Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details

Audited project

CRYPTO PHOENIX

Deployer address

0x3b7B8d99827B1059459a745d7989d6548CFEEA87

Client contacts:

CRYPTO PHOENIX team

Blockchain

Ethereum

Project website:

https://CryptoPhoenix.com

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by CRYPTO PHOENIX to perform an audit of smart contracts:
https://etherscan.io/address/0x8689d850cdf3b74a1f6a5eb60302c785b71c2fc7#code

The purpose of the audit was to achieve the following:

- **Ensure that the smart contract functions as intended.**
- **Identify potential security issues with the smart contract.**

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

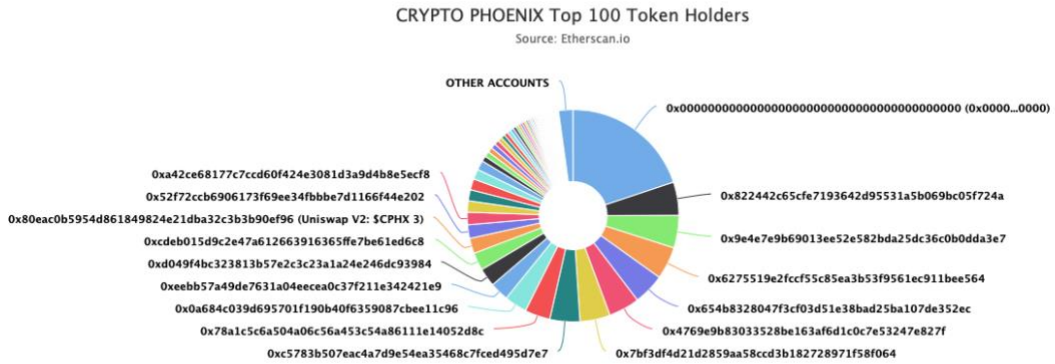
Token contract details for 14.07.2021

Contract name	CRYPTO PHOENIX
Contract address	0x8689D850CdF3b74A1F6A5eB60302c785B71c2fc7
Total supply	1,000,000,000,000,000
Token ticker	$CPHX
Decimals	18
Token holders	169
Transactions count	843
Top 100 holders dominance	97.82%
Burn fee	16
Tax fee	2
Dev fee sum	4
Manager	0x3b7b8d99827b1059459a745d7989d6548cfeea87
Contract deployer address	0x3b7B8d99827B1059459a745d7989d6548CFEEA87
Contract's current owner address	0x3b7b8d99827b1059459a745d7989d6548cfeea87

CRYPTO PHOENIX Token Distribution

The top 100 holders collectively own 97.82% (978,182,940,009,776.00 Tokens) of CRYPTO PHOENIX

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 169

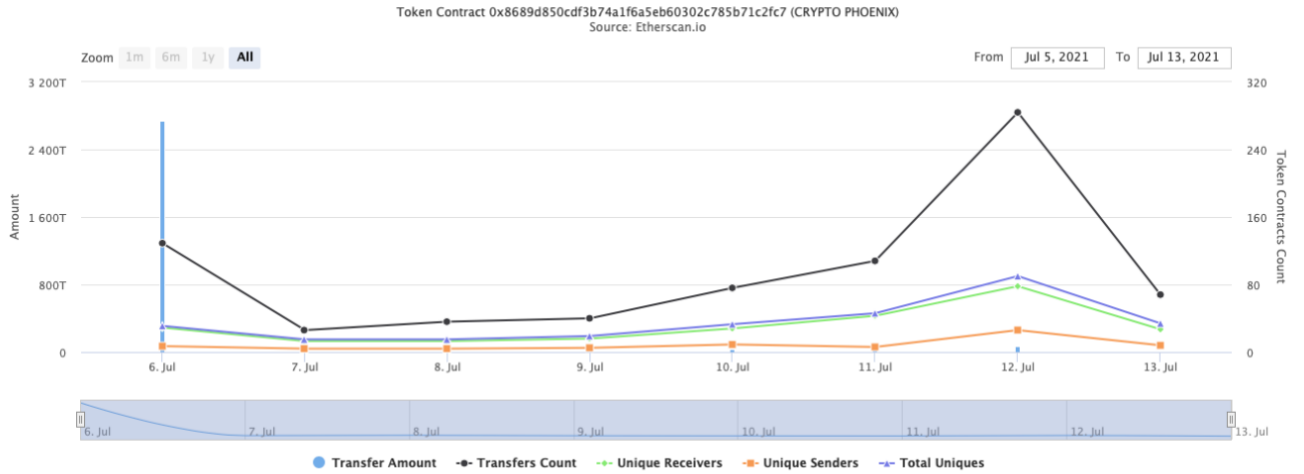


(A total of 978,182,940,009,776.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

CRYPTO PHOENIX Contract Interaction Details

Time Series: Token Contract Overview

Tue 6, Jul 2021 - Tue 13, Jul 2021



CRYPTO PHOENIX Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x0000...0000	198,964,862,281,719.91237688112085105	19.8965%
2	0x822442c65cfe7193642d95531a5b069bc05f724a	50,656,545,028,042.257039774340087419	5.0657%
3	0x9e4e7e9b69013ee52e582bda25dc36c0b0dda3e7	49,616,780,777,179.313732528510475175	4.9617%
4	0x6275519e2fccf55c85ea3b53f9561ec911bee564	49,419,985,342,740.823317394070646269	4.9420%
5	0x654b8328047f3cf03d51e38bad25ba107de352ec	47,368,288,660,875.2154869405431639	4.7368%
6	0x4769e9b83033528be163af6d1c0c7e53247e827f	47,293,241,019,310.568504292100446756	4.7293%
7	0x7bf3df4d21d2859aa58ccd3b182728971f58f064	46,200,267,877,541.999927553902121476	4.6200%
8	0xc5783b507eac4a7d9e54ea35468c7fced495d7e7	45,788,635,705,470.400879390140047166	4.5789%
9	0x78a1c5c6a504a06c56a453c54a86111e14052d8c	38,904,478,133,120.447970371879182314	3.8904%
10	0x0a684c039d695701f190b40f6359087cbee11c96	34,557,391,627,182.858128673621011807	3.4557%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Int] IERC20Metadata (IERC20)
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Int] functionStaticCall
  - [Int] functionStaticCall
  - [Int] functionDelegateCall #
  - [Int] functionDelegateCall #
  - [Prv] _verifyCallResult

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Pub] getUnlockTime
  - [Pub] lock #
    - modifiers: onlyOwner
  - [Pub] unlock #

+ Manageable (Context)

  - [Pub] <Constructor> #
  - [Pub] manager
  - [Ext] transferManagement #
    - modifiers: onlyManager

+ [Int] IUniswapV2Factory
  - [Ext] createPair #

+ [Int] IUniswapV2Router
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidityETH ($)
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ Tokenomics
  - [Pub] <Constructor> #
  - [Prv] _addFee #
  - [Prv] _addFees #
  - [Int] _getFeesCount
  - [Prv] _getFeeStruct
  - [Int] _getFee
  - [Int] _addFeeCollectedAmount #
  - [Int] getCollectedFeeTotal

+ Presaleable (Manageable)
  - [Ext] setPreseableEnabled #
    - modifiers: onlyManager

+ BaseRfiToken (IERC20, IERC20Metadata, Ownable, Presaleable, Tokenomics)
  - [Pub] <Constructor> #
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Pub] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #
  - [Ext] burn #
  - [Int] _burnTokens #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Ext] isExcludedFromReward
  - [Ext] reflectionFromToken
  - [Int] tokenFromReflection
  - [Ext] excludeFromReward #
    - modifiers: onlyOwner
  - [Int] _exclude #
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Ext] setExcludedFromFee #
    - modifiers: onlyOwner
  - [Pub] isExcludedFromFee
  - [Int] _approve #

  - [Int] _isUnlimitedSender
  - [Int] _isUnlimitedRecipient
  - [Prv] _transfer #
  - [Prv] _transferTokens #
  - [Prv] _takeFees #
  - [Int] _getValues
  - [Int] _getCurrentRate
  - [Int] _getCurrentSupply
  - [Int] _getSumOfFees
  - [Int] _isV2Pair
  - [Int] _redistribute #
  - [Int] _takeTransactionFees #

+ UniHelper (Ownable, Manageable)
  - [Ext] <Fallback> ($)
  - [Int] initializeRouterPair #
  - [Prv] _setRouterAddress #
  - [Ext] setRouterAddress #
    - modifiers: onlyManager
  - [Int] _approveDelegate #

+ Antiwhale (Tokenomics)
  - [Int] _getAntiwhaleFees

+ PhoenixAbstract (BaseRfiToken, UniHelper, Antiwhale)
  - [Pub] <Constructor> #
  - [Int] _isV2Pair
  - [Int] _getSumOfFees
  - [Int] _takeTransactionFees #
  - [Prv] _burn #
  - [Prv] _takeFee #
  - [Int] _approveDelegate #

+ CRYPTOPHOENIX (PhoenixAbstract)
  - [Pub] <Constructor> #
    - modifiers: PhoenixAbstract

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function `includeInReward()` uses the loop to find and remove addresses from the `_excluded` list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

```
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function `_getCurrentSupply` also uses the loop for evaluating total supply. It also could be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() internal view returns(uint256, uint256) {
    uint256 rSupply = _reflectedSupply;
    uint256 tSupply = TOTAL_SUPPLY;

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_reflectedBalances[_excluded[i]] > rSupply || _balances[_excluded[i]] > tSupply) return (_reflectedSupply, TOTAL_SUPPLY);
        rSupply = rSupply.sub(_reflectedBalances[_excluded[i]]);
        tSupply = tSupply.sub(_balances[_excluded[i]]);
    }
    if (tSupply == 0 || rSupply < _reflectedSupply.div(TOTAL_SUPPLY)) return (_reflectedSupply, TOTAL_SUPPLY);
    return (rSupply, tSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

Notes:

- The function `_burn()` sends burn amount to burnAddress.

Owner privileges (In the period when the owner is not renounced)

- Manager can enable and disable fees.

```
function setPreseableEnabled(bool value⬆) external onlyManager {
    isInPresale = value⬆;
}
```

- Manager can transfer manager rights.

```
function transferManagement(address newManager⬆) external virtual onlyManager {
    emit ManagementTransferred(_manager, newManager⬆);
    _manager = newManager⬆;
}
```

- Manager can change router.

```
ftrace | funcSig
function setRouterAddress(address router⬆) external onlyManager() {
    _setRouterAddress(router⬆);
}
```

- Owner can exclude from fees.

```
function setExcludedFromFee(address account⬆, bool value⬆) external onlyOwner { _isExcludedFromFee[account⬆] = value⬆; }
```

- Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

```
ftrace | funcSig
function lock(uint256 time⬆) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time⬆;
    emit OwnershipTransferred(_owner, address(0));
}
ftrace | funcSig
function unlock() public virtual {
    require(_previousOwner == msg.sender, "Only the previous owner can unlock onwership");
    require(block.timestamp > _lockTime , "The contract is still locked");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team:
https://dxsale.app/app/v2_9/dxlockview?id=1&add=0x3b7B8d99827B1059459a745d7989d6548CFEEA87&type=lplock&chain=ETH

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.