Lista DAO - audit

Security Assessment

CertiK Assessed on Apr 17th, 2025

CertiK Assessed on Apr 17th, 2025

Lista DAO - audit

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
Others	EVM Compatible	Formal Verification, Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 04/17/2025	N/A

CODEBASE

https://github.com/lista-dao/lista-token/tree/feature/lp-mint-clisbnb/contracts/dao

View All in Codebase Page

COMMITS

- e7f7157db05f25631f348c3c8ed4bd47c3da0d1a
- 327eda26b14c87d9e5ecb79b430388a3529ffd8c
- caefade1a626fde15a3f96d2cf6fa8a342bc6a03

View All in Codebase Page

Highlighted Centralization Risks

Contract upgradeability

Withdraws can be disabled

Vulnerability Summary

8	3	0	5	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

2	Centralization	2 Acknowledged	Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
0	Major		Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
2	Medium	1 Resolved, 1 Acknowledged	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
4	Minor	2 Resolved, 2 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.

0 Informational	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | LISTA DAO - AUDIT

CODEBASE | LISTA DAO - AUDIT

Repository

https://github.com/lista-dao/lista-token/tree/feature/lp-mint-clisbnb/contracts/dao

Commit

- e7f7157db05f25631f348c3c8ed4bd47c3da0d1a
- 327eda26b14c87d9e5ecb79b430388a3529ffd8c
- caefade1a626fde15a3f96d2cf6fa8a342bc6a03

AUDIT SCOPE | LISTA DAO - AUDIT

5 files audited ● 1 file with Acknowledged findings ● 1 file with Resolved findings ● 3 files without findings

ID	Repo	File	SHA256 Checksum
ERC	lista-dao/lista-token	erc20LpProvider/ERC20LpTokenProvider.sol	b74c5b36428f63a2637fba5ba1ab053dcafed1c8f6efd247d8c03547c313d078
ILT	lista-dao/lista-token	interfaces/ILpToken.sol	12052bf4b497e1d80fc94db46b00fdbd6bc1e6fd2b2dab77b380afa0a7173f66
IER	lista-dao/lista-token	interfaces/IERC20TokenProvider.sol	fe01a5f609a1af4b5aae504dbc206c03d79ab31319d5e20eba6166ea8e55dc14
ITE	lista-dao/lista-token	interfaces/IThenaErc20LpToken.sol	eb873604535c8e48ee0af62ecfe750e8586e241496432b166c2f0a1d5057970b
ISS	lista-dao/lista-token	interfaces/IStableSwap.sol	bdc1cc34f029047b530cc0e392deb7fd2b81ae5acd54c1de1af46064c8c8dbdb

APPROACH & METHODS | LISTA DAO - AUDIT

This report has been prepared for Lista DAO to discover issues and vulnerabilities in the source code of the Lista DAO - audit project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Formal Verification, Manual Review, and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | LISTA DAO - AUDIT

Out-of-Scope Components

The contract imported within the `ERC20LpTokenProvider` contract is not included in the current audit scope. The audit team assumes it has been implemented securely.

- `lpProvidableDistributor`
- `lpToken`

The MPC wallet `lpReserveAddress`, within the `ERC20LpTokenProvider` contract, is used to store reserve assets. The audit team assumes it has been implemented securely.

FINDINGS | LISTA DAO - AUDIT

8	0	2	0	2	4	0
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Lista DAO - audit. Through this audit, we have uncovered 8 issues ranging from different severity levels. Utilizing the techniques of Formal Verification, Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
ERC-01	**Centralization Related Risks**	**Centralization**	**Centralization**	**Acknowledged**
ERC-02	**Centralized Control Of Contract Upgrade**	**Centralization**	**Centralization**	**Acknowledged**
LDA-01	`clisXXX` Balance May Become Outdated	Design Issue	Medium	Acknowledged
LDA-03	Inconsistent Delegation Handling Leads To Reverts And Invalid `lpToken` Balances	Design Issue	Medium	Resolved
ERC-03	Inconsistency Between `newTotalLp` Value And Comment	Coding Issue	Minor	Resolved
ERC-04	Third-Party Dependencies	Volatile Code	Minor	Acknowledged
ILT-01	`clisBNB` Does Not Implement The `ILpToken` Interface	Coding Issue	Minor	Resolved
LDA-02	Potential Underflow In `totalReservedLp` In `_rebalanceUserLp()`	Logical Issue	Minor	Acknowledged

ERC-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	Centralization	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 363, 370, 381, 394, 401, 409	Acknowledged

Description

In the contract `AccessControlUpgradeable` the role `adminRole` has authority over the following functions:

- grantRole()
- revokeRole()

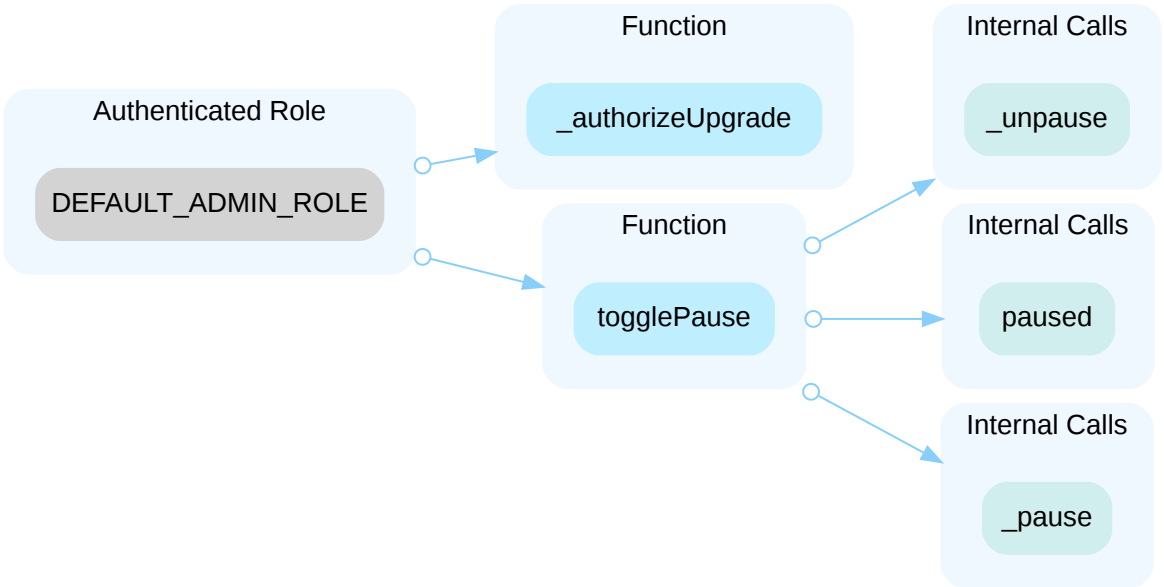
Any compromise to the `adminRole` account may allow the hacker to take advantage of this authority and grant associated role to any account or revoke the role from any account. Note that `DEFAULT_ADMIN_ROLE` is the admin role for all roles.

In the contract `AccessControlUpgradeable` the role `role` has authority over the following function:

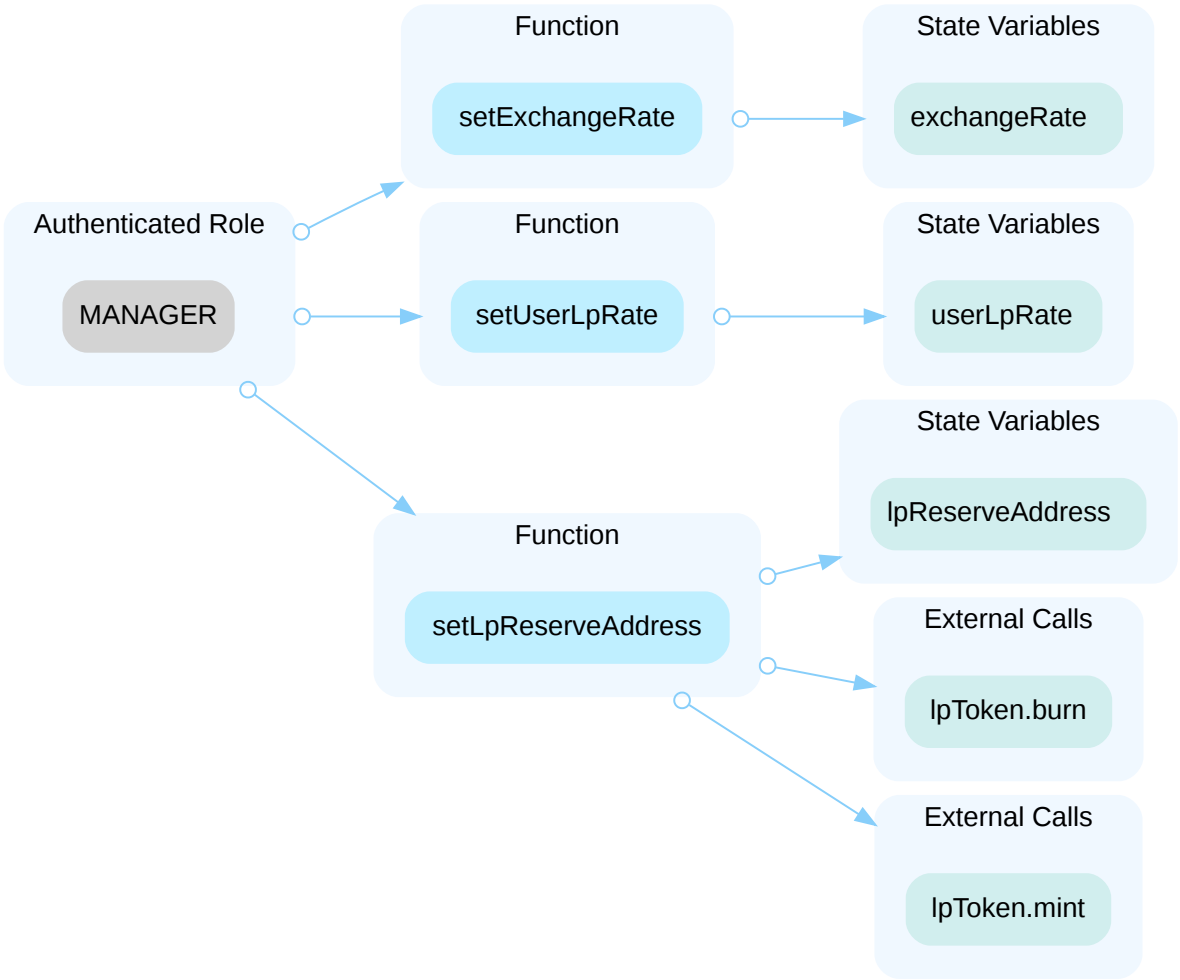
- renounceRole()

Any compromise to the `role` account may allow the hacker to take advantage of this authority and renounce corresponding privileges to functions within other contracts.

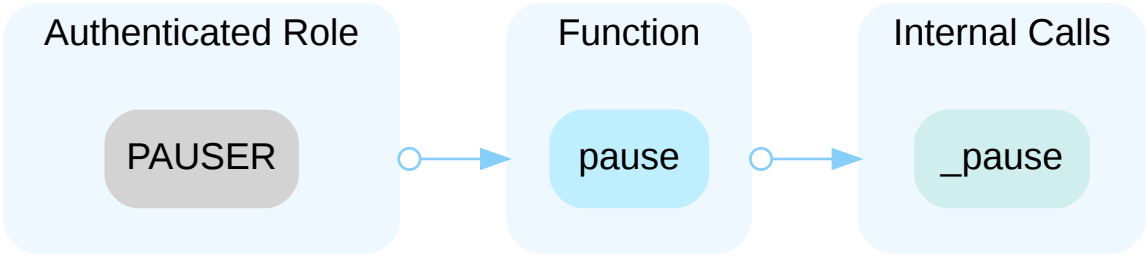
In the contract `ERC20LpTokenProvider`, the role `DEFAULT_ADMIN_ROLE` has authority over the functions shown in the diagram below. Any compromise to the `DEFAULT_ADMIN_ROLE` account may allow the hacker to take advantage of this authority and authorize contract upgrades with admin role, as well as toggle the contract pause state.



In the contract `ERC20LpTokenProvider`, the role `MANAGER` has authority over the functions shown in the diagram below. Any compromise to the `MANAGER` account may allow the hacker to take advantage of this authority and set the exchange rate, set the user liquidity provider rate, or set the LP reserve address.



In the contract `ERC20LpTokenProvider`, the role `PAUSER` has authority over the functions shown in the diagram below. Any compromise to the `PAUSER` account may allow the hacker to take advantage of this authority and pause contract execution.



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts

with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

[Lista DAO Team, 04/08/2025]:

We will ensure all deployed/pending-deploy contracts are owned and controlled by a TimeLock contract, a 3/6 multi-sig wallets act as the proposer and executor as well.

Only the TimeLock contract can perform contract upgrade, and the multi-sig wallet can call functions that requires the MANAGER role.

TimeLock: 0x07D274a68393E8b8a2CCf19A2ce4Ba3518735253

Multi-sig: 0x8d388136d578dcd791d081c6042284ced6d9b0c6

[CertiK, 04/08/2025]:

It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

ERC-02 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	Centralization	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 33	Acknowledged

Description

The `ERC20LpTokenProvider` contract functions as the implementation contract for its proxy, with the `DEFAULT_ADMIN_ROLE` having the authority to update the implementation contract of the proxy.

Any compromise of the `DEFAULT_ADMIN_ROLE` account could allow a hacker to exploit this authority, changing the implementation contract referenced by the proxy and potentially executing malicious functionality in the implementation contract.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (⅔, ⅗) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
  AND
- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.

- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
  AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
  AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

- Renounce the ownership and never claim back the privileged role;
  OR
- Remove the risky functionality.

Note: we recommend the project team consider the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Lista DAO Team, 04/08/2025]:

We will ensure all deployed/pending-deploy contracts are owned and controlled by a TimeLock contract, a 3/6 multi-sig wallets act as the proposer and executor as well.

Only the TimeLock contract can perform contract upgrade, and the multi-sig wallet can call functions that requires the MANAGER role.

TimeLock: 0x07D274a68393E8b8a2CCf19A2ce4Ba3518735253

Multi-sig: 0x8d388136d578dcd791d081c6042284ced6d9b0c6

[CertiK, 04/08/2025]:

It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

LDA-01 | `clisXXX` BALANCE MAY BECOME OUTDATED

Category	Severity	Location	Status
Design Issue	Medium	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 363, 370	Acknowledged

Description

An account's `clisXXX` balance reflects the `userStakedTokenAmount` when the `_rebalanceUserLp(address account)` function is called. However, since the `userStakedTokenAmount` is dynamic and changes based on the value of deposited LP tokens, as well as the `exchangeRate` and `userLpRate` whose values can be updated, the `clisXXX` balance remains static until the user interacts with the contract to sync it again.

Recommendation

If an account's `clisXXX` balance is intended to sync with the `userStakedTokenAmount` value, it is recommended to review the design and modify the code if necessary.

Alleviation

[Lista DAO Team, 04/08/2025]: The `clisXXX` balance will be updated by our off-chain service by calling the `syncUserLp()` function in case the balance is outdated.

LDA-03 INCONSISTENT DELEGATION HANDLING LEADS TO REVERTS AND INVALID `lpToken` BALANCES

Category	Severity	Location	Status
Design Issue	Medium	erc20LpProvider/ERC20LpTokenProvider.sol (remediation): 201, 298	Resolved

Description

In the `ERC20LpTokenProvider` contract, users who have directly staked into the `lpProvidableDistributor` contract without specifying a delegatee (i.e., `delegation[account] == address(0)`) can encounter critical issues that block future actions or cause inconsistent `lpToken` (`clisXXX`) balances.

Issue 1: Revert in `delegateAllTo()` When `userLp[account] == 0`

```
address oldDelegatee = delegation[msg.sender]; // = address(0)
_safeBurnLp(oldDelegatee, userLp[msg.sender]); // burns from address(0)
```

If a user has not set a delegatee (`delegation[msg.sender] == address(0)`) and has not called `syncUserLp()`, `deposit(uint256 _amount)`, `deposit(_amount, _delegateTo)`, or `withdraw()` to update `userLp`, calling `delegateAllTo()` results in an attempt to burn `lpToken` from `address(0)`, leading to a revert.

Issue 2: Inconsistent `lpToken` Balances and Delegation State

A user can:

1. Call `syncUserLp()` or `withdraw()` to update `userLp[account]` and mint `lpToken` to themselves without a delegatee set.
2. Later call `deposit(_amount, _delegateTo)` to deposit more and set a new delegatee.

Since the `deposit(_amount, _delegateTo)` logic only mints `lpToken` for the newly deposited amount to the new delegatee, the previously minted `lpToken` remains with the original user. This causes an inconsistent state where:

- The user holds `lpToken`, but their delegatee is no longer themselves.
- Future calls to `delegateAllTo()` or `withdraw()` revert when trying to burn `lpToken` from the delegatee who does not hold enough tokens.

Impact

- Users may get permanently **locked out** of `delegateAllTo()` and `withdraw()` due to invalid `lpToken` states.
- `lpToken` accounting becomes **inconsistent**, with tokens held by addresses that are no longer aligned with the delegation logic.

- Causes **unexpected reverts**, blocking user interactions.

Recommendation

1. Make sure `delegation[msg.sender]` is not `address(0)` before the `_safeBurnLp(oldDelegatee, userLp[msg.sender])` function call.
2. If `delegation[account]` is `address(0)`, update `delegation[account]` to be the `account` itself in the end of `_rebalanceUserLp()` function call.

Alleviation

[Lista DAO Team, 04/18/2025]:

The team heeded the advice and resolved the "issue 2" in commit caefade1a626fde15a3f96d2cf6fa8a342bc6a03.

For issue 1, after the TokenProvider is deployed we will run an off-chain service to call the `bultSyncUserLp` function timely to initialize `userLp[account]` and `delegation[account]`. Before the initialization, `delegateAllTo()` will revert if user tries to call it which is expected.

ERC-03 INCONSISTENCY BETWEEN `newTotalLp` VALUE AND COMMENT

Category	Severity	Location	Status
Coding Issue	Minor	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 300~301	Resolved

Description

The `newTotalLp` represents the total amount of newly minted `clisXXX` tokens, divided into two parts: User and Reserve, upon deposit. However, the comment at L300 states that `newTotalLp` includes three parts: Lista, User, and Reserve.

```
300         // ---- [1] Estimated LP value
301         // Total LP(Lista + User + Reserve)
302         uint256 newTotalLp = userStakedTokenAmount * exchangeRate /
RATE_DENOMINATOR;
303         // User's LP
304         uint256 newUserLp = userStakedTokenAmount * userLpRate /
RATE_DENOMINATOR;
305         // Reserve's LP
306         uint256 newReservedLp = newTotalLp - newUserLp;
```

The audit team would like to ask the development team if the current code logic aligns with the original design.

Recommendation

It is recommended to revise the code to eliminate the inconsistency.

Alleviation

[Lista DAO Team, 04/08/2025]: The team heeded the advice and resolved the issue in commit 7822ff96ebc4243a7084447698585c2c2b6e8bec.

ERC-04 | THIRD-PARTY DEPENDENCIES

Category	Severity	Location	Status
Volatile Code	Minor	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 18, 48	Acknowledged

Description

The contract is serving as the underlying entity to interact with third-party **PancakeSwap, Thena** protocols. The scope of the audit treats third-party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets.

Recommendation

We recommend that the project team constantly monitor the functionality of the **PancakeSwap, Thena** protocols to mitigate any side effects that may occur when unexpected changes are introduced.

Alleviation

[Lista DAO Team, 04/08/2025]: We have an off-chain service monitors PancakeSwap and the Thena Protocol continuously, alerts will be prompted in time if there are any suspicious activity, also the pause() function allows us to halt the protocol in time to make sure user's fund is safe.

ILT-01 | `clisBNB` DOES NOT IMPLEMENT THE `ILpToken` INTERFACE

Category	Severity	Location	Status
Coding Issue	Minor	interfaces/ILpToken.sol (pre): 9	Resolved

Description

The audit team has observed that the *`clisBNB` contract*, which is expected to implement the `ILpToken` interface, is referenced in the test file. However, the `clisBNB` contract does not define the following functions:

- `balanceWithRewardsOf()`
- `isRebasing()`
- `ratio()`
- `bondsToShares()`

test/ThenaStakingDistributor.t.sol

```
44      // LP token of Provider
45      IClisBNB clisBNB = IClisBNB(0x4b30fcAA7945fE9fDEFD2895aae539ba102Ed6F6);
46      address clisBNBOwner = 0x702115D6d3Bbb37F407aae4dEcf9d09980e28ebc;
```

contracts/dao/interfaces/ILpToken.sol

```
 6  /**
 7   * @dev Interface of the ERC20 standard as defined in the EIP.
 8   */
 9  interface ILpToken is IERC20 {
10
11      function burn(address account, uint256 amount) external;
12
13      function mint(address account, uint256 amount) external;
14
15      function balanceWithRewardsOf(address account) external returns (uint256);
16
17      function isRebasing() external returns (bool);
18
19      function ratio() external view returns (uint256);
20
21      function bondsToShares(uint256 amount) external view returns (uint256);
22
23      function decimals() external view returns (uint8);
24  }
```

Recommendation

It is recommended to revise the `ILpToken` interface if `clisBNB` should implement it.

Alleviation

[Lista DAO Team, 04/08/2025]: The team heeded the advice and resolved the issue in commit 1c80f2a34fc1d664dc69a351d9ec04b1dda29fc3.

LDA-02 POTENTIAL UNDERFLOW IN `totalReservedLp` IN `_rebalanceUserLp()`

Category	Severity	Location	Status
Logical Issue	Minor	erc20LpProvider/ERC20LpTokenProvider.sol (pre): 318~319	Acknowledged

Description

In the `_rebalanceUserLp()` function, the logic attempts to burn `oldReservedLp - newReservedLp` LP tokens from the `lpReserveAddress` and decrement `totalReservedLp` by the same amount.

The `_safeBurnLp()` function accounts for edge cases where `oldReservedLp - newReservedLp` exceeds the actual LP token balance of the `lpReserveAddress`. In such cases, it simply burns the full balance available.

However, this condition is **not mirrored** in the `totalReservedLp` calculation. Even if fewer tokens are burned due to a balance shortfall, the contract still subtracts the full `oldReservedLp - newReservedLp` from `totalReservedLp`, which could cause an **underflow error**.

```
318             _safeBurnLp(lpReserveAddress, oldReservedLp - newReservedLp);
319             totalReservedLp -= (oldReservedLp - newReservedLp);
```

Recommendation

While this issue is currently low risk due to `lpToken` being **non-transferable**, and thus external manipulation is unlikely, the inconsistency could become problematic if future changes enable token transfers or if internal logic evolves.

Alleviation

[Lista DAO Team, 04/08/2025]: The team acknowledged the finding and decided not to change the current codebase.

APPENDIX | LISTA DAO - AUDIT

Finding Categories

Categories	Description
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

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 you ("Customer" or 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, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

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 CertiK 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.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Elevating Your Entire Web3 Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.