. Wireshark mostra la informació d'aquesta capa, com ara el protocol SOCKS.

Capa de presentació: Aquesta capa fa referència als protocols que controlen la representació de les dades. Wireshark mostra la informació d'aquesta capa, com ara el protocol SSL/TLS.

Capa d'aplicació: Aquesta capa fa referència als protocols que controlen la interacció entre les aplicacions i la xarxa. Wireshark mostra la informació d'aquesta capa, com ara els protocols HTTP, FTP, SMTP, etc.

La representació més habitual és un model híbrid OSI i TCP/IP amb les 5 primeres capes:

```
> Frame 17: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface \Device\NPF_{FE44126C-7118-4588-899B-6D0C23AB524C}, id 0
> Ethernet II, Src: RealtekU_23:43:d2 (52:54:00:23:43:d2), Dst: b6:0e:14:c8:e7:de (b6:0e:14:c8:e7:de)
> Internet Protocol Version 4, Src: 10.2.222.4, Dst: 10.2.0.1
> Transmission Control Protocol, Src Port: 3389, Dst Port: 49470, Seq: 4388, Ack: 125, Len: 47
> Transport Layer Security
```

La majoria de vegades no es mostren totes les capes per cada paquet, a vegades però se'n mostren d'extres... :

```
> Frame 529: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) on interface \Device\NPF_{FE44126C-7118-4588-899B-6D0C23AB524C}, id 0
> Ethernet II, Src: RealtekU_23:43:d2 (52:54:00:23:43:d2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.2.222.4, Dst: 10.2.255.255
> User Datagram Protocol, Src Port: 138, Dst Port: 138
> NetBIOS Datagram Service
> SMB (Server Message Block Protocol)
> SMB MailSlot Protocol
> Microsoft Windows Browser Protocol
```

ACTIVITAT

1. Inicialitza una captura de paquets.

Analitza un dels paquets.

Elabora un document amb captures de pantalla sobre els detalls més significatius del paquet.

Guarda la captura de paquets i envia-la juntament amb el document.

Podem fer un filtratge per algun dels camp si seleccionem i fem:

Apply As Filter → Selected

1298 28.572334 16 Expand Subtrees .2 116 Application Data
1299 28.584643 16 Collapse Subtrees .2 93 Application Data
1300 28.584711 16 Expand All .2 54 3389 → 49470 [ACK]
1301 28.584773 16 Collapse All .2 93 Application Data
1302 28.608802 16 .2 93 Application Data
1303 28.608894 16 54 3389 → 49470 [ACK]
1304 28.627782 16 .2 121 Application Data
1305 28.630976 16
1306 28.631743 16 Apply as Filter: ip.proto == 6
1307 28.631829 16 Apply as Filter
1308 28.632507 16 Prepare as Filter
1309 28.632598 16 Conversation Filter
Colorize with Filter
Follow
Copy
Show Packet Bytes... Ctrl+Maj+O
Export Packet Bytes... Ctrl+Maj+X
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As... Ctrl+Maj+U
Go to Linked Packet
Show Linked Packet in New Window

Frame 1309: 60 bytes on wire (480 bits) captured (60 bytes) on interface 0
Ethernet II, Src: b6:0e:14:c8:00:00, Dst: b6:0e:14:c8:00:00
> Destination: Realtek
> Source: b6:0e:14:c8:00:00
Type: IPv4 (0x0800)
Padding: 0000000000
Internet Protocol Version 4, Src: 10.2.0.1, Dst: 10.2.222.4
0100 = Version: 4
.... 0101 = Header Length: 20 (bytes)
> Differentiated Services Code Point: 0
Total Length: 40
Identification: 0x2000
> 010. = Flags: 0x0000
...0 0000 0000 0000
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x26bf [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.2.0.1
Destination Address: 10.2.222.4

Aquesta instrucció genera el següent filtre:

ip.proto == 6

Podem usar “contains” per cercar strings dins les dades del paquet, per exemple:

tcp contains "NETWORK"

No.	Time	Source	Destination	Protocol	Length	Info
10	0.267724	98.114.205.102	192.150.11.111	SMB	191	Negotiate Protocol Request

> NetBIOS Session Service
v SMB (Server Message Block Protocol)
> SMB Header
v Negotiate Protocol Request (0x72)
Word Count (WCT): 0
Byte Count (BCC): 98
v Requested Dialects
v Dialect: PC NETWORK PROGRAM 1.0
Buffer Format: Dialect (2)
Name: PC NETWORK PROGRAM 1.0

0030 fa f0 7c d5 00 00 00 00 00 85 ff 53 4d 42 72 00 ..|.....SMB
0040 00 00 00 18 53 c8 00 00 00 00 00 00 00 00 00 00S.....
0050 00 00 00 00 ff fe 00 00 00 00 00 62 00 02 50 43b..PC
0060 20 4e 45 54 57 4f 52 4b 20 50 52 4f 47 52 41 4d NETWORK PROGRAM
0070 20 31 2e 30 00 02 4c 41 4e 4d 41 4e 31 2e 30 00 1.0..LANMAN1.0
0080 02 57 69 6e 64 6f 77 73 20 66 6f 72 20 57 6f 72 .Windows for Wor

Seqüència de paquets per enviar missatges

Cada paquet correspon a una seqüència de paquets que completen el missatge.

Amb ipconfig podem veure la nostra configuració de xarxa:

```
Adaptador de Ethernet Instancia de Ethernet 0 3:

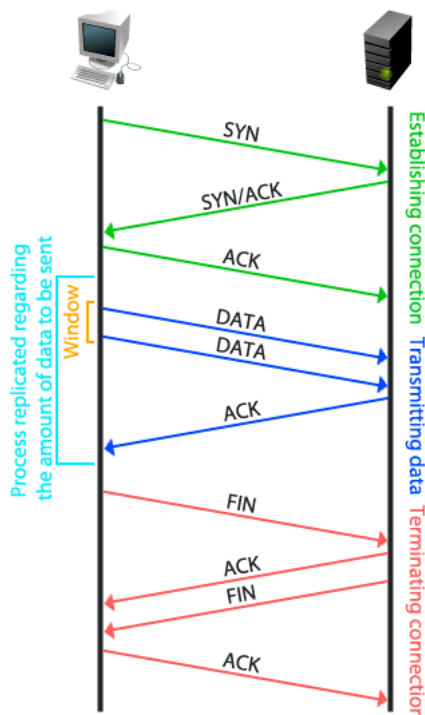
Suíjo DNS específico para la conexión. . . :
Dirección IPv4. . . . . : 10.2.222.4
Máscara de subred . . . . . : 255.255.0.0
Puerta de enlace predeterminada . . . . . :
```

I podem cercar la nostra IP entre tots els paquets capturats:

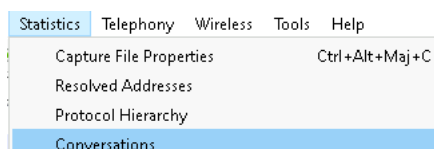
995	30.792013	10.2.222.4	10.2.0.1	TLSv1.2	107	Application Data	
996	30.792918	10.2.0.1	10.2.222.4	TCP	60	36942 → 3389 [ACK] Seq=7630 Ack=348421 Win=24567 Len=0	
997	30.792918	10.2.0.1	10.2.222.4	TCP	60	36942 → 3389 [ACK] Seq=7630 Ack=349881 Win=24558 Len=0	

En aquesta captura es pot veure una petició HTTPS a través del protocol segur TLS/SSL. I a continuació la confirmació ACK a través de TCP.

Aquesta és una representació esquematitzada d'una sessió TCP entre un client i un servidor:



Podem obtenir IP, MAC, port, etc. dels dispositius que han interactuat usant:

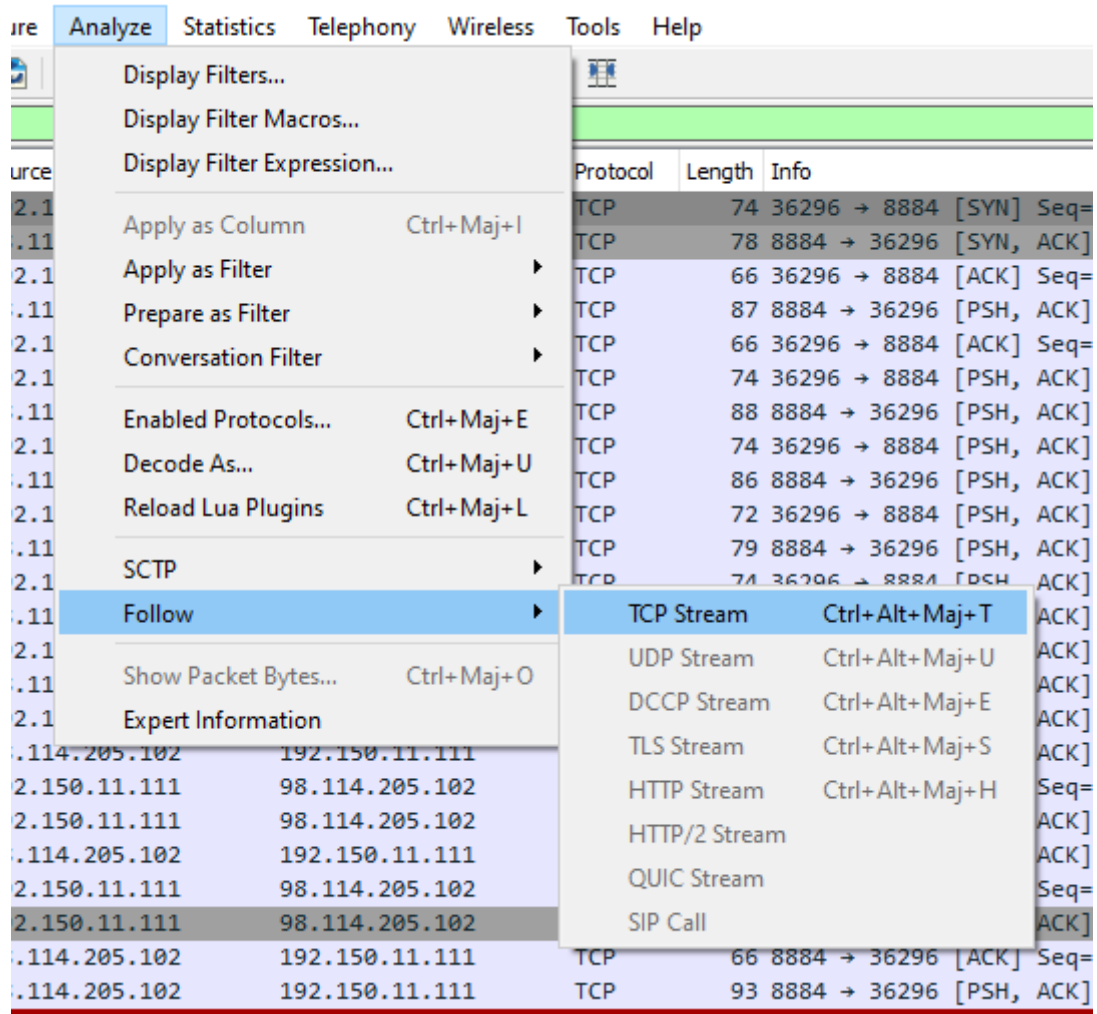


Obtenim el següent resultat:

Ethernet · 2		IPv4 · 2		IPv6	TCP · 1		UDP · 1					
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A	
10.2.222.4	10.2.0.1	1,313	446,044 KiB	628	394,603 KiB	685	51,441 KiB	0.0000000	28.7647	109,746 KiB	14,307 KiB	
10.2.222.4	10.2.255.255	1	243 bytes	1	243 bytes	0	0 bytes	9.543834	0.0000			

Anàlisi flux TCP

Els fluxos s'inicien amb paquets de sincronització o SYN.
I finalitzen amb un FIN i les confirmacions.

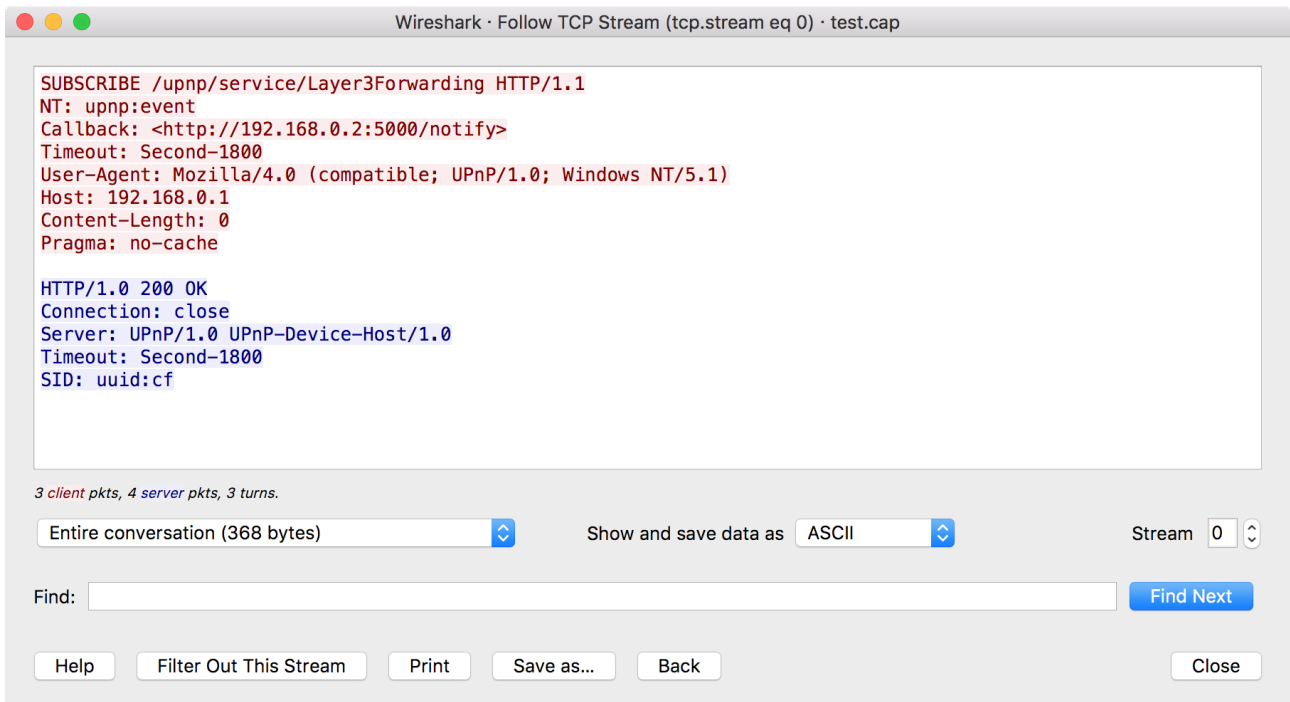


Aquesta instrucció genera el següent filtre:

`tcp.stream eq 0`

Mostrat en format YAML (un format llegible) o Hex Dump per veure un volcat en hexadecimal.

El contingut del flux es mostra en la mateixa seqüència que apareixia a la xarxa. Els caràcters no imprimibles se substitueixen per punts. El trànsit del client al servidor és de color vermell, mentre que el trànsit del servidor al client és de color blau.



Fent “save as” podem obtenir el contingut enviat.

ACTIVITAT

2. Inicialitza varies captures de paquets realitzant una acció en concret, escull les que més t'interessin:
 - Genera una petició DHCP al connectar-te a una xarxa i rebre una IP automàtica pel teu dispositiu.
 - Genera una petició DNS al convertir-se el nom de domini introduït en una direcció IP.
 - Genera una petició HTTP obrint una pàgina web.
 - Genera una petició FTP transferint arxius entre dispositius client-servidor.
 - Genera una petició SMTP enviant un correu electrònic.
 - Generant una petició ARP fent un ping a un dispositiu no registrat a la taula.

Analitza una seqüència de paquets.

Elabora un document amb captures de pantalla sobre els detalls més significatius de la seqüència de paquets.

Guarda la captura de paquets i envia-la juntament amb el document.

BIBLIOGRAFIA I WEBGRAFIA

«tosch production». <https://toschprod.wordpress.com/>

«VARONIS». <https://www.varonis.com/blog/how-to-use-wireshark>



Autor: Xavier Baubés Parramon
Aquest document es llicència sota Creative Commons versió 4.0.
Es permet compartir i adaptar el material però reconeixent-ne l'autor original.