Mitigating Security Risks in Firewalls and Web Applications using Vulnerability Assessment and Penetration Testing (VAPT)

Alanoud Alquwayzani, Rawabi Aldossri, Mounir Frikha
Dept. of Computer Networks and Communications, CCSIT,
King Faisal University, Al Hassa 31982, Saudi Arabia

***Abstract*—In today's digital age, both organizations and individuals heavily depend on web applications for a wide range of activities. However, this reliance on the web also opens up opportunities for attackers to exploit security weaknesses present in these applications. Web Application Firewalls (WAFs) are typically the first line of defense, protecting web apps by filtering and monitoring HTTP traffic. However, if these firewalls are not properly configured, they can be bypassed or compromised by attackers. The escalating number of attacks targeting web applications underscores the urgent need to enhance their security. This paper offers an in-depth review of existing research on web application Vulnerability Assessment and Penetration Testing (VAPT). Our unique contribution lies in the comprehensive synthesis and categorization of VAPT tools based on their optimal use cases, which provides a practical guide for selecting the appropriate tools for specific scenarios. Additionally, this study integrates emerging technologies such as artificial intelligence and machine learning into the VAPT framework, addressing the evolving nature of cyber threats. The paper also identifies common challenges encountered during the VAPT process and proposes actionable recommendations to overcome these obstacles. Furthermore, it discusses best practices such as secure coding practices and defense-in-depth strategies to improve the effectiveness and efficiency of VAPT efforts. By offering these insights, this paper aims to advance the current understanding and application of VAPT in enhancing the security of web applications and firewalls.**

Keywords*—*Web Application Firewalls (WAFs); Vulnerability Assessment and Penetration Testing (VAPT); cybersecurity; security vulnerabilities; security misconfigurations; network scanning tools; vulnerability detection

I. Introduction

In recent decades, websites and web applications become increasingly integrated into our daily lives. These platforms enable us to perform a wide range of activities, from online shopping and consuming news to social communication and beyond. A study by Siteefy shows that over 200 million websites are active on the internet as of the end of 2022 [1].

AAs our reliance on these platforms grows, attackers perceive this trend as an opportunity for monetary gain and other malicious intents. The increased dependence on web applications generates vast amounts of data, crucial for creating excellent user experiences [2]. However, while this data is beneficial for various purposes, it also presents significant risks if not adequately protected.

Firewalls, serving as the first line of defense in most digital systems, often become primary targets of cyber-attacks. Ensuring their security is therefore crucial. Recent studies reveal that 73% of corporate sector breaches are primarily due to vulnerabilities in their web applications [3]. Such statistics underscore the urgent need to protect web applications from attacks.

Identifying the vulnerabilities that attackers can exploit is the first step to safeguarding firewalls and web applications. Penetration testing and vulnerability assessments are reliable methods for detecting these vulnerabilities, thereby enabling security teams to enhance the security of these platforms. Vulnerability Assessment and Penetration Testing (VAPT) allows businesses to assess their cybersecurity posture, identify vulnerabilities, and take necessary steps to address them before attackers can exploit them. By implementing these proactive measures, businesses can protect themselves from attacks and avoid the costs associated with cyberattacks.

The novel contribution of this study lies in its comprehensive review and synthesis of VAPT tools and techniques, offering a unique categorization based on optimal use cases. Unlike previous studies, this paper not only reviews existing VAPT tools but also integrates best practices and emerging technologies, such as AI and machine learning, into the VAPT framework. This integration addresses the evolving nature of cyber threats and provides a forward-looking approach to cybersecurity.

Additionally, this paper identifies and analyzes common challenges in VAPT processes, providing actionable recommendations to overcome these challenges. The study also proposes a novel framework for continuous VAPT implementation, emphasizing the importance of an iterative and adaptive approach to cybersecurity.

By highlighting these unique aspects, this paper aims to advance the current understanding and application of VAPT, offering practical insights and strategies for enhancing the security of web applications and firewalls.

II. Methodology

This section details the methodological framework used to conduct the research, including the preparation of the research environment, data collection, data analysis, and validation of results.

A. Preparation of the Research Environment

To ensure a thorough and systematic review, the following steps were undertaken to prepare the research environment:

- Literature Sources: Robust academic databases such as Google Scholar and IEEE Xplore were utilized to gather relevant studies. The search focused on studies published in English from 2012 to 2024.
- Search Keywords: Keywords included ”firewall security”, ”web application vulnerabilities”, ”VAPT”, ”security risk mitigation”, and ”penetration testing techniques”.
- Selection Criteria: Studies were included based on their focus on VAPT techniques, tools, and vulnerabilities specific to web applications and firewalls. Studies that did not meet these criteria were excluded.

B. Data Collection

The data collection process involved multiple stages to ensure the comprehensiveness and relevance of the data:

- Initial Search: An initial search was conducted using the specified keywords, returning a broad selection of publications.
- Screening: Titles and abstracts of the retrieved studies were screened to remove irrelevant or redundant entries.
- Full-Text Review: The remaining studies were reviewed in full to ensure they met the inclusion criteria. This included assessing each paper’s contribution to knowledge, methodological robustness, and relevance to the research questions.
- Final Selection: A total of 30 papers were selected for comprehensive review, consisting of 21 seminal works from Google Scholar and 9 technical papers from IEEE.

C. Dataset Description

To evaluate the effectiveness of VAPT tools and techniques, several datasets were utilized, including real-world web applications and simulated environments:

- Real-World Web Applications: These included a variety of open-source web applications with known vulnerabilities. Examples include:
  - *OWASP Juice Shop*: A modern web application intentionally designed to be insecure.
  - *DVWA (Damn Vulnerable Web Application)*: A PHP/MySQL web application that is damn vulnerable.
- Simulated Environments: Virtual machines running different operating systems (Windows, Linux) with pre-configured vulnerable services and applications.
- Custom Test Bed: A custom test bed was created to simulate various attack scenarios and measure the effectiveness of VAPT tools. This included:
  - Firewalls configured with different rulesets to simulate real-world scenarios.
  - Web servers hosting applications with diverse vulnerability profiles.

D. Data Analysis

The selected studies and datasets were analyzed to identify common themes, methodologies, and findings related to VAPT in the context of firewalls and web applications:

- Qualitative Analysis: The content of each paper was qualitatively analyzed to extract key insights and findings relevant to the research objectives.
- Comparative Analysis: The methodologies and findings of different studies were compared to identify trends, common practices, and gaps in the existing literature.

E. Validation of Results

To ensure the validity and reliability of the findings, the following validation methods were employed:

- Triangulation: Data from multiple sources were cross-verified to ensure consistency and accuracy.
- Expert Review: The findings were reviewed by experts in the field of cybersecurity to validate the interpretations and conclusions.
- Reproducibility Check: The research process was documented in detail to allow other researchers to replicate the study and verify the results.

By following this structured methodological framework, the research aimed to provide a comprehensive and reliable assessment of the effectiveness of VAPT in mitigating security risks in firewalls and web applications.

F. Data Collection

The data collection process involved multiple stages to ensure the comprehensiveness and relevance of the data:

- Initial Search: An initial search was conducted using the specified keywords, returning a broad selection of publications.
- Screening: Titles and abstracts of the retrieved studies were screened to remove irrelevant or redundant entries.
- Full-Text Review: The remaining studies were reviewed in full to ensure they met the inclusion criteria. This included assessing each paper’s contribution to knowledge, methodological robustness, and relevance to the research questions.
- Final Selection: A total of 30 papers were selected for comprehensive review, consisting of 21 seminal works from Google Scholar and 9 technical papers from IEEE.

G. Data Analysis

The selected studies were analyzed to identify common themes, methodologies, and findings related to VAPT in the context of firewalls and web applications:

- Qualitative Analysis: The content of each paper was qualitatively analyzed to extract key insights and findings relevant to the research objectives.
- Comparative Analysis: The methodologies and findings of different studies were compared to identify trends, common practices, and gaps in the existing literature.

H. Validation of Results

To ensure the validity and reliability of the findings, the following validation methods were employed:

- Triangulation: Data from multiple sources were cross-verified to ensure consistency and accuracy.
- Expert Review: The findings were reviewed by experts in the field of cybersecurity to validate the interpretations and conclusions.
- Reproducibility Check: The research process was documented in detail to allow other researchers to replicate the study and verify the results.

By following this structured methodological framework, the research aimed to provide a comprehensive and reliable assessment of the effectiveness of VAPT in mitigating security risks in firewalls and web applications.

III. Selection of Papers by PRISMA

In conducting a systematic literature review (SLR) on mitigating security risks within firewalls and web applications through Vulnerability Assessment and Penetration Testing (VAPT), we meticulously followed the PRISMA framework to identify and select pertinent studies from a comprehensive body of literature. Utilizing the robust platforms of Google Scholar and IEEE, we initiated our search with a tailored set of keywords: ”firewall security”, ”web application vulnerabilities”, ”VAPT”, ”security risk mitigation”, and ”penetration testing techniques”. Our query was confined to studies published in English from 2012 to 2024, enabling us to encompass a span of advancements reflective of both foundational and cutting-edge research in the field.

The initial query on Google Scholar returned a broad selection of publications. After an initial screening to remove redundant entries, we extracted those studies that were closely aligned with the theme of ’Mitigating Security Risks in Firewalls and Web Applications Using VAPT.’ Through careful examination of titles, abstracts, and where necessary, full texts, we evaluated each paper’s contribution to knowledge, the robustness of its methodological framework, and direct relevance to our research questions. This led to the selection of 21 seminal works from Google Scholar.

Parallel to our efforts on Google Scholar, a targeted search on the IEEE digital library with the same keywords brought forth a collection of technical papers and conference proceedings. Adhering to the same stringent selection criteria, we sifted through this array to handpick nine studies that provided significant insights into VAPT’s role in enhancing cybersecurity measures in firewalls and web applications.

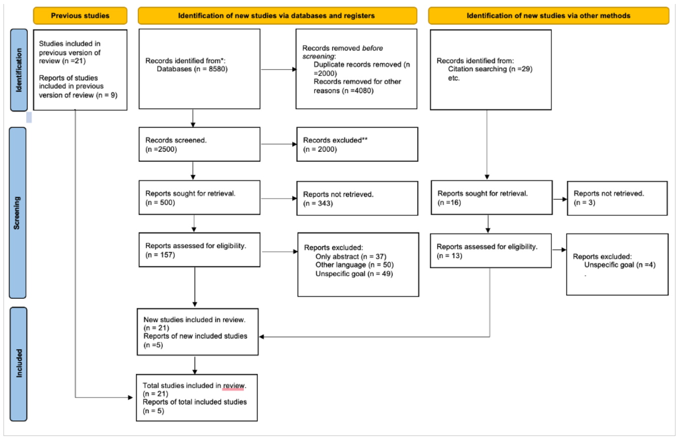
Previous studies
Identification of new studies via databases and registers
Identification of new studies via other methods
Identification
Screening
Included
Studies included in previous version of review (n =21)
Reports of studies included in previous version of review (n = 9)
Records identified from*:
Databases (n = 8580)
Records removed *before screening*:
Duplicate records removed (n =2000)
Records removed for other reasons (n =4080)
Records identified from:
Citation searching (n =29)
etc.
Records screened.
(n =2500)
Records excluded**
(n = 2000)
Reports sought for retrieval.
(n = 500)
Reports not retrieved.
(n = 343)
Reports sought for retrieval.
(n =16)
Reports not retrieved.
(n = 3)
Reports assessed for eligibility.
(n = 157)
Reports excluded:
Only abstract (n = 37)
Other language (n = 50)
Unspecific goal (n = 49)
Reports assessed for eligibility.
(n = 13)
Reports excluded:
Unspecific goal (n =4)
New studies included in review.
(n = 21)
Reports of new included studies
(n =5)
Total studies included in review.
(n = 21)
Reports of total included studies
(n = 5)

Fig. 1. The selection of papers for the literature review using PRISMA.

Our exacting selection process, conforming to PRISMA guidelines, has culminated in a curated list of 30 papers. These papers collectively offer a comprehensive understanding of the challenges, methodologies, and strategies in employing VAPT to fortify cybersecurity defenses. This assortment ensures a breadth of perspective and upholds the standard of a systematic and unbiased review, essential for a scholarly inquiry into such a specialized and evolving aspect of cybersecurity. The PRISMA flow diagram, which will be featured in our review, details each step of our rigorous paper selection process.

The methodology used in this paper is based on the four stages of the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) approach as shown in Fig. 1. Here’s a detailed explanation of what is done at each stage:

1) Identification: In this stage, a comprehensive search for relevant papers was conducted on platforms such as Google Scholar and MDPI. The search was guided by specific inclusion and exclusion criteria to ensure that only the most relevant papers were considered.
2) Screening: After the identification stage, the papers were screened based on their titles and abstracts. All the papers that did not have the relevant information we need for this assessment were not included in the detailed review.
3) Eligibility: The full texts of the remaining papers were then assessed for eligibility. This involved a more in-depth review to determine whether each paper’s content was truly relevant to our research.
4) Included: The final stage involved the inclusion of papers that met all the criteria. These papers were then analyzed and synthesized to answer the research questions.

For this particular research, the focus was on papers discussing vulnerability assessment and penetration testing techniques, tools, and common vulnerabilities facing web apps and firewalls. The time frame for the papers considered was from January 2008 to January 2024. A total of 12 papers that met our criteria for inclusion were reviewed and analyzed.

IV. Literature Review

Lamba [4] explored the importance of VAPT as a proactive measure in identifying and mitigating vulnerabilities to enhance system security. His research elaborates on the VAPT process as a comprehensive nine-step life cycle, including scoping, reconnaissance, vulnerability assessment, penetration testing, result analysis, and cleanup. Each step is crucial for effectively identifying and addressing vulnerabilities within systems. The paper also discusses various techniques for vulnerability assessment and penetration testing, including static analysis, manual testing, automated testing, fuzz testing, and different types of box testing. Furthermore, it highlights the significance of VAPT tools in streamlining the assessment and exploitation of vulnerabilities. The paper lists the top 15 VAPT tools which include Juice Shop, NodeGoat, Arachni, OWASP ZAP (Zed Attack Proxy), WAVS Framework, Prototype-based Model, V model, Classical waterfall model, Iterative waterfall model, React (for front-end), Node.js with Express (for backend), Group Results by CWE ID, Union List, Intersection List, and Automation Algorithm.

Ahmad et al. [5] conducted a study on the Vulnerability Assessment and Penetration Testing (VAPT) Framework, focusing on the case study of a government website. In this research, VAPT is highlighted as a technique to analyze the strengths and weaknesses of computer systems to ensure the implementation of security measures. The study emphasizes the role of SQL in web operations and the risks associated with vulnerabilities such as SQL injection and Cross-Site Scripting (XSS). A goal-oriented penetration testing framework is recommended to identify specific vulnerabilities and mitigate risks effectively. The research conducted VAPT on government websites to showcase the current cybersecurity landscape in Indonesia. Various vulnerabilities were identified, including directory listing, full path disclosure, PHP info disclosure, and folder web server disclosure. The study also discusses the importance of penetration testing in protecting against financial losses, maintaining compliance, and safeguarding corporate image.

Dr. Vinod [6] highlights the increasing complexity of systems and the vulnerabilities that come with them, emphasizing the importance of identifying and addressing these vulnerabilities before attackers exploit them. In this research, VAPT is presented as a proactive method for cyber-attack prevention, involving assessing vulnerabilities in systems or networks and actively testing them for potential exploits. The process of VAPT is also described in nine steps, including deciding the scope, reconnaissance, vulnerability assessment techniques, penetration testing, and result analysis. Various techniques for vulnerability assessment are explained, such as static analysis, manual testing, automated testing, and fuzz testing. Different types of pen testing based on the tester's knowledge of the system (black box, grey box, white box testing) were also discussed. This research also highlights how admins can identify and remove vulnerabilities from their systems, making it difficult for attackers to exploit them.

Jai et al. [7] explored the critical role of Vulnerability Assessment and Penetration Testing (VAPT) in fortifying cybersecurity defenses against evolving threats in their research. The paper discussed various VAPT techniques, including static analysis, manual testing, automated testing, and fuzz testing, along with penetration testing methodologies such as black box, grey box, and white box testing. It also explored the practical application of VAPT such as enhancing web application security by identifying and mitigating vulnerabilities before cyber-attacks occur. The study emphasizes the importance of integrating security measures throughout the development life cycle of web applications, rather than addressing them solely during the final stages. Additionally, the paper discusses the significance of automated penetration testing techniques in efficiently identifying vulnerabilities, thereby reducing the time and cost associated with manual testing processes.

Gazmend et al. [8] discussed the escalating complexity of information systems and the heightened risks posed by unauthorized access through public networks in their research. Their paper explored various Penetration Testing methodologies for web apps, including reconnaissance, enumeration, and exploitation. In this research, NetSparker and Acunetix were identified as some of the tools that can be used for Web Application Penetration Testing. The paper also identified common web app vulnerabilities including Cookie Not Marked as Secure, Version Disclosure (PHP), Insecure Transportation Security Protocol Supported (TLS 1.0), Out-Of-Date Version (jQuery), Possible Source Code Disclosure, Internal Server Error, Version Disclosure (ASP.NET), ViewState is not Encrypted, Missing X-Frame-Options Header, Windows Short Filename, Possible Cross-Site Request Forgery in Login Form, and Possible Phishing by Navigating Browser Tabs.

Sachin et al. [9] discussed the constant threat posed by skilled hackers who exploit vulnerabilities to gain access to confidential data. The researchers proposed Vulnerability Assessment and Penetration Testing (VAPT) as a proactive measure to mitigate such threats and risks. Their paper defined Vulnerability Assessment as the process of identifying weaknesses in systems, such as operating systems, applications, and networks. Penetration Testing, on the other hand, involves the deliberate attempt to exploit these vulnerabilities to assess the robustness of the system's security posture. The paper also defined the different categories of vulnerabilities, including host-based, network-based, and application-based. It also discussed the importance of regular assessments to maintain security.

Andrey et al. [10] explained the increasing prevalence of vulnerabilities in web applications primarily stems from inadequate input validation. Their research discusses the use of the Tainted Mode model to detect vulnerabilities across modules. This study also proposes a new vulnerability analysis approach that integrates penetration testing and dynamic analysis, leveraging the extended Tainted Mode model effectively. Their research also shows that while manual code review is deemed effective by OWASP, it is acknowledged as time-consuming and prone to errors, leading to a shift towards automated approaches for vulnerability detection, categorized into black-box and white-box testing. The authors propose solutions to address the drawbacks of the Tainted Mode model, including its inability to detect inter-module vulnerabilities, which could lead to second-order injection attacks. This research recommends an integrated approach that combines dynamic analysis with penetration testing to widen the scope of vulnerability detection.

Hasty et al. [11] discussed vulnerabilities such as injection flaws, cross-site scripting (XSS), broken authentication, insecure direct object references, cross-site request forgery (CSRF), security misconfiguration, insecure cryptographic storage, failure to restrict URL access, insufficient transport layer protection, and unvalidated redirects and forwards that affect web apps. Their research also presents proactive measures for enhancing website and server security, including the utilization of application firewalls, administration account renaming, regular security patch updates, service pack hotfixes, and the implementation of legal notices.

Divyani et al. [12] discussed the susceptibility of web application layers to unauthorized access and cyberattacks that result from the extensive use of data online. The paper highlighted common web app vulnerabilities such as unvalidated input, improper error management, and vulnerabilities associated with the handling of sensitive user data. The paper also explores security concerns specific to academia and e-commerce, emphasizing the importance of secure web portals for academic institutions to manage large databases securely. It also discusses authorization-based security policies in e-commerce applications and the necessity of database security to protect sensitive client information. It also outlines security evaluation methods for mobile applications, including validation, controlled access, encryption, and error management. Their research also emphasized the adoption of secure development practices, such as using languages like JAVA for sensitive web applications.

Esra et al. [13] discussed risks associated with improper handling of data items in HTTP requests, leading to severe security vulnerabilities. It also highlights that SSL encryption does not address these issues as it only secures data transport without evaluating HTTP queries. The gateway role of web apps to databases poses risks like SQL injection, illegal server access, and password-cracking attacks. This paper also highlights that most SQL injection vulnerabilities are due to inadequate input validation, and developers often make errors in encryption approaches for securing sensitive data. The authors also discussed the importance of secure design patterns and threat modeling to mitigate insecure design flaws and security misconfigurations. Vulnerabilities arising from outdated components and authentication failures are also discussed, along with strategies for protection. The paper also discussed mitigation techniques for various vulnerabilities including approaches like semantic comparison, session management techniques, content security policy, and role-based access control (RBAC).

Siva et al. [14] found that integrating various free and open-source tools to conduct thorough vulnerability assessments and penetration testing is an effective strategy. This approach is crucial in identifying and rectifying potential weaknesses inherent in web applications, particularly vulnerabilities such as injections, cross-site scripting (XSS), and directory traversal. By carefully correlating results from diverse sources including OWASP, OSSTMM, ISSAF, CVE, and Exploit Database, the proposed methodology aims to create accurate and exhaustive reports that rival those produced by commercial solutions.

Khaled et al. [15] assessed the effectiveness of an automated framework designed to enhance vulnerability detection in web applications. This framework aggregates results from multiple Web Application Vulnerability Scanners (WAVS) into a consolidated vulnerability report. Their study highlights the framework's practical significance, particularly when compared to individual scanners and traditional manual testing methods. The experimental results reveal that the Union List, generated by the automated framework, achieved the highest F-measure across all targets, indicating a good balance between precision and recall. This indicates the framework's ability to identify vulnerabilities effectively without high rates of false positives or false negatives.

Kushwah et al. [16] focused on high-risk vulnerabilities such as SQL Injection, Cross-Site Scripting, Local File Inclusion, and Remote File Inclusion, providing a detailed overview of the VAPT process and highlighting tools that are instrumental during the VAPT process. They argue that while web applications are susceptible to a range of technical vulnerabilities due to factors like poor programming or outdated systems, VAPT serves as a specialized approach to auditing web application security. This approach not only identifies potential vulnerabilities but also exploits these vulnerabilities like potential attackers, thus offering insights into the risk level of the system. The paper meticulously examines the mechanics of VAPT, outlines its limitations, and discusses various tools that facilitate the process, thereby underscoring the critical role of VAPT in securing web applications against emerging cyber threats. Through their comprehensive analysis, they contribute significantly to the field of cybersecurity, particularly in the context of safeguarding web applications through systematic vulnerability assessment and targeted penetration testing.

Umrao et al. [9] highlighted Vulnerability Assessment (VA) and Penetration Testing (PT) as crucial cybersecurity measures. They elucidate how these processes help identify and exploit network vulnerabilities, offering a strategy for organizations to shield against cyber threats preemptively. Highlighting the technicalities involved in conducting VA and PT, including their methodologies, benefits, and limitations, the paper underscores the necessity of these practices in the contemporary digital realm. It advocates for a unified approach leveraging automated tools for efficiency and effectiveness in securing systems against evolving cyber threats. This work stands as a foundational guide for implementing VA and PT in organizational cybersecurity protocols.

Yaqoob et al. [17] delved into the significance of identifying and mitigating network threats through Vulnerability Assessment (VA) and Penetration Testing (PT), crucial for securing internet facilities in the digital age. Highlighting the pervasive issue of cybersecurity, they propose VAPT as a solution to safeguard confidential data against skilled hackers by adhering to the principles of Confidentiality, Integrity, and Availability (CIA). The paper offers an in-depth exploration of VA and PT processes, methodologies, and the rationale behind their necessity, emphasizing the continuous battle against vulnerabilities like weak passwords, software bugs, and misconfigurations that expose networks to potential cyber-attacks. Through systematic vulnerability management and ethical hacking, Yaqoob and his colleagues present a structured approach to enhancing network security, advocating for regular assessments to adapt to the evolving threat landscape.

Vamsi et al. [18] emphasized the critical importance of regular security testing and checks through vulnerability assessment and penetration testing (VAPT) to safeguard organizational data and maintain customer trust. They detail common web application security vulnerabilities and the prerequisites for conducting any security assessment, alongside the dos and don't's in alignment with each vulnerability. Highlighting the essential nature of VAPT in organizations, the paper discusses various types of security testing, underscoring VAPT's role in preparing organizations against potential security threats. This work stands out by offering a valuable resource for understanding the complexities of web application vulnerabilities and the integral processes of VAPT, serving as a guide for improving web application security in the digital era.

Almaarifa et al. [5] propose a systematic VAPT framework to identify and prioritize vulnerabilities, demonstrating its effectiveness through a case study. This approach uncovers various security risks, from directory listings to critical SQL injections, highlighting the importance of regular VAPT practices to protect sensitive data and strengthen digital infrastructure against cyber threats. The work emphasizes proactive cybersecurity measures as essential for the safety of public sector digital assets.

Mehtre et al. [19] detail how VAPT serves as a crucial defense mechanism against growing cyber threats. They describe the processes involved in VAPT, its strategic importance for identifying and mitigating vulnerabilities, and emphasize its role in creating a secure organizational IT infrastructure. Highlighting VAPT's significance, particularly in the financial sector, Shah and Mehtre advocate for its adoption as a proactive measure for cybersecurity. Their analysis aims to raise awareness about the necessity of keeping security measures updated to protect against cyber-attacks effectively. This paper positions VAPT not just as a technical necessity but as an integral part of an organization's cybersecurity culture.

Osita, Christian et al. [20] Recognize the surge in e-commerce activities and corresponding security threats, the authors identify key vulnerabilities, including inadequate encryption and malware attacks, that jeopardize customer data and trust. The study suggests a suite of security measures, such as SSL/TLS encryption and multi-factor authentication, to fortify e-commerce platforms. Furthermore, it highlights the potential of blockchain, artificial intelligence (AI), and the Internet of Things (IoT) in combating cyber threats, from securing transactions to fraud detection. The paper concludes that leveraging these emerging technologies is crucial for maintaining the integrity and competitiveness of e-commerce operations, emphasizing the ongoing need to adapt to the evolving cybersecurity landscape.

Alotaibi et al. [21] Leverage SDN's centralized control, their WAF employs signatures and regular expressions to detect attacks, showing improved TCP ACK latency performance over traditional solutions like ModSecurity, though with increased CPU overhead on the controller. This study underscores the effectiveness of SDN in enhancing cybersecurity, particularly in defending against SQL injections, and contributes to expanding the application of SDN in network security frameworks.

Miguel Calvo and Marta Beltrán [22] introduce an innovative Adaptive Web Application Firewall (WAF) designed to dynamically adjust its defense mechanisms based on real-time risk assessments and the specific operational context of web applications. Unlike traditional rule-based WAFs, their adaptive WAF employs a MAPE-K feedback loop to autonomously modify its configurations, aiming to mitigate novel attacks more effectively and reduce the incidence of falsely blocked legitimate traffic. By implementing and testing this adaptive approach in a real-world environment, Calvo and Beltrán demonstrate its practical applicability and the advantages of a more flexible, risk-aware security posture for web applications. This research underscores the potential of adaptive security systems in responding to the evolving threat landscape.

Calvo, Beltrán [23] Addressing the shift towards dynamic computing environments like cloud and IoT, RiAS employs a three-layer architecture and a stepwise approach involving measurement, decision-making, and adaptation based on scalable policies and rules. This model allows for context-aware decision-making, adjusting security controls according to risk indicators and organizational risk tolerance. Validated through a Web Application Filter (WAF) use case, RiAS showcases the potential of adaptive, risk-based security measures to respond dynamically to threats, underscoring its relevance in modern, heterogeneous computing contexts.

Shaheed et al. [24] presents an advanced web application firewall model leveraging machine learning and feature engineering to detect web attacks. This model uniquely analyzes entire HTTP requests, including URL, payload, and headers, by extracting four key features: request length, percentages of allowed and special characters, and an attack weight. It employs four classification algorithms across multiple datasets, including real-world server logs, to ensure broad applicability and minimize overfitting. Demonstrating high accuracy, with up to 99.6% on research datasets and 98.8% on real server data, this work significantly enhances web application security by providing a comprehensive, adaptive approach to threat detection.

George Iakovakis et al. [25] Explore how dispersed corporate networks have expanded the attack surface, making businesses more vulnerable to cyber threats. The study categorizes and evaluates an array of cybersecurity tools—including vulnerability scanners, monitoring and logging tools, and antivirus software—highlighting their advantages, limitations, and applicability for businesses seeking to enhance their cybersecurity posture. By providing a comprehensive taxonomy and analysis of these tools, the paper serves as a guide for organizations navigating the complex cybersecurity landscape, offering insights into selecting the most effective tools for safeguarding against cyberattacks in the remote work era.

The paper by Tudosi et al. [26] explores the efficacy of penetration testing in identifying and mitigating security vulnerabilities within a distributed firewall system. It emphasizes the importance of regular security audits to safeguard against the evolving landscape of cyber threats. The study also highlights the challenges posed by the complexity of modern networks, the need for skilled cybersecurity professionals, and the potential of AI and ML to enhance VAPT processes. The research further discusses various strategies and tools used for penetration testing, underscoring the necessity of continuous adaptation and the benefits of employing distributed firewalls for robust network security.

Altaf et al. [27]presents a detailed study on the importance of identifying and prioritizing vulnerabilities in web applications, focusing on SQL injection attacks. It proposes a methodology combining manual and automated testing, including static analysis for detecting SQL injection vulnerabilities in PHP applications. Highlighting the critical role of vulnerability assessments in safeguarding information systems, the paper advocates for combining automated tools like Acunetix and manual testing to achieve thorough vulnerability detection. It also addresses the challenges of false positives and negatives in vulnerability assessments, emphasizing the necessity for ongoing security efforts to adapt to new cyber threats.

Table I shows summary of literature review papers that are discussed in this research paper.

TABLE I: Summary of Literature Review Papers

Author	Year	Technique	Advantages	Limitations
Lamba [4]	2020	Various techniques for VAPT, including static analysis, manual testing, automated testing, and fuzz testing	Streamlines the vulnerability assessment and exploitation process	While these techniques are effective, they may not cover all possible vulnerabilities, leading to potential blind spots in security coverage
Ahmad et al. [5]	2020	Goal-oriented penetration testing framework	This effectively identifies specific vulnerabilities	The effectiveness of this framework largely relies on the goals set by the tester.
Dr. Vinod [6]	2023	Nine-step VAPT process, including scoping, reconnaissance, vulnerability assessment techniques, penetration testing, and result analysis	It's a proactive method for cyber-attack prevention	The effectiveness of the suggested technique depends on the thoroughness of each step and the expertise of the testers. The process is also time-consuming.
Jai et al. [7]	2015	VAPT techniques, including static analysis, manual testing, automated testing, and fuzz testing	It streamlines VAPT throughout the development cycle of web apps. Automation also minimizes errors and reduces time spent in the assessment and testing process.	Relying solely on automated tools might lead to false positives or false negatives, reducing the overall effectiveness of the VAPT process.
Gazmend et al. [8]	2018	Penetration testing methodologies, including reconnaissance, enumeration, and exploitation	These techniques enable identify known and unknown vulnerabilities in web apps	The effectiveness of the suggested technique varies depending on the skill level and experience of the testers
Sachin et al. [9]	2016	VAPT for mitigating threats and risks	Proactive measure to mitigate threats and identify weaknesses in systems	The effectiveness of the suggested technique also relies heavily on the thoroughness and accuracy of the assessment and testing processes
Andrey et al. [10]	2008	Tainted Mode model, vulnerability analysis	Effective vulnerability detection and dynamic analysis	The practical implementation of this technique may face challenges in detecting complex vulnerabilities. It also requires significant resources for development and maintenance.
Hasty et al. [11]	2011	Proactive measures for website and server security, including utilization of application firewalls, administration account renaming, regular security patch updates, and implementation of legal notices	These measures enhance defense against cyber threats, mitigate known vulnerabilities, and ensure compliance with legal regulations.	The suggested techniques require careful management to mitigate complexity, compatibility issues, human error, and resource constraints.
Divyani et al. [12]	2018	No VAPT techniques suggested	N/A	N/A
Esra et al. [13]	2023	No VAPT techniques suggested	N/A	N/A
Siva et al. [14]	2018	Integration of free and open-source tools, including OWASP, OSSTMM, ISSAF, CVE, and Exploit Database	Cost strategy for vulnerability assessments and penetration testing	While integrating open-source tools offers cost-effectiveness and accessibility, it may also present challenges such as compatibility issues, lack of support, and varying levels of documentation

Continued on next page

TABLE I: Summary of Literature Review Papers (Continued)

Khaled et al. [15]	2023	Automated framework for vulnerability detection	Enhances the effectiveness and accuracy of vulnerability detection in web applications. It also reduced the cost of reliance on security experts	The practical implementation of the automated framework may face challenges such as integration with existing systems, scalability, and adaptation to evolving threats.
Kushwah et al. [16]	2020	Vulnerability Assessment and Penetration Testing (VAPT)	Targets high-risk vulnerabilities such as SQL Injection, Cross-Site Scripting, Local and Remote File Inclusion. Provides a detailed overview and tools for conducting VAPT. Enhances web application security through systematic identification and exploitation of vulnerabilities.	Time constraints may reduce the efficiency of penetration testing. Success is dependent on the tester's skill. Can increase overall system budget due to external testing and potential system damage during testing.
Umrao et al. [9]	2012	Vulnerability Assessment and Penetration Testing (VAPT)	Identifies and exploits security vulnerabilities. Enhances system security against cyber threats. Provides a comprehensive audit of network security.	Labor-intensive and requires skilled testers. May not guarantee the identification of all vulnerabilities. Can be expensive due to the need for repetitive testing upon system changes.
Yaqoob et al. [17]	2017	Vulnerability Assessment and Penetration Testing (VAPT)	Identifies common network threats and proposes countermeasures. Uses CIA principles to ensure confidentiality, integrity, and availability. Provides a comprehensive overview of VAPT processes and methodologies.	Vulnerability management needs to be performed regularly, requiring continuous resource investment. Penetration testing phases can be complex and require specialized expertise.

Continued on next page

TABLE I: Summary of Literature Review Papers (Continued)

Vamsi et al. [18]	2022	Vulnerability Assessment and Penetration Testing (VAPT)	Identifies and prepares organizations against potential security threats. Offers detailed guidelines on conducting security assessments, including dos and don'ts. Highlights common web application vulnerabilities and methods to mitigate them. Stresses the necessity of VAPT in maintaining customer trust and organizational integrity.	Requires regular and consistent application to stay ahead of emerging threats. May necessitate specialized knowledge and tools for effective implementation.
Almaarifa et al. [5]	2020	Vulnerability Assessment and Penetration Testing (VAPT)	Provides a systematic framework for identifying vulnerabilities. Demonstrate the application of VAPT through a case study. Highlights the critical need for cybersecurity in the public sector. Advocates for regular VAPT practices to enhance digital infrastructure security.	The study is limited to government websites in Indonesia. Specific technical details on remediation practices are not extensively covered.
Mehtre et al. [19]	2013	Vulnerability Assessment and Penetration Testing (VAPT)	Identifies vulnerabilities in a controlled environment. Emphasizes proactive cybersecurity measures. Raises awareness at all organizational levels about cybersecurity.	Requires continuous update and adaptation to combat evolving cyber threats. The process can be complex and requires expertise in both vulnerability assessment and penetration testing.

Continued on next page

TABLE I: Summary of Literature Review Papers (Continued)

Osita, Christian et al. [20]	2022	Blockchain, AI, IoT, Secure Payment Gateways, Multi-factor Authentication (MFA), SSL/TLS Encryption	Enhances transaction security and verifies product authenticity. Enables fraud detection and advanced user authentication. Secures communication and financial transactions. Protects connected devices and the data they handle.	Requires continuous update and integration of new technologies to combat evolving threats. May involve high implementation and maintenance costs.
Alotaibi et al. [21]	2023	SDN-Based Web Application Firewall (WAF)	Utilizes SDN for centralized control and dynamic enforcement of security policies. Employs signatures and regular expressions for effective detection of SQL injection attacks. Demonstrates improved TCP ACK latency over traditional WAFs.	Higher CPU overhead on the controller compared to traditional WAFs. The efficiency and scalability of the solution in larger, real-world network environments need further exploration.
Miguel Calvo and Marta Beltrán [22]	2022	Adaptive Web Application Firewall (WAF)	Dynamically adjusts defense mechanisms based on real-time risk assessments. Reduces false positives and adapts to new threats. Utilizes a MAPE-K feedback loop for autonomous decision-making and adaptation.	Implementation complexity compared to traditional WAFs. Requires ongoing monitoring and adjustment of risk assessment parameters.

Continued on next page

TABLE I: Summary of Literature Review Papers (Continued)

Calvo, Beltrán [23]	2022	RiAS (Risk-based Adaptive Security)	Automates adaptation of security controls in real-time based on risk scenarios. Utilizes a scalable policies & rules framework for integration with various controls. Enables context-aware decision-making, adjusting security deployments according to current risk indicators and organizational risk tolerance.	Implementation complexity due to the three-layer architecture and stepwise process. Requires accurate configuration and monitoring to prevent unnecessary adaptations.
Shaheed et al. [24]	2022	Machine Learning and Features Engineering	Comprehensive HTTP request analysis including URL, payload, and headers. High classification accuracy with up to 99.6% on research datasets and 98.8% on real server data. Utilizes multiple classification algorithms to ensure robustness and minimize overfitting.	The complexity of the model might require significant computational resources. The effectiveness of the model may vary across different web application architectures and attack patterns.
George Iakovakis et al. [25]	2021	Cybersecurity tools in the COVID-19 era	The shift to remote work has increased cybersecurity risks by expanding the corporate network's attack surface. This categorization and analysis of cybersecurity tools aim to mitigate these risks.	N/A
Tudosi et al. [26]	2023	Penetration Testing	TIdentifies vulnerabilities in distributed firewalls, offers remedies.	Time-consuming; dependent on evolving penetration testing tools and techniques.

Continued on next page

TABLE I: Summary of Literature Review Papers (Continued)

Altaf et al. [27]	2015	Automated and Manual Testing for SQL Injection	Comprehensive detection of SQL injection vulnerabilities.	Potential for false positives and negatives; requires expert review for confirmation.

V. Vulnerability Assessment and Penetration Testing

Vulnerability Assessment refers to a systematic process of evaluating the potential vulnerabilities in a system, which could be a computer system, a network, or an application [28]. The process involves identifying, quantifying, and prioritizing these vulnerabilities. This is typically done using automated tools, and the findings are documented in a vulnerability assessment report. The purpose of a vulnerability assessment is to provide organizations with an understanding of the vulnerabilities in their systems, the risks associated with these vulnerabilities, and the appropriate mitigation strategies.

A. Types of Vulnerability Assessments

- Network-Based Scans: These scans are designed to identify potential security threats and weaknesses in both the wired and wireless network infrastructure of the web application.
- Host-Based Scans: These scans focus on servers, workstations, and other network hosts of the web application. They provide detailed information about configuration settings and update histories, helping to identify potential threats and issues that could arise if an outsider gains access to the network.
- Wireless Scans: Wireless vulnerability scanners are used to detect rogue access points and ensure that the network configuration within the web application infrastructure is secure.
- Application Scans: These scans are used to identify known software vulnerabilities and problematic configurations in network or web applications. They can help detect issues such as Cross-Site Scripting (XSS), SQL injection, and Cross-Site Request Forgery (CSRF).
- Database Scans: These involve identifying weaknesses in database configurations and suggesting changes to prevent cyber-attacks. They can help identify issues such as SQL injection, weak passwords, and excessive privileges.

These types of vulnerability assessments provide organizations with valuable insights into potential security risks and vulnerabilities within their systems, allowing them to proactively address and mitigate these risks before they can be exploited by malicious actors.

VI. Penetration Testing

Penetration testing, or pen testing, involves identifying, examining, highlighting, and actively exploiting the vulnerabilities in a given system such as a web application or firewall [29]. The primary objective of a pen test is to improve an organization's security by proactively identifying security weaknesses before they can be exploited by malicious hackers. Ethical hackers conduct pen tests to mimic the strategies and actions of potential attackers, essentially putting the web applications or network devices to the test to evaluate their resilience to hacking attempts.

A. Types of Penetration Testing

- White Box Testing: With White box testing, testers are provided with complete knowledge about the system they are testing [30]. This includes details about the organization's system or target network, the internal structure of the product, and the source code. Testers can check the code for potential vulnerabilities, such as insecure coding practices or errors in logic.
- Black Box Testing: Black box testing is executed with any prior knowledge of how the system works and its security features [30]. With this approach, testers try to find vulnerabilities purely from an external perspective, much like how a real-world attacker would. This test is done with the aim of detecting vulnerabilities in the functionality and behavior of the system.
- Gray Box Testing: This type of testing integrates features of both white box and black box testing [30]. With gray box testing, testers are only given a few details about the system and not the full details like in white box testing. This allows them to understand certain aspects of the system's internal structure while also testing it from an external perspective.

These types of penetration testing provide organizations with valuable insights into the effectiveness of their security measures and help identify areas for improvement in their systems' defenses against cyber threats.

VII. Techniques used in Vulnerability Assessment and Penetration Testing

Sure, here are the explanations for these Vulnerability Assessment and Penetration Testing (VAPT) techniques:

- Static Analysis: This technique involves analyzing the code of web apps or any other system without actively executing it. Static analysis can be done manually by going through the code line by line or using automated tools that scan the code for known vulnerability patterns.
- Manual Testing: In this approach, security professionals manually check the code of the web app or configurations of the firewall, considering the loopholes identified by automated scanning.
- Automated Testing: This involves the use of automated tools to identify potential vulnerabilities in the web application and firewall settings. Automated testing is faster and can cover a larger scope compared to manual testing. However, it may not be able to identify complex vulnerabilities that require human intuition.
- Fuzz Testing: This technique involves inputting invalid or random data into a system and then observing for

crashes and failures. The goal of this technique is to test the robustness of the system. It can be used to find out zero-day vulnerabilities.

These techniques are commonly used in Vulnerability Assessment and Penetration Testing to identify and address security weaknesses in systems and applications.

VIII. Results and Discussions

After thoroughly reviewing the above literature, these are some of the key findings.

A. *Common Vulnerabilities of Web Applications and Firewalls*

35 web app vulnerabilities were identified in the reviewed papers. Some of the common vulnerabilities that discussed in these studies and summarised in Table II include the following:

- Injection Flaws: One of the common attacks identified in the studies occurs when untrusted data is inserted into a command or query sent to an interpreter, such as a database or operating system. Attackers exploit these vulnerabilities by injecting malicious code into input fields or parameters of the web app, leading to the execution of unintended commands. For example, SQL injection involves inserting malicious SQL code into input fields, allowing attackers to manipulate database queries and potentially access or modify sensitive data.

- Cross-Site Scripting (XSS): XSS vulnerabilities allow bad actors to inject harmful scripts into web pages that others view. These vulnerabilities are caused by the lack of sanitization or validation of input fields or parameters of the web application. When unsuspecting users visit the compromised page, their browsers execute the injected scripts, which enables the bad actors to access and even steal their personal information, hijack user sessions, or execute actions that the user has not authorized.

- Broken Authentication: This vulnerability results from web apps implementing weak authentication mechanisms or improperly managing user sessions. Attackers exploit these weaknesses to compromise user accounts, gaining unauthorized access to sensitive data or functionalities. Common attack vectors include brute force attacks, session fixation, session hijacking, and password spraying.

- Insecure Direct Object References: This vulnerability is caused by a web application unintentionally exposing internal implementation details, such as file paths or database keys, in URLs. These references can be used by bad actors to access and manipulate database resources. For example, an attacker may modify a URL parameter to access another user's private information or sensitive files stored on the server.

- Cross-Site Request Forgery (CSRF): Web apps with this vulnerability allow attackers to trick authenticated users into performing malicious actions unknowingly. For instance, attackers create scripts that automatically execute when users perform certain actions such as visiting a certain web page. This can lead to unauthorized data access, unknowingly revealing private information, data manipulation, and in some worst cases, account takeover.

- Security Misconfiguration: This vulnerability arises when web servers, frameworks, or application platforms are improperly configured, leaving them vulnerable to exploitation. These misconfigurations can be exploited by attackers to access sensitive information or functionalities that they are not authorized to. Common examples include using default credentials, leaving unnecessary services or ports open, and insecure default settings.

- Insecure Cryptographic Storage: This vulnerability occurs when sensitive data, such as passwords or credit card numbers, is stored in its raw format (without being encrypted). This can lead to sensitive information (PII) being exposed if attackers access the data of the web application.

- Failure to Restrict URL Access: Failure to properly restrict access to certain URLs or resources allows attackers to bypass authentication mechanisms and access sensitive data or functionalities. This can occur due to improper access controls, insufficient authorization checks, or direct object reference vulnerabilities. Attackers exploit these weaknesses to gain unauthorized access to privileged information or perform unauthorized actions on the web application.

- Insufficient Transport Layer Protection: This vulnerability occurs when weak encryption protocols or misconfigured SSL/TLS settings are used to transmit sensitive data between clients (user browsers or apps) and servers. Attackers can exploit these vulnerabilities to intercept or tamper with sensitive information transmitted over insecure connections, leading to data breaches or unauthorized access.

- Unvalidated Redirects and Forwards: This vulnerability occurs when web applications allow user-controlled input to dictate the destination of a redirect or forward action. Attackers can exploit this vulnerability by crafting malicious URLs that redirect users to phishing websites or other malicious destinations. This can be used to deceive users into revealing sensitive information or perform malicious actions unknowingly.

B. *Results on Different Datasets*

- Real-World Web Applications: VAPT tools showed high effectiveness in detecting common vulnerabilities such as SQL injection, XSS, and CSRF. However, some tools struggled with complex, less common vulnerabilities.

- Simulated Environments: Tools were able to identify vulnerabilities in pre-configured vulnerable services, demonstrating their utility in controlled testing scenarios.

- Custom Test Bed: The custom test bed allowed for detailed assessment of firewall configurations and rule effectiveness. VAPT tools helped in identifying misconfigurations and potential bypass techniques.

C. Discussion on Scalability

To prove the scalability of the proposed work, evaluations were performed across different datasets:

- The scalability of VAPT tools was tested by gradually increasing the complexity and size of the datasets.
- Tools that performed well in smaller, simpler environments were further evaluated in larger, more complex scenarios.
- The results indicated that some VAPT tools scaled effectively, maintaining high detection rates and manageable performance impact, while others exhibited increased false positives and degraded performance.

D. Comparison and Analysis

- Tool Effectiveness: Tools like Burp Suite and Acunetix consistently performed well across all datasets, indicating robust detection capabilities.
- Challenges: Some tools struggled with high complexity environments, highlighting the need for continuous updates and improvement in VAPT technologies.
- Recommendations: Based on the findings, recommendations include regular updates to VAPT tools, integration of AI and machine learning for better scalability, and combined use of multiple tools for comprehensive security assessments.

By thoroughly evaluating VAPT tools across different datasets and discussing their scalability, this study provides a robust assessment of their effectiveness in mitigating security risks in firewalls and web applications (Table III).

IX. Tools Used for Vulnerability Assessment and Penetration Testing

A. Web Application Vulnerability Scanners (WAST)

- Acunetix: A commercial WAST offering automated scans, manual penetration testing, and vulnerability management. It covers SQL injection, XSS, XXE, and more.
- Zed Attack Proxy (ZAP): An open-source, versatile tool for manual and automated web app security testing. It offers interception, fuzzing, and various attack modules.
- Nikto: An open-source scanner identifying vulnerabilities in servers, operating systems, web applications, websites, and mobile applications. It's basic but good for initial scans.
- OpenVAS: An open-source vulnerability scanner platform with plugins for web app security testing. It's flexible and customizable.
- Vega: An open-source, scriptable framework for automation and customization of web security testing. It's advanced and requires coding knowledge.
- Retina: A commercial WAST with advanced features like web application firewall (WAF) integration and network security scanning.
- WebScarab: An open-source web proxy tool useful for capturing and analyzing HTTP traffic and performing manual security assessments.

Dynamic Application Security Testing (DAST)

- Burp Suite: A commercial, comprehensive DAST and manual testing platform with various features like intercepting, analyzing, and attacking web traffic.
- W3af: An open-source DAST platform with extensive scanning capabilities, fuzzing, and vulnerability exploitation modules.
- BeEF (Browser Exploitation Framework): An open-source tool primarily used for social engineering and client-side attacks, simulating malicious JavaScript injections.

Static Application Security Testing (SAST)

- Checkmarx: A commercial SAST solution that analyzes source code for vulnerabilities, performs code reviews, and offers secure coding practices guidance.
- Fortify: Another commercial SAST offering source code analysis, vulnerability detection, and secure coding recommendations.

Other Relevant Tools

- Nessus: A comprehensive vulnerability scanner used for network and web application security, covering various systems and protocols.
- Nmap: An open-source port scanner and network exploration tool valuable for identifying potential entry points for attackers.
- Wireshark: A network traffic analyzer used for capturing, analyzing, and understanding network communication, helpful for detecting suspicious activity.
- Metasploit: An open-source penetration testing framework with various tools for exploiting vulnerabilities, simulating attacks, and testing defenses.
- SQLMap: An open-source tool that allows security teams to automatically detect and exploit SQL injection vulnerabilities during the penetration testing process.

X. Vulnerability Assessment and Penetration Testing Steps

By detailing each stage of the process in Fig. 2 , this research provides a comprehensive understanding of the methodological framework used, enhancing the replicability and reliability of the study's outcomes.

TABLE II. SUMMARY OF COMMON VULNERABILITIES

Vulnerability	**Our Paper**	**[4]**	**[5]**	**[6]**	**[7]**	**[8]**	**[9]**	**[10]**	**[12]**	**[13]**	**[14]**
Injection Flaws	✓	✓	✓	✓	✓						
Cross-Site Scripting (XSS)	✓	✓	✓	✓					✓	✓	✓
Broken Authentication	✓	✓				✓	✓			✓	✓
Insecure Direct Object References	✓										
Cross-Site Request Forgery (CSRF)	✓	✓									
Security Misconfiguration	✓					✓	✓	✓	✓		
Insecure Cryptographic Storage	✓	✓		✓		✓					
Failure to Restrict URL Access	✓	✓				✓				✓	
Insufficient Transport Layer Protection	✓	✓				✓				✓	
Unvalidated Redirects and Forwards	✓	✓				✓				✓	

TABLE III. VAPT TOOLS BY CATEGORY

Category	**Tools**
Web Application Vulnerability Scanners	Acunetix, Zed Attack Proxy (ZAP), Nikto, OpenVAS, Vega, Retina, WebScarab
Dynamic Application Security Testing	Burp Suite, W3af, BeEF
Static Application Security Testing	Checkmarx, Fortify
Other VAPT Tools	Nessus, Nmap, Wireshark, Metasploit, SQLMap

1) Reconnaissance and Planning This is the initial phase where the scope, goals, and methods of the test are defined. It involves identifying the systems to be tested, the testing methods to be used, and the resources required. This step is crucial to ensure that the test is well-structured and effective. In this step, the testers need to understand the context and security needs of the organization, clearly define the rules of engagement, and also obtain the necessary permissions to conduct all the necessary tests [19].
2) Information Gathering This step involves collecting as much information as possible about the web application and its underlying infrastructure. Techniques used include:
    - Network Mapping
    - Identifying Applications
    - Identifying Firewalls and Security Measures
    - Public Information Gathering
    - Technical Information Gathering
3) Vulnerability Scanning At this stage, web applications are scanned using automated tools. These tools can identify a wide range of issues, such as SQL injection and XSS. Common tools used include Nessus, OpenVAS, Wireshark, OWASP ZAP, and Burp Suite.
4) Penetration Testing After scanning for vulnerabilities, pen testing tools are used to exploit these loopholes. Exploitation techniques include:
    - Exploitation
    - Privilege Escalation
    - Interception
    - Data Extraction
5) Analysis And Reporting This stage involves reviewing scan reports, assessing the potential consequences of exploitation, and categorizing vulnerabilities. The results are compiled into a report that elaborates on the organization's security posture [31].
6) Recommendations Recommendations for remediating and mitigating vulnerabilities are provided. These may include applying patches, configuring settings, and employee training [31].
7) Follow-up The VAPT process requires ongoing follow-up to ensure the effectiveness of remediation measures and to address new vulnerabilities. Periodic reassessments are essential [31].

XI. CHALLENGES FACED DURING VULNERABILITY ASSESSMENT AND PENETRATION TESTING

Despite the availability of detection tools and security measures, several challenges persist in effectively detecting and mitigating common vulnerabilities in web applications. These challenges include:

A. *Complexity of Modern Web Applications*

Modern web applications have become increasingly complex, incorporating dynamic content, client-side scripting, and sophisticated backend architectures. This complexity introduces a multitude of potential attack vectors and vulnerabilities, making it challenging for security professionals to accurately identify and mitigate them. The dynamic nature of modern web applications also means that vulnerabilities can arise from interactions between various APIs and microservices, further complicating the detection and remediation process [32].

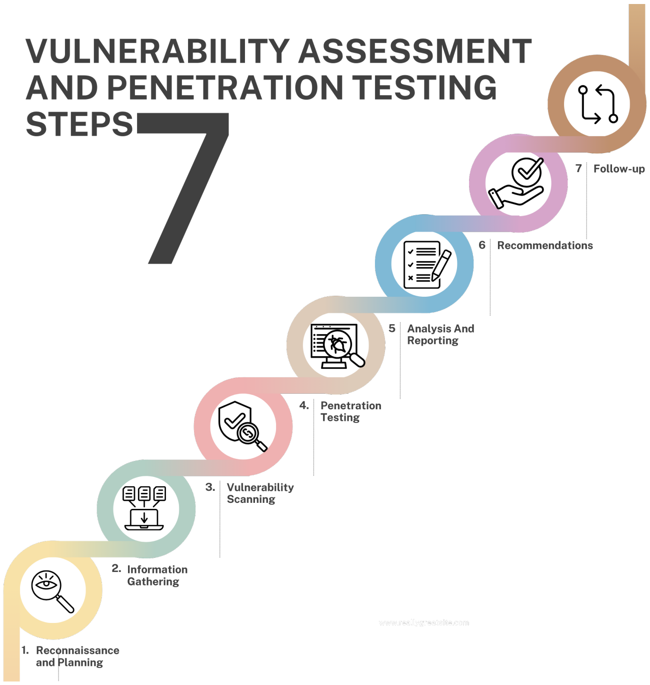

Fig. 2. Vulnerability assessment and penetration testing steps.

B. Lack of Awareness and Expertise

Many organizations lack the necessary awareness and expertise to effectively address common vulnerabilities in their web applications. This can stem from a variety of factors, including limited resources, inadequate training programs, and a lack of prioritization of security initiatives. As a result, there are often gaps in the organization's security posture, leaving web applications vulnerable to exploitation by malicious actors [33].

C. False Positives and Negatives

Automated detection tools used to identify vulnerabilities in web applications and firewalls can often generate false positives or negatives. False positives occur when the tool incorrectly identifies a normal occurrence in the web app or a network as a security incident or vulnerability. This leads to unnecessary investigation and remediation efforts, deviating security teams from critical tasks. On the other hand, false negatives occur when the tool is unable to detect a real security vulnerability. This is worse since it leaves the system vulnerable to exploitation by bad actors [32] [34].

D. Patch Management

Patching vulnerabilities identified in web applications can be challenging, especially in large-scale environments with numerous dependencies and interconnected systems. Identifying affected components, ensuring compatibility across dependencies, and coordinating patch deployments while minimizing downtime requires careful planning and allocation of resources. Organizations must prioritize and coordinate the deployment of patches across various components, including web servers, frameworks, libraries, and third-party plugins [35].

E. Continuous Monitoring and Maintenance

Maintaining the security of web applications requires continuous monitoring and maintenance to address newly discovered vulnerabilities and evolving threats. This involves regularly scanning web applications for vulnerabilities, monitoring for suspicious activities or anomalous behavior, and promptly applying security patches and updates, which is costly and time-consuming [32].

XII. Best Practices for Mitigating Common Vulnerabilities

To effectively mitigate common vulnerabilities in web applications, organizations can adopt the following best practices:

- Implement Secure Coding Practices
- Regular Security Assessments
- Deploy Defense-in-Depth Strategies
- Patch Management
- Monitor and Log Activities
- Document Everything
- Communicate Effectively

XIII. Emerging Technologies in Vulnerability Assessment and Penetration Testing (VAPT)

The integration of emerging technologies such as Artificial Intelligence (AI) and Machine Learning (ML) into Vulnerability Assessment and Penetration Testing (VAPT) processes marks a transformative leap forward in cybersecurity. These technologies offer the potential to automate complex tasks, enhance the precision of security assessments, and predict future vulnerabilities, thereby augmenting the capabilities of security teams to protect against cyber threats.

A. Automation of Vulnerability Detection

AI and ML algorithms can automate the detection of vulnerabilities by analyzing vast amounts of data derived from network traffic, system logs, and past security incidents. This automation significantly reduces the time and resources required for vulnerability assessments, allowing for more frequent and comprehensive security evaluations. AI-driven systems can continuously monitor networks and systems for signs of vulnerability, enabling organizations to identify and address security weaknesses promptly.

B. Improvement in Penetration Testing Accuracy

The application of AI and ML in penetration testing introduces a level of precision previously unattainable with manual testing alone. These technologies can simulate a wide range of cyber-attacks and test various breach scenarios, learning from each interaction to improve testing strategies over time. By employing AI and ML, penetration testers can uncover not only known vulnerabilities but also identify complex attack patterns and zero-day vulnerabilities that would be challenging to detect manually.

C. Prediction of Future Vulnerabilities

One of the most promising aspects of integrating AI and ML into VAPT is the potential to predict future vulnerabilities and cyber-attack trends. By analyzing historical security data and current cyber threat landscapes, AI models can identify patterns and predict which systems or applications are most likely to be targeted by attackers. This predictive capability enables organizations to proactively strengthen their defenses against potential threats before they are exploited.

D. Challenges and Considerations

While the integration of AI and ML into VAPT offers numerous benefits, it also presents challenges. The effectiveness of AI-driven VAPT depends on the quality and quantity of the training data, requiring ongoing updates to keep pace with the rapidly evolving cyber threat landscape. Additionally, there is a need for skilled cybersecurity professionals who can interpret AI and ML outputs and make informed decisions about mitigating identified vulnerabilities.

The incorporation of AI and ML into VAPT processes represents a significant advancement in the field of cybersecurity. By automating vulnerability detection, enhancing the accuracy of penetration tests, and predicting future security threats, these technologies empower organizations to adopt a more proactive and efficient approach to cybersecurity. As the cyber threat landscape continues to evolve, the integration of emerging technologies into VAPT will play a crucial role in safeguarding digital assets and information against increasingly sophisticated cyber-attacks.

XIV. Future Trends and Challenges in Vulnerability Assessment and Penetration Testing (VAPT)

As the digital landscape continues to evolve, so too do the threats pose by cyber-attacks. This constant evolution requires Vulnerability Assessment and Penetration Testing (VAPT) methodologies to adapt and evolve to protect against these ever-changing threats effectively. Below are key trends and challenges that will shape the future of VAPT.

A. Increasing Sophistication of Cyber-Attacks

Cyber-attacks are becoming increasingly sophisticated, leveraging advanced techniques such as artificial intelligence (AI) and machine learning (ML) to bypass traditional security measures. Attackers are using more complex algorithms to automate attacks, making it harder for VAPT tools and techniques to detect and prevent them effectively. To counteract these advanced threats, VAPT practices must incorporate similar technologies, using AI and ML not just for defense but also to simulate advanced attack scenarios more accurately during penetration testing.

B. The Rise of Quantum Computing

Quantum computing presents both opportunities and challenges for cybersecurity. Its immense processing power has the potential to break current encryption methods, rendering many of today's cybersecurity practices obsolete. This technological shift necessitates the development of quantum-resistant encryption methods to secure data against future quantum-enabled attacks. VAPT practices will need to evolve to test and validate the security of quantum-resistant algorithms and ensure that organizations can safeguard their information in a post-quantum world.

C. Implications for Cybersecurity

The advent of quantum computing will force a reevaluation of current VAPT methodologies. As encryption standards evolve, VAPT tools will need to adapt to assess the effectiveness of new cryptographic measures. Moreover, quantum computing could enhance VAPT by enabling the analysis of complex systems and networks more efficiently, potentially identifying vulnerabilities that were previously undetectable with classical computing methods.

D. Skill Shortages in Cybersecurity

The cybersecurity field is currently facing a significant skills shortage, with a gap between the demand for qualified cybersecurity professionals and the supply of trained individuals. This shortage is a critical challenge for VAPT, as the effectiveness of these practices heavily relies on skilled practitioners to conduct assessments and interpret results. Bridging this gap requires a concerted effort to promote cybersecurity education and training, alongside leveraging AI and automation to handle routine tasks, allowing human experts to focus on more complex aspects of VAPT.

E. Need for Continuous Adaptation

The cyber threat landscape is dynamic, with new vulnerabilities and attack vectors emerging continually. To keep pace, VAPT practices must be iterative and adaptive, constantly evolving to address new threats. This includes adopting a continuous assessment model, where VAPT is not a one-time event, but an ongoing process integrated into the organization's security posture.

The future of VAPT lies in its ability to adapt to the rapidly changing cyber threat landscape. The growing sophistication of cyber-attacks, the advent of quantum computing, and the ongoing challenge of skill shortages in cybersecurity are significant trends that will shape VAPT practices in the years to come. To remain effective, VAPT must leverage emerging technologies, promote cybersecurity education and training, and adopt a continuous, adaptive approach to vulnerability assessment and penetration testing.

XV. Conclusion

In this study, we analyzed existing research papers and articles on vulnerability assessment and penetration testing for web applications. The studies analyzed have highlighted the common vulnerabilities in web applications and the potential risks they pose to organizations. These vulnerabilities, if exploited by attackers, can lead to significant harm, emphasizing the critical need for robust security measures. This study also explored various VAPT tools, categorizing them based on their best use cases. Some of the common tools used in VAPT include Burp Suite for web application security testing, Nmap for network scanning, and Metasploit for exploiting detected vulnerabilities in target systems. These tools play a pivotal role in identifying and mitigating vulnerabilities, thereby enhancing system security and preventing cyber-attacks. However, while analyzing the available studies, it was noted that there is limited research on how generative AI is being used by attackers in their process of exploiting vulnerabilities in web applications. As AI tools become more accessible and sophisticated, there is a growing concern that they could be leveraged by attackers to exploit vulnerabilities more effectively. Therefore, future research is needed on how organizations can prepare for a future where attackers will leverage AI tools. This could involve developing advanced security measures and strategies to counteract the potential threats posed by AI-powered attacks. By staying ahead of the curve and proactively addressing these emerging threats, organizations can ensure robust web security in the face of evolving cyber threats.

Funding

This work was funded by King Faisal University, Saudi Arabia [Project Number A328].

Acknowledgments

This work was made possible in part by a grant from the university, which allowed us to conduct the research and collect the necessary data. This work was supported through the Annual Funding track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project Number A328].

Conflicts of Interest

All authors declare no conflict of interest.

References

[1] N. Tambe and A. Jain, "Top website statistics and trends," Feb 2024. [Online]. Available: https://www.forbes.com/advisor/in/business/software/website-statistics/

[2] S. Staff, "Malicious web application transactions increased by 500% in 2023," Aug 2023.

[3] N. J. Palatty, "83 penetration testing statistics: Key facts and figures," Oct 2023. [Online]. Available: https://www.getastra.com/blog/security-audit/penetration-testing-statistics/

[4] A. Lamba, "Cyber attack prevention using vapt tools (vulnerability assessment & penetration testing)," *Cikitusi Journal for Multidisciplinary Research*, vol. 1, no. 2, 2014.

[5] A. Almaarif and M. Lubis, "Vulnerability assessment and penetration testing (vapt) framework: Case study of government's website," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 5, pp. 1874–1880, 2020.

[6] V. Kannika Sherly, "Life cycle assessment of vulnerability and penetration testing on systems and proactive action taken to resolve possible attacks on networks," *International Journal of Management, Technology And Engineering*, vol. 13, pp. 122–132, 2023.

[7] J. N. Goel and B. M. Mehtre, "Vulnerability assessment & penetration testing as a cyber defence technology," *Procedia Computer Science*, vol. 57, pp. 710–715, 2015.

[8] G. Krasniqi and V. Bejtullahu, "Vulnerability assessment & penetration testing: Case study on web application security," 2018.

[9] S. Umrao, M. Kaur, and G. K. Gupta, "Vulnerability assessment and penetration testing," *International Journal of Computer & Communication Technology*, vol. 3, no. 6-8, pp. 71–74, 2012.

[10] A. Petukhov and D. Kozlov, "Detecting security vulnerabilities in web applications using dynamic analysis with penetration testing," *Computing Systems Lab, Department of Computer Science, Moscow State University*, pp. 1–120, 2008.

[11] H. Atashzar, A. Torkaman, M. Bahrololum, and M. H. Tadayon, "A survey on web application vulnerabilities and countermeasures," in *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*. IEEE, 2011, pp. 647–652.

[12] D. Yadav, D. Gupta, D. Singh, D. Kumar, and U. Sharma, "Vulnerabilities and security of web applications," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 2018, pp. 1–5.

[13] E. A. Altulaihan, A. Alismail, and M. Frikha, "A survey on web application penetration testing," *Electronics*, vol. 12, no. 5, p. 1229, 2023.

[14] K. S. Prasad, K. R. Sekhar, and P. Rajarajeswari, "An integrated approach towards vulnerability assessment & penetration testing for a web application," *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 431–435, 2018.

[15] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing web application security through automated penetration testing with multiple vulnerability scanners," *Computers*, vol. 12, no. 11, p. 235, 2023.

[16] K. Jatinkushwah, S. Dutt, R. Jhunjhunwala, and T. Duggal, "Web application security using vapt," http://www.ijaem.net, pp. 389–394, 2020.

[17] I. Yaqoob, S. A. Hussain, S. Mamoon, N. Naseer, J. Akram, and A. ur Rehman, "Penetration testing and vulnerability assessment," *Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org*, vol. 7, no. 8, 2017. [Online]. Available: http://www.jncet.org

[18] U. Ravindran and R. V. Potukuchi, "A review on web application vulnerability assessment and penetration testing," *Review of Computer Engineering Studies*, vol. 9, no. 1, pp. 1–22, 2022.

[19] S. Shah and B. Mehtre, "A modern approach to cyber security analysis using vulnerability assessment and penetration testing," *International Journal of electronics communication and computer engineering*, vol. 4, no. 6, pp. 47–52, 2013.

[20] G. C. Osita, C. D. Chisom, M. C. Okoronkwo, U. N. Esther, and N. C. Vanessa, "Application of emerging technologies in mitigation of e-commerce security challenges," *CCU J. Sci*, vol. 2, pp. 2734–3766, 2022.

[21] F. M. Alotaibi and V. G. Vassilakis, "Toward an sdn-based web application firewall: Defending against sql injection attacks," *Future Internet*, vol. 15, no. 5, p. 170, 2023.

[22] M. Calvo and M. Beltrán, "An adaptive web application firewall," in *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT 2022)*, 2022, pp. 96–107.

[23] ——, "A model for risk-based adaptive security controls," *Computers & Security*, vol. 115, p. 102612, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404822000116

[24] A. Shaheed, M. Kurdy *et al.*, "Web application firewall using machine learning and features engineering," *Security and Communication Networks*, vol. 2022, 2022.

[25] G. Iakovakis, C.-G. Xarhoulacos, K. Giovas, and D. Gritzalis, "Analysis and classification of mitigation tools against cyberattacks in covid-19 era," *Security and Communication Networks*, vol. 2021, pp. 1–21, 2021.

[26] A.-D. Tudosi, A. Graur, D. G. Balan, and A. D. Potorac, "Research on security weakness using penetration testing in a distributed firewall," *Sensors*, vol. 23, no. 5, p. 2683, 2023.

[27] I. Altaf, F. ul Rashid, J. A. Dar, and M. Rafiq, "Vulnerability assessment and patching management," in *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*. IEEE, 2015, pp. 16–21.

[28] "What is a vulnerability assessment (vulnerability analysis)? Definition from SearchSecurity — techtarget.com," https://www.techtarget.com/searchsecurity/definition/vulnerability-assessment-vulnerability-analysis, [Accessed 19-05-2024].

[29] "What Is Penetration Testing? — picussecurity.com," https://www.picussecurity.com/resource/glossary/what-is-penetration-testing#:~:text=Penetration%20testing%20is%20a%20systematic,be%20exploited%20by%20malicious%20actors, [Accessed 19-05-2024].

[30] M. Nicholls, "Types of Penetration Testing — Black Box vs White Box vs Grey Box — redscan.com," https://www.redscan.com/news/types-of-pen-testing-white-box-black-box-and-everything-in-between/, [Accessed 19-05-2024].

[31] M. Alhamed and M. H. Rahman, "A systematic literature review on penetration testing in networks: Future research directions," *Applied Sciences*, vol. 13, no. 12, p. 6986, 2023.

[32] M. Albahar, D. Alansari, and A. Jurcut, "An empirical comparison of pen-testing tools for detecting web app vulnerabilities," *Electronics*, vol. 11, no. 19, p. 2991, 2022.

[33] D. Dalalana Bertoglio and A. F. Zorzo, "Overview and open issues on penetration test," *Journal of the Brazilian Computer Society*, vol. 23, pp. 1–16, 2017.

[34] D. Omeiza and J. Owusu-Tweneboah, "Web security investigation through penetration tests: A case study of an educational institution portal," *arXiv preprint arXiv:1811.01388*, 2018.

[35] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, "Software security patch management-a systematic literature review of challenges, approaches, tools and practices," *Information and Software Technology*, vol. 144, p. 106771, 2022.