



# PROJET SSI

# 2024 - 2BIS1

# parties(AetB)

# Membres du groupe

Mohamed Arbi Ghanmi

Nader ben ammar

Nour Charfeddine

# Elaboration du cadre du projet

---

Pour ce projet, nous avons choisi d'utiliser les systèmes d'exploitation Ubuntu et Kali Linux. Ces deux distributions sont construites sur la base de Debian, ce qui leur confère une stabilité et une robustesse reconnues. Nous avons spécifiquement sélectionné Ubuntu pour sa polyvalence et sa convivialité générale, ainsi que Kali Linux pour sa suite d'outils intégrés spécialisés dans les tests de pénétration et le hacking éthique. En optant pour ces distributions, nous nous assurons que notre infrastructure informatique est bien adaptée aux exigences de sécurité et de performance nécessaires à notre projet.

# Partie A : SNIFFING

## Plan Du Travail :

- Explication du sniffing + exemples d'outils d'implémentation
- Collecte du trafic et identification des mots de passe entre le client et le serveur (FTP/SSH)



# SNIFFING

## Définition:

Le sniffing est une attaque passive où un attaquant intercepte et analyse le trafic réseau pour récupérer des informations sensibles telles que des identifiants de connexion, des données personnelles ou des informations confidentielles. Cette attaque se base sur l'écoute du trafic sans altérer les données transitant sur le réseau.

# SNIFFING

## Importance:

Le sniffing joue un rôle crucial dans le processus de tests de sécurité, en particulier dans le cadre des tests d'intrusion et de l'évaluation de la sécurité des réseaux.

# SNIFFING-utilité

## Détection des vulnérabilités

En analysant le trafic réseau, les testeurs de sécurité repèrent des vulnérabilités potentielles, comme la transmission non sécurisée de données sensibles ou des informations d'identification non protégées, permettant ainsi de renforcer la sécurité.

## Évaluation de la sécurité des communications

le sniffing sert à évaluer la sécurité des communications entre les composants d'un système en analysant le chiffrement, en détectant les protocoles non sécurisés, et en vérifiant la mise en place des bonnes pratiques de sécurité des communications.

## Tests de l'efficacité des mesures de sécurité

Les entreprises mettent en place diverses mesures de sécurité telles que les pare-feu, les systèmes de détection d'intrusion (IDS) et les protocoles de chiffrement. Le sniffing permet de tester l'efficacité de ces mesures en tentant de les contourner ou de les neutraliser.

# SNIFFING-utilité

## Identification des attaques

### "Man-in-the-Middle" (MITM)

Les attaques MITM sont des menaces sérieuses pour la confidentialité et l'intégrité des données. En utilisant le sniffing, les testeurs de sécurité peuvent détecter et évaluer la faisabilité d'attaques MITM, ce qui est essentiel pour renforcer la sécurité des communications.

## Analyse des flux de données

Le sniffing permet d'analyser les flux de données pour identifier des schémas de trafic anormaux ou des comportements suspects. Cela peut aider à détecter des activités malveillantes telles que des exfiltrations de données ou des tentatives d'intrusion.

## Éducation et sensibilisation

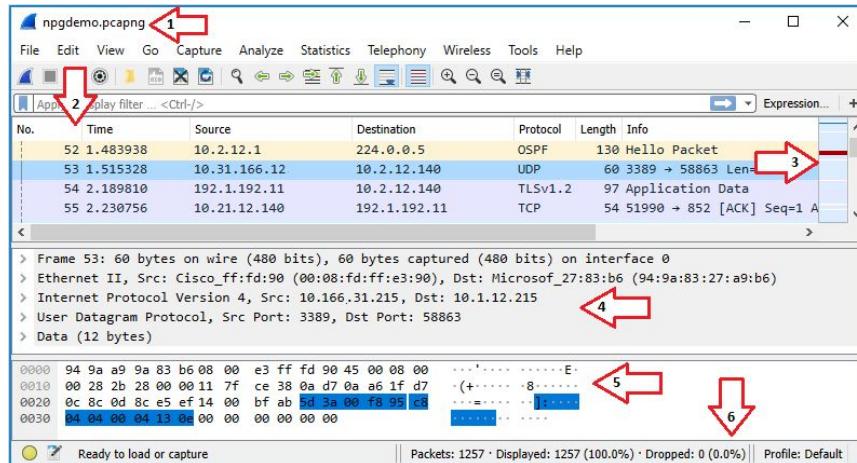
Les résultats des tests de sniffing peuvent être utilisés pour sensibiliser le personnel aux risques liés à la sécurité des données. Cela contribue à renforcer la culture de la sécurité au sein de l'organisation.

# SNIFFING-Outils

## Wireshark (anciennement Ethereal) :

Wireshark est un outil de capture de paquets open source qui permet aux utilisateurs d'analyser le trafic réseau en temps réel. Il peut être utilisé pour inspecter le contenu des paquets et extraire des informations sensibles.

### interface:



1. Title Bar
2. Packet List Pane
3. Intelligent Scrollbar
4. Packet Details Pane
5. Packet Bytes Pane
6. The Statusbar

# SNIFFING-Outils

**Tcpdump :** Tcpdump est un utilitaire en ligne de commande utilisé pour capturer et afficher le trafic réseau. Il permet aux utilisateurs de spécifier des filtres pour cibler des types spécifiques de paquets.

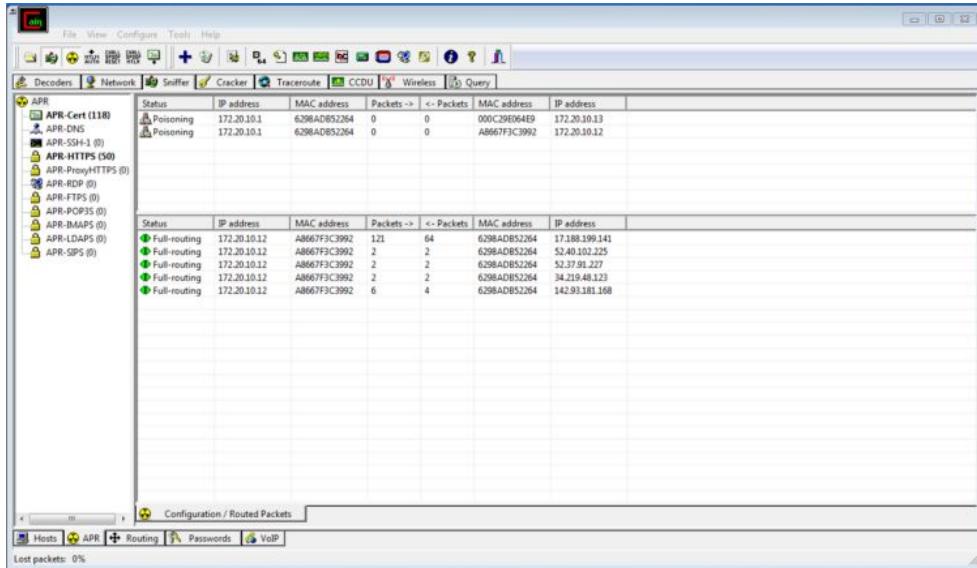
**interface(terminal):**

```
team@LHB:~$ sudo tcpdump -D
1.eth0 [Up, Running]
2.eth1 [Up, Running]
3.lo [Up, Running, Loopback]
4.any (Pseudo-device that captures on all interfaces) [Up, Running]
5.bluetooth-monitor (Bluetooth Linux Monitor) [none]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
team@LHB:~$ sudo tcpdump --list-interfaces
1.eth0 [Up, Running]
2.eth1 [Up, Running]
3.lo [Up, Running, Loopback]
4.any (Pseudo-device that captures on all interfaces) [Up, Running]
5.bluetooth-monitor (Bluetooth Linux Monitor) [none]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
team@LHB:~$
```



# SNIFFING-Outils

**Cain and Abel :** Cet outil est souvent utilisé pour le sniffing sur les réseaux locaux. Il peut capturer des paquets et effectuer des attaques par dictionnaire pour récupérer des mots de passe.  
**interface:**



# SNIFFING-Outils

**Ettercap :** Ettercap est un outil de sniffing qui peut être utilisé pour réaliser des attaques de type "Man-in-the-Middle" (MITM), où l'attaquant intercepte et modifie les communications entre deux parties sans leur consentement.

**interface:**

Applications ▾ Places ▾ Ettercap ▾

Mon 03:31 •  
ettercap 0.8.2

Start Targets Hosts View Mitm Filters Logging Plugins Info

Host List x Connections x

Host filter: [ ] Protocol filter:  TCP  UDP  Other Connection state filter:  Active  Idle  Closing  Closed  Killed

| Host                      | Port  | - | Host                                    | Port  | Proto | State  | TX Bytes | RX Bytes |
|---------------------------|-------|---|---|-------|-------|--------|----------|----------|
| 192.168.0.4               | 56494 | - | 192.168.0.255                           | 14440 | UDP   | idle   | 6642     | 0        |
| 192.168.0.52              | 57621 | - | 192.168.0.255                           | 57621 | UDP   | idle   | 616      | 0        |
| fe80::4270:9fff:fe7a:6497 | 0     | - | ff02::1                                 | 0     |       | killed | 0        | 0        |
| fe80::4270:9fff:fe7a:6497 | 0     | - | 2606:6000:66e3:f500:941d:dbea:4674:b9ed | 0     |       | idle   | 0        | 0        |
| 192.168.0.6               | 68    | - | 255.255.255.255                         | 67    | UDP   | idle   | 1200     | 0        |
| 192.168.0.6               | 137   | - | 192.168.0.255                           | 137   | UDP   | idle   | 900      | 0        |
| 192.168.0.56              | 17500 | - | 255.255.255.255                         | 17500 | UDP   | idle   | 1876     | 0        |
| 192.168.0.56              | 17500 | - | 192.168.0.255                           | 17500 | UDP   | idle   | 1876     | 0        |
| 192.168.0.14              | 5353  | - | 224.0.0.251                             | 5353  | UDP   | idle   | 1220     | 0        |
| fe80::1c39:8770:4320:4282 | 0     | - | ff02::16                                | 0     |       | idle   | 0        | 0        |
| 192.168.0.10              | 127   | - | 192.168.0.255                           | 127   | UDP   | idle   | 10650    | 0        |

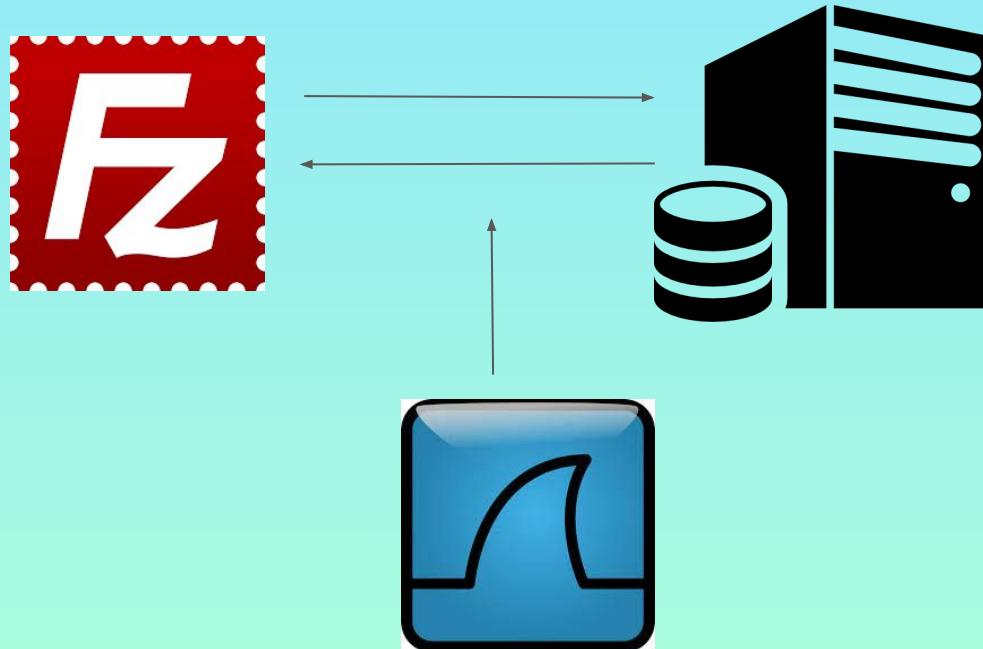
[View Details](#) [Kill Connection](#)



# Le FTP

Dans cette partie on va faire la collecte de trafics générés entre un client et un serveur, on va aussi identifier le mot de passe utilisé lors de l'utilisation du protocole d'authentification pour le service FTP

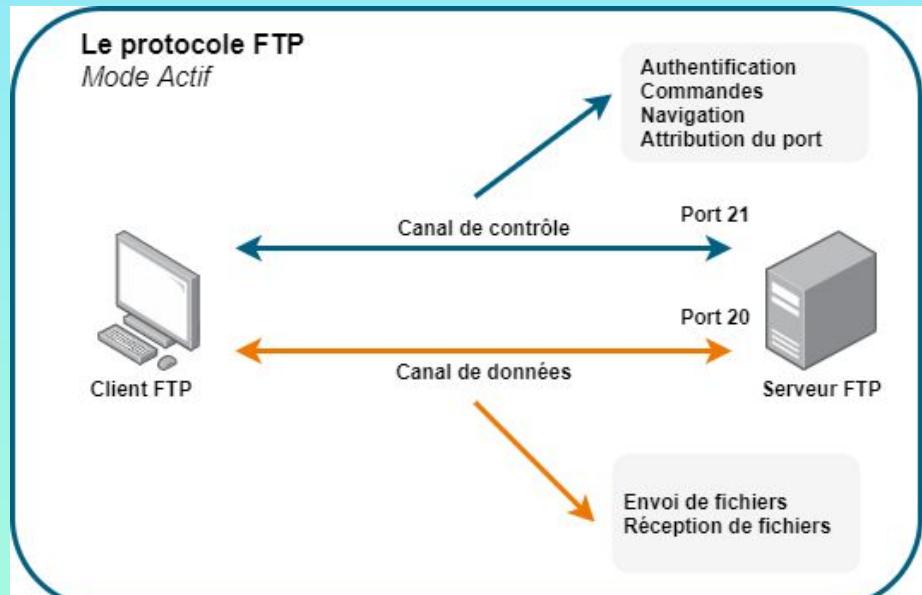
**NB:** on a ajouté un utilisateur, nommé `ftp_client`, au niveau du serveur pour ce test



# Configuration du serveur FTP

## Définition :

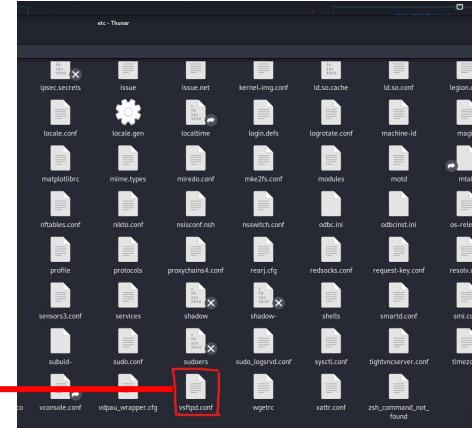
Le protocole FTP (File Transfer Protocol) est un protocole de communication utilisé pour transférer des fichiers entre des ordinateurs sur un réseau TCP/IP, tel qu'Internet. Il définit la manière dont les données sont transférées, la gestion des connexions et des sessions, ainsi que les opérations de navigation et de manipulation de fichiers sur un serveur distant.



# 1. Installation du serveur VSFTPD

- Il s'agit d'un serveur FTP open source conçu pour être rapide.
- Le projet vsftpd est maintenu par Chris Evans et est distribué sous la licence GNU GPL.

```
(root㉿kali)-[~/home/test/Downloads/compat-wireless-2010-06-26-p]
# apt-get install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cython3 debtags kali-debtags libboost-dev libboost1.83-dev libnihredis0.14 libjavascripcoregtk-4 0-18
  libopenblas-dev libopenblas-pthread-dev libopenblas0 libperl5.36 libpython3-all-dev libpython3.12
  libpython3.12-dev libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasettings5 libqt5multimediawidgets5
  librltsdr0 libubcl libwebkit2gtk-4.0-37 libxsimd-dev libzxxing2 perl-modules-5.36 python3-all-dev
  python3-backcall python3-beniget python3-debian python3-future python3-gast python3-picleshare python3-pythr
  python3-requests-toolbelt python3-rfc3986 python3-unicodecsv python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  vsftpd
0 upgraded, 0 newly installed, 0 to remove and 141 not upgraded.
Need to get 143 kB of archives.
After this operation, 353 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 vsftpd amd64 3.0.3-13+b3 [143 kB]
Fetched 143 kB in 2s (74.0 kB/s).
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 419751 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.3-13+b3_amd64.deb ...
Unpacking vsftpd (3.0.3-13+b3) ...
Setting up vsftpd (3.0.3-13+b3) ...
update-rc.d: We have no instructions for the vsftpd init script.
update-rc.d: It looks like a network service, we disable it.
Processing triggers for man-db (2.12.0-3) ...
Processing triggers for kali-menu (2023.4.7) ...
```



# Etablissement de la connexion

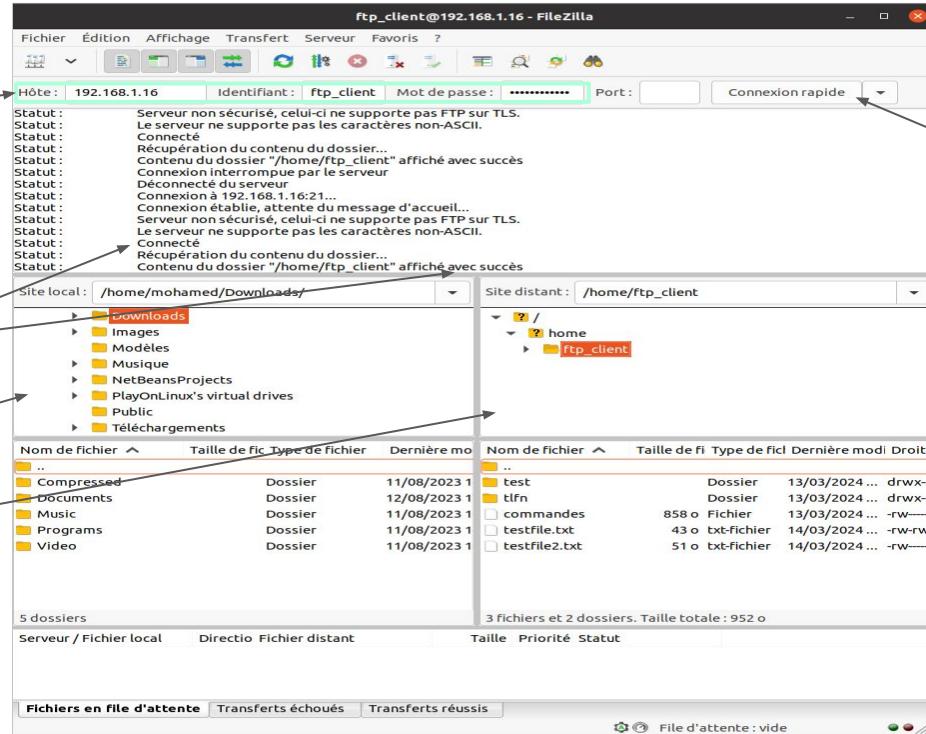
- On a utilisé FileZilla(un client FTP) pour connecter au serveur et manipuler les fichiers.

1/ Introduction les données relatives au serveur

3/ connexion réussie

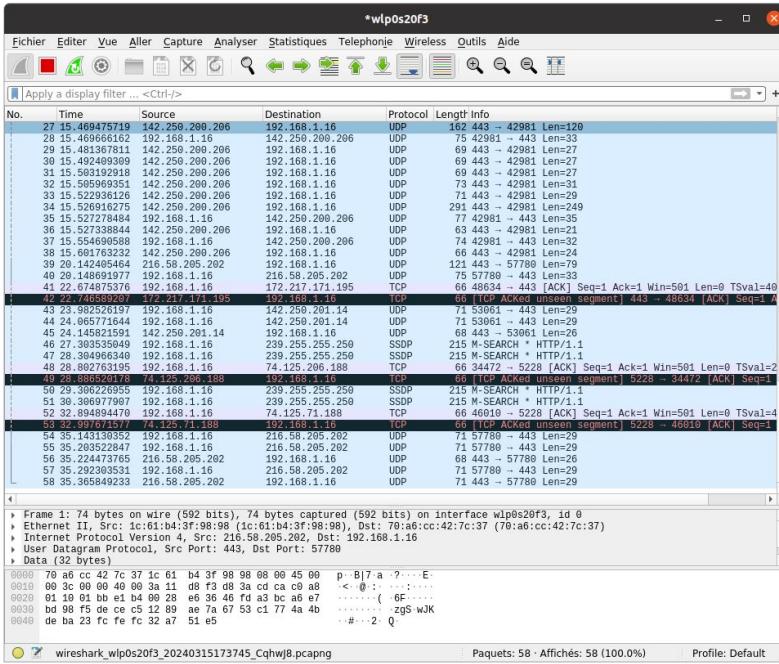
On peut voir notre pc(site local) à gauche, et le serveur(site distant) à droite

2/ établir une connexion rapide( ou bien créer un site d'après "fichier")



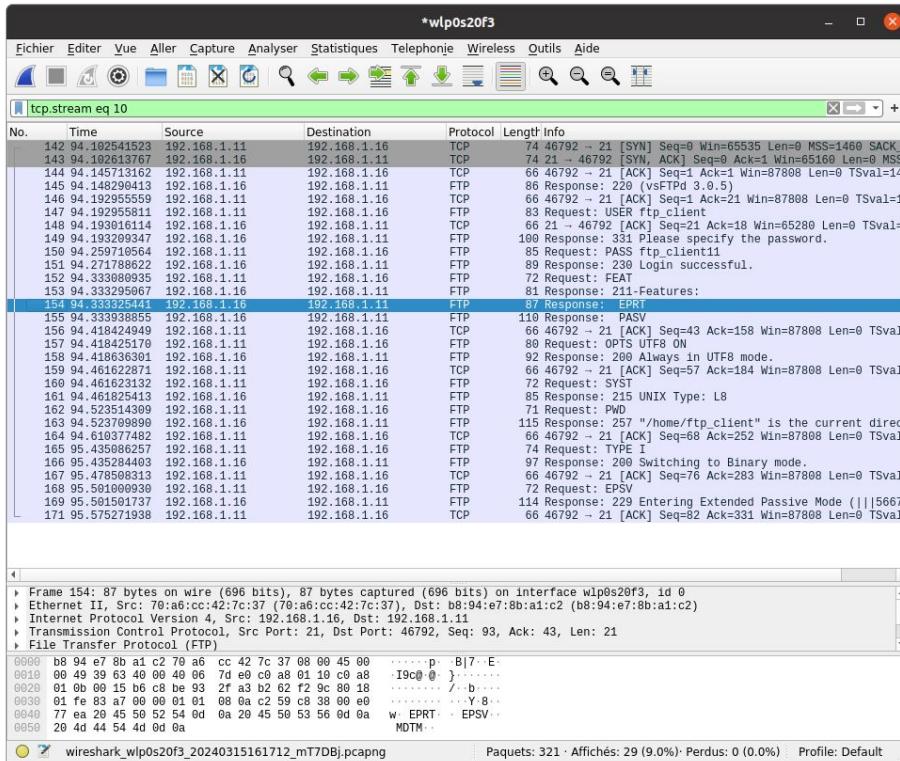
# Sniffing avec wireshark

- Lors de l'établissement de la connexion, wireshark était déjà lancé pour capturer tout ce qui se passe au niveau de notre réseau, notamment entre le client et le serveur



# Sniffing avec wireshark

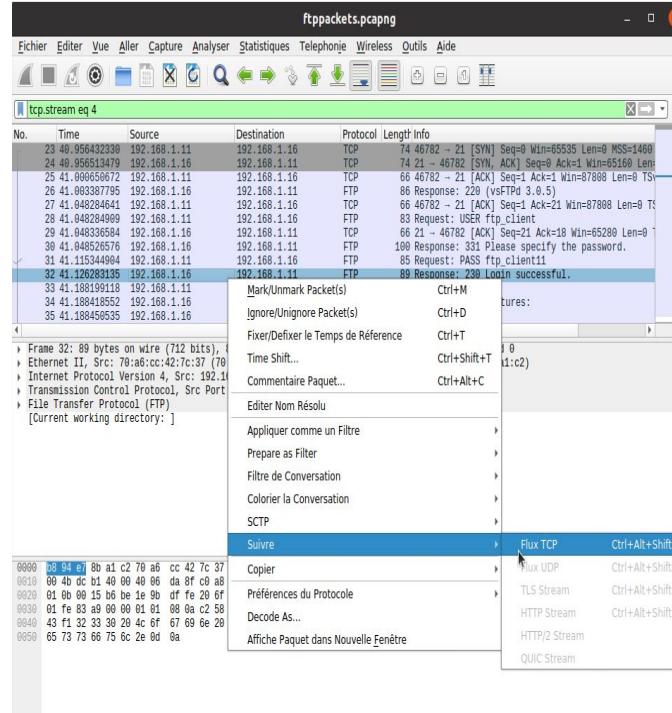
- On arrête le sniffing et on filtre par le nom de notre protocole "ftp"



On peut voir plusieurs paquets échangés entre le client et le serveur utilisant le protocole "ftp", et c'est ce qu'on cherche

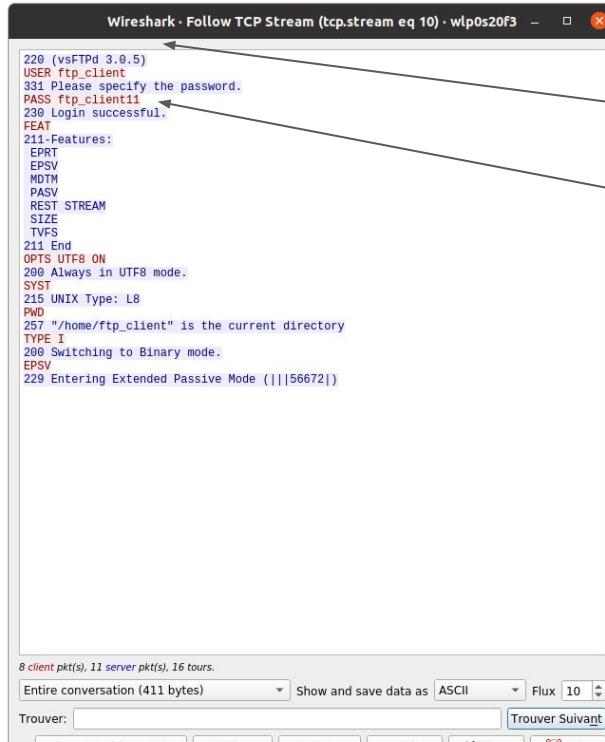
# Sniffing avec wireshark

- On suit le flux TCP de l'une de ces paquets pour dévoiler les données de l'authentification



# Divulgation du mot de passe

- Puisque FTP est un protocole qui n'est sécurisé(les données sont transférées en plein texte, non cryptées), on peut divulguer facilement le nom d'utilisateur et le mot de passe du serveur



Wireshark - Follow TCP Stream (tcp.stream eq 10) - wlp0s20f3

```
220 (vsFTPd 3.0.5)
USER ftp_client
331 Please specify the password.
PASS ftp_client11
230 Login successful.
FEAT
211-Features:
EPRT
EPSV
MDTM
PASV
REST STREAM
SIZE
TVFS
211 End
OPTS UTF8 ON
200 Always in UTF8 mode.
SYST
215 UNIX Type: L8
PWD
257 "/home/ftp_client" is the current directory
TYPE I
200 Switching to Binary mode.
EPSV
229 Entering Extended Passive Mode (|||56672|)
```

8 client pkts(s), 11 server pkts(s), 16 tours.

Entire conversation (411 bytes) Show and save data as ASCII Flux 10 Trouver Suivant Trouver: Filter Out This Stream Imprimer Sauvegarde Back Class Help

Nom d'utilisateur

Mot de passe

# Manipulation des fichiers

- Une fois on est authentifié, on peut télécharger ou envoyer des fichiers au serveur

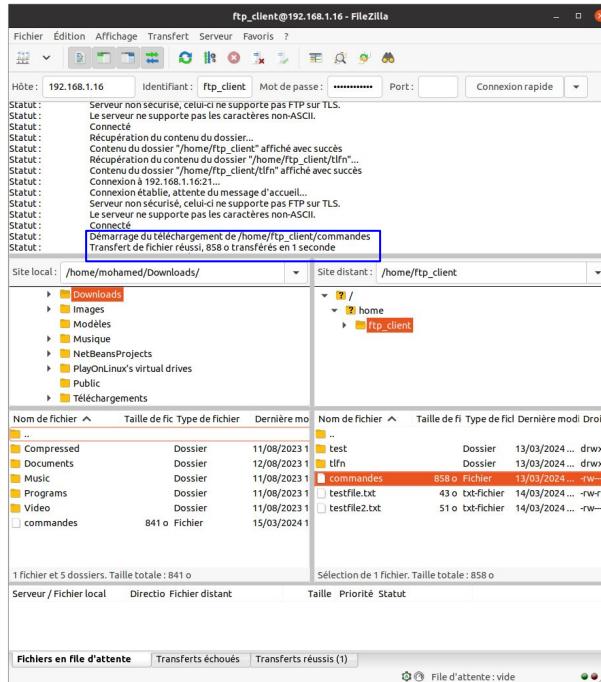


Image 1

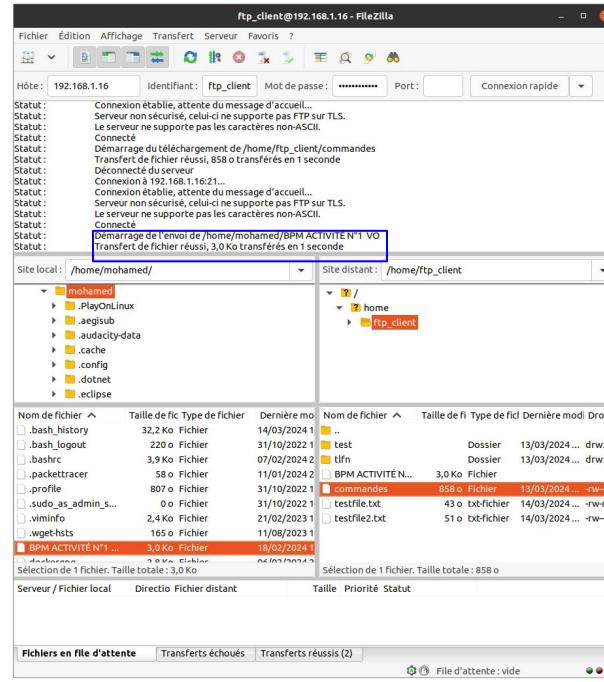


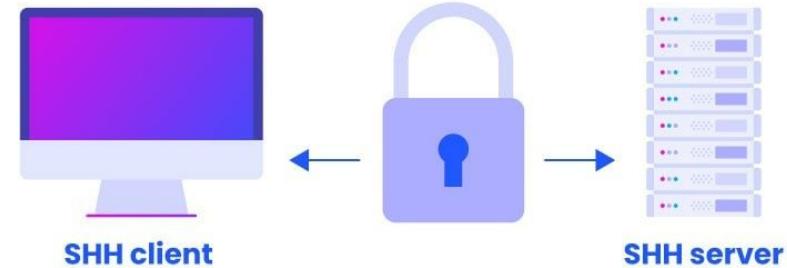
Image 2

Image 1: on a téléchargé le fichier "commandes" du serveur au pc dans le dossier Downloads

Image 2: on a envoyé le fichier "BPM ACTIVITÉ N°1 VO" du pc au serveur

# Le Protocole SSH

Le protocole SSH (Secure Shell) est un protocole de communication sécurisé conçu pour permettre l'accès à distance à des systèmes informatiques et assurer des transmissions de données sécurisées sur un réseau. Il fournit un mécanisme permettant à un utilisateur de se connecter à une machine distante de manière sécurisée et d'exécuter des commandes à distance, tout en garantissant la confidentialité, l'intégrité et l'authenticité des données échangées.



**NB:** le changement des adresses ip est dû aux changement du réseau wifi

# SSH-installation

- La commande "**apt install openssh-server**" est utilisée sur les systèmes d'exploitation basés sur Debian (comme Kali Linux) pour installer le serveur SSH.

```
zsh: corrupt history file /home/test/.zsh_history
└─[test㉿kali)-[~]
└─$ sudo su
[sudo] password for test:
└─[root㉿kali)-[/home/test]
└─# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3).
openssh-server set to manually installed.
The following packages were automatically installed and are no longer required:
  cython3 debtags kali-debtags libboost-dev libboost1.83-dev libhiredis0.14 libjavascriptcoregtk-4.0-18
  libopenblas-dev libopenblas-pthread-dev libopenblas0 libperl5.36 libpython3-all-dev libpython3.12
  libpython3.12-dev libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasettings5 libqt5multimediawidgets5
  librtlsdr0 libucl1 libwebkit2gtk-4.0-37 libxsimd-dev libzxing2 perl-modules-5.36 python3-all-dev
  python3-backcall python3-beniget python3-debian python3-future python3-gast python3-pickleshare python3-pytrans
  python3-requests-toolbelt python3-rfc3986 python3-unicodecsv python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

Broadcom
└─[root㉿kali)-[/home/test]
└─#
```

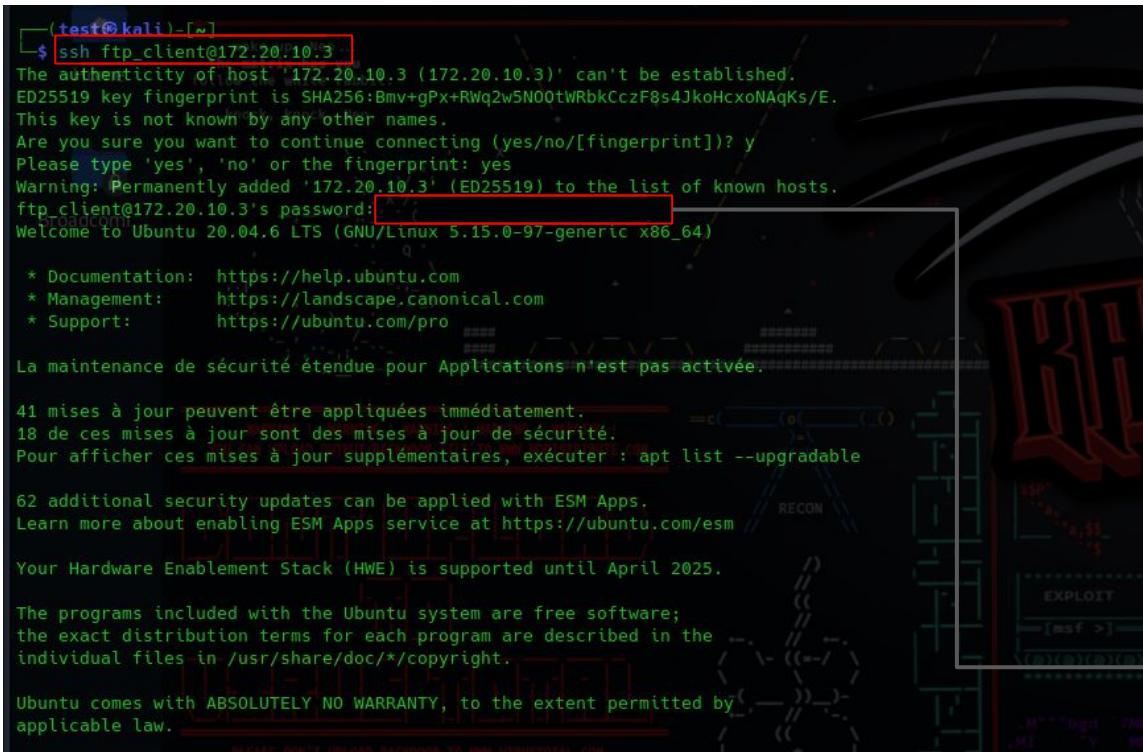
# SSH-lancement

- Après avoir lancé le service SSH via la commande "**sudo service ssh start**", nous pouvons vérifier l'état du serveur en utilisant la commande "**sudo service ssh status**". Cela nous permet de nous assurer que le serveur SSH fonctionne correctement et est prêt à accepter les connexions des clients.

```
(root㉿kali)-[~/home/test]
# sudo service ssh start
(root㉿kali)-[~/home/test]
# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-03-13 11:11:53 EDT; 24s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 2852 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2854 (sshd)
      Tasks: 1 (limit: 2273)
     Memory: 2.9M (peak: 3.3M)
        CPU: 23ms
       CGroup: /system.slice/ssh.service
               └─2854 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Mar 13 11:11:53 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Mar 13 11:11:53 kali sshd[2854]: Server listening on 0.0.0.0 port 22.
Mar 13 11:11:53 kali sshd[2854]: Server listening on :: port 22.
Mar 13 11:11:53 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Lines 1-17... skipping...
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-03-13 11:11:53 EDT; 24s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 2852 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2854 (sshd)
      Tasks: 1 (limit: 2273)
     Memory: 2.9M (peak: 3.3M)
        CPU: 23ms
       CGroup: /system.slice/ssh.service
               └─2854 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

# Connectivité du client au serveur:



```
[test@knit]:~$ ssh ftp_client@172.20.10.3
The authenticity of host '172.20.10.3 (172.20.10.3)' can't be established.
ED25519 key fingerprint is SHA256:Bmv+gPx+RWq2w5N00tWRbkCczF8s4JkoHcxoNaqKs/E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.20.10.3' (ED25519) to the list of known hosts.
ftp_client@172.20.10.3's password: [REDACTED]
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

La maintenance de sécurité étendue pour Applications n'est pas activée.

41 mises à jour peuvent être appliquées immédiatement.
18 de ces mises à jour sont des mises à jour de sécurité.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable

62 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

## Nom du serveur SSH

Après avoir entré la commande **ssh ftp\_client@172.20.10.3** sur l'ordinateur client, ce dernier sera alors invité à fournir son mot de passe afin d'accéder aux fichiers du serveur. **Adresse IP du serveur SSH**

mot de passe client

# Sniffing (Protocole SSH)

\*wlp0s20f3

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonej Wireless Outils Aide

[tcp.stream eq 17]

| No.  | Time           | Source      | Destination | Protocol | Length | Info  |
|------|----------------|-------------|-------------|----------|--------|---|
| 1753 | 279.315195968  | 172.20.10.3 | 172.20.10.2 | TCP      | 74     | 47752 -> 22 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM=1 T...   |
| 1754 | 279.315196001  | 172.20.10.3 | 172.20.10.2 | TCP      | 74     | 47752 -> 22 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM=1 T...   |
| 1755 | 279.332447828  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=1933980875...   |
| 1756 | 279.33223535   | 172.20.10.2 | 172.20.10.3 | SSHv2    | 98     | Client->Server [SSH-2.0-OpenSSH_8.2p1 Debian-3] TSval=1933980875... |
| 1757 | 279.33223535   | 172.20.10.3 | 172.20.10.2 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1 Ack=33 Win=65152 Len=0 TSval=309879796...   |
| 1759 | 279.345319490  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 108    | Server: Protocol [SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.11]         |
| 1760 | 279.359440729  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1 Ack=43 Win=32128 Len=0 TSval=19339809...    |
| 1761 | 279.359486308  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 1146   | Server: Key Exchange Init   |
| 1762 | 279.369965890  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 1602   | Client: Key Exchange Init   |
| 1763 | 279.369965896  | 172.20.10.3 | 172.20.10.2 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1123 Ack=1569 Win=64128 Len=0 TSval=3098...   |
| 1764 | 279.369950001  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 1144   | Client->Server Diffie-Hellman Key Exchange Init                     |
| 1765 | 279.382367727  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 502    | Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypt...     |
| 1766 | 279.394801164  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 82     | Client: New Keys  |
| 1768 | 279.437668840  | 172.20.10.3 | 172.20.10.2 | TCP      | 66     | 62 -> 47752 [ACK] Seq=1559 Ack=1633 Win=64128 Len=0 TSval=3098...   |
| 1769 | 279.457357516  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 110    | Client: Encrypted packet (len=44)                                   |
| 1770 | 279.457395803  | 172.20.10.3 | 172.20.10.2 | TCP      | 66     | 62 -> 47752 [ACK] Seq=1559 Ack=1677 Win=64128 Len=0 TSval=3098...   |
| 1771 | 279.457395805  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 113    | Server: Encrypted packet (len=44)                                   |
| 1772 | 279.467510743  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 134    | Client: Encrypted packet (len=46)                                   |
| 1773 | 279.467510743  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 118    | Server: Encrypted packet (len=52)                                   |
| 1774 | 279.541556096  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1745 Ack=1655 Win=31872 Len=0 TSval=1933...   |
| 1813 | 294.672676175  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 214    | Client: Encrypted packet (len=28)                                   |
| 1814 | 294.691168623  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 94     | Server: Encrypted packet (len=28)                                   |
| 1815 | 294.694246411  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=1893 Ack=1683 Win=31872 Len=0 TSval=1933...   |
| 1816 | 294.69557963   | 172.20.10.3 | 172.20.10.2 | SSHv2    | 178    | Client: Encrypted packet (len=112)                                  |
| 1817 | 294.69557963   | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 22 -> 47752 [ACK] Seq=1893 Ack=1683 Win=64128 Len=0 TSval=3098...   |
| 1818 | 294.933026297  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 694    | Server: Encrypted packet (len=528)                                  |
| 1819 | 294.940376694  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 542    | Client: Encrypted packet (len=476)                                  |
| 1820 | 294.940411656  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 119    | Server: Encrypted packet (len=44)                                   |
| 1821 | 294.944259178  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 526    | Client: Encrypted packet (len=466)                                  |
| 1822 | 294.944319553  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 502    | Server: Encrypted packet (len=436)                                  |
| 1823 | 294.945585189  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 174    | Server: Encrypted packet (len=108)                                  |
| 1824 | 294.947521414  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 734    | Client: Encrypted packet (len=668)                                  |
| 1825 | 294.947521414  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 288    | Server: Encrypted packet (len=668)                                  |
| 1826 | 294.957704763  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=2941 Ack=3561 Win=31872 Len=0 TSval=1933...   |
| 1827 | 295.011736438  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=2941 Ack=3767 Win=31872 Len=0 TSval=1933...   |
| 2452 | 411.108973513  | 172.20.10.3 | 172.20.10.2 | SSHv2    | 102    | Client: Encrypted packet (len=36)                                   |
| 2453 | 411.108973513  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 102    | Server: Encrypted packet (len=36)                                   |
| 2454 | 411.146310100  | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=2977 Ack=3863 Win=31872 Len=0 TSval=1934...   |
| 2455 | 411.1716869899 | 172.20.10.2 | 172.20.10.3 | SSHv2    | 102    | Client: Encrypted packet (len=36)                                   |
| 2456 | 411.1724435763 | 172.20.10.3 | 172.20.10.2 | SSHv2    | 119    | Server: Encrypted packet (len=36)                                   |
| 2458 | 411.1724435763 | 172.20.10.2 | 172.20.10.3 | TCP      | 66     | 47752 -> 22 [ACK] Seq=3048 Ack=3847 Win=31872 Len=0 TSval=1934...   |
| 2460 | 411.920758293  | 172.20.10.2 | 172.20.10.3 | SSHv2    | 102    | Client: Encrypted packet (len=36)                                   |
| 2461 | 411.961562356  | 172.20.10.3 | 172.20.10.2 | TCP      | 66     | 22 -> 47752 [ACK] Seq=3847 Ack=3049 Win=64128 Len=0 TSval=3098...   |

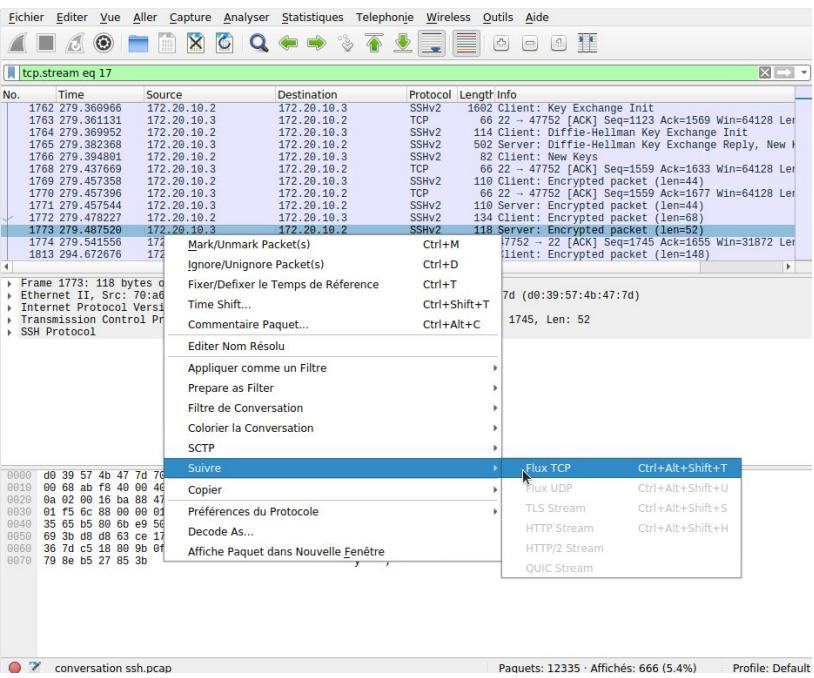
Frame 1756: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface wlp0s20f3, id 0  
Ethernet II, Src: 00:56:3d:49:06:98 (00:56:3d:49:06:98), Dst: 172.20.10.2 (172.20.10.2)  
Internet Protocol Version 4, Src: 172.20.10.4 (172.20.10.4), Dst: 172.20.10.3 (172.20.10.3)  
Transmission Control Protocol, Src Port: 47752, Dst Port: 22, Seq: 1, Ack: 1, Len: 32  
SSH Protocol

0000 70 a6 cc 42 7c 37 d9 39 57 b4 47 7d 08 00 45 10 p\_B|7 9 WKG| E:  
0010 00 56 3d 49 06 98 23 ac 14 02 02 1c 4 176C@ 0 #.....  
0020 0a 03 bc 8a 09 16 a8 d6 63 96 47 16 ad dc 88 1.....  
0030 03 84 01 00 00 00 00 00 00 00 00 00 00 00 00 00 F4 ..  
0040 87 7c f5 53 48 2d 32 2e 30 2d 49 70 65 53 53 |SSH-2.0-OpenSS  
0050 48 5f 39 2e 36 70 31 20 44 65 62 69 61 6e 2d 33 H\_9.6p1 Debian-3  
0060 0d 0a ..

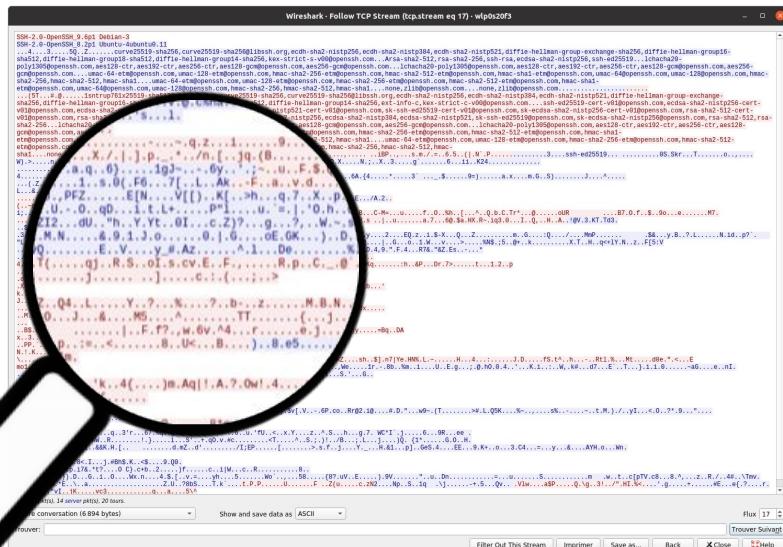
wireshark\_wlp0s20f3\_20240314132603\_7RYCyV.pcapng

En utilisant le sniffer WireShark sur la machine de l'attaquant, il est possible d'observer qu'un nouveau client s'est connecté au serveur et que des échanges de données ont débuté.

## Détection du paquet contenant le mot de passe



## Mot de passe crypté



Le cryptage des mots de passe avec SSH implique la sécurisation des informations sensibles lors de leur transmission sur un réseau à l'aide de techniques de cryptographie robustes intégrées.

# Accès client aux répertoires du serveur

## terminal Client

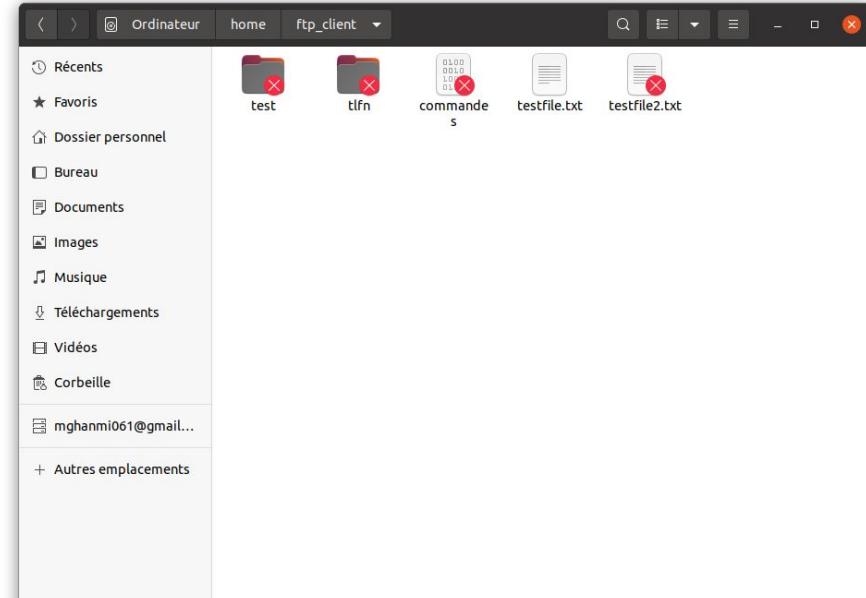
```
$ ls
commandes  test  testfile2.txt  testfile.txt  tlfn
```



La connectivité via le protocole SSH permet également de gérer et de modifier les données sur un serveur à partir d'un hôte distant.

Il est donc possible d'utiliser la commande "ls" afin de lister les fichiers présents dans un répertoire spécifique.

## Interface Serveur



# Gestion des fichiers du serveur

## Terminal Client

```
$ ls
commandes test testfile2.txt testfile.txt tlfn ZISTRIBUTE.COM
$ nano testfile2.txt
```

```
GNU nano 4.8
This is the content in the file.
```

```
GNU nano 4.8
This is the modified content in the file.
```

## Interface Serveur(après la modification du contenu)



À l'aide de la connectivité SSH, il est même possible de modifier le contenu d'un fichier à distance en utilisant la commande "**nano**", suivie du nom du fichier à modifier.



# Partie B : scan des ports et des vulnérabilités

## Plan Du Travail :

- Expliquer les différences entre les deux types de scan
- Pour chaque type de scan, comparer les deux outils les plus utilisés
- Identifier les ports ouverts sur chaque machine du réseau utilisé
- Déterminer le système d'exploitation et l'état de chaque machine du réseau
- Déterminer l'architecture du réseau utilisé
- Identifier les vulnérabilités systèmes les plus graves (critique) dans le réseau utilisé



# les différences entre les deux types de scan

Le scan des ports et le scan des vulnérabilités sont deux techniques utilisées dans le domaine de la sécurité informatique, mais elles ont des objectifs différents et utilisent des approches distinctes :

| Scan des ports  | Scan des vulnérabilités  |
|---|--|
| <p>Le scan des ports consiste à examiner les ports réseau d'une machine ou d'un réseau pour identifier quels ports sont ouverts, fermés ou filtrés.</p>   | <p>Le scan des vulnérabilités vise à identifier les failles de sécurité et les vulnérabilités potentielles dans les logiciels, les systèmes d'exploitation ou les applications.</p>  |
| <p>Les ports sont des canaux de communication utilisés par les ordinateurs pour échanger des données. Les ports ouverts peuvent indiquer des services ou des applications en cours d'exécution sur une machine.</p> | <p>Contrairement au scan des ports qui se concentre sur la disponibilité et la configuration des ports réseau, le scan des vulnérabilités cherche spécifiquement des faiblesses qui pourraient être exploitées par des attaquants.</p> |

# Scan des ports

---

Le scan de ports est une technique cruciale en sécurité informatique permettant de découvrir les ports ouverts sur un système et d'identifier les services associés. Cette information est essentielle pour évaluer la topologie réseau et déterminer la surface d'attaque d'un système. Donc on va explorer en détail deux outils majeurs de scan de ports, Nmap et Masscan

**Nmap** : Reconnu pour sa polyvalence, Nmap excelle dans la découverte de réseaux, l'identification de services, et la cartographie de réseaux. Il offre une variété d'options avancées pour des analyses personnalisées.

**Masscan** : Spécialisé dans les scans massifs, Masscan est réputé pour sa vitesse exceptionnelle, en particulier lorsqu'il s'agit de balayer l'ensemble d'une plage d'adresses IP. Il se distingue par sa rapidité dans les environnements nécessitant des analyses à grande échelle.

|                    | <b>nmap</b>  | <b>Masscan</b>  |
|--------------------|--|---|
| <b>Vitesse</b>     | Bien que Nmap soit un outil puissant, il n'est pas aussi rapide que Masscan pour les scans de ports en masse. Nmap peut prendre plus de temps pour balayer l'ensemble d'une plage d'adresses IP.   | Masscan est spécifiquement conçu pour être extrêmement rapide. Il peut balayer l'ensemble de l'espace d'adressage IPv4 en quelques minutes, ce qui en fait un choix idéal pour les scans de ports en masse. |
| <b>Précision</b>   | Nmap est souvent considéré comme plus précis en termes de détection de services et de versions. Il peut effectuer des scans plus détaillés et fournir des informations approfondies sur les services qui tournent sur les ports ouverts. | Masscan, en raison de sa vitesse élevée, peut sacrifier un peu de précision. Il peut manquer certaines informations détaillées sur les services.  |
| <b>Utilisation</b> | Nmap est polyvalent et offre une grande flexibilité. Il peut être utilisé pour divers types de scans, y compris des scans de vulnérabilités, des scans de scripts personnalisés, et bien plus encore.                                    | Masscan est principalement axé sur les scans de ports en masse. Il est moins polyvalent que Nmap, mais il excelle dans sa rapidité pour les scans massifs.  |

|                               | nmap  | Masscan  |
|-------------------------------|---|--|
| <b>Facilité d'utilisation</b> | Nmap offre une grande variété d'options de personnalisation, y compris des scripts de scan, des options de détection de versions, et des fonctionnalités avancées de configuration des paquets. | Masscan, en comparaison, est plus limité en termes d'options de personnalisation. Il est conçu pour être simple et rapide, mais cela peut se traduire par une moindre flexibilité pour certaines tâches spécifiques. |
| <b>Personnalisation</b>       | Nmap a une courbe d'apprentissage plus douce en raison de sa documentation détaillée et de son interface utilisateur conviviale.  | Masscan, bien que simple à utiliser, peut nécessiter une certaine adaptation en raison de sa rapidité et de son approche axée sur les scans de ports en masse.   |

# Scan des vulnérabilités

---

Le scan de vulnérabilités est une étape cruciale dans la gestion de la sécurité, visant à identifier les faiblesses potentielles dans un système. Cette approche proactive permet de prendre des mesures préventives avant qu'elles ne soient exploitées par des attaquants. donc on va prendre comme exemple deux outils importants dans ce type de scan, Nessus et openvas.

Nessus : Un scanner de vulnérabilités largement utilisé, Nessus offre une base de données exhaustive et des fonctionnalités avancées. Il propose une version gratuite pour un usage personnel.

OpenVAS : En tant qu'alternative open source à Nessus, OpenVAS propose des fonctionnalités similaires de scan de vulnérabilités. Il est entièrement open source et gratuit.

|  | <b>Nessus</b>  | <b>OpenVAS</b>   |
|--|--|--|
| <b>Nature Open Source</b>                | À l'origine, Nessus était un outil open source, mais il est devenu un produit commercial avec une version gratuite pour un usage personnel et non commercial (Nessus Essentials). La version complète de Nessus, avec des fonctionnalités avancées, est payante. | OpenVAS est un outil de scan de vulnérabilités entièrement open source, ce qui signifie que toutes ses fonctionnalités sont gratuites et disponibles pour le grand public.   |
| <b>Base de Données de Vulnérabilités</b> | Nessus a une base de données de vulnérabilités très étendue et régulièrement mise à jour. La version payante offre des mises à jour plus fréquentes et un accès à une base de données plus complète.   | OpenVAS utilise une base de données de vulnérabilités publiques (NVT - Network Vulnerability Tests) qui est également mise à jour régulièrement. Cependant, certains utilisateurs estiment que la base de données de Nessus est plus complète. |
| <b>Interface Utilisateur</b>             | Nessus offre une interface utilisateur conviviale et intuitive, ce qui le rend accessible aux utilisateurs de tous niveaux de compétence. La version payante propose des fonctionnalités supplémentaires et une interface plus riche.                            | L'interface utilisateur d'OpenVAS est fonctionnelle, mais certains utilisateurs la trouvent moins conviviale par rapport à celle de Nessus. Cependant, des améliorations ont été apportées au fil du temps.                                    |

|                                | <b>Nessus</b>   | <b>OpenVAS</b>   |
|--------------------------------|---|--|
| <b>Performances et Vitesse</b> | La version payante de Nessus est souvent considérée comme plus rapide que la version gratuite (Nessus Essentials). Cela est dû en partie à des fonctionnalités avancées telles que le pré-analyse et l'optimisation des performances. | OpenVAS peut être plus lent que Nessus dans certains cas, en particulier lorsqu'il s'agit de traiter un grand nombre d'actifs. Cependant, des optimisations ont également été apportées pour améliorer les performances. |
| <b>Coût</b>                    | Nessus propose une version gratuite (Nessus Essentials) pour un usage personnel et non commercial. La version professionnelle est payante, avec des fonctionnalités avancées.   | OpenVAS est entièrement gratuit et open source. Il ne nécessite pas de coût d'acquisition, mais peut nécessiter des ressources matérielles supplémentaires pour gérer des analyses importantes.                          |

# nmap

Pour cette partie on va utiliser nmap, on l'a installé simplement avec la commande suivante:  
Sudo apt install nmap

**Pourquoi on a choisi nmap? Il permet de faire quoi?**  
Nmap, abréviation de "Network Mapper", est un outil de balayage réseau gratuit et open-source utilisé pour découvrir les hôtes et les services sur un réseau informatique, créant ainsi une cartographie du réseau. Il est conçu pour analyser rapidement de grands réseaux, mais peut également être utilisé pour des hôtes individuels. Nmap envoie des paquets spécialement conçus à l'hôte cible, puis analyse les réponses pour déterminer des informations sur les hôtes et les services disponibles sur le réseau.



# Identification des ports ouverts

Pour identifier les ports ouverts sur chaque machine du réseau on a utilisé la commande suivante:

```
sudo nmap -sT -p 1-1000 192.168.1.1/24
```

Et c'est ce qu'on a obtenu:

```
mohamed@modell:~$ sudo nmap -sT -p 1-1000 192.168.1.1/24
[sudo] Mot de passe de mohamed :
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-13 20:53 CET
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0065s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
MAC Address: 1C:61:B4:3F:98:98 (Unknown)

Nmap scan report for 192.168.1.12
Host is up (0.044s latency).
All 1000 scanned ports on 192.168.1.12 are closed
MAC Address: BC:91:B5:FF:57:4B (Infinix mobility limited)

Nmap scan report for 192.168.1.13
Host is up (0.035s latency).
All 1000 scanned ports on 192.168.1.13 are closed
MAC Address: 60:14:B3:1A:C9:0C (CyberTAN Technology)

Nmap scan report for modell (192.168.1.16)
Host is up (0.00013s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http

Nmap done: 256 IP addresses (4 hosts up) scanned in 11.50 seconds
```

Adresse ip du réseau(routeur)

# Identification des ports ouverts

On peut aussi utiliser la commande suivante:

```
sudo nmap -sS -p 1-1000 192.168.1.1/24
```

La différence entre les deux c'est que la première commande fait une conversation TCP complète alors que la deuxième -sS (stands for stealthy) ne complète pas la conversation TCP

On peut de même faire identifier les ports ouverts sur une machine spécifique en remplacer l'adresse du réseau par l'adresse ip de la machine, comme le montre l'image ci dessous

```
mohamed@modell:~$ sudo nmap -sS -p 1-1000 192.168.1.16
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-17 02:33 CET
Nmap scan report for modell (192.168.1.16)
Host is up (0.0000090s latency).

Not shown: 997 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

# Détection du système d'exploitation et l'état de chaque machine du réseau

Pour déterminer le système d'exploitation et l'état de chaque machine du réseau, on a utilisé la commande suivante:

```
sudo nmap -O 192.168.1.1/24
```

Cette tentative de voir le s.e de la machine peut réussir ou échouer.

La commande qu'on a déjà mentionné peut nous fournir l'état des machines connectant au réseau mais on peut aussi utiliser la commande "sudo nmap -sV 192.168.1.1/24" pour effectuer une détection de version sur les services qui répondent, comme il est illustré dans cette image

```
Nmap scan report for modell (192.168.1.16)
Host is up (0.0000090s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 79.07 seconds
```



mohamed@modell:~



```
mohamed@modell:~$ sudo nmap -o 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-13 21:33 CET
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.022s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
5431/tcp  open  park-agent
MAC Address: 1C:61:B4:3F:98:98 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

TCP/IP fingerprint:

```
OS:SCAN(V=7.80%E=4%D=3/13%T=21%CT=1%CU=31944%PV=Y%DS=1%DC=D%G=Y%M=1C61B4%T
OS:M=65F20D8E%P=x86_64-pc-linux-gnu)SEQ(SP=9B%GCD=1%ISR=9D%TI=I%CI=I%II=RI%
OS:TS=U)OPS(O1=M5B4NW0$NN%02=M5B4NW0$NN%03=M5B4NW0$NN%04=M5B4NW0$NN%05=M5B4%
OS:NH0$NN%06=M5B4SN)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=2000)EC
OS:(R=Y%DF=N%T=1E%W=2000%O=M5B4NW0$NN%CC=%Q=)T1(R=Y%DF=N%T=1E%S=0%A=S+F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=N%T=1E%W=0%S=0%A=S+F=AS%O=M5B4NW0$NNNNLL
OS:ZLLL%RD=0%Q=)T4(R=Y%DF=N%T=1E%W=0%S=A%Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=N
OS:%T=1E%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%T=1E%W=0%S=A%A=Z%F=R%O=%R
OS:D=0%Q=)T7(R=Y%DF=N%T=1E%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=1E%IPL
OS:=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=1E%CD=S)
```

Network Distance: 1 hop

```
Nmap scan report for 192.168.1.11
Host is up (0.034s latency).
All 1000 scanned ports on 192.168.1.11 are closed
MAC Address: B8:94:E7:8B:A1:C2 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

```
Nmap scan report for 192.168.1.12
Host is up (0.037s latency).
All 1000 scanned ports on 192.168.1.12 are closed
MAC Address: BC:91:B5:FF:57:4B (Infinix mobility limited)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

```
Nmap scan report for 192.168.1.13
Host is up (0.030s latency).
All 1000 scanned ports on 192.168.1.13 are closed
MAC Address: 60:14:B3:1A:C9:0C (CyberTAN Technology)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

```
Host is up (0.000071s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
```

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 256 IP addresses (5 hosts up) scanned in 19.49 seconds

Nmap n'arrive pas à détecter le s.e de cet appareil

Par contre, il a réussi à détecter le s.e de cette machine

# Definition d'architecture

---

On détermine l'architecture du réseau en utilisant la commande :

Sudo nmap –traceroute 192.168.1.1/24

Elle utilise Nmap avec l'option --traceroute pour effectuer une trace du routeur (192.168.1.1) vers toutes les adresses IP dans la plage spécifiée (192.168.1.1/24).

Nmap commence par découvrir les hôtes actifs dans la plage d'adresses IP spécifiée (192.168.1.1/24) et puis il utilise la fonction de trace de route pour déterminer le chemin réseau (les sauts ou "hops") que les paquets prennent pour atteindre cet hôte depuis le routeur principal (192.168.1.1).

# Definition d'architecture

The screenshot shows a terminal window titled "mohamed@modell: ~" running on a dark-themed desktop environment. The terminal displays the output of several Nmap scans. The first scan is for the gateway at 192.168.1.1, showing open ports for FTP, Telnet, Domain, HTTP, and HTTPS, along with a park-agent service. The second scan is for host 192.168.1.13, which is up but has all scanned ports closed. The third scan is for host 192.168.1.14, also up but with all scanned ports closed. The terminal also shows the traceroute for each host, indicating the path and latency from the scanner's perspective.

```
mohamed@modell:~$ sudo nmap --traceroute 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-13 22:00 CET
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0070s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
5431/tcp  open  park-agent
MAC Address: 1C:61:B4:3F:98:98 (Unknown)

TRACEROUTE
HOP RTT      ADDRESS
1  7.05 ms  _gateway (192.168.1.1)

Nmap scan report for 192.168.1.13
Host is up (0.015s latency).
All 1000 scanned ports on 192.168.1.13 are closed
MAC Address: 60:14:B3:1A:C9:0C (CyberTAN Technology)

TRACEROUTE
HOP RTT      ADDRESS
1  15.01 ms  192.168.1.13

Nmap scan report for 192.168.1.14
Host is up (0.036s latency).
All 1000 scanned ports on 192.168.1.14 are closed
MAC Address: 9A:BE:A9:3E:76:23 (Unknown)

TRACEROUTE
HOP RTT      ADDRESS
1  36.47 ms  192.168.1.14

Nmap scan report for modell (192.168.1.16)
Host is up (0.0000080s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.27 seconds
mohamed@modell:~$
```

# Détection des vulnérabilités du système

---

On peut détecter les vulnérabilités du système en exécutant cette commande:

Sudo nmap –script vuln 192.168.1.1/24

elle utilise Nmap avec l'option --script vuln pour exécuter des scripts de vulnérabilité sur toutes les adresses IP dans la plage spécifiée (192.168.1.1/24).

Nmap commence par découvrir les hôtes actifs dans la plage d'adresses IP spécifiée et pour chaque hôte actif découvert, il exécute des scripts de vulnérabilité qui sont inclus dans la base de données de scripts de Nmap. Ces scripts sont conçus pour rechercher et détecter des vulnérabilités connues sur les services et les systèmes d'exploitation.

**NB:** On peut utiliser le logiciel OpenVAS pour effectuer la détection, mais nous avons choisi d'opter pour Nmap car il est polyvalent et capable de le faire

# Détection des vulnérabilités du système

```
mohamed@modell:~$ sudo nmap --script vuln 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-13 21:50 CET
Stats: 0:01:13 elapsed; 253 hosts completed (2 up), 2 undergoing Script Scan
NSE Timing: About 95.16% done; ETC: 21:51 (0:00:03 remaining)
Stats: 0:01:55 elapsed; 253 hosts completed (2 up), 2 undergoing Script Scan
NSE Timing: About 99.31% done; ETC: 21:51 (0:00:01 remaining)
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0044s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_sslv2-drown:
23/tcp    open  telnet
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
53/tcp    open  domain
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-aspx-debug: ERROR: Script execution failed (use -d to debug)
|_http-cookie-flags:
|  :
|  SessionID:
|    httponly flag not set
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
443/tcp   open  https
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-aspx-debug: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_ssl-ccs-injection: No reply from server (TIMEOUT)
|_sslv2-drown:
5431/tcp  open  park-agent
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
MAC Address: 1C:61:B4:3F:98:98 (Unknown)

Nmap scan report for 192.168.1.13
Host is up (0.041s latency).
All 1000 scanned ports on 192.168.1.13 are closed
MAC Address: 60:14:B3:1A:C9:0C (CyberTAN Technology)

Nmap scan report for modell (192.168.1.16)
Host is up (0.0000090s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_sslv2-drown:
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
```

Adresse ip du reseau  
(routeur)

Aucune vulnérabilité  
détectée

On a aussi rencontré  
ce problème avec  
clamav scripts de  
nmap

# Conclusion

Dans ces deux parties, nous avons analysé le trafic réseau et identifié les ports ouverts ainsi que les failles de sécurité. Cette analyse approfondie nous permet de renforcer la sécurité du réseau en prenant des mesures correctives et en mettant en place des politiques de sécurité appropriées.