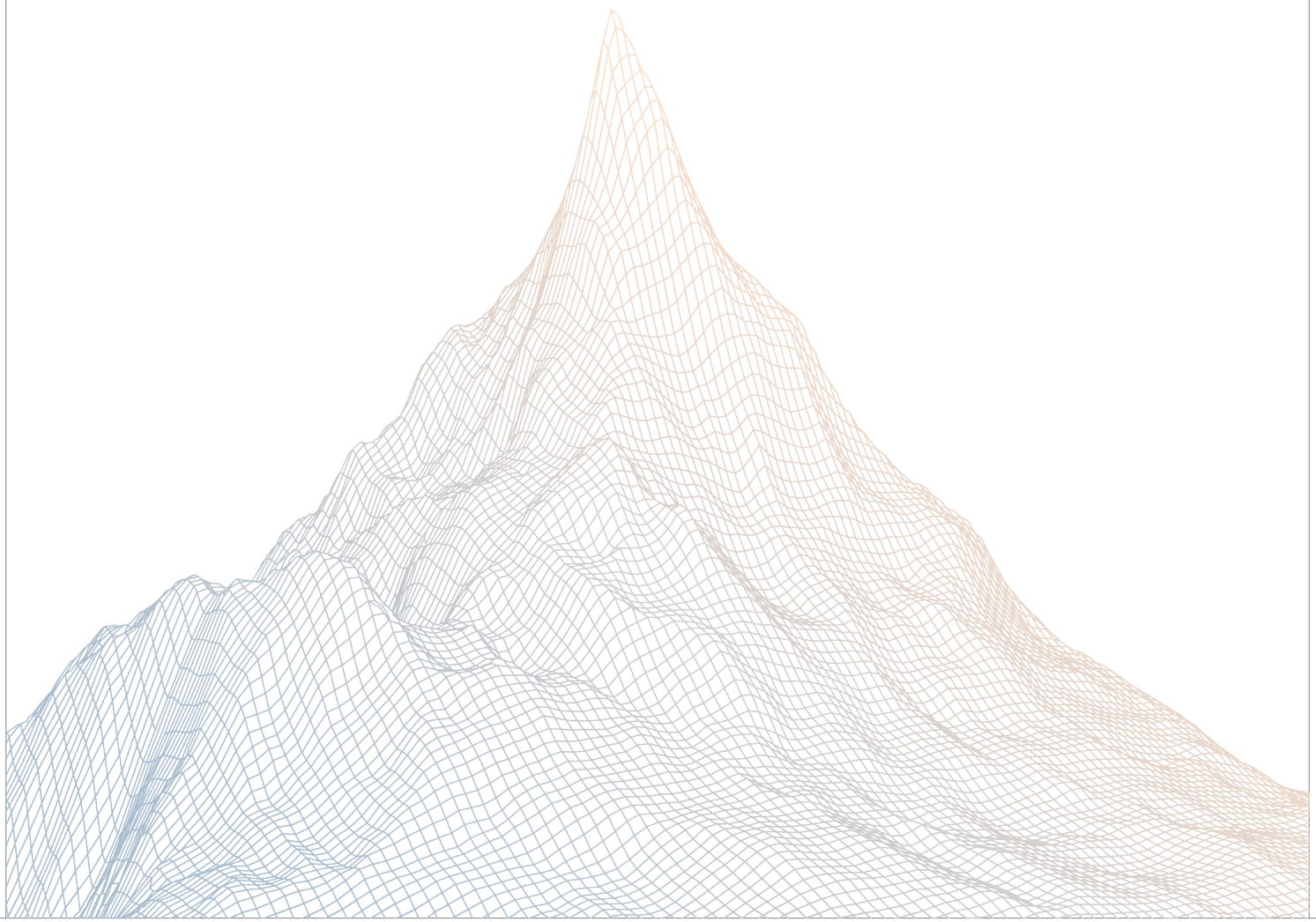
Zenith

MORE Markets

Smart Contract Security Assessment

VERSION 1.1



AUDIT DATES: February 25th to February 27th, 2025

AUDITED BY: defsec
said

Contents

1 Introduction

1.1 About Zenith

Zenith is an offering by Code4rena that provides consultative audits from the very best security researchers in the space. We focus on crafting a tailored security team specifically for the needs of your codebase.

Learn more about us at https://code4rena.com/zenith.

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 More Markets

MORE Markets is a decentralized lending protocol that lets users easily lend and borrow digital assets. It offers features like flash loans, collateral swaps and custom markets offering the ability to manage risk by isolating chosen assets. These tools help users make the most of their digital assets without needing a middleman.

The protocol is designed for permissionless market creation, removing any need for approval or oversight from a central authority. This means anyone can use the factory contract to deploy and update market parameters. The flexibility of market management enables borrowers to design innovative strategies, while offering lenders and asset managers access to sustainable yield.

2.2 Scope

The engagement involved a review of the following targets:

Target	MORE-Markets
Repository	https://github.com/MoreLabsXYZ/MORE-Markets
Commit Hash	`d7d55ea9516316fea7c16244a6f33b47df7b0b4d`
Files	`AAVE fork. Confirm changes.`

2.3 Audit Timeline

February 25, 2025	Audit start
February 27, 2025	Audit end
March 12, 2025	Report published

2.4 Issues Found

SEVERITY	COUNT
Critical Risk	0
High Risk	1
Medium Risk	1
Low Risk	1
Informational	3
Total Issues	**6**

3

Findings Summary

ID	Description	Status
H-1	stableBorrowRateEnabled Should Only Be Enabled for Stablecoins	Resolved
M-1	Lack of initial deposit in the new market listing	Acknowledged
L-1	Borrow and supply caps are not set for all assets	Acknowledged
I-1	Pyth Price Oracle might break some assumptions	Acknowledged
I-2	Assuming flowFeed decimals in AnkrRatioFeedWrapper may cause issues	Resolved
I-3	Upgrade privileged roles to multi-sig after the deployment	Acknowledged

4

Findings

4.1 High Risk

A total of 1 high risk findings were identified.

[H-1] `stableBorrowRateEnabled` Should Only Be Enabled for Stablecoins

`SEVERITY:` High	`IMPACT:` High
`STATUS:` Resolved	`LIKELIHOOD:` Medium

Target

- reservesConfig.ts#L17

Description:

The current implementation of the lending strategies allows the `stableBorrowRateEnabled` property to be set to `true` for both stablecoins and volatile assets. Specifically, the `strategyUSDCe` (a stablecoin strategy) correctly has `stableBorrowRateEnabled` set to `true`, while the `strategyCbBtc` (a volatile asset strategy) also has this property enabled. This could lead to confusion and unintended consequences, as stable borrowing rates should only apply to stablecoins.

- **strategyUSDCe**: `stableBorrowRateEnabled: true`
- **strategyCbBtc**: `stableBorrowRateEnabled: true`

Recommendations:

Set `stableBorrowRateEnabled` to `true` only for stablecoin strategies and set it to `false` for all volatile asset strategies.

More Markets: According to this issue:
https://governance.aave.com/t/aave-v2-v3-security-incident-04-11-2023/15335

We have decided to disable stable borrows for all assets that will be listed. All existing ones alos already have this feature disabled.

Zenith: Resolved through on-chain contracts.

4.2 Medium Risk

A total of 1 medium risk findings were identified.

[M-1] Lack of initial deposit in the new market listing

SEVERITY: Medium	IMPACT: Medium
STATUS: Acknowledged	LIKELIHOOD: Medium

Target

- reservesConfig.ts

Description:

The new market listing script does not include or bundle a small initial deposit. This could make the new market prone to the known empty market exploit.

Recommendations:

On the mainnet, consider always bundling the listing with a small initial deposit to avoid the empty market issue. For reference, this is the Aave's deployment transaction : here.

More Market:

Acknowledged. On the new market creation it will be bundled.

4.3 Low Risk

A total of 1 low risk findings were identified.

[L-1] Borrow and supply caps are not set for all assets

SEVERITY: Low	IMPACT: Low
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- reservesConfig.ts

Description:

Supply and borrow caps for all assets are currently not configured. This is reflected in `reservesConfig.ts` and can also be verified in the deployed configuration here.

Caps limit the maximum amount that can be supplied or borrowed for a specific asset, reducing exposure to risks such as price manipulation, oracle failures, and market crashes. Additionally, certain assets, especially less liquid or volatile ones, could pose a systemic risk if borrowing is unrestricted. Caps help prevent a single asset from destabilizing the protocol.

Recommendations:

Consider assessing each asset and setting the supply and borrow caps accordingly.

More Markets: Acknowledged

4.4 Informational

A total of 3 informational findings were identified.

[I-1] Pyth Price Oracle might break some assumptions

SEVERITY: Informational	IMPACT: Informational
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- PythAggregatorV3.sol

Description:

Ptyh is a pull oracle where anyone can permissionlessly update the on-chain price. It is important to note that since the Pyth oracle's price updates are unrestricted (with the only condition being that the timestamp in the payload is higher), a user can make this oracle return two different prices in the same transaction.

This means, for instance, a user can open a borrow position, then price is updated, causing it to become liquidatable and get liquidated within the same transaction.

Recommendations:

It is important to evaluate all potential risks associated with this Pyth oracle behavior and determine whether they are acceptable.

More Markets:

Acknowledged. We are using app called Pyth Price Scheduler, to push new prices frequently enough. Here is the reference:
https://docs.pyth.network/price-feeds/schedule-price-updates/using-scheduler

Essentially, it updates prices on the chain when any of the following conditions are met: the threshold time has passed or the difference in price from the actual price is more than X%, etc.

[I-2] Assuming `flowFeed` decimals in `AnkrRatioFeedWrapper` may cause issues

SEVERITY: Informational	IMPACT: Informational
STATUS: Resolved	LIKELIHOOD: Low

Target

- AnkrRatioFeedWrapper.sol#L23-L25
- AnkrRatioFeedWrapper.sol#L100

Description:

Inside `AnkrRatioFeedWrapper`, the decimals used are 8 to match the decimals used by `flowFeed`. However, the decimals are hardcoded, assuming that the `flowFeed` decimals will never change.

```
function decimals() public view virtual returns (uint8) {
  return uint8(8);
}
```

```
  function _getPriceOfAnkrFlowInUsd() internal view returns (int256) {
    int256 flowAnkrFlowPrice = int256(_getSafeRatioForAnkrFlow());
    int256 flowPriceInUsd = flowFeed.latestAnswer();

    // Since AnkrRatioFeed returns price of 1 FLOW in AnkrFlow, we have to
    convert it.
    // Price of AnkrFlow in FLOW is
    //
    // ankrFlowFlowPrice = 1 / flowAnkrFlowPrice
    //
    // Then to convert it to USD we need to multiply it with price of FLOW in
    USD
    //
    // flowPriceInUsd * ankrFlowFlowPrice
    //
    // or
    //
    // flowPriceInUsd / flowAnkrFlowPrice
```

```
    // Decimal of flowPriceInUsd is 8, that means that we need to convert it
    to 18 decimal
    // by multiplying by 10**(18 - 8) and then multiply by 10**8 to safe
    precision and keep
    // decimal of usd.
>>> return (flowPriceInUsd * 10 ** 10 * 10 ** 8) / flowAnkrFlowPrice;
  }
```

If the `flowFeed` decimals are changed or updated, the scaling calculation will result in an incorrect price value.

Recommendations:

Use `flowFeed` decimals instead of hardcoded value.

```
  function decimals() public view virtual returns (uint8) {
    return uint8(8);
    return flowFeed.decimals();
  }
```

Additionally, the price calculation can be simplified as follows:

```
  function _getPriceOfAnkrFlowInUsd() internal view returns (int256) {
    int256 flowAnkrFlowPrice = int256(_getSafeRatioForAnkrFlow());
    int256 flowPriceInUsd = flowFeed.latestAnswer();

    // Since AnkrRatioFeed returns price of 1 FLOW in AnkrFlow, we have to
    convert it.
    // Price of AnkrFlow in FLOW is
    //
    // ankrFlowFlowPrice = 1 / flowAnkrFlowPrice
    //
    // Then to convert it to USD we need to multiply it with price of FLOW in
    USD
    //
    // flowPriceInUsd * ankrFlowFlowPrice
    //
    // or
    //
    // flowPriceInUsd / flowAnkrFlowPrice

    // Decimal of flowPriceInUsd is 8, that means that we need to convert it
    to 18 decimal
```

```
    // by multiplying by 10**(18 - 8) and then multiply by 10**8 to safe
    precision and keep
    // decimal of usd.
    return (flowPriceInUsd * 10 ** 10 * 10 ** 8) / flowAnkrFlowPrice;
    return (flowPriceInUsd * 10 ** 18 / flowAnkrFlowPrice;
  }
```

More Markets: Resolved with @957bd3e44...

Zenith: Verified.

[I-3] Upgrade privileged roles to multi-sig after the deployment

SEVERITY: Informational	IMPACT: Informational
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- 0xC8419191Cb1A3bF4FfC022D01f857D5AFdeD01ba

Description:

In the deployment, the market owner, emergency admin, and pool admin roles are currently assigned to a single deployer address.

This configuration centralizes control, increasing the risk of unauthorized access or misuse. By transitioning these roles to a multi-sig wallet, the protocol can distribute control among multiple parties, reducing the risk of single-point failures.

Location : addresses

Recommendations:

Transition the market owner, emergency admin, and pool admin roles to multi-sig wallets. This will require multiple approvals for critical actions.

More Market: Acknowledged. We are planning to move these roles to multisig wallet and will do this asap.