

Rapport Final du Projet Pédagogique

Mise en place d'un SOC d'excellence

Environnement pour la recherche et la collecte des informations.

Etudiant:

Godwine Papin Hougavou

[https:](https://github.com/godwineHougavou/ProjetPro_Supdevinci_M1CBS)

[//github.com/godwineHougavou/ProjetPro_Supdevinci_M1CBS](https://github.com/godwineHougavou/ProjetPro_Supdevinci_M1CBS)



CFA Sup De Vinci
Master 1 Cybersécurité

April 27, 2021

Table of Contents

1	Spécification du projet	2
2	Mise en place de l'environnement de test	3
3	Développement des scripts d'automatisation des actions des analystes SOC	4
3.1	Prérequis	4
3.2	Automatisation des actions RED TEAM	5
3.3	Automatisation des actions BLUE TEAM	10
3.4	Automatisation des actions PURPLE TEAM	13
4	Conclusion	18

Chapter 1: Spécification du projet

L'objectif de ce projet se définit dans la recherche et la collecte des informations spécifiques. Il s'agit de la mise en place d'un SOC d'excellence disposant d'un environnement d'automatisation des actions quotidiennes des différentes teams du SOC: la red team, la blue team et la purple team.

- La **Red Team** de notre SOC a pour action principale, le scan automatisé de réseaux avec analyse de vulnérabilité (il s'agit de récupérer et de pouvoir analyser les vulnérabilités dans nos systèmes d'informations).
- La **Blue Team** de notre SOC a pour action principale, l'analyse d'un fichier, d'une signature de fichier, d'une adresse email, d'un mot de passe, d'une URL.
- La **Purple Team** de notre SOC a pour action principale, la recherche OSINT sur un domaine stratégique afin d'identifier des menaces, la collecte des IOC (Indicateur de Compromission), la collecte du modèle MITRE ATTCK d'une menace (Tactique, Technique, Mitigation).

Chapter 2: Mise en place de l'environnement de test

L'environnement de test à mettre en place est constitué de trois (03) machines dont deux machines virtuelles (windows et ubuntu). La troisième machine étant une machine physique avec le système d'exploitation Kali Linux, est la machine principale. Toutes les actions seront menées depuis cette machine. Les deux autres machines virtuelles seront des machines cibles sur lesquelles nous ferons des tests.

- La première machine virtuelle tourne sur un système d'exploitation windows 7 et la deuxième machine virtuelle tourne sur un système d'exploitation ubuntu 16.04. Sur la machine Ubuntu, nous avons installé une machine vulnérable meterpreter afin d'effectuer des tests de vulnérabilités.
- La machine physique est celle sur laquelle les scripts seront développés et exécutés. Durant le développement des scripts, plusieurs paquets seront installés sur cette machine.
- L'ensemble des machines virtuelles sont mises en pont (bridge) avec la machine physique.

Chapter 3: Développement des scripts d'automatisation des actions des analystes SOC

3.1 Prérequis

Afin de bien déployer l'ensemble des scripts d'automatisation des actions de chacune des teams, nous devons installer plusieurs packages.

- Python3 : <https://www.python.org/downloads/>
- Pip3 : **apt-get -y install python3-pip**
- Python3-nmap convertit les commandes Nmap en méthodes python3, ce qui facilite l'utilisation de nmap dans nos scripts python : **pip3 install python3-nmap** (<https://packaging.python.org/tutorials/installing-packages/>)
- Requests est une librairie HTTP python qui permet d'effectuer des requetes HTTP : **pip3 install requests**
- Pyhibp est une interface Python de "Have I Been Pwned?" (HIBP) avec une API publique qui permet d'interagir avec l'outil HIBP: **pip3 install pyhibp**
- Ipaddress est une librairie python qui fournit les capacités pour créer, manipuler et opérer sur des adresses et des réseaux IPv4 et IPv6: **pip3 install ipaddress**
- Scapy est un puissant programme et une bibliothèque de manipulation de paquets interactifs basés sur Python: **pip3 install scapy**
- Pyattck est un framework léger pour MITRE ATT CK Frameworks. Ce package extrait les détails des cadres MITRE Enterprise, PRE-ATTCK et Mobile: **pip3 install pyattck**
- IOC_FINDER est un package Python pour rechercher et analyser des indicateurs de compromission à partir de texte: **pip3 install ioc-finder**
- Googlesearch est une bibliothèque Python pour rechercher facilement sur Google. google-search utilise des requêtes et BeautifulSoup4 pour scraper Google: **pip3 install googlesearch-python**

Le projet est disponible sur github sous licence GNU General Public License v3.0 à l'adresse suivante https://github.com/godwineHoungavou/ProjetPro_Supdevinci_M1CBS. L'ensemble des package essentiel au bon fonctionnement des scripts est disponible dans le fichier requirements.txt. Après l'installation des différents packages et librairies, il faudra les importer dans les scripts.

```
1 #!/usr/bin/python3
2 # -*- coding: utf-8 -*-
3 import os
4 from googlesearch import search
5 from ioc_finder import find_iocs
6 import json
7 from pyattck import Attck
8 import requests
9 import time
10 import pyhibp
11 from pyhibp import pwnedpasswords as pw
12 import pprint
13 import nmap
14 import nmap3
15 import socket
16 import ipaddress
17 import re
18 import scapy.all as scapy
19 import pprint
20
```

Figure 3.1: Les librairies à importer

3.2 Automatisation des actions RED TEAM

La Red Team de notre SOC a pour action principale, le scan automatisé de réseaux avec analyse de vulnérabilité (il s'agit de récupérer et de pouvoir analyser les vulnérabilités dans nos systèmes d'informations).

Le script que nous avons développé pour automatiser les actions de la Red Team permet d'effectuer les actions suivantes:

[illegible]

Figure 3.2: Script Red Team

- **Network Scanning (Découverte d'IP sur un réseau):** cette action permet à l'analyste SOC d'effectuer la découverte des adresses IP sur un réseau. Il fournit l'adresse réseau et le script retourne l'adresse IP des machines présentes sur le réseau. Pour ce faire, l'utilisateur saisit l'adresse IP du réseau (ex: 192.168.0.1/24); on vérifie si l'adresse saisie correspond à la nomenclature d'une adresse IP. Lorsqu'une correspondance est faite, on crée un paquet ARP qui broadcasté sur le réseau et les paquets reçus en réponses sont analysés. Suite à l'analyse des paquets reçus, on arrivons à extraire l'adresse IP et MAC des machines présentes sur le réseau.

```
1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
Vous avez sélectionné: 1
Entrez l'adresse IP du réseau à scanner au format (Ex: 192.168.0.1/24): 192.168.780.265/43
Adresse IP invalide. Réessayez!
Entrez l'adresse IP du réseau à scanner au format (Ex: 192.168.0.1/24):
```

Figure 3.3: Vérification d'une adresse IP saisie

```
*****
* Copyright of Godwine Papin HOUNGAVOU, 2021 *_Blacksnow*_ *
* Sup De Vinci, M1 Cybersécurité - Projet Professionnel *
* Ce projet est disponible en licence GNU General Public License v3.0 *
* https://github.com/godwineHoungavou/ProjetPro_Supdevinci_M1CBS *
*****

Bienvenue dans cet outil d'automatisation Red Team d'analyse SOC Red Team!

Veuillez choisir une action à mener.

1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
1
Vous avez sélectionné: 1

Entrez l'adresse IP du réseau à scanner au format (Ex: 192.168.0.1/24): 192.168.0.1/26
Les hôtes disponibles sur le réseau sont:
IP MAC
192.168.0.1 40:65:a3:e3:cc:9f
192.168.0.5 40:65:a3:e3:cc:a0
root@kali:~/Bureau/ProjetM1CBS# python3 readteam_SOC.py
```

Figure 3.4: Network Scanning

- **Port Scanning sur une IP:** cette action permet à l'utilisateur de scanner les ports d'une machine en renseignant son adresse IP. Sachant que nous avons des ports allant de 1 à 65535, nous demandons à l'utilisateur de saisir son range de port à scanner (ex: 20-40). Nous utilisons la librairie nmap de python3 pour scanner les ports qui peuvent être dans plusieurs états (filtré, fermé ou ouvert).

Veuillez choisir une action à mener.

- 1) Network Scanning (Découverte d'IP sur un réseau)
- 2) Port Scanning sur une IP
- 3) Host Scanning
- 4) Banner Grabbing sur un port
- 5) Analyse de vulnérabilités
- 6) DNS discovering (DNS Brute Force)

2

Vous avez sélectionné: 2

Entrez l'adresse IP à scanner: 137.74.187.103

Entrez la plage de port à scanner dans le format suivant: Entier-Entier (Ex: 20-80). Pour scanner un port, entrez n

Saisissez la plage de ports: 20-40

Le port 20 est filtered

Le port 21 est filtered

Le port 22 est closed

Le port 23 est filtered

Le port 24 est filtered

Le port 25 est filtered

Le port 26 est filtered

Le port 27 est filtered

Le port 28 est filtered

Le port 29 est filtered

Le port 30 est filtered

Le port 31 est filtered

Le port 32 est filtered

Le port 33 est filtered

Le port 34 est filtered

Le port 35 est filtered

Le port 36 est filtered

Le port 37 est filtered

Le port 38 est filtered

Le port 39 est filtered

Le port 40 est filtered

root@kali:~/Bureau/ProjetM1CBS#

Figure 3.5: Port Scanning

- **Host Scanning:** le service host scanning permet de scanner une machine en retournant l'ensemble des services actifs et le système d'exploitation qui tourne sur la machine.

```

Veuillez choisir une action à mener.
1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
3
Vous avez sélectionné: 3

Entrez l'adresse IP à scanner: 192.168.0.13

Analyse des services et du système d'exploitation en cours...Patientez!!!

Scanning 192.168.0.13 [1 port]
Completed ARP Ping Scan at 14:12, 0.17s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:12
Completed Parallel DNS resolution of 1 host. at 14:12, 0.01s elapsed
Initiating SYN Stealth Scan at 14:12
Scanning 192.168.0.13 [1000 ports]
Discovered open port 443/tcp on 192.168.0.13
Discovered open port 902/tcp on 192.168.0.13
Completed SYN Stealth Scan at 14:12, 0.30s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.0.13
Nmap scan report for 192.168.0.13
Host is up (0.0039s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
443/tcp   open  https
902/tcp   open  iss-realsecure
MAC Address: DC:53:60:4A:EB:F1 (Intel Corporate)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Uptime guess: 6.919 days (since Tue Apr 20 16:08:47 2021)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros

```

Figure 3.6: Host Scanning

- **Banner Grabbing sur un port:** le service banner grabbing permet de récupérer le nom et la version du service qui tourne sur un port ainsi que les informations d'en-tête de ce service. Ce scan peut être effectué sur une plage de port ou sur un port spécifique.

```

Veuillez choisir une action à mener.
1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
4
Vous avez sélectionné: 4

Entrez l'adresse IP à grabber: 192.168.0.13

Entrez la plage de port à grabber dans le format suivant: Entier-Entier (Ex: 20-80). Pour grabber un port, entrez n°Port-n°Port (Ex: 25-25)
Saisissez la plage de ports: 902-902

Le port 902 tourne le service vmware-auth ,version: VMware Authentication Daemon 1.10

Informations supplémentaires grabbées sur le port 902:
220 VMware Authentication Daemon Version 1.10: SSL Required, ServerDaemonProtocol:SOAP , VMXARGS supported, NFCSSL
root@kali:~/Bureau/ProjetM1CBS#

```

Figure 3.7: Banner grabbing sur le port 902

- **Analyse de vulnérabilités:** en utilisant le Nmap Scripting Engine -NSE, nous faisons l'analyse des vulnérabilités en nous basant sur les script vulners et http-vulners-regex.

```

Veillez choisir une action à mener.
1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
5
Vous avez sélectionné: 5

Entrez l'adresse IP à analyser: 137.74.187.103

Analyse des vulnérabilités en cours sur l'hôte 137.74.187.103 .....

Host is up (0.030s latency).
rDNS record for 212.95.74.75: nc-ass-vip.sdv.fr
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd
|_http-server-header: Apache
443/tcp   open  ssl/ssl Apache httpd (SSL-only mode)
|_http-server-header: Apache

Nmap scan report for hackthissite.org (137.74.187.103)
Host is up (0.027s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    closed ssh
80/tcp    open  http-proxy HAProxy http proxy 1.3.1 or later
443/tcp   open  ssl/http-proxy HAProxy http proxy 1.3.1 or later
|_http-server-header: HackThisSite
Service Info: Device: load balancer

```

Figure 3.8: Analyse de vulnérabilité

- **DNS discovering (DNS Brute Force):** ce module nous permet de découvrir les sous-domaines appartenant au domaine saisi par l'utilisateur.

```

Veillez choisir une action à mener.
1) Network Scanning (Découverte d'IP sur un réseau)
2) Port Scanning sur une IP
3) Host Scanning
4) Banner Grabbing sur un port
5) Analyse de vulnérabilités
6) DNS discovering (DNS Brute Force)
6
Vous avez sélectionné: 6

Entrez l'adresse du host à analyser (Ex: exemple.com): supdevinci.fr
Adresse IPv4/IPv6                               Nom de domaine
-----
51.75.130.27                                     test.supdevinci.fr
51.75.130.27                                     www.supdevinci.fr
52.112.196.46                                    sip.supdevinci.fr
2603:1027::48:0:0:20                             sip.supdevinci.fr
root@kali:~/Bureau/ProjetM1CBS#

```

Figure 3.9: DNS discovering


```

1) Analyser une URL
2) Analyser un fichier
3) Analyser la signature d'un fichier
4) Vérifier si votre email est en violation de données
5) Vérifier si votre mot de passe est en violation de données
1
Vous avez sélectionné: 1

Saississez l'URL à scanner au format (exemple.com): hackthissite.org

L'URL: hackthissite.org est malveillante et a été détectée par 1 flag

root@kali:~/Bureau/ProjetM1CBS# █

```

Figure 3.12: Analyse d'URL malveillante

- **Analyser la signature d'un fichier et analyser un fichier:** ces deux actions se rapprochent dans leur fonctionnement. Pour l'analyse de la signature du fichier (Hash), nous nous basons sur l'API VirusTotal qui nous permet de savoir si un hash est malveillant ou pas. En ce qui concerne l'analyse du fichier, le processus se rapproche du précédent. En effet, lorsque le chemin absolu du fichier est saisi puis le fichier uploadé, le hash de ce dernier est calculé. C'est cet hash calculé qui est ensuite vérifié par le biais de l'API VirusTotal.

```

1) Analyser une URL
2) Analyser un fichier
3) Analyser la signature d'un fichier
4) Vérifier si votre email est en violation de données [V] Videos [S] Shopping [I] Plus Paramètres Outils
5) Vérifier si votre mot de passe est en violation de données
3
Vous avez sélectionné: 3
Envoi 315 000 résultats (0.34 secondes)

Saississez le hash de votre fichier (Ex: d41d8cd98f00b204e9800998ecf8427e): d41d8cd98f00b204e9800998ecf8427e

Le hash du fichier n'est pas malveillant

root@kali:~/Bureau/ProjetM1CBS# python3 blueteam_SOC.py

```

Figure 3.13: Analyse de la signature d'un fichier

```

1) Analyser une URL
2) Analyser un fichier
3) Analyser la signature d'un fichier
4) Vérifier si votre email est en violation de données
5) Vérifier si votre mot de passe est en violation de données
2
Vous avez sélectionné: 2

Saississez le chemin absolu du fichier à analyser (Ex: /root/Bureau/test/script.sh): /root/Bureau/ProjetM1CBS/text.JSON

Le fichier n'est pas malveillant

root@kali:~/Bureau/ProjetM1CBS# █

```

Figure 3.14: Analyse d'un fichier

- **Vérifier si votre email est en violation de données:** la première version cette fonctionnalité avait été développée avec l'API v2 de Have I Been Pwned (HIBP). Mais malheureusement, la date de ce jour cette version 2 de l'API est obsolète et donc inopérante. Il est donc conseillé de passer à la version 3 (v3). Nous n'avons pas pu déployer la v3 parce qu'elle était payante. Pour une utilisation professionnelle, nous pourrions payer la v3 de l'API HIBP.

- **Vérifier si votre mot de passe est en violation de données:** la vérification de l'efficacité du mot de passe se base sur l'outil web Have I Been Pwned (HIBP). Cet outil nous fournit une API permettant de manipuler certaines méthodes dont la vérification de l'appartenance d'un mot de passe à une fuite de données.

```

Veillez choisir une action à mener.
1) Analyser une URL
2) Analyser un fichier
3) Analyser la signature d'un fichier
4) Vérifier si votre email est en violation de données
5) Vérifier si votre mot de passe est en violation de données
Vous avez sélectionné: 5
Saississez le mot de passe à vérifier: azerty1234
Votre mot de passe a déjà fuité!
Ce mot de passe a été utilisé 4947 fois déjà.
root@kali:~/Bureau/ProjetM1CBS# python3 blueteam_SOC.py

```

Figure 3.15: Analyse d'un mot de passe ayant fuité

```

Veillez choisir une action à mener.
1) Analyser une URL
2) Analyser un fichier
3) Analyser la signature d'un fichier
4) Vérifier si votre email est en violation de données
5) Vérifier si votre mot de passe est en violation de données
Vous avez sélectionné: 5
Saississez le mot de passe à vérifier: #!H1g@45f_We4cK@5S
Votre mot n'a pas été violée ou été dans une fuite de données
root@kali:~/Bureau/ProjetM1CBS#

```

Figure 3.16: Analyse d'un mot de passe n'ayant pas fuité

3.4 Automatisation des actions PURPLE TEAM

La Purple Team de notre SOC a pour action principale, la recherche OSINT sur un domaine stratégique afin d'identifier des menaces, la collecte des IOC (Indicateur de Compromission), la collecte du modèle MITRE ATTCK d'une menace (Tactique, Technique, Mitigation)

Le script que nous avons développé pour automatiser les actions de la Red Team permet d'effectuer les actions suivantes

[illegible]

Figure 3.17: Script Purple Team

- **Recherche OSINT sur un domaine stratégique (GOOGLE DORKS):** cette fonctionnalité nous permet de faire plusieurs recherches spécifiques sur un domaine grâce aux googles dorks: il s'agit entre autre de la recherche de mot de passe utilisateur sur un domaine, la recherche des informations financière sur un domaine, l'utilisation d'une chaîne de recherche spéciale pour trouver des sites Web vulnérables ou des vulnérabilités sur un site donné et trouver des redirections ouvertes sur un domaine.

```

Bienvenue dans cet outil d'automatisation des actions PURPLE TEAM!

Veuillez choisir une action à mener.
1) Recherche OSINT sur un domaine stratégique (GOOGLE DORKS)
2) WHOIS grabbing
3) Recherche d'email sur une cible
4) Recherche de fichier sur une cible
5) Recherche de nom de domaines et sous-domaines sur une cible
6) Collecte des IOC d'une menace
7) Collecte du modèle MITRE ATTCK d'une menace

1
Vous avez sélectionné: 1
1) Recherche de mot de passe utilisateur sur un domaine
2) Recherche des informations financière sur un domaine
3) Utilisation d'une chaîne de recherche spéciale pour trouver des sites Web vulnérables ou des vulnérabilités sur un site
4) Trouver des redirections ouvertes

```

Figure 3.18: Recherche OSINT via Google Dork

- **Recherche d'email sur une cible:** la recherche des emails sur un domaine se fait grâce à l'outil theHarvester. Ce outil recherche les emails de différentes sources publiques comme les moteurs de recherche tels que Shodan, les serveurs de clés PGP, Google...
- **Recherche de fichier sur une cible:** la recherche des fichiers sur un domaine passe également par l'utilisation des googles dorks. On recherche des fichiers importants tels que: des pdf, doc, xls, txt, odt, ppt... Nous n'avons pas pu mettre une capture du programme parce qu'on a eu un message d'erreur concernant le nombre élevé de requête que nous avons déjà effectué lors de nos tests.
- **Recherche de nom de domaines et sous-domaines sur une cible:** la recherche des sous-domaines sur un domaine se fait avec l'outil sublist3r. Cet outil questionne les serveurs DNS afin de retrouver l'ensemble des sous-domaines appartenant au domaine cible.

```

Veuillez choisir une action à mener.
1) Recherche OSINT sur un domaine stratégique (GOOGLE DORKS)
2) WHOIS grabbing
3) Recherche d'email sur une cible
4) Recherche de fichier sur une cible
5) Recherche de nom de domaines et sous-domaines sur une cible
6) Collecte des IOC d'une menace
7) Collecte du modèle MITRE ATTCK d'une menace

5
Vous avez sélectionné: 5

Entrez le nom de domaine cible (Ex: exemple.com): supdevinci.fr

Recherche en cours... Veuillez patienter... !

_____Les domaines et sous-domaines découverts_____

www.supdevinci.fr
alumni.supdevinci.fr
candidature.supdevinci.fr
mailing.supdevinci.fr
r.mailing.supdevinci.fr
marisus.supdevinci.fr
test.supdevinci.fr

root@kali:~/Bureau/ProjetM1CBS#

```

Figure 3.19: Recherche des sous-domaines

- **Collecte des IOC d'une menace:** pour la collecte des IOC d'une menace, nous utilisons le package IOC_FINDER. Ce package nous permet de rechercher et d'analyser des indicateurs de compromission à partir d'un texte ou d'un fichier de log suite à une attaque.

```

Veillez choisir une action à mener.
1) Recherche OSINT sur un domaine stratégique (GOOGLE DORKS)
2) WHOIS grabbing
3) Recherche d'email sur une cible
4) Recherche de fichier sur une cible
5) Recherche de nom de domaines et sous-domaines sur une cible
6) Collecte des IOC d'une menace
7) Collecte du modèle MITRE ATTCK d'une menace

6
Vous avez sélectionné: 6

Saisissez ou collez ci-après le texte ou contenus du fichier log à partir duquel les indicateurs de compromission seront analysés.
This is just an foobar.com https://example.org/test/bingo.php. The website is accessible via this IP 130.67.34.12

Recherche des IOC en cours... Patientez!!!

Les indicateurs de compromission (IOC) collectés sont:
IOC (Indicator Of Compromise)
Valeurs

urls ['https://example.org/test/bingo.php.']
xmpp_addresses []
email_addresses_complete []
email_addresses []
ipv4_cidrs []
imphashes []
authenticashes []
domains ['example.org', 'foobar.com']
ipv4s ['130.67.34.12']
ipv6s []
sha512s []
sha256s []
sha1s []
md5s []
ssdeeps []
asns []
cves []
registry_key_paths []
google_adsense_publisher_ids []
google_analytics_tracker_ids []
bitcoin_addresses []
monero_addresses []
mac_addresses []
user_agents []
phone_numbers []
tlp_labels []
attack_mitigations {'enterprise': [], 'mobile': []}
attack_tactics {'pre_attack': [], 'enterprise': [], 'mobile': []}
attack_techniques {'pre_attack': [], 'enterprise': [], 'mobile': []}
file_paths []
root@kali:~/Bureau/ProjetM1CBS#

```

Figure 3.20: Collecte des IOC

- **Collecte du modèle MITRE ATTCK d'une menace:** la collecte du modèle MITRE ATTCK se fait grâce au framework Pyattck. Ce package extrait les détails des cadres Mitre Entreprise, PRE-ATTCK et Mobile.

```

Veillez choisir une action à mener.
1) Recherche OSINT sur un domaine stratégique (GOOGLE DORKS)
2) WHOIS grabbing
3) Recherche d'email sur une cible
4) Recherche de fichier sur une cible
5) Recherche de nom de domaines et sous-domaines sur une cible
6) Collecte des IOC d'une menace
7) Collecte du modèle MITRE ATTCK d'une menace

7
Vous avez sélectionné: 7

Entrez le cadre MITRE ATTCK que vous souhaitez collecter parmi les cadres suivants: ENTREPRISE, PRE-ATTCK et MOBILE: ENTREPRISE

-----
Les tactiques utilisées pour le Mitre Attck ENTREPRISE
-----
| Tactique | Lien |
|-----|-----|
Collection | https://attack.mitre.org/tactics/TA0009 |
Command and Control | https://attack.mitre.org/tactics/TA0011 |
Credential Access | https://attack.mitre.org/tactics/TA0006 |
Defense Evasion | https://attack.mitre.org/tactics/TA0005 |
Discovery | https://attack.mitre.org/tactics/TA0007 |
Execution | https://attack.mitre.org/tactics/TA0002 |
Exfiltration | https://attack.mitre.org/tactics/TA0010 |
Impact | https://attack.mitre.org/tactics/TA0040 |
Initial Access | https://attack.mitre.org/tactics/TA0001 |
Lateral Movement | https://attack.mitre.org/tactics/TA0008 |
Persistence | https://attack.mitre.org/tactics/TA0003 |
Privilege Escalation | https://attack.mitre.org/tactics/TA0004 |
Reconnaissance | https://attack.mitre.org/tactics/TA0043 |
Resource Development | https://attack.mitre.org/tactics/TA0042 |

#####

-----
Les techniques utilisées pour le Mitre Attck ENTREPRISE
-----
| Techniques | Lien |
|-----|-----|
Abuse Elevation Control Mechanism | https://attack.mitre.org/techniques/T1548 |
Access Token Manipulation | https://attack.mitre.org/techniques/T1134 |
Account Access Removal | https://attack.mitre.org/techniques/T1531 |
Account Discovery | https://attack.mitre.org/techniques/T1087 |
Account Manipulation | https://attack.mitre.org/techniques/T1098 |
Acquire Infrastructure | https://attack.mitre.org/techniques/T1583 |
Active Scanning | https://attack.mitre.org/techniques/T1595 |
Application Layer Protocol | https://attack.mitre.org/techniques/T1071 |
Application Window Discovery | https://attack.mitre.org/techniques/T1010 |
Archive Collected Data | https://attack.mitre.org/techniques/T1560 |
Audio Capture | https://attack.mitre.org/techniques/T1123 |
Automated Collection | https://attack.mitre.org/techniques/T1119 |
Automated Exfiltration | https://attack.mitre.org/techniques/T1020 |
BITS Jobs | https://attack.mitre.org/techniques/T1197 |
Boot or Logon Autostart Execution | https://attack.mitre.org/techniques/T1547 |

```

Figure 3.21: Collecte du modèle MITRE ATTCK ENTREPRISE 1/2

Valid Accounts	https://attack.mitre.org/techniques/T1078
Video Capture	https://attack.mitre.org/techniques/T1125
Virtualization/Sandbox Evasion	https://attack.mitre.org/techniques/T1497
Weaken Encryption	https://attack.mitre.org/techniques/T1600
Web Service	https://attack.mitre.org/techniques/T1102
Windows Management Instrumentation	https://attack.mitre.org/techniques/T1047
XSL Script Processing	https://attack.mitre.org/techniques/T1220
Domain Policy Modification	https://attack.mitre.org/techniques/T1484
Forge Web Credentials	https://attack.mitre.org/techniques/T1606

#####

Les mitigations utilisées pour le Mitre Attck ENTREPRISE

Mitigation	Lien
.bash_profile and .bashrc Mitigation	https://attack.mitre.org/mitigations/T1156
Access Token Manipulation Mitigation	https://attack.mitre.org/mitigations/T1134
Accessibility Features Mitigation	https://attack.mitre.org/mitigations/T1015
Account Discovery Mitigation	https://attack.mitre.org/mitigations/T1087
Account Use Policies	https://attack.mitre.org/mitigations/M1036
Active Directory Configuration	https://attack.mitre.org/mitigations/M1015
Antivirus/Antimalware	https://attack.mitre.org/mitigations/M1049
AppCert DLLs Mitigation	https://attack.mitre.org/mitigations/T1182
AppInit DLLs Mitigation	https://attack.mitre.org/mitigations/T1103

Figure 3.22: Collecte du modèle MITRE ATTCK ENTREPRISE 2/2

Chapter 4: Conclusion

La réalisation de ce projet portant sur la mise en place d'un environnement d'automatisation des actions d'un analyste SOC (Red Team, Blue Team et Purple Team), nous a permis d'aborder plusieurs notions en matière d'OSINT, d'analyse forensique, de pentest et de programmation. Entre autres, à travers ce projet, nous avons essentiellement appris la gestion des entrées utilisateur d'un programme, la gestion des erreurs et la manipulation des API avec le langage Python. Ce projet nous a permis d'automatiser plusieurs tâches qui s'effectuent au sein d'un SOC.