Hardware Wallet Security Audit Report

Table Of Contents

1 Executive Summary

On 2024.10.21, the SlowMist security team received the OneKey team's security audit application for OneKey Pro, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "black box lead, grey box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the project in certain scenarios. It is suggested that the project team should evaluate and consider whether these vulnerabilities need to be fixed.
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.
Suggestion	There are better practices for coding or architecture.

2 Audit Methodology

The security audit process of SlowMist security team for hardware wallet includes two steps:

- The codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.
- Manual audit of the codes for security issues. The hardware wallets are manually analyzed to look for any potential issues.

The following is a list of security audit items considered during an audit:

- Firmware Security
    - Firmware storage security audit
    - Firmware upgrade security audit
    - Firmware integrity security audit
    - Firmware decompilation security audit
    - Firmware configuration security audit
- Storage Security
    - Data storage security audit
- Exception Handling Security
    - Error handling security audit
    - Exception logs security audit
- Permission Security
    - App permissions detection
- Communication Security
    - Communication encryption security audit
- Device Interface Security
    - Interface security audit
- Business Security
    - Business logic security audit

- Authentication Security
    - Device-Based authentication security audit
- Transfer Security
    - Signature security audit
    - Deposit/Transfer security audit
    - Transaction broadcast security audit
- Secret key Security
    - Secret key generation security audit
    - Secret key storage security audit
    - Secret key usage security audit
    - Secret key backup security audit
    - Secret key destruction security audit
    - Insecure entropy source audit
    - Cryptography security audit
- Components Security
    - Third-party components security audit
- User Interaction Security
    - WYSIWYS
    - Password complexity requirements

3 Project Overview

3.1 Project Introduction

Audit Version

1. firmware-pro

   https://github.com/OneKeyHQ/firmware-pro/releases/tag/v4.10.0

   commit: 6a387f588b29885b6adebd994df417be090c210c

2. thd89

   https://github.com/OneKeyHQ/THD89/releases/tag/v1.1.4

   commit: dc9ad611843e8116290c53ab0c77109cc61d6190

3. bluetooth-firmware-pro

   https://github.com/OneKeyHQ/bluetooth-firmware-pro/releases/tag/v2.3.2

   commit: be3739f73acf09af046fa36b12cc1293b60b5cd1

Fixed Version

1. firmware-pro

   https://github.com/OneKeyHQ/firmware-pro/releases/tag/v4.11.0

   commit: a969ccccb950c36bea613a65f28bad620753e99d

2. firmware:

   https://github.com/OneKeyHQ/firmware-pro/releases/download/v4.11.0/pro.4.11.0-Stable-1204-a969ccc_thd89_0x10_app-1.1.4-0911-dc9ad61_aligned.signed.bin

Note: OneKey Pro uses GitHub Action for firmware release to ensure consistency between firmware and code. The code commit for the fixed version and the file "pro.4.11.0-Stable-1204-a969ccc_thd89_0x10_app-1.1.4-0911-dc9ad61_aligned.signed.bin" are verifiable.

3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	Allow downgrade of firmware	Firmware upgrade security audit	Suggestion	Fixed
N2	Missing security self-check functionality configuration	Firmware configuration security audit	Suggestion	Acknowledged
N3	Communication is not encrypted and authenticated.	Communication encryption security audit	Low	Acknowledged

NO	Title	Category	Level	Status
N4	Bluetooth pairing does not require unlocking the wallet	Communication encryption security audit	Medium	Fixed
N5	The communication between the app and hardware wallet is not encrypted and authentication	Communication encryption security audit	Low	Acknowledged
N6	The pending sign data replacement issue	Signature security audit	Low	Acknowledged
N7	Password complexity issue	Device-Based authentication security audit	Suggestion	Acknowledged
N8	Lack of anti-tampering design	Others	Suggestion	Acknowledged

3.3 Vulnerability Summary

[N1] [Suggestion] Allow downgrade of firmware

Category: Firmware upgrade security audit

Content

Firmware upgrades allow users to downgrade to older firmware versions. If an older firmware version contains security vulnerabilities, an attacker could exploit these vulnerabilities by updating the device to the older version.

Solution

It is recommended to add a restriction to the firmware upgrade process that prevents users from downgrading to a firmware version that is older than the current version. This will help to prevent attackers from using older or vulnerable firmware to attack hardware wallets.

Status

Fixed

[N2] [Suggestion] Missing security self-check functionality configuration

Category: Firmware configuration security audit

Content

The lack of a security self-check function is a security risk. It is important to provide this type of function so that users can verify the security of their devices and firmware themselves. This helps to ensure that the device has not been maliciously modified and that the firmware is from an official source.

Solution

It is recommended to add a feature that allows users to self-check the security of their devices and firmware.

Status

Acknowledged

[N3] [Low] Communication is not encrypted and authenticated.

Category: Communication encryption security audit

Content

In the communication process of electronic components, if encryption and authentication operations are not performed, malicious electronic components may be implanted, and then data in the communication process can be sniffed and tampered with.

- The communication between the microcontroller unit (MCU) and the low-power Bluetooth (BLE) has not implemented encryption and authentication mechanisms.
- The communication between the microcontroller unit (MCU) and the fingerprint module (FP) has not undergone encryption and authentication processing.
- The communication between the microcontroller unit (MCU) and the camera module has not taken encryption and authentication measures.
- The communication between the microcontroller unit (MCU) and the touch panel has not taken encryption and authentication measures.

Solution

It is recommended that encrypted communication and an authentication mechanism should be implemented between the microcontroller unit (MCU), Bluetooth Low Energy (BLE), fingerprint recognition module (FP), camera module, and the Touch panel.

Status

Acknowledged

[N4] [Medium] Bluetooth pairing does not require unlocking the wallet

Category: Communication encryption security audit

Content

In the current design, the Bluetooth pairing process allows hardware wallets to be paired while locked. This may cause the hardware wallet to complete pairing with malicious devices without owner authentication, and then enable malicious devices to initiate signature requests to the hardware wallet.

Solution

It is recommended that when performing Bluetooth pairing, the wallet must be unlocked first before starting the pairing process.

Status

Fixed

[N5] [Low] The communication between the app and hardware wallet is not encrypted and authentication

Category: Communication encryption security audit

Content

The mobile phone APP and hardware wallet do not have additional encrypted communication. Therefore, on the paired mobile phone, in addition to the OneKey APP wallet being able to communicate with the OneKey hardware wallet, other apps on the mobile phone can also communicate with the hardware wallet and even obtain Bluetooth communication data.

When a desktop app communicates with a hardware wallet, it is not authenticated and the communication is not encrypted. This means that the user cannot distinguish which app is sending a signing request to the hardware wallet. Therefore, a malicious app can send a signing request to the hardware wallet when the user is using it, or perform a man-in-the-middle attack to trick the user into signing a malicious request, which could then be used to steal the user's cryptocurrency assets.

Solution

It is recommended that apps should be authenticated when interacting with hardware wallets. Unauthenticated apps should not be allowed to communicate with hardware wallets. Additionally, apps and hardware wallets should use

encrypted communication to avoid man-in-the-middle attacks.

Status

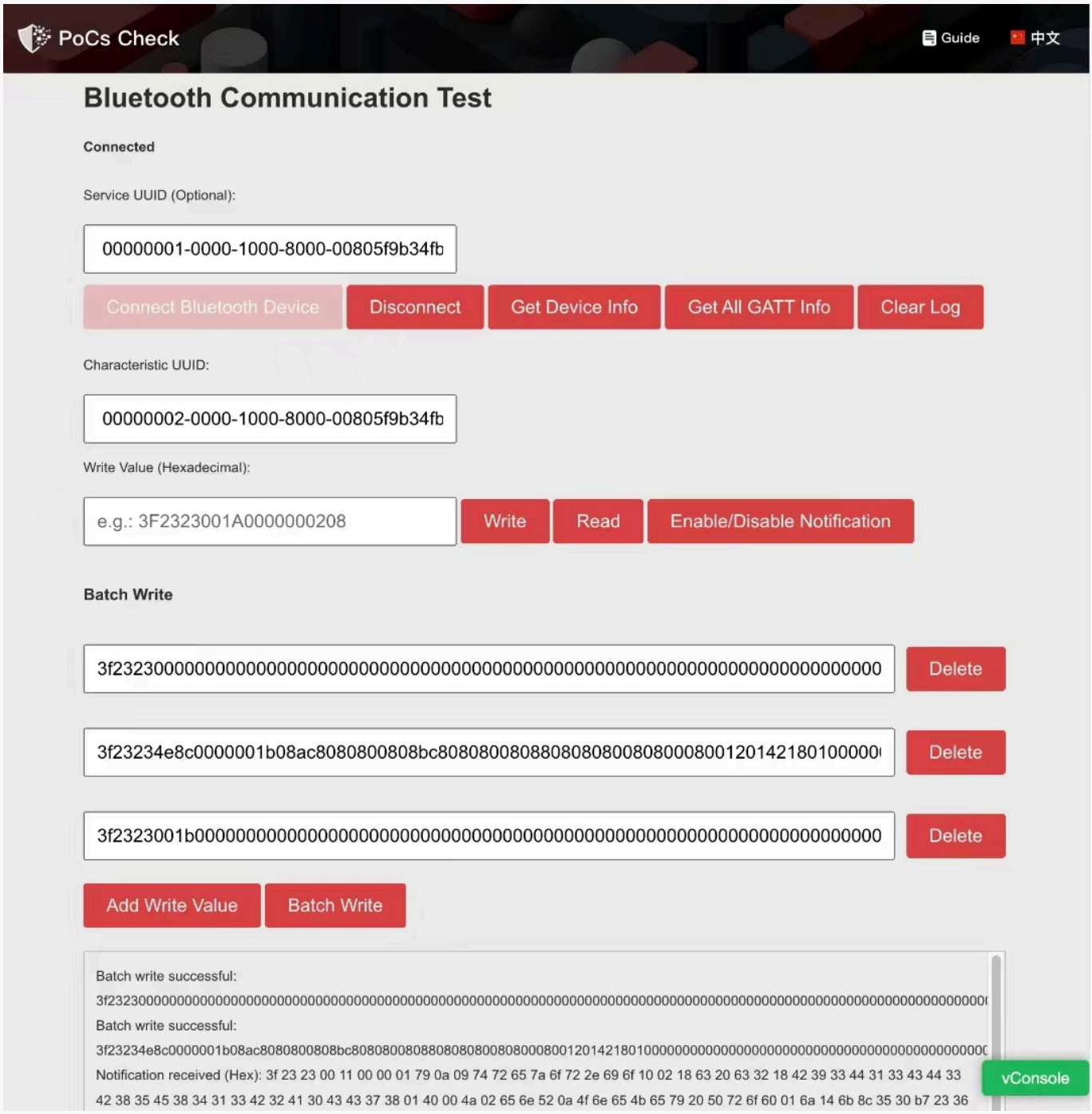
Acknowledged

[N6] [Low] The pending sign data replacement issue

Category: Signature security audit

Content

When sending a signature request to a hardware wallet via Bluetooth, when the previous signature request is not yet completed, a new signature request can be constructed to replace the old one. Malicious apps can take advantage of this issue to secretly replace the data to be signed. This kind of attack occurs in an extremely short time and it is generally difficult to identify whether the signature has been replaced.



Solution

It is recommended to add additional APP authentication, only accept signature requests from authenticated APPs, and do not receive new signature requests when the signature process is not completed. Ensure that a new signature process is started only after each signature process is completed by means of channels.

Status

Acknowledged

[N7] [Suggestion] Password complexity issue

Category: Device-Based authentication security audit

Content

The password supports 4-50 digit numeric passwords, which have a single character type. The password complexity can be improved by increasing the character type, such as requiring that the password contain letters and numbers, and the password length should be at least 8 digits.

Solution

It is recommended to increase the character type of passwords to include letters, and to require passwords to be at least 8 characters long.

Status

Acknowledged

[N8] [Suggestion] Lack of anti-tampering design

Category: Others

Content

Hardware wallets do not have tamper-proof and anti-disassembly designs. Attackers can influence the data acquisition results by modifying the acquisition modules of hardware wallets such as cameras, fingerprints, NFC, etc., and thereby implant backdoors. Attackers can also implant malicious modules in hardware wallets to influence the screen display of hardware wallets and deceive users' senses.

Solution

It is recommended to add a design to prevent disassembly and component tampering, so as to avoid attackers implanting malicious components in hardware wallets.

Status

Acknowledged

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002411100001	SlowMist Security Team	2024.10.21 - 2024.11.10	Low Risk

Summary conclusion: The SlowMist security team uses a manual and SlowMist team's analysis tool to audit the project. During the audit work, we found 1 medium-risk vulnerabilities 3 low-risk vulnerabilities and 4 suggestions.

5 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist