Quantstamp Security Assessment Certificate

October 20th 2020 — Quantstamp Verified

Curve Finance (diff)

This security assessment was prepared by Quantstamp, the leader in blockchain security

Executive Summary

Type	Liquidity Pool
Auditors	Shunsuke Tokoshima, Software Engineer Joseph Xu, Technical R&D Advisor Kevin Feng, Blockchain Researcher
Timeline	2020-09-29 through 2020-10-15
EVM	Muir Glacier
Languages	Vyper
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Low
Test Quality	High

Source Code

Repository	Commit
curve-contract	5395c5a

Total Issues	4 (1 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	1 (1 Resolved)
Low Risk Issues	2 (0 Resolved)
Informational Risk Issues	1 (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).
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 contains the results of our assessment of the two smart contracts under `contracts/pool-templates/meta`.

The contracts are overall well-written and well-tested. We identified **4 potential issues** of various severity levels: one medium, three low, and one informational severity. In addition, we provided ideas for further code and documentation improvements.

ID	Description	Severity	Status
QSP-1	Hard-coded Array Index Allows for Only One Metapool Token	Medium	Fixed
QSP-2	Block Timestamp Manipulation	Low	Acknowledged
QSP-3	Insufficient Input Validation	Low	Acknowledged
QSP-4	Implicit Assumption that Only One Base Pool Is Allowed per Metapool	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:

- SmartCheck Released v2.0

Steps taken to run the tools:

1. Install SmartCheck globally To install SmartCheck globally to your system run (administrative rights required)

```
npm install @smartdec/smartcheck -g
```

3. (Optional) Add SmartCheck as development dependency To add and install SmartCheck as development dependency to your npm project run:

```
npm install --save-dev @smartdec/smartcheck
```

5. Start the analysis To start analysis simply run:

```
smartcheck -p .
```

Findings

QSP-1 Hard-coded Array Index Allows for Only One Metapool Token

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `DepositTemplateMeta.vy`

Description: In `DepositTemplateMeta.vy` L158, the index for the base pool's LP token is hard-coded in function `remove_liquidity()`as `self.coins[1]`. This can be a problem if the metapool has multiple tokens + base pool LP token.

Recommendation: It is recommended to change `self.coins[1]` to `self.coins[MAX_COINS]` (assuming that there is only one base pool allowed per metapool and indexed at `MAX_COINS`).

Update: Addressed as of the commit `8ebe6e6`.

QSP-2 Block Timestamp Manipulation

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `SwapTemplateMeta.vy`

Description: Projects may rely on block timestamps for various purposes. However, it's important to realize that miners individually set the timestamp of a block, and attackers may be able to manipulate timestamps for their own purposes. If a smart contract relies on a timestamp, it must take this into account. In `_A()` of `SwapTemplateMeta.vy`, the returned value is dependent on `block.timestamp`. This implementation entails some risk that `Amplification coefficient` is manipulated by a malicious agent. For example, miners who happen to want to swap some tokens could interfere the development of `Amplification coefficient` as a means to get a trading advantage.

Recommendation: It would be beneficial to consider using `block.number` instead of `block.timestamp` which is more reliable.

Update: Acknowledged. The Curve team is currently on mitigating this risk by making each updating step of `Amplification coefficient` smaller. Besides, updated `Amplification coefficients` are limited to be 1/10~10× of the previous ones and this feature works as a guardrail for this risk.

QSP-3 Insufficient Input Validation

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `SwapTemplateMeta.vy, DepositTemplateMeta.vy`

Description:

1. `_A` , `_fee` and `_admin_fee` in **init**() of `SwapTemplateMeta.vy` are not validated. Values bigger than `MAX_A`, `MAX_ADMIN_FEE` or `MAX_FEE` can be set here.
2. `_owner`, `_pool_token`, `_base_pool` in `__init__()` of `SwapTemplateMeta.vy` are not validated. Providing incorrect values may result in a later discovery that the smart contract is not working as intended.
3. `_token` in `__init__()` of `DepositTemplateMeta.vy` is not validated. Providing incorrect values may result in a later discovery that the smart contract is not working as intended.

Recommendation:

1. Adding checks that ensure that `_fee` is less than or equal to `MAX_FEE` and `_admin_fee` is less than or equal to `MAX_ADMIN_FEE`.
2. Adding checks to ensure that all addresses are different from `0x0`.
3. Adding a check to ensure that the address is different from `0x0`.

Update: Deployment of metapool contracts in Curve is only allowed for Curve admins. Therefore, the risk has been considered as negligible and the Curve team has decided not to set the validations in `__init__()`.

QSP-4 Implicit Assumption that Only One Base Pool Is Allowed per Metapool

Severity: *Informational*

Status: Acknowledged

Description: The code implicitly allows only one base pool per metapool due to the implementation of `__init__()` and `MAX_COINS = N_COINS - 1`. It would not be possible to set up a metapool with two base pools with non-overlapping assets [GUSD, [3Pool], [sBTC]].

Recommendation: There is no immediate issue from this assumption but it would be good to clarify in developer documentation and external documentation (e.g., https://resources.curve.fi/faq/base-and-metapools).

Automated Analyses

SmartCheck

39 warnings (2 `VYPER_PRIVATE_MODIFIER_DONT_HIDE_DATA` in `DepositTemplateMeta.vy` and 37 `VYPER_PRIVATE_MODIFIER_DONT_HIDE_DATA` in `SwapTemplateMeta.vy`) were detected. After checking, these findings are considered as false positives.

Code Documentation

- **Update: Fixed as of the commit `3f3b74b`.** README.md: “pip install -r requirements” should read “pip install -r requirements.txt”
- **Update: Fixed as of the commit `8ebe6e6`.** DepositTemplateMeta.vy L34: "shich" should read "which"

• **Update: Some documentation(Natspec) has been added as of the commit `8ebe6e6`. Some functions in `SwapTemplateMeta.vy` are to be documented and the stableswap paper is to be pointed to in the code.** Lack of comments within the code. It would be helpful to either point to the stableswap paper at the top or to label individual calculations such as calculations of invariants etc.

Adherence to Best Practices

• Lines 92-132 in `SwapTemplateMeta.vy` include variables whose values need to be updated at compile time. There are a few variables such as `FEE_ASSET` and `BASE_POOL_COINS` that have hard-coded values but might need to be updated at compile time. It may help to check the spec and group all of the variables that need to be updated together into one block (or use commenting at each line to clearly indicate which variables need to be updated at compile time).

• In `SwapTemplateMeta.vy`, the arguments `i` and `j` for `get_dy_underlying()` and `exchange_underlying()` are best used only for calculating `base_i`, `base_j`, `meta_i`, and `meta_j` for clarity.

• Function and naming schemes can be improved for readability purposes. For example, functions names such as `get_D_mem`, `_vp_rate_ro`, `_xp_mem`, `get_D` can have more descriptive names.

• It is recommended to perform a gas analysis for potentially gas-consuming functions such as `add_liquidity()` in `DepositTemplateMeta.vy` and clarify assumptions in the documentation if any are found in the analysis.

Test Results

Test Suite Results

`pytest tests --pool template-meta` was run to test the metapool template contracts.
To summarize, 545 tests passed as of commit `5395c5a`.

Update1: As of the commit `8ebe6e6`, `pytest tests --pool template-meta` fails because the newly added `tests/pools/common/unitary/test_rate_caching.py`'s filename conflicts with `tests/pools/snow/test_rate_caching.py`'s one. The simple solution for the issue is renaming these filenames so that they are different.

Update2: As of the commit `3f3b74b`, The issue described in `Update1` has been addressed and all the tests pass. The test result as of commit `3f3b74b`is shown below.

```
============================================================================== test session starts ==============================================================================
platform darwin -- Python 3.8.5, pytest-6.0.1, py-1.9.0, pluggy-0.13.1
rootdir: /Users/tokoshimashuntasuku/Desktop/GO/QS/audits/curve/curve-contract
plugins: eth-brownie-1.11.9, xdist-1.34.0, web3-5.11.1, hypothesis-5.35.0, forked-1.3.0
collecting ...
Launching 'ganache-cli --port 8545 --gasLimit 12000000 --accounts 10 --hardfork istanbul --mnemonic brownie'...
collected 594 items

tests/test_gas.py ..                                                                                                                                                    [  0%]
tests/pools/common/integration/test_virtual_price_increases.py .                                                                                                        [  0%]
tests/pools/common/unitary/test_add_liquidity.py ........                                                                                                               [  1%]
tests/pools/common/unitary/test_add_liquidity_initial.py ....                                                                                                           [  2%]
tests/pools/common/unitary/test_claim_fees.py ......                                                                                                                    [  3%]
tests/pools/common/unitary/test_exchange.py ......................                                                                                                      [  7%]
tests/pools/common/unitary/test_exchange_reverts.py ..............                                                                                                      [  9%]
tests/pools/common/unitary/test_exchange_underlying.py ..................................................................................................................... [ 32%]
.....................................                                                                                                                                   [ 39%]
tests/pools/common/unitary/test_exchange_underlying_reverts.py ..................................                                                                       [ 45%]
tests/pools/common/unitary/test_get_virtual_price.py ....................                                                                                               [ 49%]
tests/pools/common/unitary/test_kill.py ...........                                                                                                                     [ 51%]
tests/pools/common/unitary/test_modify_fees.py ........................                                                                                                 [ 55%]
tests/pools/common/unitary/test_ramp_A_precise.py ........                                                                                                              [ 56%]
tests/pools/common/unitary/test_rate_caching.py ...................                                                                                                     [ 59%]
tests/pools/common/unitary/test_remove_liquidity.py ......                                                                                                              [ 60%]
tests/pools/common/unitary/test_remove_liquidity_imbalance.py ............                                                                                              [ 62%]
tests/pools/common/unitary/test_remove_liquidity_one_coin.py ........................                                                                                   [ 66%]
tests/pools/common/unitary/test_transfer_ownership.py ..........                                                                                                        [ 68%]
tests/pools/snow/test_coin_rates.py ....                                                                                                                                [ 69%]
tests/pools/snow/test_rate_caching_snow.py ......                                                                                                                       [ 70%]
tests/zaps/common/test_add_liquidity_initial_zap.py ...                                                                                                                 [ 70%]
tests/zaps/common/test_add_liquidity_zap.py ..............                                                                                                              [ 72%]
tests/zaps/common/test_remove_liquidity_imbalance_zap.py ............................                                                                                   [ 77%]
tests/zaps/common/test_remove_liquidity_one_coin_zap.py ........                                                                                                        [ 79%]
tests/zaps/common/test_remove_liquidity_zap.py ............                                                                                                             [ 81%]
tests/zaps/common/test_return_values.py ...                                                                                                                             [ 81%]
tests/token/test_approve.py ........................                                                                                                                    [ 86%]
tests/token/test_mint_burn.py ..........................                                                                                                                [ 90%]
tests/token/test_transfer.py ................                                                                                                                           [ 93%]
tests/token/test_transferFrom.py ..............................                                                                                                         [ 99%]
tests/token/test_version2.py ...                                                                                                                                        [100%]

================================================================================ warnings summary ================================================================================
tests/test_gas.py: 1 warning
tests/pools/common/integration/test_virtual_price_increases.py: 1 warning
tests/pools/common/unitary/test_exchange_underlying.py: 180 warnings
  /usr/local/var/pyenv/versions/3.8.5/lib/python3.8/site-packages/brownie/network/event.py:231: UserWarning: Malformed data field in event log
    warnings.warn(str(exc))

-- Docs: https://docs.pytest.org/en/stable/warnings.html
================================================================== 594 passed, 182 warnings in 691.97s (0:11:31) ==================================================================
Terminating local RPC client...
```

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

```
e3e252d2454d6b6395453e80d75938657fa1d254d08c3a3fc8a55c1360a22153 ./meta/DepositTemplateMeta.vy

4da0398d50012e51b65f1200c3258287adee874cd6933b4d9fa125446c39c281 ./meta/SwapTemplateMeta.vy
```

Tests

```
b1e927f8f52af05377267c1cabdd1fb9c7b83d529535b90eb3d854e381e56b55 ./tests/simulation.py
c6ea8d28d3c0fbc6fc5416b0f427ec6b8dc763d32d5c00702765f823d440f457 ./tests/conftest.py
0f2558bd33acea142e22c9fcfa213c15ce5e85f4c1cde3e5c14d640c55ecb904 ./tests/test_gas.py
83fcf556a70677f4354bd8016d47bb729f9b4efb5e42df76399cfcfac4e9beea ./tests/fixtures/functions.py
730109e2ec3b4bbd6a5f8d14aca276dcea02bd46995574619f284aaf8709d4c2 ./tests/fixtures/accounts.py
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 ./tests/fixtures/__init__.py
b6bbd72b350deb29089bfe85b128043d548c8eee648f07ae9c56619e4e2e6408 ./tests/fixtures/setup.py
43f02b0305c5b68f135a3cd7c7b9bf4d82e9f63da6fcc2684711c42f65546ddb ./tests/fixtures/deployments.py
a8ebcb03fe6481e8cefef8bbb71a96e2c4be101c0c84a336dfa0dbaf47bf62fb ./tests/fixtures/coins.py
a5ef901e5424f5e7bfbd841c1efcf5290db236b86afa294a625e54ee5d7cd1b9 ./tests/fixtures/pooldata.py
b84505e05172c766e8c0358d629fad7baf55cac81e18f9f11845d16e3a873a62 ./tests/zaps/common/test_add_liquidity_initial_zap.py
d640d232630d5ecbf9ae0393ac9148b47969b34842766678255fca84a83c6416 ./tests/zaps/common/test_remove_liquidity_one_coin_zap.py
3b1c86bbde71517b4de7902fe5bd6430a7b5fedaf52a406ac2bd9beb55e3fc37 ./tests/zaps/common/test_return_values.py
c6ad6b54659d1fece20da25cce3a514c45a74ba79e827939a0dc175c7daa5c2b ./tests/zaps/common/test_remove_liquidity_imbalance_zap.py
0c85ac0c643dbd9a2883958cd3259d2477dd814e978638ee8e03aca0db0a54e4 ./tests/zaps/common/test_remove_liquidity_zap.py
e1d94e44cbd12555665f976e480d84d1cbddedc0d3163f48e7e8ab469bc2cfca ./tests/zaps/common/test_add_liquidity_zap.py
04f22fdb40e888f5d41ac2208b4e803e91d1e9f443c4044ad20b68491d118016 ./tests/pools/snow/test_coin_rates.py
e9d8c2c246ced6568f556c244e22660c3e95f8e5cfc33b5572b9a65aa4eb62e6 ./tests/pools/snow/test_rate_caching_snow.py
0ced4784c3eba11952e1be381fbb3c75f39ffcf75d6c2bef02751c10327a4d70 ./tests/pools/common/integration/test_curve.py
d9ad7e417e61e088f334edd504b8f176a5c5b6758715bb27e9196ff79e927e37 ./tests/pools/common/integration/test_registry.py
30641a82755f935c780afb496af5197df735fcba6914e6dd24f14132eecb2efc ./tests/pools/common/integration/test_virtual_price_increases.py
5bc938ad42b42d3ed1ef331a507c30b32312d83d70e21e484f8727ed8ee2b987 ./tests/pools/common/integration/test_simulate_exchange.py
0750b1718c0d9d2c219c0eff4bebb2b10dcbf7b1061a6ca9a87a124352eb6c5f ./tests/pools/common/unitary/test_claim_fees.py
9bbf764629c4fac0e6711dd26a7233107b743752f1bc2d6cb357331b61bb6082 ./tests/pools/common/unitary/test_add_liquidity_initial.py
fd184c1e104b6083bedfdc6c38769b83f61a70ed92a746edd289782f1ee53f93 ./tests/pools/common/unitary/test_exchange.py
98c5679a6b766e00a7d3ab46a931245b65141d6555a5e3178cd2291b2711c54d ./tests/pools/common/unitary/test_modify_parameters.py
d673746360757befc963a625add9fecffbd406eb9506565bbc9a7e1baeec2114 ./tests/pools/common/unitary/test_modify_fees.py
cd309242a53e7430509d3e36a608a94cad27d04b44d7f447a1d6ff9b75de9e70 ./tests/pools/common/unitary/test_remove_liquidity.py
a9d83c6b1de1775d0348d4a7a2d69c63106fc4da067696173467f7804270e8c7 ./tests/pools/common/unitary/test_exchange_underlying_reverts.py
13eb0b489388e8222ab9aa232c05218682a26bcf6d40ad51849c78d11a2b69f9 ./tests/pools/common/unitary/test_exchange_reverts.py
8e2ff1c48e08fd83954ee1df7d67a335fb04de9798fc1e583a55f9ad927c8cfc ./tests/pools/common/unitary/test_exchange_underlying.py
66cd03cf49c56d1433e83e414301024d331404524e5bd9fbdce76e6f884f76ac ./tests/pools/common/unitary/test_transfer_ownership.py
0f61db953fa3f9af4864b7ff774a1729f4cbdaa662c3dceb4e8e62b4d9317971 ./tests/pools/common/unitary/test_get_virtual_price.py
5744a9a8080229c48d8d65dd40bb0adddd7c8113dbb223eae2a4ea3a011ebc12 ./tests/pools/common/unitary/test_ramp_A.py
f8e6567518e33a2dc4e7c65c3da0f841c7f81c5446c40a7842469126051ed846 ./tests/pools/common/unitary/test_remove_liquidity_imbalance.py
fcf28d6d0e1f5ae20f757db791cfee671f39587b5e2518ccfe1d521b3ed070cc ./tests/pools/common/unitary/test_add_liquidity.py
1e3b0738a17019df4025836dab3770ec77606045636bf3ac8cb77c693fb200a9 ./tests/pools/common/unitary/test_remove_liquidity_one_coin.py
9a67d4161424bf2dc45508760cdb0dcb7258cd31e52b0e16ef58ed7ad222b304 ./tests/pools/common/unitary/test_kill.py
90030c78fc2e06cbb57c3843da7dcb1e3ee72eb43ae560e656bb2df6e2c677e6 ./tests/pools/common/unitary/test_rate_caching.py
53bb204cdf197459fbf412a681073542a6bcdc056a2016dab84766a68d0c5b2f ./tests/pools/common/unitary/test_ramp_A_precise.py
38a780a9b5650eae73dd59199715a7bcf912fa85801796e12fc3e08bcf50d793 ./tests/token/conftest.py
cfb89b566029e3d04f3b7583e4f8a8ba4565218e7a519b46cb9fbfb72ce98caa ./tests/token/test_approve.py
6b8f406eca2484134429d57ba44fe7488b6e20e7590ecd7505e6389803365cb3 ./tests/token/test_mint_burn.py
562e0be062407cafbdfe8aa204d4abe0d143e5da2f36f8f7fa9735d64e92bca5 ./tests/token/test_transfer.py
a9f5b921a782cb92586d9979166f171eec0ac03f8e0eac49259b016b57aa8a63 ./tests/token/test_version2.py
b82da22a4228bc27d3c10f1d42eb7d0094df6ef196bdb8c1da14d5ca90019a8e ./tests/token/test_transferFrom.py
```

Changelog

- 2020-10-05 - Initial report
- 2020-10-13 - Updated based on commit `8ebe6e6`
- 2020-10-15 - Updated based on commit `3f3b74b`

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.