Quantstamp Security Assessment Certificate

February 14th 2023 — Quantstamp Verified

Seamoon Protocol

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

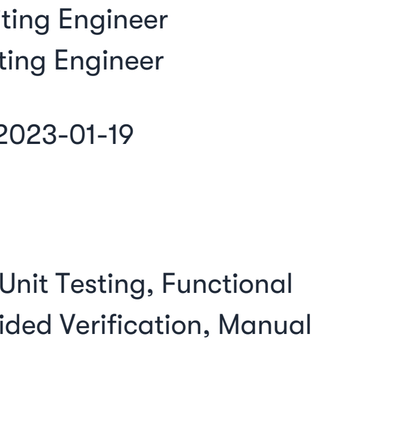
Executive Summary

Type	ERC-20 Tokens
Auditors	Guillermo Escobero, Auditing Engineer Sina Pilehchiha, Auditing Engineer Mustafa Hasan, Auditing Engineer Mostafa Yassin, Auditing Engineer
Timeline	2023-01-17 through 2023-01-19
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Whitepaper (Draft) Seamoon Project Overview - Slides
Documentation Quality	High
Test Quality	High

Source Code

Repository	Commit
dm2c/token-contracts	bb85a0a initial audit

Total Issues	**6** (5 Resolved)
High Risk Issues	**1** (1 Resolved)
Medium Risk Issues	**2** (2 Resolved)
Low Risk Issues	0 (0 Resolved)
Informational Risk Issues	**3** (2 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

Quantstamp team audited DM2E and DM2P ERC-20 smart contracts. No high-severity issues were found. Some recommendations about privileged accounts are discussed, as well as some informational issues about the Solidity version used or best coding practices.

We strongly recommend creating public technical documentation about these smart contracts, **listing the use cases and current and future goals of the project and clearly defining its privileged roles.** The project includes a test suite with high code-coverage metrics.

Fix review: Quantstamp reviewed the fixes proposed by the Seamoon team. All the issues were addressed correctly. The Seamoon team updated the whitepaper and improved the code comments of the smart contracts.

ID	Description	Severity	Status
QSP-1	Missing Documentation	⇈ High	Fixed
QSP-2	Privileged Roles and Ownership	^ Medium	Mitigated
QSP-3	Ownership Can Be Renounced	^ Medium	Fixed
QSP-4	Allowance Double-Spend Exploit	○ Informational	Acknowledged
QSP-5	Unlocked Pragma	○ Informational	Fixed
QSP-6	Design Concerns	○ Informational	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.

DISCLAIMER:
If the final commit hash provided by the client contains features that are not within the scope of the audit or an associated fix review, those features are excluded from consideration in this report.

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 v0.9.2

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 Missing Documentation

Severity: *High Risk*

Status: Fixed

Description:

1. Based on the documentation provided, the administrative roles of `DM2E` and `DM2P` are not mentioned nor explained. Due to the high impact that these users can have on the system, we find it critical to document the rationale behind this design.
2. The whitepaper mentions `DMM` token, but it is not clear if it refers to `DM2E` or `DM2P`. This is critical when explaining the token distribution and total supply cap.
3. The relation between `DM2E` and `DM2P` is not documented.

Recommendation: Provide public documentation about the goals of each administrative role, as well as if the project is using/will be using external security measures such as multi-signature accounts. Clarify the rationale of each token, and why they need to be pausable, mintable, and capped (or not).

Update: The whitepaper was updated to include detailed explanations. The client provided the following explanation:

> *Extra documents are updated for describing the goals of the administrative role and external security measures the project will use.*

QSP-2 Privileged Roles and Ownership

Severity: *Medium Risk*

Status: Mitigated

Description: The `DEFAULT_ADMIN_ROLE`, `MINTER_ROLE`, `PAUSER_ROLE`, and `BURNER_ROLE` roles are all assigned to the `msg.sender` once both the `DM2E` and `DM2P` contracts are deployed, which allows the deployer full control over both contracts.

1. `MINTER_ROLE` users are allowed to mint an arbitrary amount of tokens.
2. `PAUSER_ROLE` users are allowed to pause transactions anytime.

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. Document the goals and privileges of each role in public documentation.

Update: Addressed in commit: `caf172ab3900a4a27c2a201cafed03af182c8203`. The client provided the following explanation:

> *Extra documents are updated for describing the goals and privileges of each role.*

QSP-3 Ownership Can Be Renounced

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `DM2E.sol`, `DM2P.sol`

Description: `AccessControl` provides `revokeRole(...)` and `renounceRole(...)` functions. Users with any role can call this function and renounce their privileges. If the user is a unique administrator (role `DEFAULT_ADMIN_ROLE`) the contract will be left without administrators and no one will be able to call functions with `onlyRole(DEFAULT_ADMIN_ROLE)` modifier. Please be also aware that an administrator can revoke roles to other users (including role administrators)` .

Recommendation: Confirm this is the intended behavior. Override `revokeRole(...)` and `renounceRole(...)` functions so that ownership cannot be renounced if there is only one administrator. Multi-signature wallets are recommended to perform operations with these privileged accounts.

Update: Marked as "Fixed" by the client. Addressed in: `d58e1024bf5cfbb620e6fe841d68f8dab49399c2`.

QSP-4 Allowance Double-Spend Exploit

Severity: *Informational*

Status: Acknowledged

Description: As it presently is constructed, the contract is vulnerable to the allowance double-spend exploit, as with other ERC20 tokens.

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 client provided the following explanation:

> *The ERC20 allowance double-spend exploit can be mitigated by introducing* `increaseAllowance()` *and* `decreaseAllowance()` *methods. While the original issue remains in the code, the likelihood is reduced because users may utilize the newly added alternative methods.*

QSP-5 Unlocked Pragma

Severity: *Informational*

Status: Fixed

File(s) affected: `DM2E.sol`, `DM2P.sol`

Related Issue(s): SWC-103

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.8.*`. 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, we recommend removing the caret to lock the file onto a specific Solidity version. It is recommended to use Solidity version `0.8.16` based on risks of known bugs, new language features, and recent releases.

Update: Addressed in: `4c5a850fc6314b3e057351fe46df281b2a94cf8c`.

QSP-6 Design Concerns

Severity: *Informational*

Status: Fixed

File(s) affected: `DM2E.sol`, `DM2P.sol`

Description: `DM2E.burn()`, `DM2P.burn()`, `DM2E.burnFrom()` and `DM2P.burnFrom()` are overridden just to add the `onlyRole(BURNER_ROLE)` modifier.

Recommendation: We recommend calling `super.burn(amount)` and `super.burnFrom(account, amount)` instead of copying the original function implementations.

Update: Addressed in: `76ff124cdc1747d9e9c5369e8f9346c355c29da2`.

Automated Analyses

Slither

1. Slither detected 28 results. All of them were triaged as false positives or discussed in this document.
2. ERC-20 compliance in `DM2E` and `DM2P` was verified using `slither-check-erc`. No issues were detected.
3. **Update:** after the fix review, Slither detected 33 results. All of them were triaged as false positives or discussed in this document. ERC-20 compliance in `DM2E` and `DM2P` was verified using `slither-check-erc`. No issues were detected.

Code Documentation

No comments whatsoever are present in the contracts. It is recommended to add comments to the contracts however trivial the functionalities therein may seem. **Fixed:** inline comments were added by the development team.

Test Results

Test Suite Results

All tests passed.

Update: The Seamoon team added `8` more tests to the test suite.

```
testing for DM2E
  Deployment
     ✓ Should assign the total supply of tokens to the owner
  Transactions
     ✓ Should transfer tokens between accounts
     ✓ Should fail if sender doesn’t have enough tokens
     ✓ Should update balances after transfers
  Mint
     ✓ Should mint initial supplies correctly
     ✓ Should allow admin to mint
     ✓ Should fail to mint when users other than admin signs
  pause
     ✓ Should allow admin to paused and unpaused
     ✓ Should fail when pause by non-admin
  burn
     ✓ Should allow burn by admin
     ✓ Should fail when burn by non-admin
     ✓ Should allow burnFrom by admin
     ✓ Should fail when burnFrom by non-admin
     ✓ Should fail when exceeds the approve
  AccessControl
     ✓ Should grant initial DEFAULT_ADMIN_ROLE correctly
     ✓ Should allow admin to grant role
     ✓ Should fail when grant role by non-admin
     ✓ Should allow admin to revoke role
     ✓ Should allow admin to revoke role
     ✓ Should fail when revokeRole DEFAULT_ADMIN_ROLE by last admin
     ✓ Should fail when renounceRole DEFAULT_ADMIN_ROLE by last admin
     ✓ Should fail when revokeRole MINTER_ROLE by last admin
     ✓ Should fail when renounceRole MINTER_ROLE by last admin

testing for DM2P
  Deployment
     ✓ Should assign the total supply of tokens to the owner
  Transactions
     ✓ Should transfer tokens between accounts
     ✓ Should fail if sender doesn’t have enough tokens
     ✓ Should update balances after transfers
  Mint
     ✓ Should mint initial supplies correctly
```

```
      ✓ Shoud set cap correctly
      ✓ Should allow admin to mint
      ✓ Should fail to mint when users other than admin signs
      ✓ Should fail when exceeds the cap
    pause
      ✓ Should allow admin to paused and unpaused
      ✓ Should fail when pause by non-admin
    burn
      ✓ Should allow burn by admin
      ✓ Should fail when burn by non-admin
      ✓ Should allow burnFrom by admin
      ✓ Should fail when burnFrom by non-admin
      ✓ Should fail when exceeds the approve
    AccessControl
      ✓ Should grant initial DEFAULT_ADMIN_ROLE correctly
      ✓ Should allow admin to grant role
      ✓ Should fail when grant role by non-admin
      ✓ Should allow admin to revoke role
      ✓ Should allow admin to revoke role
      ✓ Should fail when revokeRole DEFAULT_ADMIN_ROLE by last admin
      ✓ Should fail when renounceRole DEFAULT_ADMIN_ROLE by last admin
      ✓ Should fail when revokeRole MINTER_ROLE by last admin
      ✓ Should fail when renounceRole MINTER_ROLE by last admin


  48 passing (3s)
```

Code Coverage

The code coverage analysis shows good metrics with a `90%` branch coverage. We recommend improving it to `100%`.

Update: The Seamoon team improved the branch coverage to `91.67%`.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	100	91.67	100	100	
DM2E.sol	100	91.67	100	100	
DM2P.sol	100	91.67	100	100	
All files	**100**	**91.67**	**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

`2b0b3c2f8a1dae701657c653ce0578e2505ba694a01126a186b7419e5c0e3f5d ./contracts/DM2E.sol`

`65f070bbaa9ad629b345a5548e72465a1a1f6219da63a3f269080877ce04ef33 ./contracts/DM2P.sol`

Tests

`b151fdac261d671f2b4e8b5dc6f30cde9b852cc2654dea6ec39d62fb21c57d96 ./tests/DM2E.test.ts`

`4b43bfb0fefc6d88a39196eadfc6b5fa8e8d935cf5e9f7abc9e87264e76d4e07 ./tests/DM2P.test.ts`

Changelog

- 2023-01-23 - Initial report
- 2023-02-13 - Updated the report according to commit `caf172ab3900a4a27c2a201cafed03af182c8203`.

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.