Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED SECURITY CERTIFICATE

December 20th 2021 — Quantstamp Verified

JPEG'd Part 2

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

Executive Summary

FAST RESPONSE TIMES

ALL ISSUES ADDRESSED

Type	NFT Lending
Auditors	Jose Ignacio Orlicki, Senior Engineer Cristiano Silva, Research Engineer Marius Guggenmos, Senior Research Engineer
Timeline	2021-12-07 through 2021-12-20
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Medium
Test Quality	High
Source Code	

Repository	Commit
jpegd	679bb3c

Total Issues	**10**	(0 Resolved)
High Risk Issues	1	(0 Resolved)
Medium Risk Issues	2	(0 Resolved)
Low Risk Issues	3	(0 Resolved)
Informational Risk Issues	4	(0 Resolved)
Undetermined Risk Issues	0	(0 Resolved)

0 Unresolved
10 Acknowledged
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

This report is based on new features for token vesting and token sale included in the JPEG'd project (`PreJPEG`, `TokenVesting` and `TokenSale`). We have reviewed the code, documentation, and test suite and found several issues of various severities. The test suite is very extensive but can be improved given the suggested changes from this report. Even if the features and interfaces implemented are commonly seen in Decentralized Finance (DeFi) applications it is recommended to include more public documentation on these new features. We have outlined suggestions to better follow best practices, and recommend addressing all the findings to tighten the contracts for future deployments or contract updates. We recommend addressing all the **10** findings to harden the contracts for future deployments or contract updates. We recommend against deploying the code as-is.
Update: all *10* issues were acknowledged including reaudit notes by the JPEG'd team. The documentation can still be improved. The testing is very good (100% on if-branch coverage) but can be improved further close to getting more `require()` coverage.

ID	Description	Severity	Status
QSP-1	Weak Validation of the Oracle's Output	High	Acknowledged
QSP-2	Use Of `token.decimals` Instead Of `oracle.decimals`	Medium	Acknowledged
QSP-3	Sweep Of Unexpected Tokens Missing	Medium	Acknowledged
QSP-4	Risk Of Governance Takeover	Low	Acknowledged
QSP-5	External Contracts Are Not Checked For Interface	Low	Acknowledged
QSP-6	Centralization Of Power	Low	Acknowledged
QSP-7	Ownership Can Be Renounced	Informational	Acknowledged
QSP-8	Unnecessary Dynamic Array For `supportedTokens`	Informational	Acknowledged
QSP-9	Gas optimization in `PreJPEG.release()`	Informational	Acknowledged
QSP-10	Reentrancy Guard Not Required	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 v0.8.0

Steps taken to run the tools:

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

Findings

QSP-1 Weak Validation of the Oracle's Output

Severity: *High Risk*

Status: Acknowledged

File(s) affected: `TokenSale.sol`

Description: Since Oracle is an external entity, the code must be defensive against issues that may arise in this external entity. The code is already checking if the returned price is non-zero.

Recommendation: Consider including a check for anomalous, besides non-zero, values, for example checking if the returned value is within a feasible range. Also, consider a fallback oracle for anomalous conditions.

Update: Acknowledged by the JPEG'd team and detailed response says *"The oracles are used only once in the finalizeRaise function. This function can only be called by the owner and will be called right after an oracle update with satisfying values, effectively mitigating the risk of bad oracle values."*.

QSP-2 Use Of `token.decimals` Instead Of `oracle.decimals`

Severity: *Medium Risk*

Status: Acknowledged

File(s) affected: `TokenSale.sol`

Description: Line 188 multiplies the result of `oracle.latestAnswer()` by `token.decimals` when it should be multiplying by the oracle's decimals. Unproper use of decimals can lead to some financial loss in some cases and is usually not tested in detail.

Recommendation: Replace `token.decimals` with `oracle.decimals` and store it in the `supportedTokensData` to reuse it in `getUserClaimableTokens`.

QSP-3 Sweep Of Unexpected Tokens Missing

Severity: *Medium Risk*

Status: Acknowledged

File(s) affected: `TokenSale.sol`

Description: The probability that some unsupported ERC20 tokens are sent to the TokenSale contract is high. Implementing a function to retrieve such tokens can result in some financial gain.

Recommendation: Consider adding a sweepToken(address token) function that can only be called by the owner. This function should allow transferring out any tokens that are not part of `supportedTokens`.

Update: Acknowledged by the JPEG'd team and detailed response says *"While it is possible that people will send tokens to the contract, it's very unlikely that they'll send tokens other than WETH and USDC since those are the tokens that are going to be used during the raise. Any WETH or USDC sent directly to the contract will be collected by the transferToTreasury function. As far as other tokens go, we don't think it's worth modifying the codebase just to sweep them."*.

QSP-4 Risk Of Governance Takeover

Severity: *Low Risk*

Status: Acknowledged

Description: If the voting power gets too much concentrated on a small community due to a hostile takeover, a bug, or a security incident on a token exchange, we have the risk of governance takeover. Currently, the functional parts for this audit include only vote delegation, so this issue should be fully considered when governance is fully operational.

Recommendation: The development team must discuss if there are chances of governance takeover, and what measures will be adopted to mitigate this (perhaps limiting the speed of governance operation using Time Locks). Otherwise, document this possibility.

Update: Acknowledged by the JPEG'd team and detailed response says *"We switched from on-chain governance to off-chain governance (snapshot.org). This completely removes the possibility of a governance takeover as the multisig will have the final say for every proposal."*.

QSP-5 External Contracts Are Not Checked For Interface

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `TokenSale.sol`

Description: Even if it is common practice in DeFi, is not a good practice to construct or parametrize contracts with external Oracles and external Tokens that do not satisfy the interfaces. This can lead to delays or cost of opportunity losses during launches or regular operations.

Recommendation: Consider applying ERC-165 Standard Interface Detection (https://eips.ethereum.org/EIPS/eip-165) to the small list of oracles and external tokens involved in TokenSale.

Update: Acknowledged by the JPEG'd team and detailed response says *"We don't think that checking if the oracle implements the expected interface is necessary as the contract is going to be initialized with oracles that have been already checked for compatibility."*.

QSP-6 Centralization Of Power

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: TokenSale.sol

Description: In TokenVesting contract if the contract owner (ie DEFAULT_ADMIN_ROLE) loses their key then an attacker can revoke all vested tokens. In TokenSale, if the contract owner (ie. address validated by onlyOwner) loses its key before setSaleSchedule() is called, an attacker can totally disrupt the token sale with a very small schedule.

Recommendation: Consider limiting the power of the contract owner so they cannot revoke the vesting or adjust the token sale schedule. Otherwise, document these abilities of the contract owner or privileged user.

Update: Acknowledged by the JPEG'd team and detailed response says *"The situation described in QSP-6 cannot happen as the contracts will be owned by a multisig, effectively mitigating the risk of the admin keys being lost/falling in the wrong hands."*.

QSP-7 Ownership Can Be Renounced

Severity: Informational

Status: Acknowledged

File(s) affected: All Ownable Contracts

Description: The Ownable.sol has the function renounceOwnership(). Although it can only be called by the owner, such a function leaves the contract without an owner, which surely compromises any ability to manage the contract. Below we present the code of this dangerous function.

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(address(0));
}
```

Recommendation: Overwrite this function in order to have zero risk of ending the contract without an owner losing all funds invested in the contract.

Update: Acknowledged by the JPEG'd team.

QSP-8 Unnecessary Dynamic Array For supportedTokens

Severity: Informational

Status: Acknowledged

File(s) affected: TokenSale.sol

Description: The supported tokens are already known. Thus, there is no need to use a dynamic array for storing this information.

Recommendation: Initialize the array using the supported tokens in address[] internal supportedTokens.

Update: Acknowledged by the JPEG'd team.

QSP-9 Gas optimization in PreJPEG.release()

Severity: Informational

Status: Acknowledged

File(s) affected: PreJPEG.sol

Description: We can reduce the gas consumption by making one single call to token.balanceOf(...). In its current form, two calls are being made as presented below.

```
function release() public override {
    uint256 balanceBeforeRelease = token.balanceOf(address(this));
    super.release();
    _burn(
        msg.sender,
        balanceBeforeRelease - token.balanceOf(address(this))
    );
}
```

Recommendation: Make just one single call to token.balanceOf(...) and store it in a variable.

Update: Acknowledged by the JPEG'd team.

QSP-10 Reentrancy Guard Not Required

Severity: Informational

Status: Acknowledged

File(s) affected: TokenSale.sol

Description: The TokenSale contract inherits from ReentrancyGuard and uses the nonReentrant modifier on deposit and withdrawal-related functions. Since the only external calls in those functions are at the end, there is no dangerous reentrancy possible.

Recommendation: Remove the code related to ReentrancyGuard to save a bit on gas.

Update: Acknowledged by the JPEG'd team.

Automated Analyses

Slither

Slither has detected many results out of which the majority have been filtered out as false positives and the rest have been integrated into the findings from this report.

Adherence to Best Practices

- Consider using the `virtual` keyword only for functions that you expect to be overridden by a child contract (see most functions in `vesting/TokenVesting.sol`).
- Consider using an array with a fixed size for the `supportedTokens` in `sale/TokenSale.sol` since you already know the number of tokens that will be supported.
- Consider hardcoding the addresses to `WETH/USDC` tokens to minimize errors on deployment.
- Initialize local variables to make it clear what the initial value is for readers of the code (`sale/TokenSale.sol`, L176).
- `TokenVesting` has 10 explicit revert conditions and tests only cover 6 cases. Consider having least 10 test cases for negative conditions and then test cases for positive conditions.
- `TokenSale` has 24 explicit revert conditions and tests only cover 10 cases. Consider having least 24 test cases for negative conditions and then test cases for many positive conditions too.

Test Results

Test Suite Results

The audited contracts included 20 tests (TokenVesting, TokenSale, PreJPEG).

```
PreJPEG
   ✓ should mint PreJPEG tokens on new vesting (34ms)
   ✓ should burn all tokens on revoke (52ms)
   ✓ should burn tokens on release (58ms)
   ✓ should not allow transfers (37ms)

TokenSale
   ✓ should return the correct tokens when calling getSupportedTokens (3ms)
   ✓ should return the correct oracles when calling getTokenOracles (4ms)
   ✓ should return the correct oracle when calling getOracle (4ms)
   ✓ should allow the owner to allocate tokens (40ms)
   ✓ should allow the owner to set the sale schedule (60ms)
   ✓ should allow users to deposit (321ms)
   ✓ should allow the owner to finalize the raise (196ms)
   ✓ should allow the owner to enable withdrawals (91ms)
   ✓ should allow users to withdraw (236ms)
   ✓ should allow the owner to transfer the raise to treasury (235ms)

TokenVesting
   ✓ should allow members of the vesting_controller role to vest tokens (72ms)
   ✓ shouldn't allow to release vested tokens before vesting starts (32ms)
   ✓ should allow users to release (99ms)
   ✓ should not emit tokens during the cliff period (34ms)
   ✓ should allow to claim all the tokens after vesting is over (55ms)
   ✓ should allow the owner to revoke tokens (106ms)
```

Code Coverage

The code coverage is overall very good, statement coverage is above 99%, branch coverage is above 95%, function coverage is above 98% and line coverage is above 99% also.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
`sale/`	100	100	100	100	
`TokenSale.sol`	100	100	100	100	
`vesting/`	100	100	100	100	
`PreJPEG.sol`	100	100	100	100	
`TokenVesting.sol`	100	100	100	100	
All files	**100.0**	**100.0**	**100.0**	**100.0**	

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

```
f29adb1273d5049a0c1da7c76c720ec2e753a4613c6627cd3ec06037934d7406 ./audited-contracts/TokenSale.sol
da560de20aacdb6fd7a9a6c66f9a70bf1edb94eb86cefa9975b6660969e82966 ./audited-contracts/PreJPEG.sol
22fad40068bee6bec193f3d040f6bf7e393d98925c6bedc41afb38ce9038f30e ./audited-contracts/TokenVesting.sol
```

Tests

```
916af868c2225c4623f29c2c00cb7c1c1b98d805df998ae848a3d2fad2f9d136 ./audited-tests/PreJPEG.ts
76513e470ac41ac2ea3df2f03461f1c6d352bad0c059938ba4e54714d2f9e3b8 ./audited-tests/TokenVesting.ts
7596c95d0145674bc614fcdb507a83cb79124eb7e91f52772f1013ab12f1982c ./audited-tests/TokenSale.ts
```

Changelog

- 2021-12-14 - Initial report
- 2021-12-20 - Reaudit report

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.