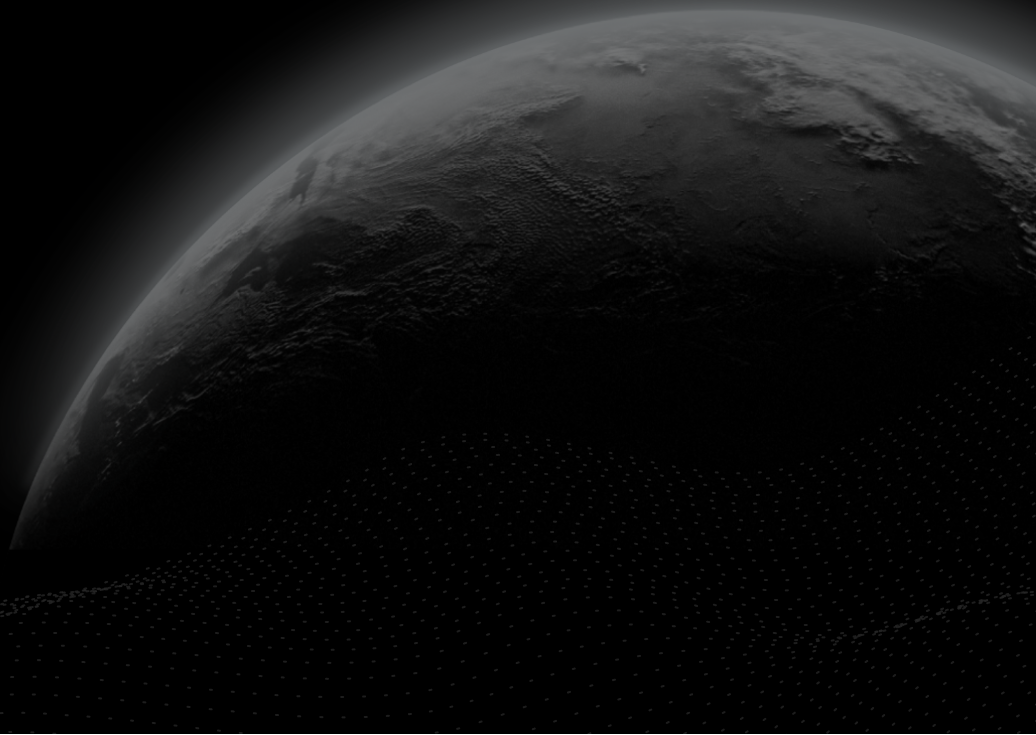
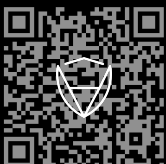




Security Assessment

Mind Network - Audit

CertiK Assessed on Jul 11th, 2024





Certik Assessed on Jul 11th, 2024

Mind Network - Audit

The security assessment was prepared by Certik, the leader in Web3.0 security.

Executive Summary

TYPES

DeFi

ECOSYSTEM

EVM Compatible

METHODS

Formal Verification, Manual Review, Static Analysis

LANGUAGE

Solidity

TIMELINE

Delivered on 07/11/2024

KEY COMPONENTS

N/A

CODEBASE

<https://github.com/mind-network/mind-restaking-contracts/>

View All in Codebase Page

COMMITTS

- [6e66b834426f06e7323af9446e7bf649abd52b60](#)
- [719d74389afaf5a0f5863f981837d4674c153b3f](#)
- [9feee4fc271cad0acbd9f630f3c19d3b2416e5d7](#)

View All in Codebase Page

Highlighted Centralization Risks

⚠ Contract upgradeability

⚠ Privileged role can remove users' tokens

⚠ Withdraws can be disabled

⚠ Privileged role can mint tokens

Vulnerability Summary



6

Total Findings

2

Resolved

2

Mitigated

0

Partially Resolved

2

Acknowledged

0

Declined

■ 0 Critical

Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.

■ 2 Major

2 Mitigated



Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.

■ 1 Medium

1 Resolved



Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.

■ 3 Minor

1 Resolved, 2 Acknowledged



Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.

■ 0 Informational

Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | MIND NETWORK - AUDIT

I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

I **Review Notes**

[Overview](#)

[External Dependencies](#)

[Addresses](#)

[Privileged Functions](#)

I **Findings**

[STR-01 : Centralized Control of Contract Upgrade](#)

[STR-02 : Centralization Risks](#)

[STR-03 : Unprotected Upgradeable Contract](#)

[STR-04 : Potential Zero Share](#)

[STR-05 : Missing Emit Events](#)

[STR-06 : Incompatibility With Deflationary Tokens](#)

I **Appendix**

I **Disclaimer**

CODEBASE | MIND NETWORK - AUDIT

Repository




<https://github.com/mind-network/mind-restaking-contracts/>

Commit

- [6e66b834426f06e7323af9446e7bf649abd52b60](#)
- [719d74389afaf5a0f5863f981837d4674c153b3f](#)
- [9feee4fc271cad0acbd9f630f3c19d3b2416e5d7](#)

AUDIT SCOPE | MIND NETWORK - AUDIT

3 files audited ● 1 file with Acknowledged findings ● 2 files with Mitigated findings

ID	Repo	File	SHA256 Checksum
● STR	mind-network/mind-restaking-contracts	 contracts/strategies/Strategy.sol	63ba4abff5ba46940931797351312faf0d456a5ec0c2515f974c1f50661dcb77
● FSB	mind-network/mind-restaking-contracts	 contracts/strategies/FixedStrategy.sol	43c9d8b964d8cf098d9c2206b224bde0bab29235279be8869b31be0ae6ed8bd6
● MTB	mind-network/mind-restaking-contracts	 contracts/strategies/MToken.sol	3f656c8e8a30f82efb8356fd69957125fd8a3ff71003c2123752e62aad6b72d1

APPROACH & METHODS | MIND NETWORK - AUDIT

This report has been prepared for Mind Network to discover issues and vulnerabilities in the source code of the Mind Network - Audit project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Formal Verification, Manual Review, and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | MIND NETWORK - AUDIT

Overview

The **Mind Network** project manages a suite of smart contracts intended for an upgradeable and pausable investment strategy for tokenized staking on its platform. It allows users to deposit and redeem tokens, with optional lock-up periods, and handles asset-to-share conversions. Features include maximum deposit and redeem limits, total asset caps, and emergency withdrawal of non-strategy tokens by the owner.

External Dependencies

In **Mind Network**, the module inherits or uses a few of the depending injection contracts or addresses to fulfill the need of its business logic. The scope of the audit treats third party entities as black boxes and assume their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets.

Addresses

The following addresses interact at some point with specified contracts, making them an external dependency. All of following values are initialized either at deploy time or by specific functions in smart contracts.

Strategy:

- `assetToken` , `shareToken` , `token` .

We assume these contracts or addresses are valid and non-vulnerable actors and implementing proper logic to collaborate with the current project.

Also, the following libraries/contracts are considered as third-party dependencies:

- `@openzeppelin/contracts/`
- `@openzeppelin/contracts-upgradeable/`

Privileged Functions

In the **Mind Network** project, the privileged roles are adopted to ensure the dynamic runtime updates of the project, which are specified in the following finding: `Centralization Risks` .

The advantage of those privileged roles in the codebase is that the client reserves the ability to adjust the protocol according to the runtime required to best serve the community. It is also worth noting the potential drawbacks of these functions, which should be clearly stated through the client's action/plan. Additionally, if the private keys of the privileged accounts are compromised, it could lead to devastating consequences for the project.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community.

Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of the

`Timelock` contract.

FINDINGS | MIND NETWORK - AUDIT



6

Total Findings

0

Critical

2

Major

1

Medium

3

Minor

0

Informational

This report has been prepared to discover issues and vulnerabilities for Mind Network - Audit. Through this audit, we have uncovered 6 issues ranging from different severity levels. Utilizing the techniques of Formal Verification, Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
STR-01	Centralized Control Of Contract Upgrade	Centralization	Major	● Mitigated
STR-02	Centralization Risks	Centralization	Major	● Mitigated
STR-03	Unprotected Upgradeable Contract	Logical Issue	Medium	● Resolved
STR-04	Potential Zero Share	Incorrect Calculation	Minor	● Acknowledged
STR-05	Missing Emit Events	Volatile Code	Minor	● Resolved
STR-06	Incompatibility With Deflationary Tokens	Volatile Code	Minor	● Acknowledged

STR-01 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	● Major	contracts/strategies/MToken.sol: 10; contracts/strategies/Strategy.sol: 18	● Mitigated

Description

`MToken` and `Strategy` serve as implementation contracts within the upgradeable framework. The `admin` role of the proxy contract in this framework possesses the authority to upgrade the implementation contracts that the proxy contracts point to.

Any compromise to the `admin` account may allow a hacker to take advantage of this authority and change the implementation contract which is pointed by proxy and therefore execute potential malicious functionality in the implementation contract.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (2/3, 3/5) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (2/3, 3/5) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
AND
- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.

- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

- Renounce the ownership and never claim back the privileged role;
OR
- Remove the risky functionality.

Note: we recommend the project team consider the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

I Alleviation

[Mind Network Team, July 10, 2024]: The team acknowledged the issue and adopted the multisign solution to ensure the private key management process at the current stage. The proxy_admin contract has transferred the ownership to a Timelock contract with a minimal 2 day delay.

Grant Role transaction hash for the Timelock contract:

- [0xb75bf7dceac4e219a78cb3430f33f57765616835a4aca2795e29a8738363410d](#)

The proposer and cancellor roles within the Timelock are given to a multisig with 2/3 signers in the sensitive function signing process.

- Grant Role transaction hash for Gnosis Safe:
[0xade3791342528146cf3acc279ff3db4285de471feec05cd36d209d060349a111](#)
- The 3 multisign addresses:
 1. EOA:[0xDB9aCdA5F77bABAE50359747870E4eB64Ba41D63](#)
 2. EOA:[0xAc4d874555CE7230108A44729565661CdA247374](#)
 3. EOA:[0xa16e241b2CAa598b62DBd9cF59d7135ffe3c49e0](#)

The Timelock contract itself is currently the only address with the `DEFAULT_ADMIN` role for the Timelock.

Documentation on the centralized roles are provided here: <https://docs.mindnetwork.xyz/minddocs/security-and-privacy/multi-sig-and-timelock>.

[CertiK, July 11, 2024]: While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it. CertiK strongly encourages the project team periodically revisit the private key security management of all above-listed addresses.

STR-02 | CENTRALIZATION RISKS

Category	Severity	Location	Status
Centralization	● Major	contracts/strategies/FixedStrategy.sol: 18; contracts/strategies/MToken.sol: 33; contracts/strategies/Strategy.sol: 55, 62, 66, 73	● Mitigated

Description

In `Strategy` contract, the `owner` role has authority over the following functions:

- `setup` : Configures the lock period, deposit and redeem limits, and the total assets cap.
- `pause` : Halts all pausable contract activities. It should be noted that the `owner` role has the ability to pause withdrawal and redemption operations, which means users will not be able to retrieve their deposited assets as expected.
- `unpause` : Resumes all previously halted contract activities.
- `withdrawAirdropToken` : Retrieves non-critical ERC20 tokens sent to the contract.

The `Strategy` contract extends from `OwnableUpgradeable` contract, the `owner` role has authority over the following functions:

- `renounceOwnership` : Leaves the contract without owner.
- `transferOwnership` : Transfers ownership of the contract to a new account (`newOwner`).

In `MToken` contract, the `STRATEGY` role has authority over the following function:

- `_update` : Updates the user's account balance. It should be noted that the `_update` function is integral to the `transfer`, `transferFrom`, `mint`, `burn`, and `burnFrom` functions. This means that the `STRATEGY` role has the capability to modify any user's account balance by invoking these functions.

The `MToken` contract extends from `AccessControlUpgradeable` contract, the `DEFAULT_ADMIN_ROLE` role has authority over the following functions:

- `grantRole` : Grants `role` to `account`.
- `revokeRole` : Revokes `role` from `account`.

In `FixedStrategy` contract, the `owner` role has authority over the following function:

- `setCampaignParam` : Sets parameters of the campaign.

Any compromise to the privileged accounts may allow the hacker to take advantage of this authority, setting state variables to irrational values, pausing contract activities, manipulating users' account balances, thereby disrupting the project's normal

functions.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign ($\frac{2}{3}$, $\frac{3}{5}$) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
OR
- Remove the risky functionality.

Alleviation

[Mind Network Team, July 10, 2024]: The team acknowledged the issue and adopted the multisign solution to ensure the private key management process at the current stage. The MToken contract has transferred the ownership to a Timelock contract with a minimal 2 day delay.

Grant Role transaction hash for the Timelock contract:

- [0xb75bf7dceac4e219a78cb3430f33f57765616835a4aca2795e29a8738363410d](#)

The proposer and cancellor roles within the Timelock are given to a multisig with 2/3 signers in the sensitive function signing process.

- Grant Role transaction hash for Gnosis Safe:
[0xc3425dee5d9ffd2634b95af73d86b111a77cdf7c3f7a5e970a08771dca4c311](#)
- The 3 multisign addresses:
 1. EOA:[0xDB9aCdA5F77bABAE50359747870E4eB64Ba41D63](#)
 2. EOA:[0xAc4d874555CE7230108A44729565661CdA247374](#)
 3. EOA:[0xa16e241b2CAa598b62DBd9cF59d7135ffe3c49e0](#)

The Timelock contract itself is currently the only address with the `DEFAULT_ADMIN` role for the Timelock.

There is currently only one Strategy contract, which is the only address with the `STRATEGY_ROLE`. The owner of the Strategy contract is the Timelock.

Documentation on the centralized roles are provided here: <https://docs.mindnetwork.xyz/minddocs/security-and-privacy/multi-sig-and-timelock>.

[CertiK, July 11, 2024]: While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it. CertiK strongly encourages the project team periodically revisit the private key security management of all above-listed addresses.

STR-03 | UNPROTECTED UPGRADEABLE CONTRACT

Category	Severity	Location	Status
Logical Issue	● Medium	contracts/strategies/Strategy.sol: 37	● Resolved

Description

The `Strategy` logic contract does not protect the initializer. An attacker could call the `initialize` function and assume ownership of the logic contract, which would enable them to perform privileged operations and deceive unsuspecting users into believing that they are interacting with the legitimate owner of the upgradeable contract.

Recommendation

We recommend adding

```
/// @custom:oz-upgrades-unsafe-allow constructor
constructor() initializer {...}
```

OR

```
/// @custom:oz-upgrades-unsafe-allow constructor
constructor() {
    ...
    _disableInitializers();
}
```

This addition will prevent the function `$INIT()` from being called directly in the implementation contract, but the proxy will still be able to initialize its storage variables.

Alleviation

[Mind Network Team, July 2, 2024]: The team heeded the advice and resolved the issue in commit [719d74389afaf5a0f5863f981837d4674c153b3f](https://github.com/mind-network/mind-network-contracts/commit/719d74389afaf5a0f5863f981837d4674c153b3f).

STR-04 | POTENTIAL ZERO SHARE

Category	Severity	Location	Status
Incorrect Calculation	Minor	contracts/strategies/Strategy.sol: 100~101, 113~114, 128~129	Acknowledged

Description

The `_convertToShares` function is vulnerable to an exploit, known as a donation attack, where an attacker can directly transfer asset tokens into the contract.

```
function _convertToShares(uint256 assets, Math.Rounding rounding) internal view
virtual returns (uint256) {
    return assets.mulDiv(shareToken.totalSupply() + 10 ** decimalsOffset,
totalAssets() + 1, rounding);
}
```

An attacker could exploit this function by directly transferring asset tokens into it, artificially inflating the `totalAssets()` value. Due to Solidity's truncation issue, this could result in users receiving zero shares upon using the `deposit` function. Consequently, the users' assets would become permanently locked within the contract with no possibility of retrieval.

In certain situations, the attacker can then burn their acquired shares to break even.

Recommendation

It is recommended to implement safeguards against direct token transfers that can manipulate the `totalAssets()` value, such as using a state variable for recording the asset balance of this contract that is contributed through the `deposit` function.

Alleviation

[Mind Network Team, July 4, 2024]: We will also deposit some initial liquidity, rather than recording the asset balance of this contract that is contributed through the deposit function.

[CertiK, July 8, 2024]: If sufficient initial liquidity is deposited, that would help alleviate the issue.

[Mind Network Team, July 10, 2024]: Issue acknowledged. I won't make any changes for the current version. We will deposit the initial liquidity before launch according to the discussion.

STR-05 | MISSING EMIT EVENTS

Category	Severity	Location	Status
Volatile Code	Minor	contracts/strategies/FixedStrategy.sol: 18~19; contracts/strategies/Strategy.sol: 55~56	Resolved

Description

The `setCampaignParam` and `setup` functions enable the `owner` role to update the following state parameters:

- `campaignUntil`
- `minBalance`
- `lockPeriod`
- `depositAmountMax`
- `redeemAmountMax`
- `totalAssetsCap`

However, it currently does not emit any events upon updating these values. Events are crucial for tracking changes in Smart Contracts, as they are logged into the blockchain and can be monitored by external entities. The absence of events in this function can lead to a lack of transparency and make it difficult for users to verify that the state has changed.

Recommendation

It is recommended to define and emit events for each state-changing action.

Alleviation

[Mind Network Team, July 2, 2024]: The team heeded the advice and resolved the issue in commit [719d74389afaf5a0f5863f981837d4674c153b3f](https://github.com/mind-network/mind-network/commit/719d74389afaf5a0f5863f981837d4674c153b3f).

STR-06 | INCOMPATIBILITY WITH DEFLATIONARY TOKENS

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/strategies/Strategy.sol: 101~102, 149~150	● Acknowledged

Description

The project design may not be compatible with non-standard ERC20 tokens, such as deflationary tokens.

The function `safeTransferFrom()` is used to move asset tokens from the sender to the recipient but fail to verify if the received token amount matches the transferred amount. This could pose an issue with fee-on-transfer tokens, where the post-transfer balance might be less than anticipated, leading to balance inconsistencies.

```
101 SafeERC20.safeTransferFrom(assetToken, user, address(this), assetAmount);
```

Scenario

When transferring deflationary ERC20 tokens, the input amount may not equal the received amount due to the charged transaction fee. For example, if a user sends 100 deflationary tokens (with a 10% transaction fee), only 90 tokens actually arrive to the contract. However, the user is minted shares meant to be equivalent to 100 asset tokens.

Recommendation

We advise the client to regulate the set of tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support non-standard ERC20 tokens.

Alleviation

[Mind Network Team, July 10, 2024]: Issue acknowledged. I won't make any changes for the current version. We will regulate the set of tokens according to the advice.

APPENDIX | MIND NETWORK - AUDIT

Finding Categories

Categories	Description
Incorrect Calculation	Incorrect Calculation findings are about issues in numeric computation such as rounding errors, overflows, out-of-bounds and any computation that is not intended.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

