Mobile Application Security: iOS Edition Bootcamp

Objective

The iOS Edition Bootcamp will teach you all you need to know to execute a successful penetration test against an iOS App. We will use an end-to-end approach to cover all the different phases of a penetration test, including static analysis, dynamic testing and reverse engineering. The instructor will share his experience and tips and tricks to attack iOS apps.

Labs will be shared as assignments during the course so you can practice the testing methodology and the latest testing tools used to analyse iOS Apps.

After successful completion of this course, you will have a complete understanding of how to test for vulnerabilities in mobile apps, assess risks, propose fixes to developers to mitigate vulnerabilities and learn how to execute tests consistently.

Prerequisites

- A macOS device that is able to run the latest macOS version (macOS Catalina 10.15)
- A jailbroken iOS device (>= iOS 11)
- The latest Xcode installed

Course Content

Topics include:

Module I: Introduction to iOS, setting up the test environment and static analysis of iOS apps

- Introduction to the course and setup of the course environment. Explanation of how testing in a simulator compares to a real iOS device.
- Overview of iOS and its security mechanisms, including hardware security, code signing, sandbox, secure boot and the security enclave.

- Overview of different mobile app types like native apps, web apps, hybrid apps and progressive web apps(PWA).
- Jailbreaking of an iOS device:
    - Overview of current available jailbreaks and how to jailbreak an iOS device
    - Differences of testing with and without jailbreak
- Distribution of apps from the developer to the penetration tester and various ways of installing (side-loading) iOS apps
- Tools needed for macOS and iOS device for a successful penetration test
- Static analysis of an iOS app binary:
    - Decrypting an app with Fairplay Encryption
    - Understanding the structure of an IPA Container
    - Using MobSF for a static scan
- Automated analysis of an iOS app with MobSF implemented in Objective-C and Swift. Interpretation of the output and identification of the attack surface.
- Analysis of the Info.plist file for App Transport Security (ATS) settings
- Software Composition Analysis (SCA) with OWASP Dependency Checker to analyse 3rd party libraries used in an iOS app for vulnerabilities.

Module 2: Kick-starting with Frida and Biometric Authentication in iOS Apps

- Differences between testing web apps and testing mobile apps, due to different attack surfaces such as biometric authentication, reverse engineering attacks and extensive usage of local data storage.
- Dynamic instrumentation
    - Frida Basics (Server, Architecture) and Frida installation
    - Frida commands (frida-ps, frida, frida-trace etc.)
    - Classes and methods with Frida
    - Frida scripts
- Static analysis of an IPA during runtime with Frida
- Introduction to tools based on Frida
    - Installing and using Passionfruit
    - Installing and using Objection
- Biometric authentication
    - Touch ID/Face ID in iOS Apps
    - Bypassing biometric authentication and best practices for secure implementation

Module 3: Network Analysis and JSON Web Token in mobile apps

- Man-in-the-Middle (MITM) attacks and how to be the MITM while testing a mobile app

- Using Burp with an iOS device, installing the CA certificate and analysing HTTP traffic using Burp Suite
- SSL Pinning
    - TLS basics and additional protection through SSL Pinning
    - Different implementations of SSL Pinning techniques
    - Different ways of bypassing SSL Pinning, by using SSL Kill Switch, Frida and Objection
- Intercepting non-HTTP traffic
- Common mobile app authentication methods - JSON Web Tokens (JWT)
    - JWT and bad/best practices in the wild
    - Different Burp modules and ways to test JWT
    - Attacks against JWT and their applications

Module 4: Assessing iOS Apps for storage of sensitive data and Reverse Engineering of Security Controls

- Xcode and its relevance for penetration testing, including log analysis with Xcode
- Testing for sensitive data in local storage
    - Different ways to store data on iOS (Core Data, SQLite, Plist and NSUserDefaults)
    - Structure of the iOS file system, and bundle and data directory of an iOS app
    - Analysis of local storage using Objection
    - Analysis of local storage using Xcode and iOS Simulator
- Storing data securely using the Keychain
- Security controls in mobile apps and how to defeat them (e.g. by bypassing jailbreak detection)
    - Common Security Controls in mobile apps (e.g. jailbreak detection, anti-debugging, etc)
    - Jailbreak detection implementation and what it checks
    - Identifying usage of jailbreak detection in an iOS app and different ways of bypassing jailbreak detection via Frida and Objection