Audit-Report Dedaub MetaMask Snap Code & Build 12.2023

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

Index

Introduction

> *"Dedaub is at the forefront in the smart contract security and auditing space. The founders, as well as many of Dedaub's auditors, have a strong academic research background together with a real-world hacker mentality to secure code."*
>
> From https://dedaub.com/about

This report, identified as DDB-01, presents the findings of a penetration test and source code audit conducted against the Dedaub MetaMask Snap and codebase, which was performed in late November and early December 2023, specifically during CW48.

The project was commissioned by Dedaub Ltd. and executed by Cure53, assigning a total of three days toward achieving the expected level of coverage. The assessment actions were divided into two distinct work packages (WPs), defined below:

- **WP1**: Source code audits against Dedaub MetaMask Snap & codebase
- **WP2**: Snap & use-case-specific feature reviews against Dedaub MetaMask Snap

Cure53 was provided with access to the respective sources and miscellaneous data to complete the tests, conforming with a white-box methodology as requested. A team of three senior testers was assigned to this project's preparation, execution, and completion, each of which possess extensive technical experience and know-how for procedures of this ilk.

Preparatory activities were finalized in November 2023, during CW47, ensuring a smooth start for the testing process.

Communication was facilitated through a dedicated and shared Slack channel between the Dedaub and Cure53 teams, allowing for seamless interaction between all involved parties. Communications were efficient with few questions requiring clarification, and the scope was well-defined and clear. No significant roadblocks were encountered at any stage, testament to the optimal project composition. Cure53 provided regular status updates on the progress of the exercise and identified findings, performing live reporting via the aforementioned Slack channel.

After the completion of the review window, the Cure53 team identified only a single finding (categorized as a security vulnerability), despite ample coverage achieved against the focus traits.

The detection of only one vulnerability, which scored a *Low* severity rating, is indicative of a steadfast security posture established for the Dedaub MetaMask Snap. The Dedaub team must be praised for their achievements, though a long-term strategy of continuous improvement and testing should be enforced to maintain this level of resilience for the MetaMask Snap.

The report will now display the *Scope*, inclusive of the test setup and available materials, in bullet-point format. This section will be followed by a chapter outlining the *Test Methodology*, demonstrating to the client the areas covered and abundant tests conducted despite the limited volume of findings. Subsequently, the report will present all findings in chronological order, distinguishing between identified vulnerabilities and general weaknesses with two distinct subsections. Each finding will be accompanied by an advanced technical description, a Proof-of-Concept (PoC) if necessary, code examples, and fix implementations for consideration.

To round off proceedings, the report will conclude with a rundown of the overall impressions gained from the testing process, as well as in-depth discussion of the perceived security posture pertaining to the Dedaub MetaMask Snap, codebase, and features.

Scope

- **Penetration tests & code audits against Dedaub MetaMask Snap codebase & build**
  - **WP1**: Source code audits against Dedaub MetaMask Snap & codebase
    - **Primary focus:**
      - General tests & attacks against browser add-ons and extension Snap-Ins, independently of specific use case as a MetaMask Snap
    - **Sources:**
      - **URL:**
        - https://github.com/Dedaub/metamask-snap
    - **Commit ID:**
      - fb485d438caf27a32f143e70fd235678dc26b9cc
    - **NPM registry:**
      - dedaub-metamask-snap@0.1.3
  - **WP2**: Snap & use-case-specific feature reviews against Dedaub MetaMask Snap
    - **Primary focus:**
      - Specific features, including (but not limited to):
      - Integration into MetaMask
      - Interaction with MetaMask features
      - Assumption of compromised or malicious server responses from Dedaub API
    - **Sources:**
      - *See WP1*
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

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 and the perceived security posture of each component. Further clarification concerning areas of investigation subject to deep-dive assessment is offered, especially in the absence of significant security vulnerabilities detected.

- The assignment commenced by reviewing the application's scope and the Snap's manifest file to verify the implementation of any unused or insecure permissions.

- The audit team conducted a thorough review of the usage of third-party libraries and integrations within the system. This evaluation aimed to assess the security practices employed by these external entities, as well as any usage of vulnerable versions. The configuration of optimum coding practices, adherence to industry standards, regular software updates, and presence of vulnerability management processes were all vetted as part of this procedure. Positively, no correlating detriments were observed here.

- Furthermore, the testing team performed dedicated investigations for either logical bugs or potential miscalculations. Fortunately, no major issues were unearthed in this realm since the core Snap logic is exclusively managed in the backend via the API in the form of transaction emulation. Nevertheless, a discrepancy was observed in the Snap's rounding mechanism, resulting in inaccurate data that subsequently led to balance misinterpretation, as detailed in ticket DDB-01-001.

- Moreover, the Snap system refrains from exposing any Remote Procedure Call (RPC) methods, significantly diminishing the plausible attack surface and thereby fortifying Snap's defensive capabilities.

- Elsewhere, the application's error handling and exception management mechanisms were inspected to ensure they provide appropriate feedback to users without simultaneously enabling data exposure. In light of this, error messages were validated to be clear and accurate, while any sensitive information that may otherwise facilitate attacker exploitation was appropriately omitted.

- Holistic appraisals of Snap's communication with the API via fetch requests revealed a robust implementation on the whole. Notably, the URL is hardcoded and the concatenated parameters remain uncontrollable, effectively restricting the attack surface and nullifying potential drawbacks such as path traversal.

- Lastly, the Cure53 consultants sought to pinpoint any methods by which to compromise the Snap if the API is breached and an adversary provides arbitrary responses. Although this behavior may display incorrect balances, the team was unable to locate any serious ramifications in this area, such as prototype pollution and/or other flaws related to subpar API response handling.

Identified Vulnerabilities

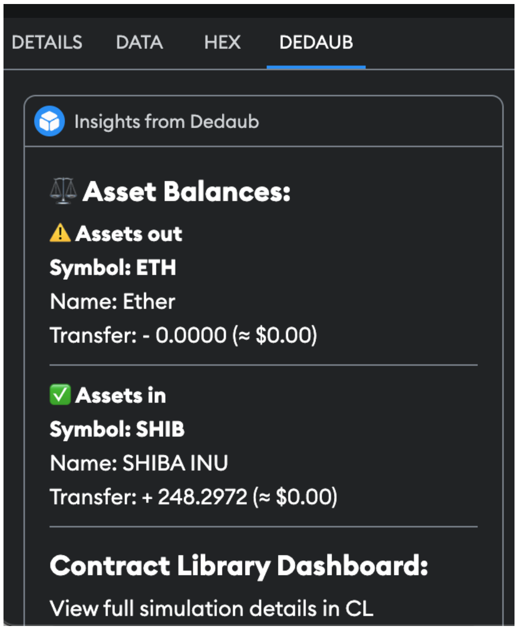
The following section lists all vulnerabilities and implementation issues identified during the testing period. Notably, findings are cited in chronological order rather than by degree of impact, with the severity rank offered in brackets following the title heading for each vulnerability. Furthermore, all tickets are given a unique identifier (e.g., *DDB-01-001*) to facilitate any future follow-up correspondence.

DDB-01-001 WP1: Incorrect rounding evokes balance misinterpretation *(Low)*

***Note**: This issue was fixed by the development team and the fix was verified by Cure53 by inspecting the corresponding commit on Github.*

Testing confirmed that the Snap UI displays numbers rounded to only four digits, which could in theory lead to a misinterpretation of numeric values. Henceforth, users could be deceived and presented with inaccurate information, particularly regarding coins with substantial values that require additional digits.

Despite the potential for manipulation, exploitation strategies are constricted to phishing and remain specific to a limited range of coins within this context. Appropriately, the ticket's attached severity score was downgraded to *Low*.



⚖️ **Asset Balances:**

⚠️ **Assets out**
Symbol: ETH
Name: Ether
Transfer: - 0.0000 (≈ $0.00)

✅ **Assets in**
Symbol: SHIB
Name: SHIBA INU
Transfer: + 248.2972 (≈ $0.00)

Contract Library Dashboard:
View full simulation details in CL

Fig.: Small ETH to SHIB transfer rounded to 0.

To mitigate this issue, Cure53 recommends rounding the numbers in question based on the decimal values, thus providing granular representation and enhancing the overall reliability of numeric displays.

Conclusions

Generally speaking, the code base was subjected to meticulous reviews and scrutiny through the application of sophisticated compromise techniques. Considering that the vast majority of those were repelled, one can conclude that the framework aspects demonstrate elevated efficacy in minimizing the attack surface. The low number of identified issues unequivocally attests to the development team's adept integration of numerous precautionary measures, significantly enhancing the degree of shielding constructed for the Dedaub MetaMask Snap.

The testing team initiated the assignment by auditing the Snap manifest file for insecure configuration patterns, including 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 limit the overarching attack surface to the smallest magnitude possible.

Despite the low propensity for offensive vectors, the Cure53 auditors conducted a sweeping evaluation of the source code to identify any potential logical vulnerabilities or error mishandling instances in specific cases. The development team’s sound paradigms to negate these vulnerabilities was noted with distinction. However, the endeavors quickly confirmed that the Snap currently leverages incorrect rounding methods, henceforth contributing to a misinterpretation of balances. Auxiliary information on this area of concern is presented in ticket DDB-01-001. A range of alternative examination strategies were performed to uncover weaknesses related to the interaction between the Snap and backend server via the API. However, all attempts were ultimately futile due to the limited controllable input and, in turn, restricted risk susceptibility.

The Snap maintained a self-contained environment by refraining from interacting with other dApps and avoiding RPC method exposure. This strategic approach effectively eliminated a broad spectrum of potential backdoor circumstances. The testing team honed in on uncovering pitfalls that could trigger unexpected behaviors in the Snap, or otherwise present inaccurate data as insights. In a similar manner, these undertakings did not identify any substandard practices. Finally, the assessors systematically investigated the project’s dependencies by searching for outdated and vulnerable libraries, though no points of contention were raised in this area either.

To conclude, following the finalization of this security audit, Cure53 materialized a significantly favorable impression of the Dedaub Wallet Snap's overall security posture. This positive assessment is further bolstered by the almost negligible number of identified faults, which validates the app's effectiveness in neutralizing a swathe of potential attack vectors.

Cure53 would like to thank Nikos Petridis from the Dedaub Ltd. team for his excellent project coordination, support, and assistance, both before and during this assignment.