

Security Assessment: CNDR TOKEN





June 13, 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	100%
 Sale Tax	100%
 Cannot Buy	Pass
 Cannot Sale	Pass
 Max Tax	100%
 Modify Tax	Yes
 Fee Check	Pass
 Is Honeypot?	Detected
 Trading Cooldown	Not Detected
 Can Pause Trade?	Pass
 Pause Transfer?	Not Detected
 Max Tx?	Fail
 Is Anti Whale?	Detected
 Is Anti Bot?	Not Detected

Contract Privilege	Description
 Is Blacklist?	Not Detected
 Blacklist Check	Pass
 is Whitelist?	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?	Detected
 Other?	Not Detected
 Holders	0
 Auditor Confidence	Critical Risk
 KYC Present	No
 KYC URL	N/A

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	0x
Name	CNDR
Token Tracker	CNDR (\$CNDR)
Decimals	9
Supply	
Platform	BASE
compiler	v0.8.23
Contract Name	Cindr
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed Contract Name

Name	Contract	Live
CINDR	0x	Yes

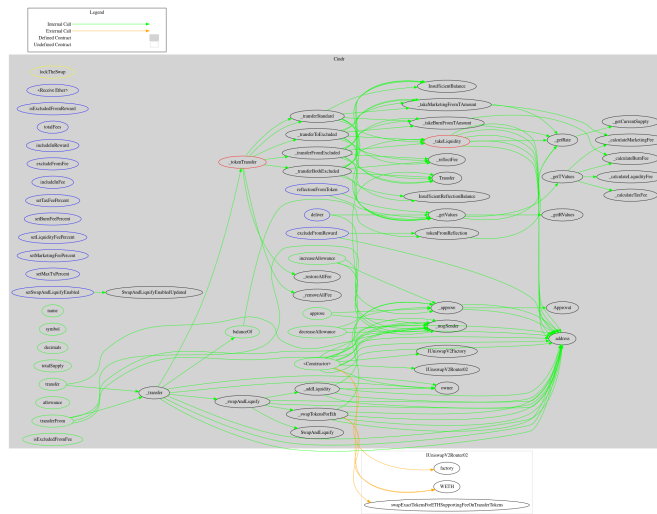
TestNet Contract was Not Assessed

Solidity Code Provided

SolidID	File Sha-1	FileName
CINDR	d59ca56e0889a0698b75d0e45050bd31ba432b0c	Cindr.sol
CINDR		
CINDR		
CINDR		
CINDR		
CINDR		

Call Graph

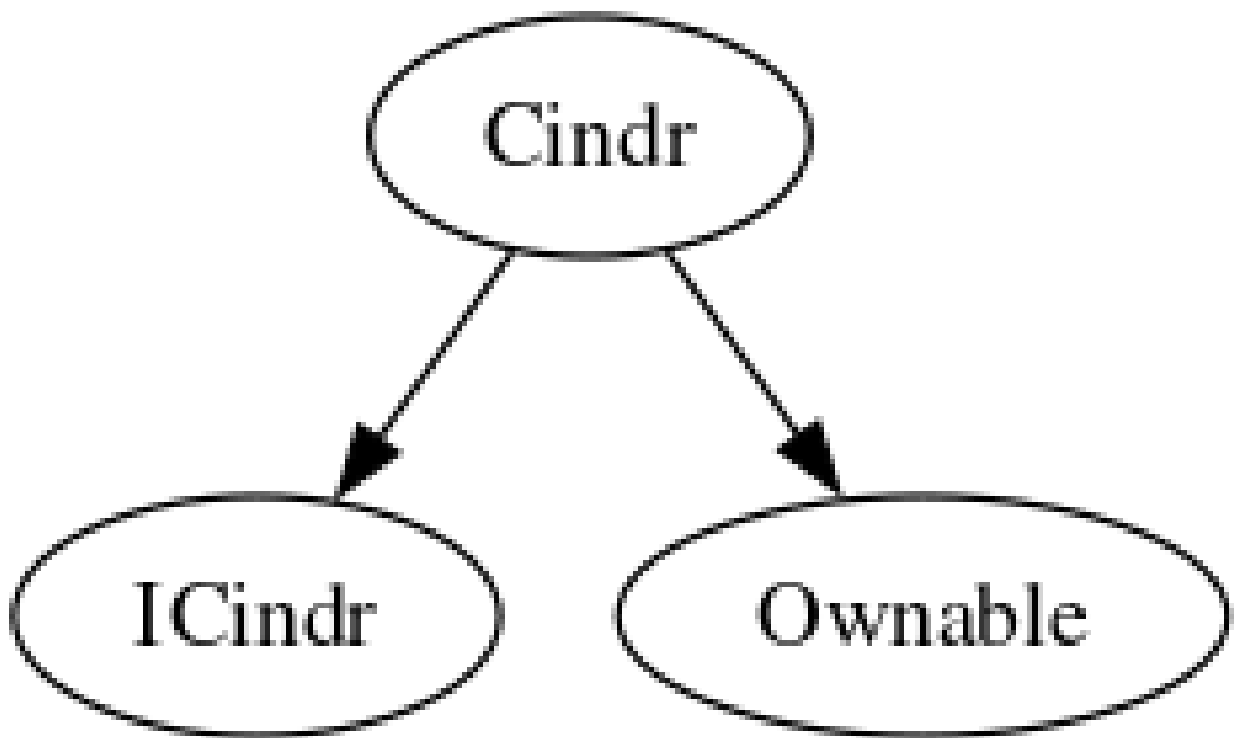
The contract for CNDR has the following call graph structure.



Inheritance

The contract for CNDR has the following inheritance structure.

The Project has a Total Supply of





Privileged Functions (onlyOwner)

Please Note if the contract is Renounced none of this functions can be executed.

Function Name	Parameters	Visibility
excludeFromReward	address account	external
includeInReward	address account	external
excludeFromFee	address account	external
includeInFee	address account	external
setTaxFeePercent	uint16 _taxFee	external
setBurnFeePercent	uint16 _burnFee	external
setLiquidityFeePercent	uint16 _liquidityFee	external
setMarketingFeePercent	uint16 _marketingFee	external
setMaxTxPercent	uint256 maxTxPercent	external
setSwapAndLiquifyEnabled	bool _enabled	external

\$CNDR-01 | Potential Sandwich Attacks.

Category	Severity	Location	Status
Security	 Medium	Cindr.sol: L: 847	 UnResolved

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by back running (after the transaction being attacked) a transaction to sell the asset. The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- swapExactTokensForETHSupportingFeeOnTransferTokens()
- addLiquidityETH()



Remediation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

References:

What Are Sandwich Attacks in DeFi — and How Can You Avoid Them?.

\$CNDR-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	 Low	Cindr.sol: L: 354 C: 14, L: 347 C: 14, L: 338 C: 14, L: 329 C: 14, L: 320 C: 14, L: 311 C: 14, L: 304 C: 14, L: 297 C: 14	 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..

\$CNDR-04 | Centralized Risk In addLiquidity.

Category	Severity	Location	Status
Coding Style	 High	Cindr.sol: L: 907 C:14	 Detected

Description

```
uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this), tokenAmount, 0, 0, owner(), block.timestamp);
```

The addLiquidity function calls the uniswapV2Router.addLiquidityETH function with the to address specified as owner() for acquiring the generated LP tokens from the \$CNDR-WBNB pool.

As a result, over time the _owner address will accumulate a significant portion of LP tokens. If the _owner is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Remediation



We advise the to address of the uniswapV2Router.addLiquidityETH function call to be replaced by the contract itself, i.e. address(this) , and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the _owner account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

1. Indicatively, here are some feasible solutions that would also mitigate the potential risk:
2. Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;
3. Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;

Introduction of a DAO / governance / voting module to increase transparency and user involvement

Project Action

\$CNDR-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	 Low	Cindr.sol: L: 262 C: 14, L: 281 C: 14, L: 297 C: 14, L: 304 C: 14, L: 311 C: 14, L: 320 C: 14, L: 329 C: 14, L: 338 C: 14, L: 347 C: 14, L: 907 C: 14, L: 900 C: 14	 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.

\$CNDR-07 | State Variables could be Declared Constant.

Category	Severity	Location	Status
Coding Style	 Low	Cindr.sol: L: 25, L:95, L:98	 Detected

Description

Constant state variables should be declared constant to save gas.



```
_decimals  
uniswapV2Router  
uniswapV2Pair
```

Remediation

Add the constant attribute to state variables that never changes.

<https://docs.soliditylang.org/en/latest/contracts.html#constant-state-variables>

\$CNDR-10 | Initial Token Distribution.

Category	Severity	Location	Status
Centralization / Privilege	 High	Cindr.sol: L: 201	 UnResolved

Description



All of the CNDR tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

Remediation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Project Action

\$CNDR-11 | Potential Reentrancy in `_swapTokensForEth`.

Category	Severity	Location	Status
Optimization	 High	Cindr.sol: L: 874 C: 14	 UnResolved

Description



The `_swapAndTransfer` function involves external calls which could be exploited for reentrancy.

Remediation

Use reentrancy guards or check-effects-interactions pattern.

Project Action

\$CNDR-13 | Extra Gas Cost For User.

Category	Severity	Location	Status
Logical Issue	 Informational	Cindr.sol: L: 829 C: 14	 Detected

Description



The user may trigger a tax distribution during the transfer process, which will cost a lot of gas and it is unfair to let a single user bear it.

Remediation

We advise the client to make the owner responsible for the gas costs of the tax distribution.

Project Action

\$CNDR-16 | Taxes can be up to 100%.

Category	Severity	Location	Status
Logical Issue	 Critical	Cindr.sol: L: 311 C: 0	 Detected

Description

The current definition of taxes can be set up to 100% for specific wallets, we suggest to modify the function not to be dynamic but to be a static resolution.

```
feelnTokens > senderBalance &&  
(feelnTokens / 100) * 95 <= senderBalance
```

due to the logic written in here may results in loss of funds.



Remediation

We advise the team to review the following logic function

```
function setFee(uint256  
redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell) public  
onlyOwner {  
    _redisFeeOnBuy = redisFeeOnBuy;  
    _redisFeeOnSell = redisFeeOnSell;  
    _taxFeeOnBuy = taxFeeOnBuy;  
    _taxFeeOnSell = taxFeeOnSell;  
}
```

Project Action

\$CNDR-21 | Max Transaction Controls.

Category	Severity	Location	Status
UnResolved	 Medium	Cindr.sol: L: 347, L: 14	 Detected

Description



The contract includes mechanisms to control the maximum transaction amount, which helps prevent large transfers that could affect the token's price stability.

Remediation

Set Reasonable Limits: Ensure that the limits are set to reasonable values that do not hinder normal trading activities. Add Governance Mechanism: Introduce a governance mechanism to allow the community to vote on changes to these limits, reducing centralization risks. Emit Events: Ensure that any changes to these limits emit events for better transparency and tracking. Validation Checks: Add validation checks to ensure the new limits are within acceptable ranges.

Project Action

\$CNDR-22 | Missing recoverETH Function.

Category	Severity	Location	Status
	 High	Cindr.sol: L: 0, L: 0	 Detected

Description

The contract currently lacks a function to recover accidentally sent ETH. This can be problematic if someone mistakenly sends ETH to the contract address or if there are any miscalculations in the contract that result in ETH being stuck.






Remediation

The absence of a recoverETH function can lead to loss of funds if ETH is accidentally sent to the contract. Implementing a recovery mechanism will allow the owner to retrieve such funds, enhancing the contract's robustness and user trust.






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	1	0
 High	4	4	0
 Medium	2	2	0
 Low	3	3	0
 Informational	1	1	0
Total	11	11	0

Social Media Checks

Social Media	URL	Result
Twitter	https://x.com/CINDRonBase	Pass
Other		
Website	https://CINDRonBase.com	Fail
Telegram	https://t.me/CINDR_on_Base	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:



Audit Result

Final Audit Score

Review	Score
Security Score	65
Auditor Score	65

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 85 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:

- Overall Classification:
- Security: Medium
- Centralization: High
- Optimization: Medium
- Transparency: Medium
- Fund Recovery: Medium
- Score: 65/100
- Overall Conclusion:
 - The Cindr token contract has a solid foundation with its reflection mechanism, auto liquidity, and fee structure. However, there are significant concerns, particularly around centralization and potential security risks. The absence of a recoverETH function adds to the medium severity issues, highlighting the need for a recovery mechanism to prevent loss of funds.
 - Addressing the high and medium severity issues will significantly enhance the contract's robustness and trustworthiness. The overall score of 65 reflects a need for improvements in decentralization, security measures, and fund recovery mechanisms. Regular audits, thorough testing, and adherence to best practices are recommended to maintain and improve the contract's integrity.

- Chat about Cindr.sol

Auditor Score =65
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.

