ARMOURINFOSEC

Hands-On Cyber Security Training · Indore, India

Mobile Application Penetration Testing

Course Syllabus

Master offensive security testing of iOS and Android applications aligned to OWASP MASVS and OWASP MASTG. Learn static and dynamic analysis, reverse engineering, runtime instrumentation with Frida and Objection, root and jailbreak detection bypass, certificate pinning bypass, IPC abuse, insecure storage discovery, and mobile backend API testing.

// COURSE INFORMATION

Course Information

DURATION	3 Months / 12 Weeks / 90 Hours
LEVEL	Advanced
MODULES	14
FORMAT	Hands-on Labs / Hybrid (Online + Indore Classroom)

// COURSE OVERVIEW

Course Overview

The Mobile Application Penetration Testing course is designed for penetration testers, application security engineers, bug bounty hunters, and mobile developers who want to specialize in iOS and Android security assessment. The curriculum is anchored on OWASP MASVS verification requirements and the OWASP Mobile Application Security Testing Guide (MASTG). You will reverse engineer APKs and IPAs, perform static and dynamic analysis, hook runtime functions with Frida and Objection, bypass root and jailbreak detection, defeat certificate pinning, exploit IPC vulnerabilities, identify insecure data storage, and assess the mobile backend API attack surface using Burp Suite. Labs run on real jailbroken iOS test devices and Android emulators.

// LEARNING OBJECTIVES

Learning Objectives

- Scope and execute end-to-end mobile penetration tests on iOS and Android
- Perform static analysis of APKs and IPAs to extract secrets and identify weaknesses
- Conduct dynamic analysis and runtime instrumentation with Frida and Objection
- Reverse engineer Android and iOS binaries with jadx, Apktool, Ghidra, and Hopper
- Bypass root, jailbreak, and integrity detection mechanisms
- Bypass TLS certificate pinning across native, Flutter, and React Native apps
- Identify and exploit IPC vulnerabilities and insecure component exposure
- Discover insecure data storage in Shared Preferences, plist, SQLite, and Keychain

- Test mobile backend APIs through Burp Suite with proper proxying setup
- Map findings to OWASP MASVS and document them following OWASP MASTG
- Produce professional mobile penetration test reports with prioritized remediation

// PREREQUISITES
Prerequisites

- Solid penetration testing or application security foundation
- Working knowledge of HTTP and web application testing with Burp Suite
- Comfort with Linux command line and Git
- Basic understanding of Java or Kotlin (Android) and Objective-C or Swift (iOS) helpful
- Familiarity with Python scripting
- Laptop with 16GB+ RAM and virtualization support

// MODULE BREAKDOWN
Module Breakdown

01 Mobile Security Landscape & Threat Modeling

- iOS vs Android security architecture
- Mobile attack surface overview
- Threat modeling for mobile applications
- OWASP MASVS verification levels (L1, L2, R)
- OWASP MASTG methodology
- Engagement scoping and rules of engagement
- Reporting standards for mobile assessments

02 Android Platform Internals

- Android architecture and runtime (ART)
- APK structure and manifest
- Application sandboxing and permissions
- Activities, Services, Broadcast Receivers, Content Providers
- Intents and intent filters
- Android Keystore and KeyChain
- SELinux on Android
- Android security updates and patch levels

03 iOS Platform Internals

- iOS architecture and Mach-O binaries
- IPA structure and code signing
- App sandbox and entitlements
- Keychain Services
- Data Protection API and file protection classes
- URL schemes and Universal Links
- App Transport Security (ATS)
- iOS security updates and SEP

04 Lab Setup: Devices, Emulators & Tooling

- Setting up Android emulators with Genymotion and AVD
- Rooting Android emulators and physical test devices
- Configuring jailbroken iOS test devices
- Installing Frida, Objection, MobSF, jadx, Apktool, Drozer
- Configuring Burp Suite with mobile proxies and CA certs
- Ghidra and Hopper for binary analysis
- Cycript for legacy iOS hooking
- Lab hygiene and isolation

05 Android Static Analysis

- Decompiling APKs with jadx and Apktool
- Reading AndroidManifest.xml for exposed components
- Hunting hardcoded secrets, API keys, and endpoints
- Reviewing smali for sensitive logic
- Detecting insecure cryptography
- Automated scanning with MobSF
- Identifying debuggable and backup flags
- Mapping findings to OWASP MASVS

06 iOS Static Analysis

- Extracting and decrypting IPAs
- Analyzing Mach-O binaries with Hopper and Ghidra
- Inspecting Info.plist and entitlements
- Hunting secrets and endpoints in binaries
- Detecting insecure cryptography and weak randomness
- Class-dump and Objective-C runtime introspection
- Reviewing Swift symbols
- Static scanning with MobSF for iOS

07 Dynamic Analysis & Runtime Instrumentation

- Frida architecture and scripting fundamentals
- Objection for guided runtime exploration
- Method hooking and replacement
- Dumping memory, classes, and keychains at runtime
- Tracing native and Java/Kotlin function calls
- Hooking Swift and Objective-C methods
- Drozer for Android attack surface enumeration
- Building reusable Frida script libraries

08 Root & Jailbreak Detection Bypass

- Common root detection techniques on Android
- Common jailbreak detection techniques on iOS
- Hooking detection routines with Frida
- Patching binaries with Apktool and Objection
- Bypassing integrity checks (SafetyNet, Play Integrity, DeviceCheck)

- Defeating emulator and debugger detection
- Repackaging and resigning apps

09 Certificate Pinning Bypass

- How TLS pinning works on Android and iOS
- Detecting pinning implementations
- Bypassing OkHttp, TrustManager, and Network Security Config pinning
- Bypassing NSURLSession and AFNetworking pinning on iOS
- Frida and Objection pinning bypass scripts
- Defeating custom and native pinning libraries
- Handling Flutter and React Native pinning

10 Insecure Data Storage

- Shared Preferences and internal storage on Android
- SQLite databases and content providers
- External storage and scoped storage risks
- NSUserDefaults and plist files on iOS
- Keychain misuse and weak access groups
- Cached files, snapshots, and pasteboard leakage
- Logging sensitive data
- Auditing backups and cloud sync

11 IPC & Component Vulnerabilities

- Exported activities, services, and broadcast receivers
- Intent redirection and intent spoofing
- Insecure content providers and SQL injection
- Deep link and URL scheme hijacking
- Universal Link and App Link abuse
- PendingIntent and XPC vulnerabilities on iOS
- WebView misconfigurations and JavaScript bridge abuse
- Inter-app data leakage

12 Runtime Manipulation & Tampering

- Bypassing business logic via runtime hooks
- Forcing premium and feature flags
- Manipulating in-app purchase flows for testing
- Local authentication bypass (biometrics, PIN)
- Memory tampering and credential extraction
- Defeating obfuscation (ProGuard, R8, Swift Shield)
- Anti-tampering and integrity checks

13 Mobile Backend API Testing

- Proxying mobile traffic through Burp Suite
- Handling certificate pinning during proxying
- Authentication and session attacks against mobile APIs
- Authorization flaws and BOLA in mobile backends
- Mass assignment and parameter tampering

- GraphQL and gRPC mobile backends
- Rate limiting, abuse, and account takeover paths
- Mapping findings to OWASP API Security Top 10

14 Reporting & OWASP MASVS Verification

- Structuring mobile penetration test reports
- Severity scoring tailored to mobile context
- Mapping every finding to OWASP MASVS controls
- Producing MASTG-aligned evidence
- Communicating risk to mobile and product teams
- Retesting workflow and remediation validation
- Continuous mobile security programs

// TOOLS & HANDS-ON LABS

Tools & Hands-On Labs

- Genymotion and Android Studio emulators with rooted images
- Jailbroken iOS test devices for hands-on iOS labs
- Vulnerable Android and iOS practice applications
- MobSF preconfigured for automated static and dynamic scanning
- Frida and Objection installed across all lab devices
- jadx, Apktool, Ghidra, Hopper, and Cycript toolchain
- Drozer attack framework for Android component testing
- Burp Suite with mobile proxy and CA configuration
- Custom Frida script library for pinning and root detection bypass
- Mobile backend API targets for end-to-end engagement labs

// TRAINING MODE

Training Mode

Every Armour Infosec course runs as a unified programme delivered in two parallel modes — the same curriculum, the same trainers, the same certification, regardless of how you join.

- ✓ Online Live Classes — real-time, instructor-led, fully interactive sessions
- ✓ On-Premise Classroom Training — in-person at our Indore centre (Sudama Nagar)
- ✓ Both modes run concurrently in every batch; switch between them as your schedule needs
- ✓ Same syllabus, lab access, and certification track for online and on-premise students

// CERTIFICATIONS & CAREER OUTCOMES

Certifications & Career Outcomes

This course aligns with industry-recognised certifications and prepares graduates for offensive-security, application-security, and infrastructure-security roles.

- OSCP+ (Offensive Security Certified Professional+) — Advanced Offensive Security Certification
- CEH (Certified Ethical Hacker)
- MASE (Mobile Application Security Expert)
- GMOB (GIAC Mobile Device Security Analyst)

- OWASP MASVS and MASTG aligned methodology
- eMAPT (eLearnSecurity Mobile Application Penetration Tester) foundations

// ENROL WITH ARMOUR INFOSEC

Enrol With Armour Infosec

Reach out to discuss enrolment, batch schedule, and lab access. Our Indore training centre runs both in-person and live online cohorts with placement assistance.

PHONE	+91 99777 47168
EMAIL	info@armourinfosec.com
ADDRESS	674, Sudama Dwar, Narendra Tiwari Marg, Sudama Nagar, Indore, Madhya Pradesh 452009
WEBSITE	https://armourinfosec.com

Scan to View Course Online

https://www.armourinfosec.com/training/mobile-application-penetration-testing/