Protect your business, revenue, and reputation with

Promon IP Protection Pro™

Promon IP Protection Pro™ delivers cross-platform, binary code obfuscation, protecting app code from reverse engineering and unauthorized modification. Your iOS and Android apps can be secured in minutes and ready for distribution, helping you to protect your intellectual property and prevent financial loss, customer churn, reputational damage, or regulatory compliance violations.

Compatibility and support

Platforms

- iOS
- Android

Languages

- Swift
- Rust
- Objective-C
- C
- C++

Architectures

- ARM
- Intel

Today, mobile apps have become vital touchpoints connecting enterprises with users across various platforms. However, this evolution brings significant challenges, particularly in safeguarding intellectual property. Promon IP Protection Pro™ is designed to address these challenges head-on.

Mobile apps, now integral to the secured enterprise infrastructure, are increasingly targeted by sophisticated cyber threats. These threats range from reverse engineering using tools like Ghidra, IDA Pro, and Hopper, to in-app fraud in sectors like financial services and retail. While server-side applications may be secure, mobile apps often remain vulnerable, potentially becoming gateways for attackers to access server-side applications.

The OWASP MASVS includes a section on resilience, which recognizes measures such as code obfuscation and anti-tampering as critical for increasing an app's resilience against reverse engineering.

Companies will soon be challenged by the intersection of generative AI and reverse engineering. This technology has the capability to quickly analyze extensive codebases, identifying patterns in obfuscated code to assist in de-obfuscation. Moreover, sophisticated AI models are being developed to potentially reverse obfuscated code back to its original form, making automated decoding a looming threat.

In this rapidly evolving digital landscape, adherence to the OWASP Mobile Application Security Verification Standard (MASVS) is crucial. OWASP MASVS sets the benchmark for mobile app security, emphasizing the importance of measures like code obfuscation and anti-tampering to enhance resilience against reverse engineering threats. Furthermore, as mobile apps frequently handle sensitive user data, complying with stringent data collection regulations such as GDPR, CCPA, and HIPAA is not just a best practice, but a necessity. This dual focus on security and compliance requires apps not only to be secure and reliable but also guardians of user data.

Secure your innovation with Promon IP Protection Pro™

- **Protect your valuable intellectual property**

  Keep your unique algorithms, app features, and essential processes safe from reverse engineering and unauthorized use and copying. You also prevent reputational damage, ensuring your brand remains trusted and respected in the market.

- **Safeguard your revenue streams**

  Avert reverse engineering attempts that could cause unauthorized access to your app code, data breaches or pirated app distribution. Secure in-app purchases, subscriptions, and premium features, keeping your revenue models safe.

- **Streamline your app code security for faster market launch**

  IP Protection Pro accelerates app launch while ensuring security, offering a quick, low-code setup, compatibility with multiple languages and frameworks, and seamless CI/CD system integration for agile development.

- **Meet strict regulatory requirements**

  Meet mobile app compliance and security requirements like OWASP MASVS standards, protecting code and data from tampering, and building user trust with its simplicity and efficiency.

- **Future proof your security**

  IP Protection Pro keeps you at the forefront of cyber defense, combating advanced threats such as reverse engineering and AI-driven attacks, ensuring ongoing protection against fraud and data theft.

What sets Promon IP Protection Pro™ apart

Cross-platform, post-compile, binary code obfuscation	IP Protection Pro secures your code across multiple platforms by operating on the binary code after it's compiled. It's compatible with a range of programming languages and supports both Android NDK and Xamarin.iOS. It protects your app's code as well as third-party libraries and SDKs without relying on Bitcode, offering the same protection on both Android and iOS, and is optimized for ARM and Intel architectures for reduced maintenance.
Low-code integration with no impact on toolchain	IP Protection Pro offers an easy-to-configure, low-code integration process that speeds up deployment and accelerates time to market. Protect your app's code quickly with minimal training. The solution is designed for efficiency, fitting smoothly into your CI/CD pipelines without altering your build system toolchain, unlike other code obfuscation products on the market.
Bindings for extra security	For added security, IP Protection Pro uses cryptographic binding to the app, ensuring that if its security features are tampered with, the app will not function. This adds an extra layer of protection against unauthorized modifications.
Extending SHIELD's multilayered protection	Expanding Promon SHIELD®'s obfuscation features, IP Protection Pro adds enhanced protection for your app's binary code. Runtime controls such as anti-debug, anti-hooking, and environmental controls such as root or jailbreak detection, screenshot prevention and keyboard injection prevention are provided through Promon SHIELD® all the time and in real time.

Promon IP Protection Pro™ code obfuscation techniques

Section encryption

Section encryption ensures that a binary (executable or library) cannot be statically analyzed (i.e., understood while on disk). Much of the binary file is encrypted by "sections" to prevent such analysis. Decryption begins shortly after start-up and after runtime integrity has been proven. Derivation of the keys used to decrypt each section depends on the Shield library being in a valid state.

Control flow abstraction

Control flow abstraction diverts call instructions within the code sections to a central dispatch function that hides the links between code blocks.With control flow abstraction, an attacker won't be able to see where a code jump goes to and whether it's to an external dependency or another internal symbol. Trying to extract a call graph from the code will be of limited use, because all calls go to the same place, and the graph is effectively flattened.

Block splitting (experimental)

If you have a large symbol with few or no dependencies, however, or if you want to increase the obfuscation of particular symbols, then block splitting can be useful. Block splitting takes the code blocks in one or more symbols/functions, splits them into smaller fragments, shuffles them with unrelated code, and inserts jumps to reconnect the control flow. Block splitting happens before control flow abstraction and so, if indirect links are chosen, this flow is hidden by control flow abstraction.

Integrity checking

To ensure that your code has not been tampered with (e.g., patched, hooked, or a breakpoint inserted), a checksum network covering all the application code is embedded. This feature is tightly integrated with control flow abstraction, so to use integrity checking, you must have control flow abstraction enabled. By default, both features are enabled to a limited level.

Debug stripping

Binary libraries and executables contain a surprising amount of debug information, even on "release" builds and especially on "debug" builds. Debug stripping removes debug information from the binary. Effective use of the debug stripping tool requires that symbols are not stripped from the release binaries before they are protected.