

The Mobile Security Imperatives for SoftPOS Providers— and How Zimperium Can Help



Executive Summary

With the introduction of new standards and innovations in the financial and mobile payment industry, the software-based point-of-sale (SoftPOS) segment is poised for exponential growth—but only if critical security challenges are addressed. This ebook offers an in-depth look at emerging standards and security requirements in the SoftPOS segment. It also details how Zimperium and its industry-leading mobile security solutions enable SoftPOS solution providers a fast track to gaining PCI MPoC certification and, in turn, faster time to market.

An Introduction to SoftPOS and Why It's Poised for Extreme Growth

In the POS market, SoftPOS is gaining significant traction. Simply put, SoftPOS is the practice of using a smartphone to accept contactless card and mobile payments via a mobile application instead of a hardware payment terminal. In addition, these SoftPOS solutions enable merchants to receive payments on NFC-enabled mobile devices, such as Android or iOS smartphones and tablets.

SoftPOS isn't new. Small merchants and payment brands have been piloting mobile applications to accept contactless payments for a few years now, and solution providers like MyPinPad, Rubean, VivaWallet, Synthesis, Symbiotic, and PayFelix have emerged to serve this burgeoning market.

While SoftPOS first emerged on Android way back in 2017, Apple only unveiled its Tap to Pay on iPhone solution in 2022. With the release of the Mobile Payment on COTS (MPoC) standard by the Payment Card Industry (PCI) at the end of 2022, the first SoftPOS vendors will now begin to achieve their PCI MPoC certification. This will pave the way for wide adoption of SoftPOS by merchants.

The Emergence of MPoC and its Implications

The MPoC standard aims to provide an industry-wide standard for SoftPOS solutions. **By complying with the standard, SoftPOS solutions will enable merchants to receive payments securely on NFC-enabled devices, including smartphones and tablets running on Android and iOS.** PCI MPoC will accelerate merchants' transition to digital transactions and the global adoption of SoftPOS solutions by both small and micro merchants but also larger retailers and adjacent markets where SoftPOS technology can be beneficial (e.g. transportation and authentication).



Why the MPoC Standard Is Different

Until the end of 2022, PCI had two standards in the mobile payments arena: the Software-based PIN Entry on COTS (SPoC) Standard and the Contactless Payments on COTS (CPoC) Standard. These existing standards imposed restrictions for solution developers and end-users as they either had to use an external physical device (the secure card reader in the case of SPoC) or could not accept payment above the cardholder verification method (CVM) limit, as PIN was not supported in the case of CPoC.

In contrast to these existing standards, the new MPoC standard introduces modularity, new certification options, and new use cases, including support for offline transactions, component certification, and software-based PIN without the need to use a secure card reader.

Next to the functional expansion and advancements in certification options, the MPoC standard introduces a fundamental change for PCI in the security requirements themselves, which are moving from highly prescriptive to objective-based security requirements.

The objective-based security requirements bring an important shift from prescribing what a developer must do (for example, obfuscate their code) to what the solution needs to achieve (in the same example, to be highly resistant to reverse engineering). This critical change in the nature of the security requirements not only brings more design and implementation freedom to developers but also changes the approach to security from simplistic compliance to actual security assurance. This shift is like comparing the “letter of the law” to the “spirit of the law.”

By enabling any merchant to accept electronic (card or mobile-based) payments instead of cash, SoftPOS offers enormous potential. However, moving from traditional hardware-based POS technology to SoftPOS solutions comes with a challenge: securing the payment data.

Why Security is Key to Realizing the Potential of MPoC

The PCI MPoC standard is expected to accelerate merchants’ global adoption of SoftPOS solutions. To participate in the growth that will be fueled by the MPoC standard, developers will need to have their solutions PCI MPoC certified. This certification requires solutions to be evaluated by PCI-accredited security labs to ensure that the solutions effectively comply with the security requirements of the standard, which includes measurable security robustness requirements.

In contrast to PCI SPoC and CPoC, solution developers aiming to gain PCI MPoC certification must ensure that their SoftPOS solutions meet the attacker resistance thresholds as specified in the MPoC security requirements. This includes the protection of cryptographic keys and resistance to advanced reverse engineering and tampering of the SoftPOS mobile applications. In addition, **solutions must offer visibility into threats and compromise of the COTS platform as part of the attestation and monitoring system.**

These requirements are defined to prevent the disclosure or manipulation of assets such as the cardholder’s Primary Account Number (PAN) and PIN data.

The High Stakes of Addressing Security Imperatives

To guard against attacks and subsequent fraud, SoftPOS solutions must resist all relevant attacks and threats, including malware, and those instigated by criminal organizations, remote attackers, and malicious actors with physical access to the device running the SoftPOS app.

The stakes are high. If the SoftPOS app isn't adequately protected, the solutions can be abused in several ways, including consumers or attackers faking or refunding payments, merchants performing unauthorized transactions, and criminal organizations collecting card data for card-not-present fraud (CNP). Further, as the scale of SoftPOS adoption grows, the scale of this exposure will expand as well, exposing merchants, payment processors and issuers, and ultimately, consumers.

In the following sections, we look at two of the most critical requirements for securing SoftPOS solutions, including the specific threats that need to be addressed, why other approaches are falling short, and how Zimperium solutions uniquely address the demands from SoftPOS development teams looking to get a secure solution to market.

Requirement 1. Secure Mobile Applications

Traditionally, in-person POS technology has been limited to hardware devices that housed physical POS terminals. These physical terminals are purpose-built—the only thing they do is process transactions. Such POS terminals were designed and built with security in mind and relied primarily on security provided by the hardware platform. While there are ways to compromise such payment terminals, these systems must be validated against the PIN Transaction Security (PTS) standards, which require safeguards against pertinent attack vectors.

Over time, these purpose-built payment terminals evolved and transitioned to Android-based platforms, which meant systems started to be exposed to both hardware and software attacks.

The move to SoftPOS makes this evolution complete, presenting a transition from purpose-built hardware secured solutions to pure software-based mobile applications on consumer smartphones. In the process, SoftPOS introduces a much broader attack surface, particularly when compared with traditional hardware-based POS terminals.

Risks Posed by Mobile Devices

To effectively protect a mobile application against attacks, development teams must have a deep understanding of the relevant threats and the associated technical and business risks they pose.



SoftPOS solutions run on many different models of smartphones, with widely varying permutations of hardware and software. While traditional POS terminals were largely considered trusted platforms, this is not the case for mobile devices.

This distrust is well founded. There are also numerous and continuous new examples of zero-day and remote exploits for both Android and iOS smartphones, including exploits that impact the hardware-based security, source code, and trusted execution environments (TEEs) inside these devices. According to Zimperium's 2023 Mobile Threat Report, in 2022 an average of 77,000 unique malware samples were discovered each month. For the year, 925,000 unique malware samples were detected, up from 611,000 in 2021, which represents a jump of 51%.

Quite simply, with SoftPOS, the underlying platform provided by the smartphone can't be considered "trusted" – and this is fully recognized in the PCI MPoC standard. This means the SoftPOS application **can't solely depend on the security of the device or its mobile platform operating system.**

This should not come as a surprise as smartphones, in contrast to payment terminals, have not been designed, developed, or secured as payment terminals.

Exacerbating matters is the fact that mobile applications typically run on a wide variety of devices, including those from a range of vendors, with varying models and form factors, different versions of hardware and OSs, different security patching levels, apps downloaded from different app stores, and so on. Given this, application developers need to design and develop an application that effectively protects itself and the assets it processes, such as cryptographic keys, PIN, PAN, and transaction data. In the future, and with the release of iOS 17 this fall, iOS will allow sideloaded apps on iOS, instead of requiring attainment of apps from the AppStore. This will increase the risk for apps in general, as iOS will be an easier target for bad actors trying to exploit terminals on that system.

Common Attacks and Protection for Mobile Applications

Mobile applications are subject to various types of attacks, typically categorized as **static** and **dynamic** analysis. Let's take a closer look at each of these attack types.

Static Analysis

Static analysis refers to the practice in which malicious actors retrieve a target application and reverse the binary code back to a human-readable format. By analyzing the source code, an attacker can understand the functionality inside the application and extract static secrets, such as fixed or unprotected credentials, tokens, or cryptographic keys. Often, attackers will take a mobile application, reverse engineer it, and add malicious functionality. For example, an attacker can embed the ability to capture user credentials or authentication data. They'll then repackage the application with the malicious code and upload it to the app store, where it's hard to distinguish from the app of the original publisher.

Application developers can employ several practices to protect against static analysis and complicate the practice of app repackaging. For example, they can employ advanced code obfuscation, tamper resistance, and integrity protection.

Dynamic Analysis

Dynamic analysis is the practice of analyzing and manipulating an application during its execution—meaning during runtime. For example, attackers can run the application in a debugger or emulator to analyze its behavior or hook into its processes to extract data or manipulate its behavior.

How Attackers Leverage Static and Dynamic Analysis

The knowledge attackers gain from the static and dynamic analysis can be abused in various attack scenarios. With these insights, attackers can:

- wage so-called repackaging attacks, creating malicious versions of legitimate applications
- employ attacks by cloning deployed instances of an application and its data
- create malware that can exploit the legitimate application
- abuse vulnerabilities remotely or instrument the application



Limitations of Existing Tools and Approaches

Limitations of Hardware-based Technologies

Securing SoftPOS solutions is not an easy task. Doing so requires a thorough understanding of the solution, its design, and the security technologies, as well as the expertise to implement it with your engineers. In theory, solution developers have three main technologies available to help secure their SoftPOS app:

- Software-based security technology
- TEE
- Secure Element (SE)

In practice, hardware-based technology, such as TEEs and SEs, has proven to be restricted to smartphone OEMs as common application developers don't have access to this technology. In addition to the lack of access, fragmentation of the hardware-based technology has led most SoftPOS developers to secure their solution with software-based security technology to be able to offer extensive support for many different device brands and smartphone models and different operating systems.

Limitations of Traditional Software-Based Approaches

When it comes to software-based security, SoftPOS developers have been left with several options, which each have significant limitations:

- **Static software protection tools.** While there are software protection tools available to secure mobile applications, the challenge is that most app protection tools are static in nature and can't detect most security risks that are introduced by insecure platforms. Consequently, there is an urgent need for advanced threat detection capabilities. Particularly, since the future of SoftPOS solutions is on COTS devices, having an effective solution that works on all devices is critical.
- **Platform-based attestation.** Platform-based attestation solutions are easy to use because they are available out-of-the-box. However, these solutions also have specific drawbacks. Some have limited platform support. For example, some are only available on Android Google Mobile Services (GMS) and not on Android Open Source Project (AOSP). Many of these solutions can have a negative impact on performance. Further, alternatives rely on online detection, while such online services may sometimes become unavailable or unreachable.



The Requirements: Attestation that Yields Self-Protecting Apps

To establish robust mobile application security, teams must employ a defense-in-depth strategy. This includes identifying the risks and threats applicable to the mobile app and establishing a proper security design. This design must provide adequate safeguards against reverse engineering, tampering, and key and data extraction.

Therefore, it's critical that mobile applications are capable of protecting themselves. They need to remain secure when running in an insecure environment, even one that has been infected with malware or compromised as a result of exploited vulnerabilities in the main OS (for example, Android or iOS), and even when local or remote attackers are directly targeting the mobile application.

To safeguard mobile applications against static and dynamic analysis, teams must embed the build-time protections outlined above. However, these static defenses are not enough for SoftPOS applications, which need robust, ongoing security.

In order to establish a risk management-based approach to mobile application security, the mobile application needs to be able to attest to the status of the environment in which it is operating. In other words, the mobile application needs to be aware of potential and active threats in its surrounding environment. This **requires the ability to gain runtime threat visibility** and take appropriate mitigating actions if threats are detected.

Having actionable threat visibility at runtime fuels the next evolution of mobile application security and risk management, enabling policy-driven application self-protection. While many static, build-time solutions pose significant limitations, most SoftPOS solution developers lack the time, resources, and expertise needed to create the advanced attestation capabilities they require.



The Solution: Zimperium zDefend for Device Attestation

zDefend, one of the pillars of Zimperium MAPS, offers dynamic, on-device attestation capabilities that are powered by machine learning. Furthermore, these defenses are independent of the underlying COTS platform or hardware.

zDefend has a number of unique characteristics that are highly beneficial for SoftPOS developers. By using Zimperium zDefend, developers can accelerate their time to market, boost security, and better meet their engineering and business objectives faster and more efficiently.

Mapping Zimperium zDefend to the PCI MPoC Standard

Attestation and monitoring represent core components of PCI MPoC. zDefend is a fundamental building block within an effective attestation and monitoring system. zDefend offers industry-leading detection capabilities that enable attestation of applications and COTS platforms. zDefend has been designed to meet the relevant security requirements defined in the MPoC standard while equipping SoftPOS developers with a flexible, highly configurable solution.



- Simple to implement and supports multiple platforms, including Android GMS, Android AOSP, and iOS
- Can deliver threat or attestation data easily and securely to the SoftPOS Attestation and Monitoring back-end systems
- Provides complete on-device detection for both known and unknown threats
- Continues to detect all threats, even when the device is offline
- Has minimal impact on the responsiveness and performance of applications
- Offers both proactive and reactive threat detection
- Has detection capabilities that can be updated over-the-air (OTA), providing continuous up-to-date protection without requiring applications to be rebuilt, redistributed, or republished
- Is massively scalable, with deployments exceeding 75 million users, and without placing limitations on the number of attestations that can be performed



Mapping zDefend against PCI MPoC

Mobile Payments on COTS (MPoC) defines security requirements, test requirements, and guidance for entities involved in the development, deployment, and operation of the mobile payment acceptance solutions that use COTS devices.

As zDefend is a fundamental building block for threat detection within an effective A&M system, the below table shows the MPoC requirements related to the attestation and monitoring software and indicates which are covered by Zimperium zDefend and which are the responsibility of zDefend integrators.

| Req. | Description | Explanation and compliance |
|--------|--|--|
| 1C-1.1 | Documentation on the coverage of the attestation and monitoring (A&M) must exist. | Zimperium provides comprehensive documentation on the detection capabilities of zDefend so that MPoC developers can readily integrate this in their MPoC documentation. |
| 1C-1.2 | The A&M functionality must cover the complete lifecycle of the MPoC SDK and MPoC Application, starting from installation through decommission. | zDefend enables MPoC developers to detect threats from the moment the MPoC application is installed to the moment of removal. |
| 1C-1.3 | The attestation and monitoring (A&M) checks must cover the entire security-sensitive MPoC SDK code and execution flows that handle assets. | zDefend enables MPoC developers to proactively query zDefend if there are any active threats, e.g., before executing a specific function. MPoC developers can use callbacks to be informed in realtime in case a threat is detected at any moment. |
| 1C-1.4 | The A&M system must attest the COTS platform and MPoC Application / MPoC SDK. | zDefend enables MPoC developers to detect mobile malware, system tampering (e.g. rooting, jailbreaking), debuggers and emulators and app tampering events (e.g. hooking). |
| 1C-2.1 | The information that is collected for attestation and monitoring purposes must be documented. | Zimperium provides comprehensive documentation on which data is collected and also enables MPoC developers to configure this. MPoC developers can readily integrate this in their MPoC documentation. |
| 1C-2.2 | The A&M data must reflect the current state of the MPoC SDK, MPoC Application, COTS platform, and peripheral devices, in addition to any security-relevant changes or measurements that have occurred since the last communication to the A&M backend. | zDefend provides the attestation data on the MPoC SDK, MPoC Application and COTS OS. The MPoC developer is responsible for implementing the A&M Back-end Systems to keep track of the current state and state changes. |
| 1C-2.3 | The A&M data sent to the backend must contain a freshness indicator. The A&M data and the freshness indicator must be protected against tampering. | The threat data provides by the zDefend SDK can be consumed by the SoftPOS solution directly on-device, in which case the SoftPOS developer will implement the secure channel and protection of threat data. The second option is to retrieve the threat data through the zConsole to which zDefend is connected. The secure channel between the zDefend SDK and zConsole meets the applicable secure channel and data protection requirements. |
| 1C-2.4 | The A&M functionality must include an aspect within the MPoC SDK which performs continual monitoring. | See 1C-1.3. zDefend is active while the MPoC SDK is active and can inform the MPoC SDK of any detected threat in real-time through callbacks |
| 1C-3.1 | Documentation must exist that describes the actions that can be taken if the attestation and monitoring (A&M) indicates that the MPoC SDK, MPoC Application, or COTS platform is potentially compromised. | Not applicable. Implementing and documenting response actions upon detected threats is the responsibility of the MPoC developer. |
| 1C-3.2 | Potential tampering events detected by the MPoC SDK must be reported to the attestation and monitoring (A&M) backend. | zDefend provides two ways of collecting the threat data. MPoC developers can collect all threat data through a single, secure API provided by zConsole. Alternatively, MPoC developers can collect the threat data directly from the zDefend SDK and report it directly to their A&M backend. |
| 1C-3.3 | It must be possible for the MPoC SDK to disable processing in the event of tamper indications. | Not applicable. zDefend will provide the tamper indications. Any response action, including disabling processing, is the responsibility of the MPoC developer. |

| Req. | Description | Explanation and compliance |
|--------|--|--|
| 1C-3.4 | The MPoC SDK must perform an attestation with the A&M back-end periodically and at least every 60 minutes of continuous operation, or suspend further payment processing until such attestation is performed. | zDefend performs continuous scanning while the MPoC SDK or MPoC Application is active. The MPoC developer can register a callback in order to be notified of any threat detections. Method-calls can be used to retrieve the current active device status / active threats. |
| 1C-3.5 | An MPoC SDK that is suspended or otherwise halted must perform an A&M attestation prior to any payment processing. | zDefend can be used inform the MPoC SDK of any active threats by querying the zDefend SDK to perform an attestation at any moment in time. |
| 1C-3.6 | A&M results that may require the cessation of payment acceptance must not be communicated through the COTS device. | Not applicable. This is the responsibility of the MPoC developer and is not related to threat detection and attestation. |
| 1C-4.1 | The protections provided to the A&M system must be detailed. | The zDefend SDK is protected against tampering and reverse engineering through Zimperium's zShield, as detailed in the implementation guide. |
| 1C-4.2 | A secure channel must not be relied upon as the sole protection for A&M data during transmission. | See 1C-2.3 |
| 1C-4.3 | The local time source used by the MPoC SDK must be secured against tampering or alteration. | Not applicable. This is applicable to the MPoC SDK itself. Regardless, zDefend follows PCI MPoC security lab recommendations on this aspect and meets this requirement. |
| 1C-4.4 | The A&M backend must be able to detect failures in the A&M functions within the MPoC SDK. | Not applicable. This is a A&M backend related capability and does not related to the zDefend SDK. |
| 1C-4.5 | The A&M used by the MPoC SDK must be resistant to tampering to an attack rating of 25 points using the attack- costing framework in Appendix B. | In order to tamper with the zDefend SDK an attacker has to evade the zDefend detection capabilities and overcome the protections of the SoftPOS application and zDefend SDK itself. The zDefend SDK is additionally protected against tampering and reverse engineering through Zimperium's zShield. |
| 1C-5.1 | A&M back-end operation security guidance information must exist that explains how the attestation and monitoring (A&M) is securely configured and operated. | Not applicable. This is the responsibility of the MPoC solution developer. Zimperium zDefend does come with detailed documentation explaining all possible configuration options for the threat detection capabilities of the zDefend SDK. |
| 1C-5.2 | The security guidance information must detail how to securely integrate the A&M software component into the MPoC Application. | Not applicable. The MPoC solution developer is responsible for writing the Security Guidance. Zimperium supports this with the detailed implementation guides. |
| 1C-5.3 | Security guidance information must detail what A&M features are configurable, the processes for setting or changing these configurations, and how the applicable settings may affect the security and functionality of the overall MPoC Solution. | zDefend comes with detailed supportive documentation explaining all detection and configuration options. |
| 1F-2.2 | The MPoC SDK A&M component must support a separate attestation policy for offline operation. | Supported. zDefend performs its detection completely on-device. As such zDefend is not reliant on any backend system for detection of any threats and will continue to function in full when the mobile device or the A&M backend is offline. |
| 1F-2.5 | The results of the A&M security checks that are not dependent on the backend must be resistant to tampering to an attack rating of 25 points using the attack-costing framework in Appendix B. The verification/assessment of these security check results cannot be delayed until connection to the A&M backend is reestablished. | See 1C-4.5 and 1F-2.2. With zDefend, all detection capabilities remain effective, regardless of being online or offline. |

Requirement 2.

Secure PIN Entry

PCI MPoC enables software-based PIN entry on the same COTS device that interacts with the NFC-enabled consumer payment method, including debit and credit cards, wearable devices, and mobile wallets. As a result, **this standard represents a significant breakthrough within the payment card industry**. However, to employ this standard, organizations must address significant security requirements in order to safeguard the PIN data from unauthorized disclosure.

Risks: How PINs are Exposed

Software-based PIN entry on COTS mobile devices introduces significant security risks. That's because traditional graphical user interfaces (GUIs) in mobile applications are built using components provided by the operating system. These user-interface components are vulnerable to several attack techniques that enable malicious actors and malware to intercept or retrieve PIN data.

Here are a few of the common techniques that attackers can use to steal sensitive information, including PINs:

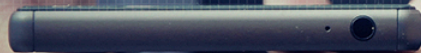
- Screen recording
- Activity hijacking
- Clickjacking or tapjacking

The above attacks are typically deployed by first tricking the merchant into installing a malicious app. This malicious app then begins monitoring the activities of the SoftPOS app and attempts to intercept sensitive information. If a malicious actor acquires elevated device privileges, an even stronger class of attacks is possible. (This technique is customarily called "rooting" in Android devices or "jailbreaking" in iOS devices). This access gives the attacker additional power, enabling them to monitor, record, and analyze all memory and execution processes on the mobile device. These access methods can even enable side-channel attacks, in which threat actors use device peripherals, such as the gyroscope or accelerometer, to capture the PIN. All these threats have to be considered and mitigated.

Please enter your PIN

* * * * *

To combat these threats, solution providers must ensure that the PIN entry component not only thwarts as many attacks as possible but also keeps the PIN data and its encryption keys secret, even if the attacks cannot be prevented.



Limitations of Existing Approaches

Limitations of Device-Based PIN Security

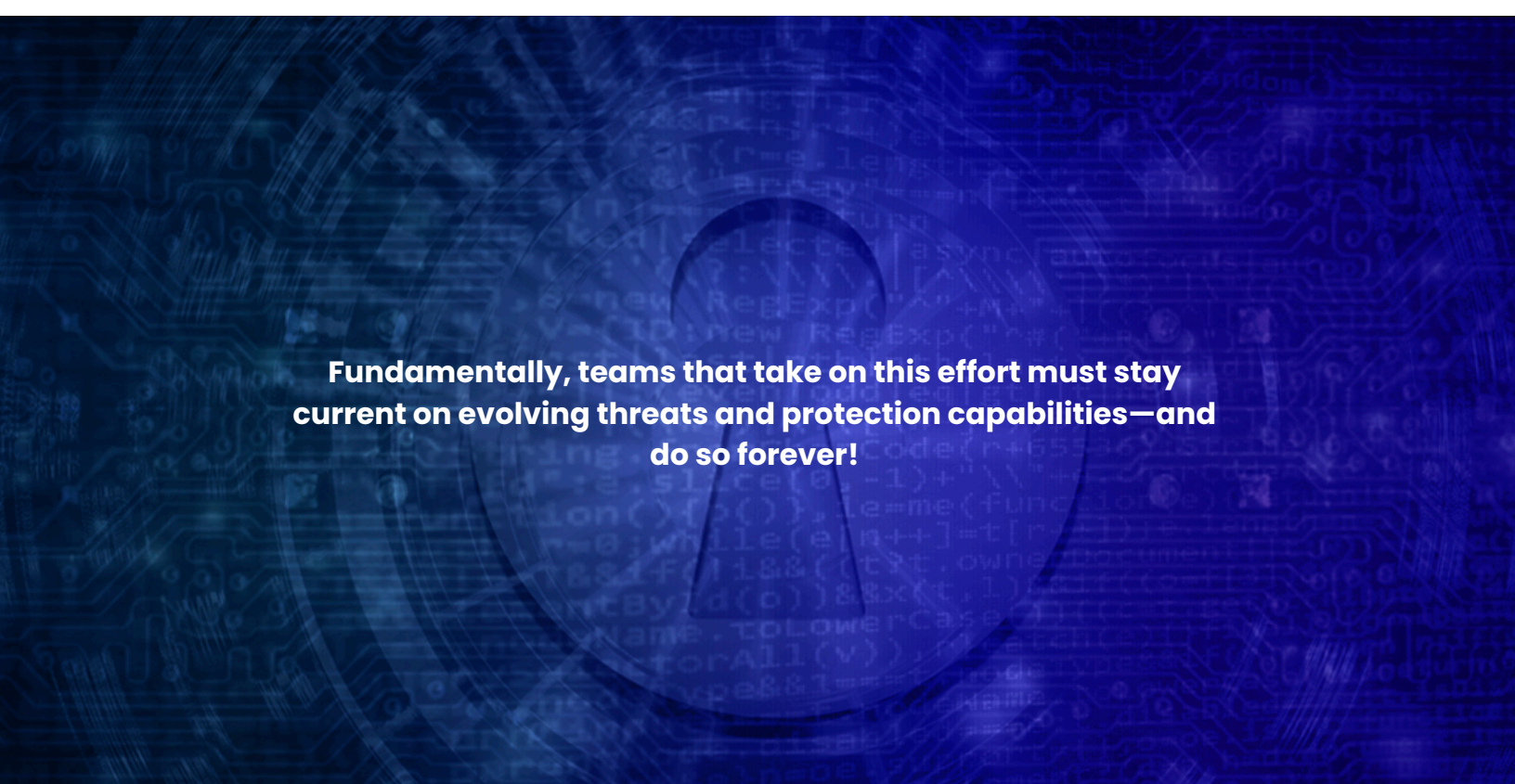
Modern and high-end Android smartphones typically secure “generic” PIN entry (for user authentication) through Trusted User Input (TUI) functionality as part of the device’s TEE. However, this functionality has not been a viable option for SoftPOS solution developers for a number of reasons:

- The TUI functionality, as part of the device TEE, **requires adaptations** to be useful for SoftPOS solutions. However, as smartphone manufacturers restrict access to the TEE and its TUI functionality, this is not a viable option for SoftPOS solution developers.
- The TUI functionality is platform specific, which introduces significant technical fragmentation and associated complexity for SoftPOS developers.
- Given it is implemented within a TEE, the security of TUI depends on how secure the TEE is. TEEs are frequently subject to attacks, which can result in threat actors gaining complete access and control over the TEE. As the patching cycles for TEEs are typically long, often spanning multiple months, this would result in an unacceptably long time frame in which the solutions would be vulnerable.

Risks of a Build Approach to Securing PIN Entry

It is possible to build capabilities for securing PIN entry in-house. While at first glance, this may be an appealing option, the reality is that this requires considerable development effort and is typically too time-consuming and cost-prohibitive for most developers. The main reason to avoid building it yourself is that **creating a secure solution requires ongoing expertise** in several areas:

- Cryptographic design and protection of cryptographic keys
- Advanced attack techniques being employed against mobile applications and platforms
- Android platform security and mobile application security capabilities
- Regulatory compliance standards



Fundamentally, teams that take on this effort must stay current on evolving threats and protection capabilities—and do so forever!

The Solution: Zimperium Secure PIN

zKeyBox provides a set of tools that enable SoftPOS developers to implement a secure GUI-based PIN entry mechanism in Android applications. Combined, these tools form an add-on feature of zKeybox called **Secure PIN**. This highly (graphical) configurable add-on is designed to enable SoftPOS developers to meet the relevant MPoC standard's requirements for securing PIN entry.

With Zimperium Secure PIN, SoftPOS developers get access to a PIN library developed by mobile security experts. This library is easy to integrate and compliant with the applicable security requirements of PCI MPoC. **Using Secure PIN will accelerate your time to market and help you optimize your engineering efforts, so you can better support key business objectives.**

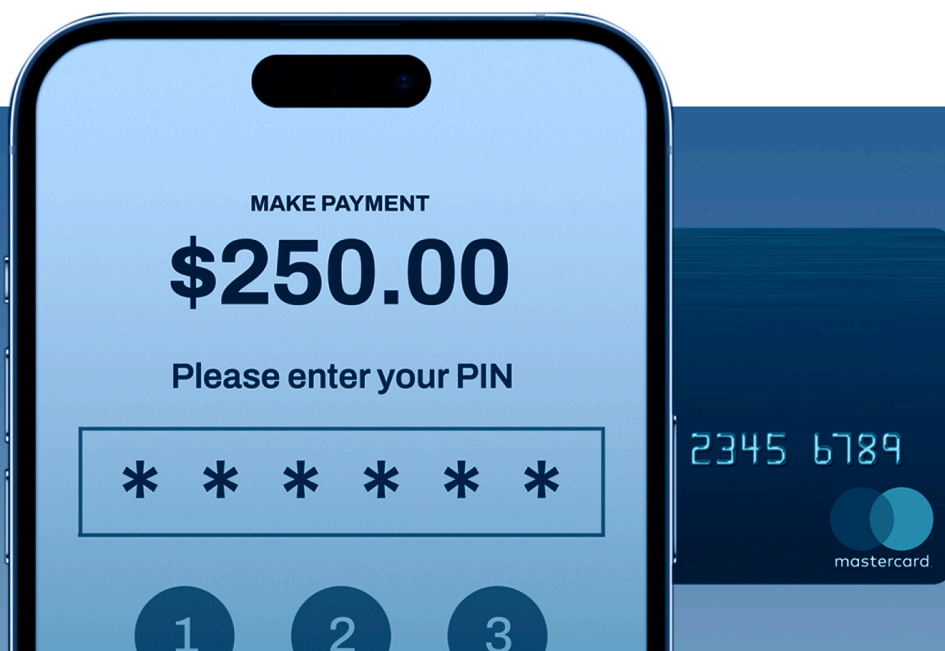
Secure PIN ensures that the PIN digits entered, the entire PIN, and the PIN encryption keys are never revealed in clear text. Secure PIN provides full support for advanced key management standards, including TR-31 and AES-compliant DUKPT, helping to ensure compatibility with payment industry standards.

Secure PIN secures PIN entry for SoftPOS solutions. This security is independent of the underlying COTS platform, meaning PIN data will remain secure even when the Android platform:

- is offline
- is outdated or not receiving security patches
- is not Google GMS (Google Mobile Service) certified, such as in the case of Android AOSP (often used on enterprise devices)

Some of the main requirements fulfilled by Secure PIN are:

- PIN digits and PIN encryption keys are never revealed in clear text
- PIN entry is aborted in the case of potential threats, such as a modified application, the PIN entry pad losing focus, or the detection of a debugger
- PIN entry does not use the system's keyboard or GUI elements
- full support for advanced key management schemes including TR-31 and AES-compliant DUKPT



MPoC COTS-Native PIN Entry Compliance

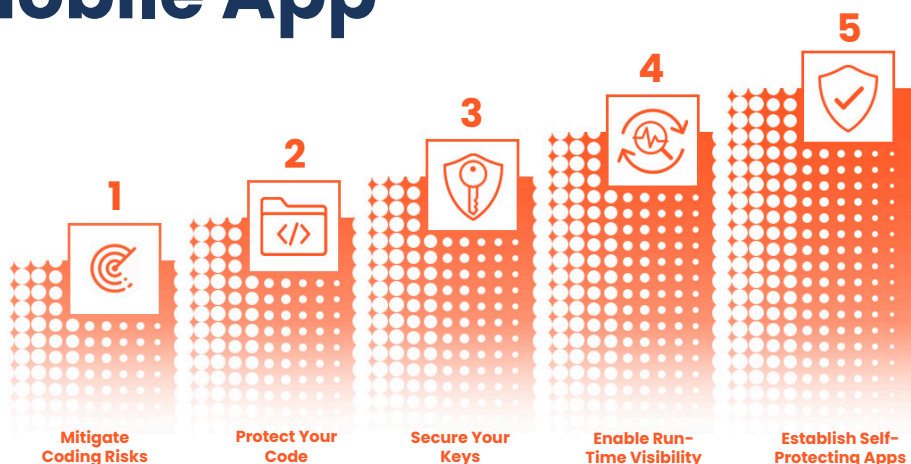
The MPoC standard defines security requirements, test requirements, and guidance for entities involved in the development, deployment, and operation of merchant-operated mobile payment acceptance solutions that allow the entry of cardholder PINs on COTS devices.

The table below demonstrates how Zimperium's solution measures up to these requirements.

| Security Requirements | Is Zimperium Compliant? |
|--|---------------------------|
| 1A-3.2 All cryptographic processes, including <i>hash</i> functions, used to provide security to the solution must adhere to <i>Appendix C Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms</i> . | Yes |
| 1A-3.4 Each key must have a single unique purpose, and no keys may be used for multiple purposes. | Yes |
| 1A-4.5 Cryptographic keys must be established using a process that ensures the entropy and confidentiality of the key. | Yes |
| 1A-4.6 The MPoC Software must support the use of HSMs for storage and operation of secret and private cryptographic keys in the back-end environments. | Yes |
| 1A-4.8 Cryptographic keys must not be protected with a key of lesser strength. | Yes |
| 1E-1.1 Documentation must exist that describes the secure capture and processing of the cardholder PIN. | Yes |
| 1E-1.2 PIN entry is supported only for chip-based transactions. | Integrator Responsibility |
| 1E-1.3 The MPoC SDK must not leak complete or partial PIN digits. The MPoC SDK must protect against side channels that use sensors present in the COTS device (e.g., accelerometers and gyroscopes) and screen capture. | Yes |
| 1E-1.4 The MPoC SDK must protect the PIN digits during entry. | Yes |
| 1E-1.5 The PIN must be encrypted into an ISO format 4-PIN block as soon as it is captured. | Yes |
| 1E-1.6 Attestation functions detecting indications of potential compromise must be executed prior to each PIN entry process. | Yes |
| 1E-1.7 The MPoC SDK must detect when another application overlays the MPoC SDK during PIN capture. In case of positive detection, the MPoC SDK must cancel any PIN entry currently in progress. | Yes |
| 1E-1.8 PIN-related data (PIN, PIN related values such as touch locations, PIN block, PIN key) must not be stored on the COTS device-persistent storage and must be erased once no longer required. | Yes |
| 1E-1.9 Offline PIN verification is supported only through the use of a PCI PTS SCRP. | Not Applicable |

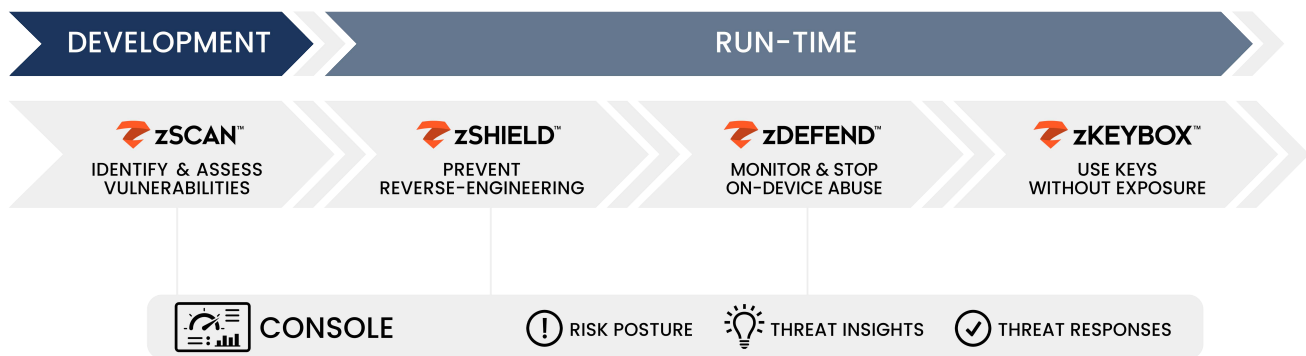
Top Level Requirements: Five Steps to Mobile App Security





By following these five steps, teams can attain the level of mobile application security maturity required in order to effectively secure their SoftPOS apps and the sensitive data and back-end services these apps access.



Solution Introduction: How Zimperium Enables Robust, PCI- Compliant Mobile App Security

With security being an essential aspect of SoftPOS solutions and PCI MPoC certification, **it is important to use proven mobile application security tools from solutions providers committed to the space.** Zimperium's Mobile Application Protection Suite (MAPS) enables mobile payment solution developers worldwide to quickly and efficiently develop secure and compliant mobile applications.



| Solutions | Value Proposition |
|---|--|
|  zSCAN™ | Helps organizations continuously discover and fix compliance, privacy, and security issues prior to being published. |
|  zSHIELD™ | Protects the source code, intellectual property (IP), and data from potential attacks like reverse engineering and code tampering. |
|  zDEFEND™ | Provides threat visibility and on-device ML-based run-time protection against device, network, phishing, and malware attacks. |
|  zKEYBOX™ | Protect your keys so they cannot be discovered, extracted, or manipulated. |