shieldify

Zaros - Token

SECURITY REVIEW

Date: 22 May 2024

CONTENTS

1. About Shieldify

Positioned as the first hybrid Web3 Security company, Shieldify shakes things up with a unique subscription-based auditing model that entitles the customer to unlimited audits within its duration, as well as top-notch service quality thanks to a disruptive 6-layered security approach. The company works with very well-established researchers in the space and have secured multiple millions in TVL across protocols, also can audit codebases written in Solidity, Vyper, Rust, Cairo, Move and Go.

Learn more about us at shieldify.org.

2. Disclaimer

This security review does not guarantee bulletproof protection against a hack or exploit. Smart contracts are a novel technological feat with many known and unknown risks. The protocol, which this report is intended for, indemnifies Shieldify Security against any responsibility for any misbehavior, bugs, or exploits affecting the audited code during any part of the project's life cycle. It is also pivotal to acknowledge that modifications made to the audited code, including fixes for the issues described in this report, may introduce new problems and necessitate additional auditing.

3. About Zaros - Token

$ZRS is the ticker for Zaros' ERC-20 utility token. By locking ZRS tokens into the protocol for up to one year, holders obtain veZRS tokens.

This grants them voting privileges within the Zaros DAO and entitles them to a portion of the trading fees generated by the leverage trading DEX. The longer the lock duration, the greater the voting power. The fee-sharing mechanism operates proportionally to each veZRS holder's voting power.

4. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4.1 Impact

- **High** - results in a significant risk for the protocol's overall well-being. Affects all or most users
- **Medium** - results in a non-critical risk for the protocol affects all or only a subset of users, but is still unacceptable
- **Low** - losses will be limited but bearable - and covers vectors similar to griefing attacks that can be easily repaired

4.2 Likelihood

- **High** - almost certain to happen and highly lucrative for execution by malicious actors
- **Medium** - still relatively likely, although only conditionally possible
- **Low** - requires a unique set of circumstances and poses non-lucrative cost-of-execution to rewards ratio for the actor

5. Security Review Summary

The security review lasted 1 day with a total of 6 hours dedicated to the audit by three researchers from the Shieldify team.

Overall, the code is well-written and heavily relies on OpenZeppelin's battle-tested contracts. The audit report contributed by identifying an issue with severity on the lower end of the spectrum, concerning proper initialization.

5.1 Protocol Summary

Project Name	**Zaros - Token**
Repository	zaros-token
Type of Project	ERC-20
Audit Timeline	1 day
Review Commit Hash	ef445110dd61fdd6e1024ab68e9385603165bc38
Fixes Review Commit Hash	9d099a9a435e5f13dc626af5628f8fbd37e59636

5.2 Scope

The following smart contracts were in the scope of the security review:

File	nSLOC
src/ZarosToken.sol	28
Total	**28**

6. Findings Summary

The following number of issues have been identified, sorted by their severity:

- **Critical** and **High** issues: **0**
- **Medium** issues: **0**
- **Low** issues: **1**

ID	Title	Severity	Status
[L-01]	Attacker Can Initialize the Implementation	Low	Fixed

7. Findings

[L-01] Attacker Can Initialize the Implementation

Severity

Low Risk

Description

The contracts are upgradable, inheriting from the Initializable contract. However, the current implementations are missing the `_disableInitializers()` function call in the constructors. Thus, an attacker can initialize the implementation. Usually, the initialized implementation has no direct impact on the proxy itself, however, it can be exploited in a phishing attack. In rare cases, the implementation might be mutable and may have an impact on the proxy.

Location of Affected Code

File: src/ZarosToken.sol#L26

Recommendation

It is recommended to call `_disableInitializers()` within the contract's constructor to prevent the implementation from being initialized:

```
+ constructor() {
+     _disableInitializers();
+ }
```

Team Response

Fixed.

shieldify

Thank you!