

WIRESHARK

Wireshark est un outil d'analyse réseau, c'est un analyseur qui permet de comprendre les mécanismes de fonctionnement des protocoles de communication transitant sur les réseaux. Wireshark est un logiciel libre et incontournable pour recueillir des informations sur le réseau. Son interface graphique permet de visualiser les données des paquets capturés. Wireshark est utilisé pour le dépannage réseau, l'analyse, le développement de logiciels et les protocoles de communication.

Le modèle de référence pour les communications réseau est le modèle **OSI** (Open Systems Interconnection). Ce modèle comporte 7 couches :

1. Couche physique : câbles, connecteurs, répéteurs, hubs.
2. Couche liaison de données : switches, bridges.
3. Couche réseau : routeurs.
4. Couche transport : protocoles TCP et UDP.
5. Couche session : gestion et terminaison des sessions entre applications (RPC, SMB).
6. Couche présentation : gère la compression et le chiffrement (SSL/TLS, JPEG, ASCII).
7. Couche application : fournit des services de communication à l'application. (HTTP, FTP, SMTP).

Quelle est la différence entre une trame et un paquet ?

- **Trame** : Une trame est une unité de données de la couche liaison de données (couche 2 du modèle OSI). Elle inclut l'en-tête de la couche liaison qui contient des informations comme les adresses MAC source et destination, ainsi que la charge utile (données).
- **Paquet** : Un paquet est une unité de données de la couche réseau (couche 3 du modèle OSI). Il inclut l'en-tête de la couche réseau qui contient des informations comme les adresses IP source et destination, ainsi que la charge utile.

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

- **PCAP (Packet Capture)** : Un format de fichier utilisé pour enregistrer les paquets capturés par des outils comme Wireshark. Il permet de sauvegarder et de partager les captures de trafic réseau.
- **PCAPNG (PCAP Next Generation)** : Une version améliorée du format PCAP, offrant des fonctionnalités supplémentaires telles que la capture de métadonnées, la compression des données, et le support multi-interface.

Installation et Utilisation de Wireshark

Partie 1 : Capturer et analyser des paquets ARP, UDP, TCP

- **Installation de Wireshark**
Mise à jour des paquets **apt update**
installation **apt install wireshark**

```
safia@debian: ~  
Paramétrage de libqt5widgets5:amd64 (5.15.8+dfsg-11) ...  
Paramétrage de qt5-gtk-platformtheme:amd64 (5.15.8+dfsg-11) ...  
Paramétrage de libqt5waylandclient5:amd64 (5.15.8-2) ...  
Paramétrage de libqt5multimedia5:amd64 (5.15.8-2) ...  
Paramétrage de libqt5printsupport5:amd64 (5.15.8+dfsg-11) ...  
Paramétrage de libqt5multimediawidgets5:amd64 (5.15.8-2) ...  
Paramétrage de libqt5multimediagsttools5:amd64 (5.15.8-2) ...  
Paramétrage de libqt5multimedia5-plugins:amd64 (5.15.8-2) ...  
Paramétrage de libqt5quick5:amd64 (5.15.8+dfsg-3) ...  
Paramétrage de libqt5svg5:amd64 (5.15.8-3) ...  
Paramétrage de libqt5waylandcompositor5:amd64 (5.15.8-2) ...  
Paramétrage de wireshark-qt (4.0.11-1~deb12u1) ...  
Paramétrage de wireshark (4.0.11-1~deb12u1) ...  
Paramétrage de qtwayland5:amd64 (5.15.8-2) ...  
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...  
Traitement des actions différées (« triggers ») pour shared-mime-info (2.2-1) ...  
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmul) ...  
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1) ...  
Traitement des actions différées (« triggers ») pour hicolor-icon-theme (0.17-2) ...  
Traitement des actions différées (« triggers ») pour gnome-menus (3.36.0-1.1) ...  
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u7) ...  
root@debian: /home/safia#
```

To direct input to this VM, move the mouse pointer inside or press

```
safia@debian: ~  
Outil de configuration des paquets  
Configuration de wireshark-common  
  
Dmccap can be installed in a way that allows members of the "wireshark" system  
group to capture packets. This is recommended over the alternative of running  
Wireshark/Tshark directly as root, because less of the code will run with  
elevated privileges.  
  
For more detailed information please see  
/usr/share/doc/wireshark-common/README.Debian.gz once the package is installed.  
  
Enabling this feature may be a security risk, so it is disabled by default. If  
in doubt, it is suggested to leave it disabled.  
  
Should non-superusers be able to capture packets?  
  
<Oui> <Non>
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

- Ajouter notre utilisateur au groupe Wireshark pour capturer des paquets sans privilèges root
sudo usermod -aG wireshark \$USER
sudo setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dmccap
 - Redémarrer la machine
sudo reboot

Ensuite, nous n'avons plus qu'à démarrer et lancer Wireshark pour capturer les paquets avec la commande :

- **wireshark**

```

/usr/sbin/usermod
/usr/share/bash-completion/completions/usermod
root@debian:/home/safia# /usr/sbin/usermod -aG wireshark safia
root@debian:/home/safia# chmod +x /usr/bin/dumpcap
root@debian:/home/safia# setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dumpcap
bash: setcap : commande introuvable
root@debian:/home/safia# apt update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
1 paquet peut être mis à jour. Exécutez « apt list --upgradable » pour le voir.
root@debian:/home/safia# apt install libcap2-bin
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
libcap2-bin est déjà la version la plus récente (1:2.66-4).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@debian:/home/safia# chmod +x /usr/bin/dumpcap
root@debian:/home/safia# setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dumpcap
bash: setcap : commande introuvable
root@debian:/home/safia# nano /etc/apt/sources.list
root@debian:/home/safia# find / -name setcap 2>/dev/null
/usr/sbin/setcap
root@debian:/home/safia# export PATH=$PATH:/sbin:/usr/sbin:/usr/local/sbin
root@debian:/home/safia# export PATH=$PATH:/sbin:/usr/sbin:/usr/local/sbin
root@debian:/home/safia# source ~/.bashrc
root@debian:/home/safia# wich setcap
bash: wich : commande introuvable
root@debian:/home/safia# which setcap
/sbin/setcap
root@debian:/home/safia# chmod +x /usr/bin/dumpcap
root@debian:/home/safia# setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dumpcap
root@debian:/home/safia# █

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.



1. Le protocole ARP (Address Resolution Protocol)

Une fois qu'on est sur l'interface graphique, on sélectionne l'interface réseau **ens33** pour capturer les paquets. Avec la commande **ip addr**, on peut lister les interfaces réseau et leurs configurations :

- Les requêtes ARP sont envoyées en broadcast, en diffusion à toutes les machines sur le réseau local, pour découvrir l'adresse MAC associée à une adresse IP spécifique. Par exemple pour l'adresse MAC 00:50:56:e6:3d:ac qui fait partie de la liste des paquets capturés dans les colonnes « source » et « destination ».
- Si on clique sur un paquet, on peut y voir les adresses MAC et IP dans les sections ethernet et ARP.

La requête ARP apparaît donc avec :

Ethernet II :

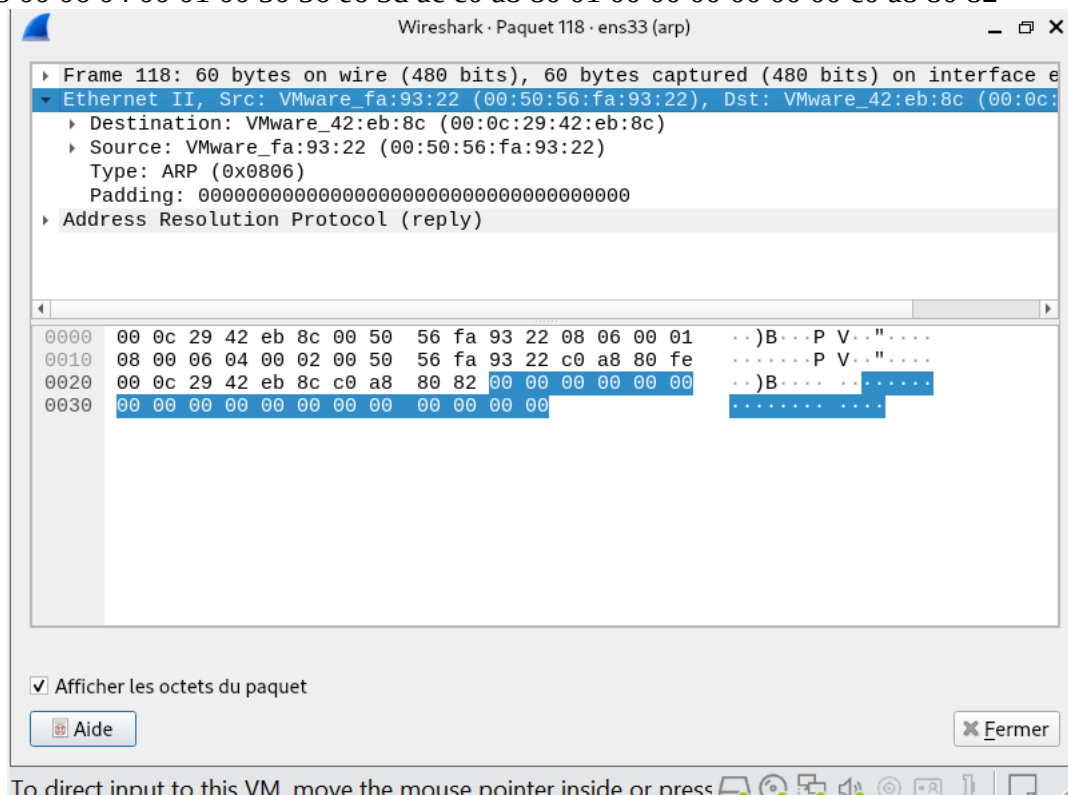
- MAC source** : VMware_e6:3d (00:50:56:e6:3d)
- Destination MAC** : Diffusion (ff:ff:ff:ff:ff:ff)

ARP :

- Type de matériel** : 0x0001 (Ethernet)
- Type de protocole** : 0x0800 (IPv4)
- Taille du matériel** : 6
- Taille du protocole** : 4
- Opcode** : 1 (demande)
- Adresse MAC de l'expéditeur** : 00:50:56:e6:3d
- Adresse IP de l'expéditeur** : 192.168.128.1
- Adresse MAC cible** : 00:00:00:00:00:00
- Adresse IP cible** : 192.168.128.130

Données Hexadécimales :

00 01 08 00 06 04 00 01 00 50 56 e6 3d ac c0 a8 80 01 00 00 00 00 00 00 c0 a8 80 82



- Utilisation du filtre ARP :
Pour le dépannage réseau, on peut diagnostiquer les problèmes de connectivité sur le réseau local. Par exemple, vérifier si les requêtes ARP reçoivent des réponses correctes et en temps opportun.
- Détection d'attaques ARP Spoofing :
Le filtre ARP peut aider à détecter les attaques d'usurpation d'ARP (ARP Spoofing). Ces attaques impliquent l'envoi de réponses ARP falsifiées pour diriger le trafic vers une adresse MAC malveillante.
- Vérification de la configuration réseau :
On peut analyser les paquets ARP pour vérifier la configuration des adresses IP et MAC sur le réseau. Cela peut aider à identifier les adresses IP en double ou les anomalies dans la table ARP.

2. Le protocole UDP

Pour capturer uniquement les paquets UDP (User Datagram Protocol), on sélectionne le filtre udp et l'interface réseau ens33. Le protocole UDP se situe au niveau de la couche 4 du modèle OSI qui concerne le transport.

Caractéristiques du protocole UDP :

- UDP est un protocole sans connexion, il n'y a pas d'établissement de connexion préalable entre l'émetteur et le récepteur avant l'envoi des données.
- Il est non fiable, il ne garantit pas la livraison des paquets. Ils peuvent être perdus, dupliqués ou livrés hors séquence. UDP n'a pas de mécanismes pour corriger ces problèmes.
- UDP a une latence faible, d'où une transmission rapide et continue des données (exemple streaming, jeux en ligne).
- Les données sont transmises sous forme de datagramme avec l'adresse source, l'adresse de destination, les données et une somme de contrôle pour l'intégrité des données.

The screenshot shows the Wireshark interface with a capture on the 'ens33' interface filtered for 'udp'. The packet list shows several SSDP (M-SEARCH) packets and one BROWSER packet. The packet details pane is expanded for the selected BROWSER packet (No. 118), showing the User Datagram Protocol (Source Port: 53888, Destination Port: 1900) and the Simple Service Discovery Protocol (SSDP) details. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
112	962.069989374	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
113	963.073872441	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
114	965.689160366	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
115	966.697186040	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
116	967.697784719	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
117	968.710479636	192.168.128.1	239.255.255.250	SSDP	217	M-SEARCH
118	977.841753246	192.168.128.1	192.168.128.255	BROWSER	243	Host Anno

Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface ens33
 Ethernet II, Src: VMware_c0:00:08 (00:50:56:00:00:08), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
 Internet Protocol Version 4, Src: 192.168.128.1, Dst: 239.255.255.250
 User Datagram Protocol, Src Port: 53888, Dst Port: 1900
 Source Port: 53888
 Destination Port: 1900
 Length: 183
 Checksum: 0x8c79 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 0]
 [Timestamps]
 UDP payload (175 bytes)
 Simple Service Discovery Protocol

0000 01 00 5e 7f ff fa 00 50 56 c0 00 00 00 00 00 00
 0010 00 cb 87 37 00 00 01 11 01 47 c0 00 00 00 00 00
 0020 ff fa d2 80 07 6c 00 b7 8c 79 4d 2e 00 00 00 00
 0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 00 00 00 00
 0040 4f 53 54 3a 20 32 33 39 2e 32 35 33 00 00 00 00
 0050 2e 32 35 30 3a 31 39 30 30 0d 0a 4d 58 3a 20
 0060 22 73 73 64 70 3a 64 69 73 63 6f 77 0a 4d 58 3a
 0070 0a 4d 58 3a 20 31 0d 0a 53 54 3a 20 64 69 61 6c
 0080 64 69 61 6c 2d 6d 75 6c 74 69 73 6f 2d 6f 72 67
 0090 2d 6f 72 67 3a 73 65 72 76 69 63 6f 6c 3a 31 0d
 00a0 6c 3a 31 0d 0a 55 53 45 52 2d 41 4d 20 47 6f
 00b0 20 47 6f 6f 67 6c 65 20 43 68 72 6f 32 35 2e
 00c0 32 35 2e 30 2e 36 34 32 32 2e 31 33 6e 64 6f
 00d0 6e 64 6f 77 73 0d 0a 0d 0a

La capture montre : Correspondant hexadécimal

Ethernet II :

- **MAC source** : VMware_c0:00:08 (00:50:56:c0:00:08)
- **Destination MAC** : Diffusion (ff:ff:ff:ff:ff)

IPv4 :

- **IP source** : 192.168.128.1
- **IP de destination** : 239.255.255.250

UDP :

- **Port source** : 53888
- **Port de destination** : 1900
- **Longueur** : 183
- **Somme de contrôle** : 0x8c79

Données Hexadécimales : d9 68 00 35 00 21 1f 92

Interprétation de la Capture

- **Découverte de Services :**
 - Les messages M-SEARCH montrent que l'ordinateur 192 . 168 . 128 . 1 recherche des services disponibles sur le réseau local.
 - Le paquet est envoyé en broadcast, donc il est reçu par tous les appareils sur le réseau local.
- **Adresses et Ports :**
 - **Adresses IP** : 192 . 168 . 128 . 1 (source) et 192 . 168 . 128 . 255 (destination, broadcast).
 - **Ports** : 53888 (source, choisi dynamiquement) et 1900 (destination, port standard pour SSDP).
- **Fonctionnalité du Protocole :**
 - SSDP utilise UDP pour sa rapidité et son efficacité. Les messages M-SEARCH permettent de découvrir rapidement les services disponibles sans établir une connexion préalable.

La capture Wireshark montre des paquets UDP utilisés par le protocole SSDP pour la découverte de services sur le réseau local. En utilisant Wireshark, nous pouvons examiner les détails des paquets UDP et comprendre comment les services sont découverts et communiquent sur le réseau.

3. Le protocole TCP

TCP (Transmission Control Protocol) est un protocole de la couche Transport (couche 4) du modèle OSI. Contrairement à UDP, TCP est un protocole orienté connexion, fiable et basé sur le flux. Il est conçu pour fournir un transfert de données fiable et ordonné entre les applications sur des réseaux IP.

Fiabilité :

- **Accusé de Réception (ACK) :** Chaque segment de données envoyé doit être confirmé par le récepteur, assurant ainsi la réception correcte des données.
- **Réémission :** Les segments non confirmés sont réémis pour garantir leur livraison.
- **Contrôle de Flux :**
 - **Fenêtre Glissante :** TCP utilise une fenêtre glissante pour contrôler le nombre de segments qui peuvent être envoyés sans accusé de réception. Cela permet d'ajuster dynamiquement le flux de données entre l'émetteur et le récepteur.
- **Contrôle de Congestion :**
 - **Algorithmes de Congestion :** TCP implémente plusieurs algorithmes de contrôle de congestion (comme le slow start, congestion avoidance, fast retransmit et fast recovery) pour éviter la surcharge du réseau.
- **Gestion de la Connexion :**
 - **Établissement de Connexion (Three-Way Handshake) :** TCP utilise un processus en trois étapes pour établir une connexion fiable entre l'émetteur et le récepteur.
 - **Fermeture de Connexion :** La connexion est fermée proprement en utilisant un processus en quatre étapes.

Capture en cours de ens33 (tcp)

Fichier Éditer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7	0.220871718	192.168.128.130	35.190.72.216	TCP	54	35676 → 443
8	0.224992839	192.168.128.130	35.190.72.216	TLSv1.3	60	Change Cipher
9	0.225468763	35.190.72.216	192.168.128.130	TCP	60	443 → 35676
10	0.237327238	192.168.128.130	35.190.72.216	TCP	54	35676 → 443
11	0.237836906	35.190.72.216	192.168.128.130	TCP	60	443 → 35676
12	0.259740929	35.190.72.216	192.168.128.130	TCP	60	443 → 35676
13	0.259841692	192.168.128.130	35.190.72.216	TCP	54	35676 → 443

Frame 10: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface ens33

Ethernet II, Src: VMware_42:eb:8c (00:0c:29:42:eb:8c), Dst: 35.190.72.216 (08:00:27:2d:7a:00)

Internet Protocol Version 4, Src: 192.168.128.130, Dst: 35.190.72.216

Transmission Control Protocol, Src Port: 35676, Dst Port: 443

Source Port: 35676

Destination Port: 443

[Stream index: 0]

[Conversation completeness: Complete, WITHIN]

[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 3515281893

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 0

Acknowledgment number (raw): 0

1010 ... = Header Length: 40 bytes (10)

To direct input to this VM, move the mouse pointer inside or press

Le protocole TCP dans cette capture :

Établissement de Connexion (Three-Way Handshake) :

- **SYN** : Le client (192 . 168 . 128 . 130) envoie un segment SYN pour initier une connexion avec le serveur (35 . 190 . 72 . 216) sur le port 443.
- **SYN-ACK** : Le serveur répond avec un segment SYN-ACK, confirmant la réception de la demande de connexion et proposant ses propres paramètres de connexion.
- **ACK** : Le client envoie un segment ACK, confirmant la réception de la réponse du serveur. La connexion est maintenant établie.
- **Transfert de Données** :
 - Une fois la connexion établie, les données peuvent être envoyées de manière fiable entre le client et le serveur.
 - Chaque segment de données envoyé est confirmé par l'autre partie à l'aide des accusés de réception (ACK).
- **Fermeture de Connexion (Four-Way Handshake)** :
 - Lorsque la transmission de données est terminée, la connexion est fermée proprement à l'aide de segments FIN et ACK.

Correspondant hexadécimal

Ethernet II :

- **Source MAC** : VMware_42:eb:8c (00:0c:29:42:eb:8c)
- **MAC de destination** : VMware_e6:3d

(00:50:56:e6:3d)

IPv4 :

- **IP source** : 192.168.128.130
- **Adresse IP de destination** : 35.190.72.216

TCP :

- **Port source** : 35676
- **Port de destination** : 443 (HTTPS)
- **Numéro de séquence** : 3515281893
- **Numéro d'accusé de réception** : 0
- **Décalage des données** : 8
- **Drapeaux** : SYN
- **Fenêtre** : 65535
- **Somme de contrôle** : 0xfa6c
- **Pointeur urgent** : 0

Données Hexadécimales :

d9 68 00 50 35 12 34 56 00 00 00 00 50 02 20 00 91 7c 00 00

Conclusion

Cette capture montre l'initialisation d'une connexion TCP entre un client et un serveur HTTPS. Le client utilise un port source dynamique (35676) et cible le port HTTPS standard (443) du serveur. Les segments SYN et SYN-ACK indiquent le début du processus de handshake pour établir une connexion fiable. Cette méthode assure une communication ordonnée et fiable entre les deux

parties, essentielle pour les applications nécessitant une intégrité des données, comme les transactions web sécurisées via HTTPS.

Mécanisme de Connexion TCP (Schéma)

Poignée de main à trois :

1. **SYN** : Client → Serveur
2. **SYN-ACK** : Serveur → Client
3. **ACK** : Client → Serveur

Fermeture de Connexion :

1. **FIN** : Client → Serveur
2. **ACK** : Serveur → Client
3. **FIN** : Serveur → Client
4. **ACK** : Client → Serveur

Installation d'un serveur DHCP et client :

```
GNU nano 7.2 /etc/dhcp/dhcpd.conf

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
subnet 192.168.128.0 netmask 255.255.255.0 {
  range 192.168.128.10 192.168.128.20;
  option routers 192.168.128.1;
  option broadcast-address 192.168.128.255;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  option domain-name "dhcp.local";
}
```

Configuration du fichier network :

```
GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dhcp

auto ens32
iface ens32 inet dhcp

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^V Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Configuration du fichier DHCP :

```
GNU nano 7.2 /etc/dhcp/dhclient.conf *
#supersede domain-name "fugue.com home.vix.com";
prepend domain-name-servers 127.0.0.1;
require subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, domain-search, host-name
        netbios-name-servers, netbios-scope, interface-mtu,
        rfc3442-classless-static-routes, ntp-servers;

#timeout 60;
#retry 60;
#reboot 10;
#select-timeout 5;
#initial-interval 2;
#script "/sbin/dhclient-script";
#media "-link0 -link1 -link2", "link0 link1";
#reject 192.33.137.209;

#alias {
# interface "eth0";
# fixed-address 192.5.5.213;

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^V Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

```

Lecture des informations d'état... Fait
isc-dhcp-client est déjà la version la plus récente (4.4.3-P1-2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:/home/safia# find / -name dhclient
/usr/sbin/dhclient
/usr/share/bash-completion/completions/dhclient
find: '/proc/2111': Aucun fichier ou dossier de ce type
find: '/run/user/1000/doc': Permission non accordée
find: '/run/user/1000/gvfs': Permission non accordée
root@debian:/home/safia# /usr/sbin/dhclient -v ens32
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens32/00:0c:29:98:a3:c4
Sending on   LPF/ens32/00:0c:29:98:a3:c4
Sending on   Socket/fallback
DHCPDISCOVER on ens32 to 255.255.255.255 port 67 interval 7
DHCPOFFER of 192.168.128.11 from 192.168.128.130
DHCPREQUEST for 192.168.128.11 on ens32 to 255.255.255.255 port 67
DHCPACK of 192.168.128.11 from 192.168.128.130
bound to 192.168.128.11 -- renewal in 257 seconds.
root@debian:/home/safia#

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```

safia@debian: ~
Sending on   LPF/ens32/00:0c:29:98:a3:c4
Sending on   Socket/fallback
DHCPDISCOVER on ens32 to 255.255.255.255 port 67 interval 7
DHCPOFFER of 192.168.128.11 from 192.168.128.130
DHCPREQUEST for 192.168.128.11 on ens32 to 255.255.255.255 port 67
DHCPACK of 192.168.128.11 from 192.168.128.130
bound to 192.168.128.11 -- renewal in 257 seconds.
root@debian:/home/safia# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:98:a3:c4 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.128.11/24 brd 192.168.128.255 scope global dynamic ens32
        valid_lft 375sec preferred_lft 375sec
    inet6 fe80::20c:29ff:fe98:a3c4/64 scope link
        valid_lft forever preferred_lft forever
root@debian:/home/safia#

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Paramètres Serveur DHCP

```
GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.2
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Parmètres DHCP client

```
GNU nano 7.2 /etc/network/interfaces *

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#auto ens33
#iface ens33 inet dhcp

auto ens32
iface ens32 inet dhcp

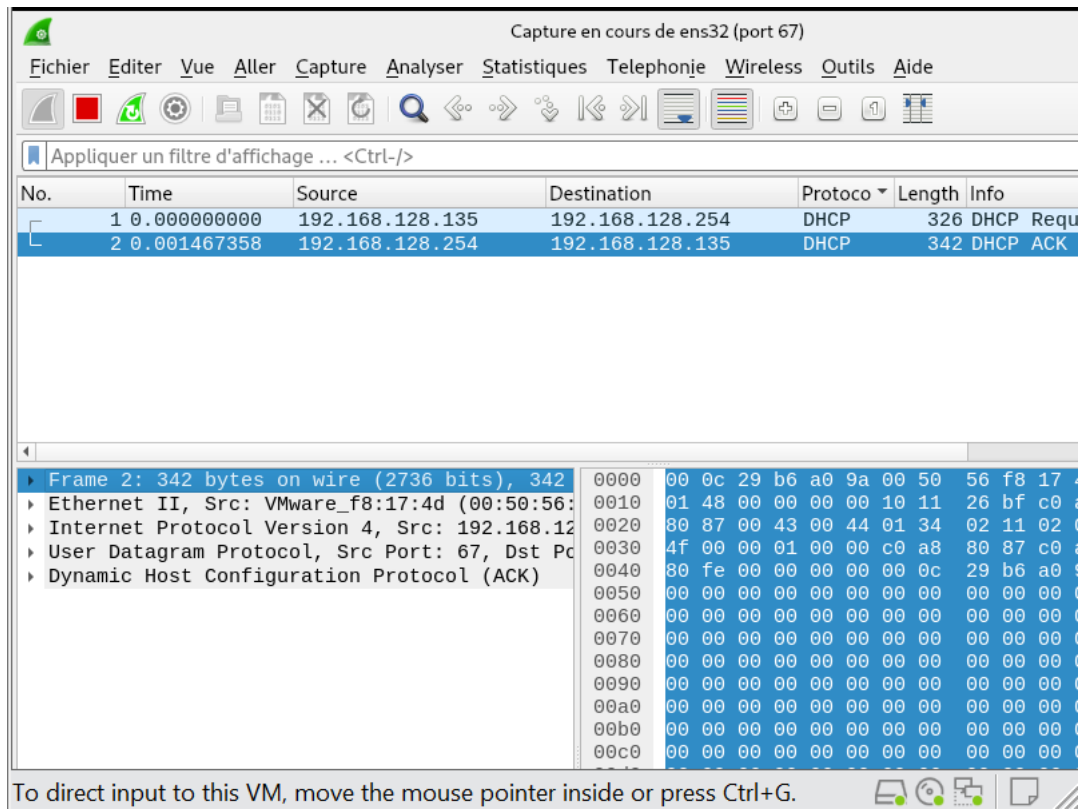
auto eth0

iface eth0 inet static
    address 192.168.1.3
    netmask 255.255.255.0
    gateway 192.168.1.1

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^V Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Echange DHCP entre 2 adresses IP Le serveur et le client.



Capture en cours de ens32 (port 67)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.128.135	192.168.128.254	DHCP	326	DHCP Request
2	0.001467358	192.168.128.254	192.168.128.135	DHCP	342	DHCP ACK

Frame 2: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface ens32

Ethernet II, Src: VMware_f8:17:4d (00:50:56:00:00:00), Dst: 02:00:00:00:00:00

Internet Protocol Version 4, Src: 192.168.128.254, Dst: 192.168.128.135

User Datagram Protocol, Src Port: 67, Dst Port: 68

Dynamic Host Configuration Protocol (ACK)

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Ces deux paquets montrent une transaction DHCP standard :

1. Le client DHCP envoie une requête DHCP (DHCP Request) pour obtenir une adresse IP.
2. Le serveur DHCP répond avec un message d'acquittement (DHCP ACK) confirmant l'attribution de l'adresse IP.

Cela semble être un échange DHCP normal sans erreurs apparentes. Les adresses MAC et les adresses IP impliquées correspondent à une configuration réseau utilisant des machines virtuelles (comme indiqué par les adresses MAC VMware).

Capture en cours de ens32 (port 443)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
614	1.741229400	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
615	1.769577337	151.101.130.49	192.168.128.135	TCP	2966	443 → 34592
616	1.769577636	151.101.130.49	192.168.128.135	TCP	1510	443 → 34592
617	1.769642343	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
618	1.772296844	151.101.130.49	192.168.128.135	TCP	7334	443 → 34592
619	1.772297141	151.101.130.49	192.168.128.135	TLSv1.2	1510	Applicati
620	1.772490497	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
621	1.773364806	151.101.130.49	192.168.128.135	TLSv1.2	4254	Applicati
622	1.773401115	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
623	1.778007241	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
624	1.778786664	151.101.130.49	192.168.128.135	TCP	60	443 → 34592
625	1.790949888	151.101.130.49	192.168.128.135	TLSv1.2	77	Encrypted
626	1.790979097	192.168.128.135	151.101.130.49	TCP	54	34592 → 443

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface
 Ethernet II, Src: VMware_b6:a0:9a (00:0c:29:b6:a0:9a), Dst: 08:00:27:00:00:00
 Internet Protocol Version 4, Src: 192.168.128.135, Destination: 151.101.130.49
 Transmission Control Protocol, Src Port: 50982, Dst Port: 443
 [Community ID: 1:IOhlw+0xzllhP7cfP/1QkBqWges=]
 TRANSUM RTE Data

Analyse de la Capture

1. Protocole :

- La capture montre principalement des paquets utilisant le protocole TCP et TLSv1.2. Les ports concernés sont le port source 50982 et le port destination 443, ce qui est typique pour le trafic HTTPS (SSL/TLS).

2. TLS/SSL Paquets :

- Les paquets 620, 621, 624, et 625 montrent des transmissions utilisant TLSv1.2, ce qui signifie que vous avez capturé des paquets SSL/TLS comme désiré.
- Le paquet 625 mentionne "Encrypted Application Data," ce qui indique que la session TLS est active et que les données d'application sont chiffrées.

Les adresses IP source et destination indiquent que les communications sont entre 192.168.128.135 et 151.101.130.49.

- Les différents paquets montrent un mélange de négociation de connexion et de transmission de données chiffrées.

*ens32 (port 443)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
614	1.741229400	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
615	1.769577337	151.101.130.49	192.168.128.135	TCP	2966	443 → 34592
616	1.769577636	151.101.130.49	192.168.128.135	TCP	1510	443 → 34592
617	1.769642343	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
618	1.772296844	151.101.130.49	192.168.128.135	TCP	7334	443 → 34592
619	1.772297141	151.101.130.49	192.168.128.135	TLSv1.2	1510	Application Data
620	1.772490497	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
621	1.773364806	151.101.130.49	192.168.128.135	TLSv1.2	4254	Application Data
622	1.773401115	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
623	1.778007241	192.168.128.135	151.101.130.49	TCP	54	34592 → 443
624	1.778786664	151.101.130.49	192.168.128.135	TCP	60	443 → 34592
625	1.790949888	151.101.130.49	192.168.128.135	TLSv1.2	77	Encrypted Data
626	1.790979097	192.168.128.135	151.101.130.49	TCP	54	34592 → 443

Frame 619: 1510 bytes on wire (12080 bits), 1510 bytes captured (12080 bits) on interface eth0

Ethernet II, Src: VMware_e6:3d:ac (00:50:56:e6:3d:ac), Dst: 192.168.128.135

Internet Protocol Version 4, Src: 151.101.130.49, Dst: 192.168.128.135

Transmission Control Protocol, Src Port: 443, Dst Port: 34592

[5 Reassembled TCP Segments (16405 bytes): #613(3960 bytes), #614(54 bytes), #615(2966 bytes), #616(1510 bytes), #617(54 bytes)]

Transport Layer Security

Transport Layer Security

TLSv1.2 Record Layer: Application Data Protocol

[Community ID: 1:2KdNe47KlLWJYhKMdZtEqEblbjM=]

Capture DNS :

Capture en cours de ens32 (port 53)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

port 53

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.128.134	192.168.128.2	DNS	75	Standard query query
2	0.005599697	192.168.128.2	192.168.128.134	DNS	91	Standard query response
3	0.006369022	192.168.128.134	192.168.128.2	DNS	75	Standard query query
4	0.011209888	192.168.128.2	192.168.128.134	DNS	103	Standard query response

Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface eth0

Ethernet II, Src: VMware_e9:e9:42 (00:0c:29:e9:e9:42), Dst: 192.168.128.134

Internet Protocol Version 4, Src: 192.168.128.134, Dst: 192.168.128.2

User Datagram Protocol, Src Port: 54196, Dst Port: 53

Domain Name System (query)

Transaction ID: 0x9341

Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

Queries

[Response To: 1]

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

La première requête est une demande de résolution DNS pour **www.google.com** de la part de 192.168.128.134 à 192.168.128.2.

La réponse indique que l'adresse IP pour **www.google.com** est 172.217.20.206.

La deuxième requête est une demande de résolution DNS pour **www.example.com**.

La réponse indique que l'adresse IP pour **www.example.com** est 93.184.216.34.

Observation des Échanges FTP sans TLS

Lors de la capture des échanges FTP sans TLS, voici ce que l'on peut observer :

. Données en Clair :

- **Nom d'utilisateur et mot de passe** : Les informations de connexion sont envoyées en clair. On peut voir les commandes **user** et **pass** avec les vrais nom d'utilisateur et du mot de passe dans les paquets.
- **Transfert de fichiers** : Les fichiers transférés entre le client et le serveur sont également visibles en clair

. Analyse des Paquets :

- **Commandes FTP** : On peut voir des commandes FTP comme **list**, **retr**, **stor**, en clair dans les paquets capturés.
- **Réponses du Serveur** : Les réponses du serveur, comme les codes de statut, sont également visibles en clair.

Capture en cours de ens32 (tcp port 443)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

port 443

No.	Time	Source	Destination	Protocol	Length	Info
21	0.686434948	192.168.128.134	93.184.215.14	TLSv1.3	85	Application Data
22	0.686435727	93.184.215.14	192.168.128.134	TCP	60	443 → 59672 [ACK] Seq=44
23	0.774119082	93.184.215.14	192.168.128.134	TLSv1.3	85	Application Data
24	0.779386639	93.184.215.14	192.168.128.134	TLSv1.3	1570	Application Data, Applic
25	0.780368285	192.168.128.134	93.184.215.14	TCP	60	59672 → 443 [ACK] Seq=12
26	0.787092651	192.168.128.134	93.184.215.14	TCP	60	59672 → 443 [FIN, ACK] S
27	0.787535230	93.184.215.14	192.168.128.134	TCP	60	443 → 59672 [ACK] Seq=59
28	0.879164287	93.184.215.14	192.168.128.134	TCP	60	443 → 59672 [FIN, PSH, A
29	0.880285379	192.168.128.134	93.184.215.14	TCP	60	59672 → 443 [ACK] Seq=12

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (584 bits) on interface ens32

Ethernet II, Src: VMware_e9:e9:42 (00:0c:29:e9:e9:42), Dst: 00:0c:29:00:00:00

Internet Protocol Version 4, Src: 192.168.128.134, Destination: 93.184.215.14

Transmission Control Protocol, Src Port: 59672, Dst Port: 443

Source Port: 59672
Destination Port: 443
[Stream index: 0]
[Conversation completeness: Incomplete, SYN]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 3923196612
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1010 = Header Length: 40 bytes (10)
Flags: 0x002 (SYN)
Window: 64240
[Calculated window size: 64240]
Checksum: 0xd11f [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (20 bytes), Maximum segment size, [Timestamps]

ens32: <live capture in progress> Paquets : 29 - Affichés : 29 (100.0%) Profil : Default

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Capture TCP :

La capture montre un segment TCP en cours sur le port 443, qui est généralement utilisé pour le protocole HTTPS.

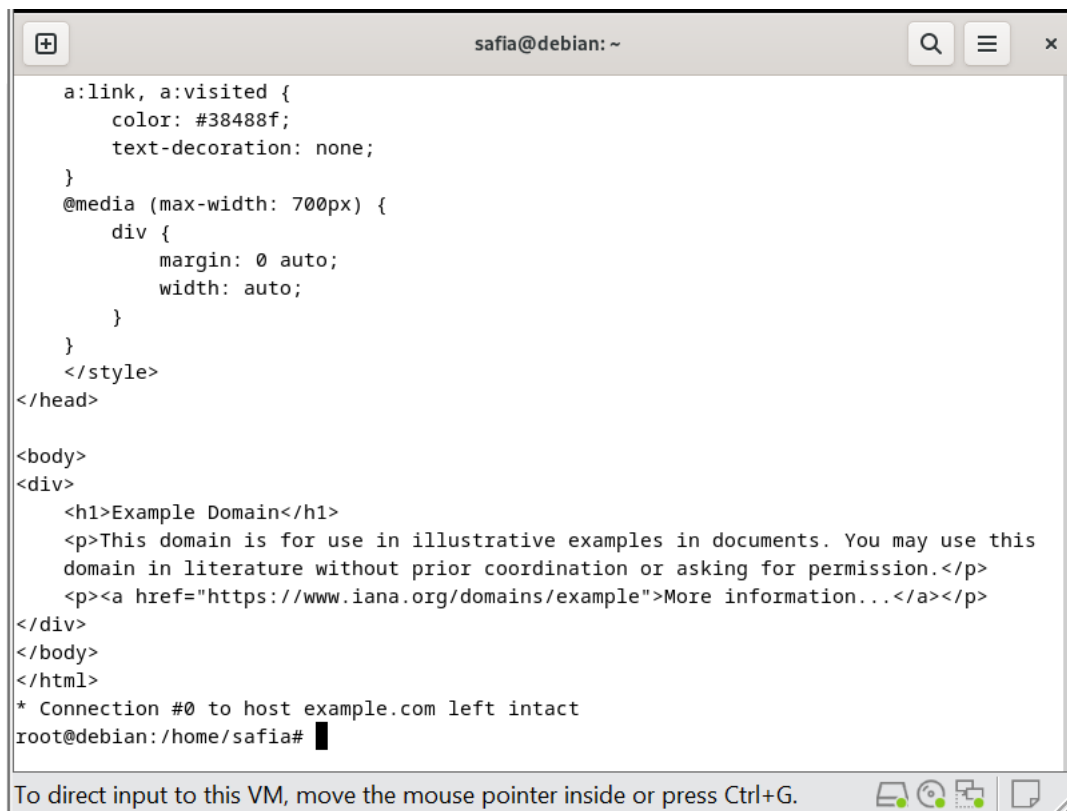
Ce paquet est un paquet SYN envoyé de 192.168.128.134 (port 59672) à 93.184.215.14 (port 443) pour établir une connexion TCP. Il fait partie du processus de handshake à trois voies de TCP, où le client envoie un paquet SYN pour initier la connexion, le serveur répond avec un paquet SYN-ACK, et le client termine le handshake avec un paquet ACK.

Capture SSL/TLS

SSL et TLS sont des protocoles utilisés pour sécuriser les communications sur un réseau informatique, et ils sont souvent utilisés dans des applications comme HTTPS, FTPS, et d'autres. Le port 443 est couramment utilisé pour HTTPS, qui utilise SSL/TLS.

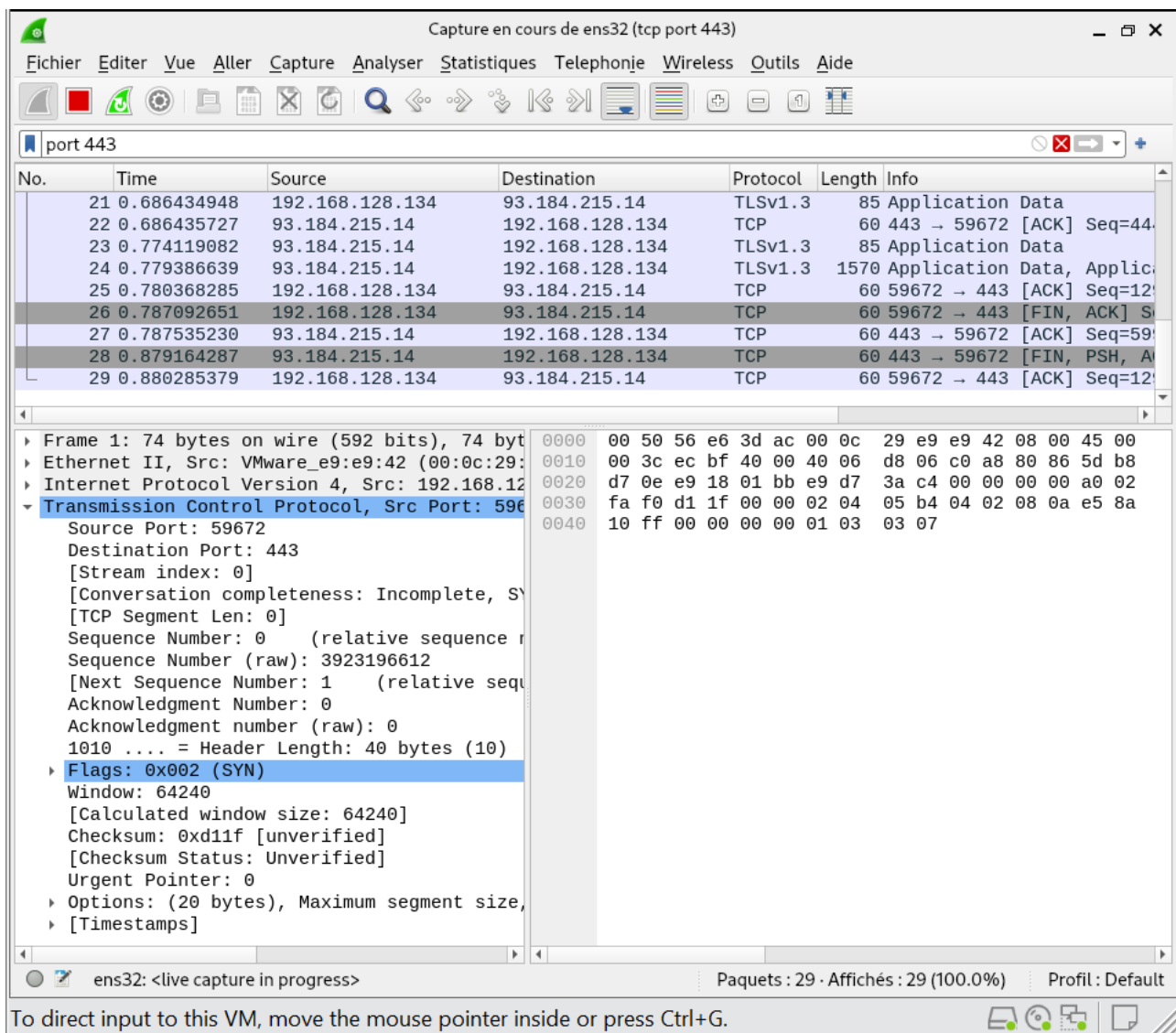
Pour générer du trafic SSL/TLS, vous pouvez simplement visiter un site web HTTPS dans un navigateur ou utiliser curl en ligne de commande :

```
curl https://www.example.com
```



```
safia@debian: ~  
  
a:link, a:visited {  
    color: #38488f;  
    text-decoration: none;  
}  
@media (max-width: 700px) {  
    div {  
        margin: 0 auto;  
        width: auto;  
    }  
}  
}</style>  
</head>  
  
<body>  
<div>  
    <h1>Example Domain</h1>  
    <p>This domain is for use in illustrative examples in documents. You may use this  
    domain in literature without prior coordination or asking for permission.</p>  
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>  
</div>  
</body>  
</html>  
  
* Connection #0 to host example.com left intact  
root@debian:/home/safia#
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



3ème Partie

Installation de Tshark :

- apt-get update
- apt-get install tshark

Pour capturer tous les paquets

- `tshark -i ens32 -w capture_all.pcap`
Pour rendre le script exécutable
- `chmod +x capture_scripts.sh`
Pour l'exécuter
- `./capture_scripts.sh`

```
Running as user "root" and group "root". This could be dangerous.
Capturing on 'ens32'
** (tshark:4454) 22:39:08.048296 [Main MESSAGE] -- Capture started.
** (tshark:4454) 22:39:08.048501 [Main MESSAGE] -- File: "/root/captures/capture_all.p
cap"
68 ^C
tshark:
root@mail:/home/safia# tshark -r ~/captures/capture_all.pcap
Running as user "root" and group "root". This could be dangerous.
  1 0.000000000 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  2 1.023976702 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  3 2.051962581 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  4 3.074136147 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  5 4.096983795 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  6 5.121337260 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  7 6.145360121 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
  8 7.168638870 VMware_e9:e9:42 → Broadcast    ARP 60 Who has 192.168.1.1? Tell 192.1
68.1.100
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Script pour l'automatisation des captures Tshark :



```
GNU nano 7.2 capture_tshark.sh *
INTERFACE="ens32"

CAPTURE_DIR="/home/safia/captures"
mkdir -p $CAPTURE_DIR

sudo tshark -i $INTERFACE -f "tcp port 80" -w $CAPTURE_DIR/capture_http.pcap
sudo tshark -i $INTERFACE -f "tcp port 443" -w $CAPTURE_DIR/capture_https.pcap
sudo tshark -i $INTERFACE -f "udp port 53" -w $CAPTURE_DIR/capture_dns.pcap
sudo tshark -i $INTERFACE -f "tcp port 21" -w $CAPTURE_DIR/capture_ftp.pcap
sudo tshark -i $INTERFACE -f "tcp port 25" -w $CAPTURE_DIR/capture_smtp.pcap

echo "Captures terminées. Les fichiers sont enregistrés dans $CAPTURE_DIR"
```

^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^_ Remplacer ^U Coller ^J Justifier ^/ Aller ligne

Pour

rendre le script exécutable

- `chmod +x /home/safia/capture_tshark.sh`
- `sudo /home/safia/capture_tshark.sh`