Fine penetration tests for fine websites

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

Pentest-Report Silent Shard Snap 06-07.2023

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

Index

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

Introduction

“Silent Shard is an MPC-based TSS complemented by cyber-physical proofs for much usable, secure, and truly decentralized support for digital wallets, exchanges and institutional asset enterprises.”

From https://silencelaboratories.com/silent-shard/

This report - identifiable by the acronym *SIL-02* - documents the scope, coverage, and findings of a penetration test and source code audit against the Silent Shard Snap and codebase.

This external survey was proposed by Silence Laboratories India Private Limited (SLIPL) in May 2023 and subsequently carried out by Cure53 in June and early July 2023, namely between CW24 and CW26. Six working days were agreed upon in advance to enable the three-person audit team ample coverage and opportunities to discover security-relevant limitations. All assessment actions were grouped into a single work package (WP), which reads as follows:

- **WP1:** Source code audits & pentests against Silent Shard Snap & codebase

One would like to stipulate here that the originally requested scope was divided into two separate reports; the second of which pertains to the Silent Shard applications and cloud functions documented under the report entitled *SIL-03*.

The project maintainers granted access to a number of assisting materials to create a seamless and productive environment for the pentesters endeavors. This included sources, comprehensive documentation, test-user credentials, and other miscellaneous items required for access or scope awareness. The methodology of choice for this particular exercise was white box.

The team performed necessary preparations ahead of the active testing window (specifically CW23 June 2023) so that the examinations could commence smoothly.

Cross-team collaborations were enabled via Slack. All participating personnel from Silence Laboratories and Cure53 were invited to join the dedicated channel and engage in the discussions, which were conducted concisely and transparently for progress updates, queries, inter alia.

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

One can positively acknowledge that no notable hindrances were encountered throughout the entirety of this exercise, testament to the ideal scope preparation and client's assistance. Live reporting was also offered but deemed surplus to requirement.

To provide a concise summary of the final outcome, the Cure53 team was only able to detect two issues affecting the targets, despite satisfactory breadth of coverage. One of those was categorized as a security vulnerability and the other was miscellaneous in nature.

Evidently, this is a miniscule yield of findings considering the magnitude of the scope, attesting to the Silence Laboratories team's performant security integrations on the whole. Cure53 concludes this procedure with a praiseworthy opinion of the framework, for which the developers and management deserve every plaudit. Nonetheless, the guidance offered via the two tickets should be heeded to resolve the outstanding fault areas and install definitive protection.

The report will now provide a more detailed explanation of the scope and test setup, as well as the available material for testing. This section will be followed by a chapter that outlines the test methodology used in this exercise. The purpose of this segment is to demonstrate to the client which areas of the software within the scope have been covered and the multitude of test stratagems conducted, despite the small volume of findings identified.

Subsequently, the report will present a list of all findings in chronological order; firstly the *Identified Vulnerabilities* will be listed, followed by the *Miscellaneous Issues*. Each finding will be accompanied by a technical description and a Proof-of-Concept (PoC) if required. Additionally, advice on mitigation or fix measures for each issue is stipulated.

Finally, the report will conclude with an overall assessment of the test conducted by Cure53. This section will discuss the team's general impressions and provide broader analysis regarding the Silent Shard Snap and codebase's security posture.

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

Scope

- **Source code audits & penetration tests against Silent Shard Snap**
  - **WP1:** Source code audits & pentests against Silent Shard Snap & codebase
    - **Sources:**
      - **URL #1:**
        - https://github.com/silence-laboratories/silent-shard-snap/
      - **Commit #1:**
        - 01e0b52a7f318b5f4665915f4e16018c787315d5
      - **URL #2:**
        - https://gitlab.com/com.silencelaboratories/shard-metamask-snaps/silentshardnewui/
      - **Commit #2:**
        - 48709ed0a60d39d26f67f17a3244afae6212abdd
  - **Documentation:**
    - **Architecture UML:**
      - https://sequencediagram.org/index.html?presentationMode=readOnly#initialData=A4Qw...
    - **TDD/RFCs v1:**
      - https://silence-laboratories.gitbook.io/metamask-snap/6TKlcSxbkxdYtYPlPPM6/
    - **PRD v1:**
      - https://silenhttps://silence-laboratories.gitbook.io/metamask-snap/6TKlcSxbkxdYtYPlPPM6/ce-laboratories.gitbook.io/metamask-snap-1/kjwX2quJyZOmIyE4LRpb/
  - **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 applied against the varying in-scope Silent Shard Snap components and codebase during the course of this engagement. With consideration to the minimal yield of findings, the following passages are provided to elucidate the team's exhaustive endeavors and summarize the degree of security efficacy offered by the targets.

WP1: Audits & pentests against Silent Shard Snap & codebase

The advanced strategies applied by Cure53 against all WP1-associated features are extrapolated below. Pertinently, the auditors placed particular emphasis on implementing compromise approaches that comply with a white-box-based testing methodology.

- The Cure53 consultants initiated proceedings by reviewing the application's scope, requirements, and provided materials. This helped to facilitate acclimatization with the application and surroundings.

- The first area of concern pertained to Cross-Site Scripting (XSS), for which the test team sought to verify whether Snap offered any potential susceptibility to XSS issues. Generally speaking, XSS is a significant threat against browser extensions. Thus, the team prioritized surveying the scope for associated risks.

- The Silent Shard platform is deserving of commendation for the sound design paradigms implemented for the purpose of minimizing the likelihood of XSS, considering that only user-controlled text is renderable within Snap. Additionally, the Silent Shard UI displays scant user-controlled data in general. These characteristics in combination serve to restrict XSS and the wider attack surface to the highest possible degree.

- The audit team reviewed the framework's adherence with web security best practices, as well as all hardening guidance related to MetaMask. These efforts verified conformance with recommended ideals.

- Cross-Site Request Forgery (CSRF) oftentimes emanates from web applications in general and can induce major risk in the eventuality an endpoint becomes exploitable. However, since the entire web application requires a bearer token for authentication, CSRF is mitigated since the token essentially acts as an anti-CSRF token.

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

- An extended appraisal of the third-party library and integration usage within the system was conducted, which aimed to itemize all assets employed by these external entities, ascertain whether any vulnerable versions were deployed, and determine their overarching resistance to security infiltration.

- The implementation of secure coding practices, adherence to industry standards, regular software updates, and presence of vulnerability management processes were all vetted as part of this procedure. Ticket SIL-02-002 addresses the detrimental behaviors observed in this regard.

- The aforementioned processes were also complemented by dedicated searches to ensure airtight data handling and protection when delivered to third-party services. In essence, Cure53 strove to verify whether any sensitive information was transferred in error to third parties.

- Elsewhere, the application's error handling and exception management mechanisms were inspected to ensure they provide appropriate feedback to users without simultaneously incurring data exposure. Positively, error messages were validated to be clear, accurate, and omitted any pertinent information that may otherwise facilitate attacker exploitation.

- The application relies on communication with external entities, such as dApps and blockchain networks. The security protection exhibited by these communications toward the prevention of unauthorized access, data leakage, or Man-in-the-Middle (MitM) attacks was systematically estimated. Moreover, the team evaluated the implementation of robust communication protocols, encryption mechanisms, and secure storage of network credentials.

- Another potential area of weakness concerns the handling of postMessage, which may evoke DOM-based XSS or correlating client-side defects if actioned incorrectly. Although this mechanism is not directly implemented by Snap, postMessage is leveraged for communication with the MetaMask extension. Nonetheless, the team's efforts towards unearthing vulnerabilities in this area yielded negligible results.

- Each RPC method was carefully reviewed for prevalent access control and injection weaknesses. Cure53 noted that communication is performed by webpages and dApps via MetaMask's *wallet_invokeSnap* request, which guarantees that the aforementioned application possesses all necessary privileges before permitting interaction with Snap.

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

- Specifically, the team instigated unrestricted and restricted approaches in an attempt to abuse the Snap's RPC methods via alternative Snaps or dApps. As a result, Cure53 noted one flaw pertaining to the *tss_sendSignRequest* method due to the lack of input sanitization, as described in ticket SIL-02-001.

- Auxiliary undertakings to achieve JavaScript execution within the Snap via the *showConfirmationMessage* function were conducted, though these were unfruitful considering that all user-controlled input is rendered as text.

- Code reviews assisted with tracking the key share process from the point of creation onward. The primary objective here was to validate whether this could be leaked by any installed functionality. All instances of access to the decrypted key shares via *getSilentShareStorage* were also audited for leakage potential.

- Furthermore, the creation of the key share using the *snap_getEntropy* method with a salt was considered appropriate and secure, since a deterministic 256-bit entropy value is generated that is specific to the Snap and user account in question.

- The system’s handling of actions that require a user's active permission - such as the initiation of pairing, creation of a new keygen, or signing and transactions requests - was inspected to determine the propensity for logic faults and bypasses. Similarly, Cure53 was unable to detect any risk-inducing operations here.

- The communication with the Cloud Functions API via fetch requests between the Snap and mobile application was assessed, which verified valid implementation on the whole. The URL is hardcoded and functions were created as a wrapper for each API call, thereby constricting the attack surface and nullifying limitations such as path traversal.

- To summarize for the *Testing Methodology* chapter, the auditors performed a comprehensive threat modeling and risk assessment exercise to identify potential vulnerabilities specific to MetaMask Snaps, for which the vectors deemed most attractive or vulnerable were prioritized in order of impact and likelihood. Mitigation strategies and recommendations are offered forthwith to provide conclusive defense-in-depth.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 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., *SIL-02-001*) to facilitate any future follow-up correspondence.

SIL-02-001 WP1: Sign request popup handles newlines incorrectly *(Low)*

The observation was made that Silent Shard Snap does not correctly handle newlines appearing in the *sign message* confirmation popup inside MetaMask.

Generally, a transaction sign request can contain a custom message to be signed. When this custom message is displayed by MetaMask, newline characters within the message shift the text, which permits a malicious dApp to achieve spoofing and integrate arbitrary text into the rest of the message to be signed.

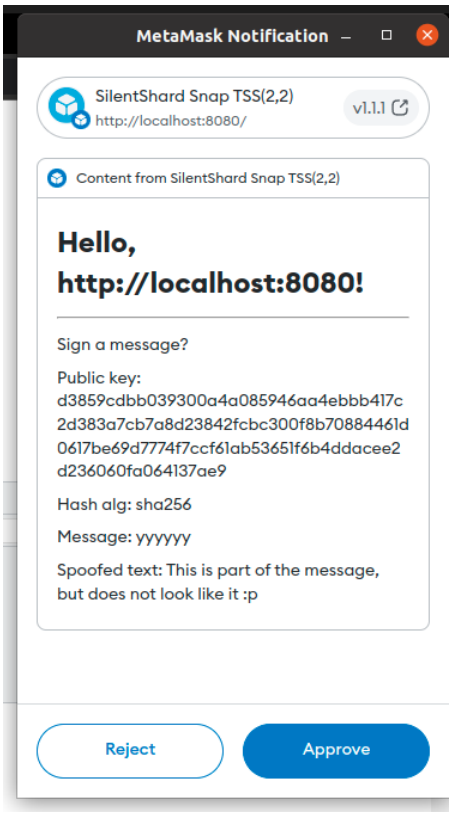

Fig.: Screenshot of spoofed signature message.

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

To mitigate this issue, Cure53 recommends ensuring that messages containing newlines are correctly handled in the app. The developer team should guarantee that messages containing a multitude of newlines are stripped or limited. Likewise, a clear separation between the custom message and remaining transaction information should be imposed.

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.

SIL-02-002 WP1: Multiple vulnerabilities via outdated dependencies *(Info)*

Cure53 noted that the application adopts components with publicly known vulnerabilities from underlying dependencies. Whilst the majority of these weaknesses are likely unexploitable under the current implementation, this behavior is still considered subpar from a security perspective due to the unnecessary persistence of undesirable security flaws. The following table summarizes all publicly known vulnerabilities affecting packages at present, utilized either directly or as an underlying dependency:

Affected project:
https://gitlab.com/com.silencelaboratories/silentshard-snap

Component	Issues	Severity
fast-xml-parser@4.0.9	Regular Expression Denial of Service (ReDoS)[1] Prototype Pollution[2]	High

This specific fault was confirmed by reviewing the following file:

Affected file:
package.json

Affected code:
```
"dependencies": {
[...]
    "crypto-js": "^4.1.1",
    "fast-xml-parser": "^4.0.9",
    "html-webpack-plugin": "^5.5.0",
[...]
},
```

[1] https://security.snyk.io/vuln/SNYK-JS-FASTXMLPARSER-5668858
[2] https://security.snyk.io/vuln/SNYK-JS-FASTXMLPARSER-3325616

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

Affected project:
https://gitlab.com/com.silencelaboratories/shard-metamask-snaps/silentshardnewui/

Component	Issues	Severity
web3@1.10.0	Server-Side Request Forgery in an indirect dependency Request[3]	Medium

Similarly, one can peruse the following file to validate this particular flaw:

Affected file:
package.json

Affected code:
```
"dependencies": {
[...]
        "typescript": "5.1.3",
        "web3": "^1.10.0"
[...]
},
```

To mitigate this issue, Cure53 suggests integrating automated task and/or commit hooks to routinely check for vulnerabilities in dependencies. A plethora of tools are available for this purpose, including the *npm audit* command[4], *Snyk*[5], and the *OWASP Dependency Check* project[6]. An optimal process here would entail regularly operating the integrated tool via an automated job that alerts a lead developer or administrator regarding known vulnerabilities in dependencies. This will help to initiate patching processes for any emerging deficiencies as soon as possible.

[3] https://github.com/advisories/GHSA-p8p7-x288-28g6
[4] https://docs.npmjs.com/cli/v7/commands/npm-audit/
[5] https://snyk.io/
[6] https://owasp.org/www-project-dependency-check/

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

Conclusions

This document's *Conclusions* segment provides an in-depth appraisal of the primary Silent Shard Snap aspects scrutinized by Cure53 during this June-July 2023 assignment. As stipulated in the introduction, the verification can be made that the developer team has implemented near-perfect security practices and functionality to ensure airtight resilience to attacks.

Generally speaking, the codebase was subjected to rigorous review and compromise techniques, which verified commendable performance in minimizing the attack surface. The limited volume of identified issues clearly indicates the development team's successful integration of abundant precautionary measures to safeguard the Silent Shard Snap. The Silence Laboratories team is acutely aware of security errors that typically blight modern systems, considering the evident internal measures that directly serve to resolve them.

Despite the highly constrained attack surface, the Cure53 team evaluated every exposed RPC method accessible in an attempt to uncover any negative security implications. Here, the *tss_sendSignRequest* method exhibited an opportunity for enhancement: the absence of adequate sanitization enables leveraging newlines to spoof the signature's message (specifically for *eth_sign*), which is further discussed in ticket SIL-02-001.

Regarding storage, the adoption of *snap_manageState* for the purpose of storing distributed keys was received with distinction, given stored content is automatically encrypted via a Snap-specific key and automatically decrypted when retrieved.

Supplementary endeavors to locate any deficiencies associated with the interaction between unauthorized dApps and Snap were initiated, though all attempts were blocked by MetaMask's own security checks and ultimately failed.

Key share leakage was pinpointed as a core area of concern due to the major security connotations evoked if facilitated. Likewise, the auditors honed in on the backup functionality since it directly utilizes the keyshare, though Cure53 verified correct encryption prior to uploading to the cloud. In essence, this flow proved unilaterally resistant to security-related damage.

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

In conclusion, following the completion of this security audit, Cure53 garnered a highly favorable impression of the Silent Shard Snap and codebase. The atypically low volume of vulnerabilities and complete negation of any *Critical*, *High*, and even *Medium* fault scenarios compounds this exceptional outcome.

Should the developer team proactively address and mitigate all findings documented in this report, Cure53 would be pleased to confirm that an optimal security foundation has been achieved. The Silence Laboratories team has made significant progress for the products in scope to date. As such, one can argue that the groundwork has already been effectively laid for future enhancements and defensive bolstering.

Cure53 would like to thank Andrei Bytes, Jay Prakash, and Daksh Garg from the Silence Laboratories India Private Limited (SLIPL) team for their excellent project coordination, support, and assistance, both before and during this assignment.