Quantstamp Security Assessment Certificate

February 24th 2023 — Quantstamp Verified

Origami

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

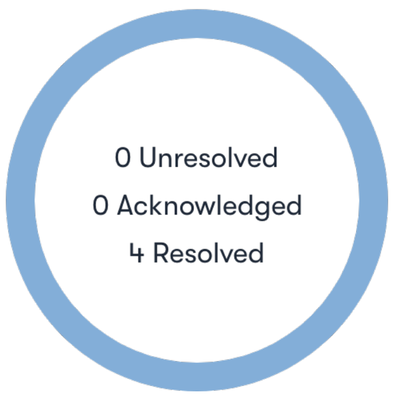
Executive Summary

Type	Upgradable ERC20 Token
Reviewers	Kacper Bak, Research Engineer Sebastian Banescu, Senior Research Engineer
Timeline	2023-02-22 through 2023-02-23
Languages	Solidity
Methods	Architecture Review, Unit Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Medium
Test Quality	High

Source Code

Repository	Commit
https://github.com/JoinOrigami/crane/blob/0b9576ac8de2e841cf88286d9f694a47472091c3/src/token/governance/ERC20Base.sol	0b9576a initial audit

Total Issues	4 (4 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	1 (1 Resolved)
Low Risk Issues	3 (3 Resolved)
Informational Risk Issues	0 (0 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).
Fixed	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

The reviewed file `ERC20Base.sol` is an upgradable ERC20 token. Overall the code is well-written, well-documented, and follows best practices (such as reusing existing libraries). We have not found any significant security vulnerabilities apart those related to the variety of roles used in the management of this contract.
Update: the team has mitigated the issues by renouncing roles for this contract deployed on Ethereum mainnet:

1. The `PAUSER_ROLE`,
2. The `MINTER_ROLE`,
3. The `DEFAULT_ADMIN_ROLE`,
4. Creating a new `ProxyAdmin`,
5. Updating the token to use the new `ProxyAdmin`, and
6. Renouncing Ownership on the new `ProxyAdmin`.

ID	Description	Severity	Status
QSP-1	Burning/transfers Still Possible when Burning/transfers Are Disabled	^ Medium	Mitigated
QSP-2	Allowance Double-Spend Exploit	v Low	Mitigated
QSP-3	Upgradability	v Low	Mitigated
QSP-4	Privileged Roles and Ownership	v Low	Mitigated

Quantstamp Review Breakdown

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

DISCLAIMER:
If the final commit hash provided by the client contains features that are not within the scope of the review or an associated fix review, those features are excluded from consideration in this report. The scope of this review has been limited to the file `src/token/governance/ERC20Base.sol`. Furthermore, mitigations to the various issues have been checked only on Ethereum mainnet.

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 reviewing 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, clarity, 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 security review.

Setup

Tool Setup:

- Slither v0.8.3

Steps taken to run the tools:

1. Install the Slither tool: `pip3 install slither-analyzer`
2. Run Slither from the project directory: `slither .`

Findings

QSP-1 Burning/transfers Still Possible when Burning/transfers Are Disabled

Severity: *Medium Risk*

Status: Mitigated

File(s) affected: `ERC20Base.sol`

Description: This issue stems from the fact that the modifer specifications and implementations are not aligned for the following two modifiers:

1. `whenBurnable` is said to "allows us to ensure that something may only occur when burning is enabled". However, the implementation contains the following condition: `hasRole(BURNER_ROLE, _msgSender()) || burnable()`, which is true also when burning is disabled and the message sender has the `BURNER_ROLE`.
2. `whenTransferrable` is said to "allows us to ensure that something may only occur when the transfers are enabled". However, the implementaiton contains the following condition: `hasRole(TRANSFERRER_ROLE, _msgSender()) || transferrable()`, which is true also when transfers are disabled and the message sender has the `TRANSFERRER_ROLE`.

This means that even if the `DEFAULT_ADMIN_ROLE` disables burning and/or transfers, users that have the `BURNER_ROLE` or the `TRANSFERRER_ROLE` will still be able to burn or transfer tokens.

Recommendation: The semantics of these two modifiers is overloaded in the sense that it wants to be a fast way of checking if transfers are enabled and if the message sender had the appropriate role to perform this action. However, this should be done by separate modifiers in order to function properly.
Note that the "obvious fix" of changing the `||` into a `&&` may not work as intended because it will result in the modifier returning false even if burning or transfers are enabled, but the message sender does not have the corresponding role to perform that action.

Update: The team has clarified that this behavior is intended despite the comments stating otherwise. Furthermore, the team never enabled `BURNER_ROLE` and `TRANSFERRER_ROLE`, disabled burning, renounced the admin role, and renounced proxy ownership.

QSP-2 Allowance Double-Spend Exploit

Severity: *Low Risk*

Status: Mitigated

File(s) affected: `ERC20Base.sol`

Description: As it presently is constructed, the contract is vulnerable to the allowance double-spend exploit, as with other ERC20 tokens. Please note that this is inherent to the ERC20 standard and not specific to this contract.

Exploit Scenario:

1. Alice allows Bob to transfer `N` amount of Alice's tokens (`N>0`) by calling the `approve()` method on `Token` smart contract (passing Bob's address and `N` as method arguments)
2. After some time, Alice decides to change from `N` to `M` (`M>0`) the number of Alice's tokens Bob is allowed to transfer, so she calls the `approve()` method again, this time passing Bob's address and `M` as method arguments
3. Bob notices Alice's second transaction before it was mined and quickly sends another transaction that calls the `transferFrom()` method to transfer `N` Alice's tokens somewhere
4. If Bob's transaction will be executed before Alice's transaction, then Bob will successfully transfer `N` Alice's tokens and will gain the ability to transfer another `M` tokens
5. Before Alice notices any irregularities, Bob calls the `transferFrom()` method again, this time to transfer `M` Alice's tokens.

Recommendation: The exploit (as described above) is mitigated through the use of functions that increase/decrease the allowance relative to its current value, such as `increaseAllowance()` and `decreaseAllowance()`. Furthermore, we recommend that developers of applications dependent on `approve()` / `transferFrom()` should keep in mind that they have to set the allowance to `0` first and verify if it was used before setting the new value.

Update: The team has informed us that besides the inline comments in the code, they are going to put that information into documentation.

QSP-3 Upgradability

Severity: *Low Risk*

Status: Mitigated

File(s) affected: `ERC20Base.sol`

Description: While upgradability is not a vulnerability in itself, token holders should be aware that the token contract can be upgraded at any given time. This audit does not guarantee the behavior of future contracts that the token may be upgraded to.

Recommendation: The fact that the contract can be upgraded and reasons for future upgrades should be communicated to users beforehand.

Update: The team disabled upgradability by changing the `ProxyAdmin`, and then renouncing ownership.

QSP-4 Privileged Roles and Ownership

Severity: *Low Risk*

Status: Mitigated

File(s) affected: `ERC20Base.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract. Specifically, the contract features the following roles:

1. `PAUSER_ROLE` can pause the contract (admin by default)
2. `MINTER_ROLE` can mint new governance tokens (admin by default). Note that there is a cap on the total supply.
3. `BURNER_ROLE` can burn governance tokens. No user is granted this role by default.
4. `TRANSFERRER_ROLE` can transfer governance tokens (admin + DAO treasury multisig).

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: The team renounced the privileged roles: `PAUSER_ROLE`, `MINTER_ROLE`, and `DEFAULT_ADMIN_ROLE`.

Automated Analyses

Slither

Slither did not report any issues.

Adherence to Best Practices

The semantics of the `TransferEnabled` and `BurnEnabled` events is inaccurate because the Boolean parameter value indicates if the transfers/burning is enabled (when `value=true`) or disabled (when `value=false`). We recommend considering renaming these events to something like `TransferStatus` and `BurnStatus`.

Test Results

Test Suite Results

```
Running 7 tests for test/token/governance/ERC20Base.t.sol:MintingGovernanceTokenTest
[PASS] testEmitsAnEventWhenMinting() (gas: 73270)
[PASS] testMinting(uint96) (runs: 256, μ: 75429, ~: 76829)
[PASS] testMintingRevertsWhenMintingMoreThanCap() (gas: 24812)
[PASS] testMintingRevertsWhenMintingMoreThanCapAfterMinting() (gas: 73856)
[PASS] testMintingRevertsWhenMintingToZeroAddress() (gas: 22779)
[PASS] testMintingRevertsWhenNotMinter() (gas: 62302)
[PASS] testMintingRevertsWhenPaused() (gas: 57738)
Test result: ok. 7 passed; 0 failed; finished in 26.16ms
```

Code Coverage

Code coverage for `ERC20Base.sol` is close to 100% in all categories.

File	% Lines	% Statements	% Branches	% Funcs
src/token/governance/ ERC20Base.sol	96.77% (30/31)	96.77% (30/31)	100.00% (2/2)	94.12% (16/17)

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

`2b95289277f7388eeb62fa3d09087b41cb7e466d9f0156f128a13bb4f4c0387f ./governance/ERC20Base.sol`

Tests

`bd5a1c900c863df476f7e63268d7bce377dfc8989260b6af4e409b08beae8cd1 ./governance/ERC20Base.t.sol`

Changelog

- 2023-02-23 - Initial report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Aave, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

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 or other making available of the report to you by Quantstamp.

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. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites&aspo; 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 any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims 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. You agree that your access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. 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. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or 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, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. 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.