



Cyberscope

Audit Report

Bananace 2

June 2023

Network ETH

Address 0xf5d1D84BA2754d2775358eBB304107cd8D20Fc43

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

Table of Contents

Analysis	1
Table of Contents	2
Review	3
Audit Updates	3
Source Files	3
Functions Analysis	5
Inheritance Graph	7
Flow Graph	8
Summary	9
Disclaimer	10
About Cyberscope	11

Review

Contract Name	BananaceV2
Compiler Version	v0.8.16+commit.07a7930e
Optimization	200 runs
Explorer	https://etherscan.io/address/0xf5d1d84ba2754d2775358ebb304107cd8d20fc43
Address	0xf5d1d84ba2754d2775358ebb304107cd8d20fc43
Network	ETH
Symbol	NANA
Decimals	18
Total Supply	696,969,696,969,696

Audit Updates

Initial Audit	12 May 2023
Corrected Phase 2	14 May 2023
Corrected Phase 3	17 May 2023
Corrected Phase 4	06 Jun 2023

Source Files

Filename	SHA256
BananaceV2.sol	5a43407c3c9f238e6d9f8c1a151db70ef511ab9cd41e7902457c5d1fcbb7b6a4

Findings Breakdown

0

● Critical	0
● Medium	0
● Minor / Informative	0

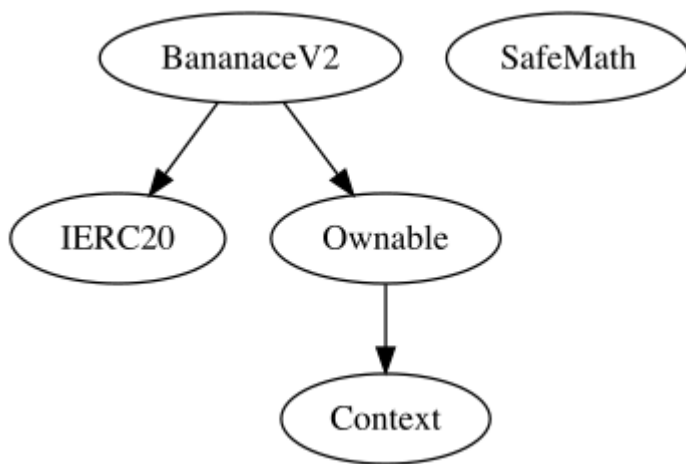
Severity		Unresolved	Acknowledged	Resolved	Other
● Critical	Critical	0	0	0	0
● Medium	Medium	0	0	0	0
● Minor / Informative	Minor / Informative	0	0	0	0

Functions Analysis

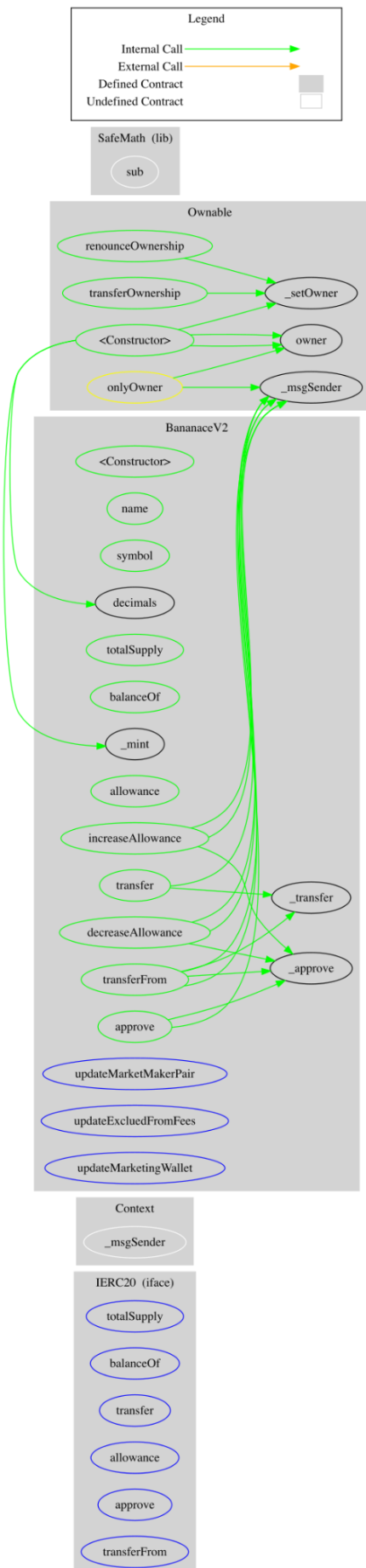
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
SafeMath	Library			

	sub	Internal		
BananaceV2	Implementation	IERC20, Ownable		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_approve	Internal	✓	
	updateMarketMakerPair	External	✓	onlyOwner
	updateExcludedFromFees	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner

Inheritance Graph



Flow Graph



Summary

Bananace contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Bananace is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The Contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The fees are fixed to 2% only for sales.

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>



Cyberscope

Audit Report

Bananace

May 2023

Network BSC

Address 0x6C8C79c1e310C879234E4Fd0e943A19e0524265f

Audited by © cyberscope

Table of Contents

Table of Contents	1
Review	2
Audit Updates	2
Source Files	2
Findings Breakdown	3
Analysis	4
MT - Mints Tokens	5
Description	5
Recommendation	5
Diagnostics	6
IDI - Immutable Declaration Improvement	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L18 - Multiple Pragma Directives	10
Description	10
Recommendation	10
L19 - Stable Compiler Version	11
Description	11
Recommendation	11
Functions Analysis	12
Inheritance Graph	15
Flow Graph	16
Summary	17
Disclaimer	18
About Cyberscope	19

Review

Contract Name	ERC20TokenOBPS
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://bscscan.com/address/0x6c8c79c1e310c879234e4fd0e943a19e0524265f
Address	0x6c8c79c1e310c879234e4fd0e943a19e0524265f
Network	BSC
Symbol	NANA
Decimals	8
Total Supply	696,969,696,969,696

Audit Updates

Initial Audit	12 May 2023
---------------	-------------

Source Files

Filename	SHA256
ERC20TokenOBPS.sol	ae453a5d0a32977c0e460acb7467b168d3ad249674821cef0c15153df2a69329

Findings Breakdown



Critical	1
Medium	0
Minor / Informative	5

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	1	0	0	0
Medium	0	0	0	0
Minor / Informative	5	0	0	0

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

MT - Mints Tokens

Criticality	Critical
Location	ERC20TokenOBPS.sol#L700
Status	Unresolved

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function. As a result, the contract tokens will be highly inflated.

```
function mint(address account, uint256 amount) public onlyOwner {  
    _mint(account, amount);  
}
```

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. 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.

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	IDI	Immutable Declaration Improvement	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L18	Multiple Pragma Directives	Unresolved
●	L19	Stable Compiler Version	Unresolved

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	ERC20TokenOBPS.sol#L678
Status	Unresolved

Description

The contract is using variables that initialize them only in the constructor. The other functions are not mutating the variables. These variables are not defined as `immutable`.

```
pinkAntiBot
```

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.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	ERC20TokenOBPS.sol#L674
Status	Unresolved

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint8 private _decimals = 8
```

Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	ERC20TokenOBPS.sol#L687
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.

```
bool _enable
```

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>.

L18 - Multiple Pragma Directives

Criticality	Minor / Informative
Location	ERC20TokenOBPS.sol#L7,33,116,200,228,617,654,669
Status	Unresolved

Description

If the contract includes multiple conflicting pragma directives, it may produce unexpected errors. To avoid this, it's important to include the correct pragma directive at the top of the contract and to ensure that it is the only pragma directive included in the contract.

```
pragma solidity ^0.8.0;  
pragma solidity >=0.5.0;  
pragma solidity ^0.8.17;
```

Recommendation

It is important to include only one pragma directive at the top of the contract and to ensure that it accurately reflects the version of Solidity that the contract is written in.

By including all required compiler options and flags in a single pragma directive, the potential conflicts could be avoided and ensure that the contract can be compiled correctly.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	ERC20TokenOBPS.sol#L7,33,116,200,228,617,669
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;  
pragma solidity ^0.8.17;
```

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.

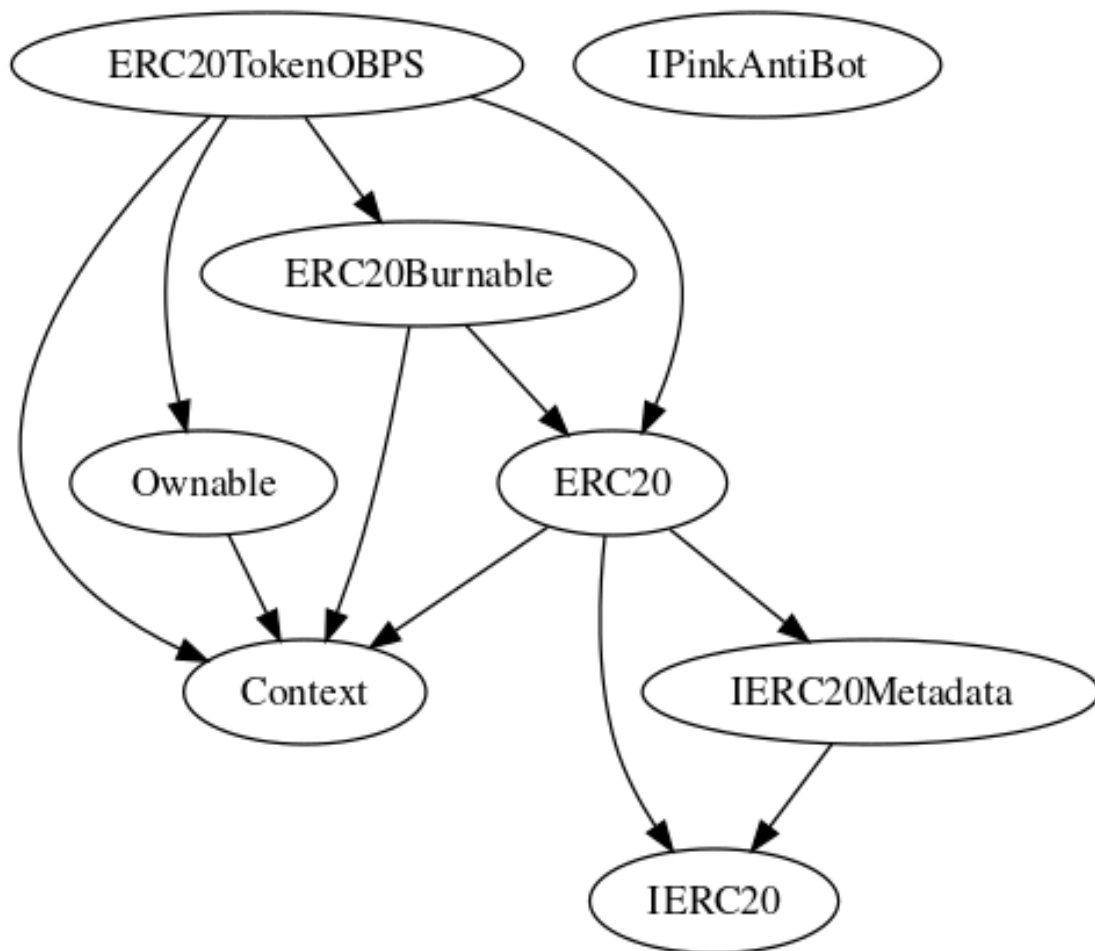
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-

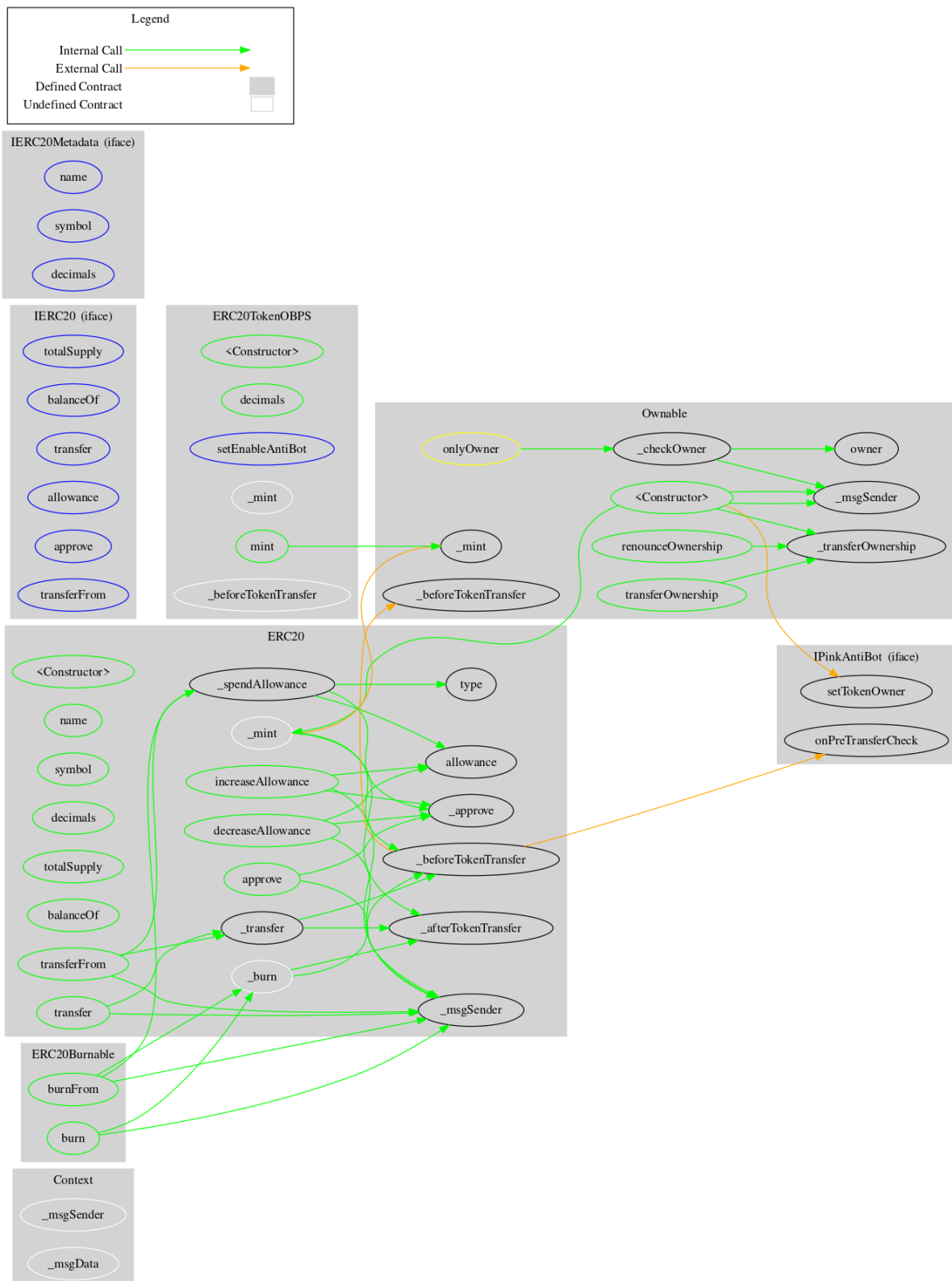
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Meta data		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	

	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
IPinkAntiBot	Interface			
	setTokenOwner	External	✓	-
	onPreTransferCheck	External	✓	-
ERC20TokenOB PS	Implementation	Context, ERC20, ERC20Burnable, Ownable		
		Public	✓	ERC20
	decimals	Public		-
	setEnableAntiBot	External	✓	onlyOwner
	_mint	Internal	✓	
	mint	Public	✓	onlyOwner
	_beforeTokenTransfer	Internal	✓	

Inheritance Graph



Flow Graph



Summary

Bananace 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 mint tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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>



Cyberscope

Audit Report

Bananace

May 2023

Network BSC

Address 0x6C8C79c1e310C879234E4Fd0e943A19e0524265f

Audited by © cyberscope

Table of Contents

Table of Contents	1
Review	2
Audit Updates	2
Source Files	3
Findings Breakdown	4
Analysis	5
ST - Stops Transactions	6
Description	6
Recommendation	6
Diagnostics	7
RDE - Redundant Decimals Extension	8
Description	8
Recommendation	8
RMO - Redundant Mint Override	9
Description	9
Recommendation	9
L19 - Stable Compiler Version	10
Description	10
Recommendation	10
Functions Analysis	11
Inheritance Graph	12
Flow Graph	13
Summary	14
Disclaimer	15
About Cyberscope	16

Review

Contract Name	ERC20TokenOB
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://bscscan.com/address/0x21d714527f1e4f62abb7f68bcd7c94e94d8121f9
Address	0x21d714527f1e4f62abb7f68bcd7c94e94d8121f9
Network	BSC
Symbol	NANA
Decimals	18
Total Supply	696,969,696,969,696

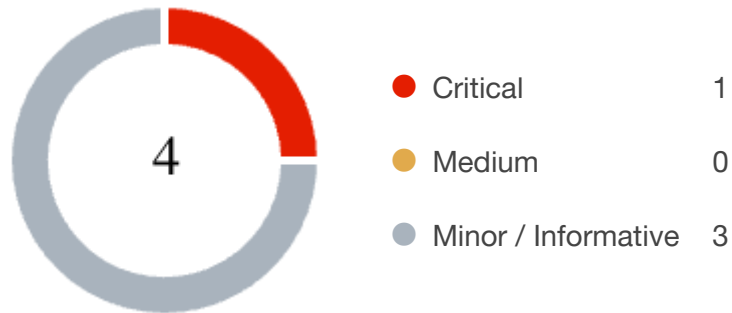
Audit Updates

Initial Audit	12 May 2023
Corrected Phase 2	14 May 2023

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249a a4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	bce14c3fd3b1a668529e375f6b70ffdf9cef 8c4e410ae99608be5964d98fa701
@openzeppelin/contracts/token/ERC20/extensions /ERC20Burnable.sol	0344809a1044e11ece2401b4f7288f414ea 41fa9d1dad24143c84b737c9fc02e
@openzeppelin/contracts/token/ERC20/extensions /IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166 689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db800 3d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a2 3a4baa0b5bd9add9fb6d6a1549814a
contracts/interfaces/IBananaAntiBot.sol	be576cc14d95e13dda2706091de23ed377 33ff6ab7086c0661635afcd01898bb
contracts/tokens/ERC20TokenOB.sol	1c7765984d00cd68ebf57bd75ebb285c8a 6719997a13822d5afd56e1b19fbe8e

Findings Breakdown



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

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

ST - Stops Transactions

Criticality	Critical
Location	contracts/tokens/ERC20TokenOB.sol#L53
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

```
if(from != owner() && to != owner()) {  
    require(amount <= _maxTxAmount, "Transfer amount exceeds the  
maxTxAmount.");  
}
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	RDE	Redundant Decimals Extension	Unresolved
●	RMO	Redundant Mint Override	Unresolved
●	L19	Stable Compiler Version	Unresolved

RDE - Redundant Decimals Extension

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L29
Status	Unresolved

Description

The contract implements the Openzeplin ERC20 standard. According to the ERC20 standard, the decimals are 18. The contract overrides the decimals method. This method returns the same number as the ERC20 standard. As a result, the override is redundant.

```
function decimals() public view virtual override returns (uint8) {  
    return DECIMALS;  
}
```

Recommendation

The team is advised to remove the decimals override since it will produce the same result.

RMO - Redundant Mint Override

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L40
Status	Unresolved

Description

The contract executes the mint method once in the constructor. The `mint()` method is overridden by the contract to allow only non-zero mints. Since the mint method is called once in the constructor, then the override of the mint method is redundant.

```
function _mint(address account, uint256 amount) internal virtual
override {
    if (amount > 0) {
        super._mint(account, amount);
    }
}
```

Recommendation

The team is advised to remove the override of the `mint()` method since it will produce the same result.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L3
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 ^0.8.17;
```

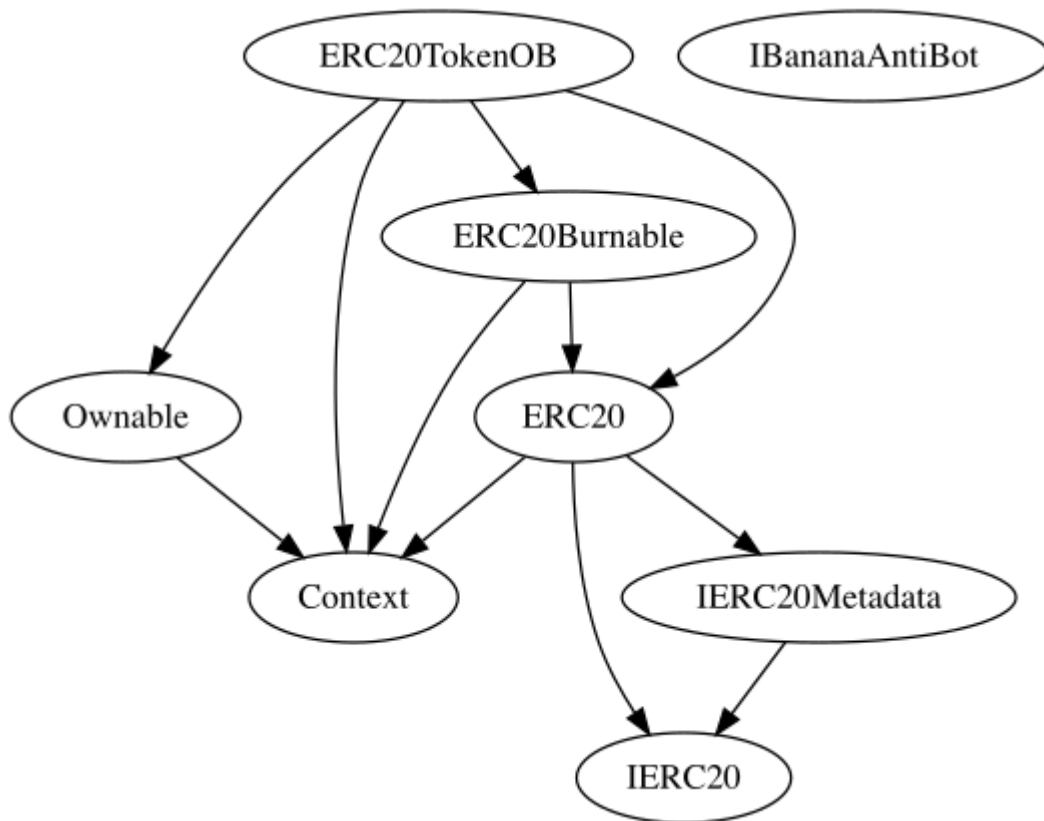
Recommendation

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.

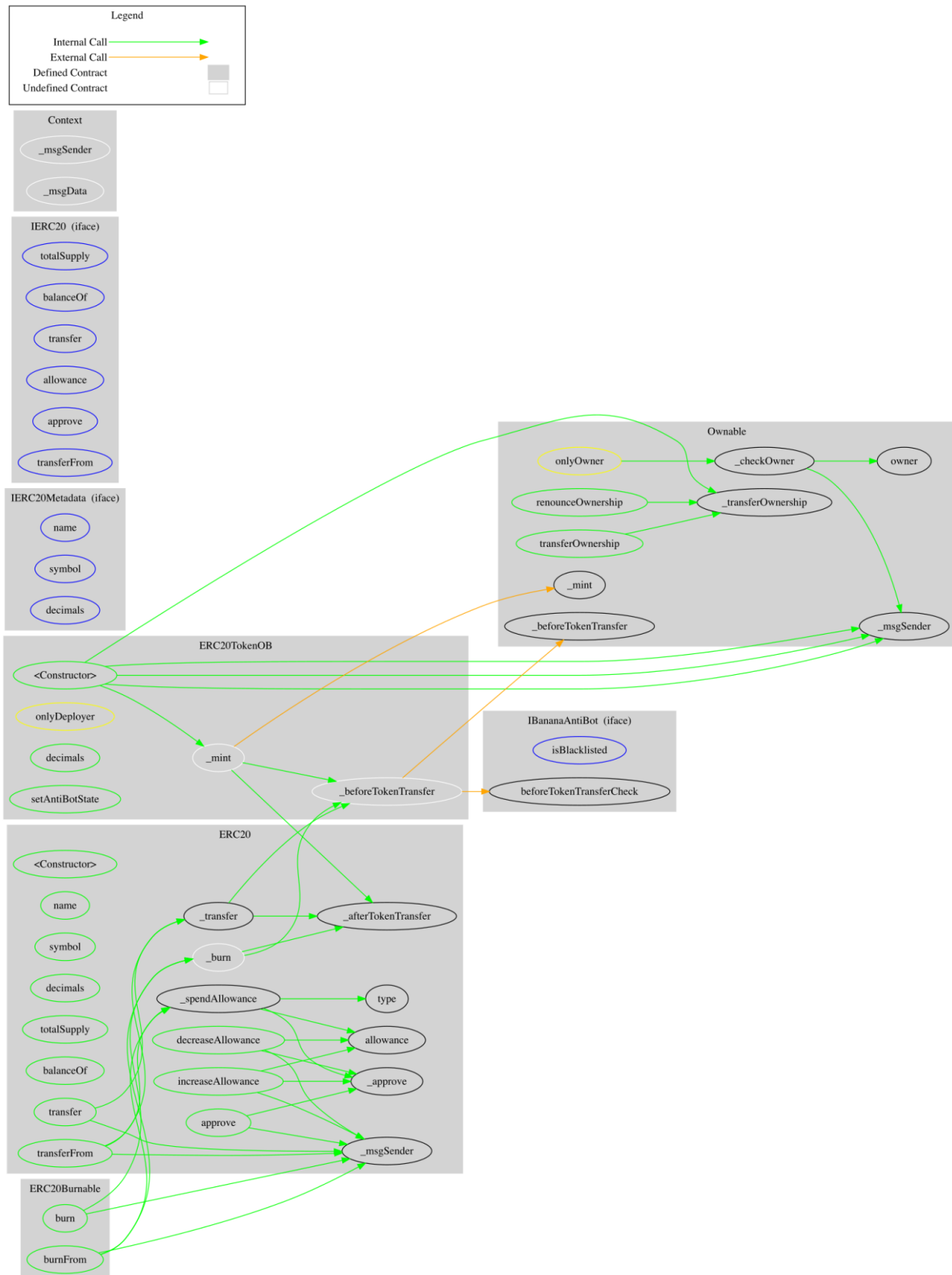
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
ERC20TokenOB	Implementation	Context, ERC20, ERC20Burnable, Ownable		
		Public	✓	ERC20
	decimals	Public		-
	setAntiBotState	Public	✓	onlyDeployer
	_mint	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

Inheritance Graph



Flow Graph



Summary

Bananace contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. The contract uses an external untrusted source in order to determine the transaction flow. The team is advised to use a commonly recognized or audited contracts for external interactions.

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>



Cyberscope

Audit Report

Bananace

May 2023

Network BSC

Address 0x3194e783fdbaff5edacb71afb6e4c8d7aa67ac61

Audited by © cyberscope

Table of Contents

Table of Contents	1
Review	2
Audit Updates	2
Source Files	3
Findings Breakdown	4
Analysis	5
Diagnostics	6
RDE - Redundant Decimals Extension	7
Description	7
Recommendation	7
RMO - Redundant Mint Override	8
Description	8
Recommendation	8
L19 - Stable Compiler Version	9
Description	9
Recommendation	9
Functions Analysis	10
Inheritance Graph	11
Flow Graph	12
Summary	13
Disclaimer	14
About Cyberscope	15

Review

Contract Name	ERC20TokenOB
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://bscscan.com/address/0x3194e783fdbaff5edacb71afb6e4c8d7aa67ac61
Address	0x3194e783fdbaff5edacb71afb6e4c8d7aa67ac61
Network	BSC
Symbol	NANA
Decimals	18
Total Supply	696,969,696,969,696

Audit Updates

Initial Audit	12 May 2023 https://github.com/cyberscope-io/audits/blob/main/nana/v1/audit.pdf
Corrected Phase 2	14 May 2023 https://github.com/cyberscope-io/audits/blob/main/nana/v2/audit.pdf
Corrected Phase 3	17 May 2023

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249a a4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	bce14c3fd3b1a668529e375f6b70ffdf9cef 8c4e410ae99608be5964d98fa701
@openzeppelin/contracts/token/ERC20/extensions /ERC20Burnable.sol	0344809a1044e11ece2401b4f7288f414ea 41fa9d1dad24143c84b737c9fc02e
@openzeppelin/contracts/token/ERC20/extensions /IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166 689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db800 3d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a2 3a4baa0b5bd9add9fb6d6a1549814a
contracts/interfaces/IPinkAntiBot.sol	8a41eb6e4420666d592566e7b99eccc149 b7e91b238f5933564ac6d8e8a50671
contracts/tokens/ERC20TokenOB.sol	2c26f2f279ef0dcfb73ec56b766fa5575635 f18ca92b3425d81dc3b9016248d9

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

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	RDE	Redundant Decimals Extension	Unresolved
●	RMO	Redundant Mint Override	Unresolved
●	L19	Stable Compiler Version	Unresolved

RDE - Redundant Decimals Extension

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L29
Status	Unresolved

Description

The contract implements the Openzeplin ERC20 standard. According to the ERC20 standard, the decimals are 18. The contract overrides the decimals method. This method returns the same number as the ERC20 standard. As a result, the override is redundant.

```
function decimals() public view virtual override returns (uint8) {  
    return DECIMALS;  
}
```

Recommendation

The team is advised to remove the decimals override since it will produce the same result.

RMO - Redundant Mint Override

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L40
Status	Unresolved

Description

The contract executes the mint method once in the constructor. The `mint()` method is overridden by the contract to allow only non-zero mints. Since the mint method is called once in the constructor, then the override of the mint method is redundant.

```
function _mint(address account, uint256 amount) internal virtual override {  
    if (amount > 0) {  
        super._mint(account, amount);  
    }  
}
```

Recommendation

The team is advised to remove the override of the `mint()` method since it will produce the same result.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/tokens/ERC20TokenOB.sol#L3
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 ^0.8.17;
```

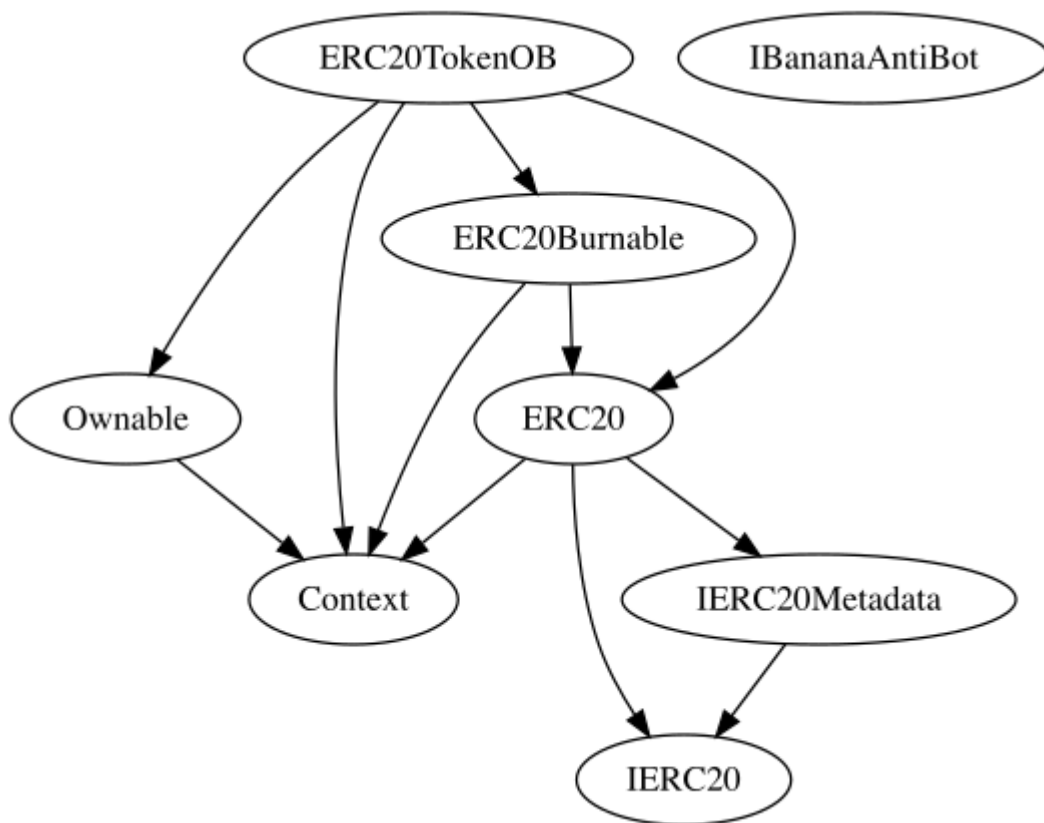
Recommendation

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.

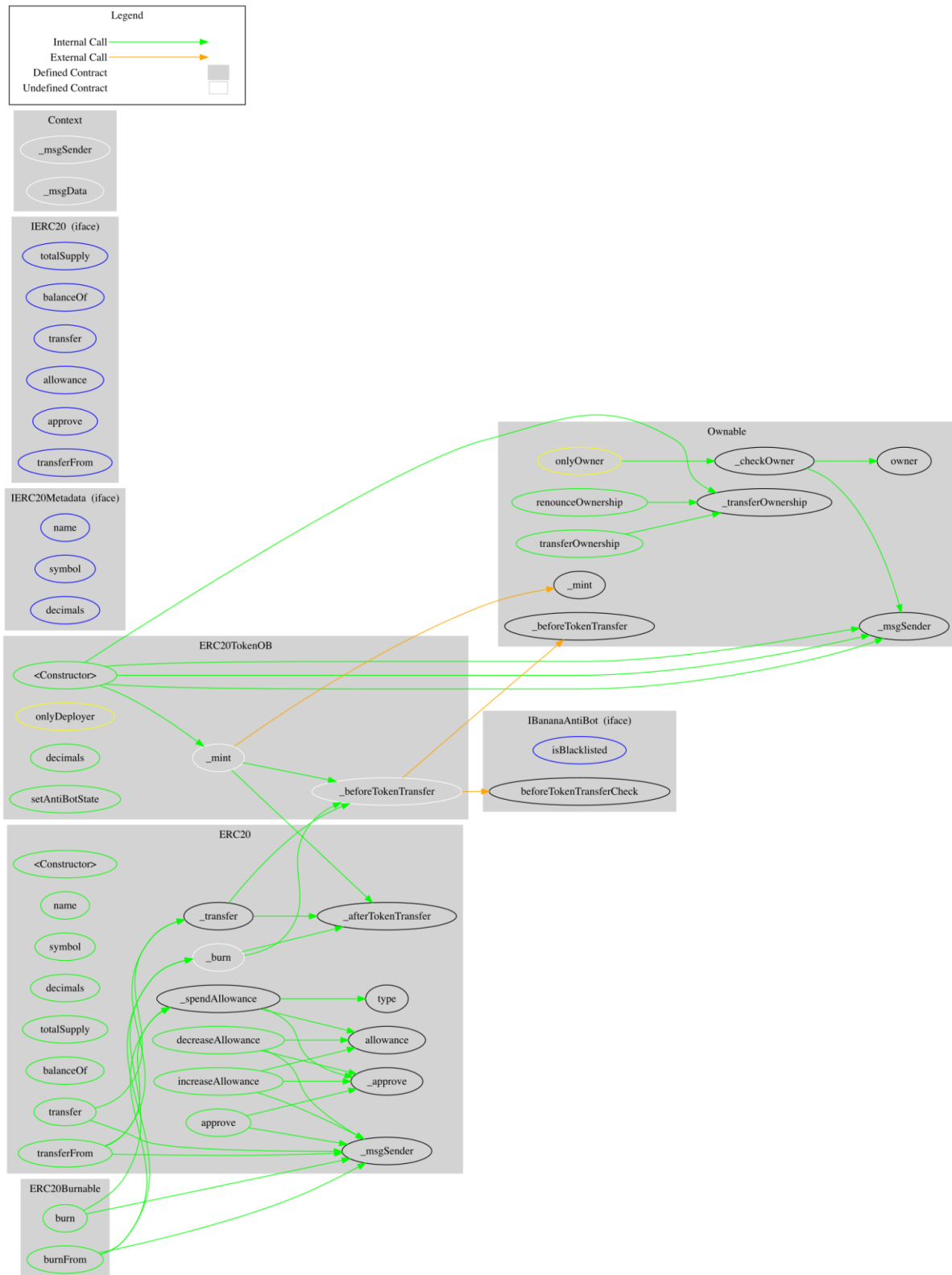
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
ERC20TokenOB	Implementation	Context, ERC20, ERC20Burnable, Ownable		
		Public	✓	ERC20
	decimals	Public		-
	setAntiBotState	Public	✓	onlyDeployer
	_mint	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

Inheritance Graph



Flow Graph



Summary

Bananace contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Bananace is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

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>