Hogwarts Management System

Web Application Penetration test

Selina Fahy

CMP319: Ethical Hacking 2

BSc Ethical Hacking Year 3

2020/21

Unit 1: plain black text.

Unit 2: highlighted yellow

Note that Information contained in this document is for educational purposes.

.

Abstract

Web application penetration testing is a method that allows for security tests to be ran against a website in order to test for security weaknesses and assess the impact these weaknesses have on the owning person(s) of the website. This report aimed to assess and document risks that a malicious hacker could pose should they have a valid account, or wish to attempt to gain access to the website.

Following the OWASP methodology, the tester conducted the tests and assessed the vulnerabilities that were found. These tests were completed using a variety of tools that would return information regarding the vulnerabilities. Many of the vulnerabilities were common and were documented within this report. At the end of the test the website was determined to be too vulnerable for immediate use without the appropriate mitigations in place.

.

Contents

.

.

.

1 INTRODUCTION

1.1 BACKGROUND

Website Application Testing is a technique that allows tests to be run on applications that are hosted on the World Wide Web, and all corresponding functionalities and interfaces. In this case, where the technique is specifically revolving around security and penetration testing, the tests are performed in order to verify if the application is secure and measure that level of security (Web Application Testing - Tutorialspoint, 2020).

With the majority of activities being able to be done through the internet there is an obvious increase in popularity of online web applications. This can, therefore, introduce the possibility of malicious third parties trying to gain access to the potentially sensitive data that can be stored or sent through the applications, specifically applications that are publically exposed (unlike an intranet on private networks, etc.).

Overall, penetration testing (pen testing) is a "preventive control measure that allows you to analyze the status of the present security layer of a system" (Guide to Web Application Penetration Testing, 2020).

Hogwarts management system has requested for a web application pen test, and a user-level account has been administered to the penetration tester, who will attempt to find vulnerabilities and errors that could be exploited by users of the website (e.g. student/staff/malicious hackers).

Through this, Hogwarts management system will be able to get an in-depth and focused report on any potential issues with the web application. As, unlike just using automated scanning and pen testing tools, a penetration tester can manually ascertain issues and deal with more complex testing in which can lead to few false negatives (Kinsbruner, 2020). This can then lead to less extra work for the corresponding team who would need to validate whether a reporting error was true or false.

1.2 AIM

The aim of this report is to conduct a web application penetration test to assess and document the risks that an attacker, who has a valid account, poses.

Through the use of a methodology, the penetration tester will conduct a structured series of attacks in hopes to identify all possible vulnerabilities. This report will endeavor to capture and explain the attacks, the meaning behind them as well as their significance to the company.

In order to achieve this the following objectives should be met:

- Active and passive reconnaissance – gaining information about the web application through the use of tools like 'nmap', 'shodan', 'nslookup' command, googling information, etc.
- Enumeration of the web application
- Using the methodology selected to find and document vulnerabilities – with the information gathered in step 1 as a basis and using tools like 'nmap', 'nessus', 'ZAP', etc. to find and exploit vulnerabilities.
- Reporting and recommendations – report all finding of vulnerabilities supported by data/evidence, categorize them and recommend 'fixes' to remove the risks of exploitation.

1.3 METHODOLOGY

The penetration tester will follow the OWASP v4 methodology (tanprathan/OWASP-Testing-Checklist, 2020). This method is a part of OWASP and was created by many cybersecurity professionals and volunteers in order to produce a comprehensive guide to web application security testing.

This methodology covers the following:

- Information gathering – tools used to gather relevant information: nmap, Burp suite, ZAP.
- Configuration and Deploy Management Testing – tools used: Nessus, Nikto, gobuster, Burp suite.
- Identity Management Testing – tools used: Burp suite.
- Authentication Testing – tools used: Burp suite, SQLmap, Mantra.
- Data Validation Testing – tools used: SQLmap, Burp suite.
- Error Handling – tools used: Burp suite.
- Cryptography – Nessus, Nikto.

2 PROCEDURE AND RESULTS

2.1 OVERVIEW OF PROCEDURE

Following the OWASP V4 methodology, the first step was information gathering and web application mapping. This has been documented with the information that was able to be gathered through the Admin authorization in order to avoid repetition later on.

2.2 PROCEDURE PART 1

2.2.1 Information gathering

Following the OWASP V4 methodology, the tester started with information gathering and the tools that had mostly been used were Burp suite and 'gobuster'.

2.2.1.1 Burp suite – Site Map

Burp suite is a tool commonly used to map out web applications as well as to attack them, for example brute forcing usernames and passwords, etc.

For the start of the pen test the tester had used Burp suite to create a site map. This is created a map of the contents of the website as the tester went from page to page and testing all the linked and user input fields that the tester came across.

However, Burp suite cannot find pages that are not linked to the main pages.

The full map of the web application can be found in Appendix A.

2.2.1.2 Gobuster

'Gobuster' is a directory brute forcing tool that allows for common directory names to be tried against a website to see if the website has these directories.

Through the use of 'gobuster' the tester was able to look for other directories (Figure 1) for the website that could not be found through the use of Burp suite. Also through the use of ZAP and authenticated access (admin) a more in-depth search was conducted (Appendix A section b).

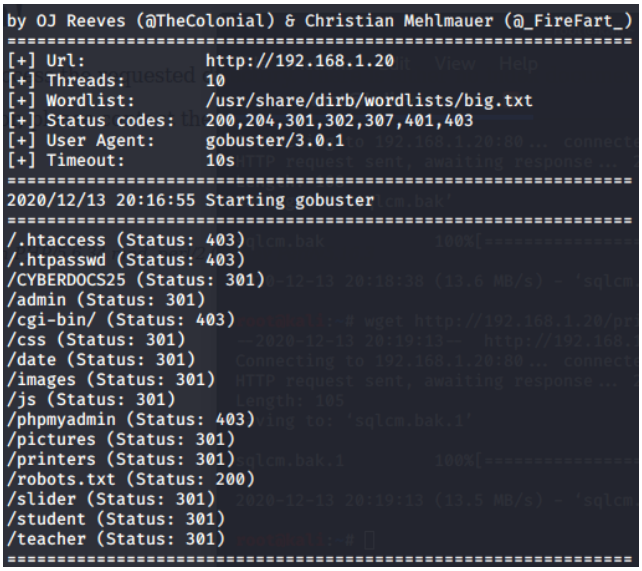


Figure 1 – Gobuster directories

Furthermore, through the use of different lists it was possible to find a few more hidden directories such as phpinfo.php (Appendix B), which revealed important information about the version of PHP being used and its directory.

Also, having a look through the /CYBERDOCS25 and the /printers directories lead to finding a file called sqlcm.bak which seems to be an SQL backup file that would monitor SQL attacks on the login pages of the website.

```
<?php $username= str_replace(array("1=1", "2=2", "UNION","Union","2=2","'b'='b'","1 =1"), "", $username); ?>
```

Figure 2 – sqlcm.bak

2.2.1.3 Robots.txt

After searching through directories the tester came across the robots.txt file. The robots.txt file is a file intended to be used by search engines that use content crawlers to get information about allowed and disallowed pages.

The Robots test file contained the directory for a text document.

Robots.txt:

*User-agent: **

Disallow: ZJDTYMXCTVQT/doornumbers.txt

The text file was extracted through the use of the 'wget' command.

Doornumbers.txt:

Keypad entry numbers for company rooms:

Room 1526 - 2468

Room 2526 - 1357

Room 3615 - 5678

The file retrieved was a file that seeming contained codes for doors, it is presumed that this would allow you to access these rooms.

2.2.1.4 Webpage Comments and Metadata for Information Leakage

Next, the tester looked through the HTML on the web pages by bringing up the developer tools and inspecting the website, as well as what could be seen through ZAP.

This allows anyone to be able to look at how the website had been constructed, viewing HMTL, CSS, JavaScript, etc.

Through the Hogwarts Management System, the tester was able to find information that was left in the comments, which can be seen Figure 3.

```
<!-- ***Note to self: Door entry number is 1846 -->
```

Figure 3 – Door key comment in source code

Furthermore, the tester was able to find reference to the phpinfo.php file that can be access on the system as seen in Figure 4, though it does claim that access should be disabled in the 'real version'.

```
<!-- *** Remember that phpinfo.php should be deleted in the real version -->
```

Figure 4 – phpinfo.php commented in CSS file

2.2.1.5 Data entry points

While the tester was moving through the website, the tester made note of all the data entry points that all users had. This was confirmed through the usage of Burp suite, which allowed for the tester to see that the information inputted by the user was being sent to the web server (Figure 5).

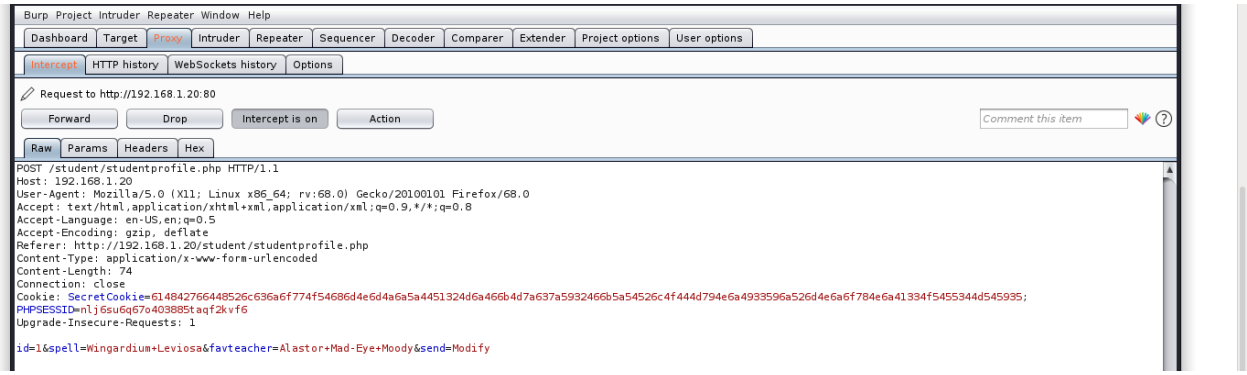


Figure 5 – input fields of the profile area captured in Burp suite

An overview of the entry points for student users:

- Login – index.php
  - Username
  - Password
- Profile – studentprofile.php
  - Picture upload
  - Favorite spell
  - Favorite teacher
- Change password – changepassword.php
  - Old password
  - New password

An overview of the entry points for admin users:

- Login – index.php
  - Username
  - Password
- Profile – profile.php
  - Picture
  - Favorite spell
  - Favorite teacher
- Change password – changepassword.php
  - Old password
  - New password
- Add teacher
  - Teacher name
- Add subject
  - Subject name

Furthermore, through the use of Burp suite, hidden fields were found on the website. These hidden fields appeared on all accounts (student/admin etc.). These fields allowed for one user to be able to edit another users' information, as these hidden fields set the 'id' (Appendix C).

2.2.2 **Configuration and deploy management testing**

2.2.2.1 Nikto

Nikto is a web server scanner that will tests for website vulnerabilities based on information about the versions used. However, given the case of version checking it is possible for Nikto to give false positives, and therefore it would require confirmation (Kali, 2020).

Through Nikto the tester was able to see various alerts for outdated versions of PHP, OpenSSL and so on (Figure 6 and 7).

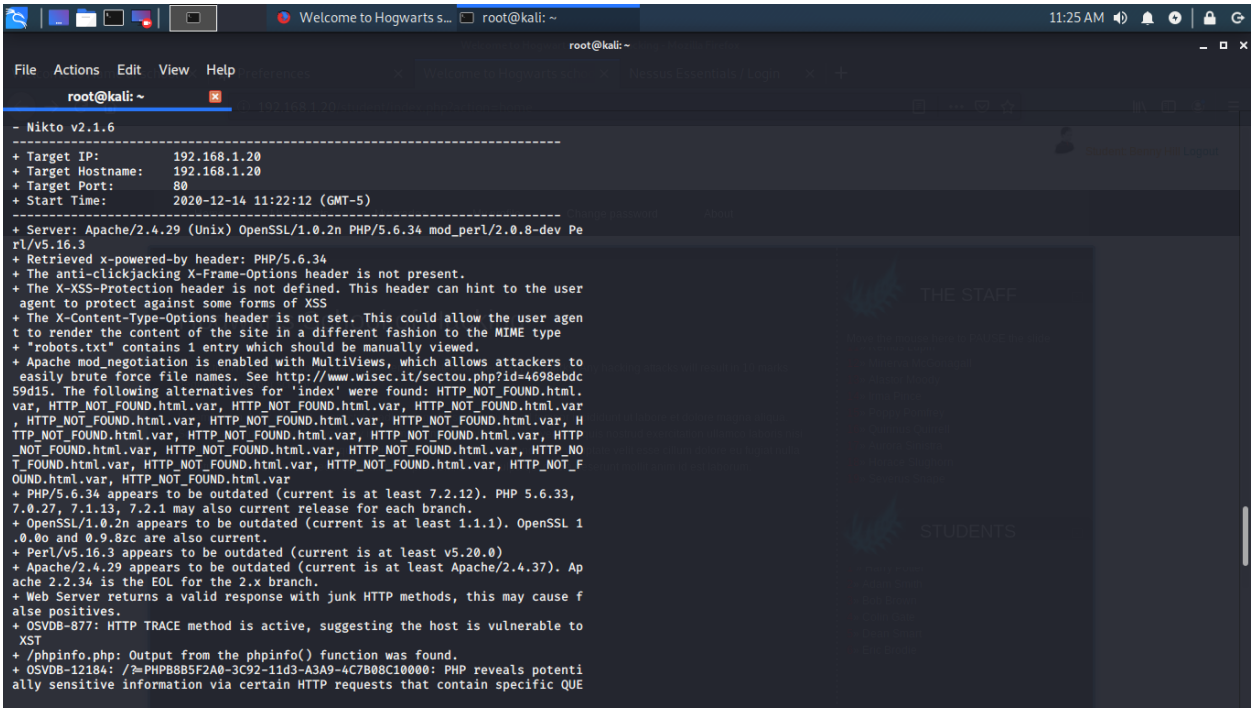


Figure 6 – Nikto output

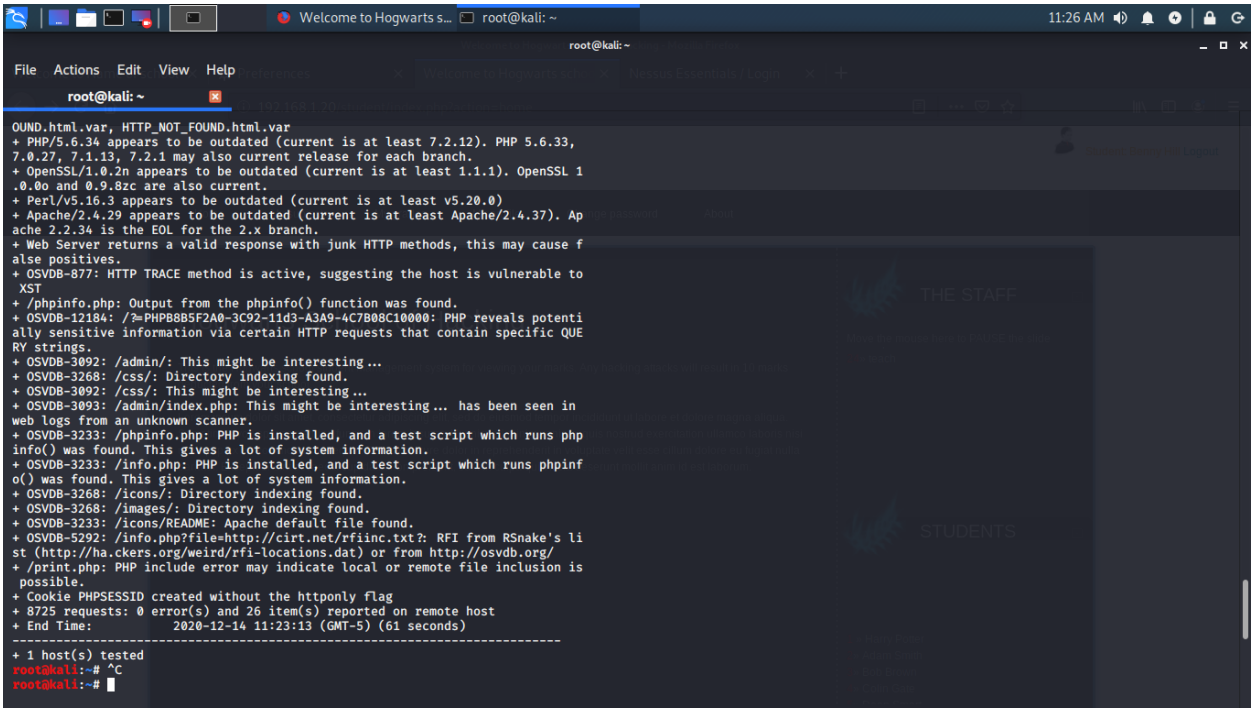


Figure 7 – Nikto output

2.2.2.2 Nessus

Nessus is a remote security scanning tool which allowed the tester to scan the web server and check for vulnerabilities (Nessus, 2020).

After downloading and activating the Nessus tool the tester was able to see the many vulnerabilities and alerts that Nessus found. An overview of what was found can be seen in Figures 8, 9, 10 and 11.

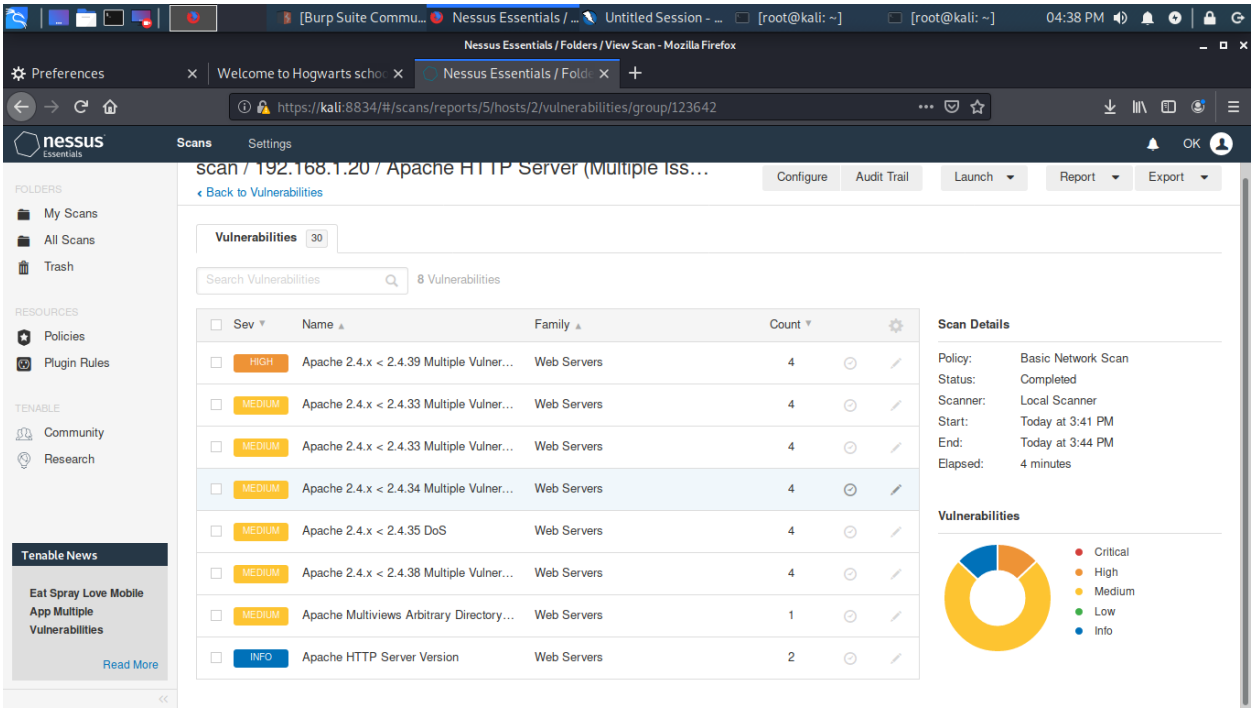


Figure 8 – Apache HTTP server vulnerabilities

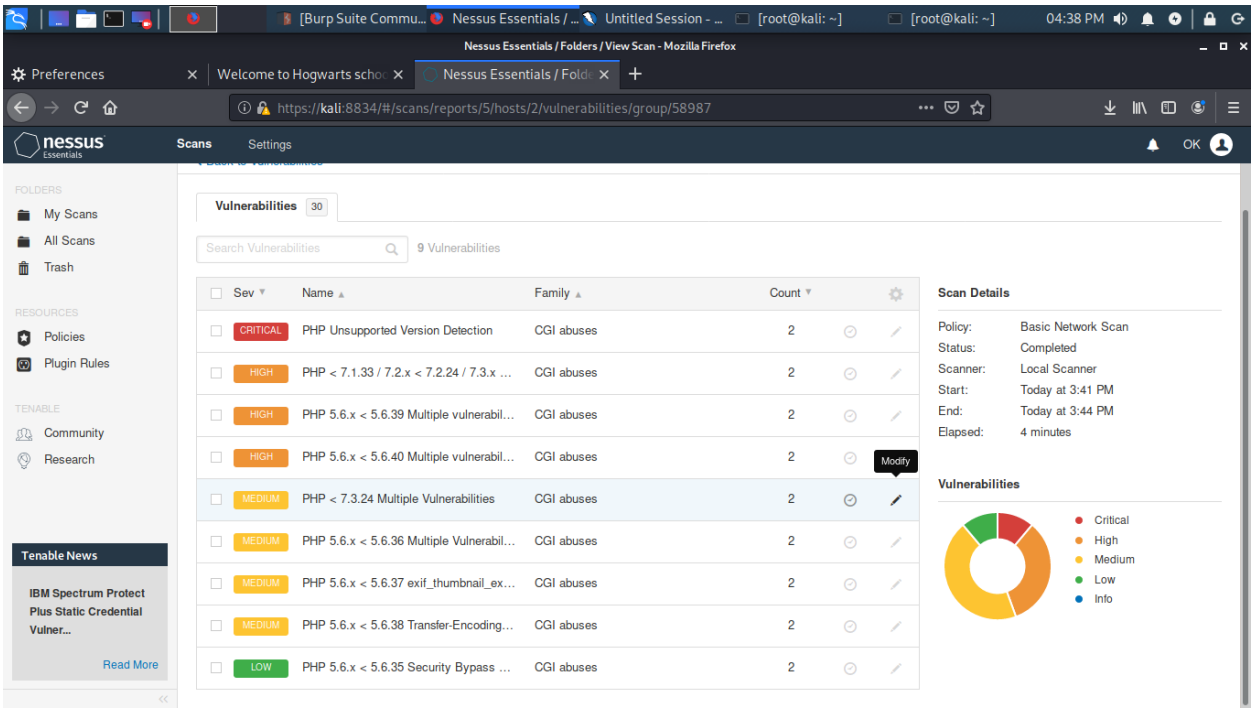


Figure 9 – PHP vulnerabilities

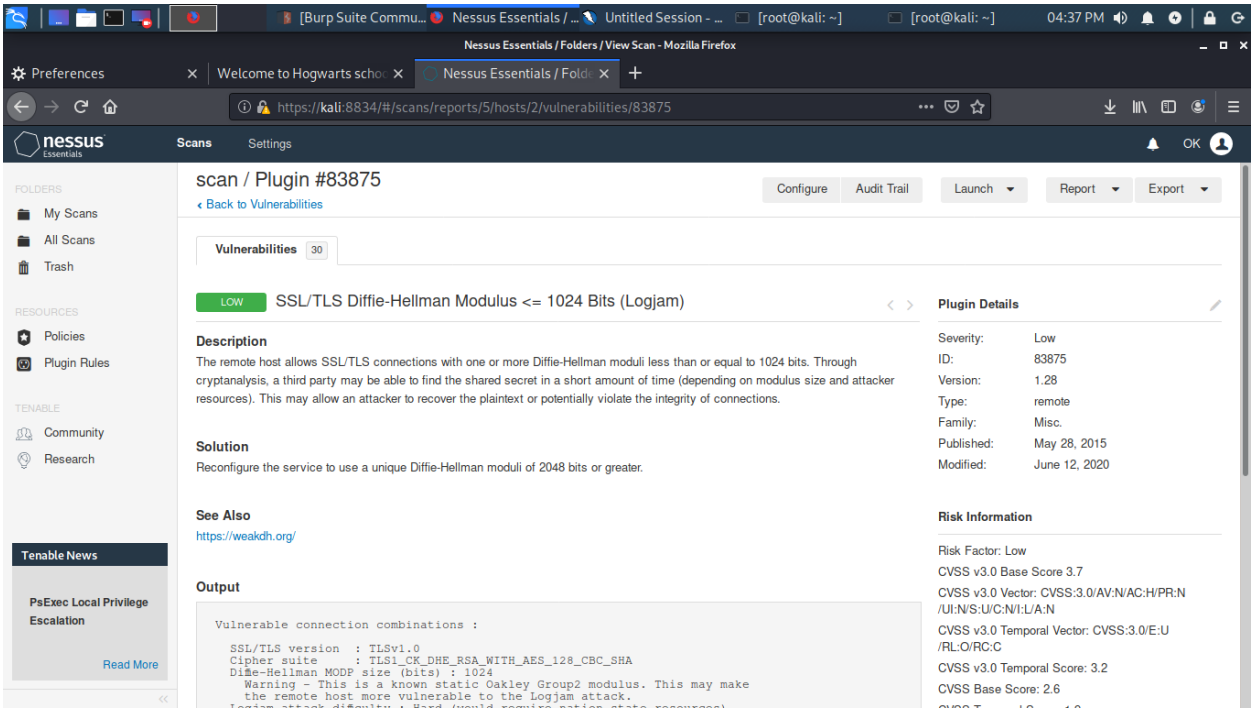


Figure 10 – SSL/TLS vulnerability

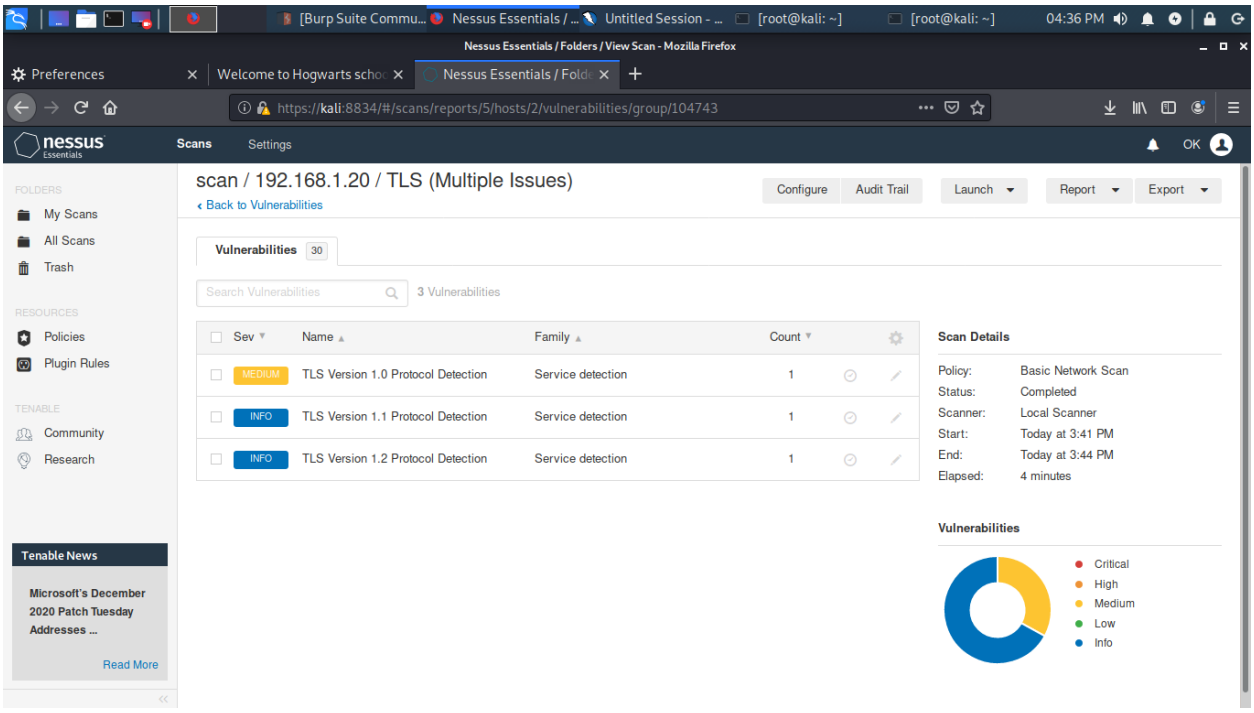


Figure 11 – TLS vulnerabilities

Overall, Nessus claims that the Hogwarts Management System is vulnerable to many Denial of Service vulnerabilities through the outdated version of OpenSSL, potential remote file inclusion with the outdates PHP version, and cross-site scripting (XSS) and other vulnerabilities for the outdates Apache server version.

2.2.3 **Identify management testing**

2.2.3.1 Weak or unenforced password policy

Having a strong password policy is of great importance when it comes to protecting a website from attacks. Specifically against brute forcing and password guessing.

During the penetration test the tester came across the 'change password' section of the menu bar, and attempted to change the password. Upon changing the password the tester did not see any form of policy in creating the new password. There was a lack of acknowledgment for the length of the password and whether or not it contained any numerical or special characters.

In Figure 12 it can be seen that the tester was able to change the password to a single letter; 'd'.

```
OldPassword=d&NewPassword=d&ConfirmPassword=d&Submit=Submit
```

Figure 12 – Password changing Burp suite

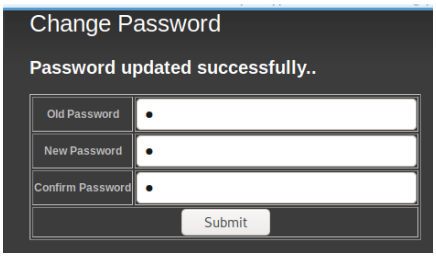


Figure 13 – The password successfully updated

2.2.3.2 Weak or unenforced username policy

At the beginning of the penetration test, Dr. Albus Dumbledore (the owner of the application) gave the username of hpotter to Harry Potter (the account used by the tester). This was a signal that the other usernames could be set up in a similar fashion. So, following this and gaining the full names of all the students and teachers (Figure 14), after successfully logging in as hpotter, the tester was able to enumerate all the usernames of students and teachers.

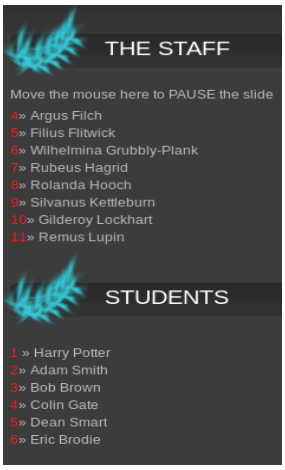


Figure 14 – Names of users

2.2.4 **Authentication testing**

2.2.4.1 Mantra

Cookies

OWASP mantra is a browser based security framework that can be used to help in exploits and vulnerability finding (OWASP Mantra - Security Framework, 2020).

Through the use of the tool OWASP mantra and Cookies Manager+ the tester was able to get the cookie and the PHPSESSID after logging in with the provided credentials (Figure 15 and Figure 16).

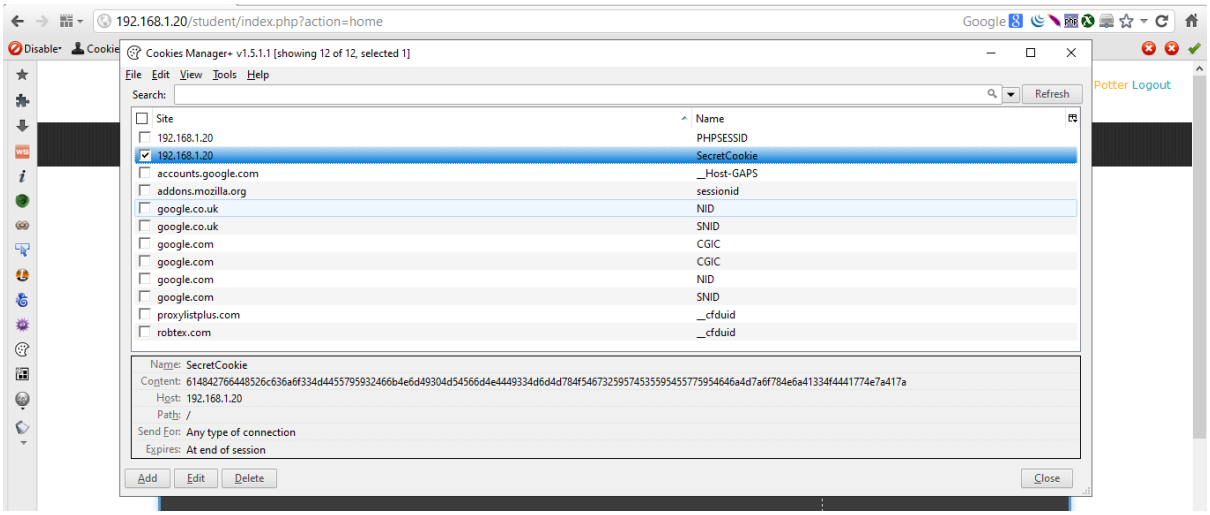


Figure 15 – Cookies Manager+

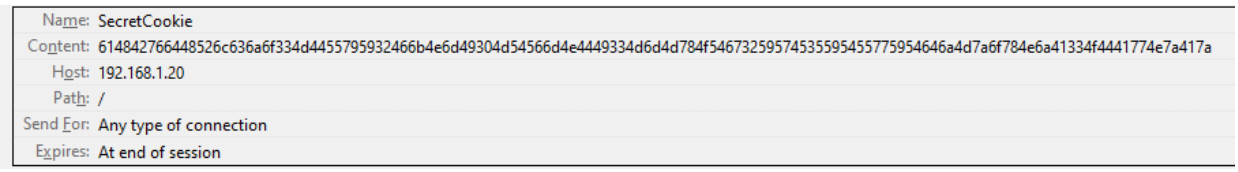


Figure 16 – Secret cookie that reveals login information

Through this the tester was able to find that the cookie was encrypted using hex and base64, providing the username followed by the password which was hashed using MD5.

This is a potential risk given that MD5 is a very weak hash, and easily broken, the detailed decryption can be seen in Appendix E.

2.2.4.2 HTTP or HTTPS – Burp suite

Through the use of Burp suite the tester was able to confirm that the website used HTTP connection over HTTPS. HTTP is an unsecure connection that sends data over port 80 (Figure 17) (What is the Difference Between HTTP and HTTPS? - KeyCDN, 2020).

Figure 17 – Burp suite connection

2.2.4.3 Weak password change or reset functionalities

On the website's menu bar that allows for the user to change their password, and had been previously explore for the detection of password policies, the tester tested for any issues regarding the changing process.

Here the tester came across the form to test for any restrictions on changing the password. The tester deliberately entered various random letters to the 'Old password' field in the form and typed in a new password ('test'), in order to test to see if the web application is checking that the user has entered the correct original password (Figure 18).

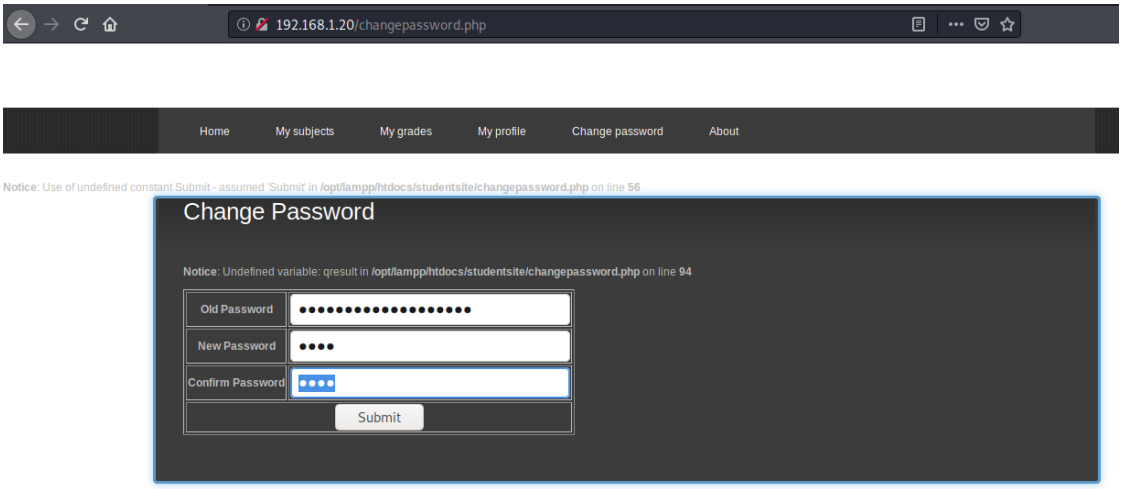


Figure 18 – Password checking test

This resulted in the old password parameter not being checked against the original password, and resulted in the password being changed to the new one (Figure 19).

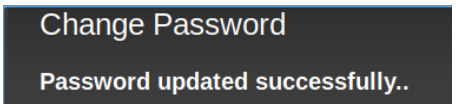


Figure 19 – Successful update

2.2.5 **Data Validation testing**

2.2.5.1 SQLmap

SQLmap was a tool employed by the tester in order to test for SQL injection. This tool automated SQL injections in order to determine if there are any SQL flaws and if it is possible to take over the database server (sqlmap: automatic SQL injection and database takeover tool, 2020).

By using Burp suite to capture the information that would be passed to the web server and saving it as a text file the tester was able to use SQLmap in order to test for SQL injection as well as attempt to gain information about the data that might be stored in the database, using the command 'sqlmap –r data.txt'.

After successfully gaining access to the database the tester requested for all the databases connected to the server to be dumped. As can be seen in Figure 20 below there are several databases that were presumed to be connected to other websites. This is a critical vulnerability as not only is Hogwarts Management System in danger if an attacker was able to successfully enumerate the database information, but other websites and users would be affected too.

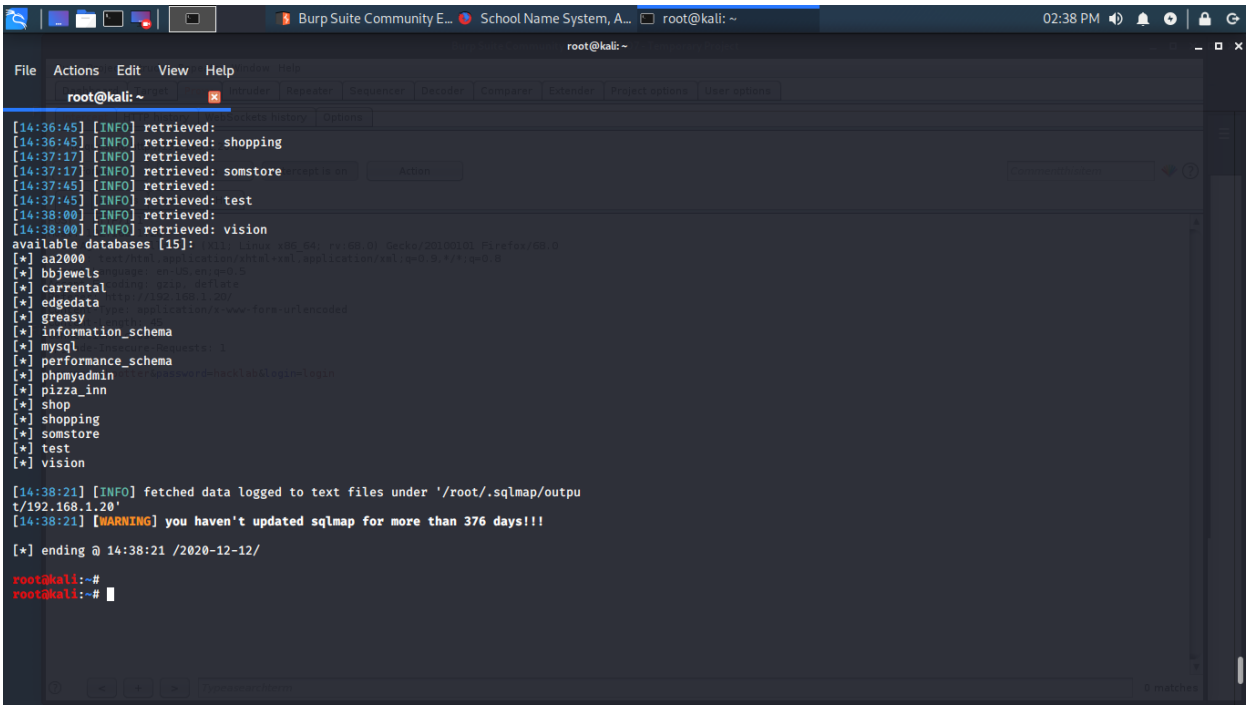


Figure 20 – databases found on the server

The tester only worked with the information regarding Hogwarts Management Systems.

Firstly, it was seen that the Hogwarts Management Systems was connected to the visions database, and after dumping all the information from there the tester found all the users credentials in the 'users' table. Through using SQLmap the hashed passwords were also able to be cracked, and the tester had access to the majority of accounts on the website. The detailed screenshots can be found in Appendix D**.**

Furthermore, access to the admin account was achieved through injecting SQL commands into the username field of the login page as seen in Figure 21 and the successful access to the admin page and user in Figures 22 and 23.

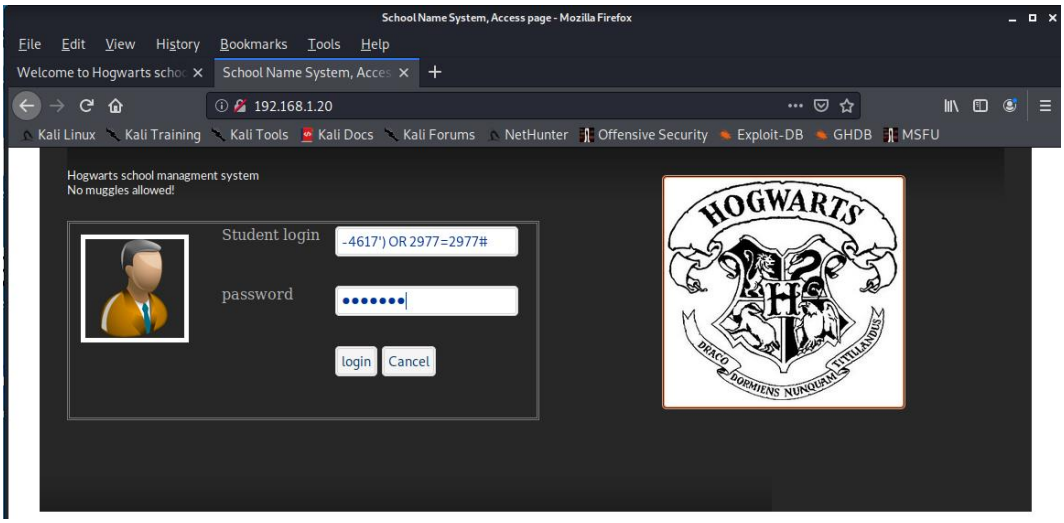


Figure 21 – SQL injection in username field

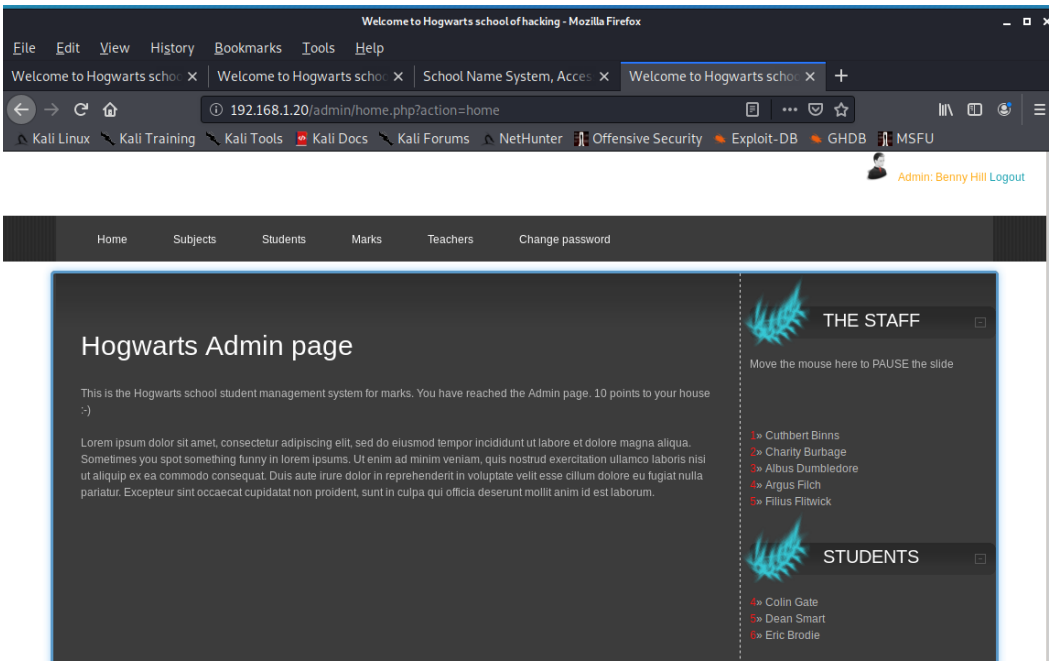


Figure 22 – Successful execution and received admin

Figure 23 – Admin user

2.2.5.2 Script injection (XSS)

Following on from the hidden input fields found in **Information gathering - Data entry points**, the tester found that some of these fields are susceptible to reflected cross-site scripting.

Reflected XSS is when JavaScript can be injected into the webpage and can display information that the user asks for, though is not stored on the web application server (Academy and scripting, 2020).

In Figure 24 and Figure 25 a successful XSS attack occurs and returns the value that the tester asked for.

Figure 24 – JavaScript injection attack

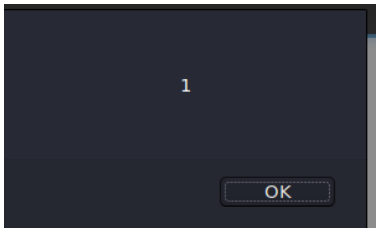


Figure 25 – JavaScript successfully reflected

This can potentially allow a user to inject malicous JavaScript code into the web page and affect other users who might be present during the session.

2.2.6 **Error handling**

2.2.6.1 Analysis of error codes

Error handling is an important part of securing a web application. Improper handling of errors can lead to leaks of important information such as; database dumps, error codes, etc. (Improper Error Handling | OWASP, 2020).

During the testing of the web application, the tester came across many errors that reveal important information about the implementations of the database servers as well as the username and passwords of the users.

In regards to the login page, we get two errors; 1) when the username is incorrect (Figure 26) and 2) when the password is incorrect (Figure 27). This allows for the attack to know when a username is correctly guess and if the password matches or not.

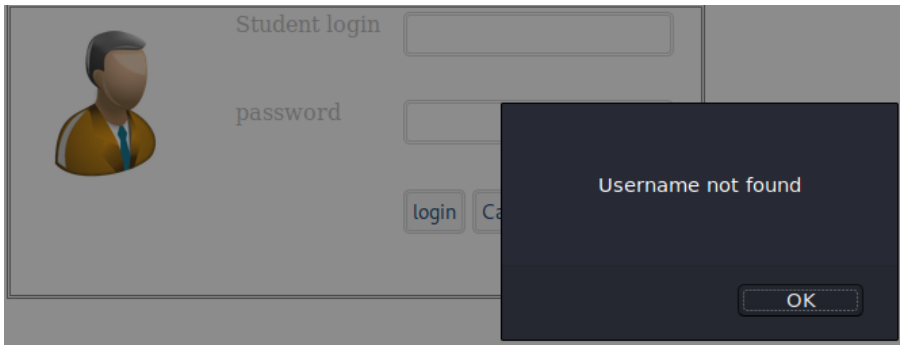


Figure 26 – Incorrect username

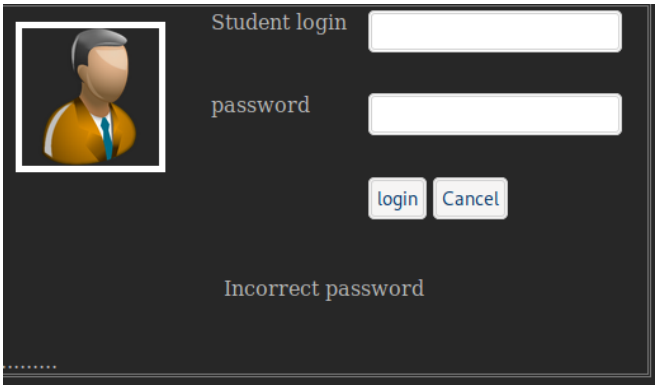


Figure 27 – Incorrect password

Furthermore, within the login area, more information about the construction of the 2 fields is revealed during a 'UNION' attack (Appendix F section a).

In other areas where users can input data, more information is revealed such as; the location of the edits and where the original information is (Figure 28).

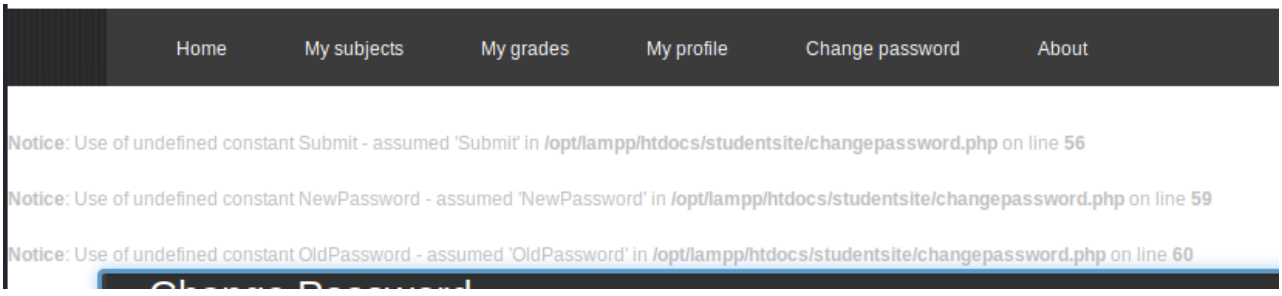


Figure 28 – location of changepassword.php as well as the new and old password

There is further errors which show more information in the admin area of the website, where SQL commands can be inject and return the syntax error as well as the command the input is being passed to (Appendix F section b).

2.2.7 **Cryptography**

2.2.7.1 Testing SSL/TSL ciphers

Following the Nessus and Nikto vulnerability alert for OpenSSL on the web application the tester went and looked for the vulnerability/exploit for this particular version (Figure 29).

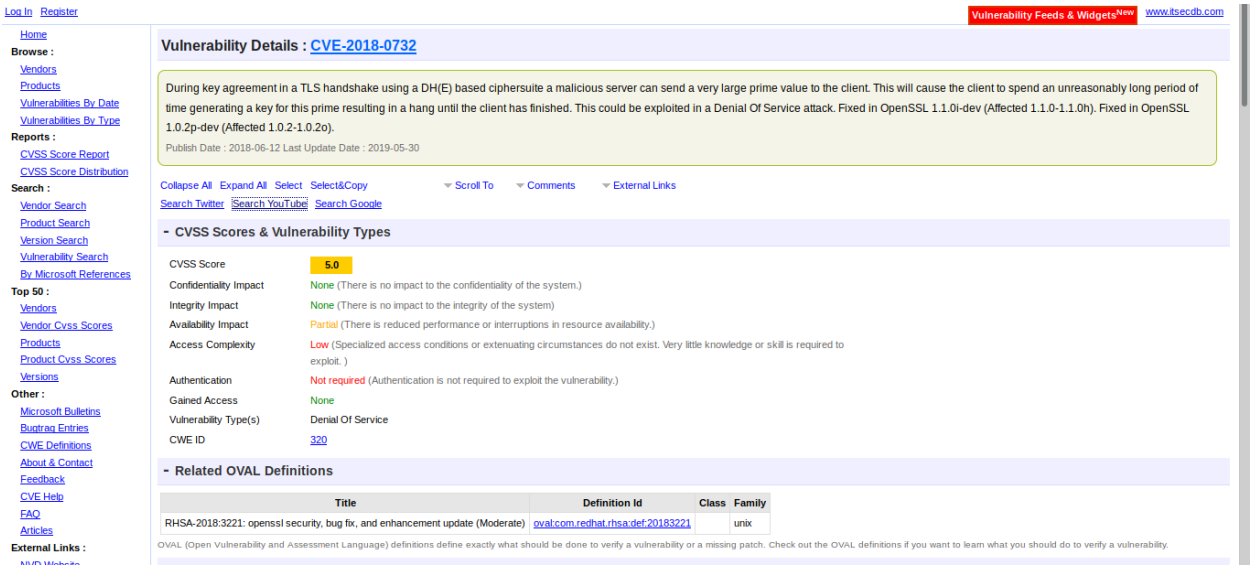


Figure 29 – OpenSSL Vulnerability exploit

For the version that is running on the web application, the vulnerability is mostly susceptible to Denial of Service attacks and overflow.

3 DISCUSSION

3.1 SOURCE CODE ANALYSIS

Source code analysis allows for analysers to manually and/or automatically find security related problems in software (Source Code Analysis Tools - Overview | CISA, 2021). Below is the documentation of vulnerabilities that the tester found having been provided the source code for the website.

3.1.1 Brute-forceable Admin password

Through the use of bruteforcing tools such as Hydra, ZAP and so on, it is possible for a malicious hacker to brute force a user's account. Through this it was possible for the tester to gain access to the administrators account. This was noted by a lack of a counting variable or function that would then restrict a user/IP address from further attempts after a set number of failed attempts have been reached.

3.1.2 Robots.txt

Through the use of the 'gobuster' tool the tester was able to find a 'robots.txt' file on the web server. After having found it and located a disallowed folder and traversing to it, found information about room numbers and keys. This security issue has the potential to allow for a malicious attacker to attempt physically entry to the company, and allow for further attacks. The robots.txt file was found in the root directory of the website and was visible to the public.

The robots.txt file should not be a file used to hide sensitive information, as it is being used here (Figure 30 and Figure 31).

```
User-agent: *
Disallow: ZJDTYMXCTVQT/doornumbers.txt
```

Figure 30 – Robots.txt

```
Keypad entry numbers for company rooms:
Room 1526 - 2468
Room 2526 - 1357
Room 3615 - 5678
```

Figure 31 – Robots.txt room keys

3.1.3 Local File Inclusion

Through having a look at the extras.php file, it was determined that the website was vulnerable to Local File Inclusion (LFI). The source code that denotes this can be seen in Figure 32.

```
<?php
  $pagetype = $_GET['type'];
include('lfifilter.php');
    include ($pagetype);
?>
```

Figure 32 – Local File Inclusion vulnerability

This vulnerability allows a malicious attack to have the ability to traverse files on the machine that hosts the web server through the '?action=home.php' by changing the 'home.php' section to something such as, '../../etc/shadow'. Though it has been noted that there has been an attempt to stop this in the form of the 'lfifilter.php' file (Figure 33), however this can be bypassed through the use of obfuscation. In such the use of '../' is changed to a different format that may not be recognized during the filtering of the input and can therefore lead to the execution of the command in the URL bar.

```
<?php
$pagetype = str_replace( array( "../", "..\"" ), "", $pagetype);
?>
```

Figure 33 – lfifilter.php

3.1.4 Hidden source code

Through looking through the source code for the website, the tester was able to find a few comments that contained some sensitive information regarding the website (Figure 34) as well as about the company/doors (Figure 35).

```
<!-- *** Remember that phpinfo.php should be deleted in the real version -->
```

Figure 34 – Comment at top of index.php file

```
<!-- ***Note to self: Door entry number is 1846 -->
```

Figure 35 – Hidden comment

These bits of information can put the website and the company at risk, as Figure 34 shows that there is a phpinfo page that will show all the information about the website and the versions of code that are being used. While Figure 35 shows once again an entry number for another room within the company.

3.1.5 Reversible cookie

After looking at the 'cookie.php' file the tester took note of the section that handled the cookie content and the hashing process.

The cookie contained the username and the md5 hashed password (Figure 36) of the user and a time stamp and then encoded it using base64 and hex (Figure 37).

```
$pass=md5($password);
```

Figure 36 – Password MD5 hash

```
<?php
$str=$username.':'.$password.':'.strtotime("now");$str = bin2hex(base64_encode($str)); setcookie("SecretCookie", $str);
?>
```

Figure 37 – Cookie.php contents

This lack of security would make it easy for a malicious attacker to gain user credentials if they are able to spoof a user's cookie. Through the use of tools such as icyberchef.com, it would be simple for a malicious hacker to decode the cookie and get the credentials.

3.1.6 Cookie attributes

Through scans, the vulnerability of HTTPOnly not being set was brought to light. HTTPOnly flag allows for basic Cross-site scripting to be filtered and even if this specific flaw is exploited it would not expose the users' cookie to a third party (HttpOnly - Set-Cookie HTTP response header | OWASP, 2021).

Therefore, looking through the cookie.php file it was noted that there was a lack of secure and HTTPOnly flags, as can be seen in Figure 37 above in the "setcookie".

3.1.7 Directory browsing

For directory browsing to occur the variable related to it in the configurations file would need to be enabled. Hogwarts management systems website has directory browsing enabled, in which a user is able to get a list of directories for the website. Though not ultimately a vulnerability, it is suspected that there is a lack of access control, which can lead to the potential harm that a malicious hacker could do if they were able to get to the directory index (Center, Definitions and listing, 2021). This may allow for a malicious hacker to be able to gain access to the php of the website and circumvent any security measures that have been put in place on the entire website.

3.1.8 Unlimited login attempts

Through tests it was found that there were no implementations regarding restricted attempts on accessing a user's account. This makes it easier for a malicious hacker to brute force user credentials and also much quicker.

Through this the tester went through the index.php file and did not find any code relating to any form of lockout after failed attempts of logging in.

3.1.9 User enumeration

While testing the website the tester found that there was a lack of appropriate error handling on the login page, and it was possible to test usernames until a 'confirmation' error was returned that informed the user that the password for the username entered was incorrect, over the username not found error (Figure 26 and Figure 27).

Having looked through the source code files it was noted within the index.php file the 'Incorrect password' error is shown when the username is valid and at least one row in the database is returned but does not satisfy the SQL command of both the username and the password entered being equal to the username and password stored in the database, which can be seen in Figure 38.

```
        $sql = "SELECT * FROM users WHERE username=('$username') AND password='$password'";
        $result = mysql_query($sql);

?>
        <?php
                //check that at least one row was returned
                $rows = mysql_num_rows($result);
                $row=mysql_fetch_array($result);

                |
if($rows>0) {
                session_start();
                $_SESSION['user_id']=$row['user_id'];
                $_SESSION['student_id']=$row['STUDENT_ID'];
                $_SESSION['level']=3;
        ?>
        <p align="center">Login Successful</p>
        <br />
        <br />
        .........
        <p align="center">
          <meta content="2;student/index.php?action=home" http-equiv="refresh" />
        </p>
        <?php


} else {

                ?>


        <p align="center">Incorrect password</p>
```

Figure 38 – Code returning the 'Incorrect Password' error

In which there is an included file (username.php) pertaining to the usernames – which returns the 'Username not found' error when a username not found in the database is entered, thus making it possible to enumerate the username/password combinations (Figure 39).

```
if($rows==0){
       echo '<script language="javascript">'; echo 'alert ("Username not found");'; echo 'window.history.back();'; echo '</script>'; die();
```

Figure 39 – Code returning the 'Username not found' error

3.1.10 No HTTPS

As has been noted by the tester, the website does not run HTTPS, but HTTP instead. This make the website more susceptible to sniffing attacks, as well as allowing any information obtained to be tampered with as there is a lack of encryption of the data being passed to and from the web server.

3.1.11 File upload

Parsing through the changepicture.php file and the studentprofile.php file, a file upload vulnerability was found. This vulnerability allows for the uploading part of the website to be abused and malicious attacks to be attempted.

The code pertaining to this vulnerability only stops the submission of file types other than images, while it is possible for a hacker to change the file type/extension after the submit button is selected through the use of a proxy, such as burp suite, where the extension can be altered.

The source code that was noted for this vulnerability can be seen in Figures 40 and 41.

```
################################################
# 1 - Filetype invalid
################################################
if ($fileuploadtype=="TYPE" || $fileuploadtype=="ALL"){
$validtypes= array("image/jpeg","image/jpg","image/png");
if(in_array($file_type,$validtypes)=== false){
        echo '<script type="text/javascript">alert("Invalid filetype detected -
        echo "<script>document.location='$nextpage'</script>";
        exit();
}
}
```

Figure 40 – File type filter

```
################################################
# 2 - Extension invalid
################################################
if ($fileuploadtype=="EXT"|| $fileuploadtype=="ALL"){
$extensions= array("jpeg","jpg","png");
if(in_array($file_ext,$extensions)=== false){
        echo '<script type="text/javascript">alert("extension not allowed, pleas
        echo "<script>document.location='$nextpage'</script>";
        exit();
}
}
```

Figure 41 – extension filter

Furthermore, there was the issue of the files uploaded being allowed to be executed by anyone. When parsing through the 2 files mentioned, the tester found an interesting line of code seen in Figure 42. This line allows for any user to be able to read, write, and execute the uploaded files (Banting, 2021).

```
chmod($target_path,0777);
```

Figure 42 – chmod permissions allowing for execution of uploaded files

3.1.12 Cross Site Request Forgery (CSRF)

The tester was able to determine that there was a fault with the update password section. After looking at the source code for this section it was noted that this is susceptible to Cross-site Request forgery. This will allow someone else to be able to alter a user's password through conventional means, such as by sending them a link that will allow the malicious attacker to change the user's password without them knowing. The source code can be seen in Figure 43 and Figure 44.

```
include("updatepassword.php");
```

Figure 43 – changepassword.php include updatepassword.php

```
<?php
                $sqlupd="UPDATE users SET password='$newpass' WHERE user_id='$studentnumber'" ;
?>
```

Figure 44 – updatepassword.php content

3.1.13 SQL Injection vulnerability

After looking through the index.php file the tester was able to locate a section in the php code that was vulnerable to SQL injection (Figure 45).

```
$sql = "SELECT * FROM users WHERE username=('$username') AND password='$password'";
```

Figure 45 – SQL injectable code

Figure 45 shows that the user's input is directly inputted into the sql command, therefore making it possible to inject malicious code.

However, it was noted that there was an attempt to filter through this, as can be seen in Figures 46 and 47.

```
//SQL countermeasure.
include 'sqlcm.php';
```

Figure 46 – SQL injection include file in index.php

```
if(preg_match("[0=0|1=1|2=2|3=3|4=4|5=5|6=6|7=7|8=8|9=9|SELECT|select|3=3|1 =1|'b'= 'b']", $username)){
```

Figure 47 – sqlcm.php contents

3.1.14 Php information disclosure vulnerability. – phpinfo.php

It was noted that there was a phpinfo.php file within the websites directory and after looking at the contents (Figure 48) found that it relayed all information about the website versions.

```
<?php

phpinfo();

?>
```

Figure 48 – phpinfo.php contents

3.1.15 Hidden guessable folder

Through the use of 'gobuster' previously the tester was aware of a hidden file called CYBERDOC25, which was confirmed when looking at the directories. Hidden folders such as this are usually places where sensitive information can be found – especially if the folder name can be bruteforced and therefore be found.

In this case it can be seen in Figure 49 that this hidden folder had the SQL database backup file in it, which lead to the tester learning earlier on about the SQL injection filters that were in place.

```
<?php $username= str_replace(array("1=1", "2=2", "UNION","Union","2=2","'b'='b'","1 =1"), "", $username); ?>
```

Figure 49 – sqlcm.bak contents

3.2 VULNERABILITIES DISCOVERED

3.2.1 Weak or unenforced username policy

A lack of a strong username policy makes it easier for a malicious hacker to enumerate the usernames of the users of the website. As was seen in section 2.2.3.2, all the usernames were the first initial followed by the last name.

After enumerating usernames of users on the website, it becomes much easier for a malicious hacker to attack the website and gain access to accounts.

3.2.2 Weak or unenforced password policy

As was noted by the tester, there was a lack of a strong password policy. This makes it easier for an attacker to guess or bruteforce the passwords against a list of usernames enumerated in the previous section. This will allow a malicious hacker to be able to gain access to user accounts.

3.2.3 Cross-site Scripting injection (XSS)

Cross-site scripting (XSS) is a common vulnerability that has the potential to be quite dangerous to the users and the website. Through XSS it is possible for a malicious hacker to gain information about a user, e.g. their unique cookie, or about the webpage/website, e.g. the version of php running.

3.2.4 Error handling

The error handling issue identified earlier, also, revealed too much information regarding the website and the folders that the information is being pulled from, as can be seen in section 2.2.6.

3.2.5 Brute-forceable Admin password

Through the use of external bruteforcing tools, such as Hydra or Burp Suite, it is possible for a malicious hacker to bruteforce any of the passwords, but most dangerously the admin password. This will allow for a malicious hacker to then have the ability to have complete control over the website and the users.

3.2.6 Reversible cookie

The cookies that are being generated by the webserver are easily decoded. With the use of outdated and easily broken encoding and hashing algorithms such as hex, base64, and md5 it would be a simple matter for an attacker to obtain the information stored inside the cookie, and in this case obtain a user's credentials.

3.2.7 Cookie attributes

On further inspection, the lack of attributes within the cookie adds to the risk that is posed against the users should a cookie be captured. The lack of a secure or HTTPOnly flag allows for unique cookies to be captured and read by third parties such as a malicious hacker.

3.2.8 Unlimited login attempts

By not tracking the number of failed attempt a certain IP address makes, can allow for a malicious hacker to have unlimited attempts in bruteforcing user credentials.

3.2.9 No HTTPS

By running the webserver on HTTP the connection made by the website and the user is not encrypted and makes it more vulnerable to man-in-the-middle attacks, where information passed between these two devices can be caught, observed, and potentially tampered with.

3.2.10 SQL Injection

The preg_match function within the sqlcm.php file is limited to only the most common actions used when manually attempting SQL injection, but not to the entire injection itself. For example; if a malicious hacker was to use a number greater than 9, a letter other than b, or even use negative numbers or capital letters, then they can successfully circumvent this filter. Furthermore, there is the use of automated tools which can be used to exploit this weakness as well.

3.2.11 Hidden guessable folder

Folders that contain potentially sensitive information that are required to be accessible from the website are most susceptible to being a target if a malicious hacker is able to identify it through the use of predetermined wordlists.

Through one such list it was possible for the tester to be able to find multiple 'hidden' sections of the website that did not have a link on the menu bar on the main page of the website. Some examples of these hidden sections would be the CYBERDOC25 folder, phpinfo.php file, and the admin login page.

3.3 COUNTERMEASURES

3.3.1 Weak or unenforced username policy

The best way to remove this vulnerability would be to set a strong username policy that will allows users to create/get unique usernames, as well as see how strong their username is.

3.3.2 Weak or unenforced password policy

A good method in order to mitigate this vulnerability would be to set a strong password policy that will allow the user to see the strength of the password that they are choosing. Furthermore, to have a minimum of 8 characters including special symbols, letters, and numbers.

3.3.3 Script injection (XSS)

One method to mitigate this vulnerability would be to remove all <script> tags as this will remove any and all vulnerable JavaScript. However, in any case where this is not ideal, it would otherwise be suggested to sanitize client inputted information.

3.3.4 Error handling

The best way to resolve this vulnerability would be to reduce the amount of information revealed in an error. For example, not to reveal any information regarding correct details entered into a login field. Moreover, not to leave any errors returned from the database open to the public, or to the directories located on the hosting machine (Figure 28 and Appendix F).

3.3.5 Brute-forceable Admin password

One method that would remove this risk would be to have the users set strong passwords – through the use of a strong password policy and showing the users how strong their passwords are.

Furthermore, this risk can be mitigated through the implementation of a lockout. After a certain number of failed attempts to get into a user's account, there should be a temporary lockout before the user can try again. The use of a counting variable for failed attempts and if else statements of the successful/unsuccessful login is one way of achieving this. A temporary ban for the IP address related to the ongoing failed attempts may also be a good method of mitigated the vulnerability.

3.3.6 Robots.txt

Some better uses for the robots.txt that have been considered would be to have the disallowed crawl be parts of the website that are not public facing, such as the admin page.

A recommended solution to this would be to remove the 'doornumbers.txt' file from the webpage and to remove the disallowed section from robots.txt or change the robots.txt disallowed to something that does not contain any form of sensitive and important information.

3.3.7 Local File Inclusion

In order to mitigate the Local file inclusion (LFI) vulnerability noted within the source code, beyond the filters that have been applied, would be to use strong input validation. The web server should restrict

the pages that can be shown to a whitelist and present an error code if a user attempts to go anywhere else.

For example; to have an if statement that allows for a user to only transverse through pages that are on a pre-written whitelist, e.g. index.php, login.php, home-page.html, etc. Then, if a user attempts to look for a page outside of this list, an error can be returned. This will stop a malicious hacker from being able to active an uploaded php file that would be stored within the 'pictures' directory.

Another way to help reduce the risk of LFI would be to have the php code run on the latest version of the php server, and to ensure that 'register_globals' are not being used (Chandel, 2021).

3.3.8 Hidden source code

The tester would recommend that these comments be removed ASAP, especially the phpinfo.php file which contains a lot of sensitive information regarding the web server, and made a note of (e.g. physically) in order to make sure they are not forgotten.

3.3.9 Reversible cookie

In order to mitigate the vulnerability of someone being able to gain access to someone else's account by spoofing their cookie would be to use more complex hashing algorithms. For example instead of using "md5", currently outdated and highly recommended to be not used, to hash the password but a newer hashing algorithm such as; bcrypt (Figure 36), Argon2i (Figure 37), or PHP password_hash default password hashing algorithm (Figure 38) (How to use bcrypt for hashing passwords in PHP? - GeeksforGeeks, 2021).

```
$options = [
    'cost' => 12,
];
$pass=password_hash(($_POST[Password]), PASSWORD_BCRYPT, $options);
```

Figure 36 – BCRYPT password hash

```
$pass=password_hash(($_POST[Password]), PASSWORD_ARGON2I);
```

Figure 37 – Argon2i password hash

```
$pass=password_hash(($_POST[Password]), PASSWORD_DEFAULT);
```

Figure 38 – Default password hash

The above mentioned hash types are newer and harder to reverse, so even if an attacker was able to get someone else's cookie, they most likely would be unable to gain that user's password.

Furthermore, another method could be to remove the sensitive data stored in the cookie (user credentials) to something else to identify the user without giving away important information about them.

3.3.10 Cookie attributes

Within a php script the HTTPOnly flag can be found in one of two places; permanently within the php.ini where it is possible to turn on the HTTPOnly flag (Figure 39) or within a function (Figure 40).

```
session.cookie_httponly = True
```

Figure 39 – HTTPOnly flag php.ini

```
void session_set_cookie_params ( int $lifetime [, string $path [, string $domain [, bool $secure= false [, bool $httponly= false ]]]] )
```

Figure 40 – HTTPOnly flag in a function

3.3.11 Directory browsing

The best way to mitigate this potential risk would be to disable directory browsing in the website configurations file.

3.3.12 Unlimited login attempts.

See section 3.3.2.

3.3.13 No HTTPS

To mitigate this vulnerability would be to run the webserver on HTTPS instead of HTTP, in order to encrypt the connection and remove the chances of a malicious hacker gaining information and potentially tampering with data of a user.

3.3.14 File upload

A method of removing this vulnerability would be to make the upload location of the file non-public, as this will not allow a malicious hacker to 'activate' a reverse shell or traverse to it. Furthermore, to reduce the likelihood of a malicious hacker uploading a malicious piece of code with an altered extension would be to use secure plug-ins that will help in preventing file upload vulnerabilities (Banu and Banu, 2021).

3.3.15 Cross Site Request Forgery (CSRF)

A method of reducing this vulnerability would be to check for the IP source of the request and ensure that the origins match the process request. Furthermore, there could be a re-authentication process to ensure that it is indeed the users requesting a password change.

3.3.16 SQL Injections

An excellent method of mitigating this vulnerability would be to use prepared statements. This will allow for the pre-written commands to be bound to the variable, and remove the ability of user inputted data affecting the command and leading to unintended access to user accounts without permission.

3.3.17 Php information disclosure

As was seen in with the hidden comments, this is a section on the website that should be removed.

3.3.18 Hidden guessable folder

A method of reducing this risk would be to use folder and file names that cannot be brute forced through the use of common word lists such as rockyou.txt. This would involve producing unique folder/file names that the appropriate users will be aware of (e.g. admin), so that they have access to the information that is relevant to them, and decreasing the likeliness of a malicious hacker obtaining it.

3.4 General Discussion

The aim of the penetration test was to conduct a series of tests in order to assess the security level of the website. These tests went to show that there are security holes within the website that can reveal unnecessary information to an attacker and can be abused to allow for certain actions to be taken, for example uploading reverse shell php code, that would damage the website or provide further sensitive information regarding the company and/or the users. This supports Dr. Dumbledore's concern that the website had some issues, as it is susceptible to many common attacks. Given the limited amount of time to work on the website, it is possible that some vulnerabilities were missed. It is also possible that through the use of some automated tools, to scan the website in search for vulnerabilities, that some vulnerabilities may not have been found/noticed.

The tests and documentations that have been reported will allow for the website developers of Hogwarts Management Systems to be able to locate and mitigate the vulnerabilities.

3.5 Future Work

If more time was available, it would have been possible to do more in-depth testing, for example testing for the 'shellshock' vulnerability that was flagged during a much later scan. Furthermore, a variety of other tools could have been used, in order to compare results and see if there may have been further vulnerabilities that one scanning tool was not able to locate.

One such tool that could have been used would be Acunetix. This tool could have been used to compare results with Nessus results. Furthermore, the tool Vega could have been used in order to further check if any of the previous tools used provided false positives, and also to get further information regarding any security settings for TLS/SSL (Vega Vulnerability Scanner, 2021).

4 CONCLUSION

4.1 CONCLUSION

In conclusion, many vulnerabilities were found and documented in the report to the client (Hogwarts Management Systems). As per the aim of this report, there are recommendations of methods to reduce the risk that these vulnerabilities pose and their impact on the company and users.

If used in its current state, ignoring all vulnerabilities or being left unattended, there would be a high risk of a malicious hacker successfully attacking the website and obtaining sensitive information and potentially doing damage to the website itself and the databases connected to it. Therefore, it is highly recommended that the website receive the appropriate attention in attending to the vulnerabilities and increasing the security and reliability of the website.

REFERENCES PART 1

For URLs, Blogs:

Owasp.org. 2020. *OWASP Web Security Testing Guide*. [online] Available at: <https://owasp.org/www-project-web-security-testing-guide/> [Accessed 14 December 2020].

Tutorialspoint.com. 2020. *Web Application Testing - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/software_testing_dictionary/web_application_testing.htm> [Accessed 14 December 2020].

Relevant Software. 2020. *Guide To Web Application Penetration Testing*. [online] Available at: <https://relevant.software/blog/penetration-testing-for-web-applications/#Why_Is_Penetration_Testing_Important> [Accessed 14 December 2020].

Tutorialspoint.com. 2020. *Web Application Testing - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/software_testing_dictionary/web_application_testing.htm> [Accessed 14 December 2020].

GitHub. 2020. *Tanprathan/OWASP-Testing-Checklist*. [online] Available at: <https://github.com/tanprathan/OWASP-Testing-Checklist> [Accessed 14 December 2020].

Kinsbruner, E., 2020. *Manual Testing Vs. Automated Testing | By Perforce*. [online] Perfecto by Perforce. Available at: <https://www.perfecto.io/blog/automated-testing-vs-manual-testing-vs-continuous-testing#:~:text=There%20are%20some%20major%20differences,with%20other%20tools%20and%20software.> [Accessed 14 December 2020].

Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/information-gathering/nikto> [Accessed 14 December 2020].

Cs.cmu.edu. 2020. *Nessus*. [online] Available at: <https://www.cs.cmu.edu/~dwendlan/personal/nessus.html> [Accessed 14 December 2020].

SourceForge. 2020. *OWASP Mantra - Security Framework*. [online] Available at: <https://sourceforge.net/projects/getmantra/> [Accessed 14 December 2020].

KeyCDN. 2020. *What Is The Difference Between HTTP And HTTPS? - Keycdn*. [online] Available at: <https://www.keycdn.com/blog/difference-between-http-and-https> [Accessed 14 December 2020].

Kedrosky, E., Pam Sornson, J. and Clark, M., 2020. *Why A Strong Password Policy Is So Important For Your Wordpress Website - Security Boulevard*. [online] Security Boulevard. Available at: <https://securityboulevard.com/2020/09/why-a-strong-password-policy-is-so-important-for-your-wordpress-website/#:~:text=Implementing%20a%20strong%20password%20policy%20is%20so%20import

ant%20because%20it,their%20way%20into%20the%20account.> [Accessed 14 December 2020].

Sqlmap.org. 2020. *Sqlmap: Automatic SQL Injection And Database Takeover Tool*. [online] Available at: <http://sqlmap.org/> [Accessed 14 December 2020].

Academy, W. and scripting, C., 2020. *What Is Reflected XSS (Cross-Site Scripting)? Tutorial & Examples | Web Security Academy*. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/cross-site-scripting/reflected> [Accessed 14 December 2020].

Owasp.org. 2020. *Improper Error Handling | OWASP*. [online] Available at: <https://owasp.org/www-community/Improper_Error_Handling> [Accessed 14 December 2020].

REFERENCES PART 2

Us-cert.cisa.gov. 2021. *Source Code Analysis Tools - Overview | CISA*. [online] Available at: <https://us-cert.cisa.gov/bsi/articles/tools/source-code-analysis/source-code-analysis-tools---overview> [Accessed 12 January 2021].

Owasp.org. 2021. *Httponly - Set-Cookie HTTP Response Header | OWASP*. [online] Available at: <https://owasp.org/www-community/HttpOnly> [Accessed 12 January 2021].

Center, S., Definitions, I. and listing, D., 2021. *Directory Listing*. [online] Portswigger.net. Available at: <https://portswigger.net/kb/issues/00600100_directory-listing#:~:text=Directory%20listings%20themselves%20do%20not%20necessarily%20constitute%20a,the%20location%20of%20sensitive%20files%20using%20automated%20tools.> [Accessed 12 January 2021].

Banting, K., 2021. *0777 Permissions Security Risk - What You Need To Know... - Business-In-Site*. [online] Business-in-site.com. Available at: <http://www.business-in-site.com/webmaster-articles/0777-permissions-security-risk-know/> [Accessed 12 January 2021].

Chandel, R., 2021. *Comprehensive Guide On Local File Inclusion (LFI)*. [online] Hacking Articles. Available at: <https://www.hackingarticles.in/comprehensive-guide-to-local-file-inclusion/> [Accessed 12 January 2021].

GeeksforGeeks. 2021. *How To Use Bcrypt For Hashing Passwords In PHP? - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/how-to-use-bcrypt-for-hashing-passwords-in-php/#:~:text=The%20bcrypt%20is%20a%20password%20hashing%20technique%20used,It%20uses%20a%20strong%20&%20robust%20hashing%20algorithm.> [Accessed 12 January 2021].

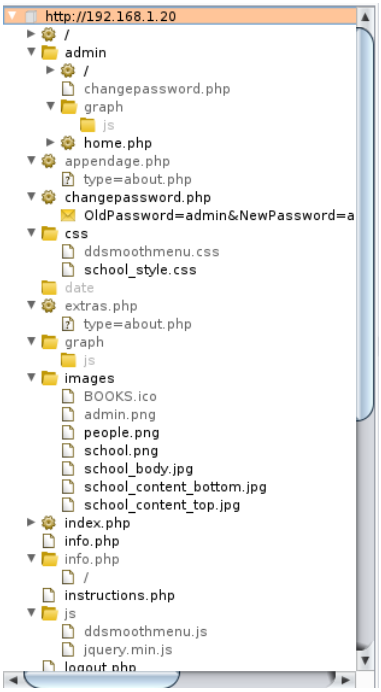
Banu, S. and Banu, S., 2021. *File Upload Vulnerability - How To Prevent Hackers From Exploiting It*. [online] MalCare. Available at: <https://www.malcare.com/blog/file-upload-vulnerability/#:~:text=%20How%20to%20Protect%20Your%20Website%20From%20File,the%20file%20upload%20function%20on%20your...%20More%20> [Accessed 12 January 2021].

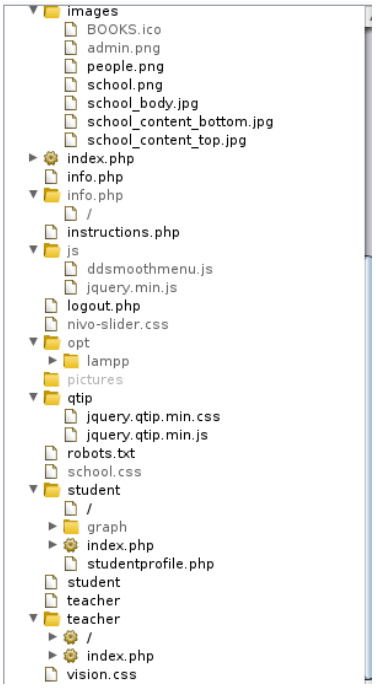
Subgraph.com. 2021. *Vega Vulnerability Scanner*. [online] Available at: <https://subgraph.com/vega/> [Accessed 12 January 2021].

APPENDICES PART 1

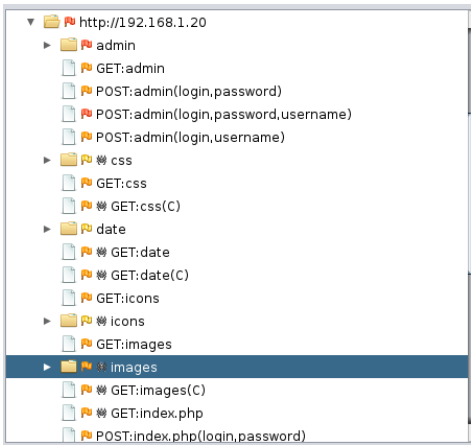
APPENDIX A – SITE MAP

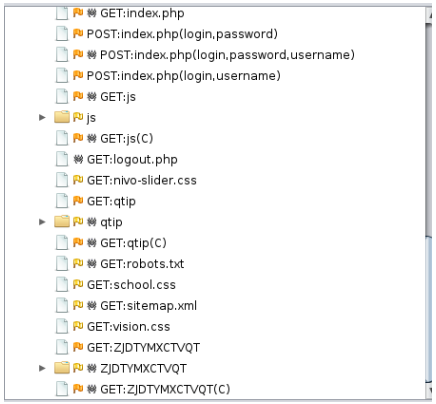
a) Site map – Burp suite





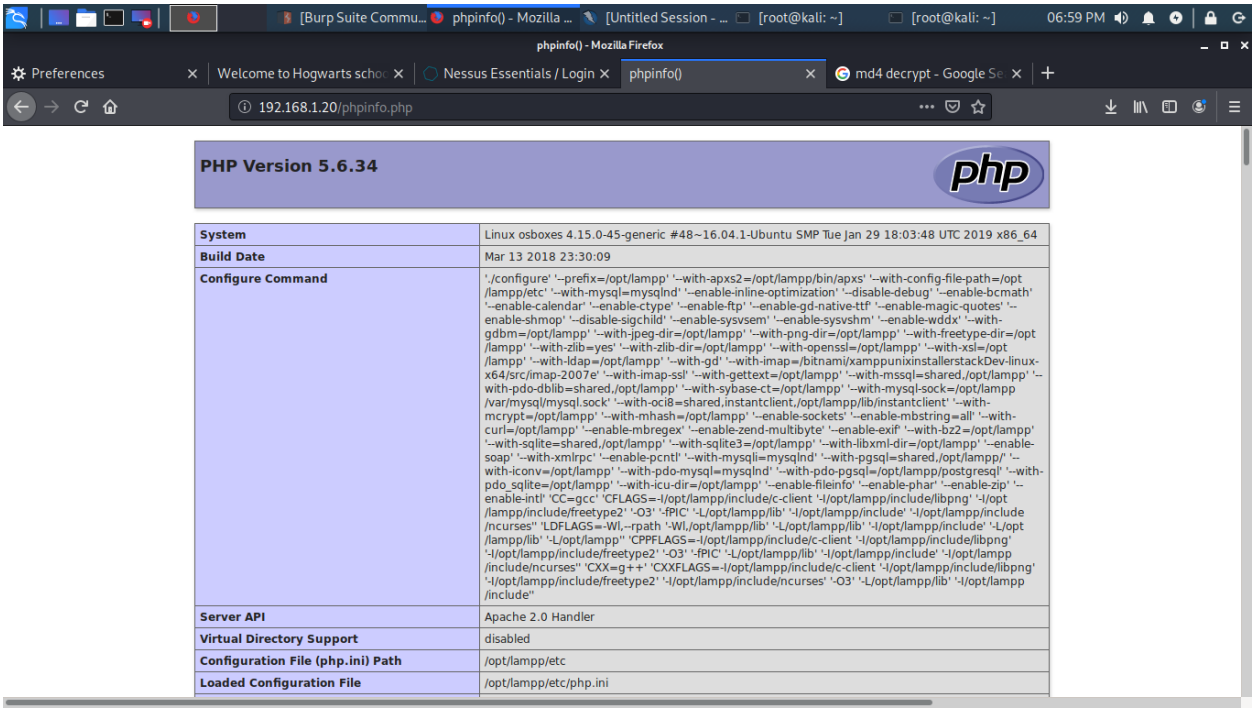
b) Site map – ZAP admin authenticated

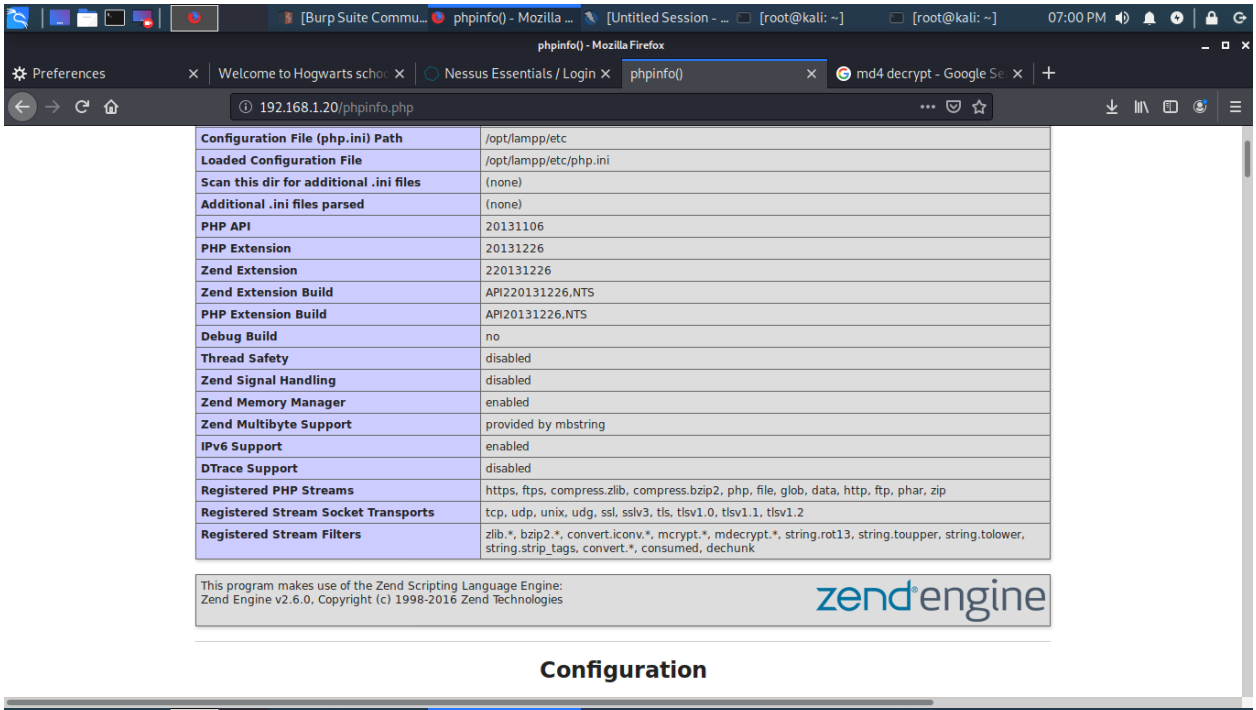




APPENDIX B – PHPINFO.PHP

a) Information about the website that was found at phpinfo.php



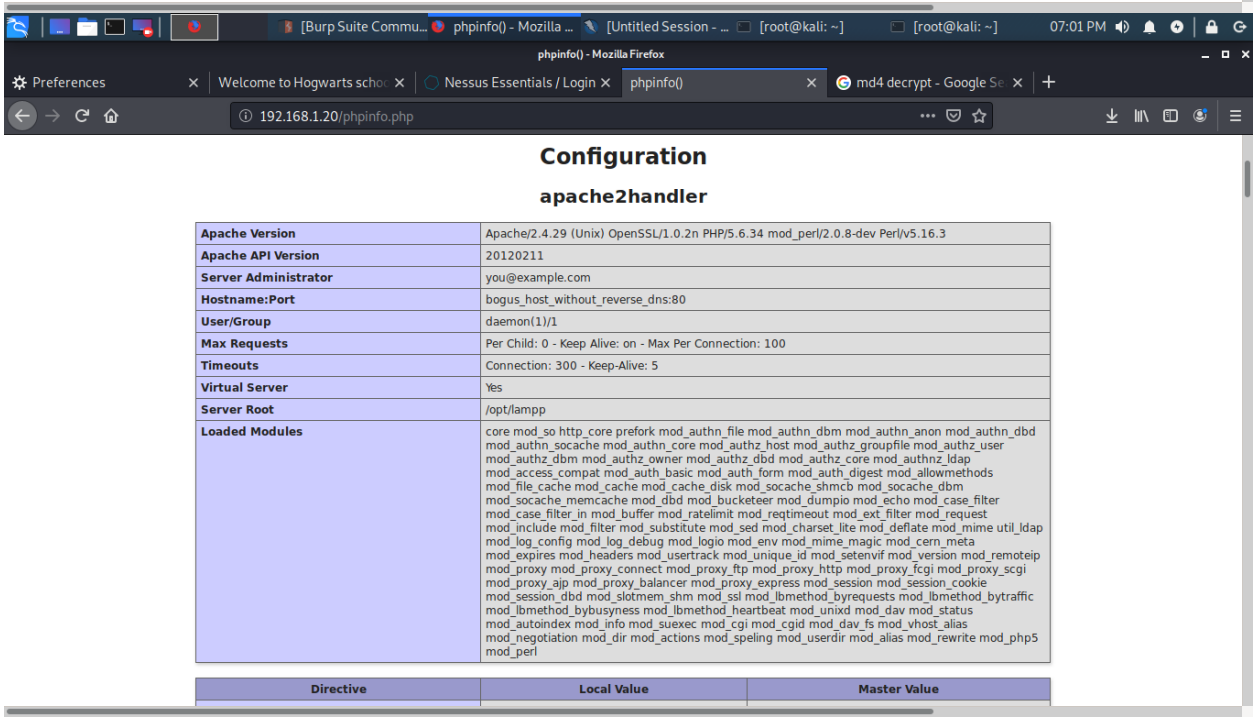


Configuration File (php.ini) Path	/opt/lampp/etc
Loaded Configuration File	/opt/lampp/etc/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, mcrypt.*, mdecrypt.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies

zend engine

Configuration

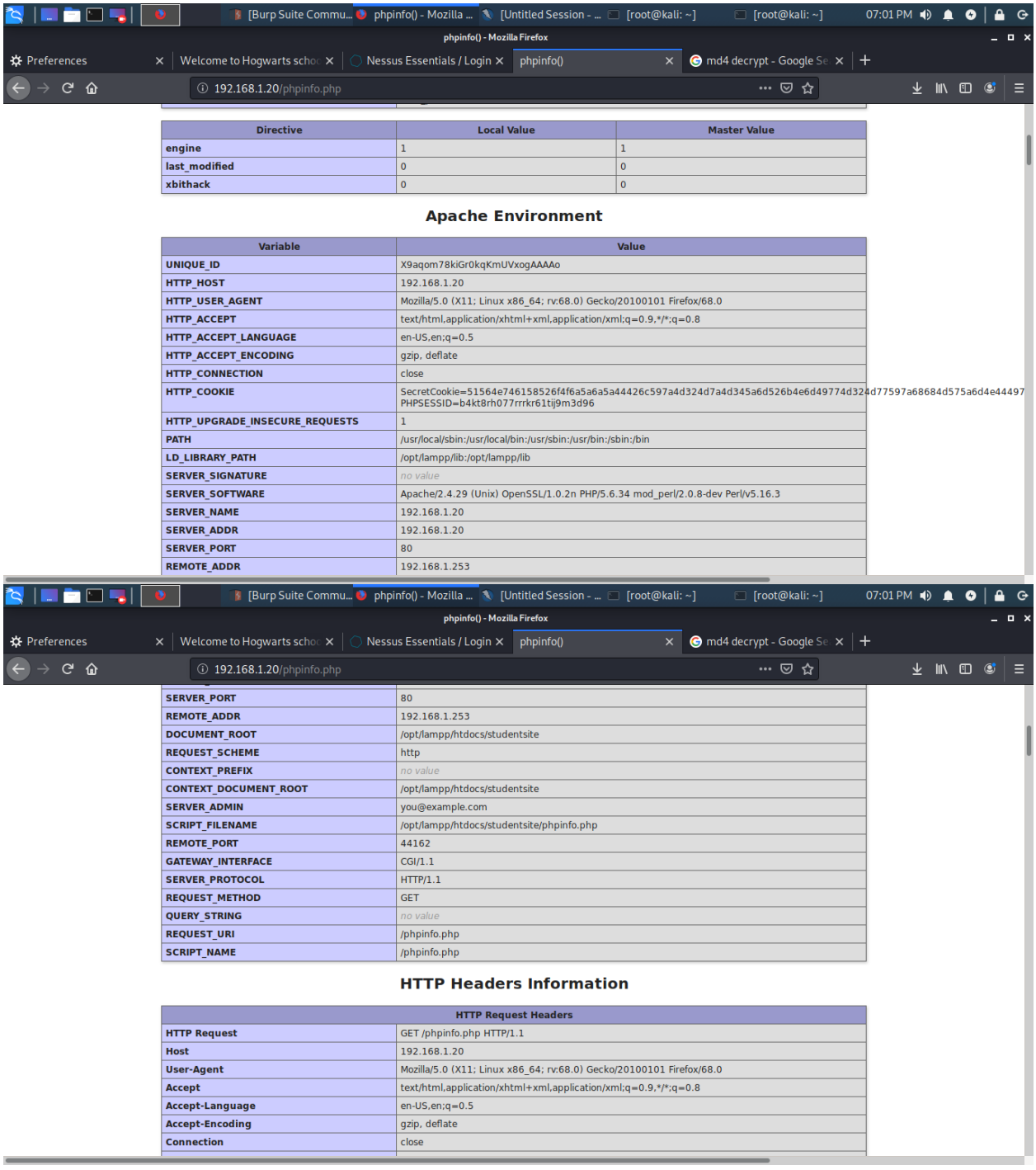


Configuration

apache2handler

Apache Version	Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
Apache API Version	20120211
Server Administrator	you@example.com
Hostname:Port	bogus_host_without_reverse_dns:80
User/Group	daemon(1)/1
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/opt/lampp
Loaded Modules	core mod_so http_core prefork mod_authn_file mod_authn_dbm mod_authn_anon mod_authn_dbd mod_authn_socache mod_authn_core mod_authz_host mod_authz_groupfile mod_authz_user mod_authz_dbm mod_authz_owner mod_authz_dbd mod_authz_core mod_authnz_ldap mod_access_compat mod_auth_basic mod_auth_form mod_auth_digest mod_allowmethods mod_file_cache mod_cache mod_cache_disk mod_socache_shmcb mod_socache_dbm mod_socache_memcache mod_dbd mod_bucketeer mod_dumpio mod_echo mod_case_filter mod_case_filter_in mod_buffer mod_ratelimit mod_reqtimeout mod_ext_filter mod_request mod_include mod_filter mod_substitute mod_sed mod_charset_lite mod_deflate mod_mime util_ldap mod_log_config mod_log_debug mod_logio mod_env mod_mime_magic mod_cern_meta mod_expires mod_headers mod_usertrack mod_unique_id mod_setenvif mod_version mod_remoteip mod_proxy mod_proxy_connect mod_proxy_ftp mod_proxy_http mod_proxy_fcgi mod_proxy_scgi mod_proxy_ajp mod_proxy_balancer mod_proxy_express mod_session mod_session_cookie mod_session_dbd mod_slotmem_shm mod_ssl mod_lbmethod_byrequests mod_lbmethod_bytraffic mod_lbmethod_bybusyness mod_lbmethod_heartbeat mod_unixd mod_dav mod_status mod_autoindex mod_info mod_suexec mod_cgi mod_cgid mod_dav_fs mod_vhost_alias mod_negotiation mod_dir mod_actions mod_speling mod_userdir mod_alias mod_rewrite mod_php5 mod_perl

Directive	Local Value	Master Value



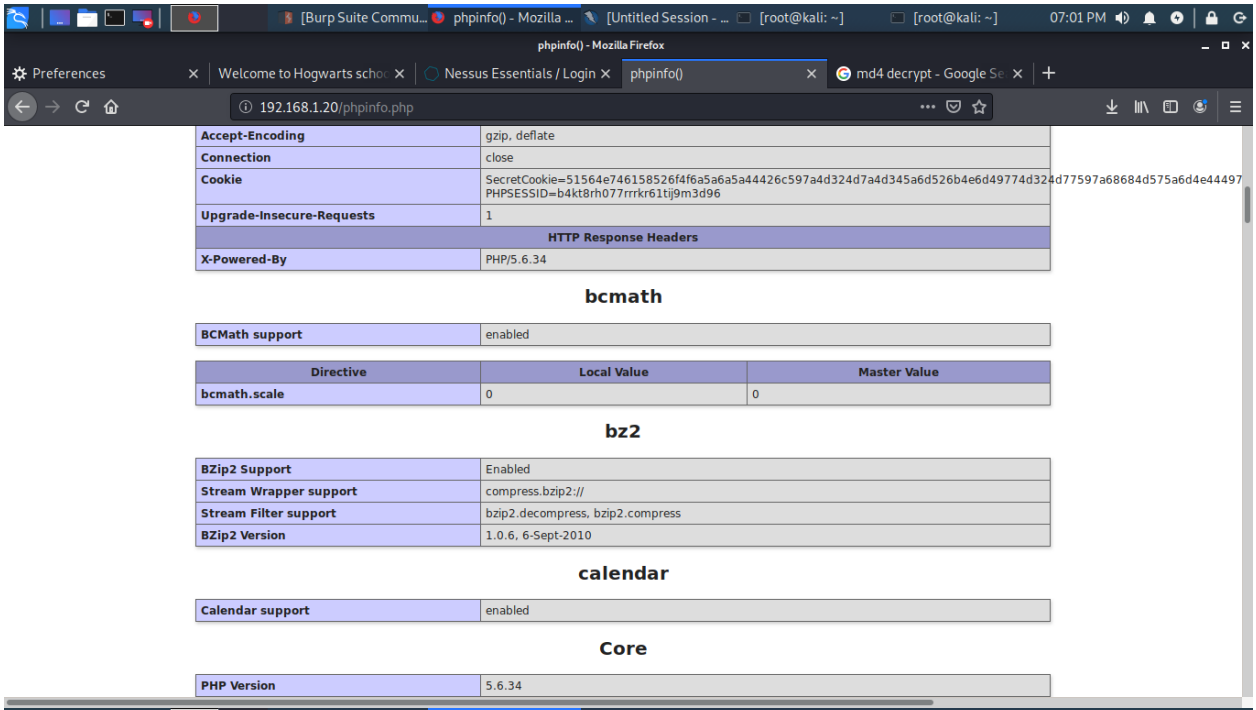
Directive	Local Value	Master Value
engine	1	1
last_modified	0	0
xbithack	0	0

Apache Environment

Variable	Value
UNIQUE_ID	X9aqom78kiGr0kqKmUVxogAAAAo
HTTP_HOST	192.168.1.20
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_CONNECTION	close
HTTP_COOKIE	SecretCookie=51564e746158526f4f6a5a6a5a44426c597a4d324d7a4d345a6d526b4e6d49774d324d77597a68684d575a6d4e44497 PHPSESSID=b4kt8rh077rrrkr61tij9m3d96
HTTP_UPGRADE_INSECURE_REQUESTS	1
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
LD_LIBRARY_PATH	/opt/lampp/lib:/opt/lampp/lib
SERVER_SIGNATURE	*no value*
SERVER_SOFTWARE	Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
SERVER_NAME	192.168.1.20
SERVER_ADDR	192.168.1.20
SERVER_PORT	80
REMOTE_ADDR	192.168.1.253
DOCUMENT_ROOT	/opt/lampp/htdocs/studentsite
REQUEST_SCHEME	http
CONTEXT_PREFIX	*no value*
CONTEXT_DOCUMENT_ROOT	/opt/lampp/htdocs/studentsite
SERVER_ADMIN	you@example.com
SCRIPT_FILENAME	/opt/lampp/htdocs/studentsite/phpinfo.php
REMOTE_PORT	44162
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	*no value*
REQUEST_URI	/phpinfo.php
SCRIPT_NAME	/phpinfo.php

HTTP Headers Information

HTTP Request Headers	
HTTP Request	GET /phpinfo.php HTTP/1.1
Host	192.168.1.20
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Connection	close



Accept-Encoding	gzip, deflate
Connection	close
Cookie	SecretCookie=51564e746158526f4f6a5a6a5a44426c597a4d324d7a4d345a6d526b4e6d49774d324d77597a68684d575a6d4e44497 PHPSESSID=b4kt8rh077rrrkr61tij9m3d96
Upgrade-Insecure-Requests	1
HTTP Response Headers	
X-Powered-By	PHP/5.6.34

bcmath

BCMath support	enabled

Directive	Local Value	Master Value
bcmath.scale	0	0

bz2

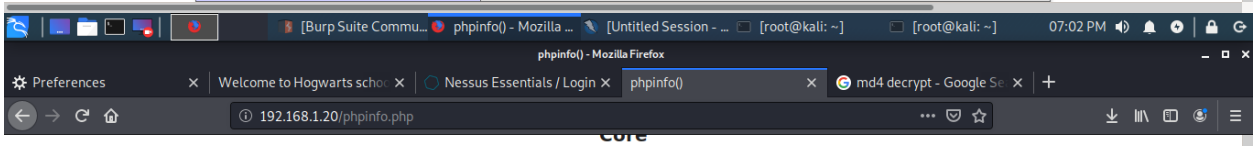
BZip2 Support	Enabled
Stream Wrapper support	compress.bzip2://
Stream Filter support	bzip2.decompress, bzip2.compress
BZip2 Version	1.0.6, 6-Sept-2010

calendar

Calendar support	enabled

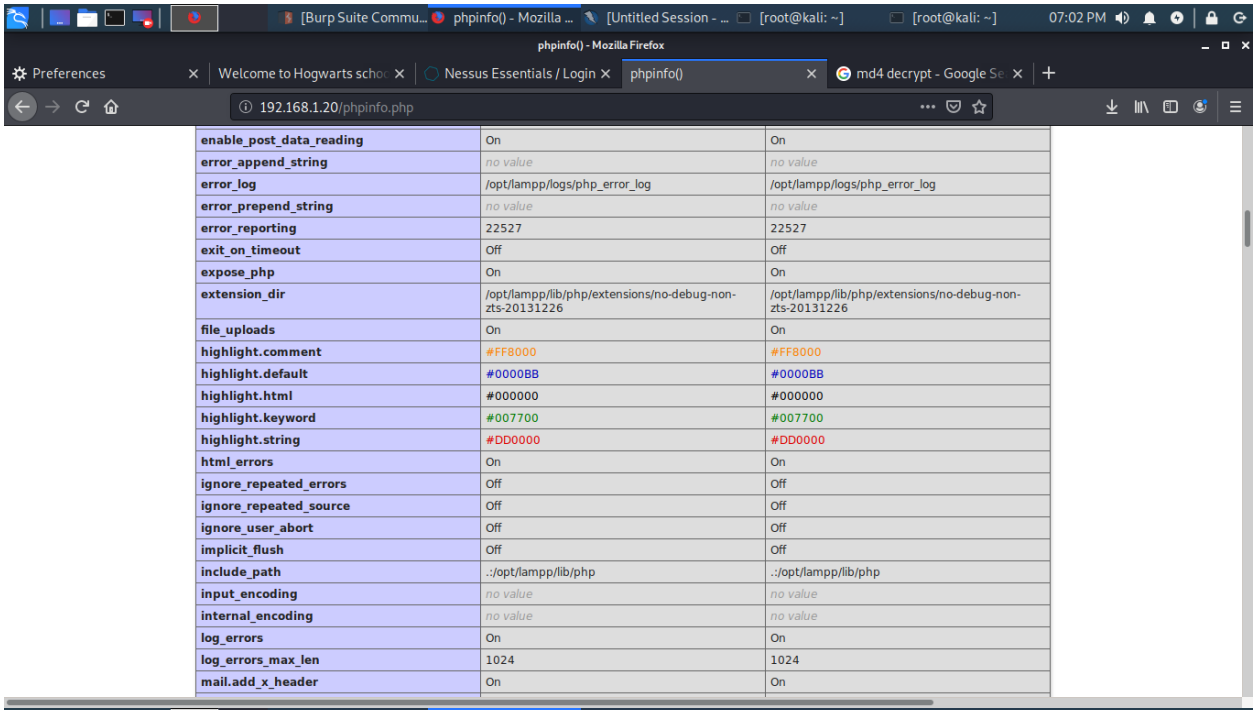
Core

PHP Version	5.6.34

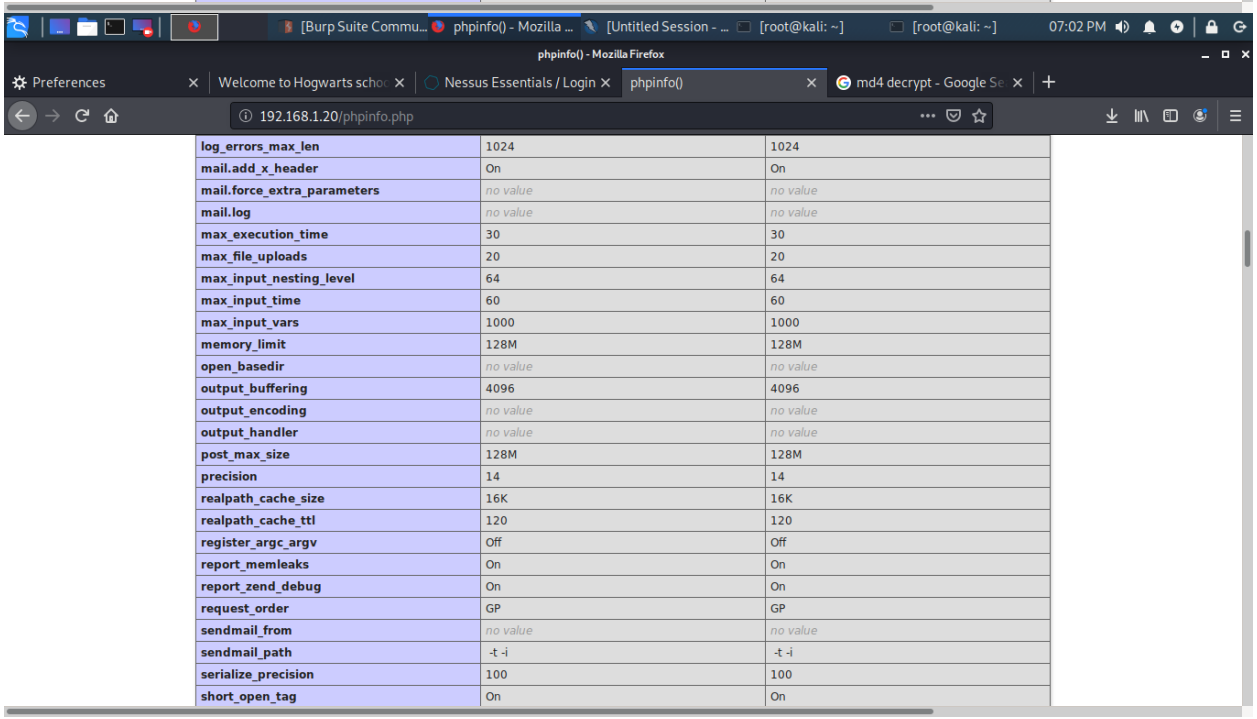


PHP Version	5.6.34

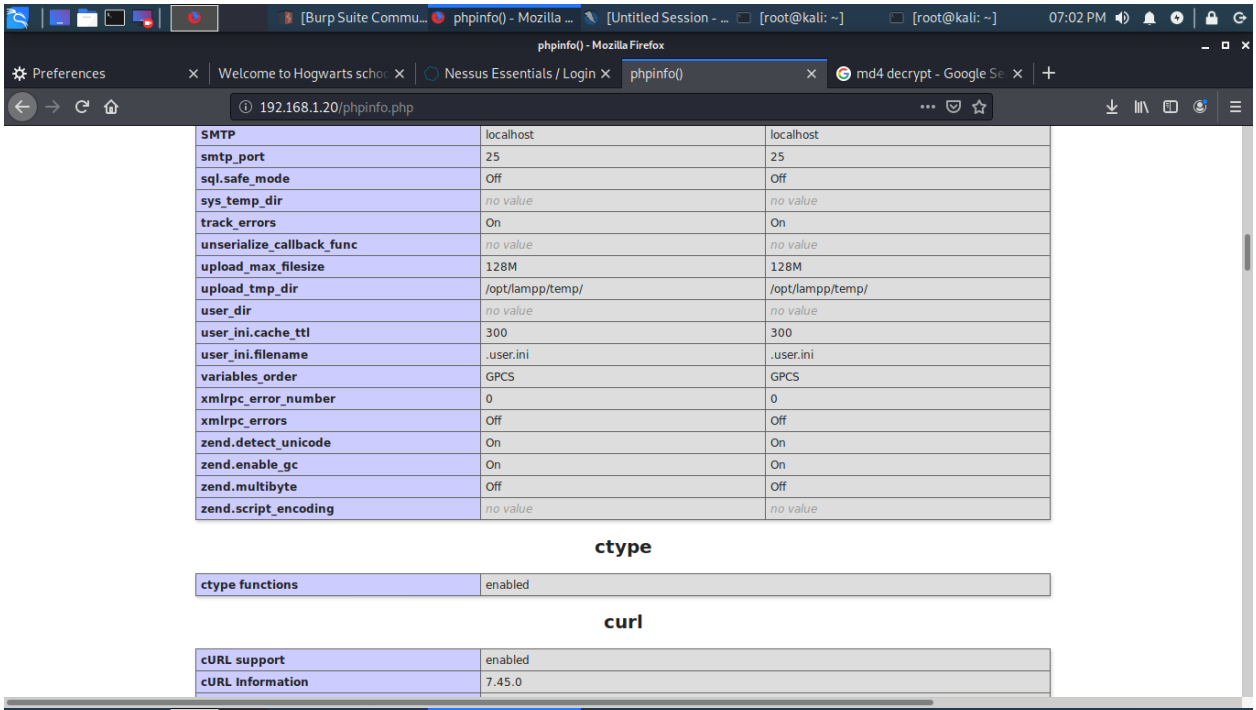
Directive	Local Value	Master Value
allow_url_fopen	On	On
allow_url_include	Off	Off
always_populate_raw_post_data	0	0
arg_separator.input	&	&
arg_separator.output	&	&
asp_tags	Off	Off
auto_append_file	*no value*	*no value*
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*
default_charset	UTF-8	UTF-8
default_mimetype	text/html	text/html
disable_classes	*no value*	*no value*
disable_functions	*no value*	*no value*
display_errors	On	On
display_startup_errors	On	On
doc_root	*no value*	*no value*
docref_ext	*no value*	*no value*
docref_root	*no value*	*no value*
enable_dl	Off	Off
enable_post_data_reading	On	On
error_append_string	*no value*	*no value*



enable_post_data_reading	On	On
error_append_string	no value	no value
error_log	/opt/lampp/logs/php_error_log	/opt/lampp/logs/php_error_log
error_prepend_string	no value	no value
error_reporting	22527	22527
exit_on_timeout	Off	Off
expose_php	On	On
extension_dir	/opt/lampp/lib/php/extensions/no-debug-non-zts-20131226	/opt/lampp/lib/php/extensions/no-debug-non-zts-20131226
file_uploads	On	On
highlight.comment	#FF8000	#FF8000
highlight.default	#0000BB	#0000BB
highlight.html	#000000	#000000
highlight.keyword	#007700	#007700
highlight.string	#DD0000	#DD0000
html_errors	On	On
ignore_repeated_errors	Off	Off
ignore_repeated_source	Off	Off
ignore_user_abort	Off	Off
implicit_flush	Off	Off
include_path	.:/opt/lampp/lib/php	.:/opt/lampp/lib/php
input_encoding	no value	no value
internal_encoding	no value	no value
log_errors	On	On
log_errors_max_len	1024	1024
mail.add_x_header	On	On



log_errors_max_len	1024	1024
mail.add_x_header	On	On
mail.force_extra_parameters	no value	no value
mail.log	no value	no value
max_execution_time	30	30
max_file_uploads	20	20
max_input_nesting_level	64	64
max_input_time	60	60
max_input_vars	1000	1000
memory_limit	128M	128M
open_basedir	no value	no value
output_buffering	4096	4096
output_encoding	no value	no value
output_handler	no value	no value
post_max_size	128M	128M
precision	14	14
realpath_cache_size	16K	16K
realpath_cache_ttl	120	120
register_argc_argv	Off	Off
report_memleaks	On	On
report_zend_debug	On	On
request_order	GP	GP
sendmail_from	no value	no value
sendmail_path	-t -i	-t -i
serialize_precision	100	100
short_open_tag	On	On



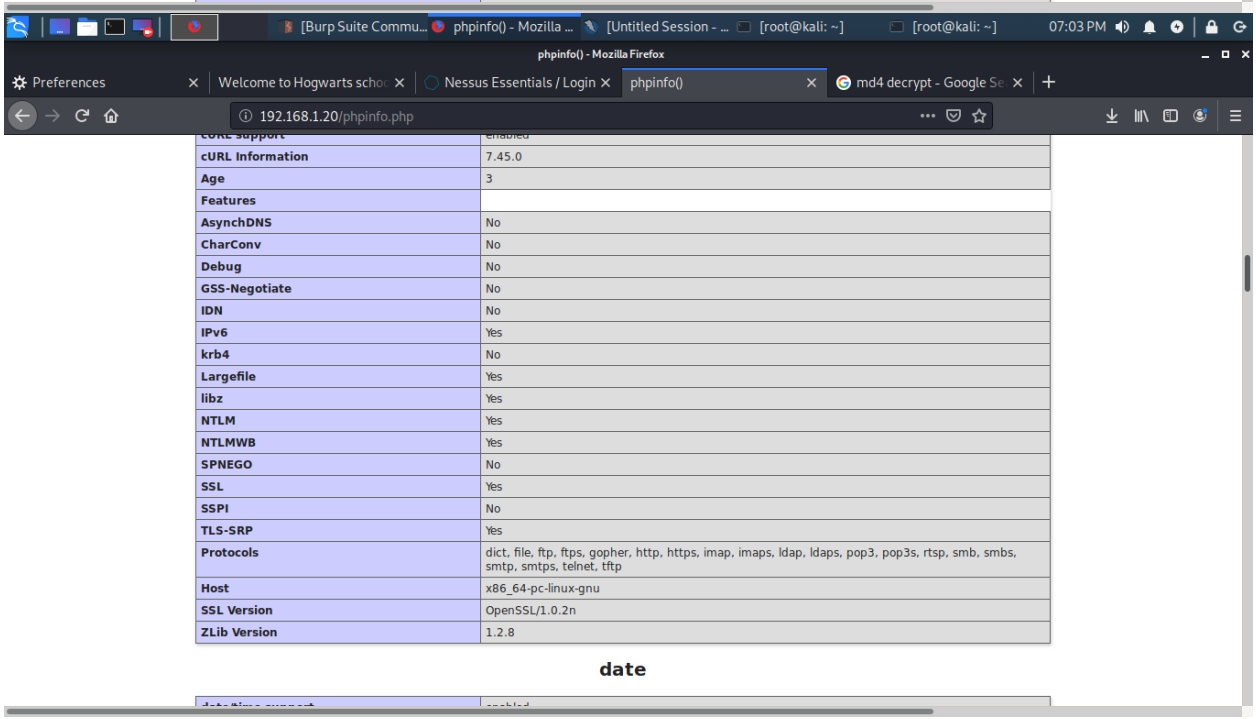
SMTP	localhost	localhost
smtp_port	25	25
sql.safe_mode	Off	Off
sys_temp_dir	*no value*	*no value*
track_errors	On	On
unserialize_callback_func	*no value*	*no value*
upload_max_filesize	128M	128M
upload_tmp_dir	/opt/lampp/temp/	/opt/lampp/temp/
user_dir	*no value*	*no value*
user_ini.cache_ttl	300	300
user_ini.filename	.user.ini	.user.ini
variables_order	GPCS	GPCS
xmlrpc_error_number	0	0
xmlrpc_errors	Off	Off
zend.detect_unicode	On	On
zend.enable_gc	On	On
zend.multibyte	Off	Off
zend.script_encoding	*no value*	*no value*

ctype

ctype functions	enabled

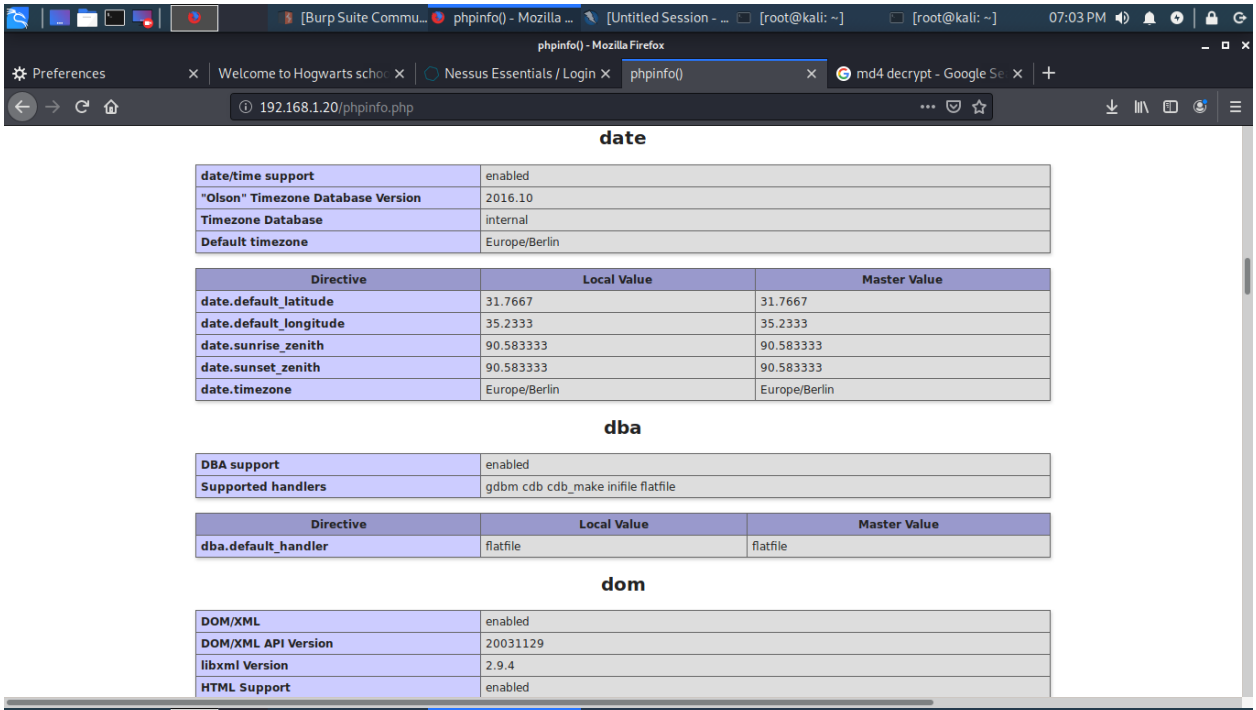
curl

cURL support	enabled
cURL Information	7.45.0



cURL Information	7.45.0
Age	3
Features	
AsynchDNS	No
CharConv	No
Debug	No
GSS-Negotiate	No
IDN	No
IPv6	Yes
krb4	No
Largefile	Yes
libz	Yes
NTLM	Yes
NTLMWB	Yes
SPNEGO	No
SSL	Yes
SSPI	No
TLS-SRP	Yes
Protocols	dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, ldaps, pop3, pop3s, rtsp, smb, smbs, smtp, smtps, telnet, tftp
Host	x86_64-pc-linux-gnu
SSL Version	OpenSSL/1.0.2n
ZLib Version	1.2.8

date



date

date/time support	enabled
"Olson" Timezone Database Version	2016.10
Timezone Database	internal
Default timezone	Europe/Berlin

Directive	Local Value	Master Value
date.default_latitude	31.7667	31.7667
date.default_longitude	35.2333	35.2333
date.sunrise_zenith	90.583333	90.583333
date.sunset_zenith	90.583333	90.583333
date.timezone	Europe/Berlin	Europe/Berlin

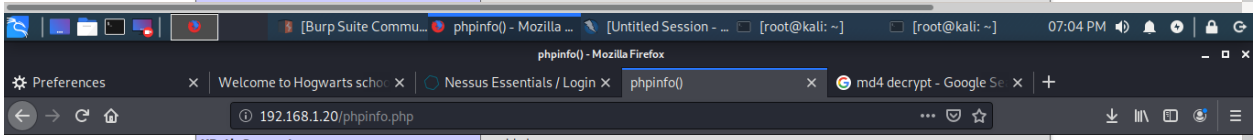
dba

DBA support	enabled
Supported handlers	gdbm cdb cdb_make inifile flatfile

Directive	Local Value	Master Value
dba.default_handler	flatfile	flatfile

dom

DOM/XML	enabled
DOM/XML API Version	20031129
libxml Version	2.9.4
HTML Support	enabled



XPath Support	enabled
XPointer Support	enabled
Schema Support	enabled
RelaxNG Support	enabled

ereg

Regex Library	Bundled library enabled

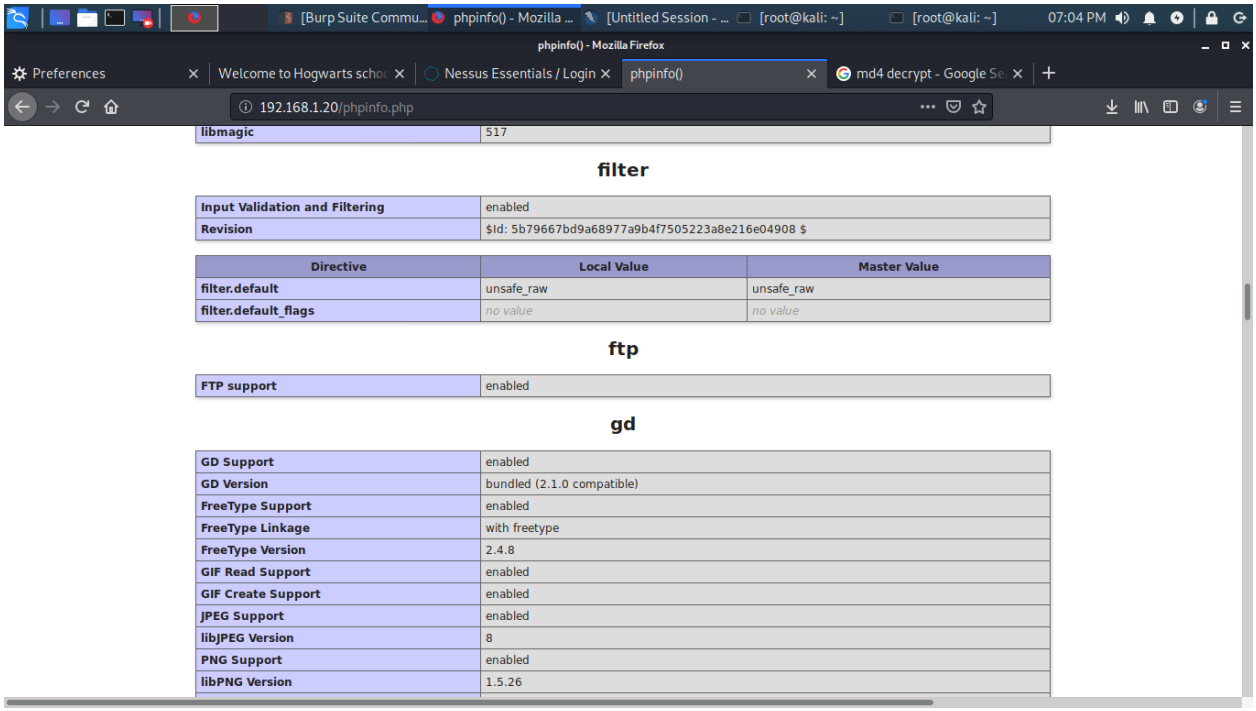
exif

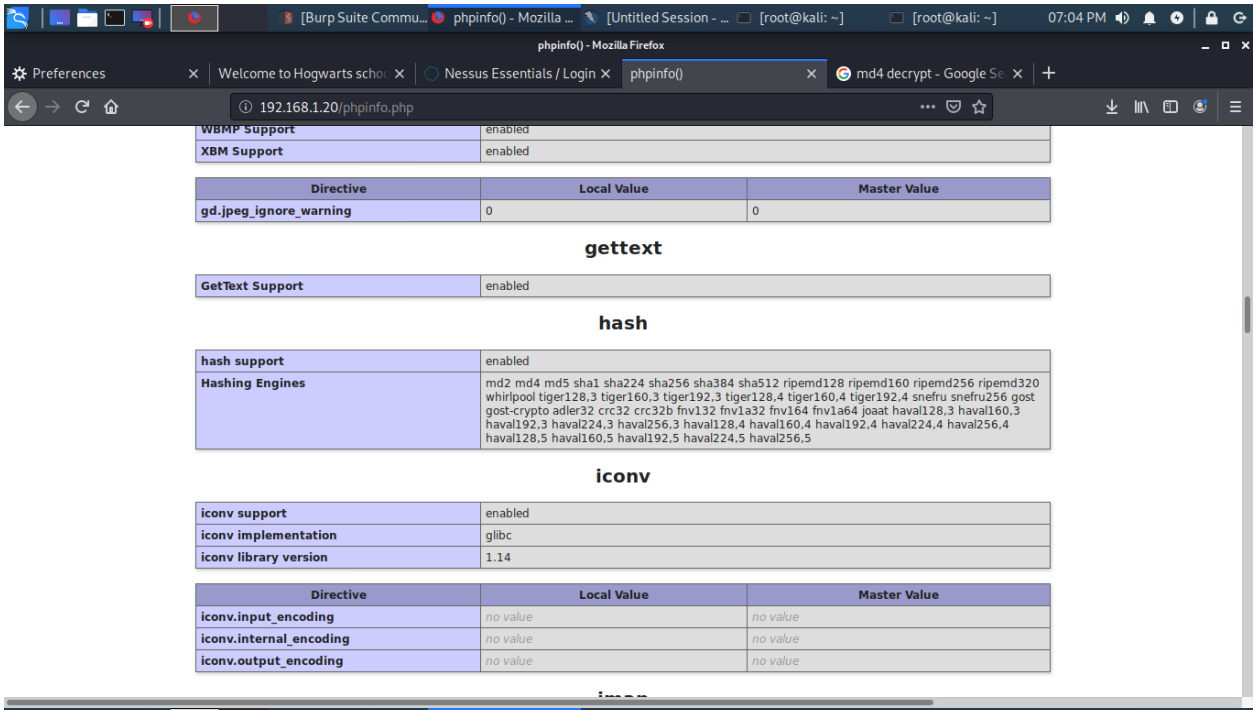
EXIF Support	enabled
EXIF Version	1.4 $Id: 1c8772f76be691b7b3f77ca31eb788a2abbcefe5 $
Supported EXIF Version	0220
Supported filetypes	JPEG,TIFF

Directive	Local Value	Master Value
exif.decode_jis_intel	JIS	JIS
exif.decode_jis_motorola	JIS	JIS
exif.decode_unicode_intel	UCS-2LE	UCS-2LE
exif.decode_unicode_motorola	UCS-2BE	UCS-2BE
exif.encode_jis	*no value*	*no value*
exif.encode_unicode	ISO-8859-15	ISO-8859-15

fileinfo

fileinfo support	enabled
version	1.0.5


[Burp Suite Commu...
phpinfo() - Mozilla ...
[Untitled Session - ...
[root@kali: ~]
[root@kali: ~]
07:04 PM
phpinfo() - Mozilla Firefox
Preferences
Welcome to Hogwarts scho
Nessus Essentials / Login
phpinfo()
md4 decrypt - Google Se
192.168.1.20/phpinfo.php
libmagic
517
filter
Input Validation and Filtering
enabled
Revision
$Id: 5b79667bd9a68977a9b4f7505223a8e216e04908 $
Directive
Local Value
Master Value
filter.default
unsafe_raw
unsafe_raw
filter.default_flags
no value
no value
ftp
FTP support
enabled
gd
GD Support
enabled
GD Version
bundled (2.1.0 compatible)
FreeType Support
enabled
FreeType Linkage
with freetype
FreeType Version
2.4.8
GIF Read Support
enabled
GIF Create Support
enabled
JPEG Support
enabled
libJPEG Version
8
PNG Support
enabled
libPNG Version
1.5.26



WBMP Support	enabled
XBM Support	enabled

Directive	Local Value	Master Value
gd.jpeg_ignore_warning	0	0

gettext

GetText Support	enabled

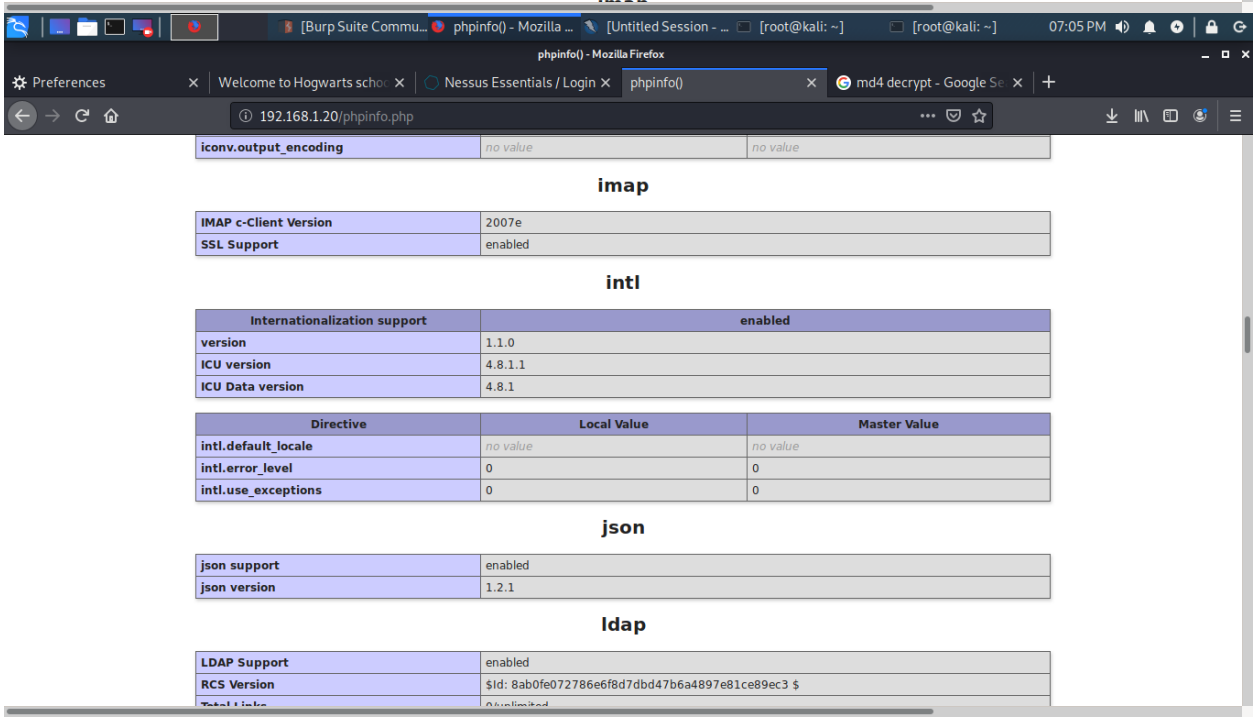
hash

hash support	enabled
Hashing Engines	md2 md4 md5 sha1 sha224 sha256 sha384 sha512 ripemd128 ripemd160 ripemd256 ripemd320 whirlpool tiger128,3 tiger160,3 tiger192,3 tiger128,4 tiger160,4 tiger192,4 snefru snefru256 gost gost-crypto adler32 crc32 crc32b fnv132 fnv1a32 fnv164 fnv1a64 joaat haval128,3 haval160,3 haval192,3 haval224,3 haval256,3 haval128,4 haval160,4 haval192,4 haval224,4 haval256,4 haval128,5 haval160,5 haval192,5 haval224,5 haval256,5

iconv

iconv support	enabled
iconv implementation	glibc
iconv library version	1.14

Directive	Local Value	Master Value
iconv.input_encoding	*no value*	*no value*
iconv.internal_encoding	*no value*	*no value*
iconv.output_encoding	*no value*	*no value*



iconv.output_encoding	*no value* *no value*

imap

IMAP c-Client Version	2007e
SSL Support	enabled

intl

Internationalization support	enabled
version	1.1.0
ICU version	4.8.1.1
ICU Data version	4.8.1

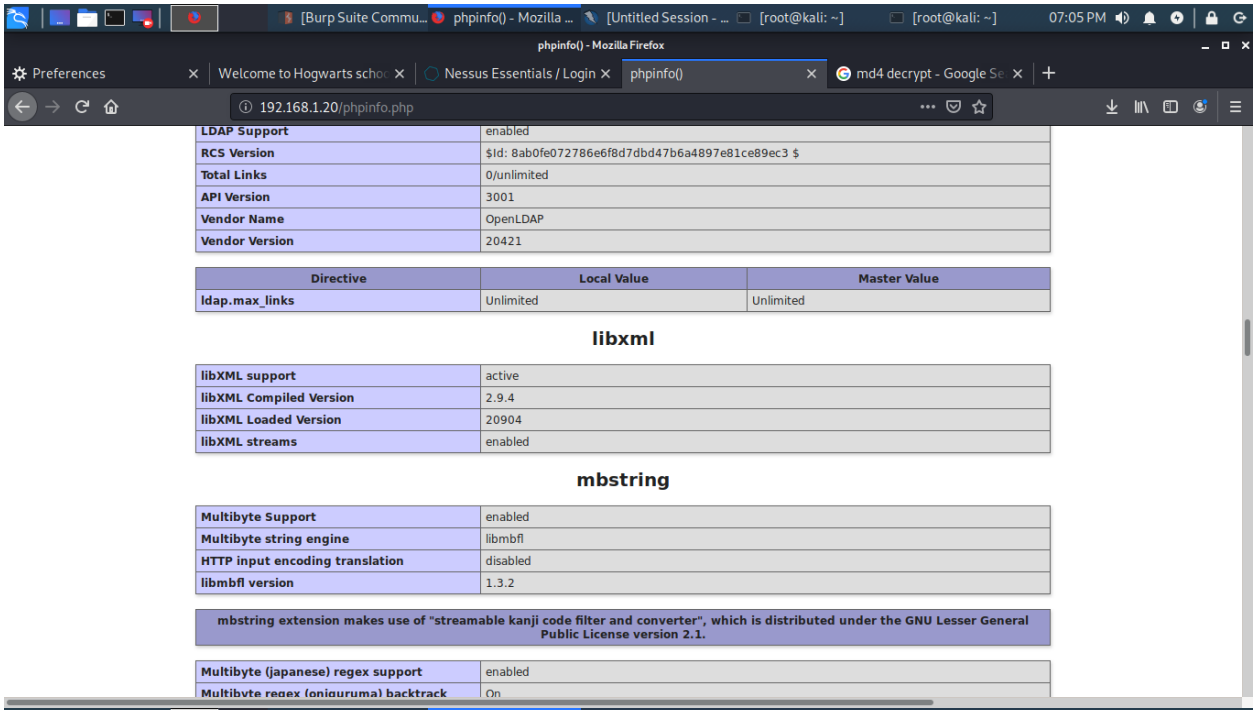
Directive	Local Value	Master Value
intl.default_locale	*no value*	*no value*
intl.error_level	0	0
intl.use_exceptions	0	0

json

json support	enabled
json version	1.2.1

ldap

LDAP Support	enabled
RCS Version	$Id: 8ab0fe072786e6f8d7dbd47b6a4897e81ce89ec3 $



LDAP Support	enabled
RCS Version	$Id: 8ab0fe072786e6f8d7dbd47b6a4897e81ce89ec3 $
Total Links	0/unlimited
API Version	3001
Vendor Name	OpenLDAP
Vendor Version	20421

Directive	Local Value	Master Value
ldap.max_links	Unlimited	Unlimited

libxml

libXML support	active
libXML Compiled Version	2.9.4
libXML Loaded Version	20904
libXML streams	enabled

mbstring

Multibyte Support	enabled
Multibyte string engine	libmbfl
HTTP input encoding translation	disabled
libmbfl version	1.3.2

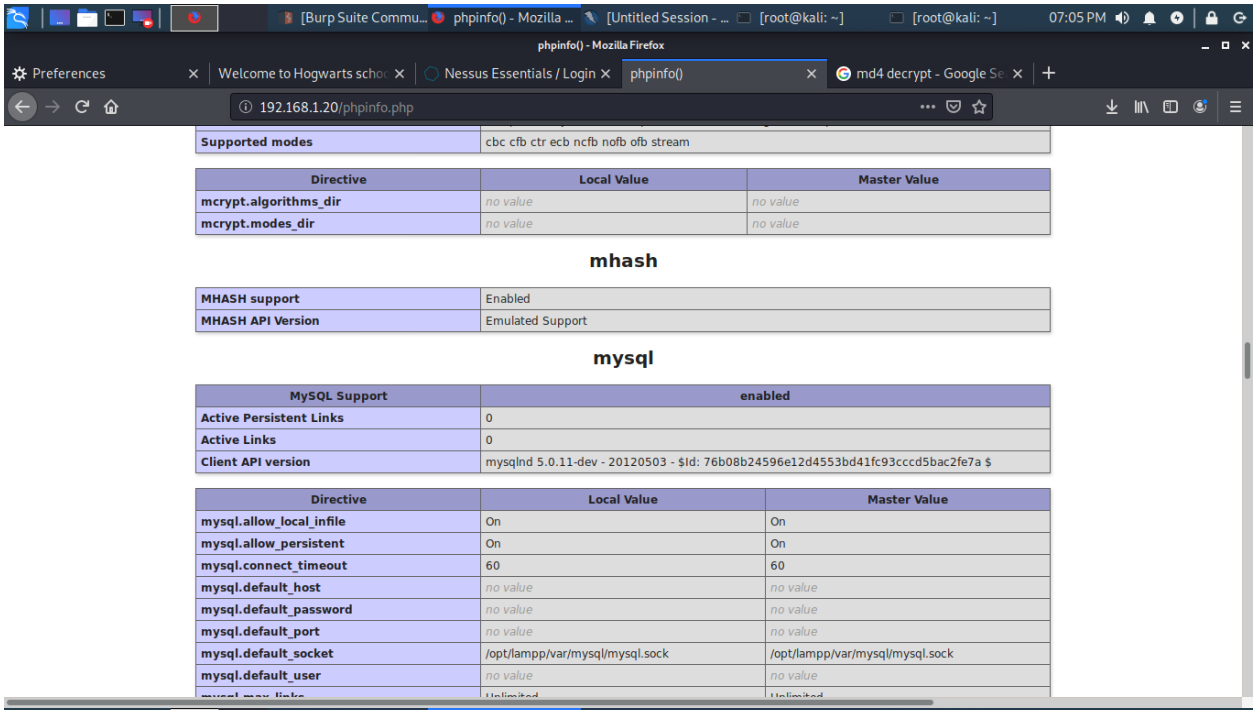
mbstring extension makes use of "streamable kanji code filter and converter", which is distributed under the GNU Lesser General Public License version 2.1.

Multibyte (japanese) regex support	enabled
Multibyte regex (oniguruma) backtrack check	On
Multibyte regex (oniguruma) version	5.9.5

Directive	Local Value	Master Value
mbstring.detect_order	*no value*	*no value*
mbstring.encoding_translation	Off	Off
mbstring.func_overload	0	0
mbstring.http_input	*no value*	*no value*
mbstring.http_output	*no value*	*no value*
mbstring.http_output_conv_mimetypes	^(text/|application/xhtml\+xml)	^(text/|application/xhtml\+xml)
mbstring.internal_encoding	*no value*	*no value*
mbstring.language	neutral	neutral
mbstring.strict_detection	Off	Off
mbstring.substitute_character	*no value*	*no value*

mcrypt

mcrypt support	**enabled**
mcrypt_filter support	**enabled**
Version	2.5.8
Api No	20021217
Supported ciphers	cast-128 gost rijndael-128 twofish arcfour cast-256 loki97 rijndael-192 saferplus wake blowfish-compat des rijndael-256 serpent xtea blowfish enigma rc2 tripledes
Supported modes	cbc cfb ctr ecb ncfb nofb ofb stream



Supported modes	cbc cfb ctr ecb ncfb nofb ofb stream

Directive	Local Value	Master Value
mcrypt.algorithms_dir	*no value*	*no value*
mcrypt.modes_dir	*no value*	*no value*

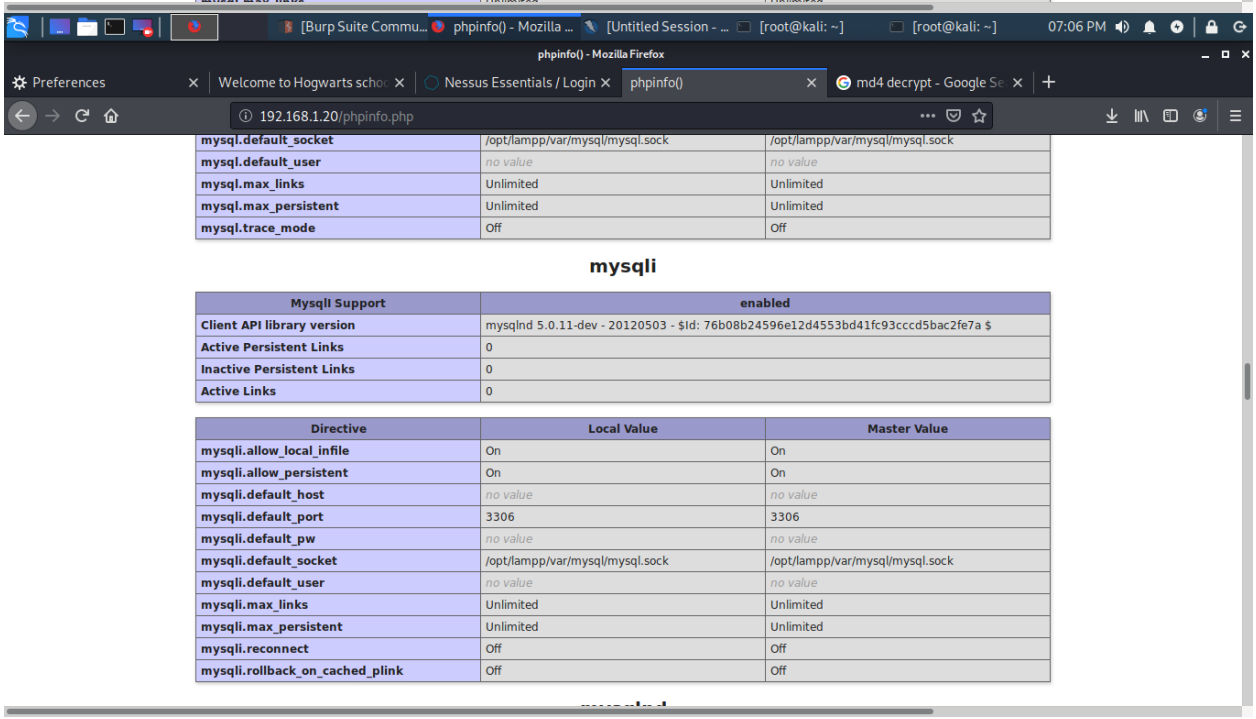
mhash

MHASH support	Enabled
MHASH API Version	Emulated Support

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	*no value*	*no value*
mysql.default_password	*no value*	*no value*
mysql.default_port	*no value*	*no value*
mysql.default_socket	/opt/lampp/var/mysql/mysql.sock	/opt/lampp/var/mysql/mysql.sock
mysql.default_user	*no value*	*no value*

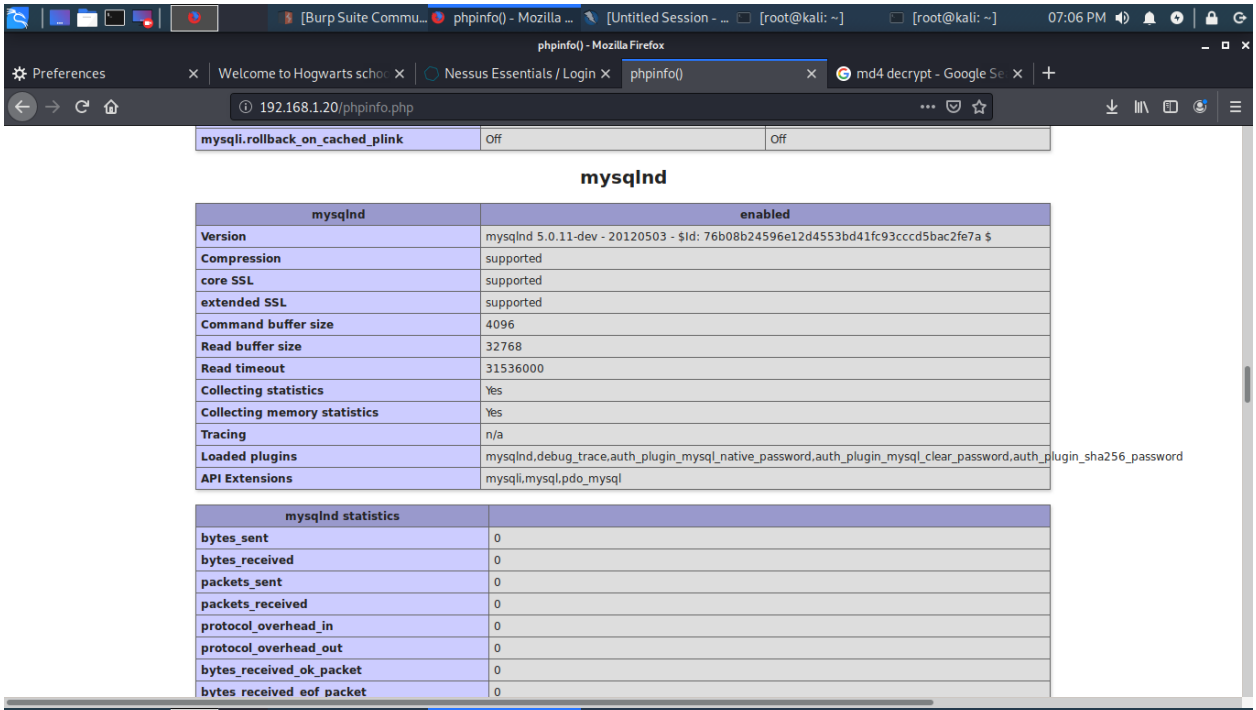


mysql.default_socket	/opt/lampp/var/mysql/mysql.sock	/opt/lampp/var/mysql/mysql.sock
mysql.default_user	*no value*	*no value*
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	/opt/lampp/var/mysql/mysql.sock	/opt/lampp/var/mysql/mysql.sock
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off
mysqli.rollback_on_cached_plink	Off	Off

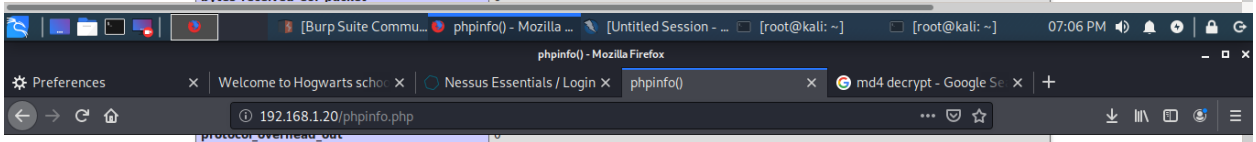


mysqli.rollback_on_cached_plink	Off	Off

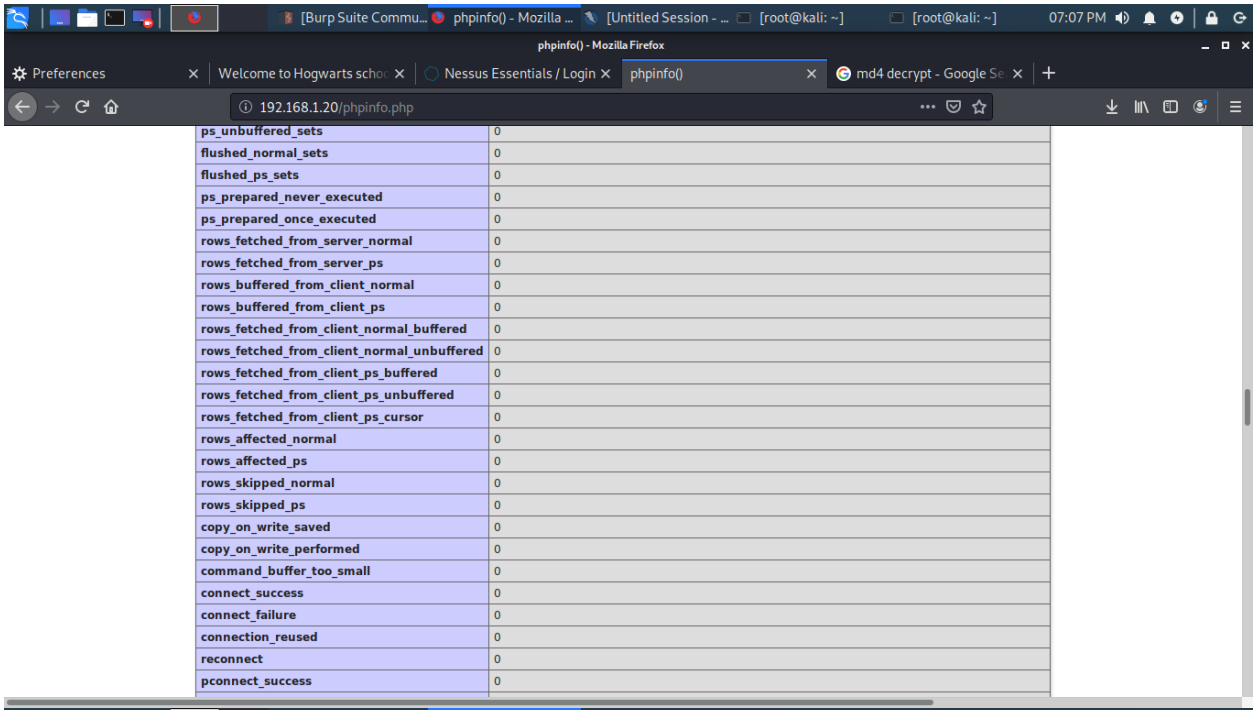
mysqlnd

mysqlnd	enabled
Version	mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $
Compression	supported
core SSL	supported
extended SSL	supported
Command buffer size	4096
Read buffer size	32768
Read timeout	31536000
Collecting statistics	Yes
Collecting memory statistics	Yes
Tracing	n/a
Loaded plugins	mysqlnd,debug_trace,auth_plugin_mysql_native_password,auth_plugin_mysql_clear_password,auth_plugin_sha256_password
API Extensions	mysqli,mysql,pdo_mysql

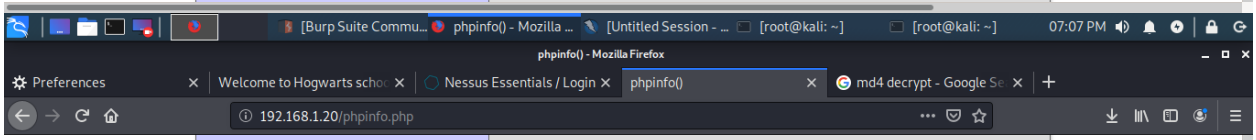
mysqlnd statistics	
bytes_sent	0
bytes_received	0
packets_sent	0
packets_received	0
protocol_overhead_in	0
protocol_overhead_out	0
bytes_received_ok_packet	0
bytes_received_eof_packet	0



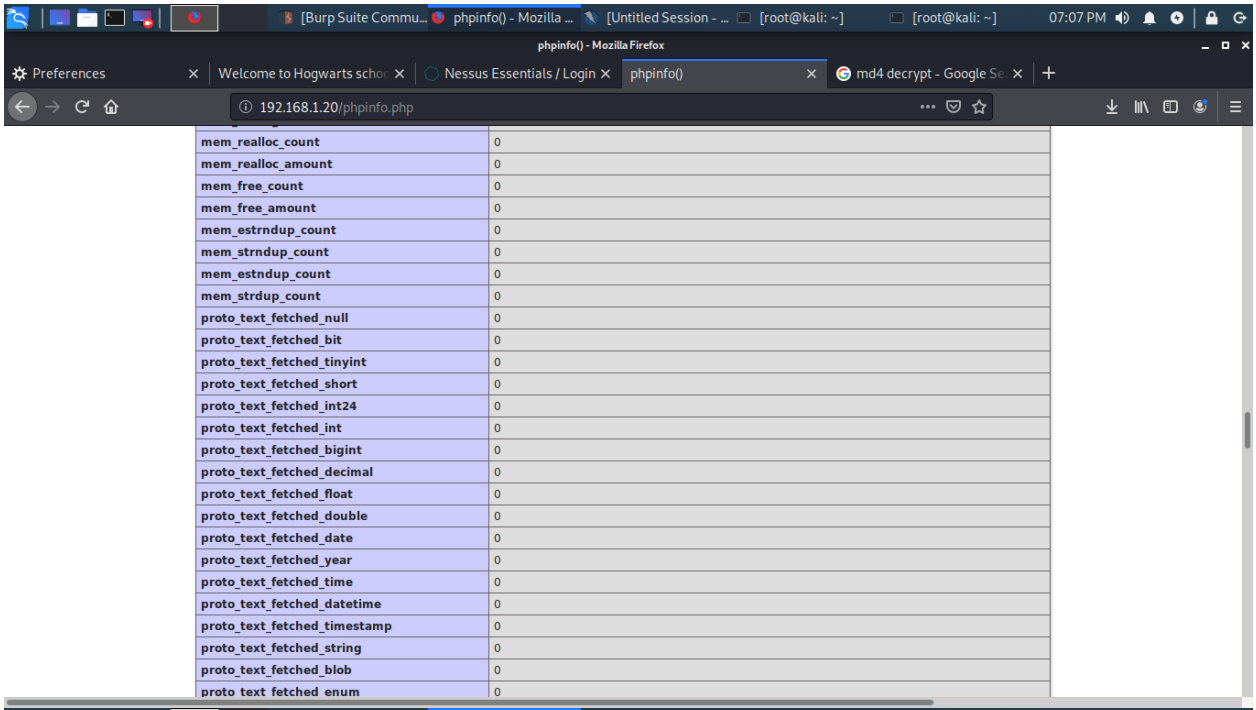
bytes_received_ok_packet	0
bytes_received_eof_packet	0
bytes_received_rset_header_packet	0
bytes_received_rset_field_meta_packet	0
bytes_received_rset_row_packet	0
bytes_received_prepare_response_packet	0
bytes_received_change_user_packet	0
packets_sent_command	0
packets_received_ok	0
packets_received_eof	0
packets_received_rset_header	0
packets_received_rset_field_meta	0
packets_received_rset_row	0
packets_received_prepare_response	0
packets_received_change_user	0
result_set_queries	0
non_result_set_queries	0
no_index_used	0
bad_index_used	0
slow_queries	0
buffered_sets	0
unbuffered_sets	0
ps_buffered_sets	0
ps_unbuffered_sets	0
flushed_normal_sets	0



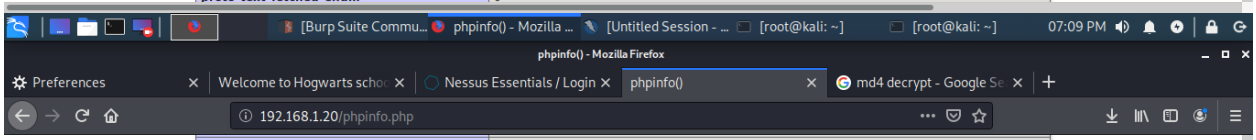
ps_unbuffered_sets	0
flushed_normal_sets	0
flushed_ps_sets	0
ps_prepared_never_executed	0
ps_prepared_once_executed	0
rows_fetched_from_server_normal	0
rows_fetched_from_server_ps	0
rows_buffered_from_client_normal	0
rows_buffered_from_client_ps	0
rows_fetched_from_client_normal_buffered	0
rows_fetched_from_client_normal_unbuffered	0
rows_fetched_from_client_ps_buffered	0
rows_fetched_from_client_ps_unbuffered	0
rows_fetched_from_client_ps_cursor	0
rows_affected_normal	0
rows_affected_ps	0
rows_skipped_normal	0
rows_skipped_ps	0
copy_on_write_saved	0
copy_on_write_performed	0
command_buffer_too_small	0
connect_success	0
connect_failure	0
connection_reused	0
reconnect	0
pconnect_success	0



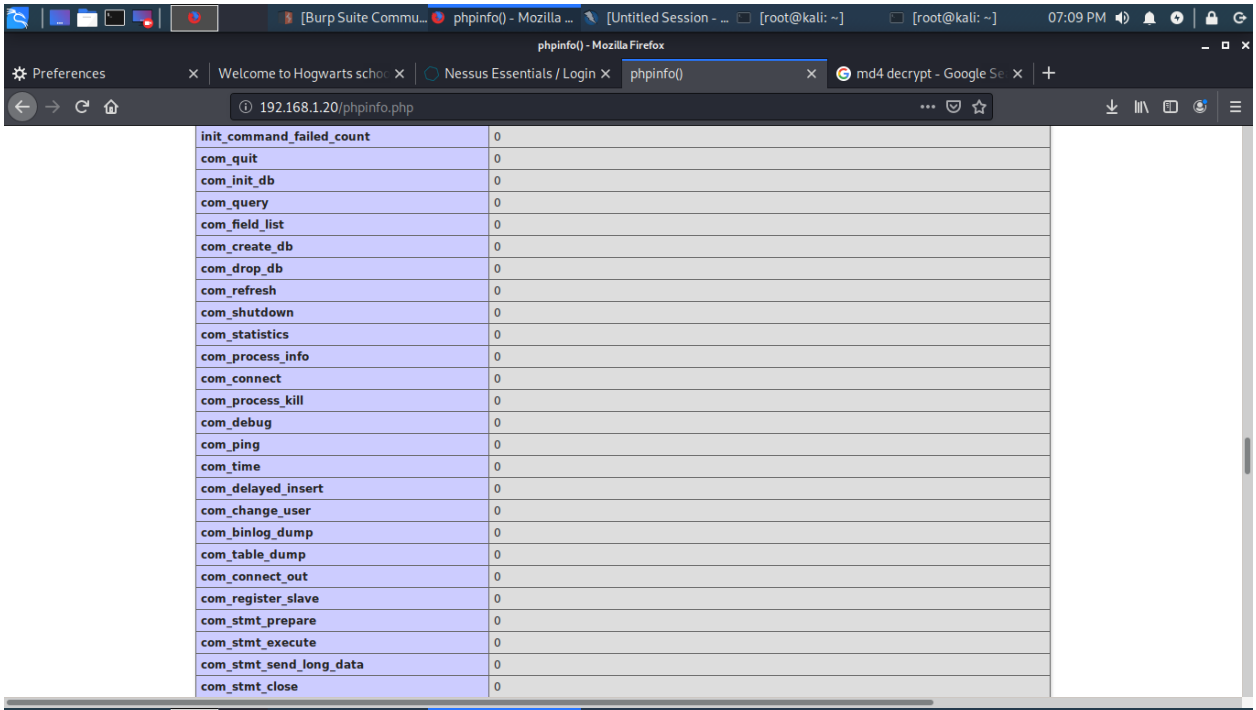
reconnect	0
pconnect_success	0
active_connections	0
active_persistent_connections	0
explicit_close	0
implicit_close	0
disconnect_close	0
in_middle_of_command_close	0
explicit_free_result	0
implicit_free_result	0
explicit_stmt_close	0
implicit_stmt_close	0
mem_emalloc_count	0
mem_emalloc_amount	0
mem_ecalloc_count	0
mem_ecalloc_amount	0
mem_erealloc_count	0
mem_erealloc_amount	0
mem_efree_count	0
mem_efree_amount	0
mem_malloc_count	0
mem_malloc_amount	0
mem_calloc_count	0
mem_calloc_amount	0
mem_realloc_count	0
mem_realloc_amount	0



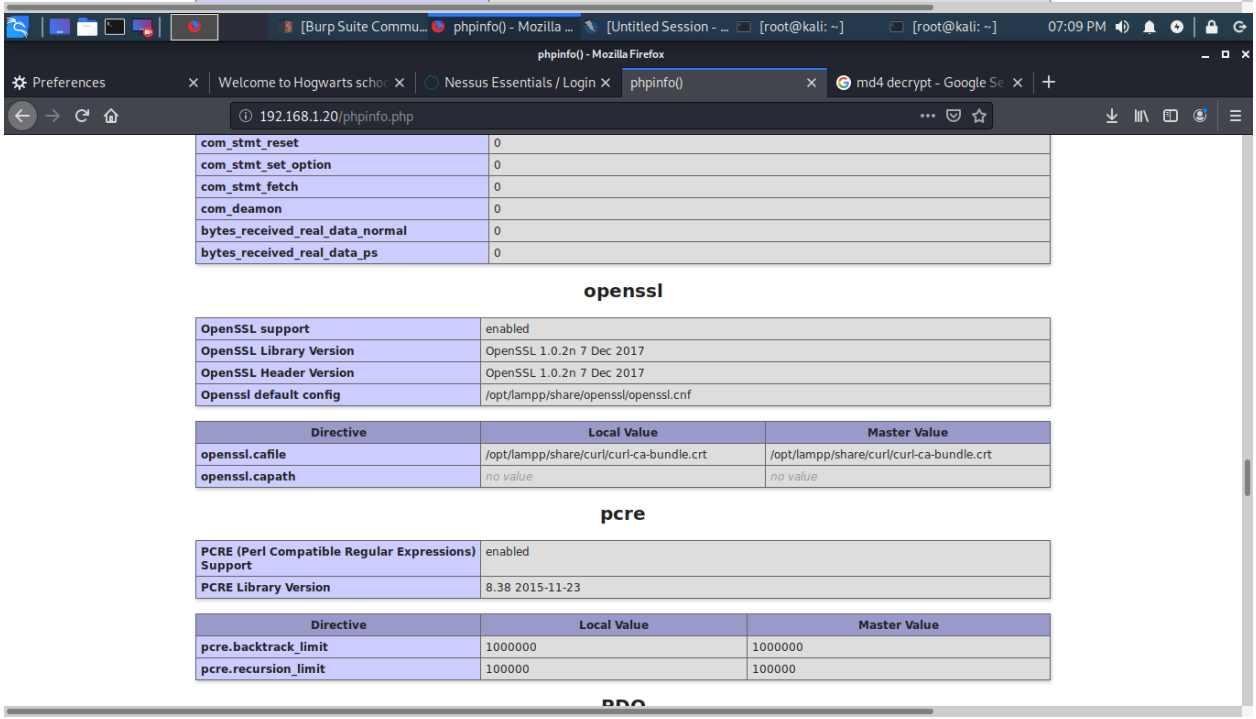
mem_realloc_count	0
mem_realloc_amount	0
mem_free_count	0
mem_free_amount	0
mem_estrndup_count	0
mem_strndup_count	0
mem_estndup_count	0
mem_strdup_count	0
proto_text_fetched_null	0
proto_text_fetched_bit	0
proto_text_fetched_tinyint	0
proto_text_fetched_short	0
proto_text_fetched_int24	0
proto_text_fetched_int	0
proto_text_fetched_bigint	0
proto_text_fetched_decimal	0
proto_text_fetched_float	0
proto_text_fetched_double	0
proto_text_fetched_date	0
proto_text_fetched_year	0
proto_text_fetched_time	0
proto_text_fetched_datetime	0
proto_text_fetched_timestamp	0
proto_text_fetched_string	0
proto_text_fetched_blob	0
proto_text_fetched_enum	0



proto_text_fetched_set	0
proto_text_fetched_geometry	0
proto_text_fetched_other	0
proto_binary_fetched_null	0
proto_binary_fetched_bit	0
proto_binary_fetched_tinyint	0
proto_binary_fetched_short	0
proto_binary_fetched_int24	0
proto_binary_fetched_int	0
proto_binary_fetched_bigint	0
proto_binary_fetched_decimal	0
proto_binary_fetched_float	0
proto_binary_fetched_double	0
proto_binary_fetched_date	0
proto_binary_fetched_year	0
proto_binary_fetched_time	0
proto_binary_fetched_datetime	0
proto_binary_fetched_timestamp	0
proto_binary_fetched_string	0
proto_binary_fetched_json	0
proto_binary_fetched_blob	0
proto_binary_fetched_enum	0
proto_binary_fetched_set	0
proto_binary_fetched_geometry	0
proto_binary_fetched_other	0
init_command_executed_count	0



init_command_failed_count	0
com_quit	0
com_init_db	0
com_query	0
com_field_list	0
com_create_db	0
com_drop_db	0
com_refresh	0
com_shutdown	0
com_statistics	0
com_process_info	0
com_connect	0
com_process_kill	0
com_debug	0
com_ping	0
com_time	0
com_delayed_insert	0
com_change_user	0
com_binlog_dump	0
com_table_dump	0
com_connect_out	0
com_register_slave	0
com_stmt_prepare	0
com_stmt_execute	0
com_stmt_send_long_data	0
com_stmt_close	0



com_stmt_reset	0
com_stmt_set_option	0
com_stmt_fetch	0
com_deamon	0
bytes_received_real_data_normal	0
bytes_received_real_data_ps	0

openssl

OpenSSL support	enabled
OpenSSL Library Version	OpenSSL 1.0.2n 7 Dec 2017
OpenSSL Header Version	OpenSSL 1.0.2n 7 Dec 2017
Openssl default config	/opt/lampp/share/openssl/openssl.cnf

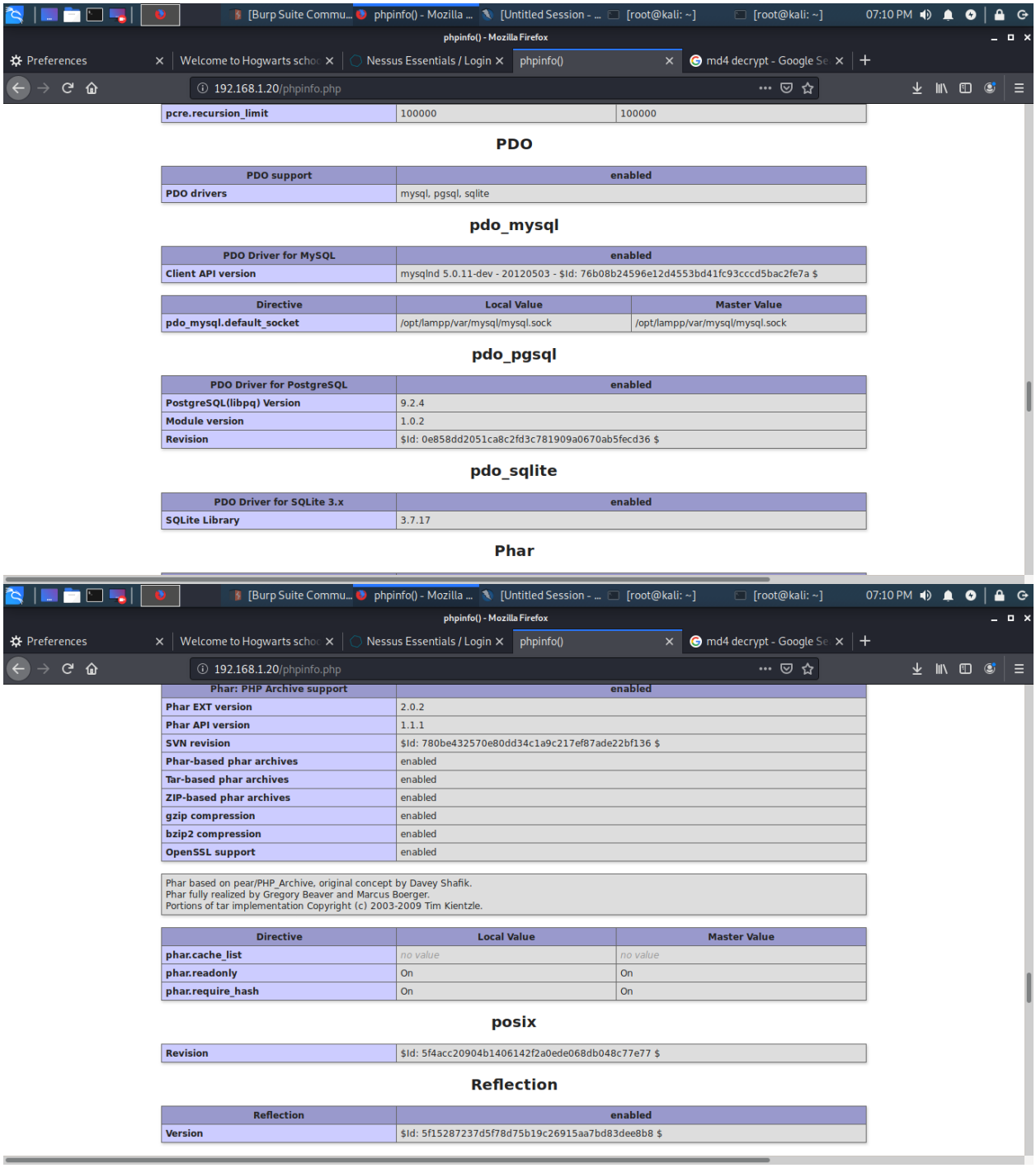
Directive	Local Value	Master Value
openssl.cafile	/opt/lampp/share/curl/curl-ca-bundle.crt	/opt/lampp/share/curl/curl-ca-bundle.crt
openssl.capath	*no value*	*no value*

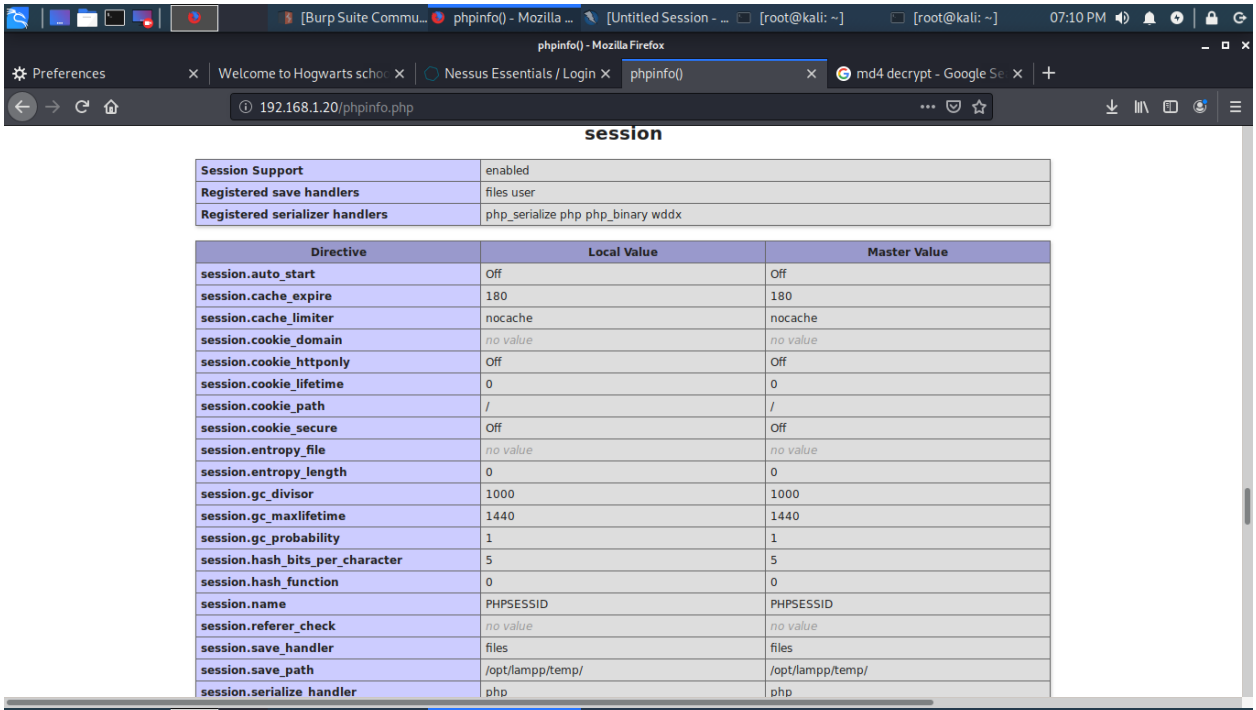
pcre

PCRE (Perl Compatible Regular Expressions) Support	enabled
PCRE Library Version	8.38 2015-11-23

Directive	Local Value	Master Value
pcre.backtrack_limit	1000000	1000000
pcre.recursion_limit	100000	100000

PDO

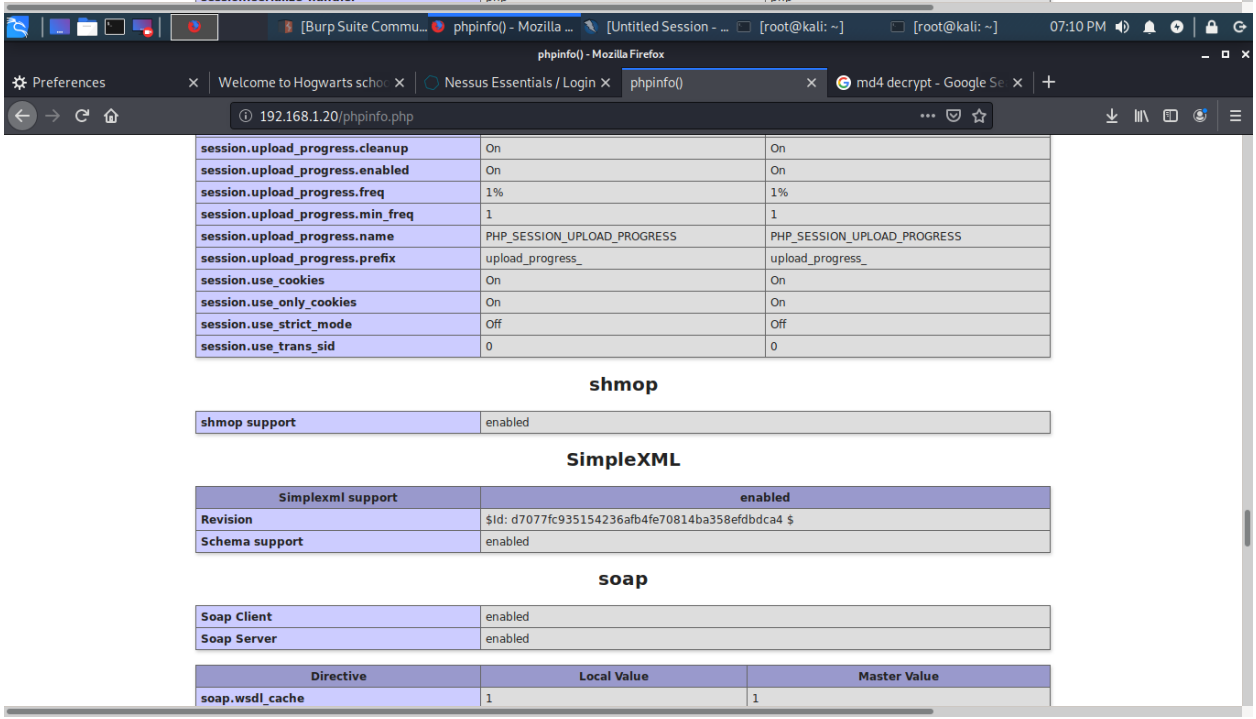

[Burp Suite Commu...
phpinfo() - Mozilla ...
[Untitled Session - ...
[root@kali: ~]
[root@kali: ~]
07:10 PM
phpinfo() - Mozilla Firefox
Preferences
Welcome to Hogwarts scho
Nessus Essentials / Login
phpinfo()
md4 decrypt - Google Se
192.168.1.20/phpinfo.php
pcre.recursion_limit
100000
100000
PDO
PDO support
enabled
PDO drivers
mysql, pgsql, sqlite
pdo_mysql
PDO Driver for MySQL
enabled
Client API version
mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $
Directive
Local Value
Master Value
pdo_mysql.default_socket
/opt/lampp/var/mysql/mysql.sock
/opt/lampp/var/mysql/mysql.sock
pdo_pgsql
PDO Driver for PostgreSQL
enabled
PostgreSQL(libpq) Version
9.2.4
Module version
1.0.2
Revision
$Id: 0e858dd2051ca8c2fd3c781909a0670ab5fecd36 $
pdo_sqlite
PDO Driver for SQLite 3.x
enabled
SQLite Library
3.7.17
Phar
Phar: PHP Archive support
enabled
Phar EXT version
2.0.2
Phar API version
1.1.1
SVN revision
$Id: 780be432570e80dd34c1a9c217ef87ade22bf136 $
Phar-based phar archives
enabled
Tar-based phar archives
enabled
ZIP-based phar archives
enabled
gzip compression
enabled
bzip2 compression
enabled
OpenSSL support
enabled
Phar based on pear/PHP_Archive, original concept by Davey Shafik.
Phar fully realized by Gregory Beaver and Marcus Boerger.
Portions of tar implementation Copyright (c) 2003-2009 Tim Kientzle.
Directive
Local Value
Master Value
phar.cache_list
no value
no value
phar.readonly
On
On
phar.require_hash
On
On
posix
Revision
$Id: 5f4acc20904b1406142f2a0ede068db048c77e77 $
Reflection
Reflection
enabled
Version
$Id: 5f15287237d5f78d75b19c26915aa7bd83dee8b8 $



session

Session Support	enabled
Registered save handlers	files user
Registered serializer handlers	php_serialize php php_binary wddx

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.cache_expire	180	180
session.cache_limiter	nocache	nocache
session.cookie_domain	*no value*	*no value*
session.cookie_httponly	Off	Off
session.cookie_lifetime	0	0
session.cookie_path	/	/
session.cookie_secure	Off	Off
session.entropy_file	*no value*	*no value*
session.entropy_length	0	0
session.gc_divisor	1000	1000
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.hash_bits_per_character	5	5
session.hash_function	0	0
session.name	PHPSESSID	PHPSESSID
session.referer_check	*no value*	*no value*
session.save_handler	files	files
session.save_path	/opt/lampp/temp/	/opt/lampp/temp/
session.serialize_handler	php	php



session.upload_progress.cleanup	On	On
session.upload_progress.enabled	On	On
session.upload_progress.freq	1%	1%
session.upload_progress.min_freq	1	1
session.upload_progress.name	PHP_SESSION_UPLOAD_PROGRESS	PHP_SESSION_UPLOAD_PROGRESS
session.upload_progress.prefix	upload_progress_	upload_progress_
session.use_cookies	On	On
session.use_only_cookies	On	On
session.use_strict_mode	Off	Off
session.use_trans_sid	0	0

shmop

shmop support	enabled

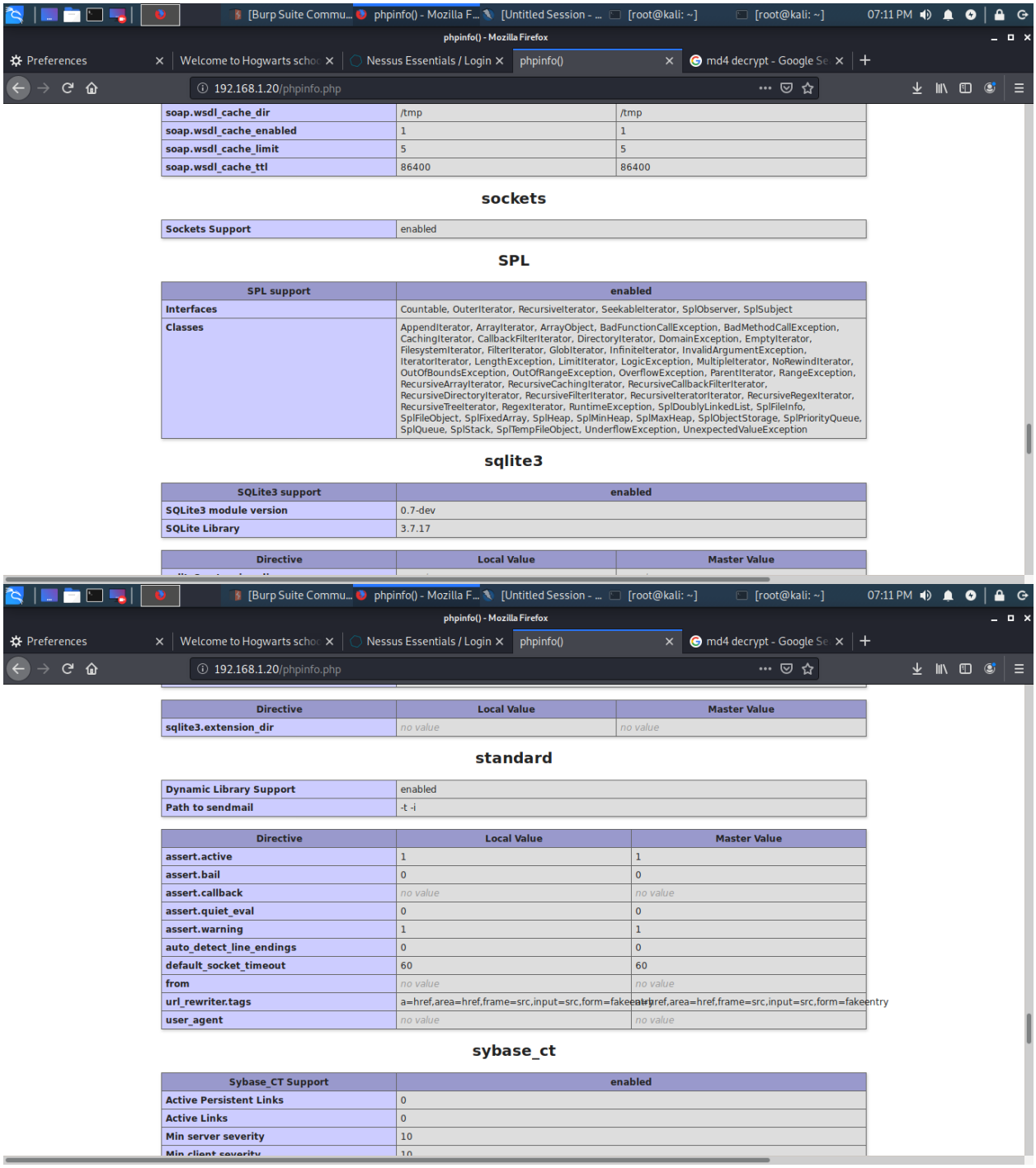
SimpleXML

Simplexml support	enabled
Revision	$Id: d7077fc935154236afb4fe70814ba358efdbdca4 $
Schema support	enabled

soap

Soap Client	enabled
Soap Server	enabled

Directive	Local Value	Master Value
soap.wsdl_cache	1	1



soap.wsdl_cache_dir	/tmp	/tmp
soap.wsdl_cache_enabled	1	1
soap.wsdl_cache_limit	5	5
soap.wsdl_cache_ttl	86400	86400

sockets

Sockets Support	enabled

SPL

SPL support	**enabled**
Interfaces	Countable, OuterIterator, RecursiveIterator, SeekableIterator, SplObserver, SplSubject
Classes	AppendIterator, ArrayIterator, ArrayObject, BadFunctionCallException, BadMethodCallException, CachingIterator, CallbackFilterIterator, DirectoryIterator, DomainException, EmptyIterator, FilesystemIterator, FilterIterator, GlobIterator, InfiniteIterator, InvalidArgumentException, IteratorIterator, LengthException, LimitIterator, LogicException, MultipleIterator, NoRewindIterator, OutOfBoundsException, OutOfRangeException, OverflowException, ParentIterator, RangeException, RecursiveArrayIterator, RecursiveCachingIterator, RecursiveCallbackFilterIterator, RecursiveDirectoryIterator, RecursiveFilterIterator, RecursiveIteratorIterator, RecursiveRegexIterator, RecursiveTreeIterator, RegexIterator, RuntimeException, SplDoublyLinkedList, SplFileInfo, SplFileObject, SplFixedArray, SplHeap, SplMinHeap, SplMaxHeap, SplObjectStorage, SplPriorityQueue, SplQueue, SplStack, SplTempFileObject, UnderflowException, UnexpectedValueException

sqlite3

SQLite3 support	**enabled**
SQLite3 module version	0.7-dev
SQLite Library	3.7.17

Directive	Local Value	Master Value
sqlite3.extension_dir	*no value*	*no value*

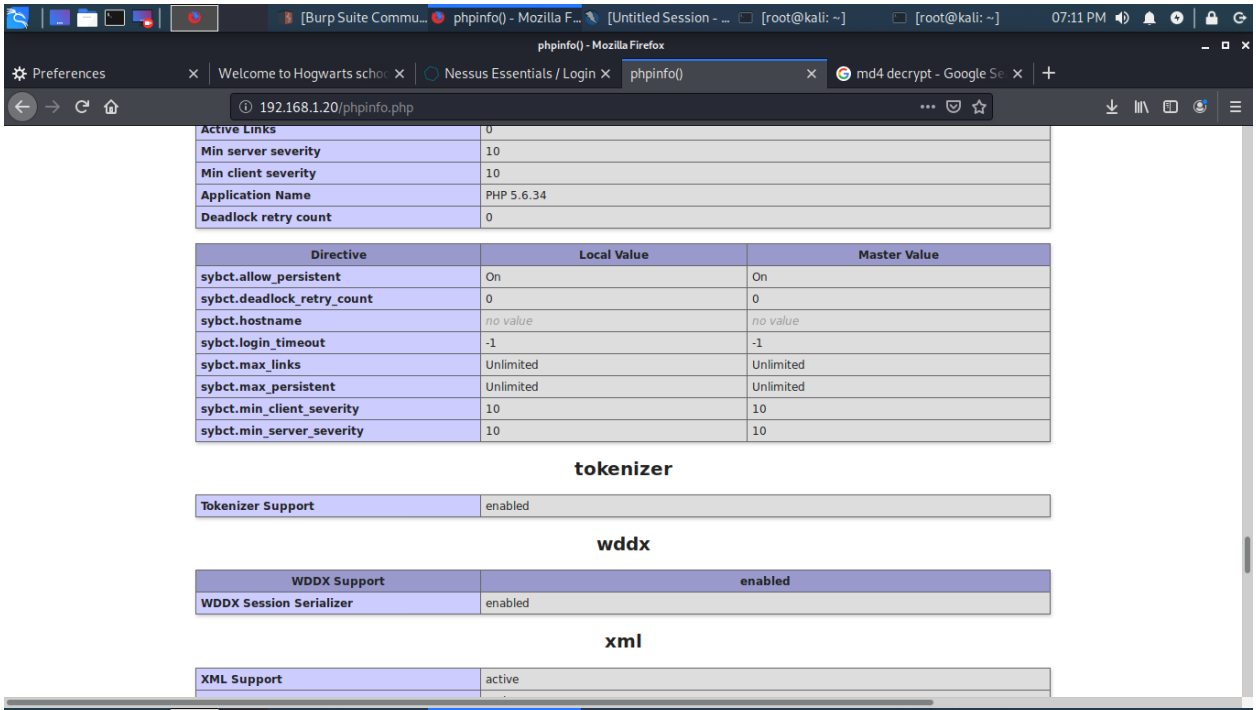
standard

Dynamic Library Support	enabled
Path to sendmail	-t -i

Directive	Local Value	Master Value
assert.active	1	1
assert.bail	0	0
assert.callback	*no value*	*no value*
assert.quiet_eval	0	0
assert.warning	1	1
auto_detect_line_endings	0	0
default_socket_timeout	60	60
from	*no value*	*no value*
url_rewriter.tags	a=href,area=href,frame=src,input=src,form=fakeentry	a=href,area=href,frame=src,input=src,form=fakeentry
user_agent	*no value*	*no value*

sybase_ct

Sybase_CT Support	**enabled**
Active Persistent Links	0
Active Links	0
Min server severity	10
Min client severity	10



Active Links	0
Min server severity	10
Min client severity	10
Application Name	PHP 5.6.34
Deadlock retry count	0

Directive	Local Value	Master Value
sybct.allow_persistent	On	On
sybct.deadlock_retry_count	0	0
sybct.hostname	*no value*	*no value*
sybct.login_timeout	-1	-1
sybct.max_links	Unlimited	Unlimited
sybct.max_persistent	Unlimited	Unlimited
sybct.min_client_severity	10	10
sybct.min_server_severity	10	10

tokenizer

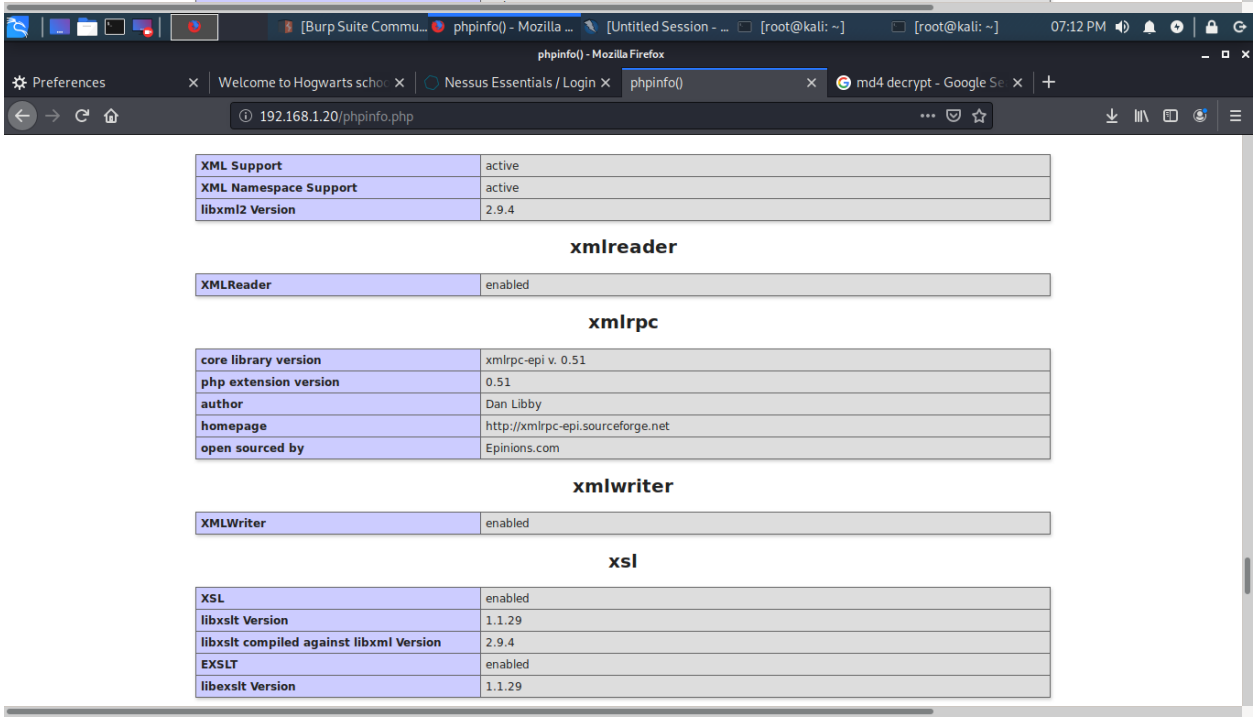
Tokenizer Support	enabled

wddx

WDDX Support	enabled
WDDX Session Serializer	enabled

xml

XML Support	active



XML Support	active
XML Namespace Support	active
libxml2 Version	2.9.4

xmlreader

XMLReader	enabled

xmlrpc

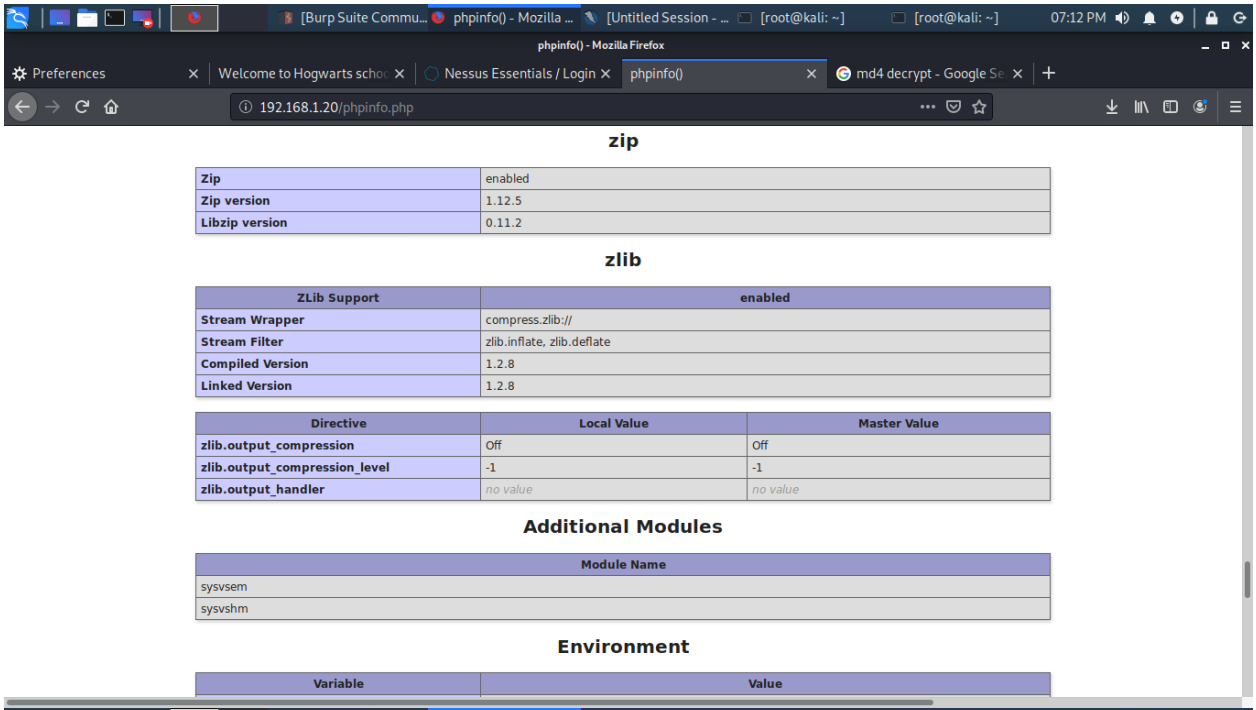
core library version	xmlrpc-epi v. 0.51
php extension version	0.51
author	Dan Libby
homepage	http://xmlrpc-epi.sourceforge.net
open sourced by	Epinions.com

xmlwriter

XMLWriter	enabled

xsl

XSL	enabled
libxslt Version	1.1.29
libxslt compiled against libxml Version	2.9.4
EXSLT	enabled
libexslt Version	1.1.29



zip

Zip	enabled
Zip version	1.12.5
Libzip version	0.11.2

zlib

ZLib Support	**enabled**
Stream Wrapper	compress.zlib://
Stream Filter	zlib.inflate, zlib.deflate
Compiled Version	1.2.8
Linked Version	1.2.8

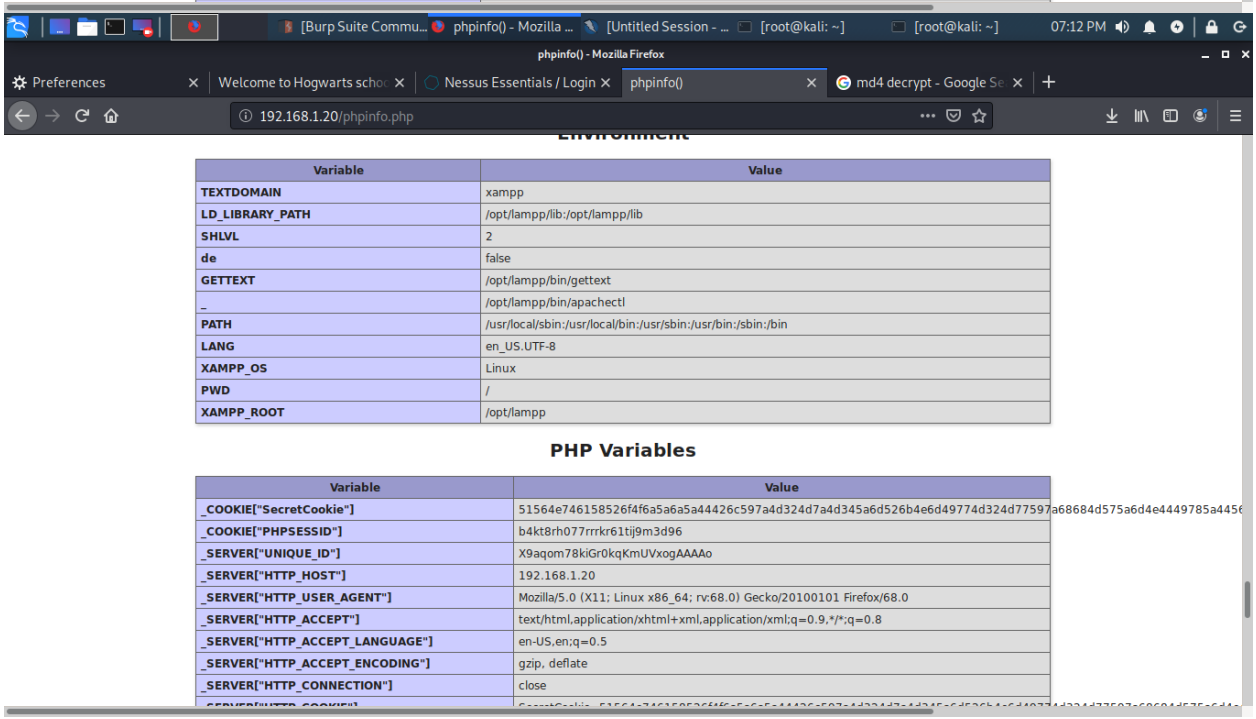
Directive	**Local Value**	**Master Value**
zlib.output_compression	Off	Off
zlib.output_compression_level	-1	-1
zlib.output_handler	*no value*	*no value*

Additional Modules

Module Name
sysvsem
sysvshm

Environment

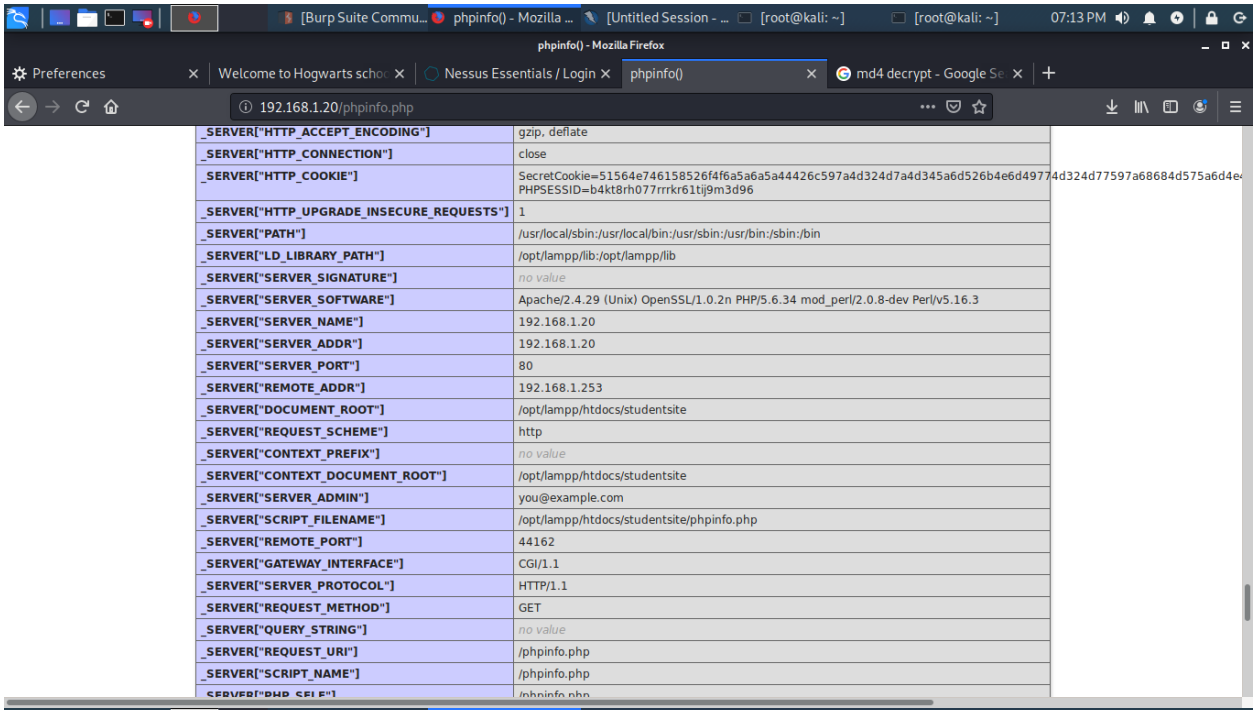
Variable	**Value**



Variable	**Value**
TEXTDOMAIN	xampp
LD_LIBRARY_PATH	/opt/lampp/lib:/opt/lampp/lib
SHLVL	2
de	false
GETTEXT	/opt/lampp/bin/gettext
_	/opt/lampp/bin/apachectl
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
LANG	en_US.UTF-8
XAMPP_OS	Linux
PWD	/
XAMPP_ROOT	/opt/lampp

PHP Variables

Variable	**Value**
_COOKIE["SecretCookie"]	51564e746158526f4f6a5a6a5a44426c597a4d324d7a4d345a6d526b4e6d49774d324d77597a68684d575a6d4e4449785a445
_COOKIE["PHPSESSID"]	b4kt8rh077rrrkr61tij9m3d96
_SERVER["UNIQUE_ID"]	X9aqom78kiGr0kqKmUVxogAAAAo
_SERVER["HTTP_HOST"]	192.168.1.20
_SERVER["HTTP_USER_AGENT"]	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
_SERVER["HTTP_ACCEPT"]	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
_SERVER["HTTP_ACCEPT_LANGUAGE"]	en-US,en;q=0.5
_SERVER["HTTP_ACCEPT_ENCODING"]	gzip, deflate
_SERVER["HTTP_CONNECTION"]	close



_SERVER["HTTP_ACCEPT_ENCODING"]	gzip, deflate
_SERVER["HTTP_CONNECTION"]	close
_SERVER["HTTP_COOKIE"]	SecretCookie=51564e746158526f4f6a5a6a5a44426c597a4d324d7a4d345a6d526b4e6d49774d324d77597a68684d575a6d4e PHPSESSID=b4kt8rh077rrrkr61tij9m3d96
_SERVER["HTTP_UPGRADE_INSECURE_REQUESTS"]	1
_SERVER["PATH"]	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
_SERVER["LD_LIBRARY_PATH"]	/opt/lampp/lib:/opt/lampp/lib
_SERVER["SERVER_SIGNATURE"]	*no value*
_SERVER["SERVER_SOFTWARE"]	Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
_SERVER["SERVER_NAME"]	192.168.1.20
_SERVER["SERVER_ADDR"]	192.168.1.20
_SERVER["SERVER_PORT"]	80
_SERVER["REMOTE_ADDR"]	192.168.1.253
_SERVER["DOCUMENT_ROOT"]	/opt/lampp/htdocs/studentsite
_SERVER["REQUEST_SCHEME"]	http
_SERVER["CONTEXT_PREFIX"]	*no value*
_SERVER["CONTEXT_DOCUMENT_ROOT"]	/opt/lampp/htdocs/studentsite
_SERVER["SERVER_ADMIN"]	you@example.com
_SERVER["SCRIPT_FILENAME"]	/opt/lampp/htdocs/studentsite/phpinfo.php
_SERVER["REMOTE_PORT"]	44162
_SERVER["GATEWAY_INTERFACE"]	CGI/1.1
_SERVER["SERVER_PROTOCOL"]	HTTP/1.1
_SERVER["REQUEST_METHOD"]	GET
_SERVER["QUERY_STRING"]	*no value*
_SERVER["REQUEST_URI"]	/phpinfo.php
_SERVER["SCRIPT_NAME"]	/phpinfo.php
_SERVER["PHP_SELF"]	/phpinfo.php
_SERVER["REQUEST_TIME_FLOAT"]	1607903906.238
_SERVER["REQUEST_TIME"]	1607903906

PHP Credits

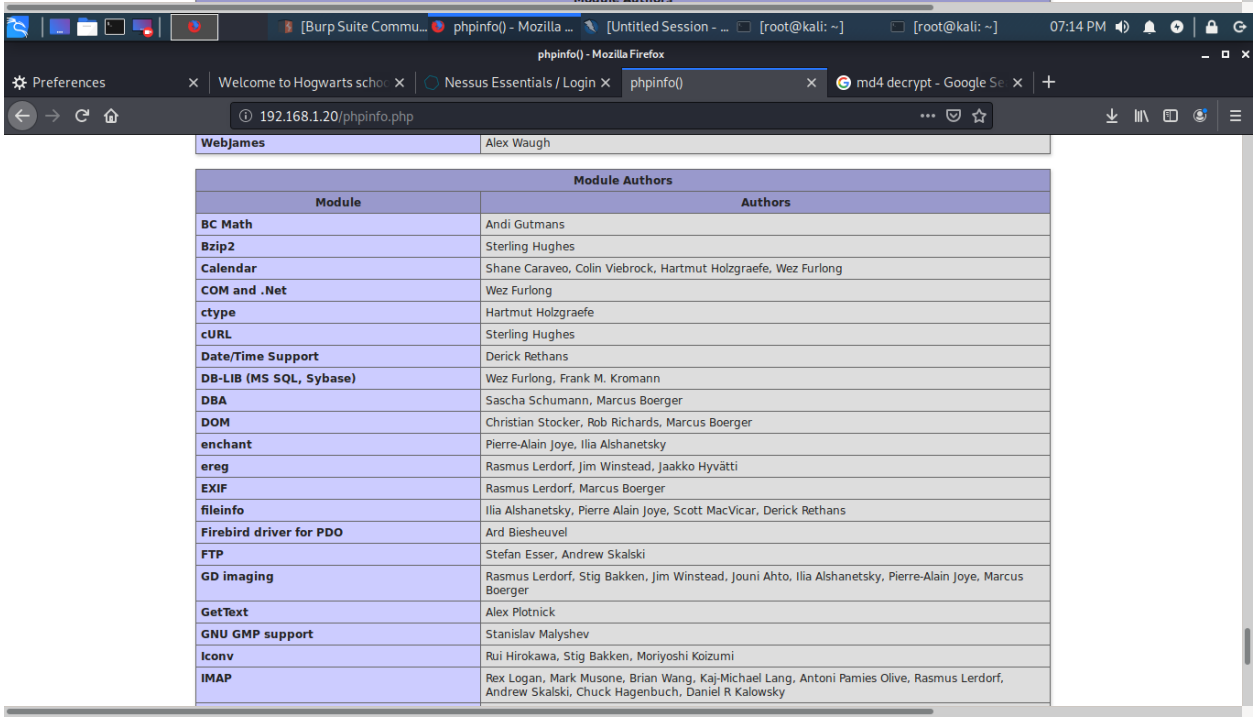
PHP Group
Thies C. Arntzen, Stig Bakken, Shane Caraveo, Andi Gutmans, Rasmus Lerdorf, Sam Ruby, Sascha Schumann, Zeev Suraski, Jim Winstead, Andrei Zmievski

Language Design & Concept
Andi Gutmans, Rasmus Lerdorf, Zeev Suraski, Marcus Boerger

PHP Authors	
Contribution	**Authors**
Zend Scripting Language Engine	Andi Gutmans, Zeev Suraski, Stanislav Malyshev, Marcus Boerger, Dmitry Stogov, Xinchen Hui, Nikita Popov
Extension Module API	Andi Gutmans, Zeev Suraski, Andrei Zmievski
UNIX Build and Modularization	Stig Bakken, Sascha Schumann, Jani Taskinen
Windows Port	Shane Caraveo, Zeev Suraski, Wez Furlong, Pierre-Alain Joye, Anatol Belski
Server API (SAPI) Abstraction Layer	Andi Gutmans, Shane Caraveo, Zeev Suraski
Streams Abstraction Layer	Wez Furlong, Sara Golemon
PHP Data Objects Layer	Wez Furlong, Marcus Boerger, Sterling Hughes, George Schlossnagle, Ilia Alshanetsky
Output Handler	Zeev Suraski, Thies C. Arntzen, Marcus Boerger, Michael Wallner

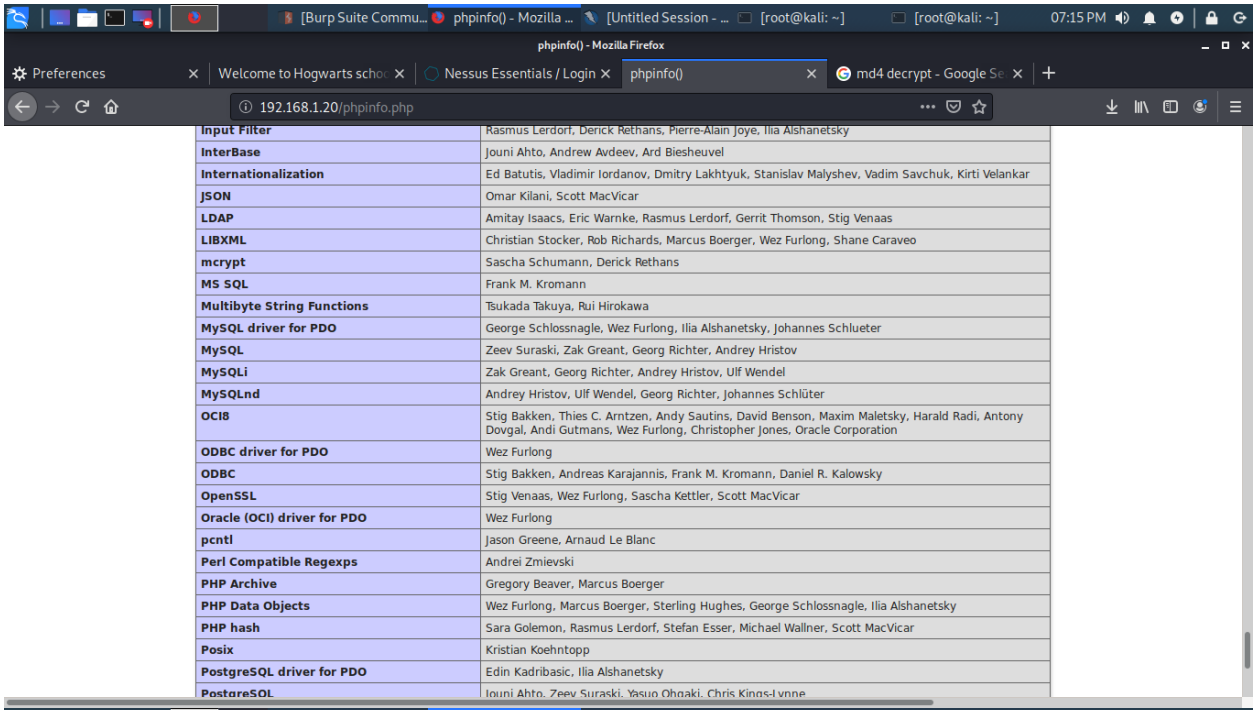
Output Handler	Zeev Suraski, Thies C. Arntzen, Marcus Boerger, Michael Wallner

SAPI Modules	
Contribution	**Authors**
AOLserver	Sascha Schumann
Apache 1.3 (apache_hooks)	Rasmus Lerdorf, Zeev Suraski, Stig Bakken, David Sklar, George Schlossnagle, Lukas Schroeder
Apache 1.3	Rasmus Lerdorf, Zeev Suraski, Stig Bakken, David Sklar
Apache 2.0 Filter	Sascha Schumann, Aaron Bannert
Apache 2.0 Handler	Ian Holsman, Justin Erenkrantz (based on Apache 2.0 Filter code)
Caudium / Roxen	David Hedbor
CGI / FastCGI	Rasmus Lerdorf, Stig Bakken, Shane Caraveo, Dmitry Stogov
CLI	Edin Kadribasic, Marcus Boerger, Johannes Schlueter, Moriyoshi Koizumi, Xinchen Hui
Continuity	Alex Leigh (based on nsapi code)
Embed	Edin Kadribasic
FastCGI Process Manager	Andrei Nigmatulin, dreamcat4, Antony Dovgal, Jerome Loyet
ISAPI	Andi Gutmans, Zeev Suraski
litespeed	George Wang
NSAPI	Jayakumar Muthukumarasamy, Uwe Schindler
phpdbg	Felipe Pena, Joe Watkins, Bob Weinand
phttpd	Thies C. Arntzen
pi3web	Holger Zimmermann
Sendmail Milter	Harald Radi
thttpd	Sascha Schumann
tux	Sascha Schumann
WebJames	Alex Waugh

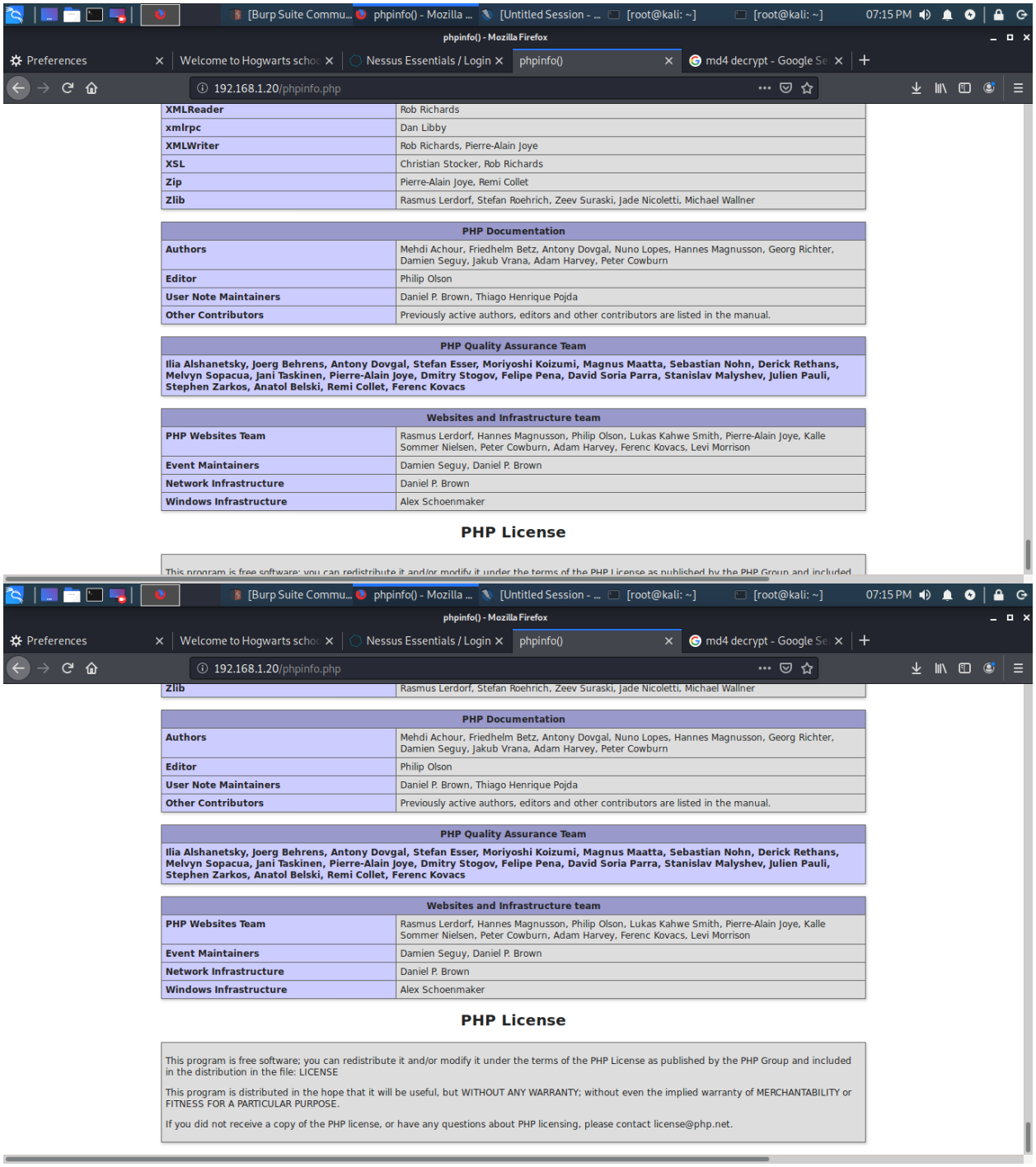


WebJames	Alex Waugh

Module Authors	
Module	**Authors**
BC Math	Andi Gutmans
Bzip2	Sterling Hughes
Calendar	Shane Caraveo, Colin Viebrock, Hartmut Holzgraefe, Wez Furlong
COM and .Net	Wez Furlong
ctype	Hartmut Holzgraefe
cURL	Sterling Hughes
Date/Time Support	Derick Rethans
DB-LIB (MS SQL, Sybase)	Wez Furlong, Frank M. Kromann
DBA	Sascha Schumann, Marcus Boerger
DOM	Christian Stocker, Rob Richards, Marcus Boerger
enchant	Pierre-Alain Joye, Ilia Alshanetsky
ereg	Rasmus Lerdorf, Jim Winstead, Jaakko Hyvätti
EXIF	Rasmus Lerdorf, Marcus Boerger
fileinfo	Ilia Alshanetsky, Pierre Alain Joye, Scott MacVicar, Derick Rethans
Firebird driver for PDO	Ard Biesheuvel
FTP	Stefan Esser, Andrew Skalski
GD imaging	Rasmus Lerdorf, Stig Bakken, Jim Winstead, Jouni Ahto, Ilia Alshanetsky, Pierre-Alain Joye, Marcus Boerger
GetText	Alex Plotnick
GNU GMP support	Stanislav Malyshev
Iconv	Rui Hirokawa, Stig Bakken, Moriyoshi Koizumi
IMAP	Rex Logan, Mark Musone, Brian Wang, Kaj-Michael Lang, Antoni Pamies Olive, Rasmus Lerdorf, Andrew Skalski, Chuck Hagenbuch, Daniel R Kalowsky

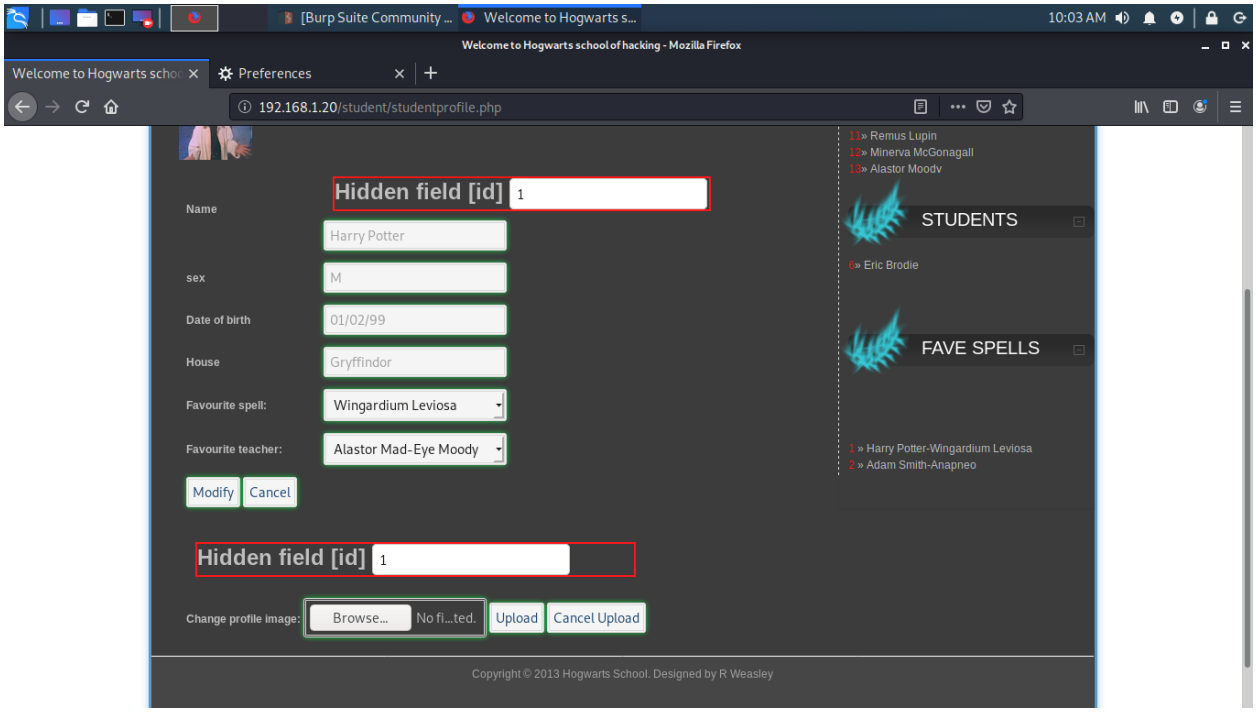


Input Filter	Rasmus Lerdorf, Derick Rethans, Pierre-Alain Joye, Ilia Alshanetsky
InterBase	Jouni Ahto, Andrew Avdeev, Ard Biesheuvel
Internationalization	Ed Batutis, Vladimir Iordanov, Dmitry Lakhtyuk, Stanislav Malyshev, Vadim Savchuk, Kirti Velankar
JSON	Omar Kilani, Scott MacVicar
LDAP	Amitay Isaacs, Eric Warnke, Rasmus Lerdorf, Gerrit Thomson, Stig Venaas
LIBXML	Christian Stocker, Rob Richards, Marcus Boerger, Wez Furlong, Shane Caraveo
mcrypt	Sascha Schumann, Derick Rethans
MS SQL	Frank M. Kromann
Multibyte String Functions	Tsukada Takuya, Rui Hirokawa
MySQL driver for PDO	George Schlossnagle, Wez Furlong, Ilia Alshanetsky, Johannes Schlueter
MySQL	Zeev Suraski, Zak Greant, Georg Richter, Andrey Hristov
MySQLi	Zak Greant, Georg Richter, Andrey Hristov, Ulf Wendel
MySQLnd	Andrey Hristov, Ulf Wendel, Georg Richter, Johannes Schlüter
OCI8	Stig Bakken, Thies C. Arntzen, Andy Sautins, David Benson, Maxim Maletsky, Harald Radi, Antony Dovgal, Andi Gutmans, Wez Furlong, Christopher Jones, Oracle Corporation
ODBC driver for PDO	Wez Furlong
ODBC	Stig Bakken, Andreas Karajannis, Frank M. Kromann, Daniel R. Kalowsky
OpenSSL	Stig Venaas, Wez Furlong, Sascha Kettler, Scott MacVicar
Oracle (OCI) driver for PDO	Wez Furlong
pcntl	Jason Greene, Arnaud Le Blanc
Perl Compatible Regexps	Andrei Zmievski
PHP Archive	Gregory Beaver, Marcus Boerger
PHP Data Objects	Wez Furlong, Marcus Boerger, Sterling Hughes, George Schlossnagle, Ilia Alshanetsky
PHP hash	Sara Golemon, Rasmus Lerdorf, Stefan Esser, Michael Wallner, Scott MacVicar
Posix	Kristian Koehntopp
PostgreSQL driver for PDO	Edin Kadribasic, Ilia Alshanetsky
PostgreSQL	Jouni Ahto, Zeev Suraski, Yasuo Ohgaki, Chris Kings-Lynne
Pspell	Vlad Krupin
Readline	Thies C. Arntzen
Recode	Kristian Koehntopp
Reflection	Marcus Boerger, Timm Friebe, George Schlossnagle, Andrei Zmievski, Johannes Schlueter
Sessions	Sascha Schumann, Andrei Zmievski
Shared Memory Operations	Slava Poliakov, Ilia Alshanetsky
SimpleXML	Sterling Hughes, Marcus Boerger, Rob Richards
SNMP	Rasmus Lerdorf, Harrie Hazewinkel, Mike Jackson, Steven Lawrance, Johann Hanne, Boris Lytochkin
SOAP	Brad Lafountain, Shane Caraveo, Dmitry Stogov
Sockets	Chris Vandomelen, Sterling Hughes, Daniel Beulshausen, Jason Greene
SPL	Marcus Boerger, Etienne Kneuss
SQLite 3.x driver for PDO	Wez Furlong
SQLite3	Scott MacVicar, Ilia Alshanetsky, Brad Dewar
Sybase-CT	Zeev Suraski, Tom May, Timm Friebe
System V Message based IPC	Wez Furlong
System V Semaphores	Tom May
System V Shared Memory	Christian Cartus
tidy	John Coggeshall, Ilia Alshanetsky
tokenizer	Andrei Zmievski, Johannes Schlueter
WDDX	Andrei Zmievski
XML	Stig Bakken, Thies C. Arntzen, Sterling Hughes
XMLReader	Rob Richards
xmlrpc	Dan Libby


[Burp Suite Commu...
phpinfo() - Mozilla ...
[Untitled Session - ...
[root@kali: ~]
[root@kali: ~]
07:15 PM
phpinfo() - Mozilla Firefox
Preferences
Welcome to Hogwarts scho
Nessus Essentials / Login
phpinfo()
md4 decrypt - Google Se
192.168.1.20/phpinfo.php
XMLReader
Rob Richards
xmlrpc
Dan Libby
XMLWriter
Rob Richards, Pierre-Alain Joye
XSL
Christian Stocker, Rob Richards
Zip
Pierre-Alain Joye, Remi Collet
Zlib
Rasmus Lerdorf, Stefan Roehrich, Zeev Suraski, Jade Nicoletti, Michael Wallner
PHP Documentation
Authors
Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy, Jakub Vrana, Adam Harvey, Peter Cowburn
Editor
Philip Olson
User Note Maintainers
Daniel P. Brown, Thiago Henrique Pojda
Other Contributors
Previously active authors, editors and other contributors are listed in the manual.
PHP Quality Assurance Team
Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi Koizumi, Magnus Maatta, Sebastian Nohn, Derick Rethans, Melvyn Sopacua, Jani Taskinen, Pierre-Alain Joye, Dmitry Stogov, Felipe Pena, David Soria Parra, Stanislav Malyshev, Julien Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs
Websites and Infrastructure team
PHP Websites Team
Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Cowburn, Adam Harvey, Ferenc Kovacs, Levi Morrison
Event Maintainers
Damien Seguy, Daniel P. Brown
Network Infrastructure
Daniel P. Brown
Windows Infrastructure
Alex Schoenmaker
PHP License
This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the PHP Group and included in the distribution in the file: LICENSE
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.

APPENDIX C – DATA ENTRY POINTS

1) Student data entry points
    a. Student hidden field in profile

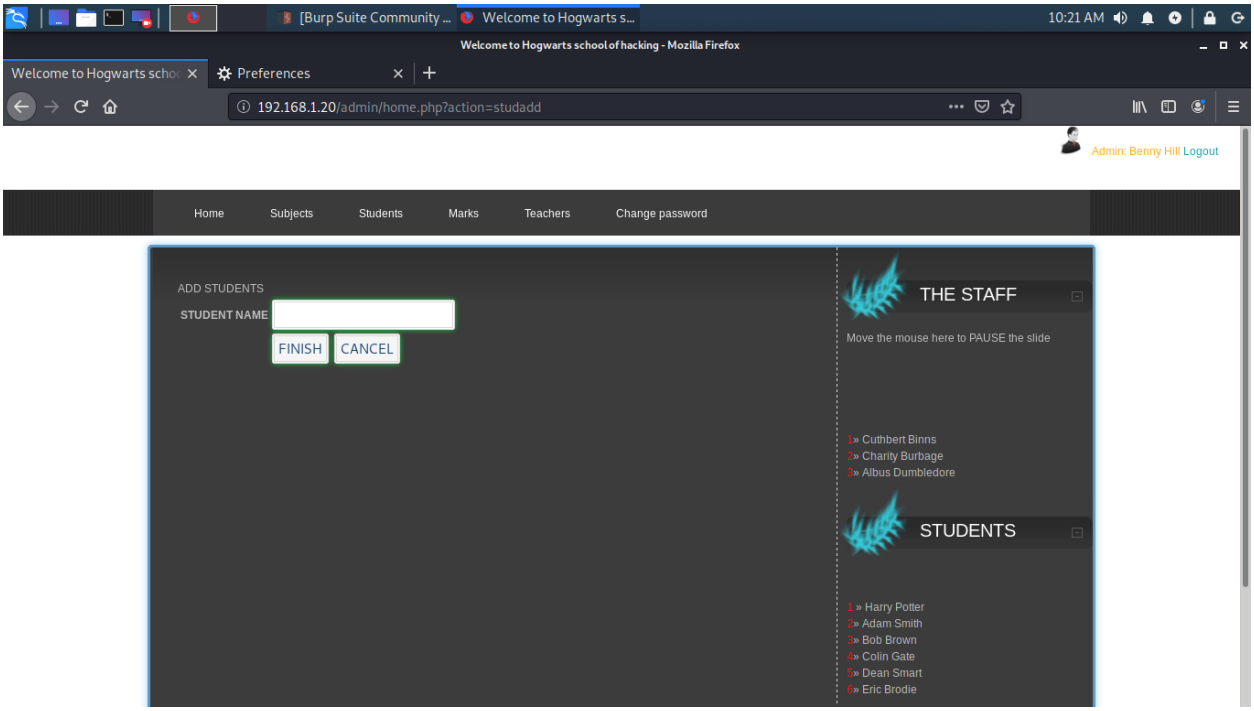


    b. Changing ‘id’ value to edit another student’s information

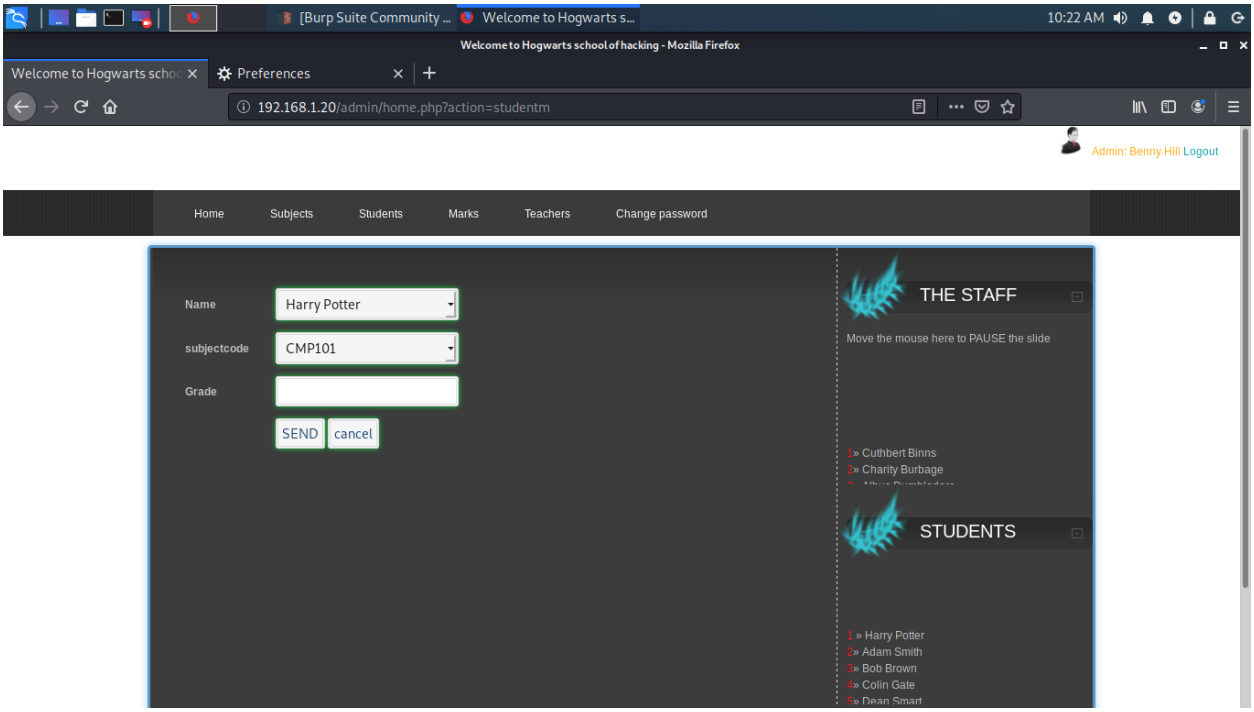
```
POST /student/studentprofile.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.20/student/studentprofile.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 74
Connection: close
Cookie: SecretCookie=614842766448526c636a6f774f54686d4e6d4a6a5a4451324d6a466b4d7a637a5932466b5a54526c4f444d794e6a4933596a526d4e6a6f784e6a41334f5455344d545935;
PHPSESSID=nlj6su6q67o403885taqf2kvf6
Upgrade-Insecure-Requests: 1

id=2&spell=Avada+Kedavra&favteacher=Avada+Kedavra&send=Modify
```

2) Admin data entry points

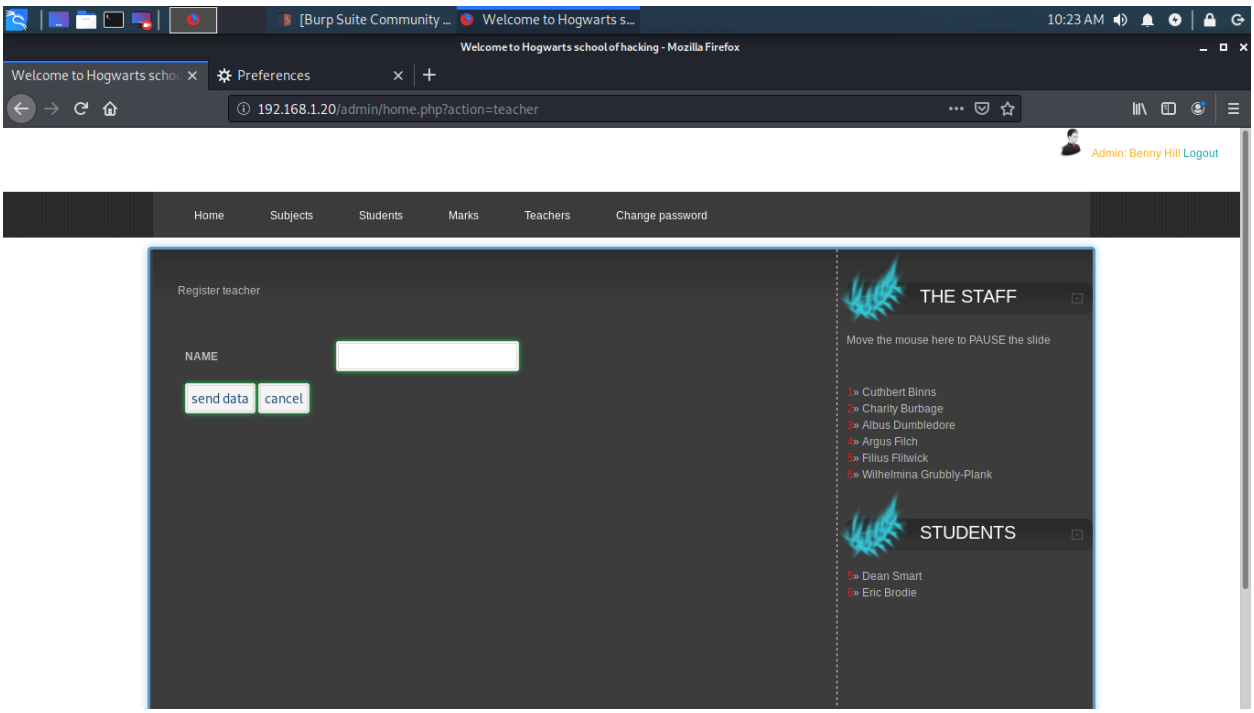
a. Add a student



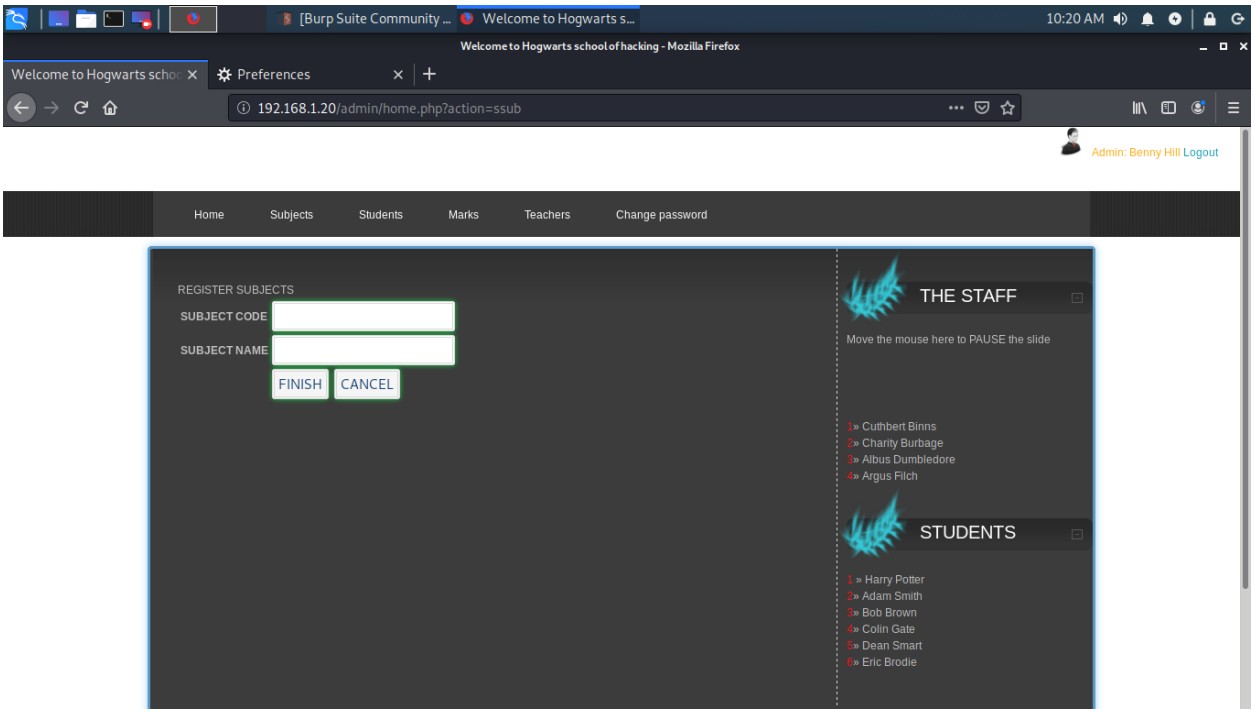
b. Add grades



c. Add a teacher

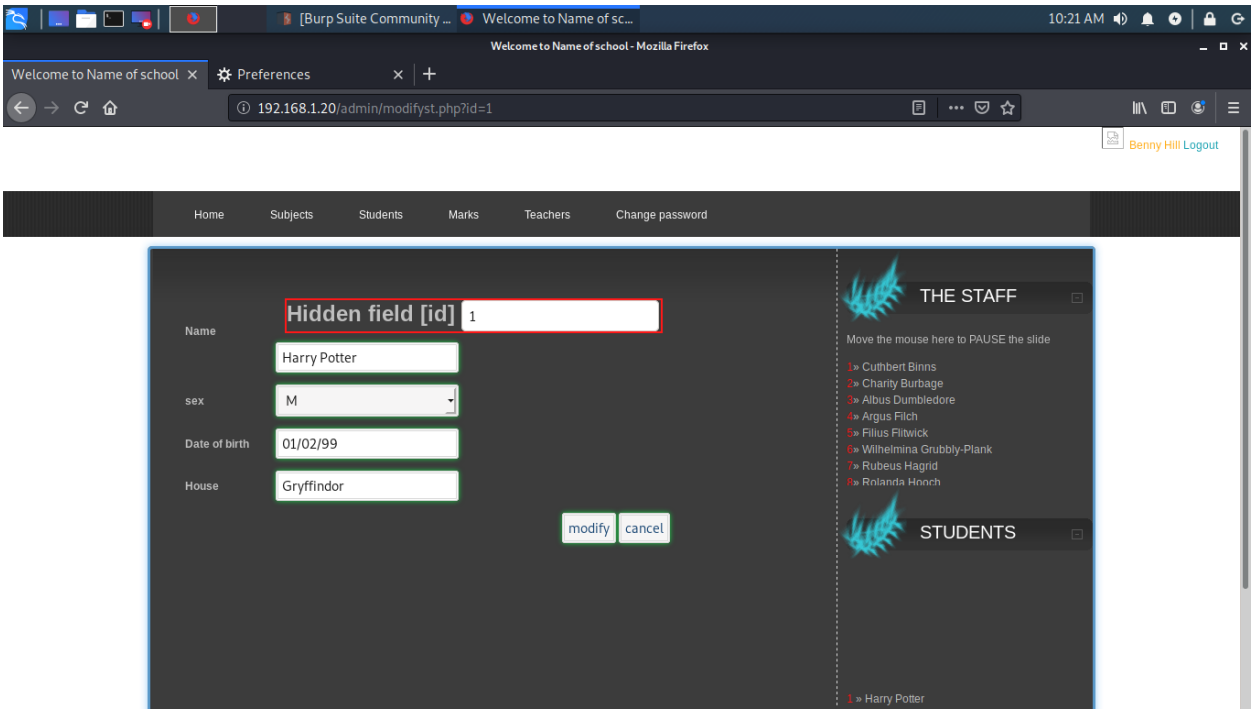


d. Add a subject

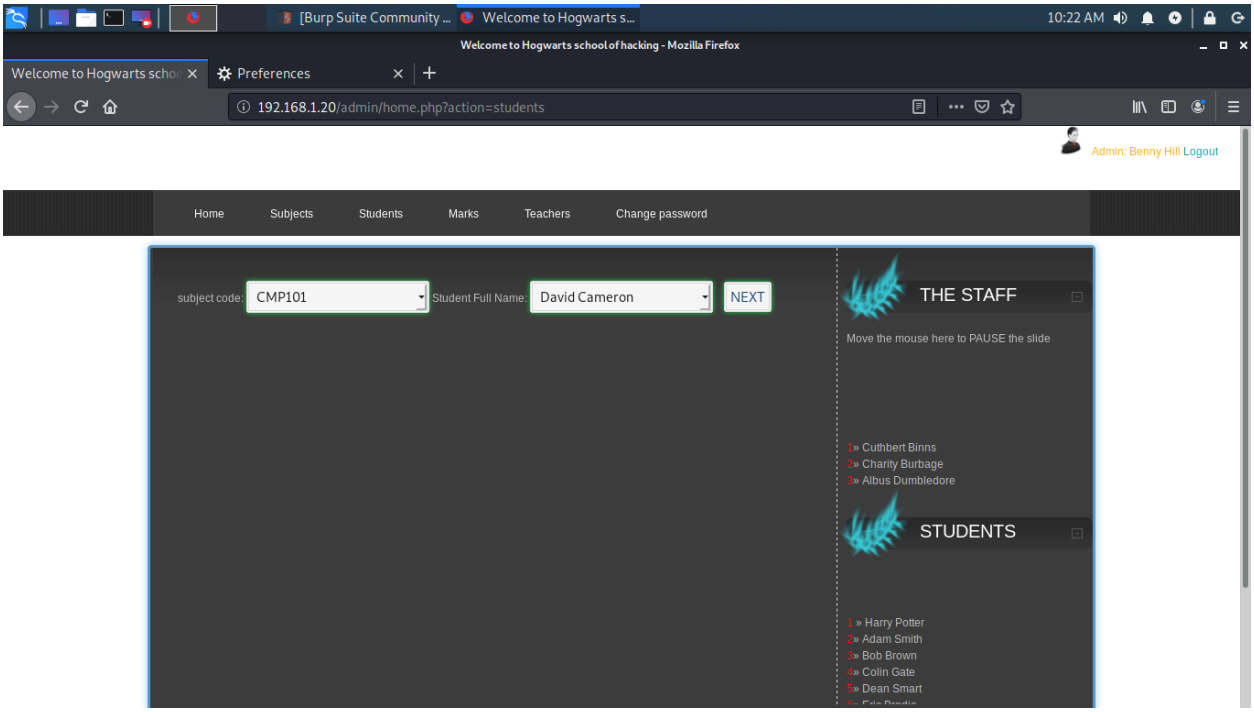


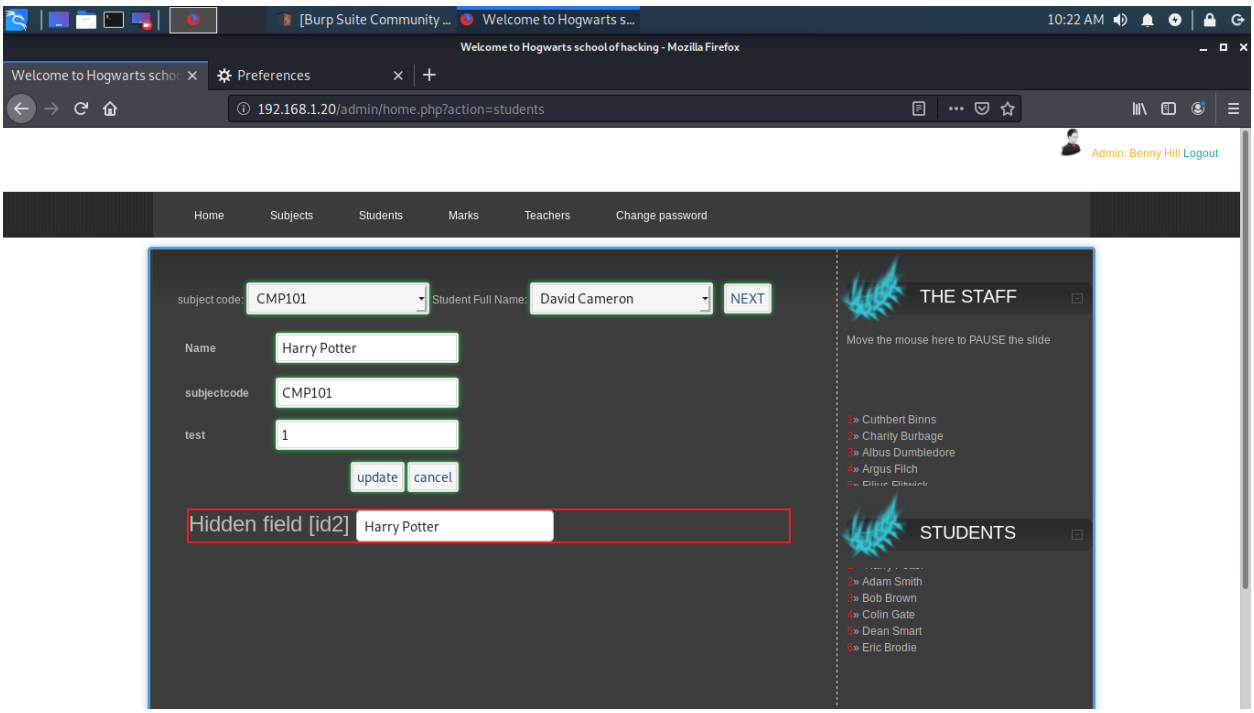
3) Admin hidden fields

a. Update student information

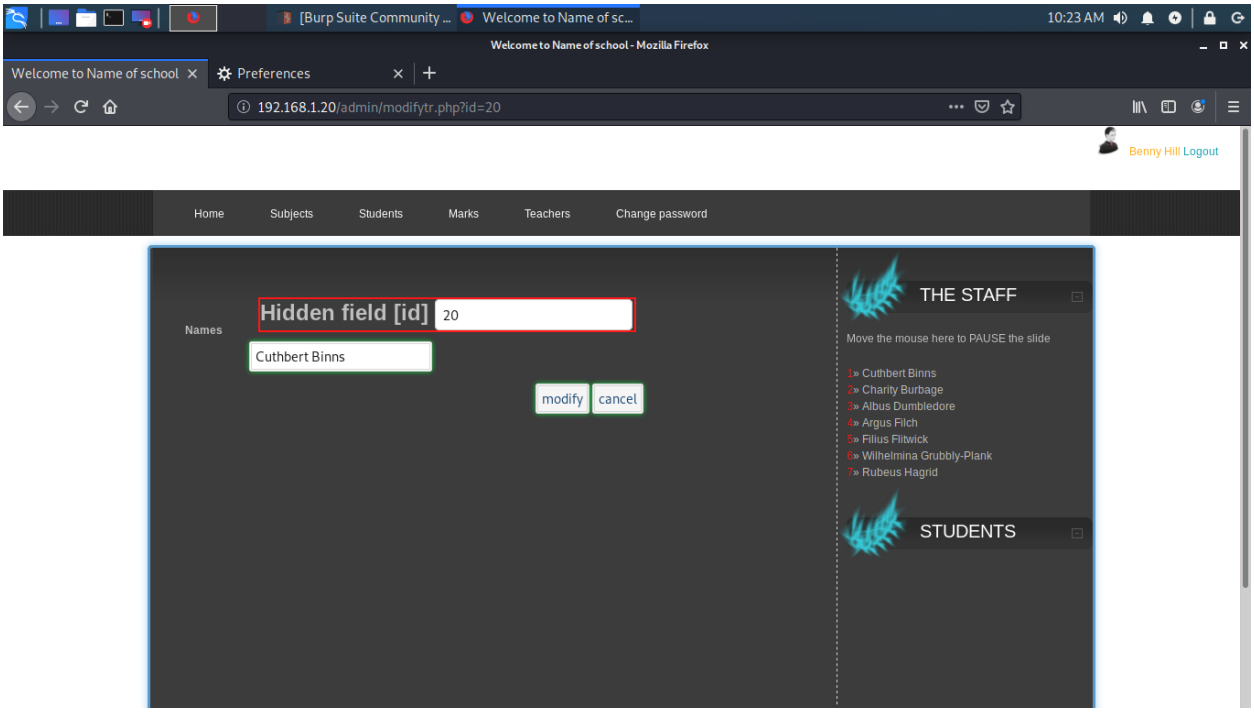


b. Update student grades





c. Edit teacher information


[Burp Suite Community ...
Welcome to Name of sc...
10:23 AM
Welcome to Name of school - Mozilla Firefox
Welcome to Name of school
Preferences
192.168.1.20/admin/modifytr.php?id=20
Benny Hill Logout
Home
Subjects
Students
Marks
Teachers
Change password
Names
Hidden field [id]
20
Cuthbert Binns
modify
cancel
THE STAFF
Move the mouse here to PAUSE the slide
1» Cuthbert Binns
2» Charity Burbage
3» Albus Dumbledore
4» Argus Filch
5» Filius Flitwick
6» Wilhelmina Grubbly-Plank
7» Rubeus Hagrid
STUDENTS

APPENDIX D – SQL INJECTION

a) The tables within the vision database

```
Database: vision
[19 tables]
+---------------------------------------------------+
| bursarystudent                                    |
| club                                              |
| clubmember                                        |
| expenditure                                       |
| grades                                            |
| image                                             |
| itempay                                           |
| nonstaff                                          |
| nonstaffpay                                       |
| payments                                          |
| spells                                            |
| staff                                             |
| student                                           |
| studentmark                                       |
| subject                                           |
| teacher                                           |
| teachercheck                                      |
| teachersalary                                     |
| users                                             |
+---------------------------------------------------+
```

b) The ‘users’ table with the usernames and the hashed passwords dumped

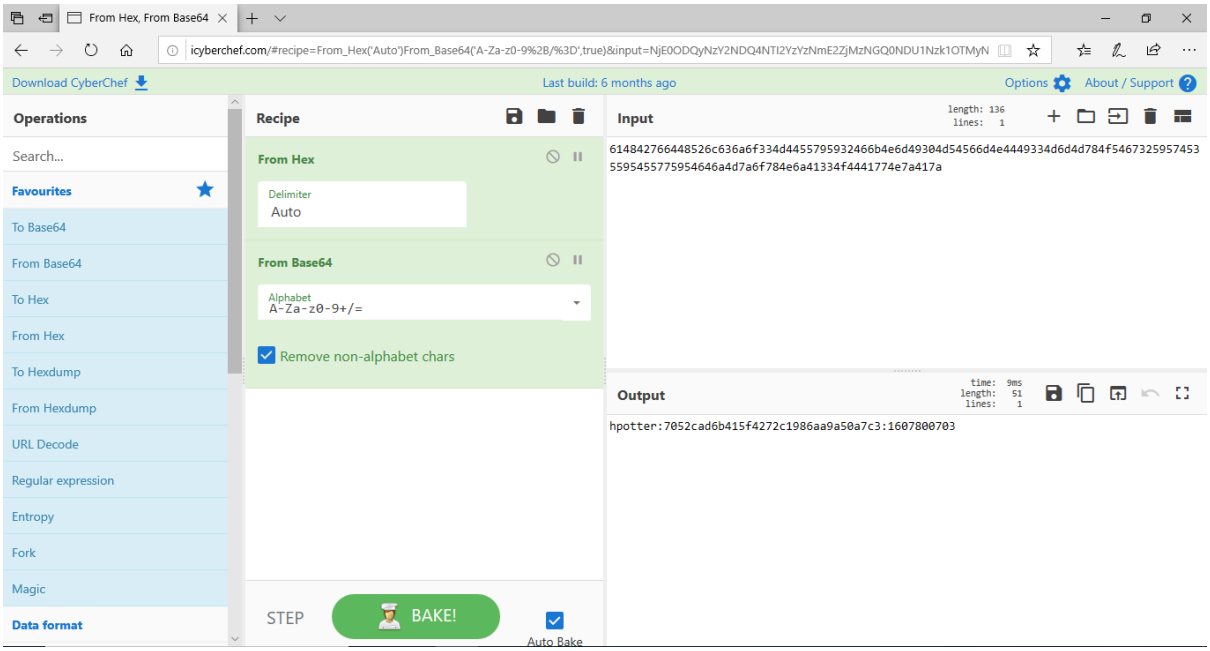
	user_id	STUDENT_	FNAME	LNAME	LEVEL	password	username
2	11	0	Benny	Hill	1	21232f297a57a5a743894a0e4a801fc3 (admin)	admin
3	14	1	Harry	Potter	3	7052cad6b415f4272c1986aa9a50a7c3	hpotter
4	15	2	Adam	Smith	3	6cd0ec36338fdd6b03c0c8a1ff421d5d (persimmon)	ASmith
5	16	3	Bob	Brown	3	6cd0ec36338fdd6b03c0c8a1ff421d5d (persimmon)	BBrown
6	17	4	Colin	Gate	3	6cd0ec36338fdd6b03c0c8a1ff421d5d (persimmon)	CGate
7	18	5	Dean	Smart	3	6cd0ec36338fdd6b03c0c8a1ff421d5d (persimmon)	DSmart
8	19	6	Eric	Brodie	3	6cd0ec36338fdd6b03c0c8a1ff421d5d (persimmon)	EBrodie
9	20	0	Cuthbert	Binns	2	12856030777f8c34d2b41294d06328e1	CBinns
10	21	0	Charity	Burbage	2	12856030777f8c34d2b41294d06328e1	CBurbage
11	22	0	Albus	Dumbledc	2	12856030777f8c34d2b41294d06328e1	ADumbledore
12	23	0	Argus	Argus Filcl	2	12856030777f8c34d2b41294d06328e1	AFilch
13	24	0	Filius	Filius Flitv	2	12856030777f8c34d2b41294d06328e1	FFlitwick
14	25	0	Wilhelmir	Grubbly-P	2	12856030777f8c34d2b41294d06328e1	WGrubbly-Plank
15	26	0	Rubeus	Hagrid	2	12856030777f8c34d2b41294d06328e1	RHagrid
16	27	0	Rolanda	Hooch	2	12856030777f8c34d2b41294d06328e1	RHooch
17	28	0	Silvanus	Kettleburı	2	12856030777f8c34d2b41294d06328e1	SKettleburn
18	29	0	Gilderoy	Lockhart	2	12856030777f8c34d2b41294d06328e1	GLockhart
19	30	0	Remus	Lupin	2	12856030777f8c34d2b41294d06328e1	RLupin
20	31	0	Minerva	McGonaga	2	12856030777f8c34d2b41294d06328e1	MMcGonagall
21	32	0	Alastor	Moody	2	12856030777f8c34d2b41294d06328e1	AMoody

users

c) The injection command suggested by SQLmap

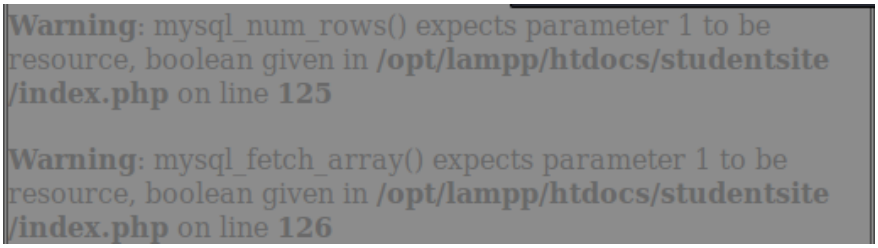
```
---
Parameter: username (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: username=admin' AND 1690=1690 AND 'Xdyh'='Xdyh&password=admin&log
in=login

    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=admin' AND (SELECT 6319 FROM (SELECT(SLEEP(5)))Vprg) AND
 'hrHc'='hrHc&password=admin&login=login
---
```

APPENDIX E – COOKIE DECRYPTION



APPENDIX F – ERROR HANDLING

a) Login page – after failed 'UNION' attack



b) Admin page– change subject grade for a student