// Private Mobile Pentest

02.26.2024 - 03.12.2024

Mobile App
Rarimo

HALBORN

Mobile App - Rarimo

Prepared by: HALBORN

Last Updated 05/06/2024

Date of Engagement by: February 26th, 2024 - March 12th, 2024

Summary

100% ⓘ OF ALL REPORTED FINDINGS HAVE BEEN ADDRESSED

ALL FINDINGS	CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
4	0	1	0	3	0

TABLE OF CONTENTS

1. Introduction

The team at Halborn was provided 2 weeks for the engagement and assigned two full-time security engineers to verify the security of the assets in scope. The security engineer is a penetration testing expert with advanced knowledge in web, mobile Android and iOS, recon, discovery & infrastructure penetration testing.

The goals of our security assessments are to improve the quality of the systems we review and to target sufficient remediation to help protect users.

2. Assessment Summary

In summary, Halborn identified several minor security issues. These include the lack of protective measures like jailbreak detection, and anti-hook and anti-debug mechanisms. Additionally, the possibility of multiple voting instances using a deactivated passport was recognized as a potential security risk. However, no sensitive data has been found stored on the device or transmitted to the remote backend server. Furthermore, bypassing SSL pinning was deemed impossible. Implementing defense-in-depth strategies such as SSL pinning is crucial for enhancing app resilience against reverse engineering and certain client-side attacks. Given that SSL pinning could not be disabled, the interception of communications by malicious actors is unlikely, unless they engage in reverse engineering to deactivate the SSL pinning in the app.

In order to figure out which requests were made, the Halborn team used cryptography hooks instrumentation with Frida which allowed to monitor while the voting process is made, the endpoints requests sent to remote server backend. Those requests were made without any sensitive data sent:

- https://issuer.polygon.robotornot.mainnet-beta.rarimo.com/v1/credentials/%7BuserDid%7D/%7BchemaID%7D - is used to get claim offers in order to retrieve verifiable credentials.
- https://rpc-api.node1.mainnet-beta.rarimo.com/rarimo/rarimo-core/identity/state/%7BIssuerIdHex%7D - is used to get current issuer state.
- https://api.stage.freedomtool.org/integrations/proof-verification-relayer/v1/verify-proof - is used to send raw EVM calldata that will be executed on a registration contract.
- https://api.mock.freedomtool.org/integrations/identity-provider-service/v1/gist-data - is used to get GIST data from rarimo, in nearly future we will get rid of this endpoint.

3. Test Approach And Methodology

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the pentest. While manual testing is recommended to uncover flaws in logic, process and implementation; automated testing techniques assist enhance coverage of the infrastructure and can quickly identify flaws in it.

The following phases and associated tools were used throughout the term of the assessment:

- Storing private keys and assets securely
- Send/Receive tokens and assets securely to another wallet
- Any attack that impacts funds, such as draining or manipulating of funds
- Application Logic Flaws
- Areas where insufficient validation allows for hostile input
- Application of cryptography to protect secrets
- Brute Force Attempts
- Input Handling
- Source code review
- Fuzzing of all input parameters
- Technology stack-specific vulnerabilities and Code Assessment
- Known vulnerabilities in 3rd party / OSS dependencies.

4. RISK METHODOLOGY

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the LIKELIHOOD of a security incident and the IMPACT should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

- **10** - CRITICAL
- **9 - 8** - HIGH
- **7 - 6** - MEDIUM
- **5 - 4** - LOW
- **3 - 1** - VERY LOW AND INFORMATIONAL

5. SCOPE

FILES AND REPOSITORY

(a) Repository: FreedomTool

(b) Assessed Commit ID: eda342d

(c) Items in scope:

Out-of-Scope:

REMEDIATION COMMIT ID:

- FreedomFreedom

Out-of-Scope: New features/implementations after the remediation commit IDs.

6. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	1	0	3	0

IMPACT X LIKELIHOOD

			HAL-06	
	HAL-03 HAL-01 HAL-04			

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-06 - USERS WITH DEACTIVATED PASSPORTS CAN VOTE MORE THAN 1 TIME	High	RISK ACCEPTED - 03/04/2024
HAL-03 - LACK OF JAILBREAK DETECTION	Low	ACKNOWLEDGED
HAL-01 - LACK OF FINGERPRINT OR PASSCODE AUTHENTICATION	Low	SOLVED - 05/05/2024
HAL-04 - LACK OF ANTI-HOOK ANTI-DEBUG MECHANISM	Low	ACKNOWLEDGED

7. FINDINGS & TECH DETAILS

7.1 (HAL-06) USERS WITH DEACTIVATED PASSPORTS CAN VOTE MORE THAN 1 TIME

// HIGH

Description

When a person loses their password, they will go to the authorities to get a new passport. The old one will get deactivated, and the new one will be used. However, the application does not check if the passport is invalidated, meaning that one person can vote more than 1 time. It is true that to do that, the person needs to get a new passport and find the old one, since to get a new one when lost the old one must be reported as lost to the authorities.

Proof of Concept

Scenario:

- A person has two passports.
- He obtained the first password nXXXXXX and lost it.
- Then, he went to get a new one. When he did that, the new passport will have a text with a note saying “the passport nYYYYY is available due to the loss of the nXXXXX”, but this is written on the new passport (first page, above the profile), and the old one gets “deactivated” on Police servers, which means that if he goes to an airport using that first deactivated passport, and they check the number or read the NFC it will show as deactivated. However, since the application does this validation offline, it does not have this kind of verification.

Score

Impact: 5

Likelihood: 4

Recommendation

In order to solve the issue, the passport ID should be checked against a Police BBDD to check if the passport is or is not deactivated.

Remediation Plan

RISK ACCEPTED: The **Rarimo team** is aware of this issue, but currently there is no correct way to fix it without reviling the user data to the backend.

7.2 (HAL-03) LACK OF JAILBREAK DETECTION

// LOW

Description

Anti-jailbreak mechanisms are not used in the iOS application. These mechanisms can help mitigate reverse engineering, application modification, and unauthorized versions of mobile applications to some extent, but few if any will be completely successful against a determined adversary. However, they can be used as part of a defense-in-depth strategy that seeks to minimize the impact and likelihood of such an attack, along with binary patching, local resource modification, method hooking, method swizzling, and heap modification.

Proof of Concept

```
[                -LT76-MBP ~ % frida-ps -Uia
  PID  Name           Identifier
----  -------------  --------------------------
1870  Filza          com.tigisoftware.Filza
1816  FreedomTool    FreedomTool.FreedomTool
1871  Settings       com.apple.Preferences
1869  Sileo          org.coolstar.SileoStore
1868  palera1n       com.samiiau.loader
   -  App Store      com.apple.AppStore
   -  Books          com.apple.iBooks
   -  Calculator     com.apple.calculator
   -  Calendar       com.apple.mobilecal
   -  Camera         com.apple.camera
   -  Clock          com.apple.mobiletimer
   -  Compass        com.apple.compass
   -  Contacts       com.apple.MobileAddressBook
   -  Exodus         exodus-movement.exodus
   -  FaceTime       com.apple.facetime
   -  Files          com.apple.DocumentsApp
   -  Find My        com.apple.findmy
   -  Fitness        com.apple.Fitness
   -  Health         com.apple.Health
   -  Home           com.apple.Home
   -  Magnifier      com.apple.Magnifier
   -  Mail           com.apple.mobilemail
   -  Maps           com.apple.Maps
   -  Measure        com.apple.measure
   -  Messages       com.apple.MobileSMS
   -  Music          com.apple.Music
   -  Notes          com.apple.mobilenotes
   -  Phone          com.apple.mobilephone
   -  Photos         com.apple.mobileslideshow
   -  Podcasts       com.apple.podcasts
   -  Reminders      com.apple.reminders
   -  Safari         com.apple.mobilesafari
   -  Shortcuts      com.apple.shortcuts
   -  Stocks         com.apple.stocks
   -  TV             com.apple.tv
   -  Tips           com.apple.tips
   -  Translate      com.apple.Translate
   -  Voice Memos    com.apple.VoiceMemos
```

```
-  Wallet          com.apple.Passbook
-  Watch           com.apple.Bridge
-  Weather         com.apple.weather
-  iTunes Store    com.apple.MobileStore
```

Score

Impact: 2

Likelihood: 2

Recommendation

The application should not allow any modifications in its operation.

- Obfuscated code.
- Add Frida and other open-source jailbreak detection tools.
- Implement jailbreak detection.

Remediation Plan

ACKNOWLEDGED: The **Rarimo team** acknowledged the issue.

Remediation Hash

https://github.com/rarimo/FreedomTool

7.3 (HAL-01) LACK OF FINGERPRINT OR PASSCODE AUTHENTICATION

// LOW

Description

During the assessment, it was observed that there was no authentication mechanism defined on the application. Even though the device has a passcode enabled or fingerprint, but it was observed that the application did not implement any security mechanism.
If a user used fingerprint or passcode on the mobile device, there is no authentication mechanism in place to protect the application and anybody who has access to the device can access the application as there will be no authentication against these. Since the likelihood is low, the severity has been downgraded.

Score

Impact: 2

Likelihood: 2

Recommendation

Ensure proper authentication implementation in order to avoid potential unauthorized attacks.

Remediation Plan

SOLVED: The **Rarimo team** solved the issue.

Remediation Hash

https://github.com/rarimo/FreedomTool

7.4 (HAL-04) LACK OF ANTI-HOOK ANTI-DEBUG MECHANISM

// LOW

Description

The tested application does not have any security features or mechanisms to prevent malicious actions, Anti Hook and Anti Debug.

Proof of Concept

- Install Frida on the jailbroken phone. Setup Jailbroken Device
- Use the Objection Tool to investigate the Anti-Hook mechanisms in the application. Objection
- Use the following command in the objection tool to investigate the Jailbroken device. `objection --gadget "<package name>" explore`
- Run the following code in the objection. `ios nsuserdefaults get`

```
[redacted]-MBP ~ % objection --gadget "FreedomTool.FreedomTool" explore
Using USB device `iPhone`
Agent injected and responds ok!

     _   _         _   _
 ___| |_|_|___ ___| |_|_|___ ___
| . | . | |  _|  _| | . |   |
|___|___| |___|___|_| |_|___|_|_|
      |___|(object)inject(ion) v1.11.0

     Runtime Mobile Exploration
        by: @leonjza from @sensepost

 [tab] for command suggestions
[FreedomTool.FreedomTool on (iPhone: 16.1.2) [usb] # ios nsuserdefaults get
{
    AKLastEmailListRequestDateKey = "2024-03-06 09:01:37 +0000";
    AKLastIDMSEnvironment = 0;
    AddingEmojiKeybordHandled = 1;
    AppleKeyboards =     (
        "en_GB",
        "es_ES",
        emoji
    );
    AppleKeyboardsExpanded = 1;
    AppleLanguages =     (
        "en-ES",
        "es-ES"
    );
    AppleLanguagesDidMigrate = 20B110;
    AppleLanguagesSchemaVersion = 3000;
    AppleLocale = "en_ES";
    ApplePasscodeKeyboards =     (
        "en_GB@sw=QWERTY;hw=Automatic",
        "es_ES@sw=QWERTY-Spanish;hw=Automatic",
        "emoji@sw=Emoji"
    );
    INNextFreshmintRefreshDateKey = "730918455.72774";
    INNextHearbeatDate = "731728787.905648";
    NSInterfaceStyle = macintosh;
    NSLanguages =     (
        "en-ES",
        "es-ES",
        en
    );
    PKKeychainVersionKey = 8;
    PKLogNotificationServiceResponsesKey = 0;
    "com.apple.content-rating.AppRating" = 1000;
    "com.apple.content-rating.ExplicitBooksAllowed" = 1;
    "com.apple.content-rating.ExplicitMusicPodcastsAllowed" = 1;
    "com.apple.content-rating.MovieRating" = 1000;
    "com.apple.content-rating.TVShowRating" = 1000;
    "org.freedomtool.isIntroFinished" = 1;
}
FreedomTool.FreedomTool on (iPhone: 16.1.2) [usb] #
```

- You can see an application does not terminate; therefore, the application does not have anti-hook or anti-tampering mechanisms.

Score

Impact: 2

Likelihood: 2

Recommendation

Anti-Debug, Anti-Hook and Integrity Check mechanism (completed in the native code), which will protect against injection of various types of scripts into it, i.e., Frida Gadgets. The application should not allow modifications in its operation.

Remediation Plan

ACKNOWLEDGED: The **Rarimo team** acknowledged the issue.

Remediation Hash

https://github.com/rarimo/FreedomTool

Halborn strongly recommends conducting a follow-up assessment of the project either within six months or immediately following any material changes to the codebase, whichever comes first. This approach is crucial for maintaining the project's integrity and addressing potential vulnerabilities introduced by code modifications.