Testing Methodologies on Web Applications

1. Functional Testing: This type of testing tests the user requirements are fulfilled or not and test functionalities with some problems. It is divided into parts:
    - **Unit testing:** In these developers do unit testing which tests and helps in debugging if needed in effective time.
    - **Integration testing**: In these developers combine units testing because to see the interaction between them.
    - **System testing**: It ensures that the product is fulfilled the specified and technical requirements. This testing is done on whole product.
    - **Regression testing**: It maintains the overall reliability and stability of software.
    - **Smoke testing**: It is basic step before validate testing. It tests basic requirement.
    - **Validate testing**: It done due to ensures that the end user requirement is fulfilled or not. It is the last stage.

2. **Non-Functional Testing**: It mainly focuses on performance, usability, scalability and reliability. It can be follows as:

- **Compatibility testing:** It tests the web application is easily compatible with O.S, Network, Browsers or not. It takes care the user requirement.
- **Performance testing**: It ensures that the web application is performance is good like response time and efficiency. It has major parts:
  - ✓ **Load testing**: In this test that the web application is sustain user load and performance is good as always.
  - ✓ **Stress testing:** It is mainly concern about the potential conditions. It tests the performance mainly.
- **Usability testing**: It concern about the best user experience with application and also, about the interaction.
- **Security testing:** It ensures about web application is safe from threats and user data is safe so they can trust product.

- **UI testing**: It is for easy interface for best user experience.

What is DevSecOps

DevSecOps stands for Development, Security and operations. It is a framework that put security in each phase of software development life cycle. It minimizes the risk of releasing code and give confidence to team from threats. It is extension of DevOps Model. It is necessary to known to all security isn't slower the process instead it saves debugging time for a lot of issues.

Difference between SAST and DAST

SAST stands for Static Application Security testing

- It is white box testing.
- This type of testing is developer approach testing and application is test from inside out.
- Identifying and fixing bug is easier with little cost.

- It requires source code to perform testing and it doesn’t identify time related issues.
- It scans static code that’s why it is known as static application security testing and it done in early stages of software development cycle.
- It may be having long duration of testing experience by tester.

DAST stands Dynamic Application Security Testing

- It is black box testing.
- It known as hackers approach and test perform from outside to in.
- It is expensive approach as it is find vulnerability in end.
- It can find issue in run time environment.
- As it is scanning dynamic code so it is dynamic application security testing and perform at end stage of software development life cycle.
- It doesn’t need source code for execution and tester doesn’t knowledge about design or framework.
- It performs Fastly.

Various SAST and DAST tools

SAST tools:

SonarQube -> Support many languages.

Checkmarx -> It also support multiple language.

Veracode -> Cloud base platform.

Codacy -> Provide report with GitHub.

Brakeman -> Specific for Ruby on Rail framework.

DAST tools:

Owasp ZAP -> Supports automated and manual scanning.

Burp suite -> Offers tools like Intruder, Scanner.

Nessus -> Detects Security issues in network etc.

Snyk -> Offers both SAST and DAST capabilities.

Appscan -> Integrate with DevOps.

How to Secure Web Application Efficiently

- Implementation of security in each phase of Software Development life cycle with the help of DevSecOps.
- Monitoring the user activities or logs and proper access control.
- Proper sanitization of the input.
- Use multi-Factor authentication.
- Do Regular Security testing and be updated not outdated.
- Use web application firewalls to block unwanted traffic.
- Set up data encryption and manages security configuration.

Lab 1 and Lab 2

Lab: OS command injection, simple case

APPRENTICE

LAB Solved

This lab contains an OS command injection vulnerability in the product stock checker.

The application executes a shell command containing user-supplied product and store IDs, and returns the raw output from the command in its response.

To solve the lab, execute the `whoami` command to determine the name of the current user.

ACCESS THE LAB

Solution

Community solutions

Lab: Manipulating WebSocket messages to exploit vulnerabilities

APPRENTICE

LAB Solved

This online shop has a live chat feature implemented using WebSockets.

Chat messages that you submit are viewed by a support agent in real time.

To solve the lab, use a WebSocket message to trigger an `alert()` popup in the support agent's browser.

ACCESS THE LAB

Solution

Community solutions