ZIMPERIUM®
MOBILE THREAT DEFENSE

App Shielding vs In-App Protection

April 2020

Mobile devices and the apps operating on them expose your backend systems to cyberattacks. Attackers exploit vulnerabilities in mobile operating systems and in your apps to spy on your users, grab their private data, or even steal their money. In response, many mobile app developers are using app shielding, sometimes called "app hardening," to protect intellectual property and reduce the chances of reverse engineering an app.

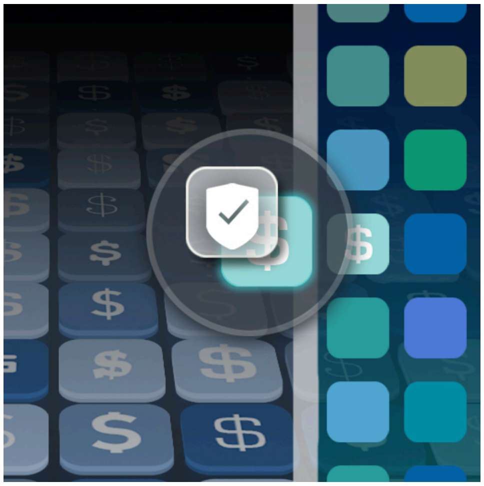
App hardening or app shielding are important to prevent reverse engineering, but will not detect a real-time cyberattack in the environment your app is running on a risky mobile device. Mobile apps need additional in-app protection to protect against attacks to your applications from exploits and fraud.

> *"The in-app protection market refers to security solutions implemented within the application (instead of the network or the operating system, for example) to make the application more resistant to attacks such as malicious data exfiltration, intrusion, tampering and reverse engineering. Enterprises use in-app protection to safeguard their software-based assets, and to protect their organization and customers from fraudulent attacks."*
>
> - Gartner "Market Guide for In-App Protection" by Dionisio Zumerle and Manjunath Bhat, July 3, 2019

Financial, healthcare, and, civic service apps containing personally identifiable information and intellectual property should implement both app shielding and in-app protection. Here, I'll describe the cases for each and provide example apps implementing in-app protection and what data they are protecting.

WHAT IS APP SHIELDING?

Application shielding is a set of technologies that modify and obfuscate an application's binary code. App shielding makes an application more tamper-resistant, protecting against intellectual property theft, piracy, and vulnerability discovery by reverse engineering, and unauthorized access. It creates a more resilient app making it more difficult to reverse engineer by obfuscating and encrypting the binary code.



App shielding capabilities include:

- **Code Obfuscation** – Obfuscation is the deliberate act of modifying source or machine code that is difficult for humans to understand. Making the code difficult to understand dissuades an attacker from trying to dig through your code for potential flaws, exploits, or reverse engineering your IP.

- **Debugging Detection** – The app prevents and detects debugging and responds to a debugger being present. All available debugging protocols must be covered.

- **Emulator Detection** – Emulator detection enables the app to detect when it is operating in an emulator. Emulators are used to reverse engineer an application and identify its communications with other services.

- **Root or Jailbreak Detection** – Jailbreak or root detection detects if a user removed restrictions that Apple or Google place on devices. Jailbreak detection is important but does not detect device compromises.

- **App Tampering** – While obfuscation helps prevent reverse engineering your static (non-running) code, attackers could still try to reverse your app by 'hooking' it while it's running. App tampering includes methods to detect if an attacker is attempting to reverse engineer your app while it's running.

WHAT IS IN-APP PROTECTION?

In-app protection differs from app shielding in that it detects real-time malware, network, and operating system attacks from inside the app. Mobile threat defense technology is placed inside a mobile application to detect and remediate threats to the app and the device.

In-app protection protects your backend systems from being compromised by mobile malware or vulnerable mobile devices your users may be carrying. While you cannot control your mobile users' device health, you can limit the devices from communicating with you if you detect malware, risky configurations, or network attacks on your users' mobile devices.

Some of the real-time threat data in-app protection provides you in addition to jailbreak and emulator detection include:

- **Malware Detection** – In-app protection detects multiple types of malware on the device and recommends remediations to the user. As users continue to migrate to mobile, so does malware. Several malware samples and remote access tools, RATs, like Bankbot, Monokle, Anubis, and Cerebus have appeared on mobile devices, and this trend will continue as more companies push more services to mobile devices. Many of these RATs monitor clipboard, password entry and even notifications to gather additional data about the user and credentials.

- **Zero Day Detection** – In-app protection detects bugs being exploited via the file system and operating system behaviors. Zero days are detected by dynamically monitoring indicators of compromise vs looking against a cloud library of known exploits, or simply the OS version / patch level installed on the device.

- **Network or Wi-Fi Mitigation** – In-app protection detects network connection manipulations and Man-in-the-Middle (MiTM) attacks. Your app should be able to identify "man in the middle" (MiTM), SSL stripping, and attempts to proxy or decrypt your users' app traffic to remediate the attack and produce threat forensics about the threat event. Network attacks represent the most significant percentage of worldwide mobile device attacks.

- **Device Configuration Risk** – Many conditions increase the risk exposure of mobile endpoints – the majority of which originate from the user being the administrator of his/her device. Users choose whether or not to update the operating system to patch known vulnerabilities, have a PIN code, or to jailbreak their device. In-app protection inside of your app provides you visibility into the health and condition, and ultimately the risk profile of those using your mobile apps. The information provides you intelligence on how to enforce conditional access or use cases based on measurable risks.

WHAT IS THE DIFFERENCE BETWEEN APP SHIELDING AND IN-APP PROTECTION?

App shielding and in-app protection are complementary. In many instances, for comprehensive risk reduction you should use both technologies in your mobile app for resiliency and to obtain mobile attack data. Deciding on which depends on what your application does and what type of information it stores and transacts with regularly.

Mobile development teams invest many hours in designing and building intuitive mobile apps. Many perform standard security checks and try to follow good coding practice. However, it's also critically important to secure the application during runtime. Mobile apps rely on the OS to provide a secure foundation to operate correctly. If a device is compromised, the entire security foundation of the mobile app is compromised as well.

A self-defending app is capable of operating independently of the device's native security features. Mobile apps with runtime security technology or in-app protection are capable of detecting malicious activity independently from native security features. The independent detection identifies real-time threats in the wild and ultimately avoids unacceptable risk exposure while protecting transactions and data on both the app and server sides.

HOW VULNERABLE ARE THE OPERATING SYSTEMS?

Mobile app developers rely on the underlying operating systems for security. However, while the mobile operating systems themselves are hardened to protect against attacks, they are not foolproof. Thousands of new vulnerabilities are discovered each year. Researchers and malicious actors alike continually test both Android and iOS looking for vulnerabilities through bug bounty programs, internal research, or to sell zero-days to the highest bidder. A remotely delivered persistent exploit on iOS can fetch up to $2M for an enterprising researcher.

In 2019, Mobile OS vendors created patches for 1,161 security vulnerabilities. Apple patched 306 CVEs (Common Vulnerabilities and Exposures), 64% of which were considered "critical" security threats.

In 2019, Google patched 855 CVEs, the majority of which (54%) were considered "critical" or "high" security threats. Critical CVEs are vulnerabilities allowing remote code execution or remotely bypassing security features. High severity CVEs, if successfully executed, could allow an attacker remote access to data or to bypass OS security features.

Even if patches for all of the vulnerabilities were delivered in record time, users also have to upgrade their devices to up their security. Relying on a consumer to patch, especially given the number of devices that are so old they cannot be upgraded any longer, is quite a gamble

WHY DO YOU NEED IT?

If you are following the OWASP Mobile Top 10, you most likely are already implementing app shielding and hardening procedures and tools. OWASP suggests you implement controls for authentication, data storage, cryptography, and reverse engineering. However, there is often room for significant improvement in mobile security and app hardening.

Zimperium previously inspected apps from the banking, travel, and retail sectors for security, privacy, and regulatory issues including the OWASP Mobile Top 10. We found most of the apps failed reverse engineering checks. Reverse engineering an app enables attackers to discover vulnerabilities and communication methods. One of the most notable examples of failing to implement hardening correctly and revealing weaknesses via a mobile app is Tesco bank.

Vulnerabilities in the bank's mobile applications left the door open for cybercriminals to brute force their way in to steal deposits. Many of the affected accounts were not mobile banking customers, however, the theft initiated from a poorly secured mobile app.

You may also need to implement in-app protection to comply with specific regulations and policies. There are mandates in PSD2 requiring mobile app developers to be able to detect third-party manipulations and to implement separate environments for authentication. GDPR and the California Consumer Privacy Act also enforce data privacy and the ability to remove private data. Other mandates like PCI DSS and HIPAA seek to reduce fraud in the payment and healthcare industries.

WHAT HAPPENS WHEN YOU INSTALL IN-APP PROTECTION IN A MOBILE APP?

A global bank recognized it had millions of customers and hundreds of thousands of employees connecting to its backend systems via fifty different enterprise and consumer mobile apps. This bank's IT staff realized it was unable to determine the health of its customers' or employees' devices interacting with bank systems. This bank found it necessary to gain visibility into risks associated with mobile devices interacting with its systems.

This US bank then conducted an exhaustive search and tested every mobile security solution available. The team unanimously chose Zimperium to secure its employee devices, consumer banking, and internal employee mobile apps. Zimperium's flexibility to support multiple EMMs, clouds, and devices proved Zimperium was the most enterprise-capable mobile security partner.

The first project the bank implemented under its advanced mobile security program was embedding mobile threat defense into its consumer mobile banking app. With threat defense deployed to its mobile banking apps, the bank now identifies suspicious transactions conducted on compromised or risky devices.

Before deployment, the bank lacked risk intelligence from customer devices. Now, it finally has data to pinpoint mobile banking fraud attributable to compromised devices or networks.

Immediately following the app update, consumers began conducting mobile banking transactions with the cyber protected app. The data surprised bank officials. The health of its customers' mobile devices (and the networks they were connecting to) was alarming.

In the first 30 days, the bank recorded nearly a million threats to its customers' mobile devices. Threat data is recorded only when the banking app is open. If the app is closed, the bank doesn't receive threat data from users since the sessions in other apps don't pose a fraud risk to the bank.

During those first 30 days, several million mobile banking users updated the apps and started generating mobile attack forensic data:

- **276,000** unsecured Wi-Fi detections;
- **1433** rogue access point detections;
- **495,000** devices had no PIN code;
- **166,000** enabled 3rd Party App Store access;
- **16,000** rooted or jailbroken devices;
- **1,000** recorded a system or file tampering event; and
- **1,500** rooting apps and another 500 apps containing malware, trojans, and spyware on its consumers' smartphones

The bank now has actionable data on how to prevent fraud via its mobile channels on more than $1.1B. All of this was made possible by updating an existing app with a mobile threat defense SDK that took less than 10 minutes to configure.

CONTACT US

For more information on how to build secure apps, obfuscate its code, or enable in-app protection, please contact us for a customized evaluation.

4055 Valley View, Dallas, TX 75244
Tel: (1) 844.601.6760
info@zimperium.com
www.zimperium.com

ZIMPERIUM®