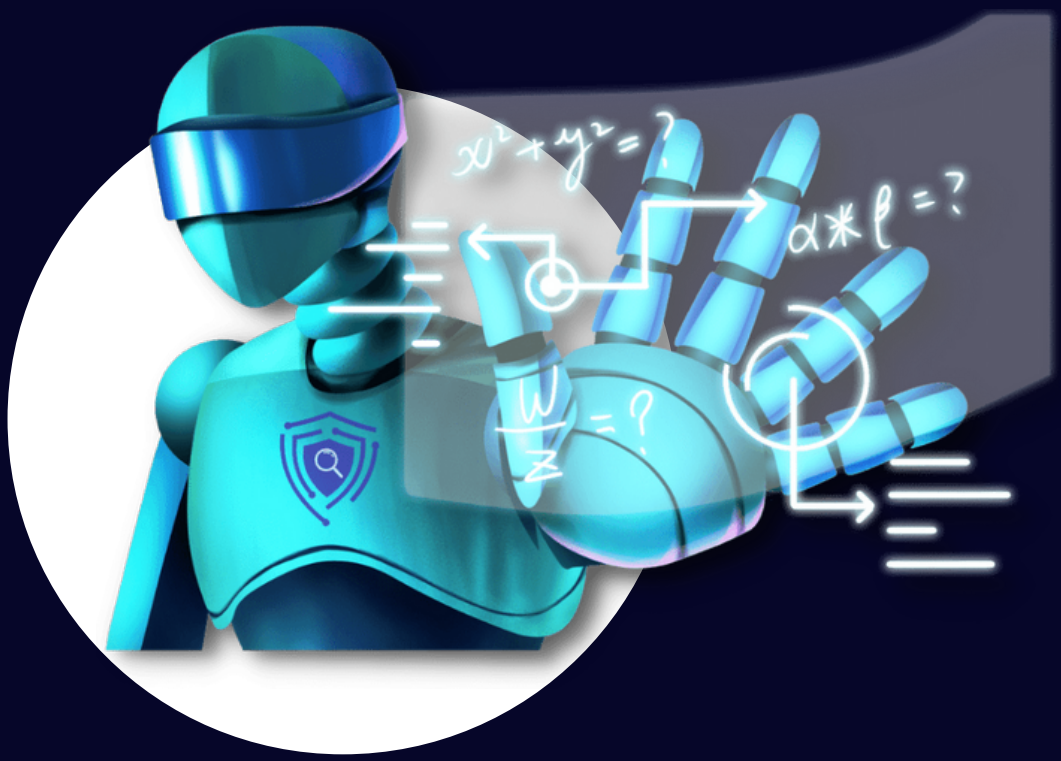
SMART CONTRACT AUDITS AND BLOCKCHAIN SECURITY

SAFUAUDIT

SMART CONTRACT AUDITS AND BLOCKCHAIN SECURITY



PROJECT: **INRD (SWAP)**

DATE: May 20, 2022

www.safuaudit.com

INTRODUCTION

Client	InrdSwap
Language	Solidity
Contract address	0x9D07f13EB4E9b6D95Bf350aE73aEB9ec08b3e27B
Owner	0x543ab47ac1600329d48e762972511d58c8a83756
Deployer	0x543ab47ac1600329d48e762972511d58c8a83756
SHA1-Hash	1aa392842f44adf8117e888394d11f4caccad9b9
Decimals	-
Supply	-
Platform	Binance Smart Chain
Compiler	v0.8.13+commit.abaa5c0e
Optimization	Yes with 200 runs
Website	https://inrdcoin.com/
Telegram	https://t.me/INRD8
Twitter	https://twitter.com/INRDcoin

This audit only provides info on the swap Smart Contract that will be implemented in the application.

TABLE OF CONTENTS

SafuAudit

APPROACH

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

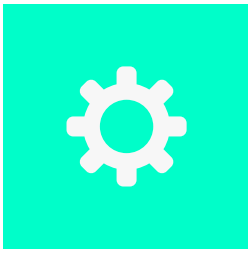
Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability



Tools

- Remix IDE
- Mythril
- Open Zeppelin Code Analyzer
- Solidity Code Complier
- Hardhat

RISK CLASSIFICATION

CRITICAL

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

MEDIUM

Issues on this level could potentially bring problems and should eventually be fixed.

MINOR

Issues on this level are minor details and warning that can remain unfixed but would be better fixed at some point in the future

INFORMATIONAL

Information level is to offer suggestions for improvement of efficacity or security for features with a risk free factor.

CONTRACT INSPECTION 🔎

Imported contracts or frameworks used:

ReentrancyGuard	Implementation			
SafeMath	Library			
Ownable	Implementation			
Token	Interface			

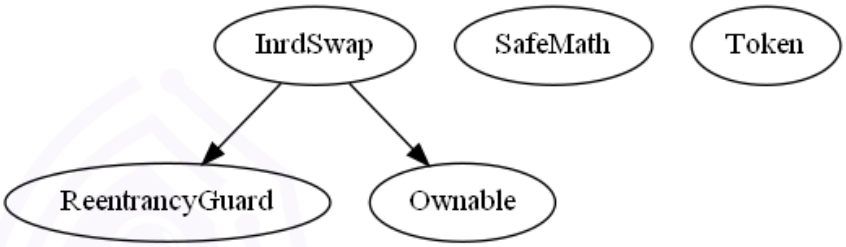
Tested Contract File:

File Name	SHA-1 Hash
InrdSwap.sol	1aa392842f44adf8117e888394d11f4caccad9b9

InrdSwap	Implementation	Ownable, ReentrancyGuard		
└	<Constructor>	Public ❗	🛑	NO❗
└	<Receive Ether>	External ❗	💵	NO❗
└	buyInrd	External ❗	🛑	nonReentrant
└	sellInrd	External ❗	🛑	nonReentrant
└	updateBuyFee	External ❗	🛑	onlyOwner
└	updateSellFee	External ❗	🛑	onlyOwner
└	addStableToken	External ❗	🛑	onlyOwner
└	updateTokenStablePrice	External ❗	🛑	onlyOwner
└	updateTokenAddress	External ❗	🛑	onlyOwner
└	withdrawTokens	External ❗	🛑	nonReentrant onlyOwner
└	withdrawCrypto	External ❗	🛑	nonReentrant onlyOwner
└	tokenBalance	Public ❗		NO❗
└	usdtToToken	External ❗		NO❗
└	tokenToUsdt	External ❗		NO❗
└	usdtToTokenFee	External ❗		NO❗
└	tokenToUsdtFee	External ❗		NO❗
└	bnbBalance	Public ❗		NO❗

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable
🔐	Private function
🔒	Internal function
NO❗	Function has no modifier

INHERITANCE TREE



Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

OWNER PRIVILEGES

Fees

- Buy Fees: 1%
- Sell Fees: 0%

Fees privileges

- Owner can set fees up to 100%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount

- Owner can't set max Tx amount

Pause function

- Owner can't pause trading

Blacklist

- Owner can't blacklist

SafuAudit

MANUAL FUNCTIONS ANALYSIS

The contract is verified to check if functions do and work as they should and malicious code is not inserted.

	Tested	Result
Transfer	Yes	Passed
Total Supply	Yes	N/A
Buy Back	Yes	N/A
Burn	Yes	N/A
Mint	Yes	N/A
Rebase	Yes	N/A
Pause	Yes	N/A
Blacklist	Yes	N/A
Lock	Yes	N/A
Max Transaction	Yes	N/A
Transfer Ownership	Yes	Passed
Renounce Ownership	Yes	N/A

SafuAudit

VULNERABILITIES TEST

ID	Description	
V-01	Function Default Visibility	**Passed**
V-02	Integer Overflow and Underflow	**Passed**
V-03	Outdated Compiler Version	**Passed**
V-04	Floating Pragma	**Minor**
V-05	Unchecked Call Return Value	**Passed**
V-06	Unprotected Ether Withdrawal	**Passed**
V-07	Unprotected SELF-DESTRUCT Instruction	**Passed**
V-08	Re-entrancy	**Passed**
V-09	State Variable Default Visibility	**Passed**
V-10	Uninitialized Storage Pointer	**Passed**
V-11	Assert Violation	**Passed**
V-12	Use of Deprecated Solidity Functions	**Passed**
V-13	Delegate Call to Untrusted Callee	**Passed**
V-14	DoS with Failed Call	**Passed**
V-15	Transaction Order Dependence	**Passed**
V-16	Authorization through tx.origin	**Passed**
V-17	Block values as a proxy for time	**Passed**

V-18	Signature Malleability	**Passed**
V-19	Incorrect Constructor Name	**Passed**
V-20	Shadowing State Variables	**Passed**
V-21	Weak Sources of Randomness from Chain Attributes	**Passed**
V-22	Missing Protection against Signature Replay Attacks	**Passed**
V-23	Lack of Proper Signature Verification	**Passed**
V-24	Requirement Violation	**Medium**
V-25	Write to Arbitrary Storage Location	**Passed**
V-26	Incorrect Inheritance Order	**Passed**
V-27	Insufficient Gas Griefing	**Passed**
V-28	Arbitrary Jump with Function Type Variable	**Passed**
V-29	DoS With Block Gas Limit	**Passed**
V-30	Typographical Error	**Passed**
V-31	Right-To-Left-Override control character (U+202E)	**Passed**
V-32	Presence of unused variables	**Passed**
V-33	Unexpected Ether balance	**Passed**
V-34	Hash Collisions With Multiple Variable Length Arguments	**Passed**
V-35	Message call with the hardcoded gas amount	**Passed**
V-36	Code With No Effects (Irrelevant/Dead Code)	**Passed**
V-37	Unencrypted Private Data On-Chain	**Passed**

FINDINGS

ID	Category	Issue	Severity
V-03	Vulnerabilities	Floating Pragma	Minor
V-24	Vulnerabilities	Requirement Violation	Medium
CE-01	Centralization	Fees up to 100%	Medium
CE-07	Centralization	Withdraw tokens without limit	Medium
CE-08	Centralization	Change swap token price	Medium
CS-01	Coding Standards	Too Many Digits	Informational

SafuAudit

V-01: Floating Pragma

Line #6

```
6    pragma solidity ^0.8.13;
```

Description

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Recommandation

We advise locking at the lowest pragma version that the contract can be compiled at. For example: pragma solidity 0.8.4;

SafuAudit

V-24: Requirement violation

Line #216

```
require(Token(_tokenAddr).transfer(beneficiary, Token(_tokenAddr).balanceOf(address(this))));
```

Description

A requirement was violated in a nested call and the call was reverted as a result.

Recommandation

Make sure valid inputs are provided to the nested call (for instance, via passed arguments).

SafuAudit

CE-01, CE-07, CE-08: Centralization Risks

CE-01: Fees up to 100%
Line #193-196, 198-201

```
function updateBuyFee(uint256 _buyFee, uint256 _buyDivisor) external onlyOwner{
    buyFee = _buyFee;
    buyFeeDivisor = _buyDivisor;
}

function updateSellFee(uint256 _sellFee, uint256 _sellDivisor) external onlyOwner{
    sellFee = _sellFee;
    sellFeeDivisor = _sellDivisor;
}
```

CE-07: Withdraw tokens without a limit
Line #215-217

```
function withdrawTokens(address _tokenAddr, address beneficiary) external nonReentrant onlyOwner {
    require(Token(_tokenAddr).transfer(beneficiary, Token(_tokenAddr).balanceOf(address(this))));
}
```

CE-08: Change swap token price
Line #207-209

```
function updateTokenStablePrice(uint256 newTokenValue) external onlyOwner {
    tokenPriceUsdt = newTokenValue;
}
```

Description

The role OnlyOwner has authority over the above functions that can manipulate the project functionality without restrictions. Any compromise to the owner account may allow a hacker to take advantage of this authority.

Recommandation

- Functions need to have at least one restriction.
  For example: setting fees to a maximum of 25%, setting tokenPriceUsdt with a limit

CS-01: Coding Standards

Line # 125: Too Many Digits

```
uint256 public tokenPriceUsdt = 13157894730000000;
```

Description

Literals with many digits are difficult to read and review.

Recommandation

- Use the scientific notation to improve readability.

SafuAudit

GOOD PRACTICES

- The owner cannot mint new tokens after deployment
- The owner cannot stop or pause the contract
- The owner cannot set a transaction limit
- The smart contract utilizes "SafeMath" to prevent overflows

```
function mul(uint256 a, uint256 b) internal pure returns (uint256) {
    if (a == 0) {
        return 0;
    }
    uint256 c = a * b;
    assert(c / a == b);
    return c;
}

function div(uint256 a, uint256 b) internal pure returns (uint256) {
    // assert(b > 0); // Solidity automatically throws when dividing by 0
    uint256 c = a / b;
    // assert(a == b * c + a % b); // There is no case in which this doesn't hold
    return c;
}

function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    assert(b <= a);
    return a - b;
}

function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    assert(c >= a);
    return c;
}
```

WEBSITE

Website	https://inrdcoin.com/
Domain Registry	http://tucowsdomains.com
Domain Expiry Date	2023-01-12
Response Code	200
SSL Checker and HTTPS Test	Passed
Deprecated HTML tags	Passed
Robots.txt	Passed
Sitemap Test	Informative
SEO Friendly URL	Passed
Responsive Test	Passed
JS Error Test	Passed
Console Errors Test	Passed
Site Loading Speed Test	0.91 seconds - Passed
HTTP2 Test	Passed
Safe Browsing Test	Passed

DISCLAIMER

SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice, or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

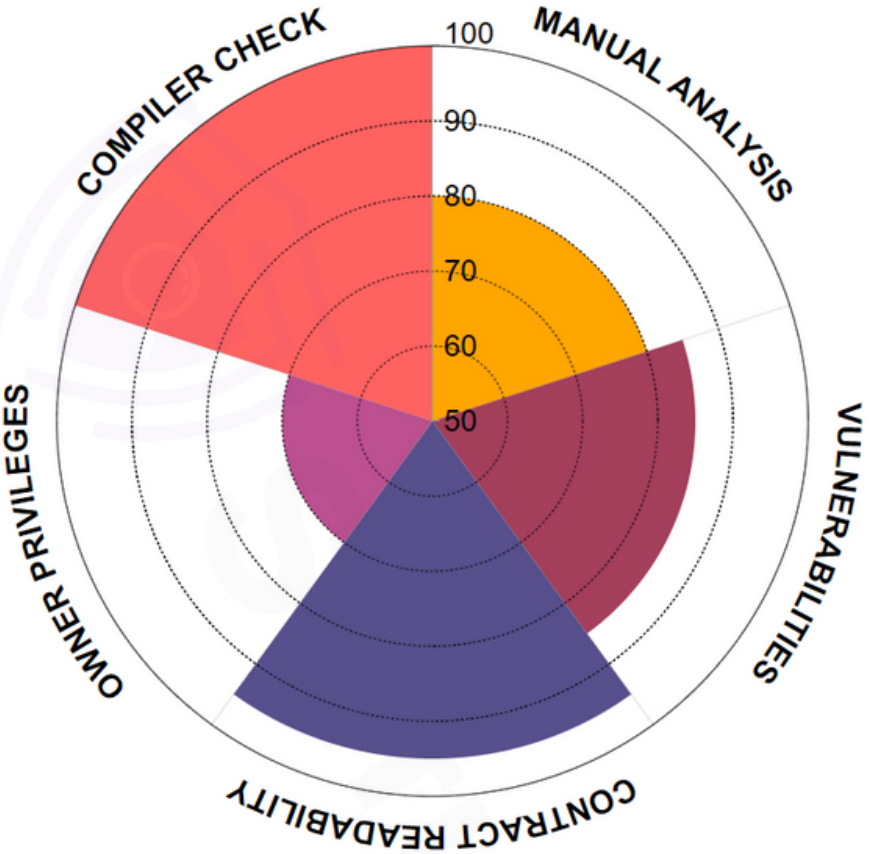
Accuracy of Information

SafuAudit will strive to ensure the accuracy of the information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyze the on-chain smart contract source code and to provide a basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only — we recommend proceeding with several independent audits Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability or a hack. Be aware that active smart contract owner privileges constitute an elevated impact on the smart contract safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

RATING



Manual Analysis

Vulnerabilities

Contract Readability

Owner Privileges

Compiler Check

Final Score: **86**

CONCLUSION

Contract InrdSwap does not contain any severe issues. Centralization risk is medium: owner can set fees up to 100%, withdraw tokens and stablecoins from contract and set the price of the token on swap without a limit.

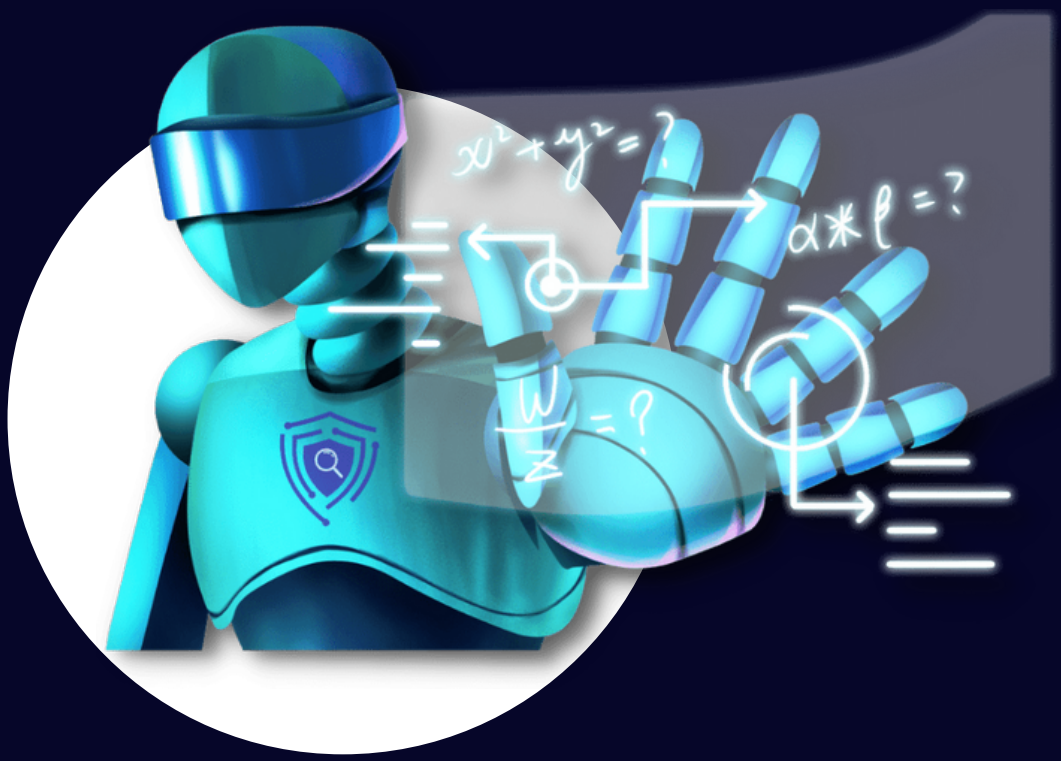
SafuAudit has tested the security based on manual and automated tests. Please note that we don't offer any warranties for business model.

SafuAudit

SMART CONTRACT AUDITS AND BLOCKCHAIN SECURITY

SAFUAUDIT

SMART CONTRACT AUDITS AND BLOCKCHAIN SECURITY



"Only in growth, reform, and change, paradoxically enough, is true security to be found."

www.safuaudit.com