iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** X **Month of publication:** October 2024

DOI: https://doi.org/10.22214/ijraset.2024.64566

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

Unveiling the Underbelly of Web Application Vulnerabilities: A Critical Exploration

Mistry Mahima Ankit[1], Kiran Dodiya[2]
[1]*M.Sc. Cyber Security, NSIT-IFSCS Affiliated to National Forensic Sciences University, Gandhinagar, Gujarat, INDIA*
[2]*Assistant Professor & Program Co-Ordinator of DFIS (Cyber Security & Digital Forensics) NSIT-IFSCS (Affiliated to National Forensic Sciences University), Gandhinagar, Gujarat, INDIA*

Abstract: ***This study reveals web application weaknesses and demonstrates how frequent security flaws permit unauthorized entry to web solutions. Many web applications are at risk due to the secrecy of the data they store. Recognizing this theme plays a key role in identifying the threats in play. We examined the OWASP's Top 10 weaknesses together with Session Hijacking and Weak Password Management.***
Violent Monkey shows the methods to take advantage of this breach by mixing practical exploration with tools, including Burp Suite and Nmap, Wireshark, and browser extensions Cookie Editor. With Cookie Editor at hand, session hijacking happens in a moment as session cookies can be easily gathered with Google Dorking and transmitted to a premium account. Violent monkey effectively represents a key illustration of privilege escalation. An authorized user can access premium features by placing a script in the client component of a web service.
Errors occur in managing passwords because a 10-thousand-character password gets made and endorsed without input validation from the system. Thanks to these weaknesses, hackers gain unauthorized access and compromise data. Investigating strong vulnerability management motivates this project and encourages additional research into how machine learning can identify weaknesses and provide timely threat-related data.
Keywords: web application flaws, disclosure of confidential data, web application security weakness and their rudiments

I. INTRODUCTION

The most important applications on the web today include social networking and online banking, which are more appealing to attackers as people catch on. Flaws in web-based application development result in more attackers. By targeting these flaws, users, and providers face security breaches and repercussions from cybercriminals exploiting their access. This paper examines the hidden risks of web applications while concentrating on how minor errors can lead to unauthorized access to protected systems. Top applications in today's web ecosystem include social networks and online banking platforms, which attract more threat vectors as users understand them.

Flaws in the design and development of web applications cause a rise in attack numbers. Attackers will exploit these defects to breach private data protection and skirt safety protocols. The potential increase in access rights and outcomes for users and service companies could occur.

This work analyses the riskier implications of vulnerabilities in web applications and how minor mistakes may lead to unauthorized entry into secured assets. I am curious about how hackers gain private data from sites without alerting the users. Actions employed cover cookie theft and running scripts. All these techniques take advantage of the security gaps to penetrate high-level accounts or disclose information that should not be exposed to the public. A common problem highlighted is the alteration of session cookies containing login details that usually lack protection. Furthermore, the intruder can assume a selectable user by using or changing valid cookies.

Access to capabilities and resources will be granted to those users. By letting hackers act in users' browsers through unsecured client-side scripts, they can be traced and discovered easily. This study aims to demonstrate how these attacks fit into real-life situations by presenting case examples where the author can effortlessly exploit these weaknesses. Investigating these examples will improve comprehension of web application security breaches, frequent vulnerabilities, and relevant defenses. We hope to discover the weaknesses and encourage better protection measures by examining this closely. Crucial remedies for the problems identified are essential for preserving the trustworthiness and stability of the online realm.[1].

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

1	Finding Vulnerability in Web Applications by Using Pentesting	The study emphasizes penetration testing's role in uncovering vulnerabilities like SQL injection and cross-site scripting, which are essential for understanding and mitigating the dark side of web application security risks[2].
2	Identifying and Mitigating Common Web Application Vulnerabilities	The paper focuses on identifying and mitigating web application vulnerabilities, emphasizing the importance of securing applications against threats like SQL injection and XSS before real-world attacks occur[3].
3	Vulnerabilidades en aplicaciones web, amenazas y ataques	Web application vulnerabilities, such as code injection and XSS, expose systems to malicious attacks, leading to unauthorized access, data breaches, and significant security risks for organizations[4].
4	Web-based Vulnerabilities Modulation: A Comprehensive Study on Web Vulnerabilities	The study focuses on prevalent web vulnerabilities like SQL injection and XSS, analyzing how attackers exploit these weaknesses to compromise data confidentiality, integrity, and availability[5].
5	Analysis of vulnerabilities and security problems of web applications	The analysis highlights various web application vulnerabilities, including broken access control and SQL injections, emphasizing the need for comprehensive security testing throughout the software development life cycle[6].
6	Gestão de Vulnerabilidade em Aplicações WEB: Exploração de SQL Injection	The paper emphasizes the significance of adopting OWASP guidelines to mitigate SQL Injection vulnerabilities, highlighting the need for robust security practices in web applications to prevent data compromise[7].
7	Assessing Web Application Security Through Vulnerabilities in Programming Languages and Environments	The paper emphasizes the prevalence of vulnerabilities in web applications, highlighting the need for early cybersecurity integration in the software development lifecycle to mitigate risks effectively[8].
8	Attacks on Web Applications	The chapter discusses various web application vulnerabilities, including insecure design and security misconfiguration, highlighting the risks of data theft and exploitation by attackers[9].
9	Systematic Review of Common Web-Application Vulnerabilities	Web application vulnerabilities, such as XSS and SQL injection, enable attackers to exploit systems, leading to unauthorized data access and significant security breaches[10].
10	Web Application Vulnerabilities and Best Practices: A Comprehensive Analysis	The paper analyzes various web application vulnerabilities, including SQL injection and XSS, highlighting their risks and proposing best practices for effectively mitigating these security threats[11].

Table 1 Literature Review

A. Problem Statement

Web applications are increasingly vulnerable to security threats such as SQL Injection, Cross-Site Scripting (XSS), and inadequate access controls, resulting in significant risks, including data breaches, account takeovers, and unauthorized access to sensitive information. Despite the availability of security tools and best practices, many applications still exhibit fundamental weaknesses due to insufficient input validation, poor access control mechanisms, and misconfigured security settings. This research aims to identify these vulnerabilities, analyze exploitation techniques, assess their impacts, and recommend effective mitigation strategies to enhance web application security.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

III. METHODOLOGY

A. Research Approach

A combination of hands-on testing and automatic tools was established to reveal flaws in web applications. The main tools used in vulnerability assessment are Burp Suite, Nmap, and Wireshark. Some Google plugins like Nmap and Violentmonkey helped me evaluate the website. We discovered additional technologies through Wappalyzer's insights into the website. It allowed me to find specific focal points that emphasize particular vulnerabilities. Cookie Editor enabled us to exploit a premium account successfully. Violentmonkey helped us run scripts that led to acquiring a pro account. The system permitted a password extending to around 10 thousand characters- uncovering a possible flaw in password check systems. Administration access is granted by tacking "/admin" on most websites on the URL endpoint. Often, this acts as a gateway displaying a susceptible entry that web application security must manage.

B. Tools Used in the Research

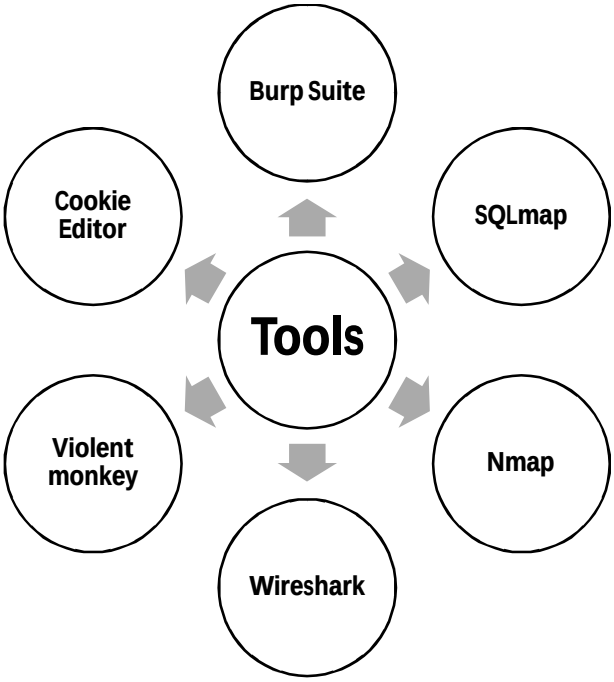


Figure 1 Tools Used in the Research

During this research, various tools were employed to identify, analyze, and exploit vulnerabilities within web applications. Each tool served a specific purpose, contributing to the overall effectiveness of the penetration testing process. Below is a detailed overview of the key tools used:

1) Burp Suite

Burp Suite is a powerful web application security testing framework that provides many features for identifying vulnerabilities. Its main components include:

a) Proxy Server: Burp Suite is an intercepting proxy that allows users to capture and modify HTTP/S requests and responses. This capability is critical for analyzing how the web application handles input and output data.
b) Scanner: The automated scanner identifies common vulnerabilities such as SQL Injection, XSS, and CSRF, providing a baseline security assessment[12].
c) Intruder: This tool performs automated attacks on web applications, such as brute force attacks against login forms or session tokens.
d) Repeater: Users can modify and resend individual requests to test the application's response to different inputs.

By utilizing Burp Suite, the research could efficiently test for vulnerabilities while monitoring how the application processed various input types.

©IJRASET: All Rights are Reserved

2) SQLmap

SQLmap is an open-source penetration testing tool designed to detect and exploit SQL Injection vulnerabilities. Key features include:

a) Automated Testing: SQLmap automates the process of identifying SQL Injection flaws, significantly reducing the time and effort required for manual testing.

b) Database Management: Once a vulnerability is confirmed, SQLmap can enumerate databases, tables, and columns, providing insights into the backend's structure.

c) Privilege Escalation: SQLmap can execute arbitrary SQL commands, allowing testers to escalate privileges and access sensitive data.

d) SQLmap proved invaluable in the research by enabling a detailed exploration of SQL Injection vulnerabilities and demonstrating how they could be exploited to access confidential information[13].

3) Nmap

Nmap (Network Mapper) is a widely used network scanning tool that helps identify open ports and services on target systems. Key functionalities include:

a) Service Discovery: Nmap can detect running services on various ports, providing insights into potential vulnerabilities associated with specific software versions.

b) OS Detection: The tool can identify the operating system and version running on a host, allowing for targeted attacks based on known vulnerabilities.

c) Scripting Engine: Nmap's scripting capabilities enable automated vulnerability scanning, enhancing the depth of the assessment.

In this research, Nmap facilitated the identification of services that could be potential targets for exploitation and revealed misconfigured servers.[14][14].

4) Wireshark

Wireshark is a network protocol analyzer that captures and displays data packets traveling over a network. Its capabilities include:

a) Traffic Analysis: Wireshark allows detailed inspection of the transmitted data, essential for understanding how web applications communicate with users and servers.

b) Session Reconstruction: The tool can reconstruct sessions, making it easier to identify security weaknesses in communication between clients and servers.

By employing Wireshark, the research could monitor network traffic to identify unencrypted sensitive data, providing insights into potential vulnerabilities in data transmission.

5) Violent Monkey

Violent monkey is a user script manager that allows users to run JavaScript scripts on web pages. Its functionalities include:

a) Script Injection: Users can inject scripts that automate website actions, useful for testing web applications against various attack vectors.

b) Customization: Violentmonkey enables custom scripts to bypass client-side protections, making testing for XSS and other vulnerabilities easier.

This tool was particularly useful in automating attacks and testing how web applications handled unexpected input.

6) Cookie Editor

Cookie Editor is a browser extension that allows users to view, modify, and delete cookies stored in their browser. Key features include:

a) Session Hijacking: By editing cookies, users can simulate different user sessions, making it possible to test how well a web application protects against session fixation attacks.

b) Testing Authentication Flaws: The tool can manipulate session cookies to gain unauthorized access to user accounts.

Using Cookie Editor allowed for practical demonstrations of session management vulnerabilities, showcasing how they could be exploited.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Impact of VulnerabilitiesWeb application vulnerabilities can be highly important, and the consequences can involve one or many people, organizations, and the Internet society. Failure to address these risks results in monetary loss, a negative impact on reputation, and possible lawsuits.

Below are some of the most serious consequences that may result from these vulnerabilities:

With the increase in Data breaches and theft of sensitive information. They may also speak about SQL Injection or other types of security misconfiguration and exploitable access control, through which one may disclose the obligatory information. It can be customer information, financial records, or trade information, to name a few. User Accounts Highjacked: When exploiting vulnerabilities, they can also seize user accounts, which can be financial, corporate, email, or any other kind of sensitive account one may possess. Violation of Privacy: When personally identifiable Information (PII) is disclosed, it means a massive breach of the PII through action such as identity theft, prosecution, or violation of state or federal statutes (GDPR or HIPAA).

IV. RESULTS AND DISCUSSION

The analysis identified several critical vulnerabilities in web applications, highlighting significant security risks. The findings from the experiments include:

1) *Successful Exploitation Across Multiple Attack Vectors:* Both manual techniques and automated tools revealed that vulnerabilities could be easily exploited, allowing unauthorized access to sensitive areas such as databases and administrative panels.
2) *Inadequate Input Validation:* Many applications failed to properly sanitize user inputs, enabling attackers to inject malicious scripts and manipulate SQL queries. This lack of comprehensive validation was a significant contributor to security failures.
3) *Poor Access Control Mechanisms:* Weak access controls often allowed direct access to administrative sections by modifying URLs (e.g., appending "/admin"). This vulnerability exposed applications to unauthorized control.
4) *Weak Session Management and Tokenization:* Insecure session handling made it easy for attackers to hijack sessions and impersonate legitimate users, compromising user privacy and security.
5) *Misconfigured Security Settings:* Many applications had incorrect permissions, default configurations, and detailed error messages that exposed sensitive information and facilitated further exploitation.

These vulnerabilities emphasize the need for enhanced security measures. Key lessons drawn from the research include:

a) *Secure Coding Practices:* Implementing input sanitization, prepared statements, and role-based access control can significantly reduce vulnerabilities.
b) *Auditing and Security Testing:* Regular penetration testing and code reviews are essential for identifying and addressing weaknesses before they can be exploited.
c) *Effective Use of Web Application Firewalls (WAFs):* WAFs can help mitigate risks by blocking malicious HTTP requests.
d) *User Awareness:* Educating users about security best practices can reduce the risk of social engineering and session hijacking.
e) *Multi-Factor Authentication (MFA):* Implementing MFA, especially for administrative access, provides an additional layer of protection. Failure to address these vulnerabilities can lead to severe consequences, including data breaches, financial losses, reputational damage, regulatory penalties, and website downtime. The impact of these vulnerabilities extends beyond individual organizations, posing risks to users, stakeholders, and the broader Internet ecosystem. Addressing these issues is imperative to safeguard sensitive information and maintain user trust.

V. CONCLUSION

The exposures explored in this research include SQL Injection (SQLi), Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS), and inadequate control of access to the privileged regions of web applications. Manual testing included tools like Burp Suite, Nmap, and Wireshark, as well as a couple of browser extensions like Cookie Editor and ViolentMonkey, to find and take advantage of these holes to leak out private information and premium signed-in accounts. In general, we observed significant administrative vulnerabilities, as many sites can be opened using the URL with /admin after the URL. This underlines the importance of a vulnerability management plan, i.e., input validation, creating anti-CSRF tokens, and using Content Security Policies (CSP). Application areas contain two primary mechanisms whereby websites protect themselves: multiple authentication factors (MFA) and role-based access control (RBAC). Regarding the ethical perspective, during all tests, we behaved appropriately since test areas were virtually located and did not impact real-world systems. That is why ethical hacking is essential to maintaining the integrity of security systems, which is the article's main conclusion.

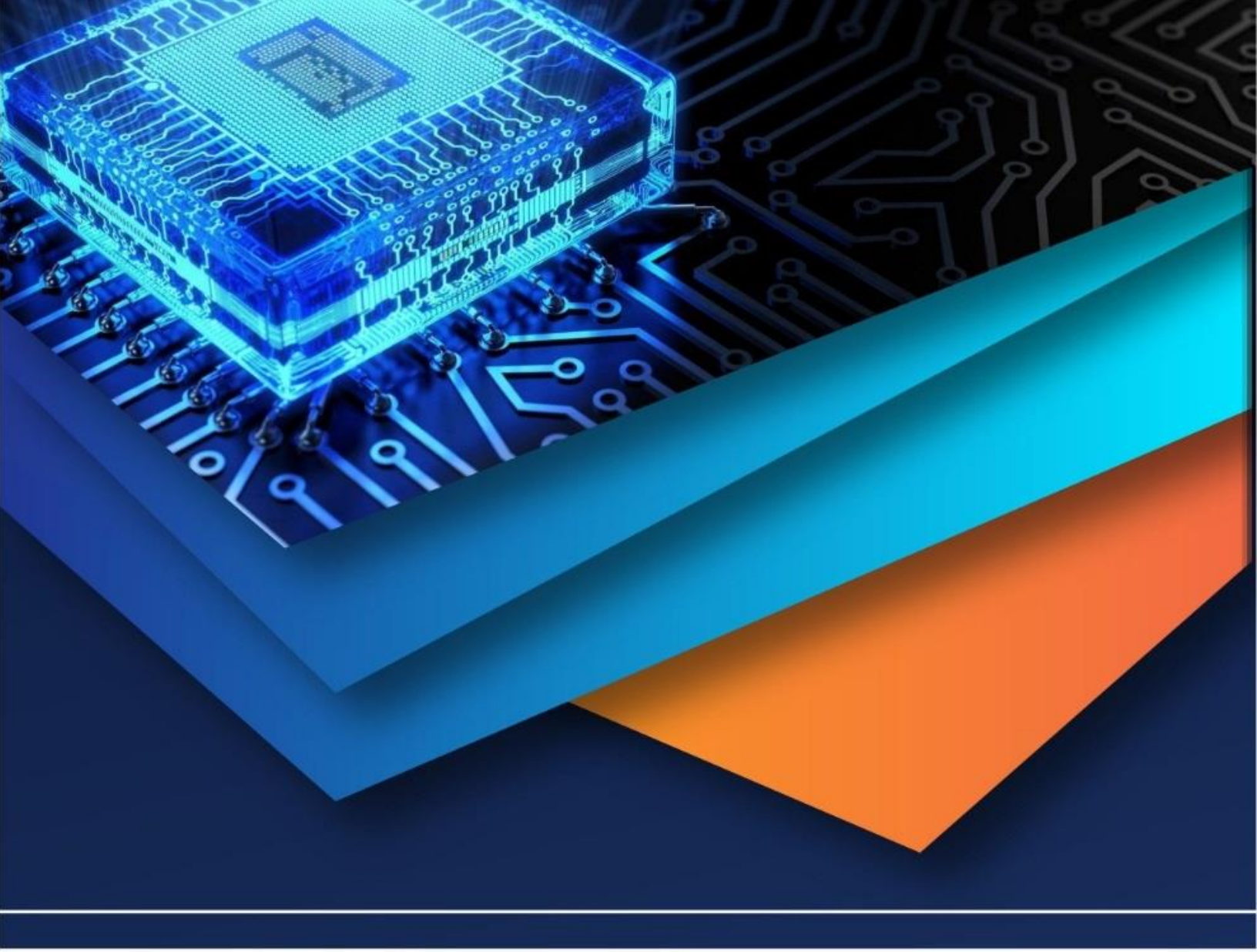
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Vulnerability management is not a stand-and-delivered process but a never-ending process that comprises sustainment, testing, and monitoring. Further study may identify other weaknesses in another setting, including cloud structures and emerging application programming interface (API), and discuss how to incorporate machine learning approaches to enhance the protection strategies. This paper shows that software developers should engage security professionals in advance to reach more effective methods to prevent future risks worldwide.

REFERENCES

[1] "What are web threats and online Internet threats?" Accessed: Oct. 13, 2024. [Online]. Available: https://www.kaspersky.com/resource-center/threats/web
[2] A. T. Sagayam, G. V Back, R. C. Marchany, D. R. Raymond, and K. Luther, "LIDS: An Extended LSTM Based Web Intrusion Detection System With Active and Distributed Learning," 2021.
[3] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," J Comput Syst Sci, vol. 80, no. 5, pp. 973–993, Aug. 2014, doi: 10.1016/J.JCSS.2014.02.005.
[4] Dinesh Gopal Dommeti and Persis Voola, "Identifying and Mitigating Common Web Application Vulnerabilities," South Asian Journal of Engineering and Technology, vol. 13, no. 2, pp. 1–10, Oct. 2023, doi: 10.26524/sajet.2023.13.9.
[5] M. Tabassum, A. W. Muzaffar, W. H. Butt, and S. A. Lashari, "Selecting third-party libraries: the web developers' perspective," International Journal of Software Engineering and Knowledge Engineering, Aug. 2024, doi: 10.1142/S0218194024500402.
[6] "(PDF) Formal Analysis of Vulnerabilities of Web Applications Based on SQL Injection (Extended Version)." Accessed: Oct. 13, 2024. [Online]. Available: https://www.researchgate.net/publication/301819223_Formal_Analysis_of_Vulnerabilities_of_Web_Applications_Based_on_SQL_Injection_Extended_Version
[7] J. H. B. Johny, W. A. F. B. Nordin, N. M. B. Lahapi, and Y. B. Leau, "SQL Injection Prevention in Web Application: A Review," Communications in Computer and Information Science, vol. 1487 CCIS, pp. 568–585, 2021, doi: 10.1007/978-981-16-8059-5_35.
[8] T. D. Kerr-Smith, S. S. Tirumala, and M. Andrews, "Assessing Web Application Security Through Vulnerabilities in Programming Languages and Environments," Te Pukenga, Jul. 2024, pp. 62–69. doi: 10.34074/proc.240109.
[9] "Web Application Attack: What Is It and How to Defend Against It?" Accessed: Oct. 13, 2024. [Online]. Available: https://www.acunetix.com/websitesecurity/web-application-attack/
[10] S. Rafique, M. Humayun, Z. Gul, A. Abbas, and H. Javed, "Systematic Review of Web Application Security Vulnerabilities Detection Methods," Journal of Computer and Communications, vol. 03, no. 09, pp. 28–40, 2015, doi: 10.4236/JCC.2015.39004.
[11] "Web Application Vulnerabilities and Best Practices: A Comprehensive Analysis – IJSREM." Accessed: Oct. 13, 2024. [Online]. Available: https://ijsrem.com/download/web-application-vulnerabilities-and-best-practices-a-comprehensive-analysis/
[12] A. Alquwayzani, R. Aldossri, and M. Frikha, "Mitigating Security Risks in Firewalls and Web Applications using Vulnerability Assessment and Penetration Testing (VAPT)," International Journal of Advanced Computer Science and Applications, vol. 15, no. 5, pp. 1348–1364, Jun. 2024, doi: 10.14569/IJACSA.2024.01505136.
[13] "10 Best Penetration Testing Tools: You Must Know - Software Testing Stuff." Accessed: Oct. 13, 2024. [Online]. Available: https://www.softwaretestingstuff.com/penetration-testing-tools
[14] "Understanding the Benefits of Using Tails Linux for Penetration Testing | Medium." Accessed: Oct. 13, 2024. [Online]. Available: https://medium.com/@techlatest.net/understanding-the-benefits-of-using-tails-linux-for-penetration-testing-f3f03e7dc10d

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)