Absolute AppSec - Practical Secure Code Review (2 days)

Overview

The Practical Secure Code Review training course is designed to teach developers and security professionals a repeatable process for reviewing source code for security flaws. It addresses multiple common challenges in modern secure code review, including quickly distilling an application, pull request, or feature into functional and security aspects. Students will be able to build personal secure code review techniques by gleaning from our past adventures in performing hundreds of code reviews and the lessons we've learned along the way. We will share a proven methodology to perform security analysis of any source code repository and identify security flaws, no matter the size of the code base, or the framework, or the language.

Requirements

- Laptop with wireless and virtual machine (VMWare/Virtual Box) capabilities.
- Attendees should be familiar with the development process (SDLC) and where security code reviews fit into the process.
- Attendees must have experience using an IDE, running command-line tools, and be able to read application source code.
- Attendee must have knowledge of the OWASP Top 10 and other common vulnerabilities.

Past Training Experience

Seth Law & Ken Johnson are experienced trainers who have taught versions of this course at multiple conferences around the world over the past 6 years, including OWASP AppSec USA/EU, KernelCon, AppSec Day, Black Hat, and DEF CON.

Detailed Training Description

Learn a proven methodology for discovering vulnerabilities in code through secure code reviews against any language or framework, no matter the amount of code AND learn how to optimize processes and discovery utilizing your own RAG AI toolset. Whether analyzing code as a consultant, internal resource, or bug bounty researcher, enhance your bug-hunting techniques and code review skills using a strategy surpassing security review checks covered by language-specific guidance and automated tools plagued by false positives and learn how an emerging technology (AI) can augment your manual approach.

During the training, you will learn and practice a methodology developed by Seth and Ken (co-hosts of the Absolute AppSec podcast) to find bugs in hundreds of code bases, including web3, mobile, and web applications and also learn how to utilize AI technologies such as Langchain, Vector Databases, and Retrieval Augment Generation (RAG). Students gain the confidence to take on code-review projects, knowing how to organize their limited time, avoiding unnecessary time sinks and focusing on an application's security-relevant files and functions.

Training Outline/Agenda

Day 1 - Theory:

- **Overview (1 hour)**
    - Introductions
    - Philosophy
    - What to Expect
    - The Circle-K Framework
    - Approach
    - Tools/Lab Setup
    - OWASP Top 10
- **Code Review Methodology Overview (30 mins)**
    - Introduction to Methodology
    - General Code Review Principles
    - Application Overview & Risk Assessment
        - Behavior Profile
        - Technology Stack
        - Application Archeology
        - AI - Build your own chatbot & vectorize the data to optimize
    - Note Taking
    - Application Overview & Risk Assessment Exercise
- **Information Gathering (1 hour)**
    - Info Gathering Activities
    - Mapping
        - Generic Web App Mapping
        - Application Flow
            - Rails
            - Node.js
            - Django
            - .Net
            - Java
        - AI - Use Langchain & Huggingface's models to perform this analysis
    - Mapping Exercise
    - Authorization Functions
        - How are users identified?
        - Identify its purpose
        - What could go wrong?
        - AI - Use Langchain & Huggingface's models to perform this analysis
    - Authorization Functions Exercise
- **Authorization (1.5 hours)**
    - Authorization Review
    - Authorization Review Vulnerabilities
        - Broken Access Control
        - Sensitive Data Exposure

        - Mass Assignment
        - Business Logic Flaws
    - AI - Use Langchain & Huggingface's models to perform this analysis
    - Authorization Review Checklist
    - Authorization Exercise
- **Authentication (1.5 hours)**
    - Authentication Review
    - Authentication Review Vulnerabilities
        - Broken Authentication
        - User Enumeration
        - Session Management
        - Authentication Bypass
        - Brute-Force Attacks
    - Authentication Review Checklist
    - Authentication Exercise
    - AI - Use Langchain & Huggingface's models to perform this analysis
- **Auditing (1/2 hour)**
    - Auditing Review
    - Auditing Review Vulnerabilities
        - Sensitive Data Exposure
        - Logging Vulnerabilities
    - Auditing Review Checklist
    - Auditing Review Exercise
- **Injection (1 hour)**
    - Injection Review
    - Injection Review Vulnerabilities
        - SQL Injection
        - Cross-Site Scripting (XSS)
        - XML External Entities (XXE)
        - Server-Side Request Forgery (SSRF)
    - AI - Use Langchain & Huggingface's models to perform this analysis
    - Injection Review Checklist
    - Injection Review Exercise
- **Cryptographic Analysis (1/2 hour)**
    - Cryptographic Analysis Review
    - Cryptographic Analysis Vulnerabilities
        - Encoding vs. Encryption
        - Hashing
        - Stored Secrets
    - AI - Use Langchain & Huggingface's models to perform this analysis
    - Cryptographic Analysis Checklist
    - Cryptographic Analysis Exercise
- **Configuration Review (1/2 hour)**
    - Configuration Review

    - Configuration Review Vulnerabilities
        - Framework gotchas
        - Configuration files
        - Dependency Analysis
    - AI - Use Langchain & Huggingface's models to perform this analysis
    - Configuration Review Checklist

Day 2 – Workshop

- **Technical Hands-On Review (3 hours)**
    - Django Vulnerable Task Manager:
- **Lab Review of Open-Source Applications (4 hours)**
    - Students divide in groups
    - Review an OSS application
- **Presentation of OSS Results (1 hour)**