Quantstamp Security Assessment Certificate

November 30th 2021 — Quantstamp Verified

StormX Governance Token

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

Executive Summary

Type	ERC20
Auditors	Jan Gorzny, Blockchain Researcher Jake Bunce, Research Engineer Roman Rohleder, Research Engineer
Timeline	2021-10-04 through 2021-10-15
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Low
Test Quality	High

Source Code

Repository	Commit
athens-token	0c590bf

Total Issues	9 (7 Resolved)
High Risk Issues	1 (1 Resolved)
Medium Risk Issues	0 (0 Resolved)
Low Risk Issues	4 (3 Resolved)
Informational Risk Issues	4 (3 Resolved)
Undetermined Risk Issues	0 (0 Resolved)

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

Summary of Findings

Quantstamp has reviewed the StormX governance token and found several issues that have been addressed. Some were likely unavoidable (e.g., QSP-4 and QSP-6), but others (e.g., QSP-2 and QSP-5) were easily fixed and improved the quality of the code. QSP-1, the only high severity issue, is a result of the use case for the token, may result in unexpected voting scenarios, but is mitigated. Otherwise, the code is of fairly high quality with lots of comments, and there are many tests.

ID	Description	Severity	Status
QSP-1	Voting can be manipulated	High	Mitigated
QSP-2	Unlocked Pragma	Low	Fixed
QSP-3	Missing Input Validation	Low	Fixed
QSP-4	Race Conditions / Front-Running	Low	Acknowledged
QSP-5	Greedy Contract	Low	Fixed
QSP-6	Gas Usage / `for` Loop Concerns	Informational	Acknowledged
QSP-7	Privileged Roles and Ownership	Informational	Fixed
QSP-8	Redundant constructor code	Informational	Fixed
QSP-9	Constructor argument verification	Informational	Mitigated

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

Steps taken to run the tools:

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

Findings

QSP-1 Voting can be manipulated

Severity: *High Risk*

Status: Mitigated

File(s) affected: `Governance.sol`

Description: `lock()` does not implement a timer to denote the lock period. This means there is the scope for abuse of voting in the following scenario:

1. A snapshot vote is created
2. A malicious user buys the governance token on the open market
3. The malicious user uses the recently purchased tokens to vote on the snapshot
4. The tokens are then sold back to the market by the malicious user

In this scenario the vote is no longer reflective of the governance community and has been swayed by the malicious actor.

Recommendation: Consider the impact of the maniuplation of governance voting and whether this is a desirable attribute of the governance token. If it is not desirable, one way to mitigate this scenario is to implement a timer on the locked tokens where a voter may only vote while the tokens are locked and the lock period is greater than the snapshot vote period.

Update: This is mitigated by the proposed use of snapshot.org, which uses wallet balance for configured block number in the proposal. For example:

1. I create a proposal with block number 100 and have 1 token on the wallet A
2. when it's block number 101, I send my 1 token to wallet B so I don't have any more tokens on wallet A
3. when it's block number 102, I still vote on the proposal with voting power 1 using wallet A (because it had 1 token when it was block number 100)

This is documented in snapshot https://docs.snapshot.org/proposals/vote.

QSP-2 Unlocked Pragma

Severity: *Low Risk*

Status: Fixed

File(s) affected: `Governance.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".

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

Update: The pragma has been locked.

QSP-3 Missing Input Validation

Severity: *Low Risk*

Status: Fixed

File(s) affected: `Governance.sol`

Description: Function `initialize` does not check that `owner_` is non-zero. This may result in no owner.

Recommendation: Add the check or make it clear in the documentation that this is acceptable.

Update: A check has been added.

QSP-4 Race Conditions / Front-Running

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `Governance.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.
In this case, `initialize` could be front-run.

Recommendation: Make sure you have a way to re-deploy the contract so that it can be set to the right address, if you are front-run the first time it is deployed.

Update: The team has acknowledged this issue: "Since we are deploying the debut release of the contract, there is no value for someone in front-runs because we will catch that we are not the owner of the contract. Even if that happens, we will re-deploy the contract, making the attack pointless."

QSP-5 Greedy Contract

Severity: *Low Risk*

Status: Fixed

File(s) affected: `Governance.sol`

Description: A greedy contract is a contract that can receive ether which can never be redeemed.
In this case, tokens which are transferred to this contract may be locked.

Recommendation: In accordance with best practices, to prevent tokens being accidentally stuck in the ERC20 contract itself, it is recommended to prevent the transferal of tokens to the contracts address. This can be achieved, by i.e. adding require statements to transfer functions, similar to `require(recipient != address(this));`.

Update: A check has been added to prevent the contract from receiving these tokens to the `transfer` function and to the `transferFrom` function.

QSP-6 Gas Usage / `for` Loop Concerns

Severity: *Informational*

Status: Acknowledged

File(s) affected: `Governance.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. `transfers()` does not check the length of the arrays passed. Arrays with a very large number of elements could cause this function to revert due to exceeding the block size during execution.

Recommendation: Consider adding an upper bound to the length of arrays passed into this function to ensure it will always complete.

Update: The team has acknowledged this finding: "The owner/company mainly uses “transfers” function to distribute the tokens in batches, and regular users are not likely to execute this function. A note in readme file and contract comment has been added to indicate this assumption."

QSP-7 Privileged Roles and Ownership

Severity: *Informational*

Status: Fixed

File(s) affected: `Governance.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract.
In this case, the contract owner initially has all the tokens.

Recommendation: This centralization of power needs to be made clear to the users, especially depending on the level of privilege the contract allows to the owner.

Update: While the `owner` still exists, there is now also a description of the role in the README.

QSP-8 Redundant constructor code

Severity: *Informational*

Status: Fixed

File(s) affected: `Governance.sol`

Description: `initialize()` calls `__Governance_init()` with the same arguments passed to `initialize()`. This can be better implemented with a single upgradable `constructor`.

Recommendation: Use a single `initialize()` function to handle the initialization of contract variables at deployment time.

Update: Redundant initialization code has been moved to `initialize()` function.

QSP-9 Constructor argument verification

Severity: *Informational*

Status: Mitigated

File(s) affected: `Governance.sol`

Description: `initialize()` does not emit an event with the arguments supplied at deploy time. This is useful to verify the intended arguments are in use within the contract.

Recommendation: Emit an event after the arguments are used in the contract, or introspect the variables from the deployment tooling.

Update: From the team: "[A test] `scripts/deploy-token.ts` has been added to the repository, and it prints all the deployment arguments in the console. Units tests also validate the supplied initialization arguments."
We consider the issue mitigated since there is no event emitted, but there is clearly an intent to verify the initial value's correctness (unfortunately, we cannot verify that the team will inspect the variables, though they have increased our confidence that they will pay attention to this issue by adding the test.)

Automated Analyses

Slither

Slither reported that some functions could be made `external`, some variables shadow others (which appears intentional), the pragma is too new to be trusted, and that some functions are not in mixed case (which also appears intentional).

Adherence to Best Practices

1. For improved readability it is recommended to have a maximum line length of 79 or 99. Therefore following lines should be shortened accordingly, which exceed these limits: `Governance.sol`: L50, L102, L103, L115, L125 and L132. **Update:** The line length is capped at 99.

Test Results

Test Suite Results

```
Governance
  ERC20
     ✓ reverts if initialize() called when owner is zero address (57ms)
     ✓ has correct name
     ✓ has correct symbol
     ✓ has correct number of decimals
     ✓ has correct total supply
  Staking
     ✓ reads locked wallet balance successfully and emits TokenLocked event
     ✓ reads unlocked balance successfully and emits TokenUnlocked event (59ms)
  Transfers
     ✓ reverts when transfering to the contract (43ms)
     ✓ sends transfer successfully (78ms)
     ✓ reverts a transfer if not enough unlocked token (41ms)
     ✓ reverts transferFrom if not enough unlocked token (53ms)
     ✓ uses transferFrom successfully (87ms)
     ✓ reverts lock if not enough unlocked token (38ms)
     ✓ locks successfully (67ms)
     ✓ reverts unlock if not enough locked token
     ✓ unlocks tokens successfully (203ms)
     ✓ reverts if input lengths do not match in transfers
     ✓ reverts if any transfer fails
     ✓ uses transfers successfully
  Upgradability
     ✓ reverts if initialize() called more than once


20 passing (4s)
```

Code Coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	100	100	100	100	
Governance.sol	100	100	100	100	
All files	**100**	**100**	**100**	**100**	

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

```
9dcc5f2cbef97b2828b049a4dff29efdd4035fb90a60506c62756d2740058808 ./contracts/Governance.sol
```

Tests

```
4b5236de3f70dc766963e72de4686ea504b950fb9a064ddaea656c463e1e55bc ./test/Governance.spec.ts
37f38b00d98d231a241f305661649c41c44b85a6d9ab7d0ea250a2b1cdb447b3 ./test/shared.ts
6478ef7c1c7ced3733833d697280f9c111a15594681a15226f35d01b3c7183d1 ./test/types.ts
```

Changelog

- 2021-10-05 - Initial report [cd7d902]
- 2021-10-14 - Revised report [751206f]
- 2021-10-15 - Revised report [0c590bf]

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.