Mobile Application Hacking & Security

5 days Hands On Course

AppSec Labs Ltd.
info@appsec-labs.com | https://appsec-labs.com | 23 HaTa'as St., Kfar Saba 44641 Israel | T: +972-9-7485005 | F: +972-9-7730595

Mobile Application Hacking and Security
5-day hands on course

Course description

This course will focus on the techniques and tools for testing the security of Android and iOS mobile applications. During this course, the students will learn about important topics such as the Android and iOS Security model, the emulator, how to perform static analysis, traffic manipulation, and dynamic analysis.

By taking this course, you will be able to perform penetration testing on Android and iOS mobile applications and expose potential vulnerabilities in the tested application.

The objectives of the course are:

- Understand the Android and iOS application threat landscape
- Perform penetration testing on android and iOS mobile apps
- Identify vulnerabilities and exploit them
- Operate AppSec Labs' unique ***AppUse*** customized VM for android pen-testing (https://appsec-labs.com/appuse)
- Operate AppSec Labs' unique ***iNalyzer*** app for iOS pen-testing (https://appsec-labs.com/inalyzer/)

Target audience

Members of the security / software development team:

- Security penetration testers
- Mobile developers
- Advanced QA teams

Prerequisites

Before attending this course, students should be familiar with:

- Common security concepts
- Basic knowledge of the Linux OS
- Java background and basic knowledge of the Android development platform
- Advantage: C / C++ background and basic knowledge of the iOS development platform

Hardware/Software requirements

Please make sure that each machine has:

- At least 4GB of RAM (6GB is highly recommended)
- 30GB of free HD space
- VMware player (free) or VMware workstation (commercial)
- Wireless connectivity in the class – a dedicated router accessible from the class' network
- Must: Participants must bring iPhone/IPad devices with iOS 8.x or iOS 9.x to the course. The devices have to be jailbroken.

Course topics

Day 1

Android: Introduction to Android security

- Mobile application threat model
- The Android linux OS security
- ART (Android runtime) technology
- The Android security mechanisms
- Application file system isolation & insecure file access
- The permission model
- Database isolation
- The Android emulator VS. physical device
- The debug bridge
- Rooting
- AppUse VM
- **Lab - Android Emulator, ADB and Database Isolation**
- **Lab - Build your own malware app and steal other app files**

Android: Static analysis - Reverse engineering & patching

- The APK file package
- APK extraction - Investigating layout, manifest, permissions and binaries
- Extracting the content of the classes.dex file
- Using smali/baksmali Dalvik assembler/disassembler
- Decompiling the application
- Using dex2jar
- Reverse engineer the app and change its behavior
- Decompile / disassmble the dex classes using smali/baksmali
- Code patching - Modifying the code
- Recompile the patched application
- Resign the APK
- **Lab - Recovering protected secrets**
- **Lab - Application patching**

- **Homework**

Day 2

Android: Component & IPC security

- Major component types – Activity, Service, Content provider, Broadcast receiver
- The intent structure
- The intent filter
- Component permissions and visibility
- Authenticating Callers of Components
- Binder interface
- Pending intents
- Direct component invocation by unauthorized apps
- Unprotected content providers
- Securely activating components
- Avoiding access to restricted screens
- Common IPC vulnerabilities
- **Lab - Invoking Internal Activities Using Malicious Intents**
- **Lab - Attacking broadcast receivers**

Android: Traffic analysis and manipulation

- Common vulnerabilities related to traffic
- Proxies and sniffers
- Sensitive information transmission
- Importing SSL certificates & trusted CA's
- Bypassing server certificate validations
- Exposing insecure traffic
- Validating server certificates and avoiding man-in-the-middle
- SSL Pinning
- WebView vulnerabilities
- **Lab - Parameter Manipulation Using a Proxy**

- **Homework**

Day 3

Android: Application dynamic runtime analysis

- Monitoring process activity
- Observing file access
- Monitoring network connectivity
- Analyzing logs using logcat
- Memory dumps and analysis
- Smali Debugging
- Setting breakpoints
- Native debugging with GDB
- Runtime instrumentation and manipulation using ReFrameworker
- Runtime hooks using Xposed and Frida
- **Lab - Memory dumps and objects analysis**
- **Lab - Smali debugging**

iOS: Introduction to iOS Security

- Welcome to iOS
- What makes mobile security so different?
- OWASP Top 10 Mobile
- What is iOS?
- iOS Device Architecture
- iOS Security Model
- iOS File System isolation
- Application Sandbox
- The iOS Pen-Testing Environment
- Lab Setup overview
- Device Setup
- Jailbreaking iOS
- Cydia Installations
- Laptop Installation
- AppSec-Labs iNalyzer
- **Lab - Setting-up and Exploring the iOS environment**

- **Homework**

Day 4

iOS: Application Static Analysis

- Static Analysis
- The IPA file package
- IPA file deployment and manual installation
- The CodeResources file
- Anti-tampering configuration
- Tampering with IPA Content
- Investigating the Application contents – View Controllers
- Investigating Info.plist file
- Listing all CFUR types on a device
- Investigating Binaries
- iOS Binary Application Structure Encryption
- Decrypting Binary - concept
- Static analysis of a decrypted Binary
- Investigating binary content
- Reversing Interfaces
- Using iNalyzer for static analysis
- **Lab - Binary Static Analysis manual and automated**

iOS: Application Storage Analysis

- Application Storage Analysis
- File System access security
- File System Data Protection Class
- Application storages
- Property list files (.plist)
- Tampering with Property list files (.plist)
- Investigating Plist files – plutil.
- Database files (.db/.sqlite)
- Snapshots Storage
- Persistent Cookies
- Investigating Logs
- Keyboard Cache
- Cryptographic failures
- Keychain access
- iNalyzer Storage Snapshot

iOS: Traffic Manipulation

- Traffic Analysis and Manipulation
- Common architecture
- Bad Session Management
- Phone identifiers used in authentication
- Credentials leakage
- Client information sent to advertisement/analytics server
- Server side vulnerabilities
- Sniffing and intercepting traffic
- SSL obstacles
- Importing SSL certificates & trusted CA's
- Bypassing server certificate validations
- **LAB: Catch and manipulate application's traffic**

Day 5

iOS: Temporary runtime manipulation

- Temporary Runtime Manipulation
- Why do we need temporary runtime manipulation?
- Temporary runtime manipulation tools
- Objective-C class interposing
- Runtime manipulation with Cycript
- Runtime manipulation with iNalyzer Dashboard
- **Lab - Objective C and runtime manipulation using iNalyzer and cycript**

iOS: Persistent runtime manipulation

- Persistent Runtime Manipulation
- Means of persistent manipulation
- Persistent runtime manipulation technique
- Persistent runtime manipulation - backstage
- Persistent runtime manipulation frameworks
- Theos Injection Framework
- iNalyzer header dump
- iNalyzer class dump reference
- Reversing iOS Binary
- Remote debugging with GDB
- **Lab - Persistent runtime manipulation using Theos**