Bido

Feedbacks

Fri Mar 15 2024

✉ contact@scalebit.xyz https://twitter.com/scalebit_

ScaleBit

1 Summary

During the audit, we identified 6 issues of varying severity, listed below.

ID	Title	Severity	Status
BID–1	Unused Libraries	Minor	Pending
BID–2	Redundant Checks	Minor	Pending
BID–3	`_unstake` Emit Wrong Event	Minor	Pending
BID–4	Use `!= 0` instead of `> 0` for Unsigned Integer Comparison	Minor	Pending
SBT–1	Uncalled Initialization Function	Minor	Pending
WBT–1	Immutable Parameters	Minor	Pending

2 Findings

BID–1 Unused Libraries

Severity: Minor

Status: Pending

Code Location:

Bido.sol#6

Descriptions:

The `Math256` library is imported into the contract, but it is not used.

Suggestion:

It is recommended to delete unused libraries.

BID-2 Redundant Checks

Severity: Minor

Status: Pending

Code Location:

Bido.sol#137

Descriptions:

Before the start of `staking`, you need to call the `initialize` function to initialize the contract. At this time, there is already a fund in the contract, recorded at the address `0xdead`, and no one can withdraw this fund. When calling the `stake` function, you also need to pass a fund, so the funds in the contract at this time are equal to the initial funds plus the funds passed in by the user. This value is always greater than the funds passed in by the user.

Suggestion:

It is recommended to delete redundant require checks.

BID-3 `_unstake` Emit Wrong Event

Severity: Minor

Status: Pending

Code Location:

Bido.sol#175

Descriptions:

In the `_unstake` function, it will emit the `UnStaked` event with the parameters of the caller and the number of unstakes. However, the unstake amount here uses `msg.value`. In this context, `msg.value` is always 0. which will confuse the users and disturb the on-chain data.

Suggestion:

It is recommended to change the unstake amount in the `_unstake` event to the correct parameter.

BID-4 Use `!= 0` instead of `> 0` for Unsigned Integer Comparison

Severity: Minor

Status: Pending

Code Location:

Bido.sol#157,166

Descriptions:

When dealing with unsigned integer types, comparisons with `!= 0` are cheaper than with `> 0`.

```
require(_amount > 0, "UNSTAKE_ZERO");
```

Suggestion:

It is recommended to use `!= 0` instead of `> 0` for unsigned integer comparison.

SBT-1 Uncalled Initialization Function

Severity: Minor

Status: Pending

Code Location:

StBTCPermit.sol#193-200

Descriptions:

The `_initializeEIP712StBTC` function is defined in the `StBTCPermit` contract. This function is used to initialize the address of the `eip712StBTC` contract. However, this function is `internal` and there is no superior function to call it, which will cause `_initializeEIP712StBTC` to be unavailable.

Suggestion:

It is recommended to confirm whether this situation conflicts with the design concept.

WBT-1 Immutable Parameters

Severity: Minor

Status: Pending

Code Location:

WstBTC.sol#27

Descriptions:

The `stBTC` parameter is defined in the contract. This parameter will only be initialized in the `constructor` and will not be changed subsequently.

Suggestion:

It is recommended to change this parameter to an `immutable` type.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non–exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.