The Evolution of Penetration Testing in the Era of AI

Errol Baloyi, Mpho Letshwenyo, Mamello Mtshali and Alex Ramantswana
Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa

EBaloyi2@csir.co.za
MLetshwenyo@csir.co.za
MMtshali3@csir.co.za
ARamantswana@csir.co.za

Abstract: Over the past several decades, penetration testing has transitioned from a predominantly manual, expert-driven activity to a mature discipline supported by automation, modular frameworks, and artificial intelligence (AI)-assisted tools. This study provides a descriptive review of the historical evolution of penetration testing tools, highlighting the major technological and methodological advancements that have shaped the field. In addition, a practical comparative evaluation of two widely used tools, Burp Suite Professional and the Open Worldwide Application Security Project (OWASP) Zed Attack Proxy (ZAP) was conducted using a controlled vulnerable web application, Damn Vulnerable Web Application (DVWA), to assess their performance and usability in a realistic testing environment. The study further examines the impact of AI on the contemporary and emerging landscape of penetration testing tools. The findings suggest that AI is augmenting existing tools through enhanced automation and more effective vulnerability identification, while simultaneously enabling new paradigms in both offensive and defensive cybersecurity practices. This work contributes to the understanding of the evolving role of penetration testing in an AI-influenced context and discusses the implications of these developments for researchers, practitioners, and tool developers.

Keywords: Artificial intelligence, Penetration testing, Cybersecurity, Vulnerability assessment

1. Introduction

Today, amid the digital era, cyberspace is constantly under threat of being compromised by malicious actors. Often referred to as cyber threat actors (CTAs), these individuals or groups intentionally cause harm or engage in malicious activities within cyberspace. However, not all threat actors are the same, they differ in terms of motivation, resources, and skill. This distinction is illustrated by Sailio, Latvala and Szanto (2020), who categorize threat actors as nation-state sponsored, cybercriminals, ideologically driven attackers, insider threats, or thrill seekers. According to Ravindran and Potukuchi (2022), the number of threat actors around the world is likely to continue growing due to the increasing number of web applications. Furthermore, the exponential growth of the Internet has worsened the situation, with the emergence of large language models (LLMs) which have further complicated the cyber landscape.

The arrival of new and advanced applications, while rich in features, has also increased system vulnerabilities. Which according to Deng et al (2024), securing these systems has become a formidable challenge. As a result, offensive security methods such as vulnerability assessment, penetration testing (VAPT) have become essential components of the cybersecurity lifecycle. To fully grasp the concept of VAPT, it is essential to first understand what constitutes a vulnerability in a system. According to Khera et al (2019), a vulnerability is a flaw or loophole in an application that allows a threat actor to gain unauthorized system privileges and access sensitive data, which can then be exploited for malicious purposes. Similarly, Goel and Mehtre (2015) define a vulnerability as a weakness, either due to a bug in implementation or a design flaw that enables an attacker to harm users and escalate their access privileges.

1.1 Vulnerability Assessment vs. Penetration Testing

Today, many institutions often find themselves in situations where they are required whether by law, stakeholders, or customers to secure their systems and data against a growing number of vulnerabilities. This has made the use of VAPT increasingly necessary. According to Khera et al (2019), VAPT helps organizations proactively discover and identify vulnerabilities before attackers can exploit them. Rahman (2025) further emphasises that VAPT enables organizations to detect security loopholes that could otherwise be leveraged by malicious actors to launch attacks against their systems.

Thus far, in the current study, vulnerability assessment and penetration testing have been used interchangeably, however, they differ in scope and purpose. According to Khera et al (2019), vulnerability assessment is the process of identifying and discovering weaknesses or loopholes in a system using automated tools such as Nessus. These vulnerabilities can provide potential entry points for attackers. In contrast, penetration testing (or pentesting) is typically a manual process that follows a vulnerability assessment and involves actively

exploiting the identified weaknesses in a controlled and systematic manner usually after obtaining proper authorization from the organization. Nevertheless, for the purposes of this study, the term "penetration testing" is used to collectively refer to both vulnerability assessment and penetration testing.

1.2 The Importance of Penetration Testing in Cybersecurity

Penetration tests are conducted to identify vulnerabilities within a system and to determine how they can be mitigated. These tests involve simulating various types of attacks on the target system, effectively mimicking the behaviour of a malicious actor (Alhamed and Rahman, 2023). Vulnerabilities across different information assets are assessed either manually or through automated tools (Moreno et al, 2025). It is important to clarify that a penetration test is performed by a penetration tester (pentester), not a threat actor (Božić, Penevski and Adamović, 2019). The pentester is typically tasked with identifying security flaws in areas such as web applications, network configurations, or the overall infrastructure design of the organization that has engaged their services. According to Khera et al (2019), many organizations suffer from poorly designed software and hardware, as well as system misconfigurations.

As a result, penetration testing is a critical practice not only does it help address these specific issues, but it also acknowledges the reality that achieving a completely vulnerability-free system is nearly impossible. However, by identifying and eliminating as many vulnerabilities as possible through penetration testing, organizations can significantly strengthen their security posture (Khera et al, 2019; Goel and Mehtre, 2015). For instance, the 2013 Adobe security breach, which exposed data of approximately 153 million users (BBC, 2025). The leaked data included user IDs, names, and encrypted information such as passwords and credit card details. In response, Adobe issued a white paper in which it assured clients of its commitment to data security. The company emphasized that it was conducting regular penetration testing of its products and services to identify and fortify weak points (Adobe, 2024). Since then, Brown (2025) notes that this ongoing testing strategy has helped Adobe consistently reinforce its defences, preventing any further major breaches of comparable scale.

2. Research Methodology

This paper adopts a descriptive review approach to analyse the evolution of penetration testing tools from before the year 2000 through to the present. Tools were selected based on their historical significance, widespread usage, and representation of emerging trends in penetration testing, especially as influenced by AI in recent years. Each tool was assessed using a consistent framework, which included the following criteria: creator, description, primary use, and licensing. Sources included official tool documentation, peer-reviewed publications, cybersecurity blogs, community forums, and public repositories such as GitHub. The review spans more than two decades to highlight shifts in design philosophy, usage patterns, and the growing integration of AI in modern tools.

To supplement the review, a basic practical assessment was conducted using a controlled testbed. Burp Suite Professional and OWASP ZAP were selected for the hands-on comparison due to their prominence in web application security testing and their contrasting licensing models (commercial versus open source). Furthermore, Burp Suite Professional incorporates vendor-supported AI integration to enhance vulnerability analysis and automation, whereas OWASP ZAP lacks native AI functionality and instead depends on third-party or community-developed extensions to achieve comparable capabilities. The widely used DVWA platform was employed as the target environment, chosen for its reproducible vulnerabilities and compatibility with the scanners.

3. The Genesis and Evolution of Penetration Testing

The term hacking originated in the 1960s at the Massachusetts Institute of Technology (MIT), specifically within the Tech Model Railroad Club (TMRC), where members sought to "hack" circuits to enhance the performance of their model trains (Whitaker and Newman, 2006). Over time, the term evolved to describe activities involving unauthorized access to computer systems. Despite this shift in association, the initial push for the development of advanced computer penetration techniques came from the United States Department of Defence (Hunt, 2012).

At that time, computers were not yet widespread, and the concept of pentesting was largely confined to military and intelligence communities, where it was known as “red teaming” a practice involving controlled, adversarial simulations to evaluate defensive capabilities (Horkan, 2025). A significant turning point occurred in the late 1960s during the Joint Computer Conference, where the need to assess digital systems for security vulnerabilities was formally recognized within both academic and operational contexts. This period also

coincided with the military's use of national time-sharing computer systems, further reinforcing the importance of such security evaluations (Hunt, 2012).

In 1969, Advanced Research Projects Agency Network (ARPANET), a United States defence program was launched as a precursor to the modern Internet, funded by the Advanced Research Projects Agency (ARPA), later renamed Defence Advanced Research Projects Agency (DARPA). ARPANET fostered a culture of exploration, which contributed to the emergence of early hacking practices (DARPA, 2025). This period also saw the formation of "Tiger Teams," elite units within the National Security Agency (NSA) and the Department of Defence, tasked with attempting to breach United States military computer systems (Horkan, 2025). These efforts are widely regarded as the first instances of penetration testing, with the Tiger Teams recognised as the earliest practitioners of the discipline (Palatty, 2025; Hunt, 2012; Schaefer, 2004).

Furthermore, these teams served as early prototypes for the red team/blue team model that is now widely employed in modern cybersecurity exercises (Palatty, 2025). However, as the field was still in its infancy, the "Tiger Teams" experienced limited success. This prompted the development of the 1972 report by J.P. Anderson, which provided detailed step-by-step guidelines to enhance the effectiveness of such teams. Remarkably, many of the procedures outlined in Anderson's 1972 report continue to form the foundational framework for contemporary penetration testing practices (Palatty, 2025; Hunt, 2012; Schaefer, 2004).

3.1 Early Days: Manual Exploitation and Custom Scripts

The expansion of personal computing and enterprise networks during the 1980s and 1990s corresponded with a growing need for systematic approaches to security testing. Penetration testing, once confined to classified government operations, increasingly entered the private sector in response to the emerging vulnerabilities associated with networked systems and the early Internet (Horkan, 2025). The introduction of some of the penetration testing tools discussed below marked a critical development in the evolution of the practice.

Table 1: Pre-2000s Tools

Tool	Creator	Description	Primary Use	Licensing
Internet Security Systems Scanner (ISS Scanner)	Developed by Christopher Klaus in 1992 at the Georgia Institute of Technology and acquired by IBM in 2006.	A pioneering tool designed to identify security vulnerabilities within networked systems.	Network testing, providing administrators insights into potential weaknesses.	Commercial; integrated into IBM's security offerings. References: (Cummings, 2025; NBC News, 2006).
Security Administrator Tool for Analysing Networks (SATAN)	Developed by Dan Farmer and Wietse Venema, and released on April 5, 1995, it was later superseded by the MITRE CVE system.	A software vulnerability scanner with web interface and context-sensitive tutorials, using signatures to identify common network-related problems.	Automated identification of known network-exploitable vulnerabilities.	Open source. References: (Network Encyclopaedia, 2024).
Ethereal/Wireshark	Ethereal was developed by Gerald Combs in 1997 and renamed Wireshark in 2006.	Network protocol analyser with deep packet inspection capabilities for real-time traffic capture and interception.	Analysing virtual network traffic in VPCs; identifying malicious activities and abnormal network behaviours.	Licensed under the GNU General Public License, version 2. References: (Anand and Singh, 2021; Nadimpalli et al, 2025; O'Neill, 2025).
Nmap or Network Mapper	Developed by Gordon "Fyodor" Lyon in 1997.	Is a tool used to scan networks to find hosts that are running on a network, identify the services within that network such as a web server and discover security vulnerabilities	Network discovery and security auditing.	Open source. References: (Adam, Widyawan and Putra, 2023; Lyon, 2025; Singh et al, 2024).

Tool	Creator	Description	Primary Use	Licensing
Snort	Developed by Martin Roesch, founder of Sourcefire in 1998, and acquired by Cisco in 2013.	A network intrusion detection and prevention system (IDPS) with real-time traffic analysis, packet logging, and detection capabilities.	Network testing (intrusion detection/prevention); detecting probes, attacks, buffer overflows, and port scans.	Licensed under the GNU General Public License (GPLv2+) and actively maintained by Cisco. References: (Lenaerts-Bergmans, 2025).

3.2 The Rise of Frameworks and Automation (2000s - Early 2010s)

The early 2000s introduced a more systematic and standards-driven era of penetration testing. The creation of the Open Web Application Security Project (OWASP) in 2001 played a pivotal role in this transition, offering guidance for testing web applications, specifically with the release of OWASP first penetration testing framework in 2003 (Palatty, 2025). Consequently, the current study details below the pen-testing tools that emerged during that period.

Table 2: 2000s-Early 2010s Tools

Tool	Creator	Description	Primary Use	Licensing
Open-Source Security Testing Methodology Manual (OSSTM)	Developed by Pete Herzog in 2001 at the Institute for Security and Open Methodologies.	It is a comprehensive, peer-reviewed methodology for operational security assessments covering five core channels.	Vulnerability and security assessments, penetration testing, ethical hacking.	Open source. References: (Raynaud, 2025).
Nessus	Developed by Renaud Deraison in 1998 and commercialized by Tenable Network Security in 2002.	An automated vulnerability scanning tool that detect security flaws in networks, servers, applications, and cloud environments.	Penetration testing, compliance auditing (PCI DSS, HIPAA, CIS), continuous security monitoring.	Commercial and open-source alternative is OpenVAS. References: (Deraison, 2025; Raynaud, 2025).
Metasploit Framework	Developed by H.D. Moore in 2003 and acquired by Rapid7 in 2009.	A penetration testing framework with extensive database of exploits, payloads, and post-exploitation modules.	Network testing, web application testing, exploitation, post-exploitation, security validation.	Primarily open source (BSD license); commercial editions: Metasploit Express and Pro. References: (Chipeta, 2025; Kennedy et al, 2011; Nadimpalli et al, 2025).
Burp Suite	Developed by Dafydd Stuttard in 2003 and later evolved by PortSwigger, Dafydd Stuttard's security company.	A comprehensive web application security testing suite acting as intercepting proxy with scanning, spidering, and attack capabilities.	Web application penetration testing, and vulnerability assessments.	Proprietary; free Community Edition and paid Professional/Enterprise editions. References: (PortSwigger, 2025).
Acunetix	Developed by Nick Galea in 2005.	Specialized web application security testing tool with automated scanning for SQL Injection, XSS, and other vulnerabilities.	Web application testing and auditing.	Commercial. References: (Acunetix, 2025; Briffa, 2025).
Aircrack-ng	Developed by Thomas 'Mister X' D'Otreppe in 2006.	It is a collection of tools for assessing Wi-Fi network security, monitoring, and cracking encryption.	Wireless network security testing and assessment.	Open source. References: (Cooper, 2025; Hurley et al, 2007).

3.3 Specialization and Cloud Integration (Mid-2010s - Present)

Since the mid-2010s, penetration testing has undergone a pivotal transformation driven by cloud adoption, DevOps practices, and increasingly sophisticated cyber threats fuelled by growing Internet usage. This shift emphasized established security frameworks such as the Payment Card Industry Data Security Standard (PCI DSS), to become increasingly critical amid rising online fraud (Fitzgerald and Rubbinaccio, 2025; Barney, 2025). Tool development evolved along three paths: legacy tools adapted for cloud use, cloud-native tools addressing risks like Infrastructure as Code (IaC) misconfigurations, and evasion techniques targeting cloud-native defences (Morris, 2025).

Penetration testing frameworks matured to guide methodologies and tool selection, with the OWASP Testing Guide endorsing tools like Burp Suite for API testing and WPScan for CMS vulnerabilities. Standards such as PTES structured assessments through phases like threat modelling and post-exploitation, leveraging tools like Nmap and Metasploit, while the NIST Cybersecurity Framework standardized compliance with tools like Nessus (Adam, Widyawan and Putra, 2023). As cloud environments grow in complexity, tools continue to specialize, spanning cloud-native scanners, cloud-adapted legacy solutions, payload generators, and advanced evasion tools (Mallick and Nath, 2024).

Table 3: Mid-2010s - Present Tools

Tool	Creator	Description	Primary Use	Licensing
Scout Suite	Developed by NCC group.	It is a multi-cloud security auditing tool using APIs to collect configuration data and generate HTML reports categorizing findings by severity.	Security posture assessments across AWS, Google Cloud, and Microsoft Azure; investigating security features across cloud services.	Licensed under the GNU General Public License v2 (GPLv2). References: (Hossam, 2025; Kesseli, 2025).
WPScan	Developed by the WPScan Team and now owned by Automattic Inc.	Vulnerability assessment tool for WordPress with highest GitHub metrics.	WordPress-specific security assessments including plugins, themes, and error log exposure.	Licensed under a dual-license model called the WPScan Source License. References: (Farrell, 2022).
Phantom Evasion	Developed by oddcod3 and archived in a GitHub repository.	Antivirus evasion tool generating fully undetectable payloads via obfuscation and reducing detection.	Payload generation for cloud penetration testing; limited by cloud behavioural analysis and runtime protection.	Licensed under the GNU General Public License version 3 (GPL3). References: (Adam and Sufyanu, 2021; Tigner, Wimmer and Rebman, 2021).
MSFVenom	Part of Metasploit Framework	Payload generation tool creating cloud-optimized payloads that evade defences through encrypted reverse TCP sessions.	Bypassing sandboxed cloud VMs.	Part of Metasploit (BSD license). References: (Pandey, 2025; Tigner, Wimmer and Rebman, 2021).

4. Penetration Testing Tools in the Age of AI

The current study has so far provided a detailed examination of traditional penetration testing tools. While these methods have been effective in the past, they often struggle to keep up with today's rapidly evolving threat landscape as also noted by (Thummala, Gupta and Siddharth, 2025). In response to this challenge, automated penetration testing has emerged as a viable solution, enabling quicker and more consistent scanning and testing of networks, applications, and systems for vulnerabilities. Accordingly, the following section explores the implications of pentesting tools in the era of AI. However, it is important to note that the tools mentioned below do not represent the full range of tools available. As Goel and Mehtre (2015) points out, there are numerous open-source and commercial pentesting tools currently on the market. These tools are developed by industry experts with the aim of delivering more efficient, dynamic, and comprehensive security assessments (Božić, Penevski and Adamović, 2019).

4.1 AI as an Augmentation of Existing Tools

AI in penetration testing has changed traditional methods of testing by improving existing tools through methods like automated vulnerability prioritization, adaptive threat simulation, and predictive analytics. This innovation helps in fixing issues that exist in traditional methods like high false positives, scalability issues in complex environments and inconsistent workflows. AI-powered tools like Tenable.io prioritize vulnerabilities by linking threat intelligence exploit feasibility, and asset criticality; this in turn reduces manual triage efforts (Hossam, 2025). Integrated tools like AIPenPool combine legacy tools like Nmap and Metasploit with Machine Learning (ML) algorithms, this improves detection accuracy while reducing false positives by using verification algorithms that compare vulnerability data against historical records (Kumar et al, 2025).

AI improves offensive capabilities by simulating advanced persistent threats (APTs). Tools like Microsoft Security Copilot mimic complex attack patterns by learning from past breaches. This allows for dynamic red-team exercises that respond to defensive actions in real-time (Hossam, 2025). Platforms like Cyberwheel automatically run attack chains that follow the MITRE ATT&CK framework by using reinforcement learning (RL). They choose methods based on the network's weaknesses and its current state. This is an improvement from scripted tools like MITRE Caldera, which uses various hacking tools but does not have the ability to make decisions on their own (Kumar et al, 2025). RL agents and Cyberwheel can apply their capabilities in different situations, they maintain strong performance even with new network setups, something traditional automated tools struggle to do.

Operational efficiency is also improved through AI integration, automated workflows seamless chain tools, reducing manual intervention. This makes security accessible to non-experts through user-friendly interfaces, it also increases the capability of organizations without sacrificing depth. For instance, AIPenTool's guided attack suggestions enable users to execute complex tests like SQL injection or SSL vulnerability assessments with minimal expertise (Kumar et al, 2025). Despite these improvements, challenges still exist. Due to noisy inputs reducing the accuracy of ML models, data quality is crucial. AI tools are still not widely used in the real world. Simulators like Cyberwheel have not yet gone through extensive testing in real-world settings. Additionally, there are some ethical concerns that have been raised, requiring frameworks like Safe Reinforcement Learning to prevent uncontrolled autonomous attacks. Semi-autonomous systems allow AI to suggest attack paths while needing human approval, balancing autonomy with ethical oversight (Fernandes et al, 2025). Additionally, integrating AI with older tools requires a lot of changes, especially for real-time adaptation to unknown vulnerabilities. However, below the current study provides a comparison of an AI integration versus non-AI approaches in automated web application testing tool.

4.2 Practical Demonstration: AI vs Non-AI Tool

As outlined in the methodology, the present study conducts a comparative analysis of two web application testing tools: Burp Suite Professional, a widely used commercial solution, and OWASP ZAP 2.16.1, an open-source counterpart. A summary of the comparative results is presented in Figure 1.

BURP

Severity	Confidence: Certain	Firm	Tentative	Total
High	5	1	0	6
Medium	4	7	1	12
Low	1	0	0	1
Information	36	7	0	43
False Positive	0	0	0	0

ZAP

Risk	Confidence: User Confirmed	High	Medium	Low	Total
High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Medium	0 (0.0%)	2 (18.2%)	2 (18.2%)	0 (0.0%)	4 (36.4%)
Low	0 (0.0%)	1 (9.1%)	3 (27.3%)	0 (0.0%)	4 (36.4%)
Informational	0 (0.0%)	1 (9.1%)	2 (18.2%)	0 (0.0%)	3 (27.3%)
Total	0 (0.0%)	4 (36.4%)	7 (63.6%)	0 (0.0%)	11 (100%)

Figure 1: VA Scan Results

The results presented in Figure 1 are derived from basic automated scans. Both tools demonstrated strong performance. However, Burp exhibited superior detection capabilities compared to ZAP, which only identified 11 vulnerabilities, whereas Burp Suite detected a total of 61. Similar findings were reported by Albalawi et al (2023), who observed that ZAP generally reports fewer vulnerabilities than Burp. Nonetheless, Albalawi et al

(2023) also noted that Burp scans typically take longer, a pattern the present study also observed. Similarly, Deeb (2024) also employed DVWA as a testbed to evaluate several leading web application security tools, including ZAP and Burp Suite, and concluded that Burp Suite achieves an excellent balance between detection accuracy and resource utilization.

4.3 The Dual-Edged Sword: AI for Attackers

AI or ML, while critical in strengthening cybersecurity defences, also pose a pernicious threat by empowering threat actors to conduct increasingly sophisticated, AI-generated exploits (Lewis et al, 2025). According to Stanham (2025), threat actors are leveraging several key strategies to accelerate and enhance their attacks. One such strategy is the use of automated and adaptive social engineering, where AI is employed to craft convincing phishing emails, create fake websites, generate deepfake audio and visuals, inject malicious prompts or code, and carry out chatbot-driven fraudulent activities. Another method involves the evasion of security mechanisms.

Stanham (2025) further notes that attackers are using AI to bypass CAPTCHAs, poison security data models, and exploit malicious generative pre-trained transformers (GPTs) to undermine traditional defence systems. Additionally, AI-assisted vulnerability discovery, as highlighted by Kerner (2025), enables the rapid identification of software and system flaws through the analysis of large volumes of data, significantly accelerating the exploitation process. Finally, AI continues to enable threat agents to redefine and adapt their strategies in real time, posing an escalating challenge to existing cybersecurity frameworks.

5. Conclusion and Future Work

The findings of this study demonstrate that the process of conducting VAPT has evolved substantially since the early stages of information security and the tools employed for VAPT have also advanced in parallel. Previously, such practices were confined to the military and defence organizations due to the sensitive nature of their operations. However, with the expansion and increasing sophistication of the threat landscape, cybersecurity has become a fundamental concern across a wide spectrum of organizations. Moreover, the integration of AI has further transformed VAPT tools. This evolution is evident in the comparative assessment conducted in this study, where Burp, a tool incorporating AI-driven capabilities, identified a greater number of vulnerabilities compared to ZAP. Consequently, future research should prioritize the development of ethically aligned AI systems for cybersecurity, alongside the creation of frameworks that ensure accountability and transparency in AI-assisted penetration testing. Finally, future work will expand this study by incorporating additional web security assessment tools and benchmarking them against standardized evaluation metrics to enable a more comprehensive comparative analysis.

Ethics declaration: Ethical clearance was not required, as this study did not involve human participants.

AI declaration: AI was not utilised in this study.

References

Acunetix. (2025) “Acunetix - Crunchbase Company Profile & Funding”, [online], https://www.crunchbase.com/organization/acunetix.

Adam, A.S., and Sufyanu, Z. (2021) “Performance comparison of PyRAT and Phantom antivirus software evasion tools”, *Journal of Information Security*, Vol. 2, No. 1.

Adam, H.M., Widyawan and Putra, G.D. (2023) “A review of penetration testing frameworks, tools, and application areas”, *IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE 2023),* pp 319-324.

Adobe. (2024) “Adobe Application Security Overview”.

Albalawi, N., Alamrani, N., Aloufi, R., Albalawi, M., Aljaedi, A., & Alharbi, A. R. (2023) “The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities”, *Electronics*, Vol. 12, No. 12.

Alhamed, M. and Rahman, M.M.H. (2023) “A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions”, *Applied Sciences*, Vol. 13, No. 12, p 6986.

Anand, P. and Singh, A.S. (2021) “Penetration testing security tools: A comparison” *2021 10th International Conference on System Modeling and Advancement in Research Trends (SMART 2021),* pp 182–184.

Barney, N. (2025) “What is PCI DSS? Requirements and Compliance”, [online], https://www.techtarget.com/searchsecurity/definition/PCI-DSS-Payment-Card-Industry-Data-Security-Standard.

BBC. (2025) “Adobe hack: At least 38 million accounts breached”, [online], https://www.bbc.com/news/technology-24740873.

Božić, K., Penevski, N. and Adamović, S. (2019) “Penetration Testing and Vulnerability Assessment: Introduction, Phases, Tools and Methods”, *SINTEZA 2019*, pp 229-234.

Briffa, M. (2025) “Maltese Success Story: Acunetix World Leading in Cybersecurity” [online], https://newtech.mt/blog-news/maltese-success-story-acunetix-world-leading-in-cybersecurity/.
Brown, C. (2025) “Penetration Testing Examples: Real-World Scenarios & Insights”, [online], https://www.vikingcloud.com/blog/penetration-testing-examples.
Chipeta, C. (2025) “What is Metasploit?”, [online], https://www.upguard.com/blog/metasploit.
Cooper, S. (2025) “Aircrack-ng Review for 2024 & the Best Alternatives (Paid & Free)”, [online], https://www.comparitech.com/net-admin/aircrack-ng-review/.
Cummings, D. (2025) “Atlanta History: Notes from the ISS S-1 IPO Filing”, [online], https://davidcummings.org/2012/09/07/atlanta-history-notes-from-the-iss-s-1-ipo-filing/.
DARPA. (2025) “ARPANET”, [online], https://www.darpa.mil/news/features/arpanet.
Deeb R. (2024) “Evaluating Web Application Vulnerability Scanners: Introducing the RD-Score for Comprehensive Performance Assessment”, *International Journal of Open Information Technologies,* Vol. 12, No. 11.
Deng, G. et al. (2024) “PentestGPt: Evaluating and Harnessing Large Language Models for Automated Penetration Testing”, *USENIX Security Symposium 2024*.
Deraison, R. (2025) “Nessus Turns 20!”, [online], https://www.tenable.com/blog/nessus-turns-20.
Farrell, S. (2022) “Abstraction and automation of WordPress vulnerability scanning”, Unpublished manuscript.
Fernandes, R., Lopes, N., Goncalves, J., and Cosgrove, J. (2025) “Comparing traditional hacking tools and AI-driven alternatives”, *13th International Symposium on Digital Forensics and Security (ISDFS 2025).*
Fitzgerald, A. and Rubbinaccio, M. (2025) “PCI DSS History: How the Standard Came To Be”, [online], https://secureframe.com/blog/pci-history.
Goel, J.N. and Mehtre, B.M. (2015) “Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology”, *Procedia Computer Science*, pp 710-715.
Horkan, W. (2025) “A Brief History of Penetration Testing: From Tiger Teams to PTaaS – Horkan”, [online], https://horkan.com/2025/06/19/a-brief-history-of-penetration-testing-from-tiger-teams-to-ptaas.
Hossam, A. (2025) “Best tools for cloud penetration testing in 2025”, [Online], https://deepstrike.io/blog/best-tools-for-cloud-penetration-testing-in-2025.
Hunt, E. (2012) “US government computer penetration programs and the implications for cyberwar”, *IEEE Annals of the History of Computing*, Vol. 34, No. 3, pp 4-21.
Hurley, C., Rogers, R., Thornton, F., Connelly, D. and Baker, B. (2007) “WarDriving and Penetration Testing with Windows”, *Syngress*, pp 93-117.
Infosec. (2025) “The history of penetration testing”, [online], https://www.infosecinstitute.com/resources/penetration-testing/the-history-of-penetration-testing/.
Kennedy, D., O’Gorman, J., Kearns, D. and Aharoni, M. (2011) “Metasploit: The Penetration Tester’s Guide”, No Starch Press.
Kerner, S.M. (2025) “AI-powered attacks: What CISOs need to know now”, [online], https://www.techtarget.com/searchsecurity/feature/AI-powered-attacks-What-CISOSs-need-to-know-now.
Kesseli, T. (2025) “Security in cloud environments”, Unpublished manuscript.
Khera, Y., Kumar, D., Sujay, S. and Garg, N. (2019) “Analysis and Impact of Vulnerability Assessment and Penetration Testing”, *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*, pp 525-530.
Kumar, N.P., Vijay Kumar Gowda, K.K., and Dhanushree, M.S. (2025) “AIPenTool: A unified approach to automated penetration testing for enhanced network and web application security”, *3rd International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE 2025)*.
Lenaerts-Bergmans, B. (2025) “Snort Explained: Understanding Snort Rules and Use Cases”, [online], https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/snort-rules/.
Lewis, C., Kristensen, I., Caso, J. and Fuchs, J. (2025) “AI is the greatest threat—and defense—in cybersecurity today. Here’s why”, [online], https://www.mckinsey.com/about-us/new-at-mckinsey-blog/ai-is-the-greatest-threat-and-defense-in-cybersecurity-today.
Lyon, G. (2025) “Nmap Preface. Nmap Network Scanning”, [online], https://nmap.org/book/preface.html.
Mallick, A.I. and Nath, R. (2024) “Navigating the cybersecurity landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments”, [Online], https://worldscientificnews.com/navigating-the-cyber-security-landscape-a-comprehensive-review-of-cyber-attacks-emerging-trends-and-recent-developments/.
Moreno, A.C. et al. (2025) “Analysis of Autonomous Penetration Testing Through Reinforcement Learning and Recommender Systems”, *Sensors*, Vol. 25, No. 1.
Morris, K. (2025) “Infrastructure as Code: Designing and Delivering Dynamic Systems for the Cloud Age”, O’Reilly Media, Sebastopol.
Nadimpalli, A.V., Sai, A.S. and Kamalesh, M.D. (2025) “Analyzing and implementing cloud security and penetration testing tools: A review”, *Journal of Cloud Security*, Vol. 8, No. 2, pp 110-125.
NBC News. (2006) “IBM to acquire Internet Security Systems”, [online], https://www.nbcnews.com/id/wbna14482359.
Network Encyclopaedia. (2024) “Satan tool: Security administrator tool for analyzing networks. NETWORK ENCYCLOPEDIA”, [online], https://networkencyclopedia.com/security-administrator-tool-for-analyzing-networks-satan/.
O’Neill, T. (2025) “Wireshark – from ethereal to today. Network Visibility”, [online], https://www.garlandtechnology.com/blog/wireshark-from-ethereal-to-today.

Palatty, N.J. (2025) “A Brief History of Penetration Testing - Astra Security Blog. Astra Security Audit”, [online], https://www.getastra.com/blog/security-audit/history-penetration-testing/?utm_source=chatgpt.com.

Pandey, B., Shukla, M.M., Pandey, P., and Bakar, W.A. (2025) “Development of malware in Windows sandbox and Kali MSFvenom for hacking Android and Windows operating systems”, *International Journal of Information Technology*, Vol. 17, No. 3, pp 1831–1838.

Pautov, A. (2024) “OWASP ZAP: A comprehensive guide to web application security testing, Medium”, [online], https://medium.com/@1200km/owasp-zap-a-comprehensive-guide-to-web-application-security-testing-6c247f4be39b.

PortSwigger. (2025) “Burp Suite documentation”, [online], https://portswigger.net/burp/documentation.

Rahman, H. (2025) “Real-Time Network Traffic Monitoring System”, [online], https://scde.jntuh.ac.in/lmsportal/alumniuploads/programdocuments_29052025/Network_monitoring_system_Using_wireshark.pdf.

Ravindran, U. and Potukuchi, R.V. (2022) “A Review on Web Application Vulnerability Assessment and Penetration Testing”, *Review of Computer Engineering Studies*, Vol. 9, No. 1, pp 1-22.

Raynaud, C. (2025) “Open-Source Security Testing Methodology Manual”, [online], https://raynaudc.wordpress.com/2011/12/20/open-source-security-testing-methodology-manual-2/.

Sailio, M., Latvala, O.M. and Szanto, A. (2020) “Cyber Threat Actors for the Factory of the Future”, *Applied Sciences*, Vol. 10, No. 12, p 4334.

Schaefer, M. (2004) “If A1 is the answer, what was the question? An edgy naïf’s retrospective on promulgating the trusted computer systems evaluation criteria”, *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pp 204-228.

Singh, A., Sharma, E.S., Reddy, B.K., Soni, P., Ghuman, S.S., and Gill, U.S. (2024) “Automated network vulnerability assessment with Nmap: A comprehensive approach*”, 2nd International Conference on Advanced Computing and Communication Technologies (ICACCTech 2024)*, pp 208–214.

Stanham, L. (2025) “Most Common AI-Powered Cyberattacks. CrowdStrike”, [online], https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/ai-powered-cyberattacks/.

Thummala, V.R., Gupta, A.K. and Siddharth, E. (2025) “Enhancing Cybersecurity with AI-Powered Penetration Testing Tools”, *1st Int. Conf. on Holistic Approaches to Achieve Sustainable Development Goals with AI (ICASDGAI-2025).*

Tigner, M., Wimmer, H. and Rebman, C.M. (2021) “Analysis of Kali Linux penetration tools: A survey of hacking tools”, *International Conference on Electrical, Computer, and Energy Technologies (ICECET 2021).*

Whitaker, A. and Newman, D.P. (2006) *Penetration Testing and Network Defense*, Cisco Press.