SALUS SECURITY

SEP 2024

CODE SECURITY ASSESSMENT

LORENZO PROTOCOL

Overview

Project Summary

- Name: Lorenzo StakePlan - Incremental Audit
- Platform: EVM-compatible chains
- Language: Solidity
- Repository:
  - https://github.com/Lorenzo-Protocol/BNB_YAT_StakePlan
- Audit Range: See Appendix - 1

Project Dashboard

Application Summary

Name	Lorenzo StakePlan - Incremental Audit
Version	v2
Type	Solidity
Dates	Sep 30 2024
Logs	Sep 30 2024; Sep 30 2024

Vulnerability Summary

Total High-Severity issues	0
Total Medium-Severity issues	1
Total Low-Severity issues	0
Total informational issues	1
Total	2

Contact

E-mail: support@salusec.io

Risk Level Description

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for clients' reputations or serious financial implications for clients and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental to the client's reputation if exploited, or is reasonably likely to lead to a moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
Informational	The issue does not pose an immediate risk, but is relevant to security best practices or defense in depth.

Content

Introduction

1.1 About SALUS

At Salus Security, we are in the business of trust.

We are dedicated to tackling the toughest security challenges facing the industry today. By building foundational trust in technology and infrastructure through security, we help clients to lead their respective industries and unlock their full Web3 potential.

Our team of security experts employ industry-leading proof-of-concept (PoC) methodology for demonstrating smart contract vulnerabilities, coupled with advanced red teaming capabilities and a stereoscopic vulnerability detection service, to deliver comprehensive security assessments that allow clients to stay ahead of the curve.

In addition to smart contract audits and red teaming, our Rapid Detection Service for smart contracts aims to make security accessible to all. This high calibre, yet cost-efficient, security tool has been designed to support a wide range of business needs including investment due diligence, security and code quality assessments, and code optimisation.

We are reachable on Telegram (https://t.me/salusec), Twitter (https://twitter.com/salus_sec), or Email (support@salusec.io).

1.2 Audit Breakdown

The objective was to evaluate the repository for security-related issues, code quality, and adherence to specifications and best practices. Possible issues we looked for included (but are not limited to):

- Risky external calls
- Integer overflow/underflow
- Transaction-ordering dependence
- Timestamp dependence
- Access control
- Call stack limits and mishandled exceptions
- Number rounding errors
- Centralization of power
- Logical oversights and denial of service
- Business logic specification
- Code clones, functionality duplication

1.3 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release and does not give any warranties on finding all possible security issues with the given smart contract(s) or blockchain software, i.e., the evaluation result does not guarantee the nonexistence of any further findings of security issues.

Findings

2.1 Summary of Findings

ID	Title	Severity	Category	Status
1	Centralization risk	Medium	Centralization	Acknowledged
2	Missing return value check	Informational	Code Quality	Acknowledged

2.2 Notable Findings

Significant flaws that impact system confidentiality, integrity, or availability are listed below.

1. Centralization risk	
Severity: Medium	Category: Centralization
Target: - contracts/StakePlan/StakePlanHub.sol	

Description

In the `StakePlanHub` contract, there exists a privileged role called `admin` and `gov`. The `admin` has the authority to execute some key functions such as `adminPause`, `adminUnpause`, `setBTCCustodyAddress`.

If the `admin` or `gov`'s private key is compromised, an attacker could trigger these functions to steal staking tokens from the contract.

Recommendation

We recommend transferring privileged accounts to multi-sig accounts with timelock governors for enhanced security. This ensures that no single person has full control over the accounts and that any changes must be authorized by multiple parties.

Status

This issue has been acknowledged by the team.

2.3 Informational Findings

2. Missing return value check	
Severity: Informational	Category: Code Quality
Target: - contracts/StakePlan/StakePlanHub.sol	

Description

The government role will incorporate the capability to add supported Bitcoin contract addresses via the `` `addSupportBtcContractAddress` `` function. However, in the event that an attempt is made to add an already existing Bitcoin contract address, the add function will return false. In such cases, it is unnecessary to emit an event, thereby avoiding unnecessary system responses and informational redundancy.

contracts/StakePlan/StakePlanHub.sol:L200-L211

```
function addSupportBtcContractAddress(
    address[] memory btcContractAddress_
) external onlyGov {
    for (uint256 i = 0; i < btcContractAddress_.length; i++) {
        address btcContractAddress = btcContractAddress_[i];
        if (btcContractAddress == address(0)) {
            revert InvalidAddress();
        }
        _btcContractAddressSet.add(btcContractAddress);
        emit BTCContractAddressAdd(btcContractAddress);
    }
}
```

Recommendation

Check the `` `add` ``'s return value, do not emit the event if the return value is false.

Status

This issue has been acknowledged by the team.

Appendix

Appendix 1 - Files in Scope

This audit covered the following files in commit d3c495e:

File	SHA-1 hash
StakePlan.sol	57b1c3990b107077b710c2e69698059595f40529
StakePlanHub.sol	84405257129054397f3c8313c61b728087701387
stBTCMintAuthority.sol	85abeca2bc78ddddf01e9c1c32f759cab937ebea
StakePlanHubStorage.sol	9acf0da5e6f5a87c03a884a7f905e1ed9ce2e27d

And we audit the following files in commit 26fc320:

File	SHA-1 hash
StakePlanHub.sol	c01fbdaf2dcc9581fcb9adea1d590f7a63761988
stBTCMintAuthority.sol	85abeca2bc78ddddf01e9c1c32f759cab937ebea
StakePlanHubStorage.sol	46255c9f3a24af80dd4dea74878ff455a2b3b2f3