DexGuard
GUARDSQUARE

State-of-the-art protection for Android apps

Hackers can use freely available tools to decompile and inspect your Android applications and SDKs to gain insight into their internal logic. This leaves apps vulnerable to various forms of abuse, including intellectual property theft, cloning, credential harvesting, API key extraction and code tampering.

DexGuard has been designed to protect native (**Java**, **Kotlin**) and cross-platform (**Unity**, **Cordova**, **Ionic**, **Flutter**, **React Native**, **and other JavaScript-based**) apps and SDKs for Android against reverse engineering and tampering. DexGuard applies multiple obfuscation and encryption techniques to the code of applications and SDKs and integrates runtime self-protection mechanisms (RASP). The applied layers of protection make it virtually impossible to gain access to their internal logic and to modify their intended behavior.

Seamless integration

- DexGuard is a command-line tool that processes, optimizes and protects Android applications and libraries. It enables you to fully protect your application or SDK without requiring you to share or alter the source code.
- DexGuard offers built-in support for both native Android (Java, Kotlin) and cross-platform applications (Cordova, Ionic, Flutter, React Native, Unity). DexGuard's functionality can be extended with the NDK add-on to process and protect native libraries.
- DexGuard provides functionality to help you efficiently and effectively protect your application or SDK; its **Protection Report** helps you validate and improve your protection setup before release.
- DexGuard integrates seamlessly with Guardsquare's real-time threat monitoring platform, **ThreatCast**. ThreatCast gives you visibility into the actual threats facing your app and enables you to adapt your security configuration to the constantly evolving threat landscape. Free ThreatCast access is included in your Guardsquare license.
- DexGuard is backward compatible with ProGuard. This makes it easy to upgrade: you can reuse your ProGuard configuration and implement DexGuard's additional layers of protection.

DexGuard protects your applications and SDKs against static analysis using multiple code hardening techniques

Name obfuscation

DexGuard obfuscates the names of classes, fields, methods and native libraries, as well as the names of resources, resource files, asset files and resource XML attributes.

Control flow obfuscation

DexGuard obfuscates the control flow of the code inside the methods to hinder automated and manual code analysis.

Arithmetic obfuscation

DexGuard protects proprietary formulas by transforming simple arithmetic and logical expressions into difficult-to-analyze code.

Data encryption

DexGuard 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 virtual machines.

Call hiding

DexGuard adds reflection to access-sensitive APIs, such as the standard Android APIs for signature validation or cryptographic operations.

Native code obfuscation | Requires NDK add-on

DexGuard hardens native libraries against reverse engineering and tampering, including the interface between native libraries and application code.

Removal of Android logging code

DexGuard removes logging, debugging and testing code to thwart any attempt at exploiting this information.

DexGuard shields your applications and SDKs against dynamic analysis and live attacks using various runtime self-protection mechanisms (RASP)

Certificate checks

DexGuard gives your application the ability to ensure it has been signed with the original certificate.

Root detection

DexGuard enables your application or SDK to control whether it is running on a rooted device or a device using a root cloaking framework.

Tamper detection

DexGuard enables your application or SDK to detect illegitimate code modifications and to verify the integrity of individual files.

Debugger and emulator checks

DexGuard enables your application or SDK to verify the integrity of its environment by detecting the use of debugging tools and emulators.

Hook detection

DexGuard enables your application or SDK to detect and prevent attempts by hooking frameworks to modify its behavior.