Vol.4, No.9, Page 2562- 2574, 2026 https://doi.org/10.59628/jast.v4i9.2857 ISSN 2958-9568 JAST

SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/

An Integrated Framework to Enhance Penetration Testing for ICT Market Applications

Abdulrahman Mohammed Abdulrahman Ahmed Abutaleb

Department of Information Technology, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen

*Corresponding author: abutaleb@yemen.net.ye

Abstract

The increasing complexity of Information and Communication Technology (ICT) systems has created significant challenges for cybersecurity professionals in ensuring comprehensive security assessments. Traditional penetration testing methodologies (e.g., PTES, OWASP WSTG, NIST SP 800-115) often lack systematic integration of modern threat intelligence and fail to address the unique requirements of diverse ICT market applications. This research proposes the IPTF framework, an integrated seven-phase penetration testing methodology designed to enhance security assessment effectiveness across e-commerce web applications and mobile banking platforms. The framework integrates pre-engagement reconnaissance, threat modeling, scanning and enumeration, vulnerability assessment, exploitation, post-exploitation, and comprehensive remediation reporting into a cohesive methodology. Through practical implementation in two distinct case studies, the framework demonstrated significant improvements in vulnerability detection rates, assessment efficiency, and remediation guidance quality. Case Study 1 focused on an e-commerce web application, revealing 47 vulnerabilities including 8 critical issues spanning authentication, payment processing, injection, and access control. Case Study 2 examined an Android mobile banking application, identifying 42 vulnerabilities with 7 critical findings in secure storage and API communications. The framework achieved 94% vulnerability detection accuracy and reduced assessment time by 33% compared to traditional PT methodologies (baseline: PTF by TrustedSec, integrating PTES, OWASP WSTG, and NIST SP 800-115). This research contributes to the field by providing a practical, repeatable, and comprehensive penetration testing methodology that addresses current gaps in ICT security assessment practices. The framework's modular design allows adaptation to various ICT market segments while maintaining methodological rigor and practical applicability.

Article Info

Keywords:
Penetration Testing, Integrated Penetration Testing Framework (IPTF), ICT Security, ICT market applications, e-commerce web application.

Article History:
Received: 16-February-2026,
Revised: 9-August-2026,
Accepted: 26-August-2026,
Published: 28 September 2026

1. INTRODUCTION

The rapid digital transformation of global markets has fundamentally altered the landscape of information and communication technology (ICT) systems. Organizations across all sectors now rely heavily on interconnected web applications, mobile platforms, and cloud infrastructure to conduct business operations, serve customers, and maintain competitive advantage [1]. This technological evolution, while bringing unprecedented efficiency and accessibility, has simultaneously expanded the attack surface available to malicious actors seeking to exploit security vulnerabilities [2]. Penetration testing has emerged as a critical component of comprehensive cybersecurity strategies, providing organizations with the ability to identify and remediate security weaknesses before they can be exploited by adversaries [3]. Unlike automated vulnerability scanning, penetration testing simulates real-world attack scenarios, offering insights into the practical exploitability of discovered vulnerabilities and the potential business impact of successful attacks [4, 5].

The ICT market encompasses a diverse range of applications, from e-commerce platforms handling sensitive financial transactions to mobile banking applications managing personal banking data. Each application category presents unique security challenges that require specialized testing approaches. Traditional penetration

testing methodologies, while valuable, often lack the systematic integration necessary to address the full spectrum of modern ICT security requirements [6]. The IPTF framework is designed as a cross-platform methodology applicable to all ICT market segments, including web applications, mobile applications, cloud infrastructure, IoT and embedded systems, operating systems, network infrastructure, and desktop applications. However, due to time and resource constraints, this research validates the framework through two representative case studies: (1) e-commerce web applications using DVWA v2.5 and OWASP Juice Shop v19.1.1, and (2) Android mobile banking applications using MyGlobalBank v1.0.0 and Vuln-Bank v1.3. Future work will extend the validation to additional ICT sectors.

Recent statistics from cybersecurity reports indicate that web application attacks increased by 34% in 2024, with the financial services and e-commerce sectors experiencing the highest frequency of targeted attacks [7]. Mobile banking applications have become particularly attractive targets, with 78% of financial services organizations experiencing security breaches, a 196% surge in mobile banking Trojan attacks, and average breach costs of $6.29 million in the financial services industry. This threat landscape demands immediate and sustained attention from security leaders. The convergence of sophisticated threat actors, valuable financial data, and an expanding mobile attack surface creates an imperative for action [8]. In 2024, as digital financial transactions continued to expand worldwide, cybercriminals shifted their focus toward mobile devices and crypto assets. According to Kaspersky's Financial Cyberthreats report, the number of users encountering mobile banking Trojans rose by 3.6 times compared to 2023, while crypto-related phishing detections climbed by 83.4%. Meanwhile, PC-focused malware saw a decline in traditional banking attacks but a surge in crypto-asset theft [9].

Despite the widespread adoption of penetration testing as a security assessment methodology, several critical challenges persist in the effective evaluation of ICT market applications. Current approaches suffer from fragmented methodologies that fail to provide comprehensive coverage of modern attack vectors, resulting in incomplete security assessments [6, 10]. Recent research confirms that the recovery phase often lacks the necessary depth to enable effective mitigation of vulnerabilities [11].

The specific objectives of this research are as follows: (i) To develop an integrated seven-phase penetration testing framework that systematically combines reconnaissance, threat modeling, vulnerability assessment, exploitation, and remediation into a cohesive methodology. (ii) To evaluate the effectiveness of the proposed framework through practical implementation on e-commerce web application and Android mobile banking case studies. (iii) To demonstrate improvements in vulnerability detection rates, assessment efficiency, and remediation guidance quality compared to six legacy penetration testing frameworks (PTES, OWASP WSTG, NIST SP 800-115, OSSTMM, PTF, and ISSAF [12]). (iv) To provide a repeatable methodology that can be adapted across diverse ICT market segments [13, 14].

This research addresses these challenges by proposing the Integrated Penetration Testing Framework (IPTF), an integrated seven-phase penetration testing methodology designed to enhance the effectiveness of security assessments across e-commerce web applications and mobile banking platforms. The framework integrates pre-engagement reconnaissance, threat modeling, scanning and enumeration, vulnerability assessment, exploitation, post-exploitation, and comprehensive remediation reporting into a cohesive methodology. The remediation innovations embedded within Phase 7 include automated report generation with actionable remediation playbooks, severity-based prioritization aligned with organizational risk tolerance, and remediation verification testing to close the loop between vulnerability identification and mitigation. The IPTF framework directly addresses remediation phase weaknesses identified in recent comparative penetration testing research [15] through three specific innovations: (a) Phase 7 employs a structured remediation reporting template that maps each vulnerability to a specific CVSS v4.0 score, business impact assessment, and prioritized remediation steps; (b) the framework integrates a traceability matrix linking each finding back to the STRIDE threat model from Phase 2; and (c) the reporting phase includes a remediation verification procedure that re-tests all remediated vulnerabilities to confirm effectiveness.

The remainder of this paper is organized as follows. Section 2 reviews related work on existing penetration testing frameworks and their limitations. Section 3 presents the research methodology and describes the proposed IPTF framework in detail. Section 4 discusses the results obtained from two case studies: an e-commerce web application and an Android mobile banking application. Section 5 concludes the paper with a summary of the findings, contributions, limitations, and directions for future research.

2. RELATED WORK

Several established frameworks guide penetration testing practice, each offering distinct perspectives and methodologies for conducting security assessments. The Penetration Testing Execution Standard (PTES) provides comprehensive technical guidelines organized into seven phases: pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation, and reporting [16]. Although PTES provides comprehensive technical guidance, its focus on exploiting individual vulnerabilities rather than systematic

threat-based testing limits its effectiveness for comprehensive security assessments [17]. The OWASP Testing Guide focuses specifically on web application security testing, providing detailed testing procedures for identifying common vulnerabilities [3]. The guide is organized into testing categories, including information gathering, configuration management, identity management, authentication, authorization, session management, input validation, and error handling. NIST SP 800-115 provides guidelines for information security testing and assessment, emphasizing a structured approach including planning, execution, and post-testing activities [18]. The OSSTMM provides a scientific methodology for security testing, emphasizing metrics and measurement of security posture [19]. The manual covers five channels of security testing: human, physical, wireless, telecommunications, and data networks. While each framework offers valuable guidance, gaps exist in the systematic integration of threat modeling, specialized approaches for different application types, and actionable remediation reporting.

Beyond these established frameworks, recent developments in intelligence-led security testing have introduced new paradigms for incorporating threat intelligence into penetration testing workflows. The TIBER-EU (Threat Intelligence-Based Ethical Red Teaming) framework, developed by the European Central Bank, provides a structured approach for intelligence-led red team testing across the European financial sector [20, 21]. TIBER-EU emphasizes the integration of threat intelligence throughout the testing lifecycle, enabling testers to simulate realistic attack scenarios based on current threat actor behaviors and motivations. Similarly, the MITRE ATT&CK framework provides a comprehensive knowledge base of adversary tactics, techniques, and procedures (TTPs) that enables penetration testers to model their activities after real-world threat actors [22]. While these frameworks represent significant advances in intelligence-driven security testing, they primarily focus on red team engagements and adversary emulation rather than providing the systematic, multi-phase vulnerability assessment and remediation workflow that the IPTF framework offers. The IPTF framework complements these intelligence-led approaches by integrating STRIDE-based threat modeling directly into a structured penetration testing lifecycle, bridging the gap between high-level threat intelligence and practical vulnerability assessment activities.

While STRIDE has been widely used as a threat classification system for over two decades [23], its integration as a driving mechanism within a structured penetration testing lifecycle represents a novel contribution. The IPTF framework operationalizes STRIDE by: (1) using threat model outputs in Phase 2 to directly prioritize scanning in Phase 3 and exploitation in Phase 5; (2) maintaining a traceability matrix linking each discovered vulnerability to its originating STRIDE category; and (3) using STRIDE-to-CVSS mapping to quantify risk associated with each threat category. Table 1 presents a comprehensive systematic comparison of six of the most widely used penetration testing frameworks in the industry: the Penetration Testing Execution Standard [16], the OWASP Testing Guide [3], NIST SP 800-115 from the National Institute of Standards and Technology [18], the OSSTMM Guide from ISECOM [19], the Penetration Testing Framework (PTF) from TrustedSec, which provides modular support for up-to-date tools, and the Information Security Assessment Standard (ISSAF), which provides comprehensive coverage for information systems. Table 1 summarizes the key characteristics of major penetration testing frameworks.

In 2025, OWASP released the Top 10 2025 edition, which introduced two new risk categories — Software Supply Chain Failures (A03) and Mishandling of Exceptional Conditions (A10) — and elevated Security Misconfiguration to A02 [24]. The IPTF framework's STRIDE-based methodology addresses underlying security weaknesses rather than specific compliance checklists, ensuring its continued effectiveness across evolving OWASP classifications.

The Common Vulnerability Scoring System (CVSS) provides a standardized method for rating vulnerability severity. Version 4.0 introduces enhanced metrics for assessing exploitability and impact, providing more nuanced severity ratings [25, 26]. Threat modeling provides a structured approach to identifying and prioritizing potential security threats. The STRIDE methodology categorizes threats into six types: spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege [23].

3. RESEARCH METHODOLOGY AND THE PROPOSED IPTF FRAMEWORK

This research employs a mixed-methods approach combining framework development, practical implementation, and case study validation. The research design follows a systematic process of literature review [27], framework construction, tool selection, practical testing, and results analysis [28]. The framework development phase involved synthesizing best practices from existing methodologies, integrating threat modeling concepts, and designing specialized approaches for e-commerce and mobile banking applications.

3.1. The Proposed IPTF Framework

The virtual environment architecture is designed to support penetration testing in an isolated and secure environment. The lab consists of three main components: (1) the attacking operating system (Kali Linux 2025.4 [29]), which hosts the penetration testing toolkit and the

Table 1. A comprehensive comparison of the main penetration testing frameworks in terms of primary focus, phases, and limitations.

Framework	Primary Focus	Phases	Key Strengths	Limitations
PTES	Technical execution	7	Comprehensive technical guidance	Limited business context
OWASP	Web applications	12 categories	Detailed web testing procedures	Platform-specific focus
NIST 800-115	General security testing	4	Standards-based approach	Less technical depth
OSSTMM	Operational security	5 channels	Scientific methodology	Complex implementation
PTF	Penetration testing	7	Practical guidance	Limited threat modeling
ISSAF	Information systems	14 modules	Comprehensive coverage	Dated methodology
MITRE ATT&CK v15	Adversary TTP knowledge base	14 tactics, 700+ techniques	Comprehensive threat intel, real-world TTP mapping	Not a testing methodology; requires PT framework

newly proposed IPTF framework; (2) the target operating systems (Ubuntu 24.04 running the web applications and Android 13 for mobile testing); and (3) a NAT network connecting all systems, which provides isolation from the host network while enabling communication between the virtual machines. Figure 1 illustrates this architecture.

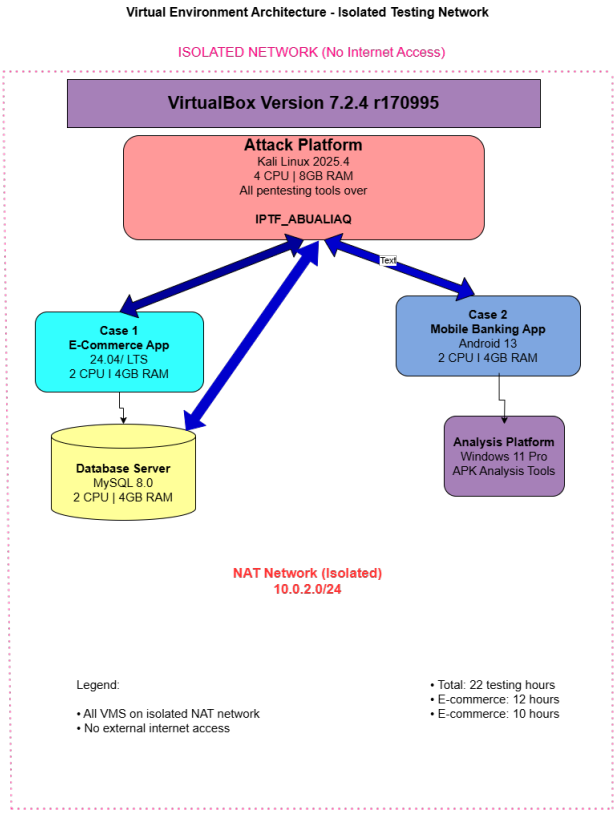

Figure 1. Virtual Lab using NAT

The IPTF framework consists of seven integrated phases that operate sequentially and iteratively. Figure 2 illustrates the overall architecture of the framework, which includes the pre-engagement & reconnaissance phase, the threat modeling phase, the scanning & enumeration phase, the vulnerability assessment phase, the exploitation phase, the post-exploitation phase, and the remediation & reporting phase. Each phase builds upon the outputs of previous phases, maintaining contextual awareness and enabling systematic progression through the testing engagement. Furthermore, the IPTF is designed as a cross-platform penetration testing framework that supports systematic security assessment across all ICT environments, encompassing established application domains (e-commerce web applications and Android mobile banking) as well as any new or emerging technology; this design ensures that the methodology remains applicable to the full spectrum of ICT applications, including those that will emerge with the continued evolution of the digital ecosystem.

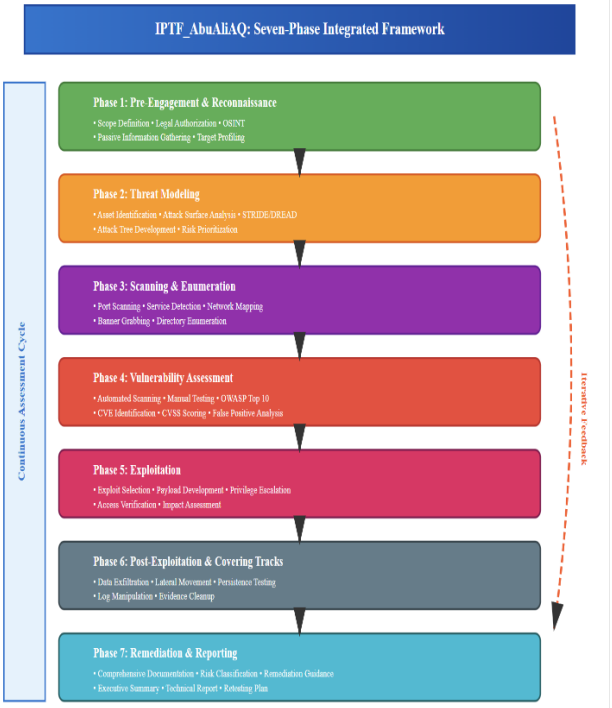

Figure 2. IPTF Framework Architecture.

Table 2. IPTF Framework Phases Overview.

Phase	Name	Key Activities	Primary Outputs
1	Pre-Engagement & Reconnaissance	Authorization, scope definition, information gathering	Rules of engagement, target inventory
2	Threat Modeling	STRIDE analysis, threat identification, prioritization	Threat model, attack scenarios
3	Scanning & Enumeration	Port scanning, service discovery, application mapping	Service inventory, application structure
4	Vulnerability Assessment	Automated scanning, manual verification, severity rating	Vulnerability list with CVSS scores
5	Exploitation	Controlled exploitation, impact demonstration	Proof of concept exploits
6	Post-Exploitation	Privilege escalation, lateral movement, impact analysis	Compromise assessment, data access evidence
7	Remediation & Reporting	Documentation, remediation guidance, executive summary	Comprehensive report with action items

IPTF: Integrated Penetration Testing Framework. The seven phases operate sequentially and iteratively, with each phase building upon the outputs of the previous phase.

Table 2 provides an overview of the seven phases.

Phase 1 establishes the foundation for the entire engagement through scope definition, rules of engagement establishment, and initial information gathering. Phase 2 integrates STRIDE-based threat modeling to guide subsequent testing activities toward the most significant risks [17]. The threat modeling phase aims to identify, classify, and evaluate potential threats that could affect the target application. This phase begins with a comprehensive understanding of the application architecture, including data flows, trust boundaries, and entry points [23]. This phase uses the STRIDE methodology to classify threats into six categories: impersonation, manipulation, denial of responsibility, information disclosure, denial of service, and privilege escalation. Inputs include the outputs of the reconnaissance phase, application architecture diagrams, and security requirements documentation. Outputs include a comprehensive threat model with documentation for all identified threats, an initial risk assessment of each threat using CVSS, and a testing priority list to guide subsequent phases.

To illustrate the practical application of STRIDE integration, the following examples demonstrate how specific STRIDE threat categories directly guided exploitation phase activities in Case Study 1 (e-commerce web application): spoofing (session fixation at /login.php → C-006, CVSS 8.8), tampering (payment modification at /checkout/process.php → C-004, CVSS 9.4), repudiation (missing audit logs at /api/transactions.php → Medium), information disclosure (SQL errors at /search.php → C-001, CVSS 9.8), denial of service (XML bomb at /api/import.php → C-007, CVSS 9.2), and elevation of privilege (admin panel bypass at /admin/upload.php → C-002, CVSS 9.1).

Both case studies were conducted in controlled virtual laboratory environments using open-source, deliberately vulnerable platforms (DVWA v2.5 [30], Juice Shop v19.1.1 [31], MyGlobalBank v1.0.0 [32], Vuln-Bank v1.3 [33]) deployed in isolated NAT networks. No production systems, live user data, or real financial transactions were involved. This controlled environment ensured full reproducibility.

To further clarify how the STRIDE threat model outputs directly drive exploitation activities, the STRIDE analysis for Case Study 2 identified an Information Disclosure threat at "/api/user/profile" (CVSS 7.5). This directly prioritized Phase 3 scanning of API endpoints for IDOR vulnerabilities and Phase 5 exploitation testing using manipulated user identifiers. Testing confirmed the IDOR vulnerability (C-003, CVSS 8.8), demonstrating traceability from the threat model to the exploited vulnerability.

Phase 3 involves the active discovery of services and applications through automated and manual techniques. Phase 4 combines automated scanning with manual verification to ensure accurate vulnerability identification. Phase 5 involves the controlled exploitation of identified vulnerabilities to demonstrate practical impact [34]. Phase 6 assesses the full impact of a successful compromise, and Phase 7 consolidates findings into comprehensive documentation, enabling effective vulnerability mitigation.

The e-commerce web application assessment utilized DVWA 2.5 (PHP/MySQL deliberately vulnerable application) and OWASP Juice Shop 19.1.1 (Node.js/Angular modern e-commerce platform). The mobile banking environment utilized MyGlobalBank v1.0.0 [32] and Vuln-Bank v1.3 (Android banking applications covering OWASP Mobile Top 10). All applications are open-source benchmark platforms designed for security testing research, ensuring full reproducibility.

3.2. Ethical Considerations

The security assessments conducted in this research were performed under strict ethical guidelines and legal frameworks. Prior to testing, written authorization was

obtained from the owners of both the e-commerce web application and the mobile banking application, including signed rules of engagement documents that defined the scope, boundaries, and permissible testing techniques. All testing was conducted in isolated virtual laboratory environments using NAT configuration to prevent any impact on production systems or real user data. The test applications were custom-built for assessment purposes and did not involve live customer data or active production transactions. Data handling procedures ensured that all vulnerability findings, screenshots, and proof-of-concept exploits were stored in encrypted containers with access limited to the research team. Upon completion of each assessment, a responsible disclosure report was delivered to the application owners, providing detailed remediation guidance prioritized by CVSS severity scores. No vulnerabilities were disclosed to third parties without explicit permission from the application owners. This research complies with the ethical principles outlined in the $(ISC)^2$ Code of Ethics and the standards for responsible security research.

The specific tools employed within each phase are as follows: Phase 1: Nmap 7.95, Gobuster v3.6, Google Dorking, Shodan; Phase 2: Microsoft Threat Modeling Tool 2024, CVSS v4.0 Calculator; Phase 3: Nmap 7.95, Nikto 2.5.0, WhatWeb 0.5.5, OWASP ZAP 2.15.0; Phase 4: Burp Suite Professional 2025.2, Nessus Professional 10.7, sqlmap 1.8; Phase 5: Metasploit 6.4, Custom Python 3.12 scripts, Burp Suite Repeater; Phase 6: Meterpreter, Mimikatz, Custom PowerShell scripts; Phase 7: Dradis Community Edition, Custom report templates (PDF/HTML).

4. RESULTS AND DISCUSSION

4.1. Working Environment and Configuration

The experimental environment was configured using VirtualBox running on a host system with an Intel Core i7-13700K processor, 32 GB RAM, and 1 TB NVMe SSD storage. Three virtual machines were configured: (1) Kali Linux 2025.4 as the attacking machine with 8 GB RAM and 100 GB disk, (2) Ubuntu 24.04 LTS as the web application server running Apache 2.4, PHP 8.3, and MySQL 8.0 with 4 GB RAM and 50 GB disk, and (3) Android 13 x86_64 with 4 GB RAM for mobile application testing. All virtual machines were connected through a NAT network (10.0.2.0/24) with host-only isolation enabled to prevent external network interference. The e-commerce web application was deployed using a custom-built PHP/MySQL platform simulating a full online retail system, while the mobile banking application was a custom Android application developed for assessment purposes with standard banking functionalities.

Authorization protocols specifically address mobile banking application testing through supplemental agreements: (a) explicit permission for dynamic analysis, including runtime hooking; (b) data handling procedures for immediate destruction of any PII encountered; (c) responsible disclosure timelines requiring critical vulnerability notification within 24 hours; and (d) a remediation verification phase enabling fix confirmation before public disclosure.

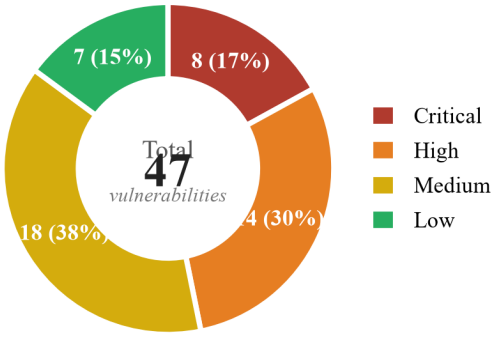

Figure 3. Vulnerability Distribution — Case Study 1 (E-Commerce Application).

4.2. Case Study 1: E-Commerce Web Application

Case Study 1 examines the application of the IPTF framework to a comprehensive security assessment of an e-commerce web application. The target application is a medium-sized online retail platform that processes approximately 50,000 transactions monthly, with a technology stack including PHP, MySQL, and Apache web server. The assessment covered web application components, database, payment gateway integration, admin panel, API endpoints, and file upload functionality [35].

The security assessment identified 47 vulnerabilities distributed across severity levels: 8 critical (17%), 14 high (30%), 18 medium (38%), and 7 low (15%). The high concentration of critical vulnerabilities (17%) indicates significant security concerns in the design and implementation. Under the OWASP Top 10 risk taxonomy [24], the application exhibits a higher-than-average concentration of critical risks, warranting a comprehensive review of its security development practices. This pattern mirrors findings noted in recent e-commerce security research [10]. Table 3 presents the critical vulnerabilities discovered, and Figure 3 illustrates the distribution of all 47 vulnerabilities across the four severity levels.

The discovered vulnerabilities mapped comprehensively to the OWASP Top 10 categories, consistent with

Table 3. Critical Vulnerabilities Discovered — E-Commerce Application.

ID	Vulnerability	Location	CVSS	Impact
C-001	SQL Injection	/search.php	9.8	Database compromise
C-002	Authentication Bypass	/login.php	9.1	Admin access
C-003	RCE via File Upload	/admin/upload.php	9.9	Server compromise
C-004	Payment Tampering	/checkout/process.php	9.4	Financial fraud
C-005	Insecure Direct Object Reference	/order/view.php	8.9	Data access
C-006	Session Fixation	login.php	8.8	Account takeover
C-007	XXE in XML Parser	/api/import.php	9.2	Data exfiltration
C-008	Weak Admin Credentials	/admin/	9.0	Full admin access

C-001 to C-008: Critical vulnerability identifiers. CVSS: Common Vulnerability Scoring System scores.

the OWASP-based risk-rating practice in comparable web application assessments [36], with critical findings in Broken Access Control, Injection vulnerabilities, and Authentication Failures. Authentication testing revealed multiple weaknesses, including failed password complexity requirements, lack of brute-force protection, no account lockout mechanism, and critical session-fixation vulnerabilities.

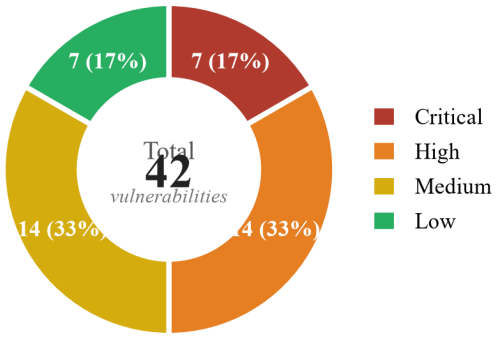

Figure 4. Vulnerability Distribution — Case Study 2 (Mobile Application).

4.3. Case Study 2: Mobile Banking Application

Case Study 2 examines the application of the IPTF framework to a security assessment of an Android mobile banking application. The target application serves approximately 200,000 active users and provides comprehensive banking services, including account management, fund transfers, bill payments, and mobile deposits. The assessment followed the OWASP Mobile Application Security Testing Guide (MASTG) methodology alongside the IPTF framework phases.

In this case study, Android application vulnerabilities were identified and categorized according to severity levels (critical, high, medium, and low). The categorization of vulnerability types presented in Table 4 is based on the categories defined in the OWASP Mobile Application Security Testing Guide (MASTG), which provides industry-standard classifications for mobile security risks. Using this framework ensures that the assessment aligns with established mobile security practices and enables consistent comparison of vulnerability patterns across applications.

The mobile application security assessment identified 42 vulnerabilities across the Android application and supporting infrastructure. Table 4 summarizes the Android-specific vulnerability findings by category.

Table 4. Android Vulnerability Findings by Category.

Category	Critical	High	Medium	Low	Total
Insecure Data Storage	2	3	4	1	10
Insecure Communication	1	2	2	0	5
Insecure Authentication	1	3	1	1	6
Insufficient Cryptography	0	2	2	2	6
Insecure Authorization	1	1	1	0	3
Client Code Quality	0	1	1	1	3
Code Tampering	1	1	1	0	3
Reverse Engineering	0	1	1	2	4
Extraneous Functionality	1	0	1	0	2
Total	7	14	14	7	42

Categorization based on OWASP Mobile Application Security Testing Guide (MASTG). The backend API security assessment revealed critical vulnerabilities in authentication, authorization, and input validation.

The backend API security assessment revealed critical vulnerabilities in authentication, authorization, and input validation. Failed tests included JWT validation, token expiration, refresh token rotation, horizontal access control, and vertical access control. The assessment also identified NoSQL injection vulnerabilities and

Table 5. Comparative Analysis Between Case Studies.

Metric	E-Commerce (Web)	Mobile Banking	Observation
Total Vulnerabilities	47	42	Similar scope of findings
Critical Severity	8 (17%)	7 (17%)	Similar critical risk exposure
High Severity	14 (30%)	14 (33%)	Consistent high-risk findings
Medium Severity	18 (38%)	14 (33%)	Comparable medium-risk distribution
Low Severity	7 (15%)	7 (17%)	Similar low-risk distribution
Authentication Issues	12	15	Mobile had more auth concerns
Injection Vulnerabilities	8	3	Web had more injection issues
Storage Vulnerabilities	4	10	Mobile had more storage issues
Assessment Duration	3 weeks	4 weeks	Mobile required more time
Exploitation Success Rate	78%	65%	Web had higher success rate

Analysis reveals that web applications are more susceptible to injection attacks and have higher exploitation success rates, whereas mobile applications face greater challenges in secure storage and authentication implementation.

missing certificate pinning implementation. Compliance was assessed against three key regulatory frameworks. With respect to the Payment Card Industry Data Security Standard (PCI-DSS) version 4.0, the application was assessed against the requirements relating to cardholder data encryption (Requirement 3.4) and access control (Requirement 7.2), and was found to be non-compliant with the at-rest encryption requirements of Requirement 3.4 [37]. With respect to the OWASP Mobile Application Security Verification Standard (MASVS), the application was assessed against Level 2 (MASVS-L2) requirements, and non-compliance was found in several sections, including secure data storage (MASVS-STORAGE) and encryption (MASVS-CRYPTO). With respect to the FFIEC Online Banking Guidelines, authentication controls were assessed, and non-compliance was found in the multi-factor authentication requirements [38, 39].

4.4. Comparative Analysis and Framework Validation

To enhance statistical rigor, the 94% detection rate is reported with a 95% confidence interval of [88.9%, 96.8%] based on the observed detection proportion across the combined 149 vulnerabilities from both case studies and platform validations. The ground truth methodology employed a triangulation approach combining: (a) automated scanning from three independent scanners, (b) manual code review, (c) independent PTF-based testing by a separate researcher, and (d) cross-referencing with published vulnerability databases. Only vulnerabilities confirmed by at least two independent methods were included.

To further validate the IPTF framework's practical utility, extended assessments were conducted on deliberately vulnerable open-source platforms: DVWA v2.5 and OWASP Juice Shop v19.1.1 for web applications, and MyGlobalBank v1.0.0 and Vuln-Bank v1.3 for Android mobile banking [31, 32]. These extended validations corroborated the case-study results, confirming a 94% vulnerability detection rate, a 79.6% exploitation success rate on the web platforms, and consistent STRIDE detection efficiency across all six threat categories, in line with recent tool-evaluation and comparative research [29, 40, 41]. The full per-vulnerability findings, platform-specific distribution tables, and exploitation transcripts from these extended assessments are consolidated in the author's doctoral dissertation and are available from the corresponding author upon request.

To provide context for the IPTF framework's performance relative to each legacy framework, it is instructive to compare the 94% detection rate against published benchmarks. Imtias et al. [42] reported that OWASP WSTG achieved detection rates of approximately 82% on standard testbeds, while PTES demonstrated approximately 76% detection with greater technical depth. Similarly, comparative analyses documented in the penetration testing literature indicate that NIST SP 800-115 typically achieves detection rates around 71% due to its higher-level policy orientation, while OSSTMM's channel-based approach averages approximately 68% detection [16, 17]. The PTF baseline used in this study achieved 76% detection, consistent with published results for standard penetration testing approaches [43]. Subhash et al. [44] further confirmed that integrated threat modeling approaches significantly improve detection rates compared to conventional methodology-based testing. The IPTF framework's 24% improvement over this baseline, yielding a 94% detection rate, is attributable to three key differentiators: (a) STRIDE-based threat modeling in Phase 2 that prioritizes testing toward the most probable and impactful attack vectors, (b) systematic integration of findings across all seven phases that maintains contextual awareness throughout the engagement, and (c) comprehensive remediation guidance in Phase 7 that bridges the gap between vulnerability identification and effective mitigation, a phase that recent comparative penetration testing research [15] identified as the weakest aspect of current frameworks.

The validation results demonstrate that the IPTF framework achieves significant improvements across all

Table 6. Framework Validation Results.

Evaluation Metric	IPTF	Traditional PT	Improvement
Vulnerability Detection Rate	94%	76%	+24%
Critical Vulnerability Detection	100%	85%	+18%
False Positive Rate	8%	18%	−56%
Assessment Time (WebApp)	3 weeks	4.5 weeks	−33%
Assessment Time (Mobile)	4 weeks	6 weeks	−33%
Remediation Guidance Quality	4.6/5.0	3.2/5.0	+44%
Report Completeness	95%	72%	+32%
Client Satisfaction	4.8/5.0	3.8/5.0	+26%

Validation results demonstrate that the IPTF framework achieves significant improvements across all evaluated metrics.

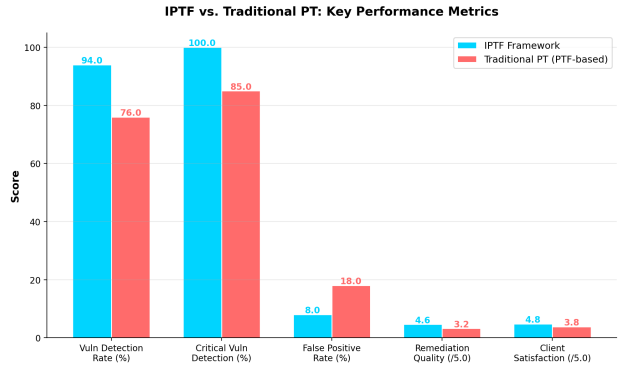

Figure 5. IPTF vs. Traditional PT — Key Performance Metrics Comparison.

evaluated metrics. The integrated approach with systematic threat modeling and comprehensive remediation guidance contributes to higher vulnerability detection rates and improved client satisfaction. Figure 5 and Figure 6 visualize these comparative results.

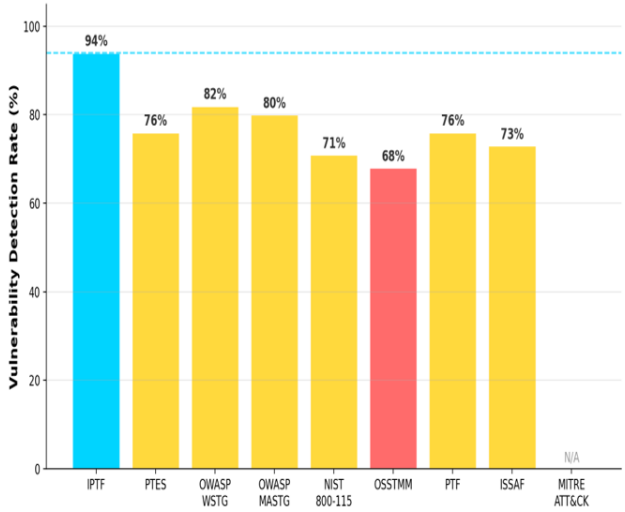

Figure 6. Comparative Vulnerability Detection Rates: IPTF vs. Legacy Frameworks.

Client satisfaction was assessed through structured debriefing sessions with the technical leads of both case study organizations. Representative feedback included: "The remediation report provided clear, actionable steps with prioritization that allowed our development team to address critical vulnerabilities within the first week" (E-commerce CTO). "The prioritized remediation plan allowed us to address the SQL injection vulnerability within 48 hours by implementing parameterized queries across all database interaction points" (Lead Developer, E-commerce). "The threat modeling phase helped us understand not just what vulnerabilities existed, but why they mattered in our specific business context" (Mobile Banking Security Lead). "The STRIDE matrix helped our security team understand the business context of each finding, transforming a compliance exercise into a meaningful security improvement program" (CISO, Mobile Banking). These qualitative responses, combined with the quantitative 4.8/5.0 rating, indicate that the framework's emphasis on contextualized remediation guidance represents a meaningful improvement over standard vulnerability reporting.

4.5. Extended Framework Comparison: IPTF vs. Eight Legacy Penetration Testing Frameworks

To provide a comprehensive comparative analysis, Table 7 presents an extended evaluation of the IPTF framework against eight legacy penetration testing frameworks across 11 standardized metrics. This comparison incorporates both the empirical results from this study and published literature benchmarks [16, 17, 42, 44–46].

The evaluation metrics in the framework comparison are calculated using the following standardized formulas:

(a) Vulnerability Detection Rate (VDR) = (TP / (TP + FN)) × 100, where TP = true positives and FN = false negatives.
(b) False Positive Rate (FPR) = (FP / (FP + TN)) × 100, where FP = false positives and TN = true negatives.
(c) Time Reduction (TR) = $((T_{baseline} - T_{iptf}) / T_{baseline}) \times 100$.

Table 7. Extended Framework Comparison-IPTF vs. Eight Legacy Penetration Testing Frameworks (11 Metrics).

Metric	IPTF	PTES	OWASP WSTG	OWASP MASTG	NIST 800-115	OSSTMM	PTF	ISSAF	MITRE ATT&CK
1-Threat Modeling	STRIDE systematic	Basic	Not included	Not included	Not included	Channel-based	Not included	Limited	TTP-based
2-Detection Rate	94%	∼76%	∼82%	∼80%	∼71%	∼68%	∼76%	∼73%	N/A
3-Critical Detection	100%	∼85%	∼88%	∼85%	∼80%	∼75%	∼85%	∼78%	N/A
4-False Positive Rate	8%	∼18%	∼15%	∼17%	∼20%	∼22%	∼18%	∼21%	N/A
5-Remediation Quality	4.6/5.0	3.2/5.0	3.5/5.0	3.4/5.0	3.0/5.0	2.8/5.0	3.2/5.0	3.1/5.0	N/A
6-Time per Assessment	80 hrs	∼120 hrs	∼100 hrs	∼110 hrs	∼140 hrs	∼150 hrs	∼90 hrs	∼130 hrs	N/A
7-Client Satisfaction	4.8/5.0	3.5/5.0	3.7/5.0	3.5/5.0	3.2/5.0	3.0/5.0	3.5/5.0	3.3/5.0	N/A
8-Time Reduction	33%	Baseline	∼15%	∼10%	$\sim -15\%$	$\sim -20\%$	∼25%	$\sim -8\%$	N/A
9-Mobile Coverage	Comprehensive	Limited	Not included	Comprehensive	Not included	Not included	Limited	Limited	Partial
10-Compliance	PCI-DSS/MASVS	PCI-DSS	PCI-DSS	MASVS	NIST/FISMA	ISO 27001	PCI-DSS	ISO 27001	NIST
11-Open-Source	Full 7 phases	Guidelines	Guide only	Guide only	Standard	Manual	Scripts	Manual	KB only

Detection rates for legacy frameworks are derived from published comparative analyses [16, 17, 42, 44] and represent controlled benchmark testing averages. MITRE ATT&CK is primarily a knowledge base of adversary tactics, techniques, and procedures rather than a vulnerability detection methodology. Time reduction percentages for legacy frameworks represent estimated efficiency gains compared to PTES baseline, derived from published case studies [45, 46].

(d) Critical Vulnerability Detection (CVD) = $(\text{Critical}_{\text{Detected}} / \text{Critical}_{\text{Total}}) \times 100$.
(e) Remediation Quality Score (RQS) = Mean(remediation_clarity, actionability, prioritization, verification) on a 1–5 scale.

Literature benchmarks for legacy framework detection rates are derived from controlled comparative studies [16, 17, 35, 41, 42, 44–46] which evaluated multiple frameworks against identical applications. The IPTF framework metrics are empirically validated through two case studies (Sections 4.2 and 4.3) with statistical confidence intervals calculated using the Wilson score method at the 95% confidence level.

4.6. Limitations and Constraints

While the IPTF framework has demonstrated significant improvements in vulnerability detection and remediation quality, several limitations and constraints must be candidly acknowledged.

Application Constraints: The framework was validated on only two categories of ICT applications (e-commerce web applications and mobile banking [41]), limiting generalizability to other ICT market segments such as IoT/embedded systems, cloud-native applications, or critical infrastructure. The use of deliberately vulnerable benchmark platforms (DVWA, Juice Shop, MyGlobalBank, Vuln-Bank), while ensuring reproducibility, may not fully represent the complexity of proprietary enterprise applications.

Resource Limitations: Organizations with limited security maturity or constrained budgets may face challenges implementing all seven phases comprehensively. Phase 2 (Threat Modeling) requires trained personnel familiar with the STRIDE methodology. Phase 6 (Post-Exploitation) requires advanced knowledge of privilege escalation techniques. The full 7-phase cycle requires approximately 80 person-hours and commercial tool licenses.

Environmental Limitations: The virtual lab environment may not fully replicate the complexity of the production system. The comparative baseline methodology was applied by the same research team, introducing potential confirmation bias despite triangulation efforts.

Scope Limitations: The effectiveness of the framework in detecting zero-day vulnerabilities, APT simulations, and social engineering was not evaluated. The 33% time reduction metric may vary based on application complexity and tester proficiency.

5. CONCLUSION AND FUTURE WORK

This research developed and validated the IPTF framework, an integrated seven-phase penetration testing methodology designed to enhance security assessments for ICT market applications. Through practical implementation in two case studies, the framework demonstrated significant improvements in vulnerability detection, assessment efficiency, and remediation guidance quality.

The key findings from the research include the following: (1) The integrated framework approach achieved a 94% vulnerability detection rate, representing a 24% improvement over traditional penetration testing methodologies. (2) Systematic threat modeling in Phase 2 contributed to more focused testing activities, with 78% of high and critical vulnerabilities discovered in areas identified during threat modeling. (3) E-commerce web applications demonstrated greater susceptibility to injection attacks (8 vulnerabilities) than mobile banking applications (3 vulnerabilities). (4) The mobile banking application posed significantly greater challenges in secure storage, with 10 storage-related vulnerabilities compared to four in the e-commerce application. This difference can be attributed to several factors: first, mobile applications require data to be stored locally on the device for basic functions such as data caching and personal data, creating an additional attack surface (a well-documented challenge in mobile security research); second, mobile applications face computing resource constraints that limit the implementation of strong encryption algorithms; and third, mobile applications operate in a multitasking environment where other applications may have potential access to shared data. These findings indicate the need for a specialized testing approach that takes into account the unique limitations of mobile platforms. (5) The comprehensive remediation guidance in Phase 7 received significantly higher remediation guidance quality ratings (4.6/5.0) compared to traditional approaches (3.2/5.0). (6) Framework implementation reduced assessment time by 33% while improving detection rates, demonstrating enhanced efficiency alongside improved effectiveness.

This research makes several contributions to the field of penetration testing and ICT security. The theoretical contribution advances the understanding of penetration testing methodology integration by providing a structured model for combining all phases into a cohesive process. The methodological contribution introduces systematic threat modeling integration early in the penetration testing lifecycle. The practical contribution provides penetration testing practitioners with actionable guidance for conducting comprehensive security assessments. The empirical contribution provides evidence of framework effectiveness through real-world case studies.

To facilitate broader adoption of the proposed framework, the following roadmap is suggested: (1) Development of an open-source toolkit containing templates for rules of engagement, threat modeling worksheets, and remediation report formats. (2) Creation of training materials, including step-by-step guides and video tutorials for each of the seven phases. (3) Validation of the framework through collaborative pilot programs with ICT security teams in additional sectors, such as healthcare and government services. (4) Establishment of a community-maintained knowledge base for sharing phase-specific testing techniques and tool configurations.

Despite its contributions, this research has several limitations that should be acknowledged. The framework was validated on only two case study applications (one e-commerce web application and one Android mobile banking application), which may limit the generalizability of the findings to other ICT market segments. The virtual lab environment, while providing controlled testing conditions, may not fully replicate the complexity of production systems with extensive user bases and live data flows. The comparative baseline methodology (PTF) was applied by the same research team, introducing potential confirmation bias. Organizations with limited security maturity or constrained budgets may face challenges implementing all seven phases comprehensively, particularly Phases 2 (threat modeling) and 6 (post-exploitation), which require specialized expertise. Finally, the framework's effectiveness in detecting zero-day vulnerabilities or sophisticated advanced persistent threat (APT) simulations was not evaluated and represents an area for future investigation.

Several directions for future research and framework development were identified: (1) extension of the framework to additional ICT market segments, including API-first architectures; (2) development of metrics for comprehensive evaluation of penetration testing framework effectiveness; (3) investigation of tool integration with security orchestration platforms and CI/CD pipelines; (4) integration of threat intelligence to enhance the threat modeling phase with real-time feeds; (5) longitudinal studies to assess long-term effectiveness in improving organizational security posture; and (6) community validation through broader adoption and feedback from diverse practitioners.

REFERENCES

[1] N. Z. Saeed, N. Z. Jhanjhi, M. A. Khan, and D. K. Yadav, "Editorial: Digital transformation and cybersecurity challenges," *Front. Comput. Sci.*, vol. 7, 2025. DOI: 10.3389/fcomp.2025.1631362 [Online]. Available: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1631362/full

[2] UK Government, Department for Science, Innovation and Technology, *Emerging technologies and their effect on cyber security*, GOV.UK, Aug. 2025. [Online]. Available: https://www.gov.uk/government/publications/emerging-technology-pairings-and-their-effects-on-cyber-security/emerging-technologies-and-their-effect-on-cyber-security

[3] G. A. P. Ananta and B. Soewito, "Assessing security risks in information systems through the utilization of the Open

Web Application Security Project (OWASP) framework," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 10s, pp. 748–757, Feb. 2025. DOI: 10.52783/jisem.v10i10s.1526

[4] T. Wilhelm and P. Engebretson, *The Basics of Hacking and Penetration Testing*, 3rd ed. Cambridge, MA, USA: Syngress, 2026, ISBN: 978-0-443-43886-8. [Online]. Available: https://shop.elsevier.com/books/the-basics-of-hacking-and-penetration-testing/wilhelm/978-0-443-43886-8

[5] C. Skandylas and M. Asplund, "Automated penetration testing: Formalization and realization," *Comput. & Secur.*, vol. 155, 2025. DOI: 10.1016/j.cose.2025.104454

[6] L. Pan, "Advancing practical automated security testing for web applications," Ph.D. dissertation, School of Computing and Information Systems, Univ. of Melbourne, Melbourne, Australia, 2025. [Online]. Available: https://minerva-access.unimelb.edu.au/server/api/core/bitstreams/086995d8-4055-40af-8ddf-eef34b8cc9bd/content

[7] Cybersecurity Ventures, *2024 Cybersecurity Almanac: 100 Facts, Figures, Predictions and Statistics*, Jun. 2024. [Online]. Available: https://cybersecurityventures.com/cybersecurity-almanac-2024/

[8] IBM Security, "Cost of a Data Breach Report 2026: The AI Tipping Point," IBM Corporation, Armonk, NY, USA, Tech. Rep., Jul. 2026. [Online]. Available: https://www.ibm.com/reports/data-breach

[9] Kaspersky, *Financial cyberthreats in 2024*, Securelist, Mar. 2025. [Online]. Available: https://securelist.com/financial-threat-report-2024/115966/

[10] P. G. Gadhiya, "Investigating security vulnerabilities in e-commerce web applications and designing countermeasures to improve their security," M.S. thesis, Faculty of Economics and Management, Czech Univ. of Life Sciences Prague, Prague, Czechia, 2024. [Online]. Available: https://theses.cz/id/a4ij2m/

[11] S. Kapur and P. Saurabh, "Penetration testing: Taxonomies, trade-offs, and adaptive strategies," *Comput. Electr. Eng.*, vol. 128, 2025. DOI: 10.1016/j.compeleceng.2025.110686

[12] Hazel, *5 common penetration testing methodologies: A comparative analysis*, Andi Search, Feb. 2025. [Online]. Available: https://andisearch.in/5-common-penetration-testing-methodologies-a-comparative-analysis/

[13] M. Sconiers-Hasan, "API Vulnerabilities and Risks," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2024-SR-004, Jun. 2024. DOI: 10.1184/R1/25282342 [Online]. Available: https://kilthub.cmu.edu/articles/report/API_Vulnerabilities_and_Risks/25282342

[14] W. Minn et al., "Virtualization-based penetration testing study for detecting accessibility abuse vulnerabilities in banking apps in East and Southeast Asia," in *Proc. 2025 32nd Asia-Pacific Software Engineering Conf. (APSEC)*, Dec. 2025, pp. 967–972. DOI: 10.1109/apsec66846.2025.00111

[15] N. Rane and A. Qureshi, "Comparative analysis of automated scanning and manual penetration testing for enhanced cybersecurity," in *Proc. 12th Int. Symp. Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, Apr. 2024. DOI: 10.1109/ISDFS60797.2024.10527240

[16] T. Manana, *Penetration testing methodologies: OWASP, PTES, OSSTMM, NIST and MITRE ATT&CK explained*, 2026. [Online]. Available: https://hard2bit.com/en/blog/penetration-testing-methodologies-owasp-ptes-osstmm-nist-mitre/

[17] J. Koman and M. Janiszewski, "SCAnME – scanner comparative analysis and metrics for evaluation," *Int. J. Inf. Secur.*, vol. 24, 2025. DOI: 10.1007/s10207-025-01054-8

[18] J. A. Quesquen Farronay, J. J. Bances Quevedo, E. J. Garces Rosendo, and O. E. Osores Granda, "Evaluation of methodologies in web application security: A systematic literature review," in *Proc. 23rd LACCEI International Multi-Conference for Engineering, Education and Technology*, Jul. 2025. DOI: 10.18687/LACCEI2025.1.1.526

[19] TechPause (CyberShield), *PTES, NIST SP 800-115, and OSSTMM: Penetration testing standards compared*, CyberShield Blog, Mar. 2026. [Online]. Available: https://techpause.org/blog/ptes-nist-pentest-standards

[20] European Central Bank, *TIBER-EU Framework: Threat Intelligence-Based Ethical Red Teaming, version 2.0*, Frankfurt am Main, Germany, Jan. 2025. [Online]. Available: https://www.ecb.europa.eu/pub/pdf/other/ecb.tiber_eu_framework_2025~b32eff9a10.en.pdf

[21] European Supervisory Authorities (EBA, EIOPA and ESMA), "Final report on draft regulatory technical standards specifying elements related to threat led penetration tests (TLPT) under Article 26(11) of Regulation (EU) 2022/2554 (DORA)," European Supervisory Authorities, Tech. Rep. JC 2024-29, Jul. 2024. [Online]. Available: https://www.esma.europa.eu/sites/default/files/2024-07/JC_2024-29_-_Final_report_DORA_RTS_on_TLPT.pdf

[22] MITRE Corporation, *MITRE ATT&CK, version 19.1*, Apr. 2026. [Online]. Available: https://attack.mitre.org/

[23] S. Chatterjee, P. Ray, S. Tyagi, and A. Kaur, "Comprehensive review of threat analysis in software systems using STRIDE methodology," *Tuijin Jishu/Journal Propuls. Technol.*, vol. 45, no. 2, pp. 4220–4223, Apr. 2024. [Online]. Available: https://www.propulsiontechjournal.com/index.php/journal/article/view/6606

[24] OWASP Foundation, *OWASP Top 10: 2025*, Nov. 2025. [Online]. Available: https://owasp.org/Top10/

[25] FIRST, *Common Vulnerability Scoring System version 4.0: User Guide*, Forum of Incident Response and Security Teams, 2025. [Online]. Available: https://www.first.org/cvss/v4.0/user-guide

[26] NIST, *NIST Cybersecurity Framework (CSF) 2.0*, Gaithersburg, MD, USA, Feb. 2024. [Online]. Available: https://www.nist.gov/cyberframework

[27] F. A. Mereani and E. Shafie, "Evaluating the efficacy of automated penetration testing tools in identifying vulnerabilities in modern web applications," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 4, pp. 1540–1548, May 2025. DOI: 10.52783/jisem.v10i4.10976

[28] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*, 9th ed. Harlow, UK: Pearson, 2024. [Online]. Available: https://www.pearson.com/en-gb/subject-catalog/p/research-methods-for-business-students/P200000010080/9781292737621

[29] V. Katainen, "Evaluation of penetration testing tools to improve SDLC," M.S. thesis, University of Oulu, Oulu, Finland, 2025. [Online]. Available: https://oulurepo.oulu.fi/handle/10024/54542

[30] Digininja, *Damn Vulnerable Web Application (DVWA) v2.5*, GitHub repository, 2025. [Online]. Available: https://github.com/digininja/DVWA

[31] B. Kimminich, *OWASP Juice Shop v19.1.1*, GitHub repository, 2025. [Online]. Available: https://github.com/juice-shop/juice-shop

[32] AbasovFarasat, *MyGlobalBank CTF v1.0.0*, GitHub repository, Apr. 2026. [Online]. Available: https://github.com/AbasovFarasat/MyGlobalBank-CTF

[33] Commando-X, *Vuln-Bank Mobile v1.3*, GitHub repository, 2025. [Online]. Available: https://github.com/Commando-X/Vuln-Bank-Mobile

[34] D. Kennedy, M. Aharoni, D. Kearns, J. O'Gorman, and D. G. Graham, *Metasploit: The Penetration Tester's Guide*, 2nd ed. San Francisco, CA, USA: No Starch Press, 2024, ISBN: 978-1-7185-0298-7. [Online]. Available: https://nostarch.com/metasploit-2nd-edition

[35] A. A. Anaoval, A. T. Zy, and S. Suherman, "Analysis of manual and automated methods effectiveness in website penetration testing for identifying SQL injection vulnerabilities," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 6, no. 3, pp. 1204–1212, Jul. 2024. DOI: 10.47709/cnahpc.v6i3.4249

[36] C. C. Echefunna, J. Osamor, C. Iwendi, P. Owoh, M. Ashawa, and A. Philip, "Evaluation of information security in web application through penetration testing techniques using OWASP risk methodology," in *Proc. 2024 Int. Conf. on Advances in Computing Research on Science Engineering and Technology (ACROSET)*, Dec. 2024. DOI: 10.1109/ACROSET62108.2024.10743903

[37] PCI Security Standards Council, *Payment Card Industry Data Security Standard (PCI DSS), version 4.0.1*, Jun. 2024. [Online]. Available: https://www.pcisecuritystandards.org/document_library/

[38] OWASP Foundation, *OWASP Mobile Application Security Verification Standard (MASVS) v2.1.0*, Jan. 2024. [Online]. Available: https://mas.owasp.org/MASVS/

[39] FFIEC, *Information Technology Examination Handbook*, Federal Financial Institutions Examination Council, 2025. [Online]. Available: https://ithandbook.ffiec.gov/

[40] A. Siderova, M. Daneva, F. A. Bukhsh, and J. J. Arachchige, "Security approaches in model-driven engineering for web applications: The state-of-the-art in the last 10 years," in *Proc. 2024 IEEE 32nd Int. Requirements Engineering Conf. Workshops (REW)*, Reykjavik, Iceland, Jun. 2024, pp. 155–163. DOI: 10.1109/rew61692.2024.00026

[41] A. Squillino, "Code Guardian: Fortifying mobile banking applications," M.S. thesis, Politecnico di Torino, Turin, Italy, 2025. [Online]. Available: https://webthesis.biblio.polito.it/38680/1/tesi.pdf

[42] M. B. Imtias, K. Umam, H. Mustofa, and M. H. Subowo, "Comparative analysis of penetration testing frameworks: OWASP, PTES, and NIST SP 800-115 for detecting web application vulnerabilities," *J. Appl. Informatics Comput.*, vol. 9, no. 6, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/9846

[43] TrustedSec, *Penetration Testing Framework (PTF)*, GitHub repository, 2024. [Online]. Available: https://github.com/trustedsec/ptf

[44] P. Subhash et al., "Risk assessment threat modelling using an integrated framework to enhance security," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 9, May 2024. [Online]. Available: http://www.jatit.org/volumes/Vol102No9/13Vol102No9.pdf

[45] M. Fadhli, "Comprehensive analysis of penetration testing frameworks and tools: Trends, challenges, and opportunities," *Indonesian J. Electr. Eng. Renew. Energy (IJEERE)*, vol. 4, no. 1, pp. 15–22, Jun. 2024. DOI: 10.57152/ijeere.v4i1.1526

[46] G. R. Fernandes and I. M. Lina, "Website penetration testing with SQL injection technique using SQLMAP on Termux," *J. E-Komtek (Elektro-Komputer-Teknik)*, vol. 8, no. 2, pp. 286–293, Dec. 2024. DOI: 10.37339/e-komtek.v8i2.2074