Wallet Application Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.03.15, the SlowMist security team received the Hana team's security audit application for Hana wallet android, 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/grey box lead, white 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 wallet application 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 wallet application is manually analyzed to look for any potential issues.

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

NO.	Audit Items	Result
1	App runtime environment detection	Fixed
2	Code decompilation detection	Fixed
3	App permissions detection	Passed
4	File storage security audit	Passed
5	Communication encryption security audit	Fixed
6	Interface security audit	Passed
7	Business security audit	Passed
8	WebKit security audit	Passed
9	App cache security audit	Passed
10	WebView DOM security audit	Fixed
11	SQLite storage security audit	Passed
12	Deeplinks security audit	Passed
13	Client-Based Authentication Security audit	Fixed
14	Signature security audit	Fixed
15	Deposit/Transfer security audit	Passed
16	Transaction broadcast security audit	Passed

NO.	Audit Items	Result
17	Secret key generation security audit	Fixed
18	Secret key storage security audit	Passed
19	Secret key usage security audit	Passed
20	Secret key backup security audit	Fixed
21	Secret key destruction security audit	Passed
22	Screenshot/screen recording detection	Confirmed
23	Paste copy detection	Fixed
24	Keyboard keystroke cache detection	Fixed
25	Background obfuscation detection	Fixed
26	Suspend evoke security audit	Passed
27	AML anti-money laundering security policy detection	Passed
28	Others	Fixed
29	User interaction security	Confirmed

3 Project Overview

3.1 Project Introduction

Audit Version

https://github.com/Hana-Technology/hana-app

commit:2c53c6d260cf46e821e4b5afe8af03303cad5696

hana-app.apk v1.0.36 (sha256:9e4d618ba88122baf4f1e28e89bf9be250b58a1c742b44de1b2ce05baa404c83)

Fixed Version

https://github.com/Hana-Technology/hana-app commit: a45f4f2fd881e376d6811d4ac585c61363bcc3c4

hana-app.apk v1.0.45 (39) (sha256:a1b50f70e76f9140a8758dc6e257633c76d2a0db8d94f40292e1089679186610)

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Runtime environment detection issues	App runtime environment detection	Low	Fixed
N2	Decompilation security issues	Code decompilation detection	Low	Fixed
N3	usesCleartextTraffic configuration enhancement	Communication encryption security audit	Suggestion	Fixed
N4	Missing screenshot/screen recording detection	Screenshot/screen recording detection	Suggestion	Confirmed
N5	Lack of security reminders	Paste copy detection	Suggestion	Fixed
N6	Lack of secure keyboard	Keyboard keystroke cache detection	Suggestion	Fixed
N7	Background obfuscation issue	Background obfuscation detection	Suggestion	Fixed
N8	Blind signing lacks security reminder	Signature security audit	Low	Fixed
N9	URL validation can be bypassed	WebView DOM security audit	Low	Fixed
N10	Client-Based Authentication issue	Client-Based Authentication Security audit	Low	Fixed
N11	Log leak mnemonic and password	Secret key generation security audit	High	Fixed
N12	Secret key backup issue	Secret key backup security audit	Medium	Fixed

NO	Title	Category	Level	Status
N13	The wallet address is not fully displayed	Others	Suggestion	Fixed
N14	User interaction issue	User interaction security	Suggestion	Confirmed

3.3 Vulnerability Summary

[N1] [Low] Runtime environment detection issues

Category: App runtime environment detection

Content

The wallet does not detect whether the device is rooted, and lacks simulator environment detection, there is also no detection of the developer mode of the phone. When the developer mode is enabled on the mobile phone, the operating environment may be at risk. The application of the mobile phone can be debugged through the developer mode.

Solution

It is recommended that the wallet add root detection, emulator detection, and developer mode detection to remind users that the current environment is not safe.

Status

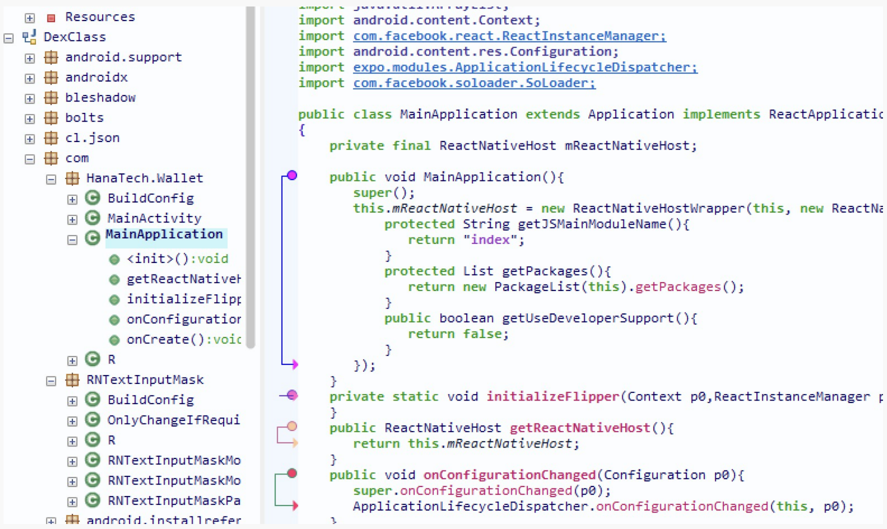
Fixed

[N2] [Low] Decompilation security issues

Category: Code decompilation detection

Content

The App does not obfuscate the code, and the Java code can be obtained by decompiling.



```
Resources
DexClass
  android.support
  androidx
  bleshadow
  bolts
  cl.json
  com
    HanaTech.Wallet
      BuildConfig
      MainActivity
      MainApplication
        <init>():void
        getReactNativeH
        initializeFlipp
        onConfiguratior
        onCreate():voic
      R
    RNTextInputMask
      BuildConfig
      OnlyChangeIfRequi
      R
      RNTextInputMaskMo
      RNTextInputMaskMo
      RNTextInputMaskPa
    android.installrefer
```

```
import android.content.Context;
import com.facebook.react.ReactInstanceManager;
import android.content.res.Configuration;
import expo.modules.ApplicationLifecycleDispatcher;
import com.facebook.soloader.SoLoader;

public class MainApplication extends Application implements ReactApplicatic
{
    private final ReactNativeHost mReactNativeHost;

    public void MainApplication(){
       super();
       this.mReactNativeHost = new ReactNativeHostWrapper(this, new ReactNa
           protected String getJSMainModuleName(){
              return "index";
           }
           protected List getPackages(){
              return new PackageList(this).getPackages();
           }
           public boolean getUseDeveloperSupport(){
              return false;
           }
       });
    }
    private static void initializeFlipper(Context p0,ReactInstanceManager p
    }
    public ReactNativeHost getReactNativeHost(){
       return this.mReactNativeHost;
    }
    public void onConfigurationChanged(Configuration p0){
       super.onConfigurationChanged(p0);
       ApplicationLifecycleDispatcher.onConfigurationChanged(this, p0);
```

Solution

It is recommended to harden the App, obfuscate the java code.

Status

Fixed

[N3] [Suggestion] usesCleartextTraffic configuration enhancement

Category: Communication encryption security audit

Content

usesCleartextTraffic is set to true to allow communication using HTTP.

- android/app/src/main/AndroidManifest.xml#L18

```
<application android:name=".MainApplication" android:label="@string/app_name"
android:icon="@mipmap/ic_launcher" android:roundIcon="@mipmap/ic_launcher_round"
android:allowBackup="false" android:theme="@style/AppTheme"
android:usesCleartextTraffic="true">
```

Solution

It is recommended to set usesCleartextTraffic to false to only allow communication using HTTPS.

Status

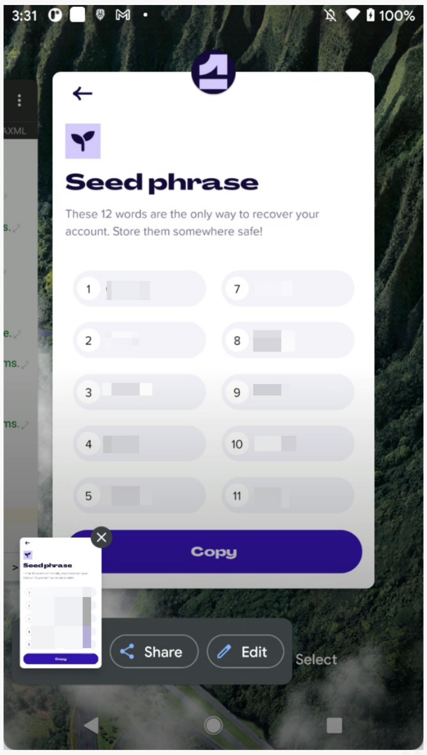
Fixed

[N4] [Suggestion] Missing screenshot/screen recording detection

Category: Screenshot/screen recording detection

Content

The APP does not have reminders for screenshots, and there are no restrictions on users taking screenshots and recordings.



Seed phrase

These 12 words are the only way to recover your account. Store them somewhere safe!

1 [redacted]	7 [redacted]
2 [redacted]	8 [redacted]
3 [redacted]	9 [redacted]
4 [redacted]	10 [redacted]
5 [redacted]	11 [redacted]

Solution

It is recommended to add screenshot/screen recording detection and prohibit screenshot/screen recording.

Status

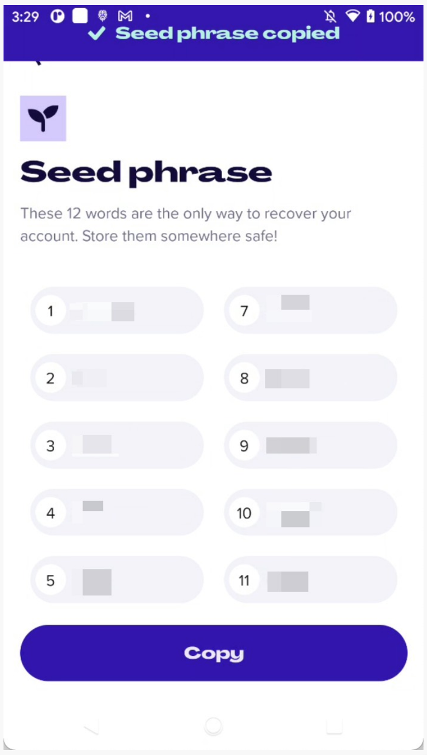
Confirmed

[N5] [Suggestion] Lack of security reminders

Category: Paste copy detection

Content

When exporting wallets, users are allowed to copy mnemonic phrases and the app lacks security reminders, which may be subject to clipboard hijacking attacks.


3:29
100%
Seed phrase copied
Seed phrase
These 12 words are the only way to recover your account. Store them somewhere safe!
1
7
2
8
3
9
4
10
5
11
Copy

Solution

It is recommended to remind users that they should record by transcribing instead of directly using the clipboard for copying.

Status

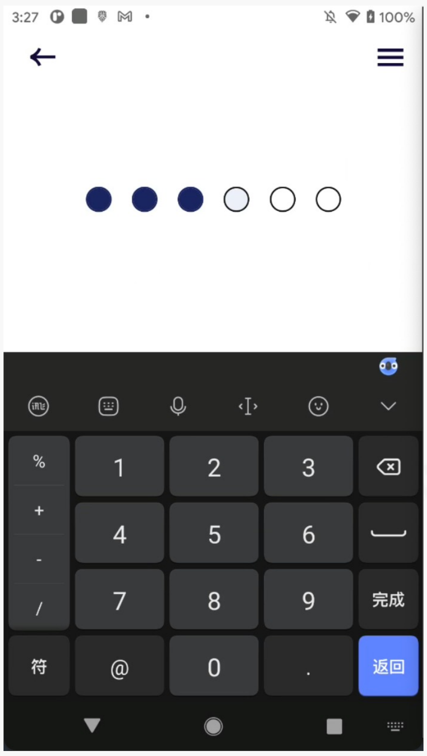
Fixed

[N6] [Suggestion] Lack of secure keyboard

Category: Keyboard keystroke cache detection

Content

The app does not use a secure keyboard, mnemonics and passwords may be stolen by the keyboard when using the app.



Solution

It is recommended to add a secure keyboard and use the secure keyboard when entering mnemonics and passwords to avoid sensitive data being recorded.

Status

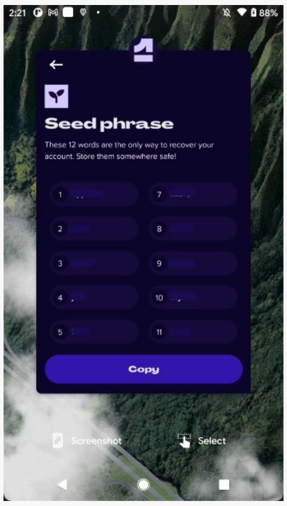
Fixed

[N7] [Suggestion] Background obfuscation issue

Category: Background obfuscation detection

Content

App UI is not obfuscation when the app is in the background.If the wallet is being exported, the mnemonic phrase may be leaked.


2:21 88%

Seed phrase

These 12 words are the only way to recover your account. Store them somewhere safe!

1 7
2 8
3 9
4 10
5 11

Solution

It is recommended to add an obfuscation mechanism to avoid sensitive data leakage.

Status

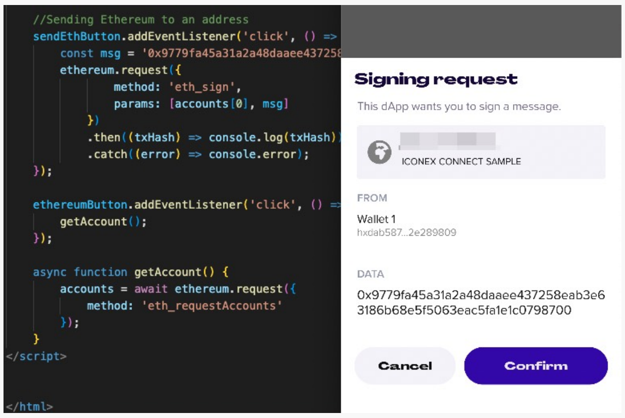
Fixed

[N8] [Low] Blind signing lacks security reminder

Category: Signature security audit

Content

When using eth_sign for the blind signature test, the wallet does not provide a security reminder for the blind signature, and the user may be at risk of being phished.


```
    //Sending Ethereum to an address
    sendEthButton.addEventListener('click', () =>
        const msg = '0x9779fa45a31a2a48daaee437258
        ethereum.request({
                method: 'eth_sign',
                params: [accounts[0], msg]
            })
            .then((txHash) => console.log(txHash)
            .catch((error) => console.error);
    });

    ethereumButton.addEventListener('click', () =>
        getAccount();
    });

    async function getAccount() {
        accounts = await ethereum.request({
            method: 'eth_requestAccounts'
        });
    }
</script>


</html>
```

Signing request

This dApp wants you to sign a message.

ICONEX CONNECT SAMPLE

FROM

Wallet 1
hxdab587...2e289809

DATA

0x9779fa45a31a2a48daaee437258eab3e63186b68e5f5063eac5fa1e1c0798700

Solution

It is recommended to detect blind signing and add security reminders.

Status

Fixed

[N9] [Low] URL validation can be bypassed

Category: WebView DOM security audit

Content

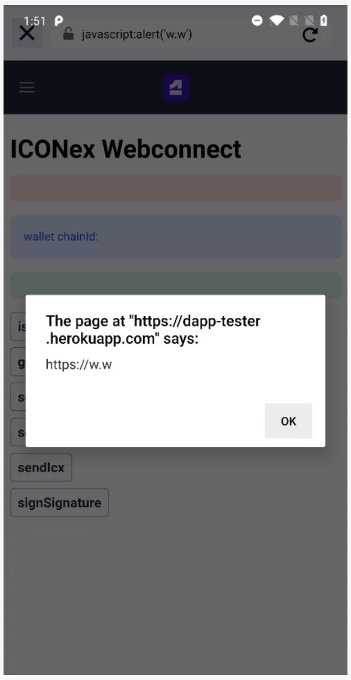
The URL verification is not perfect enough, so it can be accessed through WebView such as "javascript:alert('https://w.w')", "javascript://www.x.com/%0aalert(1)", resulting in abnormal Expected code execution.

- src/screens/browser/Browser.tsx#L334-346

```
function handleChangeUrl() {
  setIsKeyboardOpen(false);
  const hasProtocol = urlInput.includes('://');
  const isUrl = hasProtocol || urlInput.includes('.');

  const uri = isUrl
    ? `${!hasProtocol ? 'https://' : ''}${urlInput}`
    : `https://duckduckgo.com/?q=${urlInput}`;

  setUri(uri);
  setUrlInput(vanityUrl(uri));
  setHttpsUrl(isHttps(uri));
}
```

- src/screens/browser/ICXDApps.tsx#L52-61

```
function handleChangeUrl() {
  const hasProtocol = urlInput.includes('://');
  const isUrl = hasProtocol || urlInput.includes('.');

  const uri = isUrl
    ? `${!hasProtocol ? 'https://' : ''}${urlInput}`
    : `https://duckduckgo.com/?q=${urlInput}`;

  browseTo(uri);
}
```



The page at "https://dapp-tester .herokuapp.com" says:
https://w.w

Solution

It is recommended to enhance the URL matching rules to allow only HTTPS protocol access.

Status

Fixed

[N10] [Low] Client-Based Authentication issue

Category: Client-Based Authentication Security audit

Content

The wallet application is suspended in the background of the phone for a period of time, and the wallet will not be locked.

Solution

It is recommended that after the wallet is suspended in the background for a period of time, re-authentication is required to re-enter the wallet.

Status

Fixed

[N11] [High] Log leak mnemonic and password

Category: Secret key generation security audit

Content

Note: This issue was discovered during development/build and does not exist in the live version.

When the wallet is creating a password, creating a mnemonic, and changing the password, the password and mnemonic will be output in the log, which will be read by other apps.

- src/screens/onboarding/CreatePasscode.new.tsx#L128-129

```
const handleContinue = async (passcode: string, confirmPasscode: string) => {
    console.log('passcode new', passcode);
    console.log('confirmPasscode new', confirmPasscode);
```

- src/screens/onboarding/CreatePasscode.new.tsx#L155-156

```
const seedPhrase = await createVault({ passcode });
console.log('seedPhrase', seedPhrase);
```

- src/screens/onboarding/CreatePasscode.new.tsx#L187-188

```
  console.log('passcode', passcode);
  console.log('confirmPasscode', confirmPasscode);
```

Solution

It is recommended that sensitive information such as passwords and mnemonics be prohibited from being output to the log.

Status

Fixed

[N12] [Medium] Secret key backup issue

Category: Secret key backup security audit

Content

The pbkdf2 hash is stored locally and the wallet has all the ingredients to unlock the secret.

- src/stores/Vault.ts#L133-146

```
async function verifyCredentials(credentials: AuthenticateCredentials) {
  if (credentials.useBiometrics) {
    return true;
  }

  const { encryptionKey } = useInternalState.getState();

  const hashedPasscode = await hash(credentials.passcode!, encryptionKey!);
  const storedPasscode = await SecureStore.getItemAsync(
    STORAGE_KEY.PASSCODE_HASH,
    SecureStoreOptions
  );
  return hashedPasscode === storedPasscode;
}
```

- src/stores/Vault.ts#L608-619

```
async function createVault(credentials: CreateVaultCredentials) {
  const { hasVault } = useVault.getState();

  if (hasVault()) {
    throw new Error('You already have a vault.');
  }

  const encryptionKey = await generateRandomKey();
  await SecureStore.setItemAsync(STORAGE_KEY.ENCRYPTION_KEY, encryptionKey,
SecureStoreOptions);

  const hashedPasscode = await hash(credentials.passcode, encryptionKey);
  await SecureStore.setItemAsync(STORAGE_KEY.PASSCODE_HASH, hashedPasscode,
SecureStoreOptions);
```

Solution

It is recommended not to store the encryptionKey directly in the secureStore, and use PBKDF2 to derive the encryptionKey from the password.

It is recommended to only store encryptionKey when biometrics are enabled.

Status

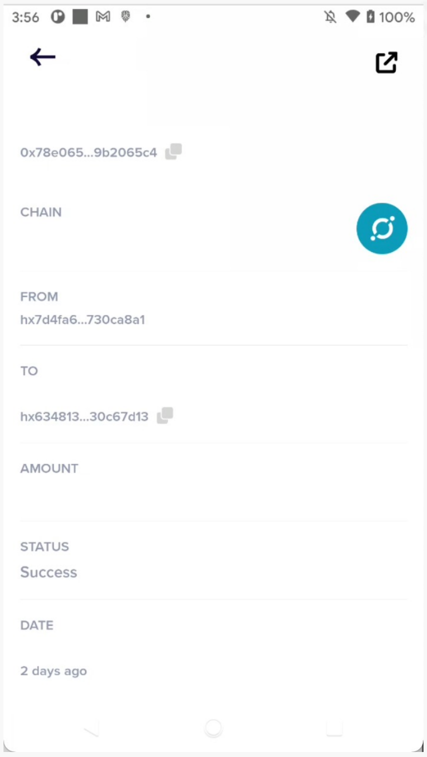
Fixed

[N13] [Suggestion] The wallet address is not fully displayed

Category: Others

Content

The wallet does not fully display the transaction address, and users need to enter the blockchain browser to view the transaction details to know the complete transaction address.This can easily be used for phishing using similar addresses.


3:56 100%

0x78e065...9b2065c4

CHAIN

FROM
hx7d4fa6...730ca8a1

TO
hx634813...30c67d13

AMOUNT

STATUS
Success

DATE
2 days ago

Solution

It is recommended that the wallet provide the function of fully displaying the transaction address to avoid phishing to deceive the transaction address.

Status

Fixed

[N14] [Suggestion] User interaction issue

Category: User interaction security

Content

Functionality	Support	Notes
WYSIWYS	●	There is no friendly parsing of the data.
AML	✗	AML strategy is not supported.
Anti-phishing	✗	Phishing detect warning is not supported.
Pre-execution	✗	Pre-execution result display is not supported.
Contact whitelisting	●	The contact whitelisting is not supported, causing similar address attacks.
Password complexity requirements	✓	The password is complexity.

Tip: ✓ Full support, ● Partial support, ✗ No support

Solution

It is recommended to enhance user interaction security.

Status

Confirmed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002303310002	SlowMist Security Team	2023.03.15 - 2023.03.31	Passed

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 high risk, 1 medium risk, 5 low risks, and 7 suggestion vulnerabilities. And 1 high risk, 1 medium risk, 5 low risks, 5 suggestion vulnerabilities were confirmed and being fixed; We extend our gratitude for Hana Wallet team recognition of SlowMist and hard work and support of relevant staff.

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.