WHITEPAPER



Secure Your Mobile Apps Against the OWASP Mobile Top 10



Executive Summary

Mobile application developers should be familiar with possible security risks mobile applications might encounter. Understanding these risks not only helps avoid pitfalls but also facilitates the development of secure applications.

The Open Worldwide Application Security Project (OWASP), a non-profit foundation, focuses on enhancing software security. OWASP offers freely available learning materials and tools designed to foster the creation of secure web and mobile applications. Among these resources is a compilation of the top ten most common threats to mobile applications, known as the OWASP Top 10.

Beyond the OWASP Top 10, the OWASP Mobile Project has produced comprehensive standards documentation and test procedures for Mobile Application Security Verification Standards (MASVS). These standards are closely linked to the OWASP Top 10 and are maintained as a living document on GitHub. While the OWASP Top 10 and the MASVS are valuable resources for all applications, they hold particular significance for financial, payment, and banking apps, as well as any application handling sensitive personal data. This document outlines how app shielding can assist you in addressing the OWASP Top 10.

App Shielding in Brief

According to Gartner, app shielding is a security solution integrated within the application to make it more resistant to attacks. Gartner classifies app shielding capabilities into prevention, detection, and "other" capabilities including Runtime Application Self-Protection (RASP). App shielding can aid developers and publishers in tackling some of the challenges highlighted by OWASP.

M1: Improper Platform Usage

This risk category covers the misuse of a platform feature, or failure to use platform security controls. This may include:

- · Correct usage of platform permissions
- Detection of custom keyboards
- Misuse of the keychain
- Android intents

App shielding detects the state of the device (root/jailbreak detection), blocks emulators, and identifies permissions enabled on the device.

M2: Insecure Data Storage

This risk category covers insecure data storage and unintended data leakage, both from disk and during runtime and user interaction. This may include:

- · Compromised file systems
- Incorrect storage of data
- Incorrect usage of keyboard cache
- Screen readers

App shielding addresses this category by blocking screen readers and key loggers, as well as offering a secure local storage mechanism with device binding (preventing the copying of data to another device).

M3: Insecure Communication

This risk category covers failure in securing the integrity of data in transit, such as poor handshakes or lack of network encryption. This may include:

- HTTP instead of HTTPS
- Incorrect SSL versions
- · Poor handshaking/weak negotiation (e.g., lack of certificate pinning)

Although app shielding itself does not offer network security, it enhances security in the implementation by protecting the application both at rest and at runtime.

M4: Insecure Authentication

This risk category covers server authentication of the end user or bad session management. This may include:

- · Failure to identify the user at all
- · Failure to maintain the user's identity
- Weaknesses in session management

This could be due to:

- Insecure authentication input
- Poor two-factor implementation
- Insecure user credentials

Although app shielding itself does not offer network security, it adds security to the implementation by protecting the application both at rest and at runtime.

M5: Insufficient Cryptography

This risk category pertains to the absence of, or inadequate use of, suitable cryptography. Cryptography is a vital component for protecting data stored on mobile devices. It's a risk area where errors can lead to severe consequences. Failing to address this category can result in:

- · Theft of app and user data
- · Unauthorized access to encrypted files

Common causes of failures in this area include:

- · Outdated or improperly configured cryptography
- Use of unproven cryptographic libraries

App shielding ensures that the security mechanisms cannot be removed from an app, protects apps against repackaging, and ensures that local data remains non-reproducible and appropriately encrypted. It accomplishes this by employing keys that are never statically present within the application.

M6: Insecure Authorization

This risk category comprises risks related to an adversary using a client to login as a legitimate user. Areas covered include:

- Password enforcement
- Token management in the client
- · Session management in the client

App shielding ensures the legitimacy of an application, and that it has not been tampered with. Code injection in an unprotected local client can bypass authentication if the application is already authorized. Furthermore, tokens and other sensitive elements can be securely stored or deleted on the device through secure local storage mechanisms.

M7: Client Code Quality

This risk category comprises basic security coding practices in application development and is an umbrella category for code-level implementation in the mobile application. Areas covered include:

- App signing
- Debugging information or access in a public application
- Exceptions management in the application
- Error handling in security controls

App shielding has self-contained functionality to protect against repackaging and verification of the application signature. Furthermore, it will remove debug information from the application code, and block debugger and emulator access. Security controls and device anomalies are managed and can be reported to the application code for specific handling.

M8: Code Tampering

This risk category covers an application's resilience against reverse engineering, as well as specific client-side attacks. It covers unauthorized modification of an application, either as downloaded from the application store or during runtime. Areas covered include:

- · Detecting whether the device is jailbroken or rooted
- Presence of debuggers
- Tampering of the executable
- Device binding
- Malware

App shielding is designed to make an application self-defending and is able to detect and react to all threats defined in M8. The reaction may be handled by the application after an anomaly is detected. The technology also supports device binding (MASVS 8.10) through secure local storage mechanisms.

M9: Reverse Engineering

This risk category is closely linked with M8 but is more focused on application analysis, code patterns, and risks related to code stealing and intellectual property theft. Areas covered include:

- Code obfuscation
- Presence of reverse engineering tools and frameworks
- Detection of running in an emulator
- Presence of debuggers
- Removal of (or bypassing) application security measures

App shielding is designed to make the application self-defending and can detect and react to all of these threats. The technology is designed not to leave evidence of its findings before reacting, and to ensure that an application is not usable in any way if an attempt is made to remove it.

M10: Extraneous Functionality

This risk involves code control and the prevention of actions such as exporting debug information or logs. It also includes hidden backdoor functionality risks.

While app shielding itself cannot prevent bad code or hidden backdoors in an application, the technology is designed not to leak security-related information or send out any data from the application. Additionally, by design, F5 Distributed Cloud Mobile App Shield directly addresses the MASVS requirement that all security controls have a centralized implementation (MASVS 1.7/ MSTG-ARCH-7).

