Mobile Security Testing Guide Hands-On

Training Course

28th & 29th of October

Table of Contents

All training courses are two full days of intensive, hands-on learning and include complimentary morning tea, lunch and afternoon tea. Tickets can be secured buy going to the website https://appsecday.io/.

Course Abstract

Have you ever wanted to know how to bypass SSL Pinning on Android or iOS? Or were just curious if it's possible to do a proper penetration test on a non-jailbroken iOS device?

If so, this training is perfect for you. All of these topics and more will be covered during our hands-on course that is based on the OWASP Mobile Security Testing Guide (MSTG) and is conducted by one of the authors himself. The OWASP MSTG is a comprehensive and open source guide on modern mobile security testing for both iOS and Android. This course will provide a customised mobile testing environment including many hands-on mobile security challenges. Wide ranges of topics will be covered such as Mobile Operating System fundamentals to using Frida (Dynamic Instrumentation Framework) to bypass advanced client-side security controls.

What attendees will learn?

This course is developed for:

- Penetration Testers that want to achieve full coverage when testing a mobile app and know how to work with an accepted industry standard for mobile testing
- Developers that want to understand how attacks against their mobile apps are executed and how they can be improved by implementing security best practices.

The goal of this course is to learn:

- the technical skills to execute a penetration test against iOS and Android mobile applications
- utilise the Mobile Security Testing Guide (MSTG) as a baseline and comprehensive methodology during mobile security assessments.
- How to mitigate vulnerabilities in mobile apps and implement the latest best practices

This training will mainly focus on:

- iOS and Android security fundamentals to understand the security mechanisms that are in place by the OS
- Preparing a penetration testing environment for iOS and Android and clarifying the limitations and benefits of each (real device, emulator, jailbroken, rooted etc.)
- Hands-on exercises that are based on iOS and Android Apps that are build specifically for each test case to gain an understanding of different vulnerabilities
- Demonstrate implementation of the latest security best practices to mitigate vulnerabilities in mobile apps or reduce the attack surface
- Methodology on conducting iOS application testing without a jailbroken device
- Basic introduction into dynamic instrumentation by using Frida and different frameworks on top of Frida (e.g. House, Passionfruit)
- Basic Reverse Engineering of iOS and Android Apps to bypass client-side security controls, such as disabling Root Detection or SSL Pinning

What attendees will be provided?

Attendees will be provided with the following content:

- All slides in PDF format used for Day 1 and Day 2
- Toolkit including all tools and scripts used during the training (Access to private Github repo)
- Several iOS and Android Apps that are used for the exercises (Access to private Github repo)

What attendees should bring?

The following prerequisites need to be fulfilled by the participants in order to be able to execute and follow all exercises:

- Laptop (Macbook or Windows laptop), with at least 8 GB Ram, 40GB of free disk space, working Wi-Fi and administrative access
- VirtualBox installed
- **iOS device (jailbroken or non-jailbroken, both is fine) with at least iOS 11**

The Android training part will be executed in a Genymotion AWS EC2-Instance. One instance will be provided for each participant.

Pre-requisites

The participants should have a basic understanding of mobile apps, interest in security and learning new things and experience with the command line.

Detailed Outline

Day 1 - Android

- Introduction of the Mobile Security Testing Guide (MSTG) and Mobile Application Security Verification Standard (MASVS)
- Key Areas in Mobile Application Security and Differences to Web Application Testing
- Overview of Android Platform and Security Mechanisms
    - Android Security Architecture (Bootloader, Permission model, Sandboxing etc.)
    - App Communication with the Operating System (IPC, Intent etc.)
    - Runtime Environment (Dalvik, ART)
- Creating an Android Testing Environment
    - Android Debug Bridge (ADB)
    - Setting up an EC2 Instance with Genymotion for testing
    - Differences and limitations between testing in an emulator and a real device
- Dynamic Analysis
    - Analysing HTTP traffic through Burp Suite
    - Analysing non-HTTP traffic (e.g. plain TCP)
- Android Application Structure
    - Decompiling APK
    - APK file structure
    - Understanding and Analysing AndroidManifest.xml
- Static Analysis
    - Identifying a deeplink vulnerability
    - Automated Static Analysis with MobSF
- Testing for Sensitive Data in Local Storage (Shared Preferences, SQLite Databases, Internal and External Storage) and secure usage of KeyStore
- Analyzing Memory by creating memory dumps of running Apps
- Dynamic Instrumentation
    - Introduction into Frida
    - Usage of House
    - Identify and hook functions of an Android App
    - Using Frida Server on a rooted device
    - Using Frida Gadget through repackaging of an app with objection
- Bypassing SSL Pinning
    - By using Xposed
    - By using Objection
    - How to bypass SSL Pinning when implemented with Network Security Configuration
- Testing of Data Exposure via Content Providers

Day 2

Android Reverse Engineering:

- Bypass root detection by patching Smali
- Using Dynamic Instrumentation to bypass root detection
- Breaking End-to-End Encryption by using Frida/Brida
- Bypass detection mechanisms for Dynamic Instrumentation (Frida)
- Dissecting Android Malware – Anubis 2.0

iOS:

- Overview of iOS Platform and Security Mechanisms
  - iOS Security Architecture (Hardware Security, Code Signing, Sandbox, Secure Boot, Security Enclave etc.)
  - Explaining IPA Container and Structure on the iOS File System
- Creating an iOS Testing Environment
  - Testing with and without Jailbreak and its limitations
  - Testing in an emulator compared to a real device
  - Hands-on: Setting up a hardware device
- Installation of iOS Apps
- Static Analysis
  - Automated Static Analysis with MobSF
  - App Transport Security (ATS)
- Dynamic Analysis
  - Analysing all HTTP traffic through Burp Suite
  - Analysing all non-HTTP traffic through a remote virtual interface
- Testing with and without Jailbreak:
  - Repackaging an IPA with Frida
  - Testing for Touch/Face ID Bypass
  - Testing for Sensitive Data in Local Storage
  - Testing for SSL Pinning Bypass
  - Using Frida and Passionfruit for iOS Apps
  - Testing for Sensitive Data in Memory
- Testing Stateless Authentication (JWT) in an iOS App
  - Dynamic Testing by using Burp Suite Professional
  - Analyse storage for refresh and access tokens
  - Apply known attacks against JWT
- Reverse Engineering
  - Basic Reverse Engineering of an iOS app
  - Bypassing Client-Side Security
    - Anti-Jailbreaking Mechanisms
    - Debugging detection
    - Tampering detection of Frida

- Runtime instrumentation and patching with Frida

Trainer Biography

Sven Schleier

https://www.linkedin.com/in/sven-schleier-98259194/

Sven began his career as Unix and Linux System administrator, just when the dot com bubble was bursting. After this phase of configuring all kind of BSD variants and Linux distributions he came in touch with Penetration Testing the first time during his Bachelor in Computer Engineering. He made several stops at big consultant companies and small boutique firms in Germany and Singapore and became specialised in Application Security and has supported and guided software development projects for iOS and Android Apps and Web Applications during the whole SDLC. Besides his day job Sven is one of the core project leaders and authors for the OWASP Mobile Security Testing Guide and OWASP Mobile Application Security Verification Standard and has created the OWASP Mobile Hacking Playground. Sven is giving talks and workshops about Mobile and Web Application Security worldwide to different audiences, ranging from developers to students and penetration testers.

Ryan Teoh

https://www.linkedin.com/in/ryan-teoh/

Ryan Teoh (OSCE, OSCP, CRT) is a Security Engineer at Grab with a strong focus on Mobile Security. He spends a considerable amount of time in iOS kernel exploitation, contributing to the iOS security testing chapter and the iOS Crackmes which are part of the OWASP Mobile Security Testing Guide. That aside, he is active on both private and public bug bounty programs and has successfully bagged several critical mobile security bugs. Ryan is a strong believer in knowledge sharing - initiated a security blog on top of facilitating workshops to security engineers, developers and students about mobile security, dynamic instrumentation and reverse engineering of mobile applications.