Android Mobile Application Hacking

AppSec Application security Labs

3-Day Hands-On Course
Course Syllabus

AppSec Labs Ltd.
info@appsec-labs.com | https://appsec-labs.com | 10 HaTa'as St., Kfar Saba 44641 Israel | T: +972-9-7485005 | F: +972-9-7730595

Android mobile application hacking

3-day hands on course

Course description

This course will focus on the techniques and tools for testing the security of Android mobile applications. During this course the students will learn about important topics such as the Android Security model, the emulator, how to perform static analysis, traffic manipulation, and dynamic analysis. By taking this course you will be able to perform security penetration testing on Android mobile applications and expose potential vulnerabilities in the tested application.

The objectives of the course are:

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

Target audience

Members of the security / software development team:

- Security penetration testers
- Android developers

Prerequisites

Before attending this course, students should be familiar with:

- Common security concepts
- Java background and basic knowledge of the Android development platform
- Basic knowledge of the Linux OS

Course topics

Day 1:

Introduction to Android security

- Mobile application threat model - What makes mobile application security so different?
- The Android linux OS security
- The Dalvik VM
- 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**
- **Homework**

Static analysis - Reverse engineering & patching the application binaries

- 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
- Decompilation
- Using dex2jar
- Reverse engineer the app and change its behavior
- Decompile / disassmble the dex classes using smali/baksmali
- Code patching - Modifying the code
- Recompile
- Resign the APK
- **Lab - Recovering protected secrets**
- **Lab - Application patching**
- **Homework**

Day 2:

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 IDA (building signatures, types etc.)
- Runtime instrumentation and manipulation using ReFrameworker
- **Lab - Memory dumps and objects analysis**
- **Lab - Bypass Application Restrictions without Modifying Any Code**
- **Homework**

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
- Using the HostnameVerifier class
- Using SSL with the HttpsURLConnection class
- Client side certificate authentication
- **Lab - Parameter Manipulation Using a Proxy**
- **Lab - Bypassing SSL Pinning**
- **Homework**

Day 3:

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
- Sticky broadcasts
- Securely activating components
- Avoiding access to restricted screens
- **Lab – Invoking Internal Activities Using Malicious Intents**
- **Lab – attacking broadcast receivers**
- **Homework**

Identifying code level vulnerabilities

- Verifying caller identity
- Whitebox approach using a code review
- Locating interesting code
- How to perform
- Detecting common code level vulnerabilities
- Using Lint
- **Lab – security code review**
- **Homework**