DMPedia Lecture Notes in Computer Science & Engineering
Volume 1, Issue 2, pp.148-155, 2026

ICICCS-2026

Penetration Testing in System Security

Shruti Agarwal, Shilpi Sharma

Amity University Uttar Pradesh

shrutiagarwal2649@gmail.com, ssharma22@amity.ed

Abstract

Cybersecurity has emerged as a key issue in contemporary computing environments due to an increasing dependence on digital systems and networked infrastructures. By discovering and exploiting vulnerabilities in a controlled and ethical manner, penetration testing has become an essential method for evaluating system security. Reducing cyber risks helps organisations assess their security posture, identify potential vulnerabilities, and implement appropriate defences. This review paper offers a comprehensive overview of the current literature on penetration testing, covering its methodologies, tools, frameworks, and applications across diverse areas, including network security, web application security, and enterprise systems. This study includes findings from various research projects, focusing on the efficacy of both manual and automated testing methodologies, including the application of structured frameworks, vulnerability scanning tools, and exploitation platforms. It also examines how standardised guidelines and methods can ensure that security assessments are systematic and reliable. The review goes into more detail on important topics, such as common attack methods (e.g., man-in-the-middle attacks, packet sniffing, SQL injection, cross-site scripting, and cross-site request forgery), and how to protect against them. It also points out how important automation and ongoing security assessment are becoming for making penetration testing more accurate and efficient. This review's results show that penetration testing is an important part of improving cybersecurity because it helps identify weaknesses before they become problems and reduces risk. The paper concludes by highlighting the need for structured, thorough penetration testing and advocating further research into advanced, automated security testing to address new cyber threats.

Keywords: - Penetration Testing, Vulnerability Assessment, Cybersecurity, Network Security, Metasploit, OWASP, NIST

1. Introduction

Organisations are now much more vulnerable to various cyber threats due to the rapid development of information technology and networked systems in the modern digital age. Ensuring the security of systems has become a crucial concern for people, businesses, and governments alike as systems become more intricate and interconnected. Penetration testing has become better known as an efficient and effective security assessment method among the strategies for protecting digital infrastructure. Penetration testing is a systematic method for assessing a system's security posture by identifying, analysing, and exploiting vulnerabilities in a controlled environment. It mimics real cyberattacks to identify vulnerabilities that malicious actors might exploit. To improve the effectiveness of penetration testing across domains such as network security, web applications, and enterprise systems, academics and professionals have developed a range of approaches, tools, and frameworks over time.

The current body of research demonstrates that penetration testing is essential for reducing risk, improving security, and adhering to accepted standards, such as structured testing frameworks and guidelines, as well as for vulnerability detection. To ensure security through evaluation, several studies emphasise the importance of implementing systematic approaches, such as lifecycle-based models, automated tools, and standardised methodologies. The practical applicability of penetration testing has also been reinforced by the incorporation of sophisticated tools like vulnerability scanners, exploitation frameworks, and operating system-based toolkits. Additionally, current research trends indicate a growing preference for automation and intelligent testing methods, enabling the detection of security flaws more quickly and effectively. SQL injection, cross-site scripting (XSS), and cross-site request

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering.
ISBN: 978-81-993813-1-5

forgery (CSRF) are common vulnerabilities that can be identified using well-known approaches such as standardised testing guides and structured assessment procedures. These developments show how penetration testing is a dynamic and essential part of cybersecurity. The goal of this review paper is to present a thorough analysis of the body of research on penetration testing, including its approaches, tools, applications, and emerging developments. The paper aims to demonstrate the importance of penetration testing for enhancing system security and to suggest avenues for further research by combining results from several studies.

1.1 Penetration Testing Framework and Standards

NIST SP 800-115

The NIST Special Publication 800-115 (Technical Guide to Information Security Testing and Assessment) is one of the most commonly used frameworks in research on penetration testing [11]. This guideline, published in 2008, provides a structured approach to security testing, including planning, execution, and post-test analysis. Several studies in this review specifically adopt or recommend NIST SP 800-115 as a foundation for their testing methodologies [4, 11, 12].Koswara et al. [4] demonstrated that NIST-guided penetration testing works in practice by applying it to a library website to identify SQL injection vulnerabilities and other web-based threats. Using these kinds of standard rules makes security assessments more credible and repeatable, thereby making them easier to defend in organisational and compliance contexts.

OWASP Testing Methodology

The Open Web Application Security Project (OWASP) Web Security Testing Guide [13] offers a thorough approach specifically for web application security. The OWASP framework addresses numerous vulnerabilities, such as SQL injection (SQLi), cross-site scripting (XSS), and cross-site request forgery (CSRF). It is especially useful in rapidly evolving threat landscapes due to its open-source nature and community-driven development. OWASP-based testing consistently identifies the most common web application vulnerabilities, according to the review [6, 13]. Particularly in large-scale web applications, the combination of automated scanning tools and OWASP guidelines results in higher vulnerability detection rates than manual testing alone as per Fig 1.

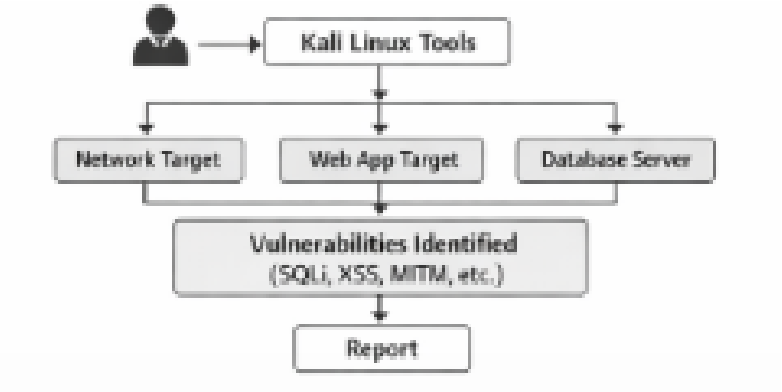

Fig 1: General Architecture of Penetration Testing Process

Lifecycle-based Approach

Several studies describe penetration testing as a phased lifecycle rather than a one-time event. Four canonical phases were identified by Bacudio et al. [8]: planning and reconnaissance, scanning, exploitation, and reporting. This lifecycle model offers a methodical framework that

reduces the risk of oversight and enhances the thoroughness of security assessments; it is also reflected in the Metasploit-based study by Tandon and Pandey [3]. This lifecycle model was expanded to enterprise settings by Al-Duwairi and Alshammari [15], who emphasised the importance of structured reporting and post-exploitation documentation as necessary conditions for successful remediation. Their work supports the idea that penetration testing is an organisational process that requires open communication between security teams and management, rather than a purely technical exercise as per Fig. 2.

Fig 2:- Lifecycle Model of Penetration Testing

Penetration Testing Tools & Techniques

Metasploit Framework

The most often noted tool in the examined literature was the Metasploit Framework. Metasploit's capabilities were thoroughly evaluated by Tandon and Pandey [3] using three different testing modalities: black-box (no prior knowledge of the system), white-box (full knowledge), and grey-box (partial knowledge). Their results verify that Metasploit greatly speeds up the exploitation stage, allowing security experts to more accurately and effectively validate vulnerabilities.

Metasploit can be used across various target environments thanks to its modular architecture, which supports a wide range of exploit modules, payloads, and auxiliary tools. As partial knowledge enables testers to concentrate resources on high-risk components without the blind spots present in pure black-box approaches, the study found that gray-box testing using Metasploit offered the best balance between reality and coverage as per Fig. 3.

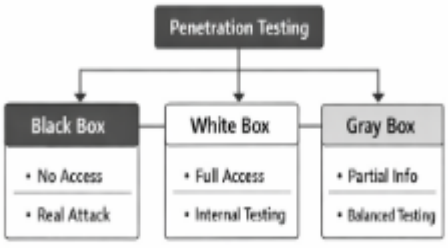

Fig 3:- Types of Penetration Testing Approach

Kali Linux Toolkit

Kali Linux, a Debian-based distribution designed specifically for penetration testing [5], includes a wide range of security tools. Denis et al. [2, 5] demonstrated how well Kali Linux-based tools can replicate network attacks, including packet sniffing and man-in-the-middle (MITM) scenarios. These tests validated the efficacy of countermeasures such as network segmentation and encrypted communication protocols, and provided important insights into network vulnerabilities.The Kali Linux toolkit is now the de facto standard for both professional and educational penetration testing environments due to its depth and accessibility. It offers thorough coverage across network, wireless, and application domains through its integration with programs such as Wireshark, Nmap, Aircrack-ng, and Burp Suite.

Automated Vulnerability Scanning

The effectiveness and scalability of security assessments have significantly improved thanks to automated penetration testing tools. The study by Scarfone et al. [12] evaluated automated vulnerability scanning against NIST guidelines, finding that automation substantially reduces the time required to identify known vulnerabilities. Large attack surfaces can be scanned in a matter of hours by programs like Nessus, OpenVAS, and Nikto—a task that would take days of manual labour. Automated tools have drawbacks despite their benefits. They frequently produce false positives and may overlook intricate, logic-based vulnerabilities that call for human judgment. The reviewed literature generally agrees that a hybrid approach that combines automated scanning for wide coverage with manual testing for depth and contextual understanding yields the best security results[6,7,12]

Vulnerability-Specific Techniques

The two most studied vulnerability categories among the reviewed studies were SQL injection and cross-site scripting [4, 6, 13]. SQL injection attacks may allow unauthorised access, modification, or deletion of data by manipulating database queries through insufficient validation of user input. Attackers can inject malicious scripts into web pages viewed by other users via cross-site scripting (XSS), making session hijacking and credential theft easier. In a comparative study of tools specifically designed for SQLi and XSS detection, Sekhar et al. [6] found significant differences in detection rates across the tools. This emphasises how crucial it is to choose tools based on the target system's unique vulnerability profile rather than relying on a single generalist tool.

1.2 Application Domain

Network Penetration Testing:-

Network penetration testing represents the most thoroughly examined area within the reviewed literature. Research by Murphy [10], Denis et al. [2, 5], and Alhamed and Rahman [7] collectively shows that simulated network attacks are highly effective at exposing structural vulnerabilities in network infrastructure. The main findings are that misconfigured network devices, unpatched software, and weak access controls are the most common ways for attacks to happen.

Murphy's NASA-based research [10] demonstrated the importance of continually monitoring network security. It showed that testing at specific times may miss vulnerabilities that come up between testing cycles. This finding supports the trend toward DevSecOps workflows that include continuous penetration testing pipelines.

Web Application Security Testing

As more businesses adopt web-based services, web application penetration testing has become increasingly important. Koswara et al. [4] and the OWASP guide [13] both stress that web applications have a different threat landscape than traditional networks. This includes client-side vulnerabilities, authentication weaknesses, and server-side injection flaws that aren't found in traditional networks.The CSRF vulnerability warrants special attention, as it is often undervalued in organisational risk assessments, despite its capacity to facilitate unauthorised state-altering actions by authenticated users [13]. For web application testing to be useful, it needs to cover both authenticated and unauthenticated attack surfaces. Automated scanning tools don't always do this.

Enterprise and Infrastructure Testing

Al-Duwairi and Alshammari [15] and Vallabhaneni and Veeramachaneni [9] concentrated on penetration testing in enterprise contexts, where complexity, scale, and regulatory requirements add dimensions not found in smaller environments. Their conclusions highlight the need for structured frameworks that support diverse system types, organisational hierarchies, and compliance requirements such as ISO 27001, PCI-DSS, and HIPAA. Additionally, scope definition and stakeholder communication present special difficulties for enterprise penetration testing. In enterprise contexts, the documentation and reporting phase which is frequently overlooked in tool-focused research emerges as a crucial differentiator that directly affects the rate and quality of remediation efforts.

Comparative Analysis of Methodologies

The studies examined demonstrate that penetration testing is becoming more mature, moving from informal, tool-driven methods to more structured, framework-guided ones. Initial research, including Scarfone et al. [11], established essential standardised guidelines. In contrast, contemporary studies by Tandon and Pandey [3] and Koswara et al. [4] illustrate the implementation of these standards in various real-world scenarios. There is a clear distinction between theoretical and practical research. Survey and review papers [1, 6, 7, 8, 11] present comprehensive overviews but may lack statistical validity, whereas experimental and case study papers [2, 3, 4, 10, 12] deliver detailed insights but may be constrained by limited applicability. A balanced research portfolio that includes both orientations is necessary for the field to move forward. The literature reviewed includes survey/review (5 papers), experimental (3 papers), framework-based (2 papers), case study (1 paper), tool-based (1 paper), and practical/analytical approaches (3 papers). This variety shows that penetration testing is a multidisciplinary field of study and a technical practice. There is a clear trend toward automation and standardisation over time. After 2020, papers consistently mention automated tools, continuous assessment models, and compliance-driven frameworks. This shows that the field is maturing and that organisations are under greater pressure to maintain strong, auditable security postures.

1.3 Findings and Emerging Trends

Key Findings

The synthesis of the literature reviewed produces several important results. Structured penetration testing frameworks consistently yield more thorough and actionable results compared to unstructured methodologies. Second, automated vulnerability scanning tools are very useful, but they need to be used with manual testing to get full coverage. Third, the web application domain is the area growing fastest in attacks, so it needs its own testing methods. Fourth, ongoing security assessment is becoming a best practice, replacing periodic point-in-time testing in environments with high risk [7, 10].

Emerging Trends

A few new trends emerge from the literature reviewed. The incorporation of artificial intelligence and machine learning into penetration testing workflows signifies an essential upcoming advancement. AI-assisted tools can learn from past penetration tests to prioritise high-value attack paths. This reduces the time required for the reconnaissance and exploitation phases [7]. Another big trend is DevSecOps integration, which adds penetration testing to continuous integration and continuous deployment (CI/CD) pipelines. This shift from periodic assessments to continuous security validation fundamentally changes how businesses think about security testing. They now see it as an ongoing process instead of a way to check compliance. Because cloud infrastructure is dynamic and temporary, traditional network-centric approaches need to be significantly modified. As a result, cloud and container security testing is also receiving more attention from researchers. Shared responsibility models, API-centric attack surfaces, and infrastructure-as-code vulnerabilities absent in conventional on-premises settings are all challenges for penetration testing in cloud environments.

2.Methodology

This systematic review was conducted in accordance with the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework, aligning with methodologies used in previous literature surveys in this field [7]. The principal research inquiries guiding this review are: Q1:- What are the most popular methods and frameworks for penetration testing? Q2:- What tools are most commonly used in research and real-life situations? Q3:- What are the most talked-about vulnerabilities and environments? Q4:- What are the limitations of existing research Fifteen studies were chosen because they were relevant to penetration testing, were published in indexed journals or conference proceedings, and covered various methodological or application aspects. The chosen papers cover the period from 2008 to 2025, ensuring that both old and new points of view are included. The papers were examined based on their topic of study, year of publication, methodology, model type, domain of representation, and principal conclusions.

Summary and Conclusion

To provide a comprehensive picture of methodological development, tool adoption, and application domain coverage in the field, this systematic review examined 15 penetration testing studies published between 2008 and 2025. The analysis demonstrates that penetration testing has evolved from an unstructured, expert-driven activity into a disciplined field supported by robust frameworks, uniform standards, and increasingly advanced tools.

The review's main conclusions are as follows: (1) Framework-based and standards-compliant testing approaches (NIST SP 800-115, OWASP) consistently produce better results than ad hoc methods. (2) The Kali Linux toolkit and Metasploit continue to be essential for conducting useful penetration tests in a variety of network and application domains. (3) Automation improves testing effectiveness, but it cannot take the place of human knowledge for sophisticated, contextual vulnerabilities. (4) Because of the distinct threat landscape of contemporary web platforms, web application security testing calls for specialised methodologies. (5) In high-risk business settings, continuous assessment models are becoming more popular than periodic testing.The development of cloud-native testing frameworks, the standardisation of reporting formats to enhance cross-organisational comparability, the integration of AI and machine learning into penetration testing automation, and longitudinal studies evaluating the long-term effects of routine penetration testing on organisational security posture should be the main areas of future research. Penetration testing will remain a crucial part of an overall cybersecurity strategy as cyber threats grow in complexity and scope.

References

[1] Vats, P., Mandot, M., & Gosain, A. (2019). A Comprehensive Literature Review of Penetration Testing & Its Applications. Proceedings of ICRITO 2020. DOI: 10.1109/ICRITO48877.2020.9197961.

[2] Denis, M., Zena, C., & Hayajneh, T. (2016). Penetration Testing: Concepts, Attack Methods, and Defence Strategies. Proceedings of the IEEE LISAT Conference.

[3] Tandon, P., & Pandey, R. (2025). A Study on Penetration Testing Using Metasploit Framework. International Journal of Research in Advanced Science and Technology, 7(5), 23–29.

[4] Koswara, W., Setiawan, R., & Nugroho, S. (2024). System Security Testing Using Penetration Testing (Library Website Case). International Journal of Engineering and Technology, 12(3), 78–84.

[5] Denis, M., Zena, C., & Hayajneh, T. (2016). Penetration Testing using Linux Tools: Attacks and Defence Strategies. Proceedings of the IEEE LISAT Conference.

[6] Sekhar, K.R., Bharti, K., & G., D. (2017). A Systematic Review of Vulnerability Analysis & Penetration Testing Tools. International Journal of Engineering and Technology (IJET), 9(2), 15–22.

[7] Alhamed, M., & Rahman, M.M.H. (2023). A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. Applied Sciences, 13, 6986. https://doi.org/10.3390/app13126986

[8] Bacudio, A., Yuan, X., Chu, B., & Jones, M. (2011). An Overview of Penetration Testing. International Journal of Network Security & Its Applications, 3, 19–38. DOI: 10.5121/ijnsa.2011.3602.

[9] Vallabhaneni, R., & Veeramachaneni, V. (2024). Understanding Penetration Testing for Evaluating Vulnerabilities and Enhancing Cyber Security. Engineering and Technology Journal, 9. DOI: 10.47191/etj/v9i10.12.

[10] Murphy, B.F. (2013). Network Penetration Testing and Research. NASA Technical Reports Server.

[11] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical Guide to Information Security Testing and Assessment. NIST Special Publication 800-115. National Institute of Standards and Technology.

[12] Yadav, S. K., Sharma, K., & Arora, A. (2018). Security integration in ddos attack mitigation using access control lists. International Journal of Information System Modeling and Design (IJISMD), 9(1), 56-76.
[13] OWASP Foundation. (2021). OWASP Web Security Testing Guide (WSTG). Open Web Application Security Project.
[14] Weidman, G. (2014). Penetration Testing: A Hands-On Introduction to Hacking. No Starch Press, San Francisco.
[15] Al-Duwairi, B., & Alshammari, E. (2015). A Survey of Network Vulnerability Assessment and Penetration Testing. International Journal of Computer Applications, 120(2), 1–7.