

DOCUMENTATION



SOMMAIRE

WIRESHARK - INTRODUCTION.....	3
Présentation de Wireshark.....	3
Questions.....	3
WIRESHARK - PARTIE I (ALCASAR).....	4
ARP.....	4
UDP.....	6
TCP.....	7
DIAGRAMME.....	8
WIRESHARK - PARTIE II (VM to VM).....	9
DHCP :.....	9
DNS.....	10
MDNS.....	12
FTP (sans TLS).....	13
FTP (avec TLS).....	14
WIRESHARK - PARTIE III (TSHARK).....	15
SSH commande tshark -i ens33.....	15
Samba commande tshark -i ens33 -Y 'smb2'.....	17
DHCP.....	19

WIRESHARK - INTRODUCTION

Présentation de Wireshark

Wireshark est un **analyseur de protocoles réseau** qui permet d'inspecter les données circulant sur un réseau informatique. Il **capture et affiche les paquets de données en temps réel**, offrant des outils puissants pour **l'analyse et le diagnostic des réseaux**. Wireshark est utilisé pour :

- **Dépannage des réseaux** : Identifier et résoudre les problèmes de performance et de configuration.
- **Sécurité des réseaux** : Analyser les trafics suspects pour détecter des intrusions ou des vulnérabilités.
- **Développement et test des protocoles** : Vérifier la conformité et l'efficacité des nouvelles implémentations de protocoles.

Questions

1. Quelle est la différence entre une trame et un paquet ?

- **Trame** : Une trame est une **unité de données de la couche liaison** (couche 2 du modèle OSI). Elle **inclut une en-tête, une charge utile** (données) et une **séquence de fin de trame** (FCS). La trame est **encapsulée dans les protocoles de la couche liaison**, comme Ethernet.
- **Paquet** : Un paquet est une **unité de données de la couche réseau** (couche 3 du modèle OSI). Il **inclut une en-tête contenant des informations de routage et une charge utile** (données). Les paquets sont **encapsulés dans les protocoles de la couche réseau**, comme IP.

2. Qu'est-ce que le format pcap/pcapng ?

- **Format pcap** : Le format de capture de paquets (pcap) est un **format de fichier standard pour l'enregistrement des données** capturées par les analyseurs réseau. Il **stocke les paquets capturés de manière séquentielle** et est largement **utilisé par les outils d'analyse** comme Wireshark.
- **Format pcapng** : Le format pcap Next Generation (pcapng) est une **version améliorée de pcap**, offrant des **fonctionnalités supplémentaires** telles que le support multi-interface, des options de métadonnées étendues et une meilleure structuration des fichiers de capture.

WIRESHARK - PARTIE I (ALCASAR)

ARP

ARP (Address Resolution Protocol) est un protocole réseau qui est utilisé pour convertir une adresse IP (Internet Protocol) en une adresse MAC (Media Access Control) physique. C'est un élément clé de la communication réseau, car il permet aux appareils de se trouver et de communiquer entre eux.

```
▶ Frame 447: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{A58FBF20-28:
▼ Ethernet II, Src: AzureWaveTec_09:16:62 (10:68:38:09:16:62), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: AzureWaveTec_09:16:62 (10:68:38:09:16:62)
  Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: AzureWaveTec_09:16:62 (10:68:38:09:16:62)
  Sender IP address: 10.10.36.190
  Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
  Target IP address: 10.10.2.68
```

Analyse du Paquet ARP (Paquet n°447)

1. **Couche 1 - Physique (Frame) :**
 - **Frame 447 :** 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
2. **Couche 2 - Liaison de Données (Ethernet II) :**
 - **Destination :** ff:ff:ff:ff:ff:ff (Broadcast)
 - **Source :** 68:38:09:16:62:0A (AzureWaveTec_09:16:62)
 - **Type :** ARP (0x0806)
3. **Couche 3 - Réseau (ARP) :**
 - **Protocole ARP :**
 - **Type de Matériel :** Ethernet (1)
 - **Type de Protocole :** IPv4 (0x0800)
 - **Taille du Matériel :** 6
 - **Taille du Protocole :** 4
 - **Opcode :** request (1)
 - **Adresse MAC Source :** 68:38:09:16:62:0A
 - **Adresse IP Source :** 10.10.36.190
 - **Adresse MAC Cible :** 00:00:00:00:00:00 (Requête ARP)
 - **Adresse IP Cible :** 10.10.2.6

Ce paquet ARP est une **requête envoyée en broadcast pour découvrir l'adresse MAC correspondant à l'adresse IP 10.10.2.68**. L'appareil émetteur à l'adresse MAC 68:38:09:16:62:0A et l'adresse IP 10.10.36.190.

UDP

UDP (User Datagram Protocol) est un protocole de transport utilisé pour envoyer des données sur Internet. Il est différent du protocole TCP (Transmission Control Protocol) car il n'établit pas de connexion fiable avec l'appareil destinataire avant d'envoyer des données. Au lieu de cela, UDP envoie simplement des paquets de données, appelés datagrammes, et espère qu'ils atteindront leur destination.

```
448 11.842019 10.10.0.104 255.255.255.255 UDP 292 45357 → 62976 Len=250
449 11.944342 fe80::2ad:24ff:fe36... ff02::1 UDP 328 49900 → 62976 Len=266

Frame 448: 292 bytes on wire (2336 bits), 292 bytes captured (2336 bits) on interface \Device\NPF_{A58FBF26...}
Ethernet II, Src: DLinkInterna_36:99:40 (00:ad:24:36:99:40), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: DLinkInterna_36:99:40 (00:ad:24:36:99:40)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.10.0.104, Dst: 255.255.255.255
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 278
  Identification: 0x8620 (34336)
  010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: UDP (17)
  Header Checksum: 0xa945 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.10.0.104
  Destination Address: 255.255.255.255
User Datagram Protocol, Src Port: 45357, Dst Port: 62976
  Source Port: 45357
  Destination Port: 62976
  Length: 258
  Checksum: 0x0f98 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 119]
  [Timestamps]
  UDP payload (250 bytes)
Data (250 bytes)
  Data [truncated]: 50052f2f384761774367414b306b4e706c4143676f4161414141414155416b674142414170454c55784a54
  [Length: 250]
```

Couche 2 (Liaison de Données) :

- Adresse MAC Source : **ad:24:36:99:40**
- Adresse MAC Destination : **ff:ff:ff:ff:ff:ff** (Broadcast)

Couche 3 (Réseau) :

- Adresse IP Source : **10.10.0.104**
- Adresse IP Destination : **255.255.255.255** (Broadcast)

Couche 4 (Transport) :

- Port Source : 45357
- Port Destination : 62976

Couche 7 (Application) :

- Données : 250 bytes (exprimées en hexadécimal)

TCP

414	10.00/151	10.10.36.190	157.240.190.63	TCP	54 5404/ → 443 [ACK] Seq=70 ACK=
439	11.333426	162.254.197.38	10.10.36.190	TCP	54 27030 → 54592 [ACK] Seq=97 Ac
8	0.462153	10.10.36.190	34.110.207.168	TLSv1.2	93 Application Data
10	0.471238	34.110.207.168	10.10.36.190	TLSv1.2	93 Application Data
31	2.599175	151.236.217.85	10.10.36.190	TLSv1.2	88 Application Data

▶	Frame 439: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{A58FBF20-2}
▼	Ethernet II, Src: Intel_3a:2e:49 (68:05:ca:3a:2e:49), Dst: AzureWaveTec_09:16:62 (10:68:38:09:16:62)
▶	Destination: AzureWaveTec_09:16:62 (10:68:38:09:16:62)
▶	Source: Intel_3a:2e:49 (68:05:ca:3a:2e:49)
	Type: IPv4 (0x0800)
▼	Internet Protocol Version 4, Src: 162.254.197.38, Dst: 10.10.36.190
	0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
▶	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
	Total Length: 40
	Identification: 0xcd1 (52945)
▶	010. = Flags: 0x2, Don't fragment
	...0 0000 0000 0000 = Fragment Offset: 0
	Time to Live: 121
	Protocol: TCP (6)
	Header Checksum: 0x9c11 [validation disabled]
	[Header checksum status: Unverified]
	Source Address: 162.254.197.38
	Destination Address: 10.10.36.190
▼	Transmission Control Protocol, Src Port: 27030, Dst Port: 54592, Seq: 97, Ack: 55, Len: 0
	Source Port: 27030
	Destination Port: 54592
	[Sequence index: 22]

Ethernet II (Couche 2 - Liaison de données)

- Adresse MAC source : **Intel_3a:2e:49 (68:05:ca:3a:2e:49)**
- Adresse MAC de destination : **AzureWaveTec_09:16:62 (10:68:38:09:16:62)**
- Type : **IPv4 (0x0800)**

Protocole Internet Version 4 (IPv4) (Couche 3 - Réseau)

- TTL (Time to Live) : **121**
- Protocole : **TCP (6)**
- Adresse IP source : **162.254.197.38**
- Adresse IP de destination : **10.10.36.190**

Protocole de contrôle de transmission (TCP) (Couche 4 - Transport)

- Port source : **27030**
- Port de destination : **54592**

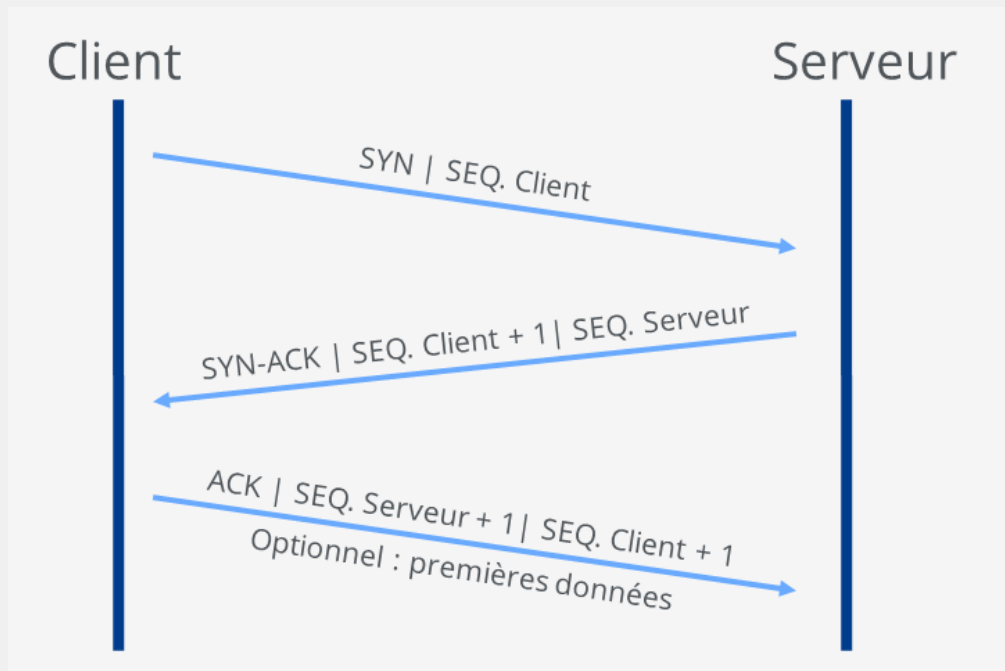
Adresses MAC sources : **68:05:ca:3a:2e:49**

Adresses IP sources : **162.254.197.38**

Adresses MAC de destination : **10:68:38:09:16:62**

Adresses IP de destination : **10.10.36.190**

DIAGRAMME



Le diagramme illustre les trois étapes essentielles du mécanisme de connexion TCP entre un client et un serveur.

1. **SYN (Synchronize) :**

- Client → Serveur : Le client **envoie un paquet SYN** pour initier la connexion, incluant un numéro de séquence initial (SEQ. Client).

2. **SYN-ACK (Synchronize-Acknowledge) :**

- Serveur → Client : Le serveur **reçoit le paquet SYN et répond avec un paquet SYN-ACK**. Ce paquet contient le numéro de séquence initial du serveur (SEQ. Serveur) et un accusé de réception pour le SYN du client (SEQ. Client + 1).

3. **ACK (Acknowledge) :**

- Client → Serveur : Le **client accuse réception du paquet SYN-ACK en envoyant un paquet ACK**. Ce paquet contient l'accusé de réception pour le SYN-ACK du serveur (SEQ. Serveur + 1) et le numéro de séquence du client (SEQ. Client + 1). À ce stade, la connexion est établie, et le client peut optionnellement envoyer les premières données.

Ce processus assure que les deux parties sont synchronisées et prêtes à échanger des données de manière fiable.

WIRESHARK - PARTIE II (VM to VM)

DHCP :

No.	Time	Source	Destination	Protocol	Length	Info
6	22.108454	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf8da7e76
9	22.108976	192.168.65.133	192.168.65.10	DHCP	342	DHCP Offer - Transaction ID 0xf8da7e76
10	22.109695	0.0.0.0	255.255.255.255	DHCP	348	DHCP Request - Transaction ID 0xf8da7e76
11	22.111272	192.168.65.133	192.168.65.10	DHCP	342	DHCP ACK - Transaction ID 0xf8da7e76
15	22.507412	192.168.65.254	192.168.65.131	DHCP	342	DHCP Offer - Transaction ID 0xf8da7e76

DHCP Discover

- **Description:** Le client, qui n'a pas encore d'adresse IP, envoie une requête DHCP Discover pour trouver un serveur DHCP sur le réseau.
- **Adresse de Destination:** (diffusion).

DHCP Offer

- **Description:** Un serveur DHCP répond à la requête Discover en proposant une adresse IP au client.
- **Adresse IP du Serveur:** Par exemple, 192.168.65.133.
- **Adresse IP Proposée au Client:** Par exemple, 192.168.65.10.

DHCP Request

- **Description:** Le client accepte l'offre en envoyant une requête DHCP Request pour la confirmer.
- **Adresse Source du Client:** 0.0.0.0.
- **Adresse de Destination:** 255.255.255.255 (diffusion).

DHCP ACK

- **Description:** Le serveur DHCP confirme l'attribution de l'adresse IP au client en envoyant une réponse DHCP ACK.
- **Adresse IP du Serveur:** Par exemple, 192.168.65.133.
- **Adresse IP Attribuée au Client:** Par exemple, 192.168.65.10.

DNS

No.	Time	Source	Destination	Protocol	Length	Info
18	1.328516	192.168.65.10	192.168.65.133	DNS	75	Standard query 0x9135 A www.example.com
19	1.330824	192.168.65.133	192.168.65.10	DNS	91	Standard query response 0x9135 A www.example.com A 192.168.65.133
29	1.332634	192.168.65.10	192.168.65.133	DNS	75	Standard query 0xfc75 AAAA www.example.com
31	1.332900	192.168.65.133	192.168.65.10	DNS	120	Standard query response 0xfc75 AAAA www.example.com SOA ns.example.com
50	6.027488	192.168.65.10	192.168.65.133	DNS	98	Standard query 0x4005 A www.example.com OPT
51	6.028266	192.168.65.133	192.168.65.10	DNS	130	Standard query response 0x4005 A www.example.com A 192.168.65.133 OPT

- **DNS Standard Query**

- **Paquet 18**
- **Source:** 192.168.65.10
- **Destination:** 192.168.65.133
- **Description:** Le client, dont l'adresse IP est 192.168.65.10, envoie une requête DNS de type A pour le domaine www.example.com au serveur DNS à l'adresse 192.168.65.133. La requête demande l'adresse IPv4 associée à ce domaine.
- **Détail:** La requête est identifiée par l'ID 0x9135.

- **DNS Standard Query Response**

- **Paquet 19**
- **Source:** 192.168.65.133
- **Destination:** 192.168.65.10
- **Description:** Le serveur DNS (192.168.65.133) répond à la requête du client. Il retourne l'adresse IPv4 associée à www.example.com, qui est 192.168.65.133.
- **Détail:** La réponse est identifiée par le même ID (0x9135) et contient un enregistrement de type A pour www.example.com.

- **DNS Standard Query (AAAA)**

- **Paquet 29**
- **Source:** 192.168.65.10
- **Destination:** 192.168.65.133
- **Description:** Le client (192.168.65.10) envoie une requête DNS de type AAAA pour www.example.com au serveur DNS (192.168.65.133). Cette requête demande l'adresse IPv6 du domaine.
- **Détail:** La requête est identifiée par l'ID 0xfc75.

- **DNS Standard Query Response (AAAA)**
 - **Paquet 31**
 - **Source:** 192.168.65.133
 - **Destination:** 192.168.65.10
 - **Description:** Le serveur DNS (192.168.65.133) répond à la requête du client. Il indique qu'il n'y a pas d'enregistrement AAAA pour www.example.com et retourne un enregistrement SOA (Start of Authority) pour ns.example.com.
 - **Détail:** La réponse est identifiée par le même ID (0xfc75) et contient un enregistrement SOA, indiquant que www.example.com n'a pas d'adresse IPv6.
- **DNS Standard Query (avec OPT)**
 - **Paquet 50**
 - **Source:** 192.168.65.10
 - **Destination:** 192.168.65.133
 - **Description:** Le client (192.168.65.10) envoie une nouvelle requête DNS de type A pour www.example.com au serveur DNS (192.168.65.133), en utilisant les options étendues DNS (EDNS0). Cette option permet des fonctionnalités avancées comme des tailles de message plus grandes.
 - **Détail:** La requête est identifiée par l'ID 0x4085 et inclut une section "OPT" pour les options EDNS0.
- **DNS Standard Query Response (avec OPT)**
 - **Paquet 51**
 - **Source:** 192.168.65.133
 - **Destination:** 192.168.65.10
 - **Description:** Le serveur DNS (192.168.65.133) répond à la requête du client en confirmant que www.example.com a l'adresse IPv4 192.168.65.133. La réponse utilise également les options étendues DNS (EDNS0).
 - **Détail:** La réponse est identifiée par le même ID (0x4085) et contient un enregistrement de type A pour www.example.com avec une section "OPT".

Résumé DNS

- **Requêtes de type A:** Le client demande l'adresse IPv4 de www.example.com et reçoit 192.168.65.133 en réponse.
- **Requêtes de type AAAA:** Le client demande l'adresse IPv6 de www.example.com, mais reçoit un enregistrement SOA indiquant qu'il n'y a pas d'adresse IPv6 disponible pour ce domaine.
- **Options DNS Étendues (EDNS0):** Le client utilise EDNS0 pour des fonctionnalités avancées, et le serveur répond en utilisant également EDNS0.

MDNS

No.	Time	Source	Destination	Protocol	Length	Info
22	3.739230	fe80::20c:29ff:fee3...	ff02::fb	MDNS	186	Standard query 0x0000 PTR _services._dns-sd._udp.local
23	3.739636	192.168.65.10	224.0.0.251	MDNS	249	Standard query 0x0000 PTR _services._dns-sd._udp.local
24	4.741398	fe80::20c:29ff:fee3...	ff02::fb	MDNS	186	Standard query 0x0000 PTR _services._dns-sd._udp.local
25	4.741572	192.168.65.10	224.0.0.251	MDNS	249	Standard query 0x0000 PTR _services._dns-sd._udp.local
28	6.741445	fe80::20c:29ff:fee3...	ff02::fb	MDNS	186	Standard query 0x0000 PTR _services._dns-sd._udp.local
29	6.741621	192.168.65.10	224.0.0.251	MDNS	249	Standard query 0x0000 PTR _services._dns-sd._udp.local

MDNS (Multicast DNS)

MDNS est un **protocole utilisé pour la résolution de noms de domaine** dans des réseaux locaux sans avoir besoin d'un serveur DNS centralisé. Il **permet aux appareils sur un réseau local de s'interroger les uns les autres** pour résoudre des noms d'hôtes en adresses IP.

- **Fonctionnalité Principale:** MDNS permet à un appareil de **demandeur l'adresse IP associée à un nom d'hôte local** (comme "imprimante.local") et de **recevoir une réponse directement des autres appareils** sur le réseau qui connaissent cette information.
- **Utilisation:** Couramment **utilisé dans des environnements où il n'y a pas de serveur DNS centralisé**, comme à la maison ou dans des petits bureaux. MDNS est souvent **utilisé pour la découverte de services réseau** comme les imprimantes, les services de partage de fichiers, et les appareils multimédia.

Paquets Capturés MDNS

1. Paquet 22

- **Source:** fe80::20c:29ff
- **Destination:** ff02::fb (IPv6 multicast)
- **Description:** Requête MDNS de type PTR cherchant les services disponibles (_services._dns-sd._udp.local) et requête pour services spécifiques comme _nvstream_dbd._tcp.local.

2. Paquet 23

- **Source:** 192.168.65.10
- **Destination:** 224.0.0.251 (IPv4 multicast)
- **Description:** Requête MDNS de type PTR cherchant les services disponibles (_services._dns-sd._udp.local) et requête pour services spécifiques comme _nvstream_dbd._tcp.local, _http._tcp.local, wsharkserver._http._tcp.local.

Résumé des Paquets Capturés

Les paquets capturés **montrent que votre appareil a envoyé des requêtes MDNS pour découvrir les services disponibles** sur le réseau local. Les requêtes **incluent des**

recherches pour des services généraux (comme `_services._dns-sd._udp.local`) et des services spécifiques (comme `_nvstream_dbd._tcp.local` et `_http._tcp.local`). Les réponses à ces requêtes permettent à votre appareil de découvrir quels services sont disponibles et où ils se trouvent sur le réseau local.

FTP (sans TLS)

No.	Time	Source	Destination	Protocol	Length	Info
149	7.328177	192.168.65.133	192.168.65.10	FTP	86	Response: 220 (vsFTPd 3.0.3)
164	10.112965	192.168.65.10	192.168.65.133	FTP	76	Request: USER ted
166	10.113311	192.168.65.133	192.168.65.10	FTP	100	Response: 331 Please specify the password.
180	10.912711	192.168.65.10	192.168.65.133	FTP	76	Request: PASS 123
182	10.939751	192.168.65.133	192.168.65.10	FTP	89	Response: 230 Login successful.
184	10.940279	192.168.65.10	192.168.65.133	FTP	72	Request: SYST
187	10.940560	192.168.65.133	192.168.65.10	FTP	85	Response: 215 UNIX Type: L8
188	10.940892	192.168.65.10	192.168.65.133	FTP	72	Request: FEAT
190	10.941152	192.168.65.133	192.168.65.10	FTP	81	Response: 211-Features:
191	10.941225	192.168.65.133	192.168.65.10	FTP	87	Response: EPRT
192	10.941435	192.168.65.133	192.168.65.10	FTP	110	Response: PASV

Détails des Paquets

1. **Paquet 149** : Réponse du serveur FTP avec le message d'accueil "**220 (vsFTPd 3.0.3)**".
2. **Paquet 164** : Requête USER avec "**ted**" comme nom d'utilisateur envoyé en clair.
3. **Paquet 166** : Réponse du serveur "**331 Please specify the password.**", demandant le mot de passe.
4. **Paquet 180** : Requête PASS avec "**123**" comme mot de passe envoyé en clair.
5. **Paquet 182** : Réponse "**230 Login successful.**", indiquant une connexion réussie.
6. **Paquets suivants** : Contiennent des commandes et réponses FTP supplémentaires (SYST, FEAT, EPRT, PASV).

Interprétation

Données Sensibles :

- Les informations de connexion, y compris le nom d'utilisateur (ted) et le mot de passe (123), sont **visibles en clair dans les paquets capturés**.

Récupération de Données :

- Oui, il est **possible de récupérer des données sensibles**, y compris les identifiants de connexion, car elles ne sont pas chiffrées.

Sécurité :

- L'**absence de chiffrement** signifie que toute personne ayant accès au réseau **peut intercepter et lire ces informations**.

FTP (avec TLS)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.65.1	192.168.65.10	SSH	90	Client: Encrypted packet (len=36)
2	0.000677	192.168.65.10	192.168.65.1	SSH	106	Server: Encrypted packet (len=52)
3	0.005092	192.168.65.10	192.168.65.1	SSH	98	Server: Encrypted packet (len=44)
4	0.005267	192.168.65.1	192.168.65.10	TCP	54	14662 → 22 [ACK] Seq=37 Ack=97 Win=512 Len=0
5	0.006690	192.168.65.10	192.168.65.1	SSH	130	Server: Encrypted packet (len=76)
6	0.061435	192.168.65.1	192.168.65.10	TCP	54	14662 → 22 [ACK] Seq=37 Ack=173 Win=511 Len=0

Détails des Paquets

1. Les paquets capturés **montrent des échanges sécurisés (TLS)**.
2. Les paquets sont marqués comme "Encrypted packet", **indiquant que le contenu des paquets est chiffré**.
3. Les numéros de paquets incluent 1, 2, 3, 4, 5, et 6, **montrant des échanges chiffrés entre le client et le serveur**.

Interprétation

Données Sensibles :

- Les informations de connexion, ainsi que **toutes les autres données échangées, sont chiffrées**.

Récupération de Données :

- Non, il n'est **pas possible de récupérer des données sensibles** à partir des paquets capturés, car ils sont chiffrés.

Sécurité :

- Le **chiffrement assure que même si les paquets sont interceptés**, les données qu'ils contiennent ne peuvent **pas être lues sans la clé de déchiffrement** appropriée

WIRESHARK - PARTIE III (TSHARK)

SSH commande tshark -i ens33

```
1 0.000000000 172.16.0.5 → 172.16.0.1 SSH 106 Server: Encrypted packet (len=52)
2 0.000220257 172.16.0.5 → 172.16.0.1 SSH 166 Server: Encrypted packet (len=112)
3 0.000409831 172.16.0.5 → 172.16.0.1 SSH 238 Server: Encrypted packet (len=184)
4 0.000419524 172.16.0.1 → 172.16.0.5 TCP 60 56966 → 22 [ACK] Seq=1 Ack=165 Win=4096 Len=0
5 0.000538176 172.16.0.5 → 172.16.0.1 SSH 210 Server: Encrypted packet (len=156)
6 0.000699997 172.16.0.5 → 172.16.0.1 SSH 166 Server: Encrypted packet (len=112)
7 0.000705529 172.16.0.1 → 172.16.0.5 TCP 60 56966 → 22 [ACK] Seq=1 Ack=505 Win=4095 Len=0
8 0.048894874 172.16.0.1 → 172.16.0.5 TCP 60 56966 → 22 [ACK] Seq=1 Ack=617 Win=4095 Len=0
9 0.749215415 172.16.0.5 → 172.16.0.1 SSH 178 Server: Encrypted packet (len=124)
```

1 0.000000000 172.16.0.5 → 172.16.0.1 SSH 106 Server: Encrypted packet (len=52)

Timestamp : 0.000000000 (ce paquet est le premier de la capture, donc son temps est 0)

Source IP : 172.16.0.5

Destination IP : 172.16.0.1

Protocol : SSH

Length : 106 bytes

Info : Serveur SSH envoie un paquet chiffré de 52 octets.

2 0.000220257 172.16.0.5 → 172.16.0.1 SSH 166 Server: Encrypted packet (len=112)

Timestamp : 0.000220257

Source IP : 172.16.0.5

Destination IP : 172.16.0.1

Protocol : SSH

Length : 166 bytes

Info : Serveur SSH envoie un paquet chiffré de 112 octets.

4 0.000419524 172.16.0.1 → 172.16.0.5 TCP 60 56966 → 22 [ACK] Seq=1 Ack=165 Win=4096 Len=0

Timestamp : 0.000419524

Source IP : 172.16.0.1

Destination IP : 172.16.0.5

Protocol : TCP

Length : 60 bytes

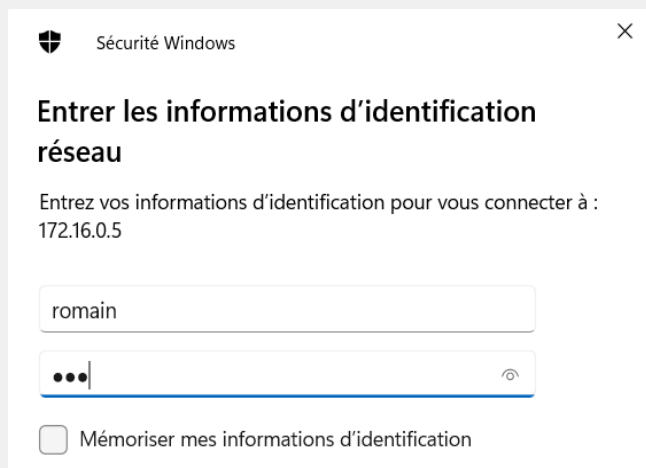
Info : Le client envoie un accusé de réception (ACK) pour le paquet reçu. Le port source est 56966 (port aléatoire côté client) et le port de destination est 22 (port SSH).

Les paquets **1, 2, 3, 5, et 6** sont des **paquets chiffrés envoyés par le serveur SSH (172.16.0.5) au client (172.16.0.1)**.

Les paquets **4, 7, et 8** sont des **accusés de réception (ACK) envoyés par le client (172.16.0.1) au serveur (172.16.0.5), confirmant la réception** des paquets précédents.

Les paquets **SSH** contiennent des **données chiffrées et ne peuvent pas être lus directement** sans la clé de déchiffrement appropriée.

Samba commande tshark -i ens33 -Y 'smb2'



Sécurité Windows

Entrer les informations d'identification réseau

Entrez vos informations d'identification pour vous connecter à :
172.16.0.5

romain

...

☐ Mémoriser mes informations d'identification

Tant qu'on ne **se connecte pas sur le serveur samba**, aucun trafic n'est détecté, une fois connecté on obtient :

```
5 4.556495118 172.16.0.1 → 172.16.0.5 SMB2 190 Create Request File: srvsvc
7 4.708936035 172.16.0.5 → 172.16.0.1 SMB2 210 Create Response File: srvsvc
8 4.708936035 172.16.0.1 → 172.16.0.5 SMB2 162 GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: srvsvc
10 4.709428295 172.16.0.5 → 172.16.0.1 SMB2 154 GetInfo Response
11 4.709741746 172.16.0.1 → 172.16.0.5 DCERPC 330 Bind: call_id: 2, Fragment: Single, 3 context items: SRVSVC V3.0 (32bit NDR), SRVSVC V3.0 (64bit ND
R), SRVSVC V3.0 (6cb71c2c-9812-4540-0300-000000000000)
12 4.709810881 172.16.0.5 → 172.16.0.1 SMB2 138 Write Response
13 4.710267577 172.16.0.1 → 172.16.0.5 SMB2 171 Read Request Len:1024 Off:0 File: srvsvc
14 4.711748951 172.16.0.5 → 172.16.0.1 SMB2 131 Read Response, Error: STATUS_PENDING
16 4.793892707 172.16.0.5 → 172.16.0.1 DCERPC 254 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280 max_recv: 4280, 3 results: Acceptance, Provi
der rejection, Negotiate ACK
17 4.794444524 172.16.0.1 → 172.16.0.5 SRVSVC 274 NetShareEnumAll request
18 4.795174887 172.16.0.5 → 172.16.0.1 SRVSVC 586 NetShareEnumAll response
19 4.795536911 172.16.0.1 → 172.16.0.5 SMB2 146 Close Request File: srvsvc
```

4.556495118 172.16.0.1 → 172.16.0.5 SMB2 190 Create Request File: srvsvc

Timestamp : 4.556495118

Source IP : 172.16.0.1

Destination IP : 172.16.0.5

Protocol : SMB2

Length : 190 octets

Info : Requête de création de fichier (Create Request) pour le fichier "srvsvc".

4.708936035 172.16.0.5 → 172.16.0.1 SMB2 210 Create Response File: srvsvc

Timestamp : 4.708936035

Source IP : 172.16.0.5

Destination IP : 172.16.0.1

Protocol : SMB2

Length : 210 octets

Info : Réponse à la demande de création de fichier (Create Response) pour le fichier "srvsvc".

4.709304353 172.16.0.1 → 172.16.0.5 SMB2 162 GetInfo Request

FILE_INFO/SMB2_FILE_STANDARD_INFO File: srvsvc

Timestamp : 4.709304353

Source IP : 172.16.0.1

Destination IP : 172.16.0.5

Protocol : SMB2

Length : 162 octets

Info : Demande d'informations (GetInfo Request) sur le fichier "srvsvc".

DHCP

87	17.407062028	0.0.0.0 → 255.255.255.255	DHCP 343 DHCP Discover	- Transaction ID 0xdc1d7565
88	17.407062409	192.168.204.254 → 192.168.204.150	DHCP 342 DHCP Offer	- Transaction ID 0xdc1d7565
89	17.407270305	0.0.0.0 → 255.255.255.255	DHCP 349 DHCP Request	- Transaction ID 0xdc1d7565
90	17.407270545	192.168.204.131 → 192.168.204.150	DHCP 342 DHCP Offer	- Transaction ID 0xdc1d7565
91	17.407498580	192.168.204.254 → 192.168.204.150	DHCP 342 DHCP ACK	- Transaction ID 0xdc1d7565
92	17.407663674	192.168.204.131 → 192.168.204.150	DHCP 342 DHCP ACK	- Transaction ID 0xdc1d7565

DHCP Discover

87. 17.407062028 0.0.0.0 -> 255.255.255.255 DHCP 343 DHCP Discover - Transaction ID 0xdc1d7565`

- Une requête **DHCP Discover** est envoyée par un client (adresse IP source `0.0.0.0` signifiant que le client n'a pas encore d'adresse IP) vers l'adresse de diffusion (`255.255.255.255`).

DHCP Offer

88. 17.407062409 192.168.204.254 -> 192.168.204.150 DHCP 342 DHCP Offer - Transaction ID 0xdc1d7565`

- Le serveur DHCP (IP `192.168.204.254`) répond avec une offre DHCP au client en utilisant l'adresse `192.168.204.150`.

- Cette offre correspond à la transaction ID `0xdc1d7565`.

DHCP Request

89. 17.407270305 0.0.0.0 -> 255.255.255.255 DHCP 349 DHCP Request - Transaction ID 0xdc1d7565`

- Le client DHCP envoie une requête DHCP (DHCP Request) pour demander la configuration offerte par le serveur.

- Utilise l'adresse source `0.0.0.0` et la destination de diffusion `255.255.255.255`.

DHCP Offer

90. 17.407270545 192.168.204.131 -> 192.168.204.150 DHCP 342 DHCP Offer - Transaction ID 0xdc1d7565`

- Une autre offre DHCP est envoyée par le serveur DHCP à l'adresse `192.168.204.150` en réponse à la requête précédente.

DHCP ACK

91. 17.407498580 192.168.204.254 -> 192.168.204.150 DHCP 342 DHCP ACK - Transaction ID 0xdc1d7565

- Le serveur DHCP envoie un accusé de réception (DHCP ACK) pour confirmer que l'adresse IP `192.168.204.150` a été allouée au client.

DHCP ACK

92. 17.407663674 192.168.204.131 -> 192.168.204.150 DHCP 342 DHCP ACK - Transaction ID 0xdc1d7565

- Une autre confirmation (DHCP ACK) est envoyée par le serveur DHCP à l'adresse `192.168.204.150`.

Résumé du processus DHCP

Discover : Le client envoie une requête pour découvrir les serveurs DHCP disponibles.

Offer : Le serveur DHCP répond avec une offre de configuration réseau.

Request : Le client demande l'adresse IP proposée dans l'offre.

ACK : Le serveur confirme la configuration attribuée.

Chaque étape est identifiée par une transaction ID unique