Guard

Smart contracts security assessment

Final report

Tariff: Standard

Parts of Four Token

November 2021

0xguard.com

hello@0xguard.com

Contents

Introduction

The report was prepared for Parts of Four Token team. The code was provided in a file after deployment report will be updated with a deployed address.

The token supports the ERC-20 interface and realizes a custom tokens redistribution concept. Commissions are taken on transfers and redistributed between other users.

Name	Parts of Four Token
Audit date	2021-11-10 - 2021-11-10
Language	Solidity
Platform	Ethereum

Contracts checked

Name	Address
P4CToken	

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	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	not 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	not 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Outdated Compiler Version	passed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

No issues were found

Medium severity issues

1. Internal balance should be recalculated (P4CToken)

The owner can add back a previously excluded address and in this will redistribute tokens. Part of token rewards will be redistributed to the included account. If an address is removed from the list of excluded and added back again, it saves tokens gained from the token redistribution as it was included in the reward distribution all the time.

Low severity issues

1. Double logical NOT operator (P4CToken)

Even number of logical negation **L:111** has no effect on the statement.

```
require(!!excludedAddressesMap[_addr], "_addr is not an excluded address.");
```

Recommendation: Logical negations operators can be removed from the line in undisrupted embedded logic way.

2. Lack of increaseAllowance and decreaseAllowance functions (P4CToken)

There is a known frontrun attack on approve/transferFrom methods.

Reccomendation: We recommend adding *increaseAllowance* and *decreaseAllowance* functions to atomically change allowance

3. Transfers may exceed gas block limit (P4CToken)

There is a for loop over an array with an unlimited number of items in the transfer() function:

```
for (uint i; i < excludedAddresses.length; i++) {
// Because this is rounded down, excludedAddresses will slowly lose funds as more
transactions are made.
// However, due to the fact that transactions are expensive and we have such a high
precision, this
// doesn't make a difference in practice.
uint256 oldBalance = internalBalances[excludedAddresses[i]];
uint256 newBalance = ((oldBalance * 1e18) / readjustmentFactor);
internalBalances[excludedAddresses[i]] = newBalance;
removedFunds += oldBalance - newBalance;
}
```

If a big number of addresses are excluded the gas usage of transfer() function (and also removeExcludedAddress()) may exceed the block gas limit which will fail all transfers.

Recommendation: The token owner should be aware of the possible problem if a big number of excluded addresses are added.

Conclusion

Parts of Four Token P4CToken contract was audited. 1 medium, 3 low severity issues were found.

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