

**Mastering iOS Penetration Testing (MIPT) — Course Content**

**Module 1 – Introduction to iOS Security & Pentesting**

* iOS architecture & security model
* iOS app lifecycle & sandboxing
* Differences between iOS and Android security
* Pentesting methodology for iOS apps

**Module 2 – Setting Up the iOS Pentesting Environment**

* macOS setup for iOS testing
* Xcode & iOS SDK installation
* Using the iOS Simulator vs. real devices
* Jailbreaking overview and tools

**Module 3 – iOS Application Fundamentals**

* IPA file structure & components
* Info.plist file analysis
* Understanding Objective-C & Swift basics
* iOS permission model

**Module 4 – Static Analysis of iOS Apps**

* Extracting IPA files from devices or App Store
* Reverse engineering with Hopper, Ghidra, and class-dump
* Searching for hardcoded credentials & API keys
* Analyzing Swift/Objective-C code for flaws

**Module 5 – Dynamic Analysis of iOS Apps**

* Using Frida & Objection for runtime instrumentation
* Hooking and modifying app behavior
* Monitoring network traffic with Burp/ZAP
* Real-time log monitoring with Console & syslog

**Module 6 – iOS Device Communication & File System**

* Accessing the iOS file system on jailbroken devices
* Extracting app sandbox data
* Understanding keychain storage

**Module 7 – Insecure Data Storage**

* Analyzing SQLite databases
* Insecure NSUserDefaults usage
* Unencrypted plist files & local storage issues

**Module 8 – Insecure Communication**

* Exploiting apps without proper TLS/SSL
* Bypassing certificate pinning
* Performing MITM attacks on iOS apps

**Module 9 – Authentication & Authorization Attacks**

* Bypassing biometric authentication (Face ID/Touch ID)
* Exploiting weak session management
* Privilege escalation within apps

**Module 10 – WebView Exploitation**

* Insecure WebView configurations in iOS apps
* JavaScript injection in WebViews
* Local file access via WebViews

**Module 11 – Insecure Code Practices**

* Improper use of cryptographic APIs
* Unvalidated input & parameter tampering
* Hardcoded API tokens and credentials

**Module 12 – Reverse Engineering & Code Patching**

* Disassembling and modifying app binaries
* Repackaging and resigning modified IPAs
* Bypassing jailbreak detection

**Module 13 – Exploiting iOS App Components**

* Deep link & URL scheme exploitation
* Attacking inter-app communication
* Abuse of custom URL handlers

**Module 14 – Jailbreaking & Privilege Escalation**

* Jailbreaking methods and tools (unc0ver, checkra1n)
* Security risks introduced by jailbreaking
* Exploiting vulnerabilities for root access

**Module 15 – iOS Malware Analysis**

* Identifying malicious iOS applications
* Dynamic malware behavior analysis
* Detecting spyware and rogue apps

**Module 16 – iOS Security Bypass Techniques**

* Bypassing jailbreak detection
* Circumventing SSL pinning
* Defeating anti-debugging protections

**Module 17 – Exploiting Native & Hybrid iOS Apps**

* Attacking Cordova, React Native, Flutter apps
* Identifying weaknesses in hybrid frameworks
* Injecting malicious JavaScript into hybrid apps

**Module 18 – Apple Watch & iOS IoT Security**

* Attack surface of Apple Watch apps
* Bluetooth & NFC exploitation
* iOS integration with smart devices

**Module 19 – Reporting & Remediation**

* Writing professional iOS pentest reports
* Creating PoCs for vulnerabilities
* Secure coding recommendations for iOS

**Module 20 – Final iOS Pentesting Project**

* End-to-end pentest of an iOS application
* Combining multiple vulnerabilities in chained exploits
* Final report & stakeholder presentation