March 17th 2020 — Quantstamp Verified

Stake Technologies Lockdrop

This smart contract audit was prepared by Quantstamp, the protocol for securing smart contracts.

Executive Summary

Type	Audit
Auditors	Martin Derka, Senior Research Engineer Kacper Bąk, Senior Research Engineer Ed Zulkoski, Senior Security Engineer
Timeline	2020-01-20 through 2020-02-11
EVM	Byzantium
Languages	Solidity, Javascript
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Source Code	Repository: ethereum-lockdrop — Commit: e6d8357
Changelog	• 2020-02-05 - Initial report • 2020-02-11 - Updated report based on 8188da93 and fe5255a
Overall Assessment	The audited repository contains smart contracts and a user-facing web application. Only the smart contracts were a subject to the audit. No documentation was provided to Quantstamp, however, it is clear that the smart contracts aim to implement factory of time-lock smart contracts for locking Ether. The implementation is minimalistic and easy to understand. Quantstamp identified one severe DoS vulnerability in the project.

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

^ High Risk	The issue puts a large number of users' sensitive information at risk, 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 risk is small or impact negligible consequences in practice (e.g., gas analysis, deployment settings).
✓ Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.

Summary of Findings

ID	Description	Severity	Status
QSP-1	Denial-of-Service (DoS)	^ High	Resolved
QSP-2	Unlocked Pragma	* Informational	Resolved

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):
Denial of Service, unsuccessful transfer of Ether, infinite lock of funds.
Toolset

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

Setup

Tool Setup:

- Truffle
- Ganache
- SolidityCoverage
- Mythril
- Truffle-Flattener
- Slither

Steps taken to run the tools:

1. Installed Truffle: `npm install -g truffle`
2. Installed Ganache: `npm install -g ganache-cli`
3. Installed the solidity-coverage tool (within the project's root directory): `npm install --save-dev solidity-coverage`
4. Ran the coverage tool from the project's root directory: `./node_modules/.bin/solidity-coverage`
5. Flattened the source code using `truffle-flattener` to accommodate the auditing tools.
6. Installed the Mythril tool from Pip: `pip3 install mythril`
7. Ran the Mythril tool on each contract: `myth -x path/to/contract`
8. Installed the Slither tool: `pip install slither-analyzer`
9. Run Slither from the project directory: `slither .`

Assessment

Findings

QSP-1 Denial-of-Service (DoS)

Severity: High Risk

Status: Resolved

File(s) affected: `Lockdrop.sol`

Description: A Denial-of-Service (DoS) attack is a situation which an attacker renders a smart contract unusable. The factory contract `Lockdrop.sol` construct a new instance of `Lock.sol` with every call to the `lock()` method. This instance is deployed to an address that can be deterministically pre-computed off-chain. On line 45, after transferring the Ether to be locked, the `Lockdrop.sol` asserts that the balance of the deployed `Lock.sol` is exactly `msg.value`. If it differs, the transaction gets reverted. If the address of the deployed `Lock.sol` has pre-existing balance, is not necessarily the case that equality is reached.

Exploit Scenario: As the addresses of the deployed `Lock.sol` instances can be pre-computed, an attacked can send Ether to the address of the next lock. The check on line 45 will then always fail and `Lockdrop.sol` will be unable to create new locks.

Recommendation: Quantstamp recommends removing the assertion on line 45 of `Lockdrop.sol`. Alternatively, the Stake Technologies team can replace it with `assert(address(lockAddr).balance >= eth);` or change the design to two-step transfer: construction followed by `address(lockAddr).call.value(msg.value)()` and assert the success of the transfer.

QSP-2 Unlocked Pragma

Severity: Informational

Status: Resolved

File(s) affected: `Lock.sol`, `Lockdrop.sol`

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.5.*`. 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." 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.

Recommendation: Quantstamp recommends locking pragma at the latest version of Solidity.

Automated Analyses

Mythril

Mythril reported control flow decisions based on timestamps. This issues is benign in the context of the contracts. It also reported a call to user-supplied address from `Lock.sol`. This issue is also benign as it cannot be exploited by anyone other than the user who locked their funds. It reports the option of anyone causing drain of Ether in `Lock.sol` after the timelock expires, which appears correct in the context of the contract (note that Ether is always sent to the user who locked it, regardless of who initiates the drain). An integer overflow in the constructor of `Lockdrop.sol` is reported as well, however, this is a benign issue that cannot be exploited after the contract is deployed. Mythril warns that `tx.origin` is used in `lock()`, which means that only externally owned accounts can lock Ether. As per the in-code comment, this is desired. It also warns against potentially failing assertion reported in QSP-1.

Slither

Slither reported potential lock of Ether in `Lock.sol` without draining function. The finding is false positive; draining function exists. It also reported unlocked pragma and the strict equality test reported by the auditors (see QSP-1 and QSP-2).

Adherence to Specification

No specification was provided for the purposes of the audit.

Code Documentation

The code is reasonable commented.

Adherence to Best Practices

The code respects best practices, with the exception of the vulnerabilities listed in this report.

Test Results

Test Suite Results

Tests are present. The test cases are adequately chosed. All tests pass.

```
  Contract: Lock
    Locking funds
      ✓ Locking funds on contract creation (425ms)
      ✓ Unlocking funds when time reached (371ms)

  Contract: Lockdrop
    Smart contract interaction
      ✓ Locking funds and emit event (593ms)
      ✓ Reject transaction without funds (4646ms)
      ✓ Reject transaction with wrong duration (382ms)
    Event collecting
      ✓ Collect Locked events (185ms)


  6 passing (7s)
```

Code Coverage

The test coverage appears good, however, it misses the branch where the equality reported in QSP-1 evaluates to false. Quantstamp recommends adding a test for it.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	100	66.67	100	100	
Lock.sol	100	100	100	100	
Lockdrop.sol	100	66.67	100	100	
All files	**100**	**66.67**	**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

```
3be5cd4922791d961ee2672d84d873f8de26cb8278d9d27fbf4337d9f0b025ba47 ./contracts/Lock.sol
1c4e30fd3aa765cb0ee219a298dead71c1c99888ddcc71575c25df3405802cf5b09 ./contracts/Migrations.sol
bb1128dc9459515074e65c9bb6c1c1db65af8895566d5e1b51a2dbfe4e3019bb4e ./contracts/Lockdrop.sol
```

Tests

```
25d255d1856d702945450925115959c9a9c816aa1cf14cbeb6bbc21e10d57fec89 ./test/1_Lock_test.js
aa25ff2bcbb8826d59565d8f5f8885f16a1470b55610fce5720a2970420c5c71 ./test/2_Lockdrop_test.js
```

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure smart contracts at scale using computer-aided reasoning tools, with a mission to help boost adoption of this exponentially growing technology.

Quantstamp's team boasts decades of combined experience in formal verification, static analysis, and software verification. Collectively, our individuals have over 500 Google scholar citations and numerous published papers. In its mission to proliferate development and adoption of blockchain applications, Quantstamp is also developing a new protocol for smart contract verification to help smart contract developers and projects worldwide to perform cost-effective smart contract security audits.

To date, Quantstamp has helped to secure hundreds of millions of dollars of transaction value in smart contracts and has assisted dozens of blockchain projects globally with its white glove security auditing 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.

Finally, Quantstamp's dedication to research and development in the form of collaborations with leading academic institutions such as National University of Singapore and MIT (Massachusetts Institute of Technology) reflects Quantstamp's commitment to enable world-class smart contract innovation.

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. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The Solidity language itself and other smart contract languages remain under development and are subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity or the smart contract programming language, or other programming aspects that could present security risks. You may risk loss of tokens, Ether, and/or other loss. A report is not an endorsement (or other opinion) of any particular project or team, and the report does not guarantee the security of any particular project. A report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. To the fullest extent permitted by law, we disclaim all warranties, express 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 website, or any website or mobile application featured in any banner or other 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. You may risk loss of QSP tokens or other loss. 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.