Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Pentest-Report Tuum Hedera Wallet Snap & Code 11.2023

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

Index

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Introduction

"Hedera Wallet Snap is a sophisticated plugin that developers can incorporate into a variety of applications to enhance MetaMask's functionality beyond its native capabilities. Specifically, Hedera Wallet Snap makes it possible to interact with Hedera Hashgraph natively without relying on Hedera JSON-RPC Relay. This plugin can be seamlessly installed as a JavaScript npm package on web applications, thereby offering a range of advanced native features of Hedera to users."

From https://docs.tuum.tech/hedera-wallet-snap/basics/introduction

This report describes the results of a security assessment of the Hedera Wallet Snap and its codebase. The project, which was implemented as a dedicated source code audit, was carried out by Cure53 in November 2023. Registered as *TUU-02*, the examination was requested by Tuum Technologies, Inc., in October 2023.

In terms of the exact timeline and specific resources allocated to the project, Cure53 completed the research in CW46 of 2023. In order to achieve the expected coverage for this task, a total of six days were invested. In addition, it should be noted that a team consisting of three senior testers was formed and assigned to the preparation, execution, and delivery of this project.

For optimal structuring and tracking of tasks, the examination was split into two separate work packages (WPs):

- **WP1**: Source code audits against Tuum Hedera Wallet Snap & codebase
- **WP2**: Code audits & feature reviews against Tuum Hedera Wallet Snap & codebase

As the titles of the WPs indicate, white-box methodology was utilized. Cure53 was provided with all crucial documentation, as well as all further means of access required to complete the tests. Additionally, sources corresponding to the test-targets were shared to make sure the project can be executed in line with the agreed-upon framework.

Overall, the project progressed effectively. To facilitate a smooth transition into the testing phase, all preparations were completed in the week preceding the audit, i.e., in CW45. Throughout the engagement, communications were conducted via a private, dedicated and shared Slack channel. Stakeholders - including the Cure53 testers and the internal staff from Tuum Technologies - could participate in discussions in this space.

The quality of the interactions throughout the test was excellent, with no outstanding queries. These steady exchanges contributed positively to the overall outcomes of this project. The scope was well prepared and clear, which played a major role in avoiding significant roadblocks during the test.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Cure53 offered frequent status updates about the test and the emerging finding, but there was no live-reporting taking place in the frames of *TUU-02*.

The Cure53 team succeeded in achieving very good coverage of the WP1-WP2 targets. Just two security-related flaws could be spotted. All of them have been categorized as general weaknesses with lower exploitation potential. The very limited number of problems and their classification as minor recommendations clearly showcase that the Hedera Wallet Snap already boasts good security premises.

Given the successful nature of the existing protective measures, Cure53 can only congratulate the development team on their excellent work towards securing the inspected Snap. Simultaneously, it should be noted that security should be monitored and general flaws considered, so as to promote good upkeep of the existing robustness within the security posture of the Hedera Wallet Snap.

The following sections first describe the scope and key test parameters, as well as how the WPs were structured and organized. Then, what the Cure53 team did in terms of attack attempts, coverage, and other test-related tasks is explained in a separate chapter on test methodology.

Next, in the absence of vulnerabilities, general findings are discussed under the miscellaneous category of flaws. Issues are presented chronologically, with corresponding technical descriptions, PoC and mitigation advice - as applicable.

The report closes with drawing broader conclusions relevant to this November 2023 project. Based on the test team's observations and collected evidence, Cure53 elaborates on the general impressions and reiterates the verdict. The final section also includes tailored hardening recommendations for the Hedera Wallet Snap complex.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Scope

- **Code audits & security reviews of Tuum Hedera Wallet Snap & codebase**
  - **WP1:** Source code audits against Tuum Hedera Wallet Snap & codebase
    - **Primary focus:**
      - General tests & attacks against browser add-ons, extension snap-ins, conducted independently of the specific use-cases in the context of a crypto-wallet snap
    - **Documentation:**
      - https://docs.tuum.tech/hedera-wallet-snap
    - **Demo video:**
      - https://www.youtube.com/watch?v=wzHn9z3CWpM&feature=youtu.be
    - **Sources:**
      - **URL:**
        - https://github.com/hashgraph/hedera-metamask-snaps/tree/8f3ac19d99d1968e970a2ce0c2650de537fd233b/packages/hedera-wallet-snap/packages/snap
    - **Commit ID:**
      - 8f3ac19d99d1968e970a2ce0c2650de537fd233b
  - **WP2**: Code audits & feature-reviews of Tuum Hedera Wallet Snap & codebase
    - **Sources & documentation:**
      - *See WP1*
    - **Primary focus:**
      - Importing Hedera accounts
      - Sending transactions
      - Retrieval of account-info
      - Further wallet-related issues
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Testing Methodology

This section outlines the testing methodology and coverage achieved during the engagement, shedding light on various components of the Hedera Wallet Snap and codebase that Cure53 inspected. Further clarification is given for the areas of investigation that were subject to deep-dive assessment, while the test team also specifies the techniques applied to evaluate the respective security posture of each component.

The test started off by the testers reviewing the application's scope, provided documentation and the *Manifest* file of the snap. This was done to generally verify the implementation of any unused permissions. With regard to Cross-Site Scripting (XSS), the testing team verified that the Snap Site and UI are seemingly not vulnerable to any potential XSS issues. Generally speaking, XSS is a significant threat for browser extensions, hence the team thoroughly surveyed areas of risk in this regard. Notably, the Hedera Wallet Snap UI does not display a great deal of user-controlled data, thus the potential for XSS is limited and the attack surface is reduced. The frontend of the Snap Site uses React, which leverages a battle-tested escaping mechanism that inherently prevents XSS issues by default. Furthermore, usage of *dangerouslySetInnerHTML,* unsanitized *href* attributes, and other potentially insecure properties were deemed to have been sufficiently avoided.

During the examination, the test team conducted a thorough review of the usage of third-party libraries and integrations within the Hedera Wallet Snap. This evaluation aimed to assess the security practices employed by external entities, checking them against any usage of vulnerable versions. This examination led to the discovery of unpatched vulnerable packages, as detailed in TUU-02-001. The test team also focused on ensuring proper handling and protection of data sent to third-party services. It appeared that no sensitive information was being transferred to third-party services.

Moreover, the team meticulously investigated potential type-assertion issues related to parameters controlled by attackers. In spite of a thorough examination, no security vulnerabilities transpired. The request parameters receive comprehensive validation in each *Remote Procedure Call (RPC)* instance. Therefore, Each *RPC* method was carefully reviewed for potential vulnerabilities, including access control and injection issues. The communication is being performed by web-pages and dApps through MetaMask's *wallet_invokeSnap* request, which safeguards the aforementioned application, ensuring that it possesses all the required privileges before being allowed to interact with the Snap component.

Furthermore, the RPC request handlers - namely *getAccountInfo*, *getAccountBalance* and *transferCrypto* - were thoroughly checked. No issues were identified. However, the code review revealed that newlines could be mishandled. This led to a potential UI spoofing attack, which has been elaborated on in TUU-02-002.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not incur an exploit but may assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy method by which to be called. Conclusively, whilst a vulnerability is present, an exploit may not always be possible. Each ticket has been given a unique identifier (e.g., *TUU-02-001*) to facilitate any future follow-up correspondence.

TUU-02-001 WP1: Unpatched packages in use *(Low)*

***Note**: The issue has been addressed by the development team and the fix was reviewed by Cure53. The issue as described no longer exists.*

Some libraries with known security vulnerabilities are used within the Hedera Wallet Snap complex. Whether these vulnerabilities are exploitable, however, depends on the relevant functionality being used in the targeted application.

Notably, the testing team was unable to comprehensively investigate potential impact of these packages during the limited time granted for this review. Thus, the implications remain unknown at this point and should be researched further internally.

Library name	Introduction agent
@axios	*@hashgraph/sdk@2.34.1*
@crypto-js	*@hashgraph/sdk@2.34.1*

Supply chain security is by no means an easy-to-handle topic or task. Quite often, there is neither an easy nor a perfect solution to be offered. It is advised to be sure to always depend on the most recent version of each library, as this will increase the chances of profiting from all of the previous flaws being found and patched.

To mitigate the existing issues as best as possible, Cure53 recommends upgrading all affected libraries and establishing a policy to ensure libraries remain up-to-date moving forward.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TUU-02-002 WP1: Newlines handled incorrectly in Snap UI (*Info*)

***Note**: The issue has been addressed by the development team and the fix was reviewed by Cure53. The issue as described no longer exists.*

The Hedera Wallet Snap lacks optimal handling for newlines that appear in some of the Snap dialogs within the MetaMask extension. This facilitates spoofing of messages for certain dialogs, which could be used to trick users into performing certain actions on behalf of the Snap component.

The impact of this problem was appropriately downgraded to *Info* because attacks would be limited to phishing attempts in this context.

Affected file:
packages/hedera-wallet-snap/packages/snap/src/rpc/account/transferCrypto.ts

Affected code:

```
    text('Are you sure you want to execute the following transaction(s)?'),
    divider(),
    text(`Memo: ${memo === null || _.isEmpty(memo) ? 'N/A' : memo}`),
    text(`Max Transaction Fee: ${maxFee ?? 1} Hbar`),
  ];
```

Steps to reproduce:

1. Have the MetaMask extension and the Hedera Wallet Snap installed.
2. Start the Hedera Pulse Snap Demo website and connect the snap.
3. Add newlines in the *memo* field when sending *HBAR*.

To mitigate this issue, Cure53 recommends stripping newlines from user-input before creating dialogs. The *snapDialog* function should be considered in this context.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Conclusions

As noted in the *Introduction,* Cure53 is impressed with the security standing of the scope examined during this November 2023 code audit of the Hedera Wallet Snap components. With only two general weaknesses, the project can certainly be viewed as being on the right track in terms of security properties and milestones.

The testing team involved in *TUU-02* wishes to underline that the codebase was subjected to rigorous reviews and probed with compromise techniques of many kinds. The fact that these efforts remained unsuccessful verifies strong efficacy in minimizing the attack surface. The limited number of identified issues - standing at two - demonstrates the development team's successful integration of abundant precautionary measures that serve their purpose of enhancing security of the Hedera Wallet Snap.

The testing team started by auditing the Snap *Manifest* file for insecure configuration patterns, including overly lax permissions. The Cure53 team concluded that the Snap only requests the required permissions, and no misconfigurations could be spotted during this investigation. This realm is aligned with best practice.

Despite the constrained attack surface, the Cure53 team evaluated the exposed RPC methods in an attempt to uncover risk-inducing behaviors. However, strict parameter validation takes place before the RPC calls. As such, secure coding patterns preemptively prevent a range of common mistakes. Furthermore, before making a transaction, the Snap deployment properly requests user-permissions and approvals.

Additional checks for issues associated with the interaction between authorized but malicious dApps and the Snap entities were conducted, but all attempts failed, i.e., they were prevented by security checks in MetaMask.

Upon scrutinizing the Snap's UI components, the Cure53 team encountered a similarly robust security posture. The dApp origin consistently appears in Snap dialogs, effectively thwarting any attempts by malicious dApps to deceive users in terms of executing actions on behalf of a trusted dApp. However, a UI spoofing attack scenario was possible due to the mishandling of newlines. Details of this flaw can be found in TUU-02-002.

The team conducted thorough checks in the frontend, focusing on XSS vulnerabilities. This is because such flaws could directly influence Snap from the site. It was then assessed whether any client-side-related security issues associated with XSS, *postMessage*, and prototype-pollution could be located.

Toward this, the testing team noted that the majority of the frontend utilizes the ReactJS framework, which features a well-tested escaping mechanism. As such, it prevents many XSS-related issues by default.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Finally, the testing team thoroughly investigated the project's dependencies, searching for outdated and vulnerable libraries. As a result, some third-party vulnerable libraries were identified and noted in TUU-02-001.

In conclusion, following the completion of this *TUU-02* security audit, Cure53 garnered a positive impression regarding the security offering established by the Hedera Wallet Snap on the whole. This viewpoint is corroborated by the testers' discoveries spanning only two miscellaneous issues, which indicates proper nullification of highly damaging attack vectors by the development team.

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.