PERFORMANCE TESTING

The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application. It is a subset of performance engineering and is also known as ***"Perf Testing"***.

"Checking the behaviour of an application by applying some load is known as performance testing."

ATTRIBUTES OF PERFORMANCE TESTING:

- **Speed** – Determines whether the application responds quickly
- **Scalability** – Determines the maximum user load the software application can handle.
- **Stability** – Determines if the application is stable under varying loads

When we use performance testing?

- We will do performance testing once the software is stable and moved to the production, and it may be accessed by the multiple users concurrently, due to this reason, some performance issues may occur. To avoid these performance issues, the tester performs one round of performance testing.

PERFORMANCE TEST WORK FLOW:

- Requirement gathering.
- Tools selection.
- Performance test plan.
- Performance test development.
- Performance test modelling.
- Test execution.
- Analysis.
- Report.

1. **Requirements Gathering & Analysis**
   Requirement gathering & Analysis is the first and the most important phase in testing. In this phase, the collection of all the information about the application from the client and other stakeholders takes place.

The key questions asked from the stakeholders are:

- Details of the application architecture and design
- The anticipated concurrent user load
- Any performance benchmarks or performance acceptance criteria
- The knowledge of high-performance critical workflows.

2. Testing Tool Selection

There are different types of performance test tool. But we should know which type of testing tool we need and how to select it.

Application type and design:

One of the primary things to consider while selecting a tool is the type and design of the application.

Ex. for performance testing of web service we may decide selecting a tool LoadUI which is a custom tool for load testing of SOAP web service.

Project Estimate:

Based on the estimate suggested for performance testing, we need to select between free tools and different licensed tools.

Tool Specific Requirement:

The given requirements of a performance test are also considered while choosing a test tool, for example, there may be a task of performing load test with a number of users, few available tools do not effectively handle this much load. So, a tool should be capable of efficiently handling the load that gets selected for testing.

Expertise required for scripting:

The expert QA also helps to select the test tool for performance testing.

3. Performance Test Planning:

The test planning phase includes the blueprint of the whole performance testing process.

The major steps performed in this phase are:

• Performance test model and environment setup
• use case scenarios to be scripted
• total user load to be simulated
• Identifying the test configuration like duration, the number of user distribution between different scenarios, etc.
• Understanding of the test result metrics, using different listeners for gathering and analyzing the test results .

4. **Performance Test Development**

• Use cases are created for the functionality identified in the test plan as the scope of PT.
• These use cases are shared with the client for their approval. This is to make sure the script will be recorded with the correct steps.
• Once approved, script development starts with a recording of the steps in use cases with the performance test tool selected during the POC (Proof of Concepts) and enhanced by performing Correlation (for handling dynamic value), Parameterization (value substitution) and custom functions as per the situation or need. More on these techniques in our video tutorials.
• The Scripts are then validated against different users.
• Parallel to script creation, the performance team also keeps working on setting up the test environment (Software and hardware).
• The performance team will also take care of Metadata (back-end) through scripts if this activity is not taken up by the client.

5. Performance Test Modeling :

Performance Load Model is created for the test execution. The main aim of this step is to validate whether the given Performance metrics (provided by clients) are achieved during the test or not. There are different approaches to create a Load model. “Little’s Law” is used in most cases.

6. Test Execution

The test script is executed for a predefined time mentioned during the planning phase. At the same time, progress is
interrogated along with the different aspects of the server like CPU usage, memory consumption, etc.

6. Test Result Analysis

This phase includes the result analysis. It combines the test scenarios for summarizing the different performanceattributes of the application and finding the performance queue. This test results information about the applicationstate is shared with all the relevant stakeholders.

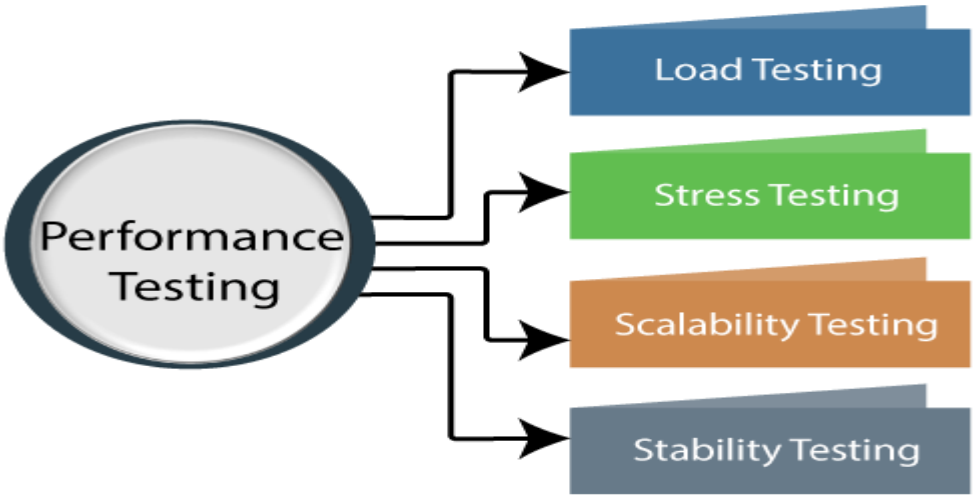
Some of the best practices that help the Result Analysis process:

• A unique and meaningful name to every test result – this helps in understanding the purpose of the test.
• Include the following information in the test result summary:
• Reason for the failure/s
• Change in the performance of the application compared to the previous test run
• Changes made in the test from the point of application build or test environment.
• It's a good practice to make a result summary after each test run so that analysis results are not compiled every
time test results are referred.
• PT generally requires many test runs to reach the correct conclusion.
• It is good to have the following points in result summary:
    - o Purpose of test
    - o Number of virtual users
    - o Scenario summary
    - o Duration of test
    - o Throughput
    - o Graphs
    - o Graphs comparison
    - o Response Time
    - o Error occurred
    - o Recommendations

8. Report

Test results should be simplified so the conclusion is clearer and should not need any derivation. Development.

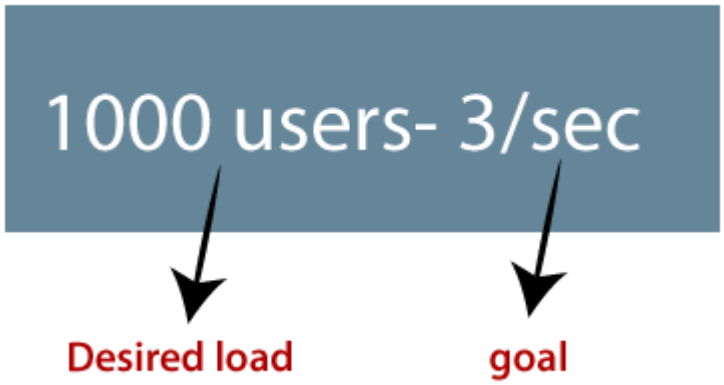
Types of Performance Testing



LOAD TESTING:

The load testing is used to check the performance of an application by applying some load which is either less than or equal to the desired load is known as load testing.

For example: In the below image, **1000 users** are the **desired load**, which is given by the customer, and **3/second** is the **goal** which we want to achieve while performing a load testing.



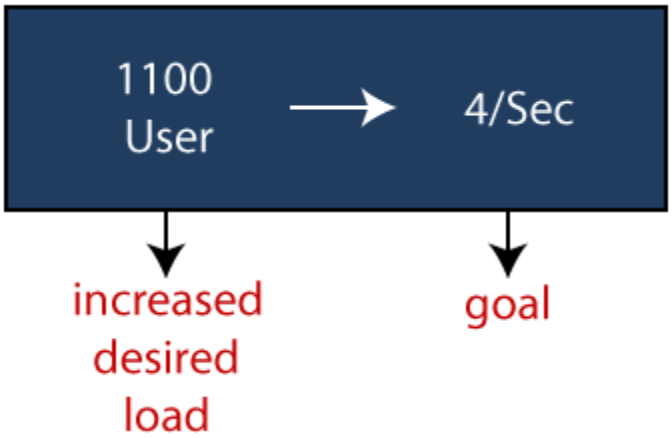
Tools

- Apache JMeter.
- LoadView.
- Micro Focus LoadRunner Cloud.
- k6.

STRESS TESTING:

The stress testing is testing, which checks the behavior of an application by applying load greater than the desired load.

For example: If we took the above example and increased the desired load 1000 to 1100 users, and the goal is 4/second. While performing the stress testing in this scenario, it will pass because the load is greater (100 up) than the actual desired load.



Tools:

- Apache JMeter.
- Load runner.
- Web load.
- Load UI.

SOAK TESTING or ENDURANCE TESTING:

is done to make sure the software can handle the expected load over a long period of time. Soak testing main goals is to check for system problem such as memory leaks. Its also called endurance testing.

TOOLS:

- Apache JMeter.
- Load runner.
- Web load.
- Load UI.

SPIKE TESTING:

tests the software's reaction to sudden large spikes in the load generated by users.

EXAMPLE

Another such example is Good Friday sales online, where the discounts are available only for a couple of hours.

TOOLS

- Apache JMeter.
- Load runner.
- Web load.
- Load UI.

VOLUME TESTING:

Under Volume Testing large no. of. Data is populated in a database, and the overall software system's behavior is monitored. The objective is to check software application's performance under varying database volumes.

Attributes of volume

- Response time.
- Data loss
- Data storage.

TOOLS

- Nosql map
- DBFIT
- HAMMER DB.

EXAMPLE:

Increasing product or user database on a website − While loading a number of items in a shopping website database, volume testing is important to conduct to make sure that the infrastructure can handle extended data load.

SCALABILITY TESTING:

Checking the performance of an application by increasing or decreasing the load in particular scales (no of a user) is known as **scalability testing**. Upward scalability and downward scalability testing are called scalability testing.

Scalability testing is divided into two parts which are as follows:

- **Upward scalability testing**
- **Downward scalability testing**

Upward scalability testing

It is testing where we **increase the number of users on a particular scale** until we get a crash point. We will use upward scalability testing to find the maximum capacity of an application.

Downward scalability testing

The downward scalability testing is used when the load testing is not passed, then start **decreasing the no. of users in a particular interval** until the goal is achieved. So that it is easy to identify the bottleneck (bug).

Tools:

- Apache JMeter.
- Load runner.
- Web load.
- Load UI.

SECURITY TESTING

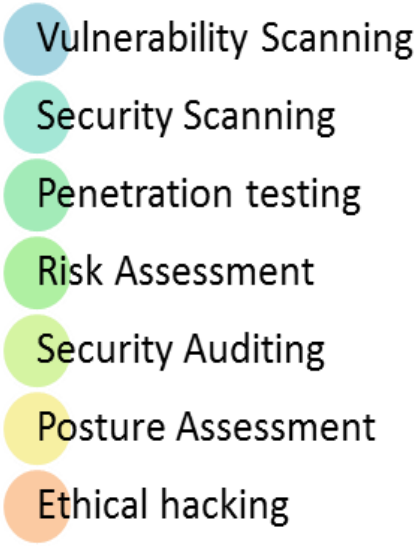
Security Testing is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders. The purpose of software testing to find weakness in software implementation, configuration or deployment.

The main purpose of security testing to determine whether a system meets its specified security requirements of security function, performance limitation and software reliability.

Principle of Security Testing: Below are the six basic principles of security testing:

- Confidentiality- (information should be access authorised person)
- Integrity- (giving the right data)
- Authentication- (identifying the user)
- Authorization- (giving permission)
- Availability- (whenever we are wanting it should ready)
- Non-repudiation- (no delay, denial)

Types of security testing:



Vulnerability Scanning:(discovering, analysing and reporting

This is done through **automated software to scan** a system against known vulnerability signatures.

Tools:

Wireshark.
Solar wind.
Beyond trust.
Nmap.

Security Scanning:

It involves identifying network and system weaknesses, and later provides solutions for reducing these risks. This scanning can be performed for both Manual and Automated scanning.

Tools:

- Wireshark.
- Solar wind.
- W3af
- Rapid7

Penetration testing:

This kind of testing simulates an attack from a malicious hacker. This testing involves analysis of a particular system to check for potential vulnerabilities to an external hacking attempt.

Tools:

- Wireshark.
- W3af.
- Vulnerability scanner.
- Nmap.

What is penetration testing?

A penetration test (pen test) is an authorized simulated attack performed on a computer system to evaluate its security. Penetration testers use the same tools, techniques, and processes as attackers to find and demonstrate the business impacts of weaknesses in a system.

Risk Assessment:

This testing involves analysis of security risks observed in the organization. Risks are classified as Low, Medium and High. This testing recommends controls and measures to reduce the risk.

Tools:

Matrix.

Decision-making.

Logic manner.

Bow-tie.

Security Auditing:

This is an internal inspection of Applications and Operating systems for security flaws. An audit can also be done via line-by-line inspection of code.

Tools:

Wireshark.
Solar wind.
W3af.
Nmap.

Ethical hacking:

It's hacking an Organization Software system. Unlike malicious hackers, who steal for their own gains, the intent is to expose security flaws in the system.

Tools:

Wireshark.
Nmap.
Traceroute.
Hashcat.

Posture Assessment:

This combines Security scanning, Ethical hacking and Risk Assessments to show an overall security posture of an organization.

Application Security Testing (AST).

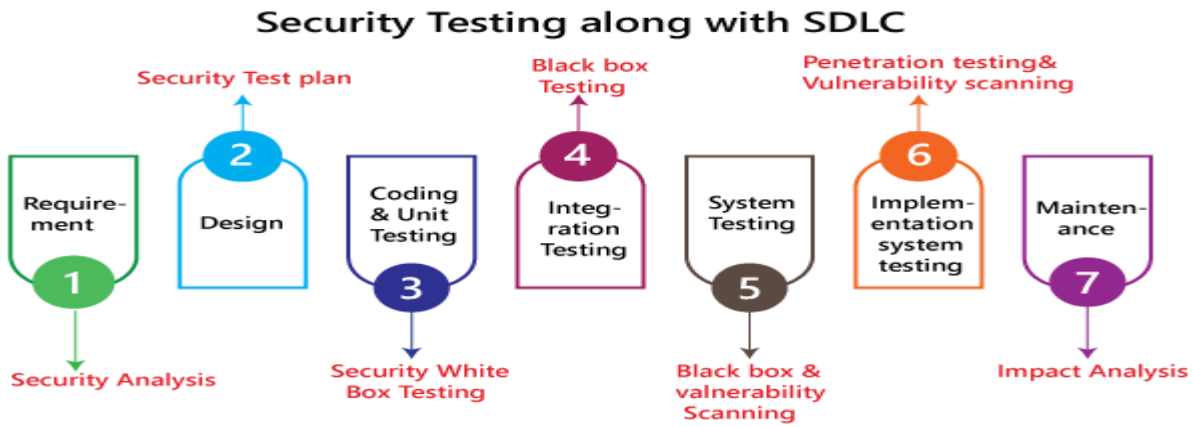
describes methods organizations can use to find and eliminate vulnerabilities in software applications. These methods involve testing, analyzing, and reporting on the security posture of a software application throughout the software development lifecycle (SDLC).

The main goal of AST is to prevent software vulnerabilities before applications are released to the market, and failing that, quickly identify and remediate them in production. Successful AST results in more robust, secure source code, greater visibility over application security issues, and improved protection against internal and external threats.

Web Application Security Testing

The goal of web application security testing is to determine whether a web application is vulnerable to attack. It covers a variety of automatic and manual techniques.

How do we perform:



1.Requirement stage

Security Procedures: In the requirement phase of SDLC, we will do the security analysis of the business needs and also verify that which cases are manipulative and waste.

2.Design stage

Security Procedures: In the design phase of SDLC, we will do the **security testing for risk** exploration of the design and also embraces the security tests at the development of the test plan.

3. Development or coding stage

Security Procedures: In the coding phase of SDLC, we will perform the white box testing along with static and dynamic testing.

4. Testing (functional testing, integration testing, system testing) stage

Security Procedures: In the testing phase of SDLC, we will do one round of **vulnerability scanning** along with black-box testing.

5. Implementation stage

Security Procedures: In the implementation phase of SDLC, we will perform **vulnerability scanning** again and also perform one round of **penetration testing**

6. Maintenance stage

Security Procedures: In the Maintenance phase of SDLC, we will do the **impact analysis** of impact areas.

Key terms are used in security testing:

Vulnerability

(Application vulnerability is a known or unknown weakness that hackers can use)

Zero days vulnerability means- therefore, it has no fix for it.

Sql injection:

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

URL manipulation:

Some web applications have an additional feature to communicate between the browser and the server in the URL. Changing some information in the URL may sometimes lead to unintended behavior by the server and this termed URL Manipulation.

XSS (Cross-Site Scripting)

When a user inserts HTML/client-side script in the user interface of a web application, this insertion is visible to other users and it is termed as XSS.

Example test case:

- A password should be in encrypted format
- Application or System should not allow invalid users
- Check cookies and session time for application
- For financial sites, the Browser back button should not work.

Methodologies/ Approach / Techniques for Security Testing

- **Tiger Box**: This hacking is usually done on a laptop which has a collection of OSs and hacking tools. This testing helps penetration testers and security testers to conduct vulnerabilities assessment and attacks.
- **BLACK BOX**: Tester is authorized to do testing on everything about the network topology and the technology.
- **Grey Box**: Partial information is given to the tester about the system, and it is a hybrid of white and black box models.

security Testing Roles

- Hackers – Access computer system or network without authorization
- Crackers – Break into the systems to steal or destroy data

- Ethical Hacker – Performs most of the breaking activities but with permission from the owner
- Script Kiddies or packet monkeys – Inexperienced Hackers with programming language skill.

Conclusion:

Security testing is the most important testing for an application and checks whether confidential data stays confidential. In this type of testing, tester plays a role of the attacker and play around the system to find security-related bugs. Security Testing is very important in Software Engineering to protect data by all means.