YDAYS

WIRESHARK

INTRODUCTION À WIRESHARK

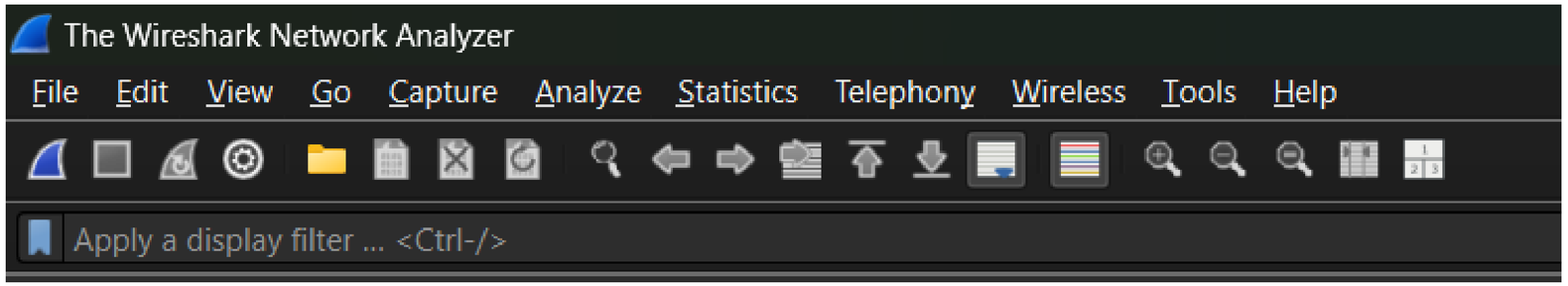
Qu'est-ce que Wireshark ?

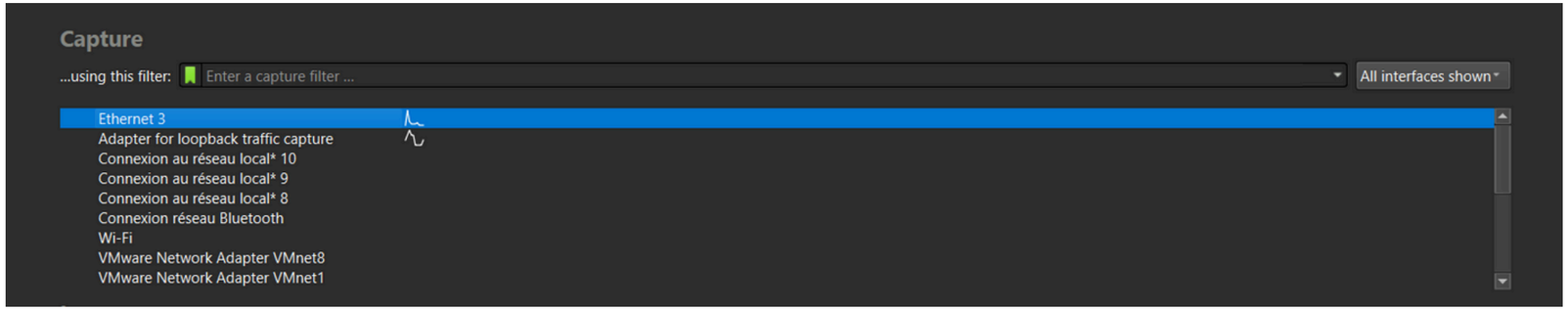
- Wireshark est un analyseur de paquets réseau gratuit et open source.
- Il permet de capturer et d'analyser le trafic réseau en temps réel.
- Wireshark aide à comprendre le comportement des protocoles réseau et est couramment utilisé pour dépanner les problèmes de réseau et étudier la sécurité.
- Il peut être utilisé sur divers systèmes d'exploitation (Linux, macOS, Windows).

INTRODUCTION À WIRESHARK

Interface Wireshark

Le premier écran qui s'affiche à l'ouverture de Wireshark est la page principale qui nous permettra de spécifier notre/nos interface(s) ainsi que d'appliquer des filtres pour affiner le trafic que nous capturons.





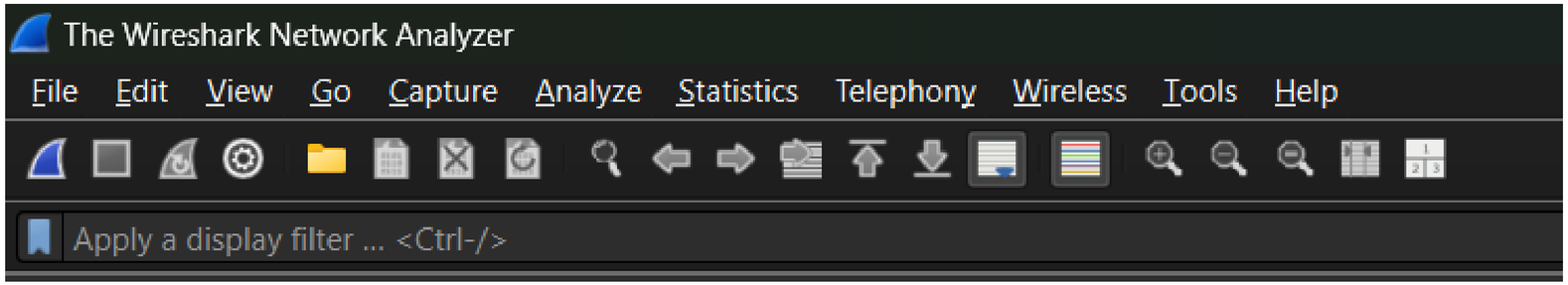
INTRODUCTION À WIRESHARK

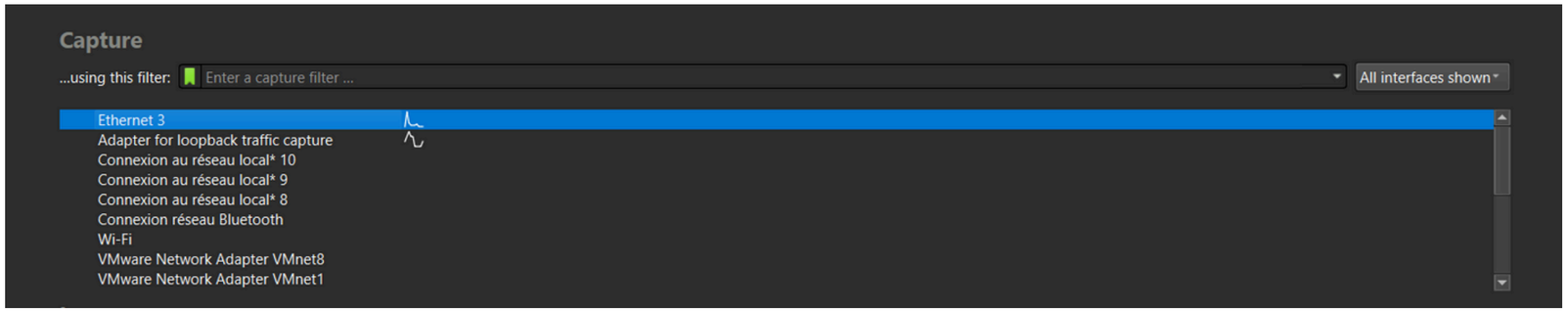
Interface Wireshark

Le premier écran qui s'affiche à l'ouverture de Wireshark est la page principale qui nous permettra de spécifier notre/nos interface(s) ainsi que d'appliquer des filtres pour affiner le trafic que nous capturons.

2 options possibles:

- Lancer la capture sur l'interface de notre choix
- Charger un PCAP pour l'analyser



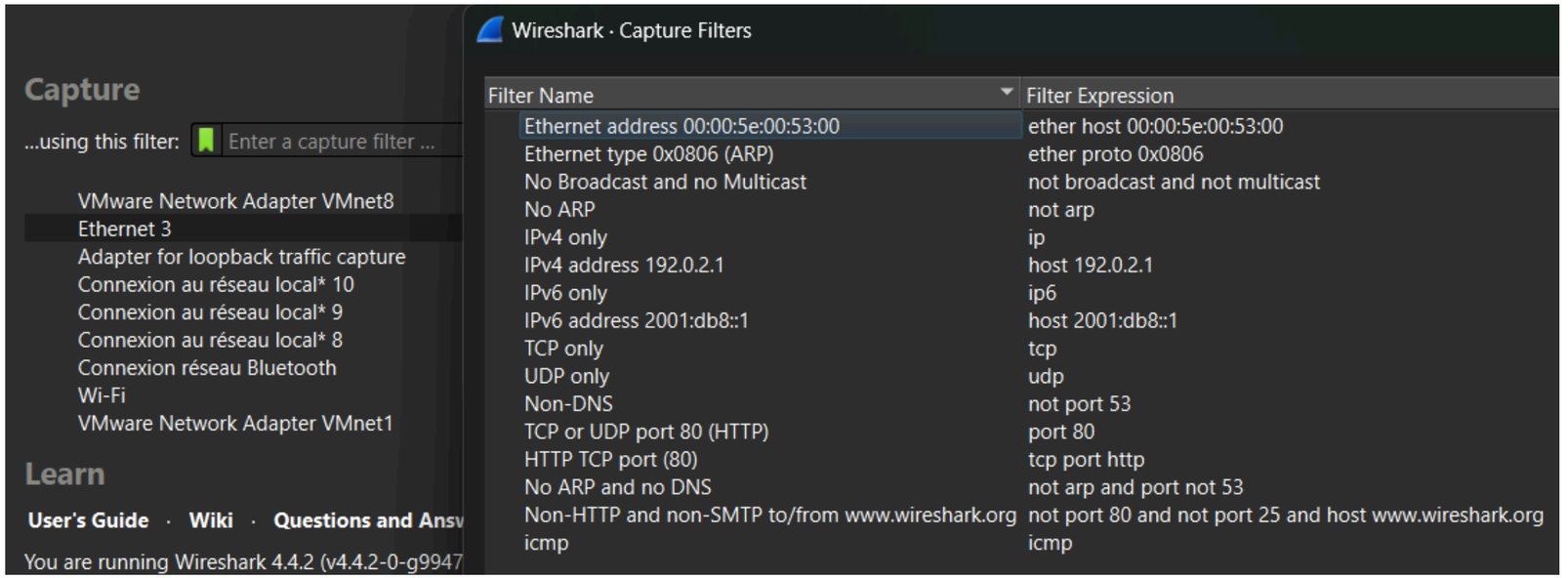


INTRODUCTION À WIRESHARK

Interface Wireshark

Si nous commençons par naviguer vers le ruban vert dans Wireshark et sélectionnons Gérer les filtres de capture, nous pouvons afficher une liste des filtres disponibles.

L’ajout d’un filtre au départ permet de réduire le nombre de paquet afin de ne garder uniquement ceux qui nous intéressent



INTRODUCTION À WIRESHARK

Interface Wireshark

Nous pouvons voir une capture avec et sans filtre

No.	Time	Source	Destination	Protocol	Length	Info
→ 1	0.000000	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 2)
← 2	0.000586	192.168.125.254	192.168.125.19	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=64 (request in 1)
3	1.014449	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 4)
4	1.014982	192.168.125.254	192.168.125.19	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=64 (request in 3)
5	2.027735	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 6)
6	2.028202	192.168.125.254	192.168.125.19	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=64 (request in 5)
7	3.033972	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 8)
8	3.034447	192.168.125.254	192.168.125.19	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64 (request in 7)

214	20.017282	2606:4700::6810:6770	2a01:e0a:5cd:bb0:a4…	TCP	74	443 → 51337 [ACK] Seq=1 Ack=43 Win=12 Len=0
215	20.099801	2606:4700::6810:6770	2a01:e0a:5cd:bb0:a4…	TLSv1.2	112	Application Data
216	20.100456	2a01:e0a:5cd:bb0:a4…	2606:4700::6810:6770	TLSv1.2	116	Application Data
217	20.102973	2606:4700::6810:6770	2a01:e0a:5cd:bb0:a4…	TLSv1.2	112	Application Data
218	20.148780	2a01:e0a:5cd:bb0:a4…	2606:4700::6810:6770	TCP	74	51337 → 443 [ACK] Seq=85 Ack=77 Win=1022 Len=0
219	20.175205	2606:4700::6810:6770	2a01:e0a:5cd:bb0:a4…	TCP	74	443 → 51337 [ACK] Seq=77 Ack=85 Win=12 Len=0
220	20.629393	192.168.125.19	35.214.136.108	TCP	55	55358 → 443 [ACK] Seq=1 Ack=1 Win=1023 Len=1
221	20.685755	35.214.136.108	192.168.125.19	TCP	66	443 → 55358 [ACK] Seq=1 Ack=2 Win=1041 Len=0 SLE=1 SRE=2
222	20.831843	192.168.125.19	104.16.102.112	TLSv1.2	96	Application Data
223	20.863290	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 224)
224	20.864135	192.168.125.254	192.168.125.19	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=64 (request in 223)
225	20.885819	104.16.102.112	192.168.125.19	TCP	60	443 → 51396 [ACK] Seq=1 Ack=43 Win=150 Len=0
226	20.920215	104.16.102.112	192.168.125.19	TLSv1.2	92	Application Data
227	20.920956	192.168.125.19	104.16.102.112	TLSv1.2	96	Application Data
228	20.934976	104.16.102.112	192.168.125.19	TCP	60	443 → 51396 [ACK] Seq=39 Ack=85 Win=150 Len=0

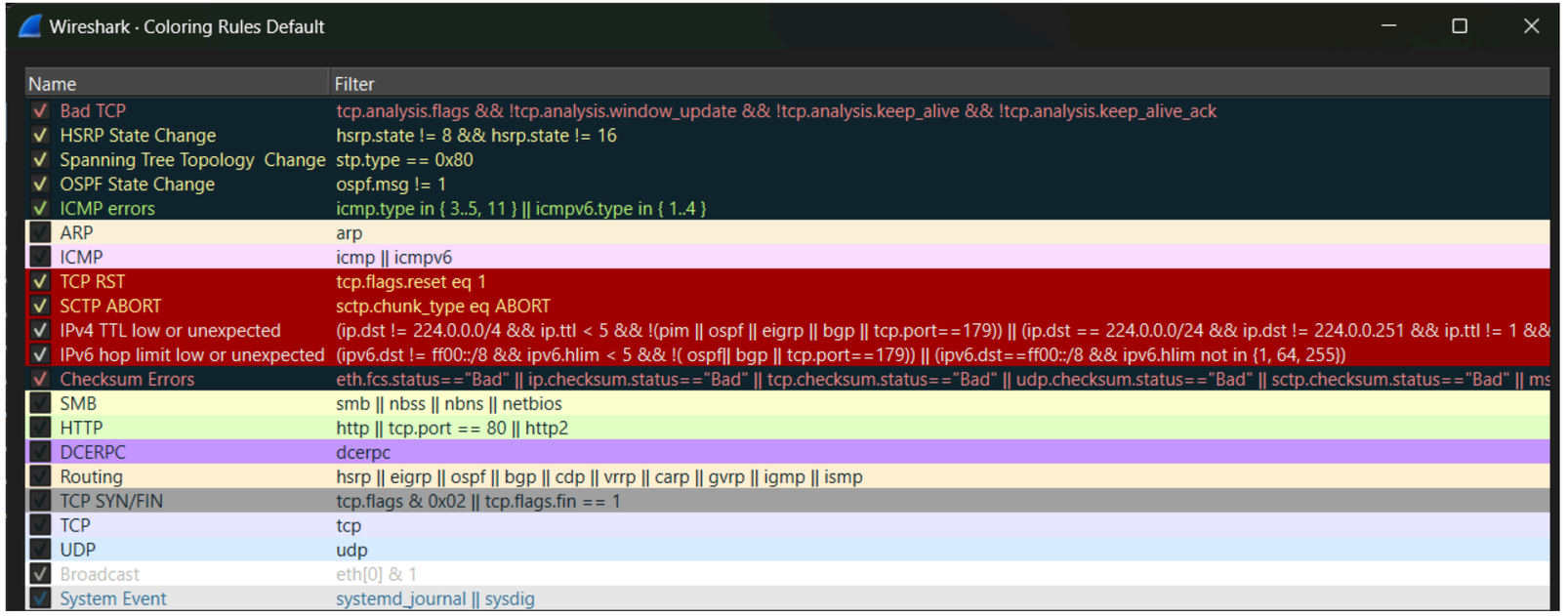
Différentes informations:

- Numéro de paquet
- Temps
- Source
- Destination
- Protocole
- Longueur
- Informations sur le paquet

INTRODUCTION À WIRESHARK

Interface Wireshark

En plus des informations rapides sur les paquets, Wireshark code également les paquets par couleur en fonction du niveau de danger et du protocole afin de pouvoir repérer rapidement les anomalies et les protocoles dans les captures.



FILTRAGE DES CAPTURES

Opérateurs de filtrage

La syntaxe des filtres de Wireshark est simple à comprendre, ce qui permet de la maîtriser rapidement. Pour tirer le meilleur parti de ces filtres, vous devez avoir une compréhension de base des opérateurs booléens et logiques.

Wireshark n'en possède que quelques-uns que vous devrez connaître :

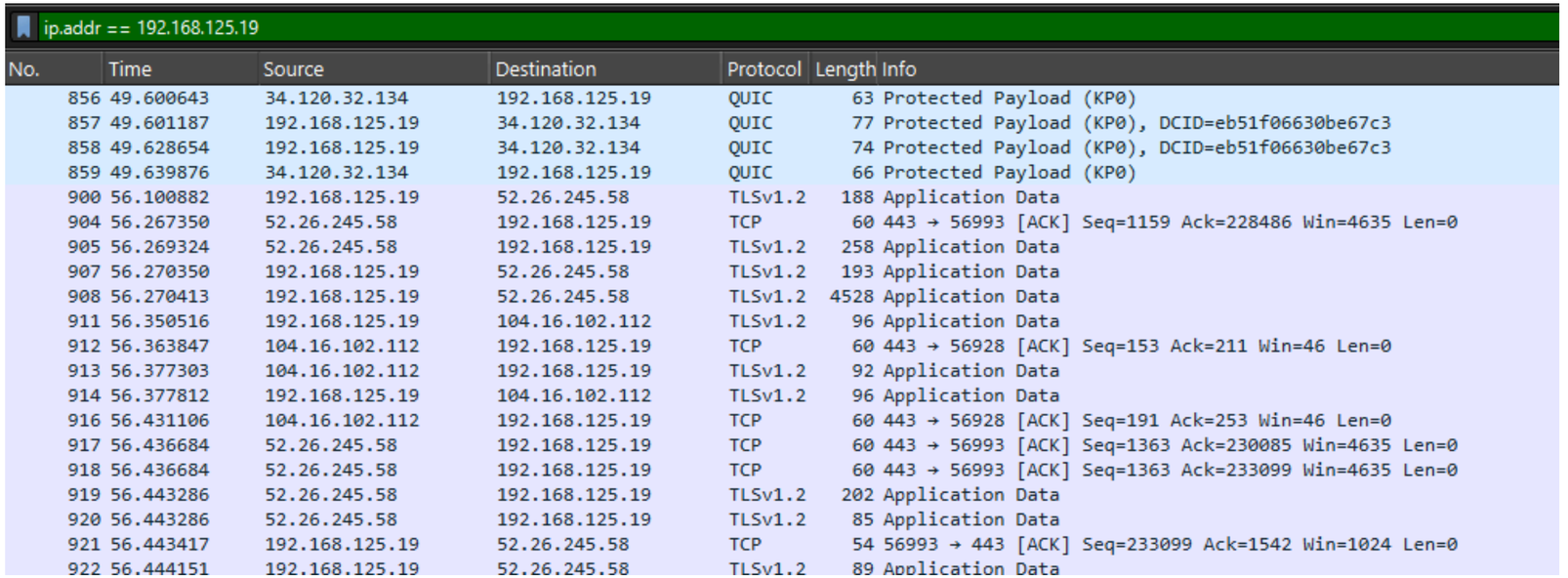
- **and - operator: and / &&**
- **or - operator: or / ||**
- **equals - operator: eq / ==**
- **not equal - operator: ne / !=**
- **greater than - operator: gt / >**
- **less than - operator: lt / <**

FILTRAGES DE BASE

Quelques exemples de filtrage

Filtrage par IP : Le premier filtre que nous allons examiner est ip.addr, ce filtre vous permettra de parcourir le trafic et de voir uniquement les paquets avec une adresse IP spécifique contenus dans ces paquets, qu'ils proviennent de la source ou de la destination.

Filtre utilisé:
ip.addr == <IP Address>



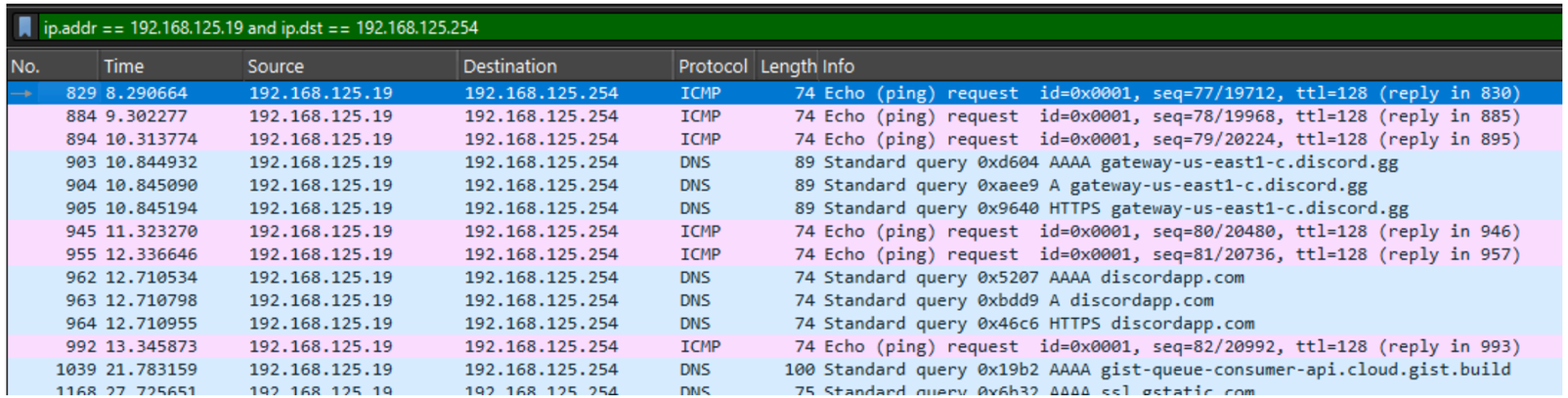
No.	Time	Source	Destination	Protocol	Length	Info
856	49.600643	34.120.32.134	192.168.125.19	QUIC	63	Protected Payload (KP0)
857	49.601187	192.168.125.19	34.120.32.134	QUIC	77	Protected Payload (KP0), DCID=eb51f06630be67c3
858	49.628654	192.168.125.19	34.120.32.134	QUIC	74	Protected Payload (KP0), DCID=eb51f06630be67c3
859	49.639876	34.120.32.134	192.168.125.19	QUIC	66	Protected Payload (KP0)
900	56.100882	192.168.125.19	52.26.245.58	TLSv1.2	188	Application Data
904	56.267350	52.26.245.58	192.168.125.19	TCP	60	443 → 56993 [ACK] Seq=1159 Ack=228486 Win=4635 Len=0
905	56.269324	52.26.245.58	192.168.125.19	TLSv1.2	258	Application Data
907	56.270350	192.168.125.19	52.26.245.58	TLSv1.2	193	Application Data
908	56.270413	192.168.125.19	52.26.245.58	TLSv1.2	4528	Application Data
911	56.350516	192.168.125.19	104.16.102.112	TLSv1.2	96	Application Data
912	56.363847	104.16.102.112	192.168.125.19	TCP	60	443 → 56928 [ACK] Seq=153 Ack=211 Win=46 Len=0
913	56.377303	104.16.102.112	192.168.125.19	TLSv1.2	92	Application Data
914	56.377812	192.168.125.19	104.16.102.112	TLSv1.2	96	Application Data
916	56.431106	104.16.102.112	192.168.125.19	TCP	60	443 → 56928 [ACK] Seq=191 Ack=253 Win=46 Len=0
917	56.436684	52.26.245.58	192.168.125.19	TCP	60	443 → 56993 [ACK] Seq=1363 Ack=230085 Win=4635 Len=0
918	56.436684	52.26.245.58	192.168.125.19	TCP	60	443 → 56993 [ACK] Seq=1363 Ack=233099 Win=4635 Len=0
919	56.443286	52.26.245.58	192.168.125.19	TLSv1.2	202	Application Data
920	56.443286	52.26.245.58	192.168.125.19	TLSv1.2	85	Application Data
921	56.443417	192.168.125.19	52.26.245.58	TCP	54	56993 → 443 [ACK] Seq=233099 Ack=1542 Win=1024 Len=0
922	56.444151	192.168.125.19	52.26.245.58	TLSv1.2	89	Application Data

FILTRAGES DE BASE

Quelques exemples de filtrage

Filtrage par SRC et DST : Le deuxième filtre examinera deux en un ainsi qu'un opérateur de filtre : ip.src et ip.dst. Ces filtres nous permettent de filtrer le trafic en fonction de la source et de la destination d'où provient le trafic.

Filtre utilisé:
ip.src == <SRC IP Address> and ip.dst == <DST IP Address>



No.	Time	Source	Destination	Protocol	Length	Info
829	8.290664	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=77/19712, ttl=128 (reply in 830)
884	9.302277	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=78/19968, ttl=128 (reply in 885)
894	10.313774	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=79/20224, ttl=128 (reply in 895)
903	10.844932	192.168.125.19	192.168.125.254	DNS	89	Standard query 0xd604 AAAA gateway-us-east1-c.discord.gg
904	10.845090	192.168.125.19	192.168.125.254	DNS	89	Standard query 0xaee9 A gateway-us-east1-c.discord.gg
905	10.845194	192.168.125.19	192.168.125.254	DNS	89	Standard query 0x9640 HTTPS gateway-us-east1-c.discord.gg
945	11.323270	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=80/20480, ttl=128 (reply in 946)
955	12.336646	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=81/20736, ttl=128 (reply in 957)
962	12.710534	192.168.125.19	192.168.125.254	DNS	74	Standard query 0x5207 AAAA discordapp.com
963	12.710798	192.168.125.19	192.168.125.254	DNS	74	Standard query 0xbdd9 A discordapp.com
964	12.710955	192.168.125.19	192.168.125.254	DNS	74	Standard query 0x46c6 HTTPS discordapp.com
992	13.345873	192.168.125.19	192.168.125.254	ICMP	74	Echo (ping) request id=0x0001, seq=82/20992, ttl=128 (reply in 993)
1039	21.783159	192.168.125.19	192.168.125.254	DNS	100	Standard query 0x19b2 AAAA gist-queue-consumer-api.cloud.gist.build
1168	27.725651	192.168.125.19	192.168.125.254	DNS	75	Standard query 0x6b32 AAAA ssl.gstatic.com

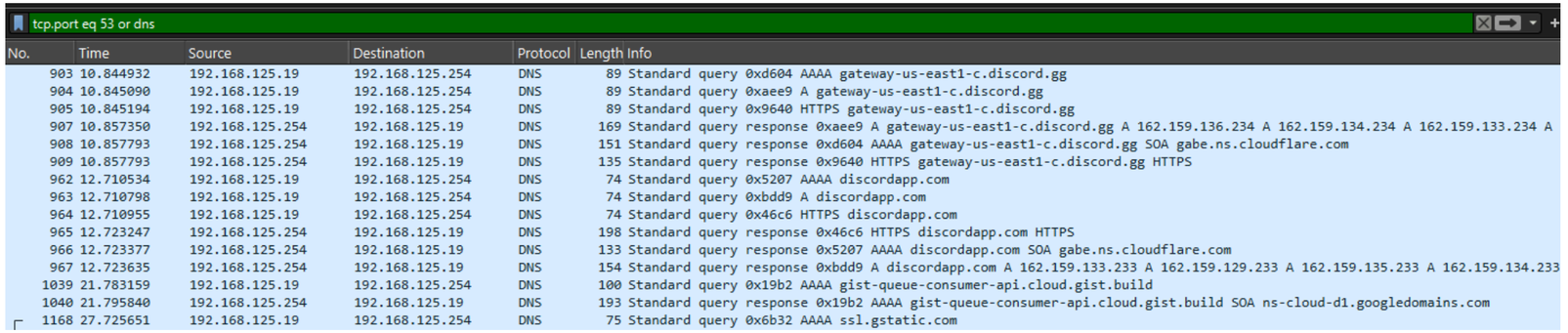
FILTRAGES DE BASE

Quelques exemples de filtrage

Filtrage par protocoles TCP : Le dernier filtre que nous aborderons est le filtre de protocole, cela vous permet de définir un port ou un protocole par lequel filtrer et peut être pratique lorsque vous essayez de garder une trace d'un protocole ou d'un port inhabituel utilisé.

Filtre utilisé:

tcp.port eq <Port #> or <Protocol Name>



No.	Time	Source	Destination	Protocol	Length	Info
903	10.844932	192.168.125.19	192.168.125.254	DNS	89	Standard query 0xd604 AAAA gateway-us-east1-c.discord.gg
904	10.845090	192.168.125.19	192.168.125.254	DNS	89	Standard query 0xaee9 A gateway-us-east1-c.discord.gg
905	10.845194	192.168.125.19	192.168.125.254	DNS	89	Standard query 0x9640 HTTPS gateway-us-east1-c.discord.gg
907	10.857350	192.168.125.254	192.168.125.19	DNS	169	Standard query response 0xaee9 A gateway-us-east1-c.discord.gg A 162.159.136.234 A 162.159.134.234 A 162.159.133.234 A
908	10.857793	192.168.125.254	192.168.125.19	DNS	151	Standard query response 0xd604 AAAA gateway-us-east1-c.discord.gg SOA gabe.ns.cloudflare.com
909	10.857793	192.168.125.254	192.168.125.19	DNS	135	Standard query response 0x9640 HTTPS gateway-us-east1-c.discord.gg HTTPS
962	12.710534	192.168.125.19	192.168.125.254	DNS	74	Standard query 0x5207 AAAA discordapp.com
963	12.710798	192.168.125.19	192.168.125.254	DNS	74	Standard query 0xbdd9 A discordapp.com
964	12.710955	192.168.125.19	192.168.125.254	DNS	74	Standard query 0x46c6 HTTPS discordapp.com
965	12.723247	192.168.125.254	192.168.125.19	DNS	198	Standard query response 0x46c6 HTTPS discordapp.com HTTPS
966	12.723377	192.168.125.254	192.168.125.19	DNS	133	Standard query response 0x5207 AAAA discordapp.com SOA gabe.ns.cloudflare.com
967	12.723635	192.168.125.254	192.168.125.19	DNS	154	Standard query response 0xbdd9 A discordapp.com A 162.159.133.233 A 162.159.129.233 A 162.159.135.233 A 162.159.134.233
1039	21.783159	192.168.125.19	192.168.125.254	DNS	100	Standard query 0x19b2 AAAA gist-queue-consumer-api.cloud.gist.build
1040	21.795840	192.168.125.254	192.168.125.19	DNS	193	Standard query response 0x19b2 AAAA gist-queue-consumer-api.cloud.gist.build SOA ns-cloud-d1.googledomains.com
1168	27.725651	192.168.125.19	192.168.125.254	DNS	75	Standard query 0x6b32 AAAA ssl.gstatic.com

FILTRAGES DE BASE

Protocole ARP avec wireshark

Ci-dessous, vous pouvez voir une capture de paquets de plusieurs requêtes et réponses ARP .

Filtre utilisé:
arp

arp

No.	Time	Source	Destination	Protocol	Length	Info
10719	155.280673	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10893	156.141246	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10906	157.369080	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10916	158.351701	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10929	159.334362	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10942	160.317260	FreeboxSas_76:29:a6	Broadcast	ARP	42	Who has 192.168.125.19? Tell 192.168.125.254
10946	160.441900	FreeboxSas_76:29:a6	Intel_71:0c:a0	ARP	42	Who has 192.168.125.13? Tell 192.168.125.254
10947	160.441921	Intel_71:0c:a0	FreeboxSas_76:29:a6	ARP	42	192.168.125.13 is at c4:23:60:71:0c:a0

Après un nettoyage du cache arp sur Windows

19049	115.196621	Intel_71:0c:a0	Broadcast	ARP	42	Who has 192.168.125.254? Tell 192.168.125.13
19050	115.200196	FreeboxSas_76:29:a6	Intel_71:0c:a0	ARP	42	192.168.125.254 is at 20:66:cf:76:29:a6
20071	124.478921	FreeboxSas_76:29:a6	Intel_71:0c:a0	ARP	42	Who has 192.168.125.13? Tell 192.168.125.254
20072	124.478961	Intel_71:0c:a0	FreeboxSas_76:29:a6	ARP	42	192.168.125.13 is at c4:23:60:71:0c:a0

FILTRAGES DE BASE

WIRESHARK

Protocole ARP avec wireshark

En regardant les détails du paquet ci-dessus, les détails les plus importants du paquet sont l'Opcode qui est l'abréviation de code d'opération et vous indiquera s'il s'agit d'une requête ou d'une réponse ARP . Le deuxième détail indiqué est la destination du paquet, qui dans ce cas, est une demande de diffusion à tous.

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: FreeboxSas_76:29:a6 (20:66:cf:76:29:a6)
    Sender IP address: 192.168.125.254
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.125.13
```

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Intel_71:0c:a0 (c4:23:60:71:0c:a0)
    Sender IP address: 192.168.125.13
    Target MAC address: FreeboxSas_76:29:a6 (20:66:cf:76:29:a6)
    Target IP address: 192.168.125.254
```

LES STATISTIQUES

Pas mal de statistiques disponibles avec Wireshark

Wireshark · Protocol Hierarchy Statistics · Wi-Fi

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	50839	100.0	37638656	329 k	0	0	0	50839
Ethernet	100.0	50839	1.9	717436	6277	0	0	0	50839
Internet Protocol Version 6	79.1	40195	4.3	1607840	14 k	0	0	0	40195
User Datagram Protocol	63.1	32057	0.7	256456	2243	0	0	0	32057
QUIC IETF	21.0	10660	17.2	6470151	56 k	10660	6386141	55 k	10976
Multicast Domain Name System	0.4	212	0.0	12694	111	212	12694	111	212
Link-local Multicast Name Resolution	0.2	93	0.0	2793	24	93	2793	24	93
HiPerConTracer Trace Service	0.0	3	0.0	3690	32	3	3690	32	3
Data	41.5	21089	54.6	20538666	179 k	21089	20538666	179 k	21089
Transmission Control Protocol	15.8	8045	0.5	179316	1569	4396	106336	930	8045
Transport Layer Security	6.5	3280	10.6	4007139	35 k	3280	3623722	31 k	3362
Data	0.7	369	0.0	7574	66	369	7574	66	369
Internet Control Message Protocol v6	0.2	93	0.0	3004	26	93	3004	26	93
Internet Protocol Version 4	20.8	10569	0.6	211400	1849	0	0	0	10569
User Datagram Protocol	5.0	2550	0.1	20400	178	0	0	0	2550
Simple Service Discovery Protocol	0.3	136	0.1	42281	369	136	42281	369	136
QUIC IETF	2.0	1012	1.0	368062	3220	1012	357253	3125	1048
NetBIOS Name Service	0.1	69	0.0	3450	30	69	3450	30	69
Multicast Domain Name System	0.4	217	0.0	12999	113	217	12999	113	217
Link-local Multicast Name Resolution	0.2	93	0.0	2793	24	93	2793	24	93
Domain Name System	2.0	1023	0.2	84986	743	1023	84986	743	1023
Transmission Control Protocol	15.7	7998	0.4	169316	1481	4190	93156	815	7998
Transport Layer Security	6.8	3433	7.7	2914072	25 k	3433	2520095	22 k	3526
Data	0.7	375	0.0	375	3	375	375	3	375
Internet Group Management Protocol	0.0	5	0.0	80	0	5	80	0	5
Internet Control Message Protocol	0.0	16	0.0	576	5	16	576	5	16
Address Resolution Protocol	0.1	75	0.0	2100	18	75	2100	18	75

LES STATISTIQUES

WIRESHARK

Pas mal de statistiques disponibles avec Wireshark

Address A	Address B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
13.107.4.254	192.168.125.13	1	54 bytes	14	1	54 bytes	0	0 bytes	59.788824	0.0000		
104.16.103.112	192.168.125.13	770	104 kB	2	444	42 kB	326	62 kB	2.692928	900.0187	371 bits/s	551 bits/s
104.18.17.5	192.168.125.13	577	73 kB	0	346	33 kB	231	40 kB	0.000000	912.1009	290 bits/s	347 bits/s
192.168.125.3	224.0.0.251	20	6 kB	90	20	6 kB	0	0 bytes	277.254339	320.3488	160 bits/s	0 bits/s
192.168.125.3	239.255.255.250	108	38 kB	6	108	38 kB	0	0 bytes	15.027706	859.0560	353 bits/s	0 bits/s
192.168.125.12	224.0.0.251	5	515 bytes	91	5	515 bytes	0	0 bytes	363.023795	64.5135	63 bits/s	0 bits/s
192.168.125.13	2.18.222.253	44	13 kB	68	20	6 kB	24	6 kB	89.105198	120.9400	409 bits/s	427 bits/s
192.168.125.13	3.219.32.83	43	11 kB	72	20	4 kB	23	7 kB	89.141590	300.3800	96 bits/s	191 bits/s
192.168.125.13	8.2.108.175	20	1 kB	71	10	660 bytes	10	540 bytes	89.140980	2.7110	1947 bits/s	1593 bits/s
192.168.125.13	13.107.5.80	41	16 kB	102	18	4 kB	23	12 kB	723.464526	124.6367	265 bits/s	771 bits/s
192.168.125.13	13.107.42.14	158	26 kB	12	84	10 kB	74	16 kB	40.917827	855.1288	93 bits/s	151 bits/s
192.168.125.13	13.248.245.213	58	22 kB	25	29	8 kB	29	13 kB	82.508418	308.7170	216 bits/s	348 bits/s
192.168.125.13	18.157.230.4	108	47 kB	56	47	23 kB	61	24 kB	85.432265	111.1241	1664 bits/s	1707 bits/s
192.168.125.13	18.195.234.25	31	9 kB	70	18	4 kB	13	5 kB	89.133080	15.2902	2006 bits/s	2623 bits/s
192.168.125.13	18.245.202.34	186	117 kB	20	84	10 kB	102	107 kB	82.112891	240.1693	316 bits/s	3568 bits/s
192.168.125.13	20.73.194.208	48	14 kB	94	28	4 kB	20	10 kB	517.806689	0.5721	59 kbps	135 kbps
192.168.125.13	20.189.173.24	36	13 kB	100	22	5 kB	14	8 kB	658.880094	100.2385	432 bits/s	643 bits/s
192.168.125.13	20.238.236.234	51	4 kB	9	22	2 kB	29	2 kB	34.271988	840.0256	15 bits/s	20 bits/s
192.168.125.13	23.77.161.49	84	32 kB	54	39	17 kB	45	15 kB	85.430795	213.2773	636 bits/s	554 bits/s
192.168.125.13	23.77.162.37	61	27 kB	62	26	7 kB	35	19 kB	88.897047	121.3534	490 bits/s	1265 bits/s
192.168.125.13	34.1.230.247	66	19 kB	40	33	13 kB	33	7 kB	82.923280	610.1618	164 bits/s	91 bits/s
192.168.125.13	34.1.248.70	27	10 kB	75	14	4 kB	13	6 kB	89.147806	0.7503	44 kbps	64 kbps
192.168.125.13	34.18.10.222	29	9 kB	88	14	3 kB	15	6 kB	177.901528	5.5270	4869 bits/s	8390 bits/s
192.168.125.13	34.36.216.150	69	19 kB	81	32	7 kB	37	12 kB	89.676975	610.1257	96 bits/s	151 bits/s
192.168.125.13	34.120.32.134	448	61 kB	10	196	29 kB	252	32 kB	34.444977	679.8570	344 bits/s	375 bits/s
192.168.125.13	34.160.72.119	250	85 kB	19	114	42 kB	136	43 kB	80.522841	708.2246	474 bits/s	487 bits/s
192.168.125.13	35.174.127.31	221	43 kB	4	130	22 kB	91	21 kB	7.475092	902.1434	194 bits/s	183 bits/s
192.168.125.13	35.214.136.108	123	35 kB	36	61	20 kB	62	15 kB	82.622193	617.1806	256 bits/s	193 bits/s
192.168.125.13	35.214.199.88	77	21 kB	32	40	14 kB	37	7 kB	82.613181	616.8717	186 bits/s	86 bits/s
192.168.125.13	35.244.174.68	58	13 kB	80	27	4 kB	31	9 kB	89.423720	610.0776	56 bits/s	117 bits/s

LES STATISTIQUES

WIRESHARK

Pas mal de statistiques disponibles avec Wireshark

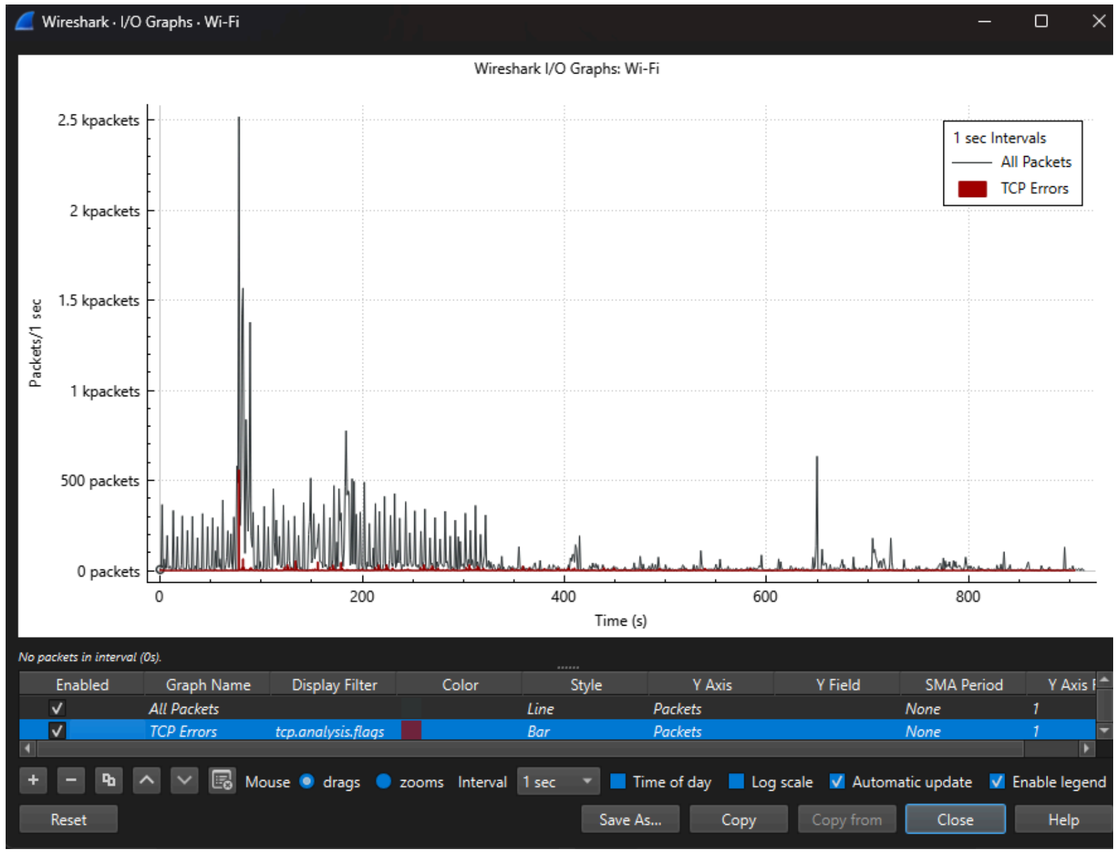
Wireshark · Packet Lengths · Wi-Fi

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	50839	740.35	42	49694	0.0556	100%	5.9800	78.446
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	8100	65.34	42	79	0.0089	15.93%	2.3800	89.305
80-159	13452	100.12	80	159	0.0147	26.46%	1.9500	78.832
160-319	1849	228.83	160	319	0.0020	3.64%	0.2800	194.512
320-639	1818	456.42	320	639	0.0020	3.58%	0.6000	89.308
640-1279	2774	1105.86	640	1277	0.0030	5.46%	0.5700	115.719
1280-2559	22627	1312.12	1280	2553	0.0247	44.51%	4.3400	202.386
2560-5119	95	3821.15	2568	5070	0.0001	0.19%	0.0700	85.465
5120 and greater	124	11207.93	5184	49694	0.0001	0.24%	0.1100	295.323

LES STATISTIQUES

WIRESHARK

Pas mal de statistiques disponibles avec Wireshark



QUESTIONS