What is Software Security Testing?

It is a type of software testing that ensures the software is free of any potential vulnerabilities or weaknesses, risks, or threats so that the software might not harm the user system and data.

By Security testing we can identify potential security risks and vulnerabilities in applications, systems and networks.

Security testing is an essential part of the software development lifecycle.

Why is Software Security Testing Required?

None of the users, business people, entrepreneurs, or organizations want to lose any information or data due to the security leaks of software. Just because a piece of software meets quality requirements related to functionality and performance does not necessarily mean that your web app software is secure. In today’s scenario, is a must to identify and address application security vulnerabilities to maintain the following:

- Security of information, databases, data history, and servers
- Customers’ trust and integrity
- Protection of web applications from future attacks

The goal of security testing

is to identify any weaknesses that could be exploited by attackers to gain access to sensitive data or interrupt system operations.

The Importance of security testing

- To identify the threats in the system.
- To measure the potential vulnerabilities of the system.
- To help in detecting every possible security risks in the system.
- To help developers in fixing the security problems through coding.

The main objectives of security testing are to:

- Identify vulnerabilities: Security testing helps identify vulnerabilities in the system, such as weak passwords, unpatched software, and misconfigured systems, that could be exploited by attackers.
- Evaluate the system's ability to withstand an attack: Security testing evaluates the system's ability to withstand different types of attacks, such as network attacks, social engineering attacks, and application-level attacks.
- Ensure compliance: Security testing helps ensure that the system meets relevant security standards and regulations, such as HIPAA, PCI DSS, and SOC2.
- Provide a comprehensive security assessment: Security testing provides a comprehensive assessment of the system's security posture, including the identification of vulnerabilities, the evaluation of the system's ability to withstand an attack, and compliance with relevant security standards.
- Help organizations prepare for potential security incidents: Security testing helps organizations understand the potential risks and vulnerabilities that they face, enabling them to prepare for and respond to potential security incidents.
- Identify and fix potential security issues before deployment to production: Security testing helps identify and fix security issues before the system is deployed to production. This helps reduce the risk of a security incident occurring in a production environment.

Security Testing: The Approach

While preparing and planning for security tests, a developer can take the following approaches:

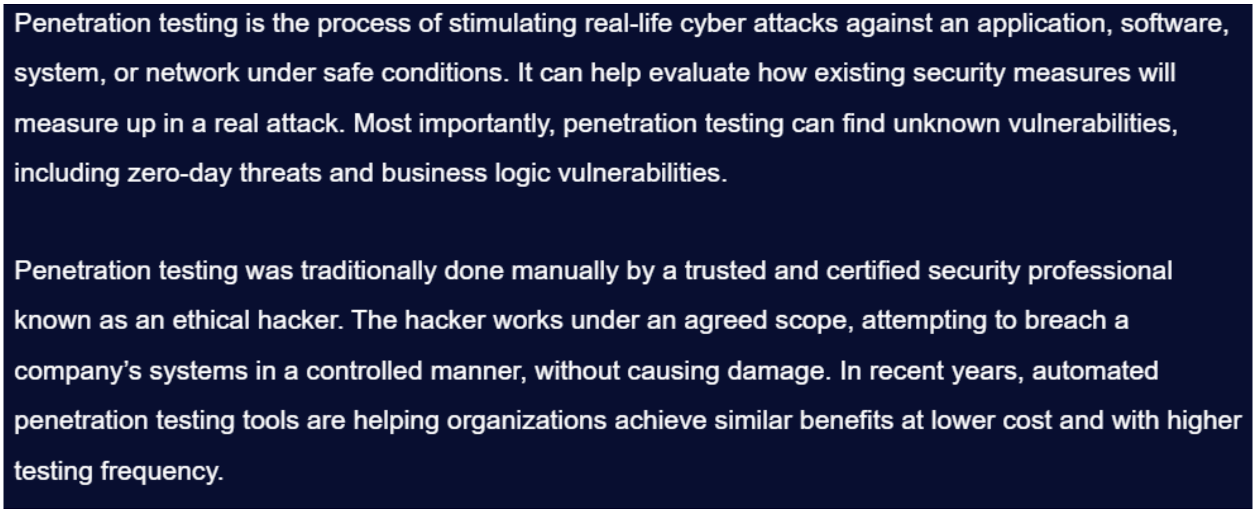
- **Architecture Study & Analysis:** The first step is understanding whether the software complies with the requirements.
- **Classify Threats:** List all potential threats and risk factors you must test.
- **Test Planning:** Run the tests based on the identified threats, vulnerabilities, and security risks.
- **Testing Tool Identification:** Software security testing tools for web applications; the developer needs to identify the relevant tools to test the software.

- **Test Case Execution:** After performing a security test, the developer should fix them manually or use any suitable open-source code.
- **Reports:** Prepare a detailed test report of the security tests you performed. It would contain a list of the vulnerabilities, threats, and issues resolved and the ones that are still pending.

Types of Security Testing

There are five different forms of security testing, each of which has a different methodology and purpose. Ideally, you will use a combination of these techniques as needed.

1. **Penetration testing (ethical hacking)** simulates an actual cyberattack to test specific systems for vulnerability.



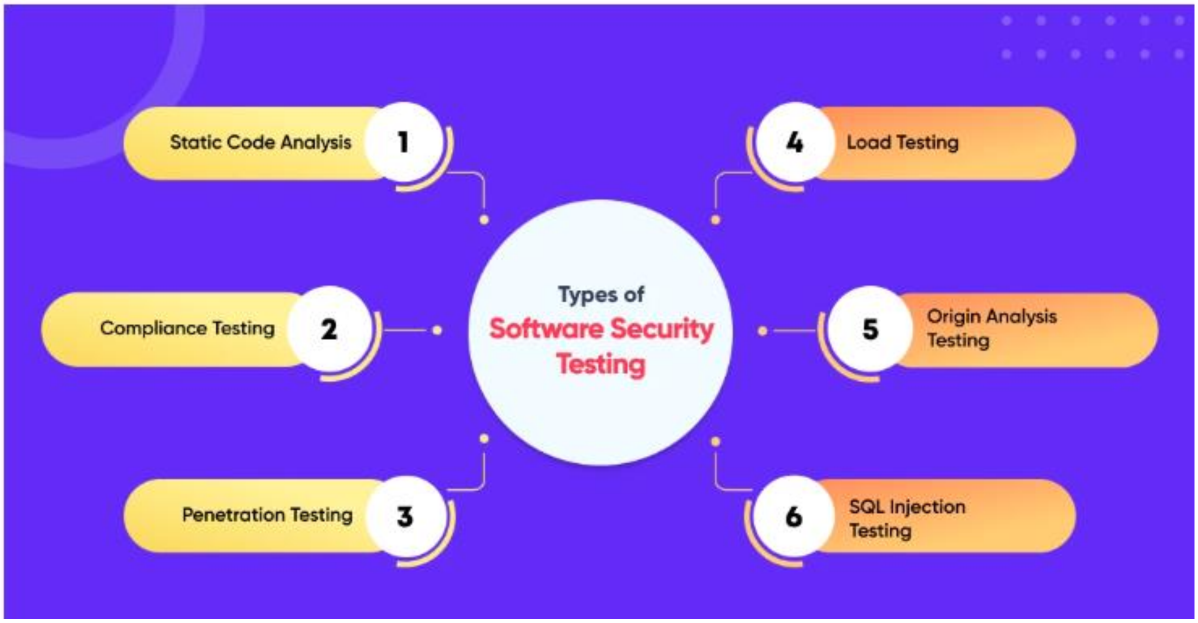
Penetration testing is the process of stimulating real-life cyber attacks against an application, software, system, or network under safe conditions. It can help evaluate how existing security measures will measure up in a real attack. Most importantly, penetration testing can find unknown vulnerabilities, including zero-day threats and business logic vulnerabilities.

Penetration testing was traditionally done manually by a trusted and certified security professional known as an ethical hacker. The hacker works under an agreed scope, attempting to breach a company’s systems in a controlled manner, without causing damage. In recent years, automated penetration testing tools are helping organizations achieve similar benefits at lower cost and with higher testing frequency.

2. **Security scanning** either manually or automatically looks for system flaws in new code.
3. **Vulnerability scanning** checks your software against lists of known vulnerabilities.
4. **Security auditing** is a line-by-line examination of your code that reveals any security holes that you may have previously missed.
5. **Security risk assessments** focus on reducing external threats, categorizing them as “low,” “medium” or “high.”

Security Testing Methods / Techniques / Approaches

The most common web app software security tests from a few years ago might not be practical today. Let’s look at the different security tests that are relevant today. We follow several web application security testing types simultaneously.



1. Static Code Analysis

This is the oldest approach and the first type of security testing most developers perform. We can perform this test manually, and developers can read through the code to find potential security flaws.

2. Compliance Testing

It’s essential for software to meet a client’s predefined policies, and we ensure this by running compliance tests. In these tests, we analyze a piece of software by comparing it with the actual configurations.

3. Penetration Testing

This software testing involves simulation attacks against newly designed software to identify the weak points. Once detected, a developer fixes the bugs within the codes.

4. Load Testing

This test measures how a piece of software performs under a heavy load. The reason behind this test is Distributed-Denial-of-Service (DDoS), an attack that aims to disrupt application availability by application or its host infrastructure with traffic or other requests.

5. Origin Analysis Testing

The popularity of open-source software has grown in the past few years. This software security testing helps developers and security admins determine where a given piece of code originated. Such testing becomes relevant when some of your source code has come from a third-party project or repository.

6. SQL Injection Testing

SQL Injection test can be done for apostrophes, brackets, commas, or quotation marks. These simple errors lead to attacks by spammers. SQL injection attacks are critical because attackers can enter the server database and get vital information.
It is not a definitive list of security tests. Enterprises might perform other security tests like Risk Assessment, Posture Assessment, Security Auditing, and even Ethical Hacking.

Security Testing Principles

The **seven** main principles of security testing are:

- **Confidentiality** – limiting access to sensitive access managed by a system.
-
- **Integrity** – ensuring that data is consistent, accurate, and trustworthy throughout its lifecycle and cannot be modified by unauthorized entities.
-
- **Authentication** – ensuring sensitive systems or data are protected by a mechanism that verifies the identity of the individual accessing them.
-
- **Authorization** – ensuring sensitive systems or data properly control access for authenticated users according to their roles or permissions.
-

- **Availability** – ensuring that critical systems or data are available for their users when they are needed.
-
- **Non-repudiation** – ensures that data sent or received cannot be denied, by exchanging authentication information with a provable time stamp.
-
- **Resilience:** a system’s total resistance to attack

Security Testing Scenarios

Below are some common use cases for security testing:

- **Verifying the server’s TLS/SSL configuration**
- **Testing for known vulnerabilities in the web application, such as SQL injection, XSS, and CSRF**
- **Performing static code analysis to verify that security best practices are being followed**
- **Testing for the presence and strength of authentication and authorization controls**
- **Performing penetration testing to identify potential weaknesses in the security architecture**
- **Ensuring that access control mechanisms are in place and correctly configured**
- **Performing data leakage tests to identify any sensitive information that may be inadvertently leaked**
- **Testing for the presence of a secure audit trail**
- **Verifying the strength and complexity of passwords**
- **Assessing whether the application is adequately protected against denial of service attacks**

Software Security Testing Responsibilities

A software security tester’s key responsibility is to protect the software data from unauthorized access and ensure if any breach happens, they can easily counter it.
Here are a few other responsibilities a software security tester has to do:

- Working with their clients from the beginning to understand their testing requirements, such as the types of devices the software will operate on.
- Planning and creating penetration methods, scripts, and tests.
- Conducting software remote and on-site testing to identify and fix security issues.
- Simulating security breaches to gauge whether your software can withstand them or not.
- Listing reports and recommendations to the management or the development team to fix them as soon as possible.
- Continuously renewing the company’s incident response and emergency recovery methods.