Cyberscope

Audit Report
aiakita

March 2024

Network	Arbitrum
Address	0x38c2fbdf53b451ae5c4027711d6fe5e1b2191b1c
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	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
●	BLC	Business Logic Concern	Unresolved
●	IDI	Immutable Declaration Improvement	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Review

Contract Name	AiAkita
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Explorer	https://arbiscan.io/address/0x38c2fbdf53b451ae5c4027711d6fe5e1b2191b1c
Address	0x38c2fbdf53b451ae5c4027711d6fe5e1b2191b1c
Network	ARBITRUM
Symbol	AiA
Decimals	6
Total Supply	296,022,887,663,397,540
Badge Eligibility	Yes

Audit Updates

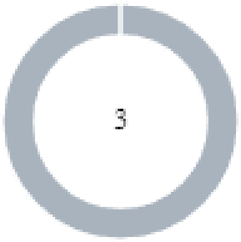
Initial Audit	07 Mar 2024

Source Files

Filename	SHA256
Ownable.sol	b0823419c4379d449c570a4a5d02382de15b17e1fc404ed9910d475adbab9024
IERC20Metadata.sol	0ff2648bc97bfaee7f608683a83e293739acb7b532a92db172790e91c8282f7b

IERC20.sol	edf1c09bf001f6f2982e60130c13413c3beea1ea3ce1365d5e49ec312ecb4d88
ERC20.sol	af482160e7ebb7c66da16714502d076c31f010f2eef70f6dedc46dfe9867fc72
Context.sol	6de5302543723d32c8eaf17becc4525936e16d9c4551455c93d306b9b72c0799
AIA.sol	e5674f46595f197d2c195ea2626e56657a6944e027d2d839ee0447185f41bb89

Findings Breakdown



●	Critical	0
●	Medium	0
●	Minor / Informative	3

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

BLC - Business Logic Concern

Criticality	Minor / Informative
Location	AIA.sol#L27
Status	Unresolved

Description

The contract is designed to impose a transaction fee on transfers, which is deducted and partly burned with a portion sent to the team wallet. This mechanism is enforced in the `transfer` function override, which includes the logic for calculating and applying transaction fees. However, the ERC20 standard also includes a `transferFrom` function, which allows tokens to be transferred from a third party's allowance. In its current implementation, the contract does not override the `transferFrom` function to apply the same fee logic. Consequently, users could potentially bypass the intended transaction fee and burn mechanism by using `transferFrom` instead of `transfer` .

```
function transfer(address recipient, uint256 amount) public
override returns (bool) {
    uint256 transactionFee = amount * TRANSACTION_FEE_PERCENT /
100;
    uint256 burnAmount = transactionFee * BURN_PERCENT / 100;
    uint256 teamWalletAmount = transactionFee - burnAmount;

    if (msg.sender != teamWalletAdr) {
        _burn(msg.sender, burnAmount);
        _transfer(msg.sender, teamWalletAdr, teamWalletAmount);
        _transfer(msg.sender, recipient, amount -
transactionFee);
    }else{
        _transfer(msg.sender, recipient, amount);
    }

    return true;
}
```

Recommendation

It is recommended for the team to re-evaluate the business logic around the transaction fee mechanism. If it should be applied only in the `transfer` function or it is required in the `transferFrom` function as well.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	AIA.sol#L23
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.

```
teamWalletAdr = _teamWalletAdr;
```

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.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	AIA.sol#L10
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.8.0;
```

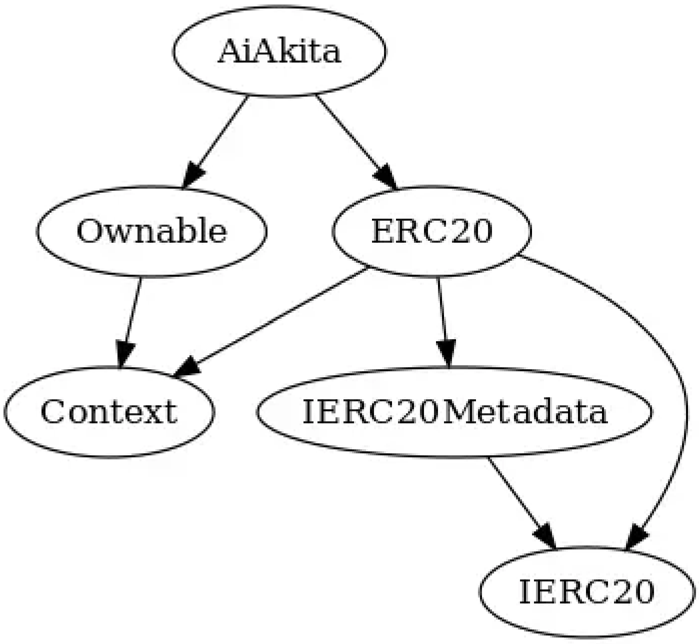
Recommendation

The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

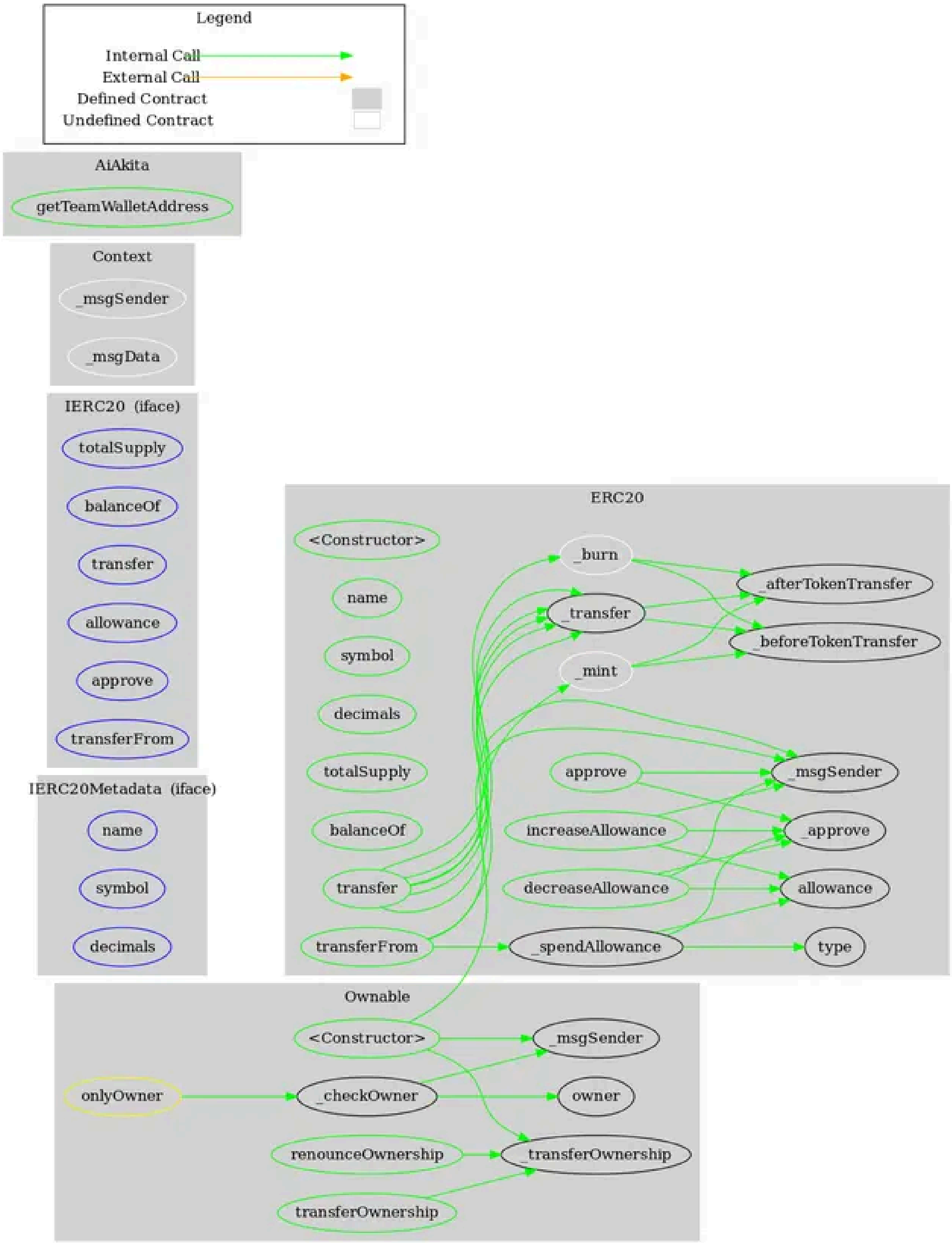
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AiAkita	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	transfer	Public	✓	-
	getTeamWalletAddress	Public		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
AiAkita
getTeamWalletAddress
Context
_msgSender
_msgData
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
IERC20Metadata (iface)
name
symbol
decimals
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
transferFrom
_burn
_transfer
_mint
_afterTokenTransfer
_beforeTokenTransfer
approve
increaseAllowance
decreaseAllowance
_spendAllowance
_msgSender
_approve
allowance
type
Ownable
<Constructor>
onlyOwner
_checkOwner
renounceOwnership
transferOwnership
_msgSender
owner
_transferOwnership

Summary

aiakita contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. aiakita is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract ownership has been renounced.

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

https://www.cyberscope.io