Zeta Markets Digital Assessment

Findings and Recommendations Report Presented to:

Zeta Markets

July 08, 2022
Version: 3.0

Presented by:

Kudelski Security, Inc.
5090 North 40th Street, Suite 450
Phoenix, Arizona 85018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Overview

Zeta Markets engaged Kudelski Security to perform a secure code assessment of the Zeta Markets, which is a Decentralized Exchange (DEX) that allows for the use of financial derivative instruments to hedge and speculate on market movements. The secure code assessment covered three main components: The Zeta flex program, which is used to mint covered call options with underlying collateral that also enables an auction mechanism. The flex-vault program is built on top of Zeta flex and is used to store the options underlying collateral. The zeta-options-main is the underlying logic behind the option that leverages Pyth as the oracle for their market data and Serum DEX for their market orderbook.

The assessment was conducted remotely by the Kudelski Security Team. Testing took place February 23rd, 2022- April 6th, 2022, and focused on the following objectives:

- Provide the customer with an assessment of the security of recently added functionality and covered-call option strategy.
- To provide a professional opinion on the maturity, efficiency, and coding practices
- To identify potential issues and include improvement recommendations based on the result of our tests.
- Soundness of financial calculations and security observations of the vault program built on top of Zeta Flex

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the Kudelski Security Teams took to identify and validate each issue, as well as any applicable recommendations for remediation.

Key Findings

The following are the major themes and issues identified during the testing period. These, along with other items, within the findings section, should be prioritized for remediation to reduce to the risk they pose.

- Missing Pyth Versioning Validations

During the test, the following positive observations were noted regarding the scope of the engagement:

- The code is clean, well-documented, and uses many best security practices.
- Adhered to Anchor workflow recommendation to test using integration tests in another language other than Rust

Scope and Rules of Engagement

Kudelski performed an Zeta Markets Digital Assessment for Zeta Markets. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

In-Scope Contracts	
zeta-flex-main	Commit: 51788500d4f75b72ff9bed88f5024176e0fba43f
zeta-vault-main	Commit: 9b731ea165e051064c3be30efa8ee3ddc3e4ca24
zeta-options-main	Commit: f68d5b686ea47e8538759102d4820c149e2b4b6d

Table 1: Scope

TECHNICAL ANALYSIS & FINDINGS

During the Zeta Markets Digital Assessment, we discovered 2 low severity findings. The remaining 2 findings were informational.

The following chart displays the findings by severity.

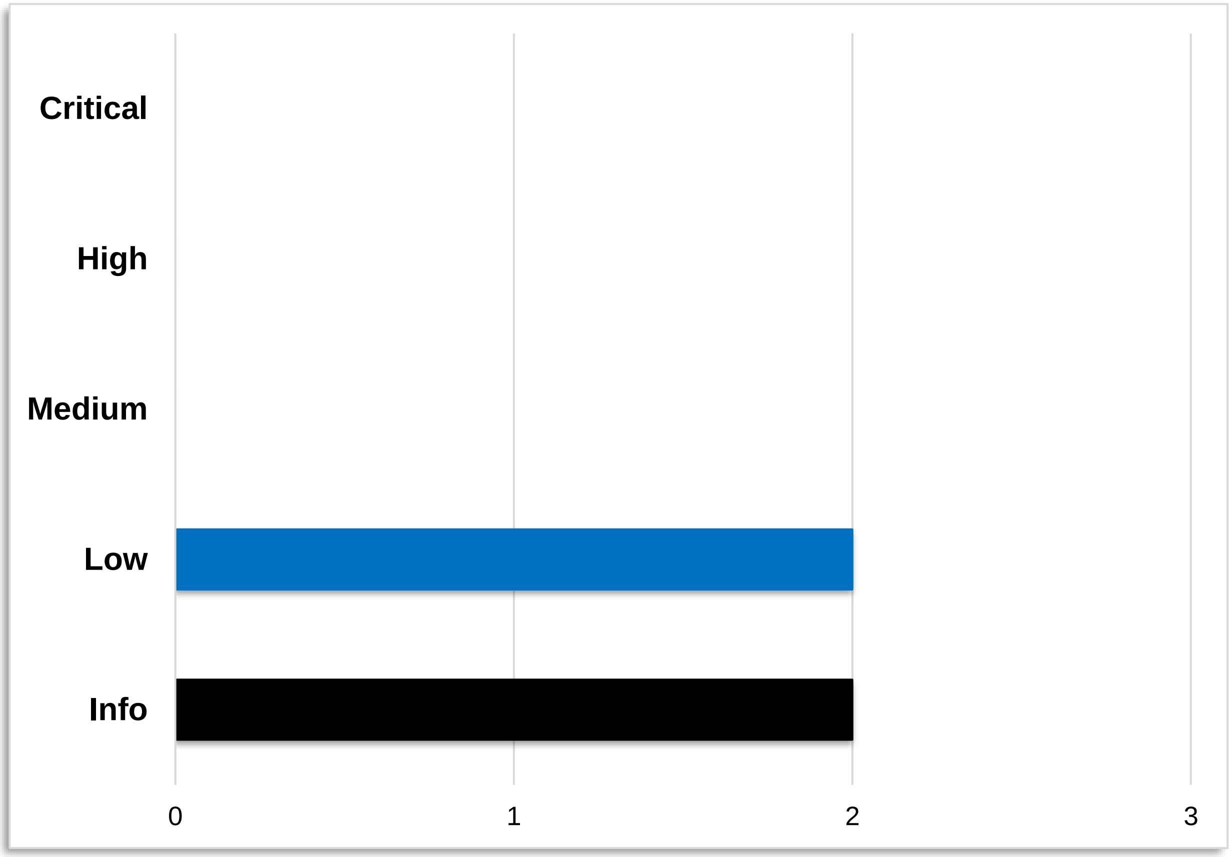

Figure 1: Findings by Severity

Findings

The *Findings* section provides detailed information on each of the findings, including methods of discovery, explanation of severity determination, recommendations, and applicable references.

The following table provides an overview of the findings.

#	Severity	Description
1	**Low**	Missing Pyth Versioning Validations
2	**Low**	Vulnerable Check_Admin() Validation
3	**Informational**	Unchecked Anchor Code
4	**Informational**	Close_margin_account() Messaging

Table 2: Findings Overview

1 – Missing Pyth Versioning Validations

Severity	Low

Impact	Likelihood	Difficulty
Medium	Low	High

Description

The code reviewed contained missing Pyth versioning validations when consuming any data from Pyth oracle feeds.

2 – Vulnerable Check_Admin() Validation

Severity	Low

Impact	Likelihood	Difficulty
Low	Low	High

Description

The check_admin function validated the admin by comparing a public key to the state.admin variable, but did not check whether the state inputs were validated.

3 – Unchecked Anchor Code

Severity	Informational

Impact	Likelihood	Difficulty
N/A	N/A	N/A

Description

Anchor “AccountInfo” types perform no account checks, but this is not obvious from the name “AccountInfo”. The “UncheckedAccount” wrapper should be used to emphasize that no checks are performed.

4 – Misleading Close_margin_account() function

Severity	**Informational**

Impact	**Likelihood**	**Difficulty**
N/A	N/A	N/A

<u>Description</u>

The close_margin_account function didn’t provide additional messaging detailing all actions being performed when closing an account. Close actions include setting account discriminator to a special variant, zeroing out lamports, etc.

METHODOLOGY

During this source code review, the Kudelski Security Services team reviewed code within the project within an appropriate IDE. During every review, the team spends considerable time working with the client to determine correct and expected functionality, business logic, and content to ensure that findings incorporate this business logic into each description and impact. Following this discovery phase the team works through the following categories:

- Authentication
- Authorization and Access Control
- Auditing and Logging
- Injection and Tampering
- Configuration Issues
- Logic Flaws
- Cryptography

These categories incorporate common vulnerabilities such as the OWASP Top 10

Tools

The following tools were used during this portion of the test. A link for more information about the tool is provided as well.

- Visual Studio 2019
- Visual Studio 2022
- Visual Studio Code
- Semgrep

Vulnerability Scoring Systems

Kudelski Security utilizes a vulnerability scoring system based on impact of the vulnerability, likelihood of an attack against the vulnerability, and the difficulty of executing an attack against the vulnerability based on a high, medium, and low rating system

Impact
The overall effect of the vulnerability against the system or organization based on the areas of concern or affected components discussed with the client during the scoping of the engagement.

> **High:**
> The vulnerability has a severe affect on the company and systems or has an affect within one of the primary areas of concern noted by the client
>
> **Medium:**
> It is reasonable to assume that the vulnerability would have a measurable affect on the company and systems that may cause minor financial or reputational damage.
>
> **Low:**
> There is little to no affect from the vulnerability being compromised. These vulnerabilities could lead to complex attacks or create footholds used in more severe attacks.

Likelihood
The likelihood of an attacker discovering a vulnerability, exploiting it, and obtaining a foothold varies based on a variety of factors including compensating controls, location of the application, availability of commonly used exploits, and institutional knowledge

> **High:**
> It is extremely likely that this vulnerability will be discovered and abused
>
> **Medium:**
> It is likely that this vulnerability will be discovered and abused by a skilled attacker
>
> **Low:**
> It is unlikely that this vulnerability will be discovered or abused when discovered.

Difficulty
Difficulty is measured according to the ease of exploit by an attacker based on availability of readily available exploits, knowledge of the system, and complexity of attack. It should be noted that a LOW difficulty results in a HIGHER severity.

> **Low:**
> The vulnerability is easy to exploit or has readily available techniques for exploit
>
> **Medium:**
> The vulnerability is partially defended against, difficult to exploit, or requires a skilled attacker to exploit.
>
> **High:**
> The vulnerability is difficult to exploit and requires advanced knowledge from a skilled attacker to write an exploit

Severity
Severity is the overall score of the weakness or vulnerability as it is measured from Impact, Likelihood, and Difficulty

KUDELSKI SECURITY CONTACTS

NAME	POSITION	CONTACT INFORMATION
Miles Nolan	**Senior Blockchain Security Analyst**	**miles.nolan@kudelskisecurity.com**
Porter Adams	**Senior Blockchain Security Analyst**	**porter.adams@kudelskisecurity.com**
Ken Toler	**Director of Application Security and Blockchain**	**ken.toler@kudelskisecurity.com**
Shannon Garcia	**Head of Offensive Security and Blockchain**	**shannon.garcia@kudelskisecurity.com**