



# RAPPORT LSAAM

## **1 / Préparation de l'environnement : machines ciblés et environnement de coding :**

## **1. INSTALLATION DE BIBLIOTHEQUES SCAPY : ( biblio responsable sur l'envoi des paquets)**

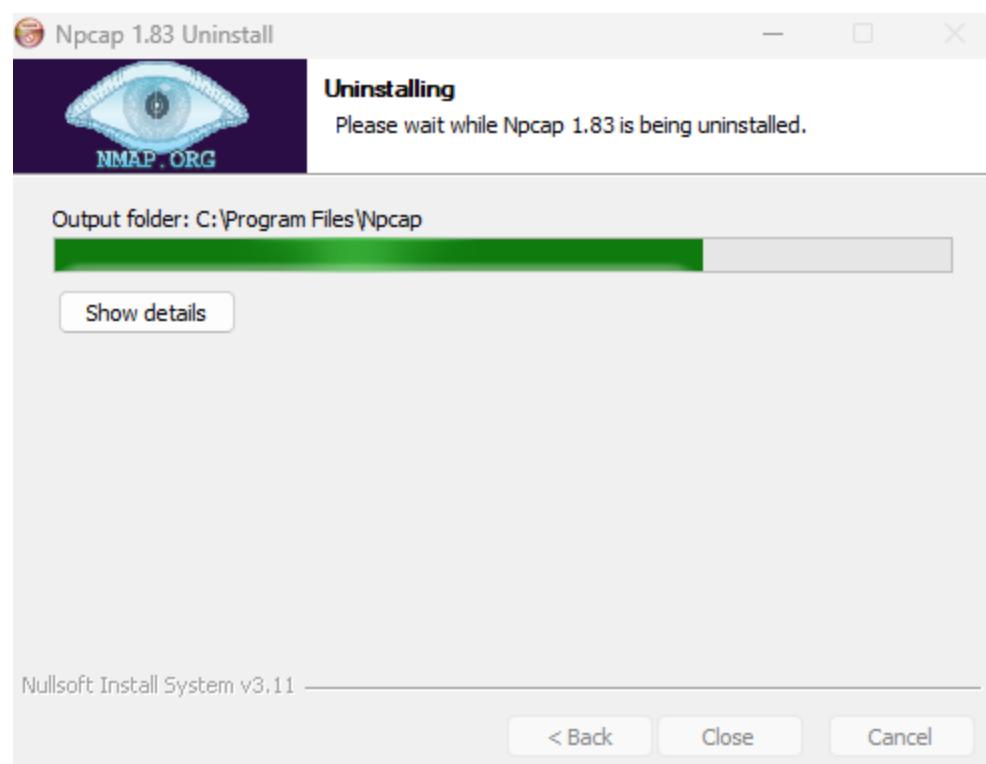
```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

PS C:\Users\HP> pip install scapy
pip : Le terme «pip» n'est pas reconnu comme nom d'applet de commande, fonction, fichier de script ou programme exécutable. Vérifiez l'orthographe du nom, ou si un chemin d'accès existe, vérifiez que le chemin d'accès est correct et réessayez.
Au caractère Ligne:1 : 1
+ pip install scapy
+ ~~~~  
+ CategoryInfo          : ObjectNotFound: (pip:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\HP> python -m pip install scapy
Collecting scapy
  Downloading scapy-2.7.0-py3-none-any.whl.metadata (5.8 kB)
  Downloading scapy-2.7.0-py3-none-any.whl (2.6 MB)
    2.6/2.6 MB 1.0 MB/s 0:00:02
Installing collected packages: scapy
```

( pip seulement ne fonctionne pas , problème de chemin cad que le système ne connaît pas ou python est installé (il faut modifier les variables d'environnement de Path )

2. ***INSTALLATION DE Npcap : Sous Windows , Scapy ne peut pas envoyer de paquets sans un petit pilote appelé Npcap.***



3. **INSTALLATION DE METASPLOITABLE** : (*machine dédiée à être vulnérabilisé par sa spécification de ports ouverts*) la faire une mise à jour pour qu'il soit compatible avec la version de workstation vm 17 et après on fait le log-in par défaut msfadmin / msfadmin

```
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]

[----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
```

```
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

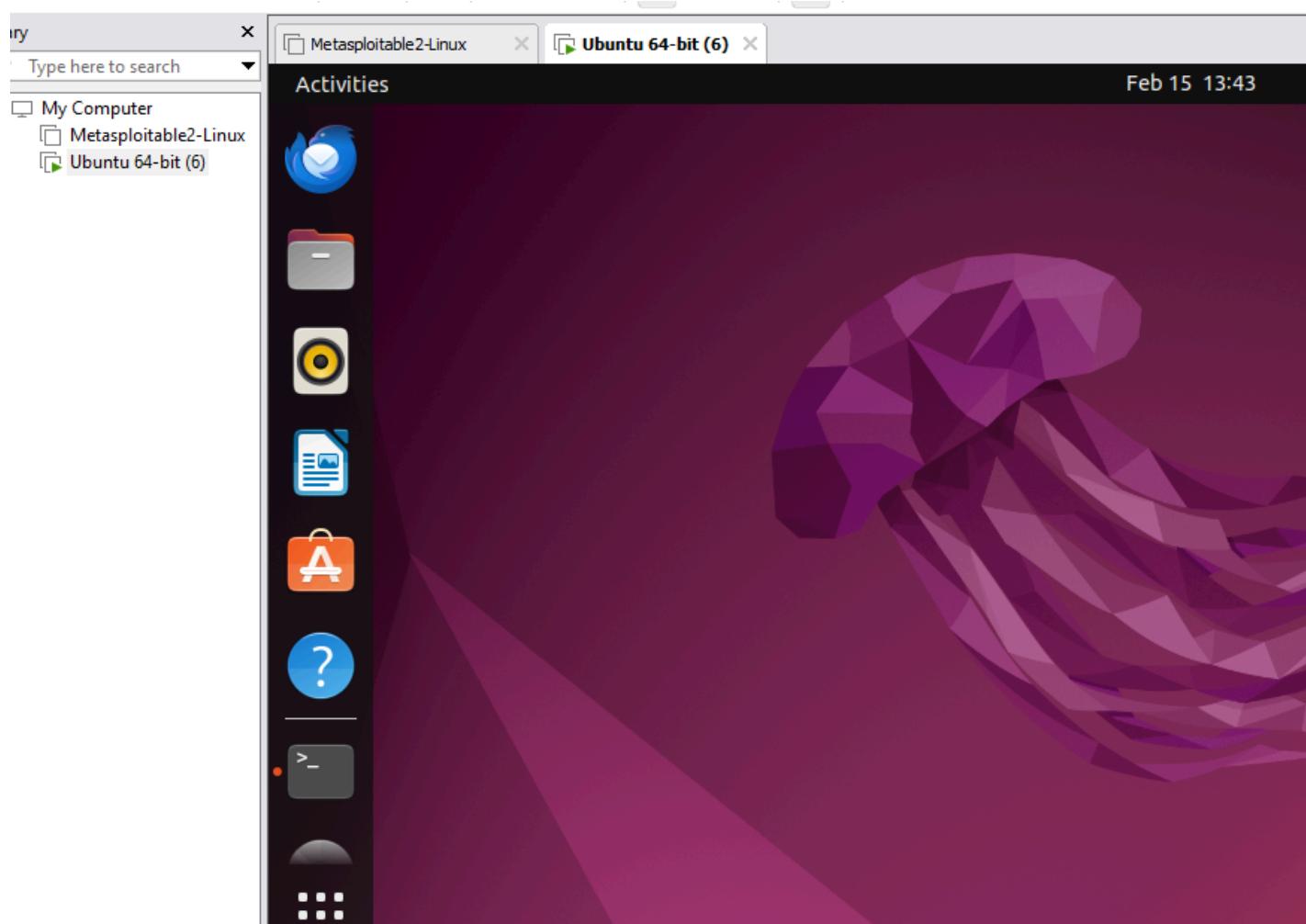
metasploitable login: msfadmin
Password:
Last login: Sun May 20 15:50:42 EDT 2012 from 172.16.123.1 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

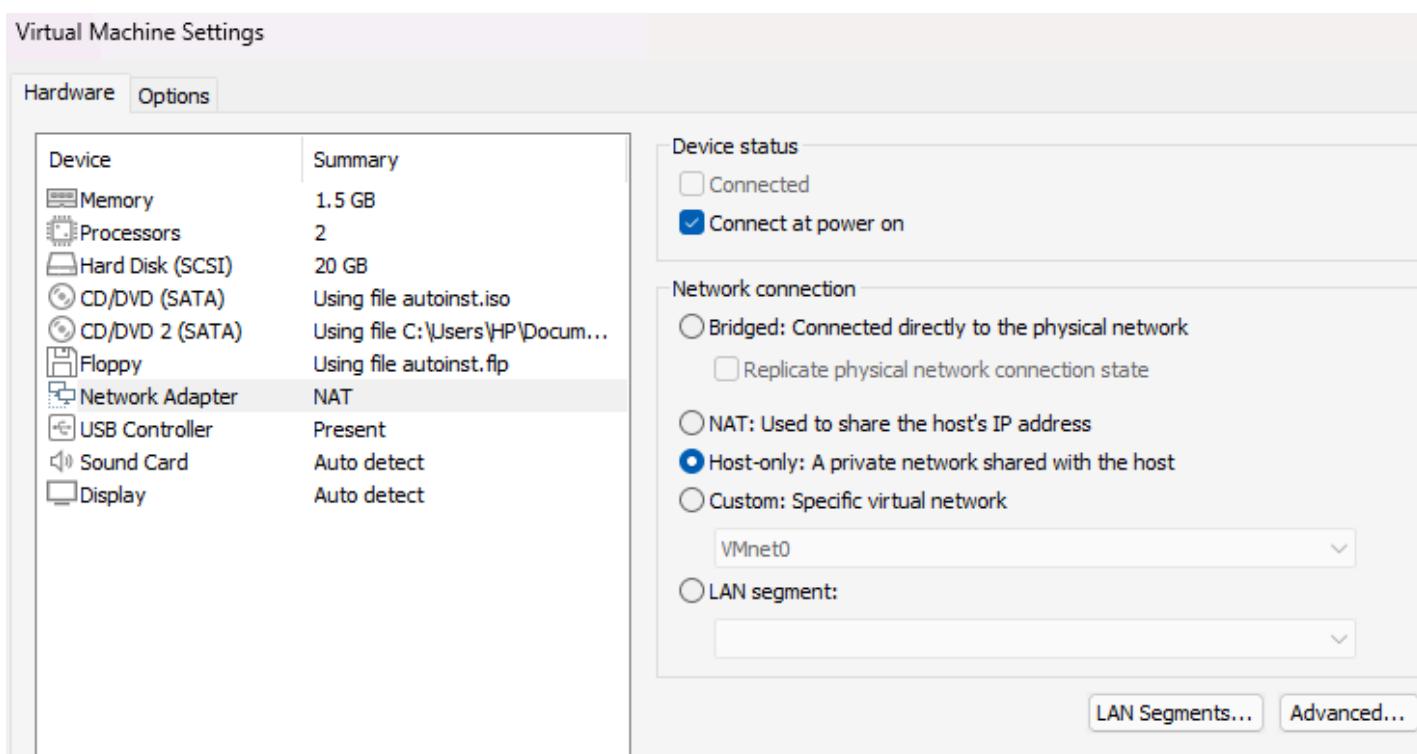
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

#### 4. Une deuxième cible mais cette fois ci une machine vm avec le OS UBUNTU:



### 5. **NB:**

Pour avoir un réseau local et privé et sécurisé et en respectant les mesures légales (ethical) j utilise alors ma machine et les deux machines vm , on configure les deux machines en mode “HOST-ONLY”



6. *En vue d'avoir l'adresse de mon réseau privé (réseau entre ma machine et les deux machines VM) on utilise la commande IPCONFIG :*

c'est exactement celle de VMNET 1: `addr reseau`, c'est alors : “`192.168.154.0/24`”

(VMnet1 c'est la carte réseau virtuelle des machines configurées en mode HOST-ONLY et VMnet8 celles configurées en mode NAT )

## **7. Vérification de l'appartenance des machines au même réseau privé :**

## METHODE 1 :

A l'aide de la commande **IFCONFIG**, on vérifie bien l'adresse IP de la première cible métasploitable est-ce qu'il appartient à mon réseau privé ?

c est exactement celle mentionnée dans la carte réseau eth0 : **inet addr 192.168.154.131**

```
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr 00:0c:29:b2:ba:26  
          inet addr:192.168.154.131 Bcast:192.168.154.255 Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:feb2:ba26/64 Scope:Link  
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
            RX packets:6 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:49 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1000  
            RX bytes:926 (926.0 B) TX bytes:5394 (5.2 KB)  
            Interrupt:17 Base address:0x2000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1 Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
            UP LOOPBACK RUNNING MTU:16436 Metric:1  
            RX packets:98 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:98 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:0  
            RX bytes:21621 (21.1 KB) TX bytes:21621 (21.1 KB)  
msfadmin@metasploitable:~$ _
```

de même pour la deuxième cible, est ce qu'il appartient bien à mon adresse réseau privé

```
Activities Terminal Feb 16 16:24  
ubuntu@ubuntu-virtual-machine:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 192.168.154.130 netmask 255.255.255.0 broadcast 192.168.154.255  
      inet6 fe80::3683:a6a1:323e:b694 prefixlen 64 scopeid 0x20<link>  
        ether 00:0c:29:1a:95:e1 txqueuelen 1000 (Ethernet)  
        RX packets 525 bytes 52628 (52.6 KB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 1360 bytes 119221 (119.2 KB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
      inet 127.0.0.1 netmask 255.0.0.0  
      inet6 ::1 prefixlen 128 scopeid 0x10<host>  
        loop txqueuelen 1000 (Local Loopback)  
        RX packets 764 bytes 60186 (60.1 KB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 764 bytes 60186 (60.1 KB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
ubuntu@ubuntu-virtual-machine:~$
```

## METHODE 2 :

OU bien on vérifie à l'aide d'un **PING** entre ma machine et la machine metasploitable ciblé à savoir si le canal est bien établi .

>>>>PING bien établie

```
C:\Users\HP>ping 192.168.154.131

Envoi d'une requête 'Ping' 192.168.154.131 avec 32 octets de données :
Réponse de 192.168.154.131 : octets=32 temps=16 ms TTL=64
Réponse de 192.168.154.131 : octets=32 temps<1ms TTL=64
Réponse de 192.168.154.131 : octets=32 temps<1ms TTL=64
Réponse de 192.168.154.131 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.154.131:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 16ms, Moyenne = 4ms

C:\Users\HP>
```

>>>>>> *ON REMARQUE bien que tous les machines appartiennent au même réseau privé*

## 2/ Scripts en Python : Découverte du réseau via SCAPY et scan des ports TCP via SOCKET :

1 >>> Etablir deux fonctions en python , une indispensable de la découverte du réseau et l'autre indispensable du scan des ports ouverts

```
main1.py
13
14     def decouverte_reseau(ip_range):
15         print("[1] Recherche des machines sur ",ip_range)
16         requete_arp = ARP(pdst=ip_range)
17         broadcast = Ether(dst="ff:ff:ff:ff:ff:ff")
18         paquet_complet = broadcast / requete_arp
19
20         reponses = srp(paquet_complet, timeout=2, verbose=False)[0]
21
22         machines_trouvees = []
23
24         print(len(reponses)," machine(s) détectée(s).")
25
26         for envoye, recu in reponses:
27             machine = {"ip": recu.psrc, "mac": recu.hwsrc}
28             machines_trouvees.append(machine)
29             print("* Trouvé : ",recu.psrc, "| MAC : ",recu.hwsrc)
30
31     return machines_trouvees
```

```
main1.py
13     C:\Users\HP\OneDrive\Desktop\PROJET LSAAM\main1.py
14     ports_comuns = {
15         80: "HTTP",
16         443: "HTTPS",
17         3306: "MySQL",
18         25: "SMTP",
19         53: "DNS",
20     }
21     for port, service in ports_comuns.items():
22         soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
23         soc.settimeout(0.5)
24         resultat = soc.connect_ex((ip_cible, port))
25
26         if resultat == 0:
27             info = f" Port {port} {service} OUVERT"
28             print(info ,"\n")
29             resultats_machine.append(info)
30             if port in [21, 23]:
31                 alerte = f"      !ALERTE : {service} n'est pas chiffré (Risque d'écoute) ! \n"
32                 print("      ",alerte)
33                 resultats_machine.append(alerte)
34             soc.close()
35
36     return resultats_machine
```

et apres on regroupe tout ceci dans un seul fichier **main.py**

```
❶ main1.py
1   from scapy.all import ARP, Ether, srp, conf
2   import socket
3   import json
4   import os
5   from datetime import datetime
6
7
8   conf.iface = "VMware Virtual Ethernet Adapter for VMnet1"
9   plage_reseau = "192.168.154.0/24"
10  nom_fichier = "inventaire.json"
11
12
13
14  def decouverte_reseau(ip_range):
15      print("[1] Recherche des machines sur ",ip_range)
16      requete_arp = ARP(pdst=ip_range)
17      broadcast = Ether(dst="ff:ff:ff:ff:ff:ff")
18      paquet_complet = broadcast / requete_arp
19
20      reponses = srp(paquet_complet, timeout=2, verbose=False)[0]
21
22      machines_trouvees = []
23
24      print(len(reponses)," machine(s) détectée(s).")
25
26      for envoye, recu in reponses:
27          machine = {"ip": recu.psrc, "mac": recu.hwsrc}
28          machines_trouvees.append(machine)
29          print("* Trouvé : ",recu.psrc, "| MAC : ",recu.hwsrc)
30
31  return machines_trouvees
32
❷ 0
```

```
❸ def audit_ports(ip_cible):
    print("\n[2] Analyse des services sur ",ip_cible)
    resultats_machine = []

    ports_comuns = {
        21: "FTP",
        22: "SSH",
        23: "Telnet",
        80: "HTTP",
        443: "HTTPS",
        3306: "MySQL",
        25: "SMTP",
        53: "DNS",
    }
    for port, service in ports_comuns.items():
        soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        soc.settimeout(0.5)
        resultat = soc.connect_ex((ip_cible, port))

        if resultat == 0:
            info = f" Port {port} {service} OUVERT"
            print(info ,"\n")
            resultats_machine.append(info)
            if port in [21, 23]:
                alerte = f" !ALERTE : {service} n'est pas chiffré (Risque d'écoute) ! \n"
                print(" ",alerte)
                resultats_machine.append(alerte)
        soc.close()

    return resultats_machine
❹ 0
```

## >>>>> Difficultés rencontrées :

lors du running de code global , j'avais toujours le message suivant !!!!

```

PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM> py main.py
--- Recherche des machines sur 192.168.154.0/24 ---
Machines trouvées :
Aucune machine trouvée. Vérifie ta configuration VMware !
PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM>

```

et tandis que la vérification des address ip manuellement montre bien qu'ils sont sur meme reseau mais aussi le passage du ping entre les machines qui force bien la configuration vm , alors c'etait probablement un problème des biblio et on dirait en global probleme sur mes scripts

C'est pourquoi j'ai essayé plusiuers tests :

#### >>Premier test : la désactivation du pare-feu windows

On dirait que le pare feu bloque le passage des requets ARP pour qu'ils circulent vers les cartes reseau des deux machines vm ,



*mais ce n'était pas ca exactement .*

#### >>Deusieme test : Quelle interface reseau utilise la biblio scapy ?

On vérifie alors quelle interface reseau utilise la biblio scapy , est-ce celle pas défaut (WIFI) ?

```

PROJET LSAAM > ⏷ test_scapy.py
1
2
3 from scapy.all import show_interfaces, conf
4 print("--- Liste des interfaces détectées ---")
5 show_interfaces()
6
7 print(f"\nInterface par défaut actuelle : {conf.iface}")
8 from scapy.all import conf
9 print(conf.use_pcap)

```

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
--- Liste des interfaces détectées ---
Source Index Name MAC IPv4 IPv6
libpcap 1 Software Loopback Interface 1 00:00:00:00:00:00 127.0.0.1 ::1
libpcap 10 WAN Miniport (Network Monitor)
libpcap 11 Microsoft Wi-Fi Direct Virtual Adapter 82:30:49:84:2b:cd 169.254.110.133 fe80::86d3:9262:ea96:bce3
libpcap 12 VMware Virtual Ethernet Adapter for VMnet1 VMware:c0:00:01 192.168.154.1 fe80::15bd:eb70:3b98:3ad5
libpcap 13 WAN Miniport (IP)
libpcap 16 WAN Miniport (IPv6)
libpcap 17 Qualcomm Atheros QCA9377 Wireless Network_ LiteonTechno:84:2b:cd 192.168.8.146 fda2:f851:ca69:a800:5cd4:aab3:c414:6167
fe80::16d4:560b:feff:1f2f
fd80::f851:ca69:a800:af2f:ba77:5b55:308c
libpcap 18 Microsoft Wi-Fi Direct Virtual Adapter #2 92:30:49:84:2b:cd 169.254.6.192 fe80::4615:c6f3:7639:b82c
libpcap 19 Realtek(R) PCI(e) Ethernet Controller CompaInform:44:e1:57 169.254.82.218 fe80::aea9:c91a:1e1e:2128
libpcap 3 VMware Virtual Ethernet Adapter for VMnet8 VMware:c0:00:08 192.168.75.1 fe80::4d2c:3776:e343:4332

```

Activer Windows  
Interface par défaut actuelle : \Device\NPF\_{EAE8B657-E9F0-433C-ADDB-DE77FD15EDA0} Accédez aux paramètres pour activer Windows.

PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM> py main.py

Ln 14, Col 33 Spaces: 4 UTF-8 CRLF {} Python 3.14.0

On remarque bien que la biblio `sacpy` en fait utilise par défaut l'interface `WIFI`, donc les requêtes ARP ne seront jamais envoyés vers mon réseau privé local qui utilise l'interface `Vmware` ce qui explique la non découverte des machines sur mon réseau

donc le problème se réfère à la non déclaration de l'interface réseau que la biblio `SCAPY` va utiliser et c'est exactement celle qui a l'index 12 intitulé "VMware Virtual Ethernet Adapter for VMnet1"

>>Troisième test : Essayer par la fonction `arping()`

c'est une fonction naïve responsable sur l'envoi des requêtes ARP on lui passant la plage adresse

```

PROJET LSAAM > der_test.py
1
2
3 from scapy.all import arping, conf
4
5 conf.iface = 12
6 print("Tentative avec la fonction native arping...")
7 arping(["192.168.154.0/24"])

```

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM> py der_test.py
Tentative avec la fonction native arping...
Begin emission
*****
Finished sending 256 packets

Received 4 packets, got 4 answers, remaining 252 packets
src      manuf   psrc
00:50:56:c0:00:01  VMware  192.168.154.1
00:0c:29:1a:95:e1  VMware  192.168.154.130
00:0c:29:b2:ba:26  VMware  192.168.154.131
00:50:56:e4:6e:14  VMware  192.168.154.254
PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM>

```

ceci alors montre que tout se passe bien si jamais on utilise la fonction `arping()` avec l'ajout de `conf.iface` ce qui montre alors que le problème était bien le non ajout de l'interface réseau grâce à `conf.iface`

>>>>>>>>>> La solution du problème :

c'est exactement l'ajout de l'interface pour que notre biblio `scapy` sait où adresser les paquets ARP, on ajoute la variable `conf.iface` et on l'infecte le bon chemin

```

1  from scapy.all import ARP, Ether, srp, conf
2  import socket
3  import json
4  import os
5  from datetime import datetime
6
7
8  conf.iface = "VMware Virtual Ethernet Adapter for VMnet1"

```

### 3/ Runing du code :

Après avoir regler les problèmes rencontrés on essaye alors de faire run a nnotre code main.py

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
C:\Users\HP\OneDrive\Desktop\PROJET LSAAM\test_scapy.py > py main.py
--- Recherche des machines sur 192.168.154.0/24 ---
Machines trouvées :
IP: 192.168.154.1 | MAC: 00:50:56:c0:00:01
IP: 192.168.154.130 | MAC: 00:0c:29:1a:95:e1
IP: 192.168.154.131 | MAC: 00:0c:29:b2:ba:26
IP: 192.168.154.254 | MAC: 00:50:56:e4:6e:14

[?] Analyse des services sur 192.168.154.1 ...
[?] Analyse des services sur 192.168.154.130 ...
[?] Analyse des services sur 192.168.154.131 ...
Port 21 ( FTP ),: OUVERT
  /!\ Risque détecté : FTP n'est pas chiffré !
Port 22 ( SSH ),: OUVERT
Port 23 ( Telnet ),: OUVERT
  /!\ Risque détecté : Telnet n'est pas chiffré !
Port 80 ( HTTP ),: OUVERT
Port 3306 ( MySQL ),: OUVERT

[?] Analyse des services sur 192.168.154.254 ...
PS C:\Users\HP\OneDrive\Desktop\PROJET LSAAM>
```

On remarque bien que on detecte tout les machines du reseau privé mais aussi des différentes ports ouverts des machines apres avoir les scanner , et spécialement la machine metasploitable qui l adresse IP 169.168.175.131 , le scann a detecter effectivement plusieurs ports ouverts

donc on peut dire alors que ca marche bien

### 3/ Gestion de la mémoire de l'outil JSON (gestion d'Assets) et gener les fichiers d'audit texte

dans le but non seulement de faire un scan comme ca et partir cad on vérifie sans avoir stocker les résultats de notre scan c est pourquoi on essaye de stocker ces résultats a l'aide d'une base de données , et a chaque fois qu on cherche a les retourner il se fait de la consulter et comme ca on peut faire des comparaisons et des analyses si jamais on avait des attaques

la base de donnees utilise est celle intégrée dans le langage de programmation Python , c est bien les fichier avec l extension .JSON , sa caractéristique de garder les résultats seulement du dernier scan réalisé nous aidera après dans la génération des fichiers d'audit textes , cela en fait aidera beaucoup en terme de gestion administratif et plutôt gestion audit , car au lieu d'avoir des lignes énormes devant ses yeux il aura alors un fichier distinct qui porte le nom du jour et heure ou le scan a été réalisé .

pour ce but , on crie une fonction sauvegarder\_inventaire\_json() qui va générer le fichier .JSON qui est notre base de donnés instantanée et au même temps il va nous generer l'affectation a notre fichier d'audit qui portera en fait le extension .txt

```

def sauvegarder_inventaire_json(ip, mac, ports):
    nom_fichier = "inventaire.json"

    if os.path.exists(nom_fichier) and os.path.getsize(nom_fichier) > 0:
        try:
            with open(nom_fichier, "r", encoding="utf-8") as f:
                inventaire = json.load(f)
        except json.JSONDecodeError:
            inventaire = {}
    else:
        inventaire = {}

    if mac not in inventaire:
        print("\n[!] NOUVEL ASSET DÉTECTÉ : ", ip)

    inventaire[mac] = {
        "ip": ip,
        "mac": mac,
        "derniere_vue": datetime.now().strftime("%d/%m/%Y %H:%M:%S"),
        "services": ports
    }

    with open(nom_fichier, "w", encoding="utf-8") as f:
        json.dump(inventaire, f, indent=4, ensure_ascii=False)

print("====")

```

```

ain1.py
print( "===== ")
print("      LSAAM - LOCAL SECURITY AUDITOR & ASSETS MONITORING      ")
print("===== ")

liste_machines = decouverte_reseau(plage_reseau)

if not liste_machines:
    print("\n Aucune machine n'a répondu. Fin de l'audit.")
else:
    date_heure = datetime.now().strftime("%d-%m-%Y %H_%M_%S")
    nom_fichier_unique = f"rapport_audit_{date_heure}.txt"

with open(nom_fichier_unique, "w", encoding="utf-8") as f:
    f.write("RAPPORT D'AUDIT DU "+date_heure+"\n")
    f.write("*50 + "\n\n")

    for machine in liste_machines:
        liste_ports_ouverts = audit_ports(machine["ip"])

        sauvegarder_inventaire_json(machine["ip"], machine["mac"], liste_ports_ouverts)

        f.write("CIBLE : " + machine['ip'] + "(MAC: "+machine['mac']+") \n")
        if liste_ports_ouverts:
            for ligne in liste_ports_ouverts:
                f.write(ligne+"\n")
        else:
            f.write(" Aucun service critique détecté.\n")
    f.write("-" * 30 + "\n\n")

```

```

        sauvegarder_inventaire_json(machine["ip"], machine["mac"], liste_ports_ouverts)

        f.write("CIBLE : " + machine['ip'] + "(MAC: "+machine['mac']+") \n")
        if liste_ports_ouverts:
            for ligne in liste_ports_ouverts:
                f.write(ligne+"\n")
        else:
            f.write(" Aucun service critique détecté.\n")
        f.write("-" * 30 + "\n\n")

        print("\n" + "*50)
        print(" AUDIT TERMINÉ !")
        print("    -> Mémoire mise à jour : ",nom_fichier)
        print("    -> Rapport généré : ",nom_fichier_unique)
        print("*50)

```

>>>>>>>>>>> Exemple du fichier .JSON :

The screenshot shows a file explorer window with a file named 'inventaire' and a code editor window with a file named 'main1.py'. The 'inventaire' file is a JSON file containing network device information. The 'main1.py' code is a Python script that performs an audit and saves the results to a JSON file.

```

{
    "00:0c:29:1a:95:e1": {
        "ip": "192.168.154.130",
        "mac": "00:0c:29:1a:95:e1",
        "derniere_vue": "19/02/2026 02:26:51",
        "services": []
    },
    "00:0c:29:b2:ba:26": {
        "ip": "192.168.154.131",
        "mac": "00:0c:29:b2:ba:26",
        "derniere_vue": "19/02/2026 02:26:52",
        "services": [
            " Port 21 FTP OUVERT",
            " !ALERTE : FTP n'est pas chiffré (Risque d'écoute) ! \n",
            " Port 22 SSH OUVERT",
            " Port 23 Telnet OUVERT",
            " !ALERTE : Telnet n'est pas chiffré (Risque d'écoute) ! \n",
            " Port 80 HTTP OUVERT",
            " Port 3306 MySQL OUVERT",
            " Port 25 SMTP OUVERT"
        ]
    },
    "00:50:56:f9:10:50": {
        "ip": "192.168.154.254",
        "mac": "00:50:56:f9:10:50",
        "derniere_vue": "17/02/2026 23:35:20",
        "services": []
    },
    "00:50:56:f4:ce:64": {
        "ip": "192.168.154.254",
        "mac": "00:50:56:f4:ce:64",
        "derniere_vue": "19/02/2026 00:09:17",
        "services": []
    }
}

```

>>>>>>>>>> Exemple d'un fichier d'audit .txt :

Modifié le : 10/12/2024 07:46

rapport\_audit\_19-02-2026\_02\_26\_42 C:\Utilisateurs\HP\OneDrive\Bureau\PROJET LSAAM Taille : 797 octet(s)  
Modifié le : 19/02/2026 02:26 RAPPORT D'AUDIT DU 19-02-2026\_02\_26\_42 = CIBLE : 192.168.154.1(MAC: 00:50:56:c0:00:01) Aucun service crit...

fct audit ports Modifié le : 15/02/2026 15:04

```
RAPPORT D'AUDIT DU 19-02-2026_02_26_42
=====
CIBLE : 192.168.154.1(MAC: 00:50:56:c0:00:01)
Aucun service critique détecté.
-----
CIBLE : 192.168.154.130(MAC: 00:0c:29:1a:95:e1)
Aucun service critique détecté.
-----
CIBLE : 192.168.154.131(MAC: 00:0c:29:b2:ba:26)
Port 21 FTP OUVERT
!ALERTE : FTP n'est pas chiffré (Risque d'écoute) !

Port 22 SSH OUVERT
Port 23 Telnet OUVERT
!ALERTE : Telnet n'est pas chiffré (Risque d'écoute) !

Port 80 HTTP OUVERT
Port 3306 MySQL OUVERT
Port 25 SMTP OUVERT
-----
CIBLE : 192.168.154.254(MAC: 00:50:56:e6:a8:98)
Aucun service critique détecté.
```

Ce travail permettra au personne chargé d'audit de bien organiser son travail mais aussi un enregistrement ciblé .

>> Réalisé par :

**El BAHRI Saida**