0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

Exilon

November 2021

0xguard.com

hello@0xguard.com

Contents

Introduction

Auto-yield BEP20 token inspired by Reflect Finance model with different default values of fees for common transfer, purchase, or sale. Fixed BUSD value of marketing fee for standard transfers. 1% burn, 2% marketing, 1% auto-yield, up to 1% reserve, and 8% (9% if zero holders for auto-yield) liquidity fees for purchases from DEX pair. The same values for sales with an additional 2% to liquidity after the first 60 minutes of open trading (and another +3% from 30 to 60 minutes, or +6% from 0 to 30 minutes). Addresses may be granted with 10 times lower commissions or excluded from fees at all.

Name	Exilon
Audit date	2021-11-28 - 2021-11-29
Language	Solidity
Platform	Binance Smart Chain

Contracts checked

Name	Address
Exilon	0x5648D13AD81C569FE36D2a4A73996BcB34f661dd

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	not 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Outdated Compiler Version	passed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. Error in distributeTokens() function (Exilon)

distributeTokens() function doesn't update the balance of callers who are excluded from distribution, see L1582. Thus the owner (or other DEFAUL_ADMIN_ROLE bearers) must pay attention when calling the excludeFromFeesDistribution() function.

Recommendation: Grant the default admin role to a proxy contract with restricted ability to call excludeFromFeesDistribution() function and renounce the admin role from the current EOA.

Medium severity issues

1. Lower commission error in _makeBuyAction() function (Exilon)

isHavingLowerCommissions[from] in L2243 is always false because the DEX pair can't be added to lower commissions mapping.

Recommendation: Correct code is isHavingLowerCommissions[to].

Low severity issues

1. Increased sell fee could be avoided (Exilon)

isSellingBig boolean flag of _makeSellAction() function is calculated by comparing the sale amount to 90% of the seller's balance, see L1866 and L1961. Splitting the sale amount to 90%-1 as the first part and the rest as the second could be used to partially avoid the increased fees on sellings.

Recommendation: Update the documentation or inform the users in any available way.

2. Lack of increase- and decreaseAllowance() functions (Exilon)

Direct modification of the allowances enables the front-run attack to spend both old and new approved values.

Recommendation: A common solution for this problem is increase/decreseAllowance functions. These functions read the current allowance values before modifying them.

3. Redundant code (Exilon)

onlyAdmin() modifier could be removed in favor of standard onlyRole() modifier of the AccessControl contract by OpenZeppelin available since v4.1 release.

4. Typos & inconsistent comments (Exilon)

Commentaries in L1375, 1473 contain typos in 'exclude'.

Commentaries in L1898, 1952 are hard to comprehend.

Conclusion

The audited contract is an RFI-like BEP20 token with a custom fee model. Open high severity issue was found in distributeTokens() function that should be restricted to excluded addresses.

40% of the total supply was added to the EXL/WBNB pair according to the initial token distribution, now (November 29, 2021) the LP pair holds about 20% of the total supply. About 86% of LP tokens is locked within locker contract.

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