0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

Hamster-money

March 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

This report has been prepared for the Hamster-money team upon their request.

The audited project is a fork of the Tomb Finance Project.

Further details about Hamster-money are available at the official website: https://hamster.money/

Name	Hamster-money
Audit date	2022-03-16 - 2022-03-17
Language	Solidity
Platform	Fantom Network

Contracts checked

Name	Address
Hamster	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/Hamster.sol
HamsterGenesisRewardPool	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/distribution/HamsterGenesisRewardPool.sol
HamsterRewardPool	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/distribution/HamsterRewardPool.sol

HShareRewardPool	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/distribution/HShareRewardPool.sol
Oracle	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/Oracle.sol
TaxOracle	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/TaxOracle.sol
TaxOfficeV2	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/TaxOfficeV2.sol
Treasury	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/Treasury.sol
HBond	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/HBond.sol
HShare	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/HShare.sol
HamsterWheel	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/HamsterWheel.sol

HamsterZapper	https://github.com/hamster-money/hamster-contracts/blob/ee153c18241a0a8ff23d021e58cf318f5a849f3f/contracts/HamsterZapper.sol

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

- Comparing the project to the Tomb Finance implementation

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

1. Tax bypass (RESPONDED) (Hamster)

Tax avoidance in the Tomb project is the main problem the team faced. The problem is that there is an invariant in the `transferFrom()` function that deducts tax for the transfer of tokens, but there is also an invariant without deduction of tax that calls the `transfer()` function. With the help of this problem, you can bypass all tax deductions if you use only the `transfer()` function and it is possible to violate the tokenomics of the project.

Recommendation: It is recommended to overload the `transfer()` function to work with tax or completely remove the tax functionality in contracts.

Hamster team response: Hamster money developers have been aware of the potential tax issue on Tomb forks. Given its full implications, the decision was made to leave the tax office functionality within the code intact and provide a user interface that allows users to add liquidity to the HAM/FTM pair, in line with the original tomb.finance solution. This way, if the protocol requires a tax to be used, the protocol will leave the liquidity operations tax free, at the same time allowing all users participate in the process.

Medium severity issues

1. Commision tokens (RESPONDED) (HamsterGenesisRewardPool)

In 168-171L when transferring a commission token, the same commission is charged. Tokens from the `_comissionTokens` set may have different commissions or change them, so the calculation of `user.amount` may be violated.

Recommendation: It is recommended to compare the balance of the token before and after the execution of the `pool.token.safeTransferFrom()` function, thus you will find out how much is spent on the commission.

Hamster team response: The Genesis pools worked as planned and no complaints have been made. The 1% commission was charged only for USDC and WFTM deposits, and was not applied to any other pool. The combination of these measures ensured the genesis period was concluded without any issues.

Low severity issues

No issues were found

Conclusion

The Hamster-money Project was compared with the Tomb Project. Hamster-money has changed the implementation of HamsterGenesisRewardPool contract.

A list of tokens with commissions has been added to the HamsterGenesisRewardPool contract, a commission will be charged when depositing these tokens. Also in this contract, the maxDeposit field has been added to the PoolInfo structure, which limits the maximum size of the pool.

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

```
UniswapV2OracleLibrary.currentBlockTimestamp() (contracts/lib/
UniswapV2OracleLibrary.sol#13-15) uses a weak PRNG: "uint32(block.timestamp % 2 ** 32)
(contracts/lib/UniswapV2OracleLibrary.sol#14)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

HShare.governanceRecoverUnsupported(IERC20,uint256,address) (contracts/
HShare.sol#84-90) ignores return value by _token.transfer(_to,_amount) (contracts/
HShare.sol#89)
Hamster.governanceRecoverUnsupported(IERC20,uint256,address) (contracts/
Hamster.sol#117-123) ignores return value by _token.transfer(_to,_amount) (contracts/
Hamster.sol#122)
TaxOfficeV2.addLiquidityETHTaxFree(uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#29-62) ignores return value by
IERC20(hamster).transferFrom(msg.sender,address(this),amtHamster) (contracts/
TaxOfficeV2.sol#47)
TaxOfficeV2.addLiquidityETHTaxFree(uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#29-62) ignores return value by
IERC20(hamster).transfer(msg.sender,amtHamster.sub(resultAmtHamster)) (contracts/
TaxOfficeV2.sol#59)
TaxOfficeV2.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#64-105) ignores return value by
IERC20(hamster).transferFrom(msg.sender,address(this),amtHamster) (contracts/
TaxOfficeV2.sol#83)
TaxOfficeV2.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#64-105) ignores return value by
IERC20(token).transferFrom(msg.sender,address(this),amtToken) (contracts/
TaxOfficeV2.sol#84)
TaxOfficeV2.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#64-105) ignores return value by
IERC20(hamster).transfer(msg.sender,amtHamster.sub(resultAmtHamster)) (contracts/
TaxOfficeV2.sol#99)
TaxOfficeV2.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#64-105) ignores return value by
IERC20(token).transfer(msg.sender,amtToken.sub(resultAmtToken)) (contracts/
TaxOfficeV2.sol#102)
TaxOfficeV2.taxFreeTransferFrom(address,address,uint256) (contracts/
TaxOfficeV2.sol#151-160) ignores return value by
```

```
IERC20(hamster).transferFrom(_sender,_recipient,_amt) (contracts/TaxOfficeV2.sol#158)
Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492) ignores return
value by IERC20(hamster).transfer(daoFund,_daoFundSharedAmount) (contracts/
Treasury.sol#478)
Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492) ignores return
value by IERC20(hamster).transfer(devFund,_devFundSharedAmount) (contracts/
Treasury.sol#484)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-
transfer

Treasury.allocateSeigniorage() (contracts/Treasury.sol#158-194) performs a
multiplication on the result of a division:
        -_seigniorage = hamsterSupply.mul(_percentage).div(1e18) (contracts/
Treasury.sol#177)
        -_savedForHamsterWheel =
_seigniorage.mul(seigniorageExpansionFloorPercent).div(10000) (contracts/
Treasury.sol#178)
HShareRewardPool.pendingShare(uint256,address) (contracts/distribution/
HShareRewardPool.sol#36-47) performs a multiplication on the result of a division:
        -_hshareReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HShareRewardPool.sol#43)
        -accHSharePerShare =
accHSharePerShare.add(_hshareReward.mul(1e18).div(tokenSupply)) (contracts/distribution/
HShareRewardPool.sol#44)
HShareRewardPool.updatePool(uint256) (contracts/distribution/
HShareRewardPool.sol#214-234) performs a multiplication on the result of a division:
        -_hshareReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HShareRewardPool.sol#230)
        -pool.accHSharePerShare =
pool.accHSharePerShare.add(_hshareReward.mul(1e18).div(tokenSupply)) (contracts/
distribution/HShareRewardPool.sol#231)
HamsterGenesisRewardPool.pendingHAMSTER(uint256,address) (contracts/distribution/
HamsterGenesisRewardPool.sol#51-62) performs a multiplication on the result of a
division:
        -_hamsterReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HamsterGenesisRewardPool.sol#58)
        -accHamsterPerShare =
accHamsterPerShare.add(_hamsterReward.mul(1e18).div(tokenSupply)) (contracts/
distribution/HamsterGenesisRewardPool.sol#59)
HamsterGenesisRewardPool.updatePool(uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#242-262) performs a multiplication on the result of a
division:
        -_hamsterReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HamsterGenesisRewardPool.sol#258)
        -pool.accHamsterPerShare =
pool.accHamsterPerShare.add(_hamsterReward.mul(1e18).div(tokenSupply)) (contracts/
distribution/HamsterGenesisRewardPool.sol#259)
HamsterRewardPool.pendingHAMSTER(uint256,address) (contracts/distribution/
HamsterRewardPool.sol#34-45) performs a multiplication on the result of a division:
        -_hamsterReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HamsterRewardPool.sol#41)
        -accHamsterPerShare =
accHamsterPerShare.add(_hamsterReward.mul(1e18).div(tokenSupply)) (contracts/
distribution/HamsterRewardPool.sol#42)
HamsterRewardPool.updatePool(uint256) (contracts/distribution/
HamsterRewardPool.sol#234-254) performs a multiplication on the result of a division:
        -_hamsterReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)
(contracts/distribution/HamsterRewardPool.sol#250)
        -pool.accHamsterPerShare =
pool.accHamsterPerShare.add(_hamsterReward.mul(1e18).div(tokenSupply)) (contracts/
distribution/HamsterRewardPool.sol#251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-
multiply

HShareRewardPool.updatePool(uint256) (contracts/distribution/
HShareRewardPool.sol#214-234) uses a dangerous strict equality:
        - tokenSupply == 0 (contracts/distribution/HShareRewardPool.sol#220)
HamsterGenesisRewardPool.updatePool(uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#242-262) uses a dangerous strict equality:
        - tokenSupply == 0 (contracts/distribution/HamsterGenesisRewardPool.sol#248)
HamsterRewardPool.updatePool(uint256) (contracts/distribution/
HamsterRewardPool.sol#234-254) uses a dangerous strict equality:
        - tokenSupply == 0 (contracts/distribution/HamsterRewardPool.sol#240)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-
strict-equalities

Reentrancy in Treasury.buyBonds(uint256,uint256) (contracts/Treasury.sol#196-219):
        External calls:
        - IBasisAsset(hamster).burnFrom(msg.sender,_hamsterAmount) (contracts/
Treasury.sol#214)
        - IBasisAsset(hamsterbond).mint(msg.sender,_bondAmount) (contracts/
Treasury.sol#215)
        State variables written after the call(s):
```

```
        - epochSupplyContractionLeft = epochSupplyContractionLeft.sub(_hamsterAmount)
(contracts/Treasury.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1

Hamster.setTaxTiersRate(uint8,uint256) (contracts/Hamster.sol#165-170) contains a
tautology or contradiction:
        - require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/
Hamster.sol#166)
Hamster.setTaxTiersTwap(uint8,uint256) (contracts/Hamster.sol#172-183) contains a
tautology or contradiction:
        - require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/
Hamster.sol#173)
Hamster._updateTaxRate(uint256) (contracts/Hamster.sol#246-256) contains a tautology or
contradiction:
        - tierId >= 0 (contracts/Hamster.sol#248)
Treasury.setMaxExpansionTiersEntry(uint8,uint256) (contracts/Treasury.sol#399-405)
contains a tautology or contradiction:
        - require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/
Treasury.sol#400)
Treasury.setSupplyTiersEntry(uint8,uint256) (contracts/Treasury.sol#450-461) contains a
tautology or contradiction:
        - require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/
Treasury.sol#451)
Treasury._calculateMaxSupplyExpansionPercent(uint256) (contracts/Treasury.sol#463-471)
contains a tautology or contradiction:
        - tierId >= 0 (contracts/Treasury.sol#464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-
contradiction

Treasury.getHamsterPrice().price (contracts/Treasury.sol#138) is a local variable never
initialized
Hamster._getHamsterPrice()._price (contracts/Hamster.sol#68) is a local variable never
initialized
Treasury.allocateSeigniorage()._savedForBond (contracts/Treasury.sol#168) is a local
variable never initialized
FixedPoint.mul(FixedPoint.uq112x112,uint256).z (contracts/lib/FixedPoint.sol#44) is a
local variable never initialized
UniswapV2Library.getAmountsOut(address,uint256,address[]).i (contracts/lib/
UniswapV2Library.sol#97) is a local variable never initialized
Treasury.getHamsterUpdatedPrice().price (contracts/Treasury.sol#69) is a local variable
```

```
never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-
local-variables

Hamster._getHamsterPrice() (contracts/Hamster.sol#67-73) ignores return value by
IOracle(hamsterOracle).consult(address(this),1e18) (contracts/Hamster.sol#68-72)
HamsterZapper.zapIn(address) (contracts/HamsterZapper.sol#48-54) ignores return value
by IWFTM(wftm).deposit{value: amountIn}() (contracts/HamsterZapper.sol#52)
TaxOfficeV2.setBurnThreshold(uint256) (contracts/TaxOfficeV2.sol#123-125) ignores return
value by ITaxable(hamster).setBurnThreshold(_burnThreshold) (contracts/
TaxOfficeV2.sol#124)
TaxOfficeV2._approveTokenIfNeeded(address,address) (contracts/TaxOfficeV2.sol#166-170)
ignores return value by IERC20(_token).approve(_router,type()(uint256).max) (contracts/
TaxOfficeV2.sol#168)
Treasury.getHamsterUpdatedPrice() (contracts/Treasury.sol#68-74) ignores return value
by IOracle(hamsterOracle).twap(hamster,1e18) (contracts/Treasury.sol#69-73)
Treasury.getHamsterPrice() (contracts/Treasury.sol#137-143) ignores return value by
IOracle(hamsterOracle).consult(hamster,1e18) (contracts/Treasury.sol#138-142)
Treasury.allocateSeigniorage() (contracts/Treasury.sol#158-194) ignores return value by
IBasisAsset(hamster).mint(address(this),_savedForBond) (contracts/Treasury.sol#189)
Treasury.buyBonds(uint256,uint256) (contracts/Treasury.sol#196-219) ignores return
value by IBasisAsset(hamsterbond).mint(msg.sender,_bondAmount) (contracts/
Treasury.sol#215)
Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492) ignores return
value by IBasisAsset(hamster).mint(address(this),_amount) (contracts/Treasury.sol#474)
HamsterGenesisRewardPool.constructor(address,address[],uint256,uint256,uint256,uint256)
(contracts/distribution/HamsterGenesisRewardPool.sol#89-110) ignores return value by
_comissionTokens.add(comissionTokens_[i]) (contracts/distribution/
HamsterGenesisRewardPool.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

HamsterWheel.setOperator(address) (contracts/HamsterWheel.sol#120-122) should emit an
event for:
        - operator = _operator (contracts/HamsterWheel.sol#121)
Treasury.setHamsterWheel(address) (contracts/Treasury.sol#386-388) should emit an event
for:
        - hamsterWheel = _hamsterWheel (contracts/Treasury.sol#387)
Treasury.setOperator(address) (contracts/Treasury.sol#435-437) should emit an event
for:
        - operator = _operator (contracts/Treasury.sol#436)
HShareRewardPool.setOperator(address) (contracts/distribution/
HShareRewardPool.sol#184-186) should emit an event for:
        - operator = _operator (contracts/distribution/HShareRewardPool.sol#185)
HamsterGenesisRewardPool.setOperator(address) (contracts/distribution/
HamsterGenesisRewardPool.sol#212-214) should emit an event for:
        - operator = _operator (contracts/distribution/
HamsterGenesisRewardPool.sol#213)
HamsterRewardPool.setOperator(address) (contracts/distribution/
HamsterRewardPool.sol#204-206) should emit an event for:
        - operator = _operator (contracts/distribution/HamsterRewardPool.sol#205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-
access-control

HamsterWheel.setLockUp(uint256,uint256) (contracts/HamsterWheel.sol#124-131) should
emit an event for:
        - withdrawLockupEpochs = _withdrawLockupEpochs (contracts/
HamsterWheel.sol#129)
        - rewardLockupEpochs = _rewardLockupEpochs (contracts/HamsterWheel.sol#130)
Treasury.setBondDepletionFloorPercent(uint256) (contracts/Treasury.sol#335-341) should
emit an event for:
        - bondDepletionFloorPercent = _bondDepletionFloorPercent (contracts/
Treasury.sol#340)
Treasury.setBootstrap(uint256,uint256) (contracts/Treasury.sol#343-351) should emit an
event for:
        - bootstrapEpochs = _bootstrapEpochs (contracts/Treasury.sol#349)
        - bootstrapSupplyExpansionPercent = _bootstrapSupplyExpansionPercent (contracts/
Treasury.sol#350)
Treasury.setDiscountPercent(uint256) (contracts/Treasury.sol#353-356) should emit an
event for:
        - discountPercent = _discountPercent (contracts/Treasury.sol#355)
Treasury.setExtraFunds(address,uint256,address,uint256) (contracts/
Treasury.sol#358-372) should emit an event for:
        - daoFundSharedPercent = _daoFundSharedPercent (contracts/Treasury.sol#369)
        - devFundSharedPercent = _devFundSharedPercent (contracts/Treasury.sol#371)
Treasury.setHamsterPriceCeiling(uint256) (contracts/Treasury.sol#378-384) should emit
an event for:
        - hamsterPriceCeiling = _hamsterPriceCeiling (contracts/Treasury.sol#383)
Treasury.setMaxDebtRatioPercent(uint256) (contracts/Treasury.sol#390-393) should emit
an event for:
        - maxDebtRatioPercent = _maxDebtRatioPercent (contracts/Treasury.sol#392)
Treasury.setMaxDiscountRate(uint256) (contracts/Treasury.sol#395-397) should emit an
event for:
        - maxDiscountRate = _maxDiscountRate (contracts/Treasury.sol#396)
Treasury.setMaxPremiumRate(uint256) (contracts/Treasury.sol#407-409) should emit an
event for:
        - maxPremiumRate = _maxPremiumRate (contracts/Treasury.sol#408)
Treasury.setMaxSupplyExpansionPercents(uint256) (contracts/Treasury.sol#419-425) should
emit an event for:
        - maxSupplyExpansionPercent = _maxSupplyExpansionPercent (contracts/
Treasury.sol#424)
Treasury.setMintingFactorForPayingDebt(uint256) (contracts/Treasury.sol#427-433) should
emit an event for:
        - mintingFactorForPayingDebt = _mintingFactorForPayingDebt (contracts/
Treasury.sol#432)
Treasury.setPremiumPercent(uint256) (contracts/Treasury.sol#439-442) should emit an
event for:
        - premiumPercent = _premiumPercent (contracts/Treasury.sol#441)
Treasury.setPremiumThreshold(uint256) (contracts/Treasury.sol#444-448) should emit an
event for:
        - premiumThreshold = _premiumThreshold (contracts/Treasury.sol#447)
HShareRewardPool.add(uint256,IERC20,bool,uint256) (contracts/distribution/
HShareRewardPool.sol#93-129) should emit an event for:
        - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/distribution/
HShareRewardPool.sol#127)
HShareRewardPool.set(uint256,uint256) (contracts/distribution/
HShareRewardPool.sol#173-182) should emit an event for:
        - totalAllocPoint = totalAllocPoint.sub(pool.allocPoint).add(_allocPoint)
(contracts/distribution/HShareRewardPool.sol#177-179)
HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256) (contracts/
distribution/HamsterGenesisRewardPool.sol#112-150) should emit an event for:
        - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/distribution/
HamsterGenesisRewardPool.sol#148)
HamsterGenesisRewardPool.set(uint256,uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#201-210) should emit an event for:
        - totalAllocPoint = totalAllocPoint.sub(pool.allocPoint).add(_allocPoint)
(contracts/distribution/HamsterGenesisRewardPool.sol#205-207)
HamsterRewardPool.add(uint256,IERC20,bool,uint256) (contracts/distribution/
HamsterRewardPool.sol#113-147) should emit an event for:
        - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/distribution/
HamsterRewardPool.sol#145)
HamsterRewardPool.set(uint256,uint256) (contracts/distribution/
HamsterRewardPool.sol#195-202) should emit an event for:
        - totalAllocPoint = totalAllocPoint.sub(pool.allocPoint).add(_allocPoint)
```

```
(contracts/distribution/HamsterRewardPool.sol#199)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-
arithmetic

HamsterWheel.setOperator(address)._operator (contracts/HamsterWheel.sol#120) lacks a
zero-check on :
                - operator = _operator (contracts/HamsterWheel.sol#121)
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamster
(contracts/Treasury.sol#256) lacks a zero-check on :
                - hamster = _hamster (contracts/Treasury.sol#264)
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamsterb
ond (contracts/Treasury.sol#257) lacks a zero-check on :
                - hamsterbond = _hamsterbond (contracts/Treasury.sol#265)
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamsters
hare (contracts/Treasury.sol#258) lacks a zero-check on :
                - hamstershare = _hamstershare (contracts/Treasury.sol#266)
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamsterO
racle (contracts/Treasury.sol#259) lacks a zero-check on :
                - hamsterOracle = _hamsterOracle (contracts/Treasury.sol#267)
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamsterW
heel (contracts/Treasury.sol#260) lacks a zero-check on :
                - hamsterWheel = _hamsterWheel (contracts/Treasury.sol#268)
Treasury.setHamsterOracle(address)._hamsterOracle (contracts/Treasury.sol#374) lacks a
zero-check on :
                - hamsterOracle = _hamsterOracle (contracts/Treasury.sol#375)
Treasury.setHamsterWheel(address)._hamsterWheel (contracts/Treasury.sol#386) lacks a
zero-check on :
                - hamsterWheel = _hamsterWheel (contracts/Treasury.sol#387)
Treasury.setOperator(address)._operator (contracts/Treasury.sol#435) lacks a zero-check
on :
                - operator = _operator (contracts/Treasury.sol#436)
HShareRewardPool.setOperator(address)._operator (contracts/distribution/
HShareRewardPool.sol#184) lacks a zero-check on :
                - operator = _operator (contracts/distribution/
HShareRewardPool.sol#185)
HamsterGenesisRewardPool.setOperator(address)._operator (contracts/distribution/
HamsterGenesisRewardPool.sol#212) lacks a zero-check on :
                - operator = _operator (contracts/distribution/
HamsterGenesisRewardPool.sol#213)
HamsterRewardPool.setOperator(address)._operator (contracts/distribution/
HamsterRewardPool.sol#204) lacks a zero-check on :
```

```
                - operator = _operator (contracts/distribution/
HamsterRewardPool.sol#205)
HShare.setTreasuryFund(address)._communityFund (contracts/HShare.sol#98) lacks a zero-
check on :
                - communityFund = _communityFund (contracts/HShare.sol#100)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-
address-validation

Treasury.getHamsterCirculatingSupply() (contracts/Treasury.sol#127-135) has external
calls inside a loop: balanceExcluded =
balanceExcluded.add(hamsterErc20.balanceOf(excludedFromTotalSupply[entryId]))
(contracts/Treasury.sol#132)
HShareRewardPool.updatePool(uint256) (contracts/distribution/
HShareRewardPool.sol#214-234) has external calls inside a loop: tokenSupply =
pool.token.balanceOf(address(this)) (contracts/distribution/HShareRewardPool.sol#219)
HamsterGenesisRewardPool.updatePool(uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#242-262) has external calls inside a loop: tokenSupply =
pool.token.balanceOf(address(this)) (contracts/distribution/
HamsterGenesisRewardPool.sol#247)
HamsterRewardPool.updatePool(uint256) (contracts/distribution/
HamsterRewardPool.sol#234-254) has external calls inside a loop: tokenSupply =
pool.token.balanceOf(address(this)) (contracts/distribution/HamsterRewardPool.sol#239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-
a-loop

Variable 'Hamster._getHamsterPrice()._price (contracts/Hamster.sol#68)' in
Hamster._getHamsterPrice() (contracts/Hamster.sol#67-73) potentially used before
declaration: uint256(_price) (contracts/Hamster.sol#69)
Variable 'Treasury.getHamsterUpdatedPrice().price (contracts/Treasury.sol#69)' in
Treasury.getHamsterUpdatedPrice() (contracts/Treasury.sol#68-74) potentially used
before declaration: uint256(price) (contracts/Treasury.sol#70)
Variable 'Treasury.getHamsterPrice().price (contracts/Treasury.sol#138)' in
Treasury.getHamsterPrice() (contracts/Treasury.sol#137-143) potentially used before
declaration: uint256(price) (contracts/Treasury.sol#139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-
declaration-usage-of-local-variables

Reentrancy in Treasury.allocateSeigniorage() (contracts/Treasury.sol#158-194):
        External calls:
        - _updateHamsterPrice() (contracts/Treasury.sol#159)
                - IOracle(hamsterOracle).update() (contracts/Treasury.sol#495)
        State variables written after the call(s):
        - _mse = _calculateMaxSupplyExpansionPercent(hamsterSupply).mul(1e14)
(contracts/Treasury.sol#170)
                - maxSupplyExpansionPercent = maxExpansionTiers[tierId] (contracts/
Treasury.sol#466)
        - previousEpochHamsterPrice = getHamsterPrice() (contracts/Treasury.sol#160)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2

Reentrancy in Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492):
        External calls:
        - IBasisAsset(hamster).mint(address(this),_amount) (contracts/Treasury.sol#474)
        - IERC20(hamster).transfer(daoFund,_daoFundSharedAmount) (contracts/
Treasury.sol#478)
        Event emitted after the call(s):
        - DaoFundFunded(now,_daoFundSharedAmount,epoch) (contracts/Treasury.sol#479)
Reentrancy in Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492):
        External calls:
        - IBasisAsset(hamster).mint(address(this),_amount) (contracts/Treasury.sol#474)
        - IERC20(hamster).transfer(daoFund,_daoFundSharedAmount) (contracts/
Treasury.sol#478)
        - IERC20(hamster).transfer(devFund,_devFundSharedAmount) (contracts/
Treasury.sol#484)
        Event emitted after the call(s):
        - DevFundFunded(now,_devFundSharedAmount,epoch) (contracts/Treasury.sol#485)
Reentrancy in Treasury._sendToHamsterWheel(uint256) (contracts/Treasury.sol#473-492):
        External calls:
        - IBasisAsset(hamster).mint(address(this),_amount) (contracts/Treasury.sol#474)
        - IERC20(hamster).transfer(daoFund,_daoFundSharedAmount) (contracts/
Treasury.sol#478)
        - IERC20(hamster).transfer(devFund,_devFundSharedAmount) (contracts/
Treasury.sol#484)
        - IERC20(hamster).safeApprove(hamsterWheel,0) (contracts/Treasury.sol#488)
        - IERC20(hamster).safeApprove(hamsterWheel,_amount) (contracts/
Treasury.sol#489)
        - IHamsterWheel(hamsterWheel).allocateSeigniorage(_amount) (contracts/
Treasury.sol#490)
        Event emitted after the call(s):
        - HamsterWheelFunded(now,_amount,epoch) (contracts/Treasury.sol#491)
Reentrancy in HamsterWheel.allocateSeigniorage(uint256) (contracts/
HamsterWheel.sol#215-228):
        External calls:
        - hamster.safeTransferFrom(msg.sender,address(this),amount) (contracts/
HamsterWheel.sol#226)
        Event emitted after the call(s):
        - RewardAdded(msg.sender,amount) (contracts/HamsterWheel.sol#227)
Reentrancy in Treasury.buyBonds(uint256,uint256) (contracts/Treasury.sol#196-219):
        External calls:
        - IBasisAsset(hamster).burnFrom(msg.sender,_hamsterAmount) (contracts/
Treasury.sol#214)
        - IBasisAsset(hamsterbond).mint(msg.sender,_bondAmount) (contracts/
Treasury.sol#215)
        - _updateHamsterPrice() (contracts/Treasury.sol#217)
                - IOracle(hamsterOracle).update() (contracts/Treasury.sol#495)
        Event emitted after the call(s):
        - BoughtBonds(msg.sender,_hamsterAmount,_bondAmount,epoch) (contracts/
Treasury.sol#218)
Reentrancy in HamsterWheel.claimReward() (contracts/HamsterWheel.sol#201-213):
        External calls:
        - hamster.safeTransfer(msg.sender,reward) (contracts/HamsterWheel.sol#210)
        Event emitted after the call(s):
        - RewardPaid(msg.sender,reward) (contracts/HamsterWheel.sol#211)
Reentrancy in HShareRewardPool.emergencyWithdraw(uint256) (contracts/distribution/
HShareRewardPool.sol#151-159):
        External calls:
        - pool.token.safeTransfer(msg.sender,_amount) (contracts/distribution/
HShareRewardPool.sol#157)
        Event emitted after the call(s):
        - EmergencyWithdraw(msg.sender,_pid,_amount) (contracts/distribution/
HShareRewardPool.sol#158)
Reentrancy in HamsterGenesisRewardPool.emergencyWithdraw(uint256) (contracts/
distribution/HamsterGenesisRewardPool.sol#179-187):
        External calls:
        - pool.token.safeTransfer(msg.sender,_amount) (contracts/distribution/
HamsterGenesisRewardPool.sol#185)
        Event emitted after the call(s):
        - EmergencyWithdraw(msg.sender,_pid,_amount) (contracts/distribution/
HamsterGenesisRewardPool.sol#186)
Reentrancy in HamsterRewardPool.emergencyWithdraw(uint256) (contracts/distribution/
HamsterRewardPool.sol#169-177):
        External calls:
        - pool.token.safeTransfer(msg.sender,_amount) (contracts/distribution/
HamsterRewardPool.sol#175)
```

```
        Event emitted after the call(s):
        - EmergencyWithdraw(msg.sender,_pid,_amount) (contracts/distribution/
HamsterRewardPool.sol#176)
Reentrancy in Treasury.redeemBonds(uint256,uint256) (contracts/Treasury.sol#313-333):
        External calls:
        - IBasisAsset(hamsterbond).burnFrom(msg.sender,_bondAmount) (contracts/
Treasury.sol#329)
        - IERC20(hamster).safeTransfer(msg.sender,_hamsterAmount) (contracts/
Treasury.sol#330)
        - _updateHamsterPrice() (contracts/Treasury.sol#331)
                - IOracle(hamsterOracle).update() (contracts/Treasury.sol#495)
        Event emitted after the call(s):
        - RedeemedBonds(msg.sender,_hamsterAmount,_bondAmount,epoch) (contracts/
Treasury.sol#332)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-3

HShare.unclaimedDevFund() (contracts/HShare.sol#24-29) uses timestamp for comparisons
        Dangerous comparisons:
        - _now > endTime (contracts/HShare.sol#26)
        - devFundLastClaimed >= _now (contracts/HShare.sol#27)
HShare.unclaimedTreasuryFund() (contracts/HShare.sol#31-36) uses timestamp for
comparisons
        Dangerous comparisons:
        - _now > endTime (contracts/HShare.sol#33)
        - communityFundLastClaimed >= _now (contracts/HShare.sol#34)
TaxOfficeV2.addLiquidityETHTaxFree(uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#29-62) uses timestamp for comparisons
        Dangerous comparisons:
        - amtHamster.sub(resultAmtHamster) > 0 (contracts/TaxOfficeV2.sol#58)
TaxOfficeV2.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (contracts/
TaxOfficeV2.sol#64-105) uses timestamp for comparisons
        Dangerous comparisons:
        - amtHamster.sub(resultAmtHamster) > 0 (contracts/TaxOfficeV2.sol#98)
        - amtToken.sub(resultAmtToken) > 0 (contracts/TaxOfficeV2.sol#101)
HShareRewardPool.pendingShare(uint256,address) (contracts/distribution/
HShareRewardPool.sol#36-47) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp > pool.lastRewardTime && tokenSupply != 0 (contracts/
distribution/HShareRewardPool.sol#41)
HShareRewardPool.getGeneratedReward(uint256,uint256) (contracts/distribution/
HShareRewardPool.sol#49-60) uses timestamp for comparisons
        Dangerous comparisons:
        - _fromTime >= _toTime (contracts/distribution/HShareRewardPool.sol#50)
        - _toTime >= poolEndTime (contracts/distribution/HShareRewardPool.sol#51)
        - _toTime <= poolStartTime (contracts/distribution/HShareRewardPool.sol#56)
HShareRewardPool.checkPoolDuplicate(IERC20) (contracts/distribution/
HShareRewardPool.sol#62-67) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HShareRewardPool.sol#64)
        - require(bool,string)(poolInfo[pid].token != _token,HShareRewardPool: existing
pool?) (contracts/distribution/HShareRewardPool.sol#65)
HShareRewardPool.constructor(address,uint256,uint256,uint256) (contracts/distribution/
HShareRewardPool.sol#74-91) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < _poolStartTime,late) (contracts/
distribution/HShareRewardPool.sol#80)
HShareRewardPool.add(uint256,IERC20,bool,uint256) (contracts/distribution/
HShareRewardPool.sol#93-129) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp < poolStartTime (contracts/distribution/
HShareRewardPool.sol#103)
        - _lastRewardTime == 0 (contracts/distribution/HShareRewardPool.sol#104)
        - _lastRewardTime < poolStartTime (contracts/distribution/
HShareRewardPool.sol#107)
        - _lastRewardTime == 0 || _lastRewardTime < block.timestamp (contracts/
distribution/HShareRewardPool.sol#112)
        - _isStarted = (_lastRewardTime <= poolStartTime) || (_lastRewardTime <=
block.timestamp) (contracts/distribution/HShareRewardPool.sol#116-118)
HShareRewardPool.governanceRecoverUnsupported(IERC20,uint256,address) (contracts/
distribution/HShareRewardPool.sol#161-171) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp < poolEndTime + 2592000 (contracts/distribution/
HShareRewardPool.sol#162)
HShareRewardPool.massUpdatePools() (contracts/distribution/
HShareRewardPool.sol#207-212) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HShareRewardPool.sol#209)
HShareRewardPool.updatePool(uint256) (contracts/distribution/
HShareRewardPool.sol#214-234) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp <= pool.lastRewardTime (contracts/distribution/
HShareRewardPool.sol#216)
```

```
HamsterGenesisRewardPool.pendingHAMSTER(uint256,address) (contracts/distribution/
HamsterGenesisRewardPool.sol#51-62) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp > pool.lastRewardTime && tokenSupply != 0 (contracts/
distribution/HamsterGenesisRewardPool.sol#56)
HamsterGenesisRewardPool.getGeneratedReward(uint256,uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#64-75) uses timestamp for comparisons
        Dangerous comparisons:
        - _fromTime >= _toTime (contracts/distribution/HamsterGenesisRewardPool.sol#65)
        - _toTime >= poolEndTime (contracts/distribution/
HamsterGenesisRewardPool.sol#66)
        - _toTime <= poolStartTime (contracts/distribution/
HamsterGenesisRewardPool.sol#71)
HamsterGenesisRewardPool.checkPoolDuplicate(IERC20) (contracts/distribution/
HamsterGenesisRewardPool.sol#77-82) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HamsterGenesisRewardPool.sol#79)
        - require(bool,string)(poolInfo[pid].token != _token,HamsterGenesisPool:
existing pool?) (contracts/distribution/HamsterGenesisRewardPool.sol#80)
HamsterGenesisRewardPool.constructor(address,address[],uint256,uint256,uint256,uint256)
(contracts/distribution/HamsterGenesisRewardPool.sol#89-110) uses timestamp for
comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < _poolStartTime,late) (contracts/
distribution/HamsterGenesisRewardPool.sol#97)
HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256) (contracts/
distribution/HamsterGenesisRewardPool.sol#112-150) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp < poolStartTime (contracts/distribution/
HamsterGenesisRewardPool.sol#123)
        - _lastRewardTime == 0 (contracts/distribution/
HamsterGenesisRewardPool.sol#124)
        - _lastRewardTime < poolStartTime (contracts/distribution/
HamsterGenesisRewardPool.sol#127)
        - _lastRewardTime == 0 || _lastRewardTime < block.timestamp (contracts/
distribution/HamsterGenesisRewardPool.sol#132)
        - _isStarted = (_lastRewardTime <= poolStartTime) || (_lastRewardTime <=
block.timestamp) (contracts/distribution/HamsterGenesisRewardPool.sol#136-138)
HamsterGenesisRewardPool.governanceRecoverUnsupported(IERC20,uint256,address)
(contracts/distribution/HamsterGenesisRewardPool.sol#189-199) uses timestamp for
comparisons
```

```
        Dangerous comparisons:
        - block.timestamp < poolEndTime + 2592000 (contracts/distribution/
HamsterGenesisRewardPool.sol#190)
HamsterGenesisRewardPool.massUpdatePools() (contracts/distribution/
HamsterGenesisRewardPool.sol#235-240) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HamsterGenesisRewardPool.sol#237)
HamsterGenesisRewardPool.updatePool(uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#242-262) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp <= pool.lastRewardTime (contracts/distribution/
HamsterGenesisRewardPool.sol#244)
HamsterRewardPool.pendingHAMSTER(uint256,address) (contracts/distribution/
HamsterRewardPool.sol#34-45) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp > pool.lastRewardTime && tokenSupply != 0 (contracts/
distribution/HamsterRewardPool.sol#39)
HamsterRewardPool.getGeneratedReward(uint256,uint256) (contracts/distribution/
HamsterRewardPool.sol#47-73) uses timestamp for comparisons
        Dangerous comparisons:
        - _toTime >= epochEndTimes[epochId - 1] (contracts/distribution/
HamsterRewardPool.sol#49)
HamsterRewardPool.checkPoolDuplicate(IERC20) (contracts/distribution/
HamsterRewardPool.sol#75-80) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HamsterRewardPool.sol#77)
        - require(bool,string)(poolInfo[pid].token != _token,HamsterRewardPool:
existing pool?) (contracts/distribution/HamsterRewardPool.sol#78)
HamsterRewardPool.constructor(address,uint256,uint256,uint256,uint256,uint256)
(contracts/distribution/HamsterRewardPool.sol#87-111) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < _poolStartTime,late) (contracts/
distribution/HamsterRewardPool.sol#95)
HamsterRewardPool.add(uint256,IERC20,bool,uint256) (contracts/distribution/
HamsterRewardPool.sol#113-147) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp < poolStartTime (contracts/distribution/
HamsterRewardPool.sol#123)
        - _lastRewardTime == 0 (contracts/distribution/HamsterRewardPool.sol#124)
        - _lastRewardTime < poolStartTime (contracts/distribution/
HamsterRewardPool.sol#127)
```

```
        - _lastRewardTime == 0 || _lastRewardTime < block.timestamp (contracts/
distribution/HamsterRewardPool.sol#132)
        - _isStarted = (_lastRewardTime <= poolStartTime) || (_lastRewardTime <=
block.timestamp) (contracts/distribution/HamsterRewardPool.sol#136)
HamsterRewardPool.governanceRecoverUnsupported(IERC20,uint256,address) (contracts/
distribution/HamsterRewardPool.sol#179-193) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp < epochEndTimes[1] + 2592000 (contracts/distribution/
HamsterRewardPool.sol#184)
HamsterRewardPool.massUpdatePools() (contracts/distribution/
HamsterRewardPool.sol#227-232) uses timestamp for comparisons
        Dangerous comparisons:
        - pid < length (contracts/distribution/HamsterRewardPool.sol#229)
HamsterRewardPool.updatePool(uint256) (contracts/distribution/
HamsterRewardPool.sol#234-254) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp <= pool.lastRewardTime (contracts/distribution/
HamsterRewardPool.sol#236)
UniswapV2OracleLibrary.currentCumulativePrices(address) (contracts/lib/
UniswapV2OracleLibrary.sol#18-42) uses timestamp for comparisons
        Dangerous comparisons:
        - blockTimestampLast != blockTimestamp (contracts/lib/
UniswapV2OracleLibrary.sol#33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-
timestamp

HShareRewardPool.updatePool(uint256) (contracts/distribution/
HShareRewardPool.sol#214-234) has costly operations inside a loop:
        - totalAllocPoint = totalAllocPoint.add(pool.allocPoint) (contracts/
distribution/HShareRewardPool.sol#226)
HamsterGenesisRewardPool.updatePool(uint256) (contracts/distribution/
HamsterGenesisRewardPool.sol#242-262) has costly operations inside a loop:
        - totalAllocPoint = totalAllocPoint.add(pool.allocPoint) (contracts/
distribution/HamsterGenesisRewardPool.sol#254)
HamsterRewardPool.updatePool(uint256) (contracts/distribution/
HamsterRewardPool.sol#234-254) has costly operations inside a loop:
        - totalAllocPoint = totalAllocPoint.add(pool.allocPoint) (contracts/
distribution/HamsterRewardPool.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-
operations-inside-a-loop
```

```
Babylonian.sqrt(uint256) (contracts/lib/Babylonian.sol#6-18) is never used and should
be removed
FixedPoint.decode(FixedPoint.uq112x112) (contracts/lib/FixedPoint.sol#57-59) is never
used and should be removed
FixedPoint.div(FixedPoint.uq112x112,uint112) (contracts/lib/FixedPoint.sol#36-39) is
never used and should be removed
FixedPoint.encode(uint112) (contracts/lib/FixedPoint.sol#26-28) is never used and
should be removed
FixedPoint.encode144(uint144) (contracts/lib/FixedPoint.sol#31-33) is never used and
should be removed
FixedPoint.reciprocal(FixedPoint.uq112x112) (contracts/lib/FixedPoint.sol#67-70) is
never used and should be removed
FixedPoint.sqrt(FixedPoint.uq112x112) (contracts/lib/FixedPoint.sol#73-75) is never
used and should be removed
SafeMath8.add(uint8,uint8) (contracts/lib/SafeMath8.sol#29-34) is never used and should
be removed
SafeMath8.div(uint8,uint8) (contracts/lib/SafeMath8.sol#103-105) is never used and
should be removed
SafeMath8.div(uint8,uint8,string) (contracts/lib/SafeMath8.sol#119-125) is never used
and should be removed
SafeMath8.mod(uint8,uint8) (contracts/lib/SafeMath8.sol#139-141) is never used and
should be removed
SafeMath8.mod(uint8,uint8,string) (contracts/lib/SafeMath8.sol#155-158) is never used
and should be removed
SafeMath8.mul(uint8,uint8) (contracts/lib/SafeMath8.sol#77-89) is never used and should
be removed
UniswapV2Library.getAmountIn(uint256,uint256,uint256) (contracts/lib/
UniswapV2Library.sol#76-86) is never used and should be removed
UniswapV2Library.getAmountOut(uint256,uint256,uint256) (contracts/lib/
UniswapV2Library.sol#62-73) is never used and should be removed
UniswapV2Library.getAmountsIn(address,uint256,address[]) (contracts/lib/
UniswapV2Library.sol#104-116) is never used and should be removed
UniswapV2Library.getAmountsOut(address,uint256,address[]) (contracts/lib/
UniswapV2Library.sol#89-101) is never used and should be removed
UniswapV2Library.getReserves(address,address,address) (contracts/lib/
UniswapV2Library.sol#40-48) is never used and should be removed
UniswapV2Library.pairFor(address,address,address) (contracts/lib/
UniswapV2Library.sol#19-37) is never used and should be removed
UniswapV2Library.quote(uint256,uint256,uint256) (contracts/lib/
UniswapV2Library.sol#51-59) is never used and should be removed
UniswapV2Library.sortTokens(address,address) (contracts/lib/UniswapV2Library.sol#12-16)
```

```
is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Safe112.add(uint112,uint112) (contracts/lib/Safe112.sol#6-11) is never used and should
be removed
Safe112.div(uint112,uint112) (contracts/lib/Safe112.sol#39-41) is never used and should
be removed
Safe112.div(uint112,uint112,string) (contracts/lib/Safe112.sol#43-53) is never used and
should be removed
Safe112.mod(uint112,uint112) (contracts/lib/Safe112.sol#55-57) is never used and should
be removed
Safe112.mod(uint112,uint112,string) (contracts/lib/Safe112.sol#59-66) is never used and
should be removed
Safe112.mul(uint112,uint112) (contracts/lib/Safe112.sol#28-37) is never used and should
be removed
Safe112.sub(uint112,uint112) (contracts/lib/Safe112.sol#13-15) is never used and should
be removed
Safe112.sub(uint112,uint112,string) (contracts/lib/Safe112.sol#17-26) is never used and
should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

UQ112x112.encode(uint112) (contracts/lib/UQ112x112.sol#12-14) is never used and should
be removed
UQ112x112.uqdiv(uint224,uint112) (contracts/lib/UQ112x112.sol#17-19) is never used and
should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

HBond (contracts/HBond.sol#8-26) should inherit from IBasisAsset (contracts/interfaces/
IBasisAsset.sol#5-12)
Oracle (contracts/Oracle.sol#8-83) should inherit from IOracle (contracts/interfaces/
IOracle.sol#5-9)
MockedWFTM (contracts/mocked/MockedWFTM.sol#7-34) should inherit from IWFTM (contracts/
interfaces/IWFTM.sol#7-10)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-
inheritance

Parameter HShare.distributeReward(address,uint256)._farmingIncentiveFund (contracts/
HShare.sol#77) is not in mixedCase
Parameter HShare.distributeReward(address,uint256)._farmingPoolAllocation (contracts/
HShare.sol#77) is not in mixedCase
Parameter HShare.governanceRecoverUnsupported(IERC20,uint256,address)._token (contracts/
HShare.sol#85) is not in mixedCase
Parameter HShare.governanceRecoverUnsupported(IERC20,uint256,address)._amount
(contracts/HShare.sol#86) is not in mixedCase
Parameter HShare.governanceRecoverUnsupported(IERC20,uint256,address)._to (contracts/
HShare.sol#87) is not in mixedCase
Parameter HShare.setDevFund(address)._devFund (contracts/HShare.sol#92) is not in
mixedCase
Parameter HShare.setTreasuryFund(address)._communityFund (contracts/HShare.sol#98) is
not in mixedCase
Parameter Hamster.isAddressExcluded(address)._address (contracts/Hamster.sol#55) is not
in mixedCase
Parameter
Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._genesisPool
(contracts/Hamster.sol#96) is not in mixedCase
Parameter Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._gen
esisPoolDistribution (contracts/Hamster.sol#97) is not in mixedCase
Parameter
Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._hamsterPool
(contracts/Hamster.sol#98) is not in mixedCase
Parameter Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._ham
sterPoolDistribution (contracts/Hamster.sol#99) is not in mixedCase
Parameter Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._air
dropWallet (contracts/Hamster.sol#100) is not in mixedCase
Parameter Hamster.distributeReward(address,uint256,address,uint256,address,uint256)._air
dropDistribution (contracts/Hamster.sol#101) is not in mixedCase
Parameter Hamster.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/Hamster.sol#118) is not in mixedCase
Parameter Hamster.governanceRecoverUnsupported(IERC20,uint256,address)._amount
(contracts/Hamster.sol#119) is not in mixedCase
Parameter Hamster.governanceRecoverUnsupported(IERC20,uint256,address)._to (contracts/
Hamster.sol#120) is not in mixedCase
Parameter Hamster.includeAddress(address)._address (contracts/Hamster.sol#125) is not
in mixedCase
Parameter Hamster.setBurnThreshold(uint256)._burnThreshold (contracts/Hamster.sol#138)
is not in mixedCase
Parameter Hamster.setHamsterOracle(address)._hamsterOracle (contracts/Hamster.sol#143)
is not in mixedCase
Parameter Hamster.setTaxCollectorAddress(address)._taxCollectorAddress (contracts/
Hamster.sol#148) is not in mixedCase
Parameter Hamster.setTaxOffice(address)._taxOffice (contracts/Hamster.sol#153) is not in
mixedCase
```

```
Parameter Hamster.setTaxRate(uint256)._taxRate (contracts/Hamster.sol#159) is not in
mixedCase
Parameter Hamster.setTaxTiersRate(uint8,uint256)._index (contracts/Hamster.sol#165) is
not in mixedCase
Parameter Hamster.setTaxTiersRate(uint8,uint256)._value (contracts/Hamster.sol#165) is
not in mixedCase
Parameter Hamster.setTaxTiersTwap(uint8,uint256)._index (contracts/Hamster.sol#172) is
not in mixedCase
Parameter Hamster.setTaxTiersTwap(uint8,uint256)._value (contracts/Hamster.sol#172) is
not in mixedCase
Parameter Hamster.excludeAddress(address)._address (contracts/Hamster.sol#193) is not
in mixedCase
Parameter HamsterWheel.initialize(IERC20,IERC20,ITreasury)._hamster (contracts/
HamsterWheel.sol#100) is not in mixedCase
Parameter HamsterWheel.initialize(IERC20,IERC20,ITreasury)._share (contracts/
HamsterWheel.sol#101) is not in mixedCase
Parameter HamsterWheel.initialize(IERC20,IERC20,ITreasury)._treasury (contracts/
HamsterWheel.sol#102) is not in mixedCase
Parameter HamsterWheel.setOperator(address)._operator (contracts/HamsterWheel.sol#120)
is not in mixedCase
Parameter HamsterWheel.setLockUp(uint256,uint256)._withdrawLockupEpochs (contracts/
HamsterWheel.sol#124) is not in mixedCase
Parameter HamsterWheel.setLockUp(uint256,uint256)._rewardLockupEpochs (contracts/
HamsterWheel.sol#124) is not in mixedCase
Parameter HamsterWheel.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/HamsterWheel.sol#230) is not in mixedCase
Parameter HamsterWheel.governanceRecoverUnsupported(IERC20,uint256,address)._amount
(contracts/HamsterWheel.sol#230) is not in mixedCase
Parameter HamsterWheel.governanceRecoverUnsupported(IERC20,uint256,address)._to
(contracts/HamsterWheel.sol#230) is not in mixedCase
Parameter Oracle.consult(address,uint256)._token (contracts/Oracle.sol#21) is not in
mixedCase
Parameter Oracle.consult(address,uint256)._amountIn (contracts/Oracle.sol#21) is not in
mixedCase
Parameter Oracle.twap(address,uint256)._token (contracts/Oracle.sol#30) is not in
mixedCase
Parameter Oracle.twap(address,uint256)._amountIn (contracts/Oracle.sol#30) is not in
mixedCase
Parameter TaxOfficeV2.excludeAddressFromTax(address)._address (contracts/
TaxOfficeV2.sol#115) is not in mixedCase
Parameter TaxOfficeV2.includeAddressInTax(address)._address (contracts/
TaxOfficeV2.sol#119) is not in mixedCase
Parameter TaxOfficeV2.setBurnThreshold(uint256)._burnThreshold (contracts/
TaxOfficeV2.sol#123) is not in mixedCase
Parameter TaxOfficeV2.setTaxCollectorAddress(address)._taxCollectorAddress (contracts/
TaxOfficeV2.sol#127) is not in mixedCase
Parameter TaxOfficeV2.setTaxExclusionForAddress(address,bool)._address (contracts/
TaxOfficeV2.sol#131) is not in mixedCase
Parameter TaxOfficeV2.setTaxExclusionForAddress(address,bool)._excluded (contracts/
TaxOfficeV2.sol#131) is not in mixedCase
Parameter TaxOfficeV2.setTaxRate(uint256)._taxRate (contracts/TaxOfficeV2.sol#135) is not
in mixedCase
Parameter TaxOfficeV2.setTaxTiersRate(uint8,uint256)._index (contracts/
TaxOfficeV2.sol#139) is not in mixedCase
Parameter TaxOfficeV2.setTaxTiersRate(uint8,uint256)._value (contracts/
TaxOfficeV2.sol#139) is not in mixedCase
Parameter TaxOfficeV2.setTaxTiersTwap(uint8,uint256)._index (contracts/
TaxOfficeV2.sol#143) is not in mixedCase
Parameter TaxOfficeV2.setTaxTiersTwap(uint8,uint256)._value (contracts/
TaxOfficeV2.sol#143) is not in mixedCase
Parameter TaxOfficeV2.setTaxableHamsterOracle(address)._hamsterOracle (contracts/
TaxOfficeV2.sol#147) is not in mixedCase
Parameter TaxOfficeV2.taxFreeTransferFrom(address,address,uint256)._sender (contracts/
TaxOfficeV2.sol#152) is not in mixedCase
Parameter TaxOfficeV2.taxFreeTransferFrom(address,address,uint256)._recipient (contracts/
TaxOfficeV2.sol#153) is not in mixedCase
Parameter TaxOfficeV2.taxFreeTransferFrom(address,address,uint256)._amt (contracts/
TaxOfficeV2.sol#154) is not in mixedCase
Parameter TaxOfficeV2.transferTaxOffice(address)._newTaxOffice (contracts/
TaxOfficeV2.sol#162) is not in mixedCase
Parameter TaxOracle.consult(address,uint256)._token (contracts/TaxOracle.sol#29) is not
in mixedCase
Parameter TaxOracle.consult(address,uint256)._amountIn (contracts/TaxOracle.sol#29) is
not in mixedCase
Parameter TaxOracle.setHamster(address)._hamster (contracts/TaxOracle.sol#50) is not in
mixedCase
Parameter TaxOracle.setWftm(address)._wftm (contracts/TaxOracle.sol#56) is not in
mixedCase
Parameter TaxOracle.setPair(address)._pair (contracts/TaxOracle.sol#62) is not in
mixedCase
Parameter Treasury.buyBonds(uint256,uint256)._hamsterAmount (contracts/
Treasury.sol#197) is not in mixedCase
```

```
Parameter Treasury.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/Treasury.sol#222) is not in mixedCase
Parameter Treasury.governanceRecoverUnsupported(IERC20,uint256,address)._amount
(contracts/Treasury.sol#223) is not in mixedCase
Parameter Treasury.governanceRecoverUnsupported(IERC20,uint256,address)._to (contracts/
Treasury.sol#224) is not in mixedCase
Parameter
Treasury.hamsterWheelGovernanceRecoverUnsupported(address,uint256,address)._token
(contracts/Treasury.sol#237) is not in mixedCase
Parameter
Treasury.hamsterWheelGovernanceRecoverUnsupported(address,uint256,address)._amount
(contracts/Treasury.sol#238) is not in mixedCase
Parameter
Treasury.hamsterWheelGovernanceRecoverUnsupported(address,uint256,address)._to
(contracts/Treasury.sol#239) is not in mixedCase
Parameter Treasury.hamsterWheelSetLockUp(uint256,uint256)._withdrawLockupEpochs
(contracts/Treasury.sol#245) is not in mixedCase
Parameter Treasury.hamsterWheelSetLockUp(uint256,uint256)._rewardLockupEpochs
(contracts/Treasury.sol#246) is not in mixedCase
Parameter Treasury.hamsterWheelSetOperator(address)._operator (contracts/
Treasury.sol#251) is not in mixedCase
Parameter
Treasury.initialize(address,address,address,address,address,uint256,address[])._hamster
(contracts/Treasury.sol#256) is not in mixedCase
Parameter Treasury.initialize(address,address,address,address,address,uint256,address[])
._hamsterbond (contracts/Treasury.sol#257) is not in mixedCase
Parameter Treasury.initialize(address,address,address,address,address,uint256,address[])
._hamstershare (contracts/Treasury.sol#258) is not in mixedCase
Parameter Treasury.initialize(address,address,address,address,address,uint256,address[])
._hamsterOracle (contracts/Treasury.sol#259) is not in mixedCase
Parameter Treasury.initialize(address,address,address,address,address,uint256,address[])
._hamsterWheel (contracts/Treasury.sol#260) is not in mixedCase
Parameter Treasury.initialize(address,address,address,address,address,uint256,address[])
._startTime (contracts/Treasury.sol#261) is not in mixedCase
Parameter Treasury.redeemBonds(uint256,uint256)._bondAmount (contracts/
Treasury.sol#314) is not in mixedCase
Parameter Treasury.setBondDepletionFloorPercent(uint256)._bondDepletionFloorPercent
(contracts/Treasury.sol#335) is not in mixedCase
Parameter Treasury.setBootstrap(uint256,uint256)._bootstrapEpochs (contracts/
Treasury.sol#343) is not in mixedCase
Parameter Treasury.setBootstrap(uint256,uint256)._bootstrapSupplyExpansionPercent
```

```
(contracts/Treasury.sol#343) is not in mixedCase
Parameter Treasury.setDiscountPercent(uint256)._discountPercent (contracts/
Treasury.sol#353) is not in mixedCase
Parameter Treasury.setExtraFunds(address,uint256,address,uint256)._daoFund (contracts/
Treasury.sol#359) is not in mixedCase
Parameter Treasury.setExtraFunds(address,uint256,address,uint256)._daoFundSharedPercent
(contracts/Treasury.sol#360) is not in mixedCase
Parameter Treasury.setExtraFunds(address,uint256,address,uint256)._devFund (contracts/
Treasury.sol#361) is not in mixedCase
Parameter Treasury.setExtraFunds(address,uint256,address,uint256)._devFundSharedPercent
(contracts/Treasury.sol#362) is not in mixedCase
Parameter Treasury.setHamsterOracle(address)._hamsterOracle (contracts/
Treasury.sol#374) is not in mixedCase
Parameter Treasury.setHamsterPriceCeiling(uint256)._hamsterPriceCeiling (contracts/
Treasury.sol#378) is not in mixedCase
Parameter Treasury.setHamsterWheel(address)._hamsterWheel (contracts/Treasury.sol#386)
is not in mixedCase
Parameter Treasury.setMaxDebtRatioPercent(uint256)._maxDebtRatioPercent (contracts/
Treasury.sol#390) is not in mixedCase
Parameter Treasury.setMaxDiscountRate(uint256)._maxDiscountRate (contracts/
Treasury.sol#395) is not in mixedCase
Parameter Treasury.setMaxExpansionTiersEntry(uint8,uint256)._index (contracts/
Treasury.sol#399) is not in mixedCase
Parameter Treasury.setMaxExpansionTiersEntry(uint8,uint256)._value (contracts/
Treasury.sol#399) is not in mixedCase
Parameter Treasury.setMaxPremiumRate(uint256)._maxPremiumRate (contracts/
Treasury.sol#407) is not in mixedCase
Parameter Treasury.setMaxSupplyContractionPercent(uint256)._maxSupplyContractionPercent
(contracts/Treasury.sol#411) is not in mixedCase
Parameter Treasury.setMaxSupplyExpansionPercents(uint256)._maxSupplyExpansionPercent
(contracts/Treasury.sol#419) is not in mixedCase
Parameter Treasury.setMintingFactorForPayingDebt(uint256)._mintingFactorForPayingDebt
(contracts/Treasury.sol#427) is not in mixedCase
Parameter Treasury.setOperator(address)._operator (contracts/Treasury.sol#435) is not
in mixedCase
Parameter Treasury.setPremiumPercent(uint256)._premiumPercent (contracts/
Treasury.sol#439) is not in mixedCase
Parameter Treasury.setPremiumThreshold(uint256)._premiumThreshold (contracts/
Treasury.sol#444) is not in mixedCase
Parameter Treasury.setSupplyTiersEntry(uint8,uint256)._index (contracts/
Treasury.sol#450) is not in mixedCase
```

```
Parameter Treasury.setSupplyTiersEntry(uint8,uint256)._value (contracts/
Treasury.sol#450) is not in mixedCase
Parameter HShareRewardPool.pendingShare(uint256,address)._pid (contracts/distribution/
HShareRewardPool.sol#36) is not in mixedCase
Parameter HShareRewardPool.pendingShare(uint256,address)._user (contracts/distribution/
HShareRewardPool.sol#36) is not in mixedCase
Parameter HShareRewardPool.getGeneratedReward(uint256,uint256)._fromTime (contracts/
distribution/HShareRewardPool.sol#49) is not in mixedCase
Parameter HShareRewardPool.getGeneratedReward(uint256,uint256)._toTime (contracts/
distribution/HShareRewardPool.sol#49) is not in mixedCase
Parameter HShareRewardPool.checkPoolDuplicate(IERC20)._token (contracts/distribution/
HShareRewardPool.sol#62) is not in mixedCase
Parameter HShareRewardPool.add(uint256,IERC20,bool,uint256)._allocPoint (contracts/
distribution/HShareRewardPool.sol#94) is not in mixedCase
Parameter HShareRewardPool.add(uint256,IERC20,bool,uint256)._token (contracts/
distribution/HShareRewardPool.sol#95) is not in mixedCase
Parameter HShareRewardPool.add(uint256,IERC20,bool,uint256)._withUpdate (contracts/
distribution/HShareRewardPool.sol#96) is not in mixedCase
Parameter HShareRewardPool.add(uint256,IERC20,bool,uint256)._lastRewardTime (contracts/
distribution/HShareRewardPool.sol#97) is not in mixedCase
Parameter HShareRewardPool.deposit(uint256,uint256)._pid (contracts/distribution/
HShareRewardPool.sol#131) is not in mixedCase
Parameter HShareRewardPool.deposit(uint256,uint256)._amount (contracts/distribution/
HShareRewardPool.sol#131) is not in mixedCase
Parameter HShareRewardPool.emergencyWithdraw(uint256)._pid (contracts/distribution/
HShareRewardPool.sol#151) is not in mixedCase
Parameter HShareRewardPool.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/distribution/HShareRewardPool.sol#161) is not in mixedCase
Parameter HShareRewardPool.set(uint256,uint256)._pid (contracts/distribution/
HShareRewardPool.sol#173) is not in mixedCase
Parameter HShareRewardPool.set(uint256,uint256)._allocPoint (contracts/distribution/
HShareRewardPool.sol#173) is not in mixedCase
Parameter HShareRewardPool.setOperator(address)._operator (contracts/distribution/
HShareRewardPool.sol#184) is not in mixedCase
Parameter HShareRewardPool.withdraw(uint256,uint256)._pid (contracts/distribution/
HShareRewardPool.sol#188) is not in mixedCase
Parameter HShareRewardPool.withdraw(uint256,uint256)._amount (contracts/distribution/
HShareRewardPool.sol#188) is not in mixedCase
Parameter HShareRewardPool.updatePool(uint256)._pid (contracts/distribution/
HShareRewardPool.sol#214) is not in mixedCase
Parameter HShareRewardPool.safeHShareTransfer(address,uint256)._to (contracts/
distribution/HShareRewardPool.sol#236) is not in mixedCase
Parameter HShareRewardPool.safeHShareTransfer(address,uint256)._amount (contracts/
distribution/HShareRewardPool.sol#236) is not in mixedCase
Parameter HamsterGenesisRewardPool.pendingHAMSTER(uint256,address)._pid (contracts/
distribution/HamsterGenesisRewardPool.sol#51) is not in mixedCase
Parameter HamsterGenesisRewardPool.pendingHAMSTER(uint256,address)._user (contracts/
distribution/HamsterGenesisRewardPool.sol#51) is not in mixedCase
Parameter HamsterGenesisRewardPool.getGeneratedReward(uint256,uint256)._fromTime
(contracts/distribution/HamsterGenesisRewardPool.sol#64) is not in mixedCase
Parameter HamsterGenesisRewardPool.getGeneratedReward(uint256,uint256)._toTime
(contracts/distribution/HamsterGenesisRewardPool.sol#64) is not in mixedCase
Parameter HamsterGenesisRewardPool.checkPoolDuplicate(IERC20)._token (contracts/
distribution/HamsterGenesisRewardPool.sol#77) is not in mixedCase
Parameter HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256)._allocPoint
(contracts/distribution/HamsterGenesisRewardPool.sol#113) is not in mixedCase
Parameter HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256)._token
(contracts/distribution/HamsterGenesisRewardPool.sol#114) is not in mixedCase
Parameter HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256)._withUpdate
(contracts/distribution/HamsterGenesisRewardPool.sol#115) is not in mixedCase
Parameter
HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256)._lastRewardTime
(contracts/distribution/HamsterGenesisRewardPool.sol#116) is not in mixedCase
Parameter HamsterGenesisRewardPool.add(uint256,IERC20,bool,uint256,uint256)._maxDeposit
(contracts/distribution/HamsterGenesisRewardPool.sol#117) is not in mixedCase
Parameter HamsterGenesisRewardPool.deposit(uint256,uint256)._pid (contracts/
distribution/HamsterGenesisRewardPool.sol#152) is not in mixedCase
Parameter HamsterGenesisRewardPool.deposit(uint256,uint256)._amount (contracts/
distribution/HamsterGenesisRewardPool.sol#152) is not in mixedCase
Parameter HamsterGenesisRewardPool.emergencyWithdraw(uint256)._pid (contracts/
distribution/HamsterGenesisRewardPool.sol#179) is not in mixedCase
Parameter
HamsterGenesisRewardPool.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/distribution/HamsterGenesisRewardPool.sol#189) is not in mixedCase
Parameter HamsterGenesisRewardPool.set(uint256,uint256)._pid (contracts/distribution/
HamsterGenesisRewardPool.sol#201) is not in mixedCase
Parameter HamsterGenesisRewardPool.set(uint256,uint256)._allocPoint (contracts/
distribution/HamsterGenesisRewardPool.sol#201) is not in mixedCase
Parameter HamsterGenesisRewardPool.setOperator(address)._operator (contracts/
distribution/HamsterGenesisRewardPool.sol#212) is not in mixedCase
Parameter HamsterGenesisRewardPool.withdraw(uint256,uint256)._pid (contracts/
distribution/HamsterGenesisRewardPool.sol#216) is not in mixedCase
```

```
Parameter HamsterGenesisRewardPool.withdraw(uint256,uint256)._amount (contracts/
distribution/HamsterGenesisRewardPool.sol#216) is not in mixedCase
Parameter HamsterGenesisRewardPool.updatePool(uint256)._pid (contracts/distribution/
HamsterGenesisRewardPool.sol#242) is not in mixedCase
Parameter HamsterGenesisRewardPool.safeHamsterTransfer(address,uint256)._to (contracts/
distribution/HamsterGenesisRewardPool.sol#264) is not in mixedCase
Parameter HamsterGenesisRewardPool.safeHamsterTransfer(address,uint256)._amount
(contracts/distribution/HamsterGenesisRewardPool.sol#264) is not in mixedCase
Parameter HamsterRewardPool.pendingHAMSTER(uint256,address)._pid (contracts/
distribution/HamsterRewardPool.sol#34) is not in mixedCase
Parameter HamsterRewardPool.pendingHAMSTER(uint256,address)._user (contracts/
distribution/HamsterRewardPool.sol#34) is not in mixedCase
Parameter HamsterRewardPool.getGeneratedReward(uint256,uint256)._fromTime (contracts/
distribution/HamsterRewardPool.sol#47) is not in mixedCase
Parameter HamsterRewardPool.getGeneratedReward(uint256,uint256)._toTime (contracts/
distribution/HamsterRewardPool.sol#47) is not in mixedCase
Parameter HamsterRewardPool.checkPoolDuplicate(IERC20)._token (contracts/distribution/
HamsterRewardPool.sol#75) is not in mixedCase
Parameter HamsterRewardPool.add(uint256,IERC20,bool,uint256)._allocPoint (contracts/
distribution/HamsterRewardPool.sol#114) is not in mixedCase
Parameter HamsterRewardPool.add(uint256,IERC20,bool,uint256)._token (contracts/
distribution/HamsterRewardPool.sol#115) is not in mixedCase
Parameter HamsterRewardPool.add(uint256,IERC20,bool,uint256)._withUpdate (contracts/
distribution/HamsterRewardPool.sol#116) is not in mixedCase
Parameter HamsterRewardPool.add(uint256,IERC20,bool,uint256)._lastRewardTime (contracts/
distribution/HamsterRewardPool.sol#117) is not in mixedCase
Parameter HamsterRewardPool.deposit(uint256,uint256)._pid (contracts/distribution/
HamsterRewardPool.sol#149) is not in mixedCase
Parameter HamsterRewardPool.deposit(uint256,uint256)._amount (contracts/distribution/
HamsterRewardPool.sol#149) is not in mixedCase
Parameter HamsterRewardPool.emergencyWithdraw(uint256)._pid (contracts/distribution/
HamsterRewardPool.sol#169) is not in mixedCase
Parameter HamsterRewardPool.governanceRecoverUnsupported(IERC20,uint256,address)._token
(contracts/distribution/HamsterRewardPool.sol#180) is not in mixedCase
Parameter HamsterRewardPool.set(uint256,uint256)._pid (contracts/distribution/
HamsterRewardPool.sol#195) is not in mixedCase
Parameter HamsterRewardPool.set(uint256,uint256)._allocPoint (contracts/distribution/
HamsterRewardPool.sol#195) is not in mixedCase
Parameter HamsterRewardPool.setOperator(address)._operator (contracts/distribution/
HamsterRewardPool.sol#204) is not in mixedCase
Parameter HamsterRewardPool.withdraw(uint256,uint256)._pid (contracts/distribution/
HamsterRewardPool.sol#208) is not in mixedCase
Parameter HamsterRewardPool.withdraw(uint256,uint256)._amount (contracts/distribution/
HamsterRewardPool.sol#208) is not in mixedCase
Parameter HamsterRewardPool.updatePool(uint256)._pid (contracts/distribution/
HamsterRewardPool.sol#234) is not in mixedCase
Parameter HamsterRewardPool.safeHamsterTransfer(address,uint256)._to (contracts/
distribution/HamsterRewardPool.sol#256) is not in mixedCase
Parameter HamsterRewardPool.safeHamsterTransfer(address,uint256)._amount (contracts/
distribution/HamsterRewardPool.sol#256) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/interfaces/IUniswapV2Pair.sol#22)
is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/interfaces/IUniswapV2Pair.sol#23)
is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/interfaces/
IUniswapV2Pair.sol#40) is not in mixedCase
Function IUniswapV2Router.WETH() (contracts/interfaces/IUniswapV2Router.sol#7) is not
in mixedCase
Struct FixedPoint.uq112x112 (contracts/lib/FixedPoint.sol#11-13) is not in CapWords
Struct FixedPoint.uq144x112 (contracts/lib/FixedPoint.sol#17-19) is not in CapWords
Parameter Epoch.setEpoch(uint256)._epoch (contracts/utils/Epoch.sol#47) is not in
mixedCase
Parameter Epoch.setPeriod(uint256)._period (contracts/utils/Epoch.sol#51) is not in
mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions

Variable Oracle.price0Average (contracts/Oracle.sol#18) is too similar to
Oracle.price1Average (contracts/Oracle.sol#19)
Variable Oracle.twap(address,uint256).price0Cumulative (contracts/Oracle.sol#32) is too
similar to Oracle.twap(address,uint256).price1Cumulative (contracts/Oracle.sol#33)
Variable Oracle.update().price0Cumulative (contracts/Oracle.sol#68) is too similar to
Oracle.update().price1Cumulative (contracts/Oracle.sol#69)
Variable Oracle.update().price0Cumulative (contracts/Oracle.sol#68) is too similar to
Oracle.twap(address,uint256).price1Cumulative (contracts/Oracle.sol#33)
Variable Oracle.price0CumulativeLast (contracts/Oracle.sol#16) is too similar to
Oracle.price1CumulativeLast (contracts/Oracle.sol#17)
Variable Oracle.twap(address,uint256).price0Cumulative (contracts/Oracle.sol#32) is too
similar to Oracle.update().price1Cumulative (contracts/Oracle.sol#69)
Variable Treasury.setExtraFunds(address,uint256,address,uint256)._daoFundSharedPercent
(contracts/Treasury.sol#360) is too similar to
Treasury.setExtraFunds(address,uint256,address,uint256)._devFundSharedPercent
```

```
(contracts/Treasury.sol#362)
Variable HamsterGenesisRewardPool._comissionTokens (contracts/distribution/
HamsterGenesisRewardPool.sol#28) is too similar to HamsterGenesisRewardPool.constructor(
address,address[],uint256,uint256,uint256,uint256).comissionTokens_ (contracts/
distribution/HamsterGenesisRewardPool.sol#91)
Variable IUniswapV2Router.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountADesired (contracts/interfaces/IUniswapV2Router.sol#11) is too
similar to IUniswapV2Router.addLiquidity(address,address,uint256,uint256,uint256,uint256
,address,uint256).amountBDesired (contracts/interfaces/IUniswapV2Router.sol#12)
Variable UniswapV2OracleLibrary.currentCumulativePrices(address).price0Cumulative
(contracts/lib/UniswapV2OracleLibrary.sol#22) is too similar to
UniswapV2OracleLibrary.currentCumulativePrices(address).price1Cumulative (contracts/lib/
UniswapV2OracleLibrary.sol#23)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-
are-too-similar

Treasury.initialize(address,address,address,address,address,uint256,address[])
(contracts/Treasury.sol#255-311) uses literals with too many digits:
        - supplyTiers = (0,500000000000000000000000,1000000000000000000000000,1500000000
000000000000000,2000000000000000000000000,5000000000000000000000000,10000000000000000000
000000,20000000000000000000000000,50000000000000000000000000) (contracts/
Treasury.sol#272-282)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-
digits

mint(address,uint256) should be declared external:
        - HBond.mint(address,uint256) (contracts/HBond.sol#20-25)
initialize(IERC20,IERC20,ITreasury) should be declared external:
        - HamsterWheel.initialize(IERC20,IERC20,ITreasury) (contracts/
HamsterWheel.sol#99-118)
rewardPerShare() should be declared external:
        - HamsterWheel.rewardPerShare() (contracts/HamsterWheel.sol#169-171)
deposit() should be declared external:
        - MockedWFTM.deposit() (contracts/mocked/MockedWFTM.sol#18-24)
withdraw(uint256) should be declared external:
        - MockedWFTM.withdraw(uint256) (contracts/mocked/MockedWFTM.sol#26-33)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (contracts/owner/Operator.sol#25-27)
getCurrentEpoch() should be declared external:
        - Epoch.getCurrentEpoch() (contracts/utils/Epoch.sol#16-18)
getLastEpochTime() should be declared external:
        - Epoch.getLastEpochTime() (contracts/utils/Epoch.sol#20-22)
getPeriod() should be declared external:
        - Epoch.getPeriod() (contracts/utils/Epoch.sol#24-26)
getStartTime() should be declared external:
        - Epoch.getStartTime() (contracts/utils/Epoch.sol#28-30)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-
function-that-could-be-declared-external
. analyzed (46 contracts with 77 detectors), 363 result(s) found
```

0x Guard