CANTINA

Clave email recovery

Security Review

Cantina Managed review by:
Blockdev, Security Researcher
Windhustler, Security Researcher

October 31, 2024

Contents

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity	Description
Critical	*Must* fix as soon as possible (if already deployed).
High	Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
Medium	Global losses <10% or losses to only a subset of users, but still unacceptable.
Low	Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.
Gas Optimization	Suggestions around gas saving practices.
Informational	Suggestions around best practices or readability.

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Clave is an easy-to-use non-custodial smart wallet powered by Account Abstraction and the Hardware Elements (e.g Secure Enclave, Android Trustzone etc...), offering a unique onboarding process.

From Oct 25th to Oct 27th the Cantina team conducted a review of clave-email-recovery on commit hash 348dce9b. The team identified a total of **4** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 0
- Low Risk: 0
- Gas Optimizations: 2
- Informational: 2

3 Findings

3.1 Gas Optimization

3.1.1 Local variable not used

Severity: Gas Optimization

Context: EmailRecoveryCommandHandler.sol#L78

Description: The function call `hexToBytes32()` is useful for safety checks but `calldataHash` local variable isn't used.

Recommendation: Don't store the return value:

```
- bytes32 calldataHash = StringUtils.hexToBytes32(newOwnerHashInEmail);
+ StringUtils.hexToBytes32(newOwnerHashInEmail);
```

Clave: Fixed in PR 2.

Cantina Managed: Verified.

3.1.2 Keep function argument in `calldata`

Severity: Gas Optimization

Context: EmailRecoveryCommandHandler.sol#L136-L148

Description: Marking function arguments to `memory` wastes gas since data is always copied from `calldata` to `memory`. If possible, it's better to avoid this copy to save gas. Both these functions copy `commandParams` to `memory` when just `commandParams[0]` is used by the function. Thus, copying `commandParams[1]` to memory isn't needed.

Recommendation: Keep these arguments in `calldata`:

```
- bytes[] memory commandParams,
+ bytes[] calldata commandParams,
```

Clave: Fixed in PR 2.

Cantina Managed: Verified.

3.2 Informational

3.2.1 `view` functions can be made `pure`

Severity: Informational

Context: EmailRecoveryCommandHandler.sol#L24, EmailRecoveryCommandHandler.sol#L65, EmailRecoveryCommandHandler.sol#L68

Description: The linked `view` functions can be made `pure` as they don't read state.

Recommendation: Convert these functions to `pure`.

Clave: Fixed in PR 2.

Cantina Managed: Verified.

3.2.2 Change text in recovery command template

Severity: Informational

Context: EmailRecoveryCommandHandler.sol#L124

Description: A more descriptive text for the recovery command template would be the one used in the EmailRecoveryCommandHandler.sol, i.e.

```
Recover account {ethAddr} using recovery hash {string}
```

as the template accepts the hash, not the actual address.

Recommendation:

```
- templates[0][3] = "to";
- templates[0][4] = "owner";
- templates[0][5] = "{string}";
+ templates[0][3] = "using";
+ templates[0][4] = "recovery";
+ templates[0][5] = "hash";
+ templates[0][6] = "{string}";
```

Clave: Fixed in PR 2.

Cantina Managed: Verified.