

OFFENSIVE ANDROID INTERNALS

Expert-Led Cybersecurity Training · Beginner to Advanced

COURSE OVERVIEW

This in-depth and immersive program offers participants an opportunity to enhance their understanding of Android Internals, Reverse Engineering as well as Android Application Exploitation. It provides a broad understanding of Android system architecture, covering topics such as Android Drivers, Modules, Linux Kernel, and the Android Binder. Participants will gain hands-on experience in reverse engineering, exploit development basics for the ARM platform, and deep dive into memory management and related vulnerabilities.

The course also covers Android's boot, recovery, rooting processes, and permissions, along with security features like DAC, CAP, SECCOMP, and SELinux. For a practical learning experience, the course covers how to extract and decrypt boot images for Android devices. The course includes hands-on exercises for symbolicating the Android kernel and porting exploits to other Android devices.

Advanced Frida techniques such as custom tracing, profiling, and memory inspection are explored with real-world applications. Case studies on prominent malware and custom malware samples designed for the course shed light on reverse engineering and advanced forensics techniques. Application Security related vulnerabilities occurring due to Android

8KSEC

components are also covered as a part of the course. The training also includes hands-on learning using vulnerable applications created for the course, and a wide range of real-world application vulnerabilities in order to give an in-depth knowledge about the different kinds of vulnerabilities in Mobile applications.

This course is designed for vulnerability researchers, penetration testers, mobile developers, and anyone eager to understand the inner workings of the Android platform and applications. This course prepares you for the **Certified Android Security Researcher (CASR)** certification exam, a hands-on assessment specifically designed to test your grasp of advanced Android security domains including userland and kernel components.

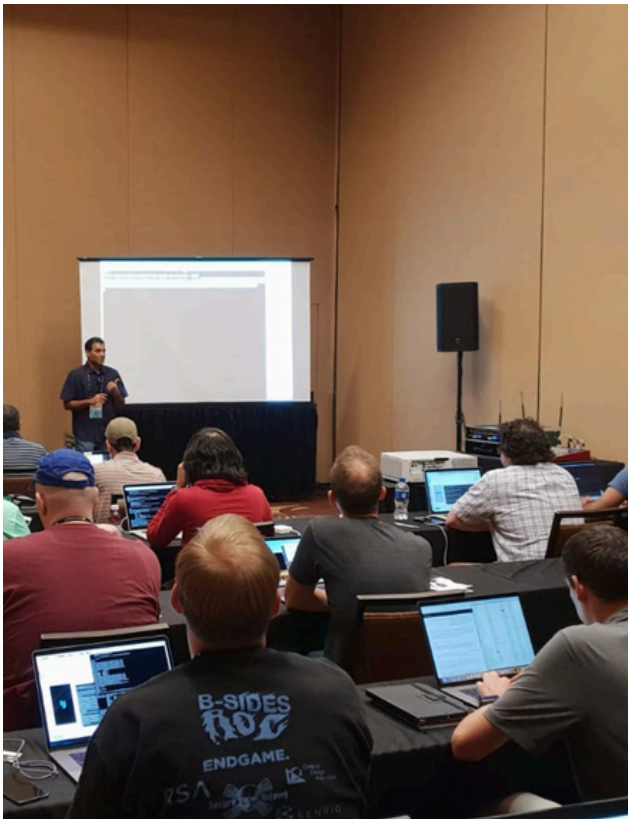
KEY LEARNING OBJECTIVES

- Understand the Android System Architecture and AOSP source code
- Learn about Android Tracing
- Grasp Android Boot, Recovery, and Rooting processes
- Get an understanding of latest ARM64 instruction set, dynamic memory management and related vulnerabilities on the ARM platform
- Acquire skills in ARM Reverse Engineering and exploit development
- Learn how to customize and build Android Kernel for Vulnerability Research
- Gain knowledge about Android Platform Permission, DAC, CAP, SECCOMP, and SELinux
- Develop practical skills in fuzzing applications and processes on Android devices
- Overview of Kernel protections and bypasses
- Reverse engineering Android binaries (Apps and system binaries)
- Get PoC applications to perform one-click exploits on Mobile apps
- Get an intro to common bug categories on Android systems
- Learn to audit Android apps for security vulnerabilities
- Understand and bypass anti-debugging and obfuscation techniques
- Get a detailed walkthrough on using IDA Pro, Hopper, Frida and other tools
- Learn how accessibility malwares work, and how to reverse engineer well-known crypto wallet stealers
- Learn how to symbolicate the Android kernel
- Learn how to extract and decrypt boot images for Android devices
- Reverse engineer and trace Android JNI bindings, including RegisterNatives
- Use tools like JNINinja, Frida, and Medusa to hook and monitor JNI methods at runtime
- Build a functional JVM environment and AFL++ Frida-mode to fuzz JNI bindings and validate crashes
- Become a Certified Android Security Researcher (CASR)

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 On-site and Virtual sessions, the attendees will be provided with Cloud based Corellium labs for performing the hands-on Android exercises without the need to carry physical phones. A 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.

8KSEC



WHO SHOULD ATTEND?

This course is designed for vulnerability researchers, malware analysts, penetration testers, mobile developers, and anyone eager to learn more about the workings of Android devices and applications.

PREREQUISITE KNOWLEDGE

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

WHAT WILL THE STUDENTS GET?

- An attempt to Certified Android Security Researcher (CASR) 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
- 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)

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)

8KSEC

COURSE SYLLABUS

Module 1: Introduction to Reverse Engineering in Android

- Key Concepts and Terminologies
- Introduction to Hopper/Ghidra
- Introduction to the ARM64 instruction set
- ARM64 security mitigations
- ARM64 calling convention
- Introduction to Java and Kotlin
- Disassembling methods
- Modifying assembly instructions
- Identifying Native Code
- Understanding the Program flow
- Identifying Cross-Platform mobile frameworks
- Reversing ARM binaries
- Exploiting a simple Heap Overflow
- Building a simple ROP chain
- Breaking ASLR with Info leaks/Brute force
- Exploit mitigations (ASLR, Heap Poisoning, PAN and more)

Module 2: Intro to Android Security

- Android Security Architecture
- Extracting APK files from Google Play
- Understanding Android application structure
- Signing Android applications
- Understanding Android ADB
- Understanding the Android file system
- Permission Model Flaws
- Attack Surfaces for Android applications

Module 3: Components

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

Module 4: Reversing Android apps

- Process of Android Apps Engineering
- Reverse Engineering for Android Apps
- Smali Learning Labs
- Examining Smali files
- Dex Analysis and Obfuscation
- Reversing Obfuscated Android Applications
- Exploiting Android Accessibility Permissions
- Reverse Engineering known complex Malwares in the Wild
- Patching Android Applications
- Android App Hooking

Module 5: Static and Dynamic analysis

- Proxying Android Traffic
- Exploiting Local Storage
- Exploiting Weak Cryptography
- Exploiting Side Channel Data Leakage
- Exploiting Content Provider Path Traversal & Info Leakage
- Multiple Manual and Automated Root Detection and Bypass Techniques
- Exploiting Weak Authorization mechanism
- Identifying and Exploiting Android Components
- Exploiting Android NDK
- Android Game Hacking

8KSEC

- Multiple Manual and Automated SSL Pinning Bypass techniques
- Exploiting Android Google Play Billing
- Firebase Exploitation
- Exploiting Biometric Authentication
- In-memory tampering
- Exploit Zip Path Traversal/ZipperDown
- Exploiting Flutter Applications
- Exploiting AWS Cognito Misconfiguration
- Exploiting Android Deep Links and WebViews

Module 6: 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
- Tracing using Frida
- Advanced App Exploitation techniques using Frida
- Frida on non-rooted Android
- Overview of Frida 17.x (Latest): agent-based architecture changes, and updates to tools like Objection and r2frida

Module 7: Securing Android Apps

- Detecting Patched Android Applications
- App Integrity Protection
- Modern Android Integrity Primitives: Play Integrity API, Android Verified Boot 4 (AVB 4), and fs-verity
- Detecting Malicious Libraries
- Detecting Emulator/Rooted Devices
- Secure Implementation of WebViews
- Implementing Anti-Debug Checks
- Detecting Suspicious Device Reset
- Detecting Proxied Applications

Module 8: Android Kernel

- Android Boot process and Bootloader interaction
- Customizing and Building Android Kernel for Vulnerability Research
- Android Rooting Process
- Debugging Android Kernel and binaries
- Extract Android kernel from Boot image
- Symbolicating the Android Kernel
- Privilege Escalation on Android
- SELinux explained
- Overview of Kernel protections and bypasses
- Modern Android Primitives: Play Integrity API, AVB 4 (Android Verified Boot), and fs-verity filesystem integrity

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 specialised cybersecurity training and consulting to several commercial and defence organisations across the United States, Europe, and the Middle East and North Africa region.

Get in touch

8kSec.io
trainings@8ksec.io