Cyberscope

Audit Report
Araracoin

November 2024

Network	BSC
Address	0xDa05cA5303D75f14e298FB8aEFF51fD2F2105803
Audited by	© cyberscope

Analysis

🔴 Critical 🟡 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔴	ST	Stops Transactions	Multisign
🔵	OTUT	Transfers User's Tokens	Passed
🔵	ELFM	Exceeds Fees Limit	Passed
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	CCR	Contract Centralization Risk	Acknowledged
●	IDI	Immutable Declaration Improvement	Unresolved
●	TUU	Time Units Usage	Unresolved
●	L08	Tautology or Contradiction	Unresolved

Table of Contents

Risk Classification

The criticality of findings in Cyberscope’s smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
● Critical	Highly Likely / High Impact
● Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
● Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	AraraCoin
Compiler Version	v0.8.24+commit.e11b9ed9
Optimazation	200 runs
Explorer	https://bscscan.com/token/0xDa05cA5303D75f14e298FB8aEFF51fD2F2105803
Address	0xDa05cA5303D75f14e298FB8aEFF51fD2F2105803
Network	BSC
Symbol	ARARA
Decimals	18
Total Supply	100,000,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	7 Nov 2024

Source Files

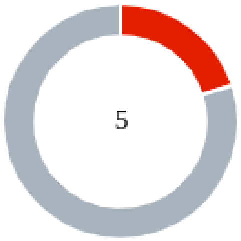
Filename	SHA256
AraraCoin.sol	a6c674e3b305658b65009d12de3c1135cb7e3dfd626963726a8084718

Introduction

AraraCoin Contract

The **AraraCoin** contract is an enhanced ERC20 token with governance, tax management, and controlled trading. Built with OpenZeppelin's `ERC20`, `ERC20Permit`, and `AccessControl` libraries, it provides standard token functionality, permit-based approvals, and role-based access control. During deployment, the total supply (100 billion tokens) is distributed across various wallets for marketing, consulting, audits, and vesting purposes. Tax handling includes a designated `taxWallet` with an adjustable tax percentage (up to 1%) and allows manager-approved exemptions. Before trading is enabled, only authorized addresses can trade, with trading and tax updates requiring multi-manager approvals for added security. This setup ensures decentralized, secure, and adaptable protocol management.

Findings Breakdown



●	Critical	1
●	Medium	0
●	Minor / Informative	4

Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	1
●	Medium	0	0	0	0
●	Minor / Informative	3	1	0	0

ST - Stops Transactions

Criticality	Critical
Location	AraraCoin.sol#L227
Status	Multisign

Description

The transactions are initially disabled for all users excluding the authorized addresses. The owner can enable the transactions for all users. Once the transactions are enable the owner will not be able to disable them again.

```
if (!tradingEnabled) {
    require(_canTrade.contains(from), "AraraCoin: Trade is 
disabled");
}
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Call the `enableTrading` function, which will initialize the trading for all users.

Team Update

The team has acknowledged that this is not a security issue and states:

Multi-sign wallet was added.

CCR - Contract Centralization Risk

Criticality	Minor / Informative
Location	AraraCoin.sol#L184,197
Status	Acknowledged

Description

The contract's functionality and behavior are heavily dependent on external parameters or configurations. While external configuration can offer flexibility, it also poses several centralization risks that warrant attention. Centralization risks arising from the dependence on external configuration include Single Point of Control, Vulnerability to Attacks, Operational Delays, Trust Dependencies, and Decentralization Erosion.

Specifically, the owner has the authority to add addresses that are allowed to trade before trading is enabled, and manage tax exemptions by adding or removing addresses from the set of tax-exempted addresses.

```
   function addExemption(address exemption) public
onlyRole(MANAGER_ROLE) {
        require(_exempted.add(exemption), "AraraCoin: Address already
exists in the exemptions");

        emit TaxExemptionUpdated(exemption, true);
    }

    function removeExemption(address exemption) public
onlyRole(MANAGER_ROLE) {
        require(_exempted.remove(exemption), "AraraCoin: Address not
found in the exemptions");

        emit TaxExemptionUpdated(exemption, false);
    }

    function addCanTrade(
        address[] calldata allowedAddresses
    ) public onlyRole(MANAGER_ROLE) {
        require(!tradingEnabled, "AraraCoin: Trading already enabled");
// Ensure trading isn't enabled yet
        require(allowedAddresses.length != 0, "AraraCoin: List of
allowed addresses cannot be empty."); // Ensure there are addresses to
add

        // Add each address in the provided list to the set of addresses
allowed to trade
        for (uint256 i = 0; i < allowedAddresses.length; i++) {
            _canTrade.add(allowedAddresses[i]);
        }
    }
```

Recommendation

To address this finding and mitigate centralization risks, it is recommended to evaluate the feasibility of migrating critical configurations and functionality into the contract's codebase itself. This approach would reduce external dependencies and enhance the contract's self-sufficiency. It is essential to carefully weigh the trade-offs between external configuration flexibility and the risks associated with centralization.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	AraraCoin.sol#L83
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
managerWallet1
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

TUU - Time Units Usage

Criticality	Minor / Informative
Location	AraraCoin.sol#L76
Status	Unresolved

Description

The contract is using arbitrary numbers to form time-related values. As a result, it decreases the readability of the codebase and prevents the compiler to optimize the source code.

```
uint256 private constant _approvalExpirationTime = 3600;
```

Recommendation

It is a good practice to use the time units reserved keywords like `seconds` , `minutes` , `hours` , `days` and `weeks` to process time-related calculations.

It's important to note that these time units are simply a shorthand notation for representing time in seconds, and do not have any effect on the actual passage of time or the execution of the contract. The time units are simply a convenience for expressing time in a more human-readable form.

L08 - Tautology or Contradiction

Criticality	Minor / Informative
Location	AraraCoin.sol#L154
Status	Unresolved

Description

A tautology is a logical statement that is always true, regardless of the values of its variables. A contradiction is a logical statement that is always false, regardless of the values of its variables.

Using tautologies or contradictions can lead to unintended behavior and can make the code harder to understand and maintain. It is generally considered good practice to avoid tautologies and contradictions in the code.

```
require(newTaxPercentage >= 0 && newTaxPercentage <= 100, "AraraCoin:
Tax percentage must be between 0 and 100 basis points (max 1%).")
```

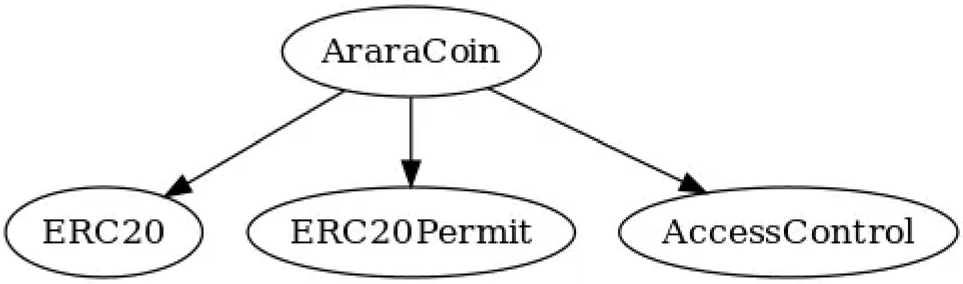
Recommendation

The team is advised to carefully consider the logical conditions is using in the code and ensure that it is well-defined and make sense in the context of the smart contract.

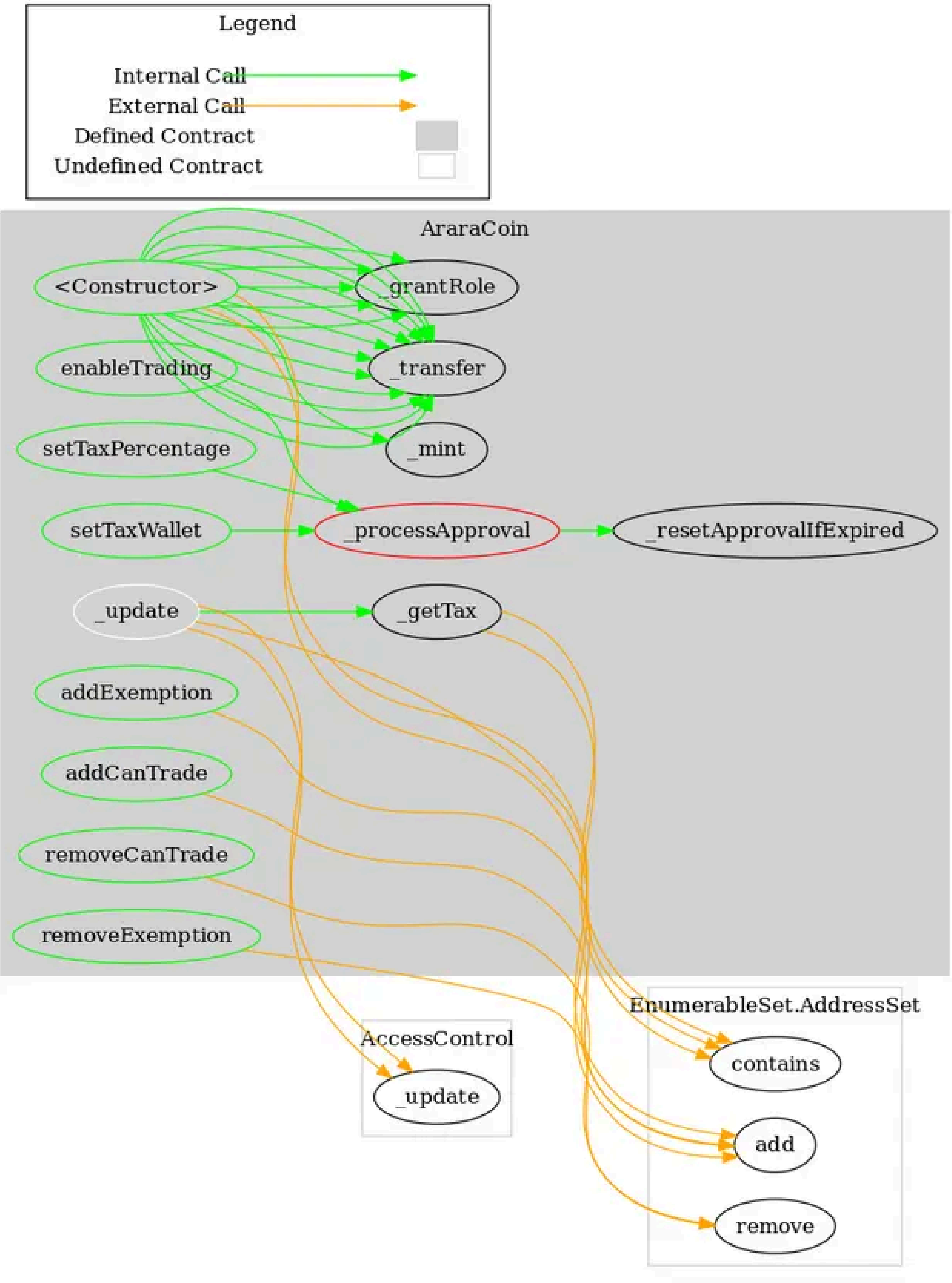
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AraraCoin	Implementation	ERC20, ERC20Permit, AccessControl		
		Public	✓	ERC20 ERC20Permit
	_resetApprovalIfExpired	Private	✓	
	_processApproval	Private	✓	
	enableTrading	Public	✓	onlyRole
	setTaxPercentage	Public	✓	onlyRole
	setTaxWallet	Public	✓	onlyRole
	addExemption	Public	✓	onlyRole
	removeExemption	Public	✓	onlyRole
	addCanTrade	Public	✓	onlyRole
	removeCanTrade	Public	✓	onlyRole
	_update	Internal	✓	
	_getTax	Private		

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
AraraCoin
<Constructor>
enableTrading
setTaxPercentage
setTaxWallet
_update
addExemption
addCanTrade
removeCanTrade
removeExemption
_grantRole
_transfer
_mint
_processApproval
_resetApprovalIfExpired
_getTax
AccessControl
_update
EnumerableSet.AddressSet
contains
add
remove

Summary

Araracoin contracts implement a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. There are some functions that can be abused by the owner like stop transactions. The multi-wallet signing pattern that the contract uses, will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of a maximum 1% fee.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. 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 Cyberscope's prior written consent. 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 regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document 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 represents an extensive assessment 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 Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope'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 Cyberscope are subject to dependencies and are 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.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

cyberscope.io