

🔗 Analyse de paquets réseau avec tcpdump et Wireshark

🔗 Objectif :

Capturer les paquets réseau sur un serveur distant (Debian), les transférer sur une machine Kali Linux avec interface graphique, puis les analyser via Wireshark.

🔗 Pré-requis :

- tcpdump installé sur le serveur : `sudo apt install tcpdump`
- Accès SSH entre le serveur et la machine Kali
- Wireshark installé sur Kali

🔗 Astuce : Utiliser Wireshark à distance via X11 forwarding

- Activez le X11 Forwarding pour lancer Wireshark depuis le serveur distant sur votre Kali (avec interface graphique).

Commande : `ssh -X user@ip_serveur`

Puis : `wireshark &` (le & permet de continuer à utiliser le terminal)

🔗 Étapes :

1🔗 Déterminer l'interface réseau utilisée

Commande : `ip a`

Recherchez l'interface connectée au réseau (ex. : `ens18`).

```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether bc:24:11:b5:0e:9e brd ff:ff:ff:ff:ff:ff
    altnam enp0s18
    inet 192.168.27.10/24 brd 192.168.27.255 scope global ens18
        valid lft forever preferred_lft forever
    inet6 fe80::be24:11ff:feb5:e9e/64 scope link
        valid lft forever preferred_lft forever
```

2🔗 Capturer les paquets

- Tout le trafic : `sudo tcpdump -i ens18 -w /tmp/capture.pcap`
- Trafic sur un port spécifique : `sudo tcpdump -i ens18 port 50271 -w /tmp/capture.pcap`

Arrêtez avec CTRL+C après quelques secondes.

```
aftec@debian:/etc/nginx/sites-available$ sudo tcpdump -i ens18 -w /tmp/capture.pcap
[sudo] Mot de passe de aftec :
tcpdump: listening on ens18, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C362 packets captured
364 packets received by filter
0 packets dropped by kernel
aftec@debian:/etc/nginx/sites-available$ █
```

3. Transférer la capture sur Kali

Commande sur Kali :

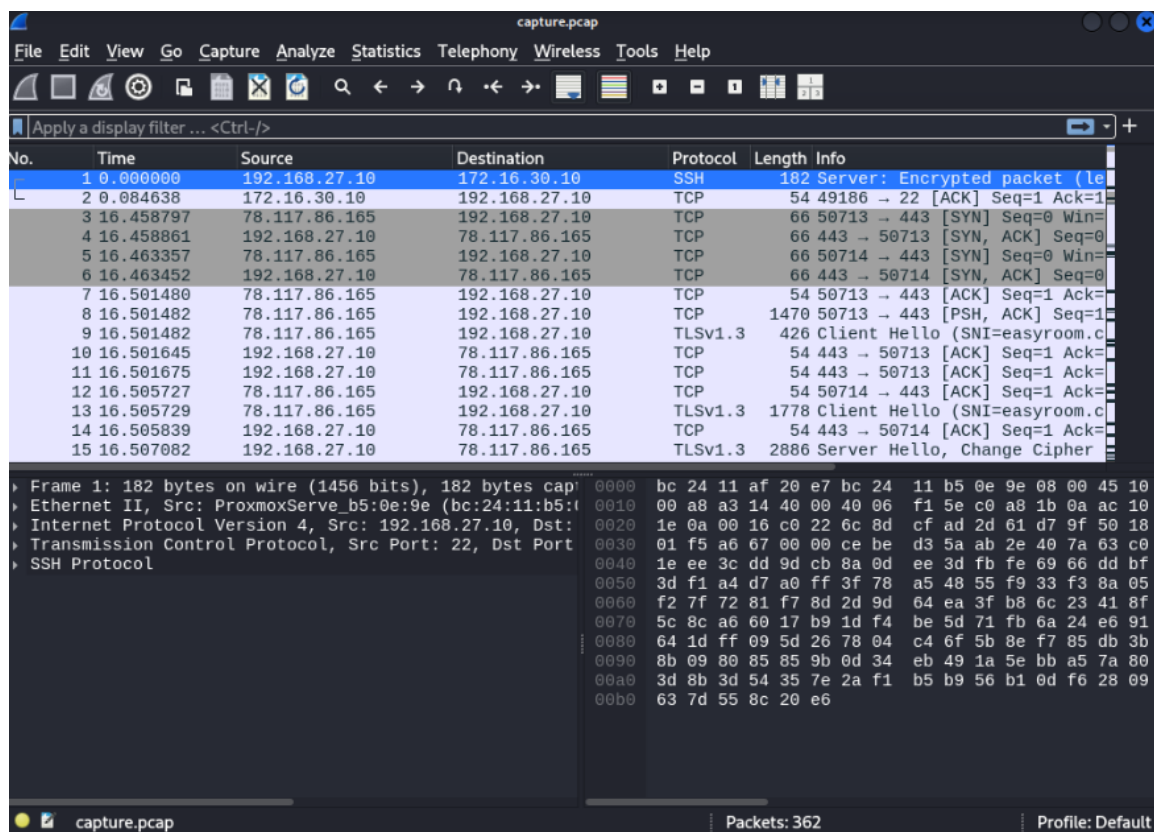
scp user@ip_serveur:/tmp/capture.pcap .

4. Ouvrir et analyser la capture

Commande : wireshark capture.pcap

```
(aftec@kali)~$ scp aftec@192.168.27.10:/tmp/capture.pcap .
aftec@192.168.27.10's password:
capture.pcap
100% 279KB 25.9MB/s 00:00

(aftec@kali)~$ wireshark capture.pcap
Completing file
capture.pcap capture_mangaverse.pcap
```



5. Utiliser les filtres dans Wireshark

Exemples de filtres utiles :

- tcp → pour ne voir que les paquets TCP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.27.10	172.16.30.10	SSH	182	Server: Encrypted packet (le
2	0.084638	172.16.30.10	192.168.27.10	TCP	54	49186 → 22 [ACK] Seq=1 Ack=1
3	16.458797	78.117.86.165	192.168.27.10	TCP	66	50713 → 443 [SYN] Seq=0 Win=
4	16.458861	192.168.27.10	78.117.86.165	TCP	66	443 → 50713 [SYN, ACK] Seq=0
5	16.463357	78.117.86.165	192.168.27.10	TCP	66	50714 → 443 [SYN] Seq=0 Win=
6	16.463452	192.168.27.10	78.117.86.165	TCP	66	443 → 50714 [SYN, ACK] Seq=0
7	16.501480	78.117.86.165	192.168.27.10	TCP	54	50713 → 443 [ACK] Seq=1 Ack=
8	16.501482	78.117.86.165	192.168.27.10	TCP	1470	50713 → 443 [PSH, ACK] Seq=1
9	16.501482	78.117.86.165	192.168.27.10	TLSv1.3	426	Client Hello (SNI=easyroom.c
10	16.501645	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
11	16.501675	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
12	16.505727	78.117.86.165	192.168.27.10	TCP	54	50714 → 443 [ACK] Seq=1 Ack=
13	16.505729	78.117.86.165	192.168.27.10	TLSv1.3	1778	Client Hello (SNI=easyroom.c
14	16.505839	192.168.27.10	78.117.86.165	TCP	54	443 → 50714 [ACK] Seq=1 Ack=
15	16.507082	192.168.27.10	78.117.86.165	TLSv1.3	2886	Server Hello, Change Cipher

- tcp.port == 443 → pour cibler un port précis

No.	Time	Source	Destination	Protocol	Length	Info
3	16.458797	78.117.86.165	192.168.27.10	TCP	66	50713 → 443 [SYN] Seq=0 Win=
4	16.458861	192.168.27.10	78.117.86.165	TCP	66	443 → 50713 [SYN, ACK] Seq=0
5	16.463357	78.117.86.165	192.168.27.10	TCP	66	50714 → 443 [SYN] Seq=0 Win=
6	16.463452	192.168.27.10	78.117.86.165	TCP	66	443 → 50714 [SYN, ACK] Seq=0
7	16.501480	78.117.86.165	192.168.27.10	TCP	54	50713 → 443 [ACK] Seq=1 Ack=
8	16.501482	78.117.86.165	192.168.27.10	TCP	1470	50713 → 443 [PSH, ACK] Seq=1
9	16.501482	78.117.86.165	192.168.27.10	TLSv1.3	426	Client Hello (SNI=easyroom.c
10	16.501645	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
11	16.501675	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
12	16.505727	78.117.86.165	192.168.27.10	TCP	54	50714 → 443 [ACK] Seq=1 Ack=
13	16.505729	78.117.86.165	192.168.27.10	TLSv1.3	1778	Client Hello (SNI=easyroom.c
14	16.505839	192.168.27.10	78.117.86.165	TCP	54	443 → 50714 [ACK] Seq=1 Ack=
15	16.507082	192.168.27.10	78.117.86.165	TLSv1.3	2886	Server Hello, Change Cipher
16	16.507106	192.168.27.10	78.117.86.165	TLSv1.3	424	Application Data, Applicatio
17	16.510166	192.168.27.10	78.117.86.165	TLSv1.3	2886	Server Hello, Change Cipher

- ip.addr == 192.168.27.10 → pour filtrer une IP spécifique

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.27.10	172.16.30.10	SSH	182	Server: Encrypted packet (le
2	0.084638	172.16.30.10	192.168.27.10	TCP	54	49186 → 22 [ACK] Seq=1 Ack=1
3	16.458797	78.117.86.165	192.168.27.10	TCP	66	50713 → 443 [SYN] Seq=0 Win=
4	16.458861	192.168.27.10	78.117.86.165	TCP	66	443 → 50713 [SYN, ACK] Seq=0
5	16.463357	78.117.86.165	192.168.27.10	TCP	66	50714 → 443 [SYN] Seq=0 Win=
6	16.463452	192.168.27.10	78.117.86.165	TCP	66	443 → 50714 [SYN, ACK] Seq=0
7	16.501480	78.117.86.165	192.168.27.10	TCP	54	50713 → 443 [ACK] Seq=1 Ack=
8	16.501482	78.117.86.165	192.168.27.10	TCP	1470	50713 → 443 [PSH, ACK] Seq=1
9	16.501482	78.117.86.165	192.168.27.10	TLSv1.3	426	Client Hello (SNI=easyroom.c
10	16.501645	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
11	16.501675	192.168.27.10	78.117.86.165	TCP	54	443 → 50713 [ACK] Seq=1 Ack=
12	16.505727	78.117.86.165	192.168.27.10	TCP	54	50714 → 443 [ACK] Seq=1 Ack=
13	16.505729	78.117.86.165	192.168.27.10	TLSv1.3	1778	Client Hello (SNI=easyroom.c
14	16.505839	192.168.27.10	78.117.86.165	TCP	54	443 → 50714 [ACK] Seq=1 Ack=
15	16.507082	192.168.27.10	78.117.86.165	TLSv1.3	2886	Server Hello, Change Cipher