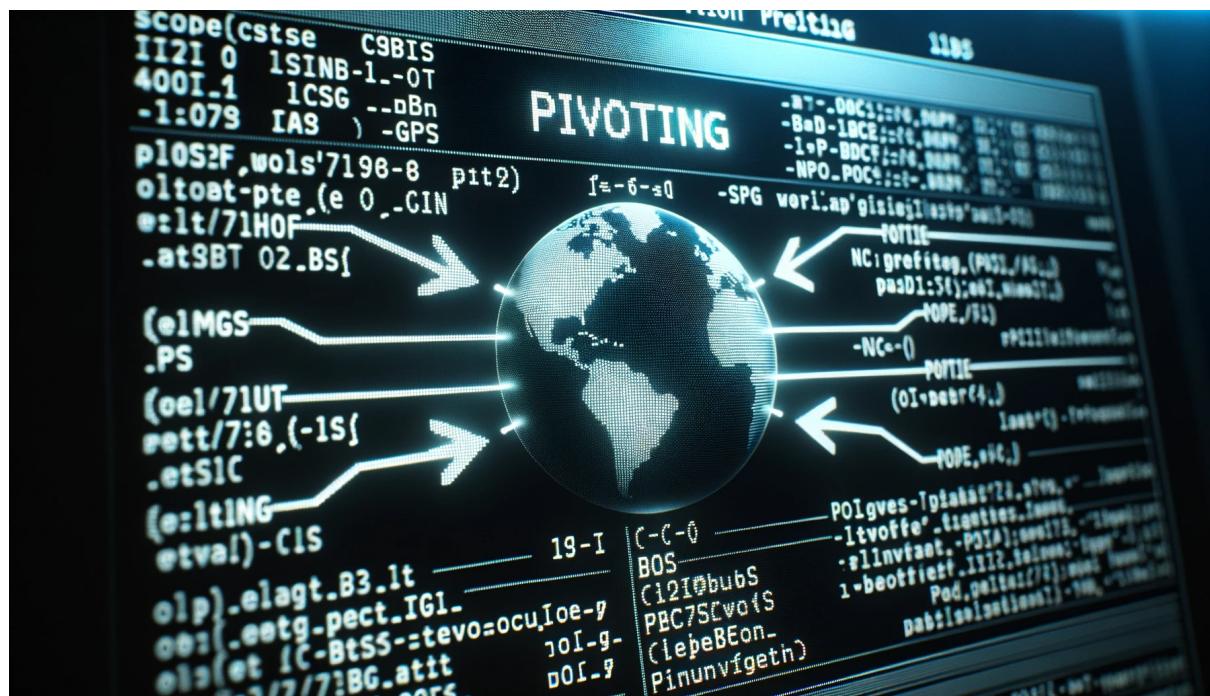


Thomas Fernandez
Antoine Garros
Redwane ZAGHOUINI



Projet CEH - Pivoting CTF



Création de la VM1

Nous allons utiliser une vm ubuntu pour cette VM

voici son réseau qu'on va modifier par la suite

```
→ ~ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:af:94:03 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 172.16.32.142/24 metric 100 brd 172.16.32.255 scope global dynamic ens1
60
        valid_lft 1776sec preferred_lft 1776sec
    inet6 fe80::20c:29ff:feaf:9403/64 scope link
        valid_lft forever preferred_lft forever
→ ~
```

L'adresse IP de la machine est 172.16.32.142

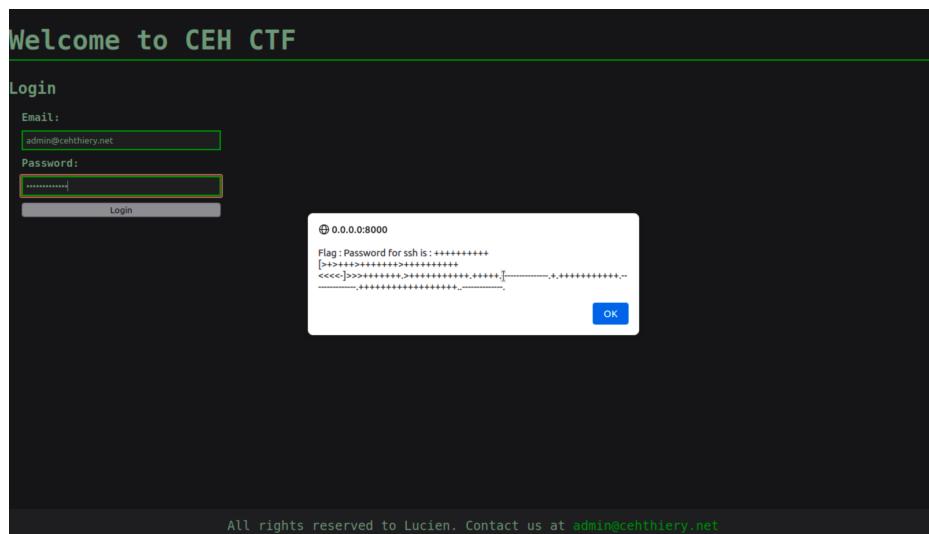
On crée un dossier pour mettre les fichiers de notre site internet que nous allons créer
Ce site aura un fichier index.html qui sera un login d'un site
et une page caché qu'on appellera admin_ceh.html qui sera vu lorsque l'attaquant aura
trouver les logs du form login
on crée un serveur http avec python

```
→ website python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [09/Oct/2023 15:40:58] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Oct/2023 15:40:58] code 404, message File not found
127.0.0.1 - - [09/Oct/2023 15:40:58] "GET /favicon.ico HTTP/1.1" 404 -
```

La première page ressemblera à ça ou il devra essayer de se connecter avec des logs qu'il trouve



Lorsqu'il trouve le login et le mot de passe, cette alerte apparaîtra avec ce message



Sur cette Vm nous vérifions bien que ssh est activé

```
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2023-10-09 17:27:21 UTC; 1h 21min left
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 1004 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1017 (sshd)
   Tasks: 1 (limit: 2185)
  Memory: 7.3M
    CPU: 207ms
   CGroup: /system.slice/ssh.service
           └─1017 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

De plus je crée le sous réseau

Je vais modifier le fichier de configuration netplan comme ici

```
[→ ~ sudo vim /etc/netplan/00-installer-config.yaml
[→ ~ sudo cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'

network:
  version: 2
  ethernets:
    ens160:
      dhcp4: no
      addresses:
        - 172.16.32.142/24
        - 172.16.32.138/24
      gateway4: 172.16.32.2
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
[→ ~ sudo netplan apply
```

Et on voit bien les 2 ips

172.16.32.142/24 et 172.16.32.138/24

Les deux adresses sont dans le sous-réseau 172.16.32.0/24

On vérifie en utilisant ip a

```
[→ ~ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:af:94:03 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 172.16.32.142/24 brd 172.16.32.255 scope global ens160
        valid_lft forever preferred_lft forever
    inet 172.16.32.138/24 brd 172.16.32.255 scope global secondary ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feaf:9403/64 scope link
        valid_lft forever preferred_lft forever
```

Création de la VM2

Nous allons créer une Vm simplement lui attribuer une adresse IP dans la plage 172.16.32.0/24, par exemple 172.16.32.144

```
[user@vmttest:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:3c:35:b2 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 172.16.32.144/24 metric 100 brd 172.16.32.255 scope global dynamic ens160
        valid_lft 1408sec preferred_lft 1408sec
    inet6 fe80::20c:29ff:fe3c:35b2/64 scope link
        valid_lft forever preferred_lft forever
```

On peut vérifier pour voir que les pings entre les 2 vm marchent bien
vm1 qui ping vm2

```
[→ ~ ping 172.16.32.138
PING 172.16.32.138 (172.16.32.138) 56(84) bytes of data.
64 bytes from 172.16.32.138: icmp_seq=1 ttl=64 time=0.723 ms
64 bytes from 172.16.32.138: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 172.16.32.138: icmp_seq=3 ttl=64 time=0.176 ms
64 bytes from 172.16.32.138: icmp_seq=4 ttl=64 time=0.153 ms
^C
--- 172.16.32.138 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.099/0.287/0.723/0.252 ms
```

cela marche bien
ainsi que vm2 qui ping vm1

```
usertest@vmtest:~$ ping 172.16.32.142
PING 172.16.32.142 (172.16.32.142) 56(84) bytes of data.
64 bytes from 172.16.32.142: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 172.16.32.142: icmp_seq=2 ttl=64 time=0.891 ms
64 bytes from 172.16.32.142: icmp_seq=3 ttl=64 time=0.917 ms
64 bytes from 172.16.32.142: icmp_seq=4 ttl=64 time=0.829 ms
^C
--- 172.16.32.142 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.829/0.922/1.053/0.081 ms
usertest@vmtest:~$ ping 172.16.32.138
PING 172.16.32.138 (172.16.32.138) 56(84) bytes of data.
64 bytes from 172.16.32.138: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 172.16.32.138: icmp_seq=2 ttl=64 time=0.995 ms
64 bytes from 172.16.32.138: icmp_seq=3 ttl=64 time=0.858 ms
64 bytes from 172.16.32.138: icmp_seq=4 ttl=64 time=0.853 ms
^C
--- 172.16.32.138 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.853/0.974/1.193/0.138 ms
```

On va créer le serveur ftp avec un flag à récupérer par la suite

```
usertest@vmtest:~$ sudo apt-get install vsftpd
[sudo] password for usertest:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ssl-cert
The following NEW packages will be installed:
  ssl-cert vsftpd
0 upgraded, 2 newly installed, 0 to remove and 38 not upgraded.
```

on modifie le fichier vsftpd.conf

```
usertest@vmtest:~$ sudo vim /etc/vsftpd.conf
```

avec les paramètres
write_enable=YES
local_enable=YES
chroot_local_user=YES

puis on restart et on voit bien que le serveur tourne

```

usertest@vmtest:~$ sudo service vsftpd restart
usertest@vmtest:~$ sudo service vsftpd status
● vsftpd.service - vsftpd FTP server
  Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
  Active: active (running) since Mon 2023-10-16 13:01:12 UTC; 5s ago
    Process: 1642 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
   Main PID: 1643 (vsftpd)
     Tasks: 1 (limit: 4504)
    Memory: 804.0K
       CPU: 7ms
      CGroup: /system.slice/vsftpd.service
              └─1643 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 16 13:01:12 vmtest systemd[1]: Starting vsftpd.service - vsftpd FTP server...
Oct 16 13:01:12 vmtest systemd[1]: Started vsftpd.service - vsftpd FTP server.

```

ensuite on va créer un flag à récupérer par la suite grâce au serveur ftp

```

usertest@vmtest:/home$ cd usertest
usertest@vmtest:~$ ls
usertest@vmtest:~$ cat flag.txt
cat: flag.txt: No such file or directory
usertest@vmtest:~$ vim flag.txt
usertest@vmtest:~$ ls
flag.txt
usertest@vmtest:~$ cat flag.txt
Bravo, vous avez réussi le pivoting

```

On active aussi ssh sur la VM 2

```

usertest@vmtest:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Drop-In: /etc/systemd/system/ssh.service.d
            └─00-socket.conf
    Active: active (running) since Mon 2023-10-16 12:58:14 UTC; 56min ago
  TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1250 (sshd)
    Tasks: 1 (limit: 4504)
   Memory: 3.2M
      CPU: 86ms
     CGroup: /system.slice/ssh.service
             └─1250 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

```

il est bien activé

Attaque des 2 vms depuis une VM Kali en utilisant le pivoting

Je nmap le réseau que je veux attaquer et je vois les ports ouverts

```
(tulmax㉿kali)-[~] $ nmap 172.16.32.142
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-11 10:10 CEST
Nmap scan report for 172.16.32.142
Host is up (0.00093s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp  open  http-proxy
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Ici je vois que j'ai le ssh d' actif et un site web sur le port 8080

Je regarde le site web http

J'arrive sur cette page

Welcome to CEH CTF

Login

Email:

Password:

All rights reserved to Lucien. Contact us at tulmax@cehthiery.net

On peut inspecter le code source pour voir si cela peut nous aider
on voit que il ya une note nous donnant un hash

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CEH CTF</title>
5   </head>
6   <body>
7     <h1>Welcome to CEH CTF</h1>
8
9     <!-- Note to myself : remove the hash of the password
10    80a19f669b02edfbc208a5386ab5036b
11
12  -->
13
14
15  <h2>Login</h2>
16  <form class="form-container" id="login-form">
17    <label for="login-email">Email:</label>
18    <input type="email" id="login-email" required>
19    <label for="login-password">Password:</label>
20    <input type="password" id="login-password" required>
21    <input type="submit" value="Login">
22  </form>
23
24
25  <footer class="footer">
26    <p>All rights reserved to tulmax. Contact us at <a href="mailto:admin@cehthiery.net">admin@cehthiery.net</a></p>
27  </footer>
28
29  <script>
30    const loginForm = document.getElementById("login-form");
31
32    loginForm.addEventListener("submit", (event) => {
```

80a19f669b02edfbc208a5386ab5036b
on peut voir à sa forme que c'est du md5

on va utiliser john the ripper avec la wordlist de rockyou_new1.txt qui va le trouver normalement si le password est pas très robuste

```
21      <input type="submit" value="Login">
22 (tulmax㉿kali)-[~]
23 $ echo "80a19f669b02edfbc208a5386ab5036b" > myhash.txt
24
25  <footer class="footer">
26
27 (tulmax㉿kali)-[~]
28 $ john --format=Raw-MD5 --wordlist=rockyou_new1.txt myhash.txt
29
30 Using default input encoding: UTF-8
31 (tulmax㉿kali)-[~] $ john --format=Raw-MD5 --wordlist=rockyou_new1.txt myhash.txt
32
33 Loaded 1 password hash (Raw-MD5 [MD5 128/128 ASIMD 4x2])
34 Warning: no OpenMP support for this hash type, consider --fork=4
35 Press 'q' or Ctrl-C to abort, almost any other key for status
36 passwordadmin (?)
37 1g 0:00:00:00 DONE (2023-10-11 10:32) 100.0g/s 1000p/s 1000c/s 1000C/s passwordadmin..12345678
38 Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
39 Session completed.
40
41 (tulmax㉿kali)-[~] $ curl -s http://127.0.0.1:8000/admin_ceh.html
42
43
44 (tulmax㉿kali)-[~] $
```

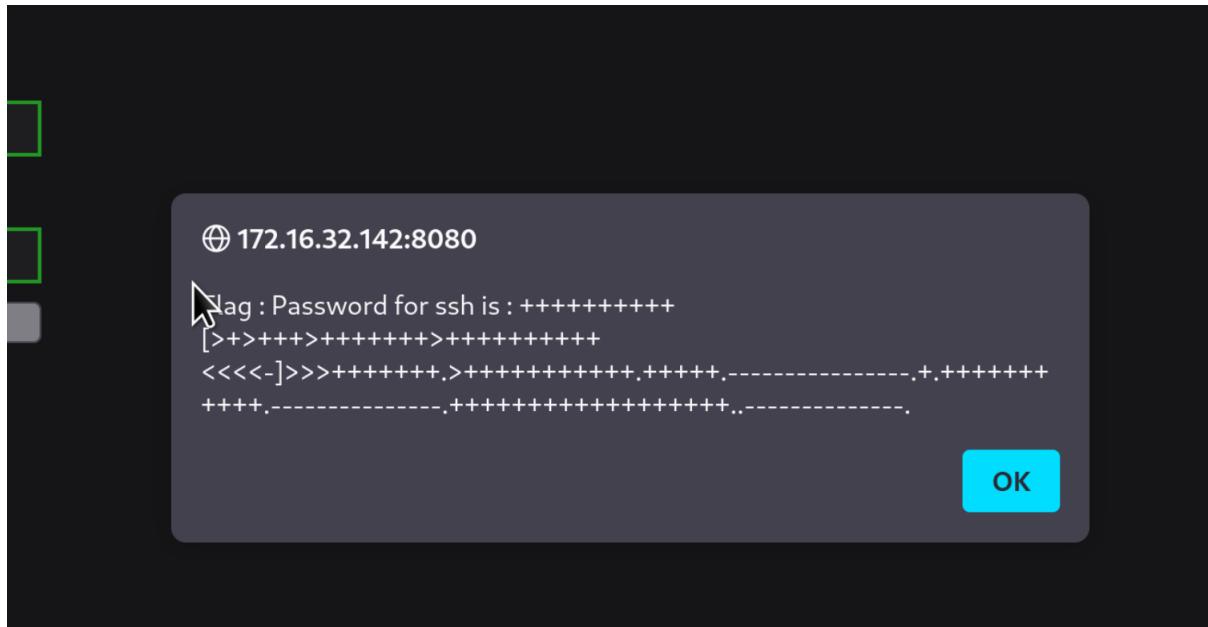
on voit que le password est passwordadmin

le footer est comme ceci

```
All rights reserved to Lucien. Contact us at tulmax@cehthiery.net
```

on va essayer plusieurs mails avec tulmax
comme tulmax@gmail.com, lucien@cehthiery.net, lucien@gmail.com, etc
Ici on peut utiliser du social engineering pour essayer de trouver un mail avec les
informations que nous avons
l'email a trouvé est tulmax@cehthiery.net

tulmax@cehthiery.net marche avec le mdp trouvé avec john the ripper



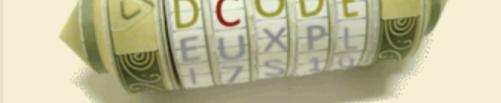
Congrats ! Welcome to the Admin Page !

On peut analyser le flag que nous avons qui est comme ceci :

: Password for ssh is :

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

On peut utiliser des sites comme <https://crackstation.net/> ou d'autres pour voir quel type de hash est utilisé pour celui-ci
mais avec notre expérience de ctf on voit que cela ressemble à du braifuck language
Utilisant ce site très utile pour déchiffrer des mots de passe
on voit que le mot de passe ssh est Motdepasse



Search for a tool

★ SEARCH A TOOL ON D^{CODE} BY KEYWORDS:
e.g. type 'sudoku'

★ BROWSE THE [FULL D^{CODE} TOOLS' LIST](#)

Results

Input: `++++++[>----.`

Arg:

Output:

Motdepasse

BRAINFUCK

Informatics > Programming Language > Brainfuck

BRAINFUCK INTERPRETER

★ BRAINFUCK CODE TO INTERPRET

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

★ ARGUMENT

★ SHOW MEMORY STATE

See also: [Leet Speak 1337](#) – [LOLCODE Language](#) – [ReverseFuck](#) – [Alphuck](#) – [JSFuck Language](#) `[]([![]]+[])` – [Binaryfuck](#)

On peut donc se connecter en ssh

```
(tulmax㉿kali)-[~]
$ ssh tulmax@172.16.32.142
The authenticity of host '172.16.32.142 (172.16.32.142)' can't be established.
ED25519 key fingerprint is SHA256:YxXk7W+v5Z3Y6iMAoOT2tdGa3zFPTnGe1wcd6qK0lI4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.32.142' (ED25519) to the list of known hosts.
tulmax@172.16.32.142's password: ab5036b
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-86-generic aarch64)

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

System information as of Wed Oct 11 08:45:19 AM UTC 2023

System load: 0.0 Processes: 286
Usage of /: 58.1% of 18.60GB Users logged in: 1
Memory usage: 39% IPv4 address for ens160: 172.16.32.142
Swap usage: 0% IPv4 address for ens160: 172.16.32.138

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

if (email === "admin@cehthiery.net" && password === "passwordadmin") {
    // If the user is the admin, show the flag
Last login: Wed Oct 11 08:41:57 2023 from 172.16.32.11 [+>++++>++++++>++++++<<<- ] >>
→ ~ whoami // Redirect to the admin page
tulmax window.location.href = "admin_ceh.html";
```

On peut bien changer de dossiers et on voit le serveur web intéressant

```
~ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos website
→ ~ cd website
→ website ls
admin_ceh.html index.html href = "admin_ceh.html"
→ website
```

on peut maintenant voir sa carte réseau

```
~ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        link/loopback 00:0c:29:af:94:03 brd ff:ff:ff:ff:ff:ff
        altname enp2s0
        inet 172.16.32.142/24 brd 172.16.32.255 scope global ens160
            valid_lft forever preferred_lft forever
            link/loopback fe80::20c:29ff:feaf:9403/64 scope link
                valid_lft forever preferred_lft forever
→ ~
```

on peut voir qu'il a plusieurs adresses ip dans la plage 172.16.32.0/24 avec cette commande nmap

```
(tulmax㉿kali)-[~] $ sudo nmap -sn 172.16.32.0/24
[sudo] password for tulmax:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-16 15:38 CEST
Nmap scan report for 172.16.32.1
Host is up (0.00028s latency).
MAC Address: 1E:57:DC:E3:95:65 (Unknown)
Nmap scan report for 172.16.32.2
Host is up (0.00046s latency).
MAC Address: 00:50:56:F5:49:03 (VMware)
Nmap scan report for 172.16.32.13
Host is up (0.00093s latency).
MAC Address: 00:0C:29:AF:94:03 (VMware)
Nmap scan report for 172.16.32.142
Host is up (0.00089s latency).
MAC Address: 00:0C:29:AF:94:03 (VMware)
Nmap scan report for 172.16.32.144
Host is up (0.0012s latency).
MAC Address: 00:0C:29:3C:35:B2 (VMware)
Nmap scan report for 172.16.32.254
Host is up (0.0020s latency).
MAC Address: 00:50:56:FF:39:AB (VMware)
Nmap scan report for 172.16.32.128
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.14 seconds
```

cette commande nous montre toutes les machines qui sont sur le réseau

On peut maintenant voir si ces machines peuvent nous permettre d'avoir accès à un serveur ftp

nous allons run cette commande pour spécifier le port 21

```
└─(tulmax㉿kali)-[~]
└─$ sudo nmap -p 21 172.16.32.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-16 15:42 CEST
Nmap scan report for 172.16.32.1
Host is up (0.00035s latency).
```

```
POR      STATE SERVICE
21/tcp    closed  ftp
MAC Address: 00:0C:29:AF:94:03 (VMware)
```

```
Nmap scan report for 172.16.32.144
Host is up (0.0015s latency).
```

et on voit que l'ip où il y a un serveur ftp est 172.16.32.144

Avec l'aide de la documentation d'orange sur le pivoting

SSH local port forwarding

Les connexions depuis le client SSH sont transférées via le serveur SSH puis vers une machine de destination.

```
$ ssh user@ssh_server -L [bind_address:]local_port:destination_host:des
```

Par exemple, ici, l'auditeur ouvre le port 32000 en local sur son poste (192.168.2.149), se connecte en SSH sur la machine compromise Srv App 1. Le serveur SSH de la machine compromise transfère toutes les requêtes que l'attaquant fait sur sa machine (127.0.0.1:32000) vers le Srv App 2 (10.42.42.2:80) en passant par Srv App 1 (192.168.2.105) via SSH (22).

```
$ ssh noraj@192.168.2.105 -L 127.0.0.1:32000:10.42.42.2:80 -N
```

Note :

L'option -N sert à ne pas exécuter de commande après l'authentification, et donc dans notre cas à ne pas lancer de shell puisque nous ne nous servons de SSH que comme d'un tunnel.

Une fois ce tunnel monté, quand l'auditeur requête 127.0.0.1, il demande en fait indirectement 10.42.42.2 (pour l'exemple d'un serveur web).

On va utiliser cette commande

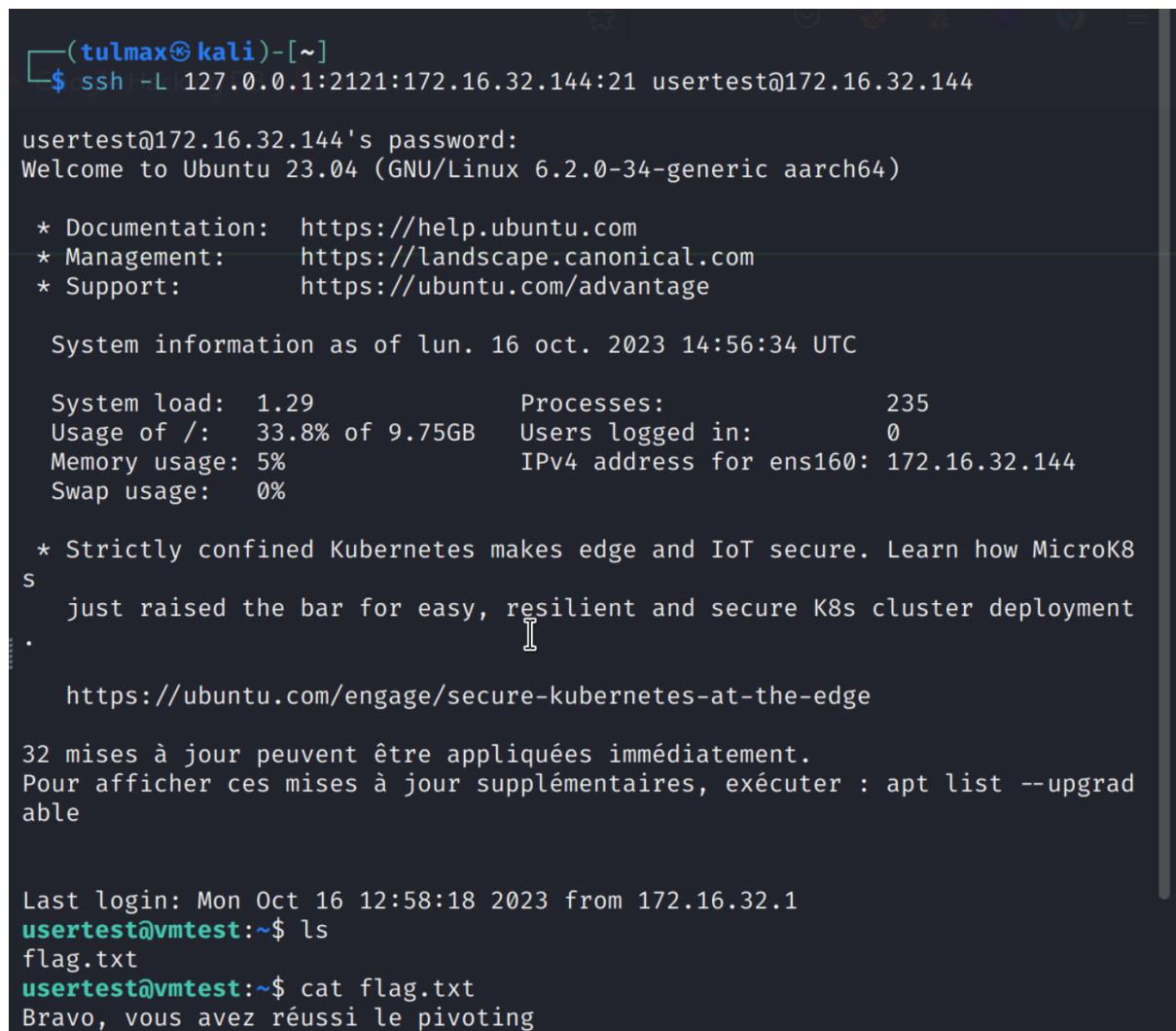
```
ssh -L 127.0.0.1:2121:172.16.32.144:21 usertest@172.16.32.144
```

-L : indique que nous allons configurer une redirection de port local.

127.0.0.1:2121 : représente le point de départ de la redirection de port. Dans ce cas, nous redirigeons le port 2121 sur la machine locale vers la machine distante (VM2).

172.16.32.144:21 : représente le point de destination de la redirection de port. Nous redirigeons le trafic du port 2121 de la VM1 vers le port 21 de la VM2. Le port 21 est le port standard utilisé par les serveurs FTP.

usertest@172.16.32.144 : c'est le nom d'utilisateur et l'adresse IP de la machine distante (VM2) à laquelle on souhaite se connecter en SSH.



```
(tulmax㉿kali)-[~]
$ ssh -L 127.0.0.1:2121:172.16.32.144:21 usertest@172.16.32.144

usertest@172.16.32.144's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-34-generic aarch64)

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

 System information as of lun. 16 oct. 2023 14:56:34 UTC

 System load:  1.29          Processes:           235
 Usage of /:   33.8% of 9.75GB  Users logged in:     0
 Memory usage: 5%            IPv4 address for ens160: 172.16.32.144
 Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8
s
 just raised the bar for easy, resilient and secure K8s cluster deployment
.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

32 mises à jour peuvent être appliquées immédiatement.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgrad
able

Last login: Mon Oct 16 12:58:18 2023 from 172.16.32.1
usertest@vmtest:~$ ls
flag.txt
usertest@vmtest:~$ cat flag.txt
Bravo, vous avez réussi le pivoting
```

Le mot de passe étant le même que la VM1

On peut considérer que dans cette entreprise le mot de passe est commun à toutes les machines du même sous réseau

On peut voir le tunnel de la vm kali avec cette commande

```
(tulmax㉿kali)-[~]
$ ps aux | grep ssh

tulmax      1348  0.0  0.1  7508  3180 ?          Ss   15:30   0:00 /usr/bin/
ssh-agent x-session-manager
tulmax      28369  0.0  0.3 14708  8092 pts/1    S+   17:11   0:00 ssh -L 21
21:127.0.0.1:21 usertest@172.16.32.144
tulmax      30502  0.0  0.0  6220  1964 pts/0    S+   17:19   0:00 grep --co
lor=auto ssh
```

On voit bien que le pivoting a bien fonctionné

Conclusion

Dans ce projet, nous avons configuré deux machines virtuelles (VM1 et VM2) au sein d'un réseau local, la VM1 servant de cible pour notre attaque. Nous avons mis en place un serveur web et un serveur FTP sur la VM1, et avons découvert un flag caché dans le code source de la page web. En utilisant une machine extérieure (VM Kali), nous avons réussi à compromettre la VM1 en utilisant une attaque de force brute pour obtenir les identifiants SSH. Après avoir obtenu l'accès SSH, nous avons utilisé la technique de pivoting pour attaquer la VM2 depuis la VM1.

Analyse du projet sur les raisons de notre accès à la VM2 :

Mots de passe faibles : L'attaque a principalement réussi en raison de l'utilisation de mots de passe faibles sur les comptes utilisateurs. Des mots de passe tels que "passwordadmin" et "Motdepasse" sont particulièrement vulnérables aux attaques de force brute.

Examen du code source du site web : L'examen du code source du site web a révélé des informations sensibles ainsi que des commentaires qui ont grandement aidé l'attaquant. Cela souligne l'importance de ne pas divulguer d'informations sensibles dans le code source d'un site web, car cela peut être exploité par des attaquants. Un code source bien conçu doit être propre, sans commentaires superflus ni informations sensibles visibles.

Partage de mots de passe : Le partage de mots de passe entre plusieurs machines au sein du même sous-réseau a également contribué à la compromission des systèmes. Ceci souligne l'importance d'utiliser des mots de passe forts et uniques pour chaque compte utilisateur.

L'utilisation de mots de passe partagés entre plusieurs machines au sein du même sous-réseau est une faille de sécurité majeure. Cela signifie qu'une fois qu'un attaquant a

compromis un seul compte utilisateur, il peut potentiellement accéder à toutes les machines du même sous-réseau qui utilisent le même mot de passe.

Configuration FTP vulnérable : La configuration FTP sur la VM2 était vulnérable, permettant ainsi à l'attaquant de créer une redirection de port (pivoting) pour accéder au serveur FTP.

Sécurité des serveurs : La sécurité des serveurs web et FTP sur la VM1 aurait pu être renforcée en utilisant des mots de passe forts, en limitant l'accès aux répertoires sensibles et en surveillant les journaux d'accès pour détecter des activités suspectes.

Limites et Inconvénients :

Nécessité que le serveur SSH soit activé : Cette méthode de pivoting repose sur la disponibilité d'un serveur SSH actif sur la machine cible. Si le serveur SSH n'est pas activé ou s'il est désactivé, cette technique de pivoting ne fonctionnera pas.

Nécessité de connaître les login/mots de passe d'un utilisateur : Pour initier une session SSH sur la machine cible, l'attaquant doit avoir connaissance des informations d'identification valides (nom d'utilisateur et mot de passe). Cela signifie qu'une attaque réussie nécessite de disposer de ces informations ou de les obtenir par des moyens non autorisés.

Aucun serveur SSH natif sous Windows (sauf beta Windows 10) : Les systèmes d'exploitation Windows n'incluent pas nativement un serveur SSH. Bien que Windows 10 inclut désormais une version bêta du serveur SSH, il n'est pas encore largement utilisé, ce qui limite la portée de cette technique de pivoting dans les environnements Windows.

Ouverture port par port : Le pivoting nécessite l'ouverture de ports spécifiques pour établir des tunnels SSH. Cela peut devenir fastidieux si plusieurs machines internes sont ciblées, car chaque tunnel nécessite l'ouverture de ports distincts.

Tunnel TCP : Cette technique de pivoting repose sur la redirection de ports TCP. Elle ne prend pas en charge les protocoles de niveau supérieur, tels que les protocoles UDP. Par conséquent, certaines applications ou services qui utilisent UDP ne peuvent pas être exploités via ce type de pivoting.

En résumé, bien que le pivoting via SSH soit une méthode efficace pour accéder à des machines internes à partir d'un point d'entrée compromis, il comporte des limitations, notamment la nécessité d'un serveur SSH actif, la connaissance des informations d'identification, et des contraintes liées aux systèmes d'exploitation et aux protocoles supportés. Les administrateurs système doivent être conscients de ces limitations pour renforcer la sécurité de leurs réseaux.