assignment

by Azam Fareed

Submission date: 22-Feb-2024 09:42AM (UTC-0500)
Submission ID: 2168146038
File name: ESSAY_ASSIGNMENT.docx (30.69K)
Word count: 5420
Character count: 32400

ABSTRACT

PHP continues to be one of the most popular programming languages for online development, being the engine behind a large number of websites and web apps. But because of its widespread use, attackers looking to take advantage of weaknesses in PHP-based online applications find it to be a highly attractive target. The numerous security flaws that affect PHP online applications are examined in this paper. These flaws range from common ones like SQL injection, cross-site scripting (XSS), and unsafe file uploads to more sophisticated ones like session hijacking and remote code execution. This paper intends to increase developer knowledge and provide them the tools they need to create more secure PHP web applications by looking at the underlying causes of these security issues and offering mitigation solutions. In the constantly growing field of web application security, it also covers the significance of secure coding techniques, continuous maintenance, and the usage of security technologies to ward off new threats.

INTRODUCTION

PHP is one of the most widely used and adaptable scripting languages in the ever-changing field of web development; it powers a large percentage of websites and web apps worldwide. PHP has become a vital component of contemporary online development due to its wide range of database and web server compatibility, ease of use, and robust community support. To guarantee the availability, confidentiality, and integrity of PHP-based online applications, developers must handle a variety of security issues and vulnerabilities brought on by this extensive use (Ariyathilake G et al, 2023).

In today's interconnected digital economy, where cyber dangers are real and data breaches can have far-reaching effects, web application security is critical. Like any other software system, PHP online applications are vulnerable to a variety of security issues, including as injection attacks, cross-site scripting (XSS), authentication flaws, and the exposing of sensitive data. Malicious actors may take use of these vulnerabilities to compromise the application, steal confidential data, interfere with services, or even obtain unauthorized access to the server infrastructure (Aslan Ö et al, 2023).

Developers, system administrators, and security experts responsible with protecting PHP online applications from potential attackers must comprehend the nature of these security vulnerabilities.

We examine prevalent vulnerabilities, underlying causes, potential repercussions, and recommended practices for mitigation in this study, which delves deeply into the topic of security challenges in PHP online applications (Neshenko N et al, 2023).

Injection attacks are one of the most common security threats in PHP web applications, with SQL injection (SQLi) being the main cause for concern. When malicious input is introduced into SQL queries that are run by the application's database, it's known as SQL injection. This enables attackers to alter the logic of the query and maybe obtain unauthorized access to private information or run arbitrary SQL commands (Agbakwuru & Njoku, 2021). This vulnerability is frequently caused by inadequate use of prepared statements or parameterized queries, which do not sufficiently isolate user input from SQL commands, and by inappropriate input validation. Another serious security flaw in PHP web applications is called cross-site scripting (XSS), which allows attackers to insert harmful code into web pages that other users are viewing (Hydara, 2021). XSS attacks can be used to take control of a victim's session cookies, deface websites, send users to malicious websites, or carry out illegal operations on their behalf. These flaws usually result from inadequate output encoding and input sanitization, which gives attackers the ability to insert malicious JavaScript code that is run in the victim's browser (Sharif, 2022).

Another prominent security hazard in PHP online applications is insecure authentication procedures, whereby poorly designed or weakly implemented authentication can result in unauthorized access to confidential data (Mathas, 2021). This covers problems including weak password hashing algorithms, inconsistent session identifiers, the absence of multi-factor authentication, and inadequate account lockout procedures. By taking advantage of these flaws, attackers can elevate their privileges within the program, assume the identity of authentic users, or obtain unauthorized access to user accounts.

PHP web applications are vulnerable to a variety of sensitive data exposure scenarios, in which private data—such as credit card numbers, passwords, or personally identifiable information (PII) may unintentionally be revealed to unapproved parties. Sensitive data storage errors, insufficient transmission security, or inadvertent exposure via error messages, logs, or debug information can all lead to this. According to Sinha et al. (2019), attackers may take advantage of these vulnerabilities to steal confidential data for financial gain, identity theft, or other nefarious intents. PHP online applications are susceptible to more complex attacks, such as session hijacking, remote

code execution (RCE), file inclusion vulnerabilities, and server misconfigurations, in addition to these widespread flaws. Unauthorized control of a user's session identifier, or "session hijacking," enables attackers to assume the identity of the victim and access their account without using the victim's credentials. Vulnerabilities related to remote code execution (RCE) occur when an attacker gains the ability to run arbitrary code on the server that hosts the PHP application, which could result in the total compromise of the system.

When an application includes external files or resources without conducting the necessary validation, it creates file inclusion vulnerabilities, such as local file inclusion (LFI) and remote file inclusion (RFI), which give attackers access to sensitive files on the server's file system or the ability to include malicious files from remote locations. The security vulnerabilities associated with PHP online applications are further compounded by server misconfigurations, insecure permissions, and obsolete software components, which give attackers more avenues for exploitation (Alrawi et al, 2019).

A multi-layered strategy including secure coding techniques, strong input validation, appropriate output encoding, safe authentication methods, frequent security audits, and proactive monitoring of web application logs and traffic patterns is needed to mitigate these security risks. It is recommended that developers follow security principles that have been set forth, make use of secure coding frameworks and libraries, and keep up with the latest developments in web application security best practices and threats (Cope, 2020).

Because of the always changing threat landscape and the possible consequences of security breaches for both enterprises and their customers, security is still a top priority when developing and deploying PHP web applications. Developers may improve the security posture of PHP online applications and protect against potential threats by knowing the common security issues, underlying vulnerabilities, and best practices for mitigation. By maintaining constant watchfulness, working together, and adhering to security-first principles, we can guarantee the integrity and robustness of PHP-based web applications against new and existing cyber threats.

THEORETICAL BACKGROUND

An increasing number of dynamic and interactive web applications are being developed as a result of PHP's widespread usage in web development. However, security flaws in PHP-based online

apps are a serious worry that come along with this quick expansion. In order to protect sensitive data, guarantee user privacy, and preserve the integrity of web systems, it is imperative that these security concerns are addressed as the internet grows more and more integrated into daily life (Abba Ari et al., 2024).

This project's impetus comes from the realization that, in the digital age, web application security is vital. Because cyber threats are becoming more sophisticated and frequent, it is critical for developers, system administrators, and security experts to stay up to date on new vulnerabilities and use effective mitigation techniques.

Moreover, the fact that PHP is widely used in web development emphasizes how important it is to comprehend and address security threats unique to this programming language. PHP is prone to a variety of security vulnerabilities, including injection attacks, authentication problems, and the exposing of sensitive data, despite its widespread use and adaptability (Abba Ari et al., 2024).

This study's motivation stems from the realization that there is a need for comprehensive guidelines and tools to successfully address security challenges, despite the growing demand for secure PHP online applications. This project intends to give developers a strong foundation for creating secure and robust PHP web applications by exploring the theoretical foundations of web application security, PHP language features, and frequent vulnerabilities.

Moreover, by investigating useful mitigation techniques, best practices, and tools designed especially for PHP-based systems, the project hopes to add to the body of knowledge on online application security. The project intends to improve the overall security posture of PHP online applications by enabling developers to proactively identify, mitigate, and avoid security problems through the dissemination of this information.

RELATED WORKS

WEB APPLICATION SECURITY

A wide range of procedures, guidelines, and technological advancements are included in web application security, with the goal of protecting web applications from attacks and weaknesses. Ensuring the security of web applications is crucial as they become more and more prevalent on the internet and ingrained in daily life. Web application security is primarily concerned with

defending online applications from several types of cyber threats, such as denial-of-service (DoS) assaults, injection attacks, data breaches, and unauthorized access. According to Sengupta et al. (2020), it comprises an all-encompassing strategy spanning several tiers of protection, such as network security, application security, user security, and infrastructure security.

Confidentiality, or making sure that only authorized users may access sensitive information, is one of the core tenets of online application security. To avoid unauthorized access or disclosure of sensitive data, this calls for the implementation of strong authentication systems, access controls, and encryption protocols.

Maintaining the correctness and reliability of data and resources is the emphasis of integrity, another essential component of web application security. Organizations can identify and stop illegal data tampering and modifications by putting procedures like digital signatures, checksums, and data validation into place.

Availability plays a crucial role in web application security by guaranteeing uninterrupted and downtime-free access to web applications for authorized users. This entails putting redundancy and failover measures in place, reducing the possibility of denial-of-service attacks, and keeping an eye out for unusual traffic patterns that might point to impending attacks or disruptions.

Web application security relies heavily on authentication and authorization systems, which confirm users' identities and provide the right access privileges according to their roles and permissions. Robust authentication procedures, such two-factor authentication and biometric authentication, lessen the possibility of unwanted access, while access controls make sure that users can only do tasks that are permitted.

In addition, a variety of best practices, instruments, and technologies created to lessen particular risks and vulnerabilities are included in web application security. This covers intrusion detection systems (IDS), web application firewalls (WAF), vulnerability assessment tools, secure coding techniques, and security incident response protocols.
In order to reduce potential risks and vulnerabilities, web application security is a broad discipline that calls for a proactive and all-encompassing strategy. Organizations may safeguard critical data and web applications from cyber attacks and provide users with a safe and reliable online experience by putting strong security measures in place.

- **INVASION OF STRUCTURED LANGUAGE**

To fully ensure information security, programmers frequently set up a segment of code in site development programs to gauge the accuracy of data entered by other users. However, given the use of PHP in website design and development, failing to include preset codes in the design could allow some unauthorized users to exploit this vulnerability to query databases, control background based on codes returned by the databases, and steal or manipulate user data, all of which pose serious risks to information security in website design. While the implementation of structured language settings has no impact on website performance, its flaws can pose a significant risk to websites' information security defense. SQL holds a significant position among the commonly used structured languages in use today (figure 1). Information security can be somewhat ensured by using SQL correctly in PHP website design. However, the question of how to combine these two elements while guaranteeing that structured language invasion codes are given full rein has currently emerged as a problem requiring an immediate fix in PHP website design.

- **XSS (Cross-site Script Execution)**

The term "XSS," which stands for "Cross Site Script Execution," refers to the basic idea that attackers might compromise user privacy by creating a website program that allows them to bypass information security defenses and gain access to other users' systems and data. Such behavior typically manifests as malicious user information theft and website attacks, both of which have a major negative impact on the information security of websites. Small-page pop-up windows on websites actually indicate that attackers are using cross-site script execution programs to trick users into downloading Trojan viruses. This paralyzes the user's information security supervision system and ultimately compromises the security of the website.

Given how frequently online applications are used in daily life, their vulnerability is growing. XSS is the most often exploited security flaw in modern online apps (Steffens, 2021). As an injecting variation, cross-site scripting modifies the client-side script that the targeted browsers have installed. When a web program uses incorrect or unencoded user input in its output, cross-site scripting (XSS) occurs. By injecting malicious scripts where a web application accepts user input, cross-site scripting (XSS) can cause serious harm to both the user and the website. Invalid inputs

may result in the sharing of personal information and the theft of user accounts and cookies (Hernández, 2022).

- **PHP VULNERABILITIES**

Even though PHP is one of the most used server-side programming languages for web development, it still has security flaws. The security and integrity of PHP-based online applications may be seriously jeopardized if these vulnerabilities are not fixed. To successfully minimize possible threats, developers, system administrators, and security professionals must understand the nature of these vulnerabilities.

SQL injection (SQLi), which happens when malicious SQL queries are injected into the application's input fields, is one of the most common vulnerabilities in PHP. Attackers may be able to run arbitrary SQL instructions as a result, or they may gain unwanted access to confidential data. Inadequate input validation and the absence of parameterized queries or prepared statements are common causes of SQL injection vulnerabilities (Ahmad & Karim, 2021).

Cross-site scripting (XSS) is another prevalent vulnerability that allows attackers to insert malicious scripts into web pages that are being viewed by other users. XSS vulnerabilities may be leveraged to take advantage of session cookies, alter webpages, or send visitors to unsafe websites. Usually, inadequate input sanitization and output encoding are the source of these vulnerabilities (Iannone et al, 2022).

Another serious weakness in PHP web applications is insecure file uploads. Attackers may upload malicious files by taking advantage of flaws in file upload features, which could result in remote code execution or unauthorized access to the file system on the server. According to Alanazi et al. (2023), this vulnerability frequently arises from improper file permissions, insufficient checks on file content, or inadequate validation of file types.

Furthermore, PHP online applications could be the target of authentication bypass attacks, in which criminals take advantage of holes in the security protocols to access private data without authorization. Weak password regulations, session identifiers that are predictable, or incorrect access control implementation can all lead to this.

It takes a proactive strategy that includes secure coding techniques, frequent security audits, and the application of security frameworks and tools to mitigate these risks. To stop SQL injection and XSS attacks, developers should use parameterized queries, output encoding, and appropriate input validation. Strict file permissions, content checks, and secure file upload techniques can all help reduce the risk of file upload vulnerabilities.
Strong password restrictions and multi-factor authentication are two examples of robust authentication techniques that can assist stop authentication bypass attempts. To fix known vulnerabilities and guarantee the general security of PHP-based web applications, regular security updates and patches should also be issued to PHP and its related libraries.

- **CYBER THREATS**

Cyber threats are a constant and changing hazard to people, businesses, and vital infrastructure around the globe in today's interconnected digital world. These dangers cover a wide range of malevolent actions carried out by hacktivists, nation-state actors, cybercriminals, and other threat actors. Organizations must comprehend the nature of cyber dangers in order to successfully protect against such attacks and reduce related risks (Ghelani, 2022).

Malware, which includes a variety of harmful software programs intended to corrupt networks, steal confidential data, or interfere with operations, is one of the most common cyberthreats. Malware encompasses several malicious programs such as viruses, worms, Trojan horses, ransomware, and spyware, each with unique traits and ways of spreading. Malware attacks are a serious risk for businesses of all sizes because they can lead to data breaches, financial losses, and reputational harm (Perera et al., 2022).

Phishing is another prevalent cyberthreat that is using false emails, websites, or messages to fool people into disclosing personal information like passwords, usernames, or bank account information. Phishing attacks frequently take advantage of human weaknesses by tricking victims into doing acts that jeopardize their security through the use of social engineering techniques. Financial fraud, identity theft, and illegal access to private networks and data are all possible outcomes of phishing assaults (Alkhalil, 2021).

One particularly sneaky cyberthreat that has grown more common in recent years is ransomware. Attackers using ransomware encrypt important information or systems, then demand payment

(often in bitcoin) to unlock the information or allow access again. Ransomware attacks have the potential to seriously impair an organization's operations, interfere with services, cause large financial losses, and harm an organization's reputation (Ryder & Yeo, 2021). Distributed denial-of-service (DDoS) attacks are another type of cyberthreat that aims to interfere with the availability of online services by flooding targeted systems with excessive traffic. DDoS attacks can be started via amplification techniques that take advantage of holes in internet protocols or by botnets made up of compromised devices. DDoS assaults have the potential to cause financial loss, service outages, and reputational harm to brands (Garg et al., 2019).

Advanced persistent threats (APTs) are another category of cyberthreats. APTs are complex, protracted campaigns that are managed by highly trained, well-funded threat actors. APTs usually target high-value assets, such government organizations, big businesses, or intellectual property, with the intention of stealing confidential data or engaging in espionage or sabotage. To avoid detection and stay persistent within targeted networks, innovative tactics, methods, and procedures (TTPs) are utilized by APTs (Zhu et al., 2021).

A multi-layered strategy including technology controls, personnel training, threat intelligence, and incident response skills is needed to mitigate cyber threats. To defend against malware and other online dangers, businesses should put strong cybersecurity measures in place, such as firewalls, intrusion detection systems (IDS), antivirus software, and encryption. Phishing simulations and employee awareness training can assist inform employees about prevalent cyberthreats and lower their vulnerability to social engineering assaults.

- **INFORM USERS OF INFORMATION SECURITY DEFENSE MEASURES**

Users of websites should be aware of information security procedures. The primary cause is the critical role that user participation plays in monitoring website information security. Users can partially prevent malevolent incursion if they direct their actions in line with pertinent standards. For example, a cross-site script execution program is genuinely present in the pop-up windows that appear while browsing websites. Clicking on the pop-up windows will allow hostile spies to access users' personal data. The fundamental idea behind cross-site script execution is that attackers use a website program's inadequate user screening as a point of entry to breach websites' information security defenses, directly infiltrate systems, and endanger the security of other users' information. Because of this phenomena, it is essential to follow users' data legally while also

educating users about information security defensive strategies when managing the information on PHP websites. The primary cause is that people's lifestyles have been greatly impacted by network technology, and the field's explosive growth has expedited the advancement of human society. As more and more people join the network family, the network is exposed to an increasing amount of user data. User data should be used legally and in a timely manner to prevent information damage. However, given the current website architecture, it is challenging for designers and developers without extensive knowledge to successfully follow dynamic data. Therefore, in order to improve information security management in PHP website design, the working principles of the Web are required, and data following capacity needs to be strengthened. Furthermore, it is important to improve the filtering of user input data because, in real life, hackers can use any user name to access website databases. To sum up, the issue of normalizing user information security needs to be tackled from the ground up.

Plans and procedures for incident response should be established by organizations in order to efficiently identify, contain, and lessen cyber dangers when they arise. Organizations can adjust their security posture by staying informed about new cyber threats and trends through proactive threat information gathering and sharing.

In conclusion, in the current digital era, cyber-attacks represent serious dangers to both individuals and enterprises. To effectively reduce associated risks, proactive and thorough cybersecurity measures are needed. Through comprehension of cyber dangers and the application of suitable mitigation tactics, entities can fortify their resilience and ward off possible cyberattacks.

- **STRENGTHENING INFORMATION SECURITY DEFENSE IN WEBSITE DESIGN**

There are a lot of issues with information security defense when designing PHP websites. Program statements, cross-site scripting, and structured language will all compromise a website's information security. Therefore, in order to improve information security defense in website design, negative issues should be addressed accordingly. For instance, in order to stop hackers from using structured language invasion to compromise websites, prominent designers are required to reserve certain codes for structured language. In order to defend against cross-site script execution, user information management must first be enthused and users must be informed of

pertinent precautions. To improve information security defense in PHP website design in the actual world, information security behavior should be normalized in practical defense and monitoring.

- **THE IMPORTANCE OF INFORMATION SECURITY DEFENSE IN PHP WEBSITE DESIGN**

WAMP technology is used by PHP and ASP, two dynamic languages, to integrate the server and database, which is crucial for optimizing website design. The modest benefits of PHP website design in real-world applications have improved websites' dependability and security. In particular, people's way of life has significantly changed as a result of the recent explosion of network technology and the onset of the network era. Networks have become indispensable to people and nations because they are present in all spheres of existence. The network transcended geographical limitations and geopolitical boundaries, making a priceless contribution to the integration of the global economy. Its widespread use has significantly improved people's quality of life. National and personal data has been exposed to networks in the network era. Information security is easily jeopardized after a network breakdown, and national information security is even more so. The significance of national information security and the attention paid to website information security by the country are evident from China's first national network security law. Because of this, the question of how to finish an effective information security defense in a PHP website design has emerged as one that needs an immediate resolution. By bolstering information security defenses in all areas of PHP website design, significant objectives related to PHP website design and promotion can be achieved, including the effective avoidance of elements that negatively impact website information safety and the promotion of safe and stable website operation. The Third International Workshop on Computer Sciences and Materials Engineering (IWMECS 2018)

RESEARCH METHODOLOGY

In this section, we specify our research hypotheses based on both complexity metrics and the security resources indicator. These metrics have been shown to be effective for vulnerability prediction in certain domains (Chakraborty, 2020). We analyze multiple dimensions of vulnerabilities: total number of vulnerabilities, total number of vulnerabilities changed over time, total number of vulnerabilities per thousand lines of code, total number of vulnerabilities changed over time, and vulnerability metrics per category. We also analyze modifications made to the

fourteen projects' combined code base. We might anticipate that over time, open source projects' vulnerability density should improve as vulnerabilities and mitigation strategies become more generally understood, particularly in regards to common vulnerabilities like SQL injection and cross-site scripting. This gives rise to three theories:
It is expected that open source web applications will become less vulnerable over time. Specifically, there will be a decrease in the density of cross-site scripting vulnerabilities and SQL injection vulnerabilities in open source web applications.

A. CODE COMPLEXITY

According to security professionals, complexity is security's worst enemy (Lewis, 2019). Complicated code is harder to test, maintain, and understand, which makes it easier for security flaws to be introduced and harder to locate and fix. Therefore, compared to simple code, complicated code ought to have more security flaws. In light of this logic, we have two theories on code complexity:

- • Compared to applications with lower code complexity, those with higher code complexity have more vulnerabilities.
- Applications with higher code complexity will increase the number of vulnerabilities in the code over time at a faster rate than applications with lower code complexity..

The depth of nested conditionals and loops is measured by the nesting complexity. Three forms of these complexity metrics are examined: total complexity, which is the sum of the complexity metrics for the entire application; average complexity, which is the average value of the metric per function; and maximum complexity, which is the complexity of the function with the highest value for the complexity metric.

B. SECURITY RESOURCES INDICATOR

Based on public security resources made available on the project website, we developed a score in our prior study to assess the significance of security to a project. A dedicated email alias for reporting security issues, a list or database of security vulnerabilities unique to the application, documentation of secure development practices, such as coding standards or methods to avoid common secure programming errors, are the four components that make up the Security Resource Indicator (SRI) metric. SRI, which ranges from 0 to 4, is the total of the four indicator components.

These indicators are different from a related set of indicators that Fortify used to study Java applications because we added the last two indicators mentioned above, which are more focused on developers than users of the application, and removed their indicator regarding easy access to security experts, which we found to be ambiguous. It is anticipated that projects whose developers prioritize security will eventually see an improvement in the code's security. While a project with a security focus would use proactive measures like developer education and code reviews to prevent vulnerabilities from being introduced into code, a project without a security focus is expected to operate in a reactive mode, fixing vulnerabilities as they are reported. Consequently, we postulate that with time, the vulnerability density of applications with high SRI values ought to decline.

CONCLUSION

The study concludes that, considering the prevalence of cyber risks in the digital sphere, it is critical to address security concerns in PHP web development. This study has identified significant vulnerabilities present in PHP applications by means of a thorough examination and analysis spanning several years. The most common ones include SQL injection, XSS, CSRF, and RCE, each of which poses different difficulties and possible threats to the confidentiality and integrity of web-based systems.

Throughout the development lifecycle, best practices and security measures must be included in order to mitigate these dangers. Software engineers need to give security top priority across the whole software engineering process, from adopting secure authentication methods to enforcing strict input validation. Our results highlight the usefulness of techniques like output escaping, parameterized queries, and secure session management in protecting PHP applications from prevalent attack vectors.

Furthermore, the study emphasizes how important it is for members of the PHP development community to maintain constant awareness and education. Developers need to be proactive in updating their knowledge and abilities to properly counter emerging risks as cyber-attacks advance and become more complex. This means using state-of-the-art tools and frameworks, keeping up with security trends, and encouraging a development culture that prioritizes security. Future studies should focus on expanding our knowledge of the dangers and mitigation techniques that are unique to PHP web development. Enhancing threat intelligence capabilities, making security controls in PHP frameworks more user-friendly, and creating thorough security

recommendations specific to the PHP ecosystem are all areas that deserve more investigation. In conclusion, handling security concerns in PHP web development is an essential component of responsible software engineering and goes beyond just being a technical need. We can protect the integrity, confidentiality, and availability of digital assets in an increasingly linked world by strengthening PHP applications against cyber-attacks by adopting proactive security measures and promoting a culture of continuous learning and growth.

REFERENCE

Ariyathilake, G. J. M., Sandeepanie, M. H. R., & Rupasinghe, P. L. (2021). SQL injection detection and prevention solution for web applications.

Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, *12*(6), 1333.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations. *IEEE Communications Surveys & Tutorials*, *21*(3), 2702-2733.

Agbakwuru, A. O., & Njoku, D. O. (2021). SQL Injection Attack on Web Base Application: Vulnerability Assessments and Detection Technique. *International Research Journal of Engineering and Technology*, *8*(3), 243-252.

Hydara, I. (2021). The limitations of cross-site scripting vulnerabilities detection and removal techniques. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(3), 1975-1980.

Sharif, M. H. U. (2022). Web Attacks Analysis and Mitigation Techniques. *International Journal of Engineering Research & Technology (IJERT)*, 10-12.

Mathas, C. M. (2021). *Secure coding practices for web applications* (Master's thesis, Πανεπιστήμιο Πειραιώς).

Sinha, P., kumar Rai, A., & Bhushan, B. (2019, July). Information Security threats and attacks with conceivable counteraction. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (Vol. 1, pp. 1208-1213). IEEE.

Alrawi, O., Lever, C., Antonakakis, M., & Monrose, F. (2019, May). Sok: Security evaluation of home-based iot deployments. In *2019 IEEE symposium on security and privacy (sp)* (pp. 1362-1380). IEEE.

Cope, R. (2020). Strong security starts with software development. *Network Security*, *2020*(7), 6-9.

Abba Ari, A. A., Ngangmo, O. K., Titouna, C., Thiare, O., Mohamadou, A., & Gueroui, A. M. (2024). Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges. *Applied Computing and Informatics*, *20*(1/2), 119-141.

Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1909-1941.

Steffens, M. (2021). Understanding emerging client-Side web vulnerabilities using dynamic program analysis.

Hernández Plaza, N. (2022). State-of-the art teaching material of the OWASP Top 10 (Master's thesis, Universitat Politècnica de Catalunya).

Ahmad, K., & Karim, M. (2021). A method to prevent SQL injection attack using an improved parameterized stored procedure. *International Journal of Advanced Computer Science and Applications*, *12*(6).

Iannone, E., Guadagni, R., Ferrucci, F., De Lucia, A., & Palomba, F. (2022). The secret life of software vulnerabilities: A large-scale empirical study. *IEEE Transactions on Software Engineering*, *49*(1), 44-63.

Alanazi, M., Mahmood, A., & Chowdhury, M. J. M. (2023). SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues. *Computers & Security*, *125*, 103028.

Ghelani, D. (2022). Cyber security, cyber threats, implications and future perspectives: A Review. *Authorea Preprints*.

Perera, S., Jin, X., Maurushat, A., & Opoku, D. G. J. (2022, March). Factors affecting reputational damage to organisations due to cyberattacks. In *Informatics* (Vol. 9, No. 1, p. 28). MDPI.

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, *3*, 563060.

Delia, E., Sveinson, K., & Ryder, S. (2022). A CRITICAL DISCOURSE ANALYSIS OF REACTIONS TO THE INAUGURAL TOUR DE FRANCE FEMMES ROUTE. In *BOOK OF ABSTRACTS* (Vol. 34, p. 51).

Battelino, T., Danne, T., Bergenstal, R. M., Amiel, S. A., Beck, R., Biester, T., ... & Phillip, M. (2019). Clinical targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time in range. *Diabetes care*, *42*(8), 1593-1603.

Zhang, Y., Zeng, G., Pan, H., Li, C., Hu, Y., Chu, K., ... & Zhu, F. (2021). Safety, tolerability, and immunogenicity of an inactivated SARS-CoV-2 vaccine in healthy adults aged 18–59 years: a randomised, double-blind, placebo-controlled, phase 1/2 clinical trial. *The Lancet infectious diseases*, *21*(2), 181-192.

Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2021). Deep learning based vulnerability detection: Are we there yet. *IEEE Transactions on Software Engineering*.

Lewis, T. G. (2019). *Critical infrastructure protection in homeland security: defending a networked nation*. John Wiley & Sons.

assignment

ORIGINALITY REPORT

11%	8%	6%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.luisolis.com Internet Source	4%
2	www.mdpi.com Internet Source	1%
3	Submitted to University of Westminster Student Paper	<1%
4	Submitted to Universiti Sains Malaysia Student Paper	<1%
5	Maureen Doyle. "An Empirical Study of the Evolution of PHP Web Application Security", 2011 Third International Workshop on Security Measurements and Metrics, 09/2011 Publication	<1%
6	www.researchgate.net Internet Source	<1%
7	Submitted to Feversham Academy Student Paper	<1%
8	Submitted to Harrisburg University of Science and Technology Student Paper	<1%

9	Submitted to HCUC Student Paper	<1%
10	Submitted to Colorado Technical University Online Student Paper	<1%
11	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
12	Submitted to University of Greenwich Student Paper	<1%
13	arxiv.org Internet Source	<1%
14	www.ncbi.nlm.nih.gov Internet Source	<1%
15	www.cybersecuritylawsrc.com Internet Source	<1%
16	Submitted to Higher Education Commission Pakistan Student Paper	<1%
17	Abdulbasit A. Darem, Asma A. Alhashmi, Tareq M. Alkhaldi, Abdullah M. Alashjaee, Sultan M. Alanazi, Shouki A. Ebad. "Cyber Threats Classifications and Countermeasures in Banking and Financial Sector", IEEE Access, 2023 Publication	<1%

18	securitytrails.com Internet Source	<1 %
19	Tinshu Sasi, Arash Habibi Lashkari, Rongxing Lu, Pulei Xiong, Shahrear Iqbal. "A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges", Journal of Information and Intelligence, 2023 Publication	<1 %
20	al-kindipublisher.com Internet Source	<1 %
21	ebin.pub Internet Source	<1 %
22	export.arxiv.org Internet Source	<1 %
23	hrmars.com Internet Source	<1 %
24	openrepository.aut.ac.nz Internet Source	<1 %
25	www.growingscience.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		