PRACTICAL MOBILE APPLICATION EXPLOITATION

Expert-Led Cybersecurity Training · Beginner to Advanced

COURSE OVERVIEW

This comprehensive course equips you with the knowledge and tools to navigate the complex world of mobile application security focusing on iOS and Android platforms. Whether you're a beginner or a seasoned security enthusiast, our fast-paced curriculum, featuring intensive hands-on labs, will empower you to effectively exploit and secure mobile apps.

We are bringing an updated version of the course with the latest tools & techniques. The training is based on exploiting vulnerable apps written by the authors, as well as exploiting a wide range of real-world application vulnerabilities. The students will get an in-depth knowledge about the different kinds of vulnerabilities in Mobile applications. The students will also learn how to reverse engineer iOS and Android Apps and system binaries. After the workshop, the students will be able to successfully pentest and secure applications running on iOS and Android platforms.

This course prepares you for the **Certificate Mobile Security Engineer (CMSE)** certification exam, a hands-on assessment specifically designed to test your ability to exploit real-world vulnerabilities commonly found in mobile applications.

8KSEC

KEY LEARNING OBJECTIVES

- Learn how to set up your own Lab environment for testing
- Learn how to Reverse engineer iOS and Android binaries (Apps and system binaries)
- Get an understanding of the ARM64 Instruction Set
- Learn Device Fingerprinting and Anti-Fraud techniques
- Get an intro to common bug categories like UaF, Heap overflow, etc
- Get PoC applications to perform 1 click exploits on Mobile apps
- Learn how to debug iOS and Android apps
- Get an intro to common bug various bug categories on Android and iOS systems
- Learn to audit iOS and Android apps for security vulnerabilities
- Understand and bypass anti-debugging and obfuscation techniques
- Learn manual and automated ways of bypassing exploit mitigations
- Learn to identify vulnerabilities in native as well as Cross-platform apps
- Learn to exploit different iPC mechanisms in iOS and Android applications
- Get a detailed walkthrough on using IDA Pro, Hopper, Ghidra, etc
- Secure Mobile apps by implementing custom solutions
- Become a Certified Mobile Security Engineer (CMSE)

WHY SHOULD YOU TAKE THIS COURSE?

This is a completely hands-on course designed for beginners and intermediate students. Instead of just slides, attendees will get a chance to exploit all of the vulnerabilities taught by the instructors. For the Onsite and Virtual sessions, the attendees will be provided with Cloud-based Corellium labs for performing the hands-on iOS and Android exercises without the need to carry physical phones. Slack channel is created before the course for the students so that they can be adequately prepared in terms of hardware and software before the class.

WHAT WILL THE STUDENTS GET?

- An attempt to CMSE (Certificate Mobile Security Engineer) certification exam
- Certificate of completion for the Training program
- Source code for vulnerable applications
- Source code for Exploit PoCs' that can be used for Bug Bounties
- All Frida Scripts used during the course
- Students will be provided with access to Corellium for iOS and Android hands-on for the duration of the course
- Students will be provided access to cloud instances for the duration of the course
- Slack access for the class and after for regular mobile security discussions

HARDWARE/SOFTWARE REQUIREMENT

- Laptop with 8+ GB RAM and 40 GB hard disk space
- Students will be provided with access to Linux cloud instances
- Students will be provided with access to Corellium for iOS hands-on and as such do not need to carry iOS devices
- Administrative access on the system

Detailed Course Setup instructions and Slack access will be sent a few weeks prior to the class

WHO SHOULD ATTEND?

This course is for penetration testers, mobile developers or anyone keen to learn mobile application security.

PREREQUISITE KNOWLEDGE

- Working knowledge of cybersecurity and pentesting fundamentals
- Basic working knowledge of iOS and Android platforms
- Basic Linux skills and command-line proficiency
- Understanding of fundamental programming concepts and looping structures in at least one higher-level language (Java, Kotlin, Objective-C, Swift, C, C++, or similar)
- Basic ARM/AARCH64 binary assembly knowledge is recommended, but not required

COURSE SYLLABUS

PART 1 - IOS EXPLOITATION

Module 1: Introduction to Reverse Engineering in iOS

- Key Concepts and Terminologies
- Introduction to Hopper/Ghidra
- Introduction to the ARM64 instruction set
- Introduction to Objective-C and Swift
- Reversing Objective-C and Swift Binaries
- Disassembling methods
- Modifying assembly instructions
- Deciphering Mangled Swift Symbols
- Identifying Native Code
- Understanding the Program flow
- Identifying Cross-Platform mobile frameworks

Module 2: Getting Started with iOS Security

- iOS security model
- App Signing, Sandboxing, and Provisioning
- iOS App Groups
- Primer to iOS 17-18 security
- Xcode Primer
- Address Sanitizer
- Exploring the iOS filesystem
- What's in a Code Signature?
- Entitlements explained
- How Sandboxing works on iOS
- Setting up lldb for Debugging
- lldb basic and advanced usage
- Setting up the testing environment
- Jailbreaking your device
- What's in a Rootless Jailbreak?
- Jailbreak Bootstraps
- Sideloading apps
- Binary protection measures
- Decrypting IPA files
- Self-signing iOS binaries

Module 3: Static and Dynamic Analysis of iOS Apps

- Bundle vs Data Container
- Finding Secrets in Code
- Dumping class information
- Insecure local data storage
- Inspecting Keychain items
- Identifying URL schemes and Universal Links
- Dynamic Analysis of iOS applications
- Method Swizzling
- Debugging apps using lldb
- Modifying ARM registers
- Basic App Instrumentation techniques using Frida
- Advance App Instrumentation techniques using Frida
- Frida on Non-Jailbroken devices
- Frida on Swift and native code
- Reversing Third Party frameworks
- Testing React Native and Flutter Apps
- Automating App Inspection

Module 4: iOS application vulnerabilities

- Tracing Crypto operations
- Side channel data leakage
- Sensitive information disclosure
- Bypassing Jailbreak Detection
- Bypassing SSL Pinning
- Bypassing Certificate transparency checks
- Exploiting iOS WebViews
- Exploiting URL schemes and Universal LInks
- Client-side injection
- Bypassing jailbreak, piracy checks
- Inspecting Network traffic
- Traffic interception over HTTP, HTTPs
- Manipulating network traffic
- Identifying iOS malware

8KSEC

Module 5: Securing iOS Applications

- AppAttest and Device Check frameworks
- Device Fingerprinting
- Detecting GPS Spoofing
- Implementing Secure Webviews
- Code Obfuscation techniques
- Protecting the Transport Layer
- Detecting Malicious Libraries
- Implementing Anti-Debug Checks
- Detecting Suspicious Device Reset
- Detecting Patched Applications
- Detecting Proxied Applications
- Jailbreak Detection Techniques

PART 2 – ANDROID EXPLOITATION

Module 6: Intro to Android Security

- Why Android
- Android Security Architecture
- Extracting APK filesfrom Google Play
- Understanding Android application structure
- Signing Android applications
- ADB – Non-Root
- Rooting Android devices
- ADB – Rooted
- Understanding the Androidfile system
- Permission Model Flaws
- Attack Surfaces for Android applications

Module 7: Android Components

- Understanding Android Components
- Introducing Android Emulator
- Introducing Android AVD
- Setting up Android Pentest Environment

Module 8: Reversing Android apps

- Process of Android Apps Engineering
- Reverse Engineering for Android Apps
- Smali Learning Labs
- Examining Smali files
- Smali vs Java
- Dex Analysis and Obfuscation
- Reversing Obfuscated Android Applications
- Exploiting Android Accessibility Permissions
- Patching Android Applications
- Reverse Engineering known complex Malware in the Wild and anti evasion techniques

Module 9: Static and Dynamic analysis

- Proxying Android Traffic
- Introduction to Certificate Transparency
- Exploiting Local Storage
- Exploiting Weak Cryptography
- Exploiting Side Channel Data Leakage
- Multiple Manual and Automated Root Detection and Bypass Techniques
- Exploiting Weak Authorization mechanism
- Identifying and Exploiting Android Components
- Analyzing Proguard, DexGuard, and other Obfuscation Techniques
- Exploiting Android NDK
- Exploiting Android WebViews
- Exploiting DeepLinks in Android
- Multiple Manual and Automated SSL Pinning Bypass techniques
- Exploiting Flutter Applications

(Continued on the next page)

Module 10: Frida and Automated Exploitation

- Exploiting Crypto using Frida
- Basic App Exploitation techniques using Frida
- Dumping Class Information using Frida
- Dumping Method Information using Frida
- Viewing and Changing Information using Frida
- Calling Arbitrary functions using Frida
- Exploiting native libraries using Frida
- Tracing using Frida
- Advance App Exploitation techniques using Frida
- Frida on non-rooted Android

Module 11: Securing Android Apps

- Detecting Patched Android Applications
- App Integrity Protection
- Detecting Malicious Libraries
- Detecting Emulator/Rooted Devices
- Secure Implementation of WebViews
- Implementing Anti-Debug Checks
- Detecting Suspicious Device Reset
- Detecting Proxied Applications

About the company

8kSec is a foremost cyber security research company offering exceptional training and consulting services to aid clients in enhancing their security stance. Our experts possess extensive experience in delivering specialized cybersecurity training and consulting to several commercial and defense organizations across the United States, Europe, and the Middle East and North Africa region.

Get in touch

8kSec.io
info@8ksec.io