Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED
SECURITY CERTIFICATE

February 16th 2022 — Quantstamp Verified

Gelt Vault V1

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

Executive Summary

Type	DeFi
Auditors	Marius Guggenmos, Senior Research Engineer Ed Zulkoski, Senior Security Engineer Souhail Mssassi, Research Engineer
Timeline	2022-01-18 through 2022-01-28
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Medium
Test Quality	High
Source Code	

Repository	Commit
gelt-contracts-v1 (initial audit)	b2b87fe
gelt-contracts-v1 (reaudit)	6f9e489

Total Issues	**6** (6 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	**1** (1 Resolved)
Low Risk Issues	**3** (3 Resolved)
Informational Risk Issues	**1** (1 Resolved)
Undetermined Risk Issues	**1** (1 Resolved)

0 Unresolved
0 Acknowledged
6 Resolved

ALL ISSUES ADDRESSED

BEST PRACTICES ADDRESSED

Documentation Issues Addressed

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 initial audit: Quantstamp has performed an audit of the Gelt Vault V1 repository. Overall, the code base is relatively small and makes use of a lot of OpenZeppelin libraries, which were not part of the audit. While the contracts are documented well, we are not aware of any documentation targeting end-users. The project's tests appear fairly extensive, although it is hard to rate the quality without coverage measurements available, which is why we conservatively evaluated the test quality as *medium* instead of *high*. The audit resulted in a total of 6 findings and an additional 1 best practice violations, described below. We confirm that none of the tests are failing when executed on our end. We recommend that all issues reported in this document be addressed.
After reaudit: Quantstamp has checked the commit hash `6f9e489` and has determined that all of the reported issues have been resolved (that is either fixed or mitigated) by the Gelt team. More details regarding each of the issues are provided in the update messages below each issue recommendation. Additionally, we promoted the test quality to *high* after finding a workaround for computing the test coverage.

ID	Description	Severity	Status
QSP-1	Redemption Fee Precision Check May Lead to Revert	^ Medium	Fixed
QSP-2	Unclear Access Control Policy May Lead to Griefing	○ Informational	Mitigated
QSP-3	`minOutputQuantity` During Redeems May Lead to Unfavorable Exchanges	⌄ Low	Fixed
QSP-4	Privileged Roles and Ownership	⌄ Low	Fixed
QSP-5	Missing Input Validation	⌄ Low	Fixed
QSP-6	Unclear `+1` Compensation in `_calcStrategyRedeemAmount`	? Undetermined	Fixed

Quantstamp Audit Breakdown

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

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 v0.8.2

Steps taken to run the tools:

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

Findings

QSP-1 Redemption Fee Precision Check May Lead to Revert

Severity: Medium Risk

Status: Fixed

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

Description: The internal function `_getStrategyRedeemFeeBps` contains the following code:

```
uint256 redemptionFee = mAsset.data().redemptionFee;

require(redemptionFee >= 1e14, "strategy redemption fee must be scaled to 18 decimals");
```

This assumes that the underlying `mAsset` will never have a fee less than 0.0001. For example, the deployed `mAsset` contract here has `redemptionFee = 6e14`. If, for example, `mStable` ever lowered the fee to `6e13`, this would still be "scaled to 18 decimals", but the function `_getStrategyRedeemFeeBps` would revert. This affects critical functions such as `voluntaryExit`.

Recommendation: Allow `redemptionFee` values below `1e14`. Note that if in practice the fees are below `1e14`, certain basis points calculations may return 0 throughout the code.

Update: The percentage calculations now use additional precision to handle lower values. Fixed in this PR.

QSP-2 Unclear Access Control Policy May Lead to Griefing

Severity: Informational

Status: Mitigated

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

Description: The functions `mintWithAuthorization` and `redeemWithAuthorization` are both operator-only functions. This means, all functions intended for end-users can only be performed through meta-transactions where the operator has to pay for the gas. This could lead to griefing scenarios where users repeatedly mint/redeem at the cost of the operator, potentially only paying minimal redemption fees.

Recommendation: Clarify if this access control policy is intended, and whether the above scenario could reasonably occur. In case it is intended, make sure to guard against it in the off-chain components that submit the transactions.

Update: The Gelt team clarified that the access control policy is as intended. Since there is no real issue with the contract itself, we have decided to downgrade this to *informational* severity.

QSP-3 `minOutputQuantity` During Redeems May Lead to Unfavorable Exchanges

Severity: Low Risk

Status: Fixed

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

Description: The function `emergencyExitStrategy` invokes `mAsset.redeem(...)`, using `1` for the `_minOutputQuantity`. If there are issues with the underlying `mAsset` contract, this may cause arbitrarily unfavorable returned quantities of `bAsset` tokens.

Recommendation: Consider using a value that is either proportional to the amount of `mAsset` tokens, or is configurable via some parameter to `emergencyExitStrategy`.

Update: The function `emergencyExitStrategy` now accepts a parameter for the `_minOutputQuantity` call. This has been added in this PR.

QSP-4 Privileged Roles and Ownership

Severity: Low Risk

Status: Fixed

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

Description: Public user-facing documentation should detail the actions that can be performed by privileged users. While we don't believe the administrators have unreasonable levels of control, it is especially important to inform users that administrators are able to halt voluntary exits by pausing the contract and thus locking access to the funds.

Recommendation: Add public documentation that clearly documents all of the actions privileged users can perform.

Update: The project's README file now documents the privileged roles. Added in this PR.

QSP-5 Missing Input Validation

Severity: Low Risk

Status: Fixed

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

Description: The `initialize` function accepts a number of important contract addresses that are permanently assigned to contract members. To avoid costly re-deployments where any of these parameters are accidentally set to `address(0)`, input validation checks should be added.

Recommendation: Verify that any address arguments in the `initialize` function are not equal to `address(0)`.

Update: The initialize function now checks all address parameters for `address(0)`. Added in this PR.

QSP-6 Unclear `+1` Compensation in `_calcStrategyRedeemAmount`

Severity: *Undetermined*

Status: Fixed

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

Description: It is not clear if the `+1` calculation is necessary in the following statement of `_calcStrategyRedeemAmount`:

```
mAssetAmount = mAsset.getRedeemExactBassetsOutput(bAssets, bAssetQuantities) + 1; // Compensate for rounding errors.
```

Note that in the deployed code here of an `mAsset`, the function `computeRedeemExact` already has a `+1` in the line `grossMasset += 1;` (see L2249).

Recommendation: Confirm whether this `+1` is needed.

Update: The Gelt team confirmed that the `+1` is required with the following statement:

> The +1 is necessary to compensate for rounding errors in the strategy since the return value of _calcStrategyRedeemAmount is used as an input to mAsset.redeem which internally uses computeRedeem that lacks the compensation for rounding errors.

Automated Analyses

Slither

Since the project uses user defined value types, which slither currently does not support, we were unable to run slither.

Code Documentation

1. There is no overview in the repository on what the project's goals are and roughly how it is implemented. This makes it hard to get started for people unfamiliar with the *mStable* or *Gelt* project. We recommend adding a quick summary of both projects in the README.
   Update: The requested documentation has been added in this PR.
2. The contracts integrate with those of the *mStable* project but there are no links to where the information about the external contracts are from. This can lead to confusion, as for example *mStable*'s `SaveWrapper` has different parameters in the official documentation than what the interface file specifies. It looks like the *mStable* docs are from an older version, however this is not immediately clear. We recommend adding links to the source of the interface file if the project does not provide a package for them.
   Update: The requested links have been added in this PR.

Adherence to Best Practices

1. Use the hash computation in the constant assignments directly instead of relying on pre-computed hashes. The solidity compiler with version 0.8 or later should be able to generate the same code, especially since optimizations are enabled.
   Update: The requested changes have been implemented in this PR.

Test Results

Test Suite Results

The test suite is fairly extensive with tests split into the four categories functional, integration, scenario and unit. The total number of tests is 119 and all of them are passing.

```
[Functional] Gelt Vault
  Scenario: A user deposits funds to the Vault
     ✓ Given the user has 1000000000 USDC
     ✓ Given the user approves the Vault to access 1000000000 USDC
     ✓ When the operator mints 1000000000 USDC with the user's signed authorisation
     ✓ Then the user should receive 100000000000000000000000 gUSDC
     ✓ When the operator deposits 1000000000 USDC to the strategy
     ✓ Then the total value of the strategy is approximately 999400000 USDC
  Scenario: A user deposits funds to the Vault
     ✓ Given the user has 10000000000 USDC
     ✓ Given the user approves the Vault to access 10000000000 USDC
     ✓ When the operator mints 10000000000 USDC with the user's signed authorisation
     ✓ Then the user should receive 1000000000000000000000000 gUSDC
     ✓ When the operator deposits 10000000000 USDC to the strategy
     ✓ Then the total value of the strategy is approximately 9994000000 USDC
  Scenario: A user redeems from the Vault
     ✓ Given the user minted by depositing 1000000000 USDC to the Vault
     ✓ Given 20000000000000000000000 mUSD total interest has been accumulated by the strategy
     ✓ When the operator withdraws 1001000000 USDC from the strategy
     ✓ When the operator redeems 1001000000 USDC worth of gUSDC with the user's signed authorisation
     ✓ Then the user should receive 1001000000 USDC
  Scenario: A user redeems from the Vault
     ✓ Given the user minted by depositing 10000000000 USDC to the Vault
     ✓ Given 20000000000000000000000 mUSD total interest has been accumulated by the strategy
     ✓ When the operator withdraws 10002000000 USDC from the strategy
     ✓ When the operator redeems 10002000000 USDC worth of gUSDC with the user's signed authorisation
     ✓ Then the user should receive 10002000000 USDC
  Scenario: User voluntarily exits the Vault
     ✓ Given the user minted by depositing 1000000000 USDC to the Vault
     ✓ When the user voluntarily exits the Vault
     ✓ Then the user should receive approximately 999000000 USDC
  Scenario: User voluntarily exits the Vault
     ✓ Given the user minted by depositing 10000000000 USDC to the Vault
     ✓ When the user voluntarily exits the Vault
     ✓ Then the user should receive approximately 9988000000 USDC

[Integration] Gelt Vault <> mStable - Execute strategy
  #executeStrategy
     ✓ should mint, save and stake the given amount of bAssets
     ✓ should unstake and redeem the given amount of bAssets

[Integration] Gelt Vault <> mStable - Exits
  #voluntaryExit
     ✓ should redeem all funds for the calling user (redeem amount <= free vault funds)
     ✓ should redeem all funds for the calling user (redeem amount > free vault funds)
     ✓ should fail to redeem when minimum output quantity is not satisfied
     ✓ should redeem even when the strategy redemption fees are outside of tolerance
     ✓ should fail to redeem to the zero address
  #emergencyExitStrategy
     ✓ should exit all positions and claim rewards from the strategy
     ✓ should not revert when there are no funds in the strategy

[Integration] Gelt Vault <> mStable - Rewards
  #claimRewards
     ✓ should claim both platform and reward tokens
     ✓ should claim no rewards when strategy value = 0
  #collectRewards
```

```
       ✓ should collect rewards to the pre-set reward collector address
       ✓ should not revert when there are no rewards to collect
       ✓ should revert when the reward collector address is unset

  [Scenario] Gelt Vault
     ✓ 3 party deposit
     ✓ 3 party deposits with partial withdrawal
     ✓ 3 party deposit partial withdrawal and redeposit
     ✓ 3 party deposit partial withdrawal and concomitant deposit

  [Unit] Gelt Vault: Access Control
    Role: Owner
       ✓ should grant the owner role on deployment to the deployer
       ✓ should make the owner role the administrator of all the other roles
       ✓ should allow owner to grant and revoke roles
       ✓ should allow owner to transfer ownership
    Role: Administrator
       ✓ should allow administrator to trigger emergency operations
       ✓ should allow administrator to configure the vault
       ✓ should disallow administrator to submit meta-transactions
       ✓ should disallow administrator to interact with the strategy
       ✓ should disallow administrator to upgrade the vault
       ✓ should disallow administrator to transfer ownership
    Role: Operator
       ✓ should allow operator to submit meta-transactions
       ✓ should disallow operator to trigger emergency operations
       ✓ should disallow operator to configure the vault
       ✓ should disallow operator to upgrade the vault
       ✓ should disallow operator to transfer ownership

  [Unit] Gelt Vault: Upgrades
     ✓ should deploy the vault via a proxy
     ✓ should update an already deployed vault
     ✓ should fail to migrate an already migrated vault
     ✓ should fail to upgrade when storage is incompatible

  [Unit] Gelt Vault: Utils
    FixedPointMath
      #add
         ✓ should add two fixed point numbers
         ✓ should revert on overflow
      #sub
         ✓ should subtract two fixed point numbers
         ✓ should revert on underflow
      #mul(UFixed256x18, UFixed256x18)
         ✓ should multiply two fixed point numbers
         ✓ should revert on overflow
      #mul(UFixed256x18, uint256)
         ✓ should multiply a fixed point number by an unsigned integer
         ✓ should revert on overflow
      #div(UFixed256x18, UFixed256x18)
         ✓ should divide two fixed point numbers
         ✓ should revert on overflow
      #div(UFixed256x18, uint256)
         ✓ should divide a fixed point number by an unsigned integer
      #floor
         ✓ should floor a fixed point number
      #toUFixed256x18(uint256)
         ✓ should return a scaled fixed point number
         ✓ should revert on overflow
      #toUFixed256x18(uint256, uint256)
         ✓ should return a fixed point number
         ✓ should revert on overflow
    PercentageMath
      #basisPoints
         ✓ should calculate the correct basis points for the given amount
         ✓ should revert when amount = 0
         ✓ should revert when bps is out of bounds

  [Unit] Gelt Vault
    #initialize
       ✓ should fail if one of the initialize parameters is the zero address
    #mintWithAuthorization
       ✓ should return the initial exchange rate when totalSupply = 0
       ✓ should mint tokens 1:100 when totalSupply == 0
       ✓ should return the correct exchange rate when totalSupply > 0
       ✓ should mint the correct amount of tokens when totalSupply > 0
    #redeemWithAuthorization
       ✓ should redeem the correct amount after initial mint
       ✓ should redeem the correct amount after multiple mints
       ✓ should fail to redeem when there are no tokens minted
       ✓ should fail to redeem when trying to redeem more tokens than minted
    #executeStrategyNetDeposit
       ✓ should revert when amount = 0
    #executeStrategyNetWithdraw
       ✓ should revert when amount = 0
    #emergencyExitStrategy
       ✓ should revert when the minimum output quantity is zero
    #sweep
       ✓ should sweep the given amount of tokens
       ✓ should revert when amount = 0
       ✓ should revert when trying to sweep a token protected by the vault
       ✓ should revert when the balance is less than the amount
    #setStrategyTolerances
       ✓ should set the strategy tolerances
       ✓ should revert when the tolerances are out of bounds
    #setRewardCollector
       ✓ should set the reward collector to the given address
       ✓ should revert when the supplied reward collector is the zero address
    #emergencyPause
       ✓ should pause the vault
       ✓ should revert when trying to pause the already paused vault
       ✓ should prevent calling vault operations while paused
       ✓ should unpause after the pause duration
    #emergencyUnpause
       ✓ should unpause the vault
       ✓ should revert when trying to unpause the already unpaused vault
       ✓ should allow calling vault operations after the vault is unpaused
    #transferOwnership
       ✓ should revert when transferring ownership to the zero address

  [Unit] Mock Vault
    #mint
       ✓ should mint tokens 1:1 when totalSupply == 0
       ✓ should mint the correct amount when totalSupply > 0
       ✓ should mint the correct amount after strategy generates yield
    #withdraw
       ✓ should withdraw the correct amount after initial mint
       ✓ should withdraw the correct amount multiple mints
       ✓ should fail to withdraw when there are no tokens minted
       ✓ should fail to withdraw when trying to withdraw more tokens than minted

·-----------------------------------------------------------------|---------------------------|-------------|-----------------------------·
|                       Solc version: 0.8.9                       ·  Optimizer enabled: true  ·  Runs: 200  ·  Block limit: 30000000 gas  |
··································································|···························|·············|······························
|  Methods                                                                                                                                |
····························|·····································|·············|·············|·············|···············|··············
|  Contract                 ·  Method                             ·  Min        ·  Max        ·  Avg        ·  # calls      ·  usd (avg)  |
····························|·····································|·············|·············|·············|···············|··············
|  ERC20Harness             ·  approve                            ·      46238  ·      46250  ·      46240  ·           22  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  ERC20Harness             ·  increaseAllowance                  ·          -  ·          -  ·      46473  ·            1  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  ERC20Harness             ·  transfer                           ·      34427  ·      51527  ·      50811  ·           24  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  ERC20Upgradeable         ·  approve                            ·      58098  ·      58110  ·      58102  ·           10  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  ERC20Upgradeable         ·  transfer                           ·      46428  ·      63552  ·      60235  ·           31  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  emergencyPause                     ·          -  ·          -  ·      74220  ·            1  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  grantRole                          ·      56349  ·      56361  ·      56355  ·           13  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  mintWithAuthorization              ·          -  ·          -  ·     151797  ·            1  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  revokeRole                         ·      34451  ·      34463  ·      34457  ·            2  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  setCollector                       ·          -  ·          -  ·      55356  ·            1  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  transferOwnership                  ·          -  ·          -  ·      55993  ·            1  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVault         ·  upgradeToAndCall                   ·          -  ·          -  ·      89564  ·            2  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVaultHarness  ·  claimGovernanceTokens              ·     119608  ·     162920  ·     148483  ·            3  ·          -  |
····························|·····································|·············|·············|·············|···············|··············
|  MstableGeltVaultHarness  ·  collectGovernanceTokens            ·      61579  ·     112558  ·      87069  ·            2  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  emergencyExitStrategy               ·      70388  ·     695493  ·     382941  ·            2  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  emergencyPause                      ·          -  ·          -  ·      69328  ·            7  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  emergencyUnpause                    ·          -  ·          -  ·      25198  ·            3  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  executeStrategyNetDeposit           ·     718649  ·     718661  ·     718652  ·           12  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  executeStrategyNetWithdraw          ·     718163  ·     718177  ·     718168  ·            3  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  grantRole                           ·          -  ·          -  ·      51463  ·          110  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  harnessExecuteStrategyNetDeposit    ·      42530  ·      60466  ·      49870  ·           11  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  harnessExecuteStrategyNetWithdraw   ·          -  ·          -  ·      67977  ·            2  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  mintWithAuthorization               ·      99383  ·     164168  ·     144646  ·           32  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  redeemWithAuthorization             ·     104937  ·     226925  ·     145968  ·            7  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  setCollector                        ·          -  ·          -  ·      50461  ·            3  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  setStrategyTolerances               ·      32325  ·      35137  ·      33731  ·            2  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  sweep                               ·          -  ·          -  ·      71813  ·            1  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness  ·  voluntaryExit                       ·     136520  ·     779795  ·     519065  ·            5  ·          -  |
······························|·····································|·············|·············|·············|···············|·············
|  Deployments                                                     ·                                         ·  % of limit   ·             |
····································································|·············|·············|·············|···············|·············
|  ERC20Harness                                                    ·          -  ·          -  ·     815998  ·        2.7 %  ·          -  |
····································································|·············|·············|·············|···············|·············
|  FixedPointMathHarness                                           ·          -  ·          -  ·     229148  ·        0.8 %  ·          -  |
····································································|·············|·············|·············|···············|·············
|  MstableGeltVault                                                ·          -  ·          -  ·    5250779  ·       17.5 %  ·          -  |
····································································|·············|·············|·············|···············|·············
|  MstableGeltVaultHarness                                         ·    5845445  ·    5858781  ·    5853138  ·       19.5 %  ·          -  |
····································································|·············|·············|·············|···············|·············
|  MstableGeltVaultV2                                              ·          -  ·          -  ·    5311844  ·       17.7 %  ·          -  |
····································································|·············|·············|·············|···············|·············
|  PercentageMathHarness                                           ·          -  ·          -  ·     198823  ·        0.7 %  ·          -  |
--------------------------------------------------------------------|-------------|-------------|-------------|---------------|-------------

  119 passing (16m)
```

Code Coverage

There are no instructions on how to compute the test coverage. We briefly tried adding coverage using solidity-coverage but ran into errors.

Update: The Gelt team informed us that their use of user defined value types (UDVT) are the reason for solidity-coverage not working. We temporarily patched the code to use a struct instead of UDVT to collect coverage data. While this is not 100% accurate, it serves well enough to see that the coverage is relatively high.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	96.86	81.25	95.92	96.91	
Authorizable.sol	60	40	75	60	… 57,62,67,68
Migratable.sol	100	100	100	100	
MstableGeltVault.sol	100	85.53	100	100	
TemporarilyPausable.sol	100	87.5	100	100	
contracts/harness/	63.33	100	52	63.33	
ERC20Harness.sol	100	100	100	100	
FixedPointMathHarness.sol	0	100	0	0	… 30,34,38,42
MstableGeltVaultHarness.sol	92.31	100	85.71	92.31	32
MstableGeltVaultV2.sol	100	100	100	100	
MstableGeltVaultV2Incompatible.sol	0	100	0	0	10
PercentageMathHarness.sol	100	100	100	100	
contracts/interface/	100	100	100	100	
IGeltVault.sol	100	100	100	100	
contracts/interface/strategy/mstable/	100	100	100	100	
IInterestBearingMasset.sol	100	100	100	100	
IMasset.sol	100	100	100	100	
ISaveWrapper.sol	100	100	100	100	
IVaultedInterestBearingMasset.sol	100	100	100	100	
contracts/lib/	75	75	69.23	76	
ECRecover.sol	71.43	50	100	71.43	62,66
EIP712.sol	100	100	100	100	
EIP712Domain.sol	100	100	100	100	
FixedPointMath.sol	55.56	100	55.56	55.56	17,23,35,40
PercentageMath.sol	100	100	100	100	
All files	**90.61**	**80.91**	**79.31**	**90.76**	

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

```
22e41ac1a7b64d7d282f9ccc68177bcdf616877ce69ed59201baefa853c6727c ./IGeltVault.sol
c5e5ea3b17b9dc3d999f81b3658eed2d86a11805a9325638aa17965e41816699 ./contracts/Migratable.sol
8e81f9346a5bf2403634d4185e7da6cde25d6d51da706f6309c8ac6e462b6058 ./contracts/Authorizable.sol
81033da8e0337480bd395446b985cd4092e011e91d4f3926940f7b7baf0e3105 ./contracts/MstableGeltVault.sol
a7a24fb29c5ba1c2dc047f0d7e93687b055dee0baa1e4589db9c1fbededf065e ./contracts/TemporarilyPausable.sol
ac05c8ad3572b0ea0915c6a14b1ca5b9ff4b29ff897d1f59cc8263b7f939bfcf ./contracts/lib/ECRecover.sol
2ee5c80c3b55c0f480423811f2cd096990363210b09c6fc68ede967caf7d3a43 ./contracts/lib/EIP712.sol
347d51b8e1286d839c0624e4c4dcea32071b0f2958613f7065851a860c74d20e ./contracts/lib/EIP712Domain.sol
9b7ff415f33067eb1ed2a3b5a0b479cafb5cc7c40ff034027daea025c14581a8 ./contracts/lib/FixedPointMath.sol
06cf19b1c38e3d0087782505afb031ab78a93cc863e16c011190855d69279f68 ./contracts/lib/PercentageMath.sol
```

Tests

```
acb5e5e189c586b0c023b5525263608abc2329e3dcd7c283444a08622e93c3da ./tests/utils/meta-transactions.ts
c483a457931693d60c54bb3425c4e3791e8c6fd8efb338d49343fcd71894e077 ./tests/utils/network.ts
0aa341a935e98b6b7e744dd4955a3c5a3fc791b6261934f04b1961c08d02a25a ./tests/utils/amount.ts
13a59b938f242f953a450a7cab0290cc5006bd09eb397826285d9a44b16d459f ./tests/utils/fixtures.ts
236c98b99f7345c457b3456d933497729049ccb84196bc7463c89a9f646c3217 ./tests/utils/eip712/ecdsa-signature.ts
740875ddb19a6f5cfe0b98de5adda451d14db1ec03df8f779be59d8d2925f022 ./tests/utils/eip712/eip712-type.ts
381bd8d0ed0b491f6d909189aaf4437807d0deda69be107eb20a5be03771fc31 ./tests/utils/eip712/eip712.ts
44c42cf33f81bcef3cba9b31e1da0f3f4c239f8b08e6b8074ff64c3aeafe4f95 ./tests/utils/eip712/field-type.ts
c08a7666391ca4b7d632a203ad636dcc81cfc72a708b50b47595b85dc2c9a283 ./tests/utils/eip712/index.ts
29574a046feb2606308772f1e1adfb219e4f16b49064b7a66180c635908f5443 ./tests/utils/eip712/eip712-types/index.ts
66eea3d9e0b3814c30e006d97749d71b308838a9bb75f988d1c0ee64928b1acf ./tests/utils/eip712/eip712-types/mint-with-authorization.ts
3f84e59dade3a192cd0ccafe616372e6f834fadf1ac68cb919f2a97614feeab6 ./tests/utils/eip712/eip712-types/redeem-with-authorization.ts
421e253955bb1654d289c58ac112b4c55cdf3df51feb0ea7c2beb57ee97d2aab ./tests/unit/gelt-vault-upgrades.spec.ts
76f4ae5e141c86559ba89361e804addcc8d2ecf1c7cc5b1e73a19f1fb581b9b2 ./tests/unit/mock-vault.spec.ts
dc5483824189ed686b5ead0e6cada9334323285d6d7f343b08c6ddde6efc20ab ./tests/unit/gelt-vault-access-control.spec.ts
c150b4d3bc9da80e533e30feb78a92af6c06e54bcbfe23a5b09d8c60202c9122 ./tests/unit/gelt-vault-utils.spec.ts
21dafbe641b2dda79e986ccdf4093b704c15d6e6110168702de2e5cb1473a6d2 ./tests/unit/gelt-vault.spec.ts
75a84c5d97b1446362f5b17f789022b0d1700ddc31b46af38bea12217b4ccf15 ./tests/unit/mocks/mock-vault.ts
f8b843c13fcb8f6033e874584877f52491a22c948f2b145067ecd442c2bc9269 ./tests/scenario/gelt-vault-scenarios.spec.ts
d23b99c635986730e2e3f24f705a58c3f65b99e7f5f74ceb75cecbf65c254ef0 ./tests/integration/gelt-vault-execute-strategy.spec.ts
90a32dd3112bbce7a9d0727fc3e961a8004f479fc0c3c9485c67f5d02f329892 ./tests/integration/integration-test-context.ts
1ea20e50aa451ffad895706bbd63ded397041c70ae83132fb6303cc7f750737a ./tests/integration/gelt-vault-exit.spec.ts
8b920e8df701d6201f9b07485f4b956c12804f761b5b58bb1a405064c59d988f ./tests/integration/gelt-vault-rewards.spec.ts
d907a67a70ce54229018659c9e1cb56d8657636c8fdce5f732db2c6cc18b1128 ./tests/functional/functional-test-context.ts
31ffa865c5281cb9da99985346962a218fbe573312ad7c6a000ce306a1c1122b ./tests/functional/gelt-vault.spec.ts
```

Changelog

- 2022-01-28 - Initial report
- 2022-02-10 - Reaudit 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.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.