Security Testing

****

*Security Testing is an essential component of software testing that is used to identify weaknesses, hazards, or dangers in software applications.

*It also assists us in preventing malicious(harm)attacks from outsiders and ensuring the security of our software applications.

Principle of Security testing:

1.Confidentiality

2.Integrity

3.Authentication

4.Authorization

5.Availability

6.Non-repudiation

Security testing tools:

1.Metasploit

2.Burp suite

3.Sqlmap

4.OWASP ZAP

5.Nessus

6.Nmap

7. Wireshark

8. Security AppScan

Confidentiality:

It is a method of safety that prevents data leaks from outsiders because it is the only way to ensure the security of our data.

Integrity:

*We will secure any data that has been altered by an unofficial people.

*The basic goal of integrity is to allow the receiver to manage the data provided by the system.

Authentication:

*Authentication is the process of verifying who a user is,

*In Other words, The process of authentication involves authenticating a person's identity and tracing the origin of a product required to gain access to private information or the system.

* Example: Accessing a user account on a website or a service provider such as Facebook or Gmail.

Authorization:

*Authorization is the process of verifying what they have access to.

*In Other words, It is the process of determining if a client is allowed to execute an action as well as receive services.

*Access control is an example of Authorization.

Availability:

The data must be kept by an official, and they also ensure that the data and statement services will be available when we need them.

Non-repudiation:

*It refers to digital security and is a method of ensuring that the sender of a communication cannot deny having transmitted the message and that the recipient cannot deny having received the message.

* Non-repudiation is used to ensure that a communication was delivered and received by the person claiming to have sent and received the message.

Types of Security testing:

1.Security Scanning

2.Risk Assessment

3.Vulnerability Scanning

4.Penetration testing

5.Security Auditing

6.Ethical hacking

7.Posture Assessment

Key Terms used in Security Testing:

Vulnerability:

This is the weakness of the web application. The cause of such “weakness” can be due to the bugs in the application, an injection (SQL/ script code), or the presence of viruses.

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.

SQL injection:

This is the process of inserting SQL statements through the web application user interface into some query that is then executed by the server.

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.

Sample Test Cases for Security Testing:

1. Try to directly access bookmarked web page without login to the system.

2. Verify that system should restrict you to download the file without sign in on the system.

3. Verify that previous accessed pages should not accessible after log out i.e. Sign out and then press the Back button to access the page accessed before.

4. Check the valid and invalid passwords, password rules say cannot be less than 6 characters, user id and password cannot be the same etc.

5. Verified that important i.e. sensitive information such as passwords, ID numbers, credit card numbers, etc should not get displayed in the input box when typing. They should be encrypted and in asterix format.

6. Check Is bookmarking disabled on secure pages? Bookmarking Should be disabled on secure pages.

7. Check Is Right Click, View, Source disabled? Source code should not be visible to user.

8. Is there an alternative way to access secure pages for browsers under version 3.0, since SSL is not compatible with those browsers?

9. Check does your server lock out an individual who has tried to access your site multiple times with invalid login/password information?

10. Verify the timeout condition, after timeout user should not able to navigate through the site.

11. Check Are you prevented from doing direct searches by editing content in the URL?

12. Verify that relevant information should be written to the log files and that information should be traceable.

13. In SSL verify that the encryption is done correctly and check the integrity of the information.

14. Verify that restricted page should not be accessible by user after session time out.

15. ID / password authentication, the same account on different machines cannot log on at the same time. So at a time only one user can login to the system with a user id.

16. ID / password authentication methods entered the wrong password several times and check if the account gets locked.

17. Add or modify important information (passwords, ID numbers, credit card number, etc.). Check if it gets reflected immediately or caching the old values.

18. Verify that Error Message does not contain malicious info so that hacker will use this information to hack web site.

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

The Advantages of Security Testing:

1. Cost Saving

2. Protection from external attacks

3. Saves Time

4. Reduced intrinsic business risk

5. Guaranteed quality product

6. Increase the demand for software

7. Overall business growth

Performance Testing

****

Definition:

*Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.

*In other words, Checking the behavior of an application by applying some load is known as performance testing.

Performance Testing Attributes:

1.Speed: It determines whether the software product responds rapidly.

2.Scalability: It determines amount of load the software product can handle at a time.

3.Stability: It determines whether the software product is stable in case of varying workloads.

4.Reliability: It determines whether the software product is secure or not.

Performance Testing Techniques:

1.Load testing:

*It is the simplest form of testing conducted to understand the behaviour of the system under a specific load.

*Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.

2.Stress testing:

It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

3.Soak testing:

*Soak Testing also known as Endurance testing, is performed to determine the system parameters under continuous expected load.

*During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues.

*The main aim is to discover the system's performance under sustained use.

4.Spike testing:

*Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system.

*The main aim is to determine whether the system will be able to sustain the workload.

Performance testing process:

The performance testing cannot be done manually since:

*We need a lot of resources, and it became a costlier approach.

*And the accuracy cannot maintain when we track response time manually.

The Performance testing process will be completed in the following steps:

1.Identify performance scenarios.

2.Plan and design performance test script.

3.Configure the test environment & distribute the load.

4.Execute test scripts.

5.Result.

6.Analysis result.

7.Identify the Bottleneck.

8.Re-run test.