Quantstamp Security Assessment Certificate

January 18th 2022 — Quantstamp Verified

Badger (Vault)

This audit report was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

Type	Yield Aggregator
Auditors	Roman Rohleder, Research Engineer Souhail Mssassi, Research Engineer Marius Guggenmos, Senior Research Engineer
Timeline	2021-12-06 through 2021-12-10
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	GitBook
Documentation Quality	High
Test Quality	High

Source Code

Repository	Commit
badger-sett-1.5	7d3b5eb
badger-sett-1.5	b2ea500

Total Issues	**16** (12 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	**2** (2 Resolved)
Low Risk Issues	**5** (4 Resolved)
Informational Risk Issues	**8** (6 Resolved)
Undetermined Risk Issues	**1** (0 Resolved)

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.
Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

After first audit: Quantstamp has performed a security audit of the Badger (Vault) smart contracts and has identified 16 issues ranging informational to medium severity. Additionally, we have found 7 deviations from best practices. Moreover, the test suite only offers 78% coverage. We recommend that branch coverage be 100% to ensure that all code paths are tested at least once. We also strongly recommend addressing all the identified issues before deploying this project in production.
After re-audit: Quantstamp has updated the report based on the responses received from the dev team. All findings have been addressed. 12 issues have been fixed, one of which mitigated. The remaining findings have been acknowledged and the corresponding acknowledgements have been attached. Slither reported 41 less findings and the coverage of the audited contracts improved from 60-80% to over 90%. 2 of the 7 best practice recommendations have been implemented. Further, documentation was added.

ID	Description	Severity	Status
QSP-1	Reentrancy	^ Medium	Fixed
QSP-2	Front run on Vault contract	^ Medium	Mitigated
QSP-3	Integer Overflow / Underflow	⌄ Low	Fixed
QSP-4	Missing input validation	⌄ Low	Fixed
QSP-5	Privileged Roles and Ownership	⌄ Low	Fixed
QSP-6	Missing initializer calls to parent contracts	⌄ Low	Fixed
QSP-7	Imprecise share calculation	⌄ Low	Acknowledged
QSP-8	Unlocked Pragma	○ Informational	Fixed
QSP-9	Events not emitted on state change	○ Informational	Fixed
QSP-10	Gas Usage / `for` Loop Concerns	○ Informational	Acknowledged
QSP-11	Block Timestamp Manipulation	○ Informational	Acknowledged
QSP-12	Unused Imports	○ Informational	Fixed
QSP-13	Function does not return a value	○ Informational	Fixed
QSP-14	Redundant check in if...else if statement	○ Informational	Fixed
QSP-15	Goal of min variable unclear	○ Informational	Fixed
QSP-16	Token-transfering functions executable during pause	? Undetermined	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

DISCLAIMER: Only contracts ` contracts/Vault.sol` and ` contracts/BaseStrategy.sol` were in scope of this audit. Even though ` contracts/Vault.sol` inherits from ` contracts/lib/SettAccessControl.sol` it was not part of the audit. Further note ` contracts/BaseStrategy.sol` is an abstract contract and hence can not be used in and by itself. In practice, concrete strategy contracts will inherit from it, which may introduce additional code or override existing functions. These final contracts were out of scope for this audit.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.8.1

Steps taken to run the tools:

Installed the Slither tool: `pip install slither-analyzer` Run Slither from the project directory: `slither .`

Findings

QSP-1 Reentrancy

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `contracts/Vault.sol`

Related Issue(s): SWC-107

Description: A reentrancy vulnerability is a scenario where an attacker can repeatedly call a function from itself, unexpectedly leading to potentially disastrous results. Here's a basic example representing the very attack which impacted The DAO in 2016:

```
    function withdraw_with_reentrancy(uint256 _amount) public {
        msg.sender.call.value(_amount)();   // ATTACKER CAN CALL AGAIN BEFORE
                                            // FUNCTION TERMINATES because `call`
                    // allows msg.sender to execute any code using fallback function
        balances[msg.sender] -= _amount;    // Balance only updated AFTER funds withdrawn
    }
```

The following functions do not follow the `Checks-Effects-Interactions` pattern, as they perform calls to external contracts before changing state variables, while at the same time not being protected through the elsewhere used `nonReentrant` modifier:

- `Vault._depositFor()`:
  . performs direct and indirect calls to external contracts by calling `balance()`, `token.balanceOf()` and `token.safeTransferFrom()` before changing state through `_mintSharesFor()`.

Recommendation: Use the OpenZeppelin ReentrancyGuard.nonReentrant modifier, as was already used in i.e. `Vault._deposit()`.

Update: Fixed by adding the `nonReentrant` modifier to `Vault._depositFor()`, as was suggested.

QSP-2 Front run on Vault contract

Severity: *Medium Risk*

Status: Mitigated

File(s) affected: `contracts/Vault.sol`

Description: In the `initialize` function of the Vault contract some important variables are initialized. The problem is that there is no restriction in place that prevents others from front-running the initialization and specifying their own addresses.

- The issue is located in the `initialize` function of the Vault contract.

Recommendation: Add a require to verify that only the Owner is the one who has the right to call this function.

Update: Mitigated by using OpenZeppelin's `AdminUpgradeabilityProxy`.

QSP-3 Integer Overflow / Underflow

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/Vault.sol`

Related Issue(s): SWC-101

Description: Integer overflow/underflow occur when an integer hits its bit-size limit. Every integer has a set range; when that range is passed, the value loops back around. A clock is a good analogy: at 11:59, the minute hand goes to 0, not 60, because 59 is the largest possible minute. Integer overflow and underflow may cause many unexpected kinds of behavior and was the core reason for the `batchOverflow` attack. Here's an example with `uint8` variables, meaning unsigned integers with a range of `0..255`. `function under_over_flow() public { uint8 num_players = 0; num_players = num_players - 1; // 0 - 1 now equals 255! if (num_players == 255) { emit LogUnderflow(); // underflow occurred } uint8 jackpot = 255; jackpot = jackpot + 1; // 255 + 1 now equals 0! if (jackpot == 0) { emit LogOverflow(); // overflow occurred } }`
Although OpenZeppelin's `SafeMath*` library is imported and used throughout the code, some unsafe artihmetic operations are still performed in the code. This is for example the case in L285, L301 and L591 of `Vault.sol`

Recommendation: Replace the unsafe arithmetic operations with their safe counterparts from the OpenZeppelin `SafeMath*` library (i.e. `a.add(b)`, instead of `a + b`, ...).

Update: Fixed by using OpenZEppelin's safe arithmetic counterparts, as suggested.

QSP-4 Missing input validation

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/Vault.sol`, `contracts/BaseStrategy.sol`

Description: It is important to validate inputs, even if they only come from trusted addresses, to avoid human error. The following functions do not have a proper validation of input parameters:

1. `Vault.initialize()` does not check that parameters `_keeper`, `_governance`, `_treasury`, `_strategist`, `_guardian` or `_badgerTree` are different from `address(0)`.
2. `Vault.deposit()` in L213 and L218 does not check that parameter `_amount` is non-zero.
3. `Vault.depositFor()` in L233 and L238 does not check that parameter `_recipient` is different from `address(0)` and `_amount` is non-zero.
4. `Vault.sweepExtraToken()` does not check that parameter `_token` is different from `address(0)`.
5. `Vault.emitNonProtectedToken()` does not check that parameter `_token` is different from `address(0)`.
6. `Vault._withdraw()` does not check that parameter `_shares` is non-zero.
7. `Vault._mintSharesFor()` does not check that parameter `recipient` is different from `address(0)` and `_amount` and `_pool` are non-zero.
8. `Vault._handleFees()` does not check that parameters `_harvestedAmount` and `harvestTime` are non-zero.
9. `Vault.setTreasury()` does not check that parameter `_treasury` is different from `address(0)`.
10. `BaseStrategy.isProtectedToken()` does not check that parameter `token` is different from `address(0)`.
11. `BaseStrategy.withdraw()` does not check that parameter `_amount` is non-zero.
12. `BaseStrategy.emitNonProtectedToken()` does not check that parameter `_token` is different from `address(0)`.
13. `BaseStrategy.withdrawOther()` does not check that parameter `_asset` is different from `address(0)`.
14. `BaseStrategy._processExtraToken()` does not check that parameter `_token` is different from `address(0)` and `_amount` is non-zero.
15. `BaseStrategy.__BaseStrategy_init()` does not check that parameter `_vault` is different from `address(0)`.

Recommendation: Add corresponding checks to these input parameters.

Update: Fixed by adding corresponding checks, as suggested.

QSP-5 Privileged Roles and Ownership

Severity: *Low Risk*

Status: Fixed

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract. The `governance` role, as setup during initialization of the `Vault` contract, may perform the following privileged actions:

1. Appoint a new `governance`, by calling `Vault.setGovernance()` with the new address, or renounce this role, by setting it to an uncontrolled address, i.e. `address(0)` and thereby block all followingly listed actions for the future.
2. Appoint a new `strategist` or unset it, by calling `Vault.setStrategist()`.
3. Appoint a new `keeper` or unset it, by calling `Vault.setKeeper()`.
4. Appoint a new `treasury` or unset it, by calling `Vault.setTreasury()`.
5. Change the `strategy` address or unset it, by calling `Vault.setStrategy()`.
6. Modify the `min` state variable, by calling `Vault.setMin()`.
7. Modify the `maxWithdrawalFee` state variable, by calling `Vault.setMaxWithdrawalFee()`.
8. Modify the `maxPerformanceFee` state variable, by calling `Vault.setMaxPerformanceFee()`.
9. Modify the `maxManagementFee` state variable, by calling `Vault.setMaxManagementFee()`.
10. Change the `guardian` address or unset it, by calling `Vault.setGuardian()`.
11. Change or unset the `guestList`, by calling `Vault.setGuestList()`.
12. Change or unset the `withdrawalFee`, by calling `Vault.setWithdrawalFee()`.
13. Change or unset the `performanceFeeStrategist`, by calling `Vault.setPerformanceFeeStrategist()`.
14. Change or unset the `performanceFeeGovernance`, by calling `Vault.setPerformanceFeeGovernance()`.
15. Change or unset the `managementFee`, by calling `Vault.setManagementFee()`.
16. Withdraw all funds from the strategy to the vault, by calling `Vault.withdrawToVault()`.
17. Withdraw all funds of a specific non-protected token from the strategy to himself. by calling `Vault.sweepExtraToken()`.
18. Initiate the `pause` state, freezing most operations, by calling `Vault.pause()`.
19. Transition into `unpaused` state, unfreezing most operations from the paused state, by calling `Vault.unpause()`.
20. Freeze or unfreeze the ability to perform deposits, by calling `Vault.pauseDeposits()` or `Vault.unpauseDeposits()`.
21. Send all available underlying tokens to the strategy, by calling `Vault.earn()`.

Similarly, the `strategist` address is able to:

1. Change or unset the `guestList`, by calling `Vault.setGuestList()`.
2. Change or unset the `withdrawalFee`, by calling `Vault.setWithdrawalFee()`.
3. Change or unset the `performanceFeeStrategist`, by calling `Vault.setPerformanceFeeStrategist()`.
4. Change or unset the `performanceFeeGovernance`, by calling `Vault.setPerformanceFeeGovernance()`.
5. Change or unset the `managementFee`, by calling `Vault.setManagementFee()`.
6. Withdraw all funds from the strategy to the vault, by calling `Vault.withdrawToVault()`.
7. Withdraw all funds of a specific non-protected token from the strategy to `governance`. by calling `Vault.sweepExtraToken()`.

The `keeper` address is able to:

1. Send all available underlying tokens to the strategy, by calling `Vault.earn()`.

The `guardian` address can:

1. Initiate the `pause` state or pause deposits, by calling `Vault.pause()` and `Vault.pauseDeposits()`, respectively.

Recommendation: This centralization of power needs to be made clear to the users, for example in the public facing documentation.

Update: Acknowledged with: `We acknowledged that governance has the ability to manage the strategy and potentially put user funds at risk, we will be using a timelock to ensure depositors can react to potentially breaking changes Additionally we'll be progressively decentralise these operation by giving ownership of the timelock to a governor contract (similar to Compounds Governance)`

QSP-6 Missing initializer calls to parent contracts

Severity: *Low Risk*

Status: Fixed

Description: The only initializer that is called from the parent contracts is `__ERC20_init`. However, there are also initializers for `PausableUpgradeable` and `ReentrancyGuardUpgradeable` that are not called. While this does not seem to be an issue at the time, it might lead to issues if these contracts ever change.

Recommendation: Add calls to all initializers of inherited contracts.

Update: Fixed by adding calls to `__Pausable_init()` and `__ReentrancyGuard_init()` at `Vault.initialize()`, as suggested.

QSP-7 Imprecise share calculation

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `contracts/Vault.sol`

Description: The shares computation in `_mintSharesFor` uses integer math without extra precision to calculate the number of shares to mint. This results in rounding errors that lead to fewer shares being issues than intended.

Recommendation: Consider adding a static multiplier to improve the precision of the shares calculation.

Update: Acknowledged with: `Agree that the imprecision can cause the loss of up to 1 share or up to 1 Wei of token, this is due to integer rounding. However the impact is quite literally limited to 1 Wei as we are multiplying first and dividing only at the end.`

QSP-8 Unlocked Pragma

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/Vault.sol`, `contracts/BaseStrategy.sol`

Related Issue(s): SWC-103

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.4.*`. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, it is recommended to remove the caret to lock the file onto a specific Solidity version.

Update: Fixed by locking all pragmas to one version (0.6.12), as suggested.

QSP-9 Events not emitted on state change

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/Vault.sol`, `contracts/BaseStrategy.sol`

Description: An event should always be emitted when a state change is performed in order to facilitate smart contract monitoring by other systems which want to integrate with the smart contract. This is not the case for the functions and the correspondingly modified state variables:

1. `Vault.setStrategy()`, `strategy`
2. `Vault.setMin()`, `min`
3. `Vault.setMaxWithdrawalFee()`, `maxWithdrawalFee`
4. `Vault.setMaxPerformanceFee()`, `maxPerformanceFee`
5. `Vault.setMaxManagementFee()`, `maxManagementFee`
6. `Vault.setGuardian()`, `guardian`
7. `Vault.setGuestList()`, `guestList`
8. `Vault.setWithdrawalFee()`, `withdrawalFee`
9. `Vault.setPerformanceFeeStrategist()`, `performanceFeeStrategist`
10. `Vault.setPerformanceFeeGovernance()`, `performanceFeeGovernance`
11. `Vault.setManagementFee()`, `managementFee`
12. `Vault.pauseDeposits()`, `pausedDeposit`
13. `Vault.unpauseDeposits()`, `pausedDeposit`
14. `BaseStrategy.setWithdrawalMaxDeviationThreshold()`, `withdrawalMaxDeviationThreshold`

Recommendation: Consider emitting an event in the aforementioned functions.

Update: Fixed by introducing new events and emitting them accordingly, as suggested.

QSP-10 Gas Usage / `for` Loop Concerns

Severity: *Informational*

Status: Acknowledged

File(s) affected: `contracts/BaseStrategy.sol`

Description: Gas usage is a main concern for smart contract developers and users, since high gas costs may prevent users from wanting to use the smart contract. Even worse, some gas usage issues may prevent the contract from providing services entirely. For example, if a `for` loop requires too much gas to exit, then it may prevent the contract from functioning correctly entirely. It is best to break such loops into individual functions as possible. In L131 of `BaseStrategy.sol` a for-loop is used to iterate over an array of protected tokens, retrieved via `BaseStrategy.getProtectedTokens()`. For exceptionally large arrays, this loop may run out of gas during iteration.

Recommendation: Add a check on the array length, to prevent arrays larger than a set maximum being processed. This maximum can be determined by performing gas analysis and/or consider mentioning this limitation in the corresponding developer guides.

Update: Acknowledged with: `We'll be keeping it that way to keep it easy for strategists While most likely a high limit of 100 tokens could cause gas issues, I would be surprised to have more than 10 protected tokens per strategy`

QSP-11 Block Timestamp Manipulation

Severity: *Informational*

Status: Acknowledged

Description: Projects may rely on block timestamps for various purposes. However, it's important to realize that miners individually set the timestamp of a block, and attackers may be able to manipulate timestamps for their own purposes. If a smart contract relies on a timestamp, it must take this into account.

Recommendation: The concrete usages in `contracts/Vault.sol` are not particularly problematic, but these limitations should be taken into account.

Update: Acknowledged with: `We believe the maximum manipulation reasonably expected should be around the avg block time, the code uses the timestamps for off chain monitoring and we can confirm that even minutes or hours should not have any dramatic impact on the math`

QSP-12 Unused Imports

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/BaseStrategy.sol`

Description: Code from `SettAccessControl` has been integrated into `BaseStrategy` and the import can be removed.

Recommendation: Remove the line `import "./lib/SettAccessControl.sol";`.

Update: Fixed by removing the unused import, as suggested.

QSP-13 Function does not return a value

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/BaseStrategy.sol`

Description: The declaration of `withdrawToVault` specifies that it returns `(uint256 balance)` but there is neither a return statement nor an assignment to `balance`.

Recommendation: Either remove the return value from the signature or actually return a value and check for it at the corresponding call-sites.

Update: Fixed by saving the return value of `balanceOf` and returning it, as suggested.

QSP-14 Redundant check in if...else if statement

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/Vault.sol`

Description: Line 278-283 contains the following code.

```
if (assetsAtHarvest != 0) {
    assetsAtLastHarvest = assetsAtHarvest;
} else if (assetsAtHarvest == 0 && _harvestedAmount == 0) {
    // If zero
    assetsAtLastHarvest = 0;
}
```

The check for `assetsAtHarvest == 0` in the `else if` branch can be removed since if it were not equal to 0 the first branch would be taken.

Recommendation: Remove the redundant check.

Update: Fixed by removing the redundant check of `assetsAtHarvest`, as suggested.

QSP-15 Goal of min variable unclear

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/Vault.sol`

Description: According to the comments on the `setMin` function, the variable `min` is supposed to be the minimum percentage of `want` tokens that should be used for earning in a strategy. However, in the current implementation it does not function as a typical minimum. Instead, it denotes the actual amount of the current balance that will be sent to the strategy for earning whenever `earn` is called.

Recommendation: Consider rethinking what the variable is supposed to do, specify the intention in a comment and rename the variable to something more specific.

Update: Fixed by renaming variable `min` to `toEarnBps`, as discussed.

QSP-16 Token-transfering functions executable during pause

Severity: *Undetermined*

Status: Acknowledged

File(s) affected: `contracts/Vault.sol`, `contracts/BaseStrategy.sol`

Description: Although the `Vault` contract inherits from `PausableUpgradeable` and makes use of the `whenNotPaused` modifier on most functions, some functions remain without this modifier, allowing them to be executed, even when in the `paused` state. In particular, the following functions are affected:

- `Vault.withdrawToVault()`
- `Vault.sweepExtraToken()`
- `Vault.emitNonProtectedToken()`
- `BaseStrategy.emitNonProtectedToken()`
- `BaseStrategy._transferToVault()`

- `BaseStrategy._processExtraToken()`

As well as the following purely virtual function definitions

- `BaseStrategy._withdrawAll()`
- `BaseStrategy._withdrawSome()`
- `BaseStrategy.harvest()`

Recommendation: Clarify whether or not this is intended behaviour and, if necessary, add the `whenNotPaused` modifier to said functions.

Update: Acknowledged with:
We have made changes to the access to allow the following:
Pausing the Vault blocks:

- Deposits
- Withdrawals

It allows:

- Earning to the Strat
- Harvesting the strat
- Bringing funds back to the strat
- Emitting reward tokens

Pausing the Strategy blocks:

- Earning to the strat
- Any type of investing
- Harvesting and Tending

It still allows:

- To move funds back to the Vault (withdrawAll)

Automated Analyses

Slither

We have run the latest version of the slither analyzer on this repository's solidity code. The tool has identified 181 issues, most of which were filtered out as false positives. The remaining issues were integrated in the findings of this report.

```
Compilation warnings/errors on ./contracts/BaseStrategy.sol:
Warning: This declaration shadows an existing declaration.
   --> ./contracts/BaseStrategy.sol:212:9:
    |
212 |         uint256 balance = IERC20Upgradeable(want).balanceOf(address(this));
    |         ^^^^^^^^^^^^^^^
Note: The shadowed declaration is here:
   --> ./contracts/BaseStrategy.sol:207:50:
    |
207 |     function withdrawToVault() external returns (uint256 balance) {
    |                                                  ^^^^^^^^^^^^^^^



BaseStrategy.__gap (contracts/BaseStrategy.sol#381) shadows:
    - PausableUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#96)
    - ContextUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31)
PausableUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#96) shadows:
    - ContextUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

BaseStrategy.want (contracts/BaseStrategy.sol#46) is never initialized. It is used in:
    - BaseStrategy.balanceOfWant() (contracts/BaseStrategy.sol#122-124)
    - BaseStrategy.deposit() (contracts/BaseStrategy.sol#191-197)
    - BaseStrategy.withdrawToVault() (contracts/BaseStrategy.sol#207-216)
    - BaseStrategy.withdraw(uint256) (contracts/BaseStrategy.sol#220-242)
    - BaseStrategy._transferToVault(uint256) (contracts/BaseStrategy.sol#289-293)
    - BaseStrategy._processExtraToken(address,uint256) (contracts/BaseStrategy.sol#309-316)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

AddressUpgradeable.isContract(address) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#26-35) uses assembly
    - INLINE ASM (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#147-164) uses assembly
    - INLINE ASM (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
    - Version used: ['0.6.12', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
    - 0.6.12 (contracts/BaseStrategy.sol#3)
    - ABIEncoderV2 (contracts/BaseStrategy.sol#4)
    - 0.6.12 (interfaces/badger/IVault.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/MathUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#3)
    - >=0.4.24<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/proxy/Initializable.sol#4)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/IERC20Upgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#3)
    - >=0.6.2<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#79-81) is never used and should be
removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#104-106) is never used
and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#129-131) is never used and should
be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#139-145) is never used and
should be removed
AddressUpgradeable.sendValue(address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
BaseStrategy._onlyGovernanceOrStrategist() (contracts/BaseStrategy.sol#90-92) is never used and should be removed
BaseStrategy._processExtraToken(address,uint256) (contracts/BaseStrategy.sol#309-316) is never used and should be removed
BaseStrategy._reportToVault(uint256) (contracts/BaseStrategy.sol#297-301) is never used and should be removed
ContextUpgradeable.__Context_init() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/MathUpgradeable.sol#27-30) is never used and should be removed
```

```
MathUpgradeable.max(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/MathUpgradeable.sol#12-14) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#37-46) is
never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20
/SafeERC20Upgradeable.sol#53-56) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20
/SafeERC20Upgradeable.sol#48-51) is never used and should be removed
SafeERC20Upgradeable.safeTransferFrom(IERC20Upgradeable,address,address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20
/SafeERC20Upgradeable.sol#26-28) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#190-193) is never used and should be
removed
SafeMathUpgradeable.mod(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#210-213) is never used and should be
removed
SafeMathUpgradeable.sub(uint256,uint256,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#170-173) is never used and should be
removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/MathUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/IERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#53-59):
    - (success) = recipient.call{value: amount}() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#57)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils
/AddressUpgradeable.sol#114-121):
    - (success,returndata) = target.call{value: value}(data) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#119)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#139-145):
    - (success,returndata) = target.staticcall(data) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#143)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function BaseStrategy.__BaseStrategy_init(address) (contracts/BaseStrategy.sol#70-79) is not in mixedCase
Parameter BaseStrategy.__BaseStrategy_init(address)._vault (contracts/BaseStrategy.sol#70) is not in mixedCase
Parameter BaseStrategy.setWithdrawalMaxDeviationThreshold(uint256)._threshold (contracts/BaseStrategy.sol#177) is not in mixedCase
Parameter BaseStrategy.withdraw(uint256)._amount (contracts/BaseStrategy.sol#220) is not in mixedCase
Parameter BaseStrategy.emitNonProtectedToken(address)._token (contracts/BaseStrategy.sol#252) is not in mixedCase
Parameter BaseStrategy.withdrawOther(address)._asset (contracts/BaseStrategy.sol#262) is not in mixedCase
Variable BaseStrategy.__gap (contracts/BaseStrategy.sol#381) is not in mixedCase
Function ContextUpgradeable.__Context_init() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#33-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#96) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (../../../../../../../../../.brownie
/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

BaseStrategy (contracts/BaseStrategy.sol#39-382) does not implement functions:
    - BaseStrategy._deposit(uint256) (contracts/BaseStrategy.sol#328)
    - BaseStrategy._harvest() (contracts/BaseStrategy.sol#357)
    - BaseStrategy._isTendable() (contracts/BaseStrategy.sol#137)
    - BaseStrategy._tend() (contracts/BaseStrategy.sol#365)
    - BaseStrategy._withdrawAll() (contracts/BaseStrategy.sol#341)
    - BaseStrategy._withdrawSome(uint256) (contracts/BaseStrategy.sol#347)
    - BaseStrategy.balanceOfPool() (contracts/BaseStrategy.sol#374)
    - BaseStrategy.balanceOfRewards() (contracts/BaseStrategy.sol#379)
    - BaseStrategy.getName() (contracts/BaseStrategy.sol#370)
    - BaseStrategy.getProtectedTokens() (contracts/BaseStrategy.sol#338)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

BaseStrategy.__gap (contracts/BaseStrategy.sol#381) is never used in BaseStrategy (contracts/BaseStrategy.sol#39-382)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

BaseStrategy.want (contracts/BaseStrategy.sol#46) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

__BaseStrategy_init(address) should be declared external:
    - BaseStrategy.__BaseStrategy_init(address) (contracts/BaseStrategy.sol#70-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
./contracts/BaseStrategy.sol analyzed (10 contracts with 75 detectors), 60 result(s) found


Compilation warnings/errors on ./contracts/Vault.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off revert strings,
or using libraries.
  --> ./contracts/Vault.sol:60:1:
   |
60 | contract Vault is ERC20Upgradeable, SettAccessControl, PausableUpgradeable, ReentrancyGuardUpgradeable {
   | ^ (Relevant source part starts here and spans across multiple lines).



ERC20Upgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#312) shadows:
    - ContextUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31)
PausableUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#96) shadows:
    - ContextUpgradeable.__gap (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Vault._mintSharesFor(address,uint256,uint256) (contracts/Vault.sol#645-657) uses a dangerous strict equality:
    - totalSupply() == 0 (contracts/Vault.sol#651)
Vault.getPricePerFullShare() (contracts/Vault.sol#230-235) uses a dangerous strict equality:
    - totalSupply() == 0 (contracts/Vault.sol#231)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Vault._withdraw(uint256) (contracts/Vault.sol#594-619):
    External calls:
    - IStrategy(strategy).withdraw(_toWithdraw) (contracts/Vault.sol#604)
    - token.safeTransfer(msg.sender,r.sub(_fee)) (contracts/Vault.sol#614)
    State variables written after the call(s):
    - _mintSharesFor(treasury,_fee,balance().sub(_fee)) (contracts/Vault.sol#618)
        - _balances[account] = _balances[account].add(amount) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#240)
    - _mintSharesFor(treasury,_fee,balance().sub(_fee)) (contracts/Vault.sol#618)
        - _totalSupply = _totalSupply.add(amount) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Vault.withdrawToVault() (contracts/Vault.sol#493-496) ignores return value by IStrategy(strategy).withdrawToVault() (contracts/Vault.sol#495)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._name (contracts/Vault.sol#148) shadows:
    - ERC20Upgradeable._name (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#43) (state variable)
Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._symbol (contracts/Vault.sol#149) shadows:
    - ERC20Upgradeable._symbol (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#44) (state variable)
Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4]).name (contracts/Vault.sol#166) shadows:
    - ERC20Upgradeable.name() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#70-72) (function)
Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4]).symbol (contracts/Vault.sol#167) shadows:
    - ERC20Upgradeable.symbol() (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#78-80) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

SettAccessControl.setStrategist(address)._strategist (contracts/lib/SettAccessControl.sol#31) lacks a zero-check on :
        - strategist = _strategist (contracts/lib/SettAccessControl.sol#33)
SettAccessControl.setKeeper(address)._keeper (contracts/lib/SettAccessControl.sol#38) lacks a zero-check on :
        - keeper = _keeper (contracts/lib/SettAccessControl.sol#40)
SettAccessControl.setGovernance(address)._governance (contracts/lib/SettAccessControl.sol#45) lacks a zero-check on :
        - governance = _governance (contracts/lib/SettAccessControl.sol#47)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in Vault._depositFor(address,uint256) (contracts/Vault.sol#565-575):
    External calls:
    - token.safeTransferFrom(msg.sender,address(this),_amount) (contracts/Vault.sol#572)
    State variables written after the call(s):
    - _mintSharesFor(_recipient,_after.sub(_before),_pool) (contracts/Vault.sol#574)
        - _balances[account] = _balances[account].add(amount) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#240)
    - _mintSharesFor(_recipient,_after.sub(_before),_pool) (contracts/Vault.sol#574)
        - _totalSupply = _totalSupply.add(amount) (../../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
Reentrancy in Vault._depositFor(address,uint256) (contracts/Vault.sol#565-575):
    External calls:
    - token.safeTransferFrom(msg.sender,address(this),_amount) (contracts/Vault.sol#572)
    Event emitted after the call(s):
    - Transfer(address(0),account,amount) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#241)
        - _mintSharesFor(_recipient,_after.sub(_before),_pool) (contracts/Vault.sol#574)
Reentrancy in Vault._withdraw(uint256) (contracts/Vault.sol#594-619):
    External calls:
    - IStrategy(strategy).withdraw(_toWithdraw) (contracts/Vault.sol#604)
    - token.safeTransfer(msg.sender,r.sub(_fee)) (contracts/Vault.sol#614)
    Event emitted after the call(s):
    - Transfer(address(0),account,amount) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#241)
        - _mintSharesFor(treasury,_fee,balance().sub(_fee)) (contracts/Vault.sol#618)
Reentrancy in Vault.reportAdditionalToken(address) (contracts/Vault.sol#338-356):
    External calls:
    - IERC20Upgradeable(_token).safeTransfer(treasury,governanceRewardsFee) (contracts/Vault.sol#349)
    - IERC20Upgradeable(_token).safeTransfer(strategist,strategistRewardsFee) (contracts/Vault.sol#350)
    - IERC20Upgradeable(_token).safeTransfer(badgerTree,newBalance) (contracts/Vault.sol#354)
    Event emitted after the call(s):
    - TreeDistribution(_token,newBalance,block.number,block.timestamp) (contracts/Vault.sol#355)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Vault.getPricePerFullShare() (contracts/Vault.sol#230-235) uses timestamp for comparisons
    Dangerous comparisons:
    - totalSupply() == 0 (contracts/Vault.sol#231)
Vault._withdraw(uint256) (contracts/Vault.sol#594-619) uses timestamp for comparisons
    Dangerous comparisons:
    - require(bool,string)(_shares != 0,0 Shares) (contracts/Vault.sol#595)
    - b < r (contracts/Vault.sol#602)
    - _diff < _toWithdraw (contracts/Vault.sol#607)
Vault._mintSharesFor(address,uint256,uint256) (contracts/Vault.sol#645-657) uses timestamp for comparisons
    Dangerous comparisons:
    - totalSupply() == 0 (contracts/Vault.sol#651)
Vault._handleFees(uint256,uint256) (contracts/Vault.sol#660-682) uses timestamp for comparisons
    Dangerous comparisons:
    - totalGovernanceFee != 0 (contracts/Vault.sol#674)
    - feeStrategist != 0 && strategist != address(0) (contracts/Vault.sol#678)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.isContract(address) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#26-35) uses assembly
    - INLINE ASM (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#147-164) uses assembly
    - INLINE ASM (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
    - Version used: ['0.6.12', '>=0.4.24<0.8.0', '>=0.6.0<0.7.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
    - 0.6.12 (contracts/Vault.sol#3)
    - 0.6.12 (contracts/lib/SettAccessControl.sol#2)
    - 0.6.12 (interfaces/badger/IStrategy.sol#3)
    - ABIEncoderV2 (interfaces/badger/IStrategy.sol#4)
    - 0.6.12 (interfaces/badger/IVault.sol#3)
    - 0.6.12 (interfaces/erc20/IERC20Detailed.sol#3)
    - >=0.6.0<0.7.0 (interfaces/yearn/BadgerGuestlistApi.sol#2)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#3)
    - >=0.4.24<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/proxy/Initializable.sol#4)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/IERC20Upgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#3)
    - >=0.6.2<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#3)
    - >=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#79-81) is never used and should be
removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#104-106) is never used
and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#129-131) is never used and should
be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#139-145) is never used and
should be removed
AddressUpgradeable.sendValue(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
ERC20Upgradeable._setupDecimals(uint8) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#293-295) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#37-46) is
never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20
/SafeERC20Upgradeable.sol#53-56) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20
/SafeERC20Upgradeable.sol#48-51) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#190-193) is never used and should be
removed
SafeMathUpgradeable.mod(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#210-213) is never used and should be
removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.7.0 (interfaces/yearn/BadgerGuestlistApi.sol#2) allows old versions
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/IERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/SafeERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#53-59):
    - (success) = recipient.call{value: amount}() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#57)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils
/AddressUpgradeable.sol#114-121):
    - (success,returndata) = target.call{value: value}(data) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#119)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#139-145):
    - (success,returndata) = target.staticcall(data) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/AddressUpgradeable.sol#143)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._token (contracts/Vault.sol#141) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._governance (contracts/Vault.sol#142) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._keeper (contracts/Vault.sol#143) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._guardian (contracts/Vault.sol#144) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._treasury (contracts/Vault.sol#145) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._strategist (contracts/Vault.sol#146) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._badgerTree (contracts/Vault.sol#147) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._name (contracts/Vault.sol#148) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._symbol (contracts/Vault.sol#149) is not in mixedCase
Parameter Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4])._feeConfig (contracts/Vault.sol#150) is not in mixedCase
Parameter Vault.deposit(uint256)._amount (contracts/Vault.sol#253) is not in mixedCase
Parameter Vault.deposit(uint256,bytes32[])._amount (contracts/Vault.sol#258) is not in mixedCase
Parameter Vault.depositFor(address,uint256)._recipient (contracts/Vault.sol#273) is not in mixedCase
Parameter Vault.depositFor(address,uint256)._amount (contracts/Vault.sol#273) is not in mixedCase
Parameter Vault.depositFor(address,uint256,bytes32[])._recipient (contracts/Vault.sol#279) is not in mixedCase
Parameter Vault.depositFor(address,uint256,bytes32[])._amount (contracts/Vault.sol#280) is not in mixedCase
Parameter Vault.withdraw(uint256)._shares (contracts/Vault.sol#287) is not in mixedCase
Parameter Vault.reportHarvest(uint256)._harvestedAmount (contracts/Vault.sol#302) is not in mixedCase
Parameter Vault.reportAdditionalToken(address)._token (contracts/Vault.sol#338) is not in mixedCase
Parameter Vault.setTreasury(address)._treasury (contracts/Vault.sol#361) is not in mixedCase
Parameter Vault.setStrategy(address)._strategy (contracts/Vault.sol#373) is not in mixedCase
Parameter Vault.setMaxWithdrawalFee(uint256)._fees (contracts/Vault.sol#390) is not in mixedCase
Parameter Vault.setMaxPerformanceFee(uint256)._fees (contracts/Vault.sol#400) is not in mixedCase
Parameter Vault.setMaxManagementFee(uint256)._fees (contracts/Vault.sol#410) is not in mixedCase
Parameter Vault.setGuardian(address)._guardian (contracts/Vault.sol#420) is not in mixedCase
Parameter Vault.setToEarnBps(uint256)._newToEarnBps (contracts/Vault.sol#433) is not in mixedCase
Parameter Vault.setGuestList(address)._guestList (contracts/Vault.sol#443) is not in mixedCase
Parameter Vault.setWithdrawalFee(uint256)._withdrawalFee (contracts/Vault.sol#452) is not in mixedCase
Parameter Vault.setPerformanceFeeStrategist(uint256)._performanceFeeStrategist (contracts/Vault.sol#461) is not in mixedCase
Parameter Vault.setPerformanceFeeGovernance(uint256)._performanceFeeGovernance (contracts/Vault.sol#471) is not in mixedCase
Parameter Vault.setManagementFee(uint256)._fees (contracts/Vault.sol#480) is not in mixedCase
Parameter Vault.emitNonProtectedToken(address)._token (contracts/Vault.sol#500) is not in mixedCase
Parameter Vault.sweepExtraToken(address)._token (contracts/Vault.sol#507) is not in mixedCase
Constant Vault._defaultNamePrefix (contracts/Vault.sol#81) is not in UPPER_CASE_WITH_UNDERSCORES
```

```
Constant Vault._symbolSymbolPrefix (contracts/Vault.sol#82) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter SettAccessControl.setStrategist(address)._strategist (contracts/lib/SettAccessControl.sol#31) is not in mixedCase
Parameter SettAccessControl.setKeeper(address)._keeper (contracts/lib/SettAccessControl.sol#38) is not in mixedCase
Parameter SettAccessControl.setGovernance(address)._governance (contracts/lib/SettAccessControl.sol#45) is not in mixedCase
Variable SettAccessControl.__gap (contracts/lib/SettAccessControl.sol#50) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#56-59) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#61-65) is not in
mixedCase
Variable ERC20Upgradeable.__gap (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#312) is not in mixedCase
Function ContextUpgradeable.__Context_init() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#33-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/PausableUpgradeable.sol#96) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#39-41) is not in
mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#43-45) is
not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#67) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (../../../../../../../../.brownie
/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ReentrancyGuardUpgradeable.__gap (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/utils/ReentrancyGuardUpgradeable.sol#67) is never used in Vault
(contracts/Vault.sol#60-683)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

initialize(address,address,address,address,address,address,address,string,string,uint256[4]) should be declared external:
    - Vault.initialize(address,address,address,address,address,address,address,string,string,uint256[4]) (contracts/Vault.sol#140-211)
getPricePerFullShare() should be declared external:
    - Vault.getPricePerFullShare() (contracts/Vault.sol#230-235)
setGovernance(address) should be declared external:
    - SettAccessControl.setGovernance(address) (contracts/lib/SettAccessControl.sol#45-48)
name() should be declared external:
    - ERC20Upgradeable.name() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#70-72)
symbol() should be declared external:
    - ERC20Upgradeable.symbol() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#78-80)
decimals() should be declared external:
    - ERC20Upgradeable.decimals() (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#95-97)
transfer(address,uint256) should be declared external:
    - ERC20Upgradeable.transfer(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#121-124)
allowance(address,address) should be declared external:
    - ERC20Upgradeable.allowance(address,address) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#129-131)
approve(address,uint256) should be declared external:
    - ERC20Upgradeable.approve(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#140-143)
transferFrom(address,address,uint256) should be declared external:
    - ERC20Upgradeable.transferFrom(address,address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#158-162)
increaseAllowance(address,uint256) should be declared external:
    - ERC20Upgradeable.increaseAllowance(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#176-179)
decreaseAllowance(address,uint256) should be declared external:
    - ERC20Upgradeable.decreaseAllowance(address,uint256) (../../../../../../../../.brownie/packages/OpenZeppelin/openzeppelin-contracts-upgradeable@3.4.0/contracts/token/ERC20/ERC20Upgradeable.sol#195-198)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
./contracts/Vault.sol analyzed (15 contracts with 75 detectors), 121 result(s) found
```

Adherence to Best Practices

1. Function `Vault.version()` and `BaseStrategy.baseStrategyVersion()` may be further restricted from being `view` functions to `pure` functions, as they do not even read any state variables. **Update** Those functions have been changed to `pure`, as suggested.

2. For improved readability it is recommended to have a maximum line length of 79 or 99. Therefore the following lines, which exceed these limits, should be adjusted accordingly:

    . `contracts/Vault.sol`: L22, L35, L39, L47, L51, L52, L56, L61, L73, L98, L104, L120, L140, L164, L170, L222, L258, L266, L292, L331, L396, L398, L405, L407, L422, L436, L439, L440, L458, L541, L586, L589 and L590.

    . `contracts/BaseStrategy.sol`: L25, L26, L29, L37, L82, L97, L117, L123, L128, L162, L163, L187, L196, L197, L206, L210, L211, L226, L231, L240, L253, L280, L282, L283, L301, L311, L314, L315, L336 and L338.

3. To improve readability and hence auditability, it is recommended to use descriptive variable names. Function `Vault._withdraw()` declares and uses variables `r` and `b` which are neither descriptive nor do they come with descriptive comments. Therefore it should be considered to rename and/or comment them.

4. Remove virtual function declarations when the class is not supposed to be used as a base class (see `Vault.sol`). **Update** The unnecessary `virtual` keywords have been removed, as recommended.

5. Use interface types directly instead of casting them on each use (`vault`, `want`, `strategy`).

6. Use compiler built-ins instead of defining your own magic values (`ONE_ETH = 1 ether`).

7. Consider caching calls to `balance()` in `Vault`, since it's a fairly expensive external call to the strategy and should stay constant most of the time.

Test Results

Test Suite Results

All 69 tests are passing.

```
tests/test_additional_token.py::test_report_an_extra_token PASSED
tests/test_additional_token.py::test_emit_additional_token_from_vault RUNNING
Transaction sent: 0xf3f12728b2a91b4380b7b6159a51e75ab318a19032798768f627a7939966bad3
tests/test_additional_token.py::test_emit_additional_token_from_vault PASSED
tests/test_additional_token.py::test_withdraw_another_token_from_strat RUNNING
Transaction sent: 0xa23f8949960ab213df1e9b280f2cad2fab381a07fe64ff1fdcbf5e009bce592a
tests/test_additional_token.py::test_withdraw_another_token_from_strat PASSED
tests/test_additional_token.py::test_withdraw_another_token_from_vault PASSED
tests/test_additional_token.py::test_security_try_rugging_want RUNNING
Transaction sent: 0xb03e58c550e5df964849d9e8eb8e89262525c7da43b0cf941a30d3d6e0083969
tests/test_additional_token.py::test_security_try_rugging_want PASSED
tests/test_additional_token.py::test_security_try_rugging_protected_token RUNNING
Transaction sent: 0xdd32ed5de320ffc7fbeda19bf4e771b82a36bccdbe76be433a270a3fc9e146de
tests/test_additional_token.py::test_security_try_rugging_protected_token PASSED
tests/test_fees_math.py::test_withdrawal_fees_are_issued_as_shares PASSED
tests/test_fees_math.py::test_performance_fees_are_issued_as_shares RUNNING
9.84251968503937e+16 98425196850393700 1
9.842519685039371e+16 98425196850393700 1
9.84251968503937e+16 98425196850393700 1
tests/test_fees_math.py::test_performance_fees_are_issued_as_shares PASSED
tests/test_fees_math.py::test_performance_fees_are_issued_to_treasury_and_strategist RUNNING
9.84251968503937e+16 98425196865742964 1
9.842519686574296e+16 98425196865742964 1
9.842520466316877e+16 98425204663168778 1
tests/test_fees_math.py::test_performance_fees_are_issued_to_treasury_and_strategist PASSED
tests/test_fees_math.py::test_zero_fee PASSED
tests/test_guestlist.py::test_add_guestlist Guestlist was deployed at:  0x5EFF382151814b1929fa960F35C333FE885C58a5
RUNNING
Transaction sent: 0xedbf6604648a84f41334eda7597d33818076fe91967b2a227d40b70edf8d7e66
tests/test_guestlist.py::test_add_guestlist PASSED
```

```
tests/test_guestlist.py::test_add_remove_guestlist Guestlist was deployed at:  0x5EFF382151814b1929fa960F35C333FE885C58a5
RUNNING
Transaction sent: 0xedbf6604648a84f41334eda7597d33818076fe91967b2a227d40b70edf8d7e66
tests/test_guestlist.py::test_add_remove_guestlist PASSED
tests/test_guestlist.py::test_deposit_guestlist Guestlist was deployed at:  0x5EFF382151814b1929fa960F35C333FE885C58a5
PASSED
tests/test_guestlist.py::test_userlimit_guestlist Guestlist was deployed at:  0x5EFF382151814b1929fa960F35C333FE885C58a5
RUNNING
Transaction sent: 0x760105f34e2de0b3d25d01efc24437ccb2378d343fe6093a266b411deabce68e
tests/test_guestlist.py::test_userlimit_guestlist PASSED
tests/test_guestlist.py::test_wantlimit_guestlist Guestlist was deployed at:  0x5EFF382151814b1929fa960F35C333FE885C58a5
RUNNING
Transaction sent: 0x29fcb92c11c560393cc807d40c69a19d533d1413a0c2716fc7999edcfcb28aa5
tests/test_guestlist.py::test_wantlimit_guestlist PASSED
tests/test_shares_math.py::test_deposit_no_initial_shares PASSED
tests/test_shares_math.py::test_deposit_some_initial_shares PASSED
tests/test_shares_math.py::test_deposit_earn_harvest RUNNING
4901960784313725490 4.901960784313726e+18 100
tests/test_shares_math.py::test_deposit_earn_harvest PASSED
tests/test_shares_math.py::test_withdrawalAll PASSED
tests/test_shares_math.py::test_withdrawalSome_more_than_deposited RUNNING
Transaction sent: 0x6fe7524663f12c5d0c386aceaaee36d22697937a7befde33a479cd86953611e5
tests/test_shares_math.py::test_withdrawalSome_more_than_deposited PASSED
tests/test_shares_math.py::test_withdrawSome PASSED
tests/test_shares_math.py::test_withdrawalAll_after_harvest PASSED
tests/test_shares_math.py::test_multiple_withdrawals PASSED
tests/functional/strategy/test_config.py::test_setWithdrawalMaxDeviationThreshold RUNNING
Transaction sent: 0xd0891f9521021906ffb16ea36d311d13cc55bc53c5ddb56f38e145f99fc97c56
Transaction sent: 0x41c0566b5036d9a76a7c8c3dd6f54ae7d3d4c14c4e9de79c5c93277af0e0b076
tests/functional/strategy/test_config.py::test_setWithdrawalMaxDeviationThreshold PASSED
tests/functional/strategy/test_config.py::test_isProtectedToken PASSED
tests/functional/strategy/test_deployment.py::test_strategy_deployment RUNNING
Transaction sent: 0xb1834a9d64f400a82ce32afa0811d1b566bb3c7b495939c5c60c413f3d0bbaf7
tests/functional/strategy/test_deployment.py::test_strategy_deployment PASSED
tests/functional/strategy/test_deposit.py::test_empty_deposit PASSED
tests/functional/vault/test_config.py::test_setTreasury RUNNING
Transaction sent: 0x4a2955ec1a041a49b12e0fc938c1b1b4845aa4498ad8b4418b0e50c6ef021b80
Transaction sent: 0xd1c1c0c8c64bb55e838c7004fc344fb2452a9d904d2af9e5eb3c18164c7529c3
tests/functional/vault/test_config.py::test_setTreasury PASSED
tests/functional/vault/test_config.py::test_setGuestList Guestlist was deployed at:  0xccc21b0987cD8351013d84bBFCC0cE89DbFeCd45
RUNNING
Transaction sent: 0x9aa621239e594b32e107f598d5e90f0e520782ea4e85e3304921224aa6e3524b
tests/functional/vault/test_config.py::test_setGuestList PASSED
tests/functional/vault/test_config.py::test_setGuardian RUNNING
Transaction sent: 0x4d2cc0d8100b5ecc096d28e56719bb13989232333ab5100fd445f760582ae95a
Transaction sent: 0xcfb096eabf26d785e482e985a63b3386c2291b4e19a0fd5c941c58c6f1e14825
tests/functional/vault/test_config.py::test_setGuardian PASSED
tests/functional/vault/test_config.py::test_setMin RUNNING
Transaction sent: 0x2db7b0eb815e4b4f53ff273cc41e641a472985b3c7f739583793d3b93f7e3050
Transaction sent: 0xa81270004a921f391025a9e563e71121cc8db944b425c1f4be1ac1a8e83b7e31
tests/functional/vault/test_config.py::test_setMin PASSED
tests/functional/vault/test_config.py::test_setMaxPerformanceFee RUNNING
Transaction sent: 0x360bdc74302f10009ae8b596e62109fc3d42a8d8581e2401bbf149fdda7fc0ae
Transaction sent: 0x7312362485a5b4e740c5f341e3e9baf2743826a8645ee5b36f3e8724e2d086a0
Transaction sent: 0x114c82d1fb810fcb6612d06cf57a2019d3ad2b11f493f29f3e30f20bb48f6fba
tests/functional/vault/test_config.py::test_setMaxPerformanceFee PASSED
tests/functional/vault/test_config.py::test_setMaxWithdrawalFee RUNNING
Transaction sent: 0x1b8765507d2c9d52af1a41c68f16462298ca12176ba3704fc1ca06482c3197e7
Transaction sent: 0x26bf5a8cd424c5b3c42f20599dc37b38f88ac4414fc4fa0a1f0f0587e7d0e3d5
Transaction sent: 0x5f72233730a67f1789a38a481db128273ef7cf820d54894131142b0402c67d74
tests/functional/vault/test_config.py::test_setMaxWithdrawalFee PASSED
tests/functional/vault/test_config.py::test_setMaxManagementFee RUNNING
Transaction sent: 0x8e897135bad0a031833ab3fe7c185b0b44e7670310dda4d11b38e9c36e6bfa09
Transaction sent: 0xd40ee475a72f893203c6483bccca4da3b51ab75fd62c7a84ce622062f8f9fab8
Transaction sent: 0xf07a89191ef137d6d7e1ab78c26fcc55d39b4a8111a4f3e9912ce453196f3909
tests/functional/vault/test_config.py::test_setMaxManagementFee PASSED
tests/functional/vault/test_config.py::test_setManagementFee RUNNING
Transaction sent: 0x3a261689a0e1b8d0a52b6eacedb20703408d7d296d314a33f3a75b6f479a1479
Transaction sent: 0xc351ae53c33a8a491fdeb812a5849da6a64f9d004a30e4971a7980d2f97d6742
tests/functional/vault/test_config.py::test_setManagementFee PASSED
tests/functional/vault/test_config.py::test_setWithdrawalFee RUNNING
Transaction sent: 0x435bc43fce4ae06375f23146a1d0dd454cf590d05ee52c7e10a32a857162c2d1
Transaction sent: 0x90ae92bdfa83eb4e420c1338421ce4ce4aa8aac626ee6dbe4c923633648a149a
tests/functional/vault/test_config.py::test_setWithdrawalFee PASSED
tests/functional/vault/test_config.py::test_setPerformanceFeeStrategist RUNNING
Transaction sent: 0x3606049f649f0ee5998dbd239c8372bb72fd87138d1a9f800d87c91d28242bae
Transaction sent: 0x459c684c33424e1b3dec1290bcf34d430c22d9dcc2acce27c33c9072594c8b11
tests/functional/vault/test_config.py::test_setPerformanceFeeStrategist PASSED
tests/functional/vault/test_config.py::test_setPerformanceFeeGovernance RUNNING
Transaction sent: 0x77ff4c12321f835282e9ce515cd9778a577d33823be8f377403a3b06ae750f95
Transaction sent: 0x505ec82f6ee70d372850dcf9b114a86af812c90899210b4b24b4e8558f5f4fb8
tests/functional/vault/test_config.py::test_setPerformanceFeeGovernance PASSED
tests/functional/vault/test_config.py::test_config_pause_unpause RUNNING
Transaction sent: 0x48c10b27b3b25519b1b9c7d2c2e4bc1cea70930d4120b35ba8f303d474cb7367
Transaction sent: 0x06ff5d64c00c73eb1f5d1b68009deb069e45f22e36c5916e44f1e0832b6ccd99
Transaction sent: 0xb93314df011799f42bfd0f777cf07db0255acb7f3a71a99f278700632c5692b5
Transaction sent: 0x3363b71f115cf153c08eba81b924b2599005d97573e1351ec5731695339d08d2
Transaction sent: 0x559ce35bd8169d078702dec24b036f13c65af76b7a14621ec87e67e09a0034b3
Transaction sent: 0x780c148cdf785168bf839df760ddb1e0a318fe3c76d333b9cb5af7a5e9850be6
Transaction sent: 0x161c07e79c8d58595d48a55e4a7e2484fac3b3c0cc0208de6637c50bc4d4f96b
Transaction sent: 0xd5613b816110a5df3466a8e124bb1d447a315d883495e282bd13357dfe560d39
tests/functional/vault/test_config.py::test_config_pause_unpause PASSED
tests/functional/vault/test_deployment.py::test_vault_deployment PASSED
tests/functional/vault/test_deployment.py::test_vault_deployment_badArgument RUNNING
Transaction sent: 0xd5cb3f4e137805a2c7ccf582168f9a4e05a1adcd9d8a3b1ac7a9afc7b3752f28
Transaction sent: 0x7f02949a72f01c5139c36be44057e99650ffaec2d0a4d55ca655f87a081dda5b
Transaction sent: 0x8d318fd5baa95f2961541b7bf83a5fa6b63ebae4135aecde0aa7b98a3154d4ef
Transaction sent: 0x07906ca7a9932686f5285e233282e0f5e0c1aa39957466d2671791b7bf7e1b0b
Transaction sent: 0xc700ed821373d4ba38768003bc7911889d96f7b5437229eeb8e98ed548b924c9
Transaction sent: 0xf48486bc2e73335a55e29bebae2c424fc327fe2a0bf98514a48b80468e486471
Transaction sent: 0x85f3735ac8f6b79acbcc3998e116235a2f7532e85b253ce1769ea59cd9a88d03
tests/functional/vault/test_deployment.py::test_vault_deployment_badArgument PASSED
tests/functional/vault/test_deposit.py::test_deposit RUNNING
Transaction sent: 0x4f9d1311dfd335650d8b1d343354e4e3ad2b18b88d528b95c014fda06a445832
Transaction sent: 0xc7168615498daadb9f2f377e609ce4e56b3459088d3b74b5cfd372deddf7df99
Transaction sent: 0x82ba097dcfeeade8212e2c59c4748c1b5966f41cedfc582edfa1deb5b5a1f7de
tests/functional/vault/test_deposit.py::test_deposit PASSED
tests/functional/vault/test_deposit.py::test_depositFor RUNNING
Transaction sent: 0x2817d86c4fe494d31a77e9bf31fb9b12dc33eebb46c5deb5b74b3f652fff2bc9
Transaction sent: 0xfcfe94cfa9e6b9580101bdb2e06fce8541e922a02c7cc5baf15af4f90c6f3877
tests/functional/vault/test_deposit.py::test_depositFor PASSED
tests/functional/vault/test_deposit.py::test_depositAll RUNNING
Transaction sent: 0x113c9264913e2c7842689738bc99279a215f54dc4f95ffd190c7dd1d7ef8877a
tests/functional/vault/test_deposit.py::test_depositAll PASSED
tests/functional/vault/test_deposit.py::test_nonreentrant RUNNING
Transaction sent: 0x882cc974a29002e20c2676dffd440a18d0434e3643717c8c181b9fb9284da397
tests/functional/vault/test_deposit.py::test_nonreentrant PASSED
tests/functional/vault/test_earn.py::test_earn RUNNING
Transaction sent: 0xc4d96b38182fb4a51aa84a9a1f1490e7800ac225de3185b2a0f17bc94f51591f
Transaction sent: 0xd09a0aa60df2074f76c2e03629fbf6d98a3c528d720d823fbeb147a334aaec4a
Transaction sent: 0xb3d40fcb10e7b1aca994292423b1a4184ac3ac2e08e5cbee78d239d19e7c6d68
tests/functional/vault/test_earn.py::test_earn PASSED
tests/functional/vault/test_name.py::test_with_default_name PASSED
tests/functional/vault/test_name.py::test_with_custom_name PASSED
tests/functional/vault/test_name.py::test_with_custom_name_default_symbol PASSED
tests/functional/vault/test_name.py::test_version PASSED
tests/functional/vault/test_report.py::test_report_failed RUNNING
Transaction sent: 0xfa3125b1ca3761d0fa1534993df2a251a84b943c22494907c0990003c87c9dfa
Transaction sent: 0x3fe2c4acb51d0a1fc49ee14ada95725190e006beabbdd1b0bfe767076d8812fc
Transaction sent: 0xe79936e1f0c700c3da764e4c997b55ad6d5518d15563a5ffd7b9d4221158064d
tests/functional/vault/test_report.py::test_report_failed PASSED
tests/functional/vault/test_report.py::test_harvest_no_balance PASSED
tests/functional/vault/test_report.py::test_report_additional_token_failed RUNNING
Transaction sent: 0x5ff905ca96cb013c88116df6e03ac19f5be0053fd51940c72bd7e0fba2d0f7de
Transaction sent: 0xa88e199f129210e5bac677e8fc7c22290c7b3a4f4912b815a6d4fbee459fda87
Transaction sent: 0x590932f31cbf128539e16091a44f0bbc1dbf5530910bf091eb8fd0f1e6aed884
Transaction sent: 0x798c15cf0748887d5cd0a831c97886230d00788d8dbe6532c37d07413b979133
Transaction sent: 0xf1252c730972e1d8bd1b688ee331ed8834ccedb7825b2bbc8a755175817bc493
Transaction sent: 0x4f4e9f9e4611e3df5dc63e4fb09f2568e4a2e95325e5f5062e6a1b5b7a68f47f
Transaction sent: 0xcca092414edce256bd1733e473a1540ff2ae783876185d6861237a60a48b829c
tests/functional/vault/test_report.py::test_report_additional_token_failed PASSED
tests/functional/vault/test_withdraw.py::test_withdrawToVault RUNNING
Transaction sent: 0xea600bb4bf35bf93722ff9bf72ea2829437ef1a838d95dda8bc1defaf683a360
tests/functional/vault/test_withdraw.py::test_withdrawToVault PASSED
tests/functional/vault/test_withdraw.py::test_withdraw RUNNING
Transaction sent: 0x9f916ab202b3c457408af3b4d95aec95eadd988256cbf6cf9bc0a4c4c724dfcd
Transaction sent: 0x24b1bd3391bb38d9cf8b87de9d3f1288e32513873a814e320d02cbe5e36c7688
tests/functional/vault/test_withdraw.py::test_withdraw PASSED
tests/functional/vault/test_withdraw.py::test_withdrawAll RUNNING
Transaction sent: 0xee51a0b37f2fb16a06151010b47707b8fead1ac54e2c3305719f120f01a494dc
```

```
tests/functional/vault/test_withdraw.py::test_withdrawAll PASSED
tests/functional/vault/test_withdraw.py::test_withdrawOther RUNNING
Transaction sent: 0x9797f5b2b8ebef38161a623181e3239ab45dcc789af7e91d3602a961fae4eb4c
tests/functional/vault/test_withdraw.py::test_withdrawOther PASSED
tests/functional/vault/test_withdraw.py::test_withdraw_lossy RUNNING
Transaction sent: 0x20870c2f9d59a375588801e83f01a9f0232fe06cb526fc125bb2454d62952593
tests/functional/vault/test_withdraw.py::test_withdraw_lossy PASSED
tests/integration/test_harvest_flow.py::test_deposit_withdraw_single_user_flow RUNNING
init_resolver DemoStrategy
Transaction sent: 0x50da5edb6a927fe593901329960f86a99f28286f1e963a4b15ab67a747679823
snap
snap
=== Compare Earn ===
=== Compare: StrategySnapshot Sett 13857666 -> 13857667 ===
+-----------------------+----------+---------+--------+
| metric                |   before |   after |   diff |
+=======================+==========+=========+========+
| balances.want.sett    |       80 |     4   |  -76   |
+-----------------------+----------+---------+--------+
| balances.want.strategy |        0 |    76   |   76   |
+-----------------------+----------+---------+--------+
| sett.available        |       76 |     3.8 |  -72.2 |
+-----------------------+----------+---------+--------+
| strategy.balanceOfWant |        0 |    76   |   76   |
+-----------------------+----------+---------+--------+
| strategy.balanceOf    |        0 |    76   |   76   |
+-----------------------+----------+---------+--------+
snap
snap
=== Compare Withdraw ===
=== Compare: StrategySnapshot Sett 13857668 -> 13857669 ===
+-------------------------+----------+---------+--------+
| metric                  |   before |   after |   diff |
+=========================+==========+=========+========+
| balances.want.sett      |      4   |    0.2  |  -3.8  |
+-------------------------+----------+---------+--------+
| balances.want.strategy  |     76   |   40    | -36    |
+-------------------------+----------+---------+--------+
| balances.want.user      |     20   |   59.8  |  39.8  |
+-------------------------+----------+---------+--------+
| balances.sett.governance |      0   |    0.2  |   0.2  |
+-------------------------+----------+---------+--------+
| balances.sett.treasury  |      0   |    0.2  |   0.2  |
+-------------------------+----------+---------+--------+
| balances.sett.user      |     80   |   40    | -40    |
+-------------------------+----------+---------+--------+
| sett.balance            |     80   |   40.2  | -39.8  |
+-------------------------+----------+---------+--------+
| sett.available          |      3.8 |    0.19 |  -3.61 |
+-------------------------+----------+---------+--------+
| sett.totalSupply        |     80   |   40.2  | -39.8  |
+-------------------------+----------+---------+--------+
| strategy.balanceOfWant  |     76   |   40    | -36    |
+-------------------------+----------+---------+--------+
| strategy.balanceOf      |     76   |   40    | -36    |
+-------------------------+----------+---------+--------+
20000000000000000 2e+17 1
snap
snap
=== Compare Withdraw ===
=== Compare: StrategySnapshot Sett 13857670 -> 13857671 ===
+-------------------------+----------+---------+---------+
| metric                  |   before |   after |    diff |
+=========================+==========+=========+=========+
| balances.want.sett      |     0.2  |   0.2   |  -1e-18 |
+-------------------------+----------+---------+---------+
| balances.want.strategy  |    40    |   0.2   | -39.8   |
+-------------------------+----------+---------+---------+
| balances.want.user      |    59.8  |  99.6   |  39.8   |
+-------------------------+----------+---------+---------+
| balances.sett.governance |     0.2  |   0.4   |   0.2   |
+-------------------------+----------+---------+---------+
| balances.sett.treasury  |     0.2  |   0.4   |   0.2   |
+-------------------------+----------+---------+---------+
| balances.sett.user      |    40    |   1e-18 | -40     |
+-------------------------+----------+---------+---------+
| sett.balance            |    40.2  |   0.4   | -39.8   |
+-------------------------+----------+---------+---------+
| sett.available          |     0.19 |   0.19  |  -1e-18 |
+-------------------------+----------+---------+---------+
| sett.totalSupply        |    40.2  |   0.4   | -39.8   |
+-------------------------+----------+---------+---------+
| strategy.balanceOfWant  |    40    |   0.2   | -39.8   |
+-------------------------+----------+---------+---------+
| strategy.balanceOf      |    40    |   0.2   | -39.8   |
+-------------------------+----------+---------+---------+
399999999999999999 4e+17 1
tests/integration/test_harvest_flow.py::test_deposit_withdraw_single_user_flow PASSED
tests/integration/test_harvest_flow.py::test_single_user_harvest_flow RUNNING
init_resolver DemoStrategy
snap
snap
=== Compare Deposit ===
=== Compare: StrategySnapshot Sett 13857664 -> 13857665 ===
+--------------------+----------+---------+--------+
| metric             |   before |   after |   diff |
+====================+==========+=========+========+
| balances.want.sett |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| balances.want.user |      100 |    50   |  -50   |
+--------------------+----------+---------+--------+
| balances.sett.user |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.balance       |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.available     |        0 |    47.5 |   47.5 |
+--------------------+----------+---------+--------+
| sett.totalSupply   |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
50000000000000000000 50000000000000000000 1
50000000000000000000 50000000000000000000 1
50000000000000000000 50000000000000000000 1
50000000000000000000 50000000000000000000 1
want.balanceOf(vault) 50000000000000000000
snap
snap
=== Compare Earn ===
=== Compare: StrategySnapshot Sett 13857665 -> 13857666 ===
+-----------------------+----------+---------+----------+
| metric                |   before |   after |     diff |
+=======================+==========+=========+==========+
| balances.want.sett    |     50   |   2.5   | -47.5    |
+-----------------------+----------+---------+----------+
| balances.want.strategy |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
| sett.available        |     47.5 |   2.375 | -45.125  |
+-----------------------+----------+---------+----------+
| strategy.balanceOfWant |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
| strategy.balanceOf    |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
Transaction sent: 0x0e53ada74bc063a64746d8dfe596f3d070a053312c0e8dc4e32239a5e1de885a
snap
snap
snap
snap
Transaction sent: 0xfe3f0e72d828a3dc50cc858690c426fdd2007a77fee577f47211b6dec10163e4
snap
snap
snap
snap
snap
snap
=== Compare Withdraw ===
=== Compare: StrategySnapshot Sett 13857675 -> 13857676 ===
+--------------------------+----------+----------+-----------+
| metric                   |   before |    after |      diff |
+==========================+==========+==========+===========+
| balances.want.sett       |    2.5   |  0.125   |  -2.375   |
+--------------------------+----------+----------+-----------+
```

```
| balances.want.strategy   |   47.5   | 25       | -22.5     |
+--------------------------+----------+----------+-----------+
| balances.want.user       |   50     | 74.875   |  24.875   |
+--------------------------+----------+----------+-----------+
| balances.sett.governance |    0     |  0.125   |   0.125   |
+--------------------------+----------+----------+-----------+
| balances.sett.treasury   |    0     |  0.125   |   0.125   |
+--------------------------+----------+----------+-----------+
| balances.sett.user       |   50     | 25       | -25       |
+--------------------------+----------+----------+-----------+
| sett.balance             |   50     | 25.125   | -24.875   |
+--------------------------+----------+----------+-----------+
| sett.available           |    2.375 |  0.11875 |  -2.25625 |
+--------------------------+----------+----------+-----------+
| sett.totalSupply         |   50     | 25.125   | -24.875   |
+--------------------------+----------+----------+-----------+
| strategy.balanceOfWant   |   47.5   | 25       | -22.5     |
+--------------------------+----------+----------+-----------+
| strategy.balanceOf       |   47.5   | 25       | -22.5     |
+--------------------------+----------+----------+-----------+
125000000000000000 1.25e+17 1
snap
snap
snap
snap
=== Compare Withdraw ===
=== Compare: StrategySnapshot Sett 13857678 -> 13857679 ===
+--------------------------+----------+----------+---------+
| metric                   |   before |    after |    diff |
+==========================+==========+==========+=========+
| balances.want.sett       |  0.125   |  0.125   |  -1e-18 |
+--------------------------+----------+----------+---------+
| balances.want.strategy   | 25       |  0.125   | -24.875 |
+--------------------------+----------+----------+---------+
| balances.want.user       | 74.875   | 99.75    |  24.875 |
+--------------------------+----------+----------+---------+
| balances.sett.governance |  0.125   |  0.25    |   0.125 |
+--------------------------+----------+----------+---------+
| balances.sett.treasury   |  0.125   |  0.25    |   0.125 |
+--------------------------+----------+----------+---------+
| balances.sett.user       | 25       |  1e-18   | -25     |
+--------------------------+----------+----------+---------+
| sett.balance             | 25.125   |  0.25    | -24.875 |
+--------------------------+----------+----------+---------+
| sett.available           |  0.11875 |  0.11875 |  -1e-18 |
+--------------------------+----------+----------+---------+
| sett.totalSupply         | 25.125   |  0.25    | -24.875 |
+--------------------------+----------+----------+---------+
| strategy.balanceOfWant   | 25       |  0.125   | -24.875 |
+--------------------------+----------+----------+---------+
| strategy.balanceOf       | 25       |  0.125   | -24.875 |
+--------------------------+----------+----------+---------+
249999999999999999 2.5e+17 1
tests/integration/test_harvest_flow.py::test_single_user_harvest_flow PASSED
tests/integration/test_harvest_flow.py::test_migrate_single_user RUNNING
init_resolver DemoStrategy
snap
snap
=== Compare Deposit ===
=== Compare: StrategySnapshot Sett 13857664 -> 13857665 ===
+--------------------+----------+---------+--------+
| metric             |   before |   after |   diff |
+====================+==========+=========+========+
| balances.want.sett |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| balances.want.user |      100 |    50   |  -50   |
+--------------------+----------+---------+--------+
| balances.sett.user |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.balance       |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.available     |        0 |    47.5 |   47.5 |
+--------------------+----------+---------+--------+
| sett.totalSupply   |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
Transaction sent: 0x1e337f179965a7267b88c20efdeeb9b3c57c2caa0638ecf84543115ed7447574
Transaction sent: 0x1e337f179965a7267b88c20efdeeb9b3c57c2caa0638ecf84543115ed7447574
Transaction sent: 0x1e337f179965a7267b88c20efdeeb9b3c57c2caa0638ecf84543115ed7447574
tests/integration/test_harvest_flow.py::test_migrate_single_user PASSED
tests/integration/test_harvest_flow.py::test_withdraw_other RUNNING
Transaction sent: 0x9593832473c58f3473c7e38ac5eae143c5df39a2db53ef644011364eac4eca4a
Transaction sent: 0x87fa338cc9810974afc217356c335b4be5413299d5da1bddfdac8b7609b9027b
Transaction sent: 0xa184c5d10afffc1829d7472e0632f4193967334f367cb0369c477893087ea729
tests/integration/test_harvest_flow.py::test_withdraw_other PASSED
tests/integration/test_harvest_flow.py::test_single_user_harvest_flow_remove_fees RUNNING
init_resolver DemoStrategy
snap
snap
=== Compare Deposit ===
=== Compare: StrategySnapshot Sett 13857675 -> 13857676 ===
+--------------------+----------+---------+--------+
| metric             |   before |   after |   diff |
+====================+==========+=========+========+
| balances.want.sett |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| balances.want.user |      100 |    50   |  -50   |
+--------------------+----------+---------+--------+
| balances.sett.user |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.balance       |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
| sett.available     |        0 |    47.5 |   47.5 |
+--------------------+----------+---------+--------+
| sett.totalSupply   |        0 |    50   |   50   |
+--------------------+----------+---------+--------+
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
5000000000000000000 5000000000000000000 1
snap
snap
=== Compare Earn ===
=== Compare: StrategySnapshot Sett 13857676 -> 13857677 ===
+-----------------------+----------+---------+----------+
| metric                |   before |   after |     diff |
+=======================+==========+=========+==========+
| balances.want.sett    |     50   |   2.5   | -47.5    |
+-----------------------+----------+---------+----------+
| balances.want.strategy |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
| sett.available        |     47.5 |   2.375 | -45.125  |
+-----------------------+----------+---------+----------+
| strategy.balanceOfWant |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
| strategy.balanceOf    |      0   |  47.5   |  47.5    |
+-----------------------+----------+---------+----------+
snap
snap
Transaction sent: 0x7b42ed32b5be579caee06a33da37d6fc036d7c70bb23b9038754e9470e1f1c99
snap
snap
snap
snap
snap
snap
snap
snap
=== Compare Withdraw ===
=== Compare: StrategySnapshot Sett 13857686 -> 13857687 ===
+--------------------------+----------+---------+----------+
| metric                   |   before |   after |     diff |
+==========================+==========+=========+==========+
| balances.want.sett       |    2.5   |  0.25   |  -2.25   |
+--------------------------+----------+---------+----------+
| balances.want.strategy   |   47.5   |  0      | -47.5    |
+--------------------------+----------+---------+----------+
```

```
| balances.want.user       |   50     | 99.75   |  49.75   |
+--------------------------+----------+---------+----------+
| balances.sett.governance |    0     |  0.25   |   0.25   |
+--------------------------+----------+---------+----------+
| balances.sett.treasury   |    0     |  0.25   |   0.25   |
+--------------------------+----------+---------+----------+
| balances.sett.user       |   50     |  0      | -50      |
+--------------------------+----------+---------+----------+
| sett.balance             |   50     |  0.25   | -49.75   |
+--------------------------+----------+---------+----------+
| sett.available           |    2.375 |  0.2375 |  -2.1375 |
+--------------------------+----------+---------+----------+
| sett.totalSupply         |   50     |  0.25   | -49.75   |
+--------------------------+----------+---------+----------+
| strategy.balanceOfWant   |   47.5   |  0      | -47.5    |
+--------------------------+----------+---------+----------+
| strategy.balanceOf       |   47.5   |  0      | -47.5    |
+--------------------------+----------+---------+----------+
250000000000000000 2.5e+17 1
Report after 4 days
Gains
-250000000000000000
gainsPercentage
-0.0025
tests/integration/test_harvest_flow.py::test_single_user_harvest_flow_remove_fees PASSED
tests/integration/test_strategy_permissions.py::test_strategy_action_permissions RUNNING
Transaction sent: 0xfb3250dfaae96602e241e4ea1b9b4dd3ec447385f206477c5bc8a32eb3d9ae56
Transaction sent: 0x5568fc98b99e57f1e8e8d4434977c979f82664db9978c0272da8b67ac447b221
Transaction sent: 0xaa130a371c1abe30b0b15f04b09e16bdcee8a6445b0ca8f543ee69ea253e9c87
Transaction sent: 0x6d9f18f26c2787a54833205931c4ce7f548390960e9544e80cc1dc0799d02579
Transaction sent: 0xa457aba90679f36f9941820038b6d0c27737606382804a70fe9acc813415e538
Transaction sent: 0x28fc92330ce4575f6a2200691b75c7af19c738458c87b8351bb3740920a796ff
Transaction sent: 0x22a7dbf4abbadd0efd363d9015c3f065df87c9572e476980137b584c77e6ab44
Transaction sent: 0x7edd232058e478dfd643a0bddc3a6ef0b49fa11d586aa944658f47d874e066bb
Transaction sent: 0x3c77bb2b5fe24b0c8f8d0b98da950a46f128f424b28d8916f2b086ec03972cec
Transaction sent: 0xf672955f700c4ede14c362f3f6d5ef75bd9ec5085c82c1568b6865eed9f16ed6
Transaction sent: 0xd32e039fb8252161376dd8c5eb6c9b4da9c5a673b2f7dc1c691fe1e839e9109b
tests/integration/test_strategy_permissions.py::test_strategy_action_permissions PASSED
tests/integration/test_strategy_permissions.py::test_strategy_pausing_permissions RUNNING
Transaction sent: 0xd38737c2fd6ed54e624b992e1c59b79cce8f3c8d229b660f0effeed2a6fec903
Transaction sent: 0x455403b95150c548478f04a0069f2b1851533ef95a92be8d09cd4e6a8109fa38
Transaction sent: 0x038b46402a101917832279c2d7682f38833b142ff0d34cc1cfdb14fb52d081ee
Transaction sent: 0xd156b086bc8a019e34bff79a7ef2546f7691e011598a4edb0b474075ee2e190b
tests/integration/test_strategy_permissions.py::test_strategy_pausing_permissions PASSED
tests/integration/test_strategy_permissions.py::test_sett_pausing_permissions RUNNING
Transaction sent: 0x31baafd55977eebc1d3406c9abb6bef2074c9e51b6d14b732a09b838fd585320
Transaction sent: 0xc8bb8e3c666c94f78b7cd225010075de47c43febdf3b0b78a448f0f97ad4d146
Transaction sent: 0x60b3f52a989f8fcc35049d00a7029135a84a861794566182ba647781f1afb85a
Transaction sent: 0xd805e8f2cef478423f26d909e31189547edea1eacaadb1e0938c37d16d338910
Transaction sent: 0x803d6a7416983ae9b30c8b26c5b931db890d69381402e6bfc4d1f666b30bea07
Transaction sent: 0x7075f42db87c177344748f3a592b0f12737c0c49ad369c95645c8dc95bd7c410
tests/integration/test_strategy_permissions.py::test_sett_pausing_permissions PASSED
tests/integration/test_strategy_permissions.py::test_sett_config_permissions RUNNING
Transaction sent: 0x9a0dfd2bb528ce1be8a9f475e11b2ae4149ee8691da556e176eb2d454be69002
Transaction sent: 0xb7aa135c0dab8859d74330b1d794638cec1e5f55d6393170ad865652cdb92b0f
Transaction sent: 0x5a72c45f24615ca235f0b6881171d3a31822f879d531f18798a773f58889895a
Transaction sent: 0x93572bfed4da9c2f382703544f70e34e1b24c79121823bf453892dca57735b1c
Transaction sent: 0x501096904339a408e297afa2873a7c60d4cf0c977286df374ba11b8594b16643
Transaction sent: 0x2253d84dcec55e2ba793c29a30e7002200b1ea07aa700175b1527d9654cfd364
Transaction sent: 0x19324b8a30c5bd5647960e8502aba0617b59663e0510c7c050902465fc91599f
tests/integration/test_strategy_permissions.py::test_sett_config_permissions PASSED
tests/integration/test_strategy_permissions.py::test_sett_earn_permissions RUNNING
Transaction sent: 0x8878012b472cc5e87cad7c8c4a9c6aa77811ce256b97dcf3e53928830fcdc74a
tests/integration/test_strategy_permissions.py::test_sett_earn_permissions PASSED
tests/integration/test_vault_token.py::test_vault_token PASSED
```

Code Coverage

```
================================== Coverage ==================================

  contract: DemoStrategy - 84.3%
    BaseStrategy.__BaseStrategy_init - 100.0%
    BaseStrategy._onlyAuthorizedActors - 100.0%
    BaseStrategy._onlyAuthorizedActorsOrVault - 100.0%
    BaseStrategy._onlyAuthorizedPausers - 100.0%
    BaseStrategy._onlyGovernance - 100.0%
    BaseStrategy._onlyNotProtectedTokens - 100.0%
    BaseStrategy._onlyVault - 100.0%
    BaseStrategy._processExtraToken - 100.0%
    BaseStrategy._transferToVault - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.isProtectedToken - 100.0%
    BaseStrategy.setWithdrawalMaxDeviationThreshold - 100.0%
    DemoStrategy._withdrawSome - 100.0%
    MathUpgradeable.min - 100.0%
    BaseStrategy.withdraw - 91.7%
    AddressUpgradeable.functionCallWithValue - 75.0%
    BaseStrategy._diff - 75.0%
    SafeERC20Upgradeable._callOptionalReturn - 75.0%
    SafeMathUpgradeable.add - 75.0%
    SafeMathUpgradeable.div - 75.0%
    SafeMathUpgradeable.sub - 75.0%
    PausableUpgradeable._unpause - 50.0%
    SafeMathUpgradeable.mul - 50.0%
    AddressUpgradeable._verifyCallResult - 37.5%

  contract: MaliciousToken - 5.4%
    MaliciousToken.transferFrom - 50.0%
    ERC20Upgradeable._approve - 0.0%
    ERC20Upgradeable._burn - 0.0%
    ERC20Upgradeable._mint - 0.0%
    ERC20Upgradeable._transfer - 0.0%
    MaliciousToken.initialize - 0.0%

  contract: MockToken - 69.2%
    ERC20Upgradeable._approve - 75.0%
    ERC20Upgradeable._burn - 75.0%
    ERC20Upgradeable._mint - 75.0%
    ERC20Upgradeable._transfer - 75.0%
    MockToken.initialize - 75.0%
    SafeMathUpgradeable.add - 75.0%
    SafeMathUpgradeable.sub - 75.0%

  contract: TestVipCappedGuestListBbtcUpgradeable - 47.9%
    TestVipCappedGuestListBbtcUpgradeable.authorized - 100.0%
    SafeMathUpgradeable.sub - 75.0%
    OwnableUpgradeable.transferOwnership - 62.5%
    TestVipCappedGuestListBbtcUpgradeable._setGuests - 62.5%
    TestVipCappedGuestListBbtcUpgradeable.proveInvitation - 0.0%

  contract: Vault - 90.2%
    SettAccessControl._onlyAuthorizedActors - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    SettAccessControl._onlyGovernanceOrStrategist - 100.0%
    Vault._calculateFee - 100.0%
    Vault._depositForWithAuthorization - 100.0%
    Vault._handleFees - 100.0%
    Vault._onlyAuthorizedPausers - 100.0%
    Vault._onlyStrategy - 100.0%
    Vault._withdraw - 100.0%
    Vault.earn - 100.0%
    Vault.reportAdditionalToken - 100.0%
    Vault.reportHarvest - 100.0%
    Vault.setGuardian - 100.0%
    Vault.setManagementFee - 100.0%
    Vault.setMaxManagementFee - 100.0%
    Vault.setMaxPerformanceFee - 100.0%
    Vault.setMaxWithdrawalFee - 100.0%
    Vault.setPerformanceFeeGovernance - 100.0%
    Vault.setPerformanceFeeStrategist - 100.0%
    Vault.setStrategy - 100.0%
    Vault.setToEarnBps - 100.0%
    Vault.setTreasury - 100.0%
    Vault.setWithdrawalFee - 100.0%
    Vault.sweepExtraToken - 100.0%
    Vault.initialize - 92.3%
    SafeMathUpgradeable.mul - 87.5%
    SafeMathUpgradeable.sub - 87.5%
    AddressUpgradeable.functionCallWithValue - 75.0%
    ERC20Upgradeable._approve - 75.0%
    ERC20Upgradeable._burn - 75.0%
    ERC20Upgradeable._mint - 75.0%
    ERC20Upgradeable._transfer - 75.0%
    SafeERC20Upgradeable._callOptionalReturn - 75.0%
    SafeMathUpgradeable.add - 75.0%
    SafeMathUpgradeable.div - 75.0%
    AddressUpgradeable._verifyCallResult - 62.5%
    PausableUpgradeable._unpause - 50.0%
    Vault._depositFor - 50.0%

View the report using the Brownie GUI
================= 69 passed, 110 warnings in 359.07s (0:05:59) =================
```

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
ee118a845dd1bf42f8b8e9ebf79f22bb6e48ce34d183114cd2153a6d95d6e34e ./contracts/Vault.sol

9303892b1d399b09d713f9aa46c21b750aff976e825dd186030945b584415c1f ./contracts/BaseStrategy.sol

9e7dc38cedd98135c5dd448b1a5fe315fdee0c5d235c8a89c217a0bd902f4e38 ./contracts/lib/SettAccessControl.sol

597e6227d3c6e8adecda81ee92e7039a181a5d3b2eb699b8288a0b7208b10a5c ./contracts/proxy/AdminUpgradeabilityProxy.sol

db4f31874d3bea383e90182737fb214d51adce16315519f9050e61f9d7e2f3dd ./contracts/test/MockToken.sol

720c86acc2f1b9334e0283346264c3f2636d5ba95b84cf8cd940f1dc2900feab ./contracts/test/TestVipCappedGuestListBbtcUpgradeable.sol

d1f7ea64ffdd799574e2f0ae674297645741b3d30d2d6fe37357acdb23b1d805 ./contracts/test/MaliciousToken.sol

e2d2f4b1676c4f10e0ffb68fc28f4c4ab19556fe4542b97248d79c60768079f7 ./contracts/test/DemoStrategy.sol
```

Changelog

- 2021-12-10 - Initial report
- 2021-12-23 - Final report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.