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

Pentest-Report MetaMask Signing Snap & Codebase 03.2024

Cure53, Dr.-Ing. M. Heiderich, A. Belkahla

Index

Introduction

"Provide automatic message signing using a pseudo randomly generated snap private key"

From https://github.com/MetaMask/message-signing-snap

This report details the findings from a penetration test and source code audit of the MetaMask Message Signing Snap and codebase. Commissioned by ConsenSys Software Inc. in March 2024, Cure53 conducted the engagement later that same month, namely during CW10. A total of three work days were dedicated to achieving the project's intended coverage and objectives.

The project comprised a single work package (WP), entitled *WP1: Source Code Audits against MetaMask Message Signing Snap & Codebase*. Cure53 employed a white-box methodology, leveraging access to source code, documentation, and other necessary resources provided by the ConsenSys Software maintainers. A team of two senior testers managed the project's various stages, including preparation, execution, and finalization.

Thorough preparation measures were completed in late February and early March 2024 (CW09). A dedicated Slack channel fostered seamless communication between the ConsenSys Software and Cure53 teams. The well-defined scope and clear communication channels minimized inquiries and ensured a smooth testing phase with regular status updates provided by Cure53. Live reporting was deemed surplus to requirements for this audit.

The Cure53 team achieved successful coverage over the defined scope and identified a single finding, which was deemed a general weakness with lower exploitation potential. Evidently, this outcome indicates a robust security posture for the MetaMask Message Signing Snap and its effective mitigation of common threats, confirming the effectiveness of the implemented security measures. As such, the developer team deserves high praise for their commitment to secure coding practices.

However, Cure53 highly recommends remaining vigilant and maintaining a focus on security best practices to ensure the continued resilience of the MetaMask Message Signing Snap against evolving threats.

The report will now document the scope and testing setup, as well as provide a comprehensive breakdown of the available materials. This will be followed by a chapter outlining the test methodology, which clarifies the techniques applied and coverage depth achieved throughout this audit. Subsequently, the report will list all findings identified in chronological order, starting with the detected vulnerabilities (albeit none were encountered here) and followed by the general weaknesses unearthed.

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

Each finding will be accompanied by a technical description, Proof-of-Concept (PoC) where applicable, plus any relevant mitigatory or preventative advice to action.

In summation, the report will finalize with a conclusion in which the Cure53 team will elaborate on the impressions gained toward the general security posture of the MetaMask Message Signing Snap and codebase, offering remediation advice and suggested follow-up actions for the project handlers to consider.

Scope

- **Crypto review & code audit against the MetaMask Message Signing Snap & codebase**
  - **WP1**: Source code audits against MetaMask Message Signing Snap & codebase
    - **Source:**
      - **URL:**
        - https://github.com/MetaMask/message-signing-snap
      - **Commit:**
        - c91fd3b1ad850d5dfc188a5bd9505df6317e5b2e
    - **NPM package:**
      - https://www.npmjs.com/package/@metamask/message-signing-snap
    - **Documentation:**
      - https://github.com/MetaMask/message-signing-snap/blob/HEAD/RPC.md
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Testing Methodology

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

- The Cure53 team initiated the assessment by examining the MetaMask Message Signing Snap's scope, accompanying documentation, and the Snap's Manifest file. This review focused on verifying the implementation of the principle of least privilege regarding permissions. By requesting only the essential permissions to function correctly, the MetaMask Message Signing Snap effectively minimizes its attack surface, thereby reducing potential security risks.

- During the examination, the test team conducted a thorough review of the usage of third-party libraries and integrations within the MetaMask Message Signing Snap system. This evaluation aimed to appraise 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 ticket MM-03-001.

- The request parameters receive comprehensive validation in each Remote Procedure Call (RPC) instance. Accordingly, each RPC method was carefully reviewed for potential vulnerabilities. The communication is performed by web pages and dApps via MetaMask's *wallet_invokeSnap* request, which in essence safeguards the aforementioned application and ensures that all required privileges are met before permitting interaction with the Snap component.

- Furthermore, the RPC request handlers were exhaustively probed. The *getPublicKey* method serves to fetch the associated public key of the entropy by leveraging the *secp256k1.getPublicKey* function. The *signMessage* method enables the signing of a designated message prefixed with *metamask:*. This utilizes a securely-generated private key via *snap_getEntropy* that is specific to the Snap and the user's account, leveraging Snap's API in the process. The cryptography operations depend on the *noble* library, which was hence subjected to systematic inspections by the Cure53 testers. A robust and secure implementation was verified in this realm.

- Elsewhere, the Cure53 team conducted an in-depth examination of the cryptography security implemented in the Snap, which relies on the *noble* package for operational purposes. Fortunately, no vulnerabilities were found within this area either.

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

- Lastly, the *signMessage* function underwent meticulous fuzz testing to identify potential abuse opportunities or exploitation avenues. Fortunately, the function remained stable throughout the testing process and exhibited resilience to manipulation in general.

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, while a vulnerability is present, an exploit may not always be possible.

MM-03-001 WP1: Unpatched packages utilized *(Low)*

Cure53 noted that some libraries with known security vulnerabilities are used within the MetaMask Message Signing Snap. Whether these vulnerabilities are exploitable, however, depends on the degree of functionality usage in the targeted application.

Due to the time limitations of this engagement, a more in-depth analysis of these libraries using a dedicated vulnerability scanner or manual code review is recommended to definitively assess their exploitability within the context of the MetaMask Message Signing Snap.

Library name	**Version**
ip	*2.0.0*

While securing the codebase itself is crucial, supply chain vulnerabilities pose a significant ongoing risk. Thus, external dependencies can introduce unforeseen weaknesses into an application. The use of libraries with known vulnerabilities within the MetaMask Message Signing Snap underscores this challenge. Unfortunately, a single foolproof solution to eliminating supply chain threats remains implausible. However, the dev team should ensure that the most recent version of each library is installed, as this will allow the associated infrastructure to benefit from patches that have been rolled out for previously identified vulnerabilities.

To mitigate this issue as effectively as possible, Cure53 recommends upgrading the affected library 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

Conclusions

The *Conclusions* section aims to provide a definitive estimation of the scope's security foundation based on the examination results achieved during this Q1 2024 review. In short, the detection of only one miscellaneous issue speaks volumes for the MetaMask team's proficient security implementation regarding the MetaMask Message Signing Snap.

The Cure53 team subjected the MetaMask Message Signing Snap's codebase to a rigorous security assessment, employing diverse penetration testing techniques. These extensive efforts yielded only a single finding, corroborating the effectiveness of the implemented security measures. This minimal number of vulnerabilities reflects the development team's successful integration of robust security practices, minimizing the attack surface and enhancing the overall security posture of the Snap.

The Cure53 team commenced the assessment by examining the Snap Manifest file for any insecure configuration patterns, particularly concerning overly broad permissions. This review confirmed that the Snap adheres to best practices by requesting only the essential permissions to function, effectively constraining the attack surface.

Despite the limited attack surface, the team further evaluated the exposed RPC methods to identify any potential risks. However, the analysis revealed the implementation of strict parameter validation prior to each RPC call. These secure coding practices proactively prevent a range of common vulnerabilities and erroneous behaviors.

The Cure53 team also investigated potential pitfalls arising from interactions between authorized (but malicious) dApps and the Snap. However, these attempts were thwarted by the existing security checks within MetaMask, demonstrating its capability to repel such threats.

Furthermore, a sweeping examination of the cryptographic logic used for message signing was ultimately unfruitful, since the team could not detect any correlating issues. This confirms the secure implementation of cryptography within the Snap. Finally, the testing team thoroughly investigated the project's dependencies, searching for outdated and vulnerable libraries. This initiative identified a third-party library that is affected by a known vulnerability, as highlighted in ticket MM-03-001.

To finalize, Cure53 is pleased to confirm that a unanimously commendable verdict was reached with regards to the security offering established by the MetaMask Message Signing Snap.

Cure53 would like to thank Prithpal Sooriya and Christian Montoya from the ConsenSys Software Inc. team for their excellent project coordination, support, and assistance, both before and during this assignment.