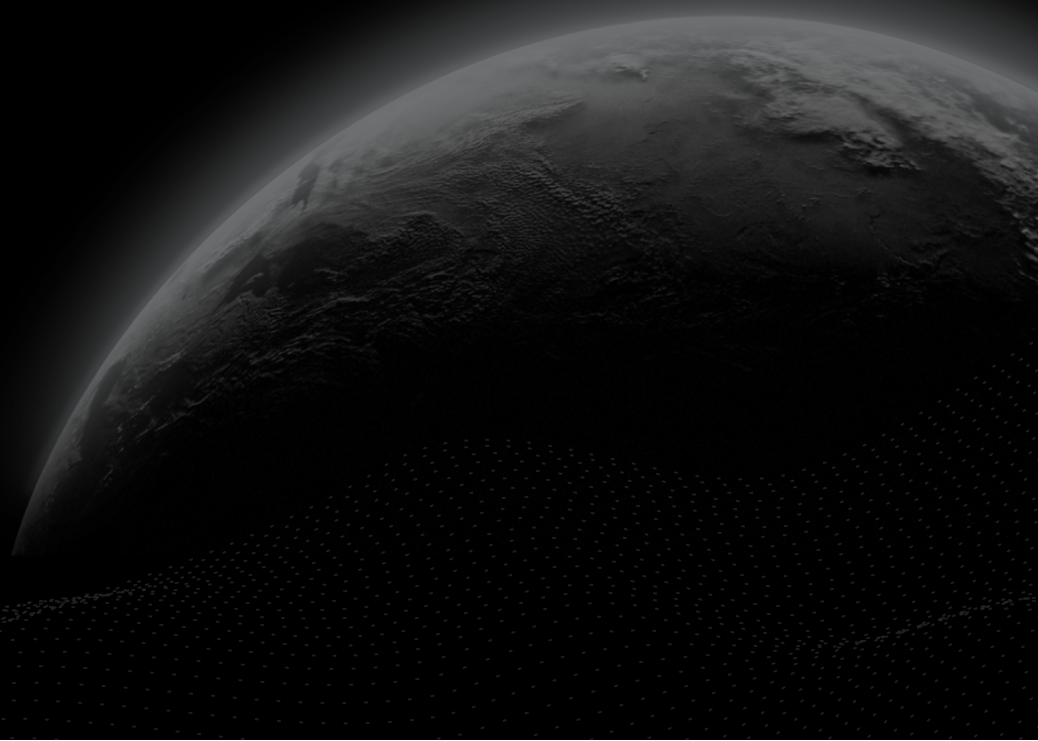
CERTIK

Security Assessment

obelisk network

CertiK Assessed on Aug 5th, 2024



CertiK Assessed on Aug 5th, 2024

obelisk network

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Binance Smart Chain (BSC) | Ethereum (ETH)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 08/05/2024	N/A

CODEBASE

update

base

View All in Codebase Page

COMMITS

962824e6b33989cd491ad7151621a4a8b0204ae9

588af44f5347acd849a8f54be7a27c1bf06828bb

View All in Codebase Page

Highlighted Centralization Risks

- Contract upgradeability
- Withdraws can be disabled
- Transfers can be paused
- Privileged role can mint tokens

Vulnerability Summary

12	11	0	0	1	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	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.
1	Major	1 Acknowledged	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.
4	Medium	4 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
5	Minor	5 Resolved	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.

2	Informational	2 Resolved	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 | OBELISK NETWORK

CODEBASE | OBELISK NETWORK

Repository

update

base

Commit

962824e6b33989cd491ad7151621a4a8b0204ae9

588af44f5347acd849a8f54be7a27c1bf06828bb

AUDIT SCOPE | OBELISK NETWORK

17 files audited ● 1 file with Acknowledged findings ● 12 files with Resolved findings ● 4 files without findings

ID	Repo	File	SHA256 Checksum
ONN	NodeDAO/obelisk-network	core/ObeliskNetwork.sol	0895f9a95a19bfc5ab5c1754017b64aedd455e078427eb2b2df0fbea66b03e4c
MSN	NodeDAO/obelisk-network	core/MintStrategy.sol	5449bc1c29089e35f121e56f18b1596cc913f86fa540b880da9e9520130c2010
SMN	NodeDAO/obelisk-network	core/StrategyManager.sol	9bc6ca66406bcc7e27fc86a45a31a2956f751b97c932ed4252225116d1bbcdb9
SND	NodeDAO/obelisk-network	modules/Strategy.sol	af71ae617f6caf3b868ff8fe9a1a5ef923b76a55eec6c8f67ede5254c1999c8e
VND	NodeDAO/obelisk-network	modules/Version.sol	c98aa07b5511c60a94c70f2127bf2c6f9d266c114d06944f39bc92d57ca5d3b6
WRN	NodeDAO/obelisk-network	modules/WithdrawalRequest.sol	2c7c6c00b566d714fe4c1a00077171e1ab267cc4c2894c3585a892ee737ea7ef
BSN	NodeDAO/obelisk-network	strategies/BaseStrategy.sol	e8d58fbd7432d134ed830652b6e9557fc178c8e0cc51416812669a5e859170d4
CSN	NodeDAO/obelisk-network	strategies/CefiStrategy.sol	7775da2dd6fd8ccd9f1a56a3fb1d2d789eff1f2b5a5d0fa79bdf0ea055304ea4
BTN	NodeDAO/obelisk-network	tokens/BaseToken.sol	159ab4289e0849cc25d520b16a35482ab06768193cb90decdd697be3a46906e7
OBT	NodeDAO/obelisk-network	tokens/OBTC.sol	9f3e1a6d9f02b9e1824fb7346a5342d722ddfaa6e60c39c8ee1deb71eb061f12
OLT	NodeDAO/obelisk-network	tokens/OLTC.sol	7d408cd2f78cb423a26ddde3a1609938d358fa164d662fbb7e70555bc36c90ee
OYB	NodeDAO/obelisk-network	tokens/OYBTCB2.sol	71b6342e5f29eb2558eef1603ed0d7ee51f22cfab8e1b7741cb66c6e71dd11e3
OYT	NodeDAO/obelisk-network	tokens/OYBTCBBL.sol	0bbc2a26fc7b4d56f426bd0a5fcea2210eb78cee583cb6bce4e88b9edf87da33

ID	Repo	File	SHA256 Checksum
AND	NodeDAO/obelisk-network	modules/Assets.sol	de1bc63ba0b8123dff2b3159bf258b217caa07e559d3136e7486dc03e5ffddb1
BLN	NodeDAO/obelisk-network	modules/BlackList.sol	857fda16c5b460a4c0ca981155ca721fdf3acfb8725ee0b1a30cfa54f9ff3155
DND	NodeDAO/obelisk-network	modules/Dao.sol	67f5da0be7a00bff1c17b64e7cf29ad62308fb8010dc0685275151037976ce6b
DSN	NodeDAO/obelisk-network	strategies/DefiStrategy.sol	8543529e6caccd029ff683d926cc00c36b85298cf528baf460d91bf07c238b07

APPROACH & METHODS | OBELISK NETWORK

This report has been prepared for obelisk network to discover issues and vulnerabilities in the source code of the obelisk network project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing 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.

FINDINGS | OBELISK NETWORK

12	0	1	4	5	2
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for obelisk network. Through this audit, we have uncovered 12 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
ONN-02	**Centralization Risks In ObeliskNetwork.Sol**	**Centralization**	**Major**	**Acknowledged**
NDA-02	Potential Reentrancy Attack (Sending Tokens)	Concurrency	Medium	Resolved
ONN-03	`ObeliskNetwork.requestWithdrawals()` Doesn't Check If Asset Matches Strategy	Volatile Code	Medium	Resolved
SMN-01	`getStakerStrategyList()` Always Fails	Logical Issue	Medium	Resolved
WRN-01	Upgradable `WithdrawalRequest` Reorganizes The Storage	Volatile Code	Medium	Resolved
NDA-01	Local Variable Shadowing	Coding Style	Minor	Resolved
NDA-04	Inherited Contracts Not Initialized In Initializer	Logical Issue	Minor	Resolved
NDA-05	Lack Of Sanity Checks	Volatile Code	Minor	Resolved
SMN-02	Wrong Events Emited	Inconsistency	Minor	Resolved
SND-01	`__Strategy_init()` Allows Duplicating Strategies	Volatile Code	Minor	Resolved
MSN-01	`IERC20Metadata` Can Be Used	Coding Style	Informational	Resolved

ID	Title	Category	Severity	Status
NDA-06	Typos	Coding Style	Informational	Resolved

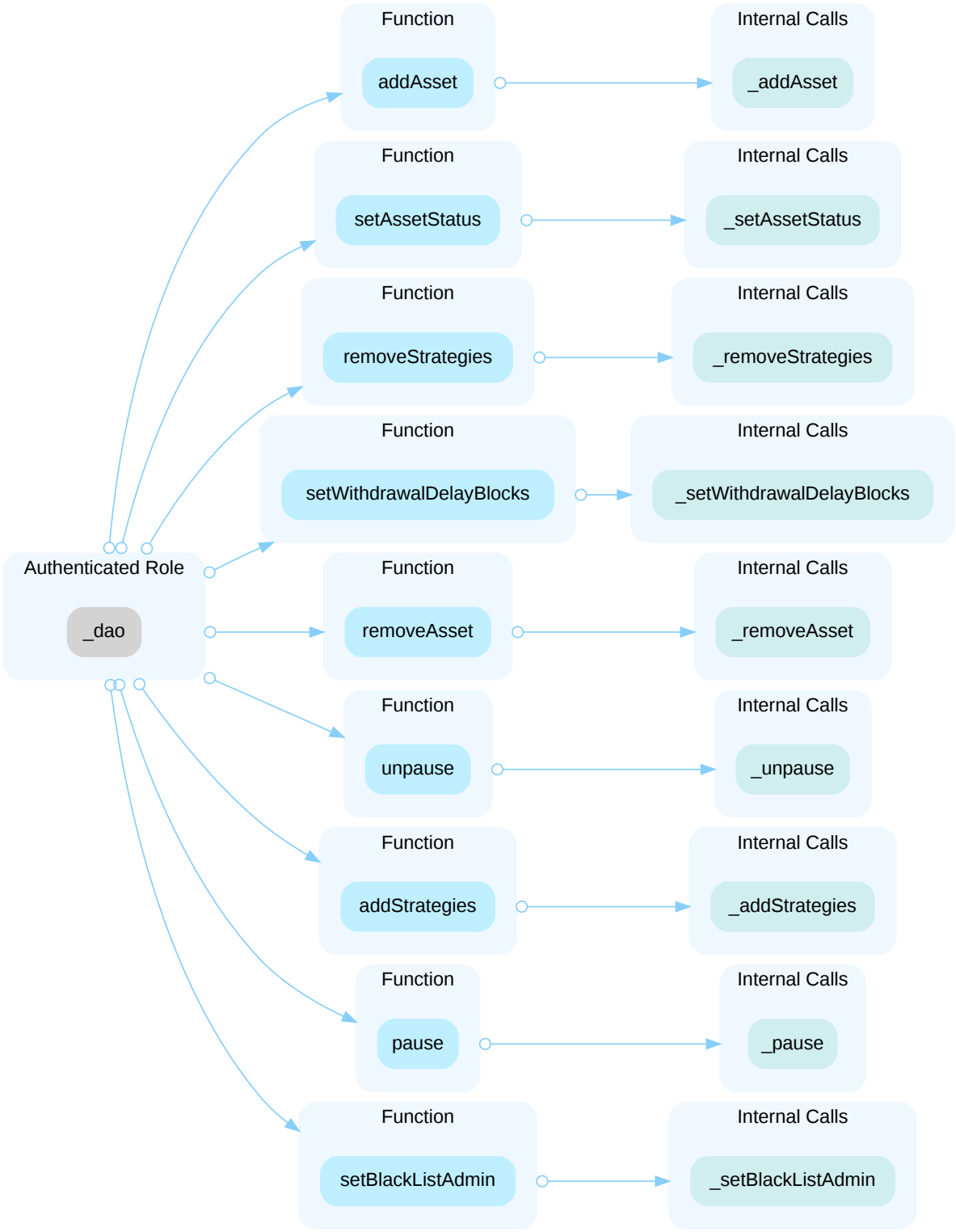
ONN-02 | CENTRALIZATION RISKS IN OBELISKNETWORK.SOL

Category	Severity	Location	Status
Centralization	Major	core/ObeliskNetwork.sol (base): 42, 102, 106, 110, 114, 118, 122, 126, 130, 151, 158	Acknowledged

Description

In the contract `ObeliskNetwork` the role `_dao` has authority over the functions shown in the diagram below. Any compromise to the `_dao` account may allow the hacker to take advantage of this authority and

- `addAsset()` / `removeAsset()` / `setAssetStatus()`
- `setBlackListAdmin()` who can `addBlackList()` / `removeBlackList()`
- `setWithdrawalDelayBlocks()` up to `MAX_WITHDRAWAL_DELAY_BLOCKS` (12.5 days)
- `addStrategies()` / `removeStrategies()`
- `pause()` / `unpause()`
- `setDao()` allows the `owner` to set a new `_dao`



In the contract `ObeliskNetwork` the role `_mintsecurity` is assigned to the `MintSecurity` contract controlled by a set of guardians. Setting a small `quorum` may allow a group of guardians to `whiteListMint()` any amount of o-tokens.

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 (⅔, ⅗) 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

[Project Team]: In the short term we will use a multi-signature wallet as the DAO address, and when it is stable, we will use DAO+Time-lock as management. Contract owner will be a Time-lock at the beginning.

NDA-02 | POTENTIAL REENTRANCY ATTACK (SENDING TOKENS)

Category	Severity	Location	Status
Concurrency	Medium	modules/WithdrawalRequest.sol (base): 139; strategies/BaseStrategy.sol (base): 150, 172; strategies/CefiStrategy.sol (base): 48~60, 57, 58	Resolved

Description

A reentrancy attack can occur when the contract creates a function that makes an external call to another untrusted contract before resolving any effects. If the attacker can control the untrusted contract, they can make a recursive call back to the original function, repeating interactions that would have otherwise not run after the external call resolved the effects.

`CefiStrategy.withdraw()` is not protected by the `nonReentrant` modifier, however, can be called by `onlyStrategyManager`. An external call to `underlyingToken` is done before the state is updated (`pendingWithdrawal` and `totalPendingWithdrawal`).

`ObeliskNetwork.claimWithdrawals()` is not protected by the `nonReentrant` modifier. An external call to `_userWithdrawal.strategy` is done before the `withdrawalQueue` is updated.

Recommendation

We recommend using the Checks-Effects-Interactions Pattern to avoid the risk of calling unknown contracts or applying OpenZeppelin ReentrancyGuard library - `nonReentrant` modifier for the aforementioned functions to prevent reentrancy attack.

ONN-03 `ObeliskNetwork.requestWithdrawals()` DOESN'T CHECK IF ASSET MATCHES STRATEGY

Category	Severity	Location	Status
Volatile Code	Medium	core/ObeliskNetwork.sol (base): 60	Resolved

Description

`ObeliskNetwork` withdrawals work this way:

1. The user calls `ObeliskNetwork.requestWithdrawals()` and provides strategy, o-token, and amount
2. It is ensured that the o-token is supported
3. The o-tokens are transferred from the user to the `ObeliskNetwork` address
4. `WithdrawalRequest._requestWithdrawals()` is called with the strategy, o-token, user, and amount
5. The arguments are remembered, no checks are performed
6. After some `_withdrawalDelayBlocks` the user calls `ObeliskNetwork.claimWithdrawals()`
7. `WithdrawalRequest._claimWithdrawals()` is called, it gets saved strategy, o-token, user, and amount
8. If `!isNativeStrategy`, then underlying tokens are transferred to the user, and it is checked that o-tokens match the underlying tokens
9. The `_userWithdrawal` is marked as `claimed`
10. The `WithdrawalsClaimed` event is emitted with o-token, amount, and destination address
11. o-tokens are burnt from the `ObeliskNetwork` address

However, if the strategy is `nativeBTCStrategy`, there are no checks for o-tokens. As a result, the user can pass any o-token (if `_isSupportedAsset()`). It will be transferred from the user and burnt, the event will be emitted, but the user will get nothing in return.

Also, the user can't cancel their requests. It is not checked if `_isPausedAsset(_token)`.

Recommendation

We recommend ensuring that only specific o-token is used with `nativeBTCStrategy`, or clarifying the intended behavior.

Alleviation

[Project Team]: nativeWithdrawStrategy represents the native asset withdrawal strategy of oLTC or oBTC.

In `_claimWithdrawals()` function the `WithdrawalsClaimed` event indicates which o-token is withdrawn, which will be enough for the custody service to handle it correctly.

SMN-01 | `getStakerStrategyList()` ALWAYS FAILS

Category	Severity	Location	Status
Logical Issue	Medium	core/StrategyManager.sol (base): 41	Resolved

Description

```
41          for (uint256 i = 0; i < _strategyListength;) {
42              if (_sharesList[i] != 0) {
43                  strategies[j] = strategyList[i];
44                  _shares[j] = _sharesList[i];
45                  j++;
46              }
47          }
```

The cycle doesn't increment `i` and never finishes.

Recommendation

We recommend incrementing the `i`.

WRN-01 UPGRADABLE `WithdrawalRequest` REORGANIZES THE STORAGE

Category	Severity	Location	Status
Volatile Code	Medium	src/modules/WithdrawalRequest.sol (update1): 186	Resolved

Description

The previous version of the `WithdrawalRequest` contract used a different storage structure. Upgradable contracts should preserve the storage structure and only add new fields to the end with adjusting of `__gap` size.

Recommendation

We recommend ensuring the `WithdrawalRequest` and the inherited contracts will not be upgraded over the existing deployments.

Alleviation

The project team confirmed that is the initial deployment, not an upgrade.

NDA-01 | LOCAL VARIABLE SHADOWING

Category	Severity	Location	Status
Coding Style	Minor	core/ObeliskNetwork.sol (base): 110; tokens/BaseToken.sol (base): 18, 18	Resolved

Description

A local variable is shadowing another component defined elsewhere. This means that when the contract accesses the variable by its name, it will use the one defined locally, not the one defined in the other place. The use of the variable may lead to unexpected results and unintended behavior.

```
110     function setAssetStatus(address _token, bool _status) external onlyDao {
```

- Local variable `_status` in `ObeliskNetwork.setAssetStatus` shadows the variable `_status` in `ReentrancyGuardUpgradeable`.

```
18      constructor(string memory _name, string memory _symbol, address _tokenAdmin)
 ERC20(_name, _symbol) {
```

- Local variable `_symbol` in `BaseToken.constructor` shadows the variable `_symbol` in `ERC20`.

```
18      constructor(string memory _name, string memory _symbol, address _tokenAdmin)
 ERC20(_name, _symbol) {
```

- Local variable `_name` in `BaseToken.constructor` shadows the variable `_name` in `ERC20`.

Recommendation

It is recommended to remove or rename the local variable that shadows another definition to prevent potential issues and maintain the expected behavior of the smart contract.

NDA-04 | INHERITED CONTRACTS NOT INITIALIZED IN INITIALIZER

Category	Severity	Location	Status
Logical Issue	Minor	modules/Version.sol (base): 31; tokens/BaseToken.sol (base): 10; tokens/OBTC.sol (base): 6; tokens/OLTC.sol (base): 6; tokens/OYBTCB2.sol (base): 6; tokens/OYBTCBBL.sol (base): 6	Resolved

Description

Contract BaseToken extends Blacklist, but the extended contract is not initialized by the current contract. Contract Version extends ReentrancyGuardUpgradeable, but the extended contract is not initialized by the current contract.

Generally, the initializer function of a contract should always call all the initializer functions of the contracts that it extends.

Recommendation

We recommend explicitly initializing the inherited contract.

NDA-05 | LACK OF SANITY CHECKS

Category	Severity	Location	Status
Volatile Code	Minor	core/MintSecurity.sol (base): 33, 39; core/MintStrategy.sol (base): 39, 40, 42, 48, 49, 52, 136, 138; core/ObeliskNetwork.sol (base): 30, 39; tokens/BaseToken.sol (base): 54, 56	Resolved

Description

Addresses are not validated before assignment or external calls, potentially allowing the use of zero addresses and leading to unexpected behavior or vulnerabilities. For example, transferring tokens to a zero address can result in a permanent loss of those tokens.

`WithdrawalRequest.withdrawalDelayBlocks` is not checked against `MAX_WITHDRAWAL_DELAY_BLOCKS`.

Recommendation

We recommend checking the validity of the function arguments.

SMN-02 | WRONG EVENTS EMITED

Category	Severity	Location	Status
Inconsistency	Minor	core/StrategyManager.sol (base): 69, 75	Resolved

Description

`StrategyManager.requestWithdrawal()` emits `UserWithdrawal`.

`StrategyManager.withdraw()` emits `UserRequestWithdrawal`.

Recommendation

We recommend changing the events emitted.

SND-01 | `__Strategy_init()` ALLOWS DUPLICATING STRATEGIES

Category	Severity	Location	Status
Volatile Code	Minor	modules/Strategy.sol (base): 19	Resolved

Description

```
16      function __Strategy_init(address[] calldata _strategies) internal
 onlyInitializing {
17          uint256 strategiesLength = _strategies.length;
18          for (uint256 i = 0; i < strategiesLength;) {
19              strategyIsWhitelisted[_strategies[i]] = true;
20              strategyList.push(_strategies[i]);
```

`Strategy.__Strategy_init()` adds `_strategies` to the `strategyIsWhitelisted` mapping and `strategyList` array. However, it doesn't check if `_strategies` has duplicates. As a result, the `strategyList` may be inconsistent.

Recommendation

We recommend adding an explicit check:

```
19      require(!strategyIsWhitelisted[_strategies[i]], "_strategies has duplicates")
;
20      strategyIsWhitelisted[_strategies[i]] = true;
```

MSN-01 | `IERC20Metadata` CAN BE USED

Category	Severity	Location	Status
Coding Style	Informational	core/MintStrategy.sol (base): 83	Resolved

Description

```
  83          uint8 sourceDecimals = IERC20Decimal(address(underlyingToken)).decimals
();
  84          uint8 targetDecimals = IERC20Decimal(address(assetAddr)).decimals();
```

`IERC20Metadata` from "openzeppelin\contracts\token\ERC20\extensions\IERC20Metadata.sol" can be used instead of `IERC20Decimal` .

Recommendation

We recommend using the interface from OpenZeppelin.

NDA-06 | TYPOS

Category	Severity	Location	Status
Coding Style	Informational	core/StrategyManager.sol (base): 24; modules/Strategy.sol (base): 61	Resolved

Description

`_strategyListength` is supposed to be `_strategyListLength`.

Recommendation

We recommend fixing the typos.

OPTIMIZATIONS | OBELISK NETWORK

ID	Title	Category	Severity	Status
ONN-01	Inefficient Memory Parameter	Inconsistency	Optimization	Resolved

ONN-01 | INEFFICIENT MEMORY PARAMETER

Category	Severity	Location	Status
Inconsistency	Optimization	core/ObeliskNetwork.sol (base): 26~40, 31	Resolved

Description

One or more parameters with `memory` data location are never modified in their functions and those functions are never called internally within the contract. Thus, their data location can be changed to `calldata` to avoid gas consumption copying from calldata to memory.

```
26      function initialize(
```

`initialize` has memory location parameters: `_tokenAddrs` .

Recommendation

We recommend changing the parameter's data location to `calldata` to save gas.

APPENDIX | OBELISK NETWORK

Finding Categories

Categories	Description
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
Concurrency	Concurrency findings are about issues that cause unexpected or unsafe interleaving of code executions.
Inconsistency	Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification.
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.