# Penetration Report of E-Commerce Website (OWASP 2021)

by

Tajudeen Miftahudeen Akorede

[Year 2022]

# 

# 

# 

# 1.0 OWASP Top10 Listings

## A01 2021: Broken Access Control

Access control is an essential mechanism in the web ecosystem. This mechanism is charged with regulating web contents, maintaining privileges and access among users and the management of authorization. Various sorts of exploitation of access control take advantage of the flaw in this system hence, the term; broken access control. With the successful execution of web attacks by hackers to achieve broken access control, Unauthorized accessed could be gained, sensitive data on servers could be accessed and stolen, user privileges could be escalated, and a distributed denial of service (DDOS) attack could be executed which causes systems to crash as a result.

One of the major ways broken access control can be executed is through the Insecure Direct Object Reference (IDOR) vulnerability. IDOR is an access control flaw that basically occurs when a servers doesn’t validate HTTP requests thoroughly. Attackers can exploit this vulnerability by manipulating HTTP requests to access paths and pages on websites that they’re not supposed to. The credentials and parameter of a browser session can be manipulated to gain access to a webpage session in another browser or a different webpage.

This can potentially cause privilege escalation, sensitive data loss and unauthorized access. With more than three hundred thousand occurrences, broken access control can also occur through SQL injections, forced browsing and web token tampering.

## A02 2021: Cryptographic Failures

With a recorded total occurrence of more than two hundred thousand, cryptographic failure was previously known as Sensitive Data Exposure in previous OWASP Top10s. This usually involves the exploitation of flaws in protection protocols of sensitive data on websites, servers and during transmission.

Cryptographic failure can occur due to weak and old encryption algorithms, use of weak passwords, revelation of stored and transmitted data in plain texts, use of deprecated hash functions like the MD5 and SHA1. Credit card details, emails, passwords, medical information and many more are majorly protected by encryption to ensure privacy. A very popular form of encryption of data in transmission is the SSL and TLS security setting.

## A03 2021: Injection

Injection occurs when an attacker supplies input into a server through the web application in form of combination of texts or files usually with a malicious intent. The supplied input can grant authorized access, reveal sensitive data or infect the server with malware. Popular instances are SQL injection, cross site scripting and file upload vulnerability.

SQL injection involves the supply of malicious SQL query into entry points on a website, this majorly exploits databases stored servers. Cross site scripting is a popular cookie session exploitation method that steals web cookies through malicious URLs containing Javascript strings supplied into entry points on websites. File upload vulnerability involves the injection of malicious PHP files on websites common through image upload points, many websites aren’t enabled to filter different file extension and as a result, this makes them susceptible to exploitation. It is usually exploited successfully when user supplied input are not sanitized and neutralized. Injection has a recorded total occurrence number of more than two hundred thousand

## A04 2021: Insecure Design

This is a vulnerability that arises from web architecture particularly during the development phase. Websites in which security measures are not implemented in their codes and design tend to be exploited in the future by attackers.

Insecure designs can occur in various forms such as rate limiting not being enabled in password logins, excessive use of web resources by users and easy prediction of admin login pages. Insecure design emphasizes the need to implement threat modelling and secure designs patterns in web architecture to prevent known attack vectors

## A05 2021: Security Misconfiguration

Unchanged default user accounts, credentials, configurations, unpatched vulnerabilities, inadequate security measures and error messages containing sensitive information can all be cited as forms in which security misconfiguration can occur. Other forms include unnecessary ports and services, continuous existence and use of default accounts and passwords.

These security misconfiguration flaws can in turn be chained by attackers into broken access control or injection exploitation in network services, websites, frameworks and servers. Attackers’ goals can also be made easier to achieve if the website and server versions and configurations are not up-to-date, this will be discussed in the next section.

## A06 2021: Vulnerable and Outdated Components

Inability to keep web resources up-to-date at regular intervals makes them vulnerable. As a result, attackers exploit this vulnerability by gaining unauthorized access, leaking sensitive data, infecting servers with malwares and rootkits among many others. Outdated components can be due to web administration personnel not keeping up with intermittent updates and patches, inexperience and laxity. Most vulnerable components are usually outdated but they can also be a result of insecure design, security misconfiguration or compatibility issues.

A very common exploitation of vulnerable and outdated components is the creation of reverse shells (RCE) which grants access to the attacker and comprises the whole system. This can further lead to distributed denial of service, malware infection, ransomware execution and sensitive data exposure. RCEs have very severe consequences when executed and they make it extremely difficult for servers to be restored after execution.

## A07 2021: Identification and Authentication Failures

Previously known as broken authentication and with a recorded total occurrence number of more than one hundred thousand, identification and authentication failures said to occur in vulnerable web authentication mechanisms such as permission of valid automated attacks, allowance of users to use weak and common passwords, ineffective two-factor authentication.

Irregular web session timeouts can also be exploited when web session tokens are recycled by the server or when the tokens are not validated properly.

## A08 2021: Software and Data Integrity Failures

This focuses majorly on the parsing of software patches, plugins extensions and updates particularly from unverified sources. This comprises the integrity and reliability of such digital entity and poses dangerous threats to web infrastructures.

Malwares, rootkits and ransomwares can be hidden in these files from untrusted sources, the exploitation of this flaw can also be potentially infectious in the case of IOT devices and the entity with compromised integrity can be programmed by attackers to reject proper and safe updates and remediation.

## A09 2021: Security Logging and Monitoring Failures

Formerly number ten in the OWASP Top10 2017, this vulnerability is said to be hard to test and exploited with a very high security impact and low occurrence number and CVEs to show for it. Logging and monitoring is a very essential record keeping process on the internet and the world of technology at large. This gives details of all activities occurring on websites and servers along with the time and duration of their occurrence.

Security logging and monitoring failures can happen when activities on websites are not properly logged, this can be very dangerous as web management personnel may not be alerted of security issues like failed logins, warnings and errors, unauthorized access and various other security attacks. This could lead to late emergency response to manage the crisis or probably even prevent the implementation of any countermeasures. This flaw can also occur when the log files and process are accessible to attackers, they could be easily cleared or stolen.

## A10 2021: Server-Side Request Forgery

A SSRF flaw is exploited when an attacker forces and manipulates a web application to send a modified request which takes the browser to an unexpected destination even when protected by a firewall. This exploitation makes the website capture a remote web resources without verifying the supplied URL by the attacker.

This can result in improper HTTP redirections, unauthorized access, sensitive data exposure, privilege escalation and complex attacks like remote code executions and distributed denial of service.

# 1.2 Aims and Objectives

The aims and objectives of this report includes the following:

* To deliver a web application for any business as a target for penetration testing.
* Apply penetration testing on the website based on OWASP Top10 2021.
* Give a report from the findings of the penetration testing.
* List and explain preventive and corrective measures to remediate the vulnerabilities.

# 

# 1.3 Methodology

This report employs a practical paradigm in examining the role of risks and vulnerabilities in the existence of web application security. Also, a qualitative research approach was chosen as a suitable methodology for the report, as this approach strengthens an understanding and explanation of outputs and also expectations behind human interaction. Here are the details of the methodology:

* The practical and literal writing aspect took place in Ibadan, Nigeria.
* Data was gathered through the following processes:
* Development of an e-commerce PHP website; E-COM assigned to the IP address 127.0.0.1 and hosting it on a local apache and MySQL server with the XAMPP software.
* Use of the following hardware, software and tools:
* Hardware:
* Software and Tools:

1. Windows 11 Operating System
2. Kali Linux 2022 Operating System
3. VMware Workstation
4. Burp Suite Community Edition
5. Xampp software
6. Local Apache 2.4.54 server
7. MySql
8. Google Chrome browser
9. Mozilla Firefox browser
10. Wordpress

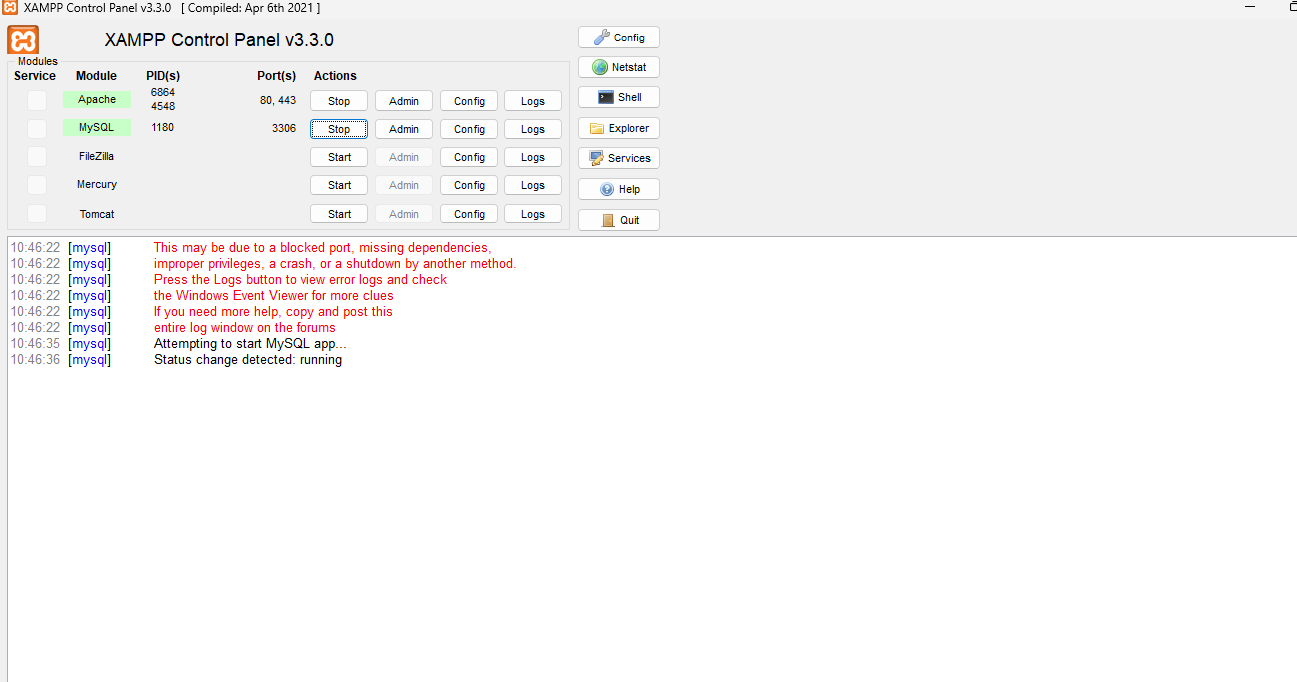
# Pen-testing The E-COM Website Based on OWASP Top10 2021

# A01 2021 Broken Access Control

An automated exploitation of this vulnerability can be demonstrated practically by the penetration testing of the E-COM website with the use of the XAMPP software and the Burp Suite Community Edition software through the following steps:

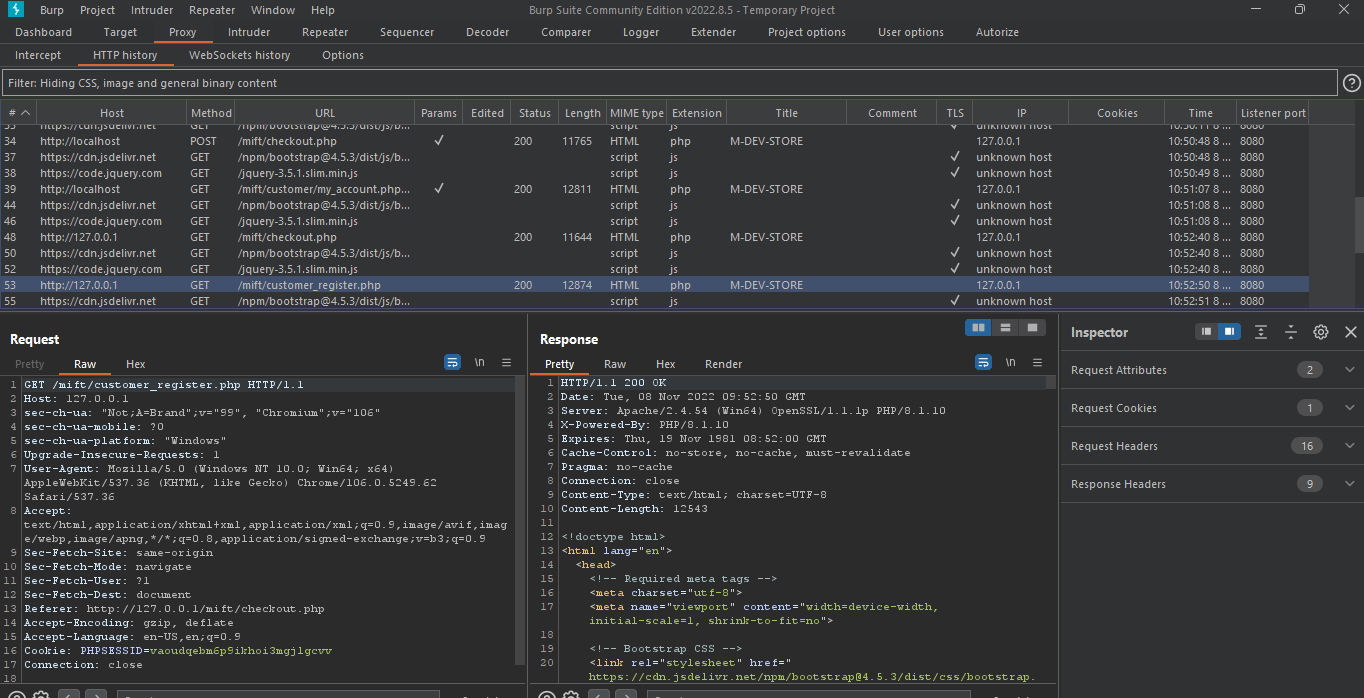
• The Burp Suite Community Edition software is installed and launched on the computer. Then Autorize extension is installed with the aid of the Jython standalone.jar file imported on the software

• The Xampp software is installed and launched which starts the Apache and MySQL server. The website source code folder is moved to the xampp/htdocs directory. This website can be accessed on the web browser app with the address localhost/website source code folder name/index.php.

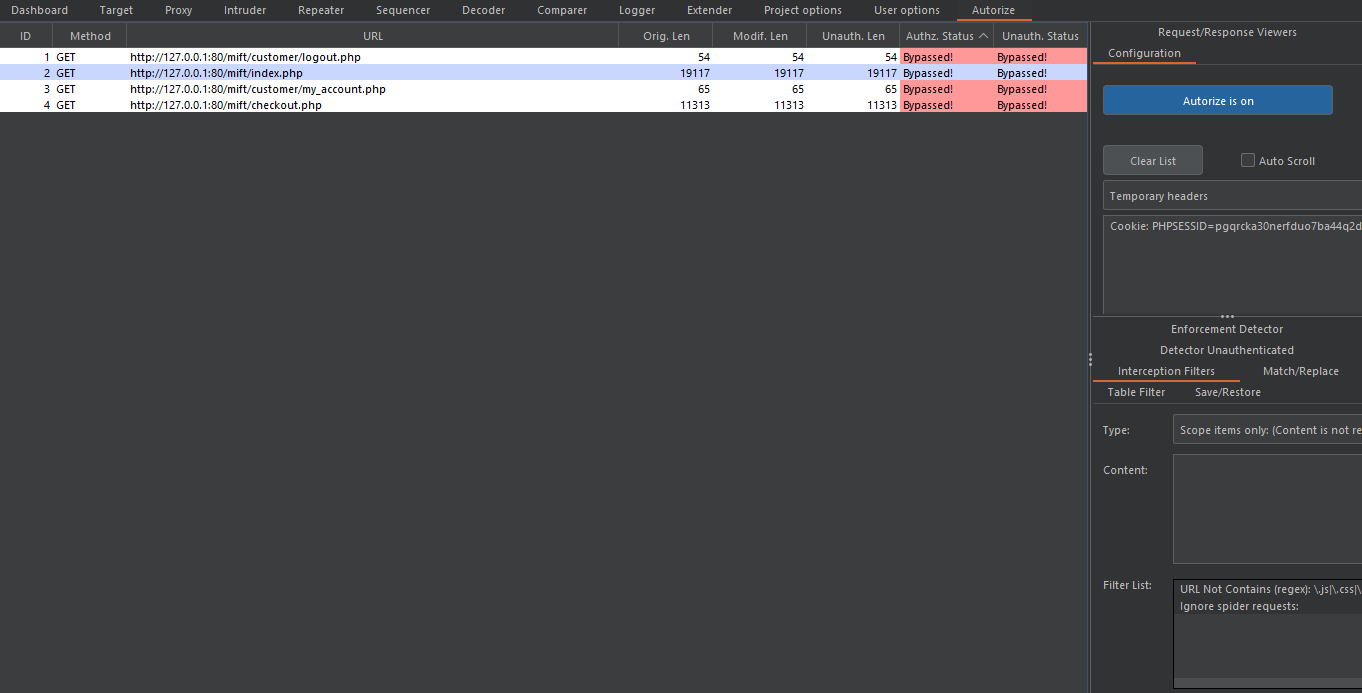


• In the proxy section of the Burp Suite application after being launched, the in-app browser is launched and the E-COM website address is accessed in two modes, first being the default and the other being the incognito mode. A response is given by the server which in turn shows the web interface.

• By reviewing the details of these two HTTP requests and response in the Burp Suite application, the default browser mode PHPSESSID cookie can swapped with the incognito mode by sending the cookie value to the Autorize panel.



This Burp Suite extension then replaces the incognito mode PHPSESSID cookie with that of the default mode when the incognito page is refreshed and sends a modified request. In a successful case, a response is gotten from the server and the Autorize panel shows a red signal of authentication status with a “bypassed” inscription.



**Mitigation**

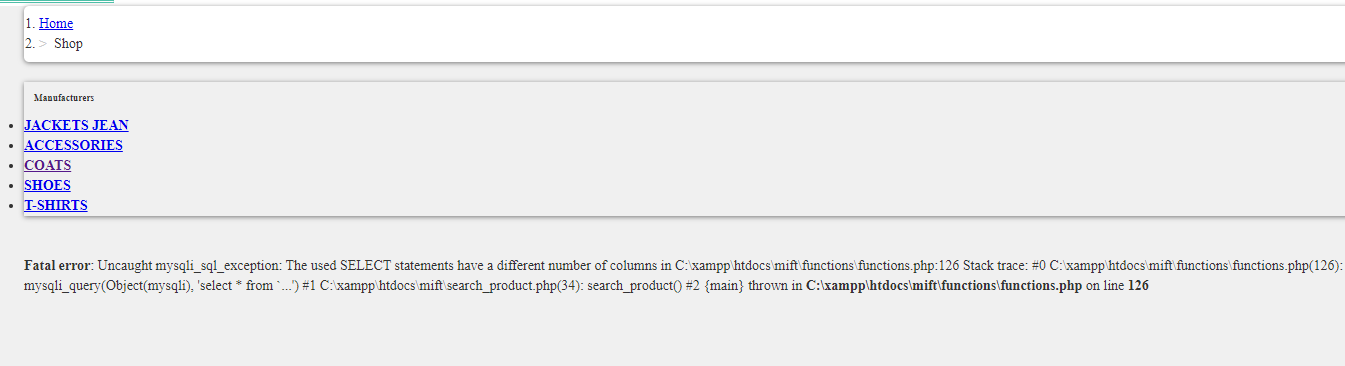
Broken access control can be prevented by limiting user privileges to necessary ones, not enabling directory listing on web servers, rate limiting of inputs to reduce automated attacks, proper logging of access control errors, regular sanitization of server data and implementation of two-factor authentication.

# A02 2021 Cryptographic Failures

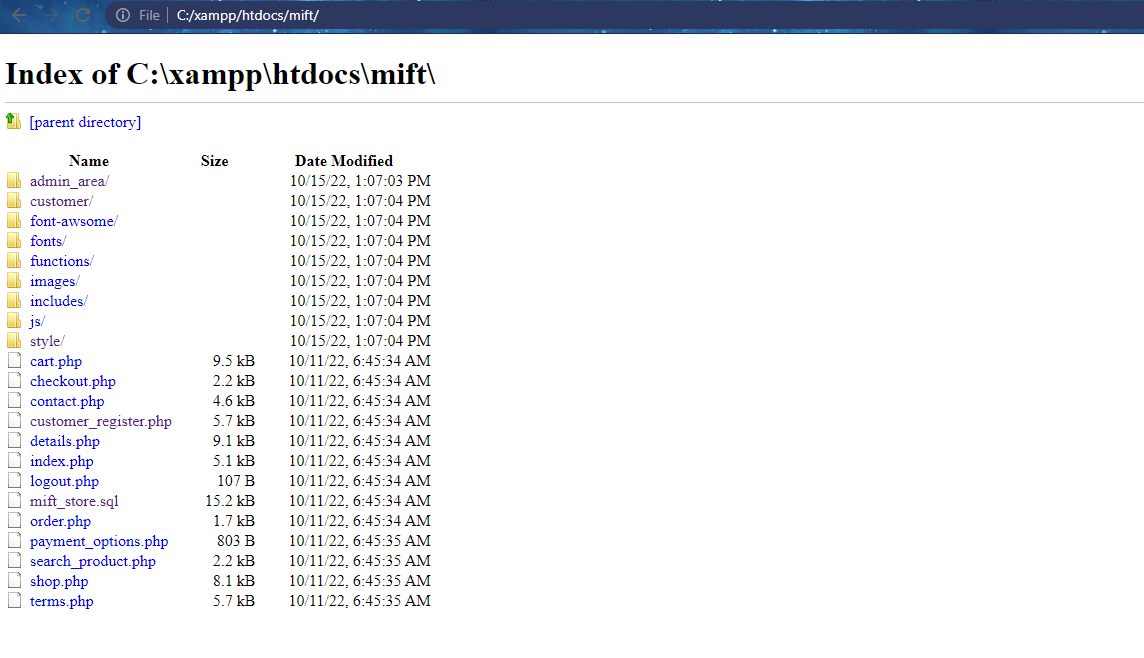
* The pentesting of the E-COM website with respect to cryptographic failure was first attempted through a basic authentication bypass SQL injection on the login page which was successful. A UNION-based database enumeration SQL injection was then attempted to check if the SQL vulnerability exploitation could be escalated.



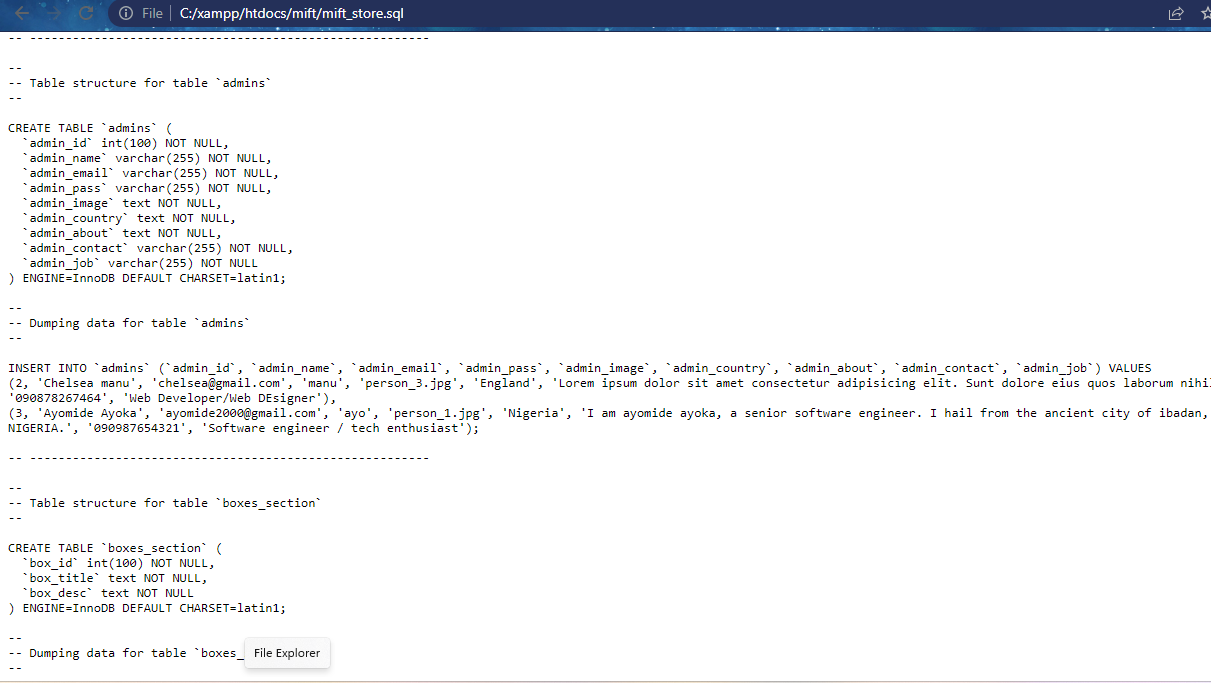
* The SQL injection resulted in a fatal error which was shown on the webpage due to a security misconfiguration flaw of revealing stack error messages to users.

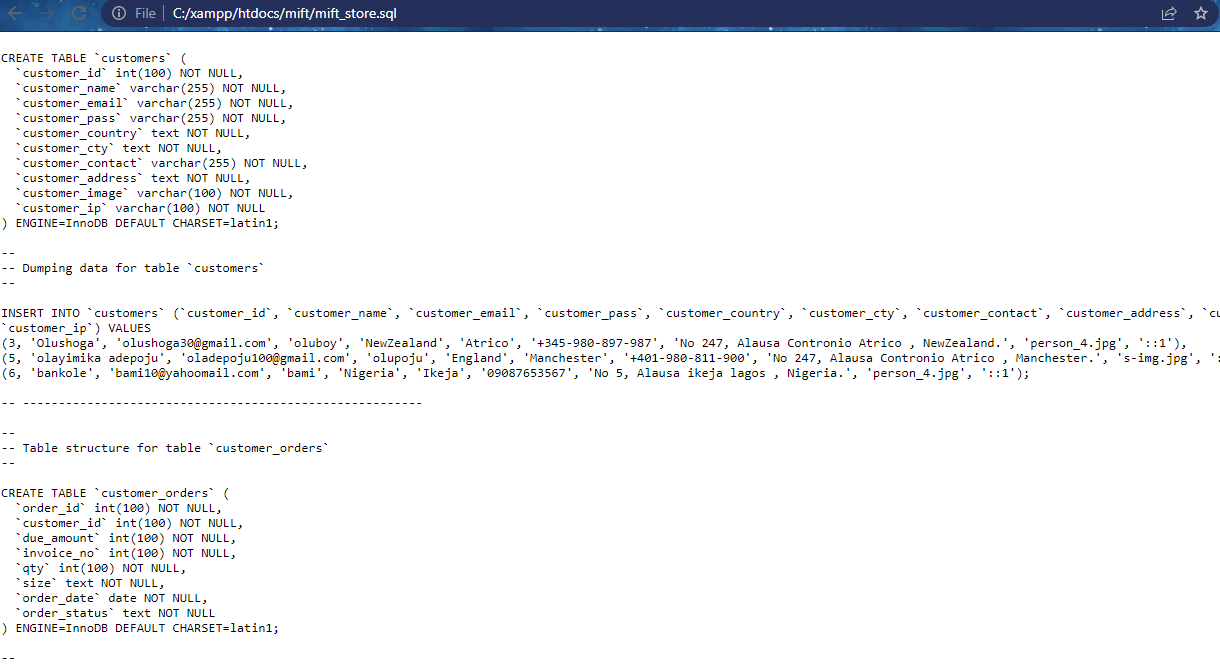


* The file path of certain files, folders and source codes is revealed is accessed. Navigating through the listed directories makes way for access the SQL database for the E-COM website.



* The list of admin and customers’ names, id, e-mail and most importantly in this section, the password is accessed and retrieved in plain text without any encryption algorithm posing restrictions.





**Mitigation**

With the respect to the failed attempt at a SQL injection on the E-COM website, a simple counter-mechanism has been initiated to encode special characters in the injected query. This is shown in the shown in the response header below

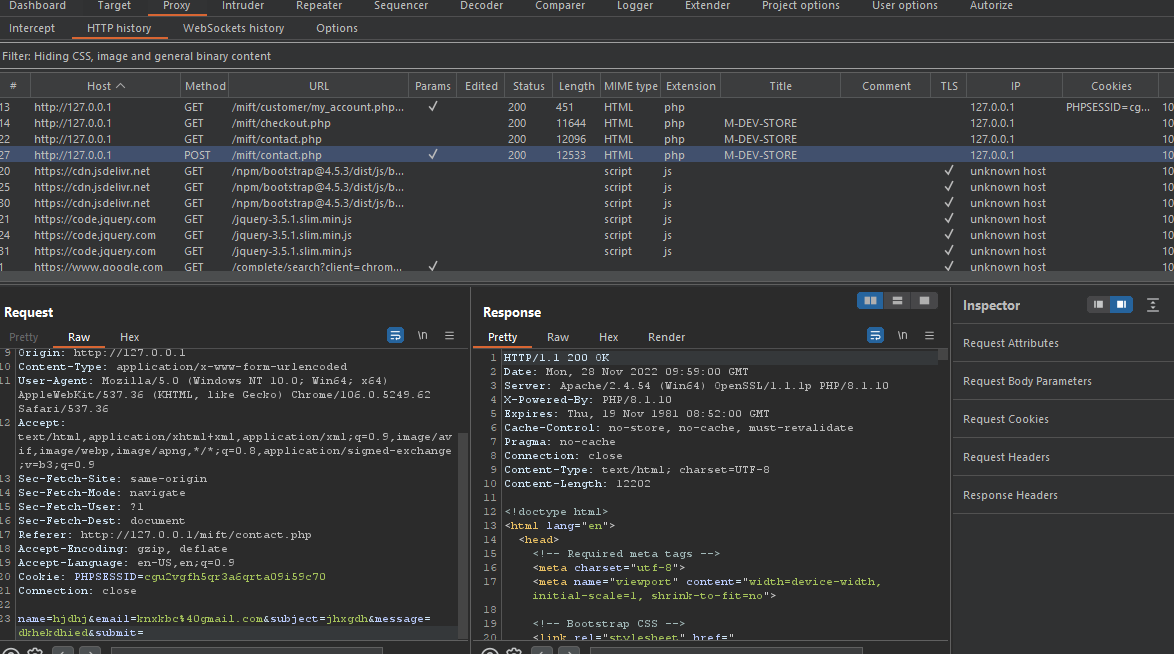


SQL injections can be further prevented through input sanitization and validation, use of filters and parameterized queries including prepared statements. The sensitive information exposure on the websites can be prevented by turning off the visibility of database errors. And sensitive details like passwords should be encrypted appropriately.

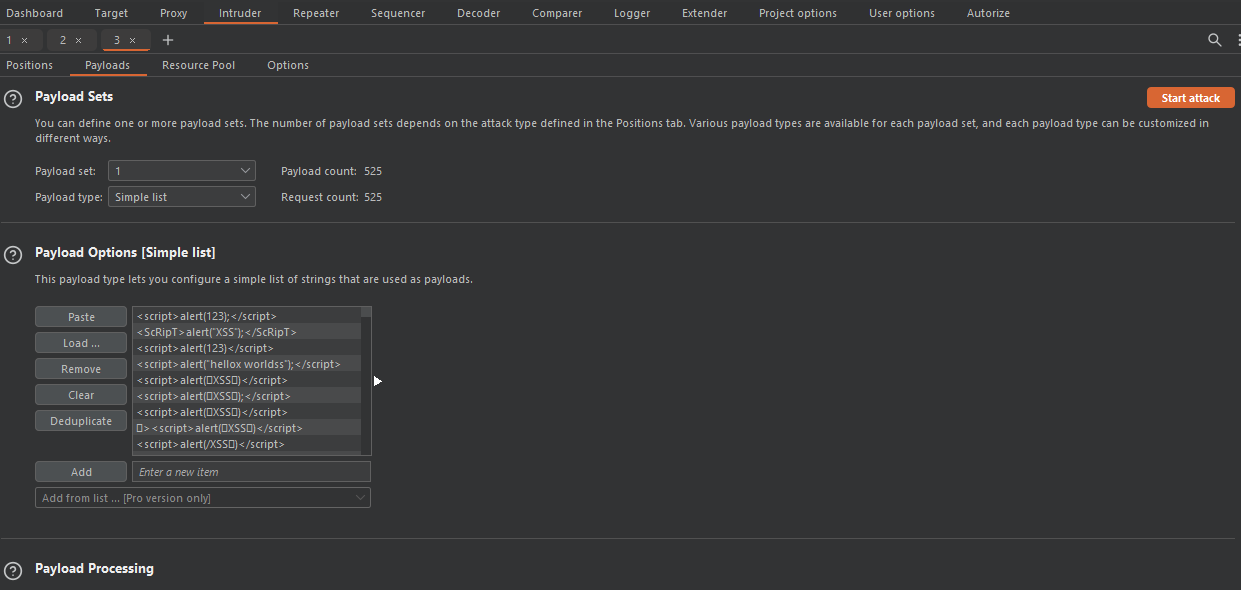
# A03 2021 Injection

Targeting the E-COM website, an automated cross site scripting attack with the BurpSuite can be performed through the following steps:

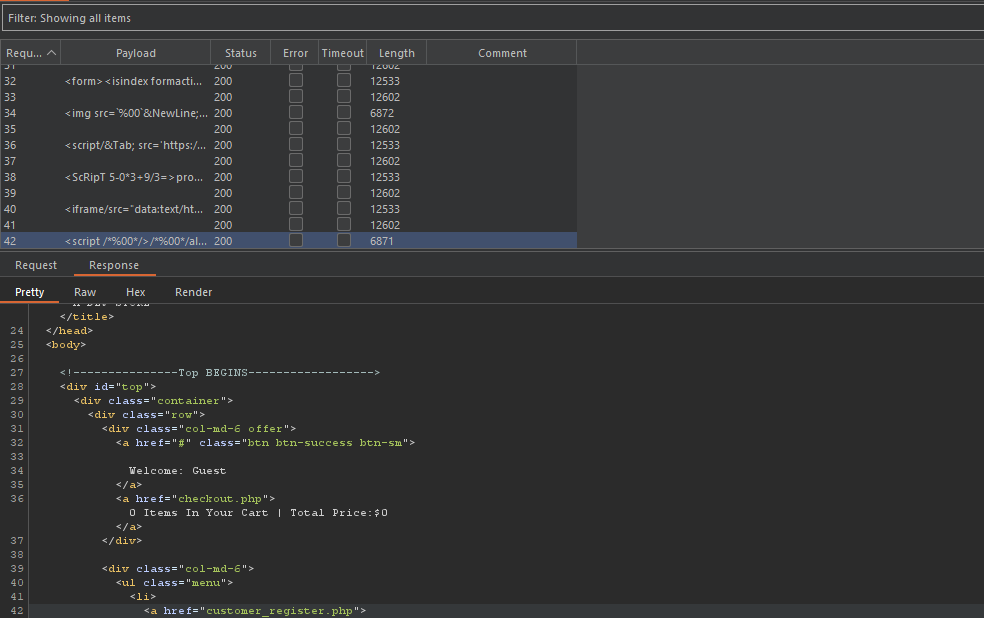
* The software and the in-built browser is launched, then the E-COM webpage with the localhost/mift-area/contact.php accessed.
* Random parameters are then filled in into the four input boxes and the request and response are reviewed in the proxy section.



* The request is then sent to the intruder and the random parameter input on the message bar is highlighted to define the payload position. A sniper attack is selected and a list of xss payloads can be imported.



* The injection of the XSS payloads is initiated and the results of the attempts are shown. If any of this payloads are shown in the response without modification, then it shows the web application is vulnerable to cross-site scripting.



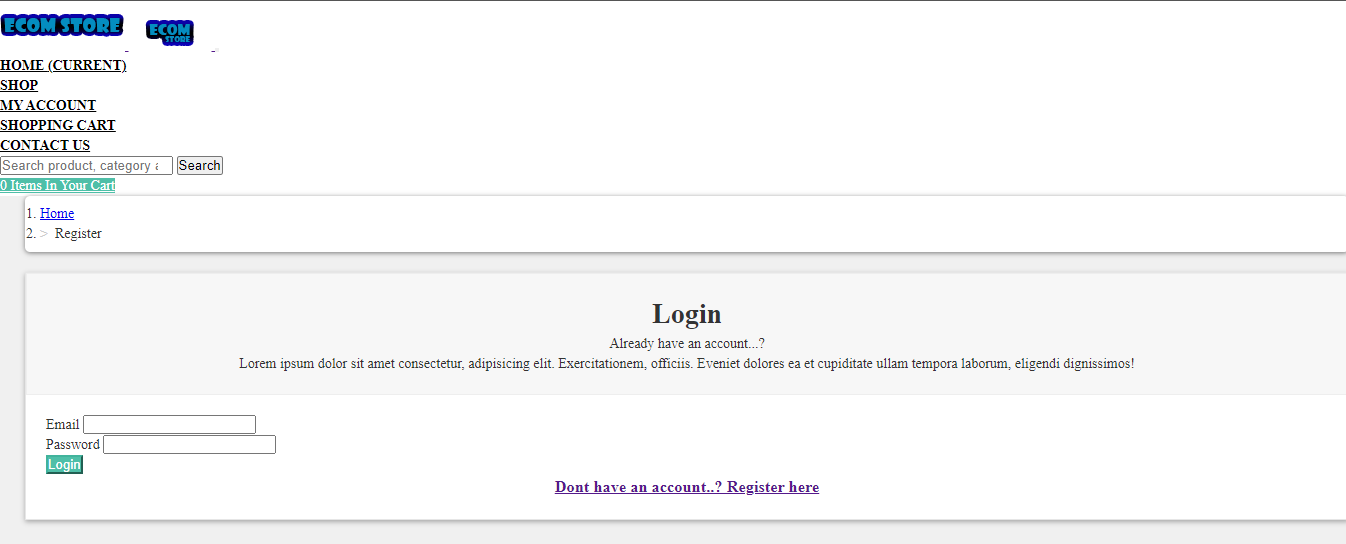
**Mitigation**

An easy way of preventing XSS attacks is the usage of white-list and blacklist filters in websites’ source codes. These filters efficiently limit the chances of a website being exploited through user-supplied input. SQL injections can be prevented by filtering user-supplied input to differentiate SQL queries from web and server data. Other forms of injection can be prevented by using secure APIs, server-side input validation measures

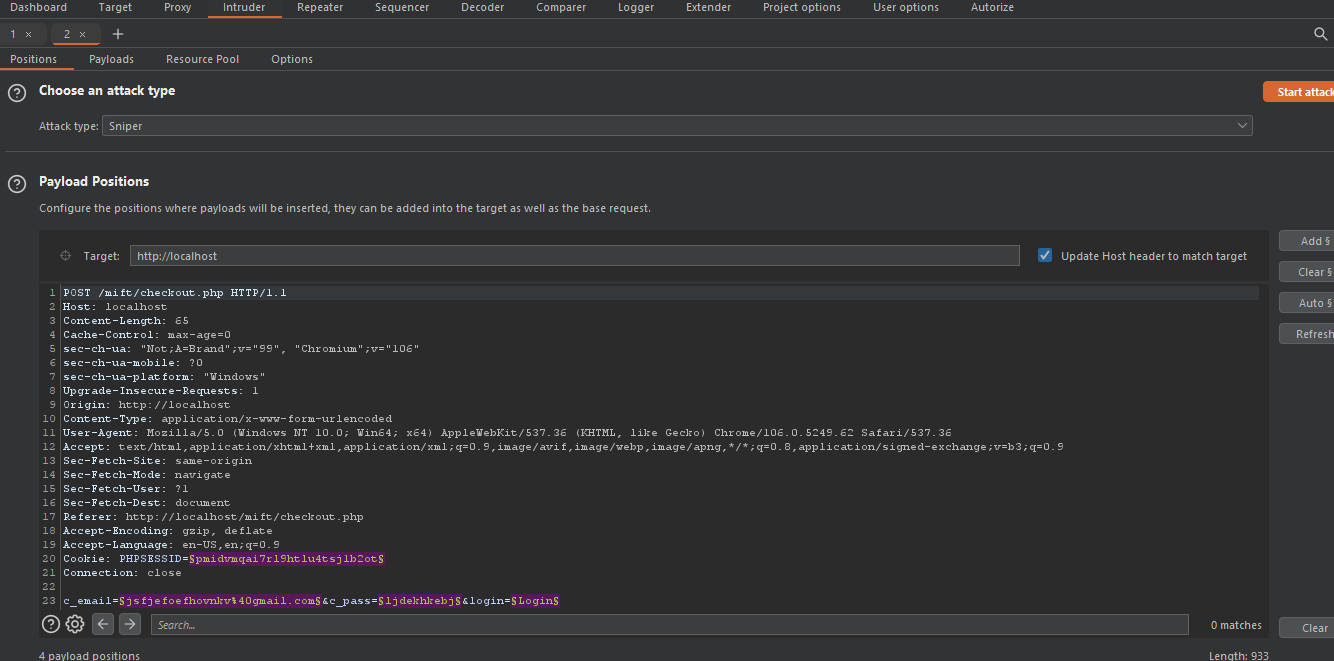
# A04 2021 Insecure Design

Exploiting the E-COM website with respect to insecure design can be performed with a bruteforce attack on the web login page in these steps:

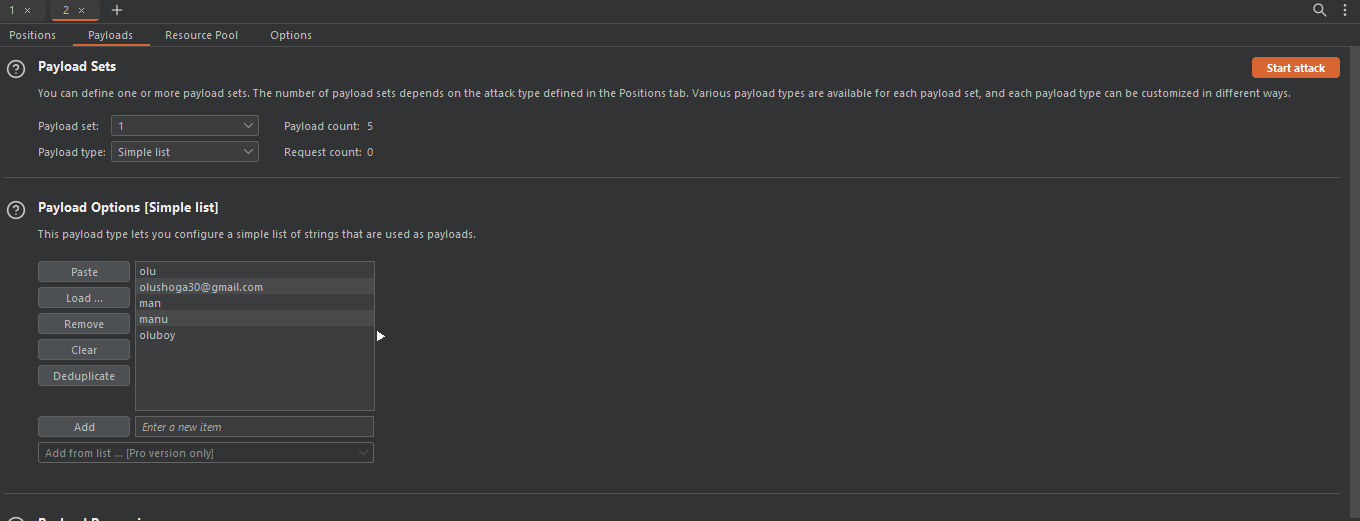
* The login page for the E-COM website is launched with the in-app browser of the Burp application.



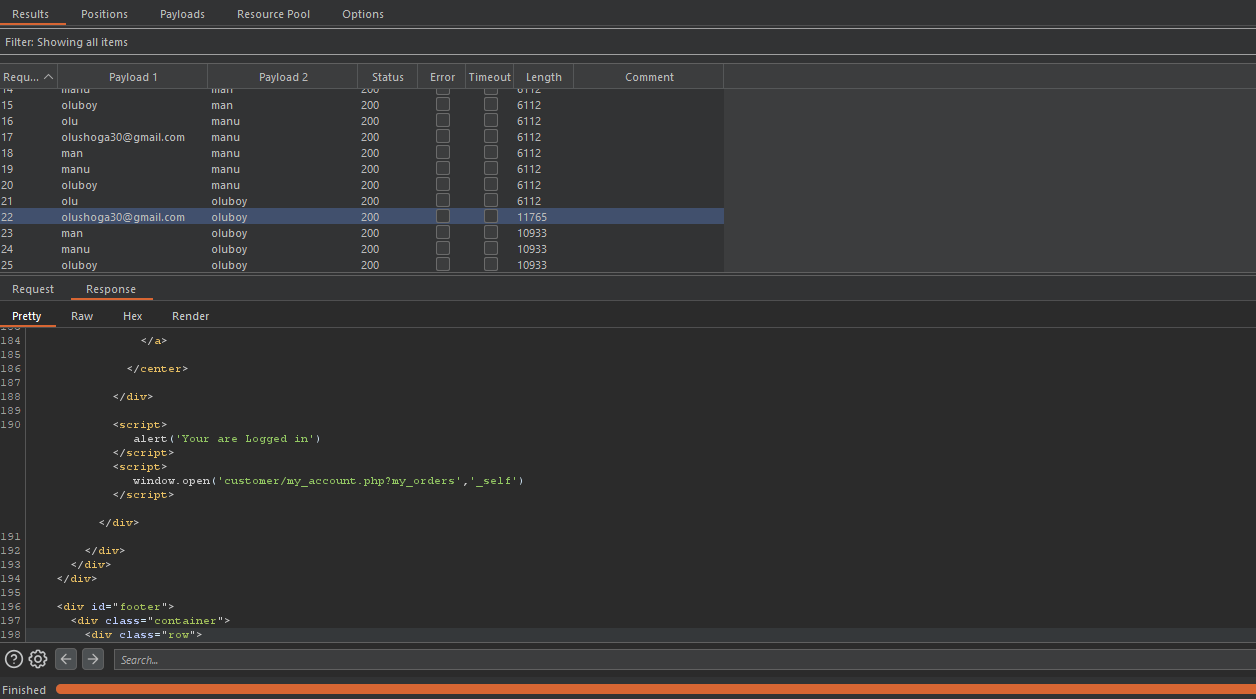
* A login is tried with random credentials, this sends an HTTP request and gives back a response from the server. The request is then sent to the intruder section.



* The payloads positons with the random login parameters and session ID cookie are cleared, then the random login parameters are highlighted and the Add button on the right side is clicked. Two payload sets are created for the username and password input bar respectively and a custom list of random and possible usernames and passwords are imported.



The attack is started and the combinations of the possible login parameters are tested. The request and response of each of these tries are reviewed and if successful, the details of a login attempt with the correct login credentials will appear.

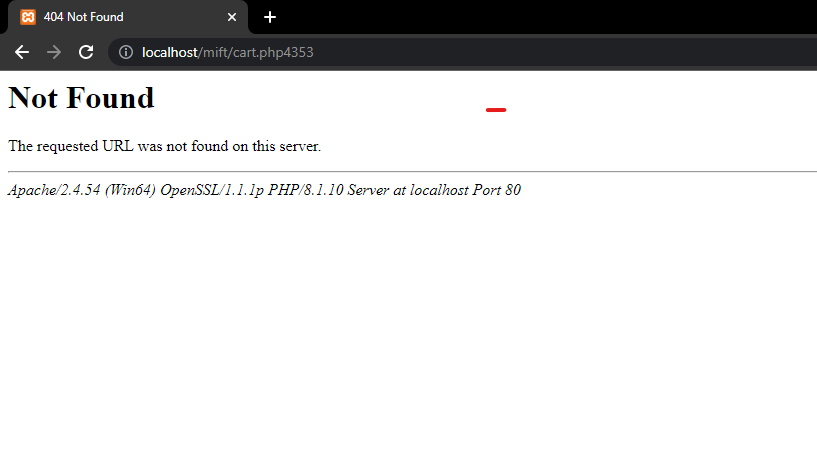


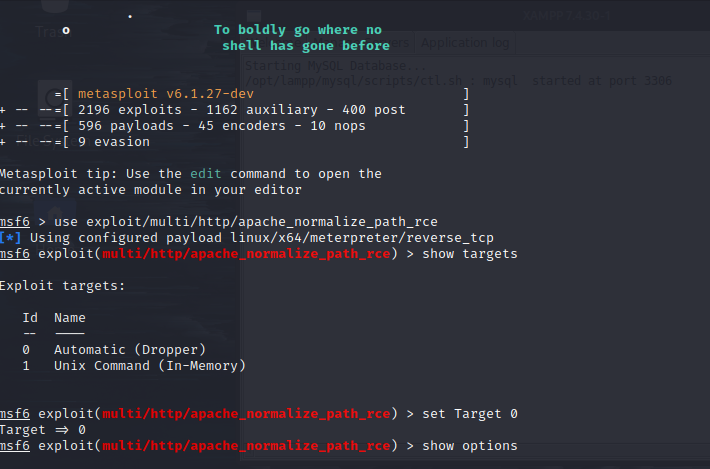
**Mitigation**

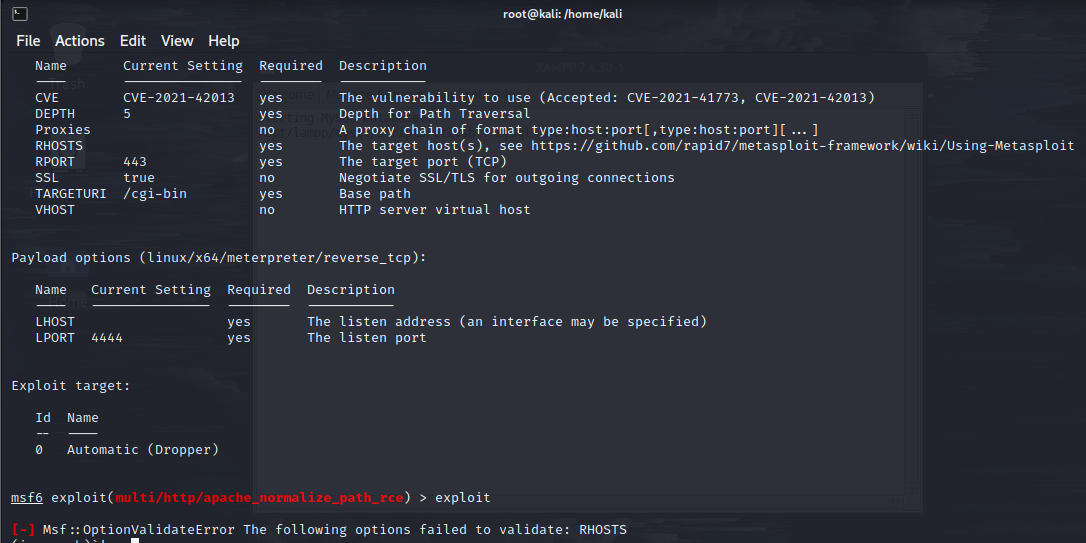
Insecure design can be mitigated through the use of efficient threat modelling, source code review of websites before being published onto the internet for the public and commercial purposes so as to consider potential attack vectors.

# A05 2021 Security Misconfiguration

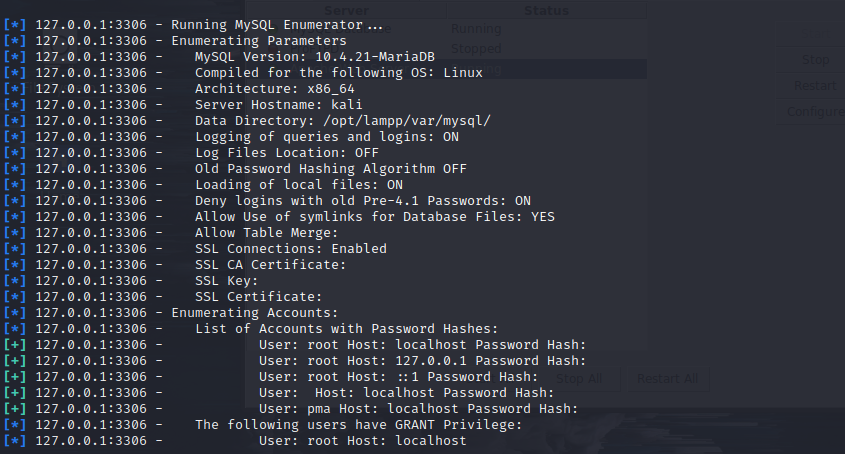
A scenario on the E-COM webpage is the 404 error message which reveals the Apache and MySql version of the hosting servers. This information however short and unimportant it seems is very sensitive as hackers can retrieve knowledge on how to exploit popular and common web server vulnerabilities from other websites like <https://www.rapid7.com> and <https://www.exploit-db.com> .

 Several Apache and MySQL server versions are known to contain bugs and flaws. An example is the traversal RCE exploitation with the metasploit framework of Apache 2.4.49/2.4.50 versions.

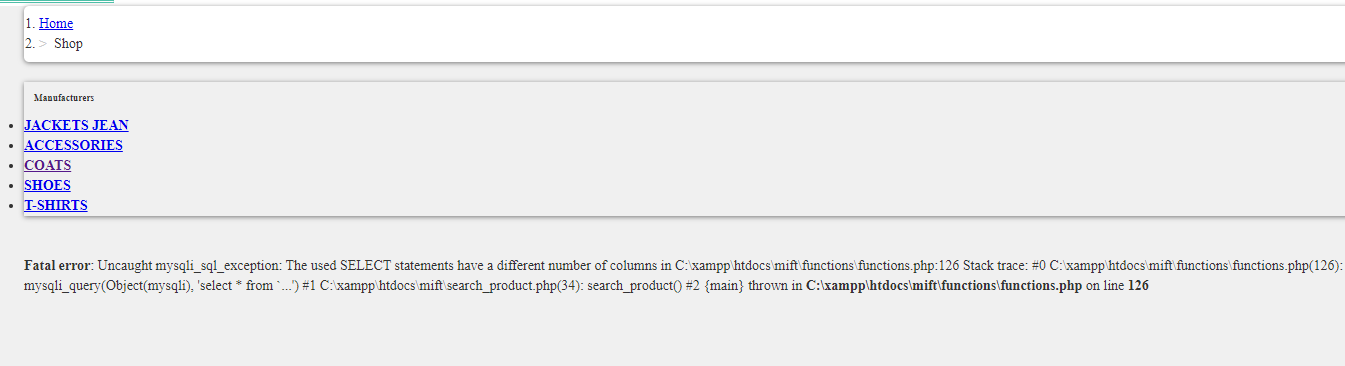




The database server can also be exploited to reveal detailed information and sensitive data about the database.



Another attack vector is the fatal error message in the cryptographic failure section revealing sensitive file paths on the website, this later leads to exposure of sensitive data.



**Mitigation**

Security misconfiguration vulnerabilities can be prevented by removing unused and unnecessary features, plugins and frameworks from the whole web infrastructure and restriction of stack traces and error messages from revealing sensitive information.

# A06 2021 Vulnerable and Outdated Components

Outdated versions of the Apache and MySQL servers have been known to be vulnerable to severe attacks like RCEs, SQL injections and authentication bypass. Targeting an outdated version of the Apache server; 2.4.50, a RCE flaw can be exploited and the version 2.4.52 contains a HTTP smuggling request vulnerability. MySQL database versions 5.7.14, 5.6.32 and 5.5.51 are known to contain RCEs and privilege escalation flaws

**Mitigation**

To prevent cyber attacks through vulnerable and outdated components, web traffic should be inspected thoroughly, proper secure design models should implemented, regular updates and maintenance of web infrastructure and use of Website Application Firewall.

# A07 2021 Identification and Authentication Failures

An exemplary reference to identification and authentication failures is the bruteforce attack on the E-COM website in the insecure design section, this is due to the website not being programmed to limit the numbers of login tries possible.

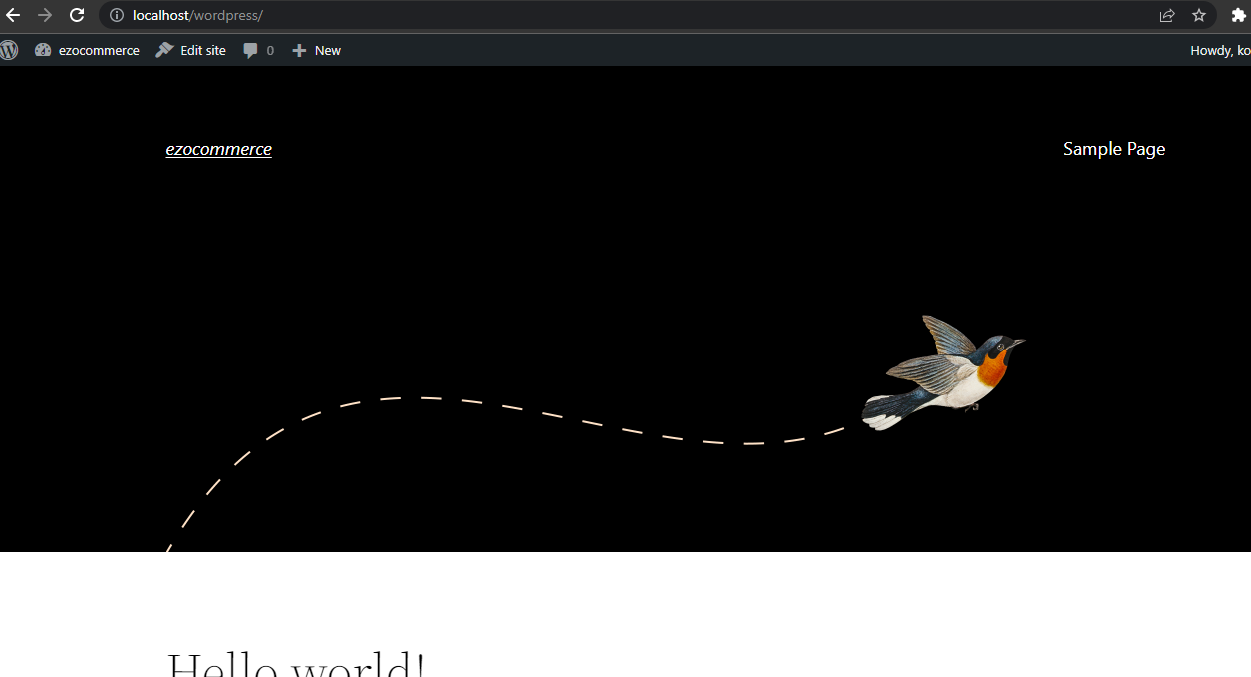
Another flaw in the website is the use of weak passwords both by the user and admin login pages and the usage of no multi-factor authentication mechanism

**Mitigation**

Identification and authentication failures can be prevented by efficient rate limiting of web logins, proper two-factor authentication, implementation of password check systems to prevent users from choosing common and weak passwords, making admin web login pages unpredictable to expose them to potential attacks and use of secure built-in server-side session managers to randomize session ID tokens frequency.

# A08 2021 Software and Data Integrity Failures

The vulnerability in regarding this doesn’t pose a direct threat to the E-COM website but through a separate wordpress site ezocommerce hosted on the same server. Wordpress has been known to utilize a wide range of web plugins to optimize website creation and management.



A couple of these extension are from untrusted sources and malicious, examples are the wp-sleeep and wp-zzz plugins. The wp-sleeep plugin redirects to malicious scam sites, infects the website database and infects important javascript and PHP source codes all because a backdoor generator was included in the plugin codes. In extreme cases, infiltrations like this pose a huge threat to the E-COM website and the server hosting the two websites generally.

**Mitigation**

Use of trusted repositories, digitally signed software by verified vendors and regular checking of frameworks, plugins and software for vulnerabilities can prevent software and data integrity failures.

# A09 2021 Security Logging and Monitoring Failures

With respect to the bruteforce attack in the insecure design section, a proper log file would have shown the number of tried logins before the user account was accessed either in the server files or in the admin webpage, regular inspection of these log files would have alerted the web management personnel and prompted the initiation of preventive and corrective measures to fix the flaw. In the case of the E-COM website, these log files don’t exist and this poses a huge threat to the security and well-being of the website.

**Mitigation**

To enact an efficient logging and monitoring mechanism for web applications, audit logs should be created record events, errors and anomalies on the website. However when creating the audit log, a lot of factors like accuracy, content and data sensitivity of the log, the duration that the logs should kept and how they should be reviewed should be considered.

# A10 2021 Server-Side Request Forgery

CVE-2021-40438 is a SSRF vulnerability on Apache servers of versions 2.4.48 and earlier. This flaw can be exploited if the mod-proxy is enabled and be used by attackers to channel modified requests to origin server of their choice.

Another popular method of exploitation is through the check stock functionality on commercial websites. The HTTP request sent by clicking this button returns a response after accessing an internal system to let users know the units of products available in stock. Interception of this response gives access for the stockAPI parameter to be manipulated and this can result in granting certain privileges to attackers. In the case of the E-COM website, the check stock functionality was deemed unnecessary as it showcases a form of bad User Interface/Experience.

## Mitigation

Further exploitation of SSRF can lead to RCE, DDOS, modification, edition and deletion of server data. This can be prevented by sanitization of all user input, and restriction of all insecure and unencrypted redirections.

# 