Smart contracts security assessment

Final report

Tariff: Standard

Crypt Finance

April 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for Crypt Finance.

The contracts allow users to stake their tokens in different strategies. The strategies are represented by MasterChef contracts. All earned rewards swaps to staked tokens and users receive interest nominated in staked tokens. Some amount of the rewards can be withheld as a commission.

The code is available at the @ConvolutedSolutions/cryptfinance-autocompounderp GitHub repository and was audited after the commit 06d14371.

Name	Crypt Finance
Audit date	2022-04-13 - 2022-04-15
Language	Solidity
Platform	Fantom Network

Contracts checked

Name	Address
StrategyManager	
VaultStrategy	

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	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Outdated Compiler Version	passed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

No issues were found

Low severity issues

1. Gas optimization (StrategyManager)

Visibility of the functions `add()`, `userStakedTokens()` can be declared as '`external`' to save gas.

2. Floating pragma (StrategyManager)

The general recommendation is that pragma should be fixed to the version you intend to deploy your contracts with. This helps to avoid deploying using an outdated compiler version and shields from possible bugs in future solidity releases.

3. Floating pragma (VaultStrategy)

The general recommendation is that pragma should be fixed to the version you intend to deploy your contracts with. This helps to avoid deploying using an outdated compiler version and shields from possible bugs in future solidity releases.

4. Gas optimization (VaultStrategy)

1. The contract uses `initialize()` function as a contract constructor. In the case of using constructor() directly, some of the variables could be declared as immutable to save gas.

2. Visibility of the function `numBurnTokens()` can be declared as '`external`' to save gas.

Conclusion

Crypt Finance StrategyManager, VaultStrategy contracts were audited. 4 low severity issues were found.

We strongly recommend writing unit tests to have extensive coverage of the codebase minimize the possibility of bugs and ensure that everything works as expected.

The contracts are dependent on the owner's account. Users interacting with the contracts have to trust the owner.

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.

Slither output

```
VaultStrategy.wrapNative() (contracts/VaultStrategy.sol#143-149) sends eth to arbitrary
user
        Dangerous calls:
        - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/VaultStrategy.sol#146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#functions-that-
send-ether-to-arbitrary-destinations

VaultStrategy._distributeFees(uint256,address) (contracts/VaultStrategy.sol#427-453)
performs a multiplication on the result of a division:
        -performanceFee = (_amount * strategyManager.performanceFee()) / 10_000
(contracts/VaultStrategy.sol#431)
        -bountyReward = (performanceFee * bountyRewardPct) / 10_000 (contracts/
VaultStrategy.sol#433)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-
multiply

VaultStrategy._burnEarnTokens(uint256) (contracts/VaultStrategy.sol#455-476) uses a
dangerous strict equality:
        - totalBurnTokenWghts == 0 || _amount == 0 (contracts/VaultStrategy.sol#458)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-
strict-equalities

Reentrancy in StrategyManager._withdraw(address,address,uint256,uint256) (contracts/
StrategyManager.sol#250-278):
        External calls:
        - _protocolEarn(strategy) (contracts/StrategyManager.sol#263)
                - _strategy.earn(address(0)) (contracts/StrategyManager.sol#306)
        - sharesRemoved = strategy.withdraw(_withdrawAmount,_to) (contracts/
StrategyManager.sol#270)
        State variables written after the call(s):
        - userPoolShares[_pid][_user] = userShares (contracts/StrategyManager.sol#272)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1

StrategyManager.earnMany(uint256[]).i (contracts/StrategyManager.sol#299) is a local
variable never initialized
StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[]).i (contracts/
```

```
StrategyManager.sol#168) is a local variable never initialized
VaultStrategy.earn(address).i (contracts/VaultStrategy.sol#257) is a local variable
never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-
local-variables

StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[]) (contracts/
StrategyManager.sol#161-174) ignores return value by
IERC20(_extraEarnTokens[i]).balanceOf(address(this)) (contracts/
StrategyManager.sol#169)
StrategyManager._deposit(uint256,uint256,address) (contracts/
StrategyManager.sol#208-231) ignores return value by userStakedPools[_for].add(_pid)
(contracts/StrategyManager.sol#228)
StrategyManager._withdraw(address,address,uint256,uint256) (contracts/
StrategyManager.sol#250-278) ignores return value by
userStakedPools[_user].remove(_pid) (contracts/StrategyManager.sol#275)
StrategyManager._protocolEarn(IVaultStrategy) (contracts/StrategyManager.sol#305-307)
ignores return value by _strategy.earn(address(0)) (contracts/StrategyManager.sol#306)
VaultStrategy.earn(address) (contracts/VaultStrategy.sol#246-312) ignores return value
by swapRouter.addLiquidity(address(token0),address(token1),token0Amt,token1Amt,0,0,addre
ss(this),block.timestamp) (contracts/VaultStrategy.sol#297-306)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

VaultStrategy.addBurnToken(address,uint256,address,address[],address[]) (contracts/
VaultStrategy.sol#315-329) should emit an event for:
        - totalBurnTokenWghts += _weight (contracts/VaultStrategy.sol#324)
VaultStrategy.setBurnToken(address,uint256,address,uint256) (contracts/
VaultStrategy.sol#341-351) should emit an event for:
        - totalBurnTokenWghts -= burnTokens[_index].weight (contracts/
VaultStrategy.sol#348)
        - totalBurnTokenWghts += _weight (contracts/VaultStrategy.sol#350)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-
arithmetic

StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[]) (contracts/
StrategyManager.sol#161-174) has external calls inside a loop:
IERC20(_extraEarnTokens[i]).balanceOf(address(this)) (contracts/
StrategyManager.sol#169)
StrategyManager._earn(uint256) (contracts/StrategyManager.sol#289-292) has external
calls inside a loop: bountyRewarded = poolInfo[_pid].strategy.earn(msg.sender)
(contracts/StrategyManager.sol#290)
```

```
VaultStrategy.wrapNative() (contracts/VaultStrategy.sol#143-149) has external calls
inside a loop: IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
VaultStrategy.tokenToEarn(address) (contracts/VaultStrategy.sol#416-425) has external
calls inside a loop: amount = IERC20(_token).balanceOf(address(this)) (contracts/
VaultStrategy.sol#419)
VaultStrategy._safeSwap(uint256,address,address,address,bool) (contracts/
VaultStrategy.sol#393-413) has external calls inside a loop: swapRouter.swapExactTokensF
orTokensSupportingFeeOnTransferTokens(_amountIn,0,path,_to,block.timestamp + 40)
(contracts/VaultStrategy.sol#406-408)
VaultStrategy._safeSwap(uint256,address,address,address,bool) (contracts/
VaultStrategy.sol#393-413) has external calls inside a loop: swapRouter.swapExactTokensF
orTokensSupportingFeeOnTransferTokens(_amountIn,0,path,_to,block.timestamp + 40)
(contracts/VaultStrategy.sol#410)
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/dependencies/
Address.sol#122-133) has external calls inside a loop: (success,returndata) =
target.call{value: value}(data) (contracts/dependencies/Address.sol#131)
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (contracts/dependencies/
SafeERC20.sol#59-66) has external calls inside a loop: newAllowance =
token.allowance(address(this),spender) + value (contracts/dependencies/
SafeERC20.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-
a-loop

Reentrancy in StrategyManager._deposit(uint256,uint256,address) (contracts/
StrategyManager.sol#208-231):
        External calls:
        - _protocolEarn(pool.strategy) (contracts/StrategyManager.sol#217)
                - _strategy.earn(address(0)) (contracts/StrategyManager.sol#306)
        - pool.stakeToken.safeTransferFrom(address(msg.sender),address(pool.strategy),_d
epositAmount) (contracts/StrategyManager.sol#221)
        - sharesAdded = pool.strategy.deposit(_depositAmount) (contracts/
StrategyManager.sol#225)
        State variables written after the call(s):
        - userPoolShares[_pid][_for] = userShares + sharesAdded (contracts/
StrategyManager.sol#227)
Reentrancy in VaultStrategy.deposit(uint256) (contracts/VaultStrategy.sol#197-217):
        External calls:
        - _farm() (contracts/VaultStrategy.sol#203)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
```

```
                - stakeToken.safeIncreaseAllowance(address(masterChef),amount)
(contracts/VaultStrategy.sol#153)
                - masterChef.deposit(pid,amount) (contracts/VaultStrategy.sol#154)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        External calls sending eth:
        - _farm() (contracts/VaultStrategy.sol#203)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        State variables written after the call(s):
        - sharesTotal = sharesTotal + sharesAdded (contracts/VaultStrategy.sol#214)
Reentrancy in VaultStrategy.emergencyWithdraw() (contracts/VaultStrategy.sol#187-192):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#187)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        State variables written after the call(s):
        - _pause() (contracts/VaultStrategy.sol#188)
                - _paused = true (contracts/dependencies/Pausable.sol#76)
        - emergencyWithdrawn = true (contracts/VaultStrategy.sol#189)
Reentrancy in VaultStrategy.pause() (contracts/VaultStrategy.sol#131-134):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#131)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        State variables written after the call(s):
        - _pause() (contracts/VaultStrategy.sol#132)
                - _paused = true (contracts/dependencies/Pausable.sol#76)
Reentrancy in VaultStrategy.unpause() (contracts/VaultStrategy.sol#136-140):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#136)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        State variables written after the call(s):
        - _unpause() (contracts/VaultStrategy.sol#138)
                - _paused = false (contracts/dependencies/Pausable.sol#88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2

Reentrancy in StrategyManager._deposit(uint256,uint256,address) (contracts/
StrategyManager.sol#208-231):
```

```
        External calls:
        - _protocolEarn(pool.strategy) (contracts/StrategyManager.sol#217)
                - _strategy.earn(address(0)) (contracts/StrategyManager.sol#306)
        - pool.stakeToken.safeTransferFrom(address(msg.sender),address(pool.strategy),_d
epositAmount) (contracts/StrategyManager.sol#221)
        - sharesAdded = pool.strategy.deposit(_depositAmount) (contracts/
StrategyManager.sol#225)
        Event emitted after the call(s):
        - Deposit(_for,_pid,_depositAmount) (contracts/StrategyManager.sol#230)
Reentrancy in StrategyManager._earn(uint256) (contracts/StrategyManager.sol#289-292):
        External calls:
        - bountyRewarded = poolInfo[_pid].strategy.earn(msg.sender) (contracts/
StrategyManager.sol#290)
        Event emitted after the call(s):
        - Earn(msg.sender,_pid,bountyRewarded) (contracts/StrategyManager.sol#291)
Reentrancy in StrategyManager._withdraw(address,address,uint256,uint256) (contracts/
StrategyManager.sol#250-278):
        External calls:
        - _protocolEarn(strategy) (contracts/StrategyManager.sol#263)
                - _strategy.earn(address(0)) (contracts/StrategyManager.sol#306)
        - sharesRemoved = strategy.withdraw(_withdrawAmount,_to) (contracts/
StrategyManager.sol#270)
        Event emitted after the call(s):
        - Withdraw(_user,_to,_pid,_withdrawAmount) (contracts/StrategyManager.sol#277)
Reentrancy in StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,addres
s,address[],address[]) (contracts/StrategyManager.sol#176-186):
        External calls:
        -
_strategy.addBurnToken(_token,_weight,_burnAddress,_earnToBurnPath,_burnToEarnPath)
(contracts/StrategyManager.sol#184)
        Event emitted after the call(s):
        - AddStrategyBurnToken(_strategy,_token,_weight,_burnAddress) (contracts/
StrategyManager.sol#185)
Reentrancy in VaultStrategy.earn(address) (contracts/VaultStrategy.sol#246-312):
        External calls:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - stakeToken.safeIncreaseAllowance(address(masterChef),amount)
(contracts/VaultStrategy.sol#153)
                - masterChef.deposit(pid,amount) (contracts/VaultStrategy.sol#154)
```

```
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        External calls sending eth:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        Event emitted after the call(s):
        - WrapNative() (contracts/VaultStrategy.sol#147)
                - wrapNative() (contracts/VaultStrategy.sol#256)
Reentrancy in VaultStrategy.earn(address) (contracts/VaultStrategy.sol#246-312):
        External calls:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - stakeToken.safeIncreaseAllowance(address(masterChef),amount)
(contracts/VaultStrategy.sol#153)
                - masterChef.deposit(pid,amount) (contracts/VaultStrategy.sol#154)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - wrapNative() (contracts/VaultStrategy.sol#258)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        External calls sending eth:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - wrapNative() (contracts/VaultStrategy.sol#258)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
```

```
        Event emitted after the call(s):
        - WrapNative() (contracts/VaultStrategy.sol#147)
                - wrapNative() (contracts/VaultStrategy.sol#258)
Reentrancy in VaultStrategy.earn(address) (contracts/VaultStrategy.sol#246-312):
        External calls:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - stakeToken.safeIncreaseAllowance(address(masterChef),amount)
(contracts/VaultStrategy.sol#153)
                - masterChef.deposit(pid,amount) (contracts/VaultStrategy.sol#154)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - wrapNative() (contracts/VaultStrategy.sol#258)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - tokenToEarn(extraEarnTokens[i]) (contracts/VaultStrategy.sol#259)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - IERC20(path[0]).safeIncreaseAllowance(address(swapRouter),_amountIn)
(contracts/VaultStrategy.sol#404)
                - swapRouter.swapExactTokensForTokensSupportingFeeOnTransferTokens(_amou
ntIn,0,path,_to,block.timestamp + 40) (contracts/VaultStrategy.sol#406-408)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
                - swapRouter.swapExactTokensForTokensSupportingFeeOnTransferTokens(_amou
ntIn,0,path,_to,block.timestamp + 40) (contracts/VaultStrategy.sol#410)
        External calls sending eth:
        - _farmHarvest() (contracts/VaultStrategy.sol#255)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        - wrapNative() (contracts/VaultStrategy.sol#256)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - wrapNative() (contracts/VaultStrategy.sol#258)
                - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/
VaultStrategy.sol#146)
        - tokenToEarn(extraEarnTokens[i]) (contracts/VaultStrategy.sol#259)
```

```
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        Event emitted after the call(s):
        - TokenToEarn(_token) (contracts/VaultStrategy.sol#423)
                - tokenToEarn(extraEarnTokens[i]) (contracts/VaultStrategy.sol#259)
Reentrancy in VaultStrategy.emergencyWithdraw() (contracts/VaultStrategy.sol#187-192):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#187)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        Event emitted after the call(s):
        - Paused(_msgSender()) (contracts/dependencies/Pausable.sol#77)
                - _pause() (contracts/VaultStrategy.sol#188)
Reentrancy in VaultStrategy.emergencyWithdraw() (contracts/VaultStrategy.sol#187-192):
        External calls:
        - _farmEmergencyWithdraw() (contracts/VaultStrategy.sol#190)
                - masterChef.emergencyWithdraw(pid) (contracts/VaultStrategy.sol#162)
        - onlyOperator() (contracts/VaultStrategy.sol#187)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        Event emitted after the call(s):
        - EmergencyWithdraw() (contracts/VaultStrategy.sol#191)
Reentrancy in VaultStrategy.farm() (contracts/VaultStrategy.sol#182-185):
        External calls:
        - _farm() (contracts/VaultStrategy.sol#183)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - stakeToken.safeIncreaseAllowance(address(masterChef),amount)
(contracts/VaultStrategy.sol#153)
                - masterChef.deposit(pid,amount) (contracts/VaultStrategy.sol#154)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        External calls sending eth:
        - _farm() (contracts/VaultStrategy.sol#183)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        Event emitted after the call(s):
        - Farm() (contracts/VaultStrategy.sol#184)
Reentrancy in VaultStrategy.pause() (contracts/VaultStrategy.sol#131-134):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#131)
```

```
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        Event emitted after the call(s):
        - Pause() (contracts/VaultStrategy.sol#133)
        - Paused(_msgSender()) (contracts/dependencies/Pausable.sol#77)
                - _pause() (contracts/VaultStrategy.sol#132)
Reentrancy in StrategyManager.removeStrategyBurnToken(IVaultStrategy,uint256)
(contracts/StrategyManager.sol#199-205):
        External calls:
        - _strategy.removeBurnToken(_index) (contracts/StrategyManager.sol#203)
        Event emitted after the call(s):
        - RemoveStrategyBurnToken(_strategy,_index) (contracts/StrategyManager.sol#204)
Reentrancy in StrategyManager.removeStrategySwapPath(IVaultStrategy,address,address)
(contracts/StrategyManager.sol#152-159):
        External calls:
        - _strategy.removeSwapPath(_token0,_token1) (contracts/StrategyManager.sol#157)
        Event emitted after the call(s):
        - RemoveStrategySwapPath(_strategy,_token0,_token1) (contracts/
StrategyManager.sol#158)
Reentrancy in
StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,uint256)
(contracts/StrategyManager.sol#188-197):
        External calls:
        - _strategy.setBurnToken(_token,_weight,_burnAddress,_index) (contracts/
StrategyManager.sol#195)
        Event emitted after the call(s):
        - SetStrategyBurnToken(_strategy,_token,_weight,_burnAddress,_index) (contracts/
StrategyManager.sol#196)
Reentrancy in StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[])
(contracts/StrategyManager.sol#161-174):
        External calls:
        - _strategy.setExtraEarnTokens(_extraEarnTokens) (contracts/
StrategyManager.sol#172)
        Event emitted after the call(s):
        - SetStrategyExtraEarnTokens(_strategy,_extraEarnTokens) (contracts/
StrategyManager.sol#173)
Reentrancy in StrategyManager.setStrategyRouter(IVaultStrategy,address) (contracts/
StrategyManager.sol#134-140):
        External calls:
        - _strategy.setSwapRouter(_router) (contracts/StrategyManager.sol#138)
        Event emitted after the call(s):
```

```
        - SetStrategyRouter(_strategy,_router) (contracts/StrategyManager.sol#139)
Reentrancy in
StrategyManager.setStrategySwapPath(IVaultStrategy,address,address,address[])
(contracts/StrategyManager.sol#142-150):
        External calls:
        - _strategy.setSwapPath(_token0,_token1,_path) (contracts/
StrategyManager.sol#148)
        Event emitted after the call(s):
        - SetStrategySwapPath(_strategy,_token0,_token1,_path) (contracts/
StrategyManager.sol#149)
Reentrancy in VaultStrategy.tokenToEarn(address) (contracts/VaultStrategy.sol#416-425):
        External calls:
        - _safeSwap(amount,_token,address(earnToken),address(this),true) (contracts/
VaultStrategy.sol#422)
                - returndata = address(token).functionCall(data,SafeERC20: low-level
call failed) (contracts/dependencies/SafeERC20.sol#92)
                - IERC20(path[0]).safeIncreaseAllowance(address(swapRouter),_amountIn)
(contracts/VaultStrategy.sol#404)
                - swapRouter.swapExactTokensForTokensSupportingFeeOnTransferTokens(_amou
ntIn,0,path,_to,block.timestamp + 40) (contracts/VaultStrategy.sol#406-408)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
                - swapRouter.swapExactTokensForTokensSupportingFeeOnTransferTokens(_amou
ntIn,0,path,_to,block.timestamp + 40) (contracts/VaultStrategy.sol#410)
        External calls sending eth:
        - _safeSwap(amount,_token,address(earnToken),address(this),true) (contracts/
VaultStrategy.sol#422)
                - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
        Event emitted after the call(s):
        - TokenToEarn(_token) (contracts/VaultStrategy.sol#423)
Reentrancy in VaultStrategy.unpause() (contracts/VaultStrategy.sol#136-140):
        External calls:
        - onlyOperator() (contracts/VaultStrategy.sol#136)
                - require(bool,string)(strategyManager.operators(msg.sender),Error:
onlyOperator, NOT_ALLOWED) (contracts/VaultStrategy.sol#80)
        Event emitted after the call(s):
        - Unpause() (contracts/VaultStrategy.sol#139)
        - Unpaused(_msgSender()) (contracts/dependencies/Pausable.sol#89)
                - _unpause() (contracts/VaultStrategy.sol#138)
Reentrancy in VaultStrategy.wrapNative() (contracts/VaultStrategy.sol#143-149):
```

```
        External calls:
        - IWNATIVE(WNATIVE).deposit{value: balance}() (contracts/VaultStrategy.sol#146)
        Event emitted after the call(s):
        - WrapNative() (contracts/VaultStrategy.sol#147)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-3

Address.isContract(address) (contracts/dependencies/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/dependencies/Address.sol#32-34)
Address._verifyCallResult(bool,bytes,string) (contracts/dependencies/
Address.sol#189-209) uses assembly
        - INLINE ASM (contracts/dependencies/Address.sol#201-204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.5.0', '>=0.6.2', '^0.8.0']
        - ^0.8.0 (contracts/StrategyManager.sol#3)
        - ^0.8.0 (contracts/VaultStrategy.sol#3)
        - ^0.8.0 (contracts/dependencies/Address.sol#3)
        - ^0.8.0 (contracts/dependencies/Context.sol#3)
        - ^0.8.0 (contracts/dependencies/EnumerableSet.sol#3)
        - ^0.8.0 (contracts/dependencies/IERC20.sol#3)
        - >=0.5.0 (contracts/dependencies/IUniswapV2Factory.sol#3)
        - >=0.5.0 (contracts/dependencies/IUniswapV2Pair.sol#3)
        - >=0.6.2 (contracts/dependencies/IUniswapV2Router01.sol#3)
        - >=0.6.2 (contracts/dependencies/IUniswapV2Router02.sol#3)
        - ^0.8.0 (contracts/dependencies/Ownable.sol#3)
        - ^0.8.0 (contracts/dependencies/Pausable.sol#4)
        - ^0.8.0 (contracts/dependencies/ReentrancyGuard.sol#3)
        - ^0.8.0 (contracts/dependencies/SafeERC20.sol#3)
        - ^0.8.0 (contracts/interfaces/IFarm.sol#3)
        - ^0.8.0 (contracts/interfaces/IStrategyManager.sol#3)
        - ^0.8.0 (contracts/interfaces/IVaultStrategy.sol#3)
        - ^0.8.0 (contracts/interfaces/IWNATIVE.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-
pragma-directives-are-used

ReentrancyGuard.nonReentrant() (contracts/dependencies/ReentrancyGuard.sol#49-61) has
costly operations inside a loop:
        - _status = _ENTERED (contracts/dependencies/ReentrancyGuard.sol#54)
ReentrancyGuard.nonReentrant() (contracts/dependencies/ReentrancyGuard.sol#49-61) has
```

```
costly operations inside a loop:
        - _status = _NOT_ENTERED (contracts/dependencies/ReentrancyGuard.sol#60)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-
operations-inside-a-loop

Address.functionCall(address,bytes) (contracts/dependencies/Address.sol#79-81) is never
used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/dependencies/
Address.sol#108-114) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/dependencies/
Address.sol#168-170) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/dependencies/
Address.sol#178-187) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/dependencies/Address.sol#141-143)
is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/dependencies/
Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/dependencies/Address.sol#54-59) is never
used and should be removed
Context._msgData() (contracts/dependencies/Context.sol#20-22) is never used and should
be removed
EnumerableSet.add(EnumerableSet.AddressSet,address) (contracts/dependencies/
EnumerableSet.sol#199-201) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/dependencies/
EnumerableSet.sol#145-147) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (contracts/dependencies/
EnumerableSet.sol#237-239) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/dependencies/
EnumerableSet.sol#183-185) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/dependencies/
EnumerableSet.sol#216-218) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/dependencies/
EnumerableSet.sol#162-164) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/dependencies/
EnumerableSet.sol#270-272) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (contracts/dependencies/
EnumerableSet.sol#223-225) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/dependencies/
EnumerableSet.sol#169-171) is never used and should be removed
EnumerableSet.remove(EnumerableSet.AddressSet,address) (contracts/dependencies/
EnumerableSet.sol#209-211) is never used and should be removed
```

```
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/dependencies/
EnumerableSet.sol#155-157) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (contracts/dependencies/
SafeERC20.sol#44-57) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (contracts/dependencies/
SafeERC20.sol#68-79) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (contracts/StrategyManager.sol#3) allows old versions
Pragma version^0.8.0 (contracts/VaultStrategy.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/Address.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/Context.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/EnumerableSet.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/IERC20.sol#3) allows old versions
Pragma version>=0.5.0 (contracts/dependencies/IUniswapV2Factory.sol#3) allows old
versions
Pragma version>=0.5.0 (contracts/dependencies/IUniswapV2Pair.sol#3) allows old versions
Pragma version>=0.6.2 (contracts/dependencies/IUniswapV2Router01.sol#3) allows old
versions
Pragma version>=0.6.2 (contracts/dependencies/IUniswapV2Router02.sol#3) allows old
versions
Pragma version^0.8.0 (contracts/dependencies/Ownable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (contracts/dependencies/ReentrancyGuard.sol#3) allows old versions
Pragma version^0.8.0 (contracts/dependencies/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IFarm.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IStrategyManager.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IVaultStrategy.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IWNATIVE.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/dependencies/
Address.sol#54-59):
        - (success) = recipient.call{value: amount}() (contracts/dependencies/
Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(contracts/dependencies/Address.sol#122-133):
        - (success,returndata) = target.call{value: value}(data) (contracts/
dependencies/Address.sol#131)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/
```

```
dependencies/Address.sol#151-160):
        - (success,returndata) = target.staticcall(data) (contracts/dependencies/
Address.sol#158)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/
dependencies/Address.sol#178-187):
        - (success,returndata) = target.delegatecall(data) (contracts/dependencies/
Address.sol#185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-
calls

Parameter StrategyManager.setOperator(address,bool)._addr (contracts/
StrategyManager.sol#82) is not in mixedCase
Parameter StrategyManager.setOperator(address,bool)._isOperator (contracts/
StrategyManager.sol#82) is not in mixedCase
Parameter StrategyManager.setPerformanceFee(uint256)._performanceFee (contracts/
StrategyManager.sol#87) is not in mixedCase
Parameter StrategyManager.setPerformanceFeeBountyPct(uint256)._performanceFeeBountyPct
(contracts/StrategyManager.sol#93) is not in mixedCase
Parameter StrategyManager.userStakedPoolLength(address)._user (contracts/
StrategyManager.sol#100) is not in mixedCase
Parameter StrategyManager.userStakedPoolAt(address,uint256)._user (contracts/
StrategyManager.sol#104) is not in mixedCase
Parameter StrategyManager.userStakedPoolAt(address,uint256)._index (contracts/
StrategyManager.sol#104) is not in mixedCase
Parameter StrategyManager.userStakedTokens(address,uint256)._user (contracts/
StrategyManager.sol#108) is not in mixedCase
Parameter StrategyManager.userStakedTokens(address,uint256)._pid (contracts/
StrategyManager.sol#108) is not in mixedCase
Parameter StrategyManager.add(IVaultStrategy)._strategy (contracts/
StrategyManager.sol#124) is not in mixedCase
Parameter StrategyManager.setStrategyRouter(IVaultStrategy,address)._strategy
(contracts/StrategyManager.sol#135) is not in mixedCase
Parameter StrategyManager.setStrategyRouter(IVaultStrategy,address)._router (contracts/
StrategyManager.sol#136) is not in mixedCase
Parameter
StrategyManager.setStrategySwapPath(IVaultStrategy,address,address,address[])._strategy
(contracts/StrategyManager.sol#143) is not in mixedCase
Parameter
StrategyManager.setStrategySwapPath(IVaultStrategy,address,address,address[])._token0
(contracts/StrategyManager.sol#144) is not in mixedCase
Parameter
```

```
StrategyManager.setStrategySwapPath(IVaultStrategy,address,address,address[])._token1
(contracts/StrategyManager.sol#145) is not in mixedCase
Parameter
StrategyManager.setStrategySwapPath(IVaultStrategy,address,address,address[])._path
(contracts/StrategyManager.sol#146) is not in mixedCase
Parameter
StrategyManager.removeStrategySwapPath(IVaultStrategy,address,address)._strategy
(contracts/StrategyManager.sol#153) is not in mixedCase
Parameter
StrategyManager.removeStrategySwapPath(IVaultStrategy,address,address)._token0
(contracts/StrategyManager.sol#154) is not in mixedCase
Parameter
StrategyManager.removeStrategySwapPath(IVaultStrategy,address,address)._token1
(contracts/StrategyManager.sol#155) is not in mixedCase
Parameter
StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[])._strategy
(contracts/StrategyManager.sol#162) is not in mixedCase
Parameter
StrategyManager.setStrategyExtraEarnTokens(IVaultStrategy,address[])._extraEarnTokens
(contracts/StrategyManager.sol#163) is not in mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._strategy (contracts/StrategyManager.sol#177) is not in mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._token (contracts/StrategyManager.sol#178) is not in mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._weight (contracts/StrategyManager.sol#179) is not in mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._burnAddress (contracts/StrategyManager.sol#180) is not in mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._earnToBurnPath (contracts/StrategyManager.sol#181) is not in
mixedCase
Parameter StrategyManager.addStrategyBurnToken(IVaultStrategy,address,uint256,address,ad
dress[],address[])._burnToEarnPath (contracts/StrategyManager.sol#182) is not in
mixedCase
Parameter StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,ui
nt256)._strategy (contracts/StrategyManager.sol#189) is not in mixedCase
Parameter StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,ui
nt256)._token (contracts/StrategyManager.sol#190) is not in mixedCase
Parameter StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,ui
nt256)._weight (contracts/StrategyManager.sol#191) is not in mixedCase
Parameter StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,ui
```

```
nt256)._burnAddress (contracts/StrategyManager.sol#192) is not in mixedCase
Parameter StrategyManager.setStrategyBurnToken(IVaultStrategy,address,uint256,address,ui
nt256)._index (contracts/StrategyManager.sol#193) is not in mixedCase
Parameter StrategyManager.removeStrategyBurnToken(IVaultStrategy,uint256)._strategy
(contracts/StrategyManager.sol#200) is not in mixedCase
Parameter StrategyManager.removeStrategyBurnToken(IVaultStrategy,uint256)._index
(contracts/StrategyManager.sol#201) is not in mixedCase
Parameter StrategyManager.deposit(uint256,uint256)._pid (contracts/
StrategyManager.sol#234) is not in mixedCase
Parameter StrategyManager.deposit(uint256,uint256)._depositAmount (contracts/
StrategyManager.sol#235) is not in mixedCase
Parameter StrategyManager.depositFor(uint256,uint256,address)._pid (contracts/
StrategyManager.sol#241) is not in mixedCase
Parameter StrategyManager.depositFor(uint256,uint256,address)._depositAmount (contracts/
StrategyManager.sol#242) is not in mixedCase
Parameter StrategyManager.depositFor(uint256,uint256,address)._for (contracts/
StrategyManager.sol#243) is not in mixedCase
Parameter StrategyManager.withdraw(uint256,uint256)._pid (contracts/
StrategyManager.sol#280) is not in mixedCase
Parameter StrategyManager.withdraw(uint256,uint256)._withdrawAmount (contracts/
StrategyManager.sol#280) is not in mixedCase
Parameter StrategyManager.emergencyWithdraw(uint256)._pid (contracts/
StrategyManager.sol#284) is not in mixedCase
Parameter StrategyManager.earn(uint256)._pid (contracts/StrategyManager.sol#294) is not
in mixedCase
Parameter StrategyManager.earnMany(uint256[])._pids (contracts/StrategyManager.sol#298)
is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._pid (contracts/VaultStrategy.sol#87) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._isLpToken (contracts/VaultStrategy.sol#88) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._addresses (contracts/VaultStrategy.sol#89) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._earnToToken0Path (contracts/VaultStrategy.sol#90) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._earnToToken1Path (contracts/VaultStrategy.sol#91) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._token0ToEarnPath (contracts/VaultStrategy.sol#92) is not in mixedCase
Parameter VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[]
,address[])._token1ToEarnPath (contracts/VaultStrategy.sol#93) is not in mixedCase
```

```
Parameter VaultStrategy.deposit(uint256)._depositAmount (contracts/
VaultStrategy.sol#198) is not in mixedCase
Parameter VaultStrategy.withdraw(uint256,address)._withdrawAmount (contracts/
VaultStrategy.sol#220) is not in mixedCase
Parameter VaultStrategy.withdraw(uint256,address)._withdrawTo (contracts/
VaultStrategy.sol#221) is not in mixedCase
Parameter VaultStrategy.earn(address)._bountyHunter (contracts/VaultStrategy.sol#247)
is not in mixedCase
Parameter
VaultStrategy.addBurnToken(address,uint256,address,address[],address[])._token
(contracts/VaultStrategy.sol#316) is not in mixedCase
Parameter
VaultStrategy.addBurnToken(address,uint256,address,address[],address[])._weight
(contracts/VaultStrategy.sol#317) is not in mixedCase
Parameter
VaultStrategy.addBurnToken(address,uint256,address,address[],address[])._burnAddress
(contracts/VaultStrategy.sol#318) is not in mixedCase
Parameter
VaultStrategy.addBurnToken(address,uint256,address,address[],address[])._earnToBurnPath
(contracts/VaultStrategy.sol#319) is not in mixedCase
Parameter
VaultStrategy.addBurnToken(address,uint256,address,address[],address[])._burnToEarnPath
(contracts/VaultStrategy.sol#320) is not in mixedCase
Parameter VaultStrategy.removeBurnToken(uint256)._index (contracts/
VaultStrategy.sol#332) is not in mixedCase
Parameter VaultStrategy.setBurnToken(address,uint256,address,uint256)._token (contracts/
VaultStrategy.sol#342) is not in mixedCase
Parameter VaultStrategy.setBurnToken(address,uint256,address,uint256)._weight
(contracts/VaultStrategy.sol#343) is not in mixedCase
Parameter VaultStrategy.setBurnToken(address,uint256,address,uint256)._burnAddress
(contracts/VaultStrategy.sol#344) is not in mixedCase
Parameter VaultStrategy.setBurnToken(address,uint256,address,uint256)._index (contracts/
VaultStrategy.sol#345) is not in mixedCase
Parameter VaultStrategy.setSwapRouter(address)._router (contracts/
VaultStrategy.sol#354) is not in mixedCase
Parameter VaultStrategy.setExtraEarnTokens(address[])._extraEarnTokens (contracts/
VaultStrategy.sol#360) is not in mixedCase
Parameter VaultStrategy.setSwapPath(address,address,address[])._token0 (contracts/
VaultStrategy.sol#378) is not in mixedCase
Parameter VaultStrategy.setSwapPath(address,address,address[])._token1 (contracts/
VaultStrategy.sol#379) is not in mixedCase
```

```
Parameter VaultStrategy.setSwapPath(address,address,address[])._path (contracts/
VaultStrategy.sol#380) is not in mixedCase
Parameter VaultStrategy.removeSwapPath(address,address)._token0 (contracts/
VaultStrategy.sol#386) is not in mixedCase
Parameter VaultStrategy.removeSwapPath(address,address)._token1 (contracts/
VaultStrategy.sol#387) is not in mixedCase
Parameter VaultStrategy.tokenToEarn(address)._token (contracts/VaultStrategy.sol#417)
is not in mixedCase
Variable VaultStrategy.WNATIVE (contracts/VaultStrategy.sol#53) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/dependencies/
IUniswapV2Pair.sol#20) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/dependencies/
IUniswapV2Pair.sol#21) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/dependencies/
IUniswapV2Pair.sol#38) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/dependencies/IUniswapV2Router01.sol#7) is
not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions

Variable VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[],
address[])._earnToToken0Path (contracts/VaultStrategy.sol#90) is too similar to VaultStr
ategy.initialize(uint256,bool,address[6],address[],address[],address[],address[])._earnT
oToken1Path (contracts/VaultStrategy.sol#91)
Variable VaultStrategy.initialize(uint256,bool,address[6],address[],address[],address[],
address[])._token0ToEarnPath (contracts/VaultStrategy.sol#92) is too similar to VaultStr
ategy.initialize(uint256,bool,address[6],address[],address[],address[],address[])._token
1ToEarnPath (contracts/VaultStrategy.sol#93)
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256
,address,uint256).amountADesired (contracts/dependencies/IUniswapV2Router01.sol#12) is
too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,u
int256,address,uint256).amountBDesired (contracts/dependencies/
IUniswapV2Router01.sol#13)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-
are-too-similar

userStakedTokens(address,uint256) should be declared external:
        - StrategyManager.userStakedTokens(address,uint256) (contracts/
StrategyManager.sol#108-116)
add(IVaultStrategy) should be declared external:
        - StrategyManager.add(IVaultStrategy) (contracts/StrategyManager.sol#123-131)
```

```
numBurnTokens() should be declared external:
        - VaultStrategy.numBurnTokens() (contracts/VaultStrategy.sol#479-481)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (contracts/dependencies/Ownable.sol#53-55)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (contracts/dependencies/Ownable.sol#61-64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-
function-that-could-be-declared-external
analyzed (24 contracts with 77 detectors), 236 result(s) found
```

0x Guard