HiLL PUBLISHING

Cybersecurity and Information Assurance, 2026, 1(1), 10-14
https://www.hillpublisher.com/Journals/cia/
ISSN Online:

Efficient Process and Method of Penetration Test in Network Security

Zhibin Liu

School of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, Heilongjiang, China.

How to cite this paper: Zhibin Liu. (2026). Efficient Process and Method of Penetration Test in Network Security. *Cybersecurity and Information Assurance*, 1(1), 10-14.
DOI: 10.26855/cia.2026.06.003

Received: November 22, 2025
Accepted: December 16, 2025
Published: January 23, 2026

***Corresponding author:** Zhibin Liu, School of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, Heilongjiang, China.

Abstract

Internet is inseparable from daily life, network security concerns every netizen, and penetration penetration test is widely used in network security. Penetration testers find vulnerabilities by mimicking the thinking operations of hackers and attacking targets. A reasonable and comprehensive approach to the penetration testing process to address the limitations of current technology. The correct and systematic penetration test process is used to simulate the intruders, and all possible attack modes hidden in the network are found out from the hacker's point of view. The process of penetration test is formed into a reasonable and efficient process, and the penetration test workers are guided to choose the best response action, so as to avoid the network security being illegally invaded. Excellent penetration testing process, from the perspective of the attacker, find hidden attack vulnerabilities, improve the existing penetration testing technology, improve the penetration testing process, improve the security efficiency.

Keywords

Cyber attacks; penetration test; network security; automatic penetration

1. Introduction

Penetration testing (PenTest) refers to the assessment of the security of IT infrastructure by attempting to exploit vulnerabilities. These vulnerabilities may exist in operating systems, service and application defects, misconfigurations, or risky user behaviors. This assessment also helps to verify the effectiveness of defense mechanisms and end-user compliance with security policies (Halfond et al., 2011). Penetration testing typically uses manual or automated techniques to compromise target devices and other potential exposure points. After a vulnerability in the target system is successfully exploited, the tester will continue to attempt attacks on resources outside the compromised system. Any security vulnerability information successfully exposed through penetration testing is usually consolidated and submitted to the client or superiors to help these professionals draw strategic conclusions and determine remediation work. The basic purpose of penetration testing is to measure the likelihood of damage to the system or end-user and to assess the consequences of such incidents on related resources or operations. Penetration testers (also known as ethical hackers) simulate hackers to assess the security of IT infrastructure. They test servers, networks, web applications, mobile devices, and other potential vulnerabilities to discover weaknesses throughout the system.

A sound penetration testing process and methodology are core influencing factors in testing. This article proposes a complete penetration testing process and methodology to facilitate penetration testing security personnel's exploration of cybersecurity and enable them to conduct work and research efficiently.

2. Applications of Penetration Testing

The information of the test targets is usually confidential, and its exposure could cause significant harm to consumers

Note: The article was originally published in *Advances in International Computer Science*, and this is an authorized translation of it.

and companies if it were obtained by a malicious attacker (Li, 2018). The cost of penetration testing is constantly rising, and engineers are working to improve the process to minimize vulnerabilities. Despite these improvements, vulnerabilities continue to occur due to the complexity of the procedures and their deployment configurations. The continued prevalence of vulnerabilities increases the importance of identifying vulnerable techniques in deployed applications.

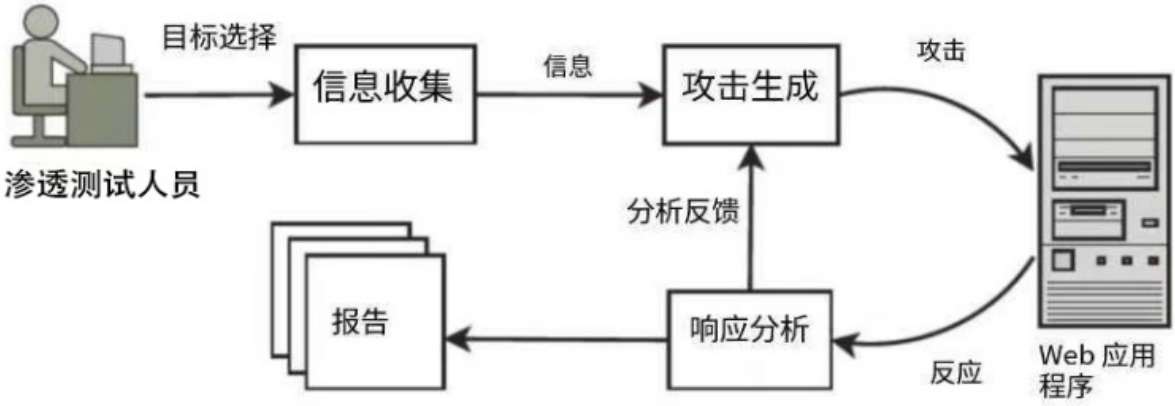

Figure 1. Penetration testing process.

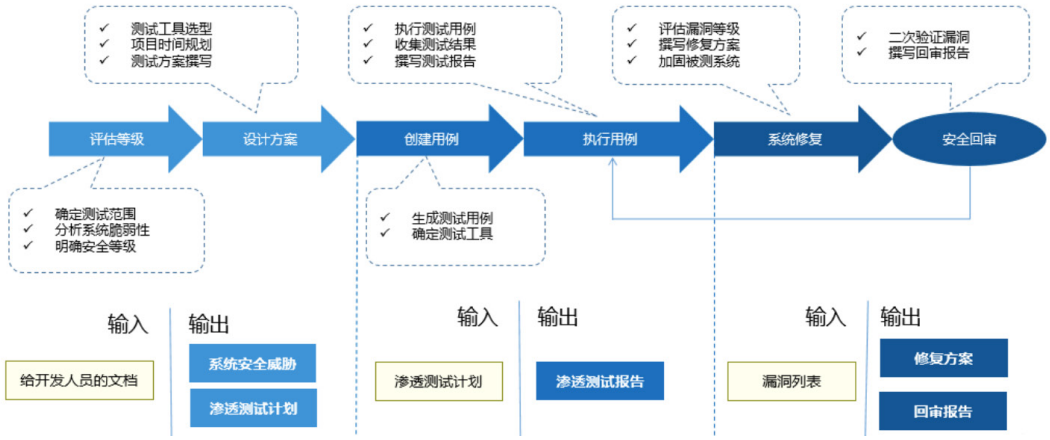

Figure 2. Penetration testing process.

The widespread use of penetration testing has led to the formation of standardization, accelerated the certification of penetration testing and the establishment of standardized "best practice" penetration testing. Although penetration testers need to perform a wide variety of tasks, they can generally be divided into three phases: information gathering, attack generation and response analysis (Li, 2020). Figure 1 shows the attack flow of penetration testing. In the first phase, information gathering, the penetration tester selects a target application and uses various techniques to obtain information about it. The results of this phase allow the penetration tester to perform the second phase, which is to develop an attack on the target application. This phase can usually be automated by customizing well-known attacks or using automated attack scripts. In the third phase, once the attack is executed, the penetration tester performs response analysis—they analyze the application's response to determine whether the attack was successful and prepare a final report on the vulnerabilities found (Lü, 2016).

3. Knowledge Graph-Based Automated Penetration Testing

Vulnerability detection features are matched with vulnerability databases for vulnerability scanning technology. Automated penetration testing verifies vulnerabilities by constructing attack payloads. By constructing real and controllable content requests, it not only has higher compatibility, lower false positive rate, fewer contracts, and faster detection speed, but can even bypass some defense rules.

Automated penetration testing based on knowledge graphs breaks away from the rigid model process. It abstracts penetration testing experience and tool structures into knowledge mappings of entity nodes. Through the connections between nodes, a penetration knowledge graph is formed. Each path's knowledge graph represents a penetration testing strategy. By expanding the scale of the knowledge graph, penetration testing capabilities can be improved. In actual penetration testing, existing penetration knowledge graphs are combined with collected data to create dynamic penetration knowledge graphs. Penetration path planning is then performed based on existing penetration data and the current penetration status, thereby improving penetration testing efficiency. Furthermore, due to the relevance of the knowledge graph, comprehensive analysis of data collected at each stage is possible. This allows data collected in later stages of penetration testing to drive earlier steps, resulting in a more comprehensive and thorough penetration test. This leads to deeper penetration, ultimately approaching the effect of human-like penetration testing. During penetration testing, the system can analyze and determine the current penetration testing status based on collected information and historical penetration testing status. Through joint reasoning and decision-making using knowledge graphs and expert systems, the optimal penetration testing path is selected, further improving penetration testing efficiency. Furthermore, the system can display the penetration process's ideas, steps, and operations in the form of penetration maps, enabling the repetition of the penetration testing process's ideas and steps, and providing penetration testers with penetration strategies.

4. Scope and Content of Penetration Testing

Website security penetration testing includes vulnerabilities such as SQL injection, cookie injection, file upload and inclusion, sensitive directory exposure, URL redirection vulnerabilities, online editor vulnerabilities, website authentication filtering vulnerabilities, remote code execution vulnerabilities, database vulnerabilities, website path vulnerabilities, XSS cross-site vulnerabilities, weak passwords, file download vulnerabilities, program function logic vulnerabilities, sending arbitrary number of SMS messages, flooding attacks, registering arbitrary number of mobile phone numbers or email addresses, bypassing API interface security authentication, and firewall bypassing.

Server security penetration includes intranet penetration, FTP privilege escalation vulnerability, SQL Server database privilege escalation vulnerability, MySQL privilege escalation vulnerability, Linux local overflow vulnerability, replacement system service vulnerability, remote desktop authentication bypass vulnerability, port mapping vulnerability, CC stress test, DDOS stress test, ARP spoofing page tampering test, DNS spoofing vulnerability, session hijacking vulnerability, and vulnerability testing on many application systems (such as virtual hosts). The three stages of attack, namely pre-attack, attack and post-attack, are adopted according to international standards, as shown in "Figure 2": pre-attack mainly refers to some information collection and vulnerability scanning process; attack process mainly uses the vulnerabilities or weak passwords found in the first stage to wait for vulnerability intrusion; post-attack refers to the subsequent work such as privilege escalation, backdoor installation, and deletion of tracking after obtaining certain permissions of the attack target (Shi & Chi, 2016). At the same time, advanced methods such as social engineering and password databases are combined to conduct attack testing.

Penetration testing aims to make the test target more secure and achieve the expected results. Web application penetration testing can be roughly divided into three stages: information gathering stage, vulnerability discovery stage, and vulnerability exploitation stage (Tang & Yu, 2009). In actual testing practice, the testing process needs to be further refined. The following is a detailed introduction to the entire system process of penetration testing in nine stages (Note: the tools, vulnerabilities, and exploitable information used in the testing process are not limited to those mentioned in the article):

4.1 Define the objectives

(1) Define the scope of security testing: such as IP address, domain name, internal and external networks, the entire website, or some pages or modules; (2) Define the security testing rules: the degree of penetration, whether further penetration is possible, time limits, whether uploaded files can be modified, whether privilege escalation is possible, whether blocking normal business operations is allowed, access methods, etc.; (3) Define the requirements: based on user needs and your own technical capabilities, determine the tasks that can be completed next.

4.2 Analyze the risks during the testing process and obtain authorization

Analyze potential risks involved in penetration testing, such as the processing of large amounts of test data, impact

on normal service use, emergency response to server anomalies, software and data corruption, data backup and recovery, and the human and material costs incurred during the testing process. The implementation plan is prepared by the testers and submitted to the client for review. Only after obtaining written authorization from the client can authorized testers conduct penetration testing on the authorized targets.

4.3 Information collection

During the information gathering phase, testers need to collect as much information as possible about the authorized target to facilitate better analysis and utilization. This information includes: open ports, relevant protocols used, programming language versions and types used, server information, application software and hardware information used, directory structure, database types and versions, all linked pages, application frameworks, etc. Security scanning tools such as NAMP, GOBY, and AWVS can be used to scan the website. WHOIS can be used to view the domain name, domain registration information, web application publisher IDs, website administrator names, unauthorized use of search engines to retrieve pages, sensitive sites, IP addresses, domain names, ports, protocols, network segments, operating system versions, all detected application version information, server information, firewall information, etc.

4.4 Vulnerability detection (manual & automatic)

For different vulnerability issues, different solutions can be developed using relevant information. Vulnerability detection tools such as AWVS, Nessus, AppScan, and w3af can be used, along with vulnerability search tools such as exploit-db to find exploits and online verification proofs (POCs).

The vulnerabilities mainly include: information disclosure vulnerabilities, logic vulnerabilities, cross-site scripting (XSS) and SQL injection vulnerabilities, CSRF, SSRF, file upload vulnerabilities, system vulnerabilities (Xu, 2017), Web Server vulnerabilities, Web application vulnerabilities, middleware vulnerabilities, incorrect server configuration, port service vulnerabilities, database server problems, database design problems, weak database passwords, communication security, plaintext transmission, tokens in cookie transmission, network protocol vulnerabilities, etc.

4.5 Vulnerability verification

Verify all exploitable vulnerabilities, build simulated real-world environments for attack testing, and combine the results from automated scanning tools with manual verification based on known vulnerability resources.

4.6 Information Analysis

Verified security vulnerabilities can be used to attack target programs. Different security vulnerabilities have different attack mechanisms. For different security vulnerabilities, further analysis and detailed attack plans are developed to ensure stable and comprehensive testing.

4.7 Exploit vulnerabilities to carry out attacks

Execute attacks to obtain internal information: infrastructure (network device information, network connections, VPNs, routing, network topology, etc.), further penetration, internal network intrusion, sensitive targets. Persistence is possible, but generally not required. Trace removal: clear relevant logs (access, operations) and uploaded files.

4.8 Organize data

Organize the penetration tools, code, POC, EXP, and adaptation methods used in the penetration process. Organize all information obtained in the penetration process. Organize the various vulnerabilities and vulnerability location information encountered in the penetration process (Zeng, Tang, & Qian, 2018), so as to form a report.

4.9 Write a report, propose solutions, and submit the report

Prepare penetration test reports, documenting the penetration test process, describing the project's security testing objectives, information gathering methods, vulnerability scanning tools and vulnerability information, attack plans, and actual attack results. Organize data according to the scope agreed upon with the client and generate reports. Analyze the causes, verification process, and impact of vulnerabilities. Propose solutions considering factors such as cost, offering reasonable, efficient, and secure solutions for all issues, and submit a test report.

5. Conclusion

Reasonable penetration testing methods and procedures can effectively assess the security of a system and propose reasonable improvement plans. Penetration testing simulates hacker intrusion into a system, obtains confidential information, forms a report on the intrusion process and detailed information, provides reports to users to identify system security threats and prompts security administrators to improve security policies to reduce security risks. Professional and systematic penetration testing can make penetration testing faster, clearer and more efficient, and more comprehensive. Even if the system is not breached, it can prove the security and stability of the test system. Penetration testing is a widely used security protection technology that helps improve the security of web applications. Penetration testing is an important line of defense for network security. A good penetration testing method can effectively improve the efficiency of penetration testing, give penetration engineers a clear process to further explore targets, protect the security environment, and safeguard national, ethnic and personal interests.

References

Halfond, W. G. J., Choudhary, S. R., Orso, A., et al. (2011). Improving penetration testing through static and dynamic analysis. *Software Testing, Verification and Reliability*, *21*(3), 195-214.

Li, L. (2018). Research on Penetration Testing Technology for Information Security of Train Control System [Doctoral dissertation, Beijing Jiaotong University].

Li, X. (2020). Research on SQL Injection Based on Web Penetration Testing. *Information and Computer (Theoretical Edition)*, *32*(3), 164-166.

Lü, J. (2016). Design and Implementation of a Network Penetration Testing Platform Based on VirtualBox [Master's thesis, South China Agricultural University].

Shi, S., & Chi, R. (2016). Computer Network Security Technology. Posts & Telecom Press.

Tang, C., & Yu, S. (2009). Evaluation of network security strategies based on security assurance capabilities. *Journal of Wuhan University (Natural Science Edition)*, *55*(1), 109-112.

Xu, F. (2017). Research on SQL Injection Vulnerability Test Case Generation Based on Context-Free Grammars [Master's thesis, Harbin Engineering University].

Zeng, P., Tang, Y., & Qian, L. (2018). Research on SQL Injection Based on Web Penetration Testing. *Network Security Technology and Application*, (12), 16-18.