ZIMPERIUM®
MOBILE THREAT DEFENSE

ZIMPERIUM zSHIELD™

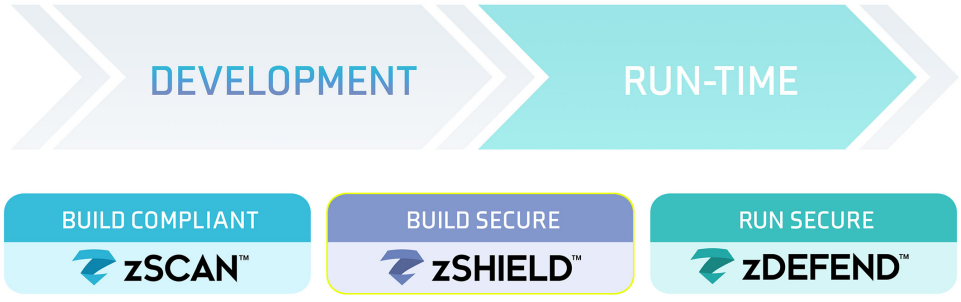
Solution Brief

STEP TWO IN MOBILE APP SECURITY: BUILDING SECURE APPS

Organizations that develop mobile apps are keenly aware of the need to secure mobile apps, but their efforts have been stymied by a highly fragmented set of solutions and no visibility into threats on end user devices. To solve both problems, Zimperium's Mobile Application Protection Suite (MAPS) identifies security and privacy risks during app development and protects/monitors apps from attacks while in use.

MAPS is comprised of three capabilities, each of which address a specific enterprise need:

Enterprise Need	MAPS Solution	Value
Build Compliant *What issues should be fixed before releasing our app?*	zSCAN™	zScan helps organizations discover and fix compliance, privacy, and security issues within mobile apps before they are released as part of the development process.
Build Secure *How can we harden our app against reverse engineering or code tampering?*	zSHIELD™	zShield app obfuscation and anti-tampering functionality protects the app from potential attacks like reverse engineering and code tampering.
Run Secure *How can we protect our app from advanced attacks on end user devices?*	zDEFEND™	zDefend SDK is embedded in apps to help detect and defend against device, network and malicious app attacks.



This solution brief explains how Zimperium zShield helps organizations harden their apps against engineering or code tampering.

HARDENING YOUR APPS WITH ZIMPERIUM zSHIELD

Once a mobile app is publicly released, attackers can inspect it for exploitable coding errors and vulnerabilities. Zimperium zShield hardens and protects the app with advanced obfuscation and anti-tampering functionality to limit attacks such as reverse engineering, piracy, removing ads, extracting assets, extracting API keys and repackaging with malware.

zShield hardens and protects your apps in three primary ways:

- Obfuscation to Prevent Reverse Engineering
- App Tampering Visibility in the Wild
- Seamless Development & Security Integrations

OBFUSCATION TO PREVENT REVERSE ENGINEERING

To prevent reverse engineering attempts utilizing static analysis, zShield deploys multiple code hardening techniques to obfuscate code visibility.

Android	
Name obfuscation	zShield obfuscates names of classes, fields, methods, native libraries, resources, resources, assets, and resource XML attributes.
Control flow obfuscation	zShield obfuscates the control flow of the code inside the methods to hinder automated and manual code analysis.
Arithmetic obfuscation	zShield protects proprietary formulas by transforming simple arithmetic and logical expressions into difficult-to-analyze code.
JavaScript obfuscation	zShield optimizes and obfuscates JavaScript files and complete cross-platform applications built with Cordova, Ionic or React Native.
Native code obfuscation	zShield obfuscates JNI function names in native libraries and in the Dalvik bytecode.
Removal of Android logging code	Logging code can provide information about the structure and execution flow of applications. zShield removes logging, debugging and testing code to thwart any attempt at exploiting this information.
Encryption	zShield encrypts sensitive strings to prevent hacking attempts through trivial searches. It also encrypts classes, asset files, resource files and native libraries.
Code virtualization	Code virtualization transforms method implementations into instructions for randomly generated VMS.
Call hiding	zShield adds reflection to access-sensitive APIs, such as the standard Android APIs for signature validation or cryptographic operations.
Protection of WebView and Cordova	zShield encrypts the contents of WebView and Cordova/Phonegap applications (HTML, CSS, JS, etc.).

iOS	
Name obfuscation	zShield obfuscates identifiers in both Swift and Objective-C code to hide semantic information from reverse engineers. The most common reflection constructs are supported out-of-the-box.
Control flow obfuscation	zShield hides the original function logic to better shield your apps against automated and manual code analysis.
String encryption	zShield encrypts sensitive strings in your apps using a random algorithm and a new key for every single string to prevent API endpoints, tokens, etc. from leaking.
Arithmetic obfuscation	zShield transforms arithmetic statements into more complex but equivalent alternatives to conceal the original computation. The transformations yield different outcomes in every single build.

APP TAMPERING VISIBILITY IN THE WILD

Unlike other obfuscation solutions that rely upon manual pen testing to demonstrate effectiveness and have no active reporting, *zShield provides immediate and on-going reporting on tampering attempts.* zShield reports app tampering events into Zimperium's administration and reporting dashboard, zConsole, and offers comprehensive forensics.

zShield protects your apps against dynamic analysis and live attacks using various runtime self-protection mechanisms.

Android	
SSL pinning	zShield confirms the protected application or SDK is connecting to the intended servers, preventing man-in-the-middle attacks.
Certificate checks	zShield ensures your application is signed with the original certificate.
Tamper detection	zShield enables your app to detect illegitimate code modifications and to verify the integrity of individual files.
Root detection	zShield enables your app to control whether it is running on a rooted device or a device using a root cloaking framework.
Debugger and emulator checks	zShield enables your app to verify the integrity of its runtime environment by detecting debugging tools and emulators.
Hook detection	zShield enables your app to detect and prevent attempts by hooking frameworks to modify its behavior.

iOS	
Jailbreak and debugger detection	zShield enables your app to monitor the integrity of its runtime environment. It also enables you to determine how your application should react when it detects a potentially harmful environment.
Hook detection	zShield enables your app to detect and prevent attempts by hooking frameworks (i.e., Frida, Cydia Substrate and fishhook) to modify its behavior.
Repackaging detection	zShield makes sure your app has not been repackaged by a third party.

SEAMLESS DEVELOPMENT & SECURITY INTEGRATIONS

zShield transparently integrates into your build process and requires no changes to your source code. It provides plugins for all common build tools and development environments like Gradle, Android Studio, Ant, Eclipse, Maven, and custom builds.

After your app is optimized and obfuscated with zShield, it will report hacking and reversing attempts directly into your security information and event management (SIEM) system for further analysis and action. zShield's standard integrations enable your security teams to view mobile threats in the same console they currently use for managing threats from traditional endpoints and networks. Depending on the attack, the security team can quickly provide details to development to fix the issues and prevent future exposures from the threat.

REDUCE YOUR MOBILE APP RISKS WITH ZIMPERIUM zSHIELD

To learn more about Zimperium zShield or receive a demonstration, contact us today.

ZIMPERIUM®