Quantstamp

Hashflow Governance

Executive Summary

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

Type	Governance
Timeline	2022-01-20 through 2022-02-02
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Source Code	• governance #f05d720 • governance #c5c1328
Auditors	• Souhail Mssassi Research Engineer • Alex Murashkin Senior Software Engineer • Sung-Shine Lee Research Engineer

Documentation quality	Undetermined
Test quality	Medium
Total Findings	5 Fixed: 4 Acknowledged: 1
High severity findings ⓘ	0
Medium severity findings ⓘ	0
Low severity findings ⓘ	2 Fixed: 2
Undetermined severity findings ⓘ	1 Acknowledged: 1
Informational findings ⓘ	2 Fixed: 2

Summary of Findings

Final Audit:
Through reviewing the code, we found **5 potential issues** of various levels of severity: `2` low-severity, `2` informational-severity and `1` undermined-severity issues. All the issues were resolved/acknowledged.

ID	DESCRIPTION	SEVERITY	STATUS
QS-1	**Possibility to remove all the minters.**	• Low ⓘ	Fixed
QS-2	**Missing address verification**	• Low ⓘ	Fixed
QS-3	**Approve Race**	• Informational ⓘ	Fixed
QS-4	**Mismatch between specs and the code**	• Informational ⓘ	Fixed
QS-5	**Undocumented design decision to use** `uint96`	• Undetermined ⓘ	Acknowledged

Assessment 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

1. Code review that includes the following
   1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   3. 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:
   1. 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.
   2. 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.

Findings

QS-1 Possibility to remove all the minters.

● Low ⓘ Fixed

> **Update**
> The team has resolved the issue in commit `c5c13289d54f952ca6b91b906d04684d17605aec` by adding a verification on the address of the minter `L(140)`.

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

Description: In the function `setMinter(address minter_)` allows setting the minter to 0x0. Once the minter is 0x0, it is not possible to add any more minters, and therefore, further minting will not be possible.

Recommendation: If renounce is desired, create a function that does that specifically. As for `setMinter`, explicitly require that the new minter cannot be the address 0.

QS-2 Missing address verification

● Low ⓘ Fixed

> **Update**
> The team has resolved the issue in commit `c5c13289d54f952ca6b91b906d04684d17605aec` by adding the necessary verifications on the constructor `L(118,123)`.

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

Description: Certain functions lack a safety check in the address, the address-type argument should include a zero-address test, otherwise, the contract's functionality may become inaccessible or tokens may be burned in perpetuity.
File(s)

- `contracts/HFT.sol` (L108);
- `contracts/HFT.sol` (L109);
- `contracts/HFT.sol` (L128);

Recommendation: It's recommended to undertake further validation prior to user-supplied data. The concerns can be resolved by utilizing a whitelist technique or a modifier.

QS-3 Approve Race

● Informational ⓘ Fixed

Update

The team has resolved the issue in commit `c5c13289d54f952ca6b91b906d04684d17605aec` by adding the `increaseAllowance` and `decreaseAllowance` functions `L(233,253)`.

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

Description: The standard ERC20 implementation contains a widely-known racing condition in its `approve` function, wherein a spender is able to witness the token owner broadcast a transaction altering their approval and quickly sign and broadcast a transaction using `transferFrom` to move the current approved amount from the owner's balance to the spender. If the spender's transaction is validated before the owner's, the spender will be able to get both approval amounts of both transactions.

Recommendation: Use `increaseAllowance` and `decreaseAllowance` functions to modify the approval amount instead of using the `approve` function to modify it.

QS-4 Mismatch between specs and the code

● Informational ⓘ Fixed

Update

The team has resolved the issue in commit `c5c13289d54f952ca6b91b906d04684d17605aec` by setting the value of `mintingAllowedAfter` as a parameter in the constructor.

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

Description: In the `HFT` contract `L(25)`, it says that `The timestamp after which minting may occur (must be set to 4 years)`, but in the constructor the `mintingAllowedAfter` variable can be set to any value greater than `now`.

Recommendation: Set the value of `mintingAllowedAfter` in the constructor to have the value `now + 4 years`

QS-5 Undocumented design decision to use `uint96`

● Undetermined ⓘ Acknowledged

Update

From the Hashflow Team :

```
This design decision was made to pack 2 checkpoint structs at the same storage to save gas when
moveDelegates is called
```

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

Description: In the constructor and other functions we remarked that there is casting operations to `uint96`, there is no mention of the purpose of such a type in the documentation, and therefore, auditors are unable to assess the related risks.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – 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.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **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).

Code Documentation

The code comes with very little inline documentation. We recommend documenting the code.

Adherence to Best Practices

1. `domainSeparator` is constant and can be pre-computed to save gas.
2. SafeMath is not needed after the version 0.8 .

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.

Files

- `d05...282 ./contracts/HFT.sol`
- `66e...1b0 ./contracts/lib/SafeMath.sol`

Tests

- `bec...ddc ./test/hft.spec.ts`
- `191...219 ./test/utils.ts`

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.8.3

Steps taken to run the tools:

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

Automated Analysis

Slither

Slither reported the following:

- Lacks a zero-check on `HFT.constructor,HFT.setMinter(address)`
- Uses timestamp for comparisons on `HFT.mint(address),HFT.delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32)` All the issues areaddressed in the report.

Test Suite Results

All tests executed successfully. We reviewed the test suite, we recommend expanding the test suite significantly to ensure that the code works as expected. Furthermore, a good test suite would contain both positive and negative test cases.

```
HFT
  ✓ set allowance via permit (214ms)
  ✓ vote delegation (487ms)
  ✓ delegate via signature (99ms)
  ✓ mints (546ms)
```

```
4 passing (2s)
```

Code Coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	74.38	45	86.36	75.21	
HFT.sol	74.38	45	86.36	75.21	... 443,446,537
contracts/lib/	57.89	33.33	50	57.89	
SafeMath.sol	57.89	33.33	50	57.89	... 150,170,171
All files	72.14	43.06	76.67	72.86	

Changelog

- 2022-01-25 - Initial report
- 2022-02-01 - Final 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, 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.

Quantstamp