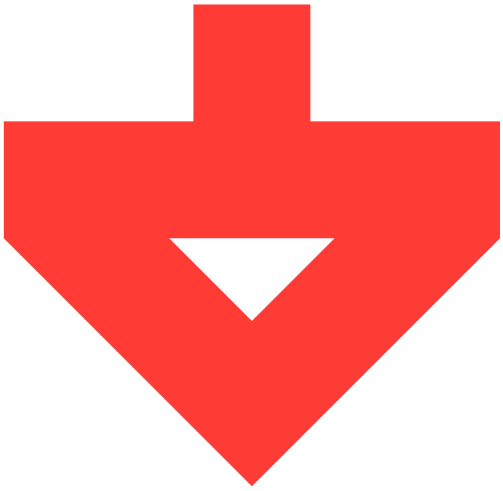
LIVENpay

Contract Audit



REPORT DATE	**November 15th, 2018**
REPORT VERSION	**3.0**

PREPARED BY

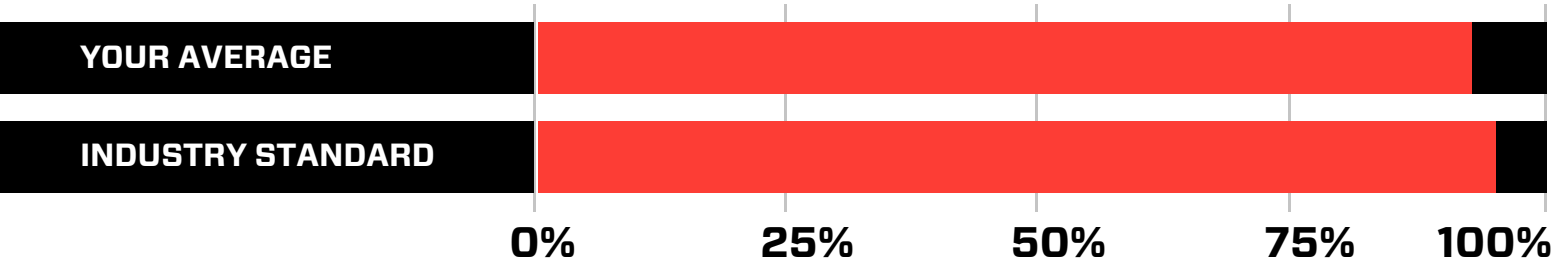
HOSHO

This document outlines the overall security of Liven's smart contract as evaluated by Hosho's Smart Contract auditing team. The scope of this audit was to analyze and document Liven's token contract codebase for quality, security, and correctness.

Contract Status

No issues were discovered in this contract during the auditing process. (See Complete Analysis)

Testable Code



Testable code is 92.58%, which is on par with the industry standard of 95%. (See Coverage Report)

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the Ethereum network's fast-paced and rapidly changing environment, we at Hosho recommend that the Liven team put in place a bug bounty program to encourage further and active analysis of the smart contract.

Table of Contents

Auditing Strategy and Techniques Applied

The Hosho team has performed a thorough review of the smart contract code, the latest version as written and updated on October 30th, 2018. All main contract files were reviewed using the following tools and processes. (See All Files Covered)

Throughout the review process, care was taken to ensure that the token contract:

- Implements and adheres to existing ERC-20 Token standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of gas, without unnecessary waste;
- Uses methods safe from reentrance attacks; and
- Is not affected by the latest vulnerabilities.

The Hosho team has followed best practices and industry-standard techniques to verify the implementation of Liven's token contract. To do so, the code is reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as they are discovered. Part of this work includes writing a unit test suite using the Meadow testing framework. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1	**Due diligence in assessing the overall code quality of the codebase.**
2	**Cross-comparison with other, similar smart contracts by industry leaders.**
3	**Testing contract logic against common and uncommon attack vectors.**
4	**Thorough, manual review of the codebase, line-by-line.**
5	**Deploying the smart contract to testnet and production networks using multiple client. implementations to run live tests.**

2.1 Summary

Liven is an ERC-20 token contract along with a crowdsale contract that accepts ETH and maintains a list of contributors for future manual token disbursement.

2.2 Coverage Report

As part of our work assisting Liven in verifying the correctness of their contract code, our team was responsible for writing a unit test suite using the Meadow testing framework.

- Branches: 90.38%
- Functions: 94.44%
- Lines: 92.92%

2.3 Failing Tests

No Failing Tests!

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or still need addressing. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to compile or operate in a significant way.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate, and is meant only as additional information.

3.1 Resolved: Does Not Follow ERC-223 Standards

CRITICAL

Contract: LivenCoin

Explanation
ERC-223 is designed to ensure that tokens cannot be transferred to a contract that are unable to explicitly handle them. The ERC-20 compatible transfer function does not have checks in place to ensure that.

Resolution
The Liven team has removed all ERC-223 functionality, resolving this issue.

3.2 Resolved: High Gas Cost Breaks Multisig Integration

CRITICAL

Contract: LivenSale

Explanation
When ETH contracts send a transaction to the fallback function of another contract, only 2300 gas is sent. A common resolution is to switch to a payable named function and make the fallback not payable.

Resolution
The Liven team has moved all of the business logic used to buy tokens to the buyTokens() function, giving multisig wallets and other contracts another path to buy tokens.

3.3 Resolved: ERC-223 Breaks Integration with Common Multisig Wallets

HIGH

Contract: LivenCoin

Explanation

Many multisig wallets (e.g. Gnosis) do not use the ERC-223 TokenFallback function, and will not be able to accept the tokens, even if they can handle the transfer of the tokens.

Resolution

The Liven team has removed all ERC-223 functionality, resolving this issue.

3.4 Can Give Allowance to 0x0

INFORMATIONAL

Contract: LivenCoin

Explanation

Approve function does not check that the target account is not 0x0.

3.5 Does Not Handle Transfer of Purchased Tokens

INFORMATIONAL

Contract: LivenSale

Explanation

This contract stores a list of contributors and contributions, but does not handle transfers of tokens to token holders.

We are grateful to have been given the opportunity to work with the Liven team.

The team of experts at Hosho, having backgrounds in all aspects of blockchain, cryptography, and cybersecurity, can say with confidence that the Liven contracts are free of any critical issues, having passed the rigorous Hosho auditing process.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

We at Hosho recommend that the Liven team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Test Suite Results

Contract: HoshoAudit.ERC20BasicStandardTests
√ erc20_Basic_Standards (0.0661480s)

Contract: HoshoAudit.ERC20Tests
√ decreaseApproval_DecreaseByHalf_EmitEvent (0.0531740s)
√ increaseApproval_Success (0.0247890s)
√ transferFrom_ApproveThenTransfer_EmitEvent (0.0748130s)
√ safeTransferFrom_ApproveThenTransfer_EmitEvent (0.0702620s)
√ fallbackFunction_Reverts (0.0049940s)
√ transferFrom_valueGreatThanAllowed_Revert (0.0365860s)
√ transferFrom_ToAccountZero_Revert (0.0326280s)
√ transferFrom_valueGreaterThanBalance_Revert (0.0055250s)
√ transfer_SendMoreThanBalance_ExpectRevert (0.0106910s)
√ transfer_ToAddressZero_ExpectRevert (0.0160110s)
√ decreaseApproval_DecreaseMoreThanAllowed_ExpectAllowanceSetTo0 (0.0477660s)
√ decreaseApproval_Success (0.0492420s)
√ allowance_CheckAmountApproved_AssertAreEqual (0.0534140s)
√ balanceOf_CheckOwnerBalance_AssertEqual (0.0445560s)
√ totalSupply_CheckTotalSupply_AssertTotal (0.0440370s)

Contract: HoshoAudit.ERC223Tests
√ ERC223Transfer_TransferToContract_TransactionCompletes (0.0689640s)
√ ERC223Transfer_TransferToAccountMoreThanBalance_TransactionReverts (0.0242310s)
√ ERC223Transfer_TransferToAddress_TransactionCompletes (0.0362950s)
√ ERC223Transfer_TransferTo0x0_TransactionReverts (0.0120970s)

Contract: HoshoAudit.LivenSaleTests
√ fallback_SendEthUnderMax_TransactionCompletes (0.0572590s)
√ fallback_SendEthAfterHittingMax_TransactionReverts (0.0573050s)
√ fallback_SendEthAfterSaleEndedByOwner_TransactionReverts (0.0570730s)
√ fallback_SendEthAfterSaleTimeEnds_TransactionReverts (0.0572680s)
√ fallback_SendEthUnderMin_TransactionReverts (0.0495290s)
√ fallback_SendEthAfterSaleTimeEndsAndExtended_TransactionCompletes (0.0573230s)
√ fallback_SendEthOverMax_TransactionCompletesReturnsEth (0.0572800s)

Contract: HoshoAudit.OwnableTests
√ TransferOwnershipTo0Revert (0.0075830s)

Contract: HoshoAudit.OwnableTests
√ RenounceOwnership_OwnershipRenounced (0.0113970s)
√ Owner_ReturnsOwner (0.0098760s)
√ TransferOwnershipPass (0.0130090s)
√ TransferOwnershipRevertFromNonOwner (0.0148800s)
√ RenounceOwnershipFailFromNotOwner (0.0112450s)
√ RevertSubtractionOverflow (0.0364240s)
√ AllowRegularDivision (0.0095190s)
√ AllowRegularMultiply (0.0082910s)
√ RevertAdditionOverflow (0.0345550s)
√ SkipOperationMult0 (0.0075740s)
√ AllowRegularAddition (0.0027900s)
√ AllowRegularSubtraction (0.0049530s)
√ RevertDivideBy0 (0.0572590s)
√ RevertMultiplyOverflow (0.0559760s)
√ safeApprove_ApproveValidAmount_Complete (0.0170970s)

Contract: HoshoAudit.SafeTransferTests
√ safeTransfer_TransferTo0x0_TransactionReverts (0.0101800s)
√ safeApprove_ApproveInvalidAmount_Revert (0.0081310s)
√ transferFrom_ApproveThenTransferTo0x0_Revert (0.0342710s)
√ transferFrom_ApproveThenTransfer_Complete (0.0408510s)

Contract: HoshoAudit.SafeTransferTests
√ safeTransfer_TransferToAccount_TransactionCompletes (0.0195620s)
√ afterUnlock_Locked_OnlyOwner (0.0112580s)
√ afterUnlock_Locked_Reverts (0.0711190s)

Appendix B

FILE	FINGERPRINT
LivenCoin.sol	A16F5D7214D3F164BBE9FEE2A6B3FCA4DA36E43E47078E2331E9924254769E46
LivenSale.sol	C67D5E2F11A8634E093D27EE36F6ED959471EDA6FB8E30114B161BCD22EADBE4
ERC20.sol	62D04608B75BEB40943FD7C1217C002A5697650628E7771D3672A195032C0400
ERC223.sol	AAFEE763603B4DCE519E4506B85DA23B0C42EBA2086D01417F94E703CDE1323D
Generic223Receiver.sol	0684EEDDA3A4A903F382043A13C50F5BC53D7EA428FA978711C78B1CD282524E
Ownable.sol	06B57668F0A542DF3872D5C6396D476A3804C06287DCDE219553DE5AC501B8C9
SafeMath.sol	3F50F4F05FEACC4A59932B7A23BC0536C3389DA4D7A2BBF2580497375F7E86A6

Appendix C

FILE	% BRANCHES	% FUNCTION	% LINES
LivenCoin.sol	92.31%	100%	100%
LivenSale.sol	100%	100%	100%
ERC20.sol	100%	100%	100%
ERC223.sol	100%	100%	100%
Generic223Receiver.sol	25%	50%	50%
Ownable.sol	100%	100%	100%
SafeMath.sol	100%	100%	100%
ALL FILES	90.38% (47/52)	94.44% (34/36)	92.92% (105/113)