Ecole Supérieure
d'Informatique et du Numérique
COLLEGE OF ENGINEERING & ARCHITECTURE

COMPTE RENDU

Cyber Sécurité - TP2 - Attaques Réseau et Sécurité des Systèmes
3e année Cybersécurité - École Supérieure d'Informatique et du Numérique (ESIN)
Collège d'Ingénierie & d'Architecture (CIA)

Étudiants : HATHOUTI Mohammed Taha
JIDAL Ilyas

Filière : Cybersecurité

Année : 2025/2026

Enseignant : M.SEBBAR & Mme.GADI

Date : 8 février 2026

Table des matières

Objectifs

L'objectif de ce TP est d'illustrer les techniques d'attaque réseau telles que l'interception et la manipulation de trafic, le scan de ports, ainsi que les attaques par déni de service (DoS) et déni de service distribué (DDoS).

Dans ce TP, nous nous concentrerons sur :

— La capture et l'analyse de paquets réseau avec tcpdump et Wireshark ;

— L'interception de trafic avec Ettercap ;

— Les attaques DoS et DDoS avec Hping3 ;

Instructions

— Le rapport de TP doit être soumis une semaine après la séance de TP sur la plateforme Moodle, en respectant la date limite mentionnée ;

— Le TP doit être réalisé individuellement en classe, mais le rapport doit être soumis en groupes de 2 étudiants maximum ;

— Les groupes de TP doivent rester les mêmes pour tous les rapports tout au long du semestre ;

Précautions Légales et Éthiques

IMPORTANT : Les attaques DoS et DDoS, ainsi que d'autres tests de pénétration, sont **illégales** si elles sont menées sans autorisation explicite. Ces tests doivent être effectués dans un environnement contrôlé et éthique (par exemple, sur des systèmes pour lesquels vous avez l'autorisation de tester la sécurité). Ce TP a été réalisé dans un environnement de laboratoire avec des machines de test dédiées.

Configuration du Réseau

Architecture du Réseau de Test

Pour ce TP, nous avons connecté deux machines physiques via un câble Ethernet :

— **Machine Ubuntu** (Attaquant) : IP = **192.168.1.20/24**

— **Machine Windows 7** (Cible) : IP = **192.168.1.10/24**

Configuration de la Machine Ubuntu

Sur la machine Ubuntu, nous avons configuré l'interface Ethernet avec l'adresse IP 192.168.1.20 :

```
sudo ip addr add 192.168.1.20/24 dev enx0c3796401539
```

```
enx0c3796401539: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.20  netmask 255.255.255.0  broadcast 0.0.0.0
        ether 0c:37:96:40:15:39  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 1 – Vérification de la configuration de l'interface réseau Ubuntu

Configuration de la Machine Windows 7

Sur la machine Windows 7, nous avons configuré l'adresse IP statique via la commande :

```
netsh interface ip set address name="Ethernet" source=static addr
   =192.168.1.10 mask=255.255.255.0
```

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2d47:225a:25a8:aa93%6
   IPv4 Address. . . . . . . . . . . : 192.168.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

FIGURE 2 – Vérification de la configuration de l'interface réseau Windows avec ipconfig

Test de Connectivité

Nous avons vérifié la connectivité entre les deux machines avec la commande ping :

Ping depuis Ubuntu vers Windows

```
ping -c 4 192.168.1.10
```

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP2$ ping -c 4 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=128 time=0.944 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=128 time=0.890 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=128 time=1.31 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=128 time=1.14 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.890/1.071/1.308/0.166 ms
```

FIGURE 3 – Test de connectivité avec ping depuis Ubuntu vers Windows (192.168.1.10)

Ping depuis Windows vers Ubuntu

```
ping 192.168.1.20
```

```
C:\Windows\System32>ping 192.168.1.20

Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time=1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\System32>
```

FIGURE 4 – Test de connectivité avec ping depuis Windows vers Ubuntu (192.168.1.20)

La connexion est bien établie avec un taux de perte de 0%, confirmant que le réseau local fonctionne correctement.

1 Partie 1 : Capture et Analyse de Paquets

1.1 Capture de Paquets avec tcpdump

1.1.1 Capture Initiale de 500 Paquets

En tant qu'administrateur, nous avons effectué une capture de 500 paquets sur l'interface réseau enx0c3796401539 :

```
sudo tcpdump -i enx0c3796401539 -c 500 -w ~/Bureau/UIR/3A/S6/
   Cybersecurite/TP2/macapture.pcap
```

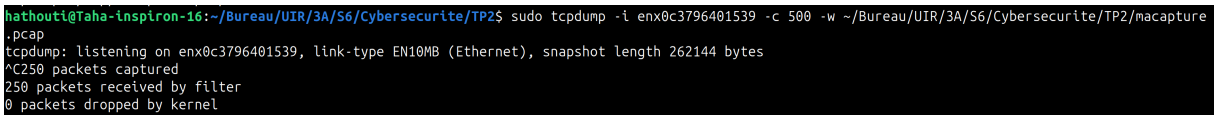

FIGURE 5 – Capture de 500 paquets avec tcpdump

Résultat : 250 paquets capturés, 250 paquets reçus par le filtre, 0 paquets perdus par le kernel.

1.1.2 Vérification du Fichier Capturé

Nous avons vérifié la nature du fichier capturé avec la commande `file` :

```
file macapture.pcap
```

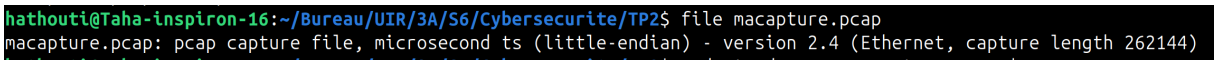

FIGURE 6 – Lecture du fichier macapture.pcap avec tcpdump

Résultat : Le fichier est identifié comme "pcap capture file, microsecond ts (little-endian) - version 2.4 (Ethernet, capture length 262144)".

1.1.3 Lecture des Paquets Capturés

Nous avons lu le contenu du fichier pcap avec tcpdump :

```
sudo tcpdump -r macapture.pcap | more
```

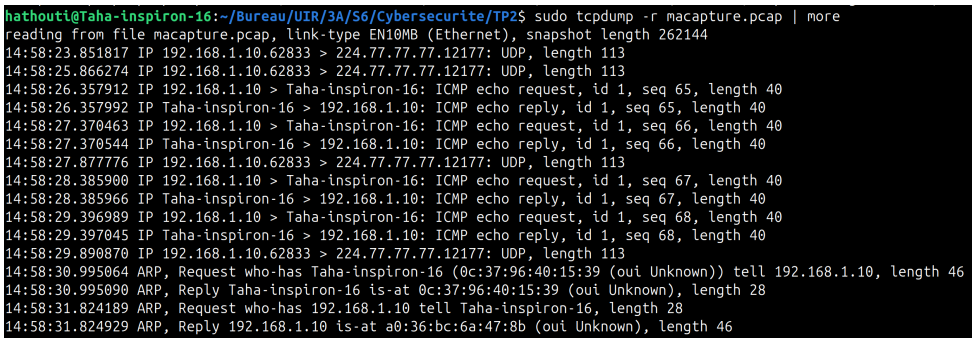

FIGURE 7 – Analyse détaillée des paquets

On observe différents types de trafic :

— **UDP** : Paquets vers 224.77.77.77 (multicast) ;

— **ICMP** : Requêtes et réponses echo (ping) ;

— **ARP** : Résolution d'adresses MAC ;

1.2 Capture Avancée

Nous avons capturé 20 paquets sans traduction d'adresse et visualisé les premiers 1500 octets de chaque paquet :

```
sudo tcpdump -i enx0c3796401539 -c 20 -n -s 1500 -w macapture2.
   pcap
sudo tcpdump -r macapture2.pcap -x
```

L'option `-n` désactive la résolution DNS, et `-x` affiche le contenu des paquets en hexadécimal.

1.3 Filtrage de Paquets

1.3.1 Capture du Trafic ARP

Nous avons appliqué un filtre pour capturer uniquement les paquets ARP (Address Resolution Protocol) :

```
sudo tcpdump ether proto 0x806
```

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP2$ sudo tcpdump ether proto 0x806
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on wlp0s20f3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:02:58.535579 ARP, Request who-has Taha-inspiron-16 tell _gateway, length 42
12:02:58.535600 ARP, Reply Taha-inspiron-16 is-at 60:a5:e2:64:75:d0 (oui Unknown), length 28
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP2$
```

FIGURE 8 – Capture du trafic ARP avec tcpdump

Observation : On voit des requêtes ARP (Request who-has) et des réponses ARP (Reply is-at) permettant la résolution d'adresses IP en adresses MAC.

1.3.2 Capture du Trafic HTTP (Port 80)

Nous avons capturé le trafic HTTP en filtrant sur le port TCP 80 :

```
sudo tcpdump -i enx0c3796401539 'tcp port 80'
```

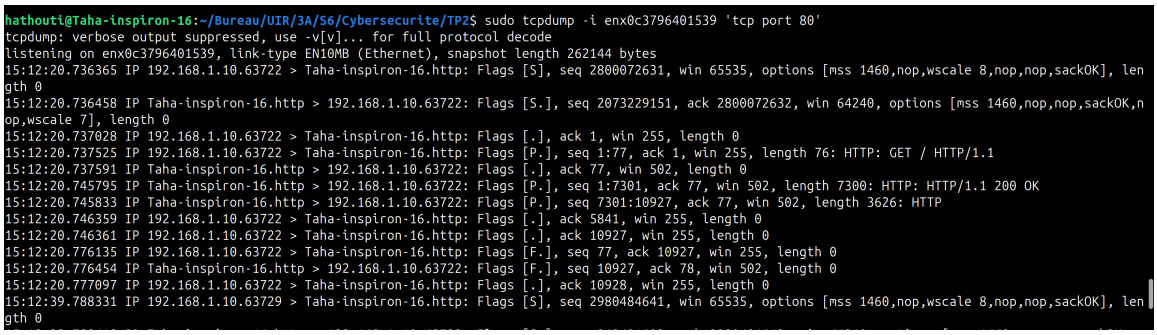

FIGURE 9 – Capture du trafic HTTP (port 80) avec tcpdump

Observation : On observe des requêtes HTTP (GET / HTTP/1.1) et des réponses (HTTP/1.1 200 OK) lors de la navigation web.

1.3.3 Capture du Trafic FTP et HTTP Combinés

Nous avons capturé simultanément le trafic FTP (port 21) et HTTP (port 80) :

```
sudo tcpdump -i enx0c3796401539 'tcp and (port 21 or 80)'
```

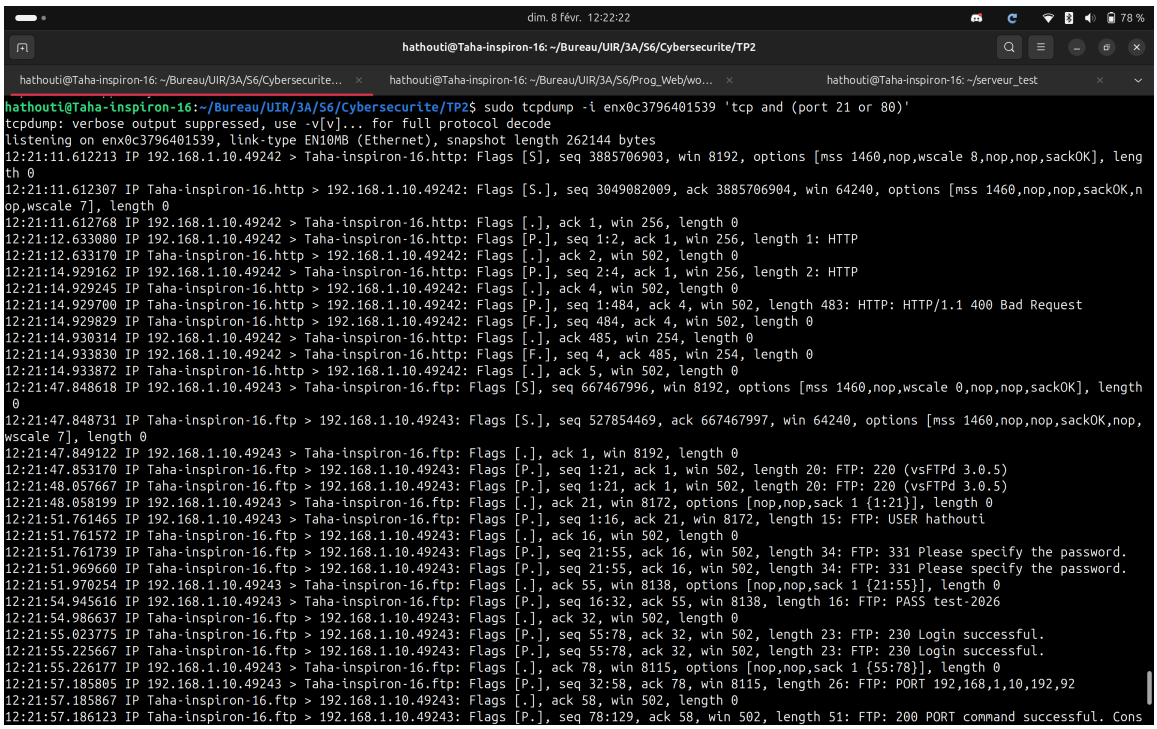

FIGURE 10 – Capture du trafic FTP et HTTP avec tcpdump

Observation : Cette capture montre à la fois :

— Des connexions FTP avec les commandes USER, PASS, et les réponses du serveur (220, 331, 230, 200) ;

— Des requêtes HTTP avec les three-way handshakes TCP ([SYN], [SYN, ACK], [ACK]) ;

1.4 Analyse avec Wireshark

Nous avons ouvert le fichier de capture avec Wireshark pour une analyse graphique plus détaillée :

```
wireshark macapture.pcap
```

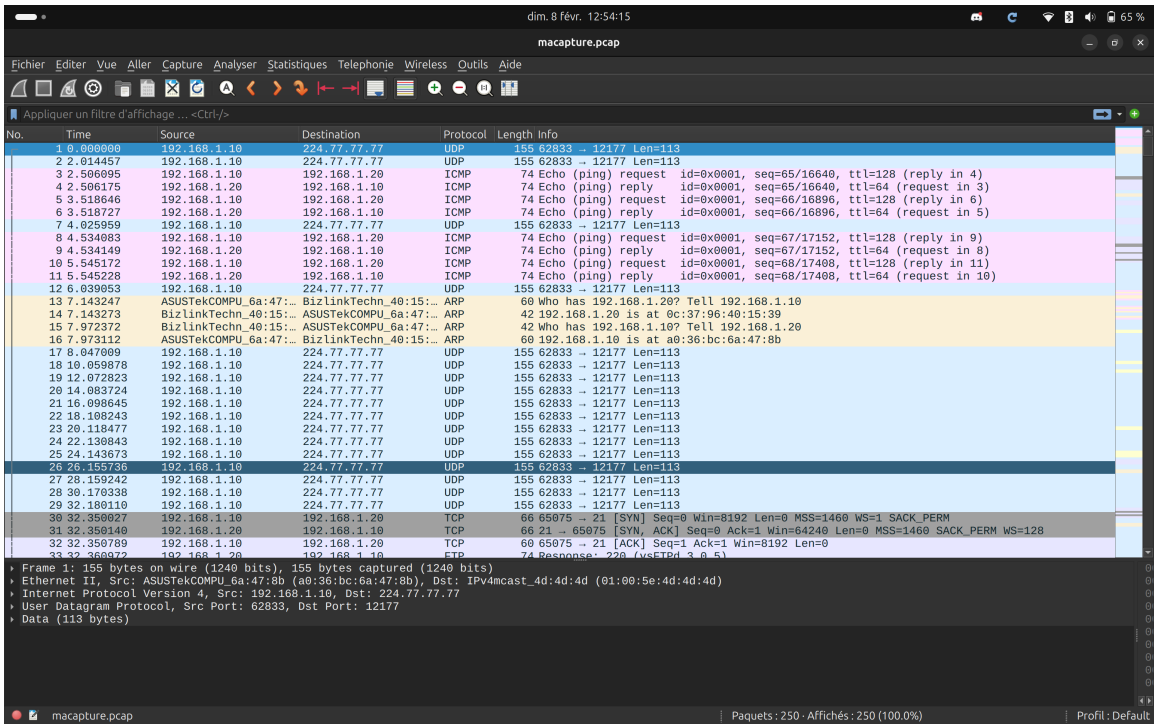

FIGURE 11 – Analyse du trafic avec Wireshark - Vue d'ensemble

Wireshark permet de :

- Visualiser les paquets de manière hiérarchique ;
- Appliquer des filtres avancés (par protocole, adresse IP, port) ;
- Analyser le contenu détaillé de chaque paquet ;
- Identifier les flux de communication ;

1.5 Interception avec Ettercap

Nous avons utilisé Ettercap pour rechercher du trafic TELNET dans notre capture :

```
ettercap -T -r macapture.pcap | grep TELNET
```

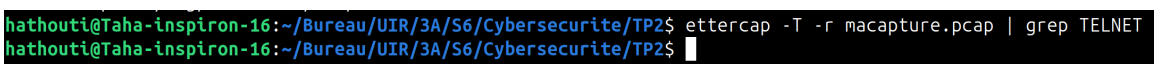

FIGURE 12 – Recherche de trafic TELNET avec Ettercap

Résultat : Aucun trafic TELNET n'a été détecté dans cette capture, ce qui est normal car nous n'avons pas établi de connexion TELNET durant cette session de capture.

2 Partie 2 : Attaques DoS et DDoS

2.1 Introduction aux Attaques par Déni de Service

Les attaques par **Déni de Service (DoS)** visent à rendre un service, un réseau ou une machine indisponible en épuisant ses ressources (CPU, mémoire, bande passante). Les attaques **DDoS (Distributed Denial of Service)** utilisent plusieurs machines pour amplifier l'impact.

Dans ce TP, nous utilisons **Hping3**, un outil puissant de génération de paquets réseau permettant de simuler différents types d'attaques DoS.

2.2 Installation de Hping3

Nous avons installé Hping3 sur la machine Ubuntu :

```
sudo apt-get update
sudo apt-get install hping3
```

2.3 Attaque SYN Flood (TCP)

2.3.1 Principe de l'Attaque SYN Flood

L'attaque **SYN Flood** exploite le mécanisme de *three-way handshake* TCP :

1. L'attaquant envoie massivement des paquets **SYN** vers la cible
2. La cible répond avec des paquets **SYN-ACK** et garde ces connexions en mémoire
3. L'attaquant ne répond jamais avec le **ACK** final
4. Les connexions semi-ouvertes s'accumulent et épuisent les ressources système

2.3.2 Lancement de l'Attaque SYN Flood

Nous avons lancé une attaque SYN flood vers le port 80 (HTTP) de la machine Windows :

```
sudo hping3 --flood --syn -p 80 192.168.1.10
```

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP2$ sudo hping3 --flood --syn -p 80 192.168.1.10
HPING 192.168.1.10 (enx0c3796401539 192.168.1.10): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.10 hping statistic ---
17324730 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

FIGURE 13 – Lancement de l'attaque SYN flood avec Hping3

Paramètres de la commande :

— `--flood` : Envoie les paquets le plus rapidement possible ;
— `--syn` : Envoie uniquement des paquets SYN (demande de connexion) ;
— `-p 80` : Cible le port 80 (HTTP) ;
— `192.168.1.10` : Adresse IP de la machine Windows 7 ;

Résultat : L'attaque a envoyé **17 324 730 paquets** avant d'être arrêtée, avec un taux de perte de 100% (aucune réponse reçue, ce qui est normal en mode flood).

2.3.3 Impact sur la Machine Windows 7

Onglet Performance - Utilisation du Processeur

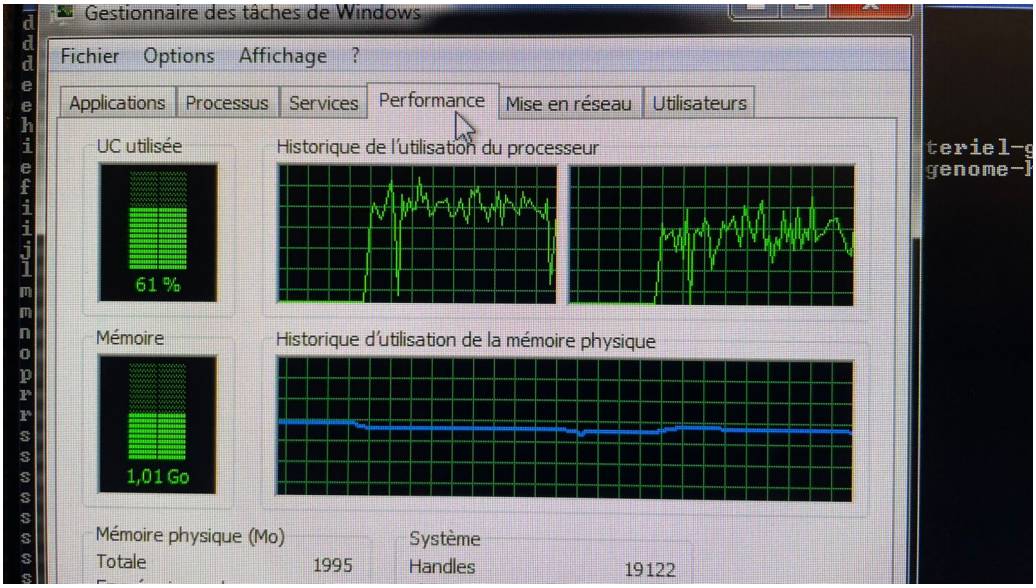

FIGURE 14 – Gestionnaire des tâches Windows - Impact sur le CPU lors de l'attaque SYN

Observations :

— **UC utilisée : 61%** - Utilisation anormalement élevée du processeur ;

— **Mémoire : 1,01 Go** - Consommation mémoire importante ;

— **Historique du processeur :** Pics d'activité visibles correspondant aux vagues de paquets SYN ;

Onglet Mise en Réseau - Utilisation du Réseau

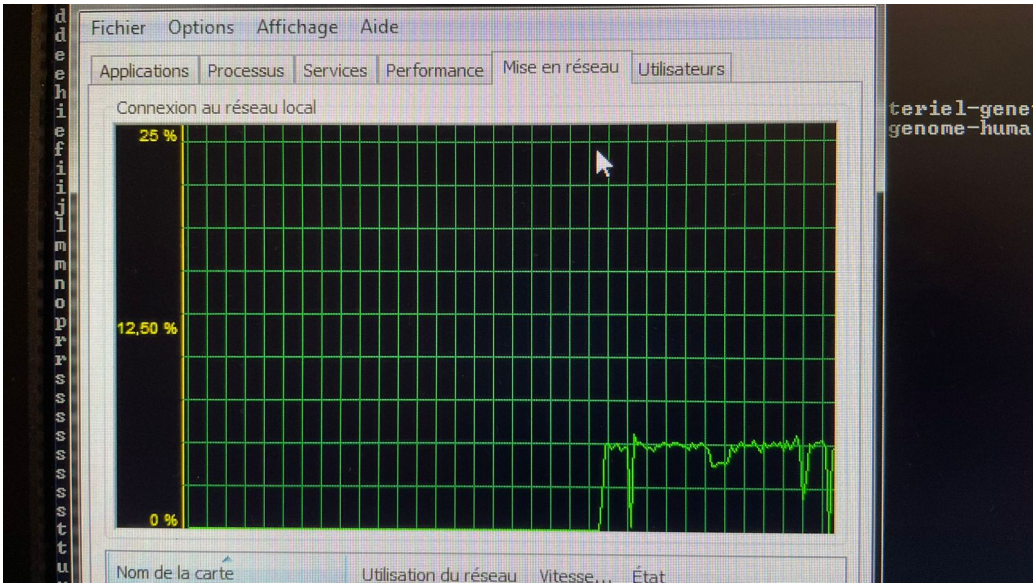

FIGURE 15 – Gestionnaire des tâches Windows - Impact réseau lors de l'attaque SYN

Observations :

— **Pic initial : 25%** d'utilisation réseau (barre jaune) ;

— **Stabilisation : 12,50%** après le pic initial ;

— **Activité anormale :** La machine n'est pas connectée à Internet, donc toute cette activité provient uniquement de l'attaque ;

2.3.4 Interprétation des Résultats - Attaque SYN

Impact constaté :

— **Surcharge CPU :** 61% d'utilisation pour traiter les faux paquets SYN ;

— **Saturation réseau :** 25% d'utilisation sans connexion Internet légitime ;

— **Épuisement des ressources :** Accumulation de connexions semi-ouvertes ;

— **Déni de service partiel :** Les ressources système sont monopolisées par l'attaque ;

Mécanisme de l'attaque :

1. Hping3 envoie des milliers de paquets SYN/seconde ;
2. Windows 7 essaie de répondre avec SYN-ACK pour chaque paquet ;
3. Le système garde en mémoire toutes ces connexions "en attente" ;
4. Les ressources (CPU, mémoire, table de connexions) s'épuisent progressivement ;

2.4 Attaque UDP Flood

2.4.1 Principe de l'Attaque UDP Flood

Contrairement au TCP, **UDP** est un protocole sans connexion (connectionless). L'attaque UDP flood consiste à :

— Envoyer massivement des paquets UDP vers un port cible ;

— Saturer la bande passante réseau ;

— Forcer le système à traiter et répondre (si le port est ouvert) ou rejeter (si fermé) ces paquets ;

2.4.2 Lancement de l'Attaque UDP Flood

Nous avons lancé une attaque UDP flood vers le port 53 (DNS) de la machine Windows :

```
sudo hping3 --flood --udp -p 53 192.168.1.10
```

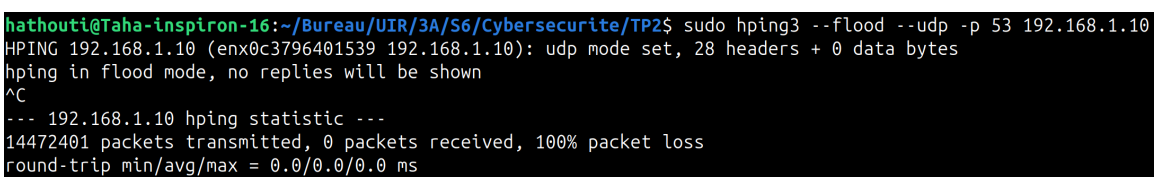

FIGURE 16 – Lancement de l'attaque UDP flood avec Hping3

Paramètres de la commande :

— `--flood` : Envoie les paquets le plus rapidement possible ;

— `--udp` : Envoie des paquets UDP (sans connexion) ;

— `-p 53` : Cible le port 53 (DNS) ;

— `192.168.1.10` : Adresse IP de la machine Windows 7 ;

Résultat : L'attaque a envoyé **14 472 401 paquets** avec 100% de perte (normal en mode flood UDP).

2.4.3 Impact sur la Machine Windows 7

Onglet Performance - Utilisation du Processeur

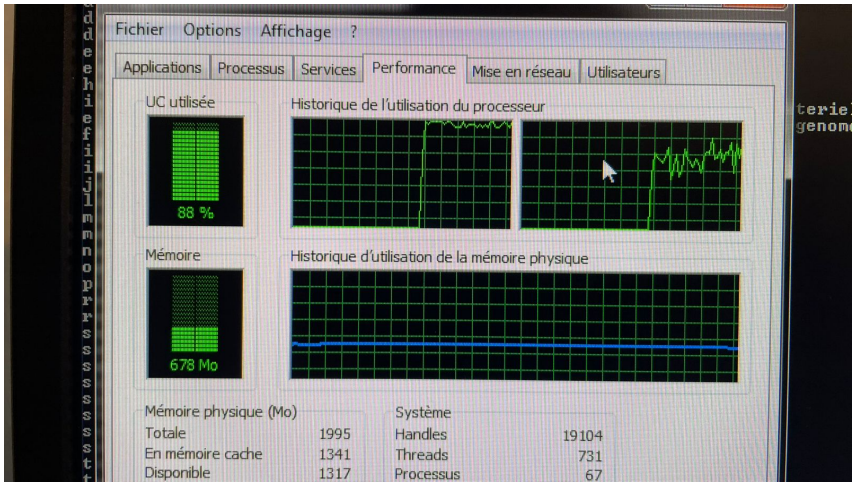

FIGURE 17 – Gestionnaire des tâches Windows - Impact CPU lors de l'attaque UDP

Observations :

— **UC utilisée : 89%** - Utilisation CPU encore plus élevée que l'attaque SYN !

— **Mémoire : 678 Mo** - Consommation mémoire réduite par rapport au SYN ;

Onglet Mise en Réseau - Utilisation du Réseau

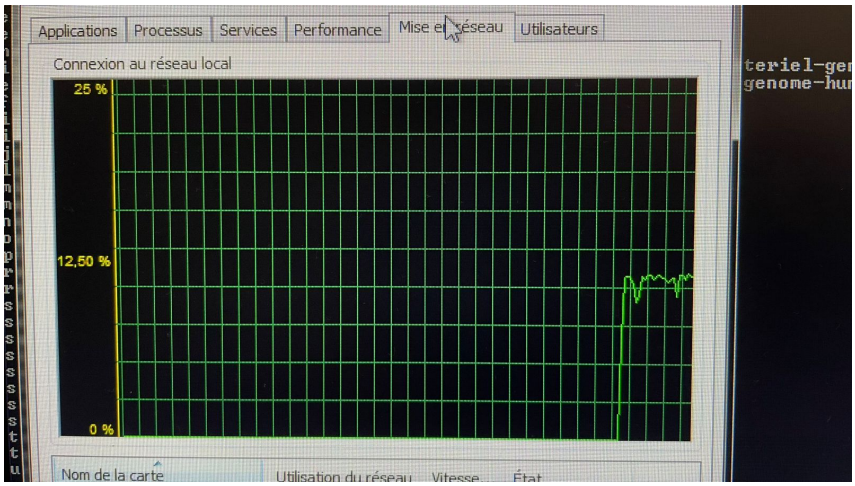

FIGURE 18 – Gestionnaire des tâches Windows - Impact réseau lors de l'attaque UDP

Observations :

— **Pic initial : 25%** d'utilisation réseau (similaire au SYN ;

— **Stabilisation :** $\sim 12,50\%$ après quelques secondes ;

— **Activité soutenue :** Le trafic UDP continue de saturer partiellement le réseau ;

2.4.4 Interprétation des Résultats - Attaque UDP

Impact constaté :

— **Surcharge CPU élevée :** 89% (plus élevé que le SYN flood) ;

— **Saturation réseau :** 25% d'utilisation initiale, puis $\sim 12,50\%$;

— **Consommation mémoire réduite :** 678 Mo (contre 1,01 Go pour SYN) ;

— **Moins de connexions ouvertes :** UDP ne crée pas de connexions persistantes ;

Différence avec le SYN Flood :

— **UDP :** Plus d'impact CPU car le système doit traiter rapidement chaque paquet sans maintenir de connexion ;

— **SYN :** Plus d'impact mémoire car les connexions semi-ouvertes s'accumulent ;

— **UDP :** Le port 53 (DNS) n'étant pas en écoute, Windows rejette les paquets, ce qui consomme du CPU ;

2.5 Comparaison des Attaques DoS

Critère	**SYN Flood**	**UDP Flood**
Protocole	TCP	UDP
Port ciblé	80 (HTTP)	53 (DNS)
Paquets envoyés	17 324 730	14 472 401
Impact CPU	61%	89%
Impact Réseau (pic)	25%	25%
Consommation Mémoire	1,01 Go	678 Mo
Type d'épuisement	Connexions + Mémoire	CPU + Bande passante
Connexions persistantes	Oui (semi-ouvertes)	Non (sans connexion)

TABLE 1 – Comparaison des impacts des attaques SYN Flood et UDP Flood

2.6 Analyse Technique des Attaques

2.6.1 Pourquoi le CPU est-il plus élevé pour UDP ?

Le CPU est plus sollicité lors de l'attaque UDP (89%) que lors du SYN flood (61%) pour les raisons suivantes :

— **Traitement immédiat :** Chaque paquet UDP doit être traité instantanément (pas de file d'attente de connexions) ;

— **Port fermé :** Le port 53 (DNS) n'étant pas en écoute sur Windows 7 ;

— **Absence de limitation :** UDP n'a pas de mécanisme de contrôle de flux comme TCP ;

2.6.2 Pourquoi la Mémoire est-elle plus élevée pour SYN ?

La consommation mémoire est plus importante lors du SYN flood (1,01 Go) car :

— **Connexions semi-ouvertes :** Windows maintient en mémoire chaque connexion TCP en attente de l'ACK final ;

— **Table de connexions :** La SYN flood remplit la table des connexions TCP (backlog queue) ;

— **Timeout long :** Les connexions semi-ouvertes restent en mémoire jusqu'au timeout (plusieurs secondes) ;

3 Conclusion

3.1 Synthèse des Observations

Ce TP nous a permis de comprendre pratiquement les techniques d'attaque réseau et leurs impacts sur les systèmes cibles :

3.1.1 Partie 1 : Capture et Analyse de Paquets

— **tcpdump** est un outil puissant pour capturer et analyser le trafic réseau en ligne de commande ;

— Les filtres permettent d'isoler des protocoles spécifiques (ARP, HTTP, FTP, TCP, UDP) ;

— **Wireshark** offre une interface graphique pour une analyse approfondie ;

— **Ettercap** permet l'interception et l'analyse de protocoles comme Telnet ou FTP ;

3.1.2 Partie 2 : Attaques DoS

Les attaques par déni de service testées ont montré des impacts significatifs :

Impact	**SYN Flood**	**UDP Flood**
CPU	61%	**89%**
Réseau (pic)	25%	25%
Mémoire	**1,01 Go**	678 Mo
Type d'attaque	Épuisement connexions	Saturation CPU

Table 2 – Récapitulatif des impacts observés

3.2 Enseignements sur la Sécurité Réseau

3.2.1 Importance de la Triade CIA

Disponibilité (Availability) :

— Les attaques DoS visent directement la **disponibilité** des services

— Un système sous attaque ne peut plus servir les utilisateurs légitimes

— La disponibilité est aussi importante que la confidentialité et l'intégrité

Confidentialité et Intégrité :

— Bien que ce TP se concentre sur la disponibilité, les captures de paquets montrent l'importance du chiffrement

— Les protocoles non chiffrés (HTTP, FTP) exposent les données sensibles

3.3 Perspectives

Ce TP a démontré l'efficacité des attaques DoS même avec un seul attaquant. Les attaques **DDoS** (Distributed), utilisant des botnets de milliers de machines, sont encore plus dévastatrices et difficiles à mitiger.