Guard

Smart contracts security assessment

Final report

Tariff: Standard

MouseHaunt

November 2021

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for the Mousehaunt team. The code was checked after commit 49f9fe.

Name	MouseHaunt
Audit date	2021-11-08 - 2021-11-08
Language	Solidity
Platform	Binance Smart Chain

Contracts checked

Name	Address
MouseHauntToken	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/MouseHauntToken.sol
WhitelistSale	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/WhitelistSale.sol
TokenAllocation	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/utils/TokenAllocation.sol
BoosterSale	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/booster/BoosterSale.sol
BMHTE	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/booster/BMHTE.sol
BMHTL	https://github.com/mousehaunt/contracts/blob/49f9fe4b666f74167b1ee2416b0116956265d573/contracts/booster/BMHTL.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

- 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Outdated Compiler Version	passed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

1. Users may not be able to claim bought tokens if owner removes allowance (TokenAllocation)

The tokens that are bought on token sale are supposed to be held on the mhtOwner account. The mhtOwner gives allowance to the TokenAllocation contract to spend tokens and when a user claims his tokens, they are transferred from the mhtOwner to the user. The mhtOwner can set allowance to 0 and users won't be able to claim their tokens.

Low severity issues

1. Lack of events (WhitelistSale)

The functions _addToWhitelist() and _removeFromWhitelist() do not emit events.

Recommendation: Create events and emit them when an address is added or removed from whitelist.

2. Contructor parameters not checked (TokenAllocation)

Parameters passed in the constructor are not checked.

Recommendation: Check at least that owner, mth are not zero addresses and unlockAtIGOPercent is less than 100:

```
constructor(
  address _mhtOwner,
  IERC20 _mht,
  uint256 _unlockAtIGOPercent,
  uint256 _cliffMonths,
  uint256 _vestingPeriodMonths
) {
  require(mhtOwner != address(0), "zero mhtOwner");
  require(mht != address(0), "zero mht");
  require(unlockAtIGOPercent <= 100, "unlockAtIGOPercent must be less than 100");
  mhtOwner = _mhtOwner;
  mht = _mht;
  unlockAtIGOPercent = _unlockAtIGOPercent;
  cliffMonths = _cliffMonths;
  vestingPeriodMonths = _vestingPeriodMonths;
}
```

3. boosters array may be declared as mapping to save gas (BoosterSale)

A mapping booster's address to booster index may be used to get booster index by its address. This will save gas in the buy() function.

Recommendation: Use mapping boosters address => uint to get array index by booster token address.

Conclusion

MouseHaunt MouseHauntToken, WhitelistSale, TokenAllocation, BoosterSale, BMHTE, BMHTL contracts were audited. 1 medium, 3 low severity issues were found.

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.

Static code analysis result

```
INFO:Detectors:
BoosterSale.buy(IERC20,uint256) (contracts/booster/BoosterSale.sol#79-101) performs a
multiplication on the result of a division:
        -_numberOfBoosters = _numberOfBoostersInWei / 1e18 (contracts/booster/
BoosterSale.sol#96)
        -busdAmountInWei = _numberOfBoosters * busdPriceInWei (contracts/booster/
BoosterSale.sol#97)
TokenAllocation._releaseAmount(TokenAllocation.UserInfo,uint256) (contracts/utils/
TokenAllocation.sol#92-132) performs a multiplication on the result of a division:
        -_amount = (userInfo.totalTokens - _unlockedAtIgo) / vestingPeriodMonths
(contracts/utils/TokenAllocation.sol#113-114)
        -_distributedTokens = _unlockedAtIgo + _amount * (_vestingIndex - 1) (contracts/
utils/TokenAllocation.sol#117-119)
TokenAllocation._releaseAmount(TokenAllocation.UserInfo,uint256) (contracts/utils/
TokenAllocation.sol#92-132) performs a multiplication on the result of a division:
        -_amount = (userInfo.totalTokens - _unlockedAtIgo) / vestingPeriodMonths
(contracts/utils/TokenAllocation.sol#113-114)
        -_remainingTokens < 2 * _amount (contracts/utils/TokenAllocation.sol#125)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-
multiply
INFO:Detectors:
Reentrancy in WhitelistSale.buy(uint256) (contracts/WhitelistSale.sol#66-83):
        External calls:
        - busd.safeTransferFrom(msg.sender,mhtOwner,busdAmount) (contracts/
WhitelistSale.sol#81)
        State variables written after the call(s):
        - _updateUserTokenAllocation(msg.sender,_mhtAmount) (contracts/
WhitelistSale.sol#82)
                - userInfo.totalTokens += totalTokens (contracts/utils/
TokenAllocation.sol#61)
                - userInfo.remainingTokens += totalTokens (contracts/utils/
TokenAllocation.sol#62)
                - userInfo.lastClaimMonthIndex = - 1 (contracts/utils/
TokenAllocation.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1
INFO:Detectors:
TokenAllocation.constructor(address,IERC20,uint256,uint256,uint256)._mhtOwner
(contracts/utils/TokenAllocation.sol#35) lacks a zero-check on :
                - mhtOwner = _mhtOwner (contracts/utils/TokenAllocation.sol#41)
```

```
BoosterSale.constructor(address,IERC20)._boosterOwner (contracts/booster/
BoosterSale.sol#23) lacks a zero-check on :
                - boosterOwner = _boosterOwner (contracts/booster/BoosterSale.sol#26)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-
address-validation
INFO:Detectors:
Reentrancy in TokenAllocation.claim() (contracts/utils/TokenAllocation.sol#156-175):
        External calls:
        - mht.safeTransferFrom(mhtOwner,msg.sender,amount) (contracts/utils/
TokenAllocation.sol#171)
        Event emitted after the call(s):
        - Claimed(msg.sender,i,amount) (contracts/utils/TokenAllocation.sol#172)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-3
INFO:Detectors:
TokenAllocation._getMonthIndexFromTimestamp(uint256) (contracts/utils/
TokenAllocation.sol#139-154) uses timestamp for comparisons
        Dangerous comparisons:
        - t <= timestamp (contracts/utils/TokenAllocation.sol#148)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-
timestamp
INFO:Detectors:
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/
Address.sol#26-36) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/
Address.sol#195-215) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#207-210)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.2']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/security/Pausable.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/
ERC20Burnable.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/
IERC20Metadata.sol#3)
```

```
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/
SafeERC20.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/
ERC165.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/
IERC165.sol#3)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeMath.sol#3)
        - ^0.8.2 (contracts/MouseHauntToken.sol#2)
        - ^0.8.2 (contracts/WhitelistSale.sol#2)
        - ^0.8.2 (contracts/booster/BMHTE.sol#2)
        - ^0.8.2 (contracts/booster/BMHTL.sol#2)
        - ^0.8.2 (contracts/booster/BoosterSale.sol#2)
        - ^0.8.2 (contracts/utils/TokenAllocation.sol#2)
        - ^0.8.2 (contracts/utils/Whitelist.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-
pragma-directives-are-used
INFO:Detectors:
BMHTE._beforeTokenTransfer(address,address,uint256) (contracts/booster/BMHTE.sol#27-33)
is never used and should be removed
BMHTL._beforeTokenTransfer(address,address,uint256) (contracts/booster/BMHTL.sol#27-33)
is never used and should be removed
MouseHauntToken._beforeTokenTransfer(address,address,uint256) (contracts/
MouseHauntToken.sol#27-33) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/Pausable.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/
```

```
ERC20Burnable.sol#3) necessitates a version too recent to be trusted. Consider
deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/
IERC20Metadata.sol#3) necessitates a version too recent to be trusted. Consider
deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/
SafeERC20.sol#3) necessitates a version too recent to be trusted. Consider deploying
with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/
ERC165.sol#3) necessitates a version too recent to be trusted. Consider deploying with
0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/
IERC165.sol#3) necessitates a version too recent to be trusted. Consider deploying with
0.6.12/0.7.6
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeMath.sol#3)
necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/MouseHauntToken.sol#2) necessitates a version too
recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/WhitelistSale.sol#2) necessitates a version too recent
to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/booster/BMHTE.sol#2) necessitates a version too recent
to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/booster/BMHTL.sol#2) necessitates a version too recent
to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/booster/BoosterSale.sol#2) necessitates a version too
recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/utils/TokenAllocation.sol#2) necessitates a version too
recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.2 (contracts/utils/Whitelist.sol#2) necessitates a version too
recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/
```

```
contracts/utils/Address.sol#54-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/
contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(node_modules/@openzeppelin/contracts/utils/Address.sol#122-133):
        - (success,returndata) = target.call{value: value}(data) (node_modules/
@openzeppelin/contracts/utils/Address.sol#131)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/
@openzeppelin/contracts/utils/Address.sol#151-160):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/
contracts/utils/Address.sol#158)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/
@openzeppelin/contracts/utils/Address.sol#178-187):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/
contracts/utils/Address.sol#185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-
calls
INFO:Detectors:
Parameter WhitelistSale.setIgoTimestamp(uint256)._igoTimestamp (contracts/
WhitelistSale.sol#57) is not in mixedCase
Parameter WhitelistSale.buy(uint256)._mhtAmount (contracts/WhitelistSale.sol#66) is not
in mixedCase
Parameter WhitelistSale.addToWhitelist(address[])._buyers (contracts/
WhitelistSale.sol#85) is not in mixedCase
Parameter WhitelistSale.removeFromWhitelist(address[])._buyers (contracts/
WhitelistSale.sol#93) is not in mixedCase
Parameter BoosterSale.configure(IERC20[],uint256[],uint256[])._boosters (contracts/
booster/BoosterSale.sol#39) is not in mixedCase
Parameter BoosterSale.configure(IERC20[],uint256[],uint256[])._busdPricePerBoosterInWei
(contracts/booster/BoosterSale.sol#40) is not in mixedCase
Parameter BoosterSale.configure(IERC20[],uint256[],uint256[])._capPerBoosterInWei
(contracts/booster/BoosterSale.sol#41) is not in mixedCase
Parameter BoosterSale.buy(IERC20,uint256)._numberOfBoostersInWei (contracts/booster/
BoosterSale.sol#79) is not in mixedCase
Parameter BoosterSale.addToWhitelist(address[])._buyers (contracts/booster/
BoosterSale.sol#103) is not in mixedCase
Parameter BoosterSale.removeFromWhitelist(address[])._buyers (contracts/booster/
BoosterSale.sol#113) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
```

```
MouseHauntToken.constructor(address) (contracts/MouseHauntToken.sol#13-17) uses
literals with too many digits:
        - _mint(owner,100000000 * 10 ** decimals()) (contracts/MouseHauntToken.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-
digits
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/
contracts/access/AccessControl.sol#129-131)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/
contracts/access/AccessControl.sol#142-144)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/
contracts/access/AccessControl.sol#160-164)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/
Ownable.sol#53-55)
name() should be declared external:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/
ERC20.sol#61-63)
symbol() should be declared external:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/
ERC20.sol#69-71)
totalSupply() should be declared external:
        - ERC20.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC20/
ERC20.sol#93-95)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/
ERC20.sol#100-102)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/
ERC20/ERC20.sol#112-115)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/
ERC20/ERC20.sol#131-134)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (node_modules/@openzeppelin/
contracts/token/ERC20/ERC20.sol#149-163)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/
```

```
contracts/token/ERC20/ERC20.sol#177-180)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/
contracts/token/ERC20/ERC20.sol#196-204)
burn(uint256) should be declared external:
        - ERC20Burnable.burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC20/
extensions/ERC20Burnable.sol#19-21)
burnFrom(address,uint256) should be declared external:
        - ERC20Burnable.burnFrom(address,uint256) (node_modules/@openzeppelin/contracts/
token/ERC20/extensions/ERC20Burnable.sol#34-41)
pause() should be declared external:
        - MouseHauntToken.pause() (contracts/MouseHauntToken.sol#19-21)
unpause() should be declared external:
        - MouseHauntToken.unpause() (contracts/MouseHauntToken.sol#23-25)
pause() should be declared external:
        - WhitelistSale.pause() (contracts/WhitelistSale.sol#49-51)
unpause() should be declared external:
        - WhitelistSale.unpause() (contracts/WhitelistSale.sol#53-55)
setIgoTimestamp(uint256) should be declared external:
        - WhitelistSale.setIgoTimestamp(uint256) (contracts/WhitelistSale.sol#57-64)
buy(uint256) should be declared external:
        - WhitelistSale.buy(uint256) (contracts/WhitelistSale.sol#66-83)
addToWhitelist(address[]) should be declared external:
        - WhitelistSale.addToWhitelist(address[]) (contracts/WhitelistSale.sol#85-91)
removeFromWhitelist(address[]) should be declared external:
        - WhitelistSale.removeFromWhitelist(address[]) (contracts/
WhitelistSale.sol#93-99)
pause() should be declared external:
        - BMHTE.pause() (contracts/booster/BMHTE.sol#19-21)
unpause() should be declared external:
        - BMHTE.unpause() (contracts/booster/BMHTE.sol#23-25)
pause() should be declared external:
        - BMHTL.pause() (contracts/booster/BMHTL.sol#19-21)
unpause() should be declared external:
        - BMHTL.unpause() (contracts/booster/BMHTL.sol#23-25)
pause() should be declared external:
        - BoosterSale.pause() (contracts/booster/BoosterSale.sol#30-32)
unpause() should be declared external:
        - BoosterSale.unpause() (contracts/booster/BoosterSale.sol#34-36)
configure(IERC20[],uint256[],uint256[]) should be declared external:
        - BoosterSale.configure(IERC20[],uint256[],uint256[]) (contracts/booster/
BoosterSale.sol#38-59)
```

```
buy(IERC20,uint256) should be declared external:
        - BoosterSale.buy(IERC20,uint256) (contracts/booster/BoosterSale.sol#79-101)
addToWhitelist(address[]) should be declared external:
        - BoosterSale.addToWhitelist(address[]) (contracts/booster/
BoosterSale.sol#103-111)
removeFromWhitelist(address[]) should be declared external:
        - BoosterSale.removeFromWhitelist(address[]) (contracts/booster/
BoosterSale.sol#113-121)
claim() should be declared external:
        - TokenAllocation.claim() (contracts/utils/TokenAllocation.sol#156-175)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-
function-that-could-be-declared-external
```

0x Guard