

Security Assessment: Chainbased TOKEN

November 21, 2024

- Audit Status: **Fail**
- Audit Edition: **Advance**
































Risk Analysis

Classifications of Manual Risk Results

Classification	Description
 Critical	Danger or Potential Problems.
 High	Be Careful or Fail test.
 Medium	Pass, Not-Detected or Safe Item.
 Low	Function Detected

Manual Code Review Risk Results

Contract Privilege	Description
 Buy Tax	1 %
 Sale Tax	1 %
 Cannot Buy	Pass
 Cannot Sale	Pass
 Max Tax	1.2%
 Modify Tax	Yes
 Fee Check	Pass
 Is Honeypot?	Not Detected
 Trading Cooldown	Not Detected
 Can Pause Trade?	Fail
 Pause Transfer?	Not-Detected
 Max Tx?	Pass
 Is Anti Whale?	Not-Detected
 Is Anti Bot?	Not-Detected

Contract Privilege	Description
 Is Blacklist?	Not-Detected
 Blacklist Check	Pass
 is Whitelist?	Not-Detected
 Can Mint?	Pass
 Is Proxy?	Not Detected
 Can Take Ownership?	Not Detected
 Hidden Owner?	Not-Detected
 Owner	No
 Self Destruct?	Not Detected
 External Call?	Not-Detected
 Other?	Not Detected
 Holders	0
 Auditor Confidence	Medium
 KYC Present	No
 KYC URL	

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	
Name	Chainbased
Token Tracker	Chainbased (BASD)
Decimals	9
Supply	500,000,000
Platform	ARBITRUM
compiler	v0.8.20+commit.a1b79de6
Contract Name	BASD
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed Contract Name

Name	Contract	Live
Chainbased		Yes

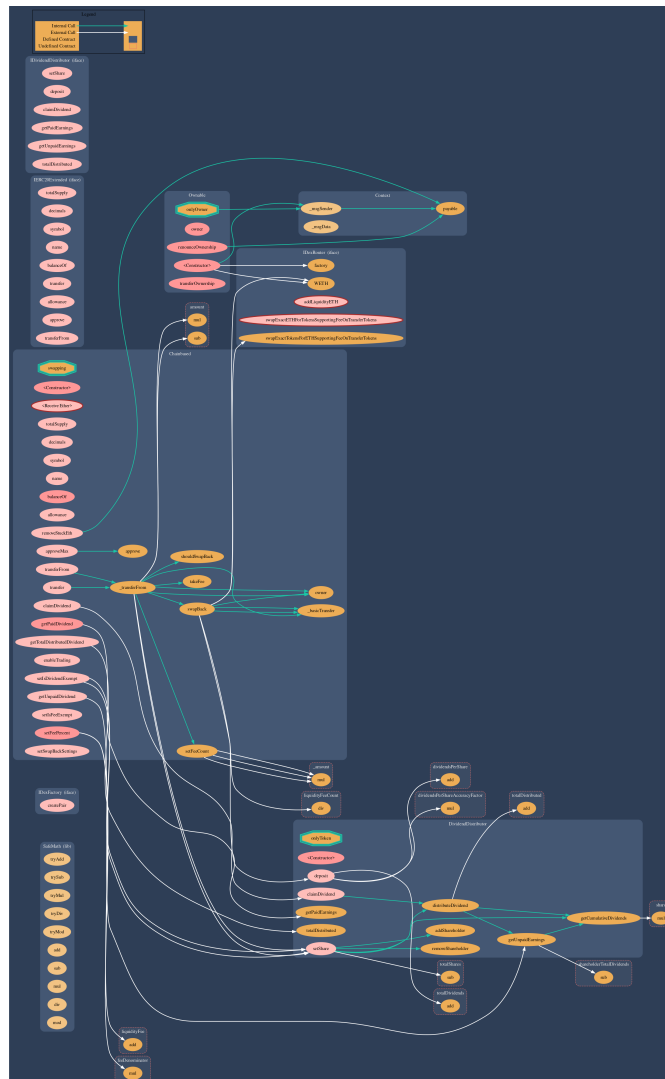
TestNet Contract was Not Assessed

Solidity Code Provided

SolidID	File Sha-1	FileName
BASD	124e1e96b7bbf800c9951f3c5515e96bd74fb01a	BASD.sol
BASD		.sol
BASD		.sol
BASD		.sol
BASD		.sol
BASD		.sol

Call Graph

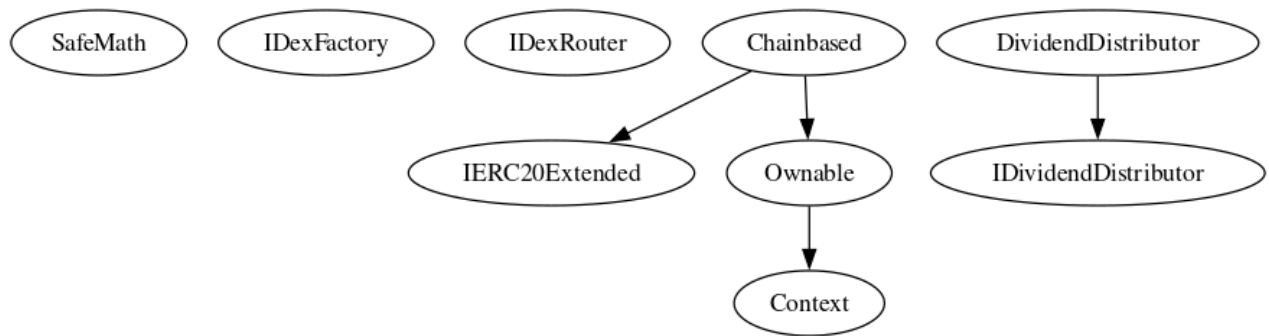
The contract for Chainbased has the following call graph structure.





Inheritance

The contract for Chainbased has the following inheritance structure.

The Project has a Total Supply of 500,000,000



BASD-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	 Low	BASD.sol: L: 0 C: 12	 Detected

Description

The given input is missing the check for the non-zero address.

The given input is missing the check for the onlyOwners need to be revisited for require..



Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
require(receiver != address(0), "Receiver is the zero address");
...
...
require(value X limitation, "Your not able to do this function");
...
```

We also recommend customer to review the following function that is missing a required validation. onlyOwners need to be revisited for require..

BASD-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	 Low	BASD.sol: L: 506 C: 12, L: 593 C: 12	 Detected



Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes. The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

BASD-14 | Unnecessary Use Of SafeMath

Category	Severity	Location	Status
Logical Issue	 Medium	BASD.sol: L: 0 C: 0	 Detected

Description

The SafeMath library is used unnecessarily. With Solidity compiler versions 0.8.0 or newer, arithmetic operations

will automatically revert in case of integer overflow or underflow.

library SafeMath {

An implementation of SafeMath library is found.

using SafeMath for uint256;

SafeMath library is used for uint256 type in contract.



Remediation

We advise removing the usage of SafeMath library and using the built-in arithmetic operations provided by the

Solidity programming language

Project Action

BASD-18 | Stop Transactions by using Enable Trade.

Category	Severity	Location	Status
Logical Issue	 Critical	BASD.sol: L: 393 C: 14	 Detected

Description



Enable Trade is present on the following contract and when combined with Exclude from fees it can be considered a whitelist process, this will allow anyone to trade before others and can represent an issue for the holders.

Remediation

We recommend the project owner to carefully review this function and avoid problems when performing both actions.

Project Action

BASD-19 | Centralization Privileges of.

Category	Severity	Location	Status
	 Medium	BASD.sol: L: 393 C: 14,L: 385 C: 14,L: 341 C: 14,L: 306 C: 14,L: 299 C: 14,L: 269 C: 14	 Detected

Description

Centralized Privileges are found on the following functions.






Remediation

undefined






Project Action

Technical Findings Summary

Classification of Risk

Severity	Description
 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.
 High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
 Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
 Low	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.
 Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
 Critical	1	2	0
 High	0	0	0
 Medium	2	1	0
 Low	2	1	0
 Informational	0	0	0
Total	5	5	0

Social Media Checks

Social Media	URL	Result
Twitter	https://x.com/chainbased_io	Pass
Other	https://t.me/ChainbasedOfficialChat , https://discord.gg/qEjSruqXXR , https://www.linkedin.com/company/chainbasedofficial/?viewAsMember=true	Pass
Website	https://www.chainbased.io	Pass
Telegram	https://t.me/ChainbasedOfficialChannel	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:



Assessment Results

Score Results

Review	Score
Overall Score	78/100
Auditor Score	80/100
Review by Section	Score
Manual Scan Score	26
Auto Scan Score	37
Advance Check Score	15

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

Audit Fail



Assessment Results

Important Notes:

- **Centralization Risks:** The owner has significant control over the contract, including fee settings and trading enablement. Consider implementing a multi-signature wallet for critical functions.
- **Reentrancy:** Functions like `claimDividend` and `swapBack` should be reviewed for reentrancy vulnerabilities. Consider using the Checks-Effects-Interactions pattern.
- **Unchecked External Calls:** Calls to `distributor.deposit` and `distributor.setShare` should include checks to handle failures gracefully.
- **Fee Calculation Complexity:** Ensure that the logic for fee calculations and transfers is thoroughly tested to prevent unexpected behavior.
- **Dividend Distribution:** Verify the accuracy of dividend calculations and ensure proper handling of edge cases, such as zero shares.
- **Gas Optimization:** Review loops and calculations for potential gas savings, especially in functions like `removeShareholder`.
- **Access Control:** Functions like `removeStuckEth` allow the owner to withdraw ETH. Ensure this is intended and consider adding restrictions or time locks.
- **Event Emissions:** Ensure all critical state changes are accompanied by event emissions for transparency and

traceability.

Auditor Score =80
Audit Fail



Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different requirements on the input variables than a setter function.

Coding Best Practices

ERC 20 Coding Standards are a set of rules that each developer should follow to ensure the code meets a set of criteria and is readable by all the developers.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocacy for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided 'as is, and Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access 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 with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.

