Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED SECURITY CERTIFICATE

November 16th 2021 — Quantstamp Verified

Covalent Staking Contract

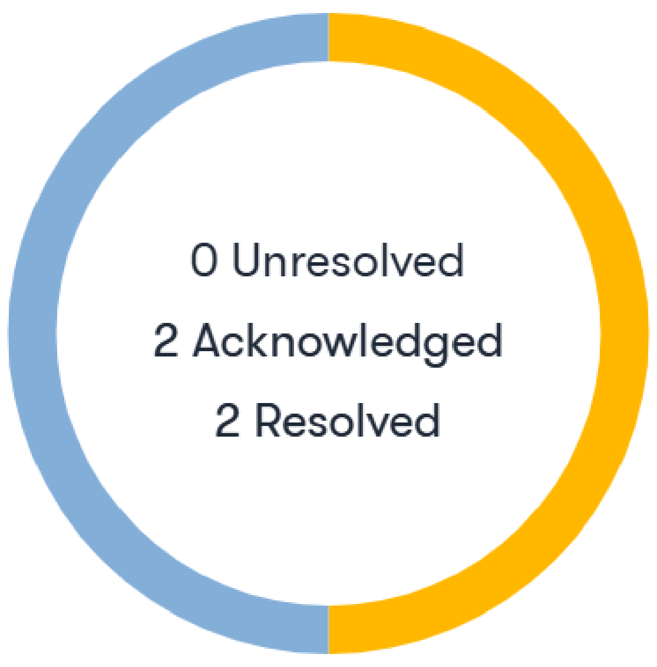
This audit report was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

Type	Staking Contract
Auditors	Ed Zulkoski, Senior Security Engineer Hisham Galal, Research Engineer Souhail Mssassi, Research Engineer
Timeline	2021-11-01 through 2021-11-16
EVM	London
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	README.md
Documentation Quality	High
Test Quality	High
Source Code	

Repository	Commit
staking-contract-covalent	5647dc0 (initial report)
staking-contract-covalent	29c8577 (final report)

Total Issues	4 (2 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	0 (0 Resolved)
Low Risk Issues	1 (1 Resolved)
Informational Risk Issues	3 (1 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).
○ 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 audit pertains to the Covalent `DelegatedStaking` contract. The code is well documented and tested. No High or Medium severity issues were found during the audit. Four Low to Informational level issues were noted.
Update: All issues have been resolved or acknowledged as of commit 29c8577.

ID	Description	Severity	Status
QSP-1	Ownership Can Be Renounced	⌄ Low	Fixed
QSP-2	Gas Usage / `for` Loop Concerns	○ Informational	Acknowledged
QSP-3	Unlocked Pragma	○ Informational	Fixed
QSP-4	Missing input validation	○ 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 v0.8.1

Steps taken to run the tools:

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

Findings

QSP-1 Ownership Can Be Renounced

Severity: *Low Risk*

Status: Fixed

File(s) affected: `DelegatedStaking.sol`

Description: All files that inherit from the `Ownable` contract could be left with no owner if the latter renounces his ownership. As one potential consequence, that could prevents update to the deployed contracts, requiring re-deploys and potential downtime.

Recommendation: Unless ownership renouncing is indeed an expected behaviour, override the renounceOwnership such that it always reverts.

QSP-2 Gas Usage / `for` Loop Concerns

Severity: *Informational*

Status: Acknowledged

File(s) affected: `DelegatedStaking.sol`

Description: Gas usage is a main concern for smart contract developers and users, since high gas costs may prevent users from wanting to use the smart contract. Even worse, some gas usage issues may prevent the contract from providing services entirely. For example, if a `for` loop requires too much gas to exit, then it may prevent the contract from functioning correctly entirely. It is best to break such loops into individual functions as possible.
The functions `getValidatorsDetails` and `getDelegatorDetails` iterate over all validators. Even though it is a view function, it is not guaranteed that nodes will return correctly if the array is too long.

Recommendation: Ensure that loop bounds will be reasonably low in practice.

Update: From the Covalent team -- "There won't be that many validators, currently there is only 10."

QSP-3 Unlocked Pragma

Severity: *Informational*

Status: Fixed

File(s) affected: `DelegatedStaking.sol`

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.4.*`. 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, it is recommended to remove the caret to lock the file onto a specific Solidity version.

QSP-4 Missing input validation

Severity: *Informational*

Status: Acknowledged

File(s) affected: `DelegatedStaking.sol`

Description: In the `setAllocatedTokensPerEpoch` function the Owner can modify the value of `allocatedTokensPerEpoch` after comparaing the value `amount` with the value 0. The problem here is that the Owner can inject a value greater than `[(2^128-1)/2]+1`, thus the `depositRewardTokens` and `takeOutRewardTokens` functions will fail because there is no uint128 that can verify the condition `amount % allocatedTokensPerEpoch == 0`.

Recommendation: Verify the `amount` in the `setAllocatedTokensPerEpoch` function.

Update: From the Covalent team -- "When endEpoch is 0, `allocatedTokensPerEpoch` could be set to `[(2^128-1)/2]+1`. However, it will be impossible to make `endEpoch > 0` from depositing tokens since the deposit amount must be greater than allocatedTokensPerEpoch. This is impossible to do since `[(2^128-1)/2]+1` is much greater than the total supply of tokens. When `endEpoch != 0`, `futureReward` will always be smaller than `[(2^128-1)/2]+1` thus `addEpochs` will be 0. This will result in revert."

Automated Analyses

Slither

1. Slither warns of potential reentrancy issues in `_redeemRewards`, `_stake`, `recoverUnstaking`, and `redelegateUnstaked`, and `transferUnstakedOut`. However, the external call is to `CQT` which should be a trusted contract. Nonetheless, we recommend strictly adhering to the checks-effects-interactions pattern where possible.
2. Slither flagged several potentially dangerous strict equalities to zero (`commissionLeftOver == 0`), however these expressions implemented the intended semantics of the functions.

Test Results

Test Suite Results

```
All together
     ✓ Should redeem, stake, unstake and recover correct # of tokens. (47907ms)

  Disabled
     ✓ Should redeem, stake, unstake, recover and redelegate correct # of tokens with disabled validators. (30754ms)

  Ownership
     ✓ Should return owner address same as signer. (253ms)
     ✓ Should not access depositRewards, takeOutRewardTokens, addValidator, disableValidator, setAllocatedTokensPerEpoch by not owner. (345ms)
     ✓ Should access depositRewardTokens, takeOutRewardTokens, addValidator, disabledValidator by owner. (511ms)
     ✓ Should not access transfer, updateExchangeRate, updateValidator, sharesToTokens, tokensToShares. (213ms)

  Add Validator
     ✓ Should change validators number. (236ms)
     ✓ Should emit event  with correct validator and operator addresses and commission rate. (241ms)
     ✓ Should add validator with correct commission rate. (235ms)

  Deposit reward Tokens
```

```
    ✓ Should change balance of the contract and the owner. (3536ms)
    ✓ Should change endEpoch. (699ms)
    ✓ Should revert with wrong inputs. (373ms)
    ✓ Should not change allocated tokens per epoch. (564ms)
    ✓ Should not change max cap multiplier. (563ms)

  Disable validator
    ✓ Should not be able to call stake after validator got disabled by the owner. (2905ms)
    ✓ Should revert when non validator or non owner calls. (415ms)
    ✓ Should revert when disabling validator twice (1062ms)

  Recover Unstaking
    ✓ Should revert when recover greater than staking (665ms)
    ✓ Should emit event when recovered unstake successfully (3837ms)

  Redeem Rewards
    ✓ Should emit redeem reward event with correct number of rewards (999ms)
    ✓ Should return number of rewards earned by validator with delegators (6995ms)

  Transfer Unstaked
    ✓ Should transfer out after cool down ends, delegator (1222ms)
    ✓ Should redelegate partially (1257ms)
    ✓ Should revert when redelegating with enabled validator (960ms)
    ✓ Should revert when validators attempt to redelegate (989ms)
    ✓ Should revert when redelegate greater than unstaked (1124ms)

  Set allocated tokens per epoch
    ✓ Should change delegation available. (1262ms)
    ✓ Should change end epoch. (655ms)
    ✓ Should revert if set to 0. (673ms)

  Set max cap multiplier
    ✓ Should change delegation available. (771ms)
    ✓ Should change max cap multiplier. (233ms)
    ✓ Should be able to stake more if multiplier increases. (826ms)
    ✓ Should revert if set to 0. (1270ms)

  Set validator commission rate
    ✓ Should change validator commission rate. (697ms)
    ✓ Should revert if set to >= 10^18. (754ms)

  Staking
    ✓ Should revert when transfer not approved (494ms)
    ✓ Should revert if exceeds max cap (1023ms)
    ✓ Should succeed when stakes filling max cap (2071ms)
    ✓ Should revert when stake by validator is less than minimum stake required (2507ms)
    ✓ Should revert when token is less than 1 (665ms)
    ✓ Should stake 1 token (941ms)
    ✓ Should change max cap when staked (2125ms)
    ✓ Should return correct delegated #  (1625ms)

  Take out reward Tokens
    ✓ Should change balance of the contract and the owner. (623ms)
    ✓ Should take out correct # of rewards. (507ms)
    ✓ Should change endEpoch. (756ms)
    ✓ Should revert with wrong inputs. (408ms)
    ✓ Should not change allocated tokens per epoch. (750ms)
    ✓ Should not change max cap multiplier. (930ms)

  Unstaking
    ✓ Should revert when unstake is more than staked (817ms)
    ✓ Should revert when unstake is too small (934ms)
    ✓ Should revert when unstake by validator is below max cap (1044ms)
    ✓ Should revert when unstake by validator turns stake into less than minimum staked required (674ms)
    ✓ Should emit event when unstaked successfully (668ms)


  54 passing (2m)
```

Code Coverage

The overall coverage is quite good, however we still recommend adding tests to improve branch and statement coverage such that they are closer to 100%. Many else-branches have not been covered. The functions `transferUnstakedOut` and `setAllocatedTokensPerEpoch` have not been fully covered.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	91.59	76.56	92.86	91.9	
DelegatedStaking.sol	91.59	76.56	92.86	91.9	… 484,485,486
All files	**91.59**	**76.56**	**92.86**	**91.9**	

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

```
e08d8024d27e3a04370cd7e7258108acdc9dd820939ed8a164a0e25dcabbac0c ./contracts/DelegatedStaking.sol
```

Tests

```
00335612fc86f413b04d733c0e8ff904d8e6c4aa3a3597d4cdea6efd30358173 ./test/fixtures.js
6ec64737adbeb0883d8aa0f8c5b5fa81efd70587f4974da909c91305ca57474c ./test/unit-tests/access.js
05e91ffc3546f1612e4f879d6338dba6f45a99ec44654375bacf420a3a3c886e ./test/unit-tests/addValidator.js
7c3d18aa1c420c746137c499829cc29352edf047d582fd2111c431f5c17c7e41 ./test/unit-tests/depositRewardTokens.js
efaea3514ace99f62156d0377945ffb57bc0e4031e75d9805e79ee776f27eca6 ./test/unit-tests/disableValidator.js
523ef14bf46838a1dbf89ad8f555ab365d83b71ff267681b52cd5582fd89eed0 ./test/unit-tests/recoverUnstaking.js
8db08f8728858047ef29a39a5765638ba9c0cb4dec4d7d12bda9b2c87d551d56 ./test/unit-tests/redeemAllRewards.js
de255b9b76f4d297877661ab5b87afe9a65a1ffed2deb5875b1ba93c17a41f0a ./test/unit-tests/redelegateUnstaked.js
6c869837c864ea9e3254cb66242bf11e871a503cfc86b7c5faa8fe2f8802b530 ./test/unit-tests/setAllocatedTokensPerEpoch.js
385eb761da4e8749adacf7fb489a92cfbe1737a6f98ecec8c325baa87468ae9b ./test/unit-tests/setMaxCapMultiplier.js
e22ed209afe3dfe1fd9b412d60f53f39da966c97398010d64ca9d3f374718b94 ./test/unit-tests/setValidatorCommissionRate.js
35a6c3abf13fb422e38feaa0077511b967e2c26d969a085fcad81a254a8dd98b ./test/unit-tests/stake.js
2e5e22a1df2b1507b50844973a66093d96ee5074e9baa44753f60d911f955c3e ./test/unit-tests/takeOutRewardTokens.js
5c105dcd5e384dc4197e6e142ed3a0bdc7d950511f39e9fcea22a50bc909a47d ./test/unit-tests/unstake.js
0bb6175953574e22b22900afa1e77729876b753295a4749f9e4c92d2e50c7471 ./test/integration-tests/all.js
084b2e129dace49e6d1bba5009cba61ea50a43cefd48e563e5a7303efc257bfc ./test/integration-tests/RewardsCalculator.js
0ae586e36fc9137450b929ebe1c103054f62842e3c1c8d95c09cc1b6c0c3a972 ./test/integration-tests/withDisabledValidator.js
```

Changelog

- 2021-11-09 - Initial report
- 2021-11-15 - Updated report based on commit 29c8577

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.