



Smart contracts security assessment

Final report

[Tariff: Standard](#)

BEBU Finance

September 2022



0xguard.com



hello@0xguard.com

Contents

1. Introduction	3
2. Contracts checked	3
3. Procedure	4
4. Known vulnerabilities checked	5
5. Classification of issue severity	6
6. Issues	6
7. Conclusion	9
8. Disclaimer	10
9. Slither output	11

Introduction

This report has been prepared for the BEBU Finance team upon their request.

The audited project is a fork of the Tomb Finance Project. The code is available at Github [BebuFinance/bebu-finance](https://github.com/BebuFinance/bebu-finance). The code was checked in the [02b769c](#) commit.

The purpose of this audit was to ensure that no issues were introduced with the changes to the original code and that known vulnerabilities (e.g. [circumventing](#) the protocol's fee system) are fixed prior to deployment.

Further details about BEBU Finance are available at the official website: <https://bebu.finance/>.

Name	BEBU Finance
Audit date	2022-08-31 - 2022-09-05
Language	Solidity
Platform	Polygon Network

Contracts checked

Name	Address
Bebu	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/TOKEN/BEBU/Bebu.sol
BEARETH1X	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol
BULLETH1X	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol

BebuRewardPool	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Reward_pool/BebuRewardPool.sol
StableFarmingRewardPool	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Reward_pool/StableFarmingRewardPool.sol
Oracle	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Oracle/Oracle.sol
BEARETH1X_Treasury	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol
BULLETH1X_Treasury	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol
Boardroom	https://github.com/BebuFinance/bebu-finance/blob/02b769c80e4dbca8c68d4be6555fb925651e4f5e/contracts/Boardroom/Boardroom.sol
Multiple contracts	

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 analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
<u>Unencrypted Private Data On-Chain</u>	passed
<u>Code With No Effects</u>	passed
<u>Message call with hardcoded gas amount</u>	passed
<u>Typographical Error</u>	passed
<u>DoS With Block Gas Limit</u>	passed
<u>Presence of unused variables</u>	passed
<u>Incorrect Inheritance Order</u>	passed
<u>Requirement Violation</u>	passed
<u>Weak Sources of Randomness from Chain Attributes</u>	passed
<u>Shadowing State Variables</u>	passed
<u>Incorrect Constructor Name</u>	passed
<u>Block values as a proxy for time</u>	passed
<u>Authorization through tx.origin</u>	passed
<u>DoS with Failed Call</u>	passed
<u>Delegatecall to Untrusted Callee</u>	passed
<u>Use of Deprecated Solidity Functions</u>	passed
<u>Assert Violation</u>	passed
<u>State Variable Default Visibility</u>	passed
<u>Reentrancy</u>	not passed

<u>Unprotected SELFDESTRUCT Instruction</u>	passed
<u>Unprotected Ether Withdrawal</u>	passed
<u>Unchecked Call Return Value</u>	passed
<u>Floating Pragma</u>	not passed
<u>Outdated Compiler Version</u>	passed
<u>Integer Overflow and Underflow</u>	passed
<u>Function Default Visibility</u>	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. Contract ownership (Multiple contracts)

Status: Open

The `governanceRecoverUnsupported()` function (found in the `Boardroom`, `BebuRewardPool`, `StableFarmingRewardPool`, `Bebu`, `BEARETH1X`, `BULLETH1X`, `BEARETH1X_Treasury` and `BULLETH1X_Treasury` contracts) can remove all tokens from the contract balance if the operator or owner role is compromised.

Recommendation: There are a large number of functions with the `onlyOperator()` modifier, there is a possibility that the operator can be compromised. It is recommended to create multiple roles for different kinds of functions to reduce the operator's problem. It is also recommended to add a time delay to the especially important set functions using the [TimelockController](#). We also recommend that you look through the entire codebase to find functions that are dangerous for you as the owner of the project (mainly set functions), if there are any, then add a call to them via multisig wallet. This will help avoid the issue of owner compromise.

Low severity issues

1. Reentrancy attack (BebuRewardPool)

Status: Open

When withdrawing, some pool tokens may be subject to a reentrancy attack. The variable `user.rewardDebt` on 262L is updated after calling `pool.token.safeTransfer()`.

Recommendation: It is recommended to update the value of the `user.rewardDebt` variable before calling `pool.token.safeTransfer()`.

2. Reentrancy attack (StableFarmingRewardPool)

Status: Open

When withdrawing, some pool tokens may be subject to a reentrancy attack. The variable `user.rewardDebt` on 381L is updated after calling `pool.token.safeTransfer()`.

Recommendation: It is recommended to update the value of the `user.rewardDebt` variable before calling `pool.token.safeTransfer()`.

3. Functions lacks validation of input parameters (StableFarmingRewardPool)

Status: Open

The `constructor()`, `addTwoPool()`, `setOperator()` contract functions do not check the input addresses against a null address.

4. Unused variable (StableFarmingRewardPool)

Status: Open

The `TOTAL_REWARDS` variable is not used in the contract.

5. Few events (Multiple contracts)

Status: Open

Many set functions from contracts are missing events when changing important values in the contract.

Recommendation: Create events for these set functions.

Conclusion

BEBU Finance Bebu, BEARETH1X, BULLETH1X, BebuRewardPool, StableFarmingRewardPool, Oracle, BEARETH1X_Treasury, BULLETH1X_Treasury, Boardroom, Multiple contracts contracts were audited. 1 medium, 5 low severity issues were found.

The BEBU Finance Project was compared with the Tomb Project. BEBU Finance has changed the implementation of Boardroom, BebuRewardPool, and Bebu contracts. A blacklist has been added.

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

Reentrancy in BebuRewardPool.deposit(uint256,uint256) (contracts/Reward_pool/BebuRewardPool.sol#224-243):

External calls:

- [- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#233)
- [- returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (contracts/utils/token/SafeERC20.sol#69)
- [- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/BebuRewardPool.sol#283)
- [- (success,returndata) = target.call{value: value}(data) (contracts/utils/Address.sol#119)
- [- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/BebuRewardPool.sol#285)
- [- pool.token.safeTransferFrom(_sender,address(this),_amount) (contracts/Reward_pool/BebuRewardPool.sol#238)

External calls sending eth:

- [- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#233)
- [- (success,returndata) = target.call{value: value}(data) (contracts/utils/Address.sol#119)

State variables written after the call(s):

- [- user.amount = user.amount.add(_amount) (contracts/Reward_pool/BebuRewardPool.sol#239)
- [- user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18) (contracts/Reward_pool/BebuRewardPool.sol#241)

Reentrancy in StableFarmingRewardPool.deposit(uint256,uint256) (contracts/Reward_pool/StableFarmingRewardPool.sol#330-360):

External calls:

- [- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/StableFarmingRewardPool.sol#339)
- [- returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (contracts/utils/token/SafeERC20.sol#69)
- [- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/StableFarmingRewardPool.sol#394)
- [- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#396)

```

☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒- pool.token.safeTransferFrom(_sender, feeTo, tax) (contracts/Reward_pool/
StableFarmingRewardPool.sol#348)
☒- pool.token.safeTransferFrom(_sender, address(this), _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#350)
☒External calls sending eth:
☒- safeBebuTransfer(_sender, _pending) (contracts/Reward_pool/
StableFarmingRewardPool.sol#339)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒State variables written after the call(s):
☒- user.amount = user.amount.add(_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#351)
☒- user.TimerStart = poolStartTime (contracts/Reward_pool/
StableFarmingRewardPool.sol#354)
☒- user.TimerStart = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#356)
☒- user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18) (contracts/
Reward_pool/StableFarmingRewardPool.sol#358)
Reentrancy in StableFarmingRewardPool.depositTwoTokens(uint256,uint256) (contracts/
Reward_pool/StableFarmingRewardPool.sol#313-319):
☒External calls:
☒- deposit(_pool0, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#317)
☒☒- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to, _BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒☒- Bebu.safeTransfer(_to, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
☒☒- pool.token.safeTransferFrom(_sender, feeTo, tax) (contracts/Reward_pool/
StableFarmingRewardPool.sol#348)
☒☒- pool.token.safeTransferFrom(_sender, address(this), _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#350)
☒- deposit(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#318)
☒☒- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to, _BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)

```

```

    (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
    Bebu.safeTransfer(_to, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
    pool.token.safeTransferFrom(_sender, feeTo, tax) (contracts/Reward_pool/
StableFarmingRewardPool.sol#348)
    pool.token.safeTransferFrom(_sender, address(this), _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#350)
    External calls sending eth:
    deposit(_pool0, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#317)
    (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
    deposit(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#318)
    (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
    State variables written after the call(s):
    deposit(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#318)
    pool.lastRewardTime = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#298)
    pool.isStarted = true (contracts/Reward_pool/StableFarmingRewardPool.sol#302)
    pool.accBebuPerShare =
    pool.accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply)) (contracts/Reward_pool/
StableFarmingRewardPool.sol#308)
    pool.lastRewardTime = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#310)
    deposit(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#318)
    totalAllocPoint = totalAllocPoint.add(pool.allocPoint) (contracts/Reward_pool/
StableFarmingRewardPool.sol#303)
    deposit(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#318)
    user.amount = user.amount.add(_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#351)
    user.TimerStart = poolStartTime (contracts/Reward_pool/
StableFarmingRewardPool.sol#354)
    user.TimerStart = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#356)
    user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18) (contracts/
Reward_pool/StableFarmingRewardPool.sol#358)
    Reentrancy in Boardroom.stake(uint256) (contracts/Boardroom/Boardroom.sol#217-228):
    External calls:
    share.safeTransferFrom(msg.sender, feeTo, tax) (contracts/Boardroom/Boardroom.sol#223)
    super.stake(amount) (contracts/Boardroom/Boardroom.sol#225)

```

```

❏- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
❏- share.safeTransferFrom(msg.sender, address(this), amount) (contracts/Boardroom/
ShareWrapper.sol#28)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏External calls sending eth:
❏- super.stake(amount) (contracts/Boardroom/Boardroom.sol#225)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏State variables written after the call(s):
❏- members[msg.sender].epochTimerStart = treasury.epoch() (contracts/Boardroom/
Boardroom.sol#226)
Reentrancy in Boardroom.withdraw(uint256) (contracts/Boardroom/Boardroom.sol#230-237):
❏External calls:
❏- claimReward() (contracts/Boardroom/Boardroom.sol#234)
❏- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏- token.safeTransfer(msg.sender, reward) (contracts/Boardroom/Boardroom.sol#250)
❏- super.withdraw(amount) (contracts/Boardroom/Boardroom.sol#235)
❏- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
❏- share.safeTransfer(msg.sender, amount) (contracts/Boardroom/ShareWrapper.sol#36)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏External calls sending eth:
❏- claimReward() (contracts/Boardroom/Boardroom.sol#234)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏- super.withdraw(amount) (contracts/Boardroom/Boardroom.sol#235)
❏- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
❏State variables written after the call(s):
❏- super.withdraw(amount) (contracts/Boardroom/Boardroom.sol#235)
❏- _balances[msg.sender] = memberShare.sub(amount) (contracts/Boardroom/
ShareWrapper.sol#35)
Reentrancy in BebuRewardPool.withdraw(uint256, uint256) (contracts/Reward_pool/
BebuRewardPool.sol#246-264):
❏External calls:

```

```

☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#255)
☒☒- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(contracts/utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/BebuRewardPool.sol#283)
☒☒- (success,returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
☒☒- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/BebuRewardPool.sol#285)
☒External calls sending eth:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#255)
☒☒- (success,returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
☒State variables written after the call(s):
☒- user.amount = user.amount.sub(_amount) (contracts/Reward_pool/
BebuRewardPool.sol#259)
Reentrancy in BebuRewardPool.withdraw(uint256,uint256) (contracts/Reward_pool/
BebuRewardPool.sol#246-264):
☒External calls:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#255)
☒☒- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(contracts/utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/BebuRewardPool.sol#283)
☒☒- (success,returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
☒☒- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/BebuRewardPool.sol#285)
☒- pool.token.safeTransfer(_sender,_amount) (contracts/Reward_pool/
BebuRewardPool.sol#260)
☒External calls sending eth:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/BebuRewardPool.sol#255)
☒☒- (success,returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
☒State variables written after the call(s):
☒- user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18) (contracts/
Reward_pool/BebuRewardPool.sol#262)
Reentrancy in StableFarmingRewardPool.withdraw(uint256,uint256) (contracts/Reward_pool/
StableFarmingRewardPool.sol#363-383):
☒External calls:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/
StableFarmingRewardPool.sol#372)
☒☒- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(contracts/utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)

```

```

☒☒- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒External calls sending eth:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/
StableFarmingRewardPool.sol#372)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒State variables written after the call(s):
☒- user.amount = user.amount.sub(_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#377)
Reentrancy in StableFarmingRewardPool.withdraw(uint256,uint256) (contracts/Reward_pool/
StableFarmingRewardPool.sol#363-383):
☒External calls:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/
StableFarmingRewardPool.sol#372)
☒☒- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/Utils/token/SafeERC20.sol#69)
☒☒- Bebu.safeTransfer(_to,_BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)
☒☒- Bebu.safeTransfer(_to,_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒- pool.token.safeTransfer(_sender,_amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#378)
☒External calls sending eth:
☒- safeBebuTransfer(_sender,_pending) (contracts/Reward_pool/
StableFarmingRewardPool.sol#372)
☒☒- (success, returndata) = target.call{value: value}(data) (contracts/Utils/
Address.sol#119)
☒State variables written after the call(s):
☒- user.TimerStart = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#380)
☒- user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18) (contracts/
Reward_pool/StableFarmingRewardPool.sol#381)
Reentrancy in StableFarmingRewardPool.withdrawTwoTokens(uint256,uint256) (contracts/
Reward_pool/StableFarmingRewardPool.sol#321-327):
☒External calls:
☒- withdraw(_pool0,amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#325)

```



```

❏❏- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/utils/token/SafeERC20.sol#69)
❏❏- Bebu.safeTransfer(_to, _BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)
❏❏- (success, returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
❏❏- Bebu.safeTransfer(_to, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
❏❏- pool.token.safeTransfer(_sender, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#378)
❏- withdraw(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#326)
❏❏- returndata = address(token).functionCall(data, SafeERC20: low-level call failed)
(contracts/utils/token/SafeERC20.sol#69)
❏❏- Bebu.safeTransfer(_to, _BebuBal) (contracts/Reward_pool/
StableFarmingRewardPool.sol#394)
❏❏- (success, returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
❏❏- Bebu.safeTransfer(_to, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#396)
❏❏- pool.token.safeTransfer(_sender, _amount) (contracts/Reward_pool/
StableFarmingRewardPool.sol#378)
❏External calls sending eth:
❏- withdraw(_pool0, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#325)
❏❏- (success, returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
❏- withdraw(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#326)
❏❏- (success, returndata) = target.call{value: value}(data) (contracts/utils/
Address.sol#119)
❏State variables written after the call(s):
❏- withdraw(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#326)
❏❏- pool.lastRewardTime = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#298)
❏❏- pool.isStarted = true (contracts/Reward_pool/StableFarmingRewardPool.sol#302)
❏❏- pool.accBebuPerShare =
pool.accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply)) (contracts/Reward_pool/
StableFarmingRewardPool.sol#308)
❏❏- pool.lastRewardTime = block.timestamp (contracts/Reward_pool/
StableFarmingRewardPool.sol#310)
❏- withdraw(_pool1, amount) (contracts/Reward_pool/StableFarmingRewardPool.sol#326)
❏❏- totalAllocPoint = totalAllocPoint.add(pool.allocPoint) (contracts/Reward_pool/
StableFarmingRewardPool.sol#303)

```

☒- `withdraw(_pool1,amount)` (`contracts/Reward_pool/StableFarmingRewardPool.sol#326`)

☒☒- `user.amount = user.amount.sub(_amount)` (`contracts/Reward_pool/StableFarmingRewardPool.sol#377`)

☒☒- `user.TimerStart = block.timestamp` (`contracts/Reward_pool/StableFarmingRewardPool.sol#380`)

☒☒- `user.rewardDebt = user.amount.mul(pool.accBebuPerShare).div(1e18)` (`contracts/Reward_pool/StableFarmingRewardPool.sol#381`)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities>

`Bebu.governanceRecoverUnsupported(IERC20,uint256,address)` (`contracts/TOKEN/BEBU/Bebu.sol#144-150`) ignores return value by `_token.transfer(_to,_amount)` (`contracts/TOKEN/BEBU/Bebu.sol#149`)

`BEARETH1X.governanceRecoverUnsupported(IERC20,uint256,address)` (`contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#412-418`) ignores return value by `_token.transfer(_to,_amount)` (`contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#417`)

`BULLETH1X.governanceRecoverUnsupported(IERC20,uint256,address)` (`contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#412-418`) ignores return value by `_token.transfer(_to,_amount)` (`contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#417`)

`BEARETH1X_Treasury._sendToBoardroom(uint256)` (`contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#264-287`) ignores return value by `IERC20(token).transfer(daoFund,_daoFundSharedAmount)` (`contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#270`)

`BEARETH1X_Treasury._sendToBoardroom(uint256)` (`contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#264-287`) ignores return value by `IERC20(token).transfer(devFund,_devFundSharedAmount)` (`contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#277`)

`BULLETH1X_Treasury._sendToBoardroom(uint256)` (`contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#264-287`) ignores return value by `IERC20(token).transfer(daoFund,_daoFundSharedAmount)` (`contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#270`)

`BULLETH1X_Treasury._sendToBoardroom(uint256)` (`contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#264-287`) ignores return value by `IERC20(token).transfer(devFund,_devFundSharedAmount)` (`contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#277`)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer>

`BebuRewardPool.pendingShare(uint256,address)` (`contracts/Reward_pool/`

`BebuRewardPool.sol#179-190`) performs a multiplication on the result of a division:

☒- `_BebuReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint)` (`contracts/`

```

Reward_pool/BebuRewardPool.sol#186)
☒-accBebuPerShare = accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply))
(contracts/Reward_pool/BebuRewardPool.sol#187)
BebuRewardPool.updatePool(uint256) (contracts/Reward_pool/BebuRewardPool.sol#201-221)
performs a multiplication on the result of a division:
☒-_BebuReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint) (contracts/
Reward_pool/BebuRewardPool.sol#217)
☒-pool.accBebuPerShare =
pool.accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply)) (contracts/Reward_pool/
BebuRewardPool.sol#218)
StableFarmingRewardPool.pendingShare(uint256,address) (contracts/Reward_pool/
StableFarmingRewardPool.sol#269-280) performs a multiplication on the result of a
division:
☒-_BebuReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint) (contracts/
Reward_pool/StableFarmingRewardPool.sol#276)
☒-accBebuPerShare = accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply))
(contracts/Reward_pool/StableFarmingRewardPool.sol#277)
StableFarmingRewardPool.updatePool(uint256) (contracts/Reward_pool/
StableFarmingRewardPool.sol#291-311) performs a multiplication on the result of a
division:
☒-_BebuReward = _generatedReward.mul(pool.allocPoint).div(totalAllocPoint) (contracts/
Reward_pool/StableFarmingRewardPool.sol#307)
☒-pool.accBebuPerShare =
pool.accBebuPerShare.add(_BebuReward.mul(1e18).div(tokenSupply)) (contracts/Reward_pool/
StableFarmingRewardPool.sol#308)
BEARETH1X._calculateTax(uint256,uint256,uint256) (contracts/TOKEN/TOKEN/ETH/
BEARETH1X.sol#308-313) performs a multiplication on the result of a division:
☒-tax = amount.mul(numerator).div(denominator).mul(taxCoefficient).div(10000) (contracts/
TOKEN/TOKEN/ETH/BEARETH1X.sol#312)
BEARETH1X._calculateReward(uint256,uint256,uint256) (contracts/TOKEN/TOKEN/ETH/
BEARETH1X.sol#336-341) performs a multiplication on the result of a division:
☒-reward = amount.mul(numerator).div(denominator).mul(rewardCoefficient).div(10000)
(contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#340)
BULLETH1X._calculateTax(uint256,uint256,uint256) (contracts/TOKEN/TOKEN/ETH/
BULLETH1X.sol#308-313) performs a multiplication on the result of a division:
☒-tax = amount.mul(numerator).div(denominator).mul(taxCoefficient).div(10000) (contracts/
TOKEN/TOKEN/ETH/BULLETH1X.sol#312)
BULLETH1X._calculateReward(uint256,uint256,uint256) (contracts/TOKEN/TOKEN/ETH/
BULLETH1X.sol#336-341) performs a multiplication on the result of a division:
☒-reward = amount.mul(numerator).div(denominator).mul(rewardCoefficient).div(10000)
(contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#340)

```

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply>

BebuRewardPool.updatePool(uint256) (contracts/Reward_pool/BebuRewardPool.sol#201-221) uses a dangerous strict equality:

☒- tokenSupply == 0 (contracts/Reward_pool/BebuRewardPool.sol#207)

StableFarmingRewardPool.updatePool(uint256) (contracts/Reward_pool/StableFarmingRewardPool.sol#291-311) uses a dangerous strict equality:

☒- tokenSupply == 0 (contracts/Reward_pool/StableFarmingRewardPool.sol#297)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities>

Reentrancy in Boardroom.blacklistFundWithdraw(address,address) (contracts/Boardroom/Boardroom.sol#270-285):

☒External calls:

☒- token.safeTransfer(_to,reward) (contracts/Boardroom/Boardroom.sol#278)

☒State variables written after the call(s):

☒- _balances[_address] = 0 (contracts/Boardroom/Boardroom.sol#282)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1>

Boardroom.setFee(uint256) (contracts/Boardroom/Boardroom.sol#138-141) contains a tautology or contradiction:

☒- require(bool,string)(_fee >= 0 && _fee <= 10000,out of range) (contracts/Boardroom/Boardroom.sol#139)

StableFarmingRewardPool.setPeriod(uint256) (contracts/Reward_pool/StableFarmingRewardPool.sol#125-128) contains a tautology or contradiction:

☒- require(bool,string)(_period >= 0 && _period <= 86400,out of range) (contracts/Reward_pool/StableFarmingRewardPool.sol#126)

StableFarmingRewardPool.setFee(uint256) (contracts/Reward_pool/StableFarmingRewardPool.sol#136-139) contains a tautology or contradiction:

☒- require(bool,string)(_fee >= 0 && _fee <= 10000,out of range) (contracts/Reward_pool/StableFarmingRewardPool.sol#137)

BEARETH1X.setTaxCoefficient(uint256) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#182-185) contains a tautology or contradiction:

☒- require(bool,string)(_taxCoefficient >= 0 && _taxCoefficient <= 20000,out of range) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#183)

BEARETH1X.setRewardCoefficient(uint256) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#205-208) contains a tautology or contradiction:

☒- require(bool,string)(_rewardCoefficient >= 0 && _rewardCoefficient <= 20000,out of range) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#206)

BEARETH1X.setTaxThreshold(uint256) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#265-268) contains a tautology or contradiction:
 ☒- require(bool,string)(_TaxThreshold >= 0 && _TaxThreshold <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#266)

BEARETH1X.setRewardThreshold(uint256) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#270-273) contains a tautology or contradiction:
 ☒- require(bool,string)(_RewardThreshold >= 0 && _RewardThreshold <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#271)

BEARETH1X.setBurnRate(uint256) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#275-278) contains a tautology or contradiction:
 ☒- require(bool,string)(_burnRate >= 0 && _burnRate <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#276)

BULLETH1X.setTaxCoefficient(uint256) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#182-185) contains a tautology or contradiction:
 ☒- require(bool,string)(_taxCoefficient >= 0 && _taxCoefficient <= 20000,out of range) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#183)

BULLETH1X.setRewardCoefficient(uint256) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#205-208) contains a tautology or contradiction:
 ☒- require(bool,string)(_rewardCoefficient >= 0 && _rewardCoefficient <= 20000,out of range) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#206)

BULLETH1X.setTaxThreshold(uint256) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#265-268) contains a tautology or contradiction:
 ☒- require(bool,string)(_TaxThreshold >= 0 && _TaxThreshold <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#266)

BULLETH1X.setRewardThreshold(uint256) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#270-273) contains a tautology or contradiction:
 ☒- require(bool,string)(_RewardThreshold >= 0 && _RewardThreshold <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#271)

BULLETH1X.setBurnRate(uint256) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#275-278) contains a tautology or contradiction:
 ☒- require(bool,string)(_burnRate >= 0 && _burnRate <= 10000,out of range) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#276)

BEARETH1X_Treasury.setPeriod(uint256) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#193-197) contains a tautology or contradiction:
 ☒- require(bool,string)(_period >= 0 && _period <= 172800,_period: out of range) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#194)

BEARETH1X_Treasury.setSupplyTiersEntry(uint8,uint256) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#232-243) contains a tautology or contradiction:
 ☒- require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#233)

BEARETH1X_Treasury.setMaxExpansionTiersEntry(uint8,uint256) (contracts/Treasury/

ETH_Treasury/BEARETH1X_Treasury.sol#245-251) contains a tautology or contradiction:
 ☒- require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#246)
 BEARETH1X_Treasury._calculateMaxSupplyExpansionPercent(uint256) (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#320-328) contains a tautology or contradiction:
 ☒- tierId >= 0 (contracts/Treasury/ETH_Treasury/BEARETH1X_Treasury.sol#321)
 BULLETH1X_Treasury.setPeriod(uint256) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#193-197) contains a tautology or contradiction:
 ☒- require(bool,string)(_period >= 0 && _period <= 172800,_period: out of range) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#194)
 BULLETH1X_Treasury.setSupplyTiersEntry(uint8,uint256) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#232-243) contains a tautology or contradiction:
 ☒- require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#233)
 BULLETH1X_Treasury.setMaxExpansionTiersEntry(uint8,uint256) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#245-251) contains a tautology or contradiction:
 ☒- require(bool,string)(_index >= 0,Index has to be higher than 0) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#246)
 BULLETH1X_Treasury._calculateMaxSupplyExpansionPercent(uint256) (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#314-322) contains a tautology or contradiction:
 ☒- tierId >= 0 (contracts/Treasury/ETH_Treasury/BULLETH1X_Treasury.sol#315)
 Epoch.setPeriod(uint256) (contracts/Utils/access/Epoch.sol#78-81) contains a tautology or contradiction:
 ☒- require(bool,string)(_period >= 0 && _period <= 172800,_period: out of range) (contracts/Utils/access/Epoch.sol#79)
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction>

BEARETH1X._gettokenPrice()._price (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#213) is a local variable never initialized
 BEARETH1X._getTokenTwapPrice()._price (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#221) is a local variable never initialized
 BULLETH1X._gettokenPrice()._price (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#213) is a local variable never initialized
 FixedPoint.mul(FixedPoint.uq112x112,uint256).z (contracts/Utils/math/FixedPoint.sol#44) is a local variable never initialized
 BULLETH1X._getTokenTwapPrice()._price (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#221) is a local variable never initialized
 UniswapV2Library.getAmountsOut(address,uint256,address[]).i (contracts/Utils/library/UniswapV2Library.sol#72) is a local variable never initialized
 BEARETH1X_Treasury.getTokenUpdatedPrice().price (contracts/Treasury/ETH_Treasury/

BEARETH1X_Treasury.sol#151) is a local variable never initialized
 BEARETH1X_Treasury.getTokenPrice().price (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#143) is a local variable never initialized
 BULLETH1X_Treasury.getTokenUpdatedPrice().price (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#151) is a local variable never initialized
 BULLETH1X_Treasury.getTokenPrice().price (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#143) is a local variable never initialized
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables>

BEARETH1X._getTokenPrice() (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#212-218) ignores return value by IOracle(tokenOracle).consult(address(this),1e18) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#213-217)
 BEARETH1X._getTokenTwapPrice() (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#220-226) ignores return value by IOracle(tokenOracle).twap(address(this),1e18) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#221-225)
 BEARETH1X.isPair(address) (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#86-105) ignores return value by IUniswapV2Router01(_recipient).factory() (contracts/TOKEN/TOKEN/ETH/BEARETH1X.sol#95-99)
 BULLETH1X._getTokenPrice() (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#212-218) ignores return value by IOracle(tokenOracle).consult(address(this),1e18) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#213-217)
 BULLETH1X._getTokenTwapPrice() (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#220-226) ignores return value by IOracle(tokenOracle).twap(address(this),1e18) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#221-225)
 BULLETH1X.isPair(address) (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#86-105) ignores return value by IUniswapV2Router01(_recipient).factory() (contracts/TOKEN/TOKEN/ETH/BULLETH1X.sol#95-99)
 BEARETH1X_Treasury.getTokenPrice() (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#142-148) ignores return value by
 IOracle(tokenOracle).consult(token,1e18) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#143-147)
 BEARETH1X_Treasury.getTokenUpdatedPrice() (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#150-157) ignores return value by
 IOracle(tokenOracle).twap(token,1e18) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#151-155)
 BEARETH1X_Treasury._sendToBoardroom(uint256) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#264-287) ignores return value by
 IBasisAsset(token).mint(address(this),_amount) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#265)
 BEARETH1X_Treasury.boardroomAddAddressBlacklist(address) (contracts/Treasury/

ETH_Treasury/BEARETH1X_Treasury.sol#364-366) ignores return value by
 IBoardroom(boardroom).addAddressBlacklist(_address) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#365)
 BEARETH1X_Treasury.boardroomRemoveAddressBlacklist(address) (contracts/Treasury/
 ETH_Treasury/BEARETH1X_Treasury.sol#368-370) ignores return value by
 IBoardroom(boardroom).removeAddressBlacklist(_address) (contracts/Treasury/ETH_Treasury/
 BEARETH1X_Treasury.sol#369)
 BULLETH1X_Treasury.getTokenPrice() (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#142-148) ignores return value by
 IOracle(tokenOracle).consult(token,1e18) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#143-147)
 BULLETH1X_Treasury.getTokenUpdatedPrice() (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#150-157) ignores return value by
 IOracle(tokenOracle).twap(token,1e18) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#151-155)
 BULLETH1X_Treasury._sendToBoardroom(uint256) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#264-287) ignores return value by
 IBasisAsset(token).mint(address(this),_amount) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#265)
 BULLETH1X_Treasury.boardroomAddAddressBlacklist(address) (contracts/Treasury/
 ETH_Treasury/BULLETH1X_Treasury.sol#362-364) ignores return value by
 IBoardroom(boardroom).addAddressBlacklist(_address) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#363)
 BULLETH1X_Treasury.boardroomRemoveAddressBlacklist(address) (contracts/Treasury/
 ETH_Treasury/BULLETH1X_Treasury.sol#366-368) ignores return value by
 IBoardroom(boardroom).removeAddressBlacklist(_address) (contracts/Treasury/ETH_Treasury/
 BULLETH1X_Treasury.sol#367)
 Reference: <https://github.com/crytic/sliether/wiki/Detector-Documentation#unused-return>



 Guard