money-market-contracts (dynamic-anchor-rate PR) - Report

Prepared for Anchor Protocol, 24 March 2022

Table of Contents

Introduction

SCV was engaged by Anchor Protocol to assist in identifying security threats and vulnerabilities that have the potential to affect their security posture. Additionally, SCV will assist the team in understanding the risks and identifying potential mitigations.

Scope

SCV performed the security assessment on the following *pull request*:

- https://github.com/Anchor-Protocol/money-market-contracts/pull/66

During the test engagement additional code were committed which remediation were applied at the same *PR* during the same week as this engagement was due.

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 Anchor Protocol. Testing includes, but is not limited to, the following:

- Understanding the application and its code base 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 and Test Coverage

SCV used a scale from **0** to **10** that represents how **SUFFICIENT(6-10)** or **NOT SUFFICIENT(0-5)** each code criteria was during the assessment:

Criteria	Status	Scale Range	Notes
Provided Documentation	**Not Sufficient**	3-4	N\A
Code Coverage Test	**Sufficient**	8-9	N\A
Code Readability	**Sufficient**	7-8	N\A
Code Complexity	**Sufficient**	6-7	N\A

Conclusion

The identified vulnerabilities pose a *low* to *no* level of technical risk to the Anchor Protocol. The provided codebase implements *dynamic rate* and has been implemented securely.

Vulnerabilities Summary

	Title and Summary	Risk	Status
1	Potential division by zero case can cause a Panic	**Low**	**Remediated**
2	Defined variable not being used	**Informational**	**Remediated**
3	English word typos found in the codebase	**Informational**	**Remediated**
4	Epoch time is calculated using past blocks heights	**Informational**	**Acknowledged**
5	Leftover TODOs in comments affects code overall quality.	**Informational**	**Pending**
6	Unnecessary use of Canonical address transformations adds complexity	**Informational**	**Acknowledged**

Detailed Vulnerabilities

Vulnerability 1: Potential division by zero case can cause a Panic

Likelihood	Impact	Risk
Rare	Low	**Low**

Notes

Anchor Protocol team remediated the issue in a subsequent PR.

Description

In the file `contract.rs#362` there is a `blocks_count` method that calculates the elapsed time between current block `(env.block.height)` and `dynrate_state.last_executed_height` by checking its difference.

```
// passed time from the last executed time
let blocks_count = Uint256::from(env.block.height - dynrate_state.
    last_executed_height);
```

The result of that operation can be zero, which would lead to an error and consequential panic when `contract.rs#378` is reached.

```
// yr change per block
let mut yield_reserve_change_pb = yield_reserve_change / Decimal256::
    from_uint256(blocks_count);
```

Recommendations

Ensure there is a check `.is_zero()` on `blocks_count` before performing arithmetic division operations.

Vulnerability 2: Defined variable not being used

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

The variable `_year` in the file `contract.rs#398` is defined but not being used anywhere along in the code.

Recommendations

Ensure defined variables are used if they are needed. If they are not needed, they can be removed from the codebase.

Vulnerability 3: English word typos found in the codebase

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

The codebase contains typos in comments that can affect overall code quality and makes it more difficult to understand.

- `confing` at `contract.rs#397` should be `config`;
- `outsite` at `contract.rs#422` should be `outside`;

Recommendations

Rename words to the correct values.

Vulnerability 4: Epoch time is calculated using past blocks heights

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

By design Anchor Protocol computes `epoch` time from past blocks heights which does not imply a direct security risk since it's securely implemented. However, using `time.seconds()` approach would be a more reliable source of measuring elapsed time.

Recommendations

SCV suggests the use of `env.block.time.seconds()` to calculated elapsed time.

Vulnerability 5: Leftover TODOs in comments affects code overall quality.

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

The codebase contains references to leftover `TODOs` that affect code the overall quality. The leftover comment was found in the `contract.rs#L570`.

Recommendations

Remove `TODO` references if unnecessary or implement them.

Vulnerability 6: Unnecessary use of Canonical address transformations adds complexity

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

Using Canoncial address transformations is no longer encouraged due its lack of efficiency in most cases. In fact, Canoncial transformations no longer saves addresses in canoncial format into the storage. The use of the Canoncial transformation has no directly security implications, although it adds an unnecessary complexity to the codebase due to back and forth transformation, which could be optimized and simplified.

Recommendations

It's recommended to use `Addr` type for addresses input validations and removing all unnecessary references to Canoncial transformations.

More information can be found in the URL link below:

- https://docs.cosmwasm.com/docs/0.16/architecture/addresses/

Document control

Document changes

Version	Date	Name	Changes
0.1	2022-03-17	Vinicius Marino	Initial report
0.2	2022-03-23	Vinicius Marino	Team communication and Pre-Release
1.0	2022-03-24	Vinicius Marino	Final report release

Document contributors

Name	Role	Email address
Vinicius Marino	Security Specialist	vini@scv.services

Appendices

Appendix A: Report Disclaimer

The content of this audit report is provided “As is”, without representations and warranties of any kind.

The author and their employer disclaim any liability for damage arising out of, or in connection with, this audit report.

Appendix B: Risk assessment methodology

A qualitative risk assessment is performed on each vulnerability to determine the impact and likelihood of each.

Risk rate will be calculated on a scale. As per criteria Likelihood vs Impact table below:

Impact \ Likelihood	Rare	Unlikely	Possible	Likely
Critical	Medium	High	Critical	Critical
Severe	Low	Medium	High	High
Moderate	Low	Medium	Medium	High
Low	Low	Low	Low	Medium
Informational	Informational	Informational	Informational	Informational

LIKELIHOOD:

- **Likely**: likely a security incident will occur;
- **Possible**: It is possible a security incident can occur;
- **Unlikely**: Low probability a security incident will occur;
- **Rare**: In rare situations, a security incident can occur;

IMPACT:

- **Critical**: May cause a significant and critical impact;
- **Severe**: May cause a severe impact;
- **Moderate**: May cause a moderated impact;
- **Low**: May cause low or none impact;
- **Informational**: May cause very low impact or none.


SCV