Available online at https://www.gyanvihar.org/researchjournals/ctm_journals.php
SGVU International Journal of Convergence of Technology and Management
E-ISSN: 2455-7528
Vol.11 Issue 1 Page No 46-53

Unseen but Critical: The Role of Medium and Low Level Severities in Websites

[1*] Ravinder Singh, [2] Dr Sarang Maruti Patil, [3] Dr. Mukesh Kumar Gupta, [4] Dr. Dipak Raghunath Patil

[1] Department of Computer Engineering, Suresh Gyan Vihar University, Jaipur, India
[2] Department of Computer Engineering, SKN Sinhgad Institute of Technology & Science, Lonavala, India,
[3] Department of Electrical Engineering, Suresh Gyan Vihar University, Jaipur, India,
[4] Department of Computer Enginerring, AVCOE, Sangamner,India,

***Abstract*—**The security and functionality of websites are critical in the digital era, where vulnerabilities can lead to significant risks. While high-severity vulnerabilities often receive immediate attention, medium and low-level severities are frequently overlooked despite their cumulative impact on system integrity. This research focuses on analyzing the role of medium and low-level severities in websites using a Dynamic Application Security Testing (DAST) approach. The study leverages DAST to identify and evaluate vulnerabilities at these levels, providing a comprehensive understanding of their prevalence, characteristics, and potential exploitation risks. Our findings reveal that medium and low-level severities, while individually less impactful, can collectively compromise system security and degrade user experience. By addressing these vulnerabilities, organizations can achieve a more robust security posture and mitigate long-term risks. The study highlights the importance of adopting a holistic security strategy that prioritizes all levels of severity and provides actionable recommendations for integrating DAST into website development and maintenance workflows.

***Keywords*—**Web Application Security, Medium-Level Severity, Low-Level Severity, Vulnerability Assessment, Severity Classification, Cybersecurity Threat Levels, Security Testing Tools, OWASP Top 10

I. Introduction

In the evolving landscape of web applications, ensuring security has become a critical challenge. While high-severity vulnerabilities like XSS[1] and SQLi[2] often dominate the attention of developers and security experts, medium and low-level severities frequently remain underestimated. These vulnerabilities, though less catastrophic on the surface, can act as gateways to larger exploits, accumulate into significant risks over time, or degrade user trust and application functionality. This research seeks to highlight the overlooked yet crucial role of medium and low-level severities in website security, advocating for a more balanced approach to vulnerability assessment and management.

This paper explores the characteristics, impact, and mitigation strategies for medium and low-severity vulnerabilities within the DAST framework. By analyzing real-world case studies and statistical data, it sheds light on patterns and trends often neglected in traditional security practices. Additionally, the research advocates for enhanced prioritization strategies and comprehensive risk management frameworks that recognize the criticality of these "unseen" vulnerabilities.

By reexamining the importance of medium and low-level severities, this study aims to contribute to a more holistic understanding of web security, ultimately promoting resilient and trustworthy web applications.

1.1 Website:

A website is a digital platform consisting of interconnected web pages hosted on a server and accessible through the internet. Designed to deliver content, services, or functionality to users, websites can range from simple informational pages to complex, interactive systems like e-commerce platforms or social networks. Websites are built using technologies like HTML, CSS, and JavaScript, often enhanced by frameworks and backend systems. They serve various purposes, including education, entertainment, business, and communication. Accessed via browsers, websites are identified by unique domain names and URLs. Websites play an important role in the digital age, connecting users to information and services globally.

1.2 Website security:

Website security focuses on protecting a website and its data from unauthorized access, misuse, and malicious attacks. With the internet playing an ever-growing role in business, communication, and daily life, ensuring robust website security is essential to protect sensitive information, maintain user trust, and uphold business continuity.

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur
Corresponding author. E-mail addresses: singhravindersingh@gmail.com

Common threats to website security include hacking attempts, malware injection, command injection[3], data breaches, DDoS attacks, CSRF[4] and phishing schemes. Such threats have the potential to expose sensitive user information, hinder website operations, and harm an organization's reputation. With the rise of automated attack tools, even small websites are frequently targeted, emphasizing the need for proactive security measures.

Core aspects of website security include regular vulnerability assessments, secure coding practices, and the implementation of security protocols such as HTTPS to encrypt data transmission. WAFs, IDS, and DAST tools can help identify and mitigate risks such as DDOS[5]. Additionally, maintaining up-to-date software, strong password policies, and access controls are essential to reducing vulnerabilities.

Security best practices also involve preparing for the possibility of breaches through backup systems, incident response plans, and educating stakeholders on cybersecurity awareness. Organizations must adopt a multi-layered approach to website security, combining technological solutions with strategic policies.

By prioritizing website security, businesses can protect their digital assets, ensure regulatory compliance, and create a safe online environment for users, fostering long-term trust and success in an interconnected digital world.

(a) Confidentiality: Confidentiality refers to the practice of protecting sensitive information from unauthorized access, ensuring that only authorized individuals can view or use it. It is a fundamental principle in cybersecurity, business, and ethics, safeguarding data privacy and trust while preventing potential misuse, breaches, or exposure of personal, organizational, or classified information.

(b) Integrity: Integrity is the quality of being honest, ethical, and adhering to strong moral principles. It reflects consistency between actions, values, and beliefs, fostering trust and respect. Demonstrating integrity means doing the right thing, even when no one is watching, and maintaining accountability in personal and professional relationships.

(c) Availability: Availability is the capacity of a system, service, or resource to stay accessible and functional whenever required. In terms of cybersecurity, it ensures uninterrupted access to data and functionality, often maintained through redundancy, failover mechanisms, and proactive measures to prevent disruptions like hardware failures or cyberattacks.

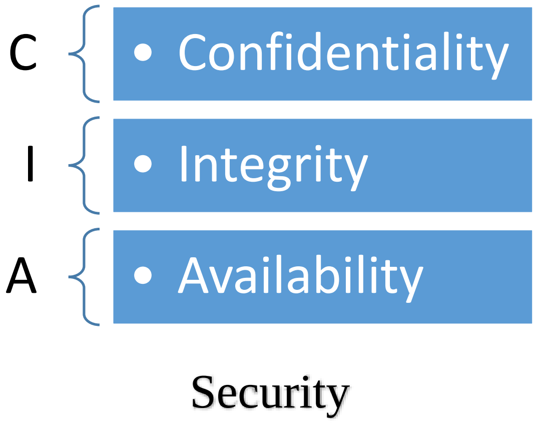

Fig:1 Security CIA Triad

As displayed in the Fig:1 Security CIA Triad, a foundational model in cybersecurity that represents three core principles: Confidentiality, Integrity and Availability, ensuring reliable access to data and systems when needed. This trio shapes the development and application of security policies, tools, and strategies aimed at safeguarding information and systems against potential threats while maintaining usability and trustworthiness. OWASP [6] is playing an crucial role in web application security by providing resources, best practices, tools, and guidelines to identify and mitigate vulnerabilities.

1.3 Website vulnerabilities

A website vulnerability refers to weaknesses or flaws in a website's code, configuration, or infrastructure that attackers can exploit to compromise security. The vulnerabilities results into unauthorized access, data theft, disruptions of services, and loss of user trust. Given the importance of websites in business operations and communication, ensuring their security and addressing vulnerabilities is essential for reliability.

Common website vulnerabilities include XSS, where a malicious script is injected by attackers to manipulate users; SQLi, where databases are compromised through malicious queries; and CSRF, which tricks users into executing unintended actions. Other issues like weak authentication, IDOR, and misconfigured servers further expose websites to potential exploits.

Vulnerabilities often arise from outdated software, poor coding practices, or improper security configurations. Tools like DAST and SAST help identify these weaknesses during development and deployment. Additionally, implementing measures such as secure coding practices, input validation, encryption, and multi-factor authentication enhances protection.

The consequences of exploiting website vulnerabilities can include data breaches, financial losses, reputational harm, and potential legal ramifications. As cyber threats evolve, proactive vulnerability management, regular security assessments, and adopting a multi-layered security approach are essential to mitigating risks. By addressing website vulnerabilities effectively, organizations can safeguard their digital assets, ensure user trust, and maintain a secure online presence.

1.4 Severity Levels of Website Vulnerabilities:

Severity levels in website vulnerabilities categorize the potential impact of a security flaw on the system. They range from low, indicating minimal risk with limited consequences, to informational, which highlights non-critical findings without immediate threat. These levels help prioritize remediation efforts by assessing the likelihood of exploitation and the effect on confidentiality, integrity, and availability, guiding developers and security teams in addressing vulnerabilities effectively and efficiently.

1.4.1 Critical Vulnerabilities

A critical vulnerability is a severe security loophole in a system, application, or network that can be easily exploited by attackers to cause significant damage. Such vulnerabilities often allow unauthorized access, data breaches, or full system

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur
Corresponding author. E-mail addresses: singhravindersingh@gmail.com

compromise with minimal effort or expertise. They pose a high risk to confidentiality, integrity, and availability, potentially leading to financial loss, reputational damage, or legal repercussions. Addressing critical vulnerabilities requires immediate action, including patching, updating, or mitigating the risk to prevent exploitation. Examples include remote code execution, privilege escalation, and unpatched zero-day vulnerabilities. Swift identification and remediation are vital to safeguard digital assets effectively.

1.4.2 High-Severity Vulnerabilities

A high vulnerability in website security refers to a flaw or weakness that poses a significant risk to the system's confidentiality, integrity, or availability. If exploited, it can result in severe consequences, such as unauthorized data access, system compromise, or operational disruptions. High vulnerabilities often require immediate attention and remediation to prevent exploitation by attackers. Examples include unpatched software, insecure authentication mechanisms, or exposed sensitive data. Addressing high vulnerabilities involves applying patches, implementing stronger security controls, and conducting regular assessments to minimize risks. Prompt action is required to maintain the security and functionality of affected systems or applications.

1.4.3 Medium-Severity Vulnerabilities

A medium vulnerability in website security represents a moderate-level threat that could potentially impact the system if exploited but typically requires specific conditions or user interaction to be effective. While it may not cause immediate critical damage, it can lead to unauthorized access, data exposure, or degraded functionality. Examples include missing security headers, insufficient input validation, or predictable session IDs. Medium vulnerabilities often serve as stepping stones for attackers when combined with other weaknesses. Addressing these vulnerabilities is essential to prevent their escalation and ensure the overall security of the system, balancing the focus between critical risks and lower-priority issues.

1.4.4 Low-Severity Vulnerabilities

A low vulnerability in website security refers to a minor weakness that poses minimal risk to the system's functionality or data integrity. These vulnerabilities are often harder to exploit or have limited impact even if successfully exploited. Examples include minor information disclosure, outdated software versions without known exploits, or cosmetic issues in error messages. While they may not require immediate attention, addressing low vulnerabilities is essential to maintain a robust security posture and prevent potential escalation when combined with other vulnerabilities. Regular monitoring and routine updates ensure that low vulnerabilities do not evolve into more significant security risks over time.

Thus understanding different level vulnerabilities is important to ensure a comprehensive security posture. While the medium and low severity vulnerabilities may seem insignificant compared to critical issues, these vulnerabilities can act as entry points for attackers or become significant when combined with other exploits. Addressing them proactively reduces the attack surface, enhances system resilience, and prevents small issues from escalating into larger threats. Addressing these vulnerabilities showcases a dedication to robust security measures and fosters user confidence and ensures compliance with industry standards. By prioritizing all severity levels, organizations can achieve more robust protection and minimize the risks.

II. Related Work

Aslan et. al [7] explore the primary causes of cyberattacks. Then author examines recent incidents, attack trends, and detection strategies. Additionally, the research highlights modern technical and non-technical approaches for early attack recognition. Leveraging emerging technologies like ML, DL, computing based on cloud, big data, and blockchain offers significant potential to address current and future cyber threats. These innovations can aid in tasks such as malware detection, intrusion prevention, spam filtering, DNS attack categorization, fraud detection, uncovering covert communication channels, and identifying advanced persistent threats. Nonetheless, certain approaches, particularly ML and DL, remain vulnerable to evasion tactics, which should be carefully addressed when designing countermeasures against sophisticated cyberattacks.

Vulnerability Assessment and Penetration Testing (VAPT) can be an effective method for preventing cyberattacks. This research highlights how VAPT can be utilized to actively mitigate potential threats by identifying vulnerabilities in systems or networks and implementing methodologies to address them. The study outlines the complete lifecycle of VAPT, detailing proactive actions taken to resolve vulnerabilities and prevent potential attacks. Additionally, it reviews commonly used VA techniques and examines popular paid as well as open-source VAPT tools. The paper provides a comprehensive guide on leveraging VAPT as a robust approach to cybersecurity and threat prevention. attacks. [8]

The study introduced an automated penetration testing method leveraging advanced machine learning techniques, specifically deep reinforcement learning. To achieve this, the researchers analyzed the Shadov system's capabilities in gathering factual data for constructing attack trees and utilized the Mulval platform for generating these trees. A novel approach was devised to create a cyber intrusion matrix using Mulval, and the Deep Q-Learning Network method was enhanced to analyze the matrix and determine the most effective attack path. In this process, reward scores, based on the CVSS rating, were assigned to each node, enabling the reduction of attack tree complexity and the identification of high-probability attack scenarios. A comparative evaluation of the proposed method demonstrated its effectiveness, highlighting its practical application in enhancing computer system security. [9]

Hu et. al [10] introduces a two-stage approach to automated penetration testing. In the initial phase, the Shodan search engine is employed to collect pertinent server information, which is subsequently utilized to build an accurate network topology. The Multi-host, Multi-stage Vulnerability Analysis (MulVAL) framework is utilized to construct an attack tree for the given topology. Conventional search algorithms are then employed to identify all possible attack paths, which are subsequently represented as a matrix compatible with deep reinforcement learning techniques. In the second stage, the Deep Q-Learning Network (DQN) method is employed to identify the most readily exploitable attack path from the available candidates. The method was evaluated across thousands of input scenarios, achieving an accuracy of 0.86 in

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur
Corresponding author. E-mail addresses: singhravindersingh@gmail.com

optimal path discovery and delivering valid solutions in additional cases.

III. METHODOLOGY

The Dynamic Application Security Testing (DAST) is a security testing method focused on identifying vulnerabilities in running applications. It evaluates an application's security by simulating real-world attack scenarios, analyzing its behavior and responses without requiring access to the source code.

DAST tools (such as HCL Appscan, Acunetix, OWASP ZAP or Burp Suite) interact with the application through its front-end, APIs, or web interfaces, scanning for issues like SQLi, XSS, authentication flaws, and misconfigurations. This approach helps uncover vulnerabilities that manifest during runtime, including those related to business logic and server configurations.

Because DAST tests applications in a live environment, it provides insights into actual security risks that could be exploited by attackers. It is widely used in development pipelines, especially in the later stages, to enhance application security. DAST complements an organization's overall security strategy by offering a practical understanding of how applications perform under potential threats. The Fig 2: DAST Methodology, describe the common functions performed during black-box testing of a web application:

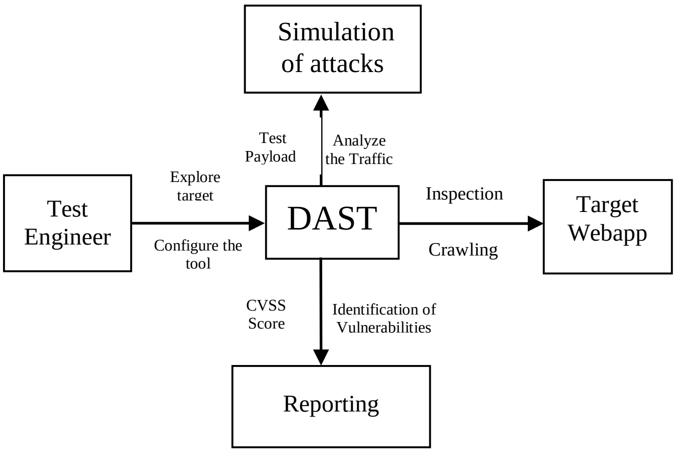

Fig: 2 DAST Methodology

1. Explore Target Website: Exploring means identifying the structure and functionalities of the target application. This includes mapping endpoints, user inputs, forms, and APIs. By analyzing these elements, DAST tools create a blueprint of the application to guide subsequent vulnerability detection and testing efforts.

2. Configuration the tool: Setting up the testing tool to align with the application's parameters is called configuration of the tool. This includes specifying the target URL, authentication credentials, scan scope, and exclusions. Proper configuration ensures accurate detection of vulnerabilities, minimizes false positives, and tailors the tool to the application's unique structure and requirements.

3. Inspection and Crawling: The step involves exploring the application to identify its structure, endpoints, and functionalities. The DAST tool mimics a user by navigating the application, mapping pages, forms, and APIs. This step ensures comprehensive coverage, enabling the tool to identify potential vulnerabilities across the application's accessible components.

4. Simulation of attacks: Simulation involves executing controlled security attacks on a live application to mimic real-world threat scenarios. This process tests for vulnerabilities such as SQLi, XSS, and authentication flaws. By analyzing the application’s responses, DAST tools identify security weaknesses and potential exploitation points.

5. Analyze the Traffic and test payload: This step involves monitoring application traffic and injecting crafted payloads to identify vulnerabilities. It analyzes requests and responses to detect anomalies, such as unexpected errors or improper handling of inputs, helping uncover issues like injection flaws, authentication weaknesses, and data exposure during runtime.

6. Identification of Vulnerability and CVSS Score: The tool detects security flaws in the application during runtime, such as injection attacks or authentication issues. Each identified vulnerability is assigned a score based on the CVSS, helping prioritize risks. The risk value depend on Impact and the severity.

7. Reporting: Reporting means record the vulnerabilities details with severity and recommended remediations. It provides detailed insights into the security issues found during testing, helping development and security teams understand the risks involve so that they can prioritize fixes to improve the application's overall security.

This study analyzes the HCL Appscan tool generated DAST reports for various web applications over the past two and a half years. The findings provide valuable insights for cybersecurity experts, researchers, and organizations, highlighting common vulnerabilities. These findings assist in recognizing and addressing security concerns, prioritizing potential risks, ensuring regulatory compliance, promoting a culture of security awareness, and strengthening the overall security framework of web applications within organizations.

IV. RESULTS AND DISCUSSION

This paper conducts a detailed examination of security weaknesses across roughly 15 user login systems, evaluating their impact and offering strategies for mitigation. The analysis draws on data gathered over the past three years. The term "Common Vulnerability Scoring System (CVSS) score" quantifies the overall security risk associated with a vulnerability by incorporating metrics from three primary categories: Base, Temporal, and Environmental. The score is calculated based on available data from one or more of these categories. As outlined by the National Vulnerability Database (NVD) on the National Institute of Standards and Technology (NIST) website, the rating criteria for CVSS Version 3.0, the critical vulnerability is defined

Vulnerability Severity	CVSS Score
Critical	9.0-10.0
High	7.0-8.9
Medium	4.0-6.9
Low	0.1-3.9

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur Corresponding author. E-mail addresses: singhravindersingh@gmail.com

None	0.0

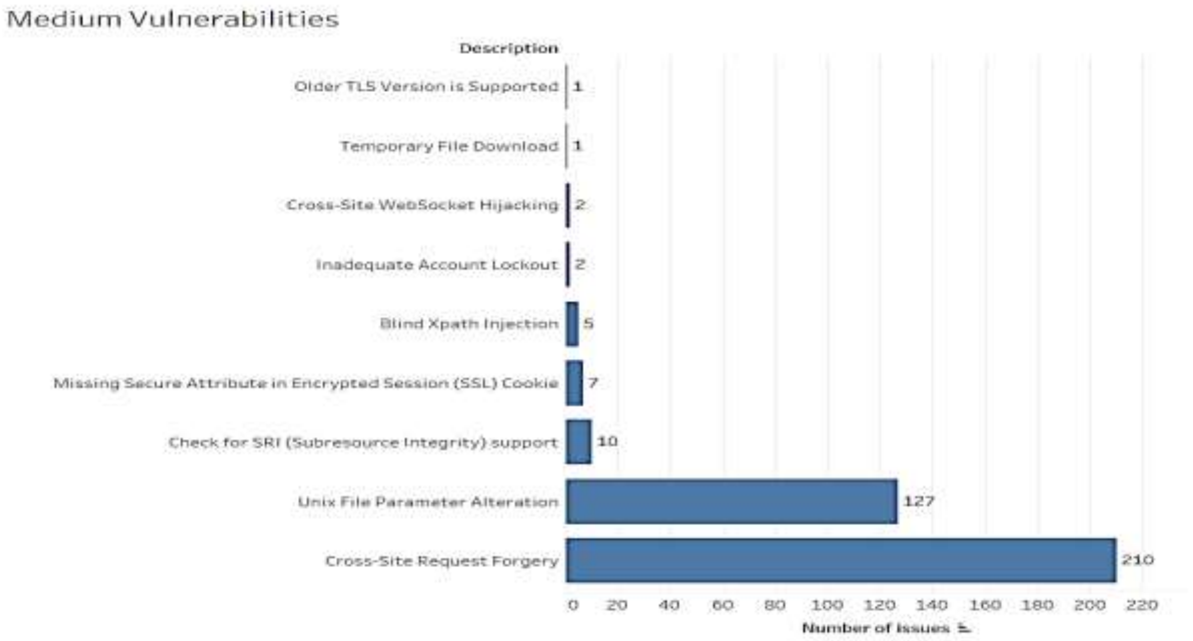

Fig 3: Medium Security Vulnerabilities

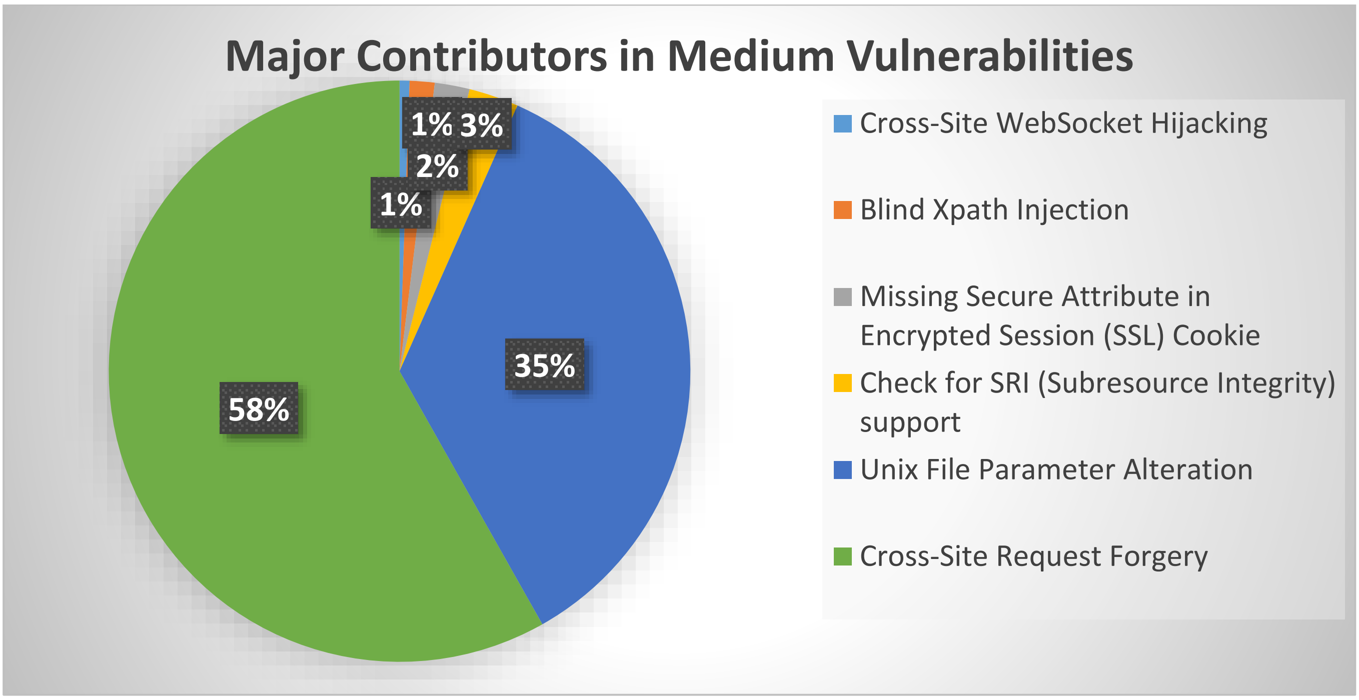

Fig 4: Major Contributors in Medium Vulnerabilities

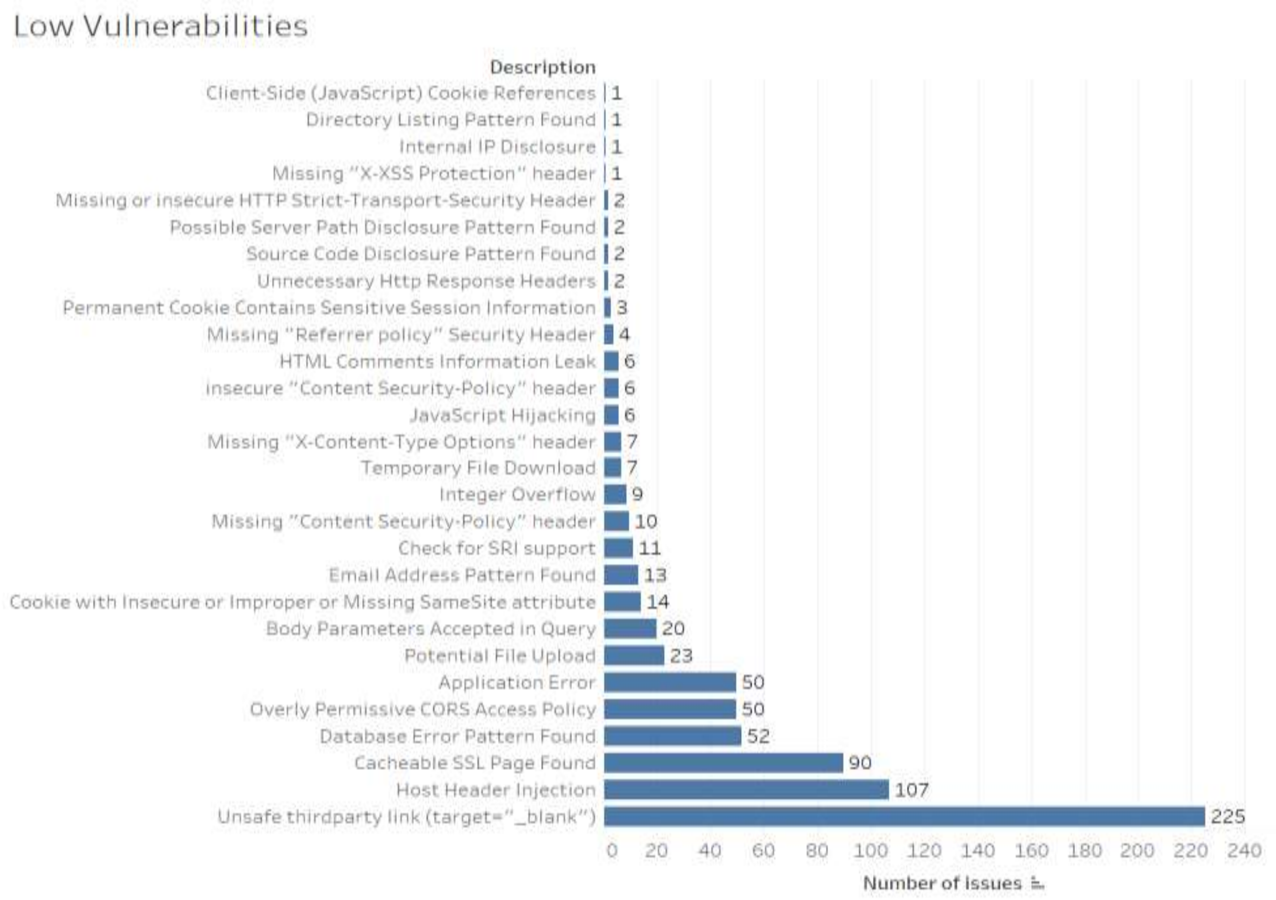

Fig 5: Low Security Vulnerabilities

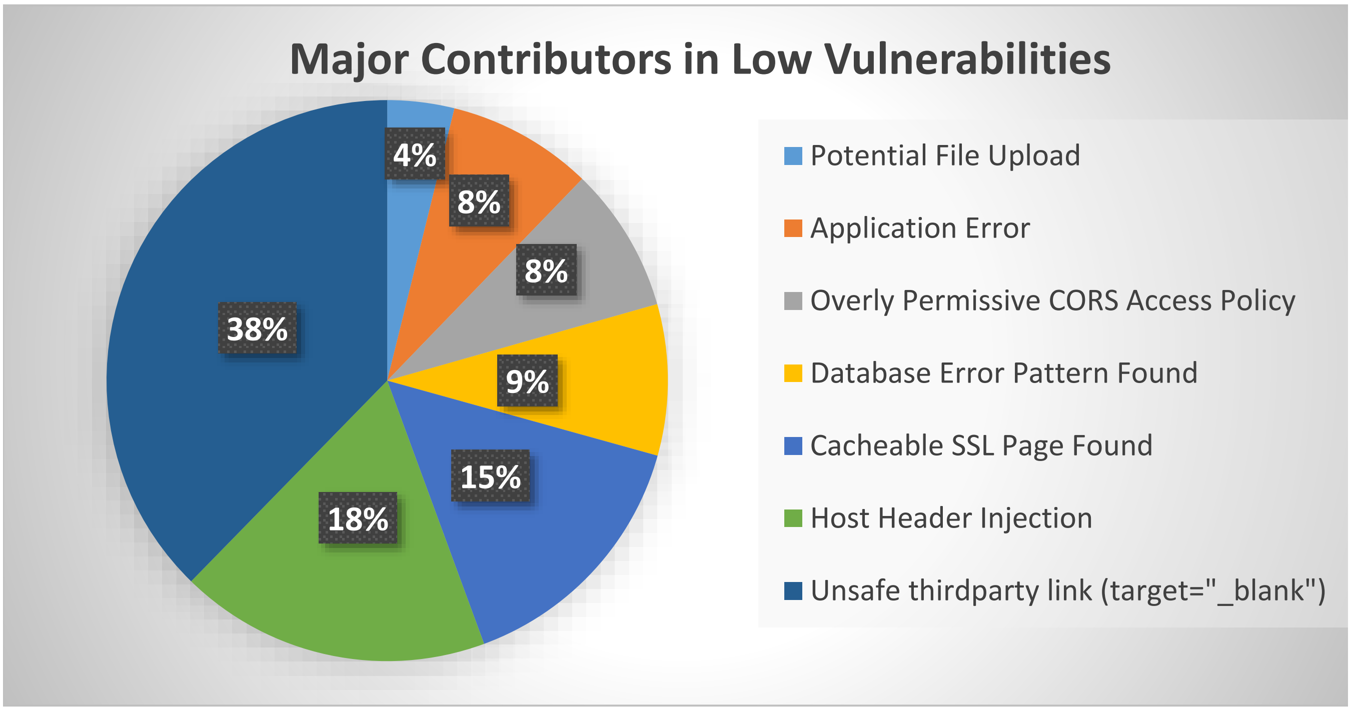

Fig 6: Major Contributors in Low Vulnerabilities

A medium vulnerability is a security flaw that, while not immediately critical, poses a moderate risk if exploited. It often requires specific conditions or some user interaction to be effective. Medium vulnerabilities can result into unauthorized access, data exposure or ddisruptions in service offered and should be addressed promptly to mitigate potential threats. The Fig 3: Medium Severity Vulnerability is showing what are the different issues observed under medium vulnerability category. In our research it is observed that ‘210’ issues of “Cross Site Request Forgery” are observed, then ‘127’ issue of “Unix File Parameter Alteration”, 10 issues of “Check for SRI” and ‘7’ issues of “Missing Secure Attribute in Encrypted SSL Cookie” are there of Medium Severity Vulnerabilities. The CVSS score is in between 4.0 to 6.9 for the said vulnerability and Fig 4: Major Contributors in Medium Vulnerabilities, indicating that the ‘Cross Site Request Forgery’ have 58% weightage, Unix File Parameter Alteration is contributing 35% and Check for SRI have 3% weightage.

Similarly, Low vulnerability refers to a state where individuals, systems, or organizations have minimal exposure to risks or threats. This is achieved through robust safeguards, adaptive strategies, and resilient frameworks, reducing susceptibility to harm. It enhances stability, safety, and confidence, fostering an environment capable of withstanding challenges and maintaining functionality under pressure.

It is crucial to address these vulnerabilities to prevent their escalation into more significant threats. As per the fig 5: Low Severity Vulnerabilities, 225 issue of ‘Unsafe third party link’ is observed, then 107 issues of “Host Header Injection”, 90 issues of “Cacheable SSL Page found”, 52 issues of “Database Error Patten found” are observed. The CVSS score is in between 0.1 to 3.9 for the Low vulnerability and fig 6: Major Contributors in Low Vulnerabilities, presenting that the ‘Unsafe third party link’ have 38% weightage, Host Header Injection is contributing 18% Cacheable SSL Page found have 15% weightage in overall count.

The CVSS score serves as the determining factor for categorizing vulnerabilities into High, Medium, or Low severity levels. Consequently, Table 2: Common Vulnerabilities with CVSS score for Medium Vulnerabilities, and Table 3: Common Vulnerabilities with CVSS score of Low vulnerabilities providing details on those issues with high CVSS scores and those with low CVSS scores.

Sl. No.	**Vulnerability**	**CVSS Score**
1.	Blind Xpath Injection	6.5
2.	Unix File Parameter Alteration	6.5
3.	Cross-Site Request Forgery	6.5
4.	Inadequate Account Lockout	6.4
5.	Temporary File Download	5.3
6.	Older TLS Version is Supported	5.3

Table 2: Common Vulnerabilities with CVSS score of Medium vulnerabilities

Sl. No.	**Vulnerability**	**CVSS Score**
1.	Cacheable SSL Page Found	3.7
2.	Overly Permissive CORS Access Policy	3.7
3.	Missing or insecure HTTP Strict-Transport-Security Header	3.7
4.	Permanent Cookie Contains Sensitive Session Information	3.1

Table 3: Common Vulnerabilities with CVSS score of Low vulnerabilities

V. CONCLUSION

The role of medium and low-severity vulnerabilities in website security is often overlooked, yet they can have significant implications when left unaddressed. This paper emphasizes the crucial need to identify, assess, and address vulnerabilities across all severity levels to maintain a robust security posture. While high-severity vulnerabilities are prioritized due to their immediate and catastrophic impact, medium and low-severity issues often serve as enablers for more sophisticated attacks, acting as entry points or escalating existing risks.

Our analysis demonstrates that these vulnerabilities, though perceived as less critical, can compromise sensitive data, degrade user trust, and provide attackers with opportunities to chain exploits. In particular, medium-severity vulnerabilities often possess a broader attack surface, while low-severity issues frequently remain unresolved due to resource constraints or underestimation of their potential impact. Together, they contribute to a cumulative risk that may rival or surpass the threat posed by a single high-severity vulnerability.

In conclusion, acknowledging the role of medium and low-level severities in website security is essential. Proactive mitigation strategies not only decrease the associated risk but also enhance the resilience and integrity of web applications in an increasingly complex.

REFERENCES

[1] Song, X.; Zhang, R.; Dong,Q.; Cui, B. Grey-Box Fuzzing Based on Reinforcement Learning for XSS Vulnerabilities. Appl. Sci. 2023, 13, 2482. https://doi.org/10.3390/app13042482

[2] Vivek Thoutam, “SQL Injection Vulnerabilities Prevention through ML IPAAS Architecture”, 2022 IJNRD | Volume 7, Issue 3 March 2022 | ISSN: 2456-4184 | IJNRD.ORG

[3] Akinmerese, O., Fasanya, S., Aderotoye, D., Adingupu, N., Ezeoke, E., Muritala, R., Lawal, O., Akingbade, B. and Ifekandu, C. (2024) Defence against Command Injection Attacks in a Distributed Network Environment. Open Access Library Journal, 11: e11491. https://doi.org/10.4236/oalib.1111491

[4] Shobhit Agrawal, “Mitigating Cross-Site Request Forgery (CSRF) Attacks Using Reinforcement Learning and Predictive Analytics”,

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur
Corresponding author. E-mail addresses: singhravindersingh@gmail.com

Applied Research in Artificial Intelligence and Cloud Computing, 2023

[5] Alashhab, A.A.; Zahid, M.S.M.; Azim, M.A.; Daha, M.Y.; Isyaku, B.; Ali, S. A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks. Symmetry 2022, 14, 1563. https://doi.org/10.3390/sym14081563

[6] OWASP: Available at https://owasp.org/www-project-top-ten. OWASP Top Ten Project, 2021

[7] Aslan, Ö.; Aktu˘g, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. Electronics 2023, 12, 1333. https://doi.org/10.3390/electronics12061333.

[8] Dr. Vinod Varma Vegesna, "Utilising VAPT Technologies (Vulnerability Assessment & Penetration Testing) as a Method for Actively Preventing Cyberattacks", International Journal of Management, Technology And Engineering, ISSN NO : 2249-7455, Volume XII, Issue VII, July 2022

[9] Semenov, S., Weilin, C., Zhang, L., & Bulba, S. (2021). Automated Penetration Testing Method Using Deep Machine Learning Technology. Advanced Information Systems, 5(3), 119–127. https://doi.org/10.20998/2522-9052.2021.3.16

[10] Hu, Z., Beuran, R., & Tan, Y. (2020). Automated Penetration Testing Using Deep Reinforcement Learning. 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). doi:10.1109/eurospw51379.2020.00010

Correspondence to: Ravinder Singh, Department of Computer Science and Information Technology, Suresh Gyan Vihar University, Jaipur
Corresponding author. E-mail addresses: singhravindersingh@gmail.com