# Hardware Wallet Security Audit Report

# Table Of Contents

# 1 Executive Summary

On 2023.09.07, the SlowMist security team received the Account Labs team's security audit application for Keystone 3, 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**

Scope1:

https://github.com/KeystoneHQ/keystone3-bootloader

commit: ee2058af32848d5f7202696c6474023d131fa85e

Scope2:

https://github.com/KeystoneHQ/keystone3-firmware/

commit: 3fa054ec48a41b2cadc8a794e24198062e3d3811

The audit scope is as follows:

- /application
- /components
- /core
- /crypto
- /rust
- /tasks
- /ui

**Fixed Version**

https://github.com/KeystoneHQ/keystone3-firmware/

commit: 23f8f72427fbff74f2a154441e636fe0070e20ba

Keystone3_1.2.0.bin(SHA256): 8153dcf2391512ecd57a9fc1727a114d829a82f86fe8f94205616fff79208559

## 3.2 Vulnerability Information

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

| NO | Title | Category | Level | Status |
| --- | --- | --- | --- | --- |
| N1 | Firmware integrity detection enhanced | Firmware integrity security audit | Suggestion | Fixed |
| N2 | Flash Data leaks | Firmware decompilation security audit | Low | Fixed |
| N3 | Authentication enhancement | Device-Based authentication security audit | Suggestion | Fixed |
| N4 | Anti tamper bypass | Secret key destruction security audit | Low | Fixed |

| NO | Title | Category | Level | Status |
| --- | --- | --- | --- | --- |
| N5 | Suggestions for clearing RAM data | Secret key usage security audit | Suggestion | Fixed |

## 3.3 Vulnerability Summary

### [N1] [Suggestion] Firmware integrity detection enhanced

**Category: Firmware integrity security audit**

**Content**

The firmware uses signature verification when upgrading to ensure that the firmware is officially released, and also checks the crc32 of the file, but does not provide users with the ability to check the reliability and integrity of the firmware in daily use of the hardware wallet.

Code location: keystone3-bootloader/app/firmware_update.c

```
 printf("start to check file crc32\r\n");
    f_lseek( & fp, headSize);
    crcCalc = 0;

    sha256_init( & ctx);
    uint8_t content_hash[32];
    for (i = headSize; i < fileSize; i += readSize) {
        ret = f_read( & fp, & g_fileUnit, FILE_UNIT_SIZE, (UINT * ) & readSize);
        if (ret) {
            FatfsError((FRESULT) ret);
            bRet = false;
            break;
        }
        //printf("i=%d,readSize=%d\r\n", i, readSize);
        crcCalc = crc32_ieee(crcCalc, g_fileUnit, readSize);
        sha256_hash( & ctx, g_fileUnit, readSize);
    }
    sha256_done( & ctx, content_hash);
    PrintArray("hash content:", content_hash, 32);
    if (crcCalc != info -> crc32) {
        printf("crc err,crcCalc=0x%08X,info->crc32=0x%08X\r\n", crcCalc, info ->
crc32);
        bRet = false;
        break;
    }
```

Code location: keystone3-bootloader/app/firmware_update.c

```
#if (SIGNATURE_ENABLE == 1)
        printf("signature=%s\r\n", info->signature);
        if (strlen(info->signature) != 128) {
            printf("error signature=%s\r\n", info->signature);
            bRet = false;
            break;
        }
        // TODO: find this public key from firmware section.
        uint8_t publickey[65] = {0};
        GetUpdatePubKey(publickey);
        PrintArray("pubKey", publickey, 65);
        if (verify_frimware_signature(info->signature, content_hash, publickey) !=
true) {
            printf("signature check error\n");
            bRet = false;
            break;
        }
#endif
```

**Solution**

It is recommended to provide the user self-test function, so that users can choose self-test to verify the integrity of the device and firmware.

**Status**

Fixed

### [N2] [Low] Flash Data leaks

**Category: Firmware decompilation security audit**

**Content**

Extract and analyze Flash data of 25Q128H, and perform decompilation and string extraction analysis, Some information was found in Flash data, including S3 links, Google Drive links, firmware information, and wallet public key information, devSn and secretKey.

```
jhttps://keystone-g3-firmware.s3.eu-central-1.amazonaws.com/release/0.7.8/763/pillar.bin?response-content-disposition=attachment&
X-Amz-Security-Token=IQoJb3JpZ2luX2VjEMb%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDmFwLW5vcnRoZWFzdC0yIkYwRAIgMIZaORV6GFuiCPeiPMDbQ5w9LGaWnuemSXEieOK8cK4CIE8PlCcT5laiXv%2F
gx78d5Rwa9id%2F0vVMZ%2FGYrrfRtVDUKpIDCI%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEQARoMNjIzMTQ3NTUyOTk1IgyEh8cBvcRSYgLj70oq5gIseUXYa6HozL8yg%2Fx9DFVDQWJbRxss%2Fqlm5c%2F0b
Wnn6OGeOiWL96awI9aJ35imPqQtnUsJA14R1ji70d9vkrqXnyAmG7NW9B%2BA5zZwRGGREkyE0Y2fhWjkTYC52YA4sI4uImeBYUkWnuJH7zyV9lwTSjrpJmBmm4uUyTFz9KzSMnPZgdfBbyc8c%2F4h87YImfDT%2F
qFhubMTXYAxQIij77%2FmnVDxiRJFAEPDF428Y%2B81baOFsNQL1B3EOGb6MNUoDqyjskCkPBscLj%2Bpz2yPlhVs%2F4kdpsOlMD8B63gfjSm9Ce0W5NgRgq8%2B5jTFTAehFpepIw45qKBmizt4o25sRCQP%2FZM
qHj10enpwHWppcm0JayaOwOWGI0ufMPZEua1ehYI7%2FHXGxmNIYN6%2FcrMfHxW5tvQwfaH3gp0I1acaM0X9Ko%2BuZmCej%2FAWI1F62ToVdVvlrjPxOSOrDF7iYW5DPvpgpeokd%2B5BMLjWsqcGOrQCFWazbt5
Hu07wr%2B1HNdiLQDMBI%2FKIaIhjM2GHxUpJPaqyzZfojepoFVXQslnZKTdABoMVC3obwxGDfgYzx%2BhwlWx8ihLIqOzkFuxRG62mZfaqdxEIUdeSHTCTibLFM%2FF69rDFvcqRzutfaS2UHqlOr18DYlvoaktfh
2VCNJIc%2FeIQEK2%2F9VtSnLsc7cZ%2BDzwOjYgqeGRU%2FFfaXaaH04nFIEV%2BURwooY8nXUsxssN5wEjAa76D6G9pnaDDGIQHKFVwRlhZxMXMJGdR%2BO21uIP6QCME9OQJvw9qqRutC%2B5MVBkQPTUYeVPyo
PcL2tE5GYkVaATJis%2F4CIPFmvNtfSV0Rz2wv1kMtCepHYCeegpbO64JiU7RKes%2FIkuX30XqbkNMqVQw89JR7vHE4ToBoISuUZp%2BzlE%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&
X-Amz-Date=20230828T160017Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAZCFT5CDRZNZC6G2P%2F20230828%2Feu-central-1%2Fs3%2Faws4_request&
X-Amz-Signature=5b32b228dbd62186bc2ee2449de849047e0469e722c6a088484393ae42a5c875_
xhttps://s3.console.aws.amazon.com/s3/object/keystone-g3-firmware?region=eu-central-1&prefix=release/0.7.8/763/pillar.bin
This resource fork intentionally left blank
Mac OS X
ATTR;
com.apple.quarantine
com.apple.macl
%com.apple.metadata:kMDItemWhereFroms
0081;64e85869;Chrome;
~;6Ep
bplist00
https://drive.usercontent.google.com/download?id=1vnzPrXrRbHek3-ismoc23fhU1ZiFgDXa&export=download&authuser=0&confirm=t&uuid=32bd82f0-fe21-48d6-97d1-0bfef6ee985a&
at=APZUnTWZ3ANuM9W4aSWx6JpDOjjF:1692948584945
This resource fork intentionally left blank
HISWEDGD
```

```
https://drive.usercontent.google.com/download?id=1zArGbi9eyMJmJm-FJVqmOMwlFRgtgNfk&export=download&
authuser=0&confirm=t&uuid=3d6f933e-6242-43ae-890c-303d1cdde07a&
at=APZUnTX1DLMfRzWn6KSWeVnSz3S3:1693816291373
This resource fork intentionally left blank
{"mark":"~update!","fileSize":3650518,"originalFileSize":6934192,"crc32":2543445009,
"originalCrc32":995842499,"encode":1,"encodeUnit":16384,"encrypt":0,
"signature":"15bfe010005c86941e0ba786fee152c5eda0728d66009862fa83e307cc742c42aced50ec2722bdf09b09c8bd5a625
047e485c37f4ae4400ccceea1462a8c84ca"}
```

```
2Begin time:2023-09-05 04:40:33
End time:2023-09-05 04:40:36
{"version":"1.0.0","key":{"btc":{"value":"xpub6CVXws3TxFFqRsHUpXaYrNTgSr7XPy5nc3p4btwkMeYGSvx5LfCCDWLBZbGNT2D6L99CMJ6VmsVjN5M8VNd2YfMBkWmZP9914Vxa38fGPUB",
"current":0},"btc_legacy":{"value":"xpub6CWzuomCz1vnWXHZYwQ8Bm8XAje2dUC7mXG99BxxzRkzfFiNMGsJxDWzU82bHtx8xjJoKArGvSj2HoRaLAcwrAdY1s7XgRJuQbjqetV7ws6","current":0},
"btc_nested_segwit":{"value":"xpub6BycH9ewxwHCjSoyX5ULCRXkWs59J61oJrxD1Q3KK2XgJw3sCFmCJCKP1Jv9bad2Uf45nNxjmzpGE1jjCzxEmprA9dasckhsBUYd3n2CUL","current":0},"ltc":
{"value":"xpub6CLD966gpa1rKy8tLetEw2FwwqkADuAYDTHqu4XYmBMg2Dw92sRFp6h8iYhTDYYj4ZmsPQ8wjcatHrEsjFirhF6PXMapEsM9cmiVr5MC8tR","current":0},"dash":
{"value":"xpub6D1n5Z6mKF1Fpx325wQsgvoj5F58ZteYKNKFKgE748wV3tBurHWSJ3oGLCdHjHVwzyDCGMqDUHR5cmA2wZ37pST1enDipv2Tk87GKzVsKPh","current":0},"bch":
{"value":"xpub6DTrYpqXyVhDZZMLeKEsatRPB5pHYvFveMkMmj6Md2cxGsyjmZvqQqSERK8Lfi6SidF4ptEpMGbPtTJTxdqpviNLMbLriRCgrbpdSiZCC79","current":0},"eth_bip44_standard":
{"value":"xpub6CRX7UArqJTaLzmsJj2PPzgmNAgKwyjc49tFeEpm2TVYbiKMVCCngP6Za9Zoyhfk5wBwuiYLG4SqWwhP9G9sS1Pforbqwrvy38LEs1yC7Q5","current":0},"eth_ledger_legacy":
{"value":"xpub6CRX7UArqJTaLzmsJj2PPzgmNAgKwyjc49tFeEpm2TVYbiKMVCCngP6Za9Zoyhfk5wBwuiYLG4SqWwhP9G9sS1Pforbqwrvy38LEs1yC7Q5","current":0},"eth_ledger_live_0":
{"value":"xpub6CRX7UArqJTaLzmsJj2PPzgmNAgKwyjc49tFeEpm2TVYbiKMVCCngP6Za9Zoyhfk5wBwuiYLG4SqWwhP9G9sS1Pforbqwrvy38LEs1yC7Q5","current":0},"eth_ledger_live_1":
{"value":"xpub6CRX7UArqJTaRxqpCycywFaYzonme7GYJw8hV4pcFTeyqSjJn6aZ6jcf2RkAYfdw48eRVbiWWjMSxMHn381is3UPzqJw1Y9mcDUT3LghjKe","current":0},"eth_ledger_live_2":
{"value":"xpub6CRX7UArqJTaTmXnVfmA5PiiBTEkx7oiCRCT1Mai4N4birSHBz4hmtPCHxCyqcV2R3fNhEaLZ89C791huHjo5VFJ8dCdGWXa8nryJiVDpx8","current":0},"eth_ledger_live_3":
{"value":"xpub6CRX7UArqJTaViq5yrj1aSdaD2gDCKyY22xcs96pMpGTYchPnLHGwQJFfE1Nc1r3fTFUJrpydmkVWSSoTiWk2V5JvuxY1cqNKY5XJERSnQX","current":0},"eth_ledger_live_4":
{"value":"xpub6CRX7UArqJTaZRpuvtuJjz9vMEnoqAdnMBT3bT8URcaqPN7AhRG9FcUce5s9MtxX3YLxTYtKgpbZjhCpefnoQ6QDge9Y6A3tatVbKx6rZ3h","current":0},"eth_ledger_live_5":
{"value":"xpub6CRX7UArqJTaa4QCPmPJUDYk6q25F82osGYQisNDEQxR7mEmzY9f5AAn3RGT2MU8ASq66GGx4xbPxRoV9NaDDMSpi4984MFDHsicMiXe2DH","current":0},"eth_ledger_live_6":
{"value":"xpub6CRX7UArqJTaeLEeR5pEemf4iwpbEvPJ4k8zfrK75o7XfwkpPvbXGKefdcbPSkJT3PWbBKw81mYzkWJ3v3FJqv2nGGfXR2VxScvYvfFjCZv","current":0},"eth_ledger_live_7":
{"value":"xpub6CRX7UArqJTagmfb78bTsA4S8Y9ty9DLTZ1FykVpvMFfbmCtMoqG5Bih6JfiNp8GkHMnK1QXJBKRYHB4RptcTCFv3Uvz83Sdsm8eksjPr8k","current":0},"eth_ledger_live_8":
{"value":"xpub6CRX7UArqJTaid611kRqsxdVicbNEKAHr2f9io4DPm3xnzYBHwhZpLHc5xSTh53dVwhTza6esUvfo1ob1vYDMifPbZZmpdx99paRibXSppG","current":0},"eth_ledger_live_9":
{"value":"xpub6CRX7UArqJTajtScSty8Gdq27AYgED5PRKq1vg53AbVNde46wnu8vUcDxbNjXVEkcTpcwtnkwW4yTPhHjLvJEPqVpJdNk5s1sS1DrydqpYZ","current":0},"trx":
{"value":"xpub6CDDY67Rg7yWcWrxbUXGZUSjUYrKD71dSxPWxfqsJhqFkuaxWNMhJcJ5yEYSv6mZAbyS3bbaBkhcGxovyAyBE1oB21KXKRh4pcYQkRmAE8H","current":0}}}
{"BTC":{"firstRecv":true,"manage":true},"ETH":{"firstRecv":true,"manage":true},"BNB":{"firstRecv":true,"manage":false},"SOL":{"firstRecv":true,"manage":false},
"DOT":{"firstRecv":true,"manage":false},"XRP":{"firstRecv":true,"manage":false},"LTC":{"firstRecv":true,"manage":false},"DASH":{"firstRecv":true,"manage":false},
"BCH":{"firstRecv":true,"manage":false},"TRX":{"firstRecv":true,"manage":false}}
e0/.-,+*)('&%$#"!  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/./.-,+*))('&%$%&'()*
+,-.-,+*)('&%$#"!
{"version":"1.0.0","setup_step":5,"bright":70,"auto_lock_screen":60,"auto_power_off":0,"vibration":1,"dark_mode":1,"usb_switch":1}
  56{"devSn":"keystone_1234","secretKey":"ASDFGHJKL"}
```

**Solution**

It is recommended that the project team check this information and ensure that no data is retained after burning, and that as far as possible, no useless data is stored in the official burn

**Status**

Fixed

## [N3] [Suggestion] Authentication enhancement

**Category: Device-Based authentication security audit**

**Content**

Keystone 3 supports Pin authentication, Password authentication and fingerprint authentication, Pin authentication and Password authentication, and will not put decryption elements in the device.Fingerprint authentication will put all the elements of decryption in the device.

**Solution**

It is recommended to use fingerprint authentication as a temporary design. Turn off fingerprint authentication every time after locking the wallet, and temporarily turn on fingerprint authentication by entering Pin or Password after unlocking.

**Status**

Fixed; After communication, the project team ensures the user experience and avoids all decryption elements from being stored in the wallet for a long time by adding the need to re-verify the password after restarting.

### [N4] [Low] Anti tamper bypass

**Category: Secret key destruction security audit**

**Content**

Keystone 3 is designed with the anti_tamper mechanism. If you disassemble Keystone 3 for analysis, the anti_tamper mechanism will be triggered to delete the data of the wallet.

The deletion operation requires the support of power supply, so the Keystone 3 built-in button battery, but the button battery is the usual operating temperature of -40 ° C to 125 ° C, can be heated by the heat gun 300 ° C for 28s, the button battery will be damaged, resulting in the anti_tamper mechanism cannot work.

Code location: keystone3-firmware/application/anti_tamper.c

```
static void TamperEraseInfo(void)
{
    uint8_t pageData[32];
    //In order to ensure that the current of the button cell can make the device erase
secrets,
    //the redundant peripherals are turned off.
    DisableAllHardware();
    Uart0OpenPort();
    Atecc608bInit();
```

```
    memset(pageData, TAMPER_MARK, 32);
    Atecc608bEncryptWrite(15, 0, pageData);
    DS28S60_Init();
    CLEAR_ARRAY(pageData);
    for (uint32_t i = 0; i < 36; i++) {
        printf("erase index=%d\n", i);
        DS28S60_HmacEncryptWrite(pageData, i);
    }
    printf("erase index=88\n");
    DS28S60_HmacEncryptWrite(pageData, 88);
    printf("erase over\n");
}
```

**Solution**

It is recommended to use fusible cut-out to redesign the circuit. When the change of voltage or temperature exceeds the threshold, fusible cut-out can trigger anti tamper.

**Status**

Fixed

### [N5] [Suggestion] Suggestions for clearing RAM data

**Category: Secret key usage security audit**

**Content**

KeyStone3 stores the mnemonic data in SRAM_MALLOC when importing the wallet, but does not clear the SRAM data when the wallet is locked, so you can see that the wallet is locked when importing the wallet, and then re-unlocked, and the previously entered mnemonic words will be displayed on the screen.

The data in SRAM was not cleared in time when the wallet was locked. that may suffer a Cold-Boot Attack in some scenarios.

Please refer to this paper: https://eprint.iacr.org/2016/769.pdf

Code location: keystone3-firmware/ui/gui_widgets/gui_mnemonic_input.c

```
void ImportSinglePhraseWords(MnemonicKeyBoard_t *mkb, KeyBoard_t *letterKb)
{
    char *mnemonic = SRAM_MALLOC(BIP39_MAX_WORD_LEN * mkb->wordCnt + 1);
    memset(mnemonic, 0, BIP39_MAX_WORD_LEN * mkb->wordCnt + 1);

    for (int i = 0, j = 0; i < mkb->wordCnt; j++, i += 3) {
```

```
        for (int k = i; k < i + 3; k++) {
            char trueBuf[12] = {0};
            GuiMnemonicGetTrueWord(lv_btnmatrix_get_btn_text(mkb->btnm, k), trueBuf);
            strcat(mnemonic, trueBuf);
            strcat(mnemonic, " ");
        }
    }
    mnemonic[strlen(mnemonic) - 1] = '\0';

    SecretCacheSetMnemonic(mnemonic);
```

The same issue also appears in the following code location, when setting the fingerprint, the wallet will not empty the RAM after the lock, resulting in the key and other information retained in the RAM.

Code location: keystone3-firmware/ui/gui_widgets/gui_setting_widgets.c

```
case DEVICE_SETTING_FINGERPRINT_PASSCODE:
        tile = lv_tileview_add_tile(g_deviceSetTileView.tileView, currentTile, 0,
LV_DIR_HOR);
        GuiWalletSetFingerPassCodeWidget(tile);
        strcpy(midLabel, _("fingerprint_passcode_mid_btn"));
        break;

    // finger module
    case DEVICE_SETTING_FINGER_MANAGER:
        tile = lv_tileview_add_tile(g_deviceSetTileView.tileView, currentTile, 0,
LV_DIR_HOR);
        GuiWalletFingerManagerWidget(tile);
        strcpy(midLabel, "Fingerprint Settings");
        structureCb = GuiFingerMangerStructureCb;
        destructCb = GuiFingerManagerDestruct;
        break;
    case DEVICE_SETTING_FINGER_ADD_ENTER:
        tile = lv_tileview_add_tile(g_deviceSetTileView.tileView, currentTile, 0,
LV_DIR_HOR);
        g_fingerAddTile = tile;
        leftBtn = NVS_BAR_CLOSE;
        leftCb = CancelCurFingerHandler;
        GuiWalletFingerAddWidget(tile);
        destructCb = GuiFingerAddDestruct;
        break;
```

Code location: keystone3-firmware/ui/gui_widgets/gui_setting_widgets.c

```
void GuiFingerManagerDestruct(void *obj, void *param)
{
    if (g_fpRecognizeTimer != NULL) {
        lv_timer_del(g_fpRecognizeTimer);
        g_fpRecognizeTimer = NULL;
    }
    ClearSecretCache();
}
```

**Solution**

It is recommended that sensitive data in RAM should be cleared immediately after the wallet is locked.

**Status**

Fixed; After communication, the creation and import of wallets are sensitive operations, and users should complete the entire creation or import process in a safe environment. Therefore, the project team reminds users that they need to complete the whole process in a safe environment when creating or importing to avoid data retention in SRAM.

# 4 Audit Result

| Audit Number | Audit Team | Audit Date | Audit Result |
| --- | --- | --- | --- |
| 0X002309260001 | SlowMist Security Team | 2023.09.07 - 2023.09.26 | Passed |

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 2 low risk vulnerabilities and 3 suggestions. All findings have been fixed.

# 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