Audit of LUASWAP Contracts

a report authored by

TomoChain Core Team and The Arcadia Group

Table of Contents

Executive Summary

The LUASWAP token and farming smart contracts were developed by TomoChain developers and are subject to be rigorously reviewed and audited. The LUA development team therefore requested a rigorous review from all TomoChain developers internally and The Arcadia Group auditing team externally for ensuring security of the contracts. The following LUASWAP smart contracts are reviewed in this report:

- LUA token smart contract deployed on Ethereum at address 0xB1f66997A5760428D3a87D68b90BfE0aE64121cC
- LuaMasterFarmer deployed at 0xb67D7a6644d9E191Cac4DA2B88D6817351C7fF62.

The team has completed the review using various methods primarily consisting of dynamic analysis, static analysis, and rigorous unit-tests. This process included a line by line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities, limiters, and reference against intended functionality as well as writing unit-tests for the smart contracts. Additionally, the audit also follows the smart contract audit checklist https://medium.com/@knownsec404team/ethereum-smart-contract-audit-checklist-ba9d1159b901 and https://diligence.consensys.net/blog/2019/09/how-to-prepare-for-a-smart-contract-audit/.

Findings

LuaToken.sol

Severity	Lines	Description	Status
Low		The use of locked and unlocked balance in LuaToken needs to be careful as the logic seems to be a bit complex. The development and auditing team have rigorously tested all functions in the token, and, within the scope of the audit, could not find any bugs/errors related to lock/unlock tokens.	Resolved
Medium	1220, 1234	The function unlock() uses the result of function canUnlockAmount(), which has a division operator, may result in incorrect released tokens. Respond to the issue, the development team decided to make additional tests for the functions. Integration tests for those functions were also done against both on TomoChain Devnet and Ethereum Ropsten Testnet (in addition to Ganache local testnet).	Resolved
Medium	1592	Timing for releasing locked tokens could be changed by function transferAll. The function is intended to transfer all locked tokens from the sender to the recipient. However, as the function updates _lastUnlockBlock of the recipient _to as the last unlock block of the sender. This could introduce potential timing release issue for the recipient _to. This issue does not affect token holder balance and the function is only used for the development team to mitigate tokens one LuaSwap launch.	Resolved

LuaMasterFarmer.sol

Severity	Lines	Description	Status

Low	1602	Input parameter _startBlock in constructor is not checked for its value. The check is to ensure that the _startBlock for counting rewards is after the current block. However, as the constructor was properly initialized, the issue is mitigated to have no security risks.	Resolved
Information	1592	There is no function to remove or update a farming pool. This is an intended behavior as removing a farming a pool is complex and prone to errors/bugs. The mitigation is to carefully review any farming pool information before deloying and adding it to master smart contracts.	Resolved
Medium		The logic of function _harvest is too complex and contains many if-else statements that are prone to errors/bugs. Upon the review and report of the issue, the development team immediately reviewed the logic in the function and simplified it. After the simplification, another rigorous review and unit-testing for the function were carried to ensure the simplified function is bug-free.	Resolved
Low	1766-1773	_harvest function transfers an amount of token to user and then immediately locks part of the transferred amount by transferring from the token holder to the farming contract. This is a valid behavior of the contract compared to the Lua token specification. However, due to locking by transferring, for some ERC-20 wallets such as Trust Wallet or Imtoken wallet, the token holder might be confused as the holder might see 2 notifications within his mobile app: 1 notification of receive XXX token while another notification of sending YYY token. This can make the holder scared as token is automatically transferred out of his wallet without his send action.	Resolved

		This is mostly related to user experience. To mitigate this issue, the team needs to make UI of LuaSwap to inform users about the transfer notifications.	

Conclusion

Based on the smart contracts code, the unit-tests, the communication between the auditing team and the development team, the analysis and review of the auditing team on the smart contracts, the auditing team concludes that:

- Within the scope of the audit, the smart contracts were well written. Some common security standards were also used by the development team.
- The issues reported by the auditing team were carefully taken into account and resolved by the development team.

Disclaimer