



September 15th 2020 — Quantstamp Verified

Frame00 - dev protocol

This security assessment was prepared by Quantstamp, the leader in blockchain security

Executive Summary

Type	Protocol for distribution, trading, and governance.				
Auditors	Jan Gorzny, Blockchain Researcher Poming Lee, Research Engineer Shunsuke Tokoshima, Software Engineer				
Timeline	2020-08-03 through 2020-08-24				
EVM	Muir Glacier				
Languages	Solidity				
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review				
Specification	<a href="#">Whitepaper</a>				
Documentation Quality	<div><div></div></div> Low				
Test Quality	<div><div></div></div> Medium				
Source Code	<table><tr><td>Repository</td><td>Commit</td></tr><tr><td><a href="#">protocol</a></td><td><a href="#">b53007b</a></td></tr></table>	Repository	Commit	<a href="#">protocol</a>	<a href="#">b53007b</a>
Repository	Commit				
<a href="#">protocol</a>	<a href="#">b53007b</a>				

Goals	<ul style="list-style-type: none"><li>Look for standard issues relating to Solidity projects.</li><li>Review the repository at large, including tests.</li></ul>
-------	--

Total Issues	10 (6 Resolved)
High Risk Issues	1 (1 Resolved)
Medium Risk Issues	2 (1 Resolved)
Low Risk Issues	3 (2 Resolved)
Informational Risk Issues	1 (1 Resolved)
Undetermined Risk Issues	3 (1 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

Frame00 uses a large-but-well-structured code base to achieve its goals. During the audit, we identified ten issues, most of which are of low severity, though there are some whose impacts are unknown. The code could benefit from additional testing (though the actual code coverage may be much higher than the table in this report implies), and stricter adherence to best practices. Quantstamp strongly recommends that the tests run before deploying this code.

ID	Description	Severity	Status
QSP-1	Unchecked Return Value	⬆️ High	Fixed
QSP-2	Input Validation Missing	⬆️ Medium	Acknowledged
QSP-3	Possible Open Access to <code>IMarketBehavior.authenticate</code>	⬆️ Medium	Fixed
QSP-4	Incorrect Length Validation	⬇️ Low	Fixed
QSP-5	Gas Usage / <code>for</code> Loop Concerns	⬇️ Low	Fixed
QSP-6	Trivial Input Validation Missing	⬇️ Low	Acknowledged
QSP-7	Unlocked Pragma	🔵 Informational	Fixed
QSP-8	Incorrect Functions	❓ Undetermined	Acknowledged
QSP-9	Race Conditions / Front-Running	❓ Undetermined	Acknowledged
QSP-10	Incorrect/Missing Visibility	❓ 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](#) v0.6.6
- [Muthril](#) v0.2.7

Steps taken to run the tools:

1. Installed the Slither tool: `pip install slither-analyzer`
2. Run Slither from the project directory: `slither .s`
3. Installed the Mythril tool from Pypi: `pip3 install mythril`
4. Ran the Mythril tool on each contract: `myth -x path/to/contract`

## Findings

### QSP-1 Unchecked Return Value

**Severity:** *High Risk*

**Status:** Fixed

**File(s) affected:** `Property.sol`

**Description:** Most functions will return a `true` or `false` value upon success. Some functions, like `send()`, are more crucial to check than others. It's important to ensure that every necessary function is checked.

In this case, the result of a `transfer` on `L146` does not have its returned value checked.

**Recommendation:** Put the specified `transfer` in a `require` statement, or check balances manually before and after the transfer.

### QSP-2 Input Validation Missing

**Severity:** *Medium Risk*

**Status:** Acknowledged

**File(s) affected:** `Dev.sol`

**Description:** Destination addresses of `deposit` and `depositFrom` in `Dev.sol` are not validated.

**Recommendation:** Insert validation as below at the beginning of the functions:

```
addressValidator().validateGroup(_to, config().propertyGroup());
```

**Update:** The team pointed out that the input is validated by the "lookup" of `Lockup.sol`. Gas fees are reduced by reducing duplicate validations, and there is no change.

### QSP-3 Possible Open Access to `IMarketBehavior.authenticate`

**Severity:** *Medium Risk*

**Status:** Fixed

**Description:** End-users (or developers of Dapps) will need to prepare a contract that inherits `IMarketBehavior` and the code of the contract can be written by them arbitrarily as far as we can tell.

Chances are that some users will not set any validation at the beginning of an `authenticate` function in such a contract.

Example contracts `MarketTest1`, `MarketTest2`, and `MarketTest3` do not implement such validations.

In such a case, attackers are able to call `IMarketBehavior.authenticate` directly and associate random properties to random markets whose `MarketBehavior` contracts are not protected. Such exploitation will burn property owners' DEV without their permissions.

**Recommendation:** In the `Market Factory` section of the white paper, strongly encourage readers to validate the caller of `IMarketBehavior.authenticate`.

### QSP-4 Incorrect Length Validation

**Severity:** *Low Risk*

**Status:** Fixed

**File(s) affected:** `VoteCounterStorage.sol`

**Description:** Depending on the encoding, the checks in `validatePropertyName` and `validatePropertySymbol` may not be accurate; see <https://ethereum.stackexchange.com/questions/13862/is-it-possible-to-check-string-variables-length-inside-the-contract> for details.

### QSP-5 Gas Usage / `for` Loop Concerns

**Severity:** *Low Risk*

**Status:** Fixed

**File(s) affected:** `PolicyFactory.sol`, `DIP7.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.

There are loops in the following functions: `DIP7.rewards` and `PolicyFactory.convergePolicy`.

### QSP-6 Trivial Input Validation Missing

**Severity:** *Low Risk*

**Status:** Acknowledged

**File(s) affected:** `various`



**Description:** Many functions accept the trivial `0x0` (and possibly other) address as input. This may lead to incorrect contract initialisation, lost funds, or simply wasted transactions.

- `Allocator.sol`: constructor, `beforeBalanceChange`.
- `AddressConfig.sol`: constructor, `setWithdraw`, `setWithdrawStorage`, `setMarketGroup`, `setPropertyFactory`, `setPropertyGroup`, `setMetricsFactory`, `setMetricsGroup`, `setPolicyGroup`, `setPolicyFactory`, `setPolicySet`, `setPolicy`, `setToken`, `setLockup`, `setLockupStorage`, `setVoteTimes`, `setVoteTimesStorage`, `setVoteCounter`, `setVoteCounterStorage`.
- `UsingConfig.sol`: constructor.
- `UsingStorage.sol`: `setStorage`, `changeOwner`.
- `Dev.sol`: constructor.
- `DevMigration.sol`: constructor.
- `EternalStorage.sol`: `changeOwner`.
- `Lockup.sol`: `lockup`, `cancel`, `withdraw`, `updateCumulativeLockedUp`, `updateStatesAtLockup`, `getLastCumulativeLockedUpAndBlock`, `difference`, `_calculateInterestAmount`, `_calculateWithdrawableInterestAmount`, `calculateWithdrawableInterestAmount`, `withdrawInterest`, `updateValues`, `addValue`, `possible`, `_property`.
- `LockupStorage.sol`: `setStorageValue`, `setStorageInterestPrice`, `setStorageLastInterestPrice`, `setStorageCumulativeGlobalRewards`, `setStorageLastCumulativePropertyInterest`, `setStorageCumulativeLockedUpUnitAndBlock`, `setStorageCumulativeLockedUpValue`, `setStoragePendingInterestWithdrawal`, `setStorageLastCumulativeLockedUpAndBlock`.
- `Market.sol`: constructor, `authenticatedCallback`.
- `MarketFactory.sol`: constructor, `create`.
- `MarketGroup.sol`: `addGroup`.
- `Metrics.sol`: constructor.
- `MetricsFactory.sol`: `create`.
- `MetricsGroup.sol`: constructor.
- `DIP1.sol`: constructor.
- `DIP3.sol`: constructor.
- `DIP7.sol`: constructor.
- `PolicyFactory.sol`: constructor, `create`.
- `PolicyGroup.sol`: constructor.
- `PolicySet.sol`: constructor, `voting`, `setVotingEndBlockNumber`, `addSet`.
- `TheFirstPolicy.sol`: constructor.
- `Property.sol`: constructor.
- `PropertyFactory.sol`: constructor, `create`.
- `PropertyGroup.sol`: constructor.
- `VoteCounter.sol`: constructor, `voteMarket`, `votePolicy`.
- `VoteCounterStorage.sol`: `setStorageAlreadyVoteMarket`, `setStorageAlreadyUseProperty`, `setStorageAlreadyVotePolicy`, `setStoragePolicyVoteCount`, `setStorageOppositeCount`
- `Withdraw.sol`: constructor, `withdraw`, `beforeBalanceChange`, `difference`.
- `WithdrawStorage.sol`: constructor, `setRewardsAmount`, `setWithdrawalLimitTotal`, `setWithdrawalLimitBalance`, `setLastWithdrawalPrice`, `setPendingWithdrawal`, `setLastCumulativeGlobalHoldersPrice`.

And various functions for files in `test/`, specifically, `UsingConfig.sol`, `LockupStorageTest.sol`, `MarketTest1.sol`, `MarketTest2.sol`, `MarketTest3.sol`, and `VoteCounterStorageTest.sol`, where such behaviour is likely not as important, but is good hygiene and may allow the test files to more closely mimic their main-net counterparts.

**Recommendation:** Add checks that important addresses are not `0x0`.  
**Update:** The team has stated that many contracts accept `0x0` addresses as legal values or make a nonsensical transaction that doesn't work. They believe that the increase in gas fees due to the addition of validation will exacerbate the UX of the majority of correct transactions.

## QSP-7 Unlocked Pragma

**Severity:** *Informational*

**Status:** Fixed

**File(s) affected:** `Allocator.sol`, `IAllocator.sol`, `AddressConfig.sol`, `UsingConfig.sol`, `IGroup.sol`, `Decimals.sol`, `Killable.sol`, `EternalStorage.sol`, `UsingStorage.sol`, `AddressValidator.sol`, `UsingValidator.sol`, `Dev.sol`, `DevMigration.sol`, `ILockup.sol`, `Lockup.sol`, `LockupStorage.sol`, `IMarket.sol`, `IMarketBehavior.sol`, `IMarketFactory.sol`, `IMarketGroup.sol`, `Market.sol`, `MarketFactory.sol`, `MarketGroup.sol`, `IMetricsFactory.sol`, `IMetricsGroup.sol`, `Metrics.sol`, `MetricsFactory.sol`, `MetricsGroup.sol`, `DIP1.sol`, `DIP3.sol`, `DIP7.sol`, `IPolicy.sol`, `IPolicyFactory.sol`, `IPolicyGroup.sol`, `IPolicySet.sol`, `PolicyFactory.sol`, `PolicyGroup.sol`, `PolicySet.sol`, `TheFirstPolicy.sol`, `IProperty.sol`, `IPropertyFactory.sol`, `Property.sol`, `PropertyFactory.sol`, `PropertyGroup.sol`, `IVoteCounter.sol`, `VoteCounter.sol`, `VoteCounterStorage.sol`, `IWithdraw.sol`, `Withdraw.sol`, `WithdrawStorage.sol`, `test/*.sol`

**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." 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.

**Recommendation:** Lock the pragma in the source code.  
**Update:** The pragma has been locked in all the files.

## QSP-8 Incorrect Functions

**Severity:** *Undetermined*

**Status:** Acknowledged

**File(s) affected:** `AddressValidator.sol`

**Description:** The function `validateAddresses` is incorrect. Passing in the addresses (x,x,y) will cause the function to return without error, even if x != y; similarly, `validate3Addresses` is incorrect. Passing in the addresses (x,x,y,z) will cause the function to return without error, even if x != y. Similarly, passing in the addresses (x,x,x,z) will cause the function to return without error, even if x != z.

**Recommendation:** Fix this behaviour or clarify these functions.  
**Update:** These functions are meant to function as "or" functions rather than "and"; so this is intentional.

## QSP-9 Race Conditions / Front-Running

**Severity:** *Undetermined*

**Status:** Acknowledged  
**File(s) affected:** `VoteCounter.sol`

**Description:** A block is an ordered collection of transactions from all around the network. It's possible for the ordering of these transactions to manipulate the end result of a block. A miner attacker can take advantage of this by generating and moving transactions in a way that benefits themselves.  
The minimum requirement of the `_up_votes` in function `marketApproval` is  $10 * 10^{18}$ . This might be too small if one wants to front run executing this line `market.toEnable()`; in function `voteMarket` in `VoteCounter.sol`. Similarly, the minimum requirement of the `_up_votes` in function `policyApproval` is  $10 * 10^{18}$ . This might be too small if one wants to front run executing this line `policyFactory.convergePolicy(_policy)` in function `votePolicy` in `VoteCounter.sol`

**Recommendation:** We have no recommendation at this time.  
**Update:** From the team: "This is not a problem. User voting updates this value. Since no one currently develops Policy and Market contracts, this small value is set for quick updates."

## QSP-10 Incorrect/Missing Visibility

**Severity:** *Undetermined*

**Status:** Fixed  
**File(s) affected:** `WithdrawStorage.sol`

**Description:** The visibility of a function or field changes determines how it can be accessed by others. Using the right visibility ensures optimal gas costs and reduces possibilities of attacks. The four types of visibility are: \* `external` - can be called by any other contract (but not within the contract itself) \* Useful for optimizing gas costs \* `public` - can be called anywhere \* `internal` - can only be called within contract, and inherited contracts will inherit this functionality \* Useful for creating functions which should not be called by anyone else \* `private` - can only be called within contract, cannot be inherited  
The function `setCumulativePrice` is “only used in testing” - but it is `public`. This requirement cannot be enforced on-chain as is. There are other public functions which may not need to be `public`.

**Recommendation:** Fix the function visibility as appropriate.  
**Update:** The team has stated that they will remove the function, but it has not yet been removed.

## Automated Analyses

### Mythril

Mythril did not finish.

### Adherence to Specification

The provided specification is lightweight and leaves much to be inferred from the code. The code appears to follow the specification.

## Code Documentation

In-code documentation could be improved - in particular, pre- and post-conditions could be clarified. Many functions are accompanied by one-sentence descriptions of their intended behaviour.

## Adherence to Best Practices

- Various functions, including `MarketTest2.authenticate`, `MarketTest1.authenticate`, `IMarketBehavior.authenticate`, `MarketTest3.authenticate`, `PolicyTestForLockup.setLockUpBlocks`, `Lockup.calculateWithdrawableInterestAmount`, `ILockup.calculateWithdrawableInterestAmount`, `IMarket.authenticate`, `Market.authenticate`, `Allocator.calculateMaxRewardsPerBlock`, `IAllocator.calculateMaxRewardsPerBlock`, `PolicyTestForWithdraw.setLockUpBlocks`, `PolicyTestForPolicyFactory.setLockUpBlocks`, `PolicyTestForVoteCounter.setLockUpBlocks`, `Killable.kill`, `UsingStorageTest.getEternalStorageAddress`, `PolicyTestForProperty.setLockUpBlocks`, `PolicyTestForTimeVote.setLockUpBlocks`, `Util.blockNumber`, `Util.createKey`, and `PolicyTestForAllocator.setLockUpBlocks` should be made `external`.
- Various numerical literals, including those in `PolicyTestForLockup.rewards`, `Lockup.slitherConstructorConstantVariables`, `PolicyTestForWithdraw.rewards`, `PolicyTestForPolicyFactory.rewards`, `PolicyTestForVoteCounter.rewards`, `Decimals.slitherConstructorConstantVariables`, `PolicyTestForTimeVote.rewards`, `LockupStorageTest.slitherConstructorConstantVariables`, `TheFirstPolicy.authenticationFee`, `TheFirstPolicy.marketApproval`, `TheFirstPolicy.policyApproval`, `TheFirstPolicy.slitherConstructorConstantVariables`, `Property.slitherConstructorConstantVariables`, `DIP1.authenticationFee`, `DIP1.marketApproval`, `DIP1.policyApproval`, and `DIP7.slitherConstructorConstantVariables` have too many zeros. Exponentiation should be used to increase readability and maintainability.
- Many variables, including `DIP1.marketVotingBlocks`, `DIP1.policyVotingBlocks`, `IMarketBehavior.schema`, `Lockup.one`, `MarketTest1.asynchronousMode`, `MarketTest1.currentBlock`, `MarketTest1.lastBlock`, `MarketTest1.metrics`, `MarketTest1.schema`, `MarketTest2.schema`, `MarketTest3.schema`, `TheFirstPolicy.lockUpBlocks`, `TheFirstPolicy.marketVotingBlocks`, and `TheFirstPolicy.policyVotingBlocks` could be made constant.
- The variables `Lockup.one`, `MarketTest1.metrics`, `MarketTest1.lastBlock`, and `MarketTest1.currentBlock` are never used locally; consider removing them.



- Constants `DIP7.basis`, `DIP7.power_basis`, `mint_per_block_and_aseet`, `Property._supply`, `Property._property_decimals`, `DIP1.basis`, `DIP1.power_basis`, `DIP1.mint_per_block_and_aseet`, `AddressValidator.errorMessage`, `LockupStorage.basis`, `TheFirstPolicy.mint_per_block_and_aseet`, `TheFirstPolicy.power_basis`, and `TheFirstPolicy.basis` are not in `UPPER_CASE_WITH_UNDERSCORES`.
- The constants `mint_per_block_and_aseet`, `TheFirstPolicy.mint_per_block_and_aseet`, and `DIP1.mint_per_block_and_aseet` have spelling errors.
- For various computations, a division is performed before a multiplication. For example, the computation involving `p` of L32 in `DIP7.sol`, L38 in `DIP1.sol`, and L38 of `TheFirstPolicy.sol`, the computation using `amount` on L188 of `LockupStorage.sol`, the computation involving `record` on L314 of `LockupStorage.sol`, and the computation involving `cLocked` on L425 of `LockupStorage.sol`. This may result in the value being zero-ed out incorrectly.
- `VoteCounter.sol`: L17: the comment: “That is, at the voting, expecting to pass a Property address for specification the voting right.” is not clear.
- `VoteCounter.sol`: L243, L253: the require error messages, “not use property” and similar on L253 is not clear; we suggest stating which variable/argument is not the desired type.
- We suggest renaming `alreadyVote` to `alreadyVoted` on L76, L162 in `VoteCounter.sol`. Similarly, we suggest renaming the variable `oppositeCount` to `againstCount` or similar (L352, L370).
- We suggest renaming the function `addOppositeCount` to `addAgainstCount` or similar in `VoteCounter.sol`.
- `Withdraw.sol`: The comment “always withdrawal” on L51 should be “always withdraw”.
- `Withdraw.sol`: The function `calculateTotalWithdrawableAmount` replicates some behaviour of the function `_calculateAmount`; perhaps these could be merged for readability and maintainability.
- `AddressValidator.sol`: We recommend renaming `validateAddress` to `validateEqualAddresses` for clarity.
- `DIP1.sol`: The numerical literals on L88, L102 have too many nines; use exponentiation and subtraction if possible.
- `DIP1.sol`: The numerical literals on L118, L108, L94 should be named constants for readability and maintainability.
- `DIP7.sol`: `SafeMath` functions are used but not imported; import that library for readability and clarity.
- `TheFirstPolicy.sol`: The numerical literals on L88, L102 have too many nines; use exponentiation and subtraction.
- `TheFirstPolicy.sol`: The numerical literals on L118, L108, L94 should be named constants for readability and maintainability.
- `TheFirstPolicy.sol`: This appears to be a clone of `DIP1.sol`, except for a numerical change on L59 and a name change. The files could be combined if possible, with the value accepted as a parameter.
- Within the `test/policy` and `test/market` folders: many functions are identical and replicated in various files. The test files could likely be refactored to improve readability and maintainability.
- `Lockup.sol` It’s unclear why there are multiple consecutive divisions by basis in a few lines (649, 604); these operations should use a value that makes it clear that this is intentional. -In `PolicySet.sol`, there is a spelling error in the function name: `getPl icySetIndexKey`; it should be `getPolicySetIndexKey`.
- Consider renaming the function `addGroup` in `MetricsGroup.sol` to `addToGroup`; similarly, consider renaming the function `removeGroup` in `MetricsGroup.sol` to `removeFromGroup`. If there are other files ending in `Group.sol`, these changes may be applicable there.
- Consider renaming the functions `create` and `destroy` in `MetricsFactory.sol` to `createMetrics` and `destroyMetrics`, respectively. Similar functions could be renamed in other `Factory.sol` files.
- In many files, the events emitted are simply `Create` and `Destroy`- these events could be more informative, for example, `MetricsCreated` or `MetricsDestroyed`, respectively. There may be other instances of such an improvement.
- Consider renaming the variable `idHashMetricsMap` in `Market.sol` to `metricsIdHashMap`.
- Consider renaming the function `fee` in `Dev.sol` to `collectFee`.
- Consider renaming the variable `count` in L67 of `VoteCounter.sol` to `vote_power` (and also modifying the error message in L71 accordingly).
- L78-79 in `Lockup.sol` can be replaced by `require(getStorageWithdrawalStatus(_property, _from) == 0, "lockup is already canceled");`.
- L118-119 in `Lockup.sol` can be replaced by `require(getStorageWithdrawalStatus(_property, msg.sender) == 0, "lockup is already canceled");`
- `IMarketBehavior.authenticate` does not need `address market` as a parameter. It can use `msg.sender`.
- The `yarn test` command does not run all tests. It should be updated so that it does; the following was used to run tests for this audit (but not coverage): `"test": "truffle test test/**/*.ts && truffle test test/**/*.ts"`,
- To fix one test in `using-storage.ts`, L46 of that file should be `validateErrorMessage(result, 'storage is set')`.
- To fix the test “The contract is killed and the function cannot be executed” test in `killable.ts`, L14 of that file should be `validateErrorMessage(result, "Returned values aren't valid, did it run Out of Gas?", false)`.
- A command like the following would be helpful: `"ganache": "ganache-cli --port 8545 --deterministic --accounts 10 -e 10000 -k istanbul -m 'curtain .... '",`.

## Test Results

### Test Suite Results

Some test suite output has been removed for the report.

**Update:** the tests are not running for the latest commit in this report. These are the old test results.

Contract: AddressConfigTest

AddressConfig; getter/setter

✓ Value set by owner(allocator) (116ms)

```
✓ Value set by non-owner(allocator) (66ms)
✓ Value set by owner(allocatorStorage) (175ms)
✓ Value set by non-owner(allocatorStorage) (117ms)
✓ Value set by owner(withdraw) (103ms)
✓ Value set by non-owner(withdraw) (129ms)
✓ Value set by owner(withdrawStorage) (133ms)
✓ Value set by non-owner(withdrawStorage) (87ms)
✓ Value set by owner(marketFactory) (80ms)
✓ Value set by non-owner(marketFactory) (56ms)
✓ Value set by owner(marketGroup) (79ms)
✓ Value set by non-owner(marketGroup) (77ms)
✓ Value set by owner(propertyFactory) (73ms)
✓ Value set by onon-wner(propertyFactory) (268ms)
✓ Value set by owner(propertyGroup) (155ms)
✓ Value set by non-owner(propertyGroup) (150ms)
✓ Value set by owner(metricsFactory) (100ms)
✓ Value set by non-owner(metricsFactory) (114ms)
✓ Value set by owner(metricsGroup) (91ms)
✓ Value set by non-owner(metricsGroup) (62ms)
✓ Value set by owner(policyFactory) (87ms)
✓ Value set by non-owner(policyFactory) (59ms)
✓ Value set by owner(policyGroup) (76ms)
✓ Value set by non-owner(policyGroup) (92ms)
✓ Value set by owner(policySet) (123ms)
✓ Value set by non-owner(policySet) (57ms)
✓ Value set by owner(token) (95ms)
✓ Value set by non-owner(token) (60ms)
✓ Value set by owner(lockup) (67ms)
✓ Value set by non-owner(lockup) (186ms)
✓ Value set by owner(lockupStorage) (92ms)
✓ Value set by non-owner(lockupStorage) (56ms)
✓ Value set by owner(voteTimes) (108ms)
✓ Value set by non-owner(voteTimes) (72ms)
✓ Value set by owner(voteTimesStorage) (70ms)
✓ Value set by non-owner(voteTimesStorage) (82ms)
✓ Value set by owner(voteCounter) (108ms)
✓ Value set by non-owner(voteCounter) (47ms)
✓ Value set by owner(voteCounterStorage) (77ms)
✓ Value set by non-owner(voteCounterStorage) (60ms)
AddressConfig; setPolicy
✓ Value set by PolicyFactory (114ms)
✓ Value set by owner (162ms)
✓ Value set by non-owner (158ms)

Contract: UsingConfigTest
UsingConfig; config
✓ You can get the address of config by setting it in the constructor. (129ms)
UsingConfig; configAddress
✓ You can get the address of config . (82ms)

Contract: DecimalsTest
Decimals; outOf
✓ outOf returns ratio of the first args out of second args (124ms)
✓ outOf returns error if the denominator is 10^36 times greater than the numerator (148ms)
Decimals; mulBasis
✓ The value multiplied by basis is returned. (105ms)
✓ Whatever you multiply 0 by 0, you get 0. (106ms)
✓ Large numbers cause overflow. (96ms)
Decimals; divBasis
✓ The value divided by basis comes back. (87ms)
✓ Zero is zero divided by whatever. (75ms)

Contract: KillableTest
Killable; kill
✓ The contract is killed and the function cannot be executed (154ms)
✓ Only deployed accounts can be killed (86ms)

Contract: EternalStorageTest
EternalStorage; getter,setter,deleter
uint
✓ get. (68ms)
✓ delete. (80ms)
✓ get initial value.
✓ cannot be set to other than the owner. (52ms)
✓ cannot be delete to other than the owner. (51ms)
byte32
✓ get. (53ms)
✓ delete. (72ms)
✓ get initial value.
✓ cannot be set to other than the owner. (59ms)
✓ cannot be delete to other than the owner. (61ms)
string
✓ get. (106ms)
✓ delete. (79ms)
✓ get initial value.
✓ cannot be set to other than the owner. (49ms)
✓ cannot be delete to other than the owner. (53ms)
bool
✓ get. (62ms)
✓ delete. (59ms)
✓ get initial value.
✓ cannot be set to other than the owner. (49ms)
✓ cannot be delete to other than the owner. (60ms)
int
✓ get. (61ms)
✓ delete. (62ms)
✓ get initial value.
✓ cannot be set to other than the owner. (41ms)
✓ cannot be delete to other than the owner. (63ms)
address
✓ get. (72ms)
✓ delete. (68ms)
✓ get initial value.
✓ cannot be set to other than the owner. (50ms)
✓ cannot be delete to other than the owner. (57ms)
EternalStorage; upgradeOwner
✓ If the owner changes, the owner can change the value. (105ms)
✓ If the owner changes, the value cannot be changed by the original owner. (79ms)
✓ Even if the owner changes, the value cannot be changed from an unrelated address. (77ms)
✓ Even if the owner changes, owner change is not executed. (56ms)

Contract: UsingStorageTest
UsingStorage; eternalStorage
✓ returns EternalStorage instance (203ms)
UsingStorage; hasStorage, createStorage
✓ If storage has not been created, an error will occur when trying to get the storage address.
✓ If storage has not been created, an error will occur when accessing storage.
✓ If storage has been created, the storage address can be obtained. (79ms)
✓ If the storage has been created, you can access the storage.
✓ Creating storage again after storage has been created results in an error. (58ms)
UsingStorage; getStorageAddress, setStorage, changeOwner
✓ Can get the value set in the storage. (40ms)
✓ the storage address is taken over, the same storage can be accessed from the takeover destination. (207ms)
✓ Before delegating authority, you can not write. (74ms)
✓ Delegation of authority is not possible from the delegate. (47ms)
✓ When delegating authority, the delegate can write to storage (111ms)
✓ When delegating authority, delegation source can not write to storage. (56ms)

Contract: AddressValidatorTest
AddressValidator; validateIllegal
✓ normal address do not cause an error.
✓ default address cause an error. (40ms)
AddressValidator; validateGroup, validateGroups
✓ No error occurs if the address belongs to a market group. (86ms)
✓ No error occurs if the address belongs to a market group.
✓ No error occurs if the address belongs to a property group. (40ms)
✓ No error occurs if the address belongs to a property group.
✓ No error occurs if the address belongs to a metrics group. (40ms)
✓ No error occurs if the address belongs to a metrics group. (55ms)
✓ No error occurs if the address belongs to a policy group. (51ms)
```



```

    ✓ No error occurs if the address belongs to a policy group. (48ms)
    ✓ No error occurs if you belong to either group(ver1). (45ms)
    ✓ No error occurs if you belong to either group(ver2). (39ms)
    ✓ An error occurs if you do not belong to either group.
AddressValidator; validateAddress, validateAddresses
    ✓ No error if addresses are the same.
    ✓ An error occurs if the address is different.
    ✓ No error if either address is the same(ver1).
    ✓ No error if either address is the same(ver2).
    ✓ No error if either address is the same(ver3).
    ✓ An error will occur if the address is different for both.

119 passing (19s)

Contract: LockupTest
  Lockup; calculateWithdrawableInterestAmount
    returns correct amount
      ✓ Alice has a 100% of interests (844ms)
      ✓ Alice has a 100% of interests after withdrawal (706ms)
      ✓ Alice has a 50% of interests (969ms)
      ✓ Alice has a 75% of interests (1297ms)
      ✓ Bob has a 30% of interests before withdrawal (462ms)
      ✓ Bob has a 25% of interests (705ms)
      ✓ Alice can withdraw 5 blocks (728ms)
      ✓ Alice has a 100% of interests (1694ms)
  scenario; single lockup
    before second run
      ✓ Alice does staking 100% of the Property's total lockups (43ms)
      ✓ Alice's withdrawable interest is 100% of the Property's interest (174ms)
    after second run
      ✓ Alice's withdrawable interest is 100% of the Property's interest (117ms)
    after additional staking
      ✓ Alice's withdrawable interest is 100% of the Property's interest (102ms)
    after withdrawal
      ✓ Alice's withdrawable interest is 100% of the Property's interest (117ms)
  scenario: multiple lockup
    before second run
      ✓ Alice does staking 100% of the Property's total lockups (49ms)
      ✓ Bob does staking 25% of the Property's total lockups, Alice's share become 80% (347ms)
      ✓ Alice's withdrawable interest is 100% of between lastBlockNumber and Bob's first deposit block interest and 80% of current interest (426ms)
      ✓ Bob's withdrawable interest is 20% of interest since the first deposit (425ms)
    after second withdrawal
      ✓ Alice's withdrawable interest is 80% of current interest (501ms)
      ✓ Bob's withdrawable interest is 20% of current interest (420ms)
    additional staking
      ✓ Bob does staking 30% of the Property's total lockups, Bob's share become 38.46153846153846%, Alice's share become 61.53846153846154% (44ms)
    after additional staking
      ✓ Alice's withdrawable interest is 80% of prev interest and 61.53846153846154% of current interest (486ms)
      ✓ Bob's withdrawable interest is 20% of prev interest and 38.46153846153846% of current interest (445ms)
    after withdrawal
      ✓ Alice's withdrawable interest is 80% of prev interest and 61.53846153846154% of current interest (391ms)
      ✓ Bob's withdrawable interest is 20% of prev interest and 38.46153846153846% before interest withdrawal by Alice and 100% current interest (440ms)
  scenario: multiple properties
    before withdrawal
      ✓ No staked Property is 0 interest (451ms)
      ✓ Alice does staking 100% of the Property1 total lockups, Property1 is 100% of the total rewards (41ms)
      ✓ Bob does staking 100% of the Property2 total lockups, Property2 is 20% of the total rewards (361ms)
      ✓ Alice's withdrawable interest is 100% of between lastBlockNumber and Bob's first deposit block interest and 80% of current interest (415ms)
      ✓ Bob's withdrawable interest is 20% of interest since the first deposit (411ms)
    after withdrawal
      ✓ No staked Property is 0 interest (399ms)
      ✓ Alice's withdrawable interest is 80% of current interest (405ms)
      ✓ Bob's withdrawable interest is 20% of current interest (453ms)
    additional staking
      ✓ No staked Property is 0 interest (483ms)
      ✓ Bob does staking 30% of the all Property's total lockups, Bob's share become 38.46153846153846%, Alice's share become 61.53846153846154% (44ms)
    after additional staking
      ✓ No staked Property is 0 interest (447ms)
      ✓ Alice's withdrawable interest is 80% of prev interest and 61.53846153846154% of current interest (398ms)
      ✓ Bob's withdrawable interest is 20% of prev interest and 38.46153846153846% of current interest (538ms)
    additional staking
      ✓ No staked Property is 0 interest (429ms)
      ✓ Alice does staking 60% of the all Property's total lockups, Alice's share become 75.96153846153847%, Bob's share become 24.03846153846154% (54ms)
    after additional staking
      ✓ No staked Property is 0 interest (440ms)
      ✓ Alice's withdrawable interest is 80% of two prev interest and 61.53846153846154% of prev interest and 75.96153846153847% of current interest (887ms)
      ✓ Bob's withdrawable interest is 20% of two prev interest and 38.46153846153846% of prev interest and 24.03846153846154% of current interest (485ms)
    after withdrawal stakes
      ✓ No staked Property is 0 interest (469ms)
      ✓ Alice's withdrawable interest (947ms)
      ✓ Bob's withdrawable interest (476ms)
  scenario: fallback legacy locking-ups
    before withdraw interest
      ✓ No staked Property is 0 interest (434ms)
      ✓ Alice's withdrawable interest is correct (422ms)
      ✓ Bob's withdrawable interest is correct (486ms)
    after withdraw interest
      ✓ No staked Property is 0 interest (412ms)
      ✓ Alice's withdrawable interest is correct (113ms)
      ✓ Bob's withdrawable interest is correct (105ms)
    after withdraw
      ✓ No staked Property is 0 interest (402ms)
      ✓ Alice's withdrawable interest is correct (104ms)
      ✓ Bob's withdrawable interest is correct (410ms)
  scenario: fallback legacy locking-ups and latest locking-ups
    before withdraw interest
      ✓ No staked Property is 0 interest (399ms)
      ✓ Alice's withdrawable interest is correct (447ms)
      ✓ Bob's withdrawable interest is correct (408ms)
    after withdraw interest
      ✓ No staked Property is 0 interest (396ms)
      ✓ Alice's withdrawable interest is correct (103ms)
      ✓ Bob's withdrawable interest is correct (91ms)
    after withdraw
      ✓ No staked Property is 0 interest (402ms)
      ✓ Alice's withdrawable interest is correct (107ms)
      ✓ Bob's withdrawable interest is correct (456ms)

63 passing (1m)
```

Code Coverage

The code coverage computed does not include all tests; only the 63 tests in the original configuration were executed by Solidity-Coverage. Since this excludes 119 tests, the actual code coverage may be much higher than is reported below.

Update: the tests are not running for the latest commit in this report. These are the old test resuts.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
allocator/	60	100	66.67	60	
Allocator.sol	60	100	66.67	60	43,45
IAAllocator.sol	100	100	100	100	



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
<b>common/config/</b>	68	100	66.67	68	
AddressConfig.sol	77.27	100	76.19	77.27	... 179,188,205
UsingConfig.sol	0	100	0	0	15,22,29
<b>common/interface/</b>	0	100	0	0	
IGroup.sol	0	100	0	0	9
<b>common/libs/</b>	0	0	0	0	
Decimals.sol	0	0	0	0	... 25,27,35,42
<b>common/lifecycle/</b>	0	0	0	0	
Killable.sol	0	0	0	0	13,21,22
<b>common/storage/</b>	0	0	0	0	
EternalStorage.sol	0	0	0	0	... 165,173,181
UsingStorage.sol	0	0	0	0	... 46,47,55,63
<b>common/validate/</b>	0	0	0	0	
AddressValidator.sol	0	0	0	0	... 70,72,73,75
UsingValidator.sol	0	100	0	0	16,23
<b>dev/</b>	21.05	12.5	42.86	21.05	
Dev.sol	40	25	60	40	... 60,68,69,70
DevMigration.sol	0	0	0	0	... 37,41,42,43
<b>lockup/</b>	69.6	65.79	42.31	69.47	
ILockup.sol	100	100	100	100	
Lockup.sol	97.53	65.79	100	97.52	958,963,964,967
LockupStorage.sol	0	100	0	0	... 425,427,434
<b>market/</b>	0	0	11.11	0	
IMarket.sol	100	100	100	100	
IMarketBehavior.sol	100	100	100	100	
IMarketFactory.sol	100	100	100	100	
IMarketGroup.sol	100	100	100	100	
Market.sol	0	0	0	0	... 228,233,240
MarketFactory.sol	0	0	50	0	... 46,47,50,51
MarketGroup.sol	0	0	16.67	0	... 39,43,44,48
<b>metrics/</b>	29.03	16.67	60	29.03	
IMetricsFactory.sol	100	100	100	100	
IMetricsGroup.sol	100	100	100	100	
Metrics.sol	0	100	0	0	12,13
MetricsFactory.sol	0	0	33.33	0	... 65,66,71,72
MetricsGroup.sol	60	25	83.33	60	... 40,41,42,43
<b>policy/</b>	13.92	11.54	23.08	13.75	
DIP1.sol	0	0	0	0	... 117,118,120
DIP3.sol	0	100	0	0	10
DIP7.sol	0	100	0	0	... 42,43,44,45
IPolicy.sol	100	100	100	100	
IPolicyFactory.sol	100	100	100	100	
IPolicyGroup.sol	100	100	100	100	
IPolicySet.sol	100	100	100	100	
PolicyFactory.sol	47.62	33.33	66.67	45.45	... 86,88,94,95
PolicyGroup.sol	57.14	25	75	57.14	28,33,34
<b>PolicySet.sol</b>	<b>24.24</b>	<b>100</b>	<b>30.77</b>	<b>23.53</b>	<b>... 78,79,83,91</b>
TheFirstPolicy.sol	0	0	0	0	... 117,118,120

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
<b>property/</b>	45.71	35.71	63.64	45.71	
IProperty.sol	100	100	100	100	
IPropertyFactory.sol	100	100	100	100	
Property.sol	0	0	0	0	... 140,145,146
PropertyFactory.sol	100	50	100	100	
PropertyGroup.sol	100	50	100	100	
<b>vote/</b>	0	0	3.45	0	
IVoteCounter.sol	100	100	100	100	
VoteCounter.sol	0	0	9.09	0	... 370,371,372
VoteCounterStorage.sol	0	100	0	0	... 183,194,203
<b>withdraw/</b>	0	0	5.88	0	
IWithdraw.sol	100	100	100	100	
Withdraw.sol	0	0	8.33	0	... 323,324,331
WithdrawStorage.sol	0	100	4.55	0	... 207,217,227
<b>All files</b>	<b>29.21</b>	<b>19.44</b>	<b>27.65</b>	<b>28.68</b>	

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

809210335c45670c34f39c4f3d6e8f32719c294bfbf38c7c863922343c61fb20 ./contracts/Migrations.sol

857ce26ffe5ececf6f6b5d099062c6949e030a80db6c7fbc9a22cabf290260d9 ./contracts/test/Util.sol

c156b0cfe9c97cb5edadc872c5d552e1d8f349ad37f63b47e4a120b1421e58ea ./contracts/test/withdraw/WithdrawStorageTest.sol

32c4d5b59b82a563e515bb7cd2b7d79834127e65046e65e818fcb4464f731b99 ./contracts/test/withdraw/WithdrawTest.sol

f5b886222035dfb3a58138884789f9b3ac9176cef9cc0df04c30995b40787f32 ./contracts/test/vote/VoteCounterStorageTest.sol

f67bb1c9c555c6f180a960a144fa87d3dc7736412783ee7e2220b50e029cd60c ./contracts/test/policy/PolicyTest1.sol

8433af22db5ca48c48ebbe618f1d87f0786843a74cb41be3f6d96eac2cf24ff7 ./contracts/test/policy/PolicyTestForAllocator.sol

e46b95864a4e5898919ecd07c8c93ae2f7ddd67a52a8851d17239085975ffb20 ./contracts/test/policy/PolicyTestForLockup.sol

01119631975d14712e12acb2ed5aaea789216d427373995b2cbaeb2bf583d242 ./contracts/test/policy/PolicyTestForPolicyFactory.sol

196397fe0a4ecff01da832ae1ce4d61564e18bcf492aaa4e8f387c44933939fd ./contracts/test/policy/PolicyTestForProperty.sol

536ffb3f91c16b13dcaab586584aa161da54f726568d8e1b6761791403bd0f70 ./contracts/test/policy/PolicyTestForTimeVote.sol

2aff03a2785d111a737daf694c19d0cfa4e1a937e799f60e51369dc9f51607eb ./contracts/test/policy/PolicyTestForVoteCounter.sol

6634f46dcecc320c722d1f7d015579bb3d598dd60661fd796e2d163a8387ffe8 ./contracts/test/policy/PolicyTestForWithdraw.sol

b495b8fbfbd935cace757e1799817022cbe9acc625c2eb8764a352b75301e8d ./contracts/test/metrics/MetricsGroupTest.sol

5526aab1365f6bbb81ed886f516459174fe2786fa23c1e76f865f786b94ea41e ./contracts/test/market/MarketTest1.sol

6839ffc99a1006b727848506f4c1ab61ed0f6545ef3ad8795154cd92241c52cc ./contracts/test/market/MarketTest2.sol

271dd25745fbc1a8ff45d5d0309643f6dafffa95532e4e923ea20005952cca62 ./contracts/test/market/MarketTest3.sol

7e42b72b199f77bcb3d1f02ae486f4d624c449ae8cf926f64697c93a93d2d2f ./contracts/test/lockup/LockupStorageTest.sol

fc1b919d125ad1bb8f76713e6d32731560ccd153f704c1ac29edf7ef1bc154ef ./contracts/test/common/storage/UsingStorage.sol

80051a0bed969bc02dc3a79e368e20f35f848dce1593df33a923bb5531a5dc69 ./contracts/test/common/lifecycle/Killable.sol

2c70cbafc4a52db5a3b81c5387bee2e503e7919ae1dd041ec9a7872e570cef5 ./contracts/test/common/libs/Decimals.sol

17ca4ae73b87d080e8be761c49710d6e2225c49ff412269e7263bf8016f1dad7 ./contracts/test/common/config/UsingConfig.sol

d167a396612eacbe94a0328f58149a1af73199f4d934229f32fb55c70bdcc999 ./contracts/src/withdraw/IWithdraw.sol

ce413145b270538b6436496038e4704bf70b18f31486d7e50af4890fc151e4ee ./contracts/src/withdraw/Withdraw.sol

4adc9824da3be7481215d7cb5eb4c70ec6c2637a20e6ea8c6fba05dc5f8cc84e ./contracts/src/withdraw/WithdrawStorage.sol

8d1399314ca86da78c4dc98342989188705c6bec1ebb9858cd1353e401701ddb ./contracts/src/vote/IVoteCounter.sol

8c756c45ca170b280e47edddc8cb1cce3aa4687eeab19d7721aa6b7b72e4e8a9 ./contracts/src/vote/VoteCounter.sol

017646ba7dfd8a91f06cb5dc7f10e55bb31bc7bae45615eebbef5662d5a89eec ./contracts/src/vote/VoteCounterStorage.sol

48951a76b64396d312ad1c3840aead1f6e67f6ec3a0642865fb8e675eb1d6cb1 ./contracts/src/property/IProperty.sol

788b604a206075b4c772a81275408db43debcf74659c773d19ac248ae1589ce0 ./contracts/src/property/IPropertyFactory.sol

d72311cef6b0b3b0ed19e48f46d9b52dfa8341f9defb31de471ec55b75801ad5 ./contracts/src/property/Property.sol

61018c3692d233423ff07e2b42d5718fcc0ddcb9cb136fe43aaa83f32ab324ea ./contracts/src/property/PropertyFactory.sol

aea1d726a31509d9d0fe6f4f0ce323fca5a479a75e4220dfc39611f5533c8d44 ./contracts/src/property/PropertyGroup.sol

f8e4d201b7b3c881e7300d663472bc0a922c77da2070112314483310dc4c9c37 ./contracts/src/policy/DIP1.sol



70be0ca4478f03bdb4028c79b5439ee1e0caef472c4898783488cf568bcdad77 ./contracts/src/policy/DIP3.sol

0430a36c5b6cf0bbf88fbb5414f10706104ceb953bbb3ddf5e6a78d702c4d885 ./contracts/src/policy/DIP7.sol

fd783133a0d506c0887ec001ec7945e0e9bb05869d24d916dc131aece30b84ce ./contracts/src/policy/IPolicy.sol

83999d603e5b599185306af81ddfdaf028defb48e8b0ebf7ac6840af03c9bcab ./contracts/src/policy/IPolicyFactory.sol

0cb5d6b9666f2ed4ecd3b24cc3d75204bff8a361d822ec60788b5fce8b426677 ./contracts/src/policy/IPolicyGroup.sol

21bae42496b974831dbd6c114e93c12e4abd4513d662555e86d472168b13c1fe ./contracts/src/policy/PolicyFactory.sol

f65c27824d88e3aa04341447250ad4f362e3ec6004547d9a18bc7bf78680f5e6 ./contracts/src/policy/PolicyGroup.sol

5cb5e9bc81e89f9ef5611d5d0bf7cb43e976c1345db62d4422f0d0d0f37affd3 ./contracts/src/policy/TheFirstPolicy.sol

f0fca976800f61c493370a5cd86e8c9ab400aa7247ed9390f74eadc7207e0d0d ./contracts/src/metrics/IMetrics.sol

e4d2251111118ec69ff0a958ec7e03d25e3a8de3799f18dc17890f9ea33d0456 ./contracts/src/metrics/IMetricsFactory.sol

f74ee45bcd49e30eaf830fba78b28bd6ac11c86e3f61e7b7718c460dec6cc61 ./contracts/src/metrics/IMetricsGroup.sol

6e4c68300e1f992f4fb2550ce9c1addf9f88bd94d8ab810c3fe085913968e0a0 ./contracts/src/metrics/Metrics.sol

ec172072fa45f9fce902ab1bf8fb2da692d3fd07552e681b7eb0233f12431994 ./contracts/src/metrics/MetricsFactory.sol

136200c7c78c570288acf491042fd783a9e230db34fb2bd22935bfc2a3175191 ./contracts/src/metrics/MetricsGroup.sol

d9aa2e0191bfbdb9b361264c4c006ef680bb9d17fe770487fbbcd5137dea64a08 ./contracts/src/market/IMarket.sol

e1e961e62fcb03c282679cfaf452ca699a65fc9734c8aaf4143dcc93f30d5329 ./contracts/src/market/IMarketBehavior.sol

8f6d580e7e336cab6741cf35ff1e6c9657a40bcd5c40d1a3a8d0b9daf21beec0 ./contracts/src/market/IMarketFactory.sol

67fd5ff87ec214d7cc517b8ff62ac5f61f620027c8343ebe6824345eeade300d ./contracts/src/market/IMarketGroup.sol

921c5797f7718cbc6e4890d0e160fa64351b33f7dfa6329dc422b380cca60e46 ./contracts/src/market/Market.sol

35ea10ebb3679e9956f6e160d92ce38bab9c976d5e1077cf113d5ed9ac16580f0 ./contracts/src/market/MarketFactory.sol

484a653ecaa8920ddfd432ace80e49231efd19601c3222b4431c12c8ab7b420d ./contracts/src/market/MarketGroup.sol

f5c17c2585d96d52978c5746aa60cd92209c15a20611fd18567ad99f90544ed0 ./contracts/src/lockup/ILockup.sol

f23da75a2f96c2a23b80c19959bf6dd9885c6d95260c0664349c839a6a399601 ./contracts/src/lockup/Lockup.sol

9fb6cc243ba59928d8d20d38eea11cc85a3681da00d474d74beb0e6da454898a ./contracts/src/lockup/LockupStorage.sol

14202e6daab61559e3be8867b2bc608f4197476772c7622bbebfafec0d1d41f8 ./contracts/src/dev/Dev.sol

a2a580b2ce84f207ffa2be37e06ac452644822b2cb2316242878911a1d36f00f ./contracts/src/dev/DevMigration.sol

0ad246932273e0d83e5197a73a8e187a65c17ac81586b95606c9552f1d655f82 ./contracts/src/common/validate/AddressValidator.sol

c29e5ca4a79684f6c08794443279485844e124cfc9c88d213f9a46b387585712 ./contracts/src/common/validate/UsingValidator.sol

0428578b9468ff13a03c7b6f00e9f1298790dff776d942857667610914643233 ./contracts/src/common/storage/EternalStorage.sol

2f70e3404df387016ecbbf42a1a9d31a53e8f4b40843cbec2b3d220421b03007 ./contracts/src/common/storage/UsingStorage.sol

cb7b788b6c7f811c41204b88a6db5d22e8bae1266cad616e9d094380c1f2fee8 ./contracts/src/common/lifecycle/Killable.sol

0276cc65d9b53a1a3f327dc1307b76c6490d42986019dca1e955daeed97f9251 ./contracts/src/common/libs/Decimals.sol

68c33f17d9c4597231ec8ef810a8fb9e15f89f00cff6176459975ede6aec60d9 ./contracts/src/common/interface/IGroup.sol

c66b1722c12968e3f98946177f5e8d0cc5eb5547d7454fa356274f78af6f632c ./contracts/src/common/config/AddressConfig.sol

fe4018319d055683acc15a6a3be0bafdb310a6eae9882ad9bc61dc8cbcd61d8 ./contracts/src/common/config/UsingConfig.sol

8fe0c162e0c519cdc3fadca30e8f8228b9eeacf4a70096d3f6403bac921e97d7 ./contracts/src/allocator/Allocator.sol

5d7edaa0b52048e6cec1b7633818e5b447ac5cfce6ca282c4f39cb6ba4b29425 ./contracts/src/allocator/IAAllocator.sol

Tests

d6b90feb2d48492af250b7c8adb330dab85e6bc36210de828a6c2b0117d3503c ./test/withdraw/withdraw-storage.ts

4912fba782bf3e3065c7113f98c30e4261d58c6db6b9daddac140d8bb2927472 ./test/withdraw/withdraw.ts

94efcf7de407951bbe81f776218b37d1c0e99fcffc6c16d9b97e8f5bfff7234b3 ./test/vote/vote-counter-storage.ts

d466dcc72d0fb5b8aa94fe3b93389b1c2f44f98de36536cda65d0ddaeea1b8ad ./test/vote/vote-counter.ts

9845003df929f443e2f5d27124314bf945712bd55c6eec767a7bd1fcb969d4e1 ./test/test-lib/const.ts

f452a96e9e052ee407d8bcd991796ca094f60625d2f3fdcacb584042bfff761 ./test/test-lib/instance.ts

e0d39f8929bddb0779c56e7ef81d58a4f4f26e4597f531b583ef59d4548bfe79 ./test/test-lib/utills/common.ts

ffd7c2add9df3884d1778b755043b716c498f6e43b5442b2527e98c9c05901fc ./test/test-lib/utills/error.ts

aea1a6cb0fd091aeb3ba981406409ff6fa614013782887d8de8d410f7b44d4bb ./test/test-lib/utills/event.ts

a40fa861f57c4dc9d79d0e1166b9398d063fb709b627f864175be5d36022f535 ./test/test-lib/utills/log.ts

cd517e1646fa733ccafc64ed7a7e713ee4b85311ee969d436b52266b0fffb7041 ./test/test-lib/utills/mint-amount.ts

336141a14caf31dff47c0d76bd23d8688450acc7534125887c7516f6aa60906d ./test/property/property-factory.ts

f0862d1759be72d6dcf7efebf48d34a2113f83710970dc8dd3a8a6945c8b95bc ./test/property/property-group.ts

8164ede9ff2b7c00aa9441b514b5b088c5fe4fa8cd03c5bc536194608b520fc9 ./test/property/property.ts

24bf2ddf677df379145e90296b1995d0f359f03d7f226b3c8d4c17c0a13fbc04 ./test/policy/dip1.ts

cab86b151b1eaa3b34057b88c7e505c0feb0fc22d6cbcb0509cf72aef4bf9b1d ./test/policy/dip3.ts

2c9bbc16529cbdc88bba85db1b01de8f2f6a2fd69f02ea4bd6061f95496315ca ./test/policy/dip7.ts

f13def59434d6b3604a202118a8dda5f3a8053aedecb5224dcffe7123c3bdfc0 ./test/policy/policy-factory.ts

a0b3cf456c805c9a00cfa592769f4cc0fd50f1e0c5698b8abe1bc10ce1575948 ./test/policy/policy-group.ts

71ecce88b11c2d62a8ffe9daf6091b052cd01dfe29be8fd4e226df27a98163ce ./test/policy/policy.ts

53d4233b12e9c080d5a465ad39b052b2fb0aa9b955d07ccf547f08599d691ee6 ./test/policy/the-first-policy.ts

a610be65796569403fb5697c02fe5cbe23a20fb592b2a4a60e1220b77149e178 ./test/metrics/metrics-factory.ts

54738a3e460241197cc5f87457364241c129f6857d8fa8997f5a51500e9f1f74 ./test/metrics/metrics-group.ts  
b79f7c5a94113be0d3c35be3f54b0338be3707e2f5c65ae0f9d1f6916153ce68 ./test/metrics/metrics.ts  
92e8e88169ef86ad5397a755e9f4b6f4055c81efb9843bc47f47217b5f261575 ./test/market/market-factory.ts  
a2f374a0e2528b1a4b8d49c9d18e510f1477966542e3e27e41f6c03cadd57002 ./test/market/market-group.ts  
2a1e14b939498475dd037c6a060811f7d8e1d22c7d385c0090e8fc8e4c838ade ./test/market/market.ts  
a0137326017ab1cb05f0174c799cdba45835f33ce6bb63e8e84ad6b7c0b5bc2c ./test/lockup/lockup-storage.ts  
48fdbaca30513186a6cf2b805fda541bad46091c774b4d6d27b625a298ced829 ./test/lockup/lockup.ts  
bf24848f6e5a0db4125e1e55f0af0d35f6213ccfa36f220c1983f606a4cb7327 ./test/dev/dev-migration.ts  
cbd6bd5ee262c1706871e4b4dbf50153dd34ed5f5c7f0b04a0f7f357a99bb1e7 ./test/dev/dev.ts  
176e7c316c24d17d18cb2cd43f8f61c6786374b3af75d87d72d582c2c7fffe76 ./test/common/validate/address-validator.ts  
ceb7cf9b32e837b37610322cfbf6d38020efc1c26f3b849b387badc071725993 ./test/common/storage/eternal-storage.ts  
449bdd86c91d1a23260995d0368e4380287bf1fb2a11ec684b7c9225b7ce3435 ./test/common/storage/using-storage.ts  
fc30c634517da212217fadd03fcc91dd395b7a5bd5ae92130ec0bf9cb44caba0 ./test/common/lifecycle/killable.ts  
9e917148d4e6a68ca3231327b89be4231ed831f3a6e071d8e03c5a20a379b76c ./test/common/libs/decimals.ts  
bf002104a4c774b2344ace7971c19f980807b6a02981e98bc6411e956422fd3f ./test/common/config/address-config.ts  
4726f981e6bff3e4fa7ea0b788748e7523782992a7fa50538a40556fdf6fed70 ./test/common/config/using-config.ts  
c9f68e7d73daa09994acf37fb66f80e61022b25a8c52a8f4bd522f22c57934c7 ./test/allocator/allocator.ts

## Changelog

- 2020-08-20 - Initial report [7a32e1e]
- 2020-08-24 - Revised report [db2544d]
- 2020-09-15 - Revised report [b53007b]



# 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.