0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

...

Troller token

October 2021

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for Troller team. The code is deployed to BSC mainnet 0x77cC0eaf92D83c3220E54b3E58Da56BE40E148F8. Users must check that the contracts they are interacting are the same as been audited.A recheck has been done for the updated contract deployed 0x270e18A7E4575C8Ce8c39253e5Cccec9D236A3e6.

All issues have been fixed or responded Successfully (documentation was added to the contract code). See team responses and updates below the issues description. Automated analysis checked for 26 issues and 25 were passed (96.16%).

Name	Troller token
Audit date	2021-10-07 - 2021-10-07
Language	Solidity
Platform	Binance Smart Chain

Contracts checked

Name	Address
Troller	https://bscscan.com/address/0x270e18A7E4575C8Ce8c39253e5Cccec9D236A3e6#code

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools

- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyse smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	passed
Code With No Effects	not passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed
Use of Deprecated Solidity Functions	passed
Assert Violation	passed

State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

Classification of issue severity

High severity High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.

Medium severity Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.

Low severity Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Exclude/include from rewards (Troller)

Dynamic correlation between rOwned and tOwned balances may be used by the owner to redistribute users balances for the owner's profit.

Recommendation: Refactor the includeInReward() function or remove it from the code. See detailed explanation here.

Update: The function includeInReward() was removed from the code. Owner should use exlcudeFromReward() with extra care.

Medium severity issues

1. Ownership unlock (Troller)

unlock() function does not update _previousOwner variable. Ownership may be transferred back after renounce if function lock() was called previously.

Recommendation: Remove function if it is not goint to be used.

Update: Unlock function was deleted, but code still has an issue: now the lock function effectively renounces ownership of the token.

2. Hardcoded addresses (Troller)

Hardcoded router address updated in constructor but has no specific update function.

Recommendation: Add function to update the router address to be ready for migration to the new dex version.

Update: Function to set a new router was added.

3. Liquify conditions (Troller)

Conditions in _transfer() function L1115 contradicts the comment L1104.

Recommendation: Update the comment or the code.

Update: Code was update to conform to the documentation.

4. Token transfers may run out of gas if big number of addresses are exluded (Troller)

Token transfers may run out of gas if big number of addresses are exluded. The _getCurrentSupply() loops over unlimited number of excluded addresses.

```
    function _getCurrentSupply() private view returns(uint256, uint256) {
        uint256 rSupply = _rTotal;
        uint256 tSupply = _tTotal;
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal,
 _tTotal);
            rSupply = rSupply.sub(_rOwned[_excluded[i]]);
            tSupply = tSupply.sub(_tOwned[_excluded[i]]);
        }
        if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
        return (rSupply, tSupply);
    }
```

Recommendation: The token owner should be extra careful when adding excluded addresses.

Update: Developers are aware of possible issue if a big number of addresses are added, extra documentation was added to the code.

Low severity issues

1. SafeMath usage (Troller)

SafeMath library is outdated and should not be used with 0.8 Solidity versinons.

Recommendation: Update vesion of the SafeMath library

Update: Issue was fixed in the update

2. Buyback designed ineffective (Troller)

Buyback functions first swaps tokens for BNB and then swaps BNB back to token.

Update: The buyback functionality was removed from the token.

3. Swaps without slippage (Troller)

Swaps are performed with 0 slippage parameters. This means that the actual swaps will be done with 100% slippage and may be frontrun. Issue may have significant impact on big token swaps.

Recommendation: It's a common way to do the swaps in the Safemoon forks, the owners of the tokens should be aware of it when setting parameters for swaps. We recommend update documentation of the function.

Update: Comments were added in the code.

4. Wrong swap deadline (Troller)

Swap in L1235 has deadline parameter set to timestamp+300, which would not work, i.e. is always true.

Recommendation: It's a common way to do the swaps in the Safemoon forks, the owners of the tokens should be aware of it when setting parameters for swaps. We recommend update documentation of the function.

Update: Developers added comment in the code that it is an acknowledged issue.

Conclusion

Troller token Troller contract was audited. 1 high, 4 medium, 4 low severity issues were found.

All issues have been fixed or responded Successfully (documentation was added to the contract code).
See team responses and updates below the issues description.
Automated analysis checked for 26 issues and 25 were passed (96.16%).

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

0x Guard