AUDIT REPORT

Eris Protocol

Phoenix Treasury

Prepared by SCV-Security

On 24th September 2024

Table of Contents

Introduction

SCV has been engaged by Eris Protocol to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

Scope Functionality

The Phoenix Directive Treasury contract facilitates the staking of virtual tokens within Terra Governance, offering a variety of financial management features including Alliance Staking, DCA and OTC swaps, milestone and vesting payments, and flexible payment schedules. It includes veto capabilities to delay payments and enforce spending limits, as well as basic on-chain price oracles. The contract allows for comprehensive querying, including user-specific actions and available balances.

Submitted Codebase

phoenix-treasury	
Repository	https://github.com/erisprotocol/contracts-ve3
Commit	`911ec9f22a793c5cee653f2011cf8eadee1c3f40`
Contract	`phoenix-treasury`
Branch	`main`

Submitted Codebase – Revisions

phoenix-treasury	
Repository	https://github.com/erisprotocol/contracts-ve3
Commit	`2c397031fde53f29e3abbcec250c731311c6e6a1`
Contract	`phoenix-treasury`
Branch	`main`

Methodologies

SCV performs a combination of automated and manual security testing based on the scope of testing. The testing performed is based on the extensive experience and knowledge of the auditor to provide the greatest coverage and value to Eris Protocol Testing includes, but is not limited to, the following:

- Understanding the application and its functionality purpose.
- Deploying SCV in-house tooling to automate dependency analysis and static code review.
- Analyse each line of the code base and inspect application security perimeter.
- Review underlying infrastructure technologies and supply chain security posture.

Code Criteria

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organisation.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorised as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

Criteria	Status	Notes
Documentation	**SUFFICIENT**	N/A
Coverage	**NOT-SUFFICIENT**	Tests were available for critical functionality, though overall coverage remains low at 8.08% (124 out of 1,534 lines of code). There is significant room for improvement in test coverage.
Readability	**SUFFICIENT**	The codebase had good readability overall and utilised many Rust and CosmWasm best practices.
Complexity	**SUFFICIENT**	N/A

Findings Summary

Summary Title	Risk Impact	Status
DCA Functionality May Be Susceptible to Sandwich Attacks	SEVERE	RESOLVED
Incorrect error message returned	INFO	RESOLVED

Audit Observations

The audit observations section is intended to present potential findings that are related to the underlying design of the protocol and would require underlying design changes to remediate that may change the overall functioning of the protocol. SCV asks that the client formulate responses to add context to validate or invalidate the following concerns.

1. Clawback function concerns

The `clawback` function in the contract presents several noteworthy considerations. While it effectively halts most contract operations by setting a global clawback flag, its implementation allows for selective asset retrieval rather than a complete clawback. This partial clawback capability, while flexible, introduces the risk of inadvertently leaving assets remaining in the contract (the clawback can be called multiple times though). Because the clawback state will effectively remove most functionality from the contract it isn't clear why the clawback allows for a partial clawback.

Additionally, the absence of more granular controls and safeguards on this critical function could expose the contract to risks from accidental triggering or single-point-of-failure scenarios. While these aspects don't necessarily constitute vulnerabilities, they represent important design choices that warrant careful consideration and clear documentation to ensure alignment with the contract's intended use and risk profile.

2. Potentially Inconsistent DCA

The `execute_dca` function does not guarantee that a dollar cost average is occurring at a regular interval. While it is enforcing a minimum time period It is relying on callers to perform the DCA action. It is not evident whether or not there is a bot that will trigger this function, but we recommend implementing a bot or incentive solution for ensuring DCA buys are made within the appropriate time interval.

Findings Technical Details

1. DCA Functionality May Be Susceptible to Sandwich Attacks

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `execute_dca` function is permissionless and also allows the caller to specify `min_received`. If the `min_received` is set too low, it could result in significant slippage, especially in illiquid markets or during high volatility periods. Additionally, this increased slippage tolerance could potentially be used in combination with a sandwich attack where a malicious actor could manipulate the asset price and extract value from the contract. It is important to note that the degree of this attack depends on the swap venue that is being used and whether it performs validation on `min_received`.

Recommendation

We recommended calculating a `min_received` value through a contract query to the swap venue within the contract rather than allowing the caller to specify one.

2. Incorrect error message returned

RISK IMPACT: INFORMATIONAL	STATUS: RESOLVED

Description

The `execute_update_milestone` function in `contracts/phoenix-treasury/src/contract.rs:294` incorrectly returns the `CannotExecuteOnlyDca` contract error.

Recommendation

We recommend updating the error type to be specific to the `execute_update_milestone` function.

Document Control

Version	Date	Notes
-	9th September 2024	Security audit commencement date.
0.1	18th September 2024	Initial report with identified findings delivered.
0.5	22nd September 2024	Fixes remediations implemented and reviewed.
1.0	24th September 2024	Audit completed, final report delivered.

Appendices

A. Appendix – Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

Risk Level	Range
CRITICAL	10
SEVERE	From 9 to 8
MODERATE	From 7 to 6
LOW	From 5 to 4
INFORMATIONAL	From 3 to 1

LIKELIHOOD and **IMPACT** would be individually assessed based on the below:

Rate	LIKELIHOOD	IMPACT
5	**Extremely Likely**	Could result in severe and irreparable consequences.
4	**Likely**	May lead to substantial impact or loss.
3	**Possible**	Could cause partial impact or loss on a wide scale.
2	**Unlikely**	Might cause temporary disruptions or losses.
1	**Rare**	Could have minimal or negligible impact.

B. Appendix – Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended for discussion purposes only. The content of this audit report is provided "as is," without representations and warranties of any kind, and SCV-Security disclaims any liability for damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

THANK YOU FOR CHOOSING

SCV SECURITY

scv.services

contact@scv.services