Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Audit-Report Tuum Hedera Identify Snap & Codebase 01.2025

Cure53, Dr.-Ing. M. Heiderich, M. Pedhapati

Index

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Introduction

“Bringing together everything that’s required to build Decentralized Identity into a new application layer of the web.”

From https://www.tuum.tech/

This report describes the results of a penetration test and source code audit against the Tuum Hedera Identify Snap and its underlying codebase.

To give some context regarding the assignment’s origination and composition, Tuum Technologies, Inc. contacted Cure53 in October 2024. The test execution was scheduled for January 2025, namely in CW02. A total of four days were invested to reach the coverage expected for this project, and a team of two senior testers was assigned to its preparation, execution, and finalization.

The methodology conformed to a white-box strategy, whereby assistive materials such as sources, as well as all further means of access required to complete the tests, were provided to facilitate the undertakings.

The work was structured using a single work package (WP), defined as:

- **WP1**: Pen.-tests & code audits against Tuum Hedera Identify Snap & codebase

It should be noted that this was not the first time Cure53 was tasked with assessing the security of the Tuum Hedera Identify Snap; it was also the focus of a previous audit held in July 2023 (see TUU-01).

All preparations were completed in December 2024, specifically during CW50, to ensure a smooth start for Cure53. Communication throughout the test was conducted through a dedicated and shared Slack channel, established to combine the teams of Tuum and Cure53. All personnel involved from both parties were invited to participate in this channel. Communications were smooth, with few questions requiring clarification, and the scope was well-prepared and clear. No significant roadblocks were encountered during the test. Cure53 provided frequent status updates and shared their findings through the aforementioned Slack channel. Live reporting was not specifically requested for this audit.

The Cure53 team achieved good coverage over the scope items, and identified only a single finding, which was classified as a security vulnerability. The overall minimal number of findings made during testing showcases the efforts made by the Tuum team in creating a well-secured system, with evidence of proactive security measures having been implemented throughout the design and development process. The development team can once again be congratulated on its excellent work, and Cure53 encourages the team to continue in the same vein.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

The report will now shed more light on the scope and testing setup, and will provide a comprehensive breakdown of the available materials. Next, the report will detail the *Test Methodology* used in this exercise. This is intended to show the client which areas of the software in scope have been covered, and which tests have been executed, despite only a single finding having been made. Following this, the report's finding will be discussed in the *Identified Vulnerabilities* section, which includes a technical description, steps to reproduce, as well as mitigation advice.

In summation, the report will finalize with a *Conclusions* chapter in which the Cure53 team will elaborate on the impressions gained toward the general security posture of the Tuum Hedera Identify Snap and its underlying codebase.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Scope

- **Code audits & security reviews against Tuum Hedera Identify Snap & related codebase**
  - **WP1:** Pen.-tests & code audits against Tuum Hedera Identify Snap & codebase
    - **Source:**
      - **URL:**
        - https://github.com/hashgraph/hedera-metamask-snaps/tree/main/packages%2Fhedera-identify-snap%2Fpackages%2Fsnap
      - **Commit:**
        - fe708c9e5f5bea7e8812b2b380125345b7cbacb3
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Test Methodology

This section documents the testing methodology applied by Cure53 during this project and discusses the resulting coverage, shedding light on how various components were examined. Further clarification concerning areas of investigation subjected to deep-dive assessment is offered, especially in the absence of significant security vulnerabilities having been detected.

- The scope of this audit focused on the Tuum Hedera Identify Snap. A white-box penetration testing strategy was employed, leveraging the availability of the Snap's open-source codebase to facilitate a comprehensive analysis. The specific commit reviewed during this audit was: *fe708c9e5f5bea7e8812b2b380125345b7cbacb3*.
- This assessment marked Cure53's second audit of the Tuum Hedera Identify Snap. The Cure53 team conducted its initial audit in July 2023 (TUU-01).
- The testing process for this iteration began by verifying the patches and mitigations implemented for the vulnerabilities reported during the initial audit. This ensured that previously identified issues had been effectively addressed before proceeding to identify any additional security weaknesses or misconfigurations in the updated version of the Snap.
- Cure53's examination began with a review of the scope, the provided documentation, and the Snap's Manifest file. This initial step aimed to ensure that the Snap did not request any unnecessary endowments or permissions. Upon review, it was confirmed that no unused or excessive permissions could be identified in the Snap's configuration, thereby indicating adherence to the principle of least privilege.
- A diff analysis between the current and previous versions of the Snap revealed a few minor changes. Upon auditing these modifications, an issue was identified in the implementation of the new custom UI within the Snap (TUU-06-001).
- The issue above specifically involved the private key persisting in memory even after the Snap was locked, following its display in the UI. This flaw presented a potential security risk, as it is advisable that sensitive data should be cleared from memory promptly, in order to prevent unauthorized access or leakage.
- Next, the Cure53 team conducted an analysis of all RPC methods, ensuring that each sensitive method utilized *snapDialog* in order to prompt user confirmation before executing any actions.
- Issues related to the lack of user confirmation dialogs - such as the previously reported concern in the GDrive configuration - were found to be properly mitigated. No further discrepancies were identified, and all other aspects of the RPC method implementation appeared to align with security best practices.
- The use of the Veramo Agent third-party library, which is utilized to generate and verify VCs and VPs, was thoroughly reviewed.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

- The assessment focused on identifying any potential misconfigurations, issues related to storage, or flaws in the verification and generation processes. A comprehensive evaluation identified no issues, and the implementation was deemed to be secure and in alignment with the intended functionality.
- The evaluation reviewed third-party libraries and integrations to ensure secure practices and to detect any use of vulnerable versions. No issues were identified in this realm.
- The application's error handling and exception management mechanisms were inspected to ensure that they provided clear and accurate feedback to users without exposing sensitive information. Error messages were validated to prevent inadvertent data exposure that could aid potential attackers.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Identified Vulnerabilities

The following section lists all vulnerabilities and implementation issues identified during the testing period. Notably, findings are cited in chronological order rather than by degree of impact, with the severity rank offered in brackets following the title heading for each vulnerability. Furthermore, all tickets are given a unique identifier (e.g., TUU-06-001) to facilitate any future follow-up correspondence.

TUU-06-001 WP1: Snap account private key persists in memory after lock *(Low)*

***Fix Note**: This issue was fixed in v0.2.1 version of Snap and the fix was verified by Cure53 in January 2025. The documented problem no longer exists.*

The Snap account private key of an Identify Snap can be viewed in a new custom UI inside MetaMask (*btn-export-snap-account-private-key)* after unlocking. Testing has shown that this key is not cleared from memory immediately after the user locks the wallet. This means that an attacker with physical access to the victim's computer can retrieve the victim's private keys, even if the MetaMask has been locked.

Steps to reproduce:

1. Ensure that Google Chrome is used and that the MetaMask with Identify Snap is installed.
2. Navigate to the “Identify Snaps” page inside MetaMask, and click on "Export Snap Account Private Key". Make note of this *privateKey*.
3. Lock the MetaMask.
4. Right-click on the extension popup, and click on “Inspect”.
5. Click on the “Memory” tab on the DevTools window.
6. Select *"ljfoeinjpaedjfecbmggjgodbgkmjkjk"* on the “Select JavaScript VM instance” option.
7. Click to take a snapshot.
8. Press CTRL+F and search for the *privateKey*
9. Observe that the key is identifiable within memory. An adversary with access to this dump can retrieve this *privateKey* by searching for all strings with private key length. This can then be used to create verifiable credentials.

To mitigate this issue, Cure53 advises clearing the memory after the user locks the wallet, doing so immediately in order to ensure that the memory is sufficiently and quickly cleared.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Conclusions

As noted in the *Introduction*, this January 2025 penetration test and source code audit was conducted by Cure53 against the Tuum Hedera Identify Snap and its underlying codebase.

From a contextual perspective, four working days were allocated to reach the coverage expected for this project. The methodology used conformed to a white-box strategy, and a team of two senior testers was assigned to the project's preparation, execution, and finalization.

This *Conclusions* section was compiled to enable the client to review the impressions and findings gained by the Cure53 team during the course of testing. On the whole, the team gained an excellent impression of the tested scope's security, although it is recommended that the single vulnerability identified during testing should be mitigated as soon as possible. This will ensure strong protection for the system and its user base.

During the assessment, the codebase underwent rigorous review, which highlighted the Snap's improved resilience. The Cure53 team employed various testing techniques, and confirmed the Tuum team's effectiveness in minimizing the available attack surface. The fact that only a single issue was identified during testing - and that this was rated with a *Low* severity - reflects the development team's successful integration of precautionary measures, and the proactive efforts that have been made to mitigate common security risks.

The review confirmed that the Snap requested only the necessary permissions, demonstrating compliance with the principle of least privilege. A diff analysis between the previous and current versions identified minor updates, and a single issue was discovered, which was related to private key persistence in memory after the Snap was locked (TUU-06-001).

Analysis of the RPC methods used confirmed the implementation of user confirmation dialogs, which used *snapDialog* properly, for all sensitive RPC methods. Additionally, the Veramo Agent library used for generating and verifying VCs and VPs was thoroughly evaluated, with no misconfigurations or issues identified.

The *decodeJWT* function underwent a thorough audit, including an in-depth review of the third-party library it relies on. Specifically, the audit examined whether the use of *deepcopy* within the function could introduce Prototype Pollution. After a comprehensive review, the Cure53 team confirmed that the function was secure.

Almost all of the RPC methods were found to incorporate explicit type checks, which were meticulously reviewed to ensure that no misconfigurations or opportunities for type check bypasses existed.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

The review of third-party integrations confirmed the absence of vulnerable versions, and demonstrated secure practices in implementation. Similarly, the application's error handling mechanisms were inspected to ensure that clear and accurate feedback was provided without exposing sensitive information. No instances of inadvertent data exposure were detected.

Overall, this evaluation revealed a well-secured system, with proactive measures evident in both its design and implementation. The presence of only a single *Low* severity issue further reflected the commitment of the Tuum Technologies team in maintaining a secure and resilient application.

Cure53 would like to thank Kiran Pachhai and Donald Bullers from the Tuum Technologies, Inc. team for their excellent project coordination, support and assistance, both before and during this assignment.