Pragmatic AppSec Testing

Course Objective

Web application security has a broad scope that spans from network communication to browser behaviors to backend applications and finally to database servers. Validating security of all these components can be a daunting task and take a considerable effort. Penetration is the most prevalent testing method used today for validating web application security. The question is, "does it cover all the basis?" Penetration testing is a black-box type testing that a QA engineer applies from the hacker's perspective. While it provides a comfort level, it does not ensure that the application has been developed with security in mind and that it meets the three basic requirements of security namely, **C**onfidentiality, **I**ntegrity, and **A**vailability (CIA). The CIA framework builds intrinsic security and thus ensures an increased confidence level. This framework should be complimented with the penetration testing.

The objective of this course is to learn how to align the security validation of a web application with the three basic elements of security namely, **C**onfidentiality, **I**ntegrity, and **A**vailability (CIA). The test effectiveness can be achieved by analyzing the requirements of each element and identifying the potential breaches that can compromise the security. The efficiency should be built by relating these breaches with the known OWASP Top 10 and other vulnerabilities and, leveraging that knowledge to identify the testing approach - static and dynamic. The student will learn how to develop requirement-based test modules and optimize the test suites before deriving the test cases. The well founded testing approaches such as unit, module, system, and regression testing will be explored as it should be applied to a specific web application. Another integrated part of learning will be, how to prepare the progress report and the executive security summary report to make the release decisions.

Intended Audience

The course is intended for DevOps (developers, security testing professionals, and support groups) to equip them with an analytical approach to create and deploy a web application that is secure and has minimal security risk. It is ideally suited for software development and software quality professionals.

Instructional Methods

The course is offered as a TWO DAY workshop with group discussions and role playing where the participants work in small teams. Each team is dedicated to explore a unique CIA element end-to-end thereby, determining the root cause, identifying the related OWASP Top 10 vulnerability and an appropriate testing methodology. The students then develop a testing technique (static and/or dynamic code analysis) and integrate it with the SDLC, especially to an iteration in an agile development environment. Each team discusses its thought process for its problem analysis and gets feedback from the other teams. By the end of the workshop, a comprehensive understanding of how to test each element of security is understood and documented as an example.

Course Overview

The course starts with building an appreciation for the three elements, Confidentiality, Integrity, and Availability, of security. The requirements for each element are established and how not meeting a requirement has been exploited by the hackers leading to the OWASP Top 10. The requirements are then customized according to the application and the course focuses on developing an acceptance criteria and a test plan including both static and dynamic code analysis. The course then takes the students through a process on how to integrate the testing with the SDLC. In addition, the course provides an in-depth discussion of when and how to incorporate the prominent tools such as ZAP and Burp Suite in the application development.

Course Outline

1. Discuss CIA (Confidentiality, Integrity, Availability) elements of security and establish requirements for each element
    a. Confidentiality Requirements - Access Control (Identification, Authentication, Authorization, Audit Trails)
        i. Data Classification
        ii. Identification – Software (Login ID), Hardware (Card Reader etc.)
        iii. Authentication – Biometric or Password
            1. Password – Crackability (complexity), # of Erroneous Tries, Aging, and Recovery Mechanisms
        iv. Authorization – Protection of Objects form subjects and processes, Reviewing Default System Configuration
        v. Audit – Logs (organization to support usability such as searches)
        vi. Known OWASP Top 10 vulnerabilities that impact confidentiality - SQL Injection, MITM Attack, Broken Authentication and Session Management
    b. Integrity Requirements
        i. Only Authorized Modification, Deletion, and Disclosure when data is stationary
        ii. No alteration or exposure of data during transport
        iii. Known OWASP Top 10 vulnerabilities – Injection (SQL, Command Line, and code Injection)
    c. Availability Requirements –
        i. Available as needed (24x7x365 or as per other criteria)
        ii. Redundancy
        iii. Known OWASP Top 10 Vulnerabilities - DDOS
2. Development of Validation Strategy
    a. Review of company historical data to establish vulnerability trends (may also consider external similar areas)
    b. Assessment and prioritization of stories with respect to CIA
    c. Establishing specific static code analysis – code review focus (input validation, encryption, buffer overflow etc.)
    d. Dynamic code analysis – Scanning for vulnerabilities, Unit testing, System testing, OWASP Vulnerability Testing
    e. Attack Surface Analysis
3. Integration into SDLC

    a. Generic Iteration
        i. Identification and prioritization of vulnerable stories (STRIDE or DREAD Methodology)
        ii. Establishing acceptance criteria for each story
        iii. Identify test cases for each story based upon acceptance criteria
        iv. Reporting test results
    b. Release Iteration
        i. Assessing scope of regression
        ii. Identifying critical vulnerability testing scope
        iii. Reporting security status
4. Use of Tools
    a. Role of tools in web application development
    b. Overview of web scanning tools (ZAP and Burp Suite)
    c. Tool setup
    d. Different ways to use the tools
    e. Interpretation of results and activities to follow

The end product of the course will be an understanding of the following aspects of a secure product development:

1. Establishing security requirements and acceptance criteria based upon CIA elements
2. Developing an effective and efficient test plan that includes both static and dynamic testing
3. Evaluation of security status at the end of each iteration as a release requirement
4. Security assessment and reporting prior to release– Evaluation of overall application security and publishing a report.
5. Efficient use of tools

Course Logistics: (may vary depending upon the environment)
When requested by the client, the course will be offered on the client facility. Following are the high level setup requirements:

- The Instructor will use his own system to present workshop material (slides etc.)
- Internet access outside the client network for illustration of models and concepts such as, attack surface analysis and scanners will be facilitated by the client
- The instructor will provide hard copies of the learning materials.

Prerequisites:
Basic understanding of web application security and the agile product development methodology.

Keywords: Cyber Security, Web Application Security, AppSec, Software Development, Software Engineering, Product Lifecycle, Software Process Improvement, Software Development Lifecycle