Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies

Master's Thesis

Design of security strategy of selected (web) application

Written by (Author): Bc. Muhammad Usman

Thesis Supervisor: Ing. Martin Havránek, Ph.D.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Muhammad Usman, BSc

Informatics

Thesis title

Design of security strategy of selected (web) application

Objectives of thesis

The main objective of this thesis is to determine the vulnerabilities and removing from the security design of the selected web application.
The partial goals of this thesis are the following:
- To explain and determine the vulnerabilities in the security design
- To conduct analysis by putting new solutions in the design
- To check the expected results of newly implemented solutions, dummy attacking
- Generalization of protection against vulnerabilities.
- Continuous testing of the system

Methodology

Design of a strategy for continuous testing of the selected application against vulnerabilities.

The actual elaboration of the diploma thesis will be preceded by analysis and studying of the selected Web application.

Based on the search, the risks of the specific environment and the vulnerabilities of the specific application will be identified.

In the practical part, the design of the security strategy will be tested against the above risks and vulnerable- cities. Based on the tested procedures, a methodology for testing the stability and security of the selected web application will be proposed.

The proposed measures will be summarized in the conclusions of the work.

The proposed extent of the thesis

60-80p.

Keywords

Security, strategy, web, application, vulnerability, software, testing, remote, code, execution

Recommended information sources

Detection of intrusions and malware, and vulnerability assessment: 16th international conference, DIMVA 2019, Gothenburg, Sweden, June 19–20, 2019, proceedings [online]. Roberto PERDISCI et al. Cham: Springer International Publishing. ISBN 0302-9743

KOFLER, M. MySQL Championship 5: [Complete Web Developer's Guide].

SQL Injection. w3schools: The World's Largest Web Developer Site. [online]. 1999-2016 [cit. 2016-01-10]. Available from: http://www.w3schools.com/sql/sql_injection.asp

Types of Security Vulnerabilities. Mac Developer Library. [online]. 11.2.2014 [cit. 2015- 12-03].

ZHANG, Bing et al. Efficiency and Effectiveness of Web Application Vulnerability Detection Approaches: A Review. ACM Computing Surveys. 2022, vol. 54, no. 9, s. 1-35. ISSN 0360-0300

Expected date of thesis defence

2022/23 SS – FEM

The Diploma Thesis Supervisor

Ing. Martin Havránek, Ph.D.

Supervising department

Department of Information Technologies

Electronic approval: 14. 7. 2022

doc. Ing. Jiří Vaněk, Ph.D.

Head of the department

Electronic approval: 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 29.03.2024

Declaration

I hereby declare that I have worked on my master's thesis titled " Design of Security Strategy of Selected (web) application " by myself and I have used only the sources mentioned at the end of the thesis. As the author of the master's thesis, I declare that the thesis does not break any copyrights. I also declare that this dissertation is my original work and has not been submitted before to any institution for assessment purposes.

I understand that the provision of incorrect information may have legal consequences.

In Prague on 31/03/2024 ____________________________

Acknowledgement

I would like to express my sincere gratitude to everyone who contributed to the completion of my thesis titled "Design of Security Strategy for a Selected (Web) Application." This research endeavour would not have been possible without the guidance and support of my thesis advisor, whose expertise and insights significantly shaped the direction of the study. I extend my appreciation to the academic faculty for their valuable feedback and constructive criticism during the various stages of the research process. Additionally, I would like to acknowledge the developers and security experts who shared their knowledge and experiences, contributing to a more comprehensive understanding of web application security. Special thanks to my family and friends for their unwavering encouragement and understanding throughout this academic journey. Their support provided the motivation needed to navigate the challenges inherent in researching and designing a robust security strategy for a web application.

Abstract

In the current landscape of digital interaction, it is essential to protect web applications from security threats. This study dives into the complexities of designing a powerful security structure for a chosen web application, zeroing in on user registration and login processes braced against SQL injection assaults. The user begins the multi-step registration process by providing an email address. They then receive a one-time password (OTP) for verification. Consequently, the client chooses distinct colour patterns for additional authentication. The user then moves on to the login page, where they enter their credentials and obtain a second OTP for further verification. A mathematical CAPTCHA further affirms the identity of the user. The user confirms their chosen colour patterns once more after OTP and CAPTCHA verification, guaranteeing a layered authentication process.

The study then guides users to a presentation page demonstrating SQL injection weaknesses. With two queries, it exhibits the possibility for corruption and the efficiency of preventive strategies such as parameterized unapproved queries. The following pages depict the results of SQL injection: one illustrates unsanctioned access aided by a basic SQL injection code, while the other exhibits the strength of systems sustained against such assaults.

This nuanced process highlights the basic significance of strong security efforts in user authentication systems, especially in moderating SQL injection weaknesses. By incorporating preventive measures at different stages, including verification of OTP, parameterized queries, and CAPTCHA challenges, the suggested technique braces the application against possible threats. Moreover, it stresses the meaning of user awareness and training in perceiving and defeating cyber threats. This exploration offers practical information and guidelines for administrators, developers, and personnel involved in decision-making trusted with shielding web applications during evolving cybersecurity challenges.

Keywords: Web application security, Security Strategy, Cybersecurity, Data encryption, Access controls, Authentication protocols, Secure coding practices, Intrusion detection systems, User education, Threat landscape.

Abstraktní

V současném prostředí digitální interakce je nezbytné chránit webové aplikace před bezpečnostními hrozbami. Tato studie se ponoří do složitosti návrhu výkonné bezpečnostní struktury pro vybranou webovou aplikaci a zaměřuje se na procesy registrace a přihlašování uživatelů, které jsou chráněny před útoky SQL injection. Uživatel zahájí vícekrokový registrační proces zadáním e-mailové adresy. Poté obdrží jednorázové heslo (OTP) pro ověření. Následně si klient vybere odlišné barevné vzory pro dodatečné ověření. Uživatel poté přejde na přihlašovací stránku, kde zadá své přihlašovací údaje a získá druhé OTP pro další ověření. Matematická CAPTCHA dále potvrzuje identitu uživatele. Uživatel potvrdí své zvolené barevné vzory ještě jednou po ověření OTP a CAPTCHA, což zaručuje vrstvený proces ověřování.

Studie pak uživatele zavede na prezentační stránku demonstrující slabiny SQL injection. Se dvěma dotazy ukazuje možnost korupce a účinnost preventivních strategií, jako jsou parametrizované neschválené dotazy. Následující stránky zobrazují výsledky SQL injection: jedna ilustruje nepovolený přístup pomocí základního kódu SQL injection, zatímco druhá ukazuje sílu systémů odolávajících takovým útokům.

Tento nuancovaný proces zdůrazňuje základní význam silných bezpečnostních snah v systémech autentizace uživatelů, zejména při zmírňování slabých stránek vkládání SQL. Začleněním preventivních opatření v různých fázích, včetně ověření OTP, parametrizovaných dotazů a výzev CAPTCHA, navrhovaná technika chrání aplikaci před možnými hrozbami. Navíc zdůrazňuje význam informovanosti uživatelů a školení ve vnímání a potírání kybernetických hrozeb. Tento průzkum nabízí praktické informace a pokyny pro administrátory, vývojáře a zaměstnance podílející se na rozhodování, kterému důvěřuje ochrana webových aplikací během vyvíjejících se výzev v oblasti kybernetické bezpečnosti.

Klíčová slova: Zabezpečení webových aplikací, Bezpečnostní strategie, Kybernetická bezpečnost, Šifrování dat, Řízení přístupu, Autentizační protokoly, Praktiky bezpečného kódování, Systémy detekce narušení, Vzdělávání uživatelů, Krajina hrozeb

Table of Contents

List of Figures

Chapter 1: Introduction

The thesis title, "Design of Security Strategy for Selected (Web) Application," encapsulates a critical exploration into the realm of safeguarding web applications against potential threats and vulnerabilities (Quyen and Van, 2023). In an era dominated by the digital landscape, the significance of robust security measures cannot be overstated, particularly in the context of web-based platforms that handle sensitive information.

In today's interconnected world, web applications serve as conduits for a multitude of transactions and exchanges, ranging from personal data storage to financial transactions (Mishra and Kaushik, 2021). However, this increased reliance on web applications also exposes them to an array of security risks, including but not limited to unauthorized access, data breaches, and malicious attacks. The evolving nature of cyber threats necessitates a proactive and adaptive approach to security strategies, emphasizing the importance of designing a comprehensive framework.

The chosen focus on a selected web application underscores the specificity and targeted nature of this thesis (Alshami et al., 2023). Rather than adopting a generic approach, the research will delve into the intricacies of a particular web application, considering its unique functionalities, user interactions, and potential vulnerabilities. This tailored strategy aims to offer insights that can be directly applied to enhance the security posture of the chosen application.

The research journey will involve a meticulous examination of existing security frameworks, methodologies, and best practices (Oladoyinbo et al., 2023). By doing so, the thesis aims to contribute to the existing body of knowledge in the field of web application security. Through the synthesis of theoretical frameworks and practical insights, the goal is to propose a well-defined and effective security strategy that can be implemented to fortify the selected web application against contemporary and future threats.

1.1 <u>Overview</u>

The thesis titled "Design of Security Strategy for Selected Web Application" involves a comprehensive exploration of the intricacies surrounding the cybersecurity landscape in the context of a specific web application. The primary objective is to formulate a robust security strategy that addresses potential vulnerabilities and safeguards sensitive data from various threats (Bandari, 2023). The research delves into the identification of potential risks and vulnerabilities within the selected web application, considering factors such as user authentication, data encryption, and secure communication protocols. The analysis also encompasses a thorough examination of contemporary security standards and best practices relevant to web applications. The thesis explores the implementation of security measures, such as intrusion detection systems, firewalls, and access controls, tailored to the specific characteristics of the chosen web application. The study emphasizes a proactive approach to security, including continuous monitoring and periodic security audits.

1.2 Research Gap

The research on the "Design of Security Strategy for Selected (Web) Application" aims to address several critical gaps in the existing literature. Firstly, there is a notable dearth of comprehensive studies that specifically focus on the design aspects of security strategies for web applications (Malhotra et al., 2021). While numerous research works delve into general web application security, the lack of a concentrated effort on the strategic design of security measures is evident. Furthermore, the evolving nature of web technologies introduces novel challenges that demand constant adaptation of security strategies. The literature review reveals a gap in the exploration of how emerging technologies, such as progressive web applications or server less architectures, impact security design (Oztemel and Gursev, 2018). Understanding these implications is crucial for ensuring the resilience of security strategies against contemporary threats. Additionally, there is a need for research that provides a holistic perspective on security, considering not only technical aspects but also human factors.

1.3 Aim and Objectives

Aim: The aim of this study is to develop an effective security strategy for a selected web application.

Objectives

The main objective of this thesis is to determine the vulnerabilities and removing from the security design of the selected web application. The partial goals of this thesis are the following:

- To explain and determine the vulnerabilities in the security design.
- To conduct analysis by putting new solutions in the design.
- To check the expected results of newly implemented solutions, dummy attacking.
- Generalization of protection against vulnerabilities.

- Continuous testing of the system.

1.3 Research Scope

The research scope for the thesis titled "Design of Security Strategy for Selected (Web) Application" encompasses a comprehensive investigation into the various facets of web application security. The study will delve into the identification and analysis of potential vulnerabilities within the selected web application, aiming to understand the evolving threat landscape (Nazah et al., 2020). This includes examining common security risks such as SQL injection, cross-site scripting, and other vulnerabilities that may compromise data integrity, confidentiality, and availability.

The research will explore contemporary security measures and technologies, evaluating their effectiveness in mitigating identified risks. This involves an in-depth examination of encryption protocols, access controls, and authentication mechanisms (El Sibai et al., 2019). The study will also consider the impact of emerging technologies such as cloud computing and Internet of Things (IoT) on the security posture of the selected web application. The thesis will assess regulatory frameworks and compliance requirements relevant to web application security, ensuring that the designed security strategy aligns with industry standards and legal obligations. The research scope extends to proposing a customized security framework tailored to the specific needs and characteristics of the selected web application, fostering a proactive approach to risk mitigation.

1.4 Rationale

The chosen topic, "Design of Security Strategy for a Selected Web Application," is crucial in the contemporary digital landscape. With the increasing frequency and sophistication of cyber threats, safeguarding web applications is imperative. This topic allows an in-depth exploration of various security measures, risk assessments, and implementation strategies tailored to a specific web application (World Health Organization, 2021). By delving into this area, one can address the unique challenges posed by the online environment, ensuring robust protection against potential vulnerabilities. Consequently, this research contributes to the broader understanding of web application security, fostering the development of effective strategies to mitigate cyber risks.

1.5 Methodology

To design a security strategy for a selected web application using a qualitative approach, employ a methodology grounded in secondary research and interpretivism philosophy (Tracy, 2019). Utilize reputable sources such as Google Scholar, online articles, journals, and newspapers to gather comprehensive insights. Begin by conducting a thorough literature review to understand existing security frameworks and vulnerabilities. Analyse case studies and real-world examples to identify best practices and pitfalls. Apply interpretivism to interpret the subjective nature of security challenges. Synthesize findings to formulate a robust security strategy, considering the evolving threat landscape. This qualitative approach ensures a nuanced understanding and effective design tailored to the specific context of the selected web application.

1.6 Limitations

Designing a security strategy for a web application is a complex endeavour with inherent limitations. Firstly, the ever-evolving landscape of cyber threats poses a significant challenge, as new and sophisticated attack vectors may outpace the static nature of a predefined strategy (Al-Doori, 2023). The need for constant updates and adaptations to emerging threats underscores the dynamic nature of cybersecurity. Secondly, the unpredictability of user behaviour introduces a human element that can be exploited through social engineering attacks. Despite robust technical measures, the success of security strategies is contingent upon user awareness and adherence to best practices, making it challenging to eliminate the human factor entirely.

Resource constraints pose a practical limitation to the implementation of comprehensive security measures (Hughes et al., 2019). Organizations may face limitations in terms of budget, expertise, or technological capabilities, hindering their ability to deploy state-of-the-art security solutions. Moreover, the reliance on third-party components introduces an element of vulnerability beyond direct control. While these components may enhance functionality, they can also become potential points of exploitation if not rigorously assessed for security. Achieving a delicate balance between stringent security and user experience presents a perpetual challenge (Distler et al., 2020). Overly complex security measures can impede usability, potentially leading to user frustration and resistance to adherence.

Chapter 2: Literature Review

2.1 Web Application Vulnerabilities

Vulnerabilities in a web application occur when there is an opening in the program's defences. Since of simple oversights like these, such as neglecting to verify or sanitise form inputs, utilising unpatched editions of web servers, or having weak security integrated into the application itself, these vulnerabilities have persisted for a long period of time (Aborujilah et al., 2022). These flaws are distinct from the more prevalent network and asset vulnerabilities. Web applications are vulnerable to these attacks because they are designed to facilitate communication between users located on different networks.

It is crucial to explore beyond typical vulnerability scanners when trying to detect holes in an organization's application security, since there is security for web applications options created particularly for apps.

2.1.2 SQL Injection Attacks

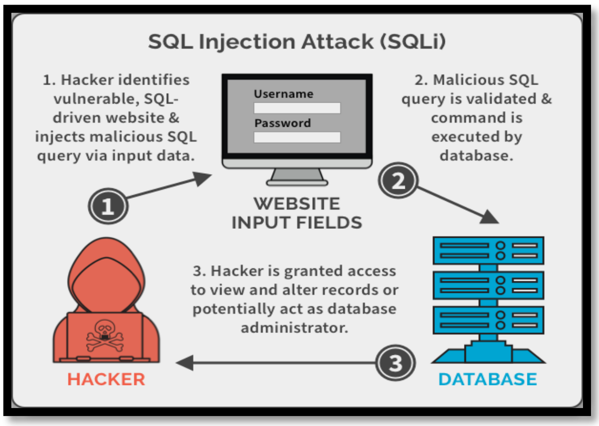

SQL Injection Attack (SQLi)
1. Hacker identifies vulnerable, SQL-driven website & injects malicious SQL query via input data.
Username
Password
2. Malicious SQL query is validated & command is executed by database.
1
2
WEBSITE INPUT FIELDS
3. Hacker is granted access to view and alter records or potentially act as database administrator.
3
HACKER
DATABASE

Figure 1: SQL Injection Attack

Image Source: (Alghawazi, Alghazzawi and Alarifi, 2022)

The widespread use of Structured Query Language (SQL) for data management and application navigation has led to the development of techniques through which malicious actors may inject their own SQL instructions into databases (Alghawazi, Alghazzawi and Alarifi, 2022). In addition to modifying, stealing, or erasing data, these instructions may provide the hacker administrative privileges on the system. Structured Query Language, or SQL for short, is a computer language for interacting with databases. Its actual pronunciation is ess-cue-el, although most people just call it "sequel." SQL is widely used to handle the information stored in databases on servers. These servers host vital information for websites and services.

This sort of server is a prime target for a SQL injection attack, which uses malicious code to coerce the server into revealing data it usually would not. This is a serious issue if the server keeps sensitive information about the website's or app's users, including such credit card information, login credentials, and other personal information that might be used to steal money or identity.

It is common for SQL injection attacks to succeed when a programme fails to properly sanitise user input by removing any code that may be interpreted as SQL (Aliero et al., 2019). An attacker might, for instance, visit a website and, using the site's search box, enter code which might cause the site's SQL server to leak every single usernames and passwords it has saved.

2.1.3 Cross-Site Scripting (XSS)

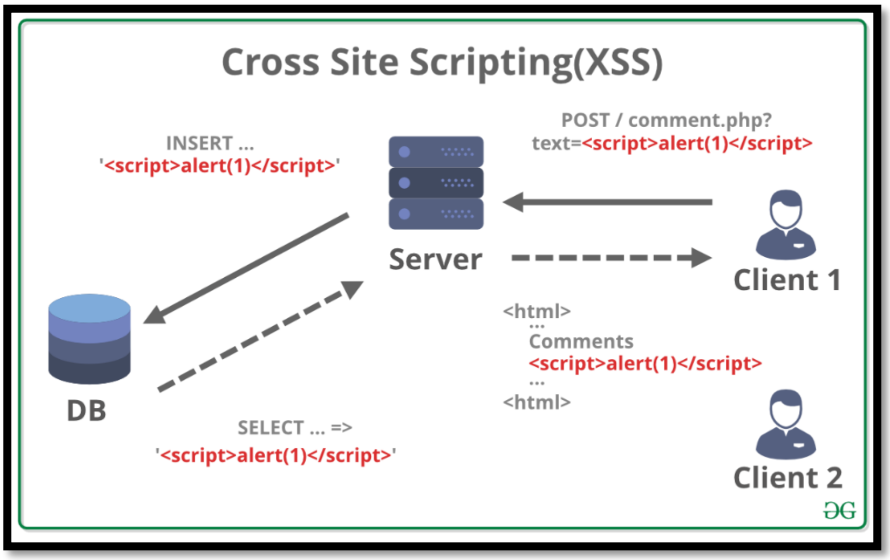

Cross Site Scripting(XSS)
INSERT ...
'<script>alert(1)</script>'
Server
POST / comment.php?
text=<script>alert(1)</script>
Client 1
DB
<html>
...
Comments
<script>alert(1)</script>
...
<html>
SELECT ... =>
'<script>alert(1)</script>'
Client 2

Figure 2 Cross-Site Scripting

Image Source: (ALMEIDA, 2021)

An attacker uses SQL injection to get access to a website and steal information contained on it, such as user passwords or financial information. A cross-site scripting assault, however, is aimed squarely at the visitors of the website under attack. This kind of attack is related to SQL injection attacks in that it involves inserting malicious code into a website or a web app (ALMEIDA, 2021). On the other hand, in this situation, the malicious file the hacker has inserted only executes in the internet device when they access the targeted website, and it targets the user directly.

Injecting malicious code into an input field is a typical method of deploying a cross-site scripting attack, that will be executed whenever a visitor views the infected website. If they write a comment on a weblog, for instance, they might potentially infect the site with malicious JavaScript by linking to it.

The credibility of an online business may be severely damaged by cross-site scripting assaults since they put consumers' data at danger invisibly. Cross-site scripting may steal any confidential information a user enters into a website or application, including login passwords, credit card numbers, and more.

2.1.4 Cross-Site Request Forgery (CSRF)

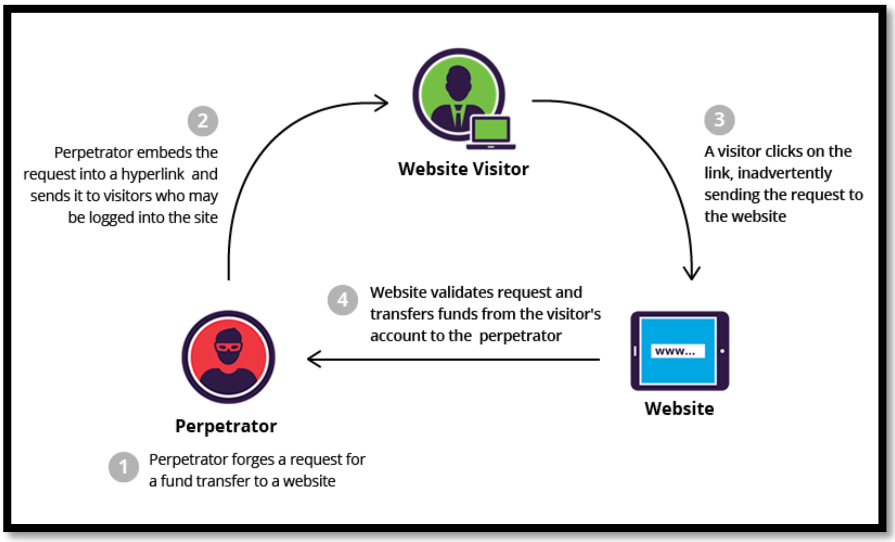

Website Visitor
2
Perpetrator embeds the request into a hyperlink and sends it to visitors who may be logged into the site
3
A visitor clicks on the link, inadvertently sending the request to the website
4
Website validates request and transfers funds from the visitor's account to the perpetrator
www...
Website
Perpetrator
1
Perpetrator forges a request for a fund transfer to a website

Figure 3: Cross-Site Request Forgery

Image Source: (Ashari et al., 2023)

One example of a web application vulnerability is the Cross-Site Request Forgery (CSRF) attack, in which a user is tricked into doing an unwanted action while logged into the app (Ashari et al., 2023). Whenever a user is tricked into making a malicious request to a web-based program, the programme will have already decided that the target and the computer are trustworthy, and thus it will carry out an action requested by the hacker. The uses for this range from innocuous tricks on users to the transmission of illegal funds.

Having extensive validation procedures in place for everybody who may view pages on your site or app is one approach to reduce the risk of attack, particularly for social media and community sites. Because of this, they will be able to confirm the user's identity by identifying the browsers and experience.

As many methods as there are for a hacker to get into an app because of web software vulnerabilities, there are equally many ways to protect oneself from them (Babu, Raj and Devi, 2020). With the aid of vulnerability scanning for web - based applications technologies, some of the most popular apps can be tracked. One can lessen the likelihood of getting hacked by utilizing these scanners, which point out vulnerable spots in existing apps and provide guidance on how to fix them.

2.2 Web Security Vulnerabilities and their solutions

Unfortunately, many businesses do not start prioritising online security best practises until after they have already had a security incident (Calzavara et al., 2020). Companies need to take both preventative and reactive measures to protect the website from attacks. The goal is to create a sense of safety concern in the viewer. In specifically, this manual is geared at helping its readers avoid 10 particularly pervasive and problematic online security errors.

2.2.1 Authentication and Authorization: A Cyber Security Primer

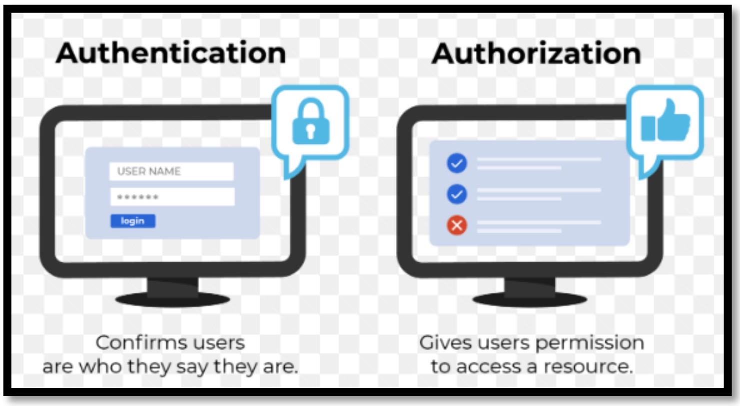

Authentication
Authorization
USER NAME
******
login
Confirms users are who they say they are.
Gives users permission to access a resource.

Figure 4 Authentication and Authorization

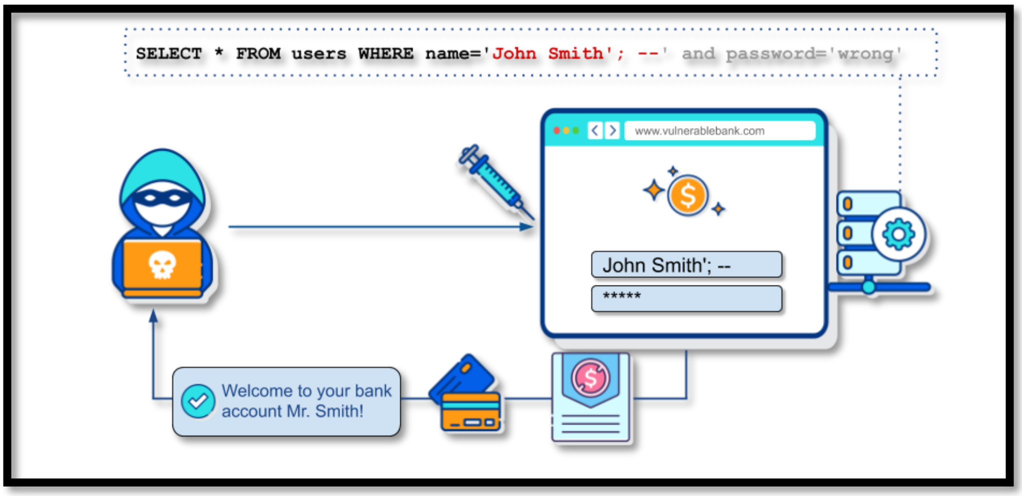
Image Source: (Dong et al., 2019)

Many developers and IT specialists are confused about the difference between authorisation and authentication. The confusion caused by shortening both phrases to "auth" does not help matters.

- Authentication — Making sure a user is who they claim to be.
- Authorization is the process of allowing a user to enter or utilise a restricted resource or carry out a restricted operation.

To rephrase, authentication refers to verifying the identity of an individual or group, whereas authorisation refers to limiting the actions of a certain individual or group (Dong et al., 2019). Let us look at ten of the most prevalent security holes on the web.

2.2.2 Injection Flaws



```
SELECT * FROM users WHERE name='John Smith'; --' and password='wrong'
```

Figure 5: Injection Flaws

Image Source: (Kumar and Goyal, 2019)

One of the most common causes of injection issues is a failure to properly filter out malicious data. Problems with injection can be introduced by the LDAP server, providing unedited data to either the database server (SQL injection), the browser, or anywhere else. Sensitive information might be exposed if an attacker could inject instructions into users' browsers and take control of them.

Incoming data from unknown sources must be filtered by the software, preferably through a whitelist. Using a blacklisted for this reason is not recommended due to the complexity involved in configuring it properly (Kumar and Goyal, 2019). A similar argument is made concerning the ease with which a hacker may circumvent a blacklist. Examples of blacklists gone wrong may usually be found in antivirus software. Incorrect patterns will not be matched.

Prevention

To prevent injection, one should do nothing more than "simply" filter the input and think about whether senders may be trusted. Because they must process every information unless it is trustworthy, filtering is a massive task (Kumi et al., 2020). Even if they successfully filter

999 of the 1,000 possible inputs into the system, there is always the risk that one of the remaining fields will be the weak point and bring the whole thing crashing down.

To add insult to injury, it is also risky to use Second Order SQL Injection to inject the results of one SQL query into another (Latchoumi, Reddy and Balamurugan, 2020). Since the database is reliable, it may appear reasonable. However, if the boundary is not secure, the information they get might have been spoofed.

Given the difficulty of effective filtering, our framework's built-in filtering features are a safe bet. They have been tested extensively and shown to be effective. Using a framework may improve security protocols, therefore it's worth considering if they aren't already.

2.2.3 Broken Authentication

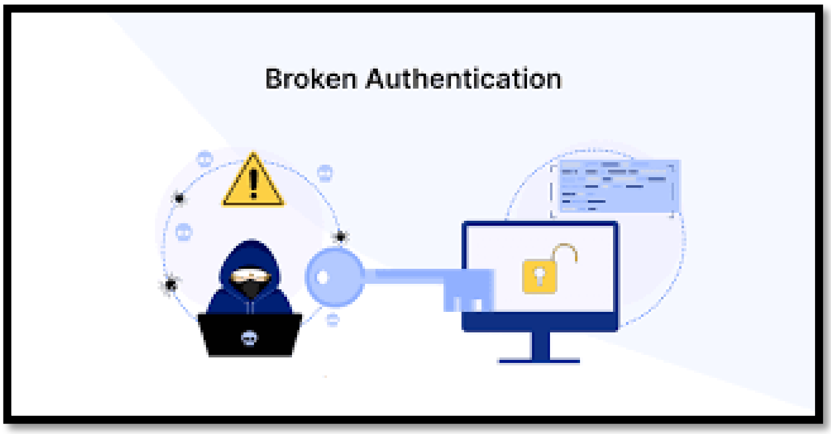


Figure 6 Broken Authentication

Image Source: (Le et al., 2019)

There is no guarantee that issues related to compromised authentication all have the same origin (Le et al., 2019). Constructing their own secure authentication key is a tall order, and so is not advised. Some of the many potential dangers are as follows.

- The session ID may be leaked in the referrer header if it were included in the URL.
- It is possible that passwords are not encrypted while being stored or sent.
- Repetitive Session IDs may make it too simple for hackers to get access.
- This session may be fixable.
- It is possible for a session to be hijacked if timeouts are not set properly, HTTP is used (which doesn't provide SSL security), etc.

Prevention

Using a platform is the simplest solution to the problem of broken authentication on the web (Loureiro, 2021). Be very careful and learn about all the possible problems that might occur if users write their own code.

2.2.4 Cross-Site Scripting (XSS)

An attacker gives the web app malicious JavaScript tags as input. The user's browser will execute this unclean input if it were returned to them (Marashdih et al., 2019). This is a common issue with input sanitization, which is a subset of injection errors. The goal of using CSS might range from just getting a client to click on a hyperlink to something significantly sinister. For instance, the script may be set to start on page load and then used to send the cookies to an attacker.

Prevention

Do not send any HTML tags back to the client. This would safeguard users against HTML injection, in which malicious code is injected into the page in plain HTML format (Nasereddin et al., 2021). This approach requires changing the value of all HTML entities to anything else. For instance, one may use the script to return the script format. Another option is to use regular expressions to remove HTML elements such as and >. However, this might be risky since certain web browsers may not recognise significantly damaged HTML. It's preferable to replace all characters with their escaped equivalents.

2.2.5 Insecure Direct Object References

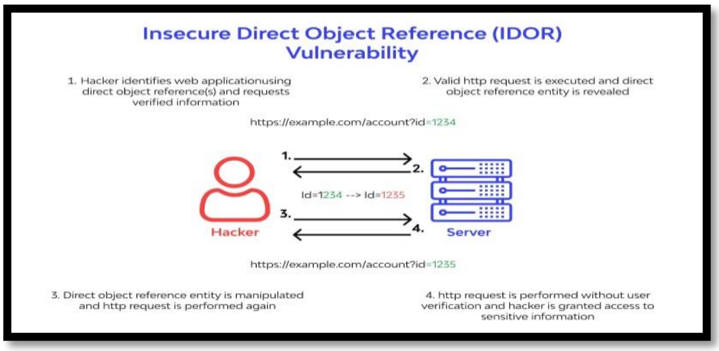

Insecure Direct Object Reference (IDOR) Vulnerability
1. Hacker identifies web applicationusing direct object reference(s) and requests verified information
2. Valid http request is executed and direct object reference entity is revealed
https://example.com/account?id=1234
1.
2.
Id=1234 --> Id=1235
3.
4.
Hacker
Server
https://example.com/account?id=1235
3. Direct object reference entity is manipulated and http request is performed again
4. http request is performed without user verification and hacker is granted access to sensitive information

Figure 7 Insecure Direct Object References

Image Source: (Ovsyannikova and Sidorenko, 2022)

This is a typical example of putting faith in user input and then having to deal with the security risk that comes along with it (Ovsyannikova and Sidorenko, 2022). When an internal item (such a file or a database key) is referenced directly, it leaves us open to attack. If authorisation isn't properly checked or is broken, the attacker may supply this reference and get access.

One such CGI argument is the file name (e.g., download.php?file=something.txt), which is read by the code and used to allow the user to download the file (Palit, Monrose and Polychronakis, 2019). If the coder forgot to include authorisation checks, an attacker may leverage this vulnerability to steal sensitive information from the PHP user's machine (e.g., the application code or random server data like backups). A password reset feature that requests user input to verify their identity is another kind of unsecured direct object reference vulnerability. After the victim visits the legitimate URL, the attacker might change the username field to read "admin" or something similar.

Prevention

Whitelist just the safe options when sufficient user authorisation has been carried out. By avoiding external storage and instead depending on data given by the client through CGI parameters, the risk is generally eliminated. It's a job perfectly adapted for session attributes, which are available in the vast majority of systems.

2.2.6 Security Misconfiguration

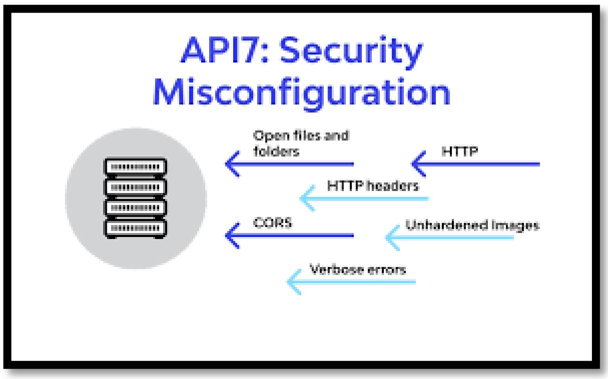


Figure 8 Security misconfiguration

Image Source: (Perez-Cabo et al., 2019)

The frequency with which they see web servers and apps that are poorly configured is high.

- Utilizing a production environment to run a programme with debugging enabled.
- Possessing the potentially dangerous feature of directory display activated on the server (Phokela et al., 2022).
- Using antiquated programmes.
- The process of keeping unneeded services active.

- Using the same password across many accounts.
- Making details about how errors are handled available to attackers.

Prevention

Get a solid "build and deploy" process that can perform tests after deployment, ideally one that is automated (Qiu et al., 2020). Post-commit hooks are the poor man's answer to safety configuration errors since they prevent code with basic credentials and development items built in from being committed.

2.2.7 Sensitive data exposure

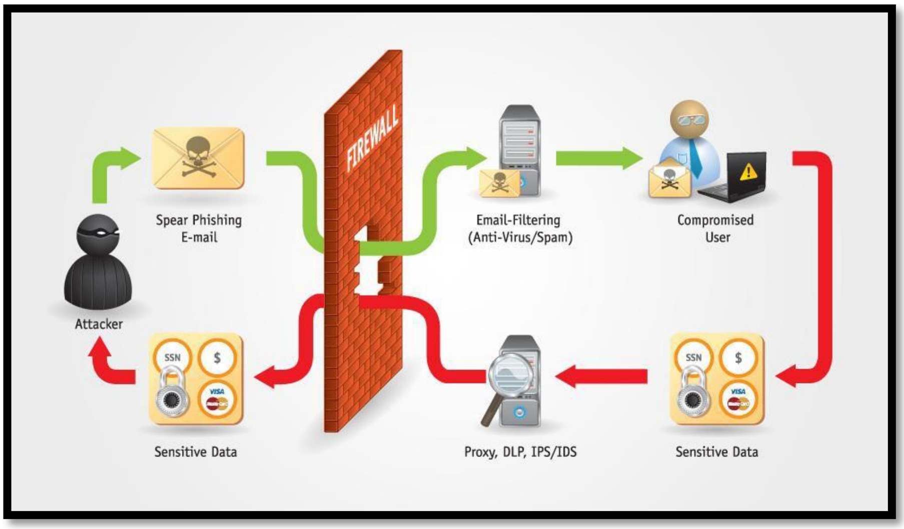


Figure 9: Sensitive data exposure

Image Source: (R Shaikh, 2019)

This cryptography and resources control flaw is a security risk on the web. Private information has to be encrypted while it is in motion or at rest (Rashid and Pajooh, 2019). Nothing will be made an exception to. Passwords and other sensitive information, such as credit card numbers, should be hashed and encrypted during transmission and storage. The crypto/hashing algorithm, of course, cannot be a slouch. AES (256 bits and higher) and RSA are the recommended online security protocols (2048 bits and up).

The importance of preventing session IDs and other sensitive information from being sent in URLs cannot be overstated. The "secure" option must be enabled for cookies containing private information.

Prevention

- Make use of HTTPS secured by a trusted certificate and PFS (Perfect Forward Secrecy). Never give up any personal information over a connection that is not encrypted using HTTPS. Cookies should be marked as secure.
- Get this security hole patched up as soon as possible. Shred private information that won't be needed in the future (Sagrillo et al., 2023). By not storing any credit card information, avoid the hassle of having to be PCI compliant. Join a reputable online payment system like Stripe or Braintree. Confidential information must be stored encrypted, and passwords must be hashed using bcrypt.

2.2.8 Missing Function Level Access Control

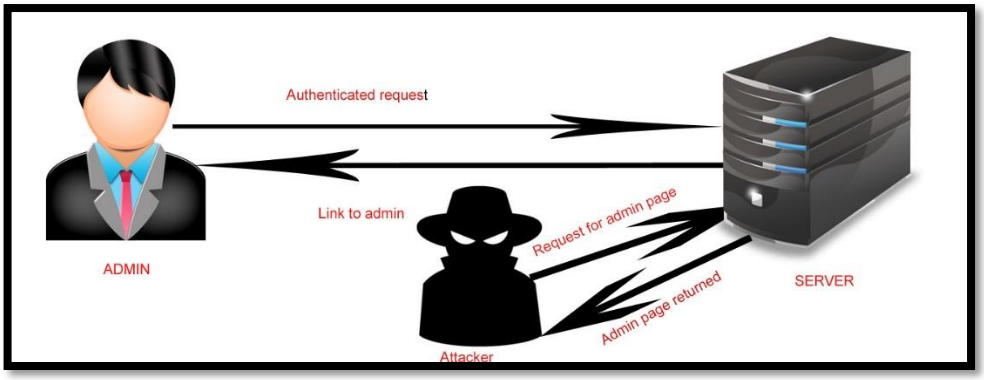


Figure 10: Missing Function Level Access Control

Image Source: (Sai, Buckley and Le Gear, 2019)

This is an error that occurs if the server method was called without first obtaining the necessary permissions. Since the UI is generated on the server, developers often think that the client has no way of accessing features that are not provided by the server (Sai, Buckley and Le Gear, 2019). An attacker may always fabricate a request to the "hidden" feature, so it's not quite that easy. If the necessary capability is not readily available, that will not discourage an attacker. here's a special /admin control panel where the button appears only if the current user has admin privileges. If authorisation is lacking, an attacker may easily find and abuse this feature.

Prevention

Normally, authorisation must be carried out on the server.

2.2.9 Cross-Site Request Forgery (CSRF)

A CSRF attack, also known as a confused deputy attack, occurs when an untrusted third party manipulates the browser into acting inappropriately on the attacker's behalf. Sites that use cross-site request forgery (CSRF) send requests to other sites using the user's browser, cookies, and session (Sarkar, 2021). The confused deputy issue arises when a user is signed into their bank on one tab of their browser, but another tab may be hijacked to have the browser abuse its credentials on the attacker's behalf. The browser acts as the agent, abusing its power to carry out the hacker's will. Please take this as an illustration: Alice plans to drain some of Todd's cash from his wallet and deposit it into her account.

Todd's browser will interpret the snippet as a link to a picture when he revisits Alice's site. The image is requested through an HTTP GET request that is sent by the browser immediately. However, instead of loading a picture in the browser, Todd's bank is instructed to wire Alice $1,500.

It should be noted that this sample not only displays the CSRF vulnerability but also shows how to change the server's state using an operand (safe)HTTP GET request (Sinha and Tripathy, 2019). It is already vulnerable due to this fact alone. For HTTP GET requests to work properly, they must be idempotent, indicating they cannot change the requested resource in any way. Never attempt to modify the server's state using a method that is not retriable.

Prevention

Put the password in a hidden form field that the outside site cannot see. That needs need to check the secret field. To access secure areas of certain websites, users may need to provide a password (Taleby et al., 2017). One thinks this is to safeguard against anyone abusing their abandoned sessions on shared machines.

2.2.10 Using Components with Known Vulnerabilities

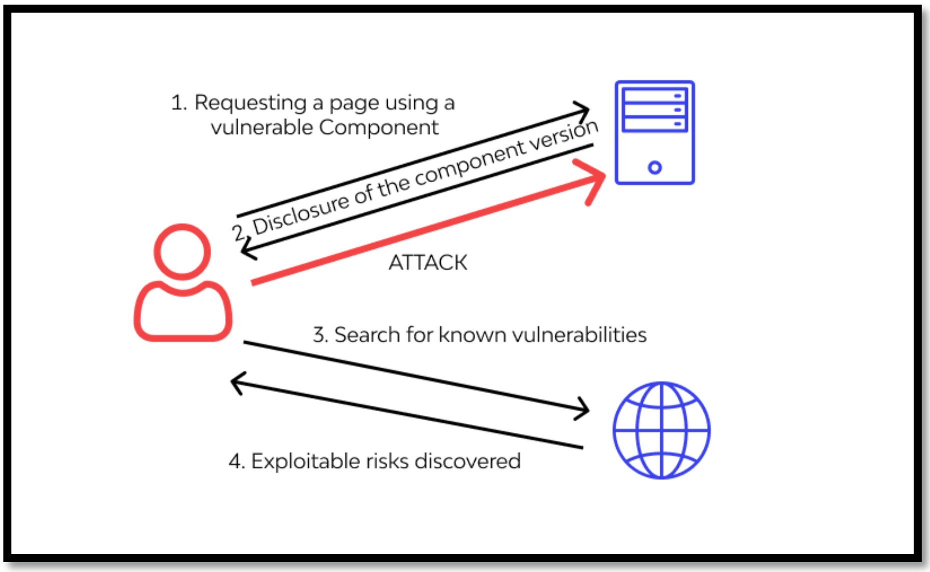

1. Requesting a page using a vulnerable Component
2. Disclosure of the component version
ATTACK
3. Search for known vulnerabilities
4. Exploitable risks discovered

Figure 11: Using Components with Known Vulnerabilities

Image Source: (Tariq et al., 2021)

That is all there is to say about it in the title. This seems like a problem that would fall under the category of deployment or maintenance. Do some investigation and maybe even an audit before implementing new code (Tariq et al., 2021). Although it could be handy to use code from a stranger on a hosting service like GitHub, doing so might leave your website vulnerable to a number of threats. Third-party software (like WordPress plugins) that went unfixed for decades in production is a common source of sites being "owned" (i.e., compromised) by an outsider.

The lesson to be learned here is that software development continues even after an app has been released to the public. Particularly if the programme makes use of third-party or open-source parts, there must be documentation, testing, and strategies for maintaining and updating the software.

Prevention

- Avoid being a lazy programmer who just copies and pastes. It is important to thoroughly examine any new code before incorporating it into the programme, since it might include bugs or even be designed to do harm (Weamie, 2022). This is a common method in which users unknowingly open themselves up to cyber threats.

- Always use the most recent versions of the programmes that rely on and set a schedule to update them frequently. Join the product newsletter to be informed about any security issues.

2.2.11 Invalidated Redirects and Forwards

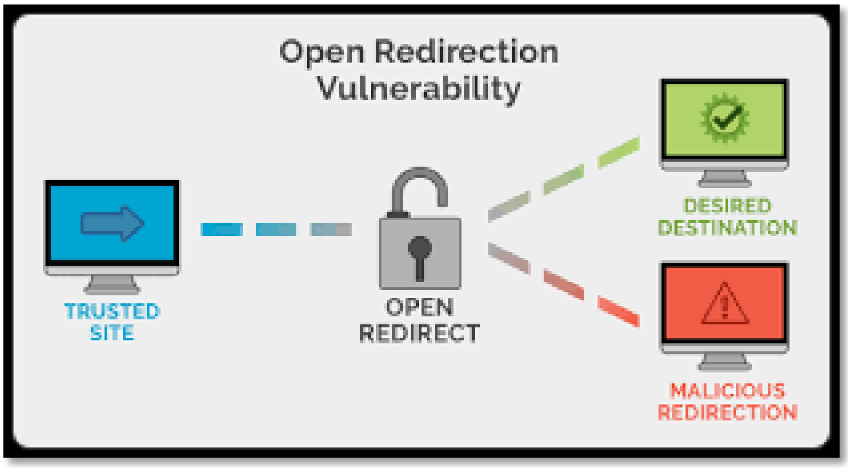


Figure 12: Unvalidated Redirects and Forwards

Image Source: (Le et al., 2019)

Yet another case of faulty input filtering. Let us pretend the destination site uses a redirect.php component that accepts a GET argument with a URL (Le et al., 2019). By changing the value of the parameter, it is possible to establish a URL on targetsite.com that leads to malwareinstall.com. This could appear to the visitor as targetsite.com/blahblahblah, which is safe to click. The user risks being sent to a virus drop page or another harmful website if they follow this link. The browser may be sent to targetsite.com/deleteprofile?confirm=1 instead.

It is important to note that header injection, which may occur if unfiltered user information is inserted into an HTTP header, is a serious security risk.

Prevention

- Avoid using redirects, since they are seldom useful.
- Maintain a permanent directory of appropriate redirect addresses for use as required.
- Include the custom parameter in the whitelist. Keep in mind that this may be challenging.

2.3 Architecture of web intrusion detection system

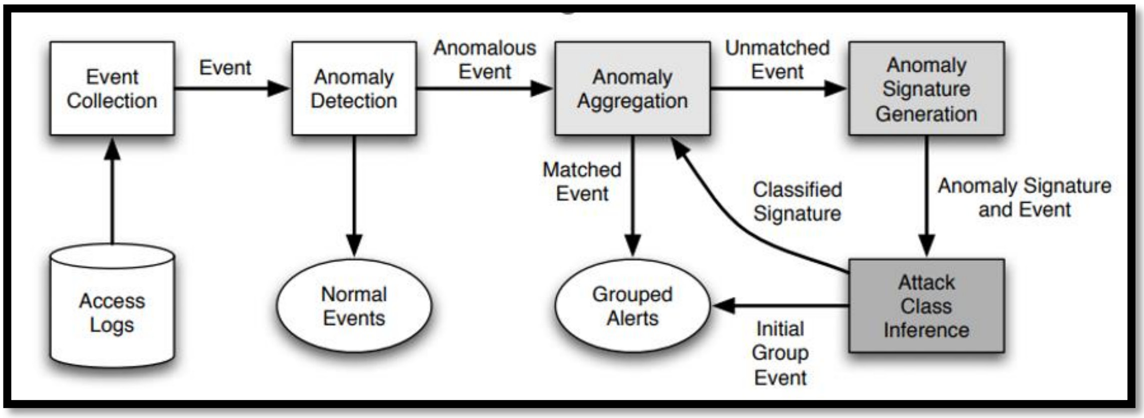


Figure 13: Architecture of web intrusion detection system

Image Source: (Loureiro, 2021)

Developing a unique technique to intrusion detection was necessary to identify unknown assaults against custom-developed software and to characterise and categorise these attacks in a meaningful manner (Loureiro, 2021). With this method, previously undiscovered threats may be uncovered by combining an event collector with an anomaly detection component. Moreover, the design has three additional parts: a component for aggregating anomalies, a component for generating anomaly signatures, and a component for inferring attack classes. By include them in the design, the study may create a system that takes advantage of anomaly detection while minimising their drawbacks. Figure depicts the architecture.

Initial event creation and standardisation is performed by the event collector. After the events have been normalised, they are sent on to the anomaly detector, which checks to see whether they are indeed abnormal (Meneghello et al., 2019). No notification is sent out if the incident is considered typical. On the other side, the anomaly aggregate selected for measurement the event if it is deemed to be abnormal. This section compares the occurrence to a databank of anomalous signatures. The goal is to establish whether an occurrence is comparable to a known outlier. An alert is instantly created and categorised alongside other comparable alerts if an occurrence matches an anomalous signature. In the absence of a match, the event is sent to the system responsible for creating anomaly signatures. In this section, the study will generalise the anomaly and create a suitable "anomaly signature." Finally, the attack class inference part uses heuristics to try to figure out what kind of assault it is. After the anomalous event has been categorised, a new "anomaly signature" is added to the previous set, and an alert is created.

2.4 Anomaly Detection Systems and Generalization

Web servers in a network are always at risk of being attacked. Therefore, protecting susceptible software with security measures is crucial to safeguarding a system (Phokela et al., 2022). Situations may be improved with the use of anomaly detection systems, which can learn a baseline of typical behaviour independently from a corpus of data. To then find new vulnerabilities, the model is use.

The Importance of Generalization in Anomaly Detection: Most anomaly detection algorithms need to generalise to accurately represent additional events outside of the initial data set (R Shaikh, 2019). A need exists for a repository of attacks versus web servers and associated applications over HTTP. As a result of generalisation, it is possible to automatically track the position, which speeds up the process of determining the heuristics necessary for an anomaly detection system to reach the accuracy necessary for usage in production. The absence of critical security flaws in the systems build and release is a need, as security is a must. There is a pressing need for more preventative steps to stave off the inevitable subsequent assault.

Approaches to Securing Systems: One approach to securing the systems involves developing unique defences for each issue that has been spotted, such as attack signatures or code fixes (Sagrillo et al., 2023). However, both approaches need human analysis of every problem and the creation of a solution that caps the reaction time at a certain interval. Since attacks by self-replicating programmes may quickly proliferate, it is crucial to have automated procedures in place that can detect and counteract security risks as they emerge. Potentially, anomaly detection systems may identify new assaults without requiring human participation.

Mechanisms of Anomaly Detection Systems: Anomaly detection software often mimics the normative patterns seen in the training data. When a system is behaving normally, it means that malicious actors are not exploiting it to undertake actions that contradict what the system's administrators have authorised. Anomalies are defined as data points that contradict the model.

Machine Learning and Anomaly Detection: One way that anomaly detection systems use machine learning is to create a baseline model of typical activity (Sarkar, 2021). If a learning system wants to do more than just remember the training data, it must be able to generalise the data, which means it must be able to construct a set that reflects an example. Since the set of examples considered the normal set is bigger than the set of instances in the empirical data, a generalising anomaly detection system takes input that is close to, but not exactly equivalent to, examples from the set of examples regarded the regular set.

Legal input is unlimited in most anomalous detection equipment, and typical behaviours are unknown and may evolve over time (Sinha and Tripathy, 2019). Anomaly detection systems in such a scenario should only utilise a subset of available training data. A model that faithfully recounts typical behaviour is the holy grail of anomaly detection systems.

2.4.1 Overgeneralize

Attacks that are similar sufficient to the empirical data may be assessed as regular or a false statement in certain instances, such as when the standard set is too vast or when the algorithm generalises in abundant findings, reducing the system's effectiveness (Weamie, 2022).

2.4.2 Under generalize:

Once the normal set is uncertain or limitless, a computer that memorises the empirical data would need sufficient capacity for the whole set of normal, which is not feasible (Dong et al., 2019). This would lead to an over-specialization and false-positive identification of common occurrences. Normal cases that are modest deviations of empirical data are often overlooked by a system that does a poor job of generalising.

Proper generalisation is a necessary condition for reliable outlier identification. A system for detecting anomalies must be able to detect them with sufficient precision before they can be implemented (Kumar and Goyal, 2019). Anomaly detection accuracy may be improved by regulating generalisation. The model of the anomaly detection system employs the data representation in a way that permits differentiation between typical and unusual data occurrences.

Chapter 3: Research Methodology

In the context of designing a security strategy for a selected web application, the research methodology involves a systematic and scientific approach to gather, analyse, and interpret data for resolving questions related to this specific topic. Researchers can employ various research methodologies based on the nature of the research, available resources, and study objectives. Quantitative research can provide numerical insights into the effectiveness of security measures, while qualitative research can delve into user perceptions and experiences (Alahmadi, Axon and Martinovic, 2022). Mixed-method research can combine both approaches, offering a comprehensive understanding. Experimental research may be employed to test specific security protocols, and case study research can provide in-depth insights into real-world security implementations. Choosing the most suitable research technique is crucial for conducting dependable, significant, and legitimate research on the design of a security strategy for a web application.

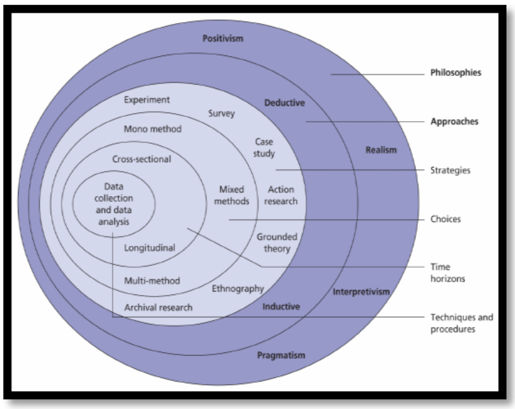


Figure 14:Research Onion

Image Source: (Akbar, 2016)

3.1 Research Philosophy

The research philosophy that guides the investigation is an essential component of the research technique (Fellows and Liu, 2021). These philosophical perspectives make it possible to select which strategy should be taken by the researcher and why doing so should be appropriate, which is derived from the research questions. The key assumptions are included in the research philosophy, which provides the researcher's perspective on how the world works. The research plan, as well as the methodologies used in that strategy, will be determined by these assumptions. The topic of research philosophy is concerned with the origin, nature, and progression of knowledge. A research philosophy may be defined as beliefs about the appropriate methods for gathering, analyzing, and applying data about certain phenomena (B Kumatongo, 2021). Although knowledge production may strike as deep, they are involved in creating acknowledge as part of the dissertation. To respond to the research question, and they must gather secondary and primary data and participate in data analysis. The conclusion the reach will represent the invention of new information. At each level of the research process, there is a set of presumptions about the information sources and the kind of knowledge being sought.

- Positivism

- Pragmatism
- Realism
- Interpretivism

3.1.1 Positivism philosophy

Positivism is an approach of thought that adheres to the concept that the only reliable information is "factual" knowledge, which is the knowledge obtained from observation (using the senses), including measurement. When doing research following positivism, the researcher's responsibility is restricted to the collecting of data and the interpretation of that data objectively. The outcomes of research conducted using these sorts of investigations may often be seen and quantified (Rahman, 2016). The positivist worldview is predicated on collecting measurable data that can be analysed statistically. It has been pointed out that positivism, as a philosophical system, is consistent with the empiricist idea that knowledge originates from human experience. It is characterized by an atomistic and ontological view of the universe, according to which "the world is made up of separate, observable elements and events that interact in an observable, determined and regular manner." Additionally, in positivist investigations, the researcher is considered separate from the study, and the research itself does not account for human motivations in any way. Second, Crowther and Lancaster contend that positivist analysis often uses an inductive method. They say this in their argument. In contrast, inductive research methodologies are often connected with a phenomenological philosophical orientation.

3.1.2 Interpretivism philosophy

Interpretivism, also known as interpretive research methodology, involves the interpretation of study elements, emphasizing the incorporation of human perspectives into research (Siponen, Soliman and Holtkamp, 2021). In this approach, researchers believe that understanding reality, whether given or socially constructed, is only possible through social constructs like language, consciousness, shared meanings, and tools. The roots of interpretivism lie in critiquing positivism in the social sciences, leading to a philosophy that places a stronger emphasis on qualitative rather than quantitative analysis. Aligned with idealism, interpretivism encompasses various approaches such as social constructivism, phenomenology, and hermeneutics, all rejecting the objectivist notion that meaning exists independently of consciousness. This approach underscores the significance of researchers, acting as social actors, acknowledging individual differences. Interpretivism studies often focus on the nuanced meanings and employ diverse research methodologies to capture multiple facets of the subject matter.

3.1.3 Pragmatism philosophy

Pragmatism, a philosophical perspective positioned between positivism and interpretivism, emphasizes a compromise (V Braun, 2020). Prioritizing the practical application of research findings, pragmatists advocate for the use of diverse methodologies and strategies tailored to the specific research topic. Flexibility is key for researchers, who are encouraged to choose the most appropriate methods, considering both qualitative and quantitative data sources. The pragmatic school of thought recognizes that different research topics may necessitate varied philosophical perspectives. Therefore, the selection of a philosophical standpoint should align with the study's objectives and the characteristics of the phenomena under investigation.

Justification

Interpretivism serves as a justified philosophical approach for the research on the design of a security strategy for a selected web application. This philosophy is rooted in the belief that reality is socially constructed, emphasizing the importance of human perspectives and subjective meanings. In the context of designing security strategies, understanding the intricate web of human interactions, perceptions, and contextual meanings is paramount.

Security strategies for web applications involve not only technical aspects but also intricate user behaviours, perceptions, and responses to security measures. Interpretivism allows researchers to delve into the subjective experiences of users, deciphering their attitudes towards security features and comprehending how these perceptions influence the effectiveness of security strategies. By recognizing that meaning is constructed through language, consciousness, and shared understandings, interpretivism provides a lens to explore the social dimensions of security practices in the digital realm.

Moreover, the design of security strategies often requires a nuanced understanding of the socio-cultural context in which web applications operate. Interpretivism, with its focus on social constructs and shared meanings, enables researchers to uncover contextual factors that shape security practices. It encourages an in-depth exploration of user interactions, considering factors such as trust, user experience, and the cultural nuances that may impact the adoption of security measures. Interpretivism is justified in the study of designing security strategies for web applications due to its ability to capture the complex interplay of human factors, social contexts, and subjective meanings, providing valuable insights for the development of effective and user-centric security measures.

3.2 Research Approach

A research approach refers to the overall method researchers employ to conduct a study and gather data for addressing research questions or testing hypotheses (Asenahabi, 2019). It involves selecting methodologies, tactics, and processes to ensure a systematic and rigorous examination. The choice of the research method should be guided by the nature of the research topic and the study's objectives. Quantitative research places a primary focus on numerical data and statistical analysis, while qualitative research emphasizes detailed data and interpretation (Mohajan, 2020). Mixed-methods research combines both approaches. Researchers need to consider factors such as resource availability, skills, and ethical considerations when determining the most suitable method. By adhering to a well-established research method, researchers can generate reliable and valid findings, contributing to the advancement of knowledge in their field.

Additionally, it is argued that research methodology is best conceptualized as a comprehensive strategy and a set of methods for conducting the study. In this regard, three broad systems of thought can be applied to the methodology: Deductive approach, Inductive approach, and Abductive approach.

3.2.1 Qualitative

Qualitative research is a method aimed at acquiring non-numerical data to gain in-depth insights into a subject (Chai et al., 2021). Diverging from statistical approaches, qualitative research is characterized by its unstructured or semi-structured nature, focusing on the "why" rather than the measurement of phenomena. The emphasis lies in descriptive exploration, delving into opinions, perspectives, and qualities rather than quantitative metrics represented in graphs or charts. Qualitative data collection often involves reputed sources such as online journals, newspaper and articles, google scholar (Chauvette, Schick-Makaroff and Molzahn, 2019). This method is commonly applied in market research, conducted in natural settings where researchers analyse subjects without interventions, trials, or control groups. By immersing themselves in the context, researchers can capture the richness of human experiences and behaviours, providing a nuanced understanding that goes beyond mere numerical representations. In essence, qualitative research serves as a valuable tool for exploring the depth and complexity of various phenomena in their natural states.

3.2.3 Quantitative

Quantitative research employs techniques focused on the systematic collection of numerical data to measure various factors (Rahman, 2016b). This method ensures an organized and structured approach to data collection, contributing to unbiased and indisputable deductions. A prominent methodology within quantitative research is grounded theory, emphasizing the systematic gathering and analysis of evidence to derive insights. This method is particularly beneficial when researchers aim to draw broad conclusions from their study and make predictions about outcomes. One significant advantage of quantitative research lies in its capacity to gather data from a relatively extensive sample size while maintaining cost-effectiveness. Surveys, a commonly used instrument in quantitative research, exemplify this efficiency. Surveys enable researchers to collect data from a large and diverse population, providing a statistically significant representation. The adaptability and cost-effectiveness of surveys make them a preferred tool, allowing researchers to reach a wide audience and analyse responses with efficiency.

3.2.4 Deductive approach

The deductive approach in research involves initiating with a broad hypothesis or principles and then narrowing down to gather supporting evidence, including concrete examples and statistical analyses (Casula, Rangarajan and Shields, 2020). This method is commonly employed in quantitative studies and scientific research. The systematic process begins with a literature search, followed by the formulation of specific hypotheses based on existing theories. Subsequently, a survey is planned and executed to collect necessary data, which is then subjected to statistical analysis. The findings drawn from this process either confirm or refute the initial theory.

The deductive method ensures an organized and rigorous research process in quantitative studies, providing evidence-based insights and facilitating hypothesis testing (Baharmand et al., 2022). This approach contributes to the expansion of knowledge on a given topic. However, it's essential to acknowledge the limitations inherent in the deductive method, such as the assumption that existing theories are comprehensive and accurate, as well as the potential oversight of emerging events or alternative explanations.

3.2.5 Inductive approach

Inductive reasoning serves as a foundational research approach, commencing its investigative journey with specific instances or empirical data and gradually progressing towards broader generalizations, intricate patterns, or the formulation of theoretical constructs. In the realm of qualitative research, various data collection methodologies, such as in-depth interviews, participant observation, and document analysis, are employed to gather rich and nuanced information (Hennink and Kaiser, 2022). Subsequently, the obtained findings undergo meticulous scrutiny to unveil discernible patterns that may hold significance for understanding complex phenomena. Researchers adeptly leverage these identified patterns to construct innovative hypotheses or refine existing ones, thereby contributing to an enhanced comprehension of the world around them. The inherent strength of inductive reasoning lies in its ability to delve into uncharted territories, facilitating the generation of testable hypotheses and offering profound insights into the intricacies of social processes. Frequently aligned with qualitative research methodologies, this approach proves particularly efficacious in addressing unexplored domains or supplementing incomplete explanations (Allemang et al., 2023). It is crucial to acknowledge the inherent limitations, including reliance on smaller sample sizes and subjective interpretations of data, which may temper the conclusiveness and generalizability of inductive deductions. These challenges do not diminish the method's value, as inductive reasoning remains an invaluable avenue for exploration, pushing the boundaries of knowledge and contributing to the continuous evolution of research methodologies.

Justification

Qualitative and deductive research approaches play pivotal roles in the design of a security strategy for selected web applications, offering distinct perspectives that contribute to a comprehensive understanding and effective implementation of security measures.

Qualitative research methodologies, such as in-depth interviews and participant observation, are instrumental in exploring the human and contextual aspects of web application security. Journals and articles in fields like cybersecurity and human-computer interaction can provide valuable insights into user behaviours, perceptions of security risks, and the social dynamics influencing security practices. Qualitative approaches allow researchers to uncover nuances and identify potential vulnerabilities that may not be apparent through quantitative methods alone. For instance, a study published in the "Journal of Cybersecurity" might delve into user perceptions of web application security, shedding light on user habits and behaviours that could impact the effectiveness of a security strategy. Additionally, articles in reputable cybersecurity magazines like "Cyber Défense Magazine" may offer practical insights and case studies, enhancing the qualitative understanding of real-world security challenges faced by web applications.

On the other hand, a deductive approach, rooted in theoretical frameworks and logical reasoning, can provide a structured foundation for designing a security strategy. Articles in academic journals focusing on cybersecurity theories and frameworks, such as the "Journal of Computer Security," may offer deductive insights into established principles and models for securing web applications. By grounding the security strategy in deductive reasoning, researchers can align their approach with proven theoretical concepts, enhancing the robustness of the proposed security measures. Furthermore, consulting reports from reputable cybersecurity organizations, as featured in publications like "Cybersecurity Ventures," can provide deductive insights into industry best practices and emerging trends. Integrating deductive reasoning with empirical evidence from qualitative studies strengthens the security strategy, ensuring a balanced and well-informed approach that addresses both theoretical principles and real-world complexities.

3.3 Data Collection Method

The term "data collection" pertains to the systematic gathering and analysis of information for research and validation purposes, employing diverse methods (H. R. and Aithal, 2022). The purpose of data collection is to investigate an issue, gaining insights into its outcomes, and discerning potential trends in the future. Utilizing various techniques aids in forming assumptions about the progression of a problem or situation that requires resolution. When seeking solutions, it is imperative to gather data from reliable sources to facilitate accurate computation and analysis.

Data collection techniques can be broadly categorized into two groups, depending on the nature of the data being collected:

- Primary Data Collection Methods
- Secondary Data Collection Methods

3.3.1 Primary data collection method

Primary sources constitute data derived first-hand from direct experiences, presenting an unmediated perspective untainted by comparisons with secondary sources (Smallwood, 2023). The essence of primary data lies in its direct relevance and immediate connection to the subject under investigation. This unfiltered nature ensures a high level of accuracy and authenticity, as it reflects the raw observations and experiences of the researcher.

Preceding data collection procedures are crucial in establishing the foundation for a study's purpose. Rigorous planning and methodical execution at this stage contribute to the reliability and relevance of the primary data collected. Researchers engage in direct interactions, surveys, or observations to capture the essence of the phenomenon they are studying, obtaining a first-hand account that holds intrinsic value. The two predominant classifications of primary data collection methods are quantitative and qualitative. Quantitative methods involve numerical data and statistical analyses, offering a structured approach to understanding patterns and relationships (Goldsack et al., 2020). Qualitative methods, on the other hand, delve into the richness of narrative data, capturing the nuances and intricacies of human experiences. Both classifications provide unique perspectives, enriching the research process and contributing to a comprehensive understanding of the study's objectives. In essence, primary data collection serves as the bedrock of research, ensuring a solid empirical foundation for subsequent analyses and interpretations.

3.3.2 Secondary data collection method

"Secondary data" is a term encompassing information gathered by an entity distinct from the organization conducting the current study or research. The acquisition of secondary data presents distinct advantages, being notably more straightforward and cost-effective compared to the collection of primary data (Nargesian, Asudeh and Jagadish, 2021). While primary data collection is recognized for providing authentic and unique insights, there are instances where organizations can derive significant benefits from incorporating secondary data and a qualitative approach.

The efficiency and cost-effectiveness of secondary data collection make it an attractive option for researchers and organizations seeking to supplement their analyses. Utilizing re-updated sources such as Google Scholar, online journals, articles, and newspapers can provide a wealth of existing information without the need for elaborate data collection processes. This approach becomes particularly advantageous when seeking a broad understanding of a topic or when time and budget constraints necessitate a pragmatic strategy.

Embracing a qualitative approach alongside secondary data enhances the depth of analysis, allowing researchers to delve into the nuances of existing information (Bryda and Costa, 2023). This combination is particularly powerful in uncovering insights, patterns, or trends that might be overlooked with a purely quantitative focus. In essence, while primary data collection is invaluable for its authenticity, secondary data, especially from reputable sources, serves as a valuable and resource-efficient complement, fostering a more comprehensive and nuanced understanding of the subject matter.

Justification

The utilization of secondary data collection methodology holds substantial justification in the context of designing a security strategy for a selected web application, particularly considering the dynamic and rapidly evolving nature of cybersecurity. Drawing from diverse sources such as journals, articles, newspapers, and magazines can enrich the research process and contribute valuable insights to the formulation of a robust security strategy.

Firstly, the field of cybersecurity is characterized by its ever-changing landscape, with new threats and vulnerabilities emerging regularly. Secondary data, obtained from up-to-date journals and articles in reputable cybersecurity publications, enables researchers to stay current with the latest trends, attack vectors, and security solutions. This ensures that the designed security strategy is informed by the most recent and relevant information, enhancing its effectiveness in addressing contemporary challenges.

Moreover, secondary data sources provide a wealth of contextual information and case studies from real-world security incidents. Analysing reports on past security breaches and successful security measures implemented in similar contexts can offer practical insights into effective strategies and potential pitfalls. Journals and articles focused on cybersecurity, such as those found in the "Journal of Cybersecurity" or reports from organizations like the Cybersecurity and Infrastructure Security Agency (CISA), can serve as valuable repositories of such information.

Additionally, reputable newspapers and magazines often feature analyses and expert opinions on cybersecurity issues. Incorporating insights from trusted news sources enhances the comprehensiveness of the secondary data, providing a broader understanding of the socio-political and economic factors influencing the security landscape. This contextual richness is crucial for tailoring a security strategy that not only addresses technical aspects but also aligns with the broader operational environment.

Furthermore, the secondary data collection methodology allows for a comprehensive review of existing frameworks, best practices, and standards in web application security. Academic journals and articles focusing on cybersecurity frameworks and standards contribute to the establishment of a solid theoretical foundation for the security strategy.

3.4 Waterfall Model

The Waterfall Model, a seminal software development methodology, is characterized by its linear and sequential approach, comprising distinct phases that unfold progressively (K Baha, 2023). While traditionally associated with software development, its principles can be judiciously adapted to the meticulous design of a security strategy for a selected web application. This model's structured nature ensures a systematic and thorough approach, crucial for addressing the intricate and evolving landscape of cybersecurity.

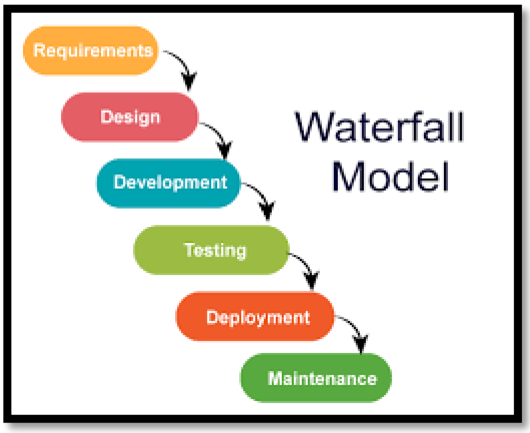


Figure 15: Waterfall Model

Image Source: (Dhruv Khanna, 2024)

Requirements Gathering: The inception of the Waterfall Model occurs with a meticulous exploration of the security requirements specific to the targeted web application (Thirunahari, 2023). This phase is foundational, involving the identification of potential threats, adherence to compliance standards, determination of user authentication needs, and consideration of other vital security aspects. A comprehensive understanding of these requirements lays the groundwork for the subsequent phases.

System Design: Building upon the gathered requirements, the next phase involves the development of a detailed system design for the security strategy (Yaacoub et al., 2020). This comprehensive design encompasses the architecture, security controls, encryption methods, and access management mechanisms. It is imperative to consider both technical and human factors during this stage, ensuring a holistic approach to security that addresses not only technological aspects but also the human elements integral to web application usage.

Implementation: With the system design in place, the focus shifts to the implementation phase (Richards, 2020). Here, the designed security strategy is put into action. This involves the deployment of identified security measures, including but not limited to firewalls, intrusion detection systems, secure coding practices, and encryption protocols. Special attention is given to ensuring the seamless integration of these security features into the web application, emphasizing a cohesive and effective implementation.

Testing: The Waterfall Model places a significant emphasis on rigorous testing to ascertain the effectiveness of the implemented security measures (T Sipponen, 2022). This phase involves thorough vulnerability assessments, penetration testing, and simulated attacks. The objective is to identify and rectify potential weaknesses, ensuring that the security strategy is robust enough to withstand real-world threats. The testing phase is a critical juncture, validating the efficacy of the security measures and refining them based on the findings.

Deployment: Upon successful testing and validation, the security strategy transitions to the deployment phase. Here, the security measures are made live and operational, actively safeguarding the web application against potential security threats. This phase marks the culmination of the design and implementation efforts, ensuring that the security strategy is ready to fulfil its protective role in a real-world environment.

Maintenance: The Waterfall Model acknowledges the need for ongoing attention and adaptation through its maintenance phase. In the realm of web application security, continuous monitoring, updates, and improvements are paramount. This phase is essential for adapting to evolving threats, addressing emerging vulnerabilities, and ensuring the long-term efficacy of the security measures. Regular maintenance activities contribute to the resilience and relevance of the security strategy over time.

3.5 Ethical Consideration

In the design of a security strategy for a selected web application, ethical considerations are paramount to ensure the integrity, confidentiality, and responsible use of information. When gathering data from re-updated sources such as Google Scholar, online journals, articles, and newspapers, it is imperative to uphold ethical standards in research practices.

The ethical responsibility involves respecting intellectual property rights. When consulting scholarly articles and journals from platforms like Google Scholar, researchers must adhere to copyright regulations and give proper attribution to authors and take papers from 2020 to

2023. Acknowledging the intellectual contributions of others promotes academic integrity and ensures that the information is used in a responsible and ethical manner.

Privacy considerations are crucial, especially when dealing with data from online sources. Researchers must be vigilant in maintaining the privacy of individuals whose information may be cited in articles or journals. Ensuring the anonymity of data subjects and refraining from disclosing sensitive details is essential to uphold ethical standards. It is also important to be mindful of potential biases or misrepresentations in the information obtained and to report findings accurately.

Additionally, the responsible use of information entails avoiding plagiarism and citing sources appropriately. Proper referencing of articles, journals, and other re-updated sources not only acknowledges the original authors but also provides transparency about the foundation of the research. This practice is essential for maintaining academic integrity and ensuring that the security strategy's design is based on credible and reliable information.

Moreover, researchers should be conscious of potential conflicts of interest that may arise when consulting certain sources. Transparency in disclosing any affiliations or relationships that could influence the research process or findings is crucial. This ensures that the security strategy is developed with objectivity and impartiality, free from undue influence. in the age of digital information, researchers must be diligent in addressing cybersecurity and data protection concerns. Safeguarding the confidentiality and integrity of the data collected from online sources is essential. Employing secure data storage practices and ensuring compliance with data protection regulations contribute to the ethical foundation of the research process.

3.5 Risk Assessment

Risk assessment for the design of a security strategy for a selected web application involves a meticulous examination of potential threats, vulnerabilities, and their potential impacts (Kaur et al., 2020). Identifying and analysing various risk factors is crucial in formulating a robust security approach. This includes assessing the sensitivity of data handled by the web application, evaluating the potential consequences of a security breach, and understanding the likelihood of different types of attacks.

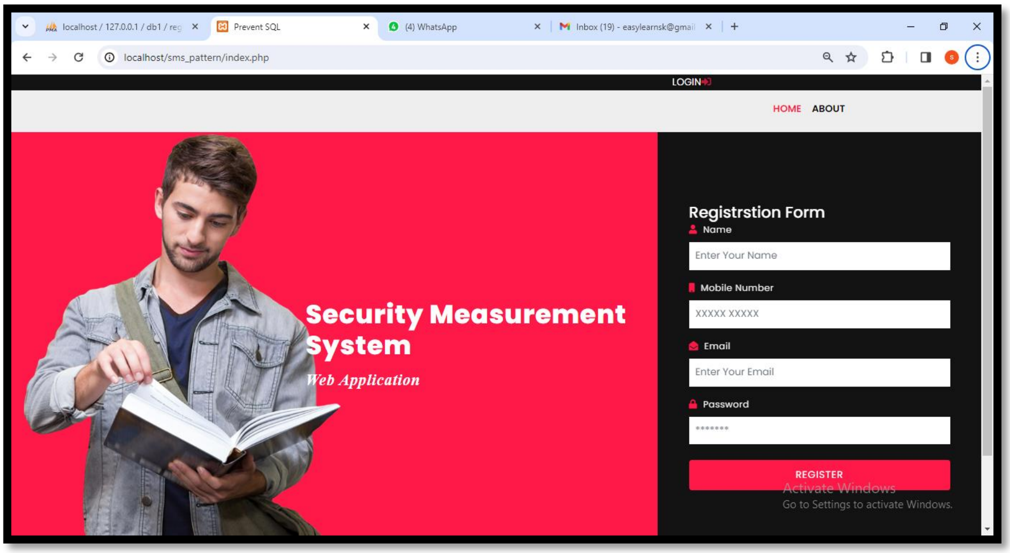
Furthermore, risk assessment involves scrutinizing the web application's architecture, codebase, and third-party integrations for vulnerabilities. Considering the evolving nature of cyber threats, continuous monitoring and periodic reassessment of risks are essential (Brass and Sowell, 2020). Threat modelling techniques can be employed to anticipate potential attack scenarios and vulnerabilities. Additionally, regulatory compliance and legal considerations are integral to the risk assessment, ensuring that the security strategy aligns with industry standards and legal requirements. A comprehensive risk assessment lays the foundation for the strategic implementation of preventive measures, detection mechanisms, and response strategies to safeguard the selected web application against a myriad of potential security challenges.

Chapter 4: Analysis and discussion

The analysis and discussion chapter of this dissertation comprised of an extensive evaluation of the security design of the chosen web application. Initial strategies stressed upon determining and comprehending present weaknesses of the system. With a detailed analysis, vulnerabilities like possible entry points for attacks and weak authentication systems were highlighted. Further, new solutions were suggested and incorporated into the security design to efficiently manage these. The solutions comprised of measures including improvised user authentication process, enhanced data encryption strategies and the execution of intrusion detection systems.

Dummy attacks were simulated to authenticate the efficiency of the newly executed solutions. This enabled and evaluation of the systems resilience against different kinds of cyber threats, involving SQL injection attacks and cross site script weaknesses. The results of these tests exhibited notable improvement in the overall security design of the application. The system demonstrated improved protection against possible threats by managing these identified vulnerabilities and execution of proactive security measures. Consistent testing and improvement guaranteed that the security system remained strong and versatile in the case of dynamics

4.1 Register – form


localhost / 127.0.0.1 / db1 / reg
Prevent SQL
(4) WhatsApp
Inbox (19) - easylearnsk@gmail
localhost/sms_pattern/index.php
LOGIN
HOME
ABOUT
Security Measurement System
Web Application
Activate Windows
Go to Settings to activate Windows.

Registrstion Form

Name: Enter Your Name

Mobile Number: XXXXX XXXXX

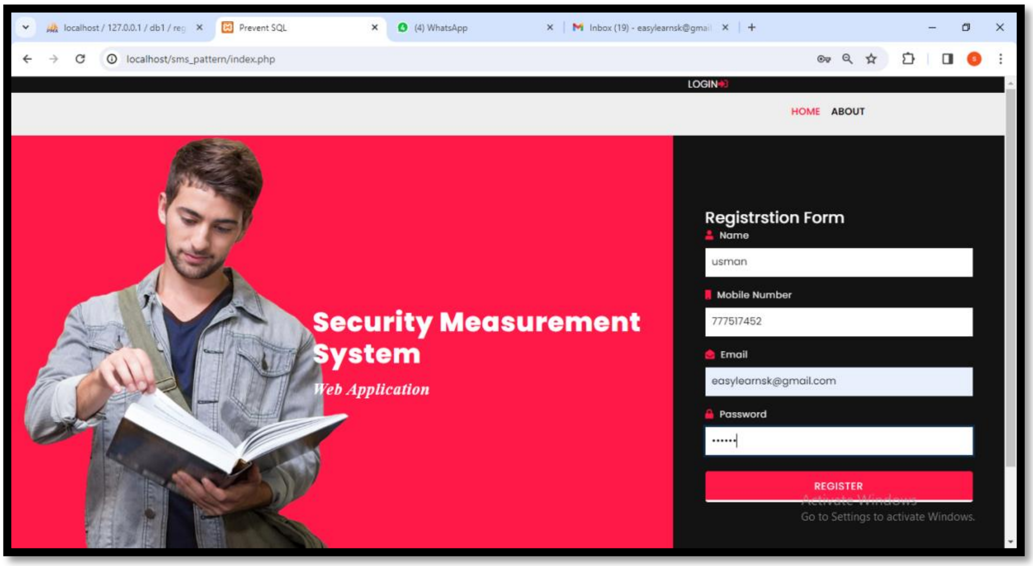
Email: Enter Your Email

Password: *******

REGISTER

Figure 16: Registration form

Image Source: Self-Created



localhost / 127.0.0.1 / db1 / reg
Prevent SQL
(4) WhatsApp
Inbox (19) - easylearnsk@gmail
localhost/sms_pattern/index.php
LOGIN
HOME
ABOUT
Security Measurement System
Web Application
REGISTER
Activate Windows
Go to Settings to activate Windows.

Registrstion Form

Name: usman

Mobile Number: 777517452

Email: easylearnsk@gmail.com

Password: ••••••

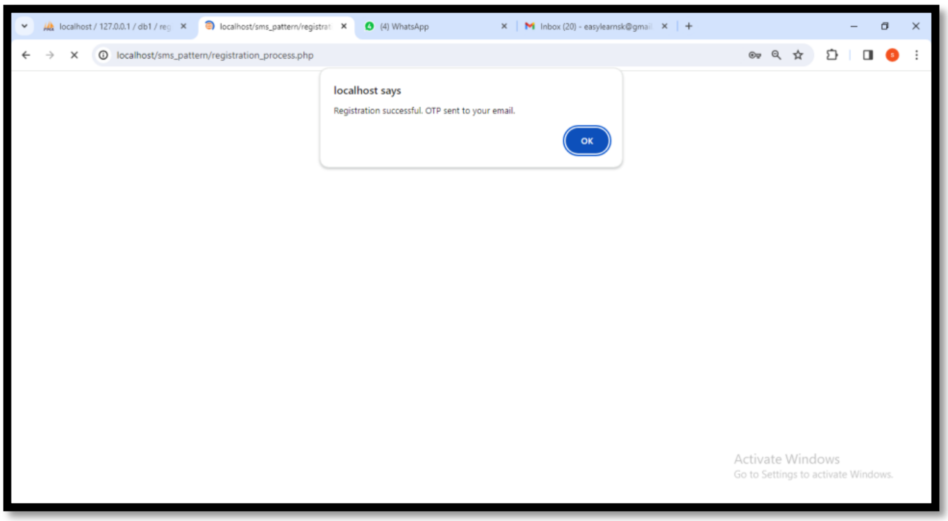
Figure 17: Registration form

Image Source: Self-Created

The above image focuses on developing a robust security framework for a specific web application. The application appears to include essential features such as user registration and a security measurement system. The user registration form collects crucial information like name, username, mobile number, email, and password. This suggests that the application deals with user accounts and sensitive data, emphasizing the need for a comprehensive security strategy (Balapour, Nikkhah and Sabherwal, 2020). Protecting user information is paramount, and the design of the security strategy should encompass measures to ensure the confidentiality, integrity, and availability of this data. The inclusion of a Security Measurement System indicates a proactive approach to security. This system likely involves tools and mechanisms to assess the application's security posture continuously. It could include vulnerability assessments, penetration testing, and monitoring mechanisms. The thesis may explore how these measures contribute to the overall security of the web application and how they align with industry best practices.

The design aspect of the security strategy implies a proactive and intentional approach to security implementation rather than a reactive one. This involves making informed decisions about encryption, access controls, authentication mechanisms, and other security controls that are crucial for safeguarding the application (Patwary et al., 2021). The analysis may delve into various security considerations, such as protection against common web application attacks (e.g., SQL injection, cross-site scripting), secure data transmission, and adherence to security standards. Also, it could discuss the importance of keeping the application updated with the latest security patches and staying vigilant against emerging threats.

Step1: This screen appears after pressing register button


localhost / 127.0.0.1 / db1 / reg
localhost/sms_pattern/registrat
(4) WhatsApp
Inbox (20) - easylearnsk@gmail
localhost/sms_pattern/registration_process.php

localhost says

Registration successful. OTP sent to your email.

OK
Activate Windows
Go to Settings to activate Windows.

Figure 18: registration successfully

Image Source: Self-Created

The above picture focuses on the post-registration process of a web application (Margeta et al., 2022). The screen that appears after pressing the register button signifies the successful completion of the registration process, and the mention of localhost suggests that the application is currently in a development or testing environment. The crucial aspect here is the subsequent step where an OTP (One-Time Password) is sent to the user's email address (Naem and Supervisor, 2020). This post-registration phase is a critical juncture in the user journey, as it involves the transmission and verification of sensitive information. The sending of an OTP to the registered email adds an extra layer of security, ensuring that the registered user is the legitimate account owner. The design of this security feature aligns with industry best practices for user authentication, enhancing the overall security of the web application.

The inclusion of OTP verification contributes to the multifactor authentication (MFA) approach, which is a cornerstone of a robust security strategy (GL Moepi, 2021). MFA significantly reduces the risk of unauthorized access by requiring users to provide multiple forms of identification, in this case, the combination of a password and a temporary OTP. This approach helps mitigate the impact of potential password compromises and enhances the security posture of the web application. The analysis may explore the technical aspects of how the OTP is generated, transmitted, and verified within the application. This could involve a discussion on cryptographic techniques, secure communication protocols, and measures to protect against common attacks targeting authentication processes, such as replay attacks. Furthermore, the analysis could delve into the user experience implications of this security strategy. It is crucial to strike a balance between security and usability to ensure that users can easily complete the registration process while still adhering to high-security standards.

Step 2: Now check the Email and find out OTP

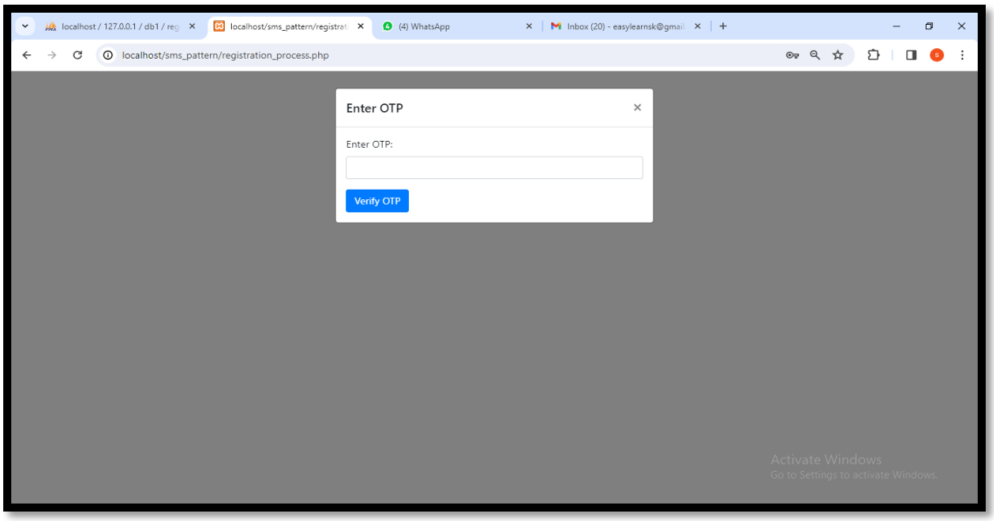
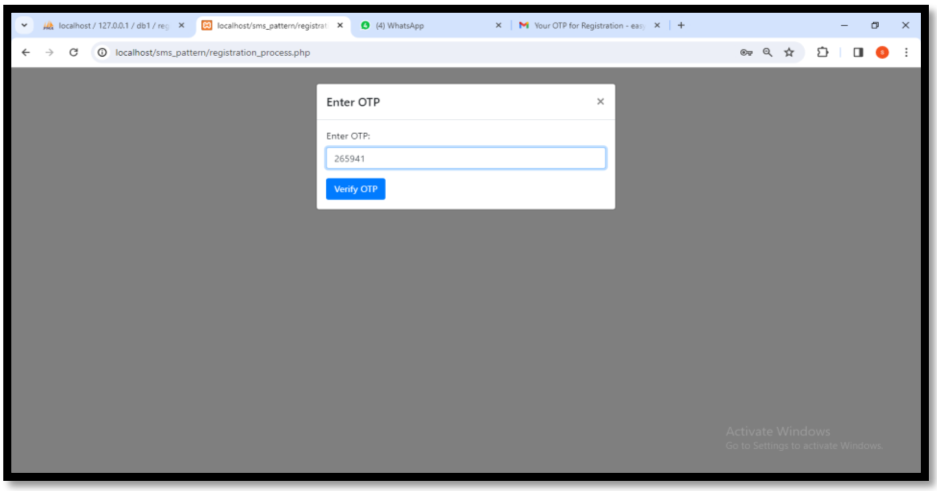

localhost / 127.0.0.1 / db1 / reg
localhost/sms_pattern/registrat
(4) WhatsApp
Inbox (20) - easylearnsk@gmail
localhost/sms_pattern/registration_process.php
Enter OTP
Enter OTP:
Verify OTP
Activate Windows
Go to Settings to activate Windows.

Figure 19: Email and find out OTP

Image Source: Self-Created


localhost / 127.0.0.1 / db1 / reg
localhost/sms_pattern/registrat
(4) WhatsApp
Your OTP for Registration - eas
localhost/sms_pattern/registration_process.php
Enter OTP
Verify OTP
Activate Windows
Go to Settings to activate Windows.

Enter OTP: 265941

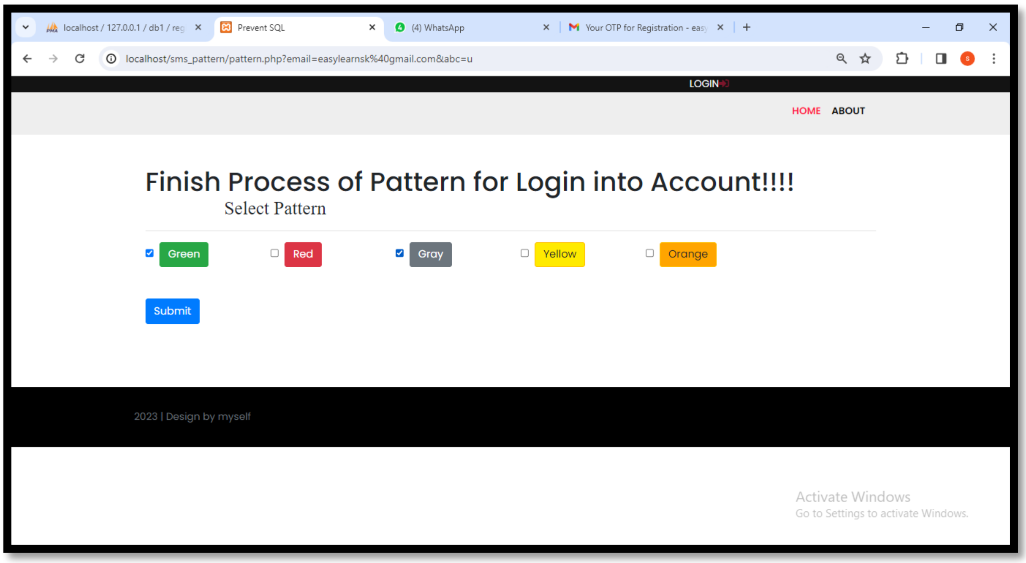
Figure 20:Verification

Image Source: Self-Created

The above image extends its analysis to the next steps in the user journey after registration – checking the email for the OTP (One-Time Password), entering the OTP, and subsequently verifying it. This phase of the application's security strategy is crucial in ensuring that only authorized users gain access to their accounts (Lyastani, Backes and Bugiel, 2023). The process begins with the user checking their email, emphasizing the reliance on email as a secure communication channel for transmitting sensitive information. The choice of email as a medium aligns with common industry practices, leveraging its widespread use and inherent security features, such as encryption during transit. The thesis could delve into the considerations behind choosing email as the communication channel and the security measures in place to protect the integrity and confidentiality of the OTP during transmission.

The step of entering the OTP represents a user-driven action, necessitating a seamless and user-friendly interface. The design of this interface should prioritize clarity and simplicity to ensure that users can easily locate and input the OTP (Furuberg and Øseth, 2023). The thesis may explore user experience considerations, such as the length of the OTP, the time window for its validity, and any usability challenges that users might encounter during this verification step. Verifying the OTP marks, the culmination of the authentication process, confirming the legitimacy of the user. This phase is integral to the overall security strategy, contributing to the establishment of a trusted user environment. The analysis might delve into the technical mechanisms involved in OTP verification, such as backend validation processes, database interactions, and measures to prevent brute-force attacks or other malicious activities targeting this step (AHY Mohammed, n.d.). Also, the analysis could consider the broader context of account recovery and security in case users encounter issues with OTP verification. It is essential to strike a balance between a stringent security posture and providing adequate support mechanisms for users who may face difficulties during this process.

Step 3: After successfully OTP verification this screen appear


localhost / 127.0.0.1 / db1 / reg
Prevent SQL
(4) WhatsApp
Your OTP for Registration - eas
localhost/sms_pattern/pattern.php?email=easylearnsk%40gmail.com&abc=u
LOGIN
HOME
ABOUT
Submit
2023 | Design by myself
Activate Windows
Go to Settings to activate Windows.

Finish Process of Pattern for Login into Account!!!!

Select Pattern

☑ Green □ Red ☑ Gray □ Yellow □ Orange

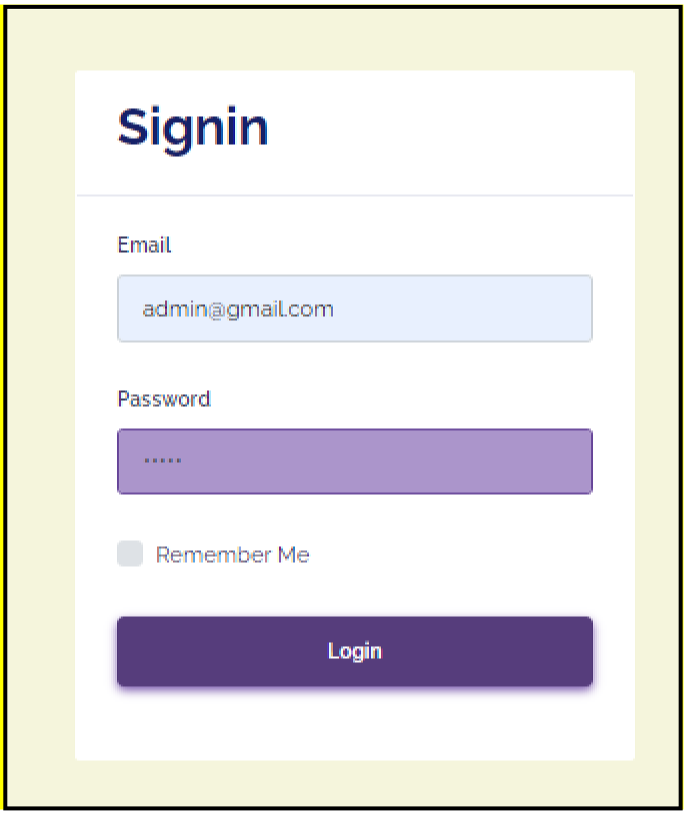
Figure 21:OTP Verification

Image Source: Self-Created

The above image depicts its analysis to the final phase of the user authentication process, where a screen appears after successful OTP verification (Adebayo et al., 2023). The message "Finish Process of Pattern for Login into Account!" suggests the implementation of a pattern-based authentication method for accessing the user account. This phase represents an additional layer of security beyond traditional password-based systems. The inclusion of a pattern-based authentication mechanism is a contemporary approach to enhancing the security posture of the web application. This method typically involves users selecting a specific pattern, often drawn on a grid of colours or shapes, to serve as a unique identifier during the login process (Gieseke et al., 2021). The thesis could explore the advantages of pattern-based authentication, such as its resistance to traditional password-based attacks and its potential for improving user experience.

The user is presented with a choice of colours, namely Green, Red, Gray, Yellow, and Orange, from which to select their pattern. This colour-centric pattern selection adds an element of personalization to the authentication process, allowing users to create a visually distinctive and memorable pattern. The "Submit" button indicates that the chosen pattern is a crucial element for subsequent logins. This pattern likely serves as an additional authentication factor, complementing the earlier stages of registration, email verification, and OTP entry. The design of this pattern-based authentication aligns with contemporary security trends that emphasize the importance of diverse authentication methods to mitigate the risk of unauthorized access (Adebimpe et al., 2023). Furthermore, the analysis might consider the user experience aspects, ensuring that the process of selecting and submitting the pattern remains intuitive and accessible. Clear guidance and feedback during this phase contribute to a positive user experience while reinforcing the importance of security (Khando et al., 2021). The final step involves submitting the chosen pattern, marking the completion of the setup process. The submission process must be secure, ensuring that the selected pattern is securely stored and associated with the user's account. The thesis could delve into the backend mechanisms involved in storing and validating these patterns, exploring encryption techniques and measures to prevent unauthorized access or tampering.

4.2 Login and Sign in



Signin

Email: admin@gmail.com

Password: .....

□ Remember Me

Figure 22: Admin Sign in

Image Source: Self-Created

The image portrays a UI intended for an admin login or sign-in page, regularly utilized in web applications or stages that request regulatory privileges (Wiefling et al., 2020). The interface highlights input fields for the client to include their email address and password, which are imperative certifications expected for validation. The email address, "admin@gmail.com," demonstrates this login form is for managers or significant-level clients.

When chosen, the "Remember Me" choice, probably a checkbox, permits the memorable framework of the client's login information for future visits, improving login endeavours (MA Puentes, 2021). The client clicks "Login" to start validation after giving their qualifications and choosing "Remember Me" whenever wanted. This simple anyway proficient plan empowers heads too safely and effectively accesses the backend or authoritative region of the stage or application.

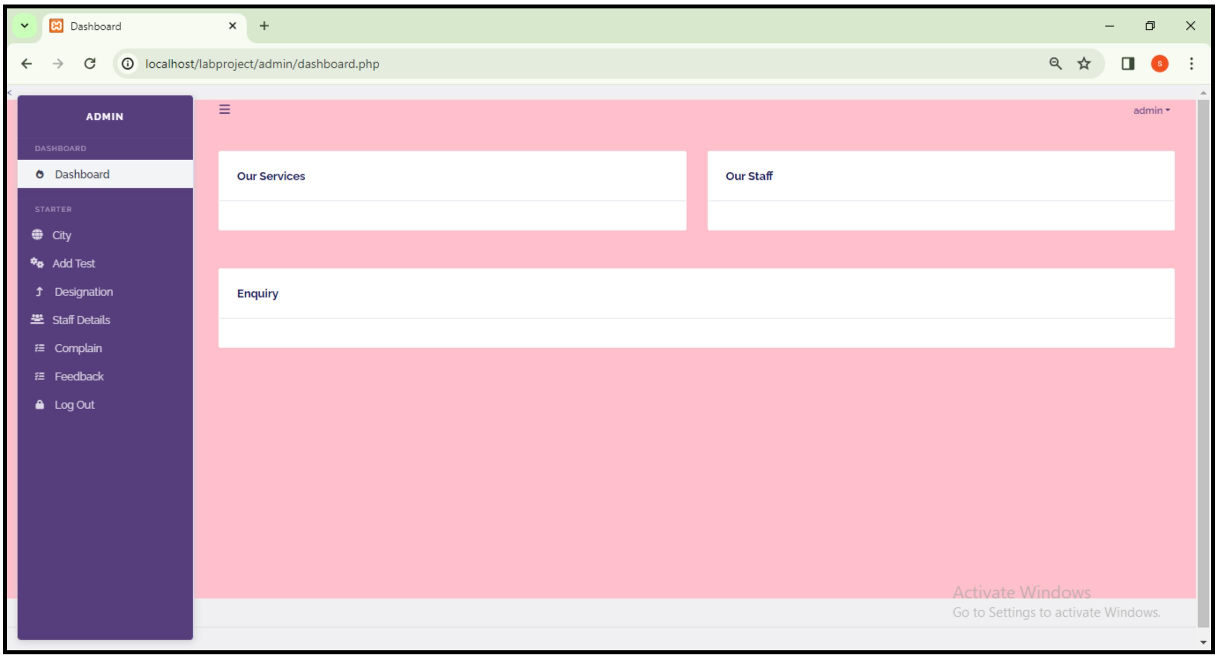


Figure 23: Dashboard

Image Source: Self-Created

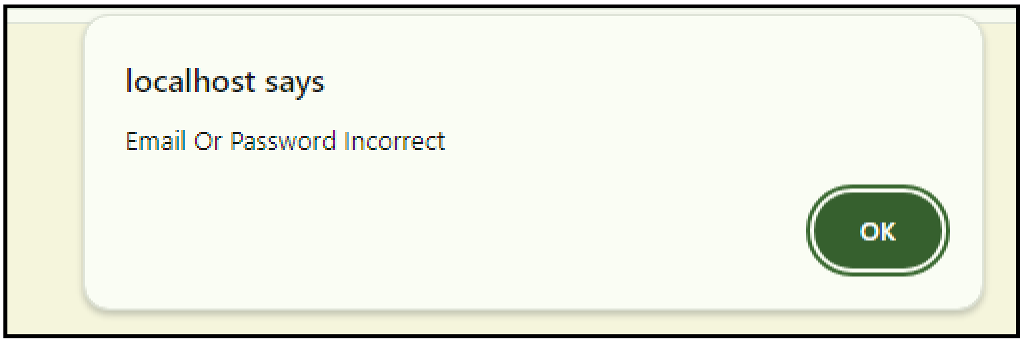
The dashboard interface shows a web application's organization part for overseeing different parts of the application or stage. The upper piece of the dashboard displays the application name "SPARTER" and gives route decisions, for example, "City," "Our Services," "Our Staff," "Enquiry," and "Activate Windows" (Manzoor et al., 2019). There is an idea to get to the settings to enact Windows, recommending that this dashboard might be related to a framework that has specific Windows initiation requirements.

Situated underneath the route choices, there is a segment marked "Add Test" which contains fields for "Designation" and "Staff Details." This region suggests usefulness for adding or managing tests and comparing staff data. Beneath, there are decisions for "E Complain" and

"Feedback," apparently giving the capacity to deal with grumblings and input got using the application.

At the lower part of the dashboard, there is a choice marked "Log Out" which permits the client to end the managerial meeting (JP Meier-Kolthoff, 2023). The design displays an efficient and proficient construction, highlighting particular names and segments assigned for different managerial exercises. In any case, the presence of the "Activate Windows" notice and the notice of localhost/labproject/admin/dashboard.php show that this connection point may be either a model or a testing climate, as these parts are not regularly found in a live creation climate.

The dashboard offers a total point of view of the program's regulatory functionalities and is by all accounts explicitly worked for easy-to-understand and effective administration of various pieces of the application (Alothman et al., 2023). By the by, considering the presence of elements, for example, the Windows enactment immediate and the localhost URL, it is pivotal to ensure that the dashboard is fittingly set up and safeguarded before its organization in a live climate.



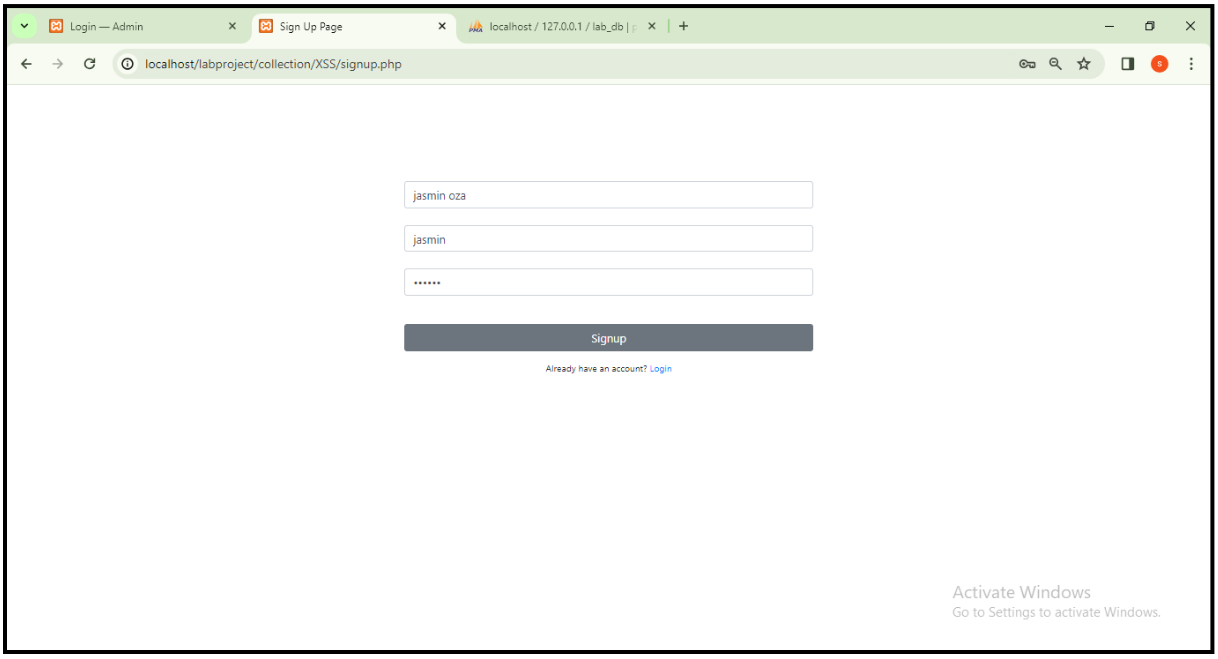
localhost says

Email Or Password Incorrect

Figure 24: Login Failed due to wrong Email or password

Image Sourced: Self-created

The image portrays a warning from a web application situated on a nearby server (localhost) expressing that the login endeavour was unsuccessful inferable from a wrong blend of email and password. The message is compact, obviously expressing to the client about the exact issue found while endeavouring to sign in. The client is incited to recognize the message by tapping the "OK" button, showing that no other activity is fundamental to the client. Web applications use this message to illuminate clients when validation comes up short and make sense of why they could not get to their records.



localhost/labproject/collection/XSS/signup.php

jasmin oza

jasmin

••••••

Already have an account? Login

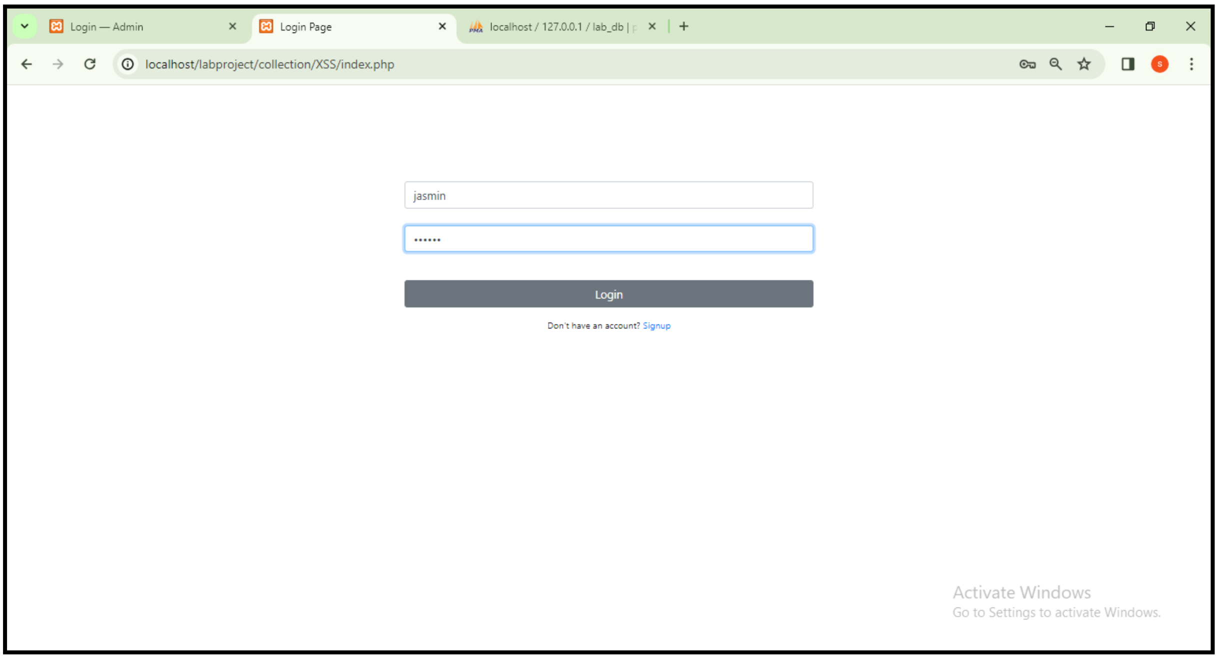
Activate Windows
Go to Settings to activate Windows.

Figure 25: Student Sign Up

Image Source: Self-created

The image shows a student login screen with fields for email or username and password. Furthermore, there is a hyperlink named "Sign Up Page" which demonstrates that there is a chance for new clients to enlist. The consideration of the "Already have an account? Login" connects gives more proof, recommending that ongoing clients can sign in by utilizing their qualifications.

The URL "localhost/labproject/collection/XSS/signup.php" shows that this login page is related to an assortment or module that arrangements with cross-site scripting (XSS), which is a security defect found in web applications (Rajkumar, Prakash and Vennila, 2022). The brief to go to settings for enactment and "Activate Windows" might be placeholders and not critical to the sign in page's working. The image portrays a traditional sign in screen that offers decisions for both signing in and enrolling, proposing the chance of a multi-step system for getting to the managerial functionalities of the program.



Login — Admin
Login Page
localhost / 127.0.0.1 / lab_db
localhost/labproject/collection/XSS/index.php

jasmin

••••••

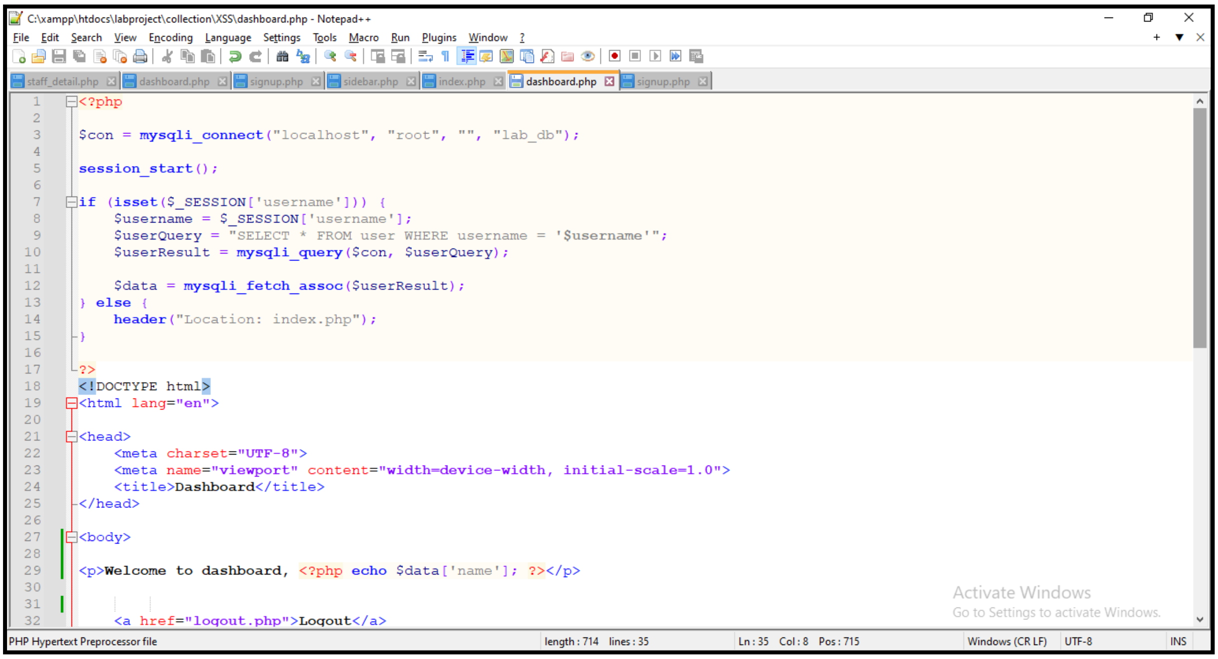
Don't have an account? Signup

Activate Windows
Go to Settings to activate Windows.

Figure 26: Student Log in

Image Source: Self-created

The image portrays a student login interface, featuring input areas for the entry of a username and password. The displayed URL, "localhost/labproject/collection/XSS/index.php," indicates that this login page is associated with a collection or module pertaining to cross-site scripting (XSS), which is a security vulnerability seen in web applications. The inclusion of the "Don't have an account? Signup" link signifies the availability of a registration option for students who lack an existing account (M Alajmi, 2020). The inclusion of the phrase "Activate Windows" and the instruction to access settings for activation are probably remnants or temporary elements that may not have any impact on the functionality of the login page. The image depicts a conventional login interface designed for students, offering them a user-friendly and direct means of accessing the application's functionalities.



C:\xampp\htdocs\labproject\collection\XSS\dashboard.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
staff_detail.php
dashboard.php
signup.php
sidebar.php
index.php
dashboard.php
signup.php

```
<?php

$con = mysqli_connect("localhost", "root", "", "lab_db");

session_start();

if (isset($_SESSION['username'])) {
    $username = $_SESSION['username'];
    $userQuery = "SELECT * FROM user WHERE username = '$username'";
    $userResult = mysqli_query($con, $userQuery);

    $data = mysqli_fetch_assoc($userResult);
} else {
    header("Location: index.php");
}

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
</head>

<body>

<p>Welcome to dashboard, <?php echo $data['name']; ?></p>


    <a href="logout.php">Logout</a>
```

Activate Windows
Go to Settings to activate Windows.
PHP Hypertext Preprocessor file
length : 714 lines : 35
Ln : 35 Col : 8 Pos : 715
Windows (CR LF)
UTF-8
INS

Figure 27: Cross Siting Code

Image Source: Self-created

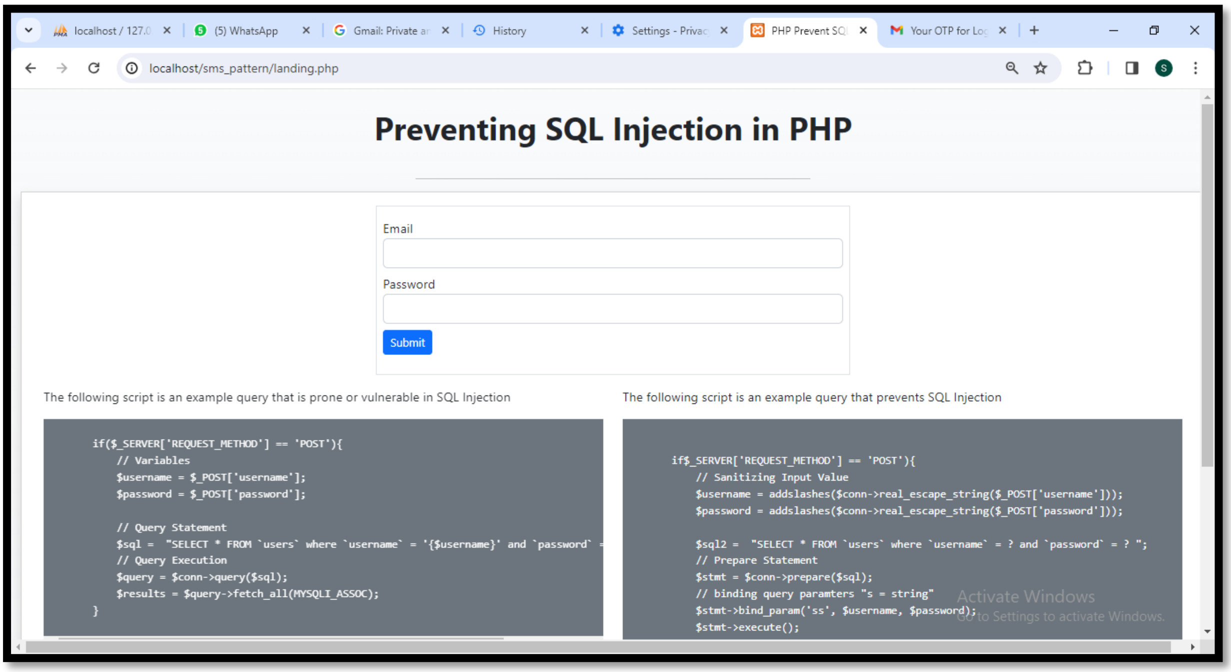
The image displays a portion of PHP code extracted from a file called "dashboard.php" within a web application project situated in the directory "C:\xampp\htdocs\labproject\collection\XSS\". The code begins with PHP opening tags" The code verifies the presence of a 'username' session variable to determine if the user is logged in. When the user is logged in, it retrieves user data from the 'user' table in the database using the username given in the session variable (Zidianakis et al., 2021). The user's name is thereafter exhibited on the dashboard by employing the PHP code. The code has a logout link that directs to a "logout.php" file, which is assumed to manage the logout feature. This code sample showcases a fundamental implementation of user authentication and session management for a dashboard page in a web application.

Cross-Site Scripting (XSS) attacks include injecting spyware and viruses into web sites that are then seen by other users, making it a type of injection attack. Web application vulnerabilities allow attackers to insert malicious scripts into innocent users' browsers. These attacks are widespread and happen when online programs do not properly check or decode user input that is incorporated in the output they produce (Rajkumar, Prakash and Vennila, 2022). Attackers can exploit XSS vulnerabilities to transmit malicious scripts to people who access compromised web pages. The browser, without awareness of the script's malicious intent, launches it under the assumption that it originated from a reliable source. This enables the script to retrieve sensitive data saved in the browser, such as passwords or session tokens, and possibly alter the content of the page.

Session hijacking

Session hijacking, often referred to as cookie hijacking, is a technique employed by criminal individuals to illegally obtain access to data or services within a computer system by abusing a legitimate computer session or session key.

Hijackers employ diverse strategies to achieve this, with certain methods being notably prevalent and efficient. Session sniffing is a technique used by attackers to monitor network traffic and obtain valid session tokens. Cross-site Scripting (XSS) is another way used by attackers to steal session cookies from unwary users by injecting malicious scripts into web sites (Chen et al., 2020). Session fixation is a method used by attackers to manipulate a user into using a session ID that the attacker has already acquired, which results in the compromise of the session. These strategies emphasize the significance of having strong security measures to safeguard against session hijacking threats.



Preventing SQL Injection in PHP

Email

Password

Submit

The following script is an example query that is prone or vulnerable in SQL Injection

```
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    // Variables
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Query Statement
    $sql =  "SELECT * FROM `users` where `username` = '{$username}' and `password` =
    // Query Execution
    $query = $conn->query($sql);
    $results = $query->fetch_all(MYSQLI_ASSOC);
}
```

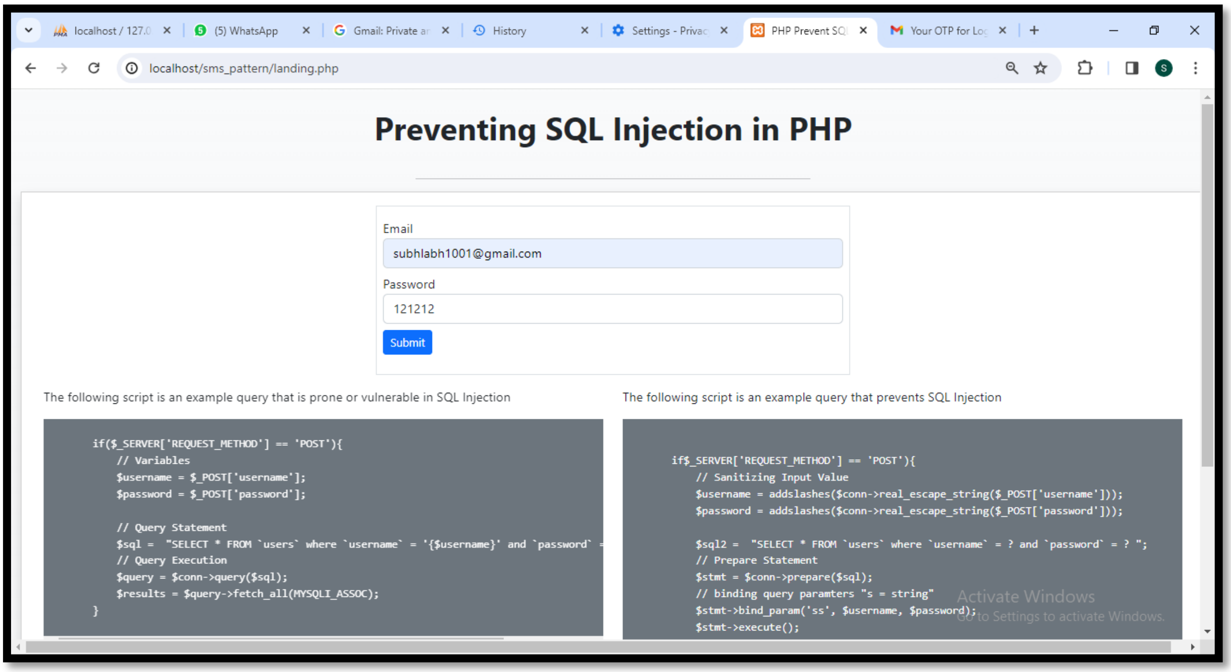
The following script is an example query that prevents SQL Injection

```
if$_SERVER['REQUEST_METHOD'] == 'POST'){
    // Sanitizing Input Value
    $username = addslashes($conn->real_escape_string($_POST['username']));
    $password = addslashes($conn->real_escape_string($_POST['password']));

    $sql2 =  "SELECT * FROM `users` where `username` = ? and `password` = ? ";
    // Prepare Statement
    $stmt = $conn->prepare($sql);
    // binding query paramters "s = string"
    $stmt->bind_param('ss', $username, $password);
    $stmt->execute();
```

Figure 28: Preventing SQL Injection n PHP

Image Source: Self-Created

New Login



Preventing SQL Injection in PHP

Email
subhlabh1001@gmail.com

Password
121212

The following script is an example query that is prone or vulnerable in SQL Injection

```
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    // Variables
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Query Statement
    $sql =  "SELECT * FROM `users` where `username` = '{$username}' and `password` =
    // Query Execution
    $query = $conn->query($sql);
    $results = $query->fetch_all(MYSQLI_ASSOC);
}
```

The following script is an example query that prevents SQL Injection

```
if$_SERVER['REQUEST_METHOD'] == 'POST'){
    // Sanitizing Input Value
    $username = addslashes($conn->real_escape_string($_POST['username']));
    $password = addslashes($conn->real_escape_string($_POST['password']));

    $sql2 =  "SELECT * FROM `users` where `username` = ? and `password` = ? ";
    // Prepare Statement
    $stmt = $conn->prepare($sql);
    // binding query paramters "s = string"
    $stmt->bind_param('ss', $username, $password);
    $stmt->execute();
```

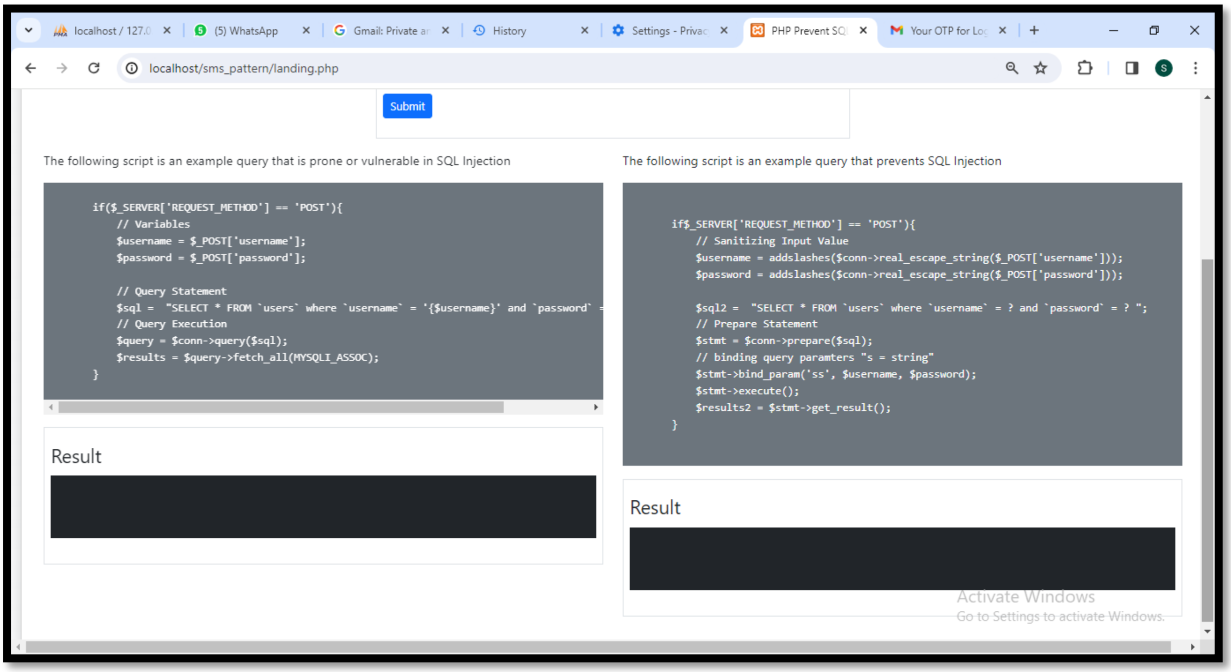
Figure 29: New Login

Image Source: Self-Created

The provided information seems to relate to a topic on preventing SQL injection in PHP, a crucial aspect of secure web development. The email address "1001@gmail.com" and the password "121212" appear to be example inputs for an authentication process within a PHP-based application.

SQL injection is a common cyber threat where attackers exploit vulnerabilities in input fields to inject malicious SQL code (Ahmad and Karim, 2021). In the context of preventing SQL injection in PHP, developers often implement secure coding practices and use parameterized queries or prepared statements to sanitize user inputs. These measures help thwart attempts by malicious users to manipulate SQL queries through input fields.

The example email and password likely serve as placeholders, illustrating the need for secure handling of user inputs to prevent SQL injection (Sabelström, 2023). Robust PHP coding practices involve validating, sanitizing, and escaping user inputs to ensure that they cannot be misused to compromise the integrity of database queries. The image prompts developers to consider and implement secure coding practices in PHP, particularly in the context of user authentication, to mitigate the risk of SQL injection attacks. Properly securing user inputs is crucial in maintaining the confidentiality and integrity of database operations within PHP-based web applications.



The following script is an example query that is prone or vulnerable in SQL Injection

```
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    // Variables
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Query Statement
    $sql =  "SELECT * FROM `users` where `username` = '{$username}' and `password` =
    // Query Execution
    $query = $conn->query($sql);
    $results = $query->fetch_all(MYSQLI_ASSOC);
}
```

Result

The following script is an example query that prevents SQL Injection

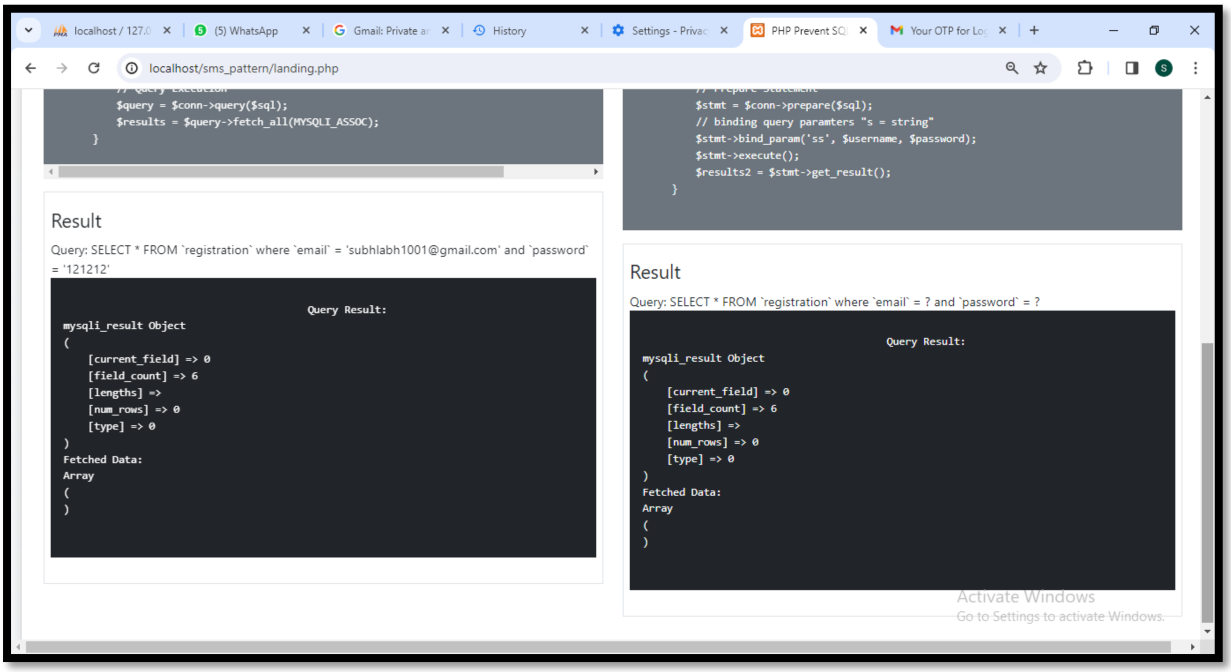
```
if$_SERVER['REQUEST_METHOD'] == 'POST'){
    // Sanitizing Input Value
    $username = addslashes($conn->real_escape_string($_POST['username']));
    $password = addslashes($conn->real_escape_string($_POST['password']));

    $sql2 =  "SELECT * FROM `users` where `username` = ? and `password` = ? ";
    // Prepare Statement
    $stmt = $conn->prepare($sql);
    // binding query paramters "s = string"
    $stmt->bind_param('ss', $username, $password);
    $stmt->execute();
    $results2 = $stmt->get_result();
}
```

Result

Figure 30:Example Query

Image Source: Self-Created

This is actual screen u can see that result column are empty but after login



```
    $query = $conn->query($sql);
    $results = $query->fetch_all(MYSQLI_ASSOC);
}
```

Result

Query: SELECT * FROM `registration` where `email` = 'subhlabh1001@gmail.com' and `password` = '121212'

```
                         Query Result:
mysqli_result Object
(
    [current_field] => 0
    [field_count] => 6
    [lengths] =>
    [num_rows] => 0
    [type] => 0
)
Fetched Data:
Array
(
)
```

```
    $stmt = $conn->prepare($sql);
    // binding query paramters "s = string"
    $stmt->bind_param('ss', $username, $password);
    $stmt->execute();
    $results2 = $stmt->get_result();
}
```

Result

Query: SELECT * FROM `registration` where `email` = ? and `password` = ?

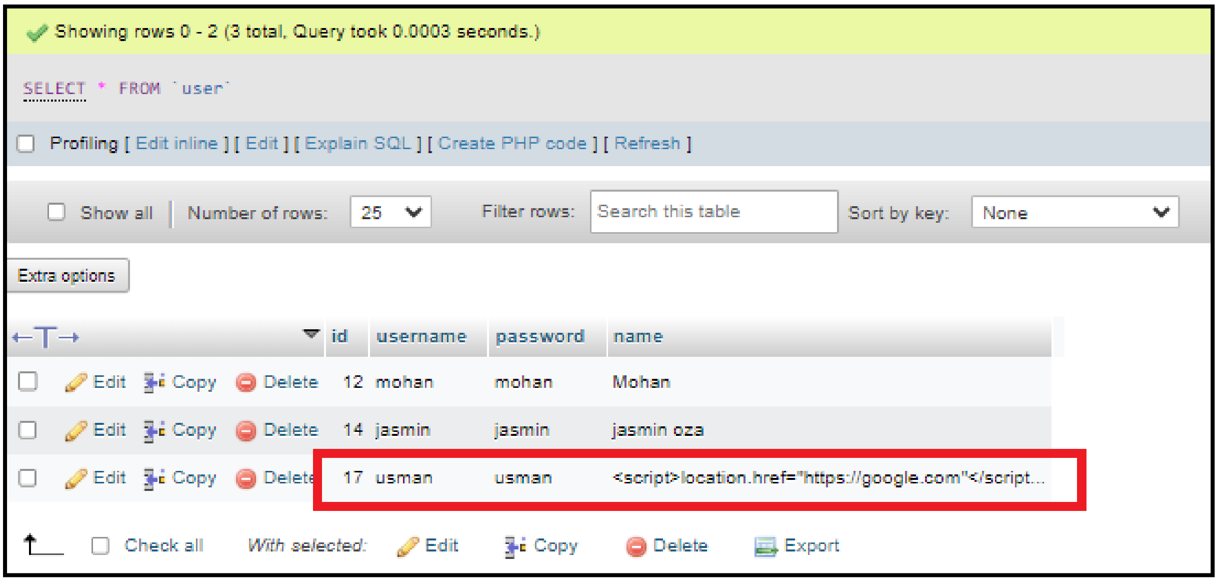
```
                         Query Result:
mysqli_result Object
(
    [current_field] => 0
    [field_count] => 6
    [lengths] =>
    [num_rows] => 0
    [type] => 0
)
Fetched Data:
Array
(
)
```

Figure 31: Result Column

Image Source: Self-Created

The provided information indicates that there is an image displaying the query result of a system (Gusenbauer and Haddaway, 2020). In the context of database systems, a query result typically refers to the outcome of executing a specific database query. This image could be showcasing the data retrieved from a database based on a particular query. The content of the query result image would depend on the nature of the executed query. It could include information such as records, columns, or statistical summaries retrieved from the database (Dogucu and Çetinkaya-Rundel, 2020). Understanding and interpreting the query result is essential for developers, analysts, or users who need insights into the data stored in the system. Analysing the query result is crucial for ensuring the accuracy and relevance of the data retrieved. It allows stakeholders to assess whether the executed query aligns with the intended purpose and requirements. Moreover, query results play a vital role in decision-making processes, providing valuable information for various applications, including reporting, analysis, or system debugging.

4.3 Database



✓ Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM `user`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Extra options
□ Check all
With selected:
Edit
Copy
Delete
Export

Number of rows: 25

Filter rows: Search this table

Sort by key: None

	id	username	password	name
□ Edit Copy Delete	12	mohan	mohan	Mohan
□ Edit Copy Delete	14	jasmin	jasmin	jasmin oza
□ Edit Copy Delete	17	usman	usman	<script>location.href="https://google.com"</script...

Figure 32: Database

Image Source: Self-Created

Prioritizing Data Protection in Web Applications: A straightforward client information base design with fields for id, username, password, and name shows that a web application security strategy should focus on client information (Mohammed Abdulridha Hussain et al., 2022). The content in the client 'Usman's password field shows SQL injection or Cross-Site Scripting (XSS) assaults, and this table, however simple, uncovered potential weaknesses. This situation features the fundamental prerequisite for solid safety efforts.

Implementing a Robust Security Framework: Security reviews, secure coding, and layered security are fundamental for a far-reaching security procedure. It is basic to utilize strong, generally acknowledged calculations to run delicate information, particularly passwords. Additionally, it is basic to have thorough information approval techniques set up injection attacks.

Mitigating XSS and Injection Attacks: Content Security Policy (CSP) headers can likewise forestall XSS assaults by distinguishing legitimate assets and content sources. This approach defends the program and its clients, yet additionally ensures adherence to information security norms, advancing certainty and trustworthiness among its client local area.

4.4 Practical

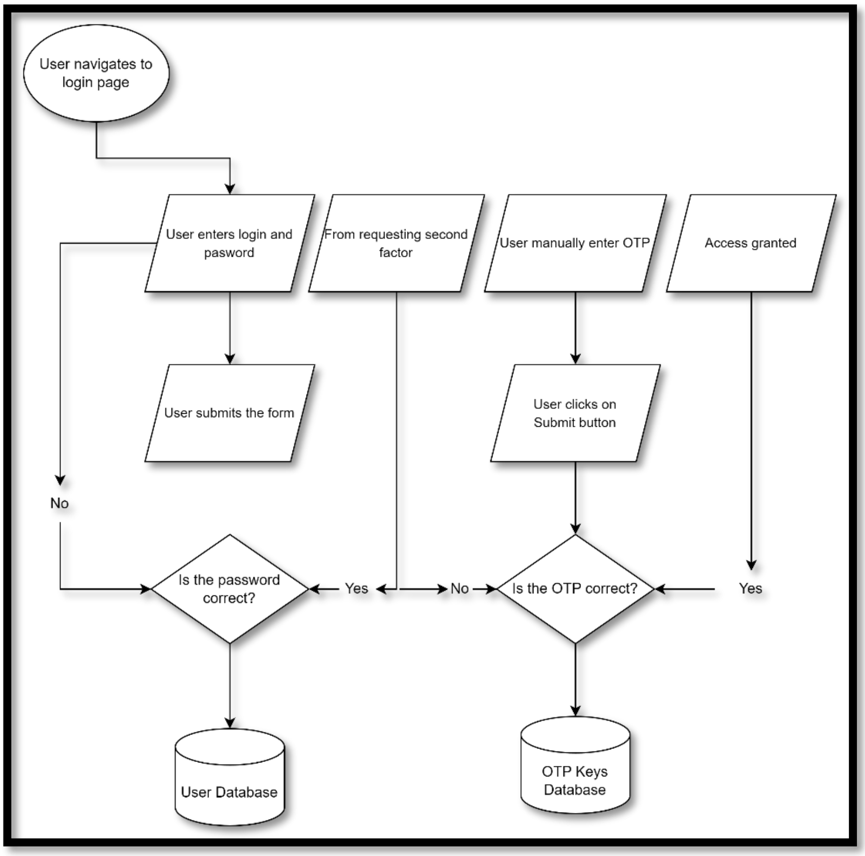


Figure 33: Flowchart

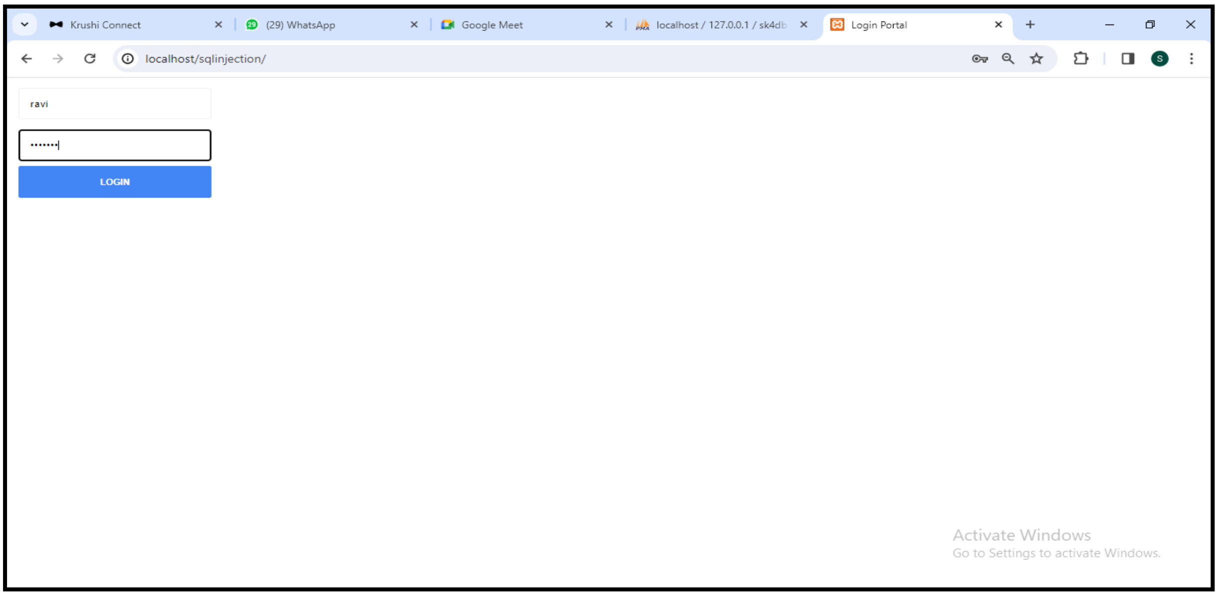
Image Source: Self-Created

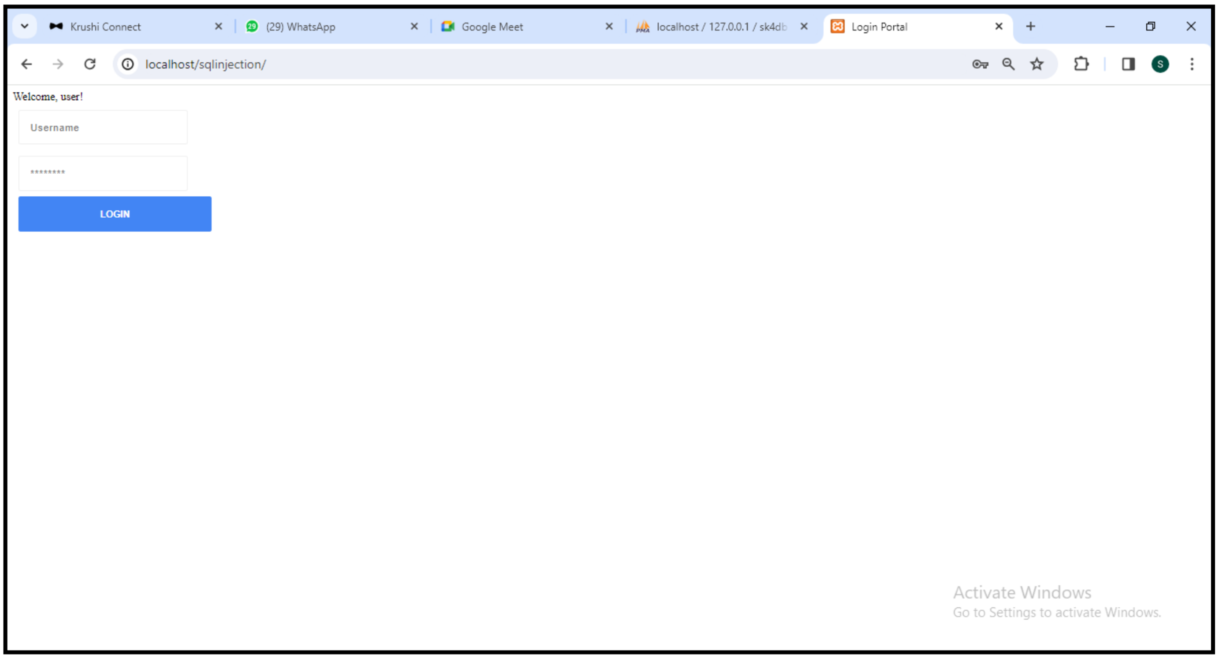
The provided flowchart outlines a step-by-step process within the context of the dissertation topic, "Design of Security Strategy for a Selected (Web) Application." The sequence delineates a user's journey through the login process, emphasizing the implementation of a robust security strategy.

Upon navigating to the login page, the user enters their login credentials, including a username and password. Subsequently, the user submits the form, initiating a critical juncture where the system queries the correctness of the entered password (Chaudhry et al., 2020). If the password is correct, the flow proceeds to the next step. However, if the password is incorrect, the user encounters a second factor authentication prompt. At this stage, the system requests a second factor for authentication, and the user is prompted to manually enter an OTP (One-Time Password). The use of OTP as a second layer of authentication enhances the security strategy by implementing a multi-factor authentication approach. After the user manually inputs the OTP, they click on the Submit button. The system then verifies the correctness of the entered OTP.

If the OTP is correct, the user is granted access, marking a successful authentication process. On the other hand, if the OTP is incorrect, access is denied, ensuring that only users with the correct second-factor authentication can proceed. The flowchart reflects the integration of an OTP mechanism, enhancing the security strategy of the selected web application.

The mention of "User database" and "OTP Keys database" indicates a structured approach to managing user information and OTP keys, emphasizing the importance of secure data storage. This security strategy aligns with the overarching dissertation topic, addressing the intricate design considerations for safeguarding a web application, particularly during the authentication process. The flowchart encapsulates a comprehensive security approach, incorporating both password-based and multi-factor authentication methods to fortify the selected web application against unauthorized access and potential threats.

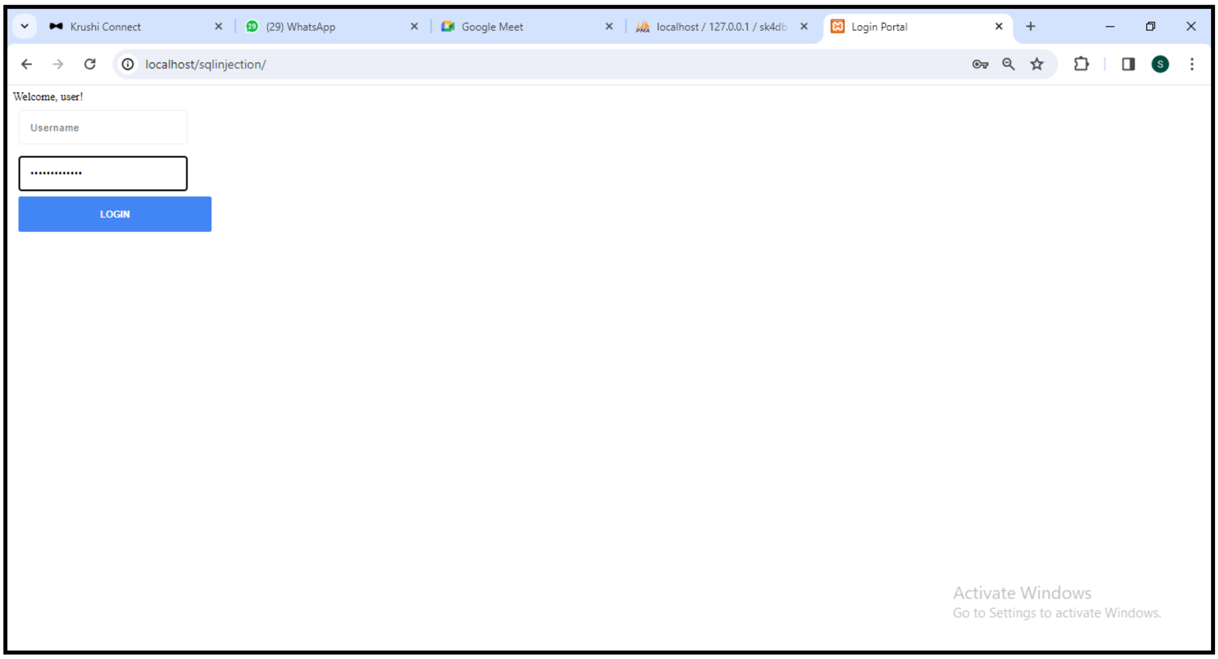




Now Sql Injection in password has been used.

test' or 1=1#

This code applies as a password and as a login…check this



Welcome, user!

Username

LOGIN

How to Prevent?

```
<?php
$hostname = "localhost";
$username = "root";
$password = "";
$dbname = "test";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
        die("Unable to connect");
}
if($_POST) {
        $uname = $_POST["username"];
        $pass = $_POST["password"];
        //Making sure that SQL Injection doesn't work
```

```
    $uname = mysqli_real_escape_string($conn, $uname);//test or 1=1

    $pass = mysqli_real_escape_string($conn, $pass);


//(Note This above two line prevnt against Sql Injection )

    $sql = "SELECT * FROM users_tutorials WHERE username = '$uname' AND password = '$pass'";

    $result = mysqli_query($conn, $sql);

    if(mysqli_num_rows($result) == 1) {

        echo "Welcome, user!";

    } else {

        echo "Incorrect Username/Password";

    }

}

?>

<!DOCTYPE html>

<html>

<head>

    <title>Login Portal</title>

    <style type="text/css">

        input[type=text],input[type=password] {

            padding: 16px;

            margin: 8px;

            border: 1px solid #f1f1f1;
				letter-spacing: 1px;

				border-radius: 3px;

				width: 240px;

			}

			input[type=submit] {

				margin-left: 8px;

				width: 274px;

				border-radius: 3px;

				border: 1px solid #4285f4;

				background-color: #4285f4;

				padding: 16px;

				color: white;

				font-weight: 600;

				cursor: pointer;

			}

		</style>

</head>

<body>

		<form action method="POST" autocompletes="off">

			<input type="text" name="username" placeholder="Username" /><br />

			<input type="password" name="password" placeholder="********" /><br />

			<input type="submit" name="login" value="LOGIN" />

		</form>

</body>
```

</html>

Chapter 5: Conclusion and Recommendation

Conclusion

The design of a security strategy for a selected web application is a multifaceted undertaking, addressing the complex interplay of technological, human, and resource-related factors. As the study conclude our exploration into this crucial aspect of modern digital landscapes, it is essential to reflect on the nuanced challenges, strategic considerations, and overarching principles that shape an effective security paradigm.

One of the primary takeaways from our examination is the inherent dynamism of the cybersecurity landscape. The continuously evolving nature of cyber threats necessitates a proactive and adaptive approach to security strategy design. A strategy that remains static in the face of emerging threats risks becoming obsolete, underscoring the importance of ongoing monitoring, threat intelligence, and regular updates. The dynamic nature of the digital realm demands a mind-set that anticipates and responds to new vulnerabilities, attack vectors, and technological advancements. The human element introduces a layer of unpredictability and complexity into the security equation. User behaviours, often influenced by factors beyond the control of security protocols, can be a source of vulnerability. Social engineering attacks, leveraging psychological manipulation, highlight the need for a comprehensive security strategy that encompasses not only technical safeguards but also user education and awareness programs. A well-rounded approach recognizes that even the most robust technical defences can be circumvented if users fall victim to phishing attempts or other manipulative tactics.

Resource constraints pose another significant challenge in the design and implementation of a security strategy. Organizations may face limitations in terms of financial resources, expertise, or technological infrastructure. It is crucial to acknowledge these constraints and prioritize security measures based on risk assessments and the criticality of assets. Strategic resource allocation becomes paramount, ensuring that available resources are optimally utilized to address the most significant threats and vulnerabilities. The reliance on third-party components introduces an additional layer of complexity. While leveraging external solutions can enhance functionality and efficiency, it simultaneously introduces potential vulnerabilities beyond direct control. Vigilant vetting of third-party components, regular security assessments, and a clear understanding of the security implications associated with external dependencies are imperative. Collaboration with vendors and maintaining a robust patch management process are crucial elements in mitigating risks associated with external dependencies.

The delicate balance between stringent security measures and user experience is a recurring theme in the design of security strategies. Overly complex security measures can impede user productivity, leading to potential resistance, workarounds, or non-compliance. Striking the right balance requires a nuanced understanding of user needs, effective communication, and the integration of security measures that are transparent and seamlessly integrated into the user experience. User education and awareness campaigns play a pivotal role in fostering a security-conscious culture and mitigating potential conflicts between security and usability. In conclusion, the design of a security strategy for a web application is a holistic and ongoing process that demands a comprehensive understanding of the digital landscape's complexities. The limitations the study have explored—evolving cyber threats, unpredictable user behaviours, resource constraints, reliance on third-party components, and the delicate balance between security and usability—underscore the need for a strategic, adaptive, and multidimensional approach to cybersecurity.

To navigate the evolving threat landscape, organizations must embrace a mind-set of continuous improvement and agility. Regular risk assessments, threat modeling, and penetration testing are essential components of this approach, enabling organizations to identify and remediate vulnerabilities before they can be exploited. Threat intelligence feeds and collaboration with industry peers further enhance the ability to anticipate and respond to emerging threats. User education and awareness programs form a critical pillar of any effective security strategy. By fostering a culture of cybersecurity awareness, organizations empower users to become active participants in their own defence against cyber threats. Training programs should cover not only technical aspects such as password hygiene and recognizing phishing attempts but also broader concepts of digital hygiene and responsible online behaviour.

Resource constraints should be addressed through strategic prioritization, aligning security investments with the organization's risk profile and critical assets. This requires a thorough understanding of the business context, the value of digital assets, and the potential impact of security incidents. Collaboration with executive leadership is essential to ensure that security considerations are integrated into overall business strategy and decision-making processes.

The reliance on third-party components necessitates a robust vendor management process. Organizations should conduct thorough security assessments before on boarding third-party solutions, establish clear contractual obligations regarding security practices, and regularly review the security posture of vendors. A proactive approach to patch management helps address vulnerabilities in a timely manner, reducing the window of exposure associated with external dependencies. Striking the right balance between security and usability requires collaboration between security teams, user experience designers, and other relevant stakeholders. Security measures should be designed with user needs in mind, minimizing disruptions to workflow while still providing effective protection. Clear communication about the rationale behind security measures helps build understanding and cooperation among users.

The design of a security strategy for a web application is an ongoing journey rather than a destination. Embracing a mind-set of continuous improvement, adaptability, and collaboration is essential in navigating the dynamic landscape of cybersecurity. By addressing the limitations and challenges discussed, organizations can cultivate a resilient security posture that not only safeguards digital assets but also contributes to the overall trust and resilience of the digital ecosystem.

Recommendation

The design of a security strategy for a selected web application is a critical undertaking in the contemporary digital landscape, where cyber threats are persistent and evolving. This recommendation provides a comprehensive guide for the development and implementation of an effective security strategy, considering the unique challenges and complexities associated with safeguarding web applications.

Understanding the Web Application's Landscape: Before embarking on the design of a security strategy, it is imperative to conduct a thorough analysis of the selected web application's landscape. Identify the application's functionalities, data flows, and potential vulnerabilities. This initial assessment sets the foundation for tailoring security measures to address specific risks.

Dynamic and Adaptive Approach: Recognize that the cybersecurity landscape is dynamic, with new threats emerging regularly. The security strategy should be designed with an adaptive mindset, allowing for continuous updates and improvements. Establish a framework for monitoring and responding to emerging threats promptly, ensuring that the security measures remain effective over time.

User Education and Awareness: Acknowledge the unpredictable nature of user behavior and the susceptibility to social engineering attacks. Incorporate a robust user education and awareness program into the security strategy. This involves regular training sessions, simulated phishing exercises, and the dissemination of best practices to empower users to recognize and mitigate security risks.

Resource Allocation and Constraints: Consider the practical constraints associated with resource availability, including budgetary limitations, technical expertise, and infrastructure capabilities. Prioritize security measures based on risk assessments and allocate resources strategically to maximize impact. Collaborate with stakeholders to secure the necessary support for the implementation of robust security measures.

Third-Party Risk Management: Acknowledge the reliance on third-party components within the web application's ecosystem. Implement a thorough third-party risk management program that assesses the security posture of external components. Establish contractual agreements that mandate adherence to security standards, ensuring that third-party integrations do not introduce vulnerabilities.

Balancing Security and User Experience: Strive for a delicate equilibrium between stringent security measures and a seamless user experience. Complex security protocols may impede usability, leading to user frustration and potential resistance. Conduct usability assessments to identify opportunities for enhancing security without compromising the user interface and experience.

Compliance and Regulatory Considerations: Align the security strategy with relevant compliance standards and regulatory requirements applicable to the web application. This may include industry-specific regulations, data protection laws, and international standards. Ensure that the security measures implemented not only protect against cyber threats but also adhere to legal and ethical considerations.

Incident Response and Recovery Planning: Develop a robust incident response and recovery plan as an integral part of the security strategy. Clearly define roles and responsibilities, establish communication protocols, and conduct regular drills to test the effectiveness of the response plan. This proactive approach ensures a swift and organized response in the event of a security incident.

Continuous Monitoring and Threat Intelligence: Implement continuous monitoring mechanisms to detect anomalies and potential security incidents in real-time. Leverage threat intelligence feeds to stay informed about emerging cyber threats relevant to the web application's industry and ecosystem. Proactive monitoring enhances the organization's ability to respond promptly to evolving threats.

Regular Security Audits and Assessments: Conduct regular security audits and assessments to evaluate the effectiveness of the implemented security measures. Engage external cybersecurity experts to perform penetration testing, vulnerability assessments, and code reviews. The insights gained from these assessments can guide ongoing improvements to the security strategy.

Reference list

Aborujilah, A., Adamu, J., Shariff, S.M. and Awang Long, Z. (2022). *Descriptive Analysis of Built-in Security Features in Web Development Frameworks*. [online] IEEE Xplore. doi:10.1109/IMCOM53663.2022.9721750.

Adebayo, O.S., Ganiyu, Shefiu Olusegun, Alli, A.A., Rufai, Salihu Ahmed, Jubril, Abdullahi Monday, Abdulazeez, L. and Gadzama, E.H. (2023). Two-layer Secured Graphical Authentication with One Time Password (OTP) Verification for a Web Based Applications. *Futminna.edu.ng*. [online] doi: http://repository.futminna.edu.ng:8080/jspui/handle/123456789/18488.

Adebimpe, L.A., Ng, I.O., Idris, M.Y.I., Okmi, M., Ku, C.S., Ang, T.F. and Por, L.Y. (2023). Systemic Literature Review of Recognition-Based Authentication Method Resistivity to Shoulder-Surfing Attacks. *Applied Sciences*, [online] 13(18), p.10040. Available at: https://www.mdpi.com/2076-3417/13/18/10040.

Ahmad, K. and Karim, M. (2021). A Method to Prevent SQL Injection Attack using an Improved Parameterized Stored Procedure. *International Journal of Advanced Computer Science and Applications*, [online] 12(6). Available at: https://pdfs.semanticscholar.org/48f1/0d1cb5dc8ff8386bdcf38eaddad320c50f8e.pdf.

AHY Mohammed, (n.d.). *An Empirical Analysis of Incorrect Account Remediation in the Case of Broken Authentication | IEEE Journals & Magazine | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/10360844/.

Akbar, A. (2016). *FACTORS INFLUENCING JOB SATISFACTION AMONG HEALTHCARE ASSISTANTS (HCA) WORKING IN DUBLIN*. [online] Available at: https://www.researchgate.net/figure/Research-Onion-Source-Sauder-etal-2011_fig1_348578861.

Alahmadi, B.A., Axon, L. and Martinovic, I. (2022). *99% False Positives: A Qualitative Study of {SOC} Analysts' Perspectives on Security Alarms*. [online] www.usenix.org. Available at: https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi.

Al-Doori, M.B.M. (2023). *Intrusion detection system in internet of things networks using machine learning techniques*. [online] openaccess.altinbas.edu.tr. Available at: http://openaccess.altinbas.edu.tr/xmlui/handle/20.500.12939/4261.

Alghawazi, M., Alghazzawi, D. and Alarifi, S. (2022). Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. *Journal of Cybersecurity and Privacy*, 2(4), pp.764–777. doi:10.3390/jcp2040039.

Aliero, M.S., Ghani, I., Qureshi, K.N. and Rohani, M.F. (2019). An algorithm for detecting SQL injection vulnerability using black-box testing. *Journal of Ambient Intelligence and Humanized Computing*, 11(1), pp.249–266. doi:10.1007/s12652-019-01235-z.

Allemang, B., Samuel, S., Greer, K., Schofield, K., Pintson, K., Patton, M., Farias, M., Sitter, K.C., Patten, S.B., Mackie, A.S. and Dimitropoulos, G. (2023). Transition readiness of youth with co-occurring chronic health and mental health conditions: A mixed methods study. *Health Expectations*, [online] 26(6), pp.2228–2244. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/hex.1382.

ALMEIDA, (2021). A study of confidential computing to prevent sensitive information exposure on information systems. *Ufcg.edu.br*. [online] doi: http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/24980.

Alothman, B., Alibrahim, O., Alenezi, N., Alhashemi, A., Alhashemi, M., Almardasi, D., Khattab, O., Joumaa, C. and Khan, M. (2023). The Development of a Secure Online System to Protect Social Networking Platforms from Security Attacks. *Applied Sciences*, [online] 13(21), p.11731. Available at: https://www.mdpi.com/2076-3417/13/21/11731.

Alshami, A., Elsayed, M., Ali, E., Eltoukhy, A.E.E. and Zayed, T. (2023). Harnessing the Power of ChatGPT for Automating Systematic Review Process: Methodology, Case Study, Limitations, and Future Directions. *Systems*, [online] 11(7), p.351. Available at: https://www.mdpi.com/2079-8954/11/7/351.

Asenahabi, B. (2019). Basics of research design: A guide to selecting appropriate research design. *International Journal of Contemporary Applied Research*, [online] 6(5), pp.76–89. Available at: http://ijcar.net/assets/pdf/Vol6-No5-May2019/07.-Basics-of-Research-Design-A-Guide-to-selecting-appropriate-research-design.pdf.

Ashari, I.F.A., Affandi, M., Putra, H.T. and Nur, M.T. (2023). Security Audit for Vulnerability Detection and Mitigation of UPT Integrated Laboratory (ILab) ITERA Website Based on OWASP Zed Attack Proxy (ZAP). *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, [online] 7(1), pp.24–34. doi:10.35870/jtik. v7i1.657.

B Kumatongo (2021). Research Paradigms and Designs with their Application in Education | Journal of Lexicography and Terminology (Online ISSN 2664-0899. Print ISSN 2517-9306). *medicine.unza.zm*. [online] Available at: https://medicine.unza.zm/index.php/jlt/article/view/551.

Babu, M.S., Raj, K.B. and Devi, D.A. (2020). Data Security and Sensitive Data Protection using Privacy by Design Technique. *EAI/Springer Innovations in Communication and Computing*, pp.177–189. doi:10.1007/978-3-030-47560-4_14.

Baharmand, H., Vega, D., Lauras, M. and Comes, T. (2022). A methodology for developing evidence-based optimization models in humanitarian logistics. *Annals of Operations Research*, [online] 319(1), pp.1197–1229. Available at: https://link.springer.com/article/10.1007/s10479-022-04762-9.

Balapour, A., Nikkhah, H.R. and Sabherwal, R. (2020). Mobile application security: Role of perceived privacy as the predictor of security perceptions. *International Journal of Information Management*, [online] 52, p.102063. Available at: https://www.sciencedirect.com/science/article/pii/S0268401219309041.

Bandari, V. (2023). Enterprise Data Security Measures: A Comparative Review of Effectiveness and Risks Across Different Industries and Organization Types. *International Journal of Business Intelligence and Big Data Analytics*, [online] 6(1), pp.1–11. Available at: https://research.tensorgate.org/index.php/IJBIBDA/article/view/3.

Brambilla, A., Mangili, S., Das, M., Lal, S. and Capolongo, S. (2022). Analysis of Functional Layout in Emergency Departments (ED). Shedding Light on the Free-Standing Emergency Department (FSED) Model. *Applied Sciences*, [online] 12(10), p.5099. Available at: https://www.mdpi.com/2076-3417/12/10/5099.

Brass, I. and Sowell, J.H. (2020). Adaptive governance for the Internet of Things: Coping with emerging security risks. *Regulation & Governance*. [online] Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/rego.12343.

Bryda, G. and Costa, A.P. (2023). Qualitative Research in Digital Era: Innovations, Methodologies and Collaborations. *Social Sciences*, [online] 12(10), p.570. Available at: https://www.mdpi.com/2076-0760/12/10/570.

Calzavara, S., Conti, M., focardi, R., Rabitti, A. and Tolomei, G. (2020). Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery. *IEEE Security & Privacy*. doi:10.1109/msec.2019.2961649.

Casula, M., Rangarajan, N. and Shields, P. (2020). The potential of working hypotheses for deductive exploratory research. *Quality & Quantity*, [online] 55(1), pp.1703–1725. Available at: https://link.springer.com/article/10.1007/s11135-020-01072-9.

Chai, H.H., Gao, S.S., Chen, K.J., Duangthip, D., Lo, E.C.M. and Chu, C.H. (2021). A Concise Review on Qualitative Research in Dentistry. *International Journal of Environmental Research and Public Health*, [online] 18(3). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7908600/.

Chaudhry, S.A., Shon, T., Al-Turjman, F. and Alsharif, M.H. (2020). Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems. *Computer Communications*, [online] 153, pp.527–537. Available at: https://www.sciencedirect.com/science/article/pii/S0140366420301687.

Chauvette, A., Schick-Makaroff, K. and Molzahn, A.E. (2019). Open Data in Qualitative Research. *International Journal of Qualitative Methods*, [online] 18, p.160940691882386. Available at: https://journals.sagepub.com/doi/abs/10.1177/1609406918823863.

Dhruv Khanna (2024). *Redirect Notice*. [online] Google.com. Available at: https://www.google.com/url?sa=i&url=https%3A%2F%2Fluciferrocks.medium.com%2Fintroduction-to-waterfall-model-of-sdlc-software-development-life-cycle-a01c7ec5e2ef&psig=AOvVaw3gMzmWyqA3yVp6L4iSC4TU&ust=1704269026911000&source=images&cd=vfe&opi=89978449&ved=2ahUKEwi0gvCAn76DAxWeqGMGHc3VC2wQr4kDegQIARB9.

Distler, V., Lenzini, G., Lallemand, C. and Koenig, V. (2020). The Framework of Security-Enhancing Friction: How UX Can Help Users Behave More Securely. *New Security Paradigms Workshop 2020*. [online] Available at: https://dl.acm.org/doi/abs/10.1145/3442167.3442173.

Dogucu, M. and Çetinkaya-Rundel, M. (2020). Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities. *Journal of Statistics Education,* [online] pp.1–11. Available at: https://www.tandfonline.com/doi/abs/10.1080/10691898.2020.1787116.

Dong, Y., Guo, W., Chen, Y., Xing, X., Zhang, Y. and Wang, G. (2019). *Towards the Detection of Inconsistencies in Public Security Vulnerability Reports Towards the Detection of Inconsistencies in Public Security Vulnerability Reports*. [online] Available at: https://www.usenix.org/system/files/sec19-dong.pdf.

El Sibai, R., Gemayel, N., Bou Abdo, J. and Demerjian, J. (2019). A survey on access control mechanisms for cloud computing. *Transactions on Emerging Telecommunications Technologies,* [online] 31(2). Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3720.

Fellows, R.F. and Liu, A.M.M. (2021). *Research Methods for Construction*. [online] *Google Books*. John Wiley & Sons. Available at: https://books.google.com/books?hl=en&lr=&id=b61JEAAAQBAJ&oi=fnd&pg=PR9&dq=The+research+philosophy+that+guides+the+investigation+is+an+essential+component+of+the+research+technique.+&ots=S40dGytXHv&sig=yQ_nCkFi-VO7-se9y_hbPXJnXFw.

Furuberg, I.L. and Øseth, M. (2023). *From Password to Passwordless: Exploring User Experience Obstacles to the Adoption of FIDO2 Authentication*. [online] ntnuopen.ntnu.no. Available at: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3093908.

Gieseke, L., Asente, P., Radomír Měch, Bedřich Beneš and Fuchs, M. (2021). A Survey of Control Mechanisms for Creative Pattern Generation. *Computer Graphics Forum,* [online] 40(2), pp.585–609. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.142658.

GL Moepi (2021). *Multi-Factor Authentication Method for Online Banking Services in South Africa | IEEE Conference Publication | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/9698724/.

Goldsack, J.C., Coravos, A., Bakker, J.P., Bent, B., Dowling, A.V., Fitzer-Attas, C., Godfrey, A., Godino, J.G., Gujar, N., Izmailova, E., Manta, C., Peterson, B., Vandendriessche, B., Wood, W.A., Wang, K.W. and Dunn, J. (2020). Verification, analytical validation, and clinical validation (V3): the foundation of determining fit-for-purpose for Biometric Monitoring Technologies (BioMeTs). *npj Digital Medicine,* [online] 3(1). Available at: https://www.nature.com/articles/s41746-020-0260-4.

Gusenbauer, M. and Haddaway, N.R. (2020). Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources. *Research Synthesis Methods*, [online] 11(2), pp.181–217. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7079055/.

H. R., G. and Aithal, P.S. (2022). *How to Choose an Appropriate Research Data Collection Method and Method Choice among Various Research Data Collection Methods and Method Choices during Ph.D. Program in India?* [online] papers.ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4275696.

Hennink, M. and Kaiser, B.N. (2022). Sample Sizes for Saturation in Qualitative Research: A Systematic Review of Empirical Tests. *Social Science & Medicine*, [online] 292(1), p.114523. Available at: https://www.mdpi.com/2076-0760/12/10/570.

Hughes, L., Dwivedi, Y.K., Misra, S.K., Rana, N.P., Raghavan, V. and Akella, V. (2019). Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda. *International Journal of Information Management*, [online] 49, pp.114–129. Available at: https://www.sciencedirect.com/science/article/pii/S0268401219302014.

Humayun, M., Niazi, M., Jhanjhi, N., Alshayeb, M. and Mahmood, S. (2020). Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study. *Arabian Journal for Science and Engineering*, [online] 45. doi:10.1007/s13369-019-04319-2.

JP Meier-Kolthoff (2023). [online] Oup.com. Available at: https://academic.oup.com/nar/article-abstract/50/D1/D801/6389592.

K Baha (2023). [online] Nii.ac.jp. Available at: https://tdu.repo.nii.ac.jp/record/2000040/files/%E8%AA%B2%E7%A8%8B%E5%8D%9A%E5%A3%AB_%E7%94%B2%E7%AC%AC175%E5%8F%B7_%E3%83%90%E3%83%BC%E3%83%8F%E3%80%80%E3%82%AB%E3%83%BC%E3%83%9E%E3%83%BC%E3%83%B3_%E2%91%A3%E5%AD%A6%E4%BD%8D%E8%AB%96%E6%96%87%E5%85%A8%E6%96%87.pdf.

Kaur, J., Khan, A.I., Abushark, Y.B., Alam, M.M., Khan, S.A., Agrawal, A., Kumar, R. and Khan, R.A. (2020). Security Risk Assessment of Healthcare Web Application Through Adaptive Neuro-Fuzzy Inference System: A Design Perspective. *Risk Management and Healthcare Policy*, [online] Volume 13, pp.355–371. Available at: https://www.tandfonline.com/doi/abs/10.2147/RMHP.S233706.

Khando, K., Gao, S., Islam, S.M. and Salman, A. (2021). Enhancing employee's information security awareness in private and public organisations: a systematic literature review. *Computers & Security*, [online] 106, p.102267. Available at: https://www.sciencedirect.com/science/article/pii/S0167404821000912.

Kumar, R. and Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities, and countermeasures: A survey. *Computer Science Review*, 33, pp.1–48. doi: 10.1016/j.cosrev.2019.05.002.

Kumi, S., Lim, C., Lee, S.-G., Oktian, Y.O. and Witanto, E.N. (2020). Automatic Detection of Security Misconfigurations in Web Applications. *Proceedings of International Conference on Smart Computing and Cyber Security*, pp.91–99. doi:10.1007/978-981-15-7990-5_8.

Latchoumi, T., Reddy, M. and Balamurugan, K. (2020). *European Journal of Molecular & Clinical Medicine Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention*. [online] Available at: https://www.ejmcm.com/article_2573_11704b94294a972d05905a301ddda208.pdf.

Le, D.-N., Kumar, R., Mishra, B.K., Chatterjee, J.M. and Khari, M. (2019). *Cyber Security in Parallel and Distributed Computing: Concepts, Techniques, Applications and Case Studies*.

[online] *Google Books*. John Wiley & Sons. Available at: https://books.google.com/books?hl=en&lr=&id=FzGtDwAAQBAJ&oi=fnd&pg=PR16&dq=Authentication+and+Authorization:+A+Cyber+Security+Primer&ots=rfOOOkzWi4&sig=qL0qcMV-Us3Lo0gxtx0_JINVlQQ.

Loureiro, S. (2021). Security misconfigurations and how to prevent them. *Network Security*, 2021(5), pp.13–16. doi:10.1016/s1353-4858(21)00053-2.

Lyastani, S.G., Backes, M. and Bugiel, S. (2023). A Systematic Study of the Consistency of Two-Factor Authentication User Journeys on Top-Ranked Websites. *Proceedings 2023 Network and Distributed System Security Symposium*. [online] Available at: https://svenbugiel.github.io/publication/ndss-23/ndss-23.pdf.

M Alajmi (2020). *A Password-Based Authentication System Based on the CAPTCHA AI Problem | IEEE Journals & Magazine | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/9173668/.

MA Puentes (2021). *Visualizing Web Application Execution Logs to Improve Software Security Defect Localization | IEEE Conference Publication | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/9825829.

Malhotra, P., Singh, Y., Anand, P., Bangotra, D.K., Singh, P.K. and Hong, W.-C. (2021). Internet of Things: Evolution, Concerns and Security Challenges. *Sensors*, [online] 21(5), p.1809. Available at: https://www.mdpi.com/1424-8220/21/5/1809.

Manzoor, A., Shah, M.A., Khattak, H.A., Din, I.U. and Khan, M.K. (2019). Multi-tier authentication schemes for fog computing: Architecture, security perspective, and challenges. *International Journal of Communication Systems*, [online] p.e4033. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4033.

Marashdih, A.W., Zaaba, Z.F., Suwais, K. and Mohd, N.A. (2019). Web Application Security: An Investigation on Static Analysis with other Algorithms to Detect Cross Site Scripting. *Procedia Computer Science*, 161, pp.1173–1181. doi: 10.1016/j.procs.2019.11.230.

Margeta, J., Hussain, R., López Diez, P., Morgenstern, A., Demarcy, T., Wang, Z., Gnansia, D., Martinez Manzanera, O., Vandersteen, C., Delingette, H., Buechner, A., Lenarz, T., Patou, F. and Guevara, N. (2022). A Web-Based Automated Image Processing Research Platform for Cochlear Implantation-Related Studies. *Journal of Clinical Medicine*, [online] 11(22), p.6640. Available at: https://www.mdpi.com/2077-0383/11/22/6640.

Meneghello, F., Calore, M., Zucchetto, D., Polese, M. and Zanella, A. (2019). IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet of Things Journal*, 6(5), pp.1–1. doi:10.1109/jiot.2019.2935189.

Mishra, L. and Kaushik, V. (2021). Application of blockchain in dealing with sustainability issues and challenges of financial sector. *Journal of Sustainable Finance & Investment*, [online] pp.1–16. Available at: https://www.tandfonline.com/doi/abs/10.1080/20430795.2021.1940805.

Mohajan, H.K. (2020). Quantitative Research: A Successful Investigation in Natural and Social Sciences. *Journal of Economic Development, Environment and People*, [online] 9(4), pp.50–79. Available at: https://www.ceeol.com/search/article-detail?id=939590.

Mohammed Abdulridha Hussain, Zaid Alaa Hussien, Zaid Ameen Abduljabbar, Ma, J., Sibahee, A., Sarah Abdulridha Hussain, Vincent Omollo Nyangaresi and Jiao, X. (2022). Provably throttling SQLI using an enciphering query and secure matching. *Egyptian Informatics Journal*, [online] 23(4), pp.145–162. Available at: https://www.sciencedirect.com/science/article/pii/S1110866522000640.

Naem, E.A.A. and Supervisor, F.M.A. (2020). *Enhance Graphical Password Authentication using One Time pad*. [online] repository.sustech.edu. Available at: https://repository.sustech.edu/handle/123456789/2492.

Nargesian, F., Asudeh, A. and Jagadish, H.V. (2021). Tailoring data source distributions for fairness-aware data integration. *Proceedings of the VLDB Endowment*, [online] 14(11), pp.2519–2532. Available at: http://vldb.org/pvldb/vol14/p2519-nargesian.pdf.

Nasereddin, M., ALKhamaiseh, A., Qasaimeh, M. and Al-Qassas, R. (2021). A systematic review of detection and prevention techniques of SQL injection attacks. *Information Security Journal: A Global Perspective*, pp.1–14. doi:10.1080/19393555.2021.1995537.

Nazah, S., Huda, S., Abawajy, J. and Hassan, M.M. (2020). Evolution of Dark Web Threat Analysis and Detection: A Systematic Approach. *IEEE Access*, [online] 8, pp.171796–171819. Available at: https://ieeexplore.ieee.org/abstract/document/9197590/.

Oladoyinbo, T.O., Adebiyi, O.O., Ugonnia, J.C., Olaniyi, O. and Okunleye, O.J. (2023). *Evaluating and Establishing Baseline Security Requirements in Cloud Computing: An Enterprise Risk Management Approach*. [online] Social Science Research Network. doi: https://doi.org/10.2139/ssrn.4612909.

Ovsyannikova, A.S. and Sidorenko, V.G. (2022). *Experience in Finding and Registering Vulnerabilities in Web Applications*. [online] IEEE Xplore. doi:10.1109/ITQMIS56172.2022.9976819.

Oztemel, E. and Gursev, S. (2018). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, [online] 31(31). Available at: https://link.springer.com/article/10.1007/s10845-018-1433-8.

Palit, T., Monrose, F. and Polychronakis, M. (2019). Mitigating data leakage by protecting memory-resident sensitive data. *Proceedings of the 35th Annual Computer Security Applications Conference*. doi:10.1145/3359789.3359815.

Patwary, A.A.-N., Naha, R.K., Garg, S., Battula, S.K., Patwary, M.A.K., Aghasian, E., Amin, M.B., Mahanti, A. and Gong, M. (2021). Towards Secure Fog Computing: A Survey on Trust Management, Privacy, Authentication, Threats and Access Control. *Electronics*, [online] 10(10), p.1171. Available at: https://www.mdpi.com/2079-9292/10/10/1171.

Perez-Cabo, D., Jimenez-Cabello, D., Costa-Pazo, A. and Lopez-Sastre, R.J. (2019). *Deep Anomaly Detection for Generalized Face Anti-Spoofing*. [online] openaccess.thecvf.com. Available at: http://openaccess.thecvf.com/content_CVPRW_2019/html/CFS/Perez-Cabo_Deep_Anomaly_Detection_for_Generalized_Face_Anti-Spoofing_CVPRW_2019_paper.html.

Phokela, K.K., Singi, K., Dey, K., Kaulgud, V. and Burden, A.P. (2022). *Framework to Assess Policy Driven Security Misconfiguration Risks in Cloud Native Application*. [online] IEEE Xplore. doi:10.1109/SecDev53368.2022.00023.

Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S. and Fang, B. (2020). A survey on Access Control in the Age of Internet of Things. *IEEE Internet of Things Journal*, pp.1–1. doi:10.1109/jiot.2020.2969326.

Quyen, V. and Van, N. (2023). *Hands-on Training for Mitigating Web Application*. [online] Available at: https://dspace02.jaist.ac.jp/dspace/bitstream/10119/18734/5/paper.pdf.

R Shaikh (2019). *Defending Cross-Site Request Forgery (CSRF) Attacks on Web Applications - ProQuest*. [online] www.proquest.com. Available at: https://search.proquest.com/openview/8535a9862f4b8e8b7c82b25cf7aec196/1?pq-origsite=gscholar&cbl=18750&diss=y.

Rahman, S. (2016a). The advantages and disadvantages of using qualitative and quantitative approaches and methods in language 'testing and Assessment' Research: a Literature Review. *Journal of Education and Learning*, [online] 6(1), pp.102–112. Available at: https://files.eric.ed.gov/fulltext/EJ1120221.pdf.

Rahman, S. (2016b). The advantages and disadvantages of using qualitative and quantitative approaches and methods in language 'testing and Assessment' Research: a Literature Review. *Journal of Education and Learning*, [online] 6(1), pp.102–112. Available at: https://files.eric.ed.gov/fulltext/EJ1120221.pdf.

Rajkumar, V., Prakash, M. and Vennila, V. (2022). Secure Data Sharing with Confidentiality, Integrity and Access Control in Cloud Environment. *Computer Systems Science and Engineering*, [online] 40(2), pp.779–793. Available at: https://pdfs.semanticscholar.org/3e4a/7285fe8e45ac8792936f6021a05b742da6ba.pdf.

Rashid, M.A. and Pajooh, H.H. (2019). *A Security Framework for IoT Authentication and Authorization Based on Blockchain Technology*. [online] IEEE Xplore. doi:10.1109/TrustCom/BigDataSE.2019.00043.

Richards, G. (2020). Designing creative places: The role of creative tourism. *Annals of Tourism Research*, [online] 85(1), p.102922. Available at: https://www.sciencedirect.com/science/article/pii/S0160738320300669.

Rodríguez, G.E., Torres, J.G., Flores, P. and Benavides, D.E. (2020). Cross-site scripting (XSS) attacks and mitigation: A survey. *Computer Networks*, 166, p.106960. doi: 10.1016/j.comnet.2019.106960.

Sabelström, A. (2023). *SQL injection attacks and countermeasures in PHP, and Cross-Site Request Forgery*. [online] *www.diva-portal.org*. Available at: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1787822.

Sagrillo, M., Guerra, R.R., Machado, R., and Bayer, F.M. (2023). A generalized control chart for anomaly detection in SAR imagery. *Computers & Industrial Engineering*, [online] 177, p.109030. doi: 10.1016/j.cie.2023.109030.

Sai, A.R., Buckley, J. and Le Gear, A. (2019). Privacy and Security Analysis of Cryptocurrency Mobile Applications. *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*. doi:10.1109/mobisecserv.2019.8686583.

Sarkar, S. (2021). Detecting Vulnerabilities of Web Application Using Penetration Testing and Prevent Using Threat Modeling. *Lecture Notes in Electrical Engineering*, pp.21–32. doi:10.1007/978-981-15-8752-8_3.

SECURING WEB APPLICATIONS ON NODE.JS PLATFORM Himachal Pradesh. (n.d.). [online] Available at: http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/6488/1/Securing%20Web%20Applications%20on%20Node.JS%20Platform.pdf.

Sinha, A.K. and Tripathy, S. (2019). CookieArmor : Safeguarding against cross-site request forgery and session hijacking. *Security and Privacy*, 2(2), p.e60. doi:10.1002/spy2.60.

Siponen, M., Soliman, W. and Holtkamp, P. (2021). Research Perspectives: Reconsidering the Role of Research Method Guidelines for Interpretive, Mixed Methods, and Design Science Research. *Scopus: 85111126616*. [online] Available at: https://osuva.uwasa.fi/handle/10024/12956.

Smallwood, A. (2023). *Primary encounters: towards a conceptual model of place relations in outdoor adventure education*. [online] insight.cumbria.ac.uk. Available at: http://insight.cumbria.ac.uk/id/eprint/7326/.

Solina Pérez, A. (2022). Cybersecurity threats and mitigation in web applications. *upcommons.upc.edu*. [online] Available at: https://upcommons.upc.edu/handle/2117/378898.

T Sipponen (2022). *Information Security Testing Plan Model for Secure Software Development*. [online] Available at: https://www.theseus.fi/bitstream/handle/10024/781556/Sipponen_Tommi.pdf?sequence=2.

Taleby, M., Li, Q., Rabbani, M. and Raza, A. (2017). A Survey on Smartphones Security: Software Vulnerabilities, Malware, and Attacks. *International Journal of Advanced Computer Science and Applications*, 8(10). doi:10.14569/ijacsa.2017.081005.

Tariq, I., Sindhu, M.A., Abbasi, R.A., Khattak, A.S., Maqbool, O. and Siddiqui, G.F. (2021). Resolving cross-site scripting attacks through genetic algorithm and reinforcement learning. *Expert Systems with Applications*, 168, p.114386. doi: 10.1016/j.eswa.2020.114386.

Thirunahari, S. (2023). A development framework for software integration projects – case study: Web app Integration with OpenWeather API. *www.researchbank.ac.nz*. [online] Available at: https://www.researchbank.ac.nz/handle/10652/6066.

Tracy, S.J. (2019). *Qualitative Research Methods: Collecting Evidence, Crafting Analysis, Communicating Impact*. [online] *Google Books*. John Wiley & Sons. Available at: https://books.google.com/books?hl=en&lr=&id=ipOgDwAAQBAJ&oi=fnd&pg=PR1&dq=To+design+a+security+strategy+for+a+selected+web+application+using+a+qualitative+approach.

V Braun (2020). *APA PsycNet*. [online] psycnet.apa.org. Available at: https://psycnet.apa.org/journals/qua/9/1/3/.

Weamie, S.J.Y. (2022). Cross-Site Scripting Attacks and Defensive Techniques: A Comprehensive Survey. *International Journal of Communications, Network and System Sciences*, [online] 15(08), pp.126–148. doi:10.4236/ijcns.2022.158010.

Wifeling, S., Patil, T., Dürmuth, M. and Luigi Lo Iacono (2020). Evaluation of Risk-Based Re-Authentication Methods. *IFIP advances in information and communication technology*, [online] pp.280–294. Available at: https://link.springer.com/chapter/10.1007/978-3-030-58201-2_19.

World Health Organization (2021). *Strategic Toolkit for Assessing Risks A comprehensive toolkit for all-hazards health emergency risk assessment*. [online] Available at: https://apps.who.int/iris/bitstream/handle/10665/348763/9789240036086-eng.pdf.

Yaacoub, J.-P.A., Salman, O., Noura, H.N., Kaaniche, N., Chehab, A. and Malli, M. (2020). Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, [online] 77, p.103201. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7340599/.

Zidianakis, E., Partarakis, N., Ntoa, S., Dimopoulos, A., Kopidaki, S., Ntagianta, A., Ntafotis, E., Xhako, A., Pervolarakis, Z., Kontaki, E., Zidianaki, I., Michelakis, A., Foukarakis, M. and Stephanidis, C. (2021). The Invisible Museum: A User-Centric Platform for Creating Virtual 3D Exhibitions with VR Support. *Electronics*, [online] 10(3), p.363. Available at: https://www.mdpi.com/2079-9292/10/3/363.