Android Application Hacking and Malware Analysis

5-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 Application Hacking and Malware Analysis – 5 day hands on course

Course description

This course will focus on the techniques and tools for testing the security and performing analysis 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 penetration testing on Android mobile applications and expose potential vulnerabilities in the tested application, and to analyze a given app in order to identify any malware behavior

The objectives of the course are:

- Understand the Android application threat landscape
- Perform penetration testing on android apps, Identify vulnerabilities and exploit them
- Perform malware analysis
- 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
- Malware analysis
- 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

Hardware/Software requirements

Please make sure that each machine has:

- At least 2GB of RAM (4GB is highly recommended), 15GB of free HD space
- VMware player (free) or VMware workstation (commercial)
- Wireless connectivity in the class – a dedicated router accessible from the class' network

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
- Setup for android app research and malware analysis
- **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:

Introduction to Android Malware

- Introduction to Android malware
- Why is it so common on Android?
- Types of malware
- Common attack vectors
- Rogue Android markets
- Repackaging
- Malicious updates
- Command & Control (C&C) servers
- Drive by attacks
- Social engineering
- Physical attacks
- Malicious payloads
- Financial charges
- Information collection
- Privilege escalation
- Case studies

Detecting malware behavior

- Permissions
- Detecting malware signatures
- Static analysis
- Building hooks using ReFrameworker for malware behavior detection
- Important strings
- Dynamic analysis
- Dynamic code Loading
- Class loaders, package context, native code, Runtime.exec, package manager
- Code loading as evasion technique
- Improper use of package names
- Unprotected storage
- Insecure downloads

- Random behavior & Sandboxing
- **Lab – malware analysis**
- **Homework**

Android anti reverse engineering

- Packers & Unpackers
- Case study - HoseDex2Jar, ApkProtect, Bangcle, Ijiami, LIAPP, PangXie
- Obfuscators & De-obfuscators
- Case study – ProGuard, DexGuard, ApkProtect, DexProtector, Allatori
- Anti-Reversing Techniques
- Custom obfuscation
- Decompiler confusion
- Anti-emulation
- Anti-rooting
- **Lab – overcoming anti reversing techniques**
- **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**

Content provider security

- Introduction to content providers
- Content URIs
- Exported providers
- Provider permissions
- Using signature protection level
- Temporary permissions
- The SQLite DB
- Local SQL injections
- Parameterized queries
- Unprotected content providers
- Verifying caller identity
- **LAB**

Android app permissions

- Application permission isolation
- The permission model
- Permission types & app restrictions
- Application signing
- Permission categories
- Creating custom permissions
- Verifying process permissions during runtime
- **LAB**

Day 4:

Cryptography Pitfalls

- Symmetric cryptography
- Asymmetric cryptography
- Hashing
- Digital signing
- PKI / certificate
- SSL protocol
- SSL cipher suite
- Insufficient transport layer protection
- LAB – Cryptography lab

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**

Device administration API

- Secure device management
- Application installation
- Using policies
- Locking devices
- Wiping sensitive data
- Creating encrypted storage
- Disabling sensitive hardware
- **LAB**

Day 5:

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**

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**