

TP B1 : Wireshark

Hego Maxence



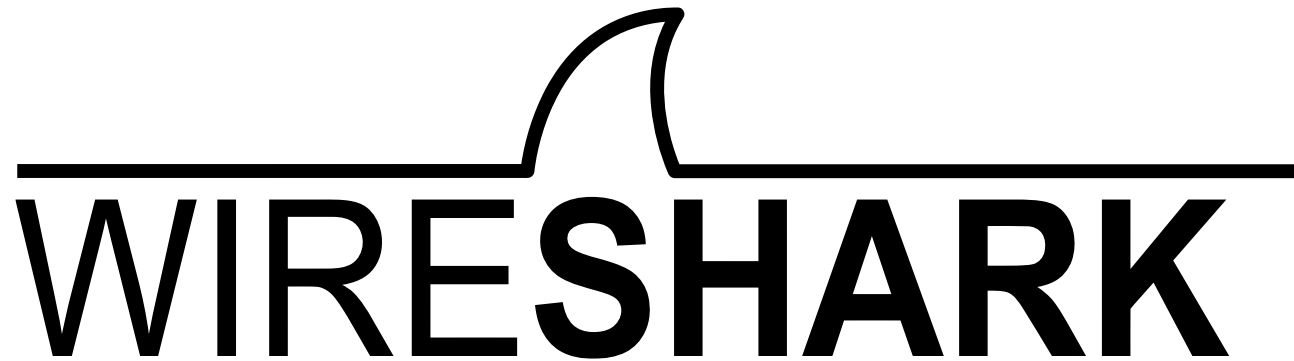
Installation :

Pour Windows :

- Télécharger Wireshark sur le site officiel : <https://www.wireshark.org/download.html>
- Exécuter en tant que administrateur : permet de capturer le trafic sur la ou les cartes réseaux

Pour Linux :

- `sudo apt install wireshark`
- `sudo wireshark`

The logo for Wireshark, featuring a stylized shark fin above the word "WIRESHARK" in a bold, sans-serif font.

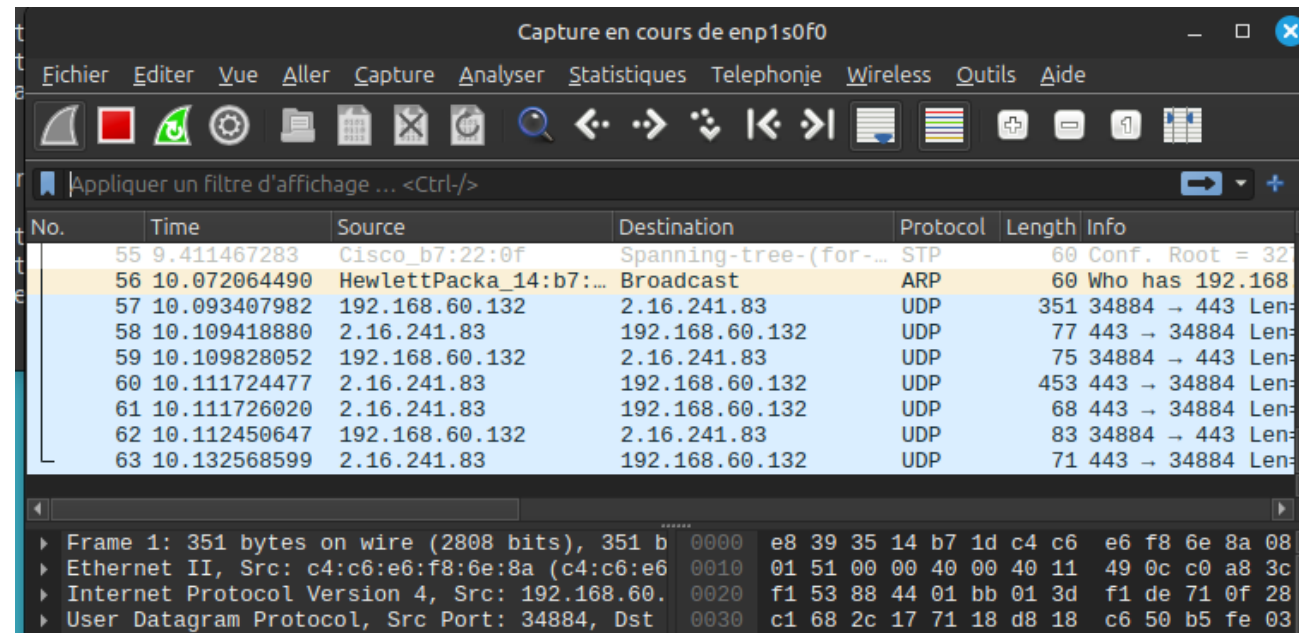
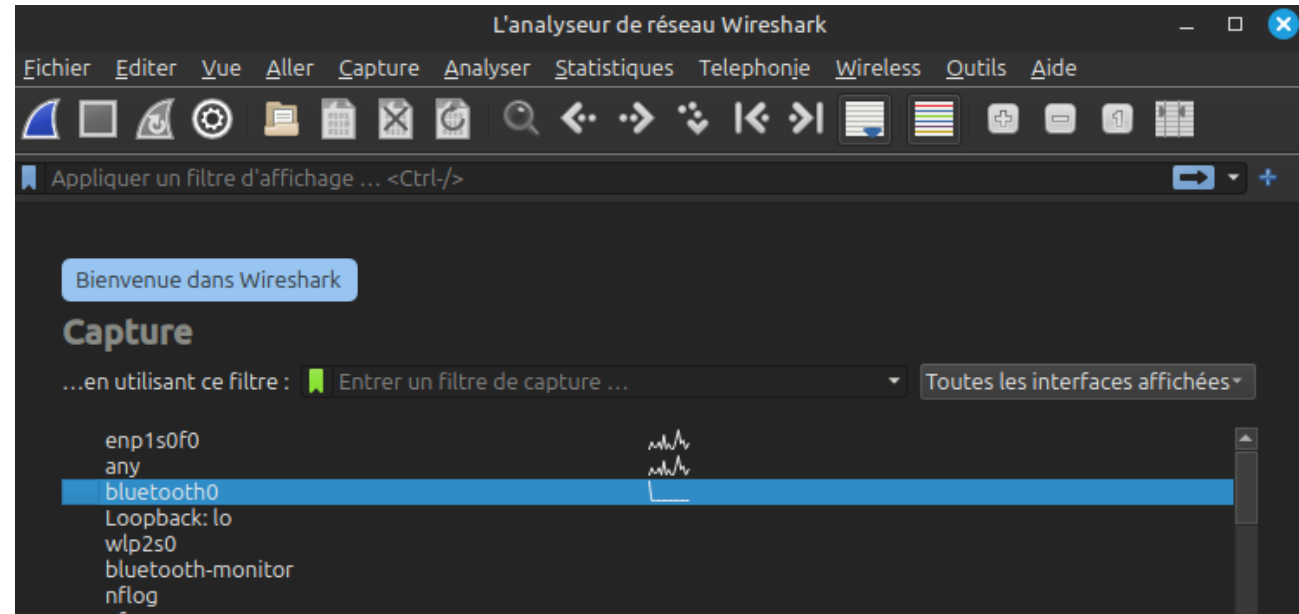
WIRESHARK

Utilisation :

Lancement wireshark :

- Une liste des interfaces apparaît
- On sélectionne notre carte réseau (ici : enp1sofo)

La capture de la trame fonctionne



Capture frame ICMP :

Ping de mon voisin :

- ping 192.168.60.133

Filtrage :

On entre le protocole que l'on souhaite analyser, ici ICMP

En cliquant sur une des frames, on obtient des informations sur cette frame.

The screenshot shows the Wireshark network protocol analyzer interface. At the top, the menu bar includes 'Fichier', 'Editer', 'Vue', 'Aller', 'Capture', 'Analyser', 'Statistiques', 'Telephonie', 'Wireless', 'Outils', and 'Aide'. Below the menu is a toolbar with various icons. A green filter bar contains the text 'icmp'. The main packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
118	13.290005705	192.168.60.132	192.168.60.133	ICMP	98	Echo (ping) request id=0
119	13.290375225	192.168.60.133	192.168.60.132	ICMP	98	Echo (ping) reply id=0
127	14.335880848	192.168.60.132	192.168.60.133	ICMP	98	Echo (ping) request id=0
128	14.336287158	192.168.60.133	192.168.60.132	ICMP	98	Echo (ping) reply id=0
131	15.359817594	192.168.60.132	192.168.60.133	ICMP	98	Echo (ping) request id=0

Below the packet list, a detailed view of frame 7410 is shown:

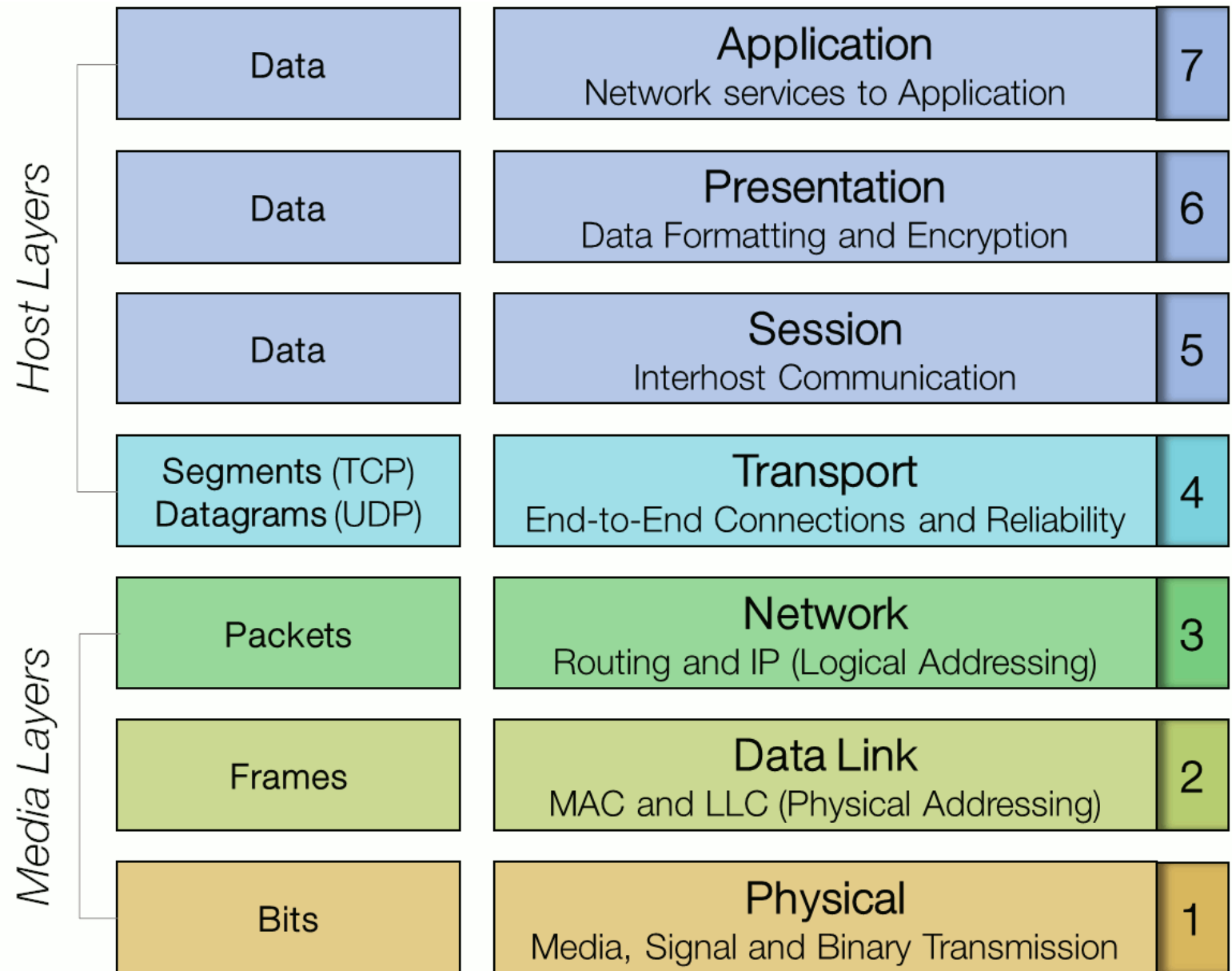
- Frame 7410: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp1s0f0, id 0
- Ethernet II, Src: c4:c6:e6:f8:6e:8a (c4:c6:e6:f8:6e:8a), Dst: Dell_87:15:c8 (98:e7:43:87:15:c8)
- Internet Protocol Version 4, Src: 192.168.60.132, Dst: 192.168.60.133
- Internet Control Message Protocol

Analyse d'une Frame : Le modèle OSI :

Le **modèle OSI** est un modèle conceptuel utiliser pour faciliter la compréhension des réseaux.

Il est divisé en **sept couches**, à chacune de ces couches est associé une **PDU ou Protocole Data Units**.

Une **Frame** est la **PDU** de la **seconde couche** de ce modèle.

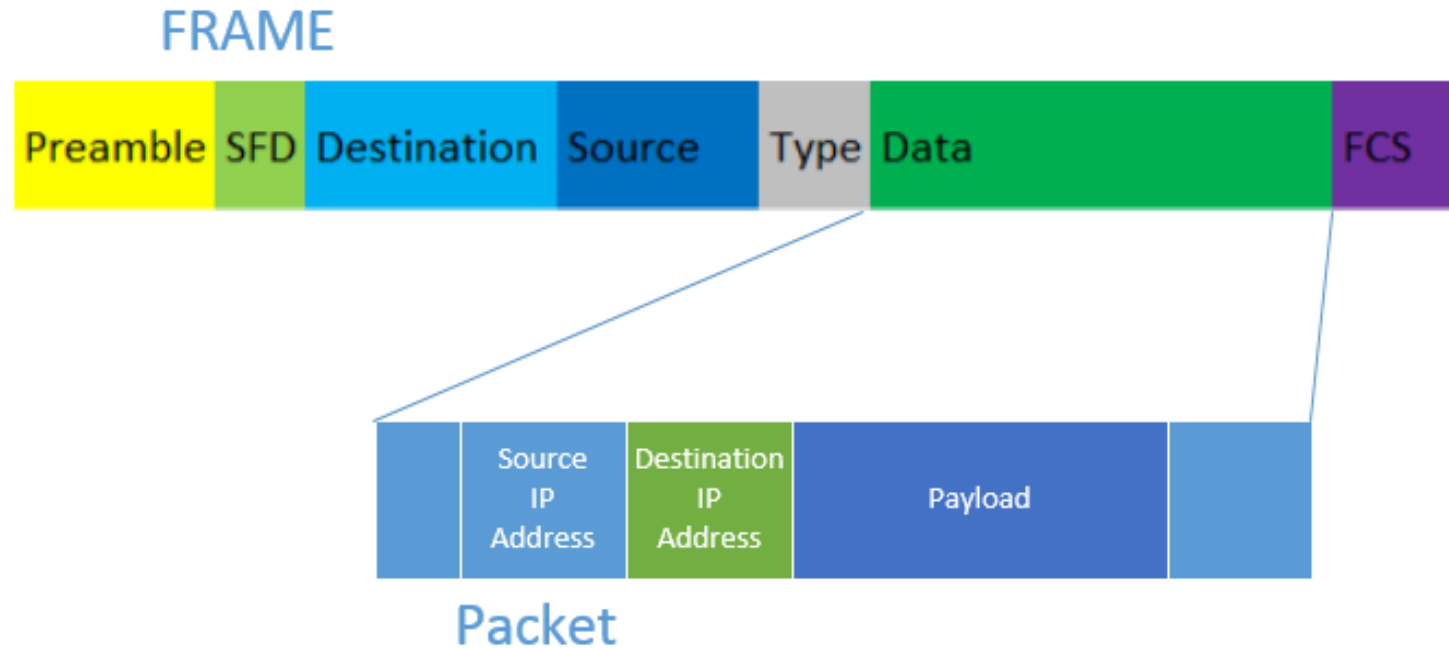


Composition d'une Frame:

Une frame = ethernet header + packet + ethernet trailer

Le packet est le PDU de la troisième couche du modèle OSI.

Le Header d'une frame est composé du :
Preamble, SFD, Destination, Source et Type.



Analyse de la frame dans Wireshark:

Dans Wireshark, on peut retrouver les éléments du Ethernet header d'une frame ICMP notamment l'adresse MAC source et destination.

Adresse Mac :

- L'adresse Mac est **un identifiant unique à chaque machine** composée de 12 caractères hexadécimaux.
- **Adresse mac source** : identifiant de la machine qui envoie le Ping
- **Adresse mac destination** : identifiant de la machine qui reçoit le Ping

```
▼ Ethernet II, Src: c4:c6:e6:f8:6e:8a (c4:c6:e6:f8:6e:8a), Dst: PCSSystemtec_af:7a:48 (08:00:27:af:7a:48)  
  ▶ Destination: PCSSystemtec_af:7a:48 (08:00:27:af:7a:48)  
  ▶ Source: c4:c6:e6:f8:6e:8a (c4:c6:e6:f8:6e:8a)  
  Type: IPv4 (0x0800)
```

La Mac adresse se trouve bien dans la partie Ethernet :

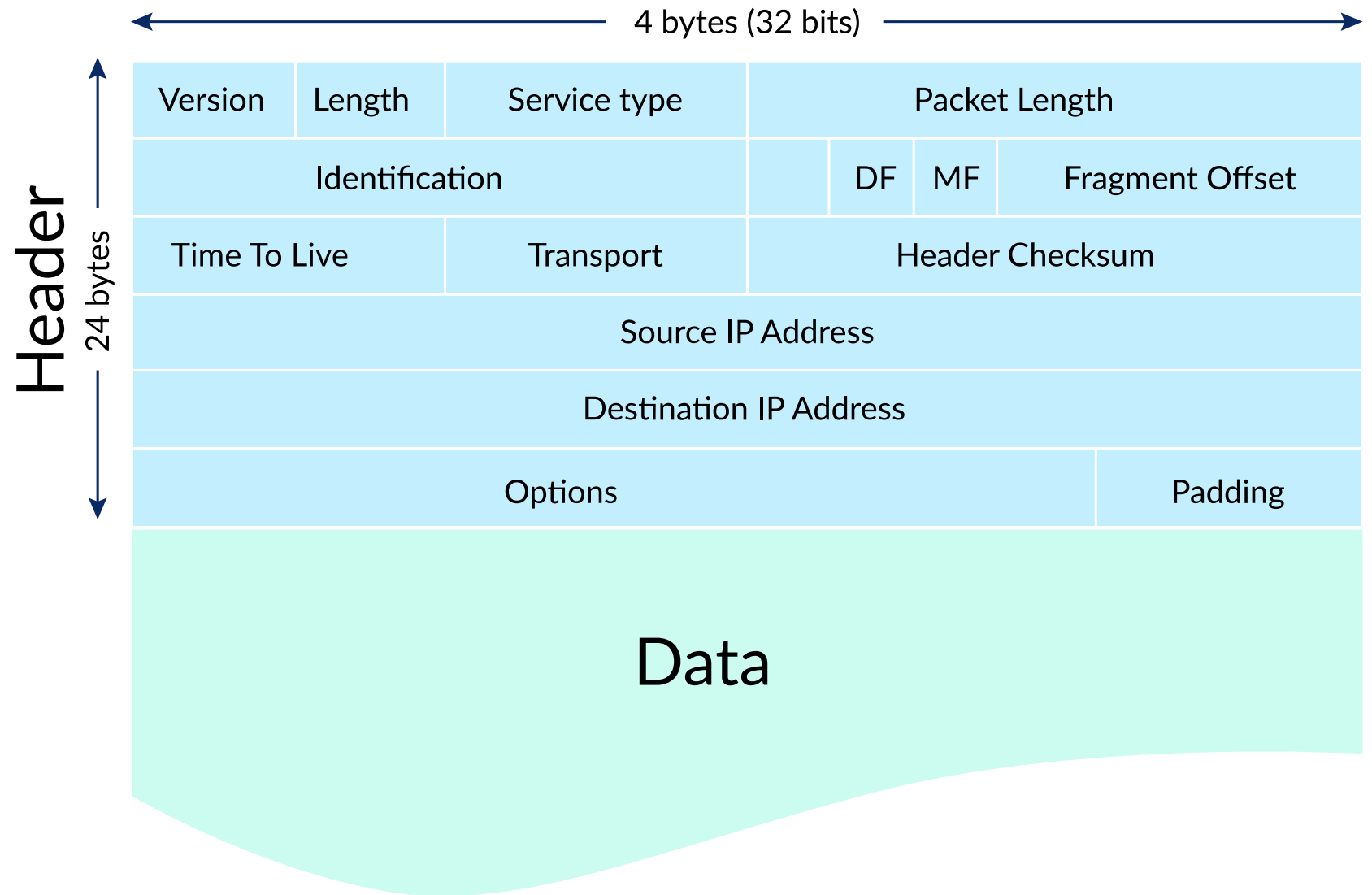
- **Destination** : 08 : 00 : 27 : af : 7a : 48
- **Source** : c4 : c6 : e6 : f8 : 6e : 8a

Composition d'un packet :

Un packet est composé du segment (PDU de la troisième couche) + de l'IPv4 header.

On retrouve dans l'IPv4 header :

L'adresse IP source et destinataire, le TTL, la taille des données ou encore le type de protocole (ICPM, TCP, UDP)



Analyse du packet dans wireshark : Adresse IP

Dans Wireshark, on peut retrouver les éléments de l'IPv4 Header.

Notamment l'adresse IP source et destinataire, le TTL ou la taille des données.

Adresse IP :

- **L'adresse IP** est l'identifiant que prennent les machines en utilisant le protocole IP.
- **L'adresse IP source** : IP de la machine qui envoie le PING
- **L'adresse IP destination** : IP de la machine qui reçoit le PING

```
▼ Internet Protocol Version 4, Src: 192.168.60.132, Dst: 192.168.60.56
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x9752 (38738)
  ▶ 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0xa949 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.60.132
    Destination Address: 192.168.60.56
```

On retrouve l'adresse IP dans l'IPv4 Header

Source Address: 192.168.60.132
Destination Address: 192.168.60.56

Analyse du packet dans wireshark : TTL et numéros de frame

Le TTL ou Time To Live et le numéro de frame font tout les deux parties de l'IPV4 header.

Time To Live :

- Le TTL ou Time To Live sert à déterminer la durée de vie d'un paquet sur le réseau.
- Il permet d'éviter que des paquets inutiles occupent la bande passante.

```
Time to Live: 64
```

identifiant :

- L'identifiant permet de repérer entre autres les anomalies ou les paquets manquants lors d'une transmission

```
Identification: 0x9752 (38738)
```

```
Frame Number: 18922
```

On peut trouver également le numéro de frame :
18922 frame capturé par Wireshark pendant l'analyse

Analyse du packet dans wireshark : Taille de la frame et des données

Avec wireshark on retrouve également la taille de la frame et la taille des données qui renseignent sur deux éléments différents.

Taille des données :

- La taille des données correspond à la taille de la charge utile ou payload du paquet
- Ce sont les données des couches supérieures encapsulées
- Si la taille dépasse 1500 octets le paquet peut être fragmenté

```
▼ Data (40 bytes)  
  Data: 101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637  
  [Length: 40]
```

Taille de la frame :

- La taille de la trame correspond à la taille total : payload + en têtes.
- Une frame a une taille minimum de 64 octet

```
▼ Frame 18922: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
```

Analyse du packet dans wireshark : Le code type ICMP

Le code type ICMP permet d'indiquer le type de message utilisé.

Il est souvent utilisé pour indiquer le type d'erreur rencontré lors d'un PING.

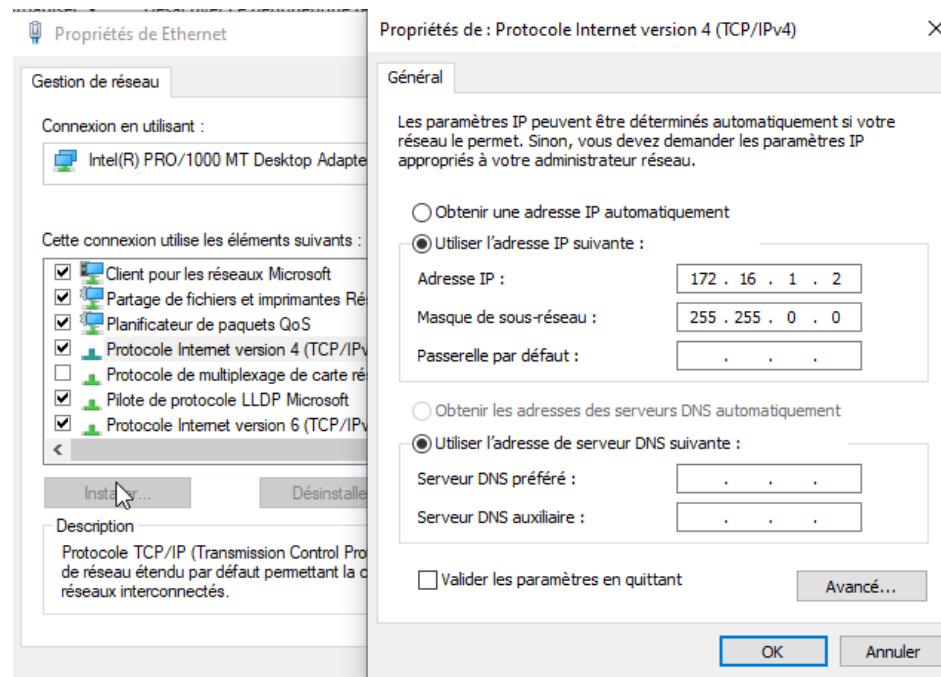
Code type ICMP :

- Le code type 0 0 correspond au code de réponse ping
- Le code type 8 0 correspond au code de requête ping
- Pour cette frame, comme il s'agit d'une echo request, le code type est 8 0.

```
Internet Control Message Protocol  
  Type: 8 (Echo (ping) request)  
  Code: 0
```

Modification des paramètres IP :

On modifie les paramètres IP pour que l'adresse IP de la VM soit 172



Après avoir modifier l'IP de la VM on Ping :

Le Ping affiche des erreurs, on voit ces erreurs dans wireshark

5948... 898.95701606/ 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=2/512, ttl=64 (no response found!)

5948... 899.981027558 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=3/768, ttl=64 (no response found!)

5948... 901.005023218 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=4/1024, ttl=64 (no response found!)

5948... 902.029005221 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=5/1280, ttl=64 (no response found!)

5948... 903.054002852 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=6/1536, ttl=64 (no response found!)

5948... 904.076990795 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=7/1792, ttl=64 (no response found!)

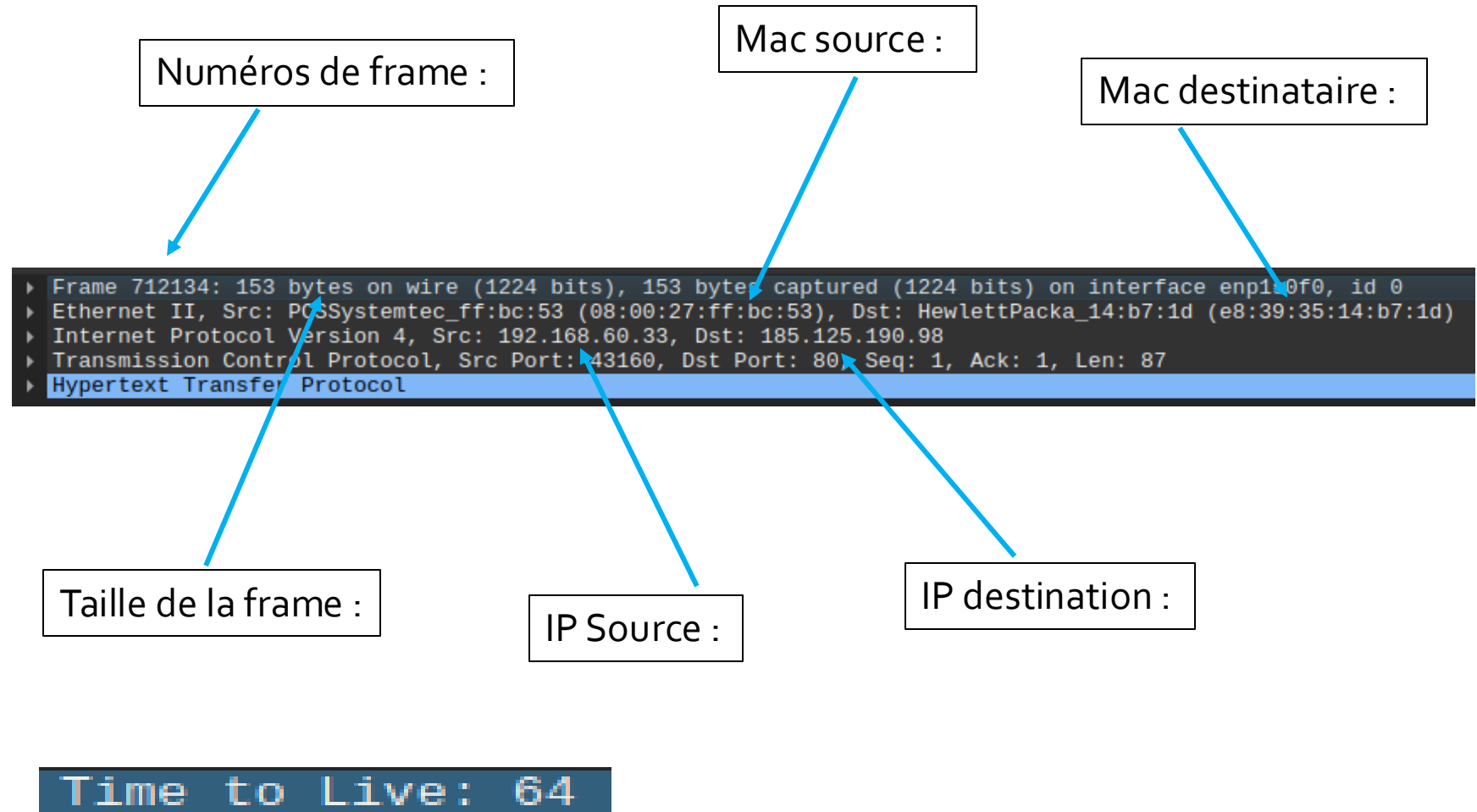
5948... 905.101042999 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=8/2048, ttl=64 (no response found!)

5948... 906.124992707 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=9/2304, ttl=64 (no response found!)

5948... 907.149020262 192.168.60.132 172.16.1.3 ICMP 98 Echo (ping) request id=0x246f, seq=10/2560, ttl=64 (no response found!)

Analyse du protocole http et https :

On analyse sur une frame les éléments clé du protocole http et https.



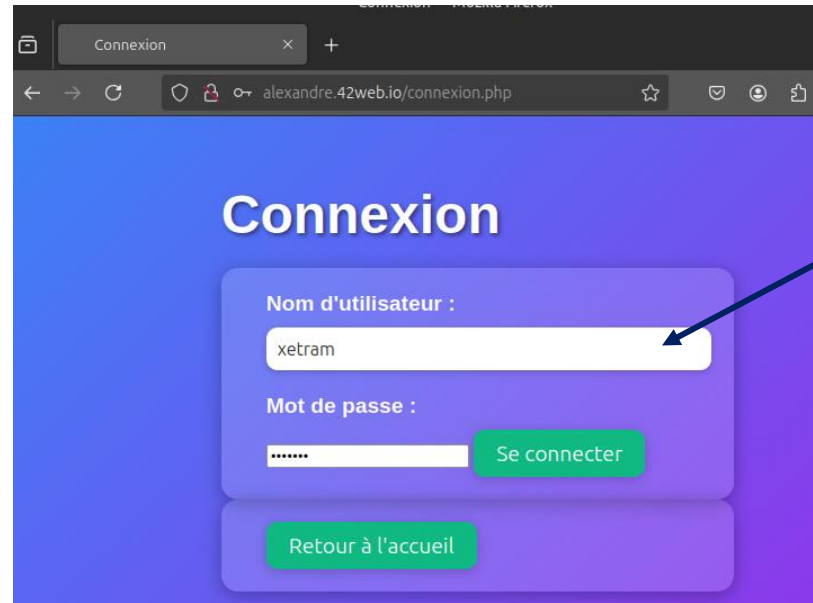
Analyse du protocole http :

On se connecte sur le site d'Alexandre.

Le site est en http et demande une connexion.

Avec Wireshark on peut retrouver le mot de passe et le login utilisé pour se connecter au site.

Le Protocole HTTP est risqué pour la protection de nos données.



On retrouve le nom d'utilisateur et le mot de passe du site en http :

```
20951 646.517770585 192.168.60.132 185.27.134.172 HTTP 729 POST /connexion.php HTTP/1.1 (application/
20953 646.706459393 185.27.134.172 192.168.60.132 HTTP 448 HTTP/1.1 302 Found
> Frame 20951: 729 bytes on wire (5832 bits), 729 bytes captured (5832 bits) on interface enp1s0f0, id 0
> Ethernet II, Src: c4:c6:e6:f8:6e:8a (c4:c6:e6:f8:6e:8a), Dst: HewlettPacka_14:b7:1d (e8:39:35:14:b7:1d)
> Internet Protocol Version 4, Src: 192.168.60.132, Dst: 185.27.134.172
> Transmission Control Protocol, Src Port: 59944, Dst Port: 80, Seq: 1461, Ack: 6385, Len: 663
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "nom_utilisateur" = "xetram"
  > Form item: "mot_de_passe" = "sio2024"
```

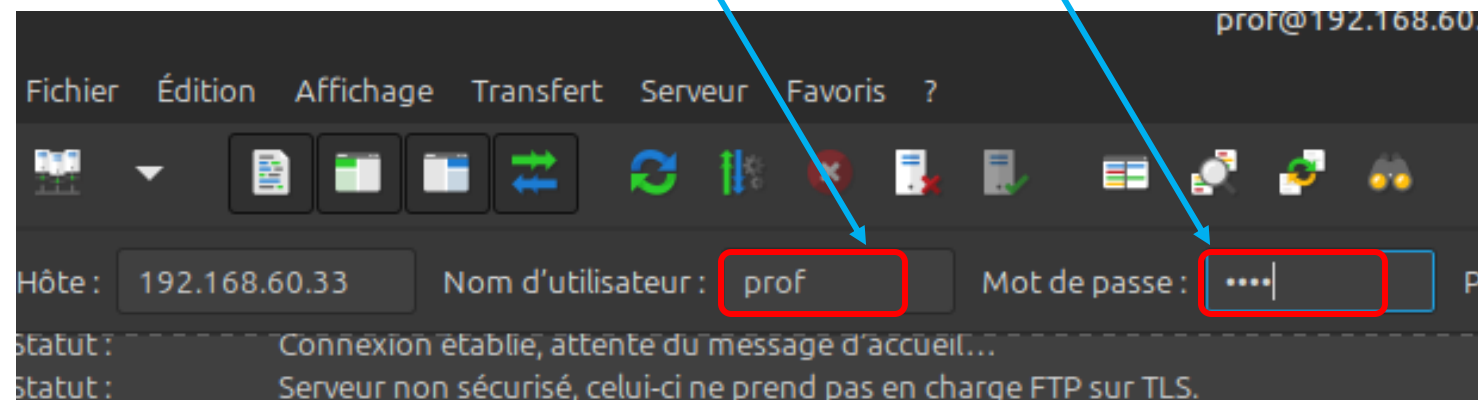
Analyse du protocole FTP :

On capture la trame de connexion d'un client vers le serveur FTP crée lors du TP précédent.

Le FTP ne chiffre pas les données, cela représente un risque pour la sécurité.

No.	Time	Source	Destination	Protocol	Length	Info
325	19.192801673	192.168.60.33	192.168.60.132	FTP	86	Response: 220 (vsFTPd 3.0.5)
327	19.192925698	192.168.60.132	192.168.60.33	FTP	76	Request: AUTH TLS
329	19.193151194	192.168.60.33	192.168.60.132	FTP	104	Response: 530 Please login with USER and PASS.
330	19.193206849	192.168.60.132	192.168.60.33	FTP	76	Request: AUTH SSL
331	19.193460829	192.168.60.33	192.168.60.132	FTP	104	Response: 530 Please login with USER and PASS.
346	20.539833665	192.168.60.132	192.168.60.33	FTP	77	Request: USER prof
347	20.540249621	192.168.60.33	192.168.60.132	FTP	100	Response: 332 Please specify the password.
349	20.540339030	192.168.60.132	192.168.60.33	FTP	77	Request: PASS prof
350	20.557716773	192.168.60.33	192.168.60.132	FTP	89	Response: 230 Login successful.

On retrouve le nom d'utilisateur et le mot de passe du serveur FTP



Le protocole SFTP :

Afin de protéger les données, on peut utiliser le protocole SFTP.

Voici les étapes pour passer du FTP au SFTP.

Pour plus d'informations, voir TP FTP

```
sio@sio2024:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
[sudo] Mot de passe de sio :
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/ssl/private/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:prof
Email Address []:
```

```
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

On génère une clé RSA qui sera utiliser pour le cryptage des données.

Commande : `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftd.pem`

On modifie le fichier `vsftpd.conf`