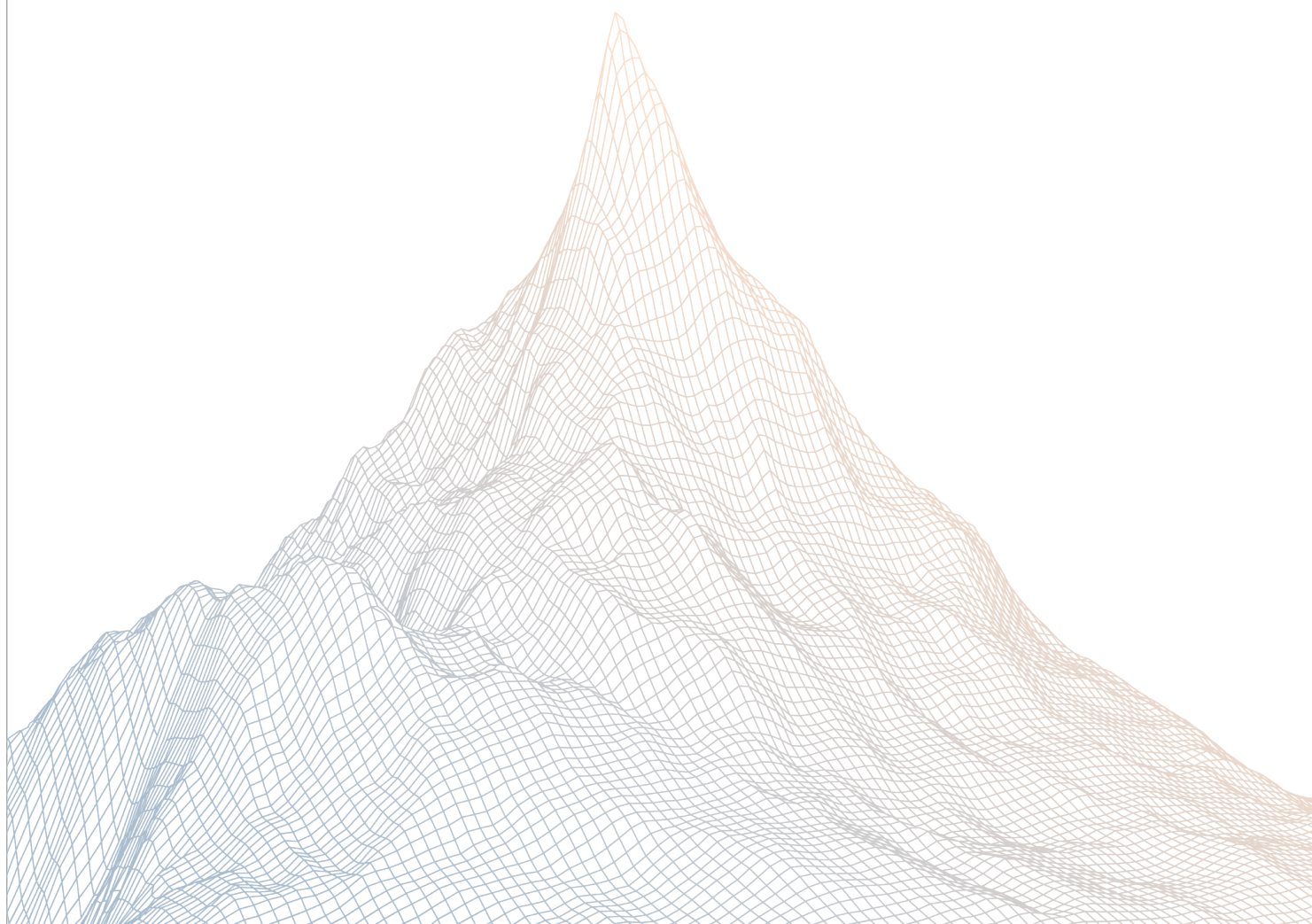


Breath of Estova

Smart Contract Security Assessment

VERSION 1.1



Contents

1	Introduction	2
1.1	About Zenith	3
1.2	Disclaimer	3
1.3	Risk Classification	3
<hr/>		
2	Executive Summary	3
2.1	About Breath of Estova	4
2.2	Scope	4
2.3	Audit Timeline	5
2.4	Issues Found	5
<hr/>		
3	Findings Summary	5
<hr/>		
4	Findings	6
4.1	Informational	7

1

Introduction

1.1 About Zenith

Zenith assembles auditors with proven track records: finding critical vulnerabilities in public audit competitions.

Our audits are carried out by a curated team of the industry's top-performing security researchers, selected for your specific codebase, security needs, and budget.

Learn more about us at <https://zenith.security>.

1.2 Disclaimer

This report reflects an analysis conducted within a defined scope and time frame, based on provided materials and documentation. It does not encompass all possible vulnerabilities and should not be considered exhaustive.

The review and accompanying report are presented on an "as-is" and "as-available" basis, without any express or implied warranties.

Furthermore, this report neither endorses any specific project or team nor assures the complete security of the project.

1.3 Risk Classification

SEVERITY LEVEL	IMPACT: HIGH	IMPACT: MEDIUM	IMPACT: LOW
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

2

Executive Summary

2.1 About Breath of Estova

Breath of Estova is a 2D action-based MMORPG where classic nostalgia meets a vast open world. Choose one of 5 kingdoms each with its own distinctive history, ideology, and motives. Build your legacy and uncover Estova's deepest mysteries—are you ready for the dangers that await?

2.2 Scope

The engagement involved a review of the following targets:

Target	FundTransferContract
Repository	https://github.com/Breath-Of-Estova/FundTransferContract.git
Commit Hash	0ee29f172f5b86d9f34fd2159cbac29d8330fe4b
Files	src/FundTransferVault.sol

2.3 Audit Timeline

December 30, 2025	Audit start
December 31, 2025	Audit end
January 5, 2026	Report published

2.4 Issues Found

SEVERITY	COUNT
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Informational	2
Total Issues	2

3

Findings Summary

ID	Description	Status
I-1	Forced native transfers are not forwarded	Resolved
I-2	Inconsistent handling of O-value transfers	Resolved

4

Findings

4.1 Informational

A total of 2 informational findings were identified.

[I-1] Forced native transfers are not forwarded

SEVERITY: Informational

IMPACT: Informational

STATUS: Resolved

LIKELIHOOD: Low

Target

- [FundTransferVault.sol](#)

Description:

The vault forwards funds only when its `deposit()` or `receive()` logic executes. However, native coins can be forced into a contract without executing its code via `selfdestruct()` from another contract. In that scenario the funds are not forwarded to the admin and remain stuck in the vault balance.

This can lead to discrepancies between the expected invariant `vault should not hold funds` and reality, which can confuse monitoring, accounting, and operational assumptions.

Recommendations:

Consider adding an `onlyOwner` rescue function.

Breath of Estova: Resolved with [@a37b8a5943 ...](#)

Zenith: Verified.

[I-2] Inconsistent handling of 0-value transfers

SEVERITY: Informational

IMPACT: Informational

STATUS: Resolved

LIKELIHOOD: Low

Target

- [FundTransferVault.sol](#)

Description:

`deposit()` rejects calls with `msg.value = 0`, while `receive()` allows them. This creates an inconsistency where a user can trigger the forwarding flow via a direct transfer with 0 value, even though `deposit()` enforces a positive amount.

Recommendations:

Make behavior consistent across entrypoints.

Breath of Estova: Resolved with [@a37b8a5943 ...](#)

Zenith: Verified.