Zellic

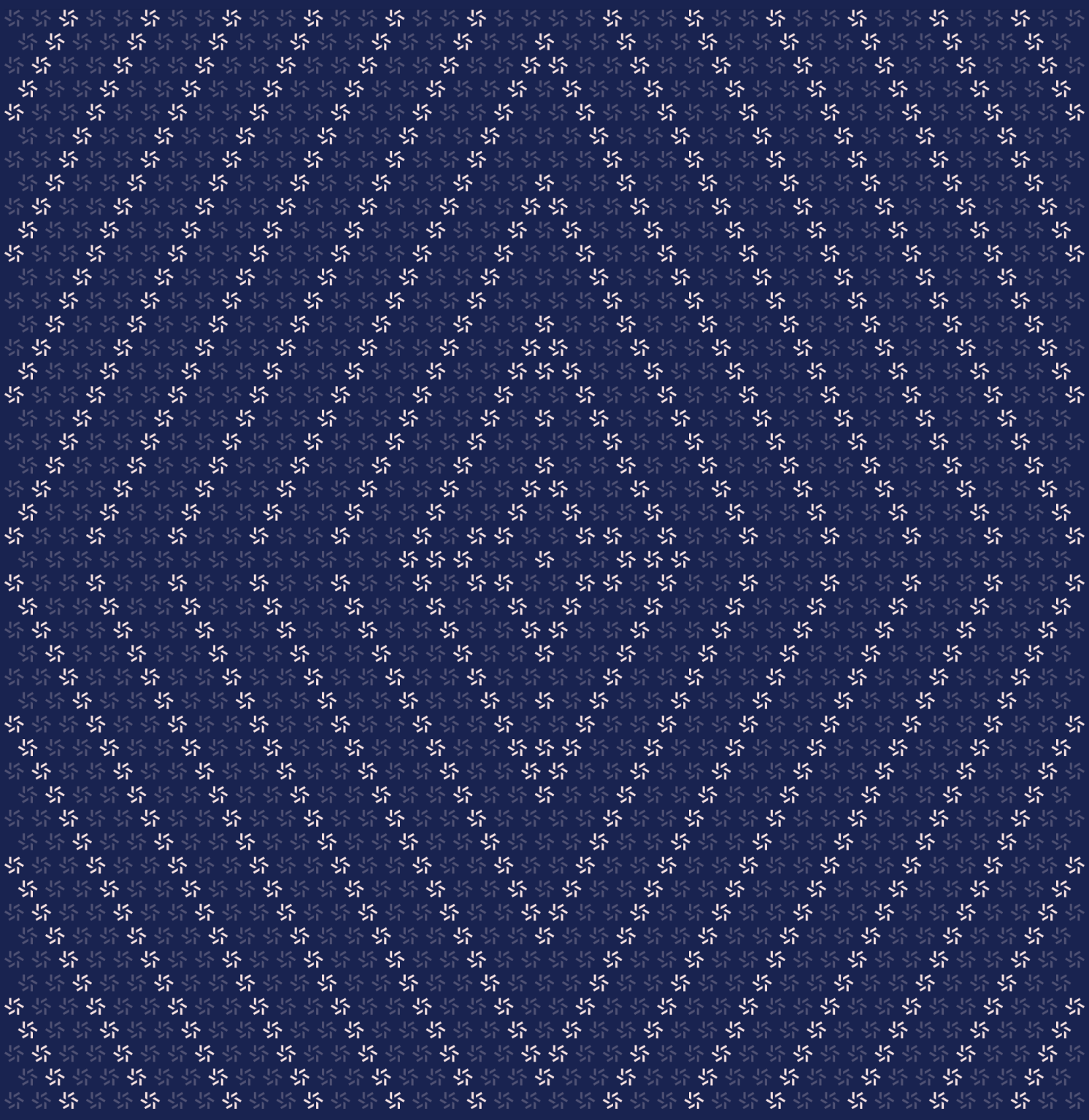
Prepared for

Plume

Prepared by
Jinheon Lee
Doyeon Park
Zellic

February 4, 2025

Cultured
Smart Contract Patch Review



Contents

About Zellic **3**

1. Overview **3**

1.1. Executive Summary 4

1.2. Results 4

2. Introduction **4**

2.1. About Cultured 5

2.2. Scope 5

3. Detailed Findings **6**

3.1. Breaking validation of increasing/decreasing position process 7

4. Discussion **8**

4.1. Patch review 9

5. Assessment Results **9**

5.1. Disclaimer 10

About Zellic

Zellic is a vulnerability research firm with deep expertise in blockchain security. We specialize in EVM, Move (Aptos and Sui), and Solana as well as Cairo, NEAR, and Cosmos. We review L1s and L2s, cross-chain protocols, wallets and applied cryptography, zero-knowledge circuits, web applications, and more.

Prior to Zellic, we founded the #1 CTF (competitive hacking) team ↗ worldwide in 2020, 2021, and 2023. Our engineers bring a rich set of skills and backgrounds, including cryptography, web security, mobile security, low-level exploitation, and finance. Our background in traditional information security and competitive hacking has enabled us to consistently discover hidden vulnerabilities and develop novel security research, earning us the reputation as the go-to security firm for teams whose rate of innovation outpaces the existing security landscape.

For more on Zellic's ongoing security research initiatives, check out our website zellic.io ↗ and follow @zellic_io ↗ on Twitter. If you are interested in partnering with Zellic, contact us at hello@zellic.io ↗.

1. Overview

1.1. Executive Summary

Zellic conducted a security assessment for Plume from January 30th to February 3rd, 2025. During this engagement, Zellic reviewed Cultured's code for security vulnerabilities, design issues, and general weaknesses in security posture.

We note that as requested by Plume, we focused on the changes made between the base Cultured commit `af802121` and the latest Cultured commit as of the time of writing at `95ee340e`.

1.2. Results

During our assessment on the scoped Cultured contracts, we discovered one finding, which was of high impact.

Additionally, Zellic recorded its notes and observations from the assessment for the benefit of Plume in the Discussion section (4.).

Breakdown of Finding Impacts

Impact Level	Count
Critical	0
High	1
Medium	0
Low	0
Informational	0

2. Introduction

2.1. About Cultured

Plume contributed the following description of Cultured:

> Cultured is a framework that allows users to trade on arbitrary data feeds, some of which will correspond very directly to real-world data ("What's the current temperature in NYC?"), and some of which will correspond in a proxied way based on real-time AI analysis of input data from Twitter, Reddit, news, etc. ("What's the sentiment on Donald Trump?"). Unlike prediction markets whose price changes purely based on orders on the platform, these indexes update on a minute-by-minute basis so that traders are always on their toes and forced to react.

2.2. Scope

The engagement involved a review of the following targets:

Cultured Contracts

Type	Solidity
Platform	EVM-compatible
Target	cultured-contracts
Repository	https://github.com/cultured-rwa/cultured-contracts ↗
Version	`Only changes between af802121...95ee340e`
Programs	`Vault` `VaultUtils`

Contact Information

The following project managers were associated with the engagement:

Jacob Goreski
Engagement Manager
jacob@zellic.io

Chad McDonald
Engagement Manager
chad@zellic.io

The following consultants were engaged to conduct the assessment:

Jinheon Lee
Engineer
jinheon@zellic.io

Doyeon Park
Engineer
doyeon@zellic.io

3. Detailed Findings

3.1. Breaking validation of increasing/decreasing position process

Target	VaultUtils.sol		
Category	Coding Mistakes	**Severity**	High
Likelihood	High	**Impact**	High

Description

The `validateIncreasePosition` and `validateDecreasePosition` functions were recently added to handle position validation.

Specifically, `validateIncreasePosition` checks whether a user's position exceeds the `maxPerUserSizeLimit`, which is expected to be set by governance. If the position exceeds this limit, the transaction is reverted.

The primary goal of this validation is to maintain the proportionality between the actual position size and the position size recorded in VaultUtils.

```
function setMaxPerUserSizeLimit(uint256 _maxPerUserSizeLimit)
    external onlyGov {
    maxPerUserSizeLimit = _maxPerUserSizeLimit;
}
```

```
function validateIncreasePosition(address _account, address /*
    _collateralToken */, address  _indexToken, uint256  _sizeDelta, bool /*
    _isLong */) external override {
    if (perUserSizeLimit[_account][_indexToken] + _sizeDelta >
    maxPerUserSizeLimit) {
        revert("Vault: max per user size limit exceeded");
    }
    perUserSizeLimit[_account][_indexToken] += _sizeDelta;
}

function validateDecreasePosition(address _account, address /*
    _collateralToken */, address _indexToken, uint256 /* _collateralDelta */,
    uint256 _sizeDelta, bool /* _isLong */, address /* _receiver */)
    external override {
    perUserSizeLimit[_account][_indexToken] -= _sizeDelta;
}
```

However, the functions lack caller validation so any user can call them.

Impact

This vulnerability can result in two issues: (1) it may disrupt the intended proportion between actual position size and the position size recorded in VaultUtils, and (2) since the functions fully trust the `_account` argument, an attacker could exploit this to block users from increasing their positions.

Recommendations

Implement caller validation to ensure that only `Vault` can invoke these functions.

Remediation

This issue has been acknowledged by Plume, and a fix was implemented in commit `3a3b04be` ↗.

4. Discussion

The purpose of this section is to document miscellaneous observations that we made during the assessment. These discussion notes are not necessarily security related and do not convey that we are suggesting a code change.

4.1. Patch review

The purpose of this section is to document the exact diffs of the codebase that were considered in scope for this audit.

We note that as requested by Plume, we focused on the changes made between the base Cultured commit `af802121` and the latest Cultured commit as of the time of writing at `95ee340e`.

Notable changes

The following were notable changes made to the codebase.

VaultUtils.sol contract updates

- A constant `PRICE_PRECISION` public uint256 and a variable `maxPerUserSizeLimit` public uint256 were added. These values serve to define the maximum allowable value of positions.
- A public mapping `perUserSizeLimit` has been added, which manages users' position size values in relation to the maximum limits set for positions.
- The `setMaxPerUserSizeLimit` function has been added, allowing governance to modify the value of `maxPerUserSizeLimit`.
- The `validateIncreasePosition` function has been updated to restrict the increase of positions by comparing the sum of the user's `perUserSizeLimit` and the `_sizeDelta` against the maximum position limit.
- The `validateDecreasePosition` function has been updated to decrease the value of `perUserSizeLimit` when a position is reduced.

5. Assessment Results

At the time of our assessment, the reviewed code was not deployed to Ethereum Mainnet.

During our assessment on the scoped Cultured contracts, we discovered one finding, which was of high impact.

5.1. Disclaimer

This assessment does not provide any warranties about finding all possible issues within its scope; in other words, the evaluation results do not guarantee the absence of any subsequent issues. Zellic, of course, also cannot make guarantees about any code added to the project after the version reviewed during our assessment. Furthermore, because a single assessment can never be considered comprehensive, we always recommend multiple independent assessments paired with a bug bounty program.

For each finding, Zellic provides a recommended solution. All code samples in these recommendations are intended to convey how an issue may be resolved (i.e., the idea), but they may not be tested or functional code. These recommendations are not exhaustive, and we encourage our partners to consider them as a starting point for further discussion. We are happy to provide additional guidance and advice as needed.

Finally, the contents of this assessment report are for informational purposes only; do not construe any information in this report as legal, tax, investment, or financial advice. Nothing contained in this report constitutes a solicitation or endorsement of a project by Zellic.