**Payconiq**

**SDK Security Guideline**

**Document Information**

|  |  |
| --- | --- |
| **Document ID** | PQI SEC-PRO-006 |
| **Classification** | Confidential |
| **Registered on** | 20-09-2018 |
| **Last Modified on** | 13-07-2023 |
| **Business Owner** | Jeffry Laan |
| **Author** | Tejas Bhosale |
| **Contact** | security@payconiq.com |
| **Version** | 2.0 |
| **Status** | Final |

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
| 20-09-2018 | 1.0 | Initial draft | Mustafa Kucukaytekin |
| 20-09-2018 | 1.1 | Added Android requirements | Andrey Kidinov |
| 20-09-2018 | 1.2 | Added iOS requirements | Andrea Tullis |
| 21-09-2018 | 1.3 | Added personal data, secure app requirements | Mustafa Kucukaytekin |
| 24-09-2018 | 1.4 |  | Remco Nabuurs |
| 24-09-2018 | 1.5 | Review | Herco Le Fevre |
| 18-08-2022 | 1.6 | Review | Nishtha Wadhawan |
| 20-09-2022 | 1.7 | Review | Nishtha Wadhawan |
| 13-07-2023 | 1.8 | Changed email of DPO | Jeffry Laan |
| 02-08-2023 | 1.9 | Campfire | Jeffry Laan |
| 08-08-2023 | 2.0 | PRC Approval | PRC |
| 16-08-2024 | 2.1 | Review | Michael Oki |

**Documented changes**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
| 16-08-2024 | 2.1 | Formatted the document using the new template and modified sections 3, 6.2, 7.2, 7.3, 7.4, 7.5, 8.1, 8.4, 8.12, 8.13 | Michael Oki |
|  |  |  |  |

**Table of Contents**

[1. Overview 4](#_Toc174724988)

[2. Purpose 4](#_Toc174724989)

[3. Scope 4](#_Toc174724990)

[4. Governance 4](#_Toc174724991)

[5. Personal Data 4](#_Toc174724992)

[6. Rules 5](#_Toc174724993)

[6.1. Rules & Recommendations 5](#_Toc174724994)

[6.2. Limitations 5](#_Toc174724995)

[6.3. SDK Tampering and Reverse Engineering 5](#_Toc174724996)

[6.4. Source Code Review and Penetration Test 5](#_Toc174724997)

[7. Recommendations 5](#_Toc174724998)

[7.1. Secure Memory Container 6](#_Toc174724999)

[7.2. Trusted Execution Environment 6](#_Toc174725000)

[7.3. Operating Systems 6](#_Toc174725001)

[7.4. Best Practices 6](#_Toc174725002)

[7.5. Cross-Site Request Forgery 6](#_Toc174725003)

[8. Technical Requirements 6](#_Toc174725004)

[8.1. Programming Languages 7](#_Toc174725005)

[8.2. Data Storage 7](#_Toc174725006)

[8.3. Data Modification 7](#_Toc174725007)

[8.4. Data Handling Rules 7](#_Toc174725008)

[8.5. PIN Policy 7](#_Toc174725009)

[8.6. Usage of not exposed functions 7](#_Toc174725010)

[8.7. Low-Level API 7](#_Toc174725011)

[8.8. Runtime Application Self Protection 8](#_Toc174725012)

[8.9. Keychain, Keystore, and Realm 8](#_Toc174725013)

[8.10. Obfuscation 8](#_Toc174725014)

[8.11. Jailbroken and Rooted States 8](#_Toc174725015)

[8.12. Network Security 8](#_Toc174725016)

[8.13. Biometric Sensors 8](#_Toc174725017)

[8.14. Registering Biometric Sensors 8](#_Toc174725018)

[8.15. Custom Keyboard 9](#_Toc174725019)

[8.16. Backup Restrictions 9](#_Toc174725020)

[8.17. Screenshot 9](#_Toc174725021)

[8.18. Running in the Background 9](#_Toc174725022)

[9. Definitions and Terms 9](#_Toc174725023)

# Overview

This document highlights the minimum security requirements for using the Payconiq SDK.

# 2. Purpose

The purpose of this document is to provide developers with a comprehensive set of instructions and best practices to secure their applications. It covers topics such as data protection, encryption, authentication, and code obfuscation, ensuring the safe and effective use of the SDK.

# 3. Scope

The security of a consumer that processes the information by the integrator’s mobile app that transfer the data with Payconiq backend systems via the Payconiq SDK is still Payconiq’s primary responsibility. Therefore, the integrator and their developers shall follow the recommendations, rules, and technical requirements in this document to protect the information services.

This security guideline is intended for Payconiq’s integrators and their developers allowing them to adapt their development process by following the requirements in this document.

# 4. Governance

* Security Team: The security team should ensure approve the SDK before it goes live
* Developers: Securely, integrate Payconiq’s SDK in their apps
* Integrators: Ensure their developers adhere to this guideline

# 5. Personal Data

The integrator shall follow the General Data Protection Regulation (GDPR) and other applicable local data protection law requirements when processing the PII, therefore the requirements of GDPR are the common goal of both Payconiq and the integrator. Payconiq doesn’t provide an exception if the integrator is not able to perform GDPR requirements.

The below is Payconiq’s Data Protection Officer contact information:

[Flora.goedefroit@payconiq.com](mailto:Flora.goedefroit@payconiq.com)

[privacy@payconiq.com](mailto:privacy@payconiq.com)

# 6. Rules

The following rules govern the usage of the SDK. Payconiq reserves the right to (temporary) block the usage of the SDK if an app or integrator are found or suspected to be in-compliant.

## 6.1. Rules & Recommendations

The integrator and app must comply with the rules and best practices as described in this document.

## 6.2. Limitations

Integrators are responsible for the correct usage of the SDK.

Abusing the SDK functionalities might result in the deactivation of the integrator’s account in cases such as:

* calling the functions in a loop
* fuzzing or injecting parameters
* performing a pentest on the SDK or Payconiq backend without prior approval

## 6.3. SDK Tampering and Reverse Engineering

Provided artifacts of the SDK shall not be modified or reverse engineered.

## 6.4. Source Code Review and Penetration Test

The integrator mobile app must be source code reviewed at least once a year, and the dynamic application security test shall be performed at least once a year. The test framework should be as minimum as defined in OWASP mobile application security framework. The report shall be shared with Payconiq upon request.

# 7. Recommendations

Recommendations are based on security best practices aim for helping integrators to protect the app. Applying the below requirements are subject to the integrator’s decision.

## 7.1. Secure Memory Container

Using the SDK, it is possible to access to personal data of the consumer. To prevent memory dump attacks, it is recommended to use a secure way to store those values in memory. Additionally, confidential information shall short-live in the memory and be overwritten when no longer needed.

## 7.2. Trusted Execution Environment

SDK leverages Secure Element and Secure Enclave usage when the hardware exists in the device, Payconiq recommends the integrator to build its application depending on this hardware.

Please note that Payconiq intends to require support for Secure Element / Secure Enclave in a future release of the SDK. Payconiq will inform the integrator about relevant timelines in 2024.

## 7.3. Operating Systems

This version of the SDK is designed to run on iOS 12+ and Android API 26 (Android 8.0+) to provide the integrators wide range of compatibility, however, Payconiq recommends the integrators to sunset old operating systems for their apps.

## 7.4. Best Practices

The developers should use the best practices ([Secure development policy](https://payconiq.sharepoint.com/:w:/g/it/EXa0SU9Qj4JKo8uLEEgST4cBirHkGVWWE-_LI3820hIMOw?e=gSPeyq), OWASP cheat sheets, etc.) for the implementation of the app to SDK. Each platform and language have specific risks, framework, practices, security functions that should be considered by the developer. i.e., avoid using system libraries which cannot be obfuscated.

## 7.5. Cross-Site Request Forgery

If the app uses an embedded display of web pages (WebView), CSRF and JS vulnerabilities should be taken into considerations.

# 8. Technical Requirements

The following technical requirements must be implemented by the integrator.

## 8.1. Programming Languages

For iOS development, Payconiq only supports Swift. Objective C is not supported by the SDK.

For Android development, Payconiq supports both Java and Kotlin. However, any new Android app should be developed in Kotlin as recommended by Google.

## 8.2. Data Storage

Payconiq related data entered by the consumer cannot be stored locally in the integrator mobile app, or any remote system other than Payconiq’s backend.

## 8.3. Data Modification

The data entered by the consumer cannot be modified/altered before passing it to SDK functions.

## 8.4. Data Handling Rules

Critical assets of the app, such as cryptographic materials or tokens shall be protected with strong encryption algorithms if the assets must be stored locally in the mobile device. Sensitive information shouldn’t be included in the stack trace.

## 8.5. PIN Policy

The SDK provides a secure memory container implementation which can be used by integrators. Passing PIN code to the SDK will always require secure memory container.

## 8.6. Usage of not exposed functions

The usage of not exposed functions is forbidden via reflection or via any other available mechanism.

Only classes and functions placed into packages:

* com.payconiq.sdk.commons
* com.payconiq.sdk.facade

or public methods or entities of PqSdk/PayconiqClient can be used.

## 8.7. Low-Level API

The SDK exposes high-level functions to deal with sensitive endpoints so secrets like encryption keys and tokens are secured.

Do NOT use low-level API to mimic any of the high-level API.

## 8.8. Runtime Application Self Protection

The app shall have runtime application self-protection measures implemented:

* Tampering detection/application signature check
* Debug detection
* SSL pinning
* Hook detection

## 8.9. Keychain, Keystore, and Realm

Any data stored locally by the SDK by any of the storage means must not be accessed or modified. For e.g., (KeyChain, UserDefaults (iOS), KeyStore or Realm (Android), etc.

## 8.10. Obfuscation

SDK users are recommended to utilize commercially available obfuscation tools like DexGuard for APKs and iXGuard for IPAs prior to final publication.

## 8.11. Jailbroken and Rooted States

SDK detects jailbroken or rooted state of the device, sends this information to Payconiq’s backend to mark user. This information will be used by Payconiq backend when processing API requests.

## 8.12. Network Security

In case the app communicates with non-Payconiq backends (i.e., the integrator or third parties), communication should be encrypted using TLS 1.2 at minimum. However, it is recommend to upgrade it to version 1.3. The exception to this requirement requires a written acceptance from Payconiq.

## 8.13. Biometric Sensors

Only OS-provided APIs shall be used for biometric sensors, such as touchID, faceID etc. Support for deprecated biometric APIs should be dropped due to potential security risks.

## 8.14. Registering Biometric Sensors

During the registration of a Biometric secret protected by biometric authentication the disclaimer message shall warn the user regarding pre-registered biometrics are authorised to perform payments.

The biometric secret is stored locally and encrypted by biometric authentication and shall be excluded from backups.

The biometric data must be deleted if a change in the biometry identity is detected. I.e., a new fingerprint is added.

## 8.15. Custom Keyboard

Custom keyboard shall be developed for PIN entry screen.

## 8.16. Backup Restrictions

The app shall not allow OS backup utilities to backup sensitive/confidential information. That information shall be marked accordingly in KeyChain, KeyStore.

## 8.17. Screenshot

Due to confidential information showed in the app screen, app shall restrict - only for Android- user to take the screenshot.

## 8.18. Running in the Background

If the app is sent to the background, all confidential information shall be removed from the view.

# 9. Definitions and Terms

Consumer or user is a Payconiq user who installed the integrator mobile app on his mobile device.

**Integrator** refers to the company that develops a mobile app which uses the Payconiq SDK.

**Integrator mobile app** or **app** refers to the integrator developed mobile app that is subject to the requirements of this document.

**The developer** refers to people who are in charge of implementing the Payconiq SDK on the integrator mobile app.

**SDK** or **Payconiq SDK** refers to Payconiq developed software development kit, which handles the network and security functions towards to Payconiq backend systems.

# 