CERTIK

EnzoBTC

Security Assessment

CertiK Assessed on Nov 26th, 2024

CertiK Assessed on Nov 26th, 2024

EnzoBTC

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
Vault	Binance Smart Chain (BSC)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 11/26/2024	N/A

CODEBASE	COMMITS
base	0e9d7a528b09dd6946a1e10b3a677f7c8ba7eb90
View All in Codebase Page	View All in Codebase Page

Highlighted Centralization Risks

Contract upgradeability

Vulnerability Summary

8	0	0	0	8	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.
2	Medium	2 Acknowledged	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
2	Minor	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.
3	Informational	3 Acknowledged	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 | ENZOBTC

CODEBASE | ENZOBTC

Repository

base

Commit

0e9d7a528b09dd6946a1e10b3a677f7c8ba7eb90

AUDIT SCOPE | ENZOBTC

18 files audited ● 12 files with Acknowledged findings ● 6 files without findings

ID	Repo	File	SHA256 Checksum
ABT	Lorenzo-Protocol/enzoBTC_contract	src/modules/Assets.sol	eb21ff0fdb61ea1510f5e47f4a27432dbb27f16ab46da1d1a2635f2a018c0686
BLB	Lorenzo-Protocol/enzoBTC_contract	src/modules/BlackList.sol	00e3a50cb459df6c11362da09cb889e5b9950ec0f9b453e80e16481db6290576
DBT	Lorenzo-Protocol/enzoBTC_contract	src/modules/Dao.sol	c4d928238aaedb650512a9de7ddf6630f164a0685d277577d3b0b3411d2ab65d
VBT	Lorenzo-Protocol/enzoBTC_contract	src/modules/Version.sol	3e287e72bc165d50de107ec8b082e87e70f7ff75d808892eb46d1c6031234a51
WBT	Lorenzo-Protocol/enzoBTC_contract	src/modules/Whitelisted.sol	c58129ee51272f8b7c6c8ec5c6894a72e871dd41c82456d11b5816b6b8d50a49
WRB	Lorenzo-Protocol/enzoBTC_contract	src/modules/WithdrawalRequest.sol	97584f5726d969c6c22e41a581b4833c4a7594cdc7f4e85d80bb2bca5077e40c
BSB	Lorenzo-Protocol/enzoBTC_contract	src/strategies/BaseStrategy.sol	2c30cd2589483f41402fd5b965127a6402519c789f39ca2d78676cd00e19d03a
CSB	Lorenzo-Protocol/enzoBTC_contract	src/strategies/CefiStrategy.sol	26ff5e928f5ae1ff11b8570a62042d65a2d539f03f25bbda8d94e1dc9e538000
BTB	Lorenzo-Protocol/enzoBTC_contract	src/tokens/BaseToken.sol	1a3450418a5769261ca61843c0fbf257fd2643c99a6e394de5a8987d012f0db5
ENB	Lorenzo-Protocol/enzoBTC_contract	src/core/EnzoNetwork.sol	6e4d0a898d7b737b76b1d67bfc6ac4eb3fe0620824121893dc11385967bacbcf
MSB	Lorenzo-Protocol/enzoBTC_contract	src/core/MintStrategy.sol	ddd59cc3c784988c3c86a3b3404e4bc200d6af2943c12305cadc4fbc33c4ad6c
SMB	Lorenzo-Protocol/enzoBTC_contract	src/core/StrategyManager.sol	5033450aa61a420b010cb668c89bcbd05d9620be686d7f17cef4fcd15b3a5c21

ID	Repo	File	SHA256 Checksum
CBT	Lorenzo-Protocol/enzoBTC_contract	src/modules/Call.sol	fbfe8647cc64f7197058266a8c479899294eb5b2df35e6eb8da2a74d4e604eb2
DSB	Lorenzo-Protocol/enzoBTC_contract	src/strategies/DefiStrategy.sol	05d4810c319274d41217df1655d1666c9d293280ed40a1040d9cb276370543a6
EBT	Lorenzo-Protocol/enzoBTC_contract	src/tokens/EnzoBTC.sol	60b49af9485c8f775a68b31bdee61614330e49f9a6b1f27f7f1c7de0f406a4ed
EBC	Lorenzo-Protocol/enzoBTC_contract	src/tokens/EnzoBTCB2.sol	ab2d247606f1f9cb457436d164e96bc9c63a52b033f5998be6ff3b33caabd5d4
EBB	Lorenzo-Protocol/enzoBTC_contract	src/tokens/EnzoBTCBBN.sol	42afc822c0829dc9a2f03983f23ceaa0946e75d3bad0e5fef46d2f947d2aad81
EBF	Lorenzo-Protocol/enzoBTC_contract	src/tokens/EnzoBTCFBTC.sol	61bb4d03ec469aab6561ab3d10303e6a97e6ff5d8464a6470e62438bad442918

APPROACH & METHODS | ENZOBTC

This report has been prepared for EnzoBTC to discover issues and vulnerabilities in the source code of the EnzoBTC 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.

THIRD-PARTY DEPENDENCY | ENZOBTC

The contract serves as the underlying entity to interact with one or more third-party protocols. The scope of the audit treats third-party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of third parties can create severe impacts, such as increasing fees of third parties, migrating to new LP pools, etc.

```
92      IMintableBurnable(_admin).whiteListMint(_mintAmount, _user);
```

```
173     IMintableBurnable(_admin).whiteListBurn(_withdrawalAmount, address(this));
```

- The functions `deposit()` and `claimWithdrawals() interact with a third-party contract with` IMintableBurnable `interface via` whiteListMint `and` whiteListBurn`.

The auditors understand that the business logic requires interaction with third parties. However, it is recommended for the team to constantly monitor the statuses of third parties to mitigate the side effects when unexpected activities are observed.

FINDINGS | ENZOBTC

8	0	1	2	2	3
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for EnzoBTC. Through this audit, we have uncovered 8 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
BTC-04	**Centralization Risks**	**Centralization**	**Major**	**Acknowledged**
BTC-01	Inherited Contracts Not Initialized In Initializer	Logical Issue	Medium	Acknowledged
WRB-01	No Cap On Fees	Logical Issue	Medium	Acknowledged
BTL-01	Local Variable Shadowing	Coding Style	Minor	Acknowledged
BTP-01	Missing Zero Address Validation	Volatile Code	Minor	Acknowledged
BTC-02	Underscore Prefix For Non-External Variables	Code Optimization	Informational	Acknowledged
BTC-03	Storage Size Convention	Coding Issue	Informational	Acknowledged
VBT-01	Contracts May Fail To Resume If Owner Renounce Ownership During Pause	Design Issue	Informational	Acknowledged

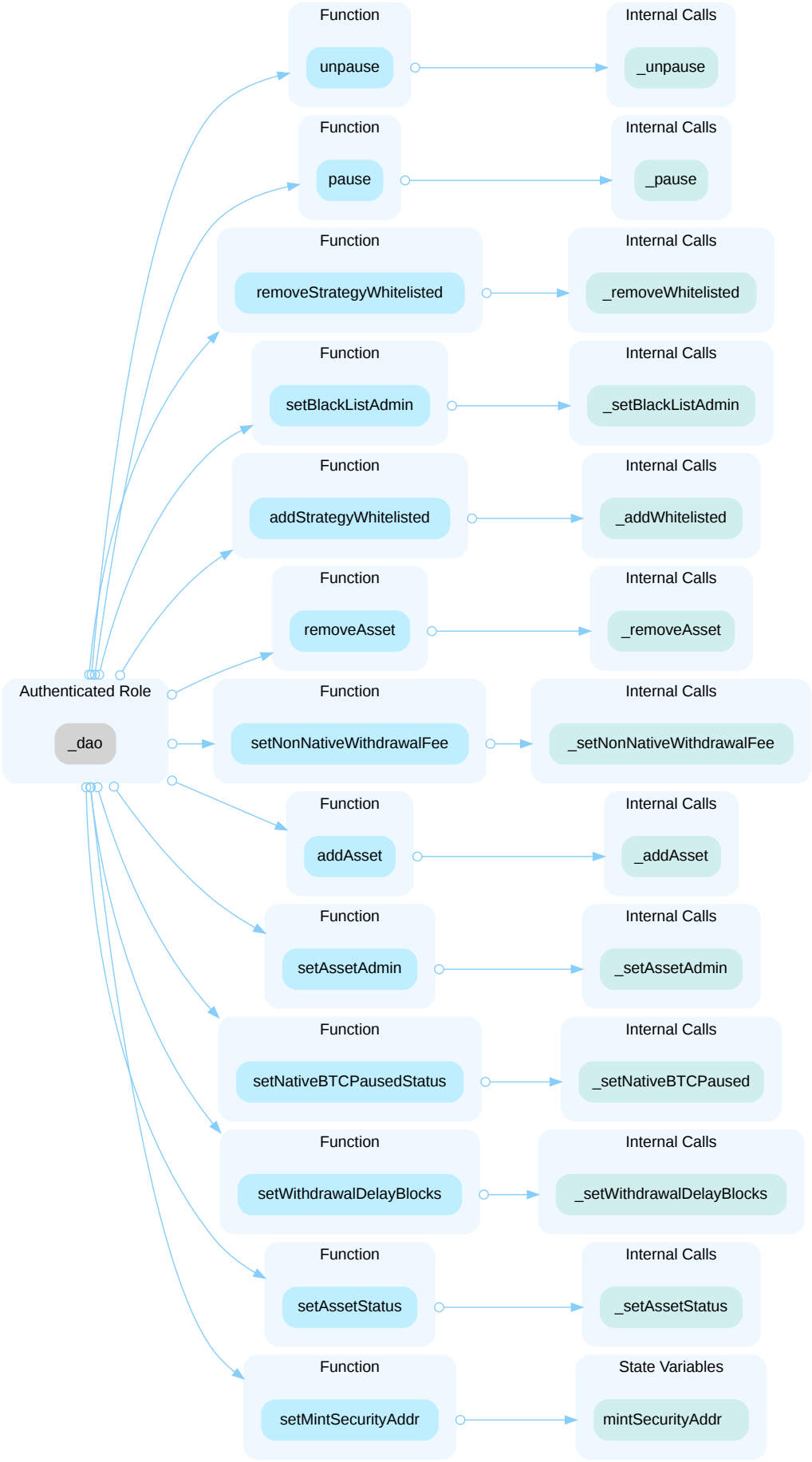
BTC-04 | CENTRALIZATION RISKS

Category	Severity	Location	Status
Centralization	Major	src/core/EnzoNetwork.sol: 63, 181, 189, 198, 207, 215, 225, 233, 241, 249, 257, 262, 270, 291, 298; src/core/MintSecurity.sol: 56, 103, 114, 134, 231, 311, 319, 326; src/core/MintStrategy.sol: 90, 118, 150, 175, 185, 193, 202, 230; src/core/StrategyManager.sol: 75, 83, 144, 151; src/modules/BlackList.sol: 32, 37; src/strategies/BaseStrategy.sol: 95, 116, 140, 163, 176, 232, 242, 250, 258; src/strategies/CefiStrategy.sol: 50, 65; src/tokens/BaseToken.sol: 53, 63, 68, 73	Acknowledged

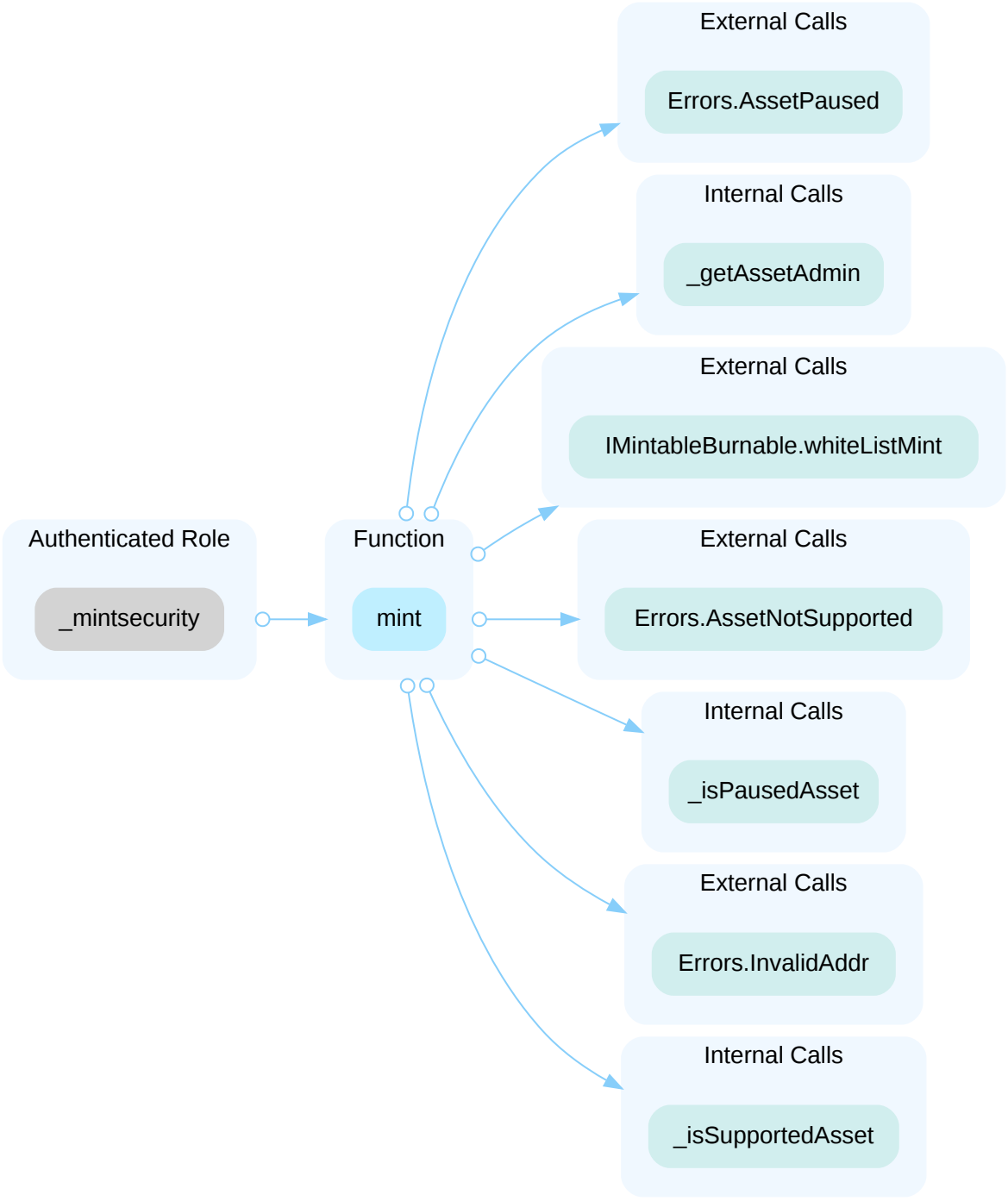
Description

In the contract `EnzoNetwork`, 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

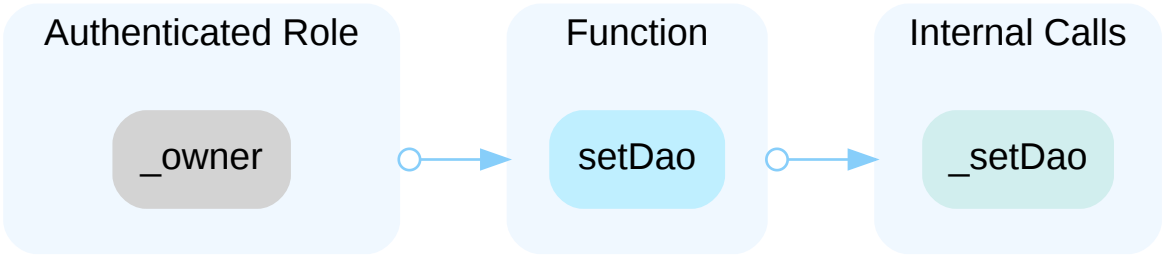
- unpause the contract
- pause contract operations
- remove strategies from whitelist
- set a blacklist admin
- add strategies to whitelist
- remove an asset
- set the non-native withdrawal fee
- add an asset token
- set asset administrator
- set native BTC paused status
- set withdrawal delay blocks
- set the asset status
- set mint security address


Function
unpause
Internal Calls
_unpause
Function
pause
Internal Calls
_pause
Function
removeStrategyWhitelisted
Internal Calls
_removeWhitelisted
Function
setBlackListAdmin
Internal Calls
_setBlackListAdmin
Function
addStrategyWhitelisted
Internal Calls
_addWhitelisted
Function
removeAsset
Internal Calls
_removeAsset
Authenticated Role
_dao
Function
setNonNativeWithdrawalFee
Internal Calls
_setNonNativeWithdrawalFee
Function
addAsset
Internal Calls
_addAsset
Function
setAssetAdmin
Internal Calls
_setAssetAdmin
Function
setNativeBTCPausedStatus
Internal Calls
_setNativeBTCPaused
Function
setWithdrawalDelayBlocks
Internal Calls
_setWithdrawalDelayBlocks
Function
setAssetStatus
Internal Calls
_setAssetStatus
Function
setMintSecurityAddr
State Variables
mintSecurityAddr

In the contract `EnzoNetwork`, the role `_mintsecurity` has authority over the functions shown in the diagram below. Any compromise to the `_mintsecurity` account may allow the hacker to take advantage of this authority and mint tokens to a specified address.

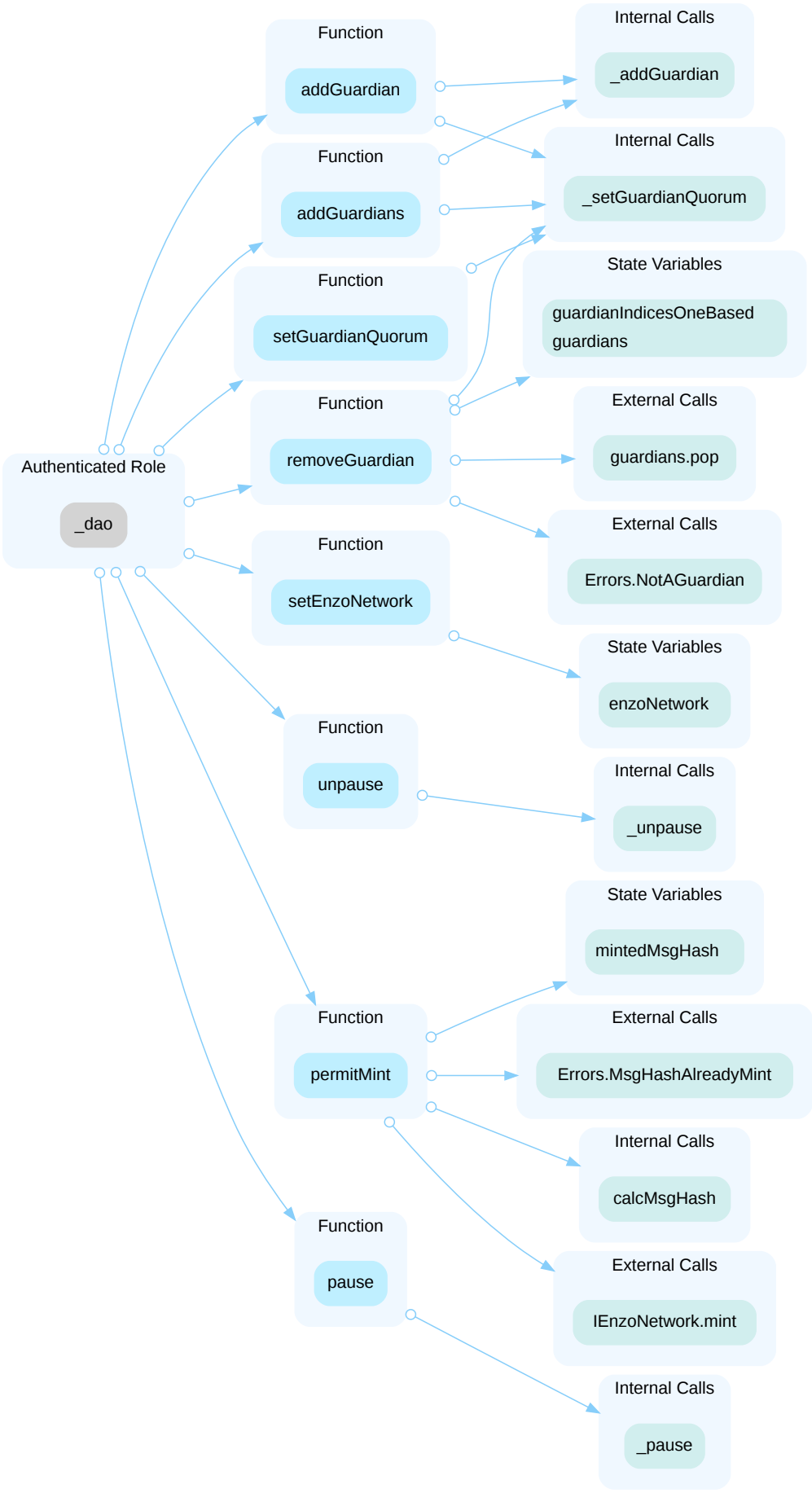


In the contract `EnzoNetwork`, the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and set the DAO address.



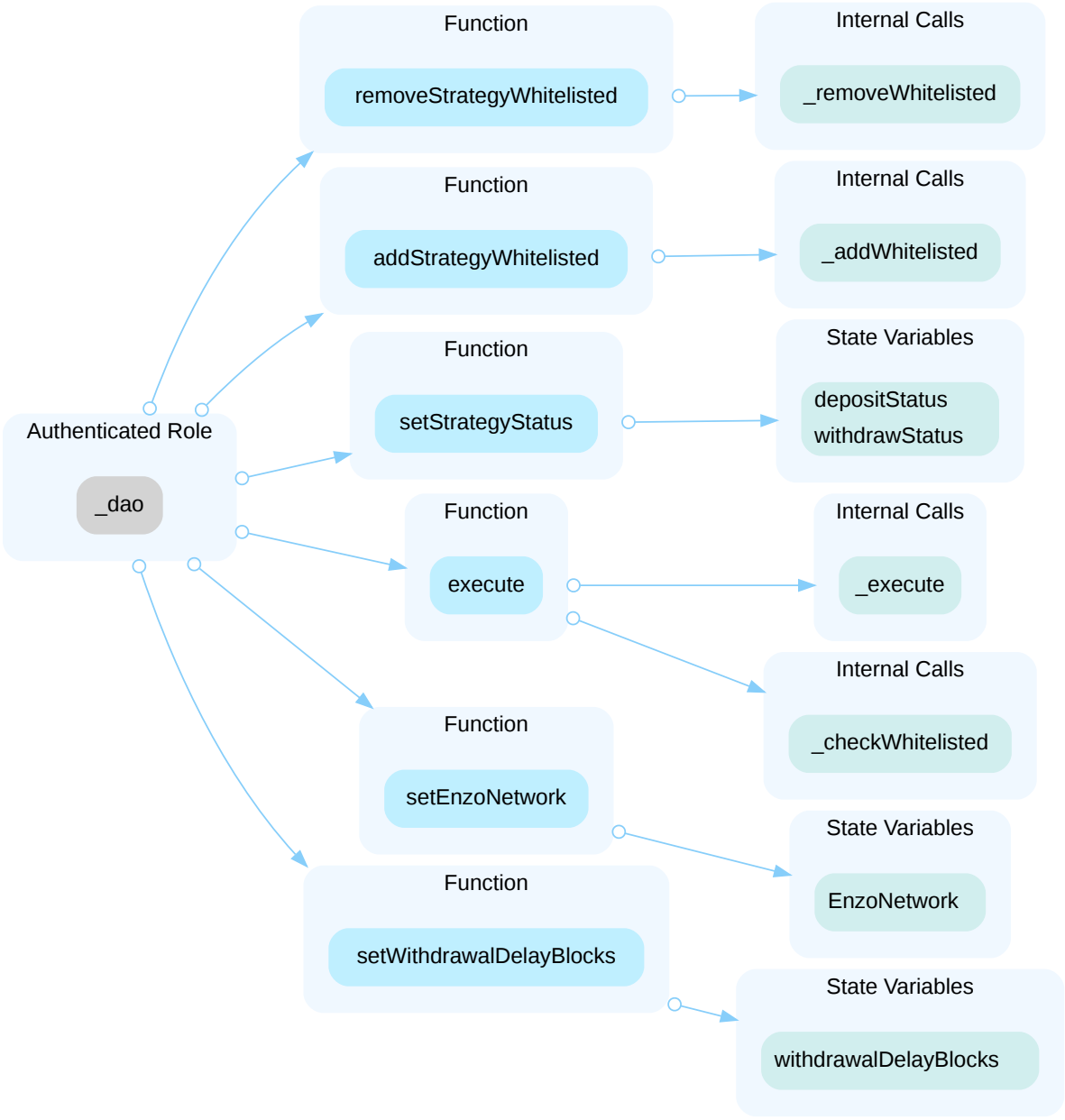
In the contract `MintSecurity`, the role `_dao` has authority over the functions shown in the diagram below. Any compromise to the `_dao` account may allow the hacker to take advantage of this authority and:

- add guardians
- set new quorum
- set the enzoNetwork address
- remove a guardian
- update the quorum
- unpause the contract
- permit minting of tokens to a destination address
- set guardian quorum
- pause the contract

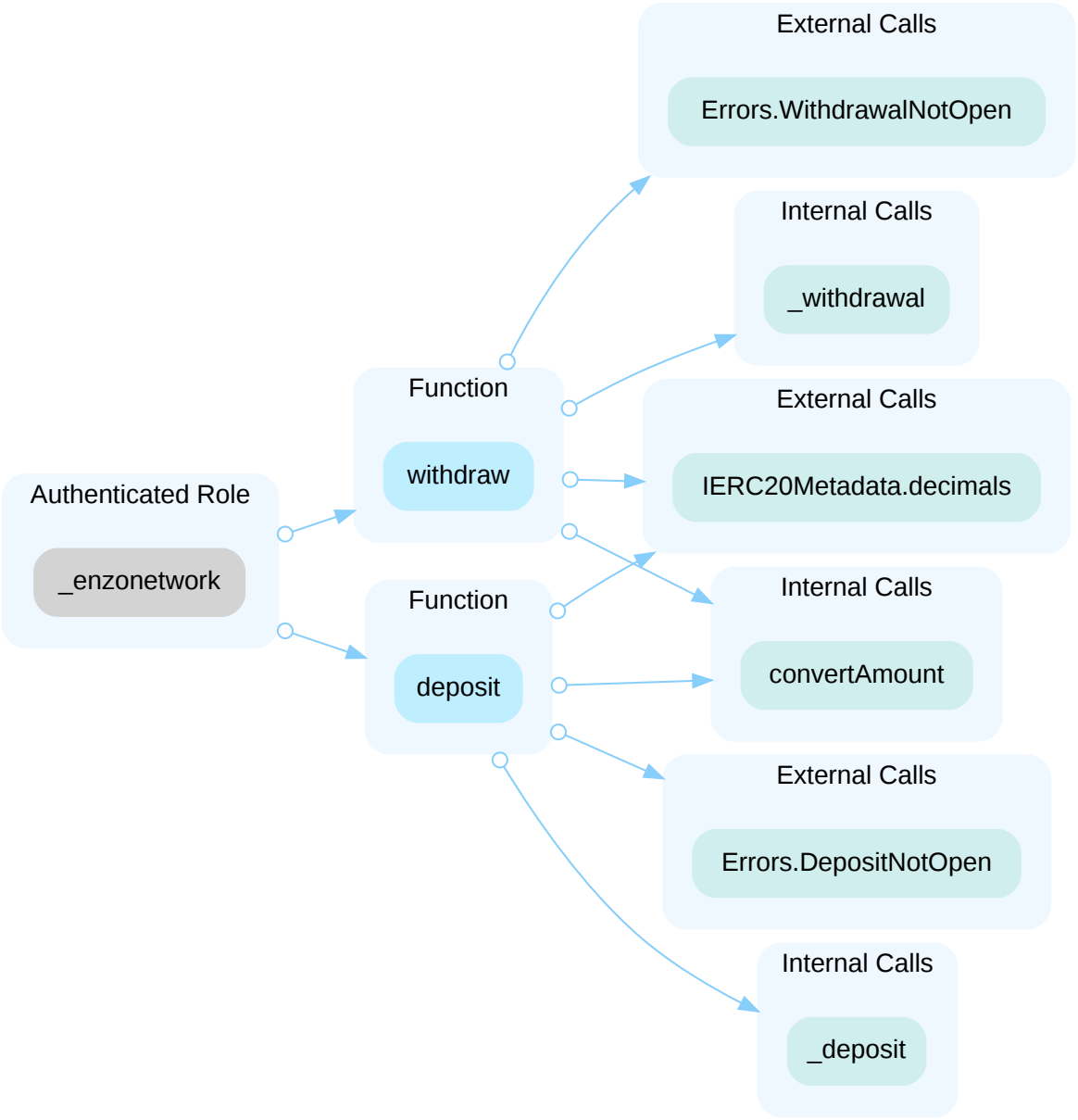

Function
addGuardian
Internal Calls
_addGuardian
Function
addGuardians
Internal Calls
_setGuardianQuorum
Function
setGuardianQuorum
State Variables
guardianIndicesOneBased
guardians
Function
removeGuardian
External Calls
guardians.pop
Authenticated Role
_dao
External Calls
Errors.NotAGuardian
Function
setEnzoNetwork
State Variables
enzoNetwork
Function
unpause
Internal Calls
_unpause
State Variables
mintedMsgHash
Function
permitMint
External Calls
Errors.MsgHashAlreadyMint
Internal Calls
calcMsgHash
Function
pause
External Calls
IEnzoNetwork.mint
Internal Calls
_pause

In the contract `MintStrategy`, 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:

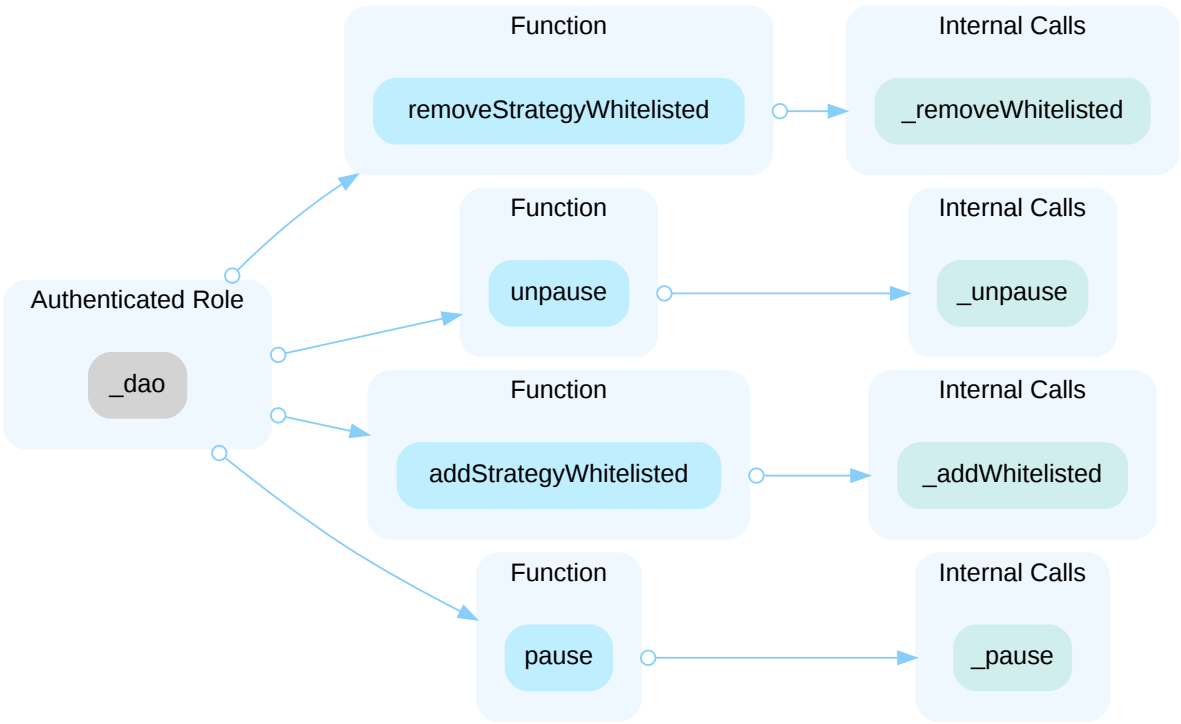
- remove strategies from whitelist
- add strategy to whitelist
- set strategy deposit and withdraw statuses
- execute transactions with provided parameter
- set the Enzo network address
- set the withdrawal delay blocks



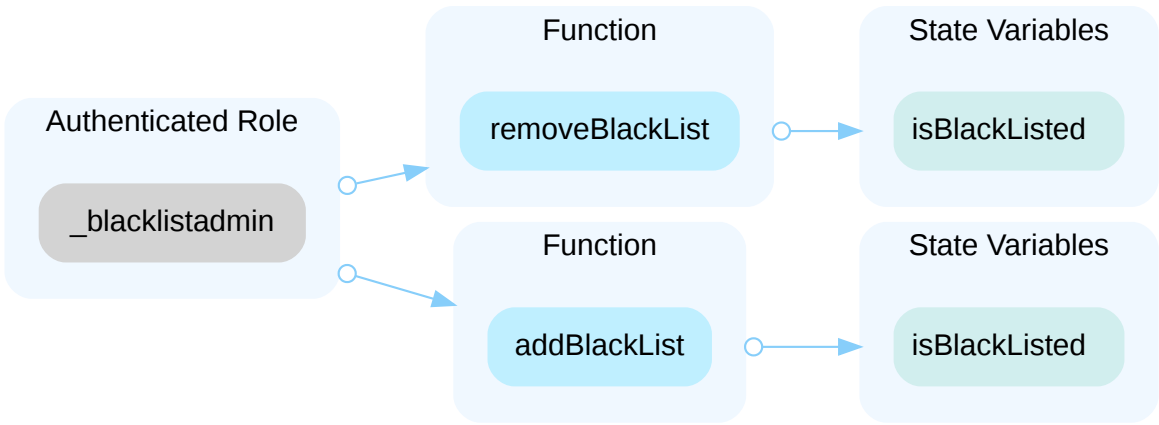
In the contract `MintStrategy`, the role `_enzonetwork` has authority over the functions shown in the diagram below. Any compromise to the `_enzonetwork` account may allow the hacker to take advantage of this authority and withdraw or deposit tokens for a user.



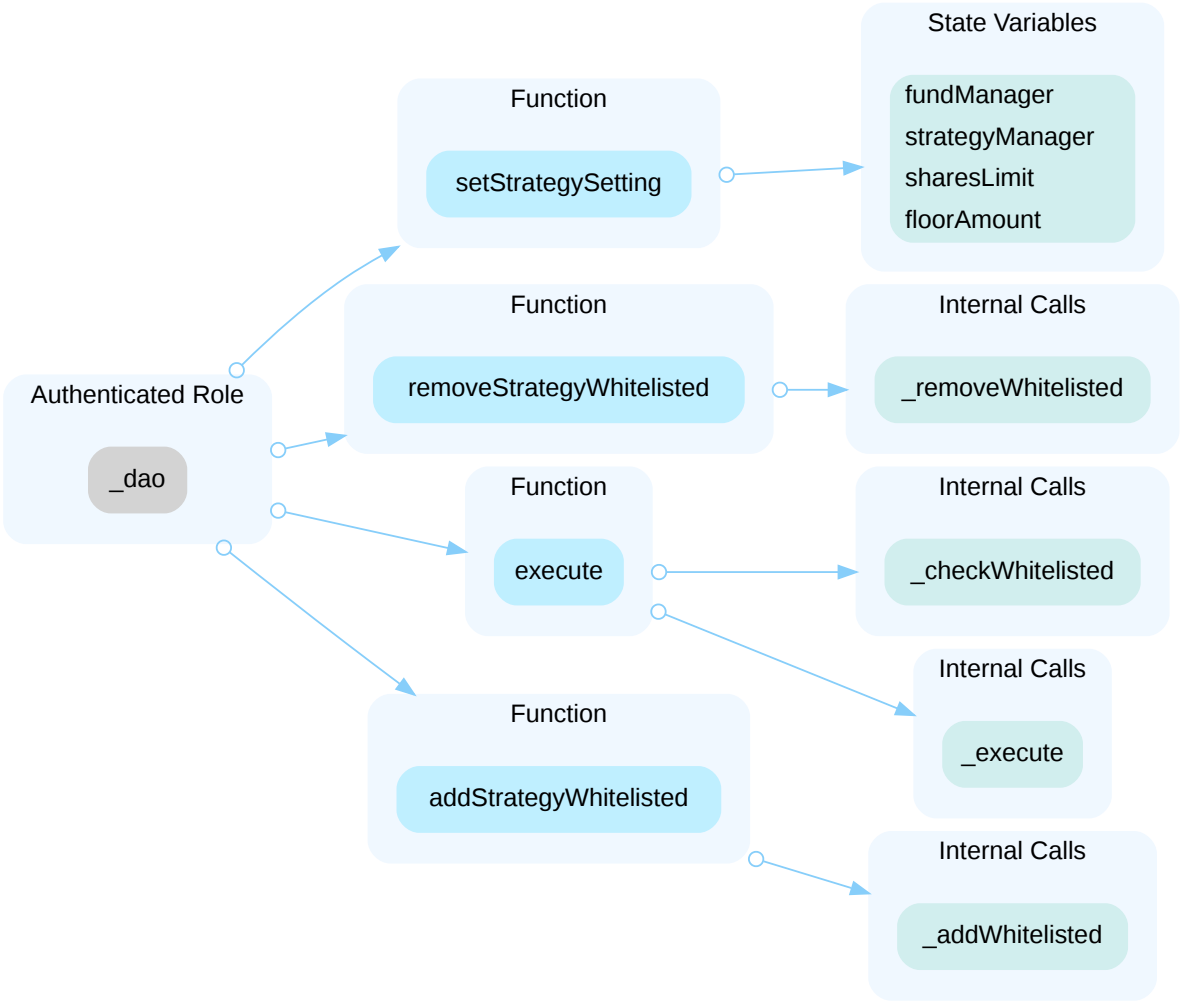
In the contract `StrategyManager`, the role `_dao` has authority over the functions shown in the diagram below. Any compromise to the `_dao` account may allow the hacker to take advantage of this authority and remove/add strategy from/to the whitelist, pause/unpause the contract.



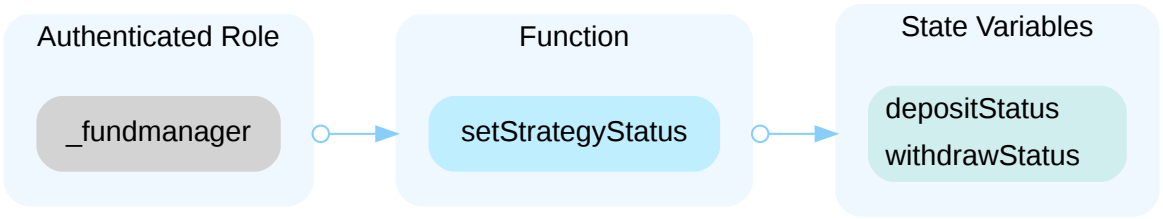
In the contract `BlackList`, the role `_blacklistadmin` has authority over the functions shown in the diagram below. Any compromise to the `_blacklistadmin` account may allow the hacker to take advantage of this authority and remove/add users from/to the blacklist.



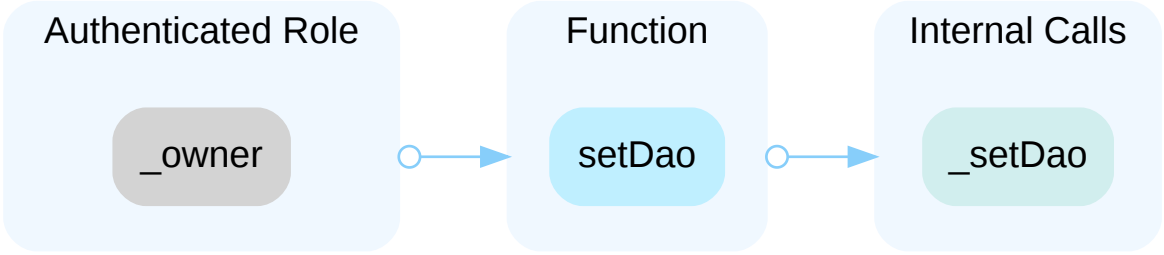
In the contract `BaseStrategy`, the role `_dao` has authority over the functions shown in the diagram below. Any compromise to the `_dao` account may allow the hacker to take advantage of this authority and set strategy settings with provided parameters, add/remove strategies to/from the whitelist, execute transactions with specified parameters.



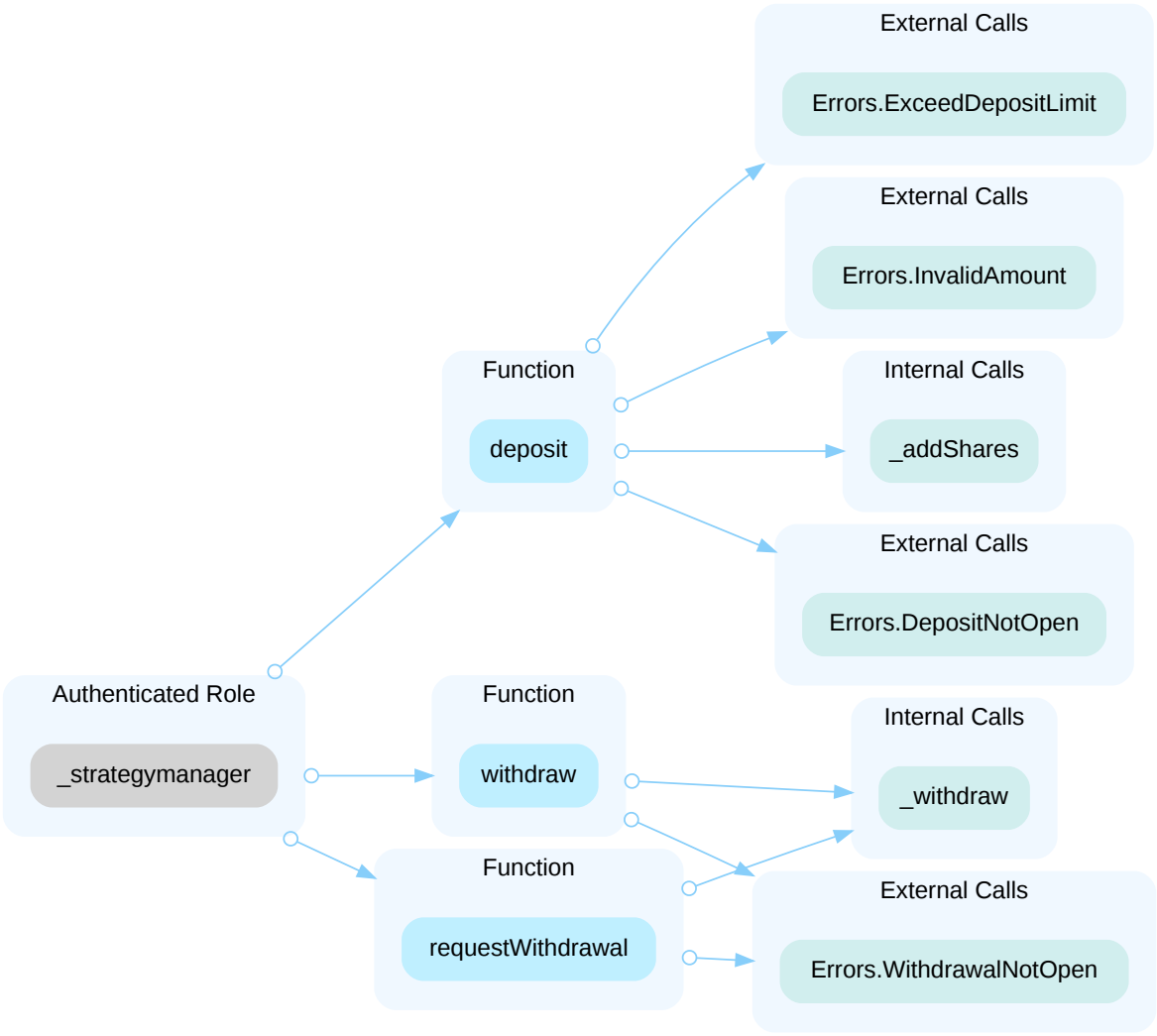
In the contract `BaseStrategy`, the role `_fundManager` has authority over the functions shown in the diagram below. Any compromise to the `_fundmanager` account may allow the hacker to take advantage of this authority and set strategy deposit and withdrawal statuses.



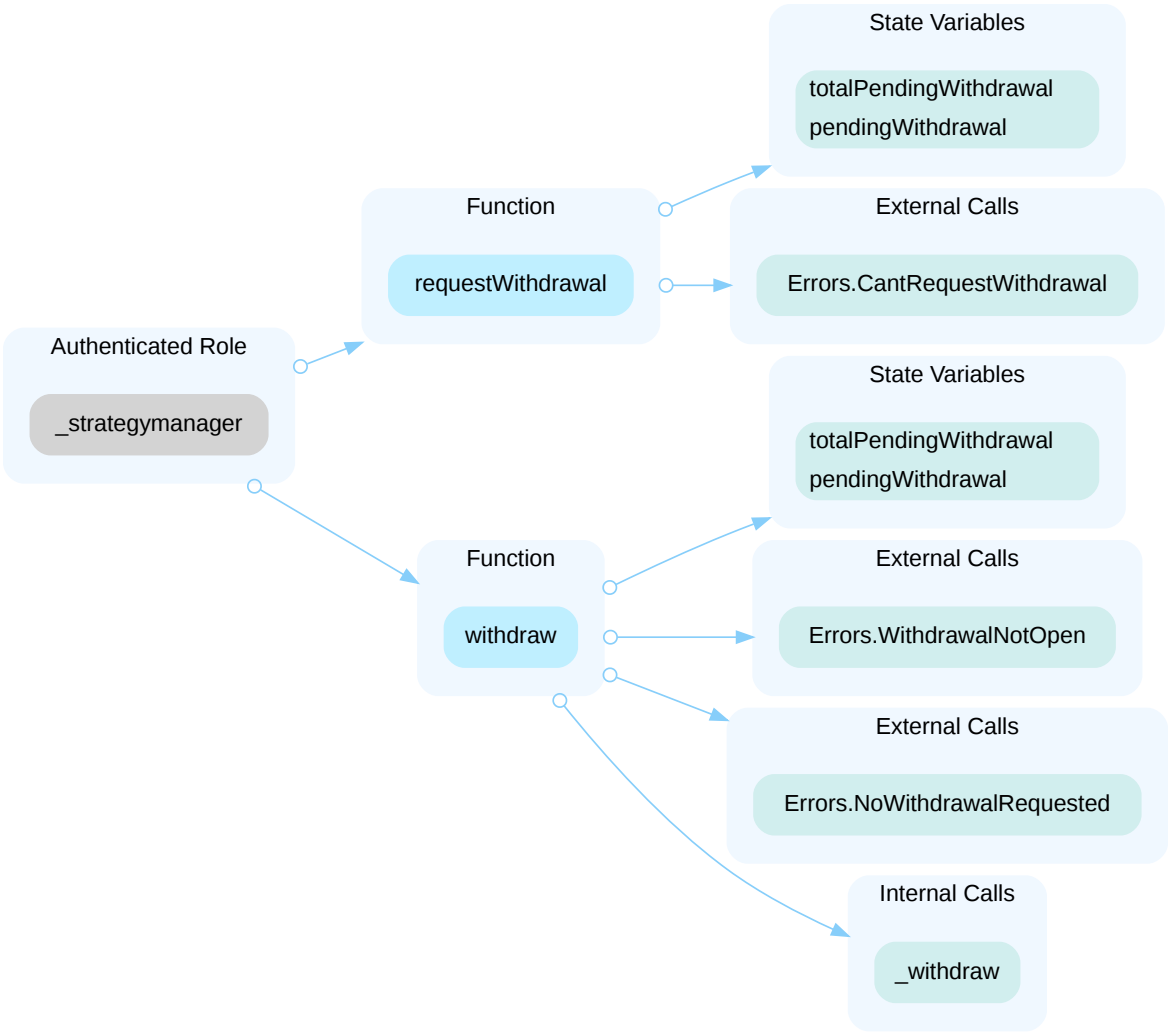
In the contract `BaseStrategy`, the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and set the `_dao` address.



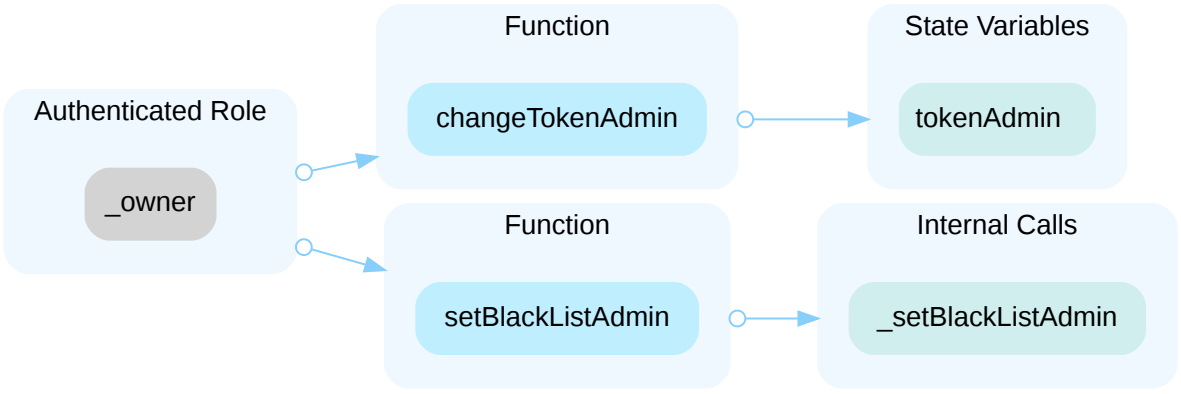
In the contract `BaseStrategy`, the role `_strategyManager` has authority over the functions shown in the diagram below. Any compromise to the `_strategyManager` account may allow the hacker to take advantage of this authority and deposit user funds to the contract, request withdrawal for a user, and withdraw funds for a user.



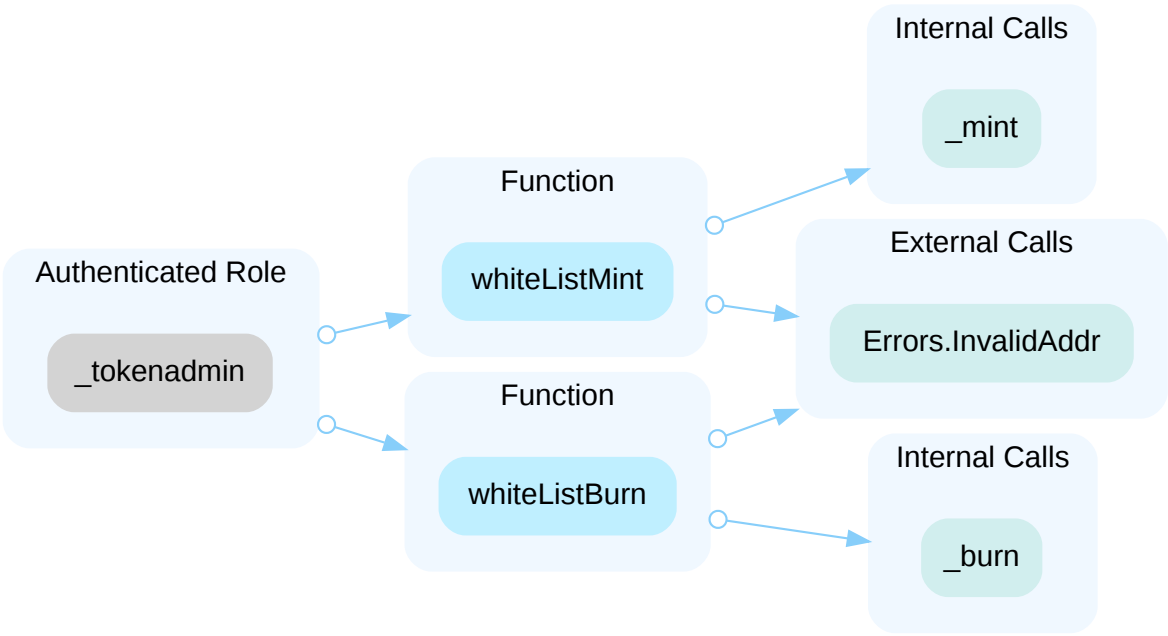
In the contract `CefiStrategy`, the role `_strategyManager` has authority over the functions shown in the diagram below. Any compromise to the `_strategyManager` account may allow the hacker to take advantage of this authority and request user withdrawals or withdraw a user's specified amount.



In the contract `BaseToken`, the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and change the `_tokenAdmin` address, and set the `blackListAdmin` admin.



In the contract `BaseToken`, the role `_tokenAdmin` has authority over the functions shown in the diagram below. Any compromise to the `_tokenAdmin` account may allow the hacker to take advantage of this authority and call `whiteListMint()` to mint to a specified account or burn tokens from an account with `whiteListBurn()`.



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

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

BTC-01 | INHERITED CONTRACTS NOT INITIALIZED IN INITIALIZER

Category	Severity	Location	Status
Logical Issue	Medium	src/core/MintStrategy.sol: 19; src/modules/Version.sol: 26	Acknowledged

Description

Contract Version extends ReentrancyGuardUpgradeable, but the current contract does not initialize the extended contract. Contract `MintStrategy` extends `Whitelisted`, but the current contract does not initialize extended 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.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

WRB-01 | NO CAP ON FEES

Category	Severity	Location	Status
Logical Issue	Medium	src/modules/WithdrawalRequest.sol: 168, 168	Acknowledged

Description

The `nonNativeWithdrawalFee` variables in the contract have no set limits, allowing the withdrawal fees to be set to any value. If the fee is set to a high value the withdrawal fee could cost too much compared to the asset to be withdrawn.

Recommendation

We recommend setting a reasonable cap on fees and providing adequate disclosure to the community.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

BTL-01 | LOCAL VARIABLE SHADOWING

Category	Severity	Location	Status
Coding Style	Minor	lib/openzeppelin-contracts-upgradeable/contracts/security/ReentrancyGuardUpgradeable.sol: 38; src/core/EnzoNetwork.sol: 233	Acknowledged

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.

```
233     function setNativeBTCPausedStatus(bool _status) public onlyDao {
```

- Local variable `_status` in `EnzoNetwork.setNativeBTCPausedStatus` shadows the variable `_status` in `ReentrancyGuardUpgradeable`.

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.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

BTP-01 | MISSING ZERO ADDRESS VALIDATION

Category	Severity	Location	Status
Volatile Code	Minor	src/core/EnzoNetwork.sol: 257, 259; src/core/MintSecurity.sol: 311, 313; src/core/MintStrategy.sol: 230, 232	Acknowledged

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.

```
313         enzoNetwork = _enzoNetwork;
```

- `_enzoNetwork` is not zero-checked before being used.

```
259         mintSecurityAddr = _mintSecurityAddr;
```

- `_mintSecurityAddr` is not zero-checked before being used.

```
232         EnzoNetwork = _EnzoNetwork;
```

- `_EnzoNetwork` is not zero-checked before being used.

Recommendation

It is recommended to add a zero-check for the passed-in address value to prevent unexpected errors.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

BTC-02 | UNDERSCORE PREFIX FOR NON-EXTERNAL VARIABLES

Category	Severity	Location	Status
Code Optimization	Informational	src/core/MintSecurity.sol: 26, 27, 29; src/core/MintStrategy.sol: 29, 30, 33; src/modules/Assets.sol: 15; src/modules/Whitelisted.sol: 15; src/modules/WithdrawalRequest.sol: 32; src/strategies/BaseStrategy.sol: 29, 30, 32, 35	Acknowledged

Description

The current contract doesn't follow the naming convention specified by Solidity DOC:

If the state variable `variable` is used as `private` or `internal` and is not exposed publicly. It should have `an underscore prefix` like `_variable`. Leading underscores allow you to immediately recognize the intent of such functions, but more importantly, if you change a function from non-external to external (including public) and rename it accordingly, this forces you to review every call site while renaming. This can be an important manual check against unintended external functions and a common source of security vulnerabilities (avoid find-replace-all tooling for this change).

Recommendation

To mitigate this issue, it is recommended to follow the naming conventions and rename the variable by adding an underscore prefix.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

BTC-03 | STORAGE SIZE CONVENTION

Category	Severity	Location	Status
Coding Issue	Informational	src/modules/Assets.sol: 114; src/modules/BlackList.sol: 52; src/modules/Dao.sol: 36; src/modules/Whitelisted.sol: 91; src/modules/WithdrawalRequest.sol: 189; src/strategies/BaseStrategy.sol: 267	Acknowledged

Description

While not a requirement, generally each upgradeable contract contains 50 storage slots in total, including already-used storage slots. In the `Blacklist` contract, the `isBlacklisted` and `blackListAdmin` variables take up two slots, and at the end of the contract, the `_gap` variable takes an extra 50 slots, bringing total storage slots to 52.

Similar storage configurations are used for the contracts `Dao`, `Whitelisted`, `WithdrawalRequest`, and `BaseStrategy`.

Recommendation

Consider modifying the `__gap` so that the sum of the storage slots used by it and the storage slots used by the other variable leads to 50 total storage slots.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

VBT-01 CONTRACTS MAY FAIL TO RESUME IF OWNER RENOUNCE OWNERSHIP DURING PAUSE

Category	Severity	Location	Status
Design Issue	Informational	src/modules/Version.sol: 25	Acknowledged

Description

The contract inherits from `Pausable` and `Ownable` at the same time.

If the owner of a smart contract renounces ownership while the contract is paused, it means that there will be no one with the necessary permissions to unpause the contract. This could result in a permanent state of pause, effectively freezing all contract functionality that is dependent on the pause state. It's crucial to design smart contracts with secure ownership transfer mechanisms and emergency procedures to prevent such situations.

Recommendation

To mitigate this issue, modify the `renounceOwnership` function to include a condition that checks whether the contract is paused. If the contract is paused, the function should revert and prevent renouncing ownership.

Alleviation

[Certik, 11/26/2024]: The team acknowledged the issue and decided to remain unchanged in the scope of the audit.

APPENDIX | ENZOBTC

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.
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
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.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

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.

Elevating Your Entire Web3 Journey

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.