



Student Name: Kerlws Youssef

Student Number: 1808236

Module Code: 6CS007

Module Name: Project and Professionalism

Project Title: Web Security

Module Leader Name: Alix Bergeret

Supervisor Name: Alix Bergeret

Reader Name:

Submission Date: 26<sup>th</sup> April 2024

(The date you handed in the assessment)

**Award Title :** BSc (Hons) Computing and Information Technology

(Award Title for your project, if in doubt refer to your course/Module Registration)



## Declaration Sheet

Presented in partial fulfilment of the assessment requirements for the above award.

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work. I confirm that the intellectual contents of the work are the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free licence to do all or any of those things referred to in section 16(i) of the Copyright Designs and Patents Act 1988. (viz: to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make an adaptation of the work).

Student Name (Print): Crolos Masaud

Student Number: 1808236

Signature: ..... Date: .....

(Must include the unedited statement above. Sign and date)

Please use an electronic signature (scan and insert)

## TABLE OF CONTENTS

SECTION 1 STUDENT DETAILS .....	1
1.0 STUDENT NAME .....	1
1.1 STUDENT NUMBER .....	1
1.2 MODULE CODE.....	1
1.3 MODULE NAME .....	1
1.4 PROJECT TITLE .....	1
1.5 MODULE LEADER NAME .....	1
1.6 SUPERVISOR NAME.....	1
1.7 READER'S NAME .....	1
1.8 DECLARATION SHEET .....	2
 SECTION 2 STATEMENT OF PROJECT DETAILS .....	5
2.1 DECLARATION SHEET .....	6
2.2 ACADEMIC QUESTION.....	6
2.3 AIMS .....	6
2.4 OBJECTIVES .....	6
2.5 ARTEFACT (PROPOSED) TO BE DEVELOPED .....	6
 SECTION 3 PROJECT PROPOSAL .....	7
3.1 INTRODUCTION.....	7
3.2 INITIAL RESEARCH INTO SOURCE OF INFORMATION .....	8
3.3 SECURITY VULNERABILITIES.....	8
3.4 SECURITY MEASURES .....	8
3.5 CROSS-SITE VULNERABILITIES .....	8
3.6 VULNERABILITIES IN PHP .....	9
3.7 SQL INJECTIONS IN PHP APPLICATIONS .....	9
3.8 SECURITY RULES IN MYSQL RELATING TO GDPR.....	9
3.9.1 ARTEFACT AND METHODOLOGY .....	9
3.9.2 PLAN/SCHEDULE .....	9
3.9.3 REFERENCES AND BIBILOGRAHPY.....	10
3.9.4 ADDITIONAL SOURCES OF INFORMATION .....	11
3.9.5 ETHICAL CONSIDERATION FORM .....	11
 SECTION 4.0 VULNERABILITES IN WEB APPLICATIONS LITERATURE REVIEW .....	12
4.1 ABSTRACT.....	12
4.2 INTRODUCTION .....	12
4.2.1 SECURITY THREATS AND VULNERABILITES IN WEB APPLICATIONS .....	13
4.3 SQL INJECTIONS .....	13
4.4 TYPES OF SQL INJECTION .....	13
4.5 SQL INJECTION THREATS .....	13
4.6 SQL MAP TOOL.....	14
4.7 CROSS SITE SCRIPTING .....	15
4.8 SECURITY SOLUTIONS FOR WEB APPLICATION .....	15
4.9.1 ENCRPTION TECHNIQUES .....	16
4.9.2 WEB APPLICATION FIREWALL .....	17
4.9.3 WEB APPLICATION VULNERABILITY SCANNER.....	17
4.9.4 ILLUSTRATION OF WEB APPLICATION VULNERABILITY SCANNER .....	19
4.9.6 CONCLUSION.....	19
4.9.5 REFERENCES .....	20

SECTION 5.0 ARTEFACT DESIGN .....	21
5.1 REQUIREMENT ANALYSIS .....	21
5.2 PURPOSE OF THE SECURE DATABASE DRIVEN WEB APPLICATION .....	21
5.3 APPLICATION REQUIREMENTS .....	21
5.4 REQUIREMENTS TO BE MET .....	21
5.5 USABILITY .....	21
5.6 USER-FRIENDLINESS .....	21
5.7 TRACEABILITY OF REQUIREMENTS .....	21
5.8 REQUIREMENT CHECKLIST .....	22
5.9.1 PROJECT DESIGN: WIREFRAME MODELS .....	23
5.9.2 HOMEPAGEMODEL .....	23
5.9.3 REGISTER MODEL .....	24
5.9.4 LOGIN MODEL .....	25
5.9.5 PRODUCT LISTING MODEL .....	26-27
SECTION 6.0 WEB APPLICATION INFRASTRUCTURE .....	28
6.1 UML DIAGRAMS .....	28
6.2 ENTITY DATABASE DIAGRAM .....	28
6.3.1 USERS TABLE .....	28
6.4.2 MENS AND WOMEN'S CLOTHES TABLE .....	28
6.4.3 CHECKOUT TABLE .....	28
6.5 USER INTERFACE DIAGRAM .....	29-30
SECTION 7.0 ARTEFACT TESTING .....	30
7.1 TESTING METHODOLOY .....	30
7.2 VULNERABILITY SCANNER METHODOLOGY .....	30
7.3 WEB APPLICATION FIREWALL METHODOLOGY .....	30
7.4 ENCRYPTION TECHNOLOGIES .....	30
SECTION 8.0 IMPLEMENTATION AND DEVELOPMENT .....	31
8.1 INTRODUCTION .....	31
8.2 WINDOWS SERVER 2019 BASE EC2 INSTANCE .....	31
8.3 AMAZON WEB SERVER EC2 .....	31
8.4 CONFIGURING EC2 INSTANCE .....	32
8.5 RPD PROTOCOL .....	33
8.6 EC2 KEY PAIRS .....	33
8.7 HTTPS AND SSL PROTOCOL .....	34
8.8 INSTALLING XAMPP TO EC2 INSTANCE .....	35
SECTION 9.0 PROJECT ARTEFACT .....	36
9.1 HOMEPAGE .....	36
9.2 REGISTRATION .....	37
9.2 PHPMYADMIN DATABASE .....	37
9.3 USERMODEL.PHP .....	37
9.4 PHP PASSWORD HASH .....	38

SECTION 10.0 STEPS TO SETTING UP AMAZON WEB APPLICATION FIREWALL .....	6
10.1 CONFIGURING VPC .....	6
10.2 VPC INTERNET GATEWAY .....	6
10.3 VPC SUBNET .....	6
10.4 ROUTE TABLE .....	6
10.5 ADDING CAPTCHA RULES ON AWS WAF .....	6
10.6 CAOTCHA VALIDITY PEROID .....	6
10.7 ADDING PROTECTION AGAINST SQL INJECTIONS RULE.....	6
10.8 ADDING RULES FOR PHP APPLICATION .....	6
10.9 REFERENCES.....	6

## Section 2: Statement of Project Details

### 2.1 Project Title

Web Security

### 2.2 Academic Question

What are the potential security risks and solutions associated with web applications?

### 2.3 Aims

- Analyse all the security risks and methods to defend against discrepancies.
- Design and implement a database driven website for storing confidential information.
- Use SSL and HTTPS for a secure connection between the client and the user.

### 2.4 Objectives

- Design a secure online e-commerce store with a secure registration and login system.
- Test the application using a vulnerability scanner to assess the level of security.
- Host the web application on AWS Cloud Server and utilize AWS security features.

### 2.5 Artefact (proposed) to be developed.

An online prototype e-commerce store will be developed as the project artefact. It will be a highly functional and secure for potential online customers for a registration and log in system to keep the user's information secure from security threats.

A database will be implemented for storing information about the products, prices, customer's details, usernames, and password.

The web application will be hosted on the Amazon Web Service cloud machine and a test-plan will be created to evaluate the results from a web application vulnerability scanner to measure the security of risks and provide solutions to tackle any discrepancies.

## Section 3: Project Proposal

### 3.1 Introduction

The prevalence of the World Wide Web makes websites and their visitors an attractive target for various types of cybercrime including data breaches, spear phishing campaigns, ransomware, and fake technical support scams. (Hsiu-Chuan Huang et al, 2017).

Organizations are relying on web sites to promote and market themselves to produce revenue and profit which make these companies a target to online threats such as Viruses, malware, and worms. These threats are dangerous and are very harmful to organizations as it can cause a negative impact with its operations.

Third party modules or components can have security vulnerabilities that developers are aware of. Vulnerability scanners are useful for testing out applications for the causes of security by, detecting a security gap which may need fixing before realising the full version of application. Companies can face problems relating to financial issues due to their budget which often results in forgetting the key essential security components.

The project will investigate the security techniques required to develop a secure online e-commerce web application. It will analyse the security risks of how the user's information is kept safe from unauthorised access via data transmission between the client and the application. Research will also involve the technical characteristics of security risks. The solutions for a secure web application will be investigated.

A design and the layout of the online e-commerce application will be created including all security features with a description of their functions. The significance of designing a user-friendly website will remain. The application will primarily be written using HTML, CSS, PHP, and MySQL.

Confirmation if the project succeeds its requirements is determined upon an appropriate test plan. The testing of the artefact will be a regular process during design and development. A test plan approach will be constructed to fathom the success of the final artefact. The final evaluation of the project will analyse on where project succeeded based on the level of security and if additional work may be needed and how the artefact could be improved.

### 3.2 Initial Research into Source of Information

#### Security Vulnerabilities

According to (Galluccio, E. et al, 2020) there are malicious forms of attacks against computers worldwide. Their purpose is to gain control of computers by finding vulnerabilities of technologies, protocols, frameworks, and in the application's infrastructure. The most common method is known as SQL injections for exploiting the syntax of a language used in databases. SQL stands for Structured Query Language are used to access unobtainable information present of a database, including usernames, and passwords for access services in an application.

Attacks against web applications are trying expose sensitive data by gaining unrestricted access by finding a weakness in application often the happens in the back-end systems. Denial of service attacks on the application level can achieve the same results as other methods of attacks against the software infrastructure (Stuttard D. & Pinto M, 2011).

#### Security Measures

Web Application Firewalls is represented as the most advanced firewall capabilities in the industry. They protect web applications by blocking malicious web traffic and application on the layer attacks known as Denial of service, SQL injection, and cross-site scripting (Palo Alto Networks). Web application firewall protects OSI layer 7 by preventing and blocking attacks on the HTTP protocol for displaying data and images to users. Vulnerability Scanners can detect any security discrepancies on the web application by scanning the URL for SQL injections, PHP syntax, DDoS, and cross-site scripting attacks.

#### HTTPS

Hypertext transfer protocol secure is used in web applications to encrypt data while data is in transmission between a web browser and a website. This is essential for data protection because the information such as bank account, addresses, and email addresses are protected from hackers stealing the confidential information. HTTP uses a protocol called Secure Socket Layer which provides authentication from a public key certificate and generates a secure session key to enforce privacy in transmission on a symmetric key. SSL protocol is integrated into an application for keeping data secure over the HTTP protocol between the client and the server. Applications with SSL enabled communicates with the client after the client sends a hello message. The server responds to the message which contains the server's certificate to the client to give the client access. The server's certificate provides the client with a public key for a validity period, and information about the owner of the server (Razumov, P. et al, 2023).

#### Amazon Web Services

AWS provides security services such as AWS, identity and access management, AWS shield, AWS network access control lists, security groups and Amazon guard duty. The identity and access management (IAM) is a web service for controlling access to AWS resources in a securely manner. IAM is responsible for controlling the users, roles, and groups to give access to resources and roles. AWS shield is affective by providing protection against DDoS attacks on the network. The security group includes a virtual firewall for allowing and denying traffic that coming inbound or outbound (Penwell T, 2023).

#### Cloud Computing with AWS

Cloud computing provides the right tools with structure and services for running and moving code to the cloud service on Amazon Web services. Users can deploy virtual computers with the most popular operating system on the market such as Windows and Linux. AWS promotes developers to develop, test and upload applications on to their cloud servers. Users can customize their virtual computer's memory, storage, and security groups according to their requirements. AWS is flexible and reliable for their robust cloud computing structure and servers located on their headquarters.



AWS can save tons of money for start-up companies who are looking build a network with computers, plus security is covered by their security policies and groups, by enabling users to create created security rules or use AWS default security rules. (Ifrah S, 2019)

### **Denial of Service**

The DDoS attack is an application layer attack which is a very dangerous cyber-attack created by attackers to interrupt operations by targeting services and sending infinite amounts of internet traffic to slow down the system, this can cause the users from accessing the web application and slow down business operations.

### **Amazon Web Services Elastic Containers**

Amazon Web Services Containers is an engine which is like Kubernetes in function however, AWS provides a container as a service allowing users to use microservices for running a single virtual private cloud. Amazon Elastic Containers are easily deployed and manged for scalable applications and provides configuration and operational tools.

### **Amazon Web Application firewall**

Amazon Web Application firewall provides security rules to protect the applications from attacks via SQL injections, cross site vulnerabilities and DDOS attacks by providing security rules to block any scripts from being injected and running from the attackers' computers. WAF detects the attacks and blocks any communications to the web application hosted on a EC2 instance. WAF another method of blocking DDoS attacks by blacklisting the attacker's IP address to prevent any communications to the EC2 instance.

### **Application layer attacks**

Application layer attacks are very dangerous as they are coming in many forms, OWASP is reliable source for tracking and trending application layer security vulnerabilities. The application layer has protocols such as Hypertext transfer protocol to transmit data between the web server and the client. Developers must test the applications before release by seeking any vulnerability gaps on the application through a vulnerability scanner to scan the application for any security holes in the application. (Russell C, 2018)

### **How Web Applications work**

Most web applications on the internet are database driven applications with a back-end database with pages that run on server-side script writing in a program language for dynamic interactions with the user and extracts information on the database to display to the user interface. E-commerce web applications are very common database driven web applications and stores an enormous amount of data to the database such as product name, description, prices, and stocks. Most database driven web applications uses PHP for creating back-end scripts to connect, retrieve, update, delete and display data from database such as MySQL and oracle.

### **Cross-site Vulnerabilities**

Cross-site scripting vulnerabilities have been known since 1996 in World Wide Web time where e-commerce started to become big in the web technology world.

The first ever major XSS worm was able to shout down on the most popular social networking web site Myspace. The virus was able spread from a single Myspace user profile page to another, which ended up infecting more than a million users in 24 hours. This caused studies on the security world to take place and research into JavaScript malware (Grossman J, 2007).

XSS can create serious security vulnerabilities for both the web site and the user. The attacker would damage a system by injecting a malicious code into a section of the application where the application accepts user input, and if the user's input is not validated, the code can transfer private information, hijack a user's account, and cause denial of service. There are three types of XSS attacks which are named reflected, stored and DOM-based. Reflected XSS are executed by the victim's browser and runs when the victim gives an input to the website. Stored XSS attacks store malicious scripts into databases, message forums, and comments fields. The malicious script is executed by the visiting user which causes the user to pass their privileges to the attacker. The reflected and stored XSS are run on the server side. Dom-based XSS attacks are run on the client-side (Hydara, I. et al, 2015).

### **Vulnerabilities in PHP**

PHP is the most popular server-side web programming language, According to W3Techs. Majority of web applications are run on PHP. PHP scripts are shared with other web applications, so it is crucial for the application to be secure and have no vulnerabilities for enforcing security on the application. This means if a part of application with the PHP scripts does indeed have any vulnerabilities, then this can have a negative effect to the rest of the application. Weak input validation is a PHP vulnerability which can be prone to SQL injection and cross-site scripting. PHP object injection is an application-level vulnerability which can allow attackers to send malicious scripts such as SQL injections, path traversal and Application Denial of service.

### **SQL Injections in PHP**

SQL injection is known as one the most dangerous web application vulnerability when developing web applications in PHP. If the user input to the SQL query is invalid, this can give an attacker a chance to manipulate the query itself which, will give the attacker control of the script and possibly breach the whole system. If the SQL Injection attack is successful, this can also mean the confidential such as customer's payment details is at risk of data theft.

### **Security Rules in MySQL relating to GDPR.**

The General Data Protection Regulation is created in the European Union for enforcing privacy law and human rights law. The data in a MySQL is highly confidential and must protected by monitoring security and preserving data privacy. Encryption rules is a regulation by the GDPR to use proper encryption techniques which should be used for security purposes and maintenance.

### **XAMPP**

XAMPP is open-source web hosting application which is responsible for hosting web application to a local server and provides a MySQL database server for storing and retrieving data for the application. The MySQL database enables operations such as create databases, modify tables, create columns, establish relationships, create indexes (primary and foreign keys), and set user permissions. XAMPP provides a PhpMyAdmin which is a tool written in PHP and MySQL to use features to browser and drop databases, tables, views, fields, and indexes. Data can be imported from CSV and SQL files to the database for creating tables and filling in data to the database.

### **CodeIgniter 4 Framework**

CodeIgniter 4 is web application framework for developing dynamic web applications, web services, and web APIs because the framework provides libraries, components, templates, database integration. The framework is heavily used by PHP and provides great performance for applications. The web application artefact will be created using the CodeIgniter 4 framework and for the development of the artefact. The framework will be used to connect the MySQL database to the web application for fetching and storing data from the user interface. The framework is Model, View, and Controller software architectural pattern. The model's purpose is to interact with the database to select a table and fields with the primary key from the database to interact with. The controller is the

control logic for binding the model and view together to route the logic together to display results. The view is displaying the data and results via a user graphical interface to users. CodeIgniter 4 Frameworks provides XSS filtering security which prevents any malicious JavaScript code that want to hijack cookies to complete malicious activities. CSRF protection is a CodeIgniter functionality to protect the user's cookie session by regenerating new CSRF cookies to protect the user's session from cross site scripting.

### **3.3 Artefact and Methodology (proposed)**

The final artefact will be a prototype e-commerce web application which will be heavily secure using encryption and firewall techniques to provide a secure login and registration system to protect confidential information such as customer's payment details, address, and identity.

The artefact is a method of answering the question "What are the potential security risks and solutions associated with web applications?". The use of this artefact will be investigated to find out if the security solutions is appropriate through tests and evaluations. The artefact will tackle all the security threats based on the research taken place for the final artefact to be secure.

The database will be built for storing customer's information through the development process of the secure online application using AWS security features. Ensuring the connection is secure by using encryption technologies accredited by the research taken place for solutions to prevent any gaps which may be a security vulnerability. XSS scripts can affect the website in negative by creating security issues so the application will have strong technologies to defend against XSS scripts and SQL injections.

A test plan will be conducted at the development and testing stage to ensure the functionality, usability and security of the application is meeting the requirements which will be used to write an evaluation of the application. A vulnerability scanner will be used to scan the web application to analyse and evaluate if the application is secure and does not have any vulnerabilities which can be prone to SQL injection attacks, DDoS attacks and Cross site scripting attacks.

### 3.4 Plan/Schedule

Milestone description	Assigned to	Progress	Start	Dags
Milestone 1 Project Proposal	Crolos Masaud			
Student Details	Crolos Masaud	100%	05/MO/2023	1
Table of Contents	Crolos Masaud	100%	05/MO/2023	1
Statement of Project Details	Crolos Masaud	100%	05/MO/2023	1
Project Proposal	Crolos Masaud	100%	05/MO/2023	7
References and Bibliography	Crolos Masaud	100%	05/MO/2023	7
Milestone 2 Literature Review				
Literature Review	Crolos Masaud	0%	17/MO/2023	14
Artefact Design and Test Plan				
Designing the Web Application	Crolos Masaud	0%	01/MV/2023	25
Creating Database	Crolos Masaud	0%	02/MV/2023	25
Implementing security techniques	Crolos Masaud	0%	03/MV/2023	14
Testing the applications	Crolos Masaud	0%	16/MV/2023	3
Professionalism Report				
Social Impact of Artefact	Crolos Masaud	0%	19/MV/2023	15
Ethical issues of Artefact	Crolos Masaud	0%	20/MV/2023	1
Legal implications	Crolos Masaud	0%	21/MV/2023	1
Security aspects of Artefact	Crolos Masaud	0%	22/MV/2023	1
Draft report	Crolos Masaud	0%	24/MV/2023	14
Advanced version of artefact	Crolos Masaud	0%	01/JZ/2023	14

### 3.5 References and Bibliography

Hsiu-Chuan Huang et al. (2017) Web Application Security: Threats, Countermeasures, and Pitfalls. Computer (Long Beach, Calif.). [Online] 50 (6), 81–85.

McDonald, M. (2020) *Web Security for Developers: real threats, practical defense*. 1st edition. San Francisco: No Starch Press.

Liu, Lin et al. (2016) A Website Security Risk Assessment Method Based on the I-BAG Model. *China communications*. [Online] 13 (5), 172–181.

Hsiu-Chuan Huang et al. (2017) Web Application Security: Threats, Countermeasures, and Pitfalls. Computer (Long Beach, Calif.). [Online] 50 (6), 81–85.

Galluccio, E. et al. (2020) *SQL Injection Strategies*. 1st edition. Packt Publishing.

Stuttard, D. & Pinto, M. (2011) *The web application hacker's handbook : finding and exploiting security flaws*. Second edition. Indianapolis, IN: John Wiley & Sons, Inc.

Razumov, P. et al. (2023) Ensuring the security of web applications operating on the basis of the SSL/TLS protocol. E3S web of conferences. [Online] 4023028–.

Palo Alto Networks. (n.d.). What Is a WAF? | Web Application Firewall Explained. [online] Available at: <https://www.paloaltonetworks.co.uk/cyberpedia/what-is-a-web-application-firewall>.

Penwell, T. (2023) *Beginning AWS Security: Build Secure, Effective, and Efficient AWS Architecture*. 1st edition. [Online]. Berkeley, CA: Apress L. P.

Grossman, J. & Grossman, J. (2007) XSS attacks cross-site scripting exploits and defense. 1st edition. Burlington, MA: Syngress.

Hydara, I. et al. (2015) Current state of research on cross-site scripting (XSS) – A systematic literature review. *Information and software technology*. [Online] 58170–186.

Steinke, G. et al. (2011) Towards an Understanding of Web Application Security Threats and Incidents. *Journal of information privacy & security*. [Online] 7 (4), 54–69.

Šušter, I. and Ranisavljević, T. (2023). OPTIMIZATION OF MYSQL DATABASE. *Journal of Process Management. New Technologies*, 11(1-2), pp.141–151. doi:<https://doi.org/10.5937/jpmnt11-44471>.

Russell, C. (2018) *Web application firewalls : securing modern web applications*. First edition. Sebastopol, CA: O'Reilly Media.

Mateo Tudela, F. et al. (2020) On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications. *Applied sciences*. [Online] 10 (24), 9119–.

Ifrah, S. (2019) *Deploy Containers on AWS: With EC2, ECS, and EKS*. 1st ed. [Online]. Berkeley, CA: Apress L. P.

Russell, C. (2018) *Web application firewalls : securing modern web applications*. First edition. Sebastopol, CA: O'Reilly Media.

phpMyAdmin (2019). *phpMyAdmin*. [online] phpMyAdmin. Available at: <https://www.phpmyadmin.net/>.

www.codeigniter.com. (n.d.). *Welcome to CodeIgniter4 — CodeIgniter 4.2.1 documentation*. [online] Available at: [https://www.codeigniter.com/user\\_guide/intro/index.html](https://www.codeigniter.com/user_guide/intro/index.html).

## Section 4: Additional Sources of information

### 4.1 Ethical Considerations Form

To be submitted via Canvas

## Section 4.0 Vulnerabilities in web applications and solutions for PHP Web Applications: Literature Review

**Abstract.** The attacks of web sites continue to grow every year with this year resulting 3,808,687,191 compromised record incidents in September 2023 according to the itgovernance dataset (Irwin L, 2023). This literature review is about using security technologies such as web application firewall and vulnerability scanning methods as a solution to protect confidential data from web application vulnerabilities. This research will have past and present vulnerabilities in web applications in the world wide web and the solutions to solve these problems mentioned above. With uses of encryption techniques to provide extra security for the web application for protecting confidential data on the internet.

### Introduction

Currently, web sites hold over millions of confidential data which are substantial to cyber-attacks every year. Web developers must build secure web applications to ensure that data integrity is enforced by finding all the vulnerabilities that an application may have by using a vulnerability scanning methods to pinpoint or the penetration method to discover any security holes. Web applications are used in our everyday lives with the use of social media, entertainment and educational purposes holding users' information which must be protected. Hackers will maliciously build scripts to try find a backdoor to the system to steal or intercept data. The high increase of web applications can possibly have errors and vulnerabilities in web applications because of the low awareness of the importance of creating secure web applications, enhance automated tools are key and essential for detecting the vulnerability in web applications. (Amankwah R, 2020).

Web database driven applications are in danger to SQL injections and DDoS. It is essential that developers test their applications and implement security defences such as web application firewall and effective coding scripts. Attackers are on the lookout for finding any vulnerabilities with back-end scripts to find a back door access to the application and damage systems. Security measures such as web applications firewall is effective in detecting and blocking the attacks to keep web application secure. Developers must test if the application is secure with the use of vulnerability scanners to prevent any security holes on the system.

## Security threats and vulnerabilities in web applications

### SQL injections

A very common web-based vulnerability is known as SQL Injection attack which allows attackers to insert malicious SQL queries to penetrate databases. The attacker's goal is to steal sensitive data from the web application. The attack process confines with an attacker injecting a malicious SQL query directly to the web application. Once the script is injected then sensitive data from the database will be extracted to the attacker. (Gupta S, 2020).

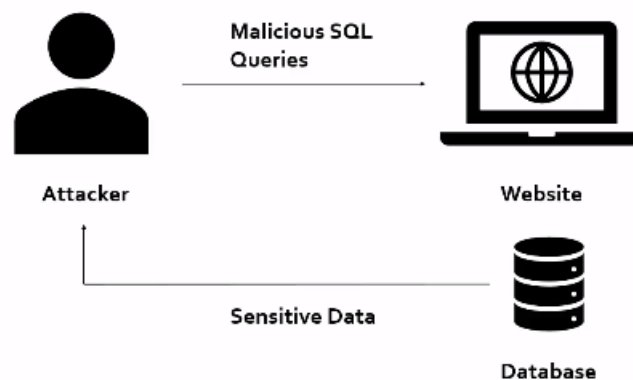


Illustration Figure 1: Gupta, S. (2020) SQL Injection Attacks: Protect Your System from Vulnerabilities.

### Types of SQL Injections

There are two types of SQL injection. The first one is a simple SQL injection which is built by linking coded string with the string entered by the user. The attacker can run the SQL injection through a union SQL command. The second type of SQL injection is known as Blind SQL injection which are used to gain access to sensitive information in database by requesting an array of true or false queries. (Gupta S, 2020).

### Simple SQL Injection

Simple SQL injection attacks a web database driven database using strings of malicious code to a specific database query. SQL injection is the most common attacks to web applications by exploiting input fields such as login forms, search boxes and URL parameters.

### **Blind SQL Injection**

Blind SQL injection sends true or false questions to determine the application's response and the attack often happens when the web application's code is vulnerable to SQL injection. The application displays error messages from the database to indicate the SQL query syntax is wrong. Blind SQL injections HTTP responses do not contain the results from the SQL query or provide any details of errors from the database. Boolean Blind SQL Injection attacks send a SQL query to the database to fetch results by promoting the application. The result is returned either by truth or false and the information can be modified within the HTTP response. Time based is another Blind SQL injection which forces the database to wait for a period before the database reacts. The attackers wait for the database to respond to the query if it is true or false. The results come from the HTTP response, and will either be created as soon as possible or it will take a while. The attacker will find out if the script returns a true or false response.

### **Out of band SQLi**

The out of band SQL injection is another form of SQL injection where the attacker does not get a response from the application through the same channel, so the script makes the application send data to their computers to receive a response from the application's database. Out of band SQL injection attacks happen if the server uses commands to trigger DNS or HTTP requests. This attack also relies on the database server to create HTTP request for delivering data to the attacker to control the database and possibly sabotage the whole database.

### **Error-based SQLi**

Error-based SQL injection is another form of in-band SQL injection method which waits for error messages from the database server to find out information on the database structure. This type of SQL injection method can specify the fields of the database. Prevention to stop this type of SQL injections, is by disabling errors on the live web application and using a file to log errors with restricted access.

### **SQL Injection Threats**

SQL injection attacks rely on some weak validation of textual input used to build database queries. Malicious created inputs are designed to threaten the confidentiality and security policies of Web sites that are database driven web applications. PHP web applications use SQL queries throughout the application which can be targeted and vulnerable to SQL injections attacks. This can be due to weak validation of the textual input used to compose SQL queries. The malicious inputs contain SQL instructions to produce queries which is different in terms of semantics to the designer's code and can cause discrepancies to the security policies to the database. It is known that a SQL injection has exposed 40 million credit cards from CardSystems, Inc. The attacker has sent malicious scripts to steal confidential data from the database and send the data through a File Transfer Protocol Server every four days. (Merlo E, 2006).

SQL databases hold sensitive data such as customer's address, payment details and their name and can lose the confidentiality once the application encounters a SQL injection attack. The authentication of the database can be lost due to poorly written SQL commands used to check usernames and passwords, which can be used to connect to a system pretending to be a user without the knowledge of the password. Integrity is a massive loss to databases because the hacker can read sensitive information and, can even make changes of delete information from a SQL Injection attack. SQL Injections are very common and often database driven web applications have suffered against SQL injection attacks. The security flaws on the database driven web applications can be detected for exploitation, through a software package.

SQL injections can give attackers power to steal innocent user's information and change the data on the database potentially destroying organisations procedures. PHP applications is a big target for SQL injections attacks due to most PHP applications uses MySQL database and scripts.

## SQL Map tool

```
File Actions Edit View Help
Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: artist=-9218 UNION ALL SELECT CONCAT(0x7162786b71,0x715561767765624555646864735151616f6b67414159637a6a795346764a72456d56714e67646876,0x7162766271),NULL,NULL-- --
----
[08:39:00] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12
[08:39:00] [INFO] fetching entries of column(s) 'uname' for table 'users' in database 'acuart'
[08:39:01] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[08:39:01] [INFO] fetching number of column(s) 'uname' entries for table 'users' in database 'acuart'
[08:39:01] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[08:39:01] [INFO] retrieved: 1
[08:39:04] [INFO] retrieved: test
Database: acuart
Table: users
[1 entry]
+-----+
|  uname  |
+-----+
|  test   |
+-----+

[08:39:11] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[08:39:11] [INFO] fetched data logged to text files under '/home/kali/.sqlmap/output/testphp.vulnweb.com'
[08:39:11] [WARNING] you haven't updated sqlmap for more than 140 days!!!
[*] ending @ 08:39:11 /2020-08-21/
```

Illustration Figure 2: Gupta, S. (2020) SQL Injection Attacks: Protect Your System from Vulnerabilities.

SQL map tool is a SQL injection method to target a specific website to perform an SQL attack. This tool is used on command lines and terminals for Linux to find out the database tables and columns to fetch confidential data such as usernames, passwords, addresses and credit card information in a web application. SQL map tool is an open-source python-based tool to find SQL injection vulnerabilities for exploitation for future SQL Injection attacks. SQL map tool can detect SQL injections vulnerabilities in a target application by writing a command with the URL, which can be used to fix the security holes that can prevent future SQL Injection attacks. (Gupta S, 2020).

## Cross Site Scripting

Cross Site Scripting is used to get the user to click on a link which contains malicious JavaScript code. These malicious JavaScript can be sent by an e-mail from the hacker to a client or be planted into a public forum. The link on the forum can be disguised as something else to get the user to click on the link to catch their victims. The JavaScript code is trying take out the user's session cookie, once the process is complete, the attacker is successful in tricking the victim. The attacker can potentially steal money from the victims via the same user's session cookie if any bank details is stored on to the web browser. The Cisco 2018 Annual Security Report shows that all web applications do indeed have at least one vulnerability. The report indicates, 40% of the attacks comes from Cross-site Scripting (XSS), which is one of the most common used techniques. The OWASP ranks it as No. 7 and is presented in two thirds of all web applications. (Rodríguez G. E, 2020).



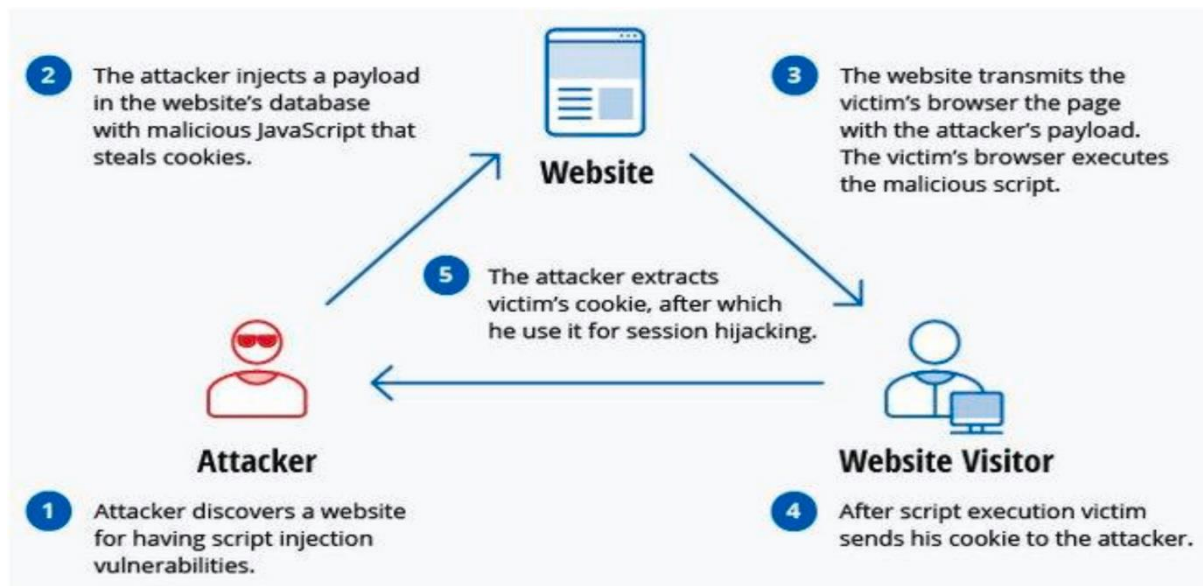


Illustration Figure 3: Abu Al-Haija, Q. (2023). Cost-effective detection system of cross-site scripting attacks using hybrid learning approach. Results in Engineering.

### Security Solutions for a secure web application

#### Encryption techniques

Encryption is used in the modern web applications for processing information into unreadable entities through data transmission between client and web application. This security technique is key for projecting information on the internet to stop hackers from attaining the confidential data. For the information to be encrypted, it requires two crucial pieces of data: the cipher and the key. The cipher is an algorithm for encrypting and decrypting data by accumulating symbols into a message of fixed size, and stream ciphers encrypts data by using continuous stream of symbols. Two major types of encryptions are symmetric key and asymmetric or public key. The method of the symmetric key encryption involves both the sender and the receiver to share a key which is used by the algorithm to encrypt or decrypt the data. The one-way encryption is a form of symmetric encryption where the message is encrypted with no decrypting in the transmission. The use of encryption on the web application will in fact provide a secure transmission for the protection of confidential information, enforcing security and data confidentiality to stop hackers from stealing user's information. (Tricia B & Bill B, 2009).

Figure 8.2. Diagram of symmetric encryption.

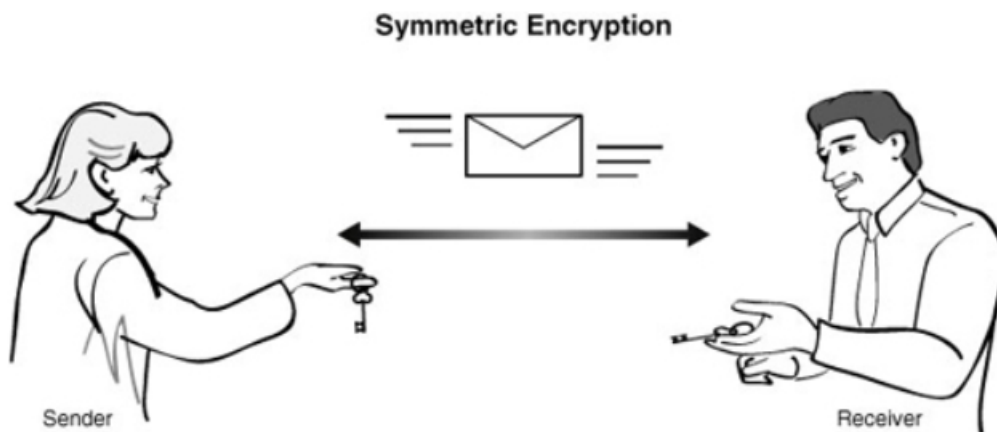


Illustration Figure 4: Tricia, B. and Bill, B. (2009). 8 Encryption - Securing PHP Web Applications

### Web Application Firewall

Amazon provides a Web application Firewall for blocking unauthorized traffic that are outbound or inbound and can also stop traffic requests to the web application to the Web application. The firewall is responsible for controlling the HTTP traffic between the web application and the internet. Web application firewall is excellent in defending a web application from cross-site-scripting, file inclusion, and SQL injections. The Web Application firewall is most the advanced version compared to a standard firewall due to extra layer of security called application-level filtering which a standard firewall does not provide. WAF is a barrier between the web application and the client on the internet because it protects the web server by detecting potentially dangerous traffic by acting as a reverse proxy. WAF is controlled by policies and provides pre-trained module to predict new incoming requests. The application will filter harmful communications and, the policies will protect the application by guarding the application against vulnerabilities. WAF uses intrusion detection systems in the application layer for security protection against SQL injections, cross site scrips, malicious php scripts and DDoS attacks (Dawadi B. R, 2023).

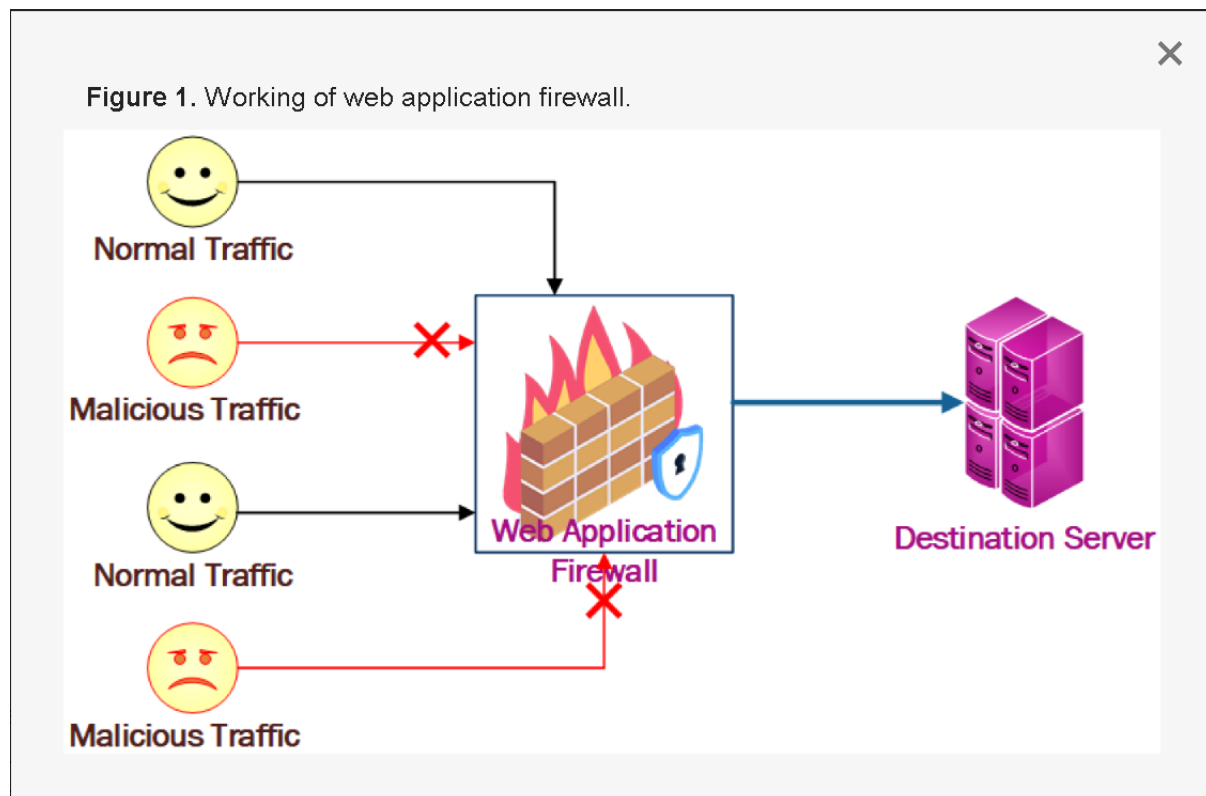
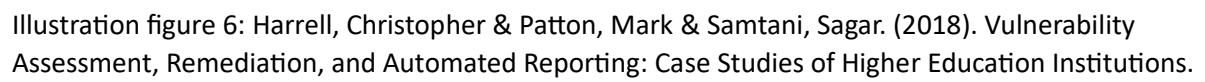


Illustration figure 5: Dawadi, B. R. et al. (2023) Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. Sensors.

### Web Application Vulnerability Scanner

A Web application vulnerability scanner is an automated tools for scanning web applications for security vulnerabilities which are often cross site scripting, SQL injections, Command Injection, and an insecure server configuration. WAVS consists of several activities that are crawling, fuzzing, and analysing. The crawler's purpose is to search for web pages in web applications. The WAV creates a simulation for attacks on the web application or scan for vulnerabilities. Web application vulnerability scanners provides logs and monitoring for data leaks and security gaps. (Al Anhar A & Suryanto Y, 2021).

An attacker may develop malicious for stealing confidential information which is highly unethical and perhaps finding out every vulnerability on the application is like acting as a hacker to stop an actual hacker from infiltrating the application. A vulnerability scanner is automated which can save developers a lot time with testing the application regardless of creating a system for a deep analysis on the application.



## **Conclusion**

### **Benefits of using a Vulnerability Scanner**

The vulnerability scanner is highly beneficial for finding out any security vulnerabilities which can aid the development of a secure applications by pinpointing and proving a solution to ensure the application does not have any vulnerabilities. It also provides results and descriptions of the vulnerability which can be evaluated to provide a solution. These vulnerabilities are used for cybercrime which involves invasion of privacy for user and cause a disturbance on software infrastructure. With the use of Vulnerability scanners, developers can use this software for preventing these discrepancies mentioned to ensure the application is secure especially with rules and regulations from GDPR, the application must be meet these guidelines. Vulnerability scanner can detect early weakness on the application by identifying the security weaknesses and flaws on the application to protect against malicious activities, this can reduce the risk of cyberattacks against the web application and against data breaches.

### **How to use Vulnerability Scanner?**

There are several vulnerability scanners which are open source or can be purchased. The most common web application scanners are Nessus, Nikto, OpenVAS, Nmap, W3af, Qualys, and Net. Vulnerability scanners are accessible via online such as debricked which allows users to add their repository from GitHub to the web application and scans the application for any vulnerability. It is an open-source program and easily accessible on the internet.

### **Why is encryption technology an effective technique for enforcing security on web applications?**

Encryption is key for protecting the confidentiality of data through data transmission between the client and the user which gives organizations peace of mind as the data is safeguarded in transit or in rest. Data protection legislation requires the integrity of data to be respected which is why encryption technology is essential for companies. Hackers will not be able to penetrate the application to steal data from organizations. HTTPS makes sure web application encrypts user's connection to the web application and protects the data that are being transferred to make sure the data is not being tampered or modified by hackers. The HTTPS makes sure that user is accessing the actual website and not a fake web application created by a hacker. (Software Quality, n.d.)

### **Benefits of Web Application Firewall**

Web Application Firewall will provide protection against malicious HTTPS traffic through filtering, monitoring, and blocking for securing web applications from hackers. Hackers will try to gain access to the application through using malicious traffic however, WAF has security barrier for preventing outbound and inbounds connections to other PCs or server which may be dangerous or compromise the whole application. AWS Web application firewall allows users to create rules for filtering web traffic based on the conditions of the rules which, includes the IP addresses, HTTP headers and body or custom URL. This means there is an extra layer of protection against web attacks that are looking to exploit vulnerabilities on the web application. (Amazon Web Services, Inc., n.d.)

## References

- Abu Al-Haija, Q. (2023). Cost-effective detection system of cross-site scripting attacks using hybrid learning approach. *Results in Engineering*, [online] 19, p.101266. doi:<https://doi.org/10.1016/j.rineng.2023.101266>.
- Altulaihan, E. A. et al. (2023) A Survey on Web Application Penetration Testing. *Electronics (Basel)*. [Online] 12 (5), 1229–.
- Al Anhar, A. & Suryanto, Y. (2021) 'Evaluation of Web Application Vulnerability Scanner for Modern Web Application', in 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). [Online]. 2021 IEEE. pp. 200–204.
- Dawadi, B. R. et al. (2023) Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. *Sensors (Basel, Switzerland)*. [Online] 23 (4), 2073–
- Gupta, S. (2020) SQL Injection Attacks: Protect Your System from Vulnerabilities.
- Harrell, Christopher & Patton, Mark & Samtani, Sagar. (2018). *Vulnerability Assessment, Remediation, and Automated Reporting: Case Studies of Higher Education Institutions*. 148-153. 10.1109/ISI.2018.8587380.
- Merlo, E. et al. (2006) 'Insider and Outsider Threat-Sensitive SQL Injection Vulnerability Analysis in PHP', in 2006 13th Working Conference on Reverse Engineering. [Online]. 2006 IEEE. pp. 147–156.
- Rodríguez, G. E. et al. (2020) Cross-site scripting (XSS) attacks and mitigation: A survey. *Computer networks (Amsterdam, Netherlands : 1999)*. [Online] 166106960–.
- Russell. (2018). *Web application firewalls : securing modern web applications (First edition.)*. O'Reilly Media.
- Sinha, S. (2019) *Bug Bounty Hunting for Web Security : find and exploit vulnerabilities in Web sites and Applications*. 1st ed. 2019. [Online]. Berkeley, CA: Apress.
- Singh, N. et al. (2016) 'SQL injection: Types, methodology, attack queries and prevention', in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). 2016 Bharati Vidyapeeth, New Delhi as the Organizer of INDIACom - 2016. pp. 2872–2876.
- Tricia, B. and Bill, B. (2009). *8 Encryption - Securing PHP Web Applications [Book]*. [online] [www.oreilly.com](http://www.oreilly.com). Available at: <https://learning.oreilly.com/library/view/securing-php-web/9780321574312/ch08.html> [Accessed 2 Nov. 2023]
- Amankwah, R. et al. (2020) An empirical comparison of commercial and open-source web vulnerability scanners. *Software, practice & experience*. [Online] 50 (9), 1842–1857.
- Irwin, L. (2023). List of Data Breaches and Cyber Attacks in 2023. [online] IT Governance UK Blog. Available at: <https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-2023>.
- Amazon Web Services, Inc. (n.d.). *Features - AWS WAF - Amazon Web Services (AWS)*. [online] Available at: <https://aws.amazon.com/waf/features/#:~:text=AWS%20WAF%20lets%20you%20create> [Accessed 9 Feb. 2024].
- Software Quality. (n.d.). *What is Hypertext Transfer Protocol Secure (HTTPS)?* [online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/HTTPS#:~:text=HTTPS%20encrypts%20the%20website%20v> isitor.

## **Requirement Analysis**

### **Purpose of the secure database driven web application**

The application is a prototype of an e-commerce web application with a registration and login form to allow users to register and login into the application with their registered credentials. The application will use HTTPS and SSL certificate to establish a secure connection between the client and the web server. The web application will be hosted on Amazon Web Services Cloud EC2 instance with a XAMPP local host server. AWS provides a web application firewall and will be used in the project to protect the application from DDOS attacks, SQL injections, and malicious PHP scripts by setting rules provided by the WAF.

### **Application's Framework**

The Application's Framework is CodeIgniter 4 which is a Model-view-controller web frameworks and provides libraries for URL mapping to assign the URL according to the views (pages), the database migration is another component supported by the framework for allowing the web application to interact with the database for fetching and posting data from the database, templating component is used to provide a consistent header for each view for the whole application.

### **Application Requirements**

The web application will consist of six pages, the first page is the home page with banner displaying images of deals on offer, the second page will a registration page to enable users to create an account, the third page will be a logging page for customers that are registered on the application, the fourth and fifth page will display men's and women's clothing once the customer clicks on an they are interested then the application will redirect the product page with a description of the name, price and image of the product.

### **Requirements to be met:**

- Logging and Registration function
- Men's and Women's clothing products page
- Encryption technology implemented on the application.
- Application hosted on Amazon Cloud
- Web application firewall installed on Amazon Cloud Server
- Vulnerability scanner to test the application based on security.

### **Usability**

Users will be able to navigate through the online e-commerce store with the nav bar located on the top of the page for easy access to different pages such as the game product pages. The customer will easily find a product they desire via a search bar, once the customer is happy with the item, then they will be able to click on the product and proceed to check out.

### **User-friendliness**

The application is compatible with mobile devices with its effective user interface, Users can register, purchase, navigate and logging from their mobile devices. The layout of the application will adapt to the mobile screen size by adjusting the page elements such as images, text, and design to for flexibility and compatibility.

### **Traceability of requirements**

During the development stage, requirements will be traced with use of test runs and test plan to evaluate the progress of the application. This will help the development of the application for ensuring the application is secure, user friendly, usable, and backwards compatible. With use the of a test plan, this will be used to test application security by using a vulnerability scanner and web

application firewall to run tests on the application. Once these tests run, then the results will be used and stored into the test plan for evaluation.

### Requirements Checklist

A requirement checklist is created to provide a list of requirements for the applications to determine the expected outcomes and actual outcomes to be achieved.

Application Requirements	Expected outcome	Tick or Cross
The web application provides a Registration and logging system.	Users can register and log in on to the application.	✓
Users can view products.	Multiple product page is expected display on the application to allow users to view products and is available for purchase.	✓
Encryption technologies has been installed for ensuring users a secure connection.	Encryption technologies are expected to provide a secure transmission between the server and the client for protecting confidential information.	✓
Web application firewall is installed for filtering and monitoring traffic inbounds and outbounds connection	Web application has installed on the server and provides monitoring of inbounds and outbounds connections.	✓
Testing the application to discover any security vulnerabilities.	Vulnerability scanner has been for finding any gaps on the systems for possible vulnerabilities on the application.	✓



## Project Design

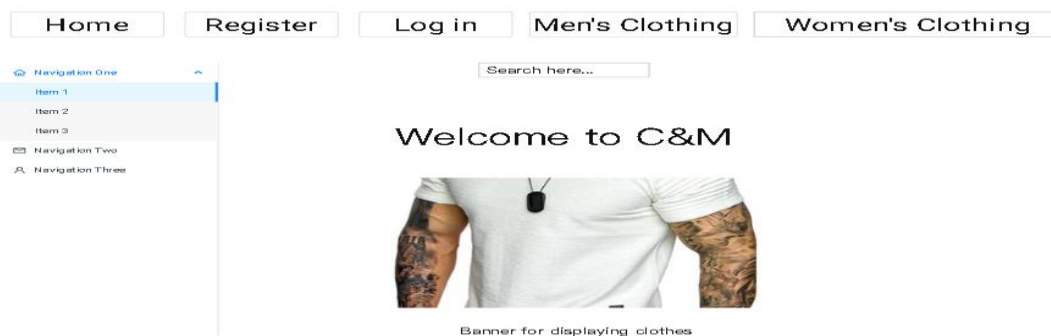
### Wireframe Models

The wireframe model provides a mock-up of the applications by providing illustrations based on content and pages. It will point out the layout, user's interface and navigation on computers and mobile devices before the initial development begins. With the use of Wireframe models, this will help to create a great user interface to provide usability, and functionalities from the requirements for the database driven web application. There will be models of the applications for each page:

- Homepage
- Register
- Logging In
- Men's and Women's Product page
- Transaction page

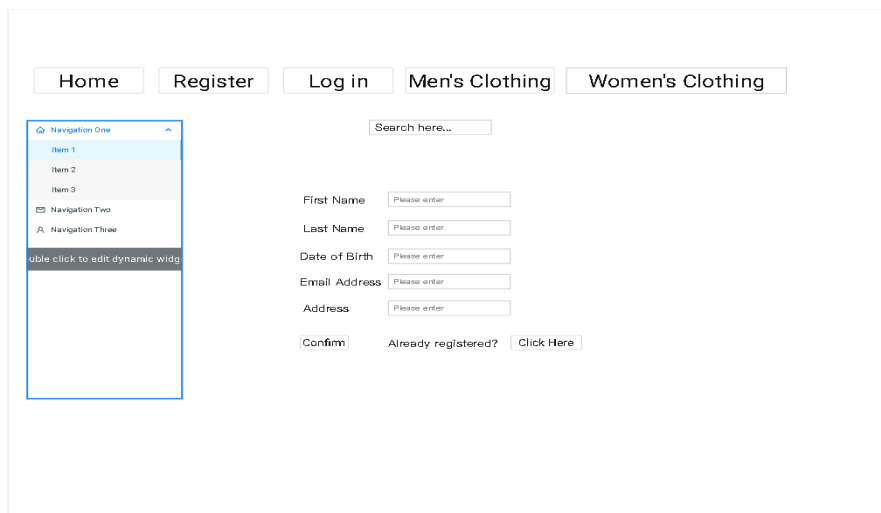
### Homepage Model

The Homepage displays the navigation bar on the left-hand side of the page to allow users to access the registration page and the logging page. This page has a banner to display a product to users with a link to the specific product. The application provides a search bar to enable users to search for products. The Homepage is developed on HTML, CSS, JavaScript, and PHP. The HTML programming language is providing a navigation bar to the registration, Login, Men's clothing, and Women's pages, according to the suitable URL. The JavaScript part of the page is responsible for creating an image slider to display deals to customers with three images and provides a button for users to change the images on the slider. CSS is used for this page for to set the font size, the size of the images for image slider, and the font type of the whole page.



## Register Page Model

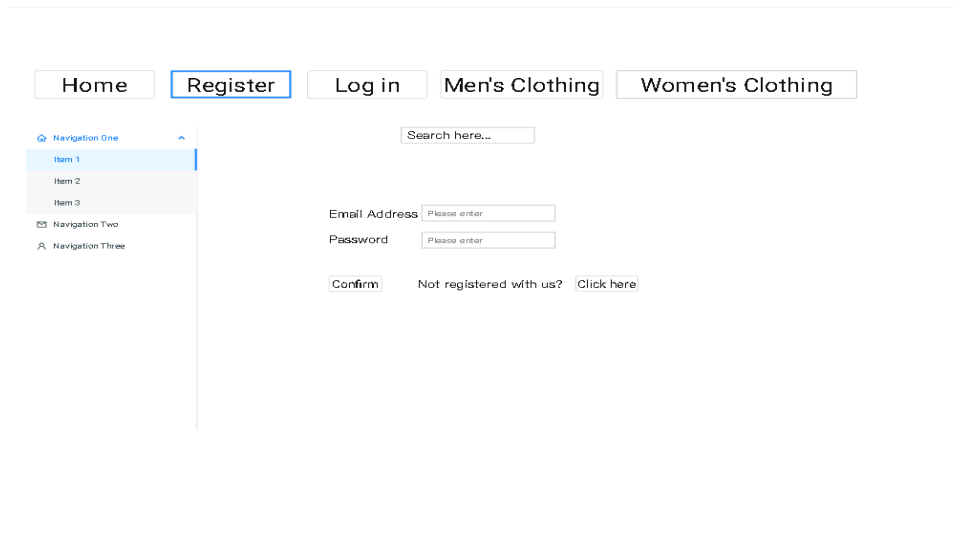
This Model provides an illustration of the registration form design which can look like on the actual artefact. Users will be able to be filling their information such as First Name, Last Name, Date of Birth, Email address and their home address. The information then will be saved to a database. The page is created by using HTML and CSS for building a registration form with input fields, PHP scripts is responsible for fetching data from the input fields to post the data to the database.



The image shows a web page layout for a registration form. At the top, there is a navigation bar with five buttons: "Home", "Register", "Log in", "Men's Clothing", and "Women's Clothing". Below the navigation bar, on the left, is a sidebar titled "Navigation One" with a list of items: "Item 1", "Item 2", "Item 3", "Navigation Two", and "Navigation Three". A search bar with the placeholder text "Search here..." is located to the right of the sidebar. The main content area contains a registration form with the following fields: "First Name", "Last Name", "Date of Birth", "Email Address", and "Address". Each field has a "Please enter" placeholder. Below the "Address" field, there is a "Confirm" button and a link "Already registered? Click Here".

## Logging Page Model

The logging page enables users to log in to the application with their credentials after they have registered. If the user does not have an account, then they will be able to by click on a link to register. The page is developed using three programming languages, which are HTML, CSS, and PHP. The HTML language is responsible for presenting the login form with input fields, CSS is used to set the font sizes and layout. The PHP language fetches data from the Users tables on the database to

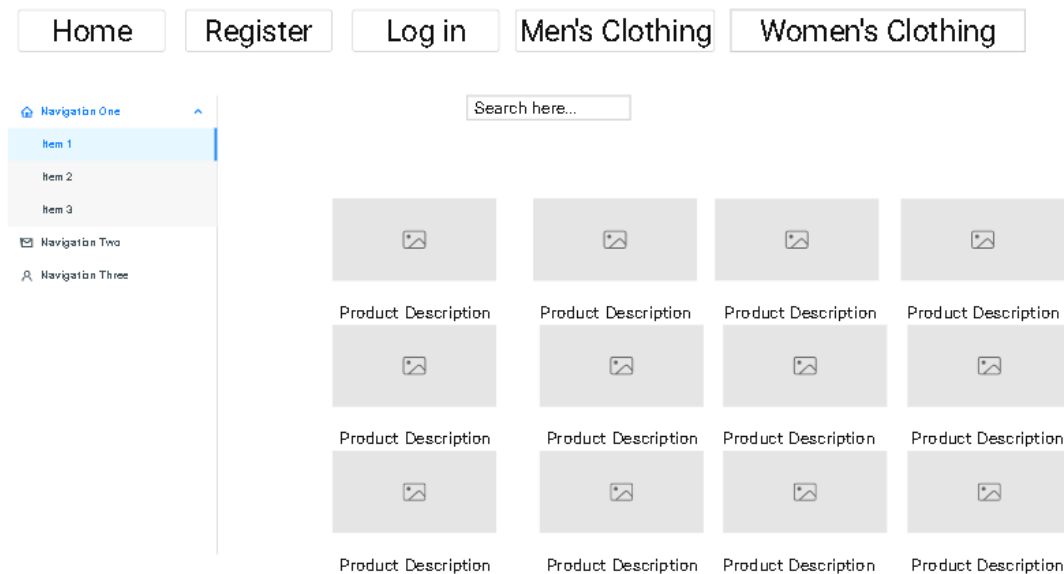


The image shows a web page layout for a login form. At the top, there is a navigation bar with five buttons: "Home", "Register", "Log in", "Men's Clothing", and "Women's Clothing". Below the navigation bar, on the left, is a sidebar titled "Navigation One" with a list of items: "Item 1", "Item 2", "Item 3", "Navigation Two", and "Navigation Three". A search bar with the placeholder text "Search here..." is located to the right of the sidebar. The main content area contains a login form with the following fields: "Email Address" and "Password". Each field has a "Please enter" placeholder. Below the "Password" field, there is a "Confirm" button and a link "Not registered with us? Click here".

validate a user session if their email address and password is valid.

### Men's and Clothing Page Model

Users can view clothes on this page and proceed to check out once they click on a product. The Men's and Women's Clothing page is created using HTML and CSS. The Product listing is created using HTML programming language to create the product cards on the page with a title and price of the product. CSS is used to add images to the product cards and set the sizes for each section on the page.



### Products listing Page Model

Once the user is happy with the product, the user will be able to click on the product which will take to client to the desire product page. This page displays the description of the product including price and size of the clothes. If the user happy with the item, then they can proceed to the payment page to purchase the item. The pages consist of four programming languages, HTML, CSS, and JavaScript. HTML is used to structure of the product listing element. JavaScript is used to allow users to change colour of the products by clicking on the buttons to confirm their choice. CSS is used to set the font size of the label for the products and determine the font size of the text on the page.

## UML Diagrams

The UML Diagram will be used to set the modelling for all stages of development for the database driven web application. An Entity-relationship model of the application's database will be designed to illustrate the database relationships between different entities. The diagram will also describe the database entities on each table that will be used on the application. The UML diagram is used to create a visualization of the software application by defining the database tables with required entities that are responsible for storing user's data.

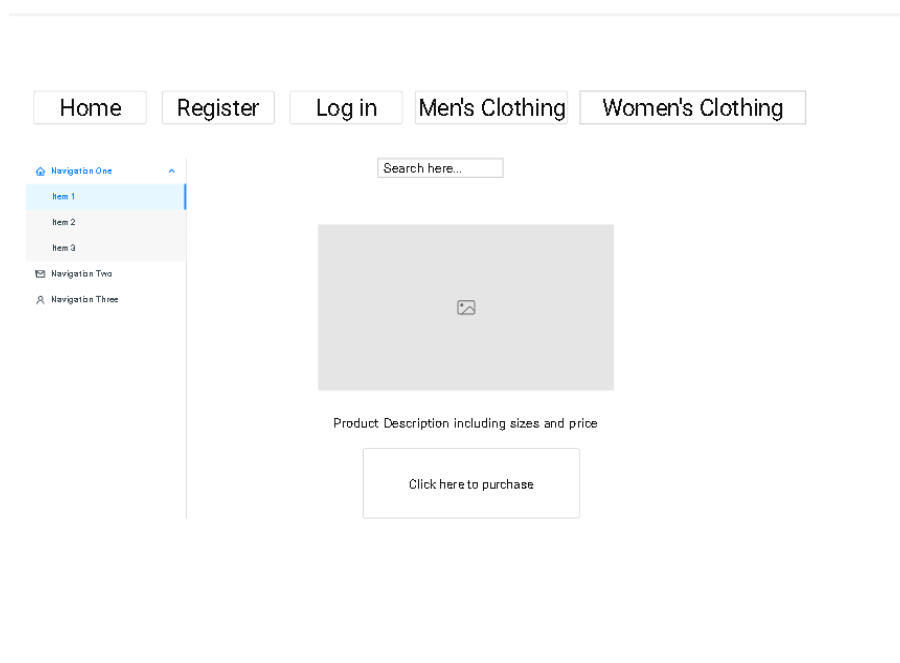
## Entity Database Diagram

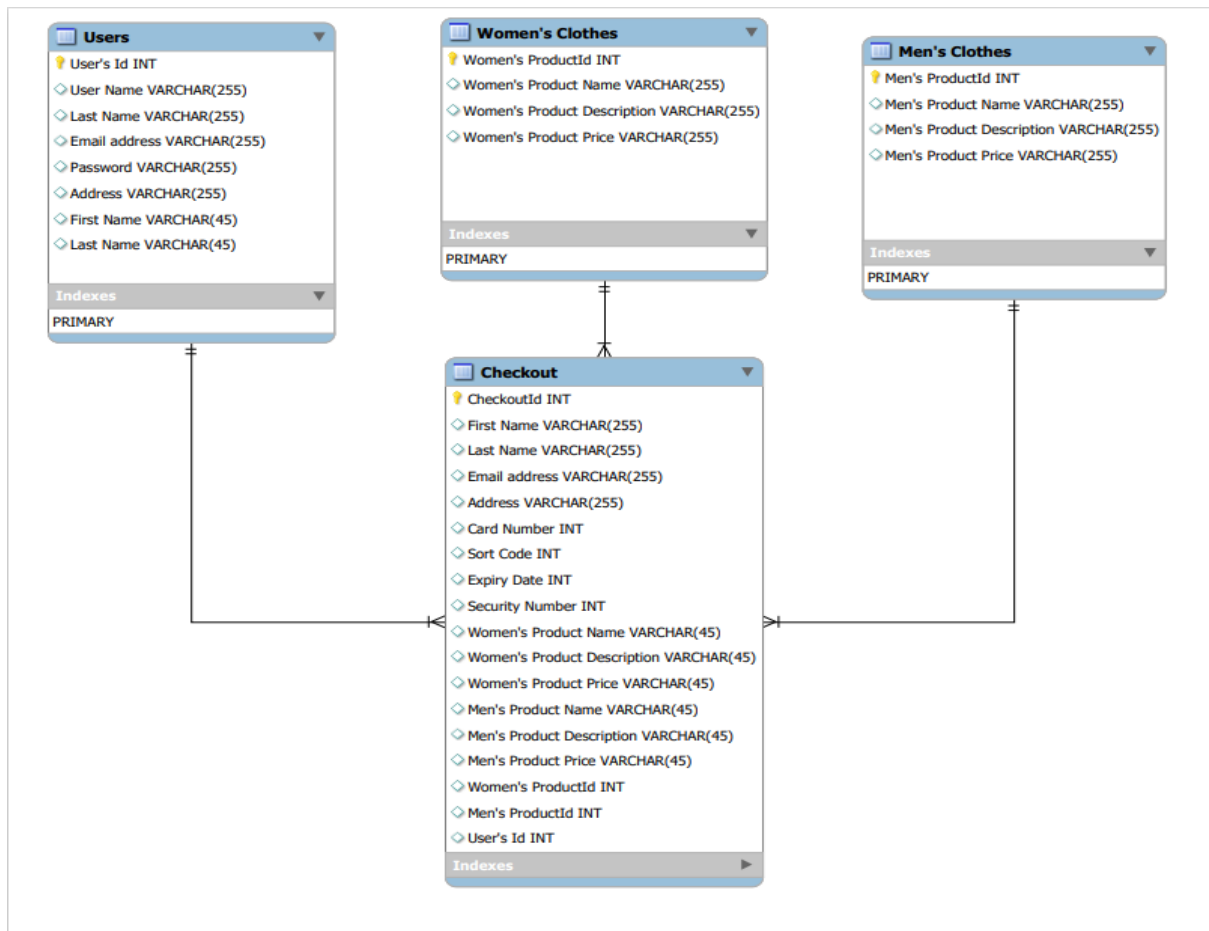
There are four Database tables on the application for storing users' information, clothes descriptions and price and establishing checkouts between the client and the application.

**Users Table:** Consists of Eight attributes for storing users' information once they have registered via the registration form on the application. The information will be retrieved from the users table to provide a logging confirmation through the information saved on the application to the table. This is responsible for a creating a login and registration system for the application. The Users table will hold user's information such as, ID, first name, last name, email address and password. The login system will validate the data from the User's database to create a user session for the user. The ID on the user's table is the primary key on the table and provides unique id number to each user's once they are registered on the system.

**Men's and Women's Clothes Table:** This table provides information about the clothes such as name, description and price and size.

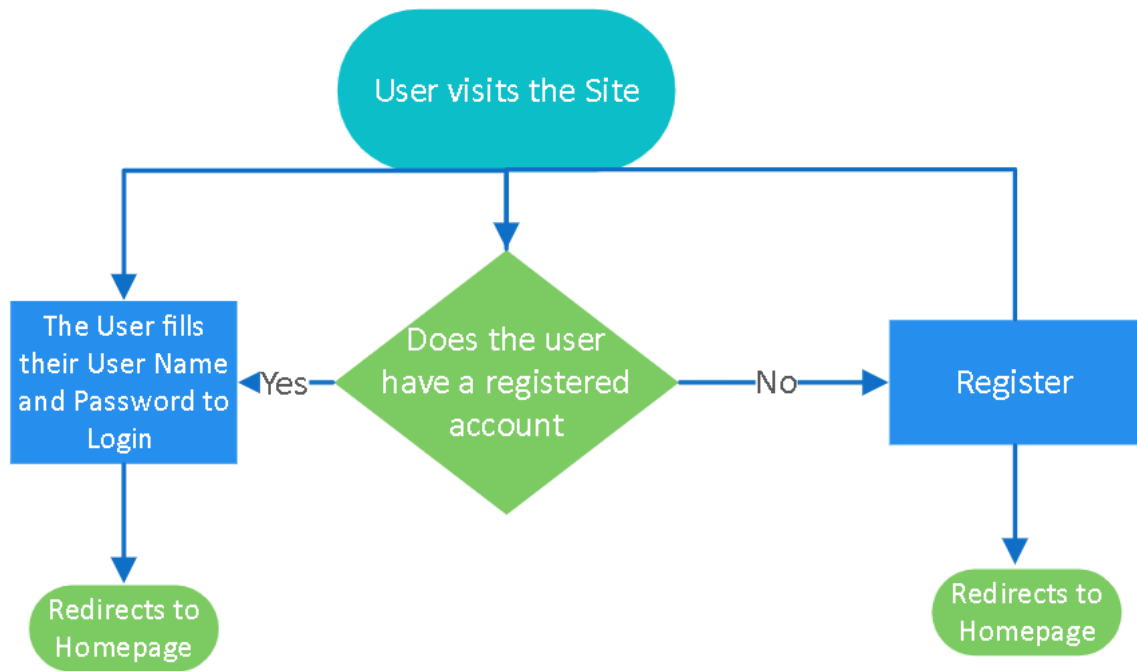
**Checkout Table:** This table is responsible for retrieving information from the users table, Women's, and Men's clothes table by the recognition of foreign keys to establish the users, the item and payment details from users for a successful checkout when purchasing a product.



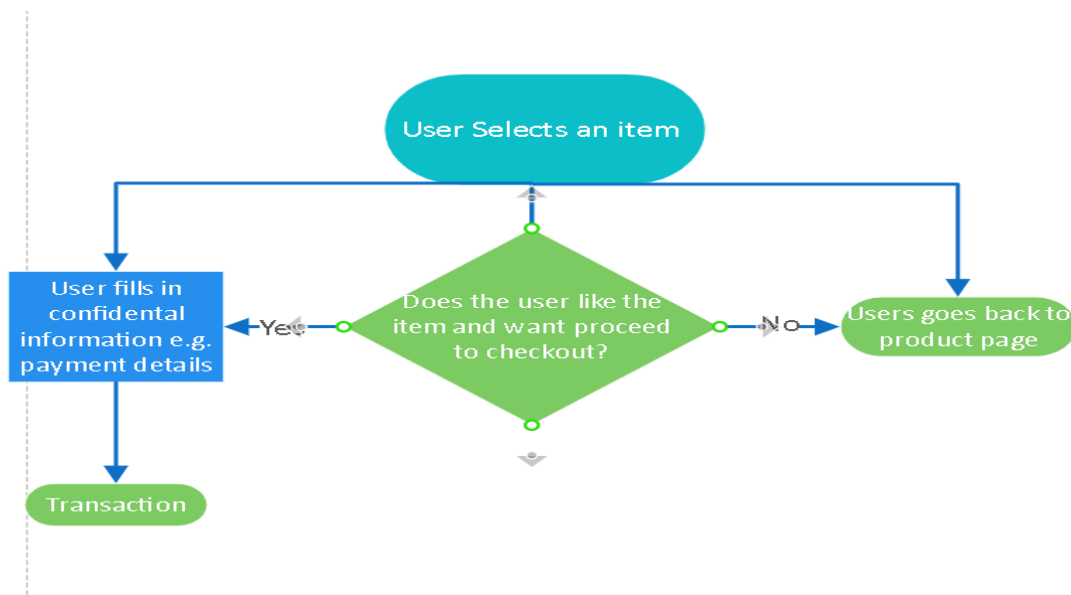


### User Interface Diagram

User registration and login interface: The user will be able to register on the application if they like to or they can browse on the application as a guest. The user will need to register on the application if they do not have an account to log in. If the user does not have an account, they can register on the application.



User checkout interface: The user browses the Men's and Women's product page to look for clothes, once they are happy to purchase the item, then they can proceed to the checkout for a successful transaction. If the user changes their mind and is not happy then they can go back to the products page to browse for more clothes.



## **Artefact Testing methodology the security of the application.**

### **Testing Methodology**

There will be tests on the application on basis of security to provide a secure application for protecting confidential data and web application from online threats. A variety of tools will be used to enforce security and test the application with a vulnerability scanner, web application firewall and uses of encryption technologies for a secure transmission of data. The results based on security will be documented on a test plan for an evaluation if the application is secure from any risks. Regarding this these tools are essential for making the application secure for users to protect against threats such as SQL injections, Cross site scripting, data theft of confidential information and Denial of Service on the application.

### **Vulnerability Scanner Methodology**

The vulnerability scanner will used to find any cyber risks which can negatively impact the application's security which help to develop a secure application by fixing the cyber risks. The penetration testing will reveal vulnerabilities that are hard to find and can help aid in the finding a solution to a security vulnerability.

### **Web Application Firewall Methodology**

The Web Application Firewall is responsible for blocking any possible threatening outbound and inbound connections to the application. This will be used to protect the application from unwanted connections through the policies set up on the firewall. AWS provides an WAF which will keep are record of all previous history of the connections to the applications including SQL injection scripts, malicious PHP scripts. Denial of Service attacks and admin protection attacks.

### **Encryption Technologies**

Encryption Technologies are used on the application for protecting confidential data of users such as payment details, Addresses, Names and Data of Births. The Encryption will hide the information on the internet to protect it from possible hackers who may want to steal the information for their own benefit.

## Implementation and Development

### Introduction


The artefact prototype is an ecommerce web application hosted on the Amazon Web Services cloud to utilize the secure protocols of HTTPS, as well as the AWS web application firewall to manage inbound and outbound connections to the web application. The web application will be developed using CodeIgniter to create a dynamic web application for register and logging systems for users, their credentials will be stored on a PHPMYADMIN database. The XAMPP web server will host the application and will be installed on the windows server 2019 cloud machine with the firewalls and security technologies implemented.


### Windows Server 2019 Base EC2 instance

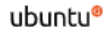
**▼ Application and OS Images (Amazon Machine Image)** [Info](#)


An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or [Browse](#) for AMIs if you don't see what you are looking for below


**Quick Start**


  
aws


  
Mac

  
ubuntu®

  
Microsoft

  
Red Hat

  
SUS

  
[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Microsoft Windows Server 2019 Base  
ami-035d8ae8de3734e5a (64-bit (x86))  
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Free tier eligible ▼

### Amazon Web Server EC2

Amazon Web Services EC2 is service provided by Amazon to allow businesses to run applications on the public cloud and enable access to their virtual computers. The EC2 instance enables access to virtual machines to deploy and develop applications without investing into hardware such as computer servers to host the web applications on the internet. EC2 instances provides configuration for security features, networking and manage cookies from the EC2 dashboard.



## Configuring EC2 instance

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
rdp ▼	TCP	3389
Source type <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security group"/>	<input type="text" value="e.g. SSH for admin desktop"/>
<input type="text" value="0.0.0.0/0"/> X		
▼ Security group rule 2 (TCP, 443, 0.0.0.0/0)		<button>Remove</button>
Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
HTTPS ▼	TCP	443
Source type <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security group"/>	<input type="text" value="e.g. SSH for admin desktop"/>
<input type="text" value="0.0.0.0/0"/> X		
▼ Security group rule 3 (TCP, 80, 0.0.0.0/0)		<button>Remove</button>
Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
HTTP ▼	TCP	80
Source type <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security group"/>	<input type="text" value="e.g. SSH for admin desktop"/>

These are the security groups for enabling RDP, HTTPS, HTTPS, DNS and SMB. RDP stands for remote desktop protocol which is a secure network communication protocol created by Microsoft for administrators to fix any issues relating server where possible a user may face problems and enable the administrator to have remote access to the user's desktop computer Enterprise Desktop. (n.d.)

### RDP protocol

Remote Desktop Protocol uses Hypertext transfer secure protocol to establish an encrypted connection to the user and Amazon Elastic Compute Cloud to set up an Amazon Elastic Computer Cloud instance to run Windows Server 2019 on the cloud servers. RDP protocols enables direct access to the Windows Server 2019 EC2 instance by downloading RDP instance directly to my computer. The file requires a PEM key generated by Amazon Web Services which then is converted to a password for access to the Windows Server 2019 EC2 instance Machine.



### EC2 Key Pairs

 securewebserver.pem	19/01/2024 21:18	PEM File	2 KB
---	------------------	----------	------

Amazon EC2 uses public key cryptography for encrypting and decrypting login information while the public key cryptography encrypts the data through the public key, and the user uses the private key to decrypt the data. docs.aws.amazon.com

HTTPS and SSL Protocol

The Hypertext transfer protocol secure is used to send encrypted data between the web browser and a website. HTTPS increases the security of data transfer for the web application. XAMPP uses Secure Socket Layer to encrypt the data transmission which enables the HTTPS to use Secure Socket Layer to provide asymmetric public key infrastructure. XAMPP provides a windows batch file to create a certificate to indicate users the identity of the remote computer which is Localhost. The Localhost is a server create from XAMPP and will host the web application.

> This PC > Local Disk (C:) > xampp > apache > crt

	Name	Date modified	Type	Size
s	makecert	1/28/2024 6:08 PM	Windows Batch File	1 KB
ts	cert	1/28/2024 6:12 PM	CONF File	2 KB
ls	localhost	1/28/2024 11:52 PM	File folder	

Certificate

GeneralDetailsCertification Path

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer

Issued to: Localhost

Issued by: Localhost

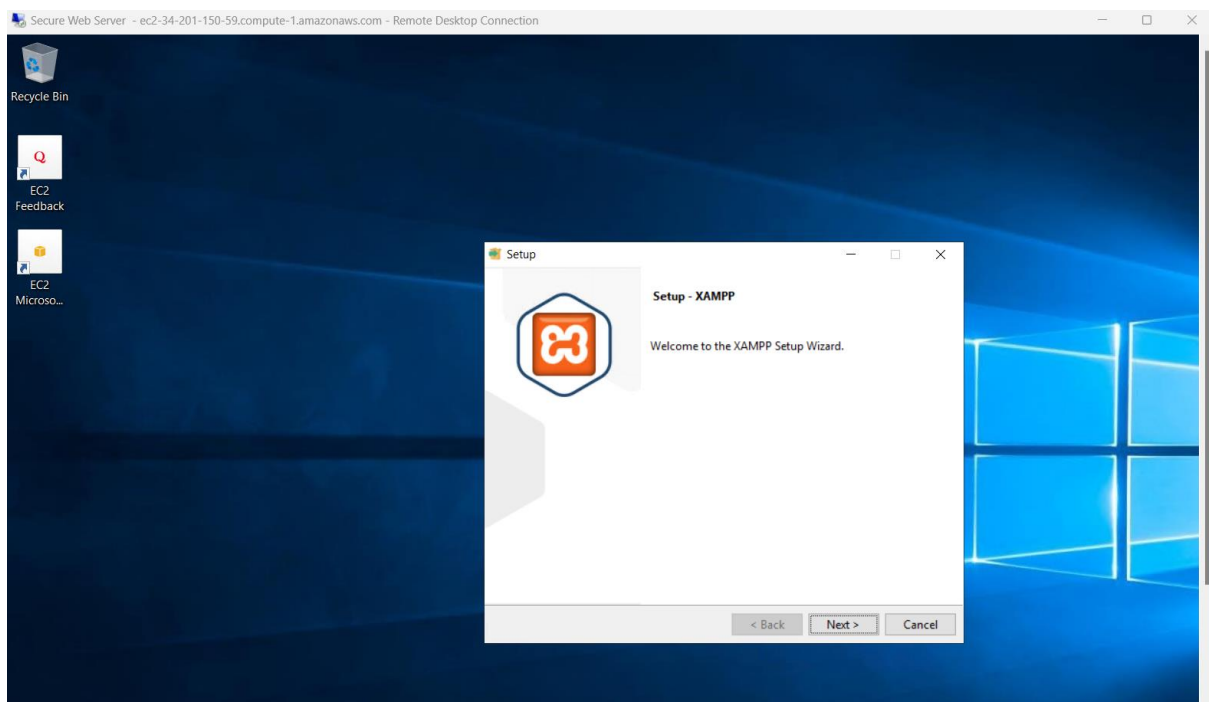
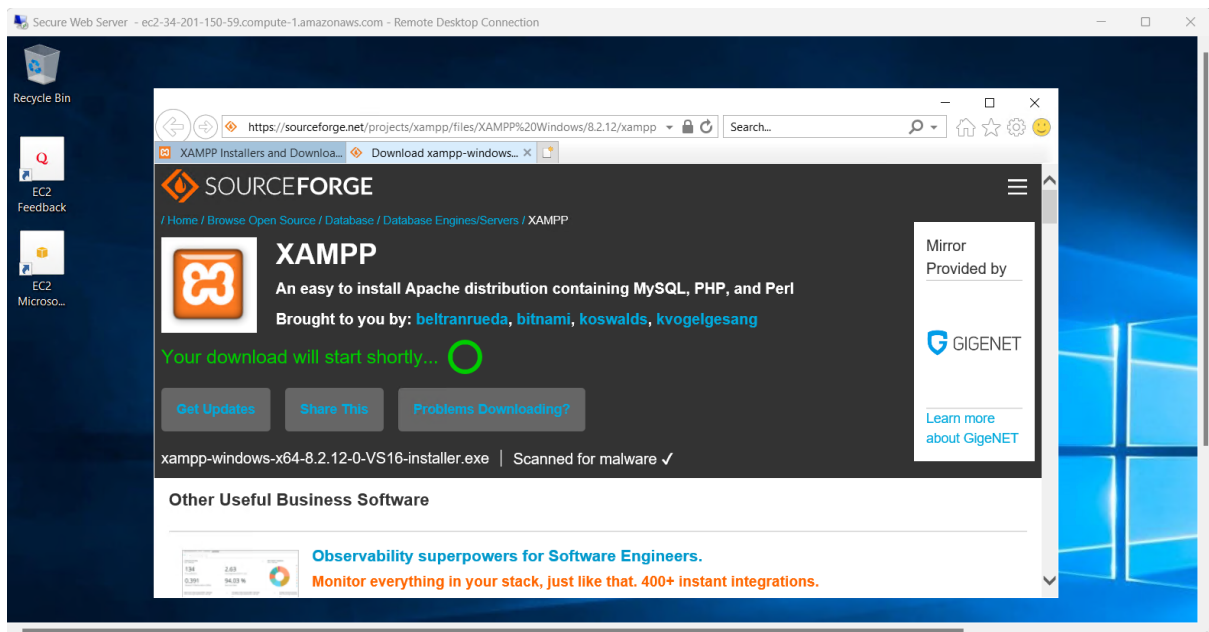
Valid from 1/28/2024 to 1/27/2025

Install Certificate...

Issuer Statement

OK

## Installing XAMPP to the Windows Server 2019 EC2 Instance

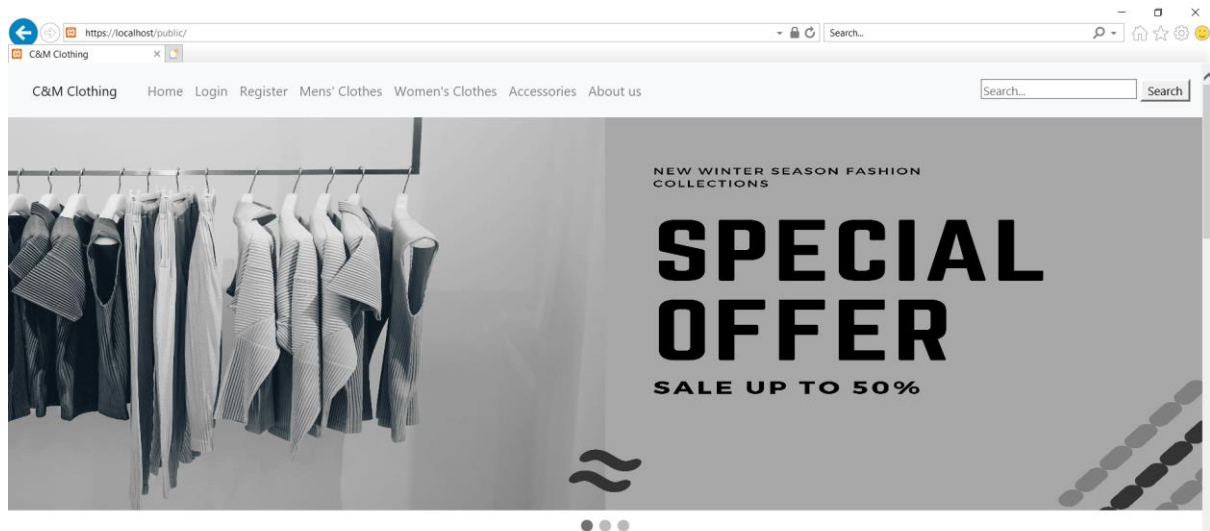


XAMPP is a web server which will be used to host the web application on the Windows Server 2019 works very well as it does not require to set up Apache, MariaDB, PHP, Perl, PhpMyAdmin, FTP and Tomcat as the Server provides the configuration already. Windows Server 2019 provides all the features and is optimized for speed and performance (console.cloud.google.com, 2024).

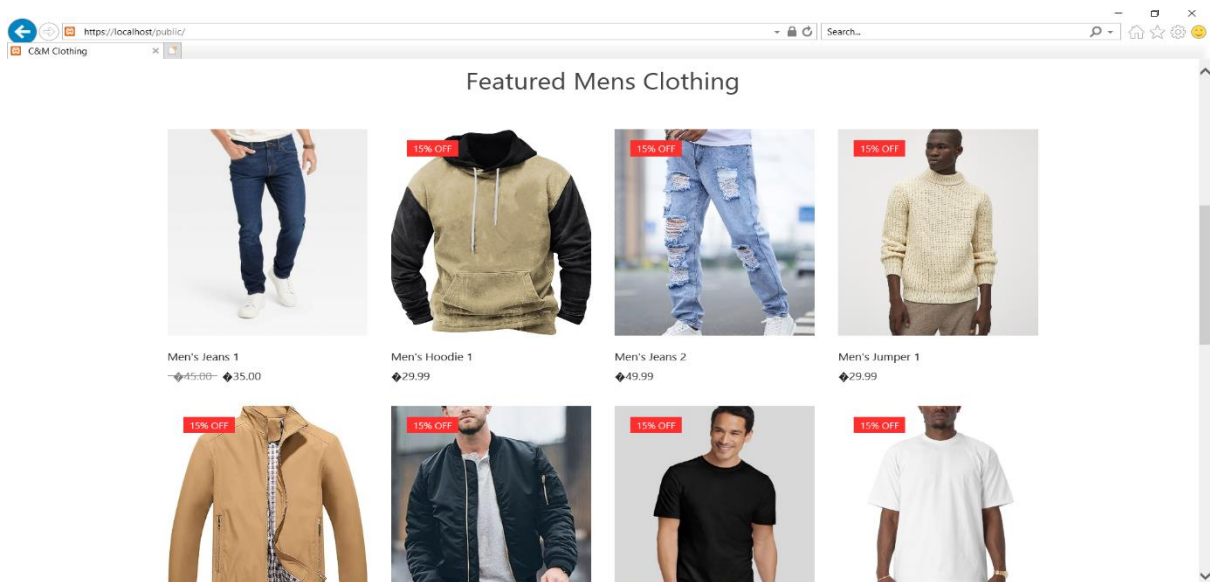
## Project Artefact: Secure Web Application

The web application is hosted on the XAMPP local server on the Hypertext protocol and provides a certificate to users to provide that the identity of this computer is correct. The homepage provides a navigation bar with links to the homepage, login form, registration form, Men's clothing page, Men's clothing page, Accessories, about us page and a search bar. The Homepage was developed using CodeIgniter 4 framework and by HTML, CSS, and JavaScript.

### Homepage



### Featured Mens Clothing



## Code Appendix for the Homepage

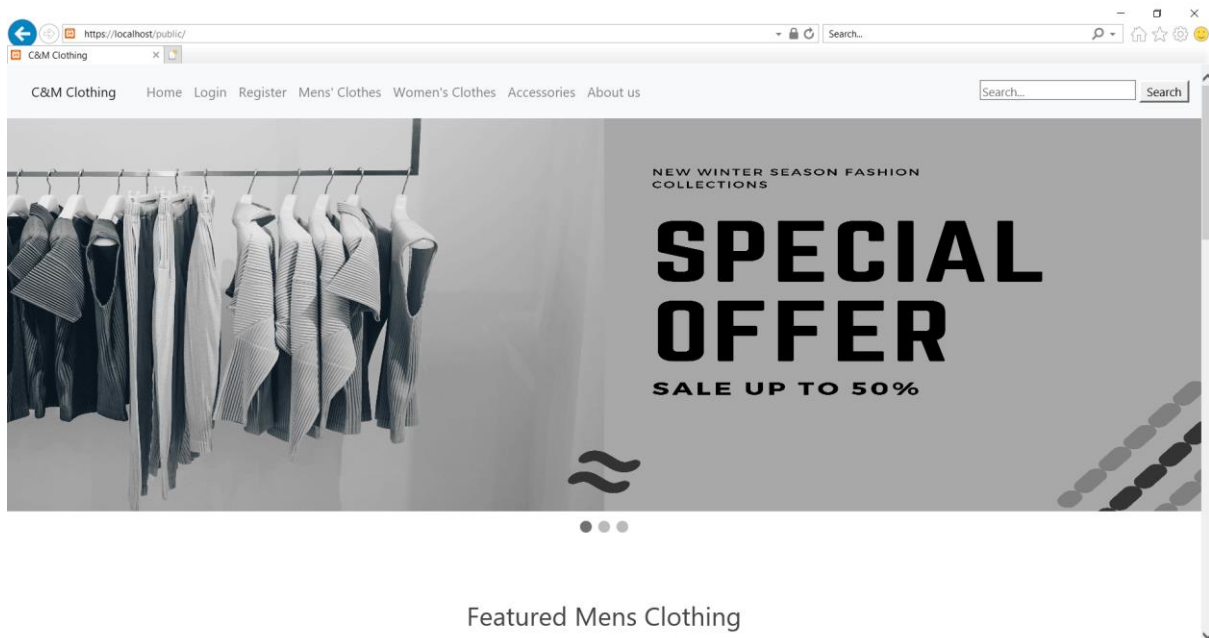
```
<div class="slider">
  

</div>
```

The Div Class slider is an html code for displaying three images on the banner for the top part of the page below the navigation bar.



```

</div>
<div class="navigation-button">
  <span class="dot active" onclick="changeSlide(0)"></span>
  <span class="dot" onclick="changeSlide(1)"></span>
  <span class="dot" onclick="changeSlide(2)"></span>
</div>
<script>
var currentImg = 0;
var imgs = document.querySelectorAll('.slider img');
let dots = document.querySelectorAll('.dot');
var interval = 3000;

// Second banner

document.getElementById('img-2').src = secondImageUrl;
document.getElementById('img-3').src = thirdImageUrl;

var timer = setInterval(changeSlide, interval);

function changeSlide(n) {
  for (var i = 0; i < imgs.length; i++) {
    imgs[i].style.opacity = 0;
    dots[i].className = dots[i].className.replace(' active', '');
  }
}

```

```

currentImg = (currentImg + 1) % imgs.length;

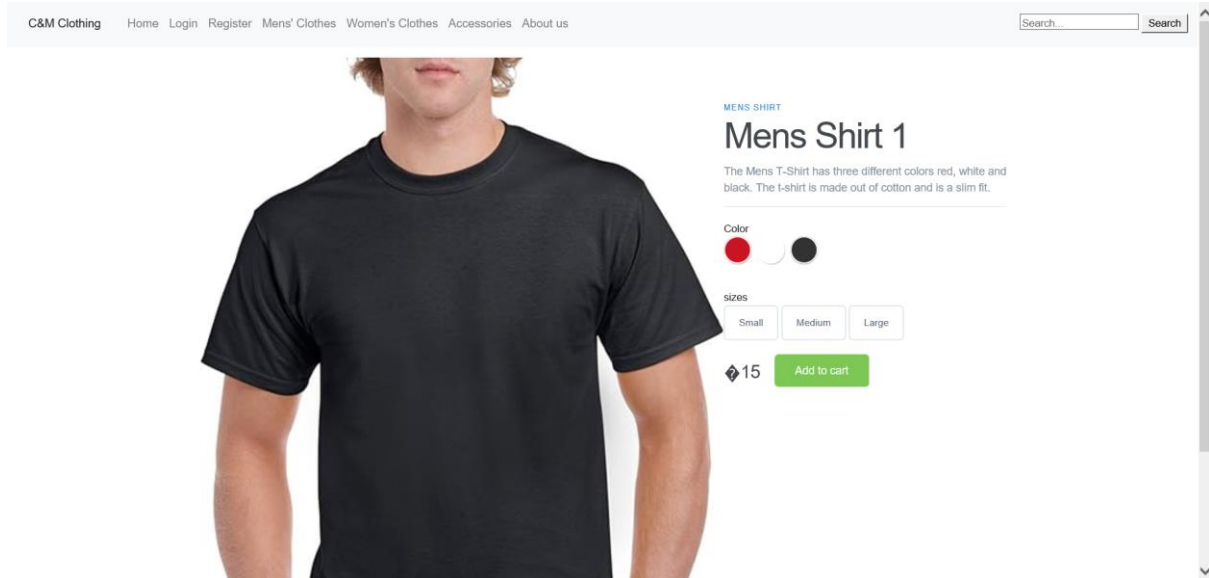
if (n != undefined) {
  clearInterval(timer);
  timer = setInterval(changeSlide, interval);
  currentImg = n;
}

imgs[currentImg].style.opacity = 1;
dots[currentImg].className += ' active';
}

```

The JavaScript enables the slider to change images automatically once the user clicks on the Navigation buttons on the slider by setting a timer on the intervals “changeSlide” to images.

## Code Appendix for the Product listing page.



The product listing page uses HTML, CSS, and JavaScript. The HTML code displays the images while the CSS sets the image size and the placement of the image. The JavaScript allows users to select the colour buttons on the page to change the colour of the product.

```
<div class="product-configuration">

  <!-- Product Color -->
  <div class="product-color">
    <span>Color</span>

    <div class="color-choose">
      <div>
        <input data-image="red" type="radio" id="red" name="color" value="red" checked>
        <label for="red"><span></span></label>
      </div>
      <div>
        <input data-image="white" type="radio" id="white" name="color" value="white">
        <label for="white"><span></span></label>
      </div>
      <div>
        <input data-image="black" type="radio" id="black" name="color" value="black">
        <label for="black"><span></span></label>
      </div>
    </div>
  </div>

  <div class="size">
    <span>sizes</span>

    <div class="cable-choose">
      <button>Small</button>
      <button>Medium</button>
      <button>Large</button>
    </div>
  </div>
</div>
```

The HTML code defines the buttons to allow users to select the colour of the product, and the cable-choose buttons enables users to select their size that they desire.



```

3
9      <!-- Product Pricing -->
0      <div class="product-price">
1          <span>15</span>
2          <a href="/public/basket/index" class="cart-btn">Add to cart</a>
3      </div>
4  </div>
5  </main>
6  <script>
7  $(document).ready(function() {
8
9      $('.color-choose input').on('click', function() {
10          var mensshirtColor = $(this).attr('data-image');
11
12          $('.active').removeClass('active');
13          $('.left-column img[data-image = ' + mensshirtColor + ']').addClass('active');
14          $(this).addClass('active');
15      });
16  });
17  </script>

```

The JavaScript enables users to change the colour of the product and display the correct image based on the button clicked.

### Registration form code appendix

The registration form is created by bootstrap and PHP to get the data and post it to the Users database table on PHPMYADMIN.

The screenshot shows a web browser window with the URL `https://localhost/public/register`. The page title is "C&M Clothing". The navigation bar includes links for Home, Login, Register, Mens' Clothes, Women's Clothes, Accessories, and About us. A search bar is located on the right. The main content area is titled "Register" and displays a "Successful Registration" message in a green box. Below the message, there are input fields for First Name (filled with "crolos"), Last Name (filled with "massaud"), Email Address (filled with "test1@gmail.com"), Password, and Confirm Password. A blue "Register" button is at the bottom left, and a link "Already have an account?" is at the bottom right.

## UsersModel.PHP

```
protected $table = 'users';
protected $primaryKey = 'id';
protected $useAutoIncrement = true;
protected $returnType = 'object';
protected $validationRules= [];
protected $validationMessages= [];
protected $skipValidation = false;

protected $allowedFields=['email', 'firstname', 'lastname','password','created_at','updated_at'];
protected $beforeInsert =['beforeInsert'];
protected $beforeUpdate =['beforeUpdate'];

protected function beforeInsert(array $data) {
    if(isset($data['data']['password'])) {
        $data['data']['password'] = password_hash($data['data']['password'], PASSWORD_DEFAULT);
    }

    return $data;
}

protected function beforeUpdate(array $data){
    $data = $this->passwordHash($data);

    return $data;
}

protected function passwordHash(array $data){
    if (!isset($data['data']['password']))
        $data['data']['password'] = password_hash($data['data']['password'], PASSWORD_DEFAULT);

    return $data;
}
```

The UserModel PHP file interacts with the users table on PhpMyAdmin by selecting all the columns such as email, first name, last name, password, created\_at and updated at. The code states the primary key to display the data. The password hash function is implemented in the file to protect the user's password by hashing the password with random symbols, letters, and numbers to hide the actual password.

## UsersController.PHP

```
<?php

namespace App\Controllers;

use App\Models\UserModel;

class Users extends BaseController
{
    private function setUserSession($user)
```

```

public function register()
{
    $data = [];
    helper(['form']);

    if ($this->request->getMethod() == 'post') {
        $rules = [
            'firstname' => 'required|min_length[3]|max_length[20]',
            'lastname' => 'required|min_length[3]|max_length[20]',
            'email' => 'required|min_length[6]|max_length[50]|valid_email|is_unique[users.email]',
            'password' => 'required|min_length[8]|max_length[255]',
            'password_confirm' => 'matches[password]',
        ];

        if (!$this->validate($rules)) {
            $data['validation'] = $this->validator;
        } else {
            $model = new UserModel();

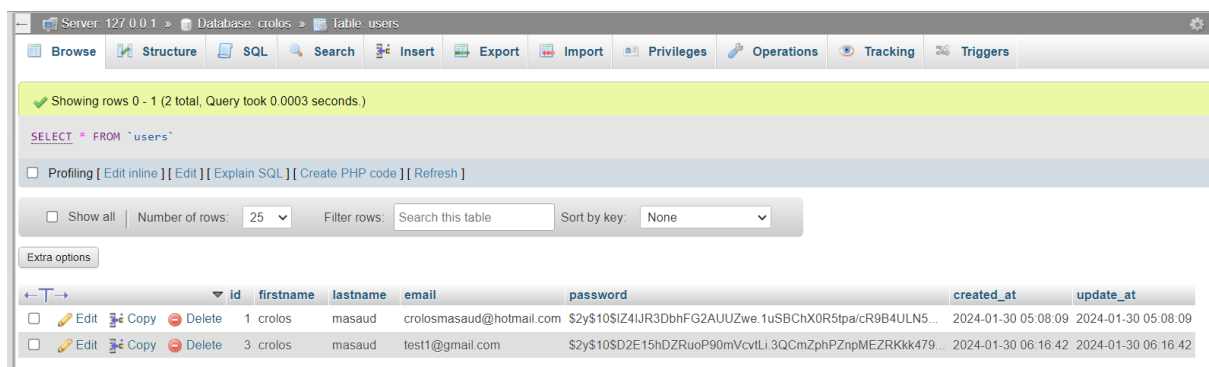
            $newData = [
                'firstname' => $this->request->getVar('firstname'),
                'lastname' => $this->request->getVar('lastname'),
                'email' => $this->request->getVar('email'),
                'password' => $this->request->getVar('password'),
            ];
            $model->save($newData);
            $session = session();
            $session->setFlashdata('success', 'Successful Registration');
        }
    }

    return view('templates/header', $data) . view('register');
}

```

The register function is responsible for getting the data listed which are firstname, lastname, email, password, and password confirm from the registration form to the database and notifies the user once the registration is successful. There are validation rules set to state the min length and max length of each field. The email validation makes sure that the user enters a unique email address meaning that the user cannot use the same email address that they already have used to register.

## PHPMYADMIN database



Server: 127.0.0.1 » Database: crolos » Table: users

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT \* FROM `users`

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	firstname	lastname	email	password	created_at	update_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	crolos	masaud	crolosmasaud@hotmail.com	\$2y\$10\$Iz4lUR3DbhFG2AUUZwe.1uSBChX0R5tpa/cR9B4ULN5...	2024-01-30 05:08:09	2024-01-30 05:08:09
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	crolos	masaud	test1@gmail.com	\$2y\$10\$D2E15hDZRuoP90mVcvtLi.3QCmZpPZnpMEZRKkk479...	2024-01-30 06:16:42	2024-01-30 06:16:42

The password is hashed and encrypted to protect and enforce privacy for the user's password and to abide by the data protection law. The database contains the user's Id, first name, last name, email address, password, real time dates of when the applicant has registered on the web application.

## Login form page code appendix

Login

Email

Password

Login

Users are to login using their registered email address and password which are saved on the users table on PhpMyAdmin. Once the function states the credentials is correct, it will redirect the user to the home page.

### UserModel.PHP

```
protected $table = 'users';
protected $primaryKey = 'id';
protected $useAutoIncrement = true;
protected $returnType = 'object';
protected $validationRules= [];
protected $validationMessages= [];
protected $skipVaildation = false;

protected $allowedFields =['email', 'firstname', 'lastname','password','created_at','updated_at'];
protected $beforeInsert =['beforeInsert'];
protected $beforeUpdate =['beforeUpdate'];

protected function beforeInsert(array $data) {
    if(isset($data['data']['password'])) {
        $data['data']['password'] = password_hash($data['data']['password'], PASSWORD_DEFAULT);
    }
    return $data;
}

protected function beforeUpdate(array $data){
    $data = $this->passwordHash($data);
    return $data;
}

protected function passwordHash(array $data){
    if (isset($data['data']['password']))
        $data['data']['password'] = password_hash($data['data']['password'], PASSWORD_DEFAULT);
    return $data;
}
```

The log in function uses the same model to select the appropriate fields on the database to validate the user if the credentials are on the database.

## UsersController.PHP

```
public function login()
{
    // Load helper functions for form validation
    helper(['form']);

    // Initialize an empty array to hold data to be sent to the view
    $data = [];

    // Check if the form has been submitted
    if ($this->request->getMethod() == 'post') {
        // Define validation rules for email and password
        $rules = [
            'email' => 'required|valid_email',
            'password' => 'required|min_length[8]|max_length[255]'
        ];

        // Validate the input data against the defined rules
        if ($this->validate($rules)) {
            // If validation passes, attempt to authenticate the user
            $model = new UserModel();
            $user = $model->where('email', $this->request->getVar('email'))->first();

            // Check if user exists and if password matches
            if ($user && password_verify($this->request->getVar('password'), $user->password)) {
                // Set user session
                $this->setUserSession($user);
                // Redirect to dashboard or any desired page after successful login
                return redirect()->to('/');
            } else {
                // Invalid email or password
                $data['error'] = 'Invalid email or password';
            }
        } else {
    }
```

The login function posts email and password, with validation rules to validate the data against the defined rules. The rules part of the code makes that the user only enters a valid email address and password. The login function also validates the user by checking if the password matches the password filed on the Users to authenticate the session.

# Steps to setting up Amazon Web Application Firewall

## Configuring VPC

vpc-071df85b38806bb2c / secure web serve vpc

Details

Resource map




CIDRs

Flow logs

Tags

Integrations

Details

<div>VPC ID</div> <div> vpc-071df85b38806bb2c</div>	<div>State</div> <div> Available</div>	<div>DNS hostnames</div> <div>Disabled</div>	<div>DNS resolution</div> <div>Enabled</div>
<div>Tenancy</div> <div>Default</div>	<div>DHCP option set</div> <div><a href="#">dopt-0dccc59ef5604a3e7</a></div>	<div>Main route table</div> <div><a href="#">rtb-0d7b0ac8fe89615c3</a></div>	<div>Main network ACL</div> <div><a href="#">acl-0337ae30994e45154</a></div>
<div>Default VPC</div> <div>No</div>	<div>IPv4 CIDR</div> <div>12.0.0.0/16</div>	<div>IPv6 pool</div> <div>-</div>	<div>IPv6 CIDR (Network border group)</div> <div>-</div>
<div>Network Address Usage metrics</div> <div>Disabled</div>	<div>Route 53 Resolver DNS Firewall rule groups</div> <div>-</div>	<div>Owner ID</div> <div> 484442361528</div>	

Amazon Virtual Private Cloud provides the user full control over network environment for resource placement, connectivity, and security. The Virtual Private Cloud enables clients to specify an IP address range for the VPC for allowing the Web Application Firewall to communicate with Elastic Container Instance including adding the subnets, and gateways with the associated security groups.

## VPC Internet Gateway

igw-0a9c298648dc81fcb / internetgateway

Details	Tags
Details	
Internet gateway ID igw-0a9c298648dc81fcb	State Attached
VPC ID vpc-071df85b38806bb2c   secure web serve vpc	Owner 484442361528

The internet gateway is a VPC component for enabling communication between the VPC and the internet. The internet gateway uses IPv4 and IPv6 traffic to connect the VPC and the internet together and does not cause bandwidth constraints on the network traffic.

## VPC Subnet

Subnets (2) info

Find resources by attribute or tag

Subnet ID : subnet-0ae8979a96f03a200

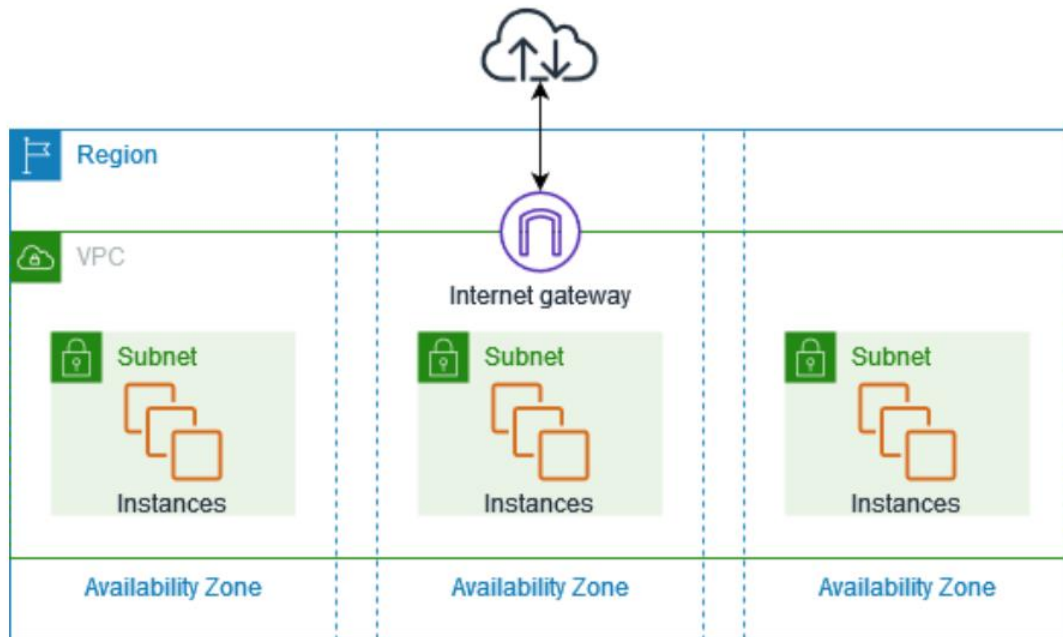
Subnet ID : subnet-0c43a1a98174fc9fa

Clear filters




1

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
<input type="checkbox"/>	securewebserversubnet-1a	subnet-0ae8979a96f03a200	Available	vpc-071df85b38806bb2c   secur...	12.0.1.0/24	-	251
<input type="checkbox"/>	securewebserversubnet-1a	subnet-0c43a1a98174fc9fa	Available	vpc-071df85b38806bb2c   secur...	12.0.2.0/24	-	251

A subnet consists of two IP addresses for the two zones and is used to connect the VPC to the internet to route the subnets.



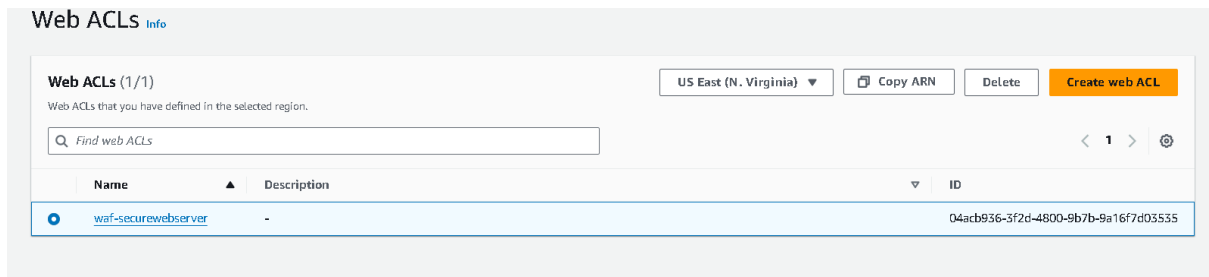
## Route Table

Details <a href="#">Info</a>	
Route table ID  rtb-022b7557ec85aac77	Main  No
VPC <a href="#">vpc-071df85b38806bb2c</a>   secure web serve vpc	Owner ID  484442361528
Explicit subnet associations —	Edge associations —

The route table is a set of rules to for stating where the network traffic of the subnets.

## Application Load Balancer

The application Load balancer is not a security solution, but it is used to preserve the services from mitigation an impact of a DoS attack.

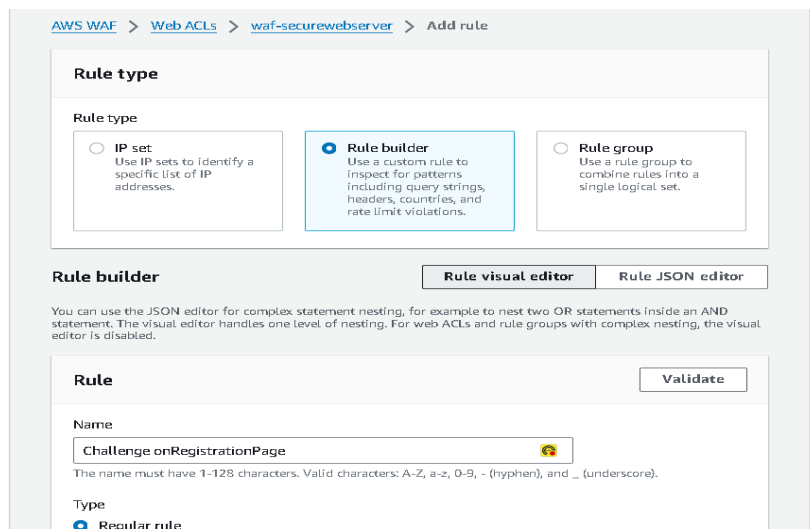


The screenshot shows the 'Web ACLs' page in the AWS IAM console. The page title is 'Web ACLs' with an 'Info' link. Below the title, it says 'Web ACLs (1/1)' and 'Web ACLs that you have defined in the selected region.' There are buttons for 'US East (N. Virginia)', 'Copy ARN', 'Delete', and 'Create web ACL'. A search bar contains 'Find web ACLs'. A table lists the web ACLs with columns 'Name', 'Description', and 'ID'. One web ACL is listed: 'waf-securewebserver' with a description '-' and ID '04acb936-3f2d-4800-9b7b-9a16f7d03535'.

Name	Description	ID
waf-securewebserver	-	04acb936-3f2d-4800-9b7b-9a16f7d03535

## Adding a Captcha rule on AWS WAF

The rule builder was used to create a custom rule to stop bots or any suspicious activity on the registration page to register on the web application. The rule builder inspects patterns relating to query strings, header, countries, and rate limit violations.



The screenshot shows the 'Add rule' page in the AWS WAF console. The breadcrumb navigation is 'AWS WAF > Web ACLs > waf-securewebserver > Add rule'. The 'Rule type' section has three options: 'IP set' (radio button), 'Rule builder' (radio button, selected), and 'Rule group' (radio button). The 'Rule builder' option is highlighted. Below this, there are tabs for 'Rule visual editor' and 'Rule JSON editor'. A note states: 'You can use the JSON editor for complex statement nesting, for example to nest two OR statements inside an AND statement. The visual editor handles one level of nesting. For web ACLs and rule groups with complex nesting, the visual editor is disabled.' The 'Rule' section has a 'Validate' button. The 'Name' field contains 'Challenge onRegistrationPage' and a warning icon. A note below the name field states: 'The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and \_ (underscore).' The 'Type' section has a radio button for 'Regular rule' which is selected.

**Rule type**

☐ IP set  
Use IP sets to identify a specific list of IP addresses.

☒ Rule builder  
Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.

☐ Rule group  
Use a rule group to combine rules into a single logical set.

**Rule builder**

**Rule visual editor** **Rule JSON editor**

You can use the JSON editor for complex statement nesting, for example to nest two OR statements inside an AND statement. The visual editor handles one level of nesting. For web ACLs and rule groups with complex nesting, the visual editor is disabled.

**Rule** **Validate**

**Name**  
Challenge onRegistrationPage

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and \_ (underscore).

**Type**  
☒ Regular rule



**If a request** matches the statement ▼

**Statement**

Inspect

URI path ▼

Match type

Starts with string ▼

String to match

/register

Text transformation

AWS WAF applies all transformations to the request before evaluating it. If multiple text transformations are added, then text transformations are applied in the order presented below with the top of the list being applied first.

None ▼

**Add text transformation**

You can add up to 10 text transformations.


### Captcha validity period

Once the user can complete the challenge capture then they will validity period of three days before they will have to complete another captcha challenge.

Choose an action to take when a request matches the statements above.

- ☐ Allow
- ☐ Block
- ☐ Count
- ☒ CAPTCHA
- ☐ Challenge

#### Immunity time

Specify how long a CAPTCHA token can be used after it's created. The AWS WAF default setting is 300 seconds. You can configure this at the web ACL level and the rule level. The web ACL configuration applies to all rules that don't have their own setting. Additional pricing applies. See [AWS WAF Pricing](#) 

250000 seconds

Enter an integer from 60 to 259,200. (259,200 seconds is 3 days.)

☒ Set a custom immunity time for this rule

► Custom request - *optional*

► Add label - *optional*

Add labels to requests that match this rule. Rules that are evaluated later in the same web ACL can reference the labels that this rule adds.

### Adding protection against SQL Injections Rule

Amazon Web Services protects web applications host on the EC2 instances by inspecting the contents of the query from a request. The web application firewall can find SQL injections payload and block the requests. WAF also analyses queries with keywords commonly used in SQL injection attacks such as OR, UNION, SELECT, and DROP. (LinkedIn, n.d.)

#### If a request matches any rule in the rule group

##### **AWSManagedRulesSQLiRuleSet**

Managed rule group name

**AWSManagedRulesSQLiRuleSet**

Vendor name

**AWS**

Version

**Default (using Version\_1.1)**

Capacity

**200**

Description

Contains rules that allow you to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries.

### Adding Rules for PHP application

The rules are set to block PHP vulnerabilities to the application to prevent security vulnerabilities to the application. The AWSManagedRulesPHPRuleSet contains rules to block requests which may be exploiting vulnerabilities related to PHP which may be injections of harmful PHP functions. The rule prevents exploiting from PHP scripts to execute codes.

##### **AWSManagedRulesPHPRuleSet**

Managed rule group name

**AWSManagedRulesPHPRuleSet**

Vendor name

**AWS**

Version

**Default (using Version\_2.0)**

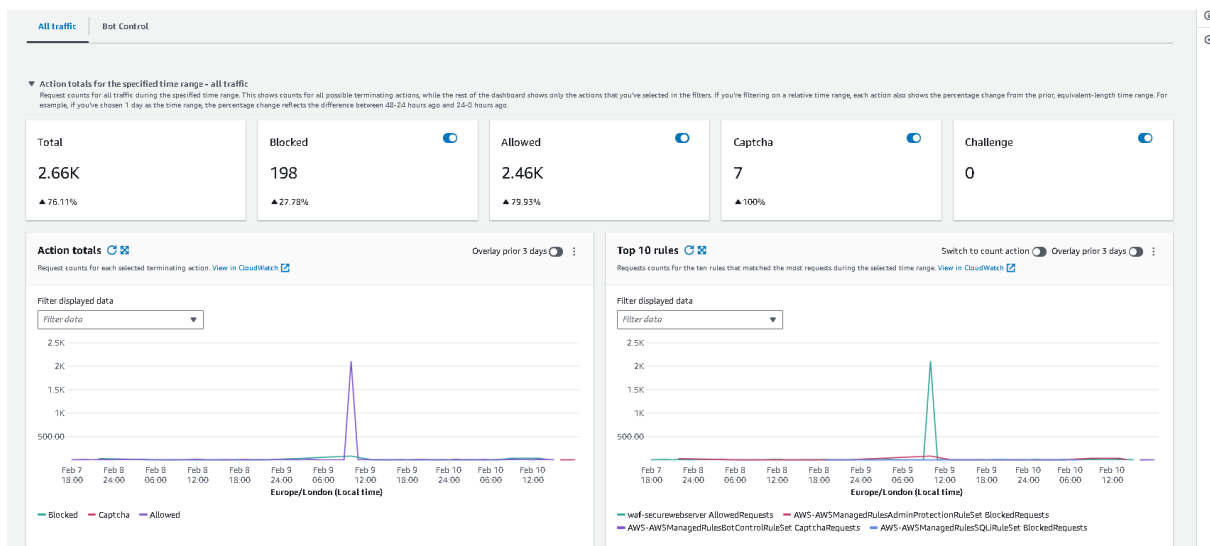
Capacity

**100**

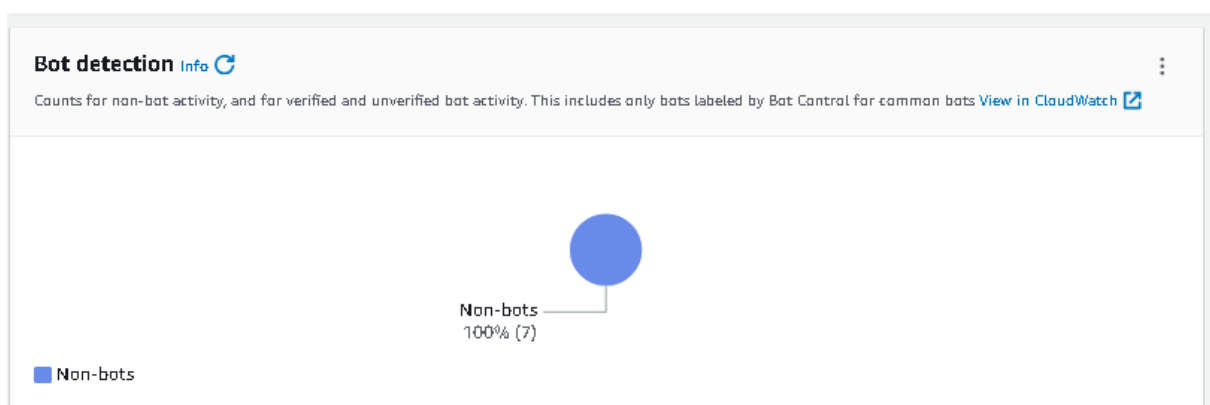
## Evaluation and Results based on Amazon Web Applications Firewall analytics in the last three days.

The ACL contains data based on the requests of users on the web application hosted on the EC2 instance. The data also contains rules such as captcha and shows how many captcha challenges has been used for users. The data of blocked request are displayed in the ACL analytics which means the Web Application Firewall is actively working by blocking requests which can be PHP scripts that are harmful to the application, or malicious bots are trying to access the web application.

All the traffic that are evaluated are specified on a time range. The ACL shows the counts of all possible terminating actions, and other part of the dashboard provides information on the actions which have been selected within the filters selected. In total, there have been two thousand and sixty-six request and one hundred and ninety-eight of the requests have been blocked due to possible harmful PHP scripts, SQL injections and bots which can potentially damage the web application. Two thousand and forty-six requests out the two thousand and sixty requests have been allowed because they have past the security requirements and rules set on the Web application firewall, which means it is a safe connection between the client and the web application. Seven of the requests have been challenged on the Captcha method to validate if the user is not a bot.

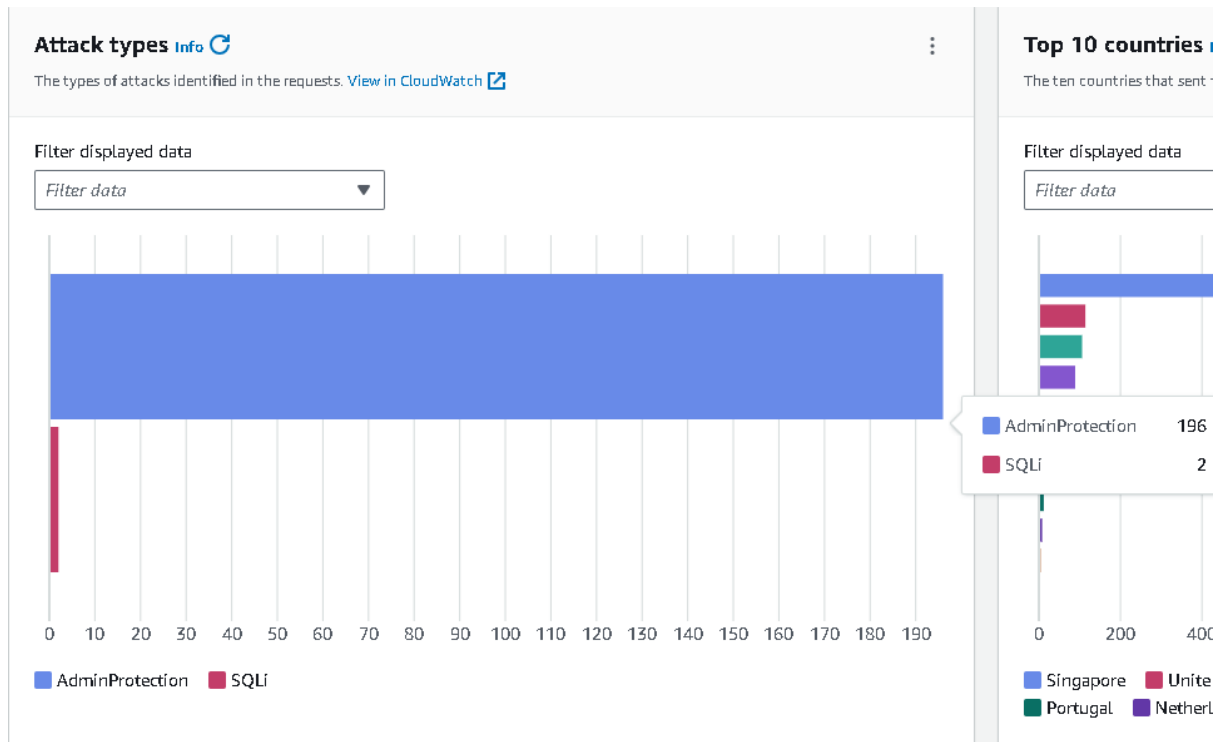


The ACL provides a pie chart for evaluating connections which may be bot requests. According to the bot detection, one hundred percent of the of bot detection analytics are non bots.



## Attack Types

ACL provides a panel for attacks type which are identified by the AWS WAF in the requests. AWS WAF uses the rules that have been set on the rule group and provides analytics for the attack types. There have been two SQL injection attacks and one hundred and ninety-six Admin attacks. All the attacks are not able to affect the web application as they have been prevented by web application firewall.



## Evaluation and Results based on Amazon Web Applications Firewall analytics in the last week.

There have been five thousand and twenty-two requests to access the web application. Out of the five thousand and twenty-two requests, three hundred and ninety requests have been blocked meaning four thousand eight hundred and thirty requests have been allowed.

## Conclusion

The Amazon Web Application Firewall is effective in protecting the web application against SQL injection attacks, AdminProtection, Cross-site Scripting attacks and malicious PHP scripts from stealing data and possibly damage the whole system infrastructure. Amazon's Web Application Firewall is great for companies and developers for protecting the web application and should consider using their security method for protecting confidential information and ensuring the application does not take any Denial-of-Service attacks.

## Evaluation for security vulnerabilities via a vulnerability scanner

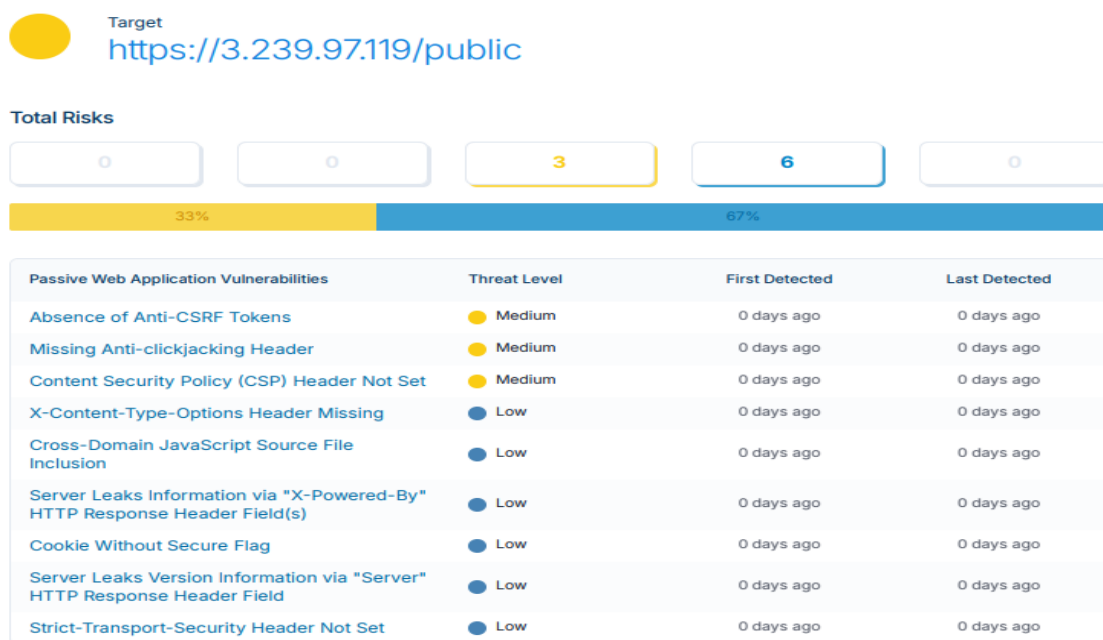
The HostScan Vulnerability has conducted a vulnerability scan by scanning the Web EC2 Hosting IP address with a report for evaluating whole applications based on low threats and high threats status for Absence of Anti-CSRF Tokens, Content Security Policy Header Not Set, Missing Secure Flag, Cooke without Secure Flags, and other possible security threats.

This Report is from OWASP ZAP, in this report they cover threat status according to the security risks mentioned above.

Note that Public IP address changes because the AMAZON EC2 Instances IP addresses often change when starting up an EC2 Instance. The IP address changes and never stays the same for this instance.

### 2.2 Target Breakdowns

The risks discovered for each target.



### Conclusions based on the Vulnerability Scanner report.

The Vulnerability Scanner has found that there are three medium threats with the web application which are Absence of Anti-CSRF Tokens, Missing Anti-clickjacking Header, and Content Security Policy Header Not Set. Based on this information, the web application has minor threats which need to be fixed before becoming one hundred percent, even a medium threat can potentially cause a negative impact. In Conclusion, the web application is secure but needs minor fixes to make the application does not have any vulnerabilities.

## References

console.cloud.google.com. (n.d.). *Google Cloud console*. [online] Available at: <https://console.cloud.google.com/marketplace/product/cloud-infrastructure-services/xampp-server-2019?project=cs012-1808236> [Accessed 19 Jan. 2024].

Enterprise Desktop. (n.d.). *What is RDP? Remote Desktop Protocol Explained*. [online] Available at: [https://www.techtarget.com/searchenterprisedesktop/definition/Remote-Desktop-Protocol-RDP#:~:text=Remote%20desktop%20protocol%20\(RDP\)%20is](https://www.techtarget.com/searchenterprisedesktop/definition/Remote-Desktop-Protocol-RDP#:~:text=Remote%20desktop%20protocol%20(RDP)%20is).

docs.aws.amazon.com. (n.d.). *Working with Amazon EC2 key pairs - AWS SDK for .NET*. [online] Available at: <https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/key-pairs.html> [Accessed 19 Jan. 2024].

Amazon Web Services, Inc. (n.d.). *Private Cloud - Amazon VPC - AWS*. [online] Available at: [https://aws.amazon.com/vpc/#:~:text=Amazon%20Virtual%20Private%20Cloud%20\(Amazon](https://aws.amazon.com/vpc/#:~:text=Amazon%20Virtual%20Private%20Cloud%20(Amazon).

LinkedIn. (n.d.). *LinkedIn Login, Sign in*. [online] Available at: <https://www.linkedin.com/pulse/how-does-waf-work-indusface-2c/> [Accessed 30 Jan. 2024].