Fine penetration tests for fine websites

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

Pentest-Report Tuum AuthFlow Snap & Codebase 08.2024

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, assigned the reference ID TUU-04, outlines the test methodology and findings of a security evaluation conducted on the Tuum AuthFlow Snap and associated codebase.

Commissioned by Tuum Technologies, Inc. in June 2024, the assessment was undertaken by Cure53 during July and early August 2024 (CW30 and CW31), with a total resource allocation of four days. A dedicated team of two senior security consultants oversaw the project from inception to completion, with preparatory activities fulfilled in July 2024 (CW29) for a smooth start.

A single Work Package (WP1) was created and defined as *WP1: Pen.-tests & code audits against Tuum Authentication Snap & codebase*. Cure53 was granted unfettered access to the system, including source code, test-user credentials, and any additional resources deemed necessary.

In context, the Snaps developed by Tuum have been analyzed by Cure53 on previous occasions, with the last review of the Hedera Wallet Snap performed in April 2024 (see TUU-03).

Effective communication was maintained through a dedicated Slack channel, facilitating collaboration among all relevant stakeholders. The well-defined project scope and clear communication channels ensured a productive testing process. Cure53 also relayed regular status updates on the progress and relevant findings.

In summary, Cure53's endeavors yielded a total of two findings, categorized as one security vulnerability and one lower-risk weakness. The concise composition of the target means that breach opportunities are few and far between from the outset. Nevertheless, the test team's diligent and extensive efforts were only able to yield a limited number of findings, suggesting a robust security posture for the Tuum Authentication Snap. However, the identified Markdown injection vulnerability (see TUU-04-001), warrants attention and remediation despite the limited likelihood of exploitation.

The remainder of the code base is effectively risk averse, exhibiting only one other *Info*-rated discovery. Based on this wholly positive verdict, Cure53 can only commend the Tuum developer team for their progress toward enhancing the security of their Snap components.

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

The report presents a number of key chapters moving forward. Firstly, the *Scope* provides all general setup information in the bullet points below. Next, the *Test Methodology* clarifies the evaluation techniques applied by Cure53 and all interesting subsequent observations, with the aim of verifying the extent of the test team's efforts in spite of the almost negligible yield of findings.

The document then lists all security problems in chronological order of detection (rather than degree of severity), grouped into two subcategories: *Identified Vulnerabilities* and *Miscellaneous Issues*. Each corresponding ticket gives a technical explanation, Proof-of-Concept (PoC) and/or steps to reproduce, code snippets, and fix or preventative advice. Lastly, the *Conclusions* section summarizes Cure53's estimation of the targets based on the evidence collected.

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

Scope

- **Code audits & security reviews against Tuum Authentication Snap & related codebase**
  - **WP1:** Pen.-tests & code audits against Tuum Authentication Snap & codebase
    - **Source code:**
      - https://github.com/tuum-tech/authflow-snap/releases/tag/0.1.0
    - **Commit ID:**
      - *3a427023d1b8c8ec8a6da32dc736a090fe17b065*
  - **Documentation:**
    - https://www.figma.com/board/tnxEBNMqS2nJwlrGvbgfOX/AuthFlow-Audit-Documentation?node-id=0-1&t=06HGA5JPK91G7qDO-1
  - **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

Testing Methodology

This section outlines the specific techniques applied and resulting coverage achieved during this project focusing on the Tuum AuthFlow Snap and codebase. Further clarification concerning areas of investigation subject to deep-dive assessment is offered in addition, particularly considering the general lack of significant findings. The test team also clarifies the tactics employed to evaluate the respective security posture of each aspect.

- The scope of the audit concentrated on the Tuum Authentication Snap exclusively, while the Snap testing site was deemed out-of-scope. This approach followed a white-box pentesting strategy since Snap's full source code is open-source.

- Cure53's examinations initiated with a review of the scope, provided documentation, and the Snap's Manifest file. This initial step served to verify whether the Snap requested any unnecessary endowments or permissions. No unused permissions were identified.

- The test team meticulously investigated any usage of third-party libraries and integrations within the Tuum Authentication Snap, which aimed to appraise the security practices employed by external entities and ensure that vulnerable versions were not in use. Positively, no correlating issues were identified.

- Each RPC method was stringently explored for potential issues related to access control, insecure user-input handling, and other JavaScript-related flaws. Communication is facilitated by web pages and dApps via MetaMask's *wallet_invokeSnap* request and registered in Snap via *onRpcRequest*, ensuring that the application possesses all necessary privileges before permitting interaction with the Snap.

- Next, Cure53 determined that all actions implemented by the Snap were intuitive, clearly communicated, and required explicit user consent. The vast majority of the RPC methods leverage the *snap_dialog* provided by MetaMask to retrieve user confirmation. However, one minor shortcoming related to user input handling was located, as discussed in ticket TUU-04-001.

- The storage of Basic and Verifiable credentials was vetted, whereby Cure53 confirmed that MetaMask Snap storage is utilized, which is encrypted and only accessible after MetaMask is unlocked.

- The communication with identity snap was explored for any discrepancies, though no faults of this ilk were detected.

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

- Lastly, the application's error handling and exception management mechanisms were closely inspected to ensure that they provide appropriate feedback to users without simultaneously exposing data. In light of this, an *Info*-rated drawback was identified and documented under ticket TUU-04-002.

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-04-001*) to facilitate any future follow-up correspondence.

TUU-04-001 WP1: Text used over copyable during dialog display *(Low)*

***Fix Note**: This issue was fixed and the fix was verified by Cure53 in mid August 2024. The documented problem no longer exists.*

Cure53 observed that the dialogs shown in the Snap display user input by utilizing the *text* method provided via the MetaMask UI. However, this process is problematic since the *text* method permits rendering control characters and Markdown. Although newlines are sanitized, Markdown text can be injected into these fields.

For instance, the RPC method allows passing friendly names separated by commas. By combining this with Markdown, an adversary could perform phishing while displaying user input by confusing the affected user.

PoC:

```
window.ethereum.request({
      method: 'wallet_invokeSnap',
    params:{
        snapId: 'local:http://localhost:8081',
        request:{
        method: 'createVerifiablePresentation',
        params: {
          credentialDescription:"asdf, Origin: **htps://google.com**",
        },
      }}})
```

To mitigate this issue, Cure53 advises enforcing that user input is never passed into the *text* method. Alternatively, the Tuum team should utilize the *copyable* method to display text (as recommended by MetaMask[1]), which ignores Markdown and other special characters.

[1] https://docs.metamask.io/snaps/learn/best-practices/security-guidelines/

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 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, while a vulnerability is present, an exploit may not always be possible.

TUU-04-002 WP1: Insecure logging to console leaks sensitive information *(Info)*

***Fix Note**: This issue was fixed and the fix was verified by Cure53 in mid August 2024. The documented problem no longer exists.*

Testing confirmed that the logging mechanism utilized in the Snap leaks sensitive information, such as passwords, in the dev console script context. This allows a malicious actor with physical access to the browser to obtain access to passwords, even after a MetaMask wallet logout.

Affected file:
authflow-snap/authflow-snap/packages/snap/src/snap-classes/SnapState.ts

Affected code:

```
 public static async outputCredentialsToConsole() {
    try {
      const credentials = await SnapState.getCredentials();

      if (credentials) {
        Object.entries(credentials).forEach(([key, item]) => {
          console.log(`Key: ${key}, Item: ${JSON.stringify(item)}`);
        });
      } else {
        console.log('No credentials found.');
      }
    } catch (error) {
      const errorMessage =
            error instanceof Error ? error.message : 'An unknown error
occurred';
                console.error(`Error in outputCredentialsToConsole: $
{errorMessage}`);
    }
  }
```

To mitigate this issue, Cure53 advises removing sensitive logging from the Snap in all affected areas.

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

Conclusions

As noted in the *Introduction*, the Tuum Authentication Snap garnered a highly favorable impression during this August 2024 penetration test regarding its security posture under the current implementation. This verdict is corroborated by the minimal number of issues outlined in this report, as well as the minor risk levels attached.

The testing team effectively covered a significant proportion of the scope area during the procedures. However, Cure53 must emphasize that the modest list of defects is likely due to the limited functionality established within the framework, as well as the constrained attack surface for Snaps in general.

The analysis was initiated by reviewing the Snap Manifest file for insecure configuration patterns and overly lax permissions. Here, Cure53 was able to verify that the Snap only requests the required permissions, while all configurations complied with wider industry best practices to reduce the overarching attack surface to the smallest possible magnitude.

Next, Cure53 closely inspected all RPC methods registered by the Snap, which aimed to verify whether one could invoke arbitrary methods without user approval, detect any insecure handling of user input, or pinpoint other logical weaknesses related to request handling from the dApps and other Snaps. These efforts led to identification of the pitfall highlighted in ticket TUU-04-001.

Since the Authentication Snap stores sensitive information such as Basic and Verifiable credentials, the test team appraised the protocols responsible for data retention. Here, Cure53 concluded that these materials were held securely in an encrypted MetaMask Snap storage. Onward, the testers vetted the efficacy of the error message and logging implementations. Due to the fact that insecure logging of pertinent details may be retrievable by an attacker with physical access to the machine (even with MetaMask locked), Cure53 sought to locate any connected flaws. This endeavor raised the concern outlined in ticket TUU-04-002.

Finally, the assessors systematically investigated the project's dependencies by searching for outdated and vulnerable libraries, though no risk-inducing behaviors were noted in this area. To conclude, following the finalization of this Q3 2024 security audit, Cure53 is pleased to confirm that the Tuum AuthFlow Snap's security posture is commendable. This positive viewpoint is bolstered by the scant number of identified faults, which validates the app's effectiveness in neutralizing potential attack vectors.

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