



OFFENSIVE IOS INTERNALS

Expert-Led Cybersecurity Training · Beginner to Advanced

COURSE OVERVIEW

This course is designed to provide a comprehensive understanding of the internals of the iOS operating system and its security features. The course will cover topics such as the iOS operating system architecture, memory management, application sandboxing, code signing etc. Students will learn the fundamental concepts and tools used in reverse engineering, and get a thorough introduction to the ARM64 architecture, including static and dynamic analysis techniques, as well as various debugging and disassembly tools. Exploit mitigations such as SPTM, TXM, PAC, PAN, PPL etc will also be discussed. Additionally, the course covers iOS application security, including topics such as encryption, and secure communication. Students will learn how to use Frida, a dynamic instrumentation framework, for reverse engineering and dynamic analysis of mobile applications. We will also discuss advanced topics such as hooking, memory manipulation, and instrumenting network communication. This course will also discuss the tools and techniques used for analyzing iOS malware. The course will also cover the different stages of iOS malware analysis, including static, dynamic, and behavioural analysis. Additionally, the course will walk the attendees through different methods of mitigating and preventing iOS malware.

This course will be a mix of lectures, practical labs, and projects designed to give students hands-on experience with iOS internals and application security. Students will gain the skills needed to reverse engineer, design, develop, and secure iOS applications.

This course prepares you for the **Certified iOS Security Engineer (CISE)** certification exam, a hands-on assessment specifically designed to test your grasp of advanced iOS security domains including userland and kernel components.

KEY LEARNING OBJECTIVES

- Introduction to ARM64 architecture
- Understand iOS app lifecycle
- Overview of the iOS Kernel and its Security Mitigations
- Reverse engineering iOS binaries (Apps and system binaries)
- Get an intro to common bug various bug categories on iOS
- Learn to audit iOS apps for security vulnerabilities
- Understand Memory allocation in Userland and Kernel
- Understand and bypass anti-debugging and obfuscation techniques
- Learn manual and automated ways of bypassing security mitigations
- Learn Device Fingerprinting and Anti-Fraud techniques
- Get a detailed walkthrough on using Ghidra, Hopper etc
- Advanced Dynamic Instrumentation using Frida
- Understanding how Rooting and Jailbreaks work
- Case Study of some known vulnerabilities
- Learn to identify vulnerabilities in native as well as Cross-platform apps
- Learn to exploit different iPC mechanisms (mach_msg, XPC etc)
- mach_msg2 , SAD_FENG_SHUI, PGZ
- Get a detailed walkthrough on using IDA Pro, Hopper, Ghidra, etc
- Secure Mobile apps by implementing custom solutions
- Become a Certified iOS Security Engineer (CISE)

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 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 CISE (Certificate iOS 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 for the duration of the course (Live On-site & Virtual Training only)
- Students will be provided access to cloud instances for the duration of the course (Live On-site & Virtual Training only)
- Slack access for the class and after for regular mobile security discussions (Live On-site & Virtual Training only)

WHO SHOULD ATTEND?

This course is specifically designed with the needs of modern iOS developers. This course will also be applicable for vulnerability researchers, penetration testers, mobile developers, or anyone keen to learn more about the iOS application security ecosystem.

PREREQUISITE KNOWLEDGE

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

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 (Live On-site & Virtual Training only)

COURSE SYLLABUS

Module 1: iOS Operating System Architecture

- Overview of iOS architecture
- iOS system libraries and frameworks
- Setting up a testing environment for iOS research
- Overview of the Mach-O Binary Format
- iOS virtual memory management
- Overview of application sandboxing and code signing in iOS

Module 2: Introduction to Reverse Engineering in iOS

- Key Concepts and Terminologies
- Introduction to Hopper/Ghidra
- Introduction to the ARM 64 instruction set
- ARM64 security mitigations
- ARM64 calling convention
- 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 3: 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
- Sideloaded apps
- Binary protection measures
- Decrypting IPA files
- Self-signing iOS binaries
- Analyzing Proprietary security Mitigations
- Overview of Past Vulnerabilities
- Intro to dyld_shared_cache

Module 4: iOS Kernel internals

- Intro to XNU kernel
- The Mach and BSD Layer
- Overview of IOKit
- Extracting the Kernelcache and Kexts
- Analyzing specific kexts AMFI, CoreTrust, Sandbox
- Sandbox Profiles
- Symbolicating a Kernelcache
- Overview of mach_msg2, SAD_FENG_SHUI, PGX
- Entitlement validation in the Kernel
- Analyzing Kernel Panic files
- Walkthrough of PAC, SPTM, PAN, GXL, PPL etc
- Patching Diffing XNU kernel

Module 5: Frida in-depth

- Overview of Frida and its capabilities
- Setting up the Frida environment
- Frida usage and commands
- Frida-trace and handlers
- Frida hooking techniques
- Frida on Swift applications
- Frida on native code
- Frida memory manipulation techniques
- Analyzing messaging apps using Frida
- Invoking custom functions with Frida

Module 6: 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

Module 7: iOS vulnerabilities

- Case Study of Sandbox Escapes
- Incorrect validation of Entitlements
- XPC Related vulnerabilities
- Case Study of a Kernel Vulnerability
- Case Study of a PAC Bypass

Module 8: iOS Malware Reversing

- Understanding different stages of a Malware
- Device Acquisition techniques
- Using Custom IOCs
- Case Study of some Public Malware

Module 9: Securing iOS Ecosystem

- 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
- Pasteboard Security Measures
- Understanding the Lockdown Mode
- Understanding Code Signature Checks



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 multiple commercial and defense organizations across the United States, Europe, and the Middle East and North Africa region.

Get in touch

[8kSec.io](https://8ksec.io)

info@8ksec.io

