APE AUDITS

FungieDAO(Child Contract)

Smart Contract Security Audit

Date 18/January/2022

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 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 ApeAudits and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ApeAudits) owe no duty of care towards you or any other person, nor does ApeAudits 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 ApeAudits 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, ApeAudits hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ApeAudits, 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.

Purpose

This document details ApeAudits' findings and recommended solutions. Our audit was performed in the weeks leading up to January 18, 2022. The contract we audited has yet to be deployed.

Findings

NO.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit	N/A	Passed
2	Race Conditions Audit	N/A	Passed
3	Authority Control Audit	Permission Vulnerability Audit Excessive Auditing Authority	Passed Passed
4	Safe Design Audit	Zeppelin Module Safe Compiler Version Hard-coded Version Fallback Function Safeuse Show Coding Security Function Return Value Security Call Function Security	Passed Passed Passed Passed Passed Passed Passed
5	Denial of Service Audit	N/A	Passed
6	Gas Optimization Audit	N/A	Passed
7	Design Logic Audit	N/A	Passed
8	Malicious Event Log Audit	N/A	Passed
9	"False Deposit" Vulnerability Audit	N/A	Passed
10	Uninitialized Storage Pointers Audit	N/A	Passed
11	Arithmetic Accuracy Deviation Audit	N/A	Passed

Contract Summary:

```
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)

+ Contract Ownable
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)
  - From Ownable
    - constructor() (internal)
    - owner() (public)
    - renounceOwnership() (public)
    - transferOwnership(address) (public)

+ Contract IERC20
  - From IERC20
    - allowance(address,address) (external)
    - approve(address,uint256) (external)
    - balanceOf(address) (external)
    - totalSupply() (external)
    - transfer(address,uint256) (external)
    - transferFrom(address,address,uint256) (external)
```

```
+ Contract ERC20
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)
  - From ERC20
    - _approve(address,address,uint256) (internal)
    - _beforeTokenTransfer(address,address,uint256) (internal)
    - _burn(address,uint256) (internal)
    - _mint(address,uint256) (internal)
    - _transfer(address,address,uint256) (internal)
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - balanceOf(address) (public)
    - constructor(string,string,uint8) (public)
    - decimals() (public)
    - decreaseAllowance(address,uint256) (public)
    - increaseAllowance(address,uint256) (public)
    - name() (public)
    - symbol() (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)
```

```
+ Contract IPancakeFactory (Most derived contract)
  - From IPancakeFactory
    - allPairs(uint256) (external)
    - allPairsLength() (external)
    - createPair(address,address) (external)
    - feeTo() (external)
    - feeToSetter() (external)
    - getPair(address,address) (external)
    - setFeeTo(address) (external)
    - setFeeToSetter(address) (external)

+ Contract TransferHelper (Most derived contract)
  - From TransferHelper
    - safeApprove(address,address,uint256) (internal)
    - safeTransfer(address,address,uint256) (internal)
    - safeTransferETH(address,uint256) (internal)
    - safeTransferFrom(address,address,address,uint256) (internal)

+ Contract IPancakeRouter01
  - From IPancakeRouter01
    - WETH() (external)
    - addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256) (external)
    - addLiquidityETH(address,uint256,uint256,uint256,address,uint256) (external)
    - factory() (external)
    - getAmountIn(uint256,uint256,uint256) (external)

    - getAmountOut(uint256,uint256,uint256) (external)

    - getAmountsIn(uint256,address[]) (external)

    - getAmountsOut(uint256,address[]) (external)

    - quote(uint256,uint256,uint256) (external)

    - removeLiquidity(address,address,uint256,uint256,uint256,address,uint256) (external)

    - removeLiquidityETH(address,uint256,uint256,uint256,address,uint256) (external)

    - removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32) (external)

    - removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32) (external)

    - swapETHForExactTokens(uint256,address[],address,uint256) (external)

    - swapExactETHForTokens(uint256,address[],address,uint256) (external)

    - swapExactTokensForETH(uint256,uint256,address[],address,uint256) (external)

    - swapExactTokensForTokens(uint256,uint256,address[],address,uint256) (external)

    - swapTokensForExactETH(uint256,uint256,address[],address,uint256) (external)

    - swapTokensForExactTokens(uint256,uint256,address[],address,uint256) (external)


+ Contract IPancakeRouter02 (Most derived contract)

  - From IPancakeRouter01

    - WETH() (external)

    - addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256) (external)

    - addLiquidityETH(address,uint256,uint256,uint256,address,uint256) (external)

    - factory() (external)
```

  - getAmountIn(uint256,uint256,uint256) (external)

  - getAmountOut(uint256,uint256,uint256) (external)

  - getAmountsIn(uint256,address[]) (external)

  - getAmountsOut(uint256,address[]) (external)

  - quote(uint256,uint256,uint256) (external)

  - removeLiquidity(address,address,uint256,uint256,uint256,address,uint256) (external)

  - removeLiquidityETH(address,uint256,uint256,uint256,address,uint256) (external)

  - removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32) (external)

  - removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32) (external)

  - swapETHForExactTokens(uint256,address[],address,uint256) (external)

  - swapExactETHForTokens(uint256,address[],address,uint256) (external)

  - swapExactTokensForETH(uint256,uint256,address[],address,uint256) (external)

  - swapExactTokensForTokens(uint256,uint256,address[],address,uint256) (external)

  - swapTokensForExactETH(uint256,uint256,address[],address,uint256) (external)

  - swapTokensForExactTokens(uint256,uint256,address[],address,uint256) (external)

- From IPancakeRouter02

  - removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,uint256) (external)

  - removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32) (external)

  - swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint256) (external)

    - swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256) (external)

    - swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256) (external)

+ Contract IPancakePair (Most derived contract)

  - From IPancakePair

    - DOMAIN_SEPARATOR() (external)

    - MINIMUM_LIQUIDITY() (external)

    - PERMIT_TYPEHASH() (external)

    - allowance(address,address) (external)

    - approve(address,uint256) (external)

    - balanceOf(address) (external)

    - burn(address) (external)

    - decimals() (external)

    - factory() (external)

    - getReserves() (external)

    - initialize(address,address) (external)

    - kLast() (external)

    - mint(address) (external)

    - name() (external)

    - nonces(address) (external)

    - permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (external)

    - price0CumulativeLast() (external)

    - price1CumulativeLast() (external)

```
    - skim(address) (external)
    - swap(uint256,uint256,address,bytes) (external)
    - symbol() (external)
    - sync() (external)
    - token0() (external)
    - token1() (external)
    - totalSupply() (external)
    - transfer(address,uint256) (external)
    - transferFrom(address,address,uint256) (external)

+ Contract IFungieDAO (Most derived contract)
  - From IFungieDAO
    - add_child_bnb_to_liquidity() (external)
    - add_external_child(address) (external)
    - disableTax() (external)
    - disable_tax_for_child(address) (external)
    - enableTax() (external)
    - enable_tax_for_child(address) (external)
    - get_child_contracts() (external)

+ Contract SafeMath (Most derived contract)
  - From SafeMath
    - add(uint256,uint256) (internal)
    - div(uint256,uint256) (internal)
    - div(uint256,uint256,string) (internal)
    - mod(uint256,uint256) (internal)
    - mod(uint256,uint256,string) (internal)
    - mul(uint256,uint256) (internal)
    - sub(uint256,uint256) (internal)
    - sub(uint256,uint256,string) (internal)
    - tryAdd(uint256,uint256) (internal)
    - tryDiv(uint256,uint256) (internal)
    - tryMod(uint256,uint256) (internal)
    - tryMul(uint256,uint256) (internal)
    - trySub(uint256,uint256) (internal)

+ Contract Child (Most derived contract)
  - From Ownable
    - constructor() (internal)
    - owner() (public)
    - renounceOwnership() (public)
    - transferOwnership(address) (public)
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)
  - From ERC20
    - _approve(address,address,uint256) (internal)
    - _beforeTokenTransfer(address,address,uint256) (internal)
    - _burn(address,uint256) (internal)
    - _mint(address,uint256) (internal)
```

  - allowance(address,address) (public)
  - approve(address,uint256) (public)
  - balanceOf(address) (public)
  - constructor(string,string,uint8) (public)
  - decimals() (public)
  - decreaseAllowance(address,uint256) (public)
  - increaseAllowance(address,uint256) (public)
  - name() (public)
  - symbol() (public)
  - totalSupply() (public)
  - transfer(address,uint256) (public)
  - transferFrom(address,address,uint256) (public)
- From Child
  - _transfer(address,address,uint256) (internal)
  - _transferExcluded(address,address,uint256) (private)
  - _transferIfNotLauched(address,address,uint256) (private)
  - _transferStandard(address,address,uint256) (private)
  - addExcluded(address) (public)
  - addLiquidity(uint256,uint256) (internal)
  - addToBlacklist(address[]) (public)
  - changeWithdrawInterval(uint256) (public)
  - constructor(uint8,uint8,uint8,uint8,uint8,address,address,uint8,address) (public)
  - getWithdrawableAmount(address) (public)
  - launch() (public)
  - receive() (external)

- removeExcluded(address) (public)

- removeFromBlacklist(address[]) (public)

- swapAndLiquify(uint256) (internal)

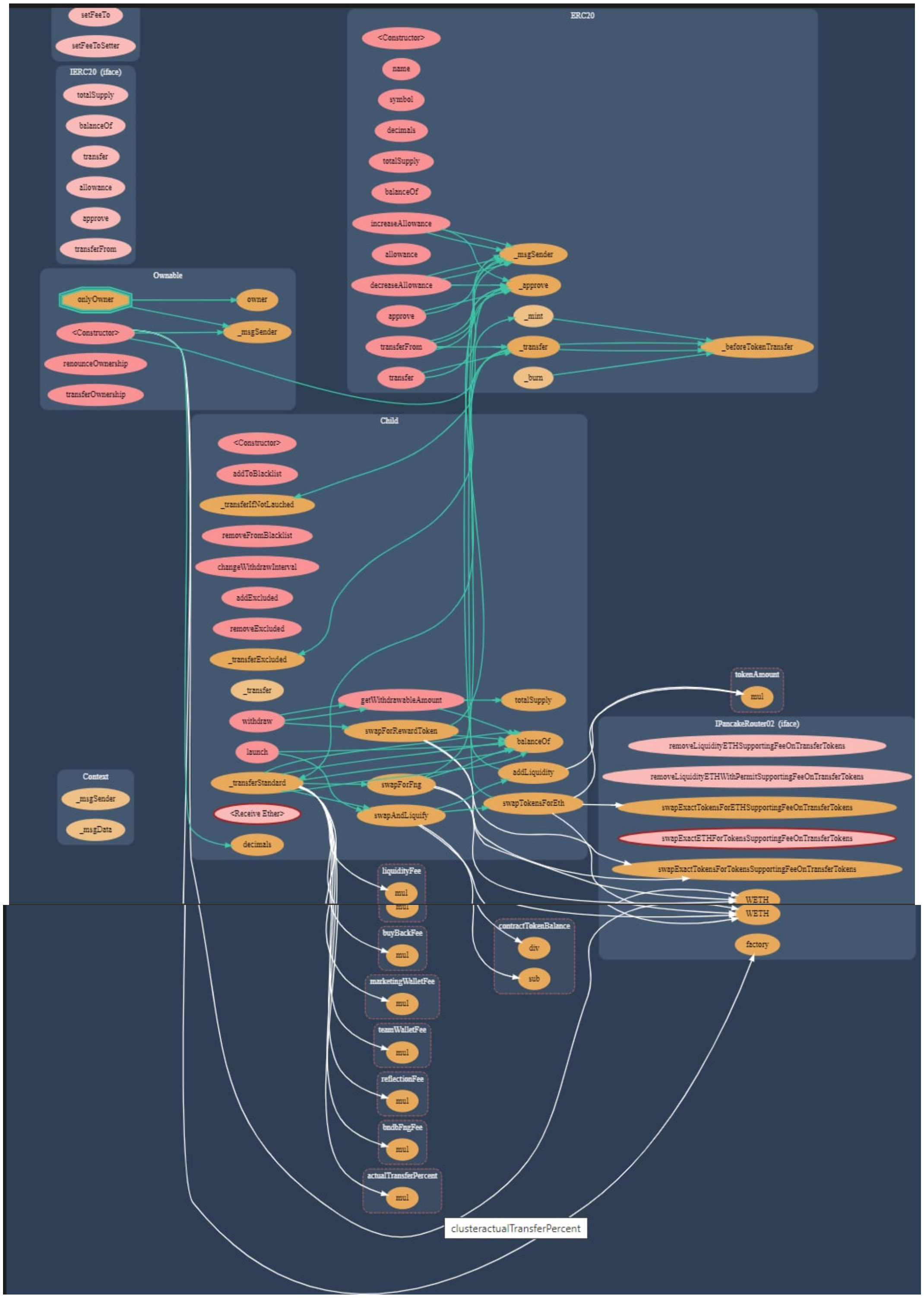
- swapForFng(uint256,address) (internal)

- swapForRewardToken(uint256,address) (internal)

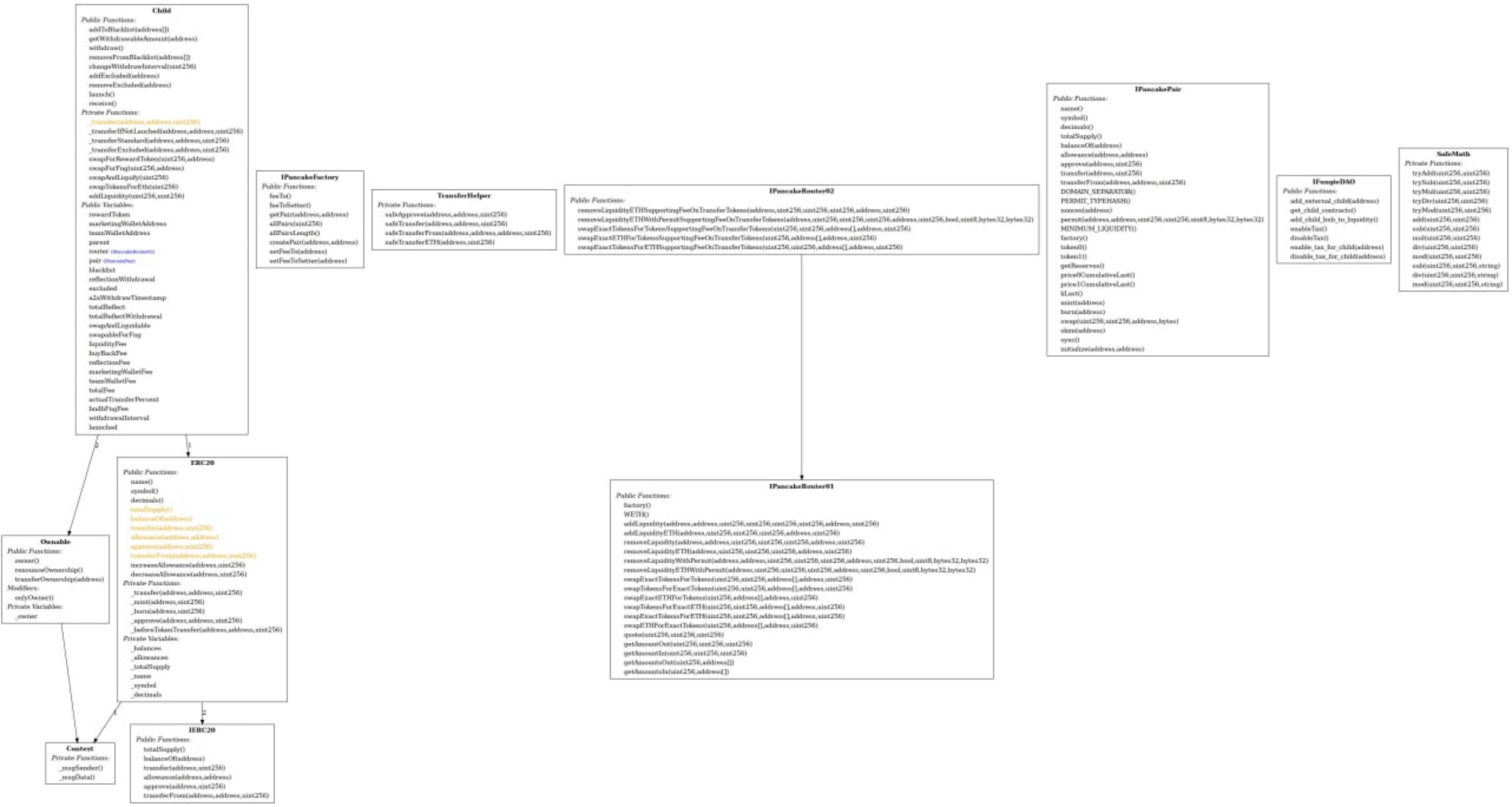
- swapTokensForEth(uint256) (internal)

- withdraw() (public)

Child.Sol.Call.Graph



Child.Sol.Inheritance.Graph.


Child
Public Functions:
addToBlacklist(address[])
getWithdrawableAmount(address)
withdraw()
removeFromBlacklist(address[])
changeWithdrawInterval(uint256)
addExcluded(address)
removeExcluded(address)
launch()
receive()
Private Functions:
_transfer(address,address,uint256)
_transferIfNotLauched(address,address,uint256)
_transferStandard(address,address,uint256)
_transferExcluded(address,address,uint256)
swapForRewardToken(uint256,address)
swapForFog(uint256,address)
swapAndLiquify(uint256)
swapTokensForEth(uint256)
addLiquidity(uint256,uint256)
Public Variables:
rewardToken
marketingWalletAddress
teamWalletAddress
parent
router (IPancakeRouter02)
pair (IPancakePair)
blacklist
reflectionWithdrawal
excluded
a2aWithdrawTimestamp
totalReflect
totalReflectWithdrawal
swapAndLiquidable
swapableForFog
liquidityFee
buyBackFee
reflectionFee
marketingWalletFee
teamWalletFee
totalFee
actualTransferPercent
bndbFogFee
withdrawalInterval
launched
2
1
ERC20
Public Functions:
name()
symbol()
decimals()
totalSupply()
balanceOf(address)
transfer(address,uint256)
allowance(address,address)
approve(address,uint256)
transferFrom(address,address,uint256)
increaseAllowance(address,uint256)
decreaseAllowance(address,uint256)
Private Functions:
_transfer(address,address,uint256)
_mint(address,uint256)
_burn(address,uint256)
_approve(address,address,uint256)
_beforeTokenTransfer(address,address,uint256)
Private Variables:
_balances
_allowances
_totalSupply
_name
_symbol
_decimals
Ownable
Public Functions:
owner()
renounceOwnership()
transferOwnership(address)
Modifiers:
onlyOwner()
Private Variables:
_owner
1
2
Context
Private Functions:
_msgSender()
_msgData()
IERC20
Public Functions:
totalSupply()
balanceOf(address)
transfer(address,uint256)
allowance(address,address)
approve(address,uint256)
transferFrom(address,address,uint256)
IPancakeFactory
Public Functions:
feeTo()
feeToSetter()
getPair(address,address)
allPairs(uint256)
allPairsLength()
createPair(address,address)
setFeeTo(address)
setFeeToSetter(address)
TransferHelper
Private Functions:
safeApprove(address,address,uint256)
safeTransfer(address,address,uint256)
safeTransferFrom(address,address,address,uint256)
safeTransferETH(address,uint256)
IPancakeRouter02
Public Functions:
removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,uint256)
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32)
swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint256)
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
IPancakeRouter01
Public Functions:
factory()
WETH()
addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256)
addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
removeLiquidity(address,address,uint256,uint256,uint256,address,uint256)
removeLiquidityETH(address,uint256,uint256,uint256,address,uint256)
removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32)
removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,bytes32)
swapExactTokensForTokens(uint256,uint256,address[],address,uint256)
swapTokensForExactTokens(uint256,uint256,address[],address,uint256)
swapExactETHForTokens(uint256,address[],address,uint256)
swapTokensForExactETH(uint256,uint256,address[],address,uint256)
swapExactTokensForETH(uint256,uint256,address[],address,uint256)
swapETHForExactTokens(uint256,address[],address,uint256)
quote(uint256,uint256,uint256)
getAmountOut(uint256,uint256,uint256)
getAmountIn(uint256,uint256,uint256)
getAmountsOut(uint256,address[])
getAmountsIn(uint256,address[])
IPancakePair
Public Functions:
name()
symbol()
decimals()
totalSupply()
balanceOf(address)
allowance(address,address)
approve(address,uint256)
transfer(address,uint256)
transferFrom(address,address,uint256)
DOMAIN_SEPARATOR()
PERMIT_TYPEHASH()
nonces(address)
permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
MINIMUM_LIQUIDITY()
factory()
token0()
token1()
getReserves()
price0CumulativeLast()
price1CumulativeLast()
kLast()
mint(address)
burn(address)
swap(uint256,uint256,address,bytes)
skim(address)
sync()
initialize(address,address)
IFungieDAO
Public Functions:
add_external_child(address)
get_child_contracts()
add_child_bnb_to_liquidity()
enableTax()
disableTax()
enable_tax_for_child(address)
disable_tax_for_child(address)
SafeMath
Private Functions:
tryAdd(uint256,uint256)
trySub(uint256,uint256)
tryMul(uint256,uint256)
tryDiv(uint256,uint256)
tryMod(uint256,uint256)
add(uint256,uint256)
sub(uint256,uint256)
mul(uint256,uint256)
div(uint256,uint256)
mod(uint256,uint256)
sub(uint256,uint256,string)
div(uint256,uint256,string)
mod(uint256,uint256,string)

High Severity Issues

No High Severity Issues found.

Moderate Severity Issues.

No Moderate Severity Issues found.

Low Severity Issues

No Low Severity Issues found.

Conclusion

All high and medium severity issues fixed in past work with the FungieDao Development Team.

ApeAudits note:
Please check the disclaimer above and note - the report is provided ‘as-is’ and makes no statements or warranties whatsoever. The report is provided only for the contract(s) mentioned in the report.