Chapitre 3
Collecter les évidences

1. Introduction

La collecte d'informations a pour but de permettre la rédaction du rapport, mais aussi d'être en mesure de répondre ultérieurement aux questions pouvant survenir sur le déroulement de la mission.

2. Utilisation et organisation autour d'espaces de travail dédiés

Pour faciliter la collecte et éviter la présence d'informations inutiles, il est nécessaire de penser, étape par étape, à ce dont nous aurons besoin par la suite.

Il peut s'agit par exemple, des adresses IP des cibles, des types de système ainsi que, lorsque cela est possible, des VHOSTS associés ou encore des types de services et des ports d'écoute utilisés par machine (comme RDP/3389 TCP ou Apache Tomcat/8080 TCP).

Ce type d'information sera collecté au travers de scans dès la phase préliminaire dite de reconnaissance active.

Par la suite, d'autres informations, plus cruciales encore, devront également être conservées, notamment :

– la vulnérabilité identifiée et/ou exploitée sur la cible ;

– l'identifiant de connexion valide sur la machine (login et mot de passe) ;

– l'adresse MAC, le type de hardware…

Afin de faciliter cette collecte d'information, et dans un souci d'efficacité, nous allons utiliser un outil connu et reconnu depuis ses débuts : Metasploit (ou MSF). Il offre une vaste gamme d'outils de sécurité offensive particulièrement adaptés à notre métier.

À ce stade, nous nous intéresserons plus particulièrement à sa fonctionnalité d'organisation en espaces de travail (workspaces).

Remarque

Cet ouvrage n'ayant pas vocation à enseigner l'utilisation du framework, nous proposerons régulièrement des liens vers des ressources utiles et inviterons les lecteurs à explorer cette boîte à outils particulièrement riche, à laquelle de nombreux ouvrages ont déjà été consacrés.

Commençons par créer un espace de travail nommé BOOK.

Cette création s'effectue simplement en trois étapes :

– le démarrage de la BDD PostgreSQL ;

– le lancement de MSF ;

– la création du workspace.

© Editions ENI - All rights reserved

```
root@kmkz-pc:~# service postgresql start
root@kmkz-pc:~# msfconsole -q
msf6 > workspace
  LIVE-FIRE
  P-Ext
* default
msf6 > workspace -a BOOK
[*] Added workspace: BOOK
[*] Workspace: BOOK
msf6 > workspace
  LIVE-FIRE
  P-Ext
  default
* BOOK
msf6 >
```

Création de l'espace de travail BOOK

Par défaut, MSF vous place automatiquement sur l'espace de travail (l'astérisque devant les informations présentées le montre), ce qui signifie que tout ce qui est lancé depuis la console MSF est automatiquement enregistré et pourra être consulté ultérieurement.

Cependant, la vie de Pentester ne se limite pas à l'utilisation d'une boîte à outils. Ainsi tout a été pensé pour importer des données afin de permettre de centraliser les informations dans MSF pour les phases d'exploitation et de post-exploitation à venir, tout en facilitant la recherche en triant efficacement. C'est ce que montre le menu d'aide du workspace ci-dessous :

```
msf6 > workspace -h
Usage:
    workspace            List workspaces
    workspace [name]     Switch workspace

OPTIONS:

    -a, --add <name>          Add a workspace.
    -d, --delete <name>       Delete a workspace.
    -D, --delete-all          Delete all workspaces.
    -h, --help                Help banner.
    -l, --list                List workspaces.
    -r, --rename <old> <new>  Rename a workspace.
    -S, --search <name>       Search for a workspace.
    -v, --list-verbose        List workspaces verbosely.
```

Tri et import/export de données dans le workspace

Exemple

Prenons l'exemple d'un simple scan de machine Windows ayant un port SMB (**445** TCP) exposé, situation fréquente en interne, qui est un des services des plus recherchés. Nous verrons pourquoi par la suite.

Nous pouvons par exemple lancer un scanner tel que Nmap (de nombreux ouvrages en couvrent également toute la puissance) contre une machine cible ayant l'adresse **192.168.178.64** et importer le résultat dans la base de données MSF.

Pour ce faire, nous utiliserons le format de sortie XML proposé par Nmap (option `-oX` pour `Output Xml`) qui donnera la sortie suivante :

```
nmap 192.168.178.64 -Pn -sS -sV -p 445 -oX /tmp/scan-smb-tcp-
445.xml
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-19 17:03 CEST
Nmap scan report for DESKTOP-084JG1D.fritz.box (192.168.178.64)
Host is up (0.00020s latency).

PORT    STATE SERVICE      VERSION
445/tcp open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds
(workgroup: WORKGROUP)
MAC Address: 08:00:27:C8:C2:EA (PCS Systemtechnik/Oracle
VirtualBox virtual NIC)
Service Info: Host: DESKTOP-084JG1D; OS: Windows; CPE:
cpe:/o:microsoft:windows
```

Une fois le scan effectué et la sortie stockée dans un fichier format XML, il reste à importer le fichier (option `db_import`), à vérifier que tout est bien en ordre et correspond à la sortie de Nmap visible ci-dessus.

Remarque

L'ensemble des options utilisées (-sS, -Pn ou encore -sV) sont disponibles dans le chapitre dédié Ressources et références en fin d'ouvrage.

© Editions ENI - All rights reserved

```
root@kmkz-pc:~# msfconsole -q
msf6 > workspace BOOK
[*] Workspace: BOOK
msf6 > db_import /tmp/scan-smb-tpc-445.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.168.178.64
[*] Successfully imported /tmp/scan-smb-tpc-445.xml
msf6 > hosts

Hosts
=====

address         mac                name                      os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----                      -------  ---------  -----  -------  ----  --------
192.168.178.64  08:00:27:c8:c2:ea  DESKTOP-084JG1D.fritz.box  Unknown                    device

msf6 > services
Services
========

host            port  proto  name          state  info
----            ----  -----  ----          -----  ----
192.168.178.64  445   tcp    microsoft-ds  open   Microsoft Windows 7 - 10 microsoft-ds workgroup: WORKGROUP

msf6 > vulns

Vulnerabilities
===============

Timestamp  Host  Name  References
---------  ----  ----  ----------
```

Informations stockées en BDD dans MSF

Cet exemple est assez bref, mais cette façon de faire permet, dans de nombreux cas, d'éviter de scanner plusieurs fois les mêmes machines et d'exporter les données pour les stocker en fichier Excel pour le suivi du patching sur de larges périmètres. C'est une façon de gagner du temps tout en étant professionnel et paré à toute éventualité.

Autre atout majeur : la fonctionnalité `creds`. Comme son nom l'indique, elle permet de stocker automatiquement les identifiants de connexion (credentials) sous différents formats. Il devient ainsi possible de déterminer précisément quels comptes ont été compromis et dans quel contexte, ou encore de procéder au cassage de hash lorsque cela s'avère nécessaire – une situation toutefois rare, que nous aborderons ultérieurement dans la partie consacrée à la post-exploitation.

```
msf6 > creds
Credentials
===========

host            origin  service        public  private     realm  private_type  JtR Format  cracked_password
----            ------  -------        ------  -------     -----  ------------  ----------  ----------------
192.168.178.64          445/tcp (smb)  tata    tatamaster         Password
```

Exemple d'utilisation avec le service associé indiquant comment ces identifiants ont été utilisés

Remarque

Outre Nmap, plusieurs solutions proposent des sorties au format XML qui peuvent être exploitées de la même manière. Dans le cas contraire, il reste possible de saisir manuellement les données dans la base, comme pour les identifiants de connexion.

3. La prise de note

Malgré cela, il est primordial de prendre des notes dans l'éditeur de son choix et de structurer les éléments afin de ne pas se perdre sur un périmètre étendu. Il n'existe pas de règle stricte, en la matière : l'essentiel est d'adopter une organisation efficace, qui permette de retrouver en temps voulu toutes les informations utiles sans accumuler de données superflues.

Il s'agit d'être en mesure de garder des traces des outils ou exploits (partie de code permettant de tirer profit d'une vulnérabilité (hors MSF)) utilisés afin d'être capables de les communiquer dans son rapport ou le cas échéant aux équipes IT qui voudraient tester leurs compétences en analyse forensique par exemple.

Il faut aussi garder des traces de son cheminement d'attaque, car cela se fait parfois sur plusieurs jours, avec plusieurs pistes différentes. Sans prise de notes, il sera alors plus difficile de l'expliquer dans le rapport final ou pendant une réunion de débriefing.

Le choix de l'éditeur importe peu, pourvu qu'il soit confortable à utiliser, qu'il s'agisse de SublimeText, CherryTree, Scite ou d'un autre outil. En complément du workspace MSF, il permet de centraliser les évidences, phase par phase, à condition d'être alimenté tout au long de l'exercice.

© Editions ENI - All rights reserved

4. Les captures d'écran et autres preuves

Ce point peut sembler secondaire, mais il fait l'objet en réalité de nombreuses discussions…

Le débat porte en général sur le fait d'afficher ou non les données sensibles d'un rapport de test d'intrusion, car ces derniers, lorsqu'ils sont bien réalisés, sont de véritables manuels de compromission pour l'entité ciblée.

Considérant que les rapports sont eux-mêmes des données sensibles classées confidentielles et à diffusion restreinte, l'ensemble des données peut être affiché sans problème à partir du moment où le rapport est remis de manière sécurisée (chiffrée), avec une classification claire, ainsi qu'une liste de diffusion établie.

La responsabilité du non-respect de la classification et de la diffusion n'incombant pas au consultant.

Par ailleurs, une capture d'écran ne doit pas se substituer aux explications nécessaires : elle sert à illustrer le propos et à rendre la preuve immédiatement compréhensible.

Par exemple, une capture d'écran montrant les résultats d'un scan réseau qui révèle l'exposition d'un service critique doit faire apparaître clairement l'information concernée, afin qu'elle soit identifiable au premier coup d'œil, éventuellement à l'aide d'un encadré.

L'objectif est de fournir une preuve indéniable des constats formulés. En effet, une configuration temporaire – lors d'un passage en production ou à la suite d'une mise à jour, par exemple – peut avoir évolué entre-temps. Il est donc essentiel de documenter précisément la situation observée et de justifier le cheminement du pentester au moment de la mission.

Remarque

Il ne faut jamais oublier qu'un test d'intrusion est un état des lieux à un instant t, sur une fenêtre donnée définie en périmètre. Il est donc important de présenter ses conclusions avec le plus de clarté possible en rognant ses captures d'écran ou en n'hésitant pas à surligner et mettre en valeur ce qui est important.

D'autres preuves peuvent être communiquées telles que les codes d'exploitation (les exploits) ou tout autre code, outils, sortie de scan, schéma ou document confidentiel utilisé.

Elles peuvent faire l'objet d'une section addendum en fin de rapport pour plus de clarté, ce qui permet aux équipes de pouvoir rejouer les scénarios mis en avant ou tout simplement d'anticiper les demandes à venir comme « avez-vous testé ceci ou cela ? ».

© Editions ENI - All rights reserved

Chapitre 4
Phases de reconnaissance

1. Introduction

Nous retiendrons ici l'approche la plus courante et la plus généraliste : celle d'un test d'intrusion classique mené depuis Internet, donc depuis l'extérieur.

Lors d'un exercice offensif de type Red Team ou encore TLPT, les phases de reconnaissance s'exécuteront sur plusieurs jours, en favorisant la discrétion.

À l'inverse, un test d'intrusion (ou *Pentest*) est circonscrit dans le temps et, le plus souvent, annoncé aux équipes. Il privilégie donc l'efficacité afin de couvrir le périmètre le plus large possible, quitte à générer davantage de bruit – toutes proportions gardées, évidemment.

Dans le contexte de cet ouvrage, nous partirons donc sur un mode assez classique d'investigation afin d'apporter de la valeur à la mission en identifiant la surface d'attaque externe réelle identifiable depuis Internet.

Durant la phase de reconnaissance, deux approches sont couramment employées : l'approche passive via l'OSINT et l'approche active en commençant à interagir avec l'infrastructure cible. Chacune possède ses avantages et ses inconvénients.