Badger

SMART CONTRACT AUDIT

ZOKYO.

Dec 11, 2020 | v. 3.0

PASS

Zokyo's Security Team has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

SCORE

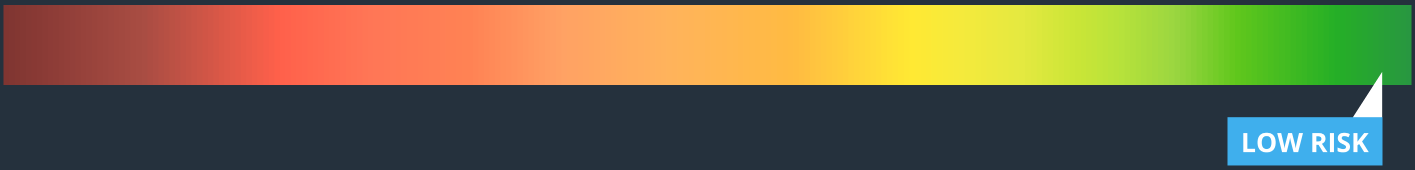
98

TECHNICAL SUMMARY

This document outlines the overall security of the Badger smart contracts, evaluated by Zokyo's Blockchain Security team.

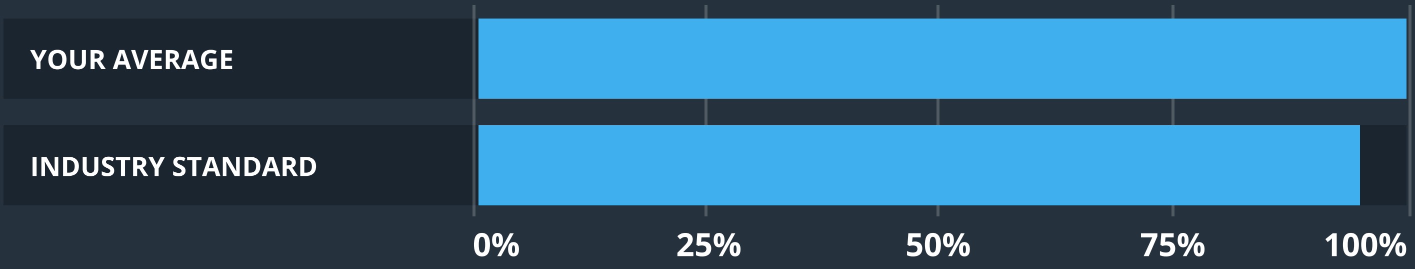
The scope of this audit was to analyze and document the Badger smart contract codebase for quality, security, and correctness.

Contract Status



There were no critical issues found during the audit.

Testable Code



Testable code is 100%, which is above the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that’s able to withstand the Ethereum network’s fast-paced and rapidly changing environment, we at Zokyo recommend that the Badger team put in place a bug bounty program to encourage further and active analysis of the smart contract.

TABLE OF CONTENTS

AUDITING STRATEGY AND TECHNIQUES APPLIED

The Smart contract's source code was taken from:
badger-system repo master branch (last commit – Emerge)

Repo – https://github.com/Badger-Finance/badger-system

Master Branch – https://github.com/Badger-Finance/badger-system/tree/master

Last Commit –
https://github.com/Badger-Finance/badger-system/commit/537ef9c44196893cb9760beca2dcc501952e9a4a

Throughout the review process, care was taken to ensure that the token contract:

- Implements and adheres to existing Token standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of gas, without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the latest vulnerabilities;
- Whether the code meets best practices in code readability, etc.

Zokyo's Security Team has followed best practices and industry-standard techniques to verify the implementation of Badger smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1 Due diligence in assessing the overall code quality of the codebase.

2 Cross-comparison with other, similar smart contracts by industry leaders.

3 Testing contract logic against common and uncommon attack vectors.

4 Thorough, manual review of the codebase, line-by-line.

SUMMARY

There were no critical issues found during the **manual audit**. Contracts are well written and structured. One finding during manual audit has no impact on contract performance or security, so it is fully production-ready.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the ability of the contract to compile or operate in a significant way.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

COMPLETE ANALYSIS

Function instead of modifier

INFORMATIONAL | UNRESOLVED

Set (73-79 lines) and **SettAccessControl** (15-25 lines) smart contracts include internal functions for modifiers roles.

Recommendation:
Use modifiers instead of functions.

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Badger DAO team

```
contract: Controller - 33.3%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    Controller.approveStrategy - 100.0%
    Controller.balanceOf - 100.0%
    Controller.setVault - 75.0%
    Controller.withdraw - 75.0%
    SettAccessControl._onlyGovernance - 75.0%
    Controller.setStrategy - 62.5%
    SettAccessControl._onlyGovernanceOrStrategist - 62.5%
    Controller._onlyApprovedForWant - 56.2%
    AddressUpgradeable._functionCallWithValue - 50.0%
    Controller.earn - 50.0%
    Controller.getExpectedReturn - 0.0%
    Controller.inCaseStrategyTokenGetStuck - 0.0%
    Controller.inCaseTokensGetStuck - 0.0%
    Controller.initialize - 0.0%
    Controller.revokeStrategy - 0.0%
    Controller.setConverter - 0.0%
    Controller.setOneSplit - 0.0%
    Controller.setRewards - 0.0%
    Controller.setSplit - 0.0%
    Controller.withdrawAll - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%

contract: ForceEther - 100.0%
    ForceEther.forceSend - 100.0%

contract: Sett - 52.6%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    Sett._blockLocked - 100.0%
    Sett._deposit - 100.0%
    Sett._lockForBlock - 100.0%
    Sett._withdraw - 100.0%
    Sett.available - 100.0%
    Sett.balance - 100.0%
    Sett.deposit - 100.0%
    Sett.earn - 100.0%
    Sett.setController - 100.0%
    Sett.setMin - 100.0%
    Sett.withdraw - 100.0%
    Sett.withdrawAll - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    SettAccessControl.setKeeper - 100.0%
    SettAccessControl.setStrategist - 100.0%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    SettAccessControlDefended._defend - 75.0%
    AddressUpgradeable._functionCallWithValue - 50.0%
    Sett._onlyController - 0.0%
    Sett.depositAll - 0.0%
    Sett.getPricePerFullShare - 0.0%
    Sett.harvest - 0.0%
    Sett.initialize - 0.0%
    Sett.trackFullPricePerShare - 0.0%
    Sett.transfer - 0.0%
    Sett.transferFrom - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControlDefended.approveContractAccess - 0.0%
    SettAccessControlDefended.revokeContractAccess - 0.0%

  contract: StrategyBadgerLpMetaFarm - 50.4%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy._safeApproveHelper - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
```

```
BaseStrategy.deposit - 100.0%
BaseStrategy.pause - 100.0%
BaseStrategy.setController - 100.0%
BaseStrategy.setGuardian - 100.0%
BaseStrategy.setPerformanceFeeGovernance - 100.0%
BaseStrategy.setPerformanceFeeStrategist - 100.0%
BaseStrategy.setWithdrawalFee - 100.0%
BaseStrategy.unpause - 100.0%
ContextUpgradeable._msgSender - 100.0%
PausableUpgradeable._pause - 100.0%
SettAccessControl._onlyGovernance - 100.0%
StrategyBadgerLpMetaFarm._deposit - 100.0%
StrategyBadgerLpMetaFarm.balanceOfPool - 100.0%
StrategyBadgerLpMetaFarm.getName - 100.0%
BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
BaseStrategy._onlyAuthorizedPausers - 83.3%
SettAccessControl._onlyAuthorizedActors - 83.3%
BaseStrategy._transferToVault - 75.0%
BaseStrategy.withdraw - 75.0%
StrategyBadgerLpMetaFarm._withdrawSome - 75.0%
AddressUpgradeable._functionCallWithValue - 50.0%
BaseStrategy._processWithdrawalFee - 50.0%
PausableUpgradeable._unpause - 50.0%
StrategyBadgerLpMetaFarm.harvest - 46.7%
BaseStrategy.withdrawAll - 33.3%
BaseStrategy.withdrawOther - 25.0%
BaseStrategy.__BaseStrategy_init - 0.0%
BaseStrategy._add_max_liquidity_uniswap - 0.0%
BaseStrategy._swap - 0.0%
PausableUpgradeable.__Pausable_init - 0.0%
PausableUpgradeable.__Pausable_init_unchained - 0.0%
PausableUpgradeable.paused - 0.0%
SettAccessControl.setGovernance - 0.0%
SettAccessControl.setKeeper - 0.0%
SettAccessControl.setStrategist - 0.0%
StrategyBadgerLpMetaFarm._onlyNotProtectedTokens - 0.0%
StrategyBadgerLpMetaFarm._withdrawAll - 0.0%
StrategyBadgerLpMetaFarm.getProtectedTokens - 0.0%
```

```
    StrategyBadgerLpMetaFarm.initialize - 0.0%

contract: StrategyBadgerRewards - 55.1%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy._safeApproveHelper - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.pause - 100.0%
    BaseStrategy.setController - 100.0%
    BaseStrategy.setGuardian - 100.0%
    BaseStrategy.setPerformanceFeeGovernance - 100.0%
    BaseStrategy.setPerformanceFeeStrategist - 100.0%
    BaseStrategy.setWithdrawalFee - 100.0%
    BaseStrategy.unpause - 100.0%
    ContextUpgradeable._msgSender - 100.0%
    PausableUpgradeable._pause - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    StrategyBadgerRewards._deposit - 100.0%
    StrategyBadgerRewards.balanceOfPool - 100.0%
    StrategyBadgerRewards.getName - 100.0%
    BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
    BaseStrategy._onlyAuthorizedPausers - 83.3%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    BaseStrategy._transferToVault - 75.0%
    BaseStrategy.withdraw - 75.0%
    StrategyBadgerRewards.harvest - 67.9%
    StrategyBadgerRewards._withdrawSome - 58.3%
    AddressUpgradeable._functionCallWithValue - 50.0%
    BaseStrategy._processWithdrawalFee - 50.0%
    PausableUpgradeable._unpause - 50.0%
    BaseStrategy.withdrawAll - 33.3%
    BaseStrategy.withdrawOther - 25.0%
    BaseStrategy.__BaseStrategy_init - 0.0%
    PausableUpgradeable.__Pausable_init - 0.0%
    PausableUpgradeable.__Pausable_init_unchained - 0.0%
    PausableUpgradeable.paused - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%
    StrategyBadgerRewards._onlyNotProtectedTokens - 0.0%
    StrategyBadgerRewards._withdrawAll - 0.0%
    StrategyBadgerRewards.getProtectedTokens - 0.0%
    StrategyBadgerRewards.initialize - 0.0%

contract: StrategyCurveGaugeRenBtcCrv - 55.6%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy._safeApproveHelper - 100.0%
    BaseStrategy._swap - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.pause - 100.0%
    BaseStrategy.setController - 100.0%
    BaseStrategy.setGuardian - 100.0%
    BaseStrategy.setPerformanceFeeGovernance - 100.0%
    BaseStrategy.setPerformanceFeeStrategist - 100.0%
    BaseStrategy.setWithdrawalFee - 100.0%
    BaseStrategy.unpause - 100.0%
    ContextUpgradeable._msgSender - 100.0%
    PausableUpgradeable._pause - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    StrategyCurveGaugeBase._deposit - 100.0%
    StrategyCurveGaugeBase._processPerformanceFees - 100.0%
    StrategyCurveGaugeBase._withdrawSome - 100.0%
    StrategyCurveGaugeBase.balanceOfPool - 100.0%
    StrategyCurveGaugeBase.getName - 100.0%
    StrategyCurveGaugeRenBtcCrv._add_liquidity_curve - 100.0%
    BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
    BaseStrategy._onlyAuthorizedPausers - 83.3%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    BaseStrategy._transferToVault - 75.0%
    BaseStrategy.withdraw - 75.0%
    StrategyCurveGaugeBase.harvest - 75.0%
    BaseStrategy._processFee - 58.3%
    AddressUpgradeable._functionCallWithValue - 50.0%
    BaseStrategy._processWithdrawalFee - 50.0%
    PausableUpgradeable._unpause - 50.0%
    BaseStrategy.withdrawAll - 33.3%
    BaseStrategy.withdrawOther - 25.0%
    BaseStrategy.__BaseStrategy_init - 0.0%
    PausableUpgradeable.__Pausable_init - 0.0%
    PausableUpgradeable.__Pausable_init_unchained - 0.0%
    PausableUpgradeable.paused - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%
    StrategyCurveGaugeBase._onlyNotProtectedTokens - 0.0%
    StrategyCurveGaugeBase._withdrawAll - 0.0%
    StrategyCurveGaugeBase.getProtectedTokens - 0.0%
    StrategyCurveGaugeBase.initialize - 0.0%
    StrategyCurveGaugeBase.setKeepCRV - 0.0%

contract: StrategyCurveGaugeSbtcCrv - 55.2%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy._safeApproveHelper - 100.0%
    BaseStrategy._swap - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.pause - 100.0%
    BaseStrategy.setController - 100.0%
    BaseStrategy.setGuardian - 100.0%
    BaseStrategy.setPerformanceFeeGovernance - 100.0%
    BaseStrategy.setPerformanceFeeStrategist - 100.0%
    BaseStrategy.setWithdrawalFee - 100.0%
    BaseStrategy.unpause - 100.0%
    ContextUpgradeable._msgSender - 100.0%
    PausableUpgradeable._pause - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    StrategyCurveGaugeBase._deposit - 100.0%
    StrategyCurveGaugeBase._processPerformanceFees - 100.0%
    StrategyCurveGaugeBase._withdrawSome - 100.0%
    StrategyCurveGaugeBase.balanceOfPool - 100.0%
    StrategyCurveGaugeSbtcCrv._add_liquidity_curve - 100.0%
    BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
    BaseStrategy._onlyAuthorizedPausers - 83.3%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    BaseStrategy._transferToVault - 75.0%
    BaseStrategy.withdraw - 75.0%
    StrategyCurveGaugeBase.harvest - 75.0%
    BaseStrategy._processFee - 58.3%
    AddressUpgradeable._functionCallWithValue - 50.0%
    BaseStrategy._processWithdrawalFee - 50.0%
    PausableUpgradeable._unpause - 50.0%
    BaseStrategy.withdrawAll - 33.3%
    BaseStrategy.withdrawOther - 25.0%
    BaseStrategy.__BaseStrategy_init - 0.0%
    PausableUpgradeable.__Pausable_init - 0.0%
    PausableUpgradeable.__Pausable_init_unchained - 0.0%
    PausableUpgradeable.paused - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%
    StrategyCurveGaugeBase._onlyNotProtectedTokens - 0.0%
    StrategyCurveGaugeBase._withdrawAll - 0.0%
    StrategyCurveGaugeBase.getName - 0.0%
    StrategyCurveGaugeBase.getProtectedTokens - 0.0%
    StrategyCurveGaugeBase.initialize - 0.0%
    StrategyCurveGaugeBase.setKeepCRV - 0.0%

contract: StrategyCurveGaugeTbtcCrv - 55.2%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy._safeApproveHelper - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.pause - 100.0%
    BaseStrategy.setController - 100.0%
    BaseStrategy.setGuardian - 100.0%
    BaseStrategy.setPerformanceFeeGovernance - 100.0%
    BaseStrategy.setPerformanceFeeStrategist - 100.0%
    BaseStrategy.setWithdrawalFee - 100.0%
    BaseStrategy.unpause - 100.0%
    ContextUpgradeable._msgSender - 100.0%
    PausableUpgradeable._pause - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    StrategyCurveGaugeBase._deposit - 100.0%
    StrategyCurveGaugeBase._processPerformanceFees - 100.0%
    StrategyCurveGaugeBase._withdrawSome - 100.0%
    StrategyCurveGaugeBase.balanceOfPool - 100.0%
    StrategyCurveGaugeTbtcCrv._add_liquidity_curve - 100.0%
    BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
    BaseStrategy._onlyAuthorizedPausers - 83.3%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    BaseStrategy._transferToVault - 75.0%
    BaseStrategy.withdraw - 75.0%
    StrategyCurveGaugeBase.harvest - 75.0%
    BaseStrategy._processFee - 58.3%
    AddressUpgradeable._functionCallWithValue - 50.0%
    BaseStrategy._processWithdrawalFee - 50.0%
    PausableUpgradeable._unpause - 50.0%
    BaseStrategy.withdrawAll - 33.3%
    BaseStrategy.withdrawOther - 25.0%
    BaseStrategy.__BaseStrategy_init - 0.0%
    PausableUpgradeable.__Pausable_init - 0.0%
    PausableUpgradeable.__Pausable_init_unchained - 0.0%
    PausableUpgradeable.paused - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%
    StrategyCurveGaugeBase._onlyNotProtectedTokens - 0.0%
```

```
    StrategyCurveGaugeBase._withdrawAll - 0.0%
    StrategyCurveGaugeBase.getName - 0.0%
    StrategyCurveGaugeBase.getProtectedTokens - 0.0%
    StrategyCurveGaugeBase.initialize - 0.0%
    StrategyCurveGaugeBase.setKeepCRV - 0.0

contract: StrategyHarvestMetaFarm - 54.9%
    AddressUpgradeable.functionCall - 100.0%
    AddressUpgradeable.isContract - 100.0%
    BaseStrategy._onlyController - 100.0%
    BaseStrategy.balanceOf - 100.0%
    BaseStrategy.balanceOfWant - 100.0%
    BaseStrategy.deposit - 100.0%
    BaseStrategy.pause - 100.0%
    BaseStrategy.setController - 100.0%
    BaseStrategy.setGuardian - 100.0%
    BaseStrategy.setPerformanceFeeGovernance - 100.0%
    BaseStrategy.setPerformanceFeeStrategist - 100.0%
    BaseStrategy.setWithdrawalFee - 100.0%
    BaseStrategy.unpause - 100.0%
    ContextUpgradeable._msgSender - 100.0%
    MathUpgradeable.min - 100.0%
    PausableUpgradeable._pause - 100.0%
    SettAccessControl._onlyGovernance - 100.0%
    StrategyHarvestMetaFarm._deposit - 100.0%
    StrategyHarvestMetaFarm._fromHarvestVaultTokens - 100.0%
    StrategyHarvestMetaFarm._postDeposit - 100.0%
    StrategyHarvestMetaFarm._processPerformanceFees - 100.0%
    StrategyHarvestMetaFarm._toHarvestVaultTokens - 100.0%
    StrategyHarvestMetaFarm.balanceOfPool - 100.0%
    StrategyHarvestMetaFarm.getName - 100.0%
    StrategyHarvestMetaFarm.isTendable - 100.0%
    StrategyHarvestMetaFarm.setFarmPerformanceFeeGovernance - 100.0%
    StrategyHarvestMetaFarm.setFarmPerformanceFeeStrategist - 100.0%
    StrategyHarvestMetaFarm._withdrawSome - 91.7%
    BaseStrategy._onlyAuthorizedActorsOrController - 87.5%
    BaseStrategy._onlyAuthorizedPausers - 83.3%
    SettAccessControl._onlyAuthorizedActors - 83.3%
    BaseStrategy._transferToVault - 75.0%
    BaseStrategy.withdraw - 75.0%
    StrategyHarvestMetaFarm.harvest - 70.5%
    StrategyHarvestMetaFarm.tend - 67.9%
    BaseStrategy._processFee - 58.3%
    AddressUpgradeable._functionCallWithValue - 50.0%
    BaseStrategy._processWithdrawalFee - 50.0%
    PausableUpgradeable._unpause - 50.0%
    BaseStrategy.withdrawAll - 33.3%
    BaseStrategy.withdrawOther - 25.0%
    BaseStrategy.__BaseStrategy_init - 0.0%
    PausableUpgradeable.__Pausable_init - 0.0%
    PausableUpgradeable.__Pausable_init_unchained - 0.0%
    PausableUpgradeable.paused - 0.0%
    SettAccessControl.setGovernance - 0.0%
    SettAccessControl.setKeeper - 0.0%
    SettAccessControl.setStrategist - 0.0%
    StrategyHarvestMetaFarm._onlyNotProtectedTokens - 0.0%
    StrategyHarvestMetaFarm._withdrawAll - 0.0%
    StrategyHarvestMetaFarm.getProtectedTokens - 0.0%
    StrategyHarvestMetaFarm.initialize - 0.0%

19 passed, 30 skipped, 86 warnings in 3162.15s (0:52:42)
```

We are grateful to have been given the opportunity to work with the Badger DAO team.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

Zokyo's Security Team recommends that the Badger team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.