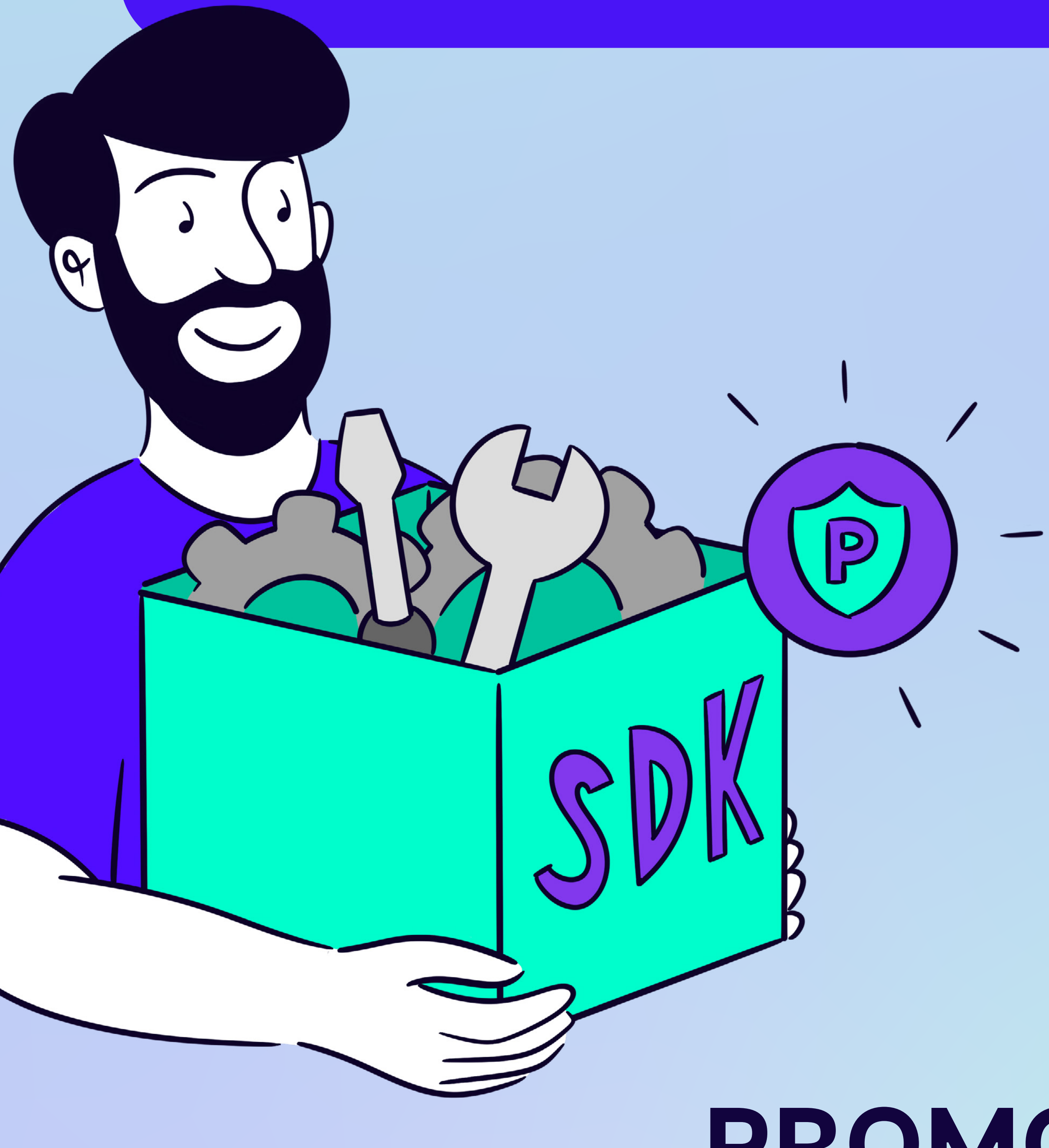
Promon SDK Protection™



PROMON

Promon SDK Protection™

Unlock the full potential of your SDKs and stay ahead of attackers

Promon SDK Protection™ delivers advanced security measures designed specifically for Software Development Kits (SDKs) across mobile platforms. By embedding robust code obfuscation and runtime controls directly into the compiled code, the solution ensures your SDKs—and the apps that rely on them—are shielded from reverse engineering, tampering, and unauthorized access.

Now includes Java-bytecode VM obfuscation

In today's mobile ecosystem, a typical app integrates around 30 SDKs, with up to 90% of its code derived from external sources. This exposes SDK providers and their customers to security risks, increasing the need to safeguard their software's integrity. The threat is real: SDK misuse or compromise can trigger not only financial repercussions but also tarnish a company's reputation.

This widespread reliance puts SDK producers—including software companies, banking entities, game engine developers, streaming services, and specialized firms in identity verification, DRM, and security—under intense scrutiny. They face the challenge of adhering to stringent regulations like eIDAS 2.0, DORA, and PSD2 while also protecting their intellectual property from reverse engineering and mitigating security risks such as repackaging attacks.

A staggering 92% of companies experienced a breach in the previous year due to vulnerabilities in applications developed in-house. They can expose your SDKs to hackers, enabling them to uncover vulnerabilities they can exploit to target your business.

For SDK providers like you, it's not just about protecting your products. Reducing the chance of vulnerabilities being passed to hosting apps adds real value to your SDKs. You need to boost security, address your customers' concerns confidently, and maintain trust.

Compatibility and support*

Platforms

- iOS
- Android native
- Android Java bytecode (AAR, JAR & ZIP)

Languages

- Swift
- Rust
- Objective-C
- C/C++
- Java
- JavaScript
- Kotlin

Architectures

- ARM (arm64, armv7 32-bit)

*Support for additional platforms and CPU architectures is upcoming.

Your strategic outcomes with Promon SDK Protection™

1. Mitigate security vulnerabilities

Protect your valuable intellectual property:
SDK Protection uses binary code obfuscation to prevent unauthorized access, reverse engineering, adversarial analysis, and proprietary code theft. This means your unique algorithms and sensitive data remain exclusive.

Strengthen your security posture:
As an SDK producer, securing your SDKs directly mitigates the risk associated with operating on untrusted devices and within vulnerable applications. This proactive approach allows you to detect and react to attacks, effectively making your SKDs more resilient

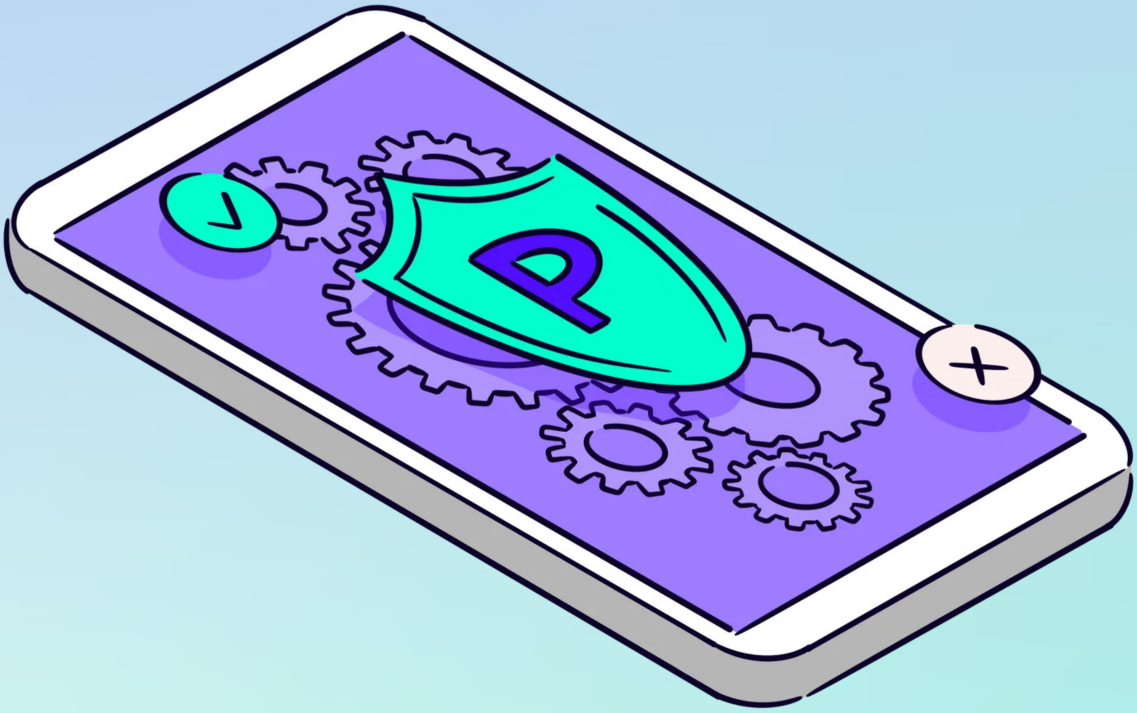
2. Reduce development and maintenance efforts

Seamless deployment:
SDK Protection uses a post-compile approach that can reduce the integration time and effort in your development frameworks. This requires no additional training, allowing your developers to focus on what they do best without worrying about security implementation details.

Lower maintenance costs:
SDK Protection is more cost-effective than typical security solutions that require adding obfuscation pre-compile into your development process. Changes in your framework, code, or workflow won't drive up your security expenses.

3. Enable uniform security across platforms

With SDK Protection, you get equal security for Android and iOS SDKs, with plans to add more platforms and architectures. As your product lineup grows into new platforms, the SDK Protection solution seamlessly adapts, keeping your technology secure without requiring additional security expertise from your developers.



What makes Promon SDK Protection™ different?

Rapid deployment

Designed for efficiency, SDK Protection can be deployed quickly, integrating into your workflow with minimal disruption. This hassle-free process accelerates the path from development to secure deployment.

Minimal developer impact

SDK Protection's post-compile approach streamlines its integration, preserving your existing codebase and workflows. Developers can concentrate on their work, free from the complexities of adding security measures.

Runtime protection

Besides code obfuscation, which provides protection from static analysis, SDK Protection enhances security when the app is in use. It guards against dynamic threats like code injection, hooking, and unauthorized debugging activities, adding a critical layer of protection for apps using your SDKs.

Scalable security

SDK Protection grows with your business. It keeps pace with new and evolving platforms, providing a robust security framework that supports your innovation and growth. A simple command line tool update is all it takes to ensure this security solution remains adaptable and up-to-date.

Promon SDK Protection™ code obfuscation techniques

NEW!

Virtual-machine (VM) obfuscation (Java)

Selected Java methods are moved out of the class-file bytecode and re-implemented inside a native virtual machine. Conventional decompilers and debuggers see only tiny stubs, while the true logic runs in opaque VM instructions, raising the skill and time required to reverse-engineer your SDK.

Section encryption (Native)

Large parts of each iOS/Android binary are AES-encrypted on disk. Keys are derived at start-up after integrity checks, blocking static analysis.

String encryption (Java)

All static strings—paths, API keys, URLs, error messages—are encrypted inside the bytecode. The clear text is revealed only in memory at runtime.

Control-flow obfuscation & flattening (Native and Java)

Control flow obfuscation and flattening inserts misleading branches, dispatcher loops and junk code. The real execution path becomes extremely hard to reconstruct in either environment.

Block splitting (Native)

Block splitting breaks sizable functions into small fragments, shuffles them with unrelated code, then links them back together with jumps. Combined with control-flow obfuscation, this makes algorithm flow almost impossible to follow.

Static-member shuffling (Java)

It moves static fields and methods across classes to break obvious data and logic groupings. This thwarts attackers who rely on clean class hierarchies.

Renaming (Java)

It replaces every class, method, and field name with meaningless identifiers and flattens the package tree. Renaming removes the semantic cues decompilers normally display.

Anti-decompilation (Java)

It alters bytecode metadata with edge-case values that crash or confuse popular decompilers. Forces attackers into slow, manual disassembly.

Debug stripping (Native and Java)

It deletes all debug symbols and source-line tables in binaries and bytecode. Without these breadcrumbs, automated inspection tools lose their map.

Integrity checking (Native)

Integrity checking embeds a checksum network over protected code sections. Any patching, hooking, or tampering trips the check and invokes your configured response.

Promon SDK Protection™ runtime controls

Hooking protection

Attackers can inject code into an Android or iOS app via code hooks, commonly facilitated by hooking frameworks. These injections are then used to modify the app, intercept messages, and read user input. SDK Protection detects the presence of code hooks and, based on your configurations, outright crashes the app and optionally calls an internal function for custom handling.

Root/Jailbreak protection

When an Android device is "rooted," it gains admin/root access to file locations that the manufacturer and/or carrier had originally restricted. On iOS devices, "jailbreaking" has a similar effect but is more about removing restrictions on the apps that can be downloaded and installed. These are actions that might be performed directly by an attacker, or they might be performed by a user who simply wants to customize their device. However, a rooted or jailbroken device is much more susceptible to malware, so it is essential to be aware of the risks. For both platforms, SDK Protection detects that the device's default restrictions are compromised and can be configured to act accordingly.

Debug protection

Attackers can run a debugger on an Android or iOS application to extract sensitive information and help them reverse engineer the app. SDK Protection detects the use of such debuggers and, based on your configurations, outright crashes the app and optionally calls an internal function for custom handling.

About Promon

Promon is the leader in proactive mobile app and SDK security. We make the world a little bit safer, one app at a time. Since 2006, some of the world's most impactful companies have trusted Promon to secure their mobile apps. Today, more than 1 billion people use a Promon-protected app.

Promon is headquartered in Oslo, Norway, with offices throughout the globe. Mobile app security is crucial to preserve and improve your business reputation. Request pricing or talk to an expert to learn more today.

Learn more »

Would you like to talk to an expert?

Mobile app security is crucial to preserve and improve your business reputation. Request pricing or talk to an expert to learn more today.

Book a meeting »

PROMON

promon.io
Promon AS • Cort Adelers Gate 30 • 0251 Oslo • Norway