Mobile Application Security Testing

[OWASP | SANS | PCI DSS | ISO27001]

Tests	Techniques	Tools	Apps

iOS Tests

- MASVS-STORAGE
- MASVS-CRYPTO
- MASVS-AUTH
- MASVS-NETWORK
- MASVS-PLATFORM
- MASVS-CODE
- MASVS-RESILIENCE

Generic Techniques

- ✓ Binary Analysis
- ✓ Tampering and Runtime Instrumentation
- ✓ Static Analysis
- ✓ Reverse Engineering
- ✓ Dynamic Analysis

Penetration Tester: Bhabesh Kumar Das

STORAGE

- Testing Local Data Storage
- Testing Memory for Sensitive Data
- Checking Logs for Sensitive Data
- Finding Sensitive Data in the Keyboard Cache
- Determining Whether Sensitive Data Is Shared with Third Parties
- Testing Backups for Sensitive Data

CRYPTO

- Verifying the Configuration of Cryptographic Standard Algorithms
- Testing Random Number Generation
- Testing Key Management

AUTH

- Testing Local Authentication

NETWORK

- Testing Data Encryption on the Network
- Testing the TLS Settings
- Testing Endpoint Identity Verification
- Testing Custom Certificate Stores and Certificate Pinning

PLATFORM

- Testing Universal Links
- Testing App Extensions
- Determining Whether Sensitive Data Is Exposed via IPC Mechanisms
- Testing UIPasteboard
- Testing App Permissions
- Testing for Sensitive Functionality Exposure Through IPC
- Testing Custom URL Schemes
- Testing UIActivity Sharing
- Determining Whether Native Methods Are Exposed Through WebViews
- Testing WebView Protocol Handlers
- Testing iOS WebViews
- Checking for Sensitive Data Disclosed Through the User Interface
- Testing Auto-Generated Screenshots for Sensitive Information

CODE

- Testing Enforced Updating
- Checking for Weaknesses in Third Party Libraries
- Make Sure That Free Security Features Are Activated
- Testing Object Persistence
- Memory Corruption Bugs

RESILIENCE

- Testing Jailbreak Detection
- Testing Emulator Detection
- Testing File Integrity Checks
- Making Sure that the App Is Properly Signed
- Testing for Debugging Code and Verbose Error Logging
- Testing Obfuscation
- Testing for Debugging Symbols
- Testing Anti-Debugging Detection
- Testing Reverse Engineering Tools Detection
- Testing whether the App is Debuggable

iOS Security Testing Techniques

- Installing Apps
- Accessing App Data Directories
- Method Hooking
- Waiting for the debugger
- Get Loaded Native Libraries
- Patching
- Dynamic Analysis on iOS
- Reverse Engineering iOS Apps
- Emulation-based Analysis
- Method Tracing
- Process Exploration
- Basic Network Monitoring/Sniffing
- Reviewing Decompiled Objective-C and Swift Code
- Sandbox Inspection
- Bypassing Certificate Pinning
- Information Gathering - API Usage
- Execution Tracing
- Getting Loaded Classes and Methods dynamically

- Automated Static Analysis
- Get Open Connections
- Runtime Reverse Engineering
- Host-Device Data Transfer
- Listing Installed Apps
- Repackaging and Re-Signing
- Library Injection
- Dumping KeyChain Data
- Get Open Files
- Symbolic Execution
- Extracting Information from the Application Binary
- Obtaining and Extracting Apps
- Native Code Tracing
- Static Analysis on iOS
- Dynamic Analysis on Non-Jailbroken Devices
- Monitoring System Logs
- Accessing the Device Shell
- Retrieving Strings
- Disassembling Native Code
- Debugging
- Retrieving Cross References
- Patching React Native Apps
- Repackaging Apps
- Reviewing Disassembled Native Code
- Reviewing Disassembled Objective-C and Swift Code
- Information Gathering - Network Communication
- Exploring the App Package
- Setting up an Interception Proxy
- Decompiling Native Code

Testing Tools

To perform static analysis, dynamic analysis, dynamic instrumentation, etc. These tools are meant to help you conduct your assessments, rather than provide a conclusive result on an application's security status. It's essential to carefully review the tools' output, as it can contain both false positives and false negatives.

we prioritize including tools that meet the following criteria:

- Open-source
- Free to use
- Capable of analyzing recent Android applications
- Regularly updated

- Strong community support

Note: The functionality of the tools can also be affected by whether you're using a rooted or jailbroken device, the specific version of the rooting or jailbreaking method, and/or the tool version itself.

Generic Tools

- r2frida
- Ghidra
- Frida
- LIEF
- Iaito
- MobSF
- RMS Runtime Mobile Security
- Objection
- Frida CodeShare

iOS Tools

- class-dump
- Plutil
- security
- ios-deploy
- Frida-cycript
- Grapefruit
- Xcode Command Line Tools
- Xcode
- iProxy
- MachoOView
- class-dump-dyld
- Keychain-Dumper
- swift-demangle
- Frida-ios-dump
- Lldb

- Cydia
- Optool
- Cycript
- xcrun
- simctl
- radare2 for iOS
- objection for iOS
- SSL Kill Switch 2
- dsdump
- Frida for iOS
- otool
- Usbmuxd
- iOSbackup
- SwiftShield
- gdb
- MobSF for iOS
- Sileo
- BinaryCookieReader
- class-dump-z
- nm - iOS

Network Tools

- Wireshark
- MITM Relay
- OWASP ZAP
- Burp Suite
- bettercap
- tcpdump