Quantstamp Security Assessment Certificate

June 14th 2022 — Quantstamp Verified

GMO GYEN

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

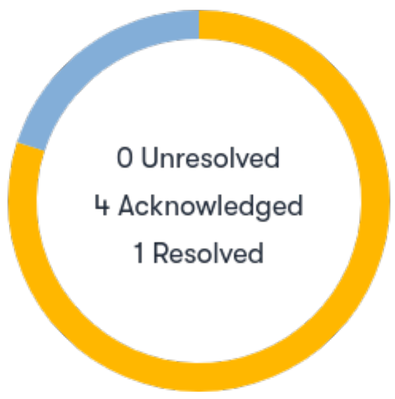
Executive Summary

Type	Token Contract
Auditors	Fayçal Lalidji, Senior Security Engineer Hisham Galal, Research Engineer
Timeline	2022-05-16 through 2022-06-14
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Readme Whitepaper
Documentation Quality	Medium
Test Quality	Low

Source Code

Repository	Commit
gmotrust-stablecoin-contract	5af3ba3
gmotrust-stablecoin-contract	c7c3261

Total Issues	**5** (1 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	**1** (1 Resolved)
Low Risk Issues	**2** (0 Resolved)
Informational Risk Issues	**2** (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

Through reviewing the code, we found **6 potential issues.** We recommend carefully re-considering the logic to ensure the safety of users. In addition, the contracts grant many privileges to trusted roles, which opens room for misuse of users' assets.
Reaudit update: All highlighted issues have been either fixed or acknowledged.

ID	Description	Severity	Status
QSP-1	Prohibited Addresses Can Still Receive Funds	^ Medium	Fixed
QSP-2	Initialization Front-Running	⌄ Low	Acknowledged
QSP-3	Zero Value Transfer Not Allowed	⌄ Low	Acknowledged
QSP-4	Documentation Is Not In Natspec Format	○ Informational	Acknowledged
QSP-5	Privileged Roles and Ownership	○ Informational	Acknowledged

Quantstamp Audit Breakdown

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

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

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

Setup

Tool Setup:

- Slither v0.8.3

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 Prohibited Addresses Can Still Receive Funds

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `Token_v1.sol`

Description: Prohibited addresses are still allowed to receive funds. It is unclear if this is part of the application design since the funds will be frozen and the only way to rescue the funds is to unprohibited the address.

Recommendation: Clarify if the described issue is a design choice.

QSP-2 Initialization Front-Running

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `Wiper.sol`, `Token_v3.sol`

Description: Multiple Initialization functions are public and can be called by anyone. An attacker might front-run the deployer and assign itself one of the initializable roles. The related functions are listed below:

- `Wiper.initializeWiper()`.
- `Token_v3.InitializeV3(...)`.

Recommendation: We recommend adding access restrictions to the listed functions or acknowledging that the deployed contracts will be verified after the initialization.

Update: "Initialization functions are designed to be called once, and we will acknowledge that the deployed contracts will be verified after the initialization".

QSP-3 Zero Value Transfer Not Allowed

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `Token_v1.sol`

Description: `Token_v1.transfer` and `Token_v1.transferFrom` do not allow transfers of zero amount which is a correct and safe behavior for ERC20 tokens. Some Decentralized apps might rely on events triggered to update or perform actions as stated in EIP20 standard, any transfer event "MUST trigger when tokens are transferred, including zero value transfers.".

Recommendation: We recommend to carefully asses the assoociated miss behavior that such logic may trigger.

Update: "It is part of the application design since transfer 0 tokens with no meaning for our application".

QSP-4 Documentation Is Not In Natspec Format

Severity: *Informational*

Status: Acknowledged

Description: Solidity contracts can use a special form of comments to provide rich documentation for functions, return variables, and more. This special form is named the Ethereum Natural Language Specification Format (NatSpec). According to the latter, “It is recommended that Solidity contracts are fully annotated using NatSpec for all public interfaces (everything in the ABI)”. However, the existing audited contracts are not in Natspec.

Recommendation: Document all contracts according to the NatSpec format.

Update: "Will check this in future".

QSP-5 Privileged Roles and Ownership

Severity: *Informational*

Status: Acknowledged

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract.
The audited implementation contains the following privileges:

- The max allowed cap to be minted can be reset to a higher value using `Token_v1.cap(...)`.
- Users wallets can be blacklisted using the implemented prohibition logic
- Users accounts balances can be wiped after it they have been prohibited using `Token_v2.wipe(...)`.

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: "It had been addressed outside of the program (README)".

Automated Analyses

Slither

Slither did not raise any significant issue.

Adherence to Best Practices

1. Consider using role-based access control by Openzepplin instead of having roles contracts defined in `/contracts/roles/*.sol`.
2. Consider re-implementing `notMoreThanCapacity` modifier such that it takes `amount` rather than `tokenSupply().add(amount)` as a parameter.

Test Results

Test Suite Results

The tests could not be generated due to `git` falling due to timeout.

Code Coverage

The code coverage could not be generated due to `git` falling due to timeout.

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

```
096948f39d4fdc9649811978cae490118b7c2583887c941782c40893637fb3dc ./contracts/Burning.sol
e15471a9030045288eeff7a63cc3bbbcdc167e33064bd1ccc104e84cabeb26e6 ./contracts/BurningFactory.sol
836908563de461c5f3cfa9b5f0d32b56f8106438f0a10bc62fef832961478b96 ./contracts/Token_v1.sol
50594f93ebf90b619873c067505f6d834acf9dddc3984df8419e5a39a5db2dbb ./contracts/GYEN.sol
f19e0b3f73a979f556cc87082dff05abbf75b4d277197af1f2a10493bac6dde8 ./contracts/ZUSD.sol
7ddfcf9ebf85ed23cb86786946f3105fc8ffcb8f249c4ccb495a55515ccd787c ./contracts/Migrations.sol
febe4f744961bb8b321c39d9a1fbfe7333202e3d40a87a701425c5b0771558da ./contracts/Token_v3.sol
e29ed09ff2163d9336cbd504bf12f5ef18db1d8eba61e27c6cd6bb56c13cf918 ./contracts/Token_v2.sol
eddc1bae7ef345a89cebb98ef620921653627f595449d7dfe30eefd2ac6474e3 ./contracts/Roles/Rescuer.sol
386d697a1e43ad16c2f9965f8cb8ca38a28d13880492d948f9fdf56b144fda5e ./contracts/Roles/Owner.sol
cc8f7faf785b3457b47e6f063b94adbb2803d42d995a8cc40285aed8e18f6bcf ./contracts/Roles/Admin.sol
98d495de9099f0be3d6ccd9271cefed0af5cc682e16b7e4e848e9d5a6d2ef8c8 ./contracts/Roles/Pauser.sol
63963c0fa57e50f3ab8394dc26af5367d0065e072724693f6fd05810558def06 ./contracts/Roles/Prohibiter.sol
5410bbd28737380654e7b7ae941d8e16f13e45d53e2ac40cf176c2ad78db5a0c ./contracts/Roles/Minter.sol
bf3f3f60dd4280e4cac64604272aa8885ec9b3a401a535a818b87aa0acdebb49 ./contracts/Roles/Capper.sol
0c5b5607682a426593c6a1bec73ec3dfac4a6ef29a25b7a507137695e53f7382 ./contracts/Roles/MinterAdmin.sol
1565319ee69fd6fefd0ca0e90619fa59f8dcb7712dbe06d120ab2e4e439fa05a ./contracts/Roles/Common.sol
ed6baf3a6615f085ea9634fa8ac2fda3d67e673d22c85480b2be18d32d790a50 ./contracts/Roles/Wiper.sol
6aab2181edfcb1d53eca73afdb1929af519e84dc7261d14fd16ebf1ad3848e51 ./contracts/Roles/Operators.sol
```

Tests

```
e2f04294e81bced62ec485ed7e85b5b3010dc7de7ab93b20ae6ca6b2d2121d9f ./test/burning.js
918300fff1ad6bc4acf0f87153b1d83a0228538c1a0efd6ed7fdc4a1949e6d5c ./test/zusd.js
8041c14aa19c55f0e287494194ccf4a8332ddb61ec57fe8c8c56bcd1dbe22787 ./test/upgrade_gyen_zusd_to_v3.js
9f501932870ea85ff1ca7e2d96399190b0e683ba16ef5585e2516a265239d306 ./test/validate_storage_slots_helper.js
955b3b476410ca16a4b350c3f2d7af7e94c8a4d42aecbb3a7ad7ec07723e60b2 ./test/gyen.js
731ca3a78c4a5065e030b8abe6abdfe72deb7230f1a8603cb16d66629687b54a ./test/burning_factory.js
2d1cbd825d7fab3481fefb1f38435f142caa619b3215de6d83cf0c0a02526895 ./test/token_v1.js
8216f6713851e45f917499c9a96c89c91ec0f683001fdee0d333bca2e9f30ed8 ./test/upgrade_gyen_zusd_to_v2.js
95fd7083f2a988a021d5f4fc35dd6029ca2a0f806714b2a45c261fdd918de001 ./test/roles/admin.js
c4125ae20de16fa1e3cfcb2a9e433e8778cbdf43846d84adcafb5f4c2dd796a0 ./test/roles/minterAdmin.js
bc1e1af6edc9d83d1ca10e3c36b7650cf1b35bf73dfd10db6638332a83e8a25e ./test/roles/owner.js
b0159fbd9c828012c04bbff4c752814c2561651ff0544ca5fe65159a256040a6 ./test/roles/pauser.js
dbd0074f07423c042c116de34885f375defa740a3e42519bd8cd4afa33327aba ./test/roles/prohibiter.js
```

Changelog

- 2022-05-18 - Initial report
- 2022-06-14 - Final report (c7c3261)

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.