Cyberscope

Audit Report

AI MASA

July 2023

Network	BSC Testnet
Address	0x8073D97B896299188681999E05DdF046fEF73fc8
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

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

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	PFM	Potential Functions Misuse	Unresolved
	L19	Stable Compiler Version	Unresolved

Table of Contents

Review

Contract Name	MASA
Compiler Version	v0.8.16+commit.07a7930e
Optimization	800 runs
Explorer	https://testnet.bscscan.com/address/0x8073d97b896299188681999e05ddf046fef73fc8
Address	0x8073d97b896299188681999e05ddf046fef73fc8
Network	BSC_TESTNET
Decimals	18

Audit Updates

Initial Audit	19 Jul 2023

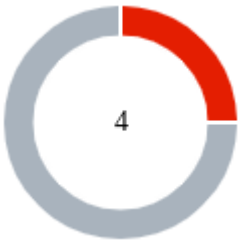
Source Files

Filename	SHA256
contracts/MASA.sol	84a78f52b4ae4382120f93ca3e65628d4ce62fa1f64efcfa33fb898e267fa335
@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol	68f5690fc266a6b48386c28cbfd72ec67c24b05a51ce26d24103577c15f61401
@openzeppelin/contracts-upgradeable/utils/StorageSlotUpgradeable.sol	05b696b46ca1be28e19dfba65ea71c3b3615bd39d19bfd8212864a16c54870fd

@openzeppelin/contracts-upgradeable/utils/CountersUpgradeable.sol	5c1ac829a429b0c2ca9b4c9ed8b78d412320e9175e45f088c4e9056ef95fbf21
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	5fb301961e45cb482fe4e05646d2f529aa449fe0e90c6671475d6a32356fa2d4
@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	1d7d481b79fd54d957c9a0696f6227f7799fec01d8ba41f5c130a7cc6b4eddc9
@openzeppelin/contracts-upgradeable/utils/math/SafeCastUpgradeable.sol	647d03e70d45c15cd9aa3afc3b32de945ec024a022614e263f33bb35c557ac94
@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol	158a0316fa289fad12c2ca764449e43e6724fb79c58fc438508d116f9af46b39
@openzeppelin/contracts-upgradeable/utils/cryptography/EIP712Upgradeable.sol	91e9d20515fa1516a9e9dd754b8a3ced5552f955b039dbe69d08c566fbd2e024
@openzeppelin/contracts-upgradeable/utils/cryptography/ECDSAUpgradeable.sol	2aee2a508bebf8e55bf78814d9d66a7a21c35c171e4010dfc3888c031f193628
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	4e09a7479aa3e7c313f8fc141c4c8fc04e0abfeb8754615ef7d78ec94c298b07
@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol	7307fb68607d3c93995797209010e5048c9cc1777f3b97dc7940f41a7327d080
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol	b97515a88e75c313eacf0a27c9439ef371d86d4c2730d3b13076640942f813df
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-ERC20PermitUpgradeable.sol	6d6ffe69a38a39c69acde1dd5edb74f80cff046c4a66d1cd816b98ca741c9a43
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol	68bcca423fc72ec9625e219c9e36306c726a347e43f3711467c579bd3f6500c8
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20VotesUpgradeable.sol	88763a9a0b498ca738c9a1c0c33a56464e0e8a2ad466426fe10b01cd9e01e2ae

@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20BurnableUpgradeable.sol	ca660e828b0c4be205a9f56f3b87b91c1fa67cfd0f6e9dbd431faea7a6280d36
@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol	c05b019a0b3bee8f3fac2da7c929f7d665b97d6d046aa35126615fff11205119
@openzeppelin/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol	7517b26ac0cee066447b94cbf7df8ad5ce91cc6ddf0fd1e3425fe978889f5eb0
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	98ce2984e449716f24043a8c11bbe969a6d34878b1d522b92c88d62708ba3376
@openzeppelin/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol	e0ac7115916f0dce0a8e80769694736f3e674bdc5b2e5853964c82004b1e1cc5
@openzeppelin/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol	f6c1a8b4512e9cc0168278c2a634b184fd86b1e39c7c283bcf34fb154236fc5d
@openzeppelin/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol	a94576fd98585c07b2a9725f7c89c910a3a1909a03f49ec2df465327c6a0ffc3
@openzeppelin/contracts-upgradeable/governance/utils/IVotesUpgradeable.sol	400936c02700eb4147c65a91a15fb6f90d074d7519f8ebce49dce78a2c917186
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	da66c17044345dc892d85bd7ddc9745d25df0b3dacfba8f84eb87c60d6e40fe3

Findings Breakdown



Critical	1
Medium	0
Minor / Informative	3

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

ST - Stops Transactions

Criticality	Minor / Informative
Location	contracts/MASA.sol#L39
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users including the owner, by calling the `pause` method.

```
    function pause() public onlyOwner {
        _pause();
    }
```

Recommendation

It is recommended to consider removing the `pause` functionality from the contract entirely, especially if there are no other intended purposes or justifiable use cases for it. Also 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.

MT - Mints Tokens

Criticality	Critical
Location	contracts/MASA.sol#L47
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 to, uint256 amount) public onlyOwner {
      _mint(to, 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.

PFM - Potential Functions Misuse

Criticality	Minor / Informative
Location	contracts/UUPSUpgradeable.sol#L72,85
Status	Unresolved

Description

The contract contains the `upgradeTo` and `upgradeToAndCall` functions that facilitate the upgradeability of the proxy's implementation. While these functions are essential for maintaining and improving the contract over time, there exists a potential risk. If these functions misused, they could redirect the proxy to point to a malicious or unintended implementation. Such a scenario could compromise the contract's intended behavior, potentially leading to loss of funds, unauthorized access, or unintended fuctionalities.

```
   function upgradeTo(address newImplementation) external
virtual onlyProxy {
        _authorizeUpgrade(newImplementation);
        _upgradeToAndCallUUPS(newImplementation, new bytes(0),
false);
    }

    function upgradeToAndCall(address newImplementation, bytes
memory data) external payable virtual onlyProxy {
        _authorizeUpgrade(newImplementation);
        _upgradeToAndCallUUPS(newImplementation, data, true);
    }
```

Recommendation

It is recommended to implement robust access controls and governance mechanisms around the `upgradeTo` and `upgradeToAndCall` functions. Only trusted entities, such as contract administrators or a multi-signature wallet, should have the authority to invoke these functions.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/MASA.sol#L2
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 ^0.8.9;
```

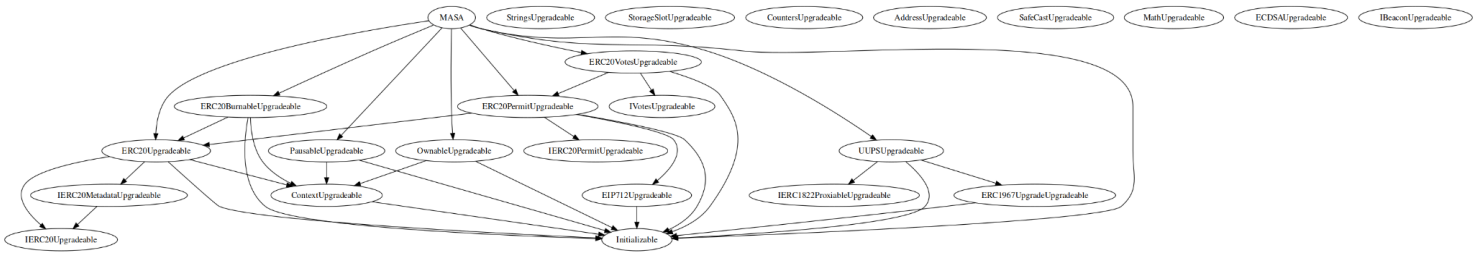
Recommendation

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.

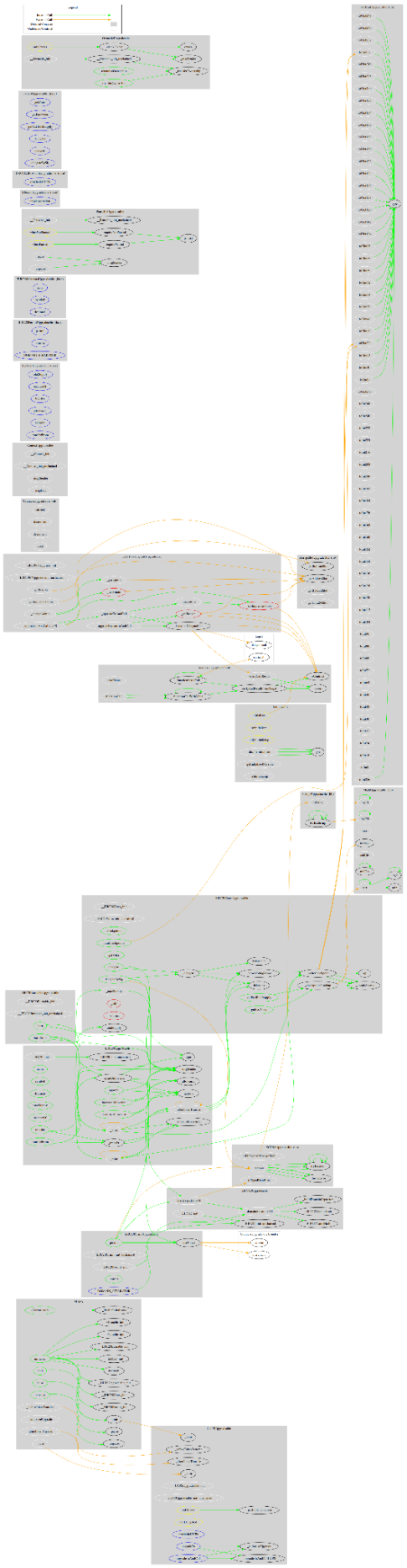
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
MASA	Implementation	Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, OwnableUpgradeable, ERC20PermitUpgradeable, ERC20VotesUpgradeable, UUPSUpgradeable		
		Public	✓	-
	initialize	Public	✓	initializer
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	mint	Public	✓	onlyOwner
	_beforeTokenTransfer	Internal	✓	whenNotPaused
	_authorizeUpgrade	Internal	✓	onlyOwner
	_afterTokenTransfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	

Inheritance Graph



Flow Graph



Summary

AI MASA 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 stop transactions and 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.

Initial Audit, 19 Jul 2023

At the time of the audit report, the contract with address

0x8073D97B8962991886819 99E05DdF046fEF73fc8 is pointed by the following proxy

address: 0x21BAEAE318F0B2B4D6FaAEc38bd3E5572905aAC6.

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