In partnership with Femto Security

smart state

Web3 security easier
than ever

Dejitaru Tsuka

Oct 02

2022

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- FA2 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unused vars
- Unchecked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions/Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

WARNING

CHECK SUMMARY:

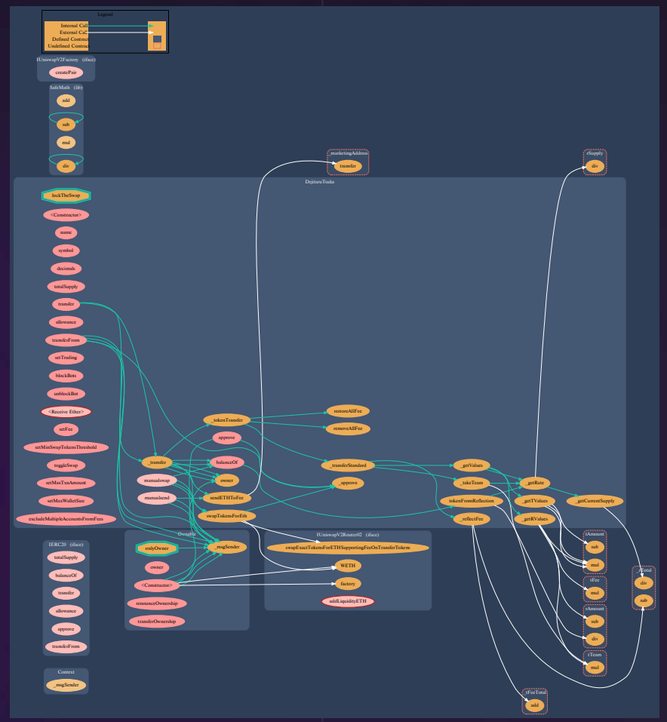
SafeMath usage for Solidity 0.8.9 is not necessary and can be removed for gas optimization.

STRUCTURE OF CONTRACT

DEJITARUTSUKA.SOL

CONTRACT METHODS ANALYSIS:

- name()
  Vulnerabilities not detected
- symbol()
  Vulnerabilities not detected
- decimals()
  Vulnerabilities not detected
- totalSupply()
  Vulnerabilities not detected



Pic. 1.1
DejitaruTsuka.sol

- balanceOf(address account)
  Vulnerabilities not detected

- transfer(address recipient, uint256 amount)
  Vulnerabilities not detected

- allowance(address owner, address spender)
  Vulnerabilities not detected

- approve(address spender, uint256 amount)
  Vulnerabilities not detected

- transferFrom(
    address sender,
    address recipient,
    uint256 amount
)
  Vulnerabilities not detected

- tokenFromReflection(uint256 rAmount)
  Vulnerabilities not detected

- removeAllFee()
  Vulnerabilities not detected

- restoreAllFee()
  Vulnerabilities not detected

- _approve(
    address owner,
    address spender,
    uint256 amount
)
  Vulnerabilities not detected

WARNING

- _transfer(
  address from,
  address to,
  uint256 amount
)

  Line 380:
  There is a condition (from == uniswapV2Pair && to != address(uniswapV2Router). Due to it users can abuse this functionality in order to evade fees. This can be achieved via calling removeLiquidityETHSupportingFeeOnTransferTokens on Uni V2. This function transfers all tokens from router's balance to the user. So the user can add a small amount of liquidity, perform a swap where they sets UniV2 Router as recepient and then call removeLiquidityETHSupportingFeeOnTransferTokens. They will get their tokens and won't pay any fees when buying tokens. This scenario is useful in case _taxFeeOnBuy and _redisFeeOnBuy are not 0.

WARNING

- swapTokensForEth(uint256 tokenAmount)
  path can be hardcoded for gas optimization.

- sendETHToFee(uint256 amount)
  Vulnerabilities not detected

WARNING

- setTrading(bool _tradingOpen)
  Function should emit an event

- manualswap()
  Vulnerabilities not detected

- manualsend()
  Vulnerabilities not detected

WARNING

- blockBots(address[] memory bots_)
  Function should emit an event

WARNING

- unblockBot(address notbot)
  Function should emit an event

- _tokenTransfer(
    address sender,
    address recipient,
    uint256 amount,
    bool takeFee
)
  Vulnerabilities not detected

- _transferStandard(
    address sender,
    address recipient,
    uint256 tAmount
)
  Vulnerabilities not detected

- _takeTeam(uint256 tTeam)
  Vulnerabilities not detected

- _reflectFee(uint256 rFee, uint256 tFee)
  Vulnerabilities not detected

- _getValues(uint256 tAmount)
  Vulnerabilities not detected

- _getTValues(
    uint256 tAmount,
    uint256 redisFee,
    uint256 taxFee
)
  Vulnerabilities not detected

- _getRValues(
    uint256 tAmount,
    uint256 tFee,
    uint256 tTeam,
    uint256 currentRate
)
  Vulnerabilities not detected

- _getRate()
  Vulnerabilities not detected

- _getCurrentSupply()
  Vulnerabilities not detected

WARNING

- setFee(uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell)
  Function should emit an event

WARNING

- setMinSwapTokensThreshold(uint256 swapTokensAtAmount)
  Function should emit an event

WARNING

- toggleSwap(bool _swapEnabled)
  Function should emit an event

WARNING

- setMaxTxnAmount(uint256 maxTxAmount)
  Function should emit an event

WARNING

- setMaxWalletSize(uint256 maxWalletSize)
  Function should emit an event

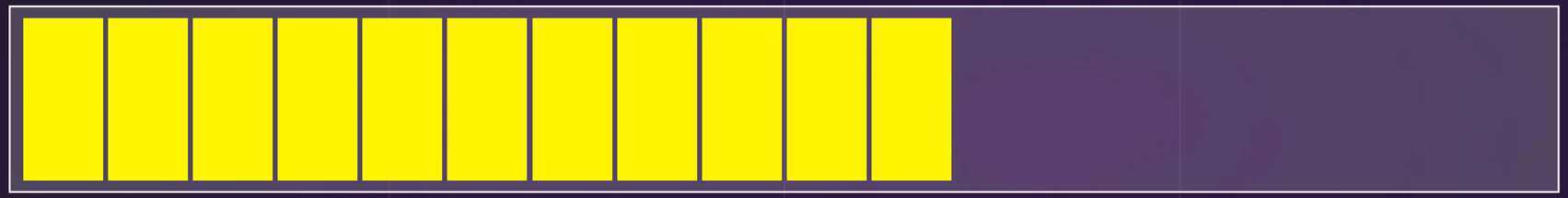
WARNING

- excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded)
  Function should emit an event

VERIFICATION CHECK SUMS

Contract Name	Solc version	Optimization	Bytecode hash (SHA 256)
DejitaruTsuka.sol	0.8.17	200	7690913bfa330e359e43ef27d10b4fdb4f7868defd9eccc5cb2abb6dcf7ba70b

PROJECT EVALUATION



6/10

smart state

GET IN TOUCH

info@smartstate.tech
smartstate.tech

View this report on smartstate.tech