Cyberscope

Audit Report

AIO Ecosystem

June 2023

Network	BSC
Address	0xe5fA0495966B124DD55B390794683bd5CffF4EFA
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	Renounced
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	RZAC	Redundant Zero Address Check	Unresolved
	MDG	Misleading Digit Grouping	Unresolved
	MCM	Misleading Comment Messages	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L05	Unused State Variable	Unresolved
	L06	Missing Events Access Control	Unresolved
	L16	Validate Variable Setters	Unresolved

Table of Contents

Analysis 1
Diagnostics 2
Table of Contents 3
Review 4
Audit Updates 4
Source Files 4
Findings Breakdown 5
BT - Burns Tokens 6
Description 6
Recommendation 6
RZAC - Redundant Zero Address Check 7
Description 7
Recommendation 7
MDG - Misleading Digit Grouping 8
Description 8
Recommendation 8
MCM - Misleading Comment Messages 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L05 - Unused State Variable 11
Description 11
Recommendation 11
L06 - Missing Events Access Control 12
Description 12
Recommendation 12
L16 - Validate Variable Setters 13
Description 13
Recommendation 13
Functions Analysis 14
Inheritance Graph 17
Flow Graph 18
Summary 19
Team Update 19
Disclaimer 20
About Cyberscope 21

Review

Contract Name	AIOToken
Compiler Version	v0.8.18+commit.87f61d96
Optimization	999999 runs
Explorer	https://bscscan.com/address/0xe5fa0495966b124dd55b390794683bd5cfff4efa
Address	0xe5fa0495966b124dd55b390794683bd5cfff4efa
Network	BSC
Symbol	AIO
Decimals	9
Total Supply	100,000,000

Audit Updates

Initial Audit	12 Jun 2023

Source Files

Filename	SHA256
AIOToken.sol	6d8ef3fc594b4b1afa64b1a5dc55419c3555b86e7397f436188f5c3d46a816f0

Findings Breakdown

Critical	1
Medium	0
Minor / Informative	7

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	1
Medium	0	0	0	0
Minor / Informative	7	0	0	0

BT - Burns Tokens

Criticality	Critical
Location	AIOToken.sol#L229
Status	Renounced

Description

The `onlyTokenBurner` role has the authority to burn tokens from a specific address. The user role may take advantage of it by calling the `burn` function. As a result, the targeted address will lose the corresponding tokens.

```
function burn(
        address account,
        uint256 amount
    ) external onlyTokenBurner returns (bool) {
        _burn(account, amount);
        return true;
    }
```

Recommendation

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

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

RZAC - Redundant Zero Address Check

Criticality	Minor / Informative
Location	AIOToken.sol#L177
Status	Unresolved

Description

The contract contains a modifier called "onlyTokenBurner" that includes a redundant check to verify if the sender is not the zero address. The check is unnecessary because it is impossible for the caller to be the zero address.

```
require(
    _msgSender() != address(0),
    "Access denied: only for 'token burners'"
    );
```

Recommendation

The team is advised to remove the redundant check for the zero address in the "onlyTokenBurner" modifier. By eliminating this unnecessary condition, the code becomes more concise, increases gas efficiency and maintains a clear focus on the essential authorization verification step.

MDG - Misleading Digit Grouping

Criticality	Minor / Informative
Location	AIOToken.sol#L162
Status	Unresolved

Description

The code attempts to assign the value of totalSupply to a constant, representing a total supply of tokens. However the assignment of the variable totalSupply contains a misleading usage of the '_' symbol when it groups the digits by four at the end of the totalSupply number.

```
uint256 public totalSupply = 10_000_0000 * 10 ** decimals;
```

Recommendation

The team is recommended to use the '_' symbol to separate digits in sets of three. This practice enhances code comprehension and facilitates better understanding, particularly when working with large numbers.

MCM - Misleading Comment Messages

Criticality	Minor / Informative
Location	AIOToken.sol#L166
Status	Unresolved

Description

The contract is using misleading comment messages. These comment messages do not accurately reflect the actual implementation, making it difficult to understand the source code. As a result, the users will not comprehend the source code's actual implementation.

The comment suggests that the `tokenBurner` address is assigned when the contract is deployed in the constructor. However the `tokenBurner` address is not actually assigned within the constructor.

```
/**
 * @dev An address that has the ability to burn tokens
 * @notice Assigned when the contract is deployed in the constructor
 */
address public tokenBurner;
```

Recommendation

The team is advised to carefully review the comment in order to reflect the actual implementation. To improve code readability, the team should use more specific and descriptive comment messages.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	AIOToken.sol#L201
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
address _tokenBurner
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L05 - Unused State Variable

Criticality	Minor / Informative
Location	AIOToken.sol#L157
Status	Unresolved

Description

An unused state variable is a state variable that is declared in the contract, but is never used in any of the contract's functions. This can happen if the state variable was originally intended to be used, but was later removed or never used.

Unused state variables can create clutter in the contract and make it more difficult to understand and maintain. They can also increase the size of the contract and the cost of deploying and interacting with it.

```
mapping(address => bool) private _isExcludedFromFee
```

Recommendation

To avoid creating unused state variables, it's important to carefully consider the state variables that are needed for the contract's functionality, and to remove any that are no longer needed. This can help improve the clarity and efficiency of the contract.

L06 - Missing Events Access Control

Criticality	Minor / Informative
Location	AIOToken.sol#L202
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
tokenBurner = _tokenBurner
```

Recommendation

To avoid this issue, it's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	AIOToken.sol#L202
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
tokenBurner = _tokenBurner
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behaviour or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

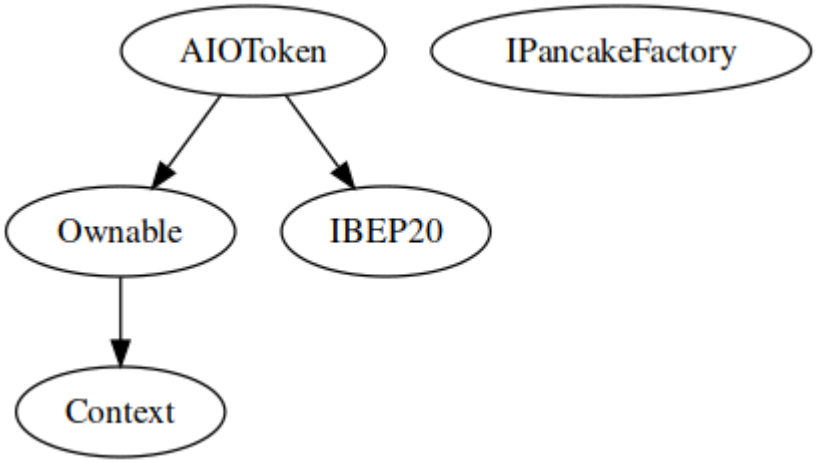
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-

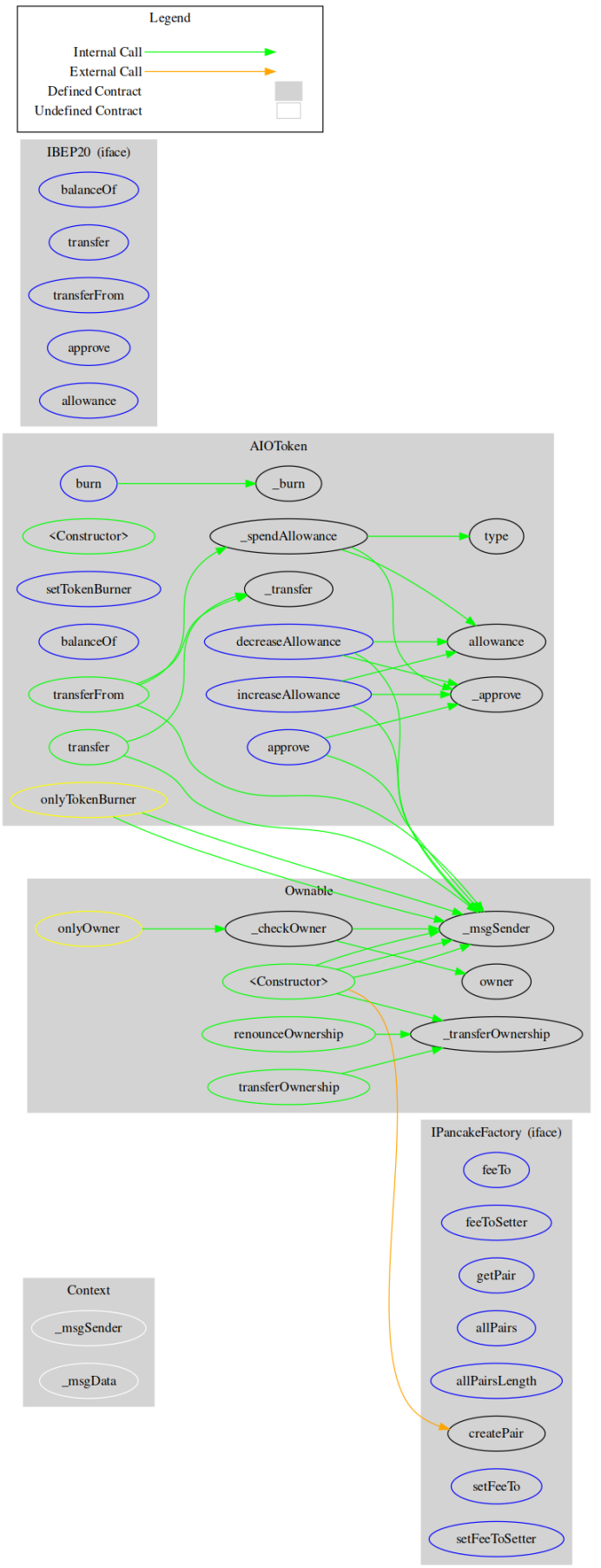
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IBEP20	Interface			
	balanceOf	External		-
	transfer	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	allowance	External		-
AIOToken	Implementation	IBEP20, Ownable		
		Public	✓	Ownable
	setTokenBurner	External	✓	onlyOwner
	balanceOf	External		-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	burn	External	✓	onlyTokenBurner
	approve	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	allowance	Public		-
	_transfer	Private	✓	

	_approve	Private	✓	
	_burn	Private	✓	
	_spendAllowance	Private	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IBEP20 (iface)
balanceOf
transfer
transferFrom
approve
allowance
AIOToken
burn
_burn
<Constructor>
_spendAllowance
type
setTokenBurner
_transfer
balanceOf
decreaseAllowance
allowance
transferFrom
increaseAllowance
_approve
transfer
approve
onlyTokenBurner
Ownable
onlyOwner
_checkOwner
_msgSender
<Constructor>
owner
renounceOwnership
_transferOwnership
transferOwnership
IPancakeFactory (iface)
feeTo
feeToSetter
getPair
allPairs
allPairsLength
createPair
setFeeTo
setFeeToSetter
Context
_msgSender
_msgData

Summary

AIO Ecosystem contract implements 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 burn tokens from any address. if the contract owner abuses the burn functionality, then the users could lost their tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Team Update

The contract's ownership has been renounced, and the information regarding the transaction can be accessed through this link: https://bscscan.com/tx/0xb9d1dfafa9ee98b428f388062bb58ab6d93475accfce5794aac0fcecac1e9c0f.

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