Cyberscope

# Audit Report

# zoozToken

March 2023

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0x132306a39d6fC1E49C3Cb6D8FE8d07d4D44B462a |
| Audited by | © cyberscope |

# Table of Contents

# Review

| | |
|---|---|
| **Contract Name** | ZOOZToken |
| **Compiler Version** | v0.8.7+commit.e28d00a7 |
| **Optimization** | 200 runs |
| **Explorer** | https://bscscan.com/address/0x132306a39d6fc1e49c3cb6d8fe8d07d4d44b462a |
| **Address** | 0x132306a39d6fc1e49c3cb6d8fe8d07d4d44b462a |
| **Network** | BSC |
| **Symbol** | ZOOZ |
| **Decimals** | 9 |
| **Total Supply** | 770,000,000 |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 16 Feb 2023<br>https://github.com/cyberscope-io/audits/blob/main/zooz/v1/audit.pdf |
| **Corrected Phase 2** | 01 Mar 2023<br>https://github.com/cyberscope-io/audits/blob/main/zooz/v2/audit.pdf |
| **Corrected Phase 3** | 13 Mar 2023 |

# Source Files

| **Filename** | SHA256 |
|---|---|
| **ZOOZToken.sol** | ecdbda204d2c247ad0ef19a55e984d73b8dbd05658a353049a35ad946bd1d1b7 |

# Roles

The contract consist of three roles. The `owner`, `manager,` and `governor` roles.

The `Owner` has the authority to

- Grant or revokes the `governor` and the `manager` role.
- Renounce or Transfer ownership.
- Set blacklisted addresses.
- Set excluded from fees addresses.

The `Manager` has the authority to

- Transfer the `manager` role.
- Configure reward address.

The `Governor` has the authority to

- Set blacklisted addresses.
- Set excluded from fees addresses.

# 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 | Unresolved |

# BC - Blacklists Addresses

| | |
|---|---|
| **Criticality** | Medium |
| **Location** | contracts/ZOOZToken.sol#L467 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the `setBlockedAddress` function.

```
function setBlockedAddress(address holderAddress, bool blocked) public
onlyGovernance()  {
        require(holderAddress != address(0), "HolderAddress can't be the zero
address");

    blockedAddresses[holderAddress][_msgSender()] = blocked;

        if(blocked) {
            emit AddressBlocked(holderAddress);
            return;
        }

        emit AddressUnblocked(holderAddress);
}
```

## 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. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# Diagnostics

Critical Medium Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | MVN | Misleading Variables Naming | Unresolved |
| ● | L04 | Conformance to Solidity Naming Conventions | Unresolved |

# MVN - Misleading Variables Naming

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/ZOOZToken.sol#L210,384 |
| **Status** | Unresolved |

## Description

Variables can have misleading names if their names do not accurately reflect the value they contain or the purpose they serve. The contract uses some variable names that are too generic or do not clearly convey the information stored in the variable. Misleading variable names can lead to confusion, making the code more difficult to read and understand.

The contract utilizes the variable `botAddresses` and the function `_isItBotAddress`. These variables and the function implement an exclude from the fees mechanism.

```
mapping (address => mapping (address => bool)) internal botAddresses;

function _isItBotAddress(address addr) internal view returns(bool) {
    return botAddresses[addr][governance1Address]
        && botAddresses[addr][governance2Address]
        && botAddresses[addr][governance3Address];
}
```

## Recommendation

It's always a good practice for the contract to contain variable names that are specific and descriptive. The team is advised to keep in mind the readability of the code.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | ZOOZToken.sol#L216 |
| **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.

```
uint256 public constant totalsupply
```

## Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

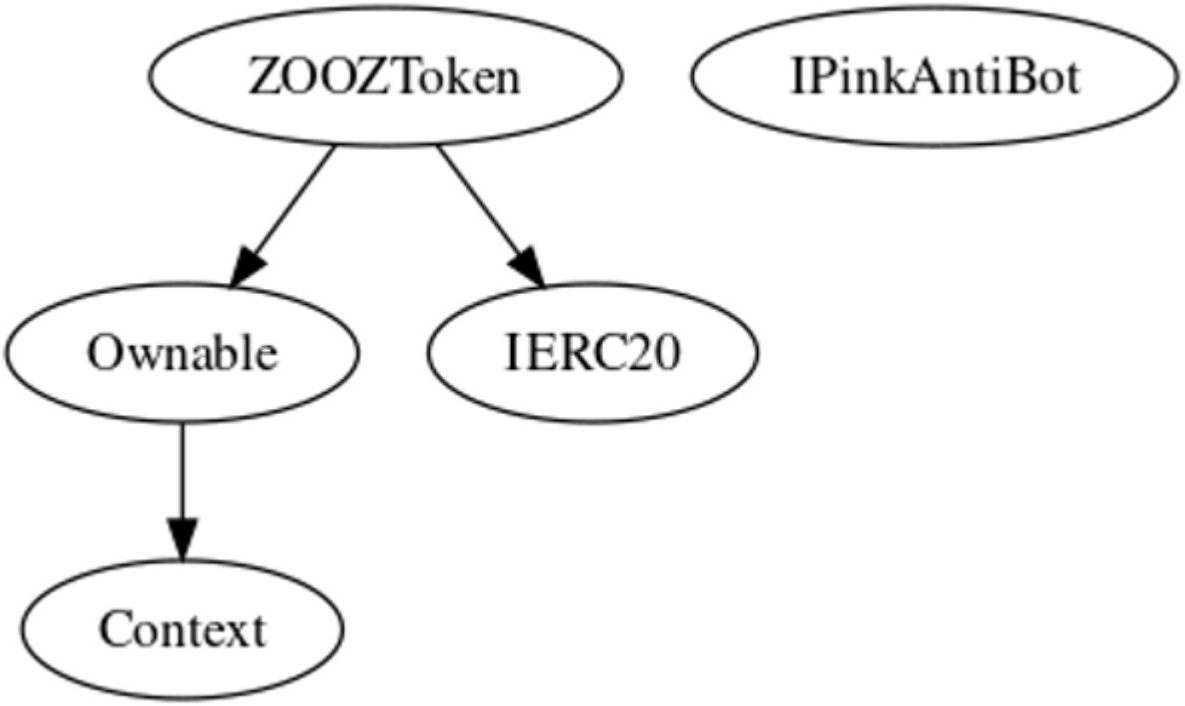
Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

# Functions Analysis

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

| | | | | |
|---|---|---|---|---|
| **ZOOZToken** | Implementation | Ownable, IERC20 | | |
| | | Public | ✓ | - |
| | totalSupply | Public | | - |
| | balanceOf | Public | | - |
| | timestampOf | Public | | - |
| | balancesOf | Public | | - |
| | transfer | Public | ✓ | - |
| | allowance | Public | | - |
| | approve | Public | ✓ | - |
| | transferFrom | Public | ✓ | - |
| | _approve | Private | ✓ | |
| | _transfer | Private | ✓ | |
| | _isBlockedAddress | Internal | | |
| | _isItBotAddress | Internal | | |
| | _getFees | Internal | | |
| | _holdDateHook | Internal | ✓ | |
| | _stdTransfer | Private | ✓ | |
| | setManagerAddress | Public | ✓ | onlyManager |
| | setRewardsTeamAddress | Public | ✓ | onlyManager |
| | setBlockedAddress | Public | ✓ | onlyGovernance |
| | setBotAddress | Public | ✓ | onlyGovernance |
| | setPair | Public | ✓ | onlyManager |
| | setEnableAntiBot | External | ✓ | onlyManager |
| | setGovernance | Public | ✓ | onlyOwner |

# Inheritance Graph



# Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
renounceOwnership
transferOwnership
<Constructor>
onlyOwner
_transferOwnership
_msgSender
owner
IPinkAntiBot (iface)
setTokenOwner
onPreTransferCheck
ZOOZToken
setBlockedAddress
setBotAddress
<Constructor>
totalSupply
balanceOf
timestampOf
balancesOf
onlyGovernance
allowance
onlyManager
transfer
setManagerAddress
setRewardsTeamAddress
transferFrom
approve
setPair
setEnableAntiBot
setGovernance
_transfer
_approve
_holdDateHook
_getFees
_isBotAddress
_isBlockedAddress
_stdTransfer
Context
_msgSender
_msgData

# Summary

ZOOZ Token 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 blacklist addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. The fee percentage decreases over time, as the time elapsed since the last transaction of the holder. If the time elapsed is less than or equal to 1 week, the fee percentage is 14%. After 1 week, the fee percentage decreases to 10% for the next 3 weeks (1 month total). After 3 months, the fee percentage decreases again to 5%, and after 6 months, the fee percentage decreases to 2%. Eventually, after more than 6 months the fee percentage reaches 0.

# 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