Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

Winry Inu

Deployer address

0xF7bF4F9FF7c183D67095676DeAe715222D78a8Aa

Client contacts:

Winry Inu team

Blockchain

Ethereum

Project website:

https://winryinu.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.

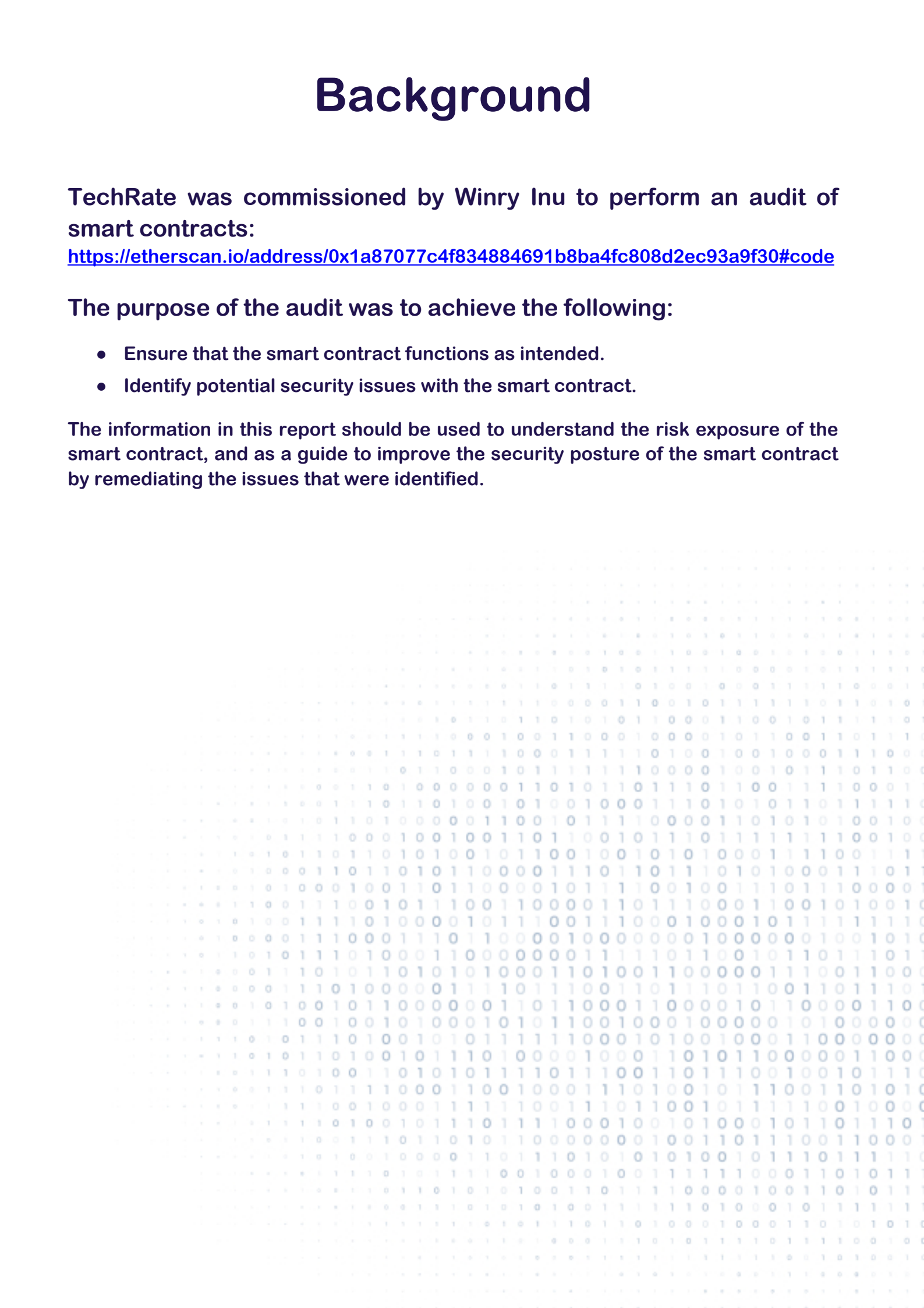
Background

TechRate was commissioned by Winry Inu to perform an audit of smart contracts:
https://etherscan.io/address/0x1a87077c4f834884691b8ba4fc808d2ec93a9f30#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

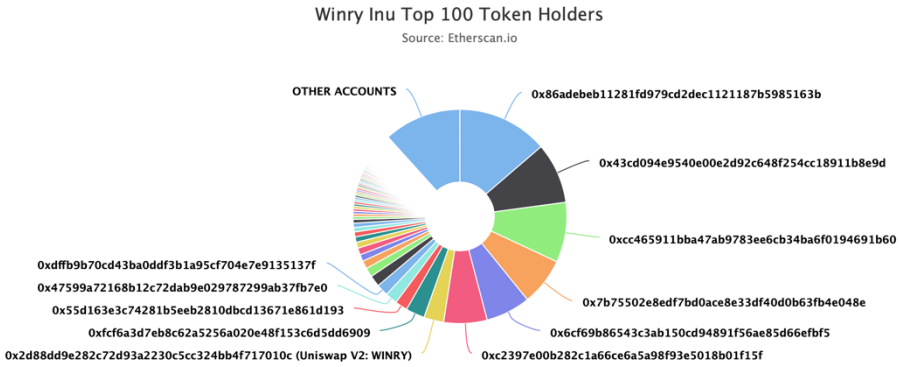
Token contract details for 03.11.2021

Contract name	Winry Inu
Contract address	0x1a87077C4F834884691B8ba4fc808D2eC93A9F30
Total supply	1,000,000,000,000,000,000
Token ticker	WINRY
Decimals	9
Token holders	1,498
Transactions count	3,426
Top 100 holders dominance	88.28%
Contract deployer address	0xF7bF4F9FF7c183D67095676DeAe715222D78a8Aa
Contract's current owner address	0x0000000000000000000000000000000000000000

Winry Inu Token Distribution

The top 100 holders collectively own 88.28% (882,837,904,567,377,000.00 Tokens) of Winry Inu

Token Total Supply: 1,000,000,000,000,000,000.00 Token | Total Token Holders: 1,498

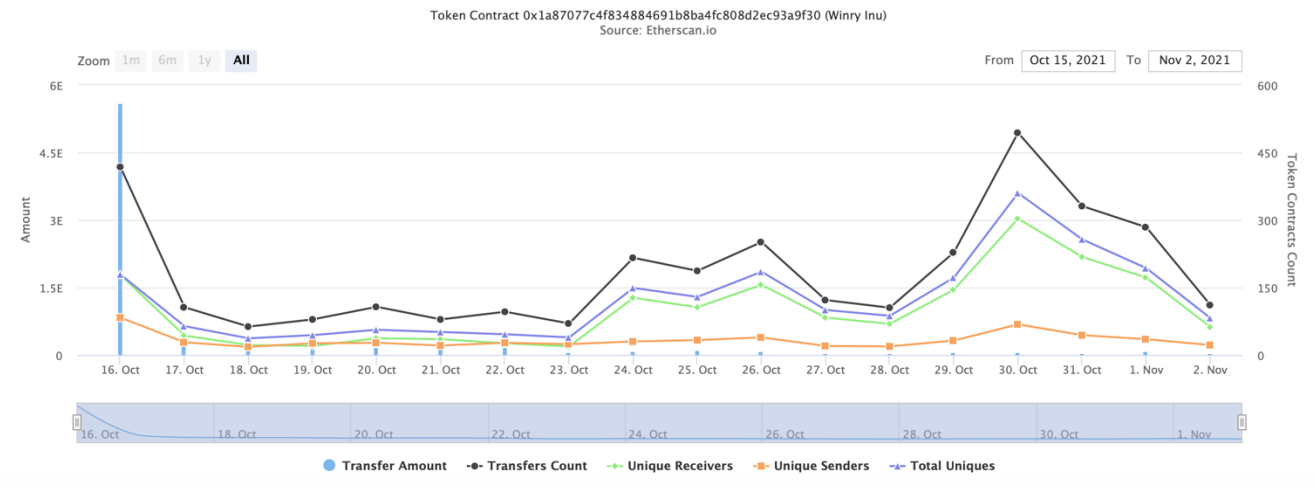


(A total of 882,837,904,567,377,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000,000.00 token)

Winry Inu Contract Interaction Details

Time Series: Token Contract Overview

Sat 16, Oct 2021 - Tue 2, Nov 2021



Winry Inu Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	0x86adebeb11281fd979cd2dec1121187b5985163b	137,187,041,162,639,000.670569378	13.7187%
2	0x43cd094e9540e00e2d92c648f254cc18911b8e9d	91,633,691,527,719,900.293764996	9.1634%
3	0xcc465911bba47ab9783ee6cb34ba6f0194691b60	90,709,279,288,961,400.897672585	9.0709%
4	0x7b75502e8edf7bd0ace8e33df40d0b63fb4e048e	72,006,006,062,563,200.134834582	7.2006%
5	0x6cf69b86543c3ab150cd94891f56ae85d66efbf5	67,438,316,686,175,900.840666741	6.7438%
6	0xc2397e00b282c1a66ce6a5a98f93e5018b01f15f	65,587,391,972,438,600.979370761	6.5587%
7	Uniswap V2: WINRY	30,104,575,044,990,000.742284195	3.0105%
8	0xfcf6a3d7eb8c62a5256a020e48f153c6d5dd6909	28,562,942,307,871,400.17334549	2.8563%
9	0x55d163e3c74281b5eeb2810dbcd13671e861d193	18,841,105,632,925,700.015440293	1.8841%
10	0x47599a72168b12c72dab9e029787299ab37fb7e0	18,453,455,355,668,800.453350372	1.8453%

Contract functions details

+ Context
  - [Int] _msgSender

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

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

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

+ [Int] IUniswapV2Factory
  - [Ext] createPair #

+ [Int] IUniswapV2Router02
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidityETH ($)

+ WinryInu (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Ext] setCooldownEnabled #
    - modifiers: onlyOwner
  - [Prv] tokenFromReflection
  - [Prv] _approve #
  - [Ext] setFeeAmountOne #
  - [Ext] setFeeAmountTwo #

- [Prv] _transfer #
- [Prv] swapTokensForEth #
  - modifiers: lockTheSwap
- [Prv] sendETHToFee #
- [Ext] openTrading #
  - modifiers: onlyOwner
- [Pub] setBots #
  - modifiers: onlyOwner
- [Pub] delBot #
  - modifiers: onlyOwner
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _isBuy
- [Prv] _takeTeam #
- [Prv] _reflectFee #
- [Ext] <Fallback> ($)
- [Ext] manualswap #
- [Ext] manualsend #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply

($) = 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 issue
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 setBots() uses the loop to add bots to list.

```
function setBots(address[] memory bots_) public onlyOwner {
    for (uint i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}
```

Recommendation:
Check that the bots array length is not too big.

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

- Owner can enable / disable cooldown (user to user trading with time offset).

```
function setCooldownEnabled(bool onoff) external onlyOwner() {
    cooldownEnabled = onoff;
}
```

- Owner can open swap trading.

```
function openTrading() external onlyOwner() {
    require(!tradingOpen,"trading is already open");
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D);
    uniswapV2Router = _uniswapV2Router;
    _approve(address(this), address(uniswapV2Router), _tTotal);
    uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this), _uniswapV2Router.WETH());
    uniswapV2Router.addLiquidityETH{value: address(this).balance}(address(this),balanceOf(address(this)),0,0,owner(),block.timestamp);
    swapEnabled = true;
    cooldownEnabled = true;
    _maxTxAmount = 50000000000000000 * 10**9;
    tradingOpen = true;
    IERC20(uniswapV2Pair).approve(address(uniswapV2Router), type(uint).max);
}
```

- Owner can add and remove bots (no transferring between this addresses).

```
function setBots(address[] memory bots_) public onlyOwner {
    for (uint i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}

function delBot(address notbot) public onlyOwner {
    bots[notbot] = false;
}
```

- Fee address wallet 2 can change fees.

```
function setFeeAmountOne(uint256 fee)  external  {
    require(_msgSender() == _feeAddrWallet2, "Unauthorized");
    _feeAddr1 = fee;
}

function setFeeAmountTwo(uint256 fee)  external  {
    require(_msgSender() == _feeAddrWallet2, "Unauthorized");
    _feeAddr2 = fee;
}
```

- Fee address wallet 1 can manual swap and send.

```
function manualswap() external {
    require(_msgSender() == _feeAddrWallet1);
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualsend() external {
    require(_msgSender() == _feeAddrWallet1);
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

Conclusion

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

Liquidity locking details are NOT provided by the team.

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.