Wallet Application Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.10.09, the SlowMist security team received the Bitizen team's security audit application for Bitizen 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.

Level	Description
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	Passed
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	Fixed
7	Business security audit	Passed
8	WebKit security audit	Passed
9	App cache security audit	Passed
10	WebView DOM security audit	Passed

NO.	Audit Items	Result
11	SQLite storage security audit	Passed
12	Deeplinks security audit	Fixed
13	Client-Based Authentication Security audit	Passed
14	Signature security audit	Passed
15	Deposit/Transfer security audit	Passed
16	Transaction broadcast security audit	Passed
17	Secret key generation security audit	Passed
18	Secret key storage security audit	Passed
19	Secret key usage security audit	Passed
20	Secret key backup security audit	Passed
21	Secret key destruction security audit	Passed
22	Screenshot/screen recording detection	Confirmed
23	Paste copy detection	Passed
24	Keyboard keystroke cache detection	Confirmed
25	Background obfuscation detection	Confirmed
26	Suspend evoke security audit	Passed
27	AML anti-money laundering security policy detection	Passed
28	Others	Fixed

3 Project Overview

3.1 Project Introduction

Audit Version

Android V1.2.0：https://play.google.com/store/apps/details?id=org.bitizen.wallet

SHA256:425ebc899a2720d5b24beb2ff0e3d3062b310e48d944f0cc314bae133a8b8f39

Fixed Version

Android V1.2.4 (The dev apk package provided by the Bitizen team)

SHA256:77955e6b6c76f18a851b650890ff0c212fba746ec814e1ec627d6e8fa644162d

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	APK unpacked and hardened	Code decompilation detection	Suggestion	Fixed
N2	usesCleartextTraffic is not closed	Communication encryption security audit	Low	Fixed
N3	Deeplink bypass authentication	Deeplinks security audit	High	Fixed
N4	Not using a secure keyboard	Keyboard keystroke cache detection	Suggestion	Confirmed
N5	Background hang without obfuscation	Background obfuscation detection	Suggestion	Confirmed
N6	No screen recording and screenshot reminders	Screenshot/screen recording detection	Suggestion	Confirmed
N7	API interface parameters can be enumerated.	Interface security audit	Low	Fixed

NO	Title	Category	Level	Status
N8	The wallet address is not fully displayed	Others	Low	Fixed

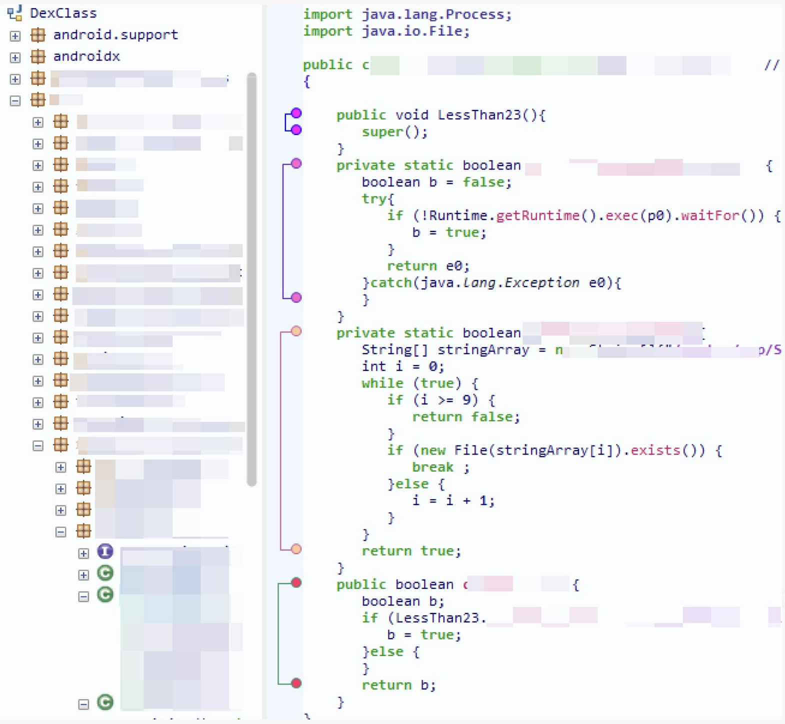
3.3 Vulnerability Summary

[N1] [Suggestion] APK unpacked and hardened

Category: Code decompilation detection

Content

The APK is not packaged for protection and code is not obfuscated.



DexClass
android.support
androidx
[redacted]

```
import java.lang.Process;
import java.io.File;

public c[redacted]     //
{

    public void LessThan23(){
       super();
    }
    private static boolean [redacted]    {
       boolean b = false;
       try{
          if (!Runtime.getRuntime().exec(p0).waitFor()) {
             b = true;
          }
          return e0;
       }catch(java.lang.Exception e0){
       }
    }
    private static boolean [redacted]
       String[] stringArray = n[redacted]p/S
       int i = 0;
       while (true) {
          if (i >= 9) {
             return false;
          }
          if (new File(stringArray[i]).exists()) {
             break ;
          }else {
             i = i + 1;
          }
       }
       return true;
    }
    public boolean c[redacted]   {
       boolean b;
       if (LessThan23.[redacted]
          b = true;
       }else {
       }
       return b;
    }
}
```

Solution

Reinforce and pack the App to avoid being decompiled. The code is obfuscated to increase the difficulty of decompilation and reading.

Status

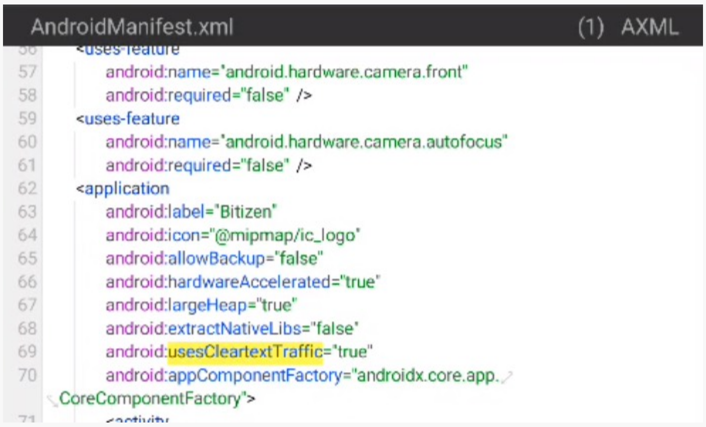
Fixed

[N2] [Low] usesCleartextTraffic is not closed

Category: Communication encryption security audit

Content

The AndroidManifest file is not configured with android:usesCleartextTraffic , and can only communicate via HTTPS when set to false.



```
<uses-feature
    android:name="android.hardware.camera.front"
    android:required="false" />
<uses-feature
    android:name="android.hardware.camera.autofocus"
    android:required="false" />
<application
    android:label="Bitizen"
    android:icon="@mipmap/ic_logo"
    android:allowBackup="false"
    android:hardwareAccelerated="true"
    android:largeHeap="true"
    android:extractNativeLibs="false"
    android:usesCleartextTraffic="true"
    android:appComponentFactory="androidx.core.app.CoreComponentFactory">
    <activity
```

Solution

Set android:usesCleartextTraffic to false.

Status

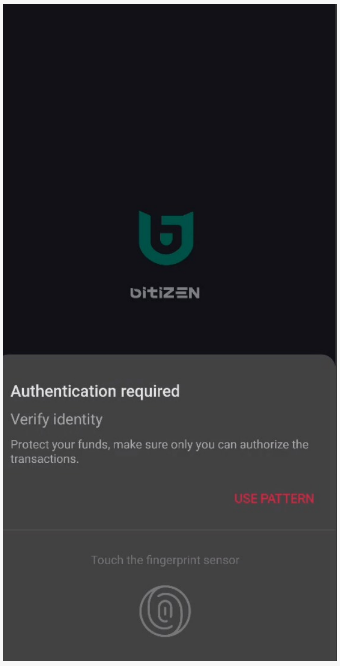
Fixed

[N3] [High] Deeplink bypass authentication

Category: Deeplinks security audit

Content

Using deeplink https://bitizen.org/wallet/wc or bitizen://wallet/wc, two requests can bypass the login authentication of the front desk.



Authentication required

Verify identity

Protect your funds, make sure only you can authorize the transactions.

Solution

When deeplink evokes an App, it should first check whether the current App is unlocked.

Status

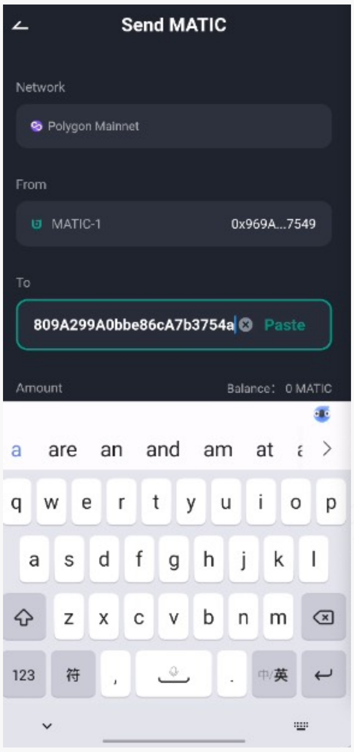
Fixed

[N4] [Suggestion] Not using a secure keyboard

Category: Keyboard keystroke cache detection

Content

Third-party input methods will collect user input, which may lead to leakage of sensitive information.



Send MATIC

Network

Polygon Mainnet

From

MATIC-1 0x969A...7549

To

809A299A0bbe86cA7b3754a Paste

Amount Balance: 0 MATIC

Solution

It is recommended to add a secure keyboard to the app.

Status

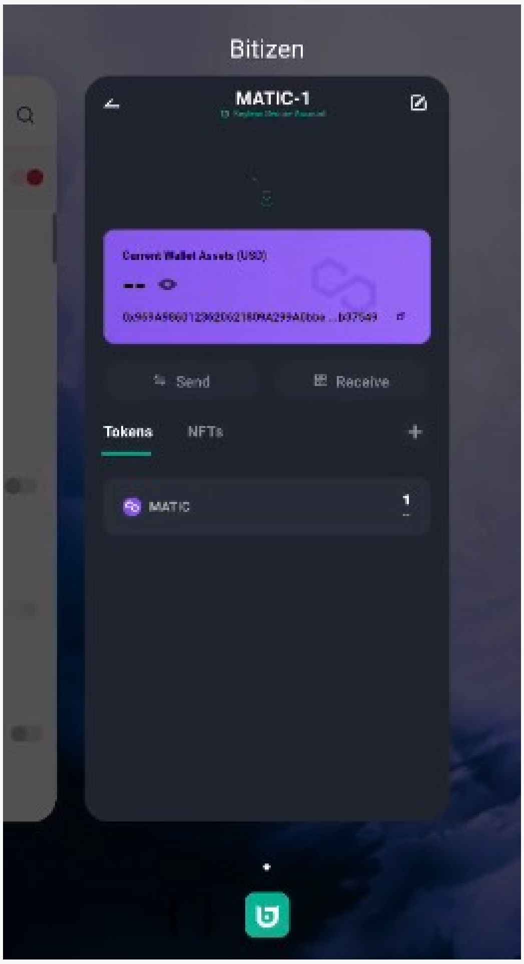
Confirmed

[N5] [Suggestion] Background hang without obfuscation

Category: Background obfuscation detection

Content

After hanging the wallet in the background, the page of the wallet needs to be obfuscated. Avoid leaking sensitive data when switching apps when the wallet is in a sensitive interface.



Solution

After hanging the wallet in the background, the page of the wallet needs to be obfuscated.

Status

Confirmed

[N6] [Suggestion] No screen recording and screenshot reminders

Category: Screenshot/screen recording detection

Content

No screen recording and screenshot reminders

Solution

It is recommended to add a reminder to record or take screenshots.

Status

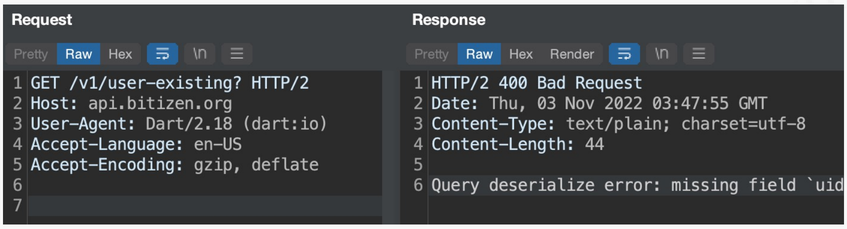
Confirmed

[N7] [Low] API interface parameters can be enumerated.

Category: Interface security audit

Content

The server will prompt the client's api request which parameters are missing and the specific parameter types, which has potential security risks.



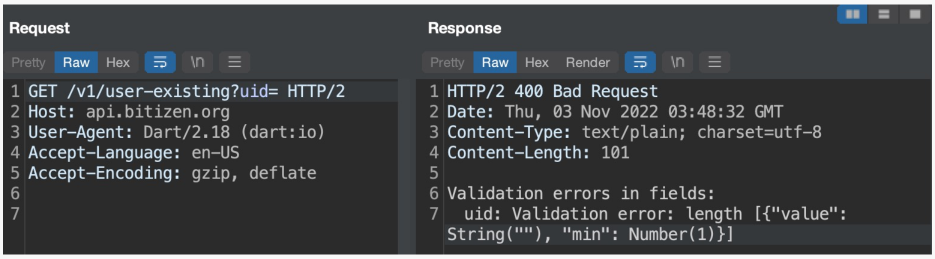
```
GET /v1/user-existing? HTTP/2
Host: api.bitizen.org
User-Agent: Dart/2.18 (dart:io)
Accept-Language: en-US
Accept-Encoding: gzip, deflate

```

```
HTTP/2 400 Bad Request
Date: Thu, 03 Nov 2022 03:47:55 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 44

Query deserialize error: missing field `uid
```

When the client only submits parameters, the specific types of the parameters are exposed.



```
GET /v1/user-existing?uid= HTTP/2
Host: api.bitizen.org
User-Agent: Dart/2.18 (dart:io)
Accept-Language: en-US
Accept-Encoding: gzip, deflate

```

```
HTTP/2 400 Bad Request
Date: Thu, 03 Nov 2022 03:48:32 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 101

Validation errors in fields:
  uid: Validation error: length [{"value":
String(""), "min": Number(1)}]
```

Solution

It is recommended to standardize the error reminders on the server side to avoid the possibility of information leakage.

Status

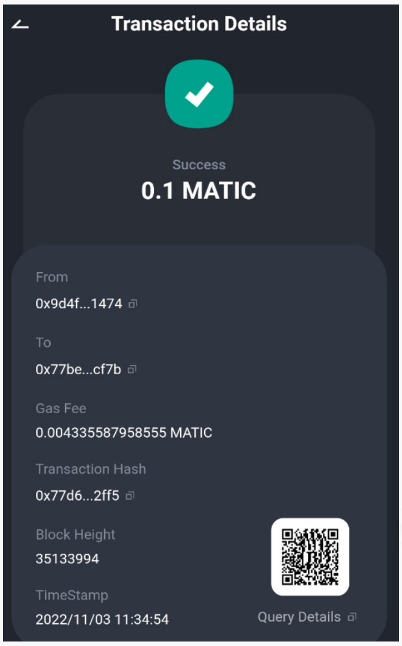
Fixed

[N8] [Low] 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.



Transaction Details

Success

0.1 MATIC

From

0x9d4f...1474

To

0x77be...cf7b

Gas Fee

0.004335587958555 MATIC

Transaction Hash

0x77d6...2ff5

Block Height

35133994

TimeStamp

2022/11/03 11:34:54

Solution

It is recommended that the wallet fully display the functions of the transaction address as much as possible to avoid phishing to deceive the transaction address.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002210270002	SlowMist Security Team	2022.10.09 - 2022.10.27	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, 3 low risk, 4 suggestion vulnerabilities. And 3 suggestion were confirmed,all other findings were fixed.We extend our gratitude for Bitizen 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.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist