Trust Audits
Blockchain Security

SMART CONTRACT AUDITS AND SECURITY ANALYSIS REPORT

COSTUMER : MASCOT NETWORK
DATE : 09 MEI 2022

Introduction

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.
The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

Document

- Project Name : Mascot Network
- Type : Token
- Platform : BEP20
- Language : Solidity
- Methods : Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
- Website : https://mascotnetwork.top/

Table of Contents

Contract Code Audit – Token Overview

Contract Name

MasToken

Contract Address

0xffA905bb70B6d43CE7a8b84F2bA8e0a334B9765D

Complier Version

v0.6.12+commit.27d51765

Total Supply

1,899,822.347269 MAS

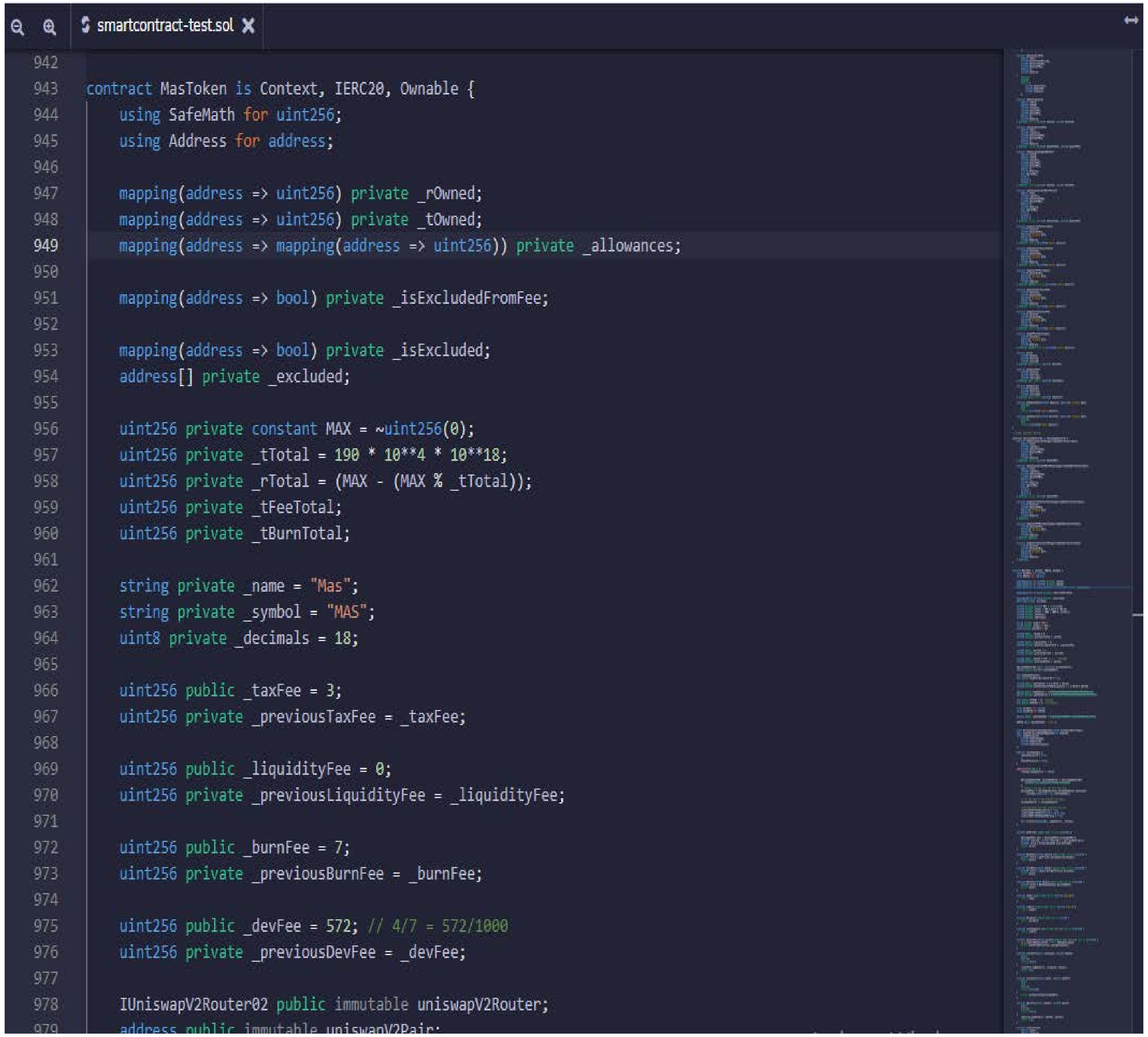
Deployer Address

0xECE95e2467DfF08F595C82E31baC6186c589e209

Decimals

18

BEP-20 Contract Code Audit – Overview



```
contract MasToken is Context, IERC20, Ownable {
    using SafeMath for uint256;
    using Address for address;

    mapping(address => uint256) private _rOwned;
    mapping(address => uint256) private _tOwned;
    mapping(address => mapping(address => uint256)) private _allowances;

    mapping(address => bool) private _isExcludedFromFee;

    mapping(address => bool) private _isExcluded;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 190 * 10**4 * 10**18;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;
    uint256 private _tBurnTotal;

    string private _name = "Mas";
    string private _symbol = "MAS";
    uint8 private _decimals = 18;

    uint256 public _taxFee = 3;
    uint256 private _previousTaxFee = _taxFee;

    uint256 public _liquidityFee = 0;
    uint256 private _previousLiquidityFee = _liquidityFee;

    uint256 public _burnFee = 7;
    uint256 private _previousBurnFee = _burnFee;

    uint256 public _devFee = 572; // 4/7 = 572/1000
    uint256 private _previousDevFee = _devFee;

    IUniswapV2Router02 public immutable uniswapV2Router;
    address public immutable uniswapV2Pair;
```

Contract Address

0xffA905bb70B6d43CE7a8b84F2bA8e0a334B9765D

Token Name

Token Mas (MAS)

Contract Creator

0xECE95e2467DfF08F595C82E31baC6186c589e209

Source Code

Solidity

Contract Name

MasToken

Other Settings

default evmVersion, MIT license

Compiler Version

v0.6.12+commit.27d51765

Optimization Enabled

No with 200 runs

The contract code is **verified** on BSCScan.

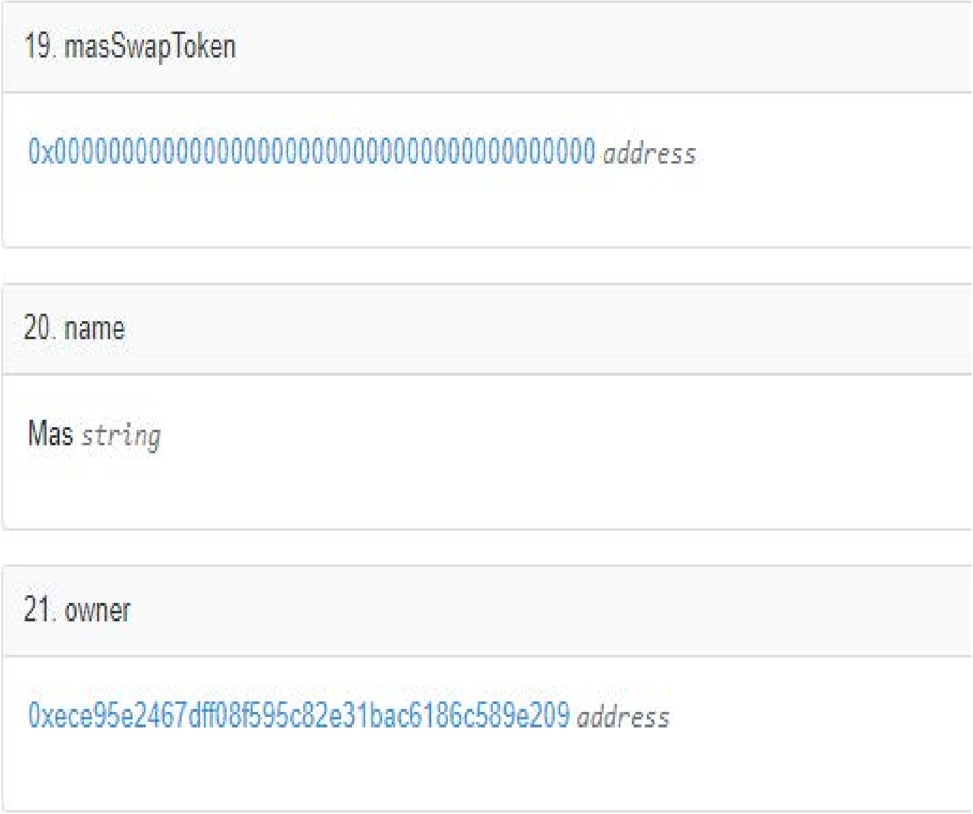
Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility (x3)	Complete	Complete	✓ Low Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on BSCScan.

The vulnerabilities listed above were not found in the token's Smart Contract

Contract Code Audit – Contract Ownership



19. masSwapToken

0x0000000000000000000000000000000000000000 *address*

20. name

Mas *string*

21. owner

0xece95e2467dff08f595c82e31bac6186c589e209 *address*

The contract ownership is not currently renounced.

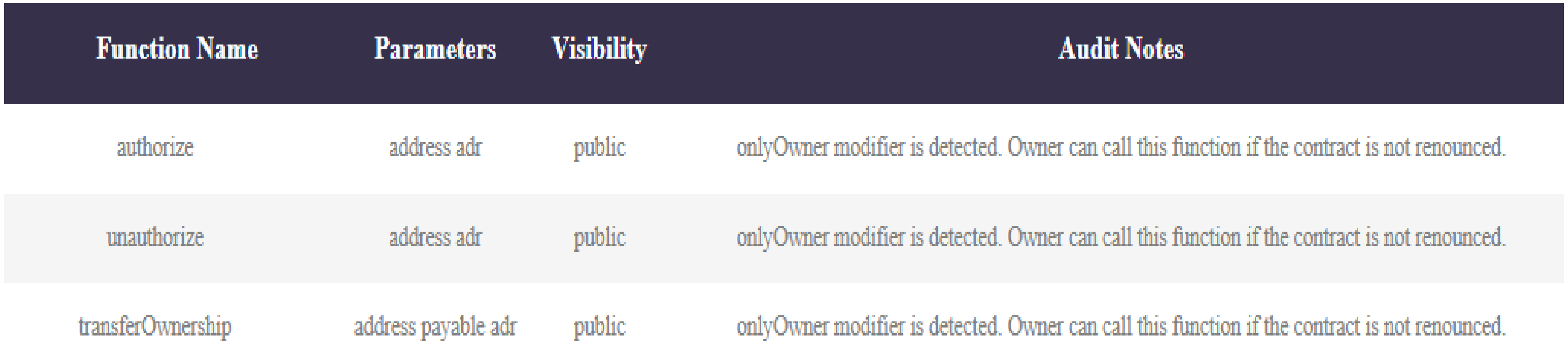
Owner address :
0xECE95e2467DfF08F595C82E31baC6186c589e209

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

We recommend:
- Establishing a Time-Lock with reasonable latency
- Assignment of privileged roles to multi-signature wallets

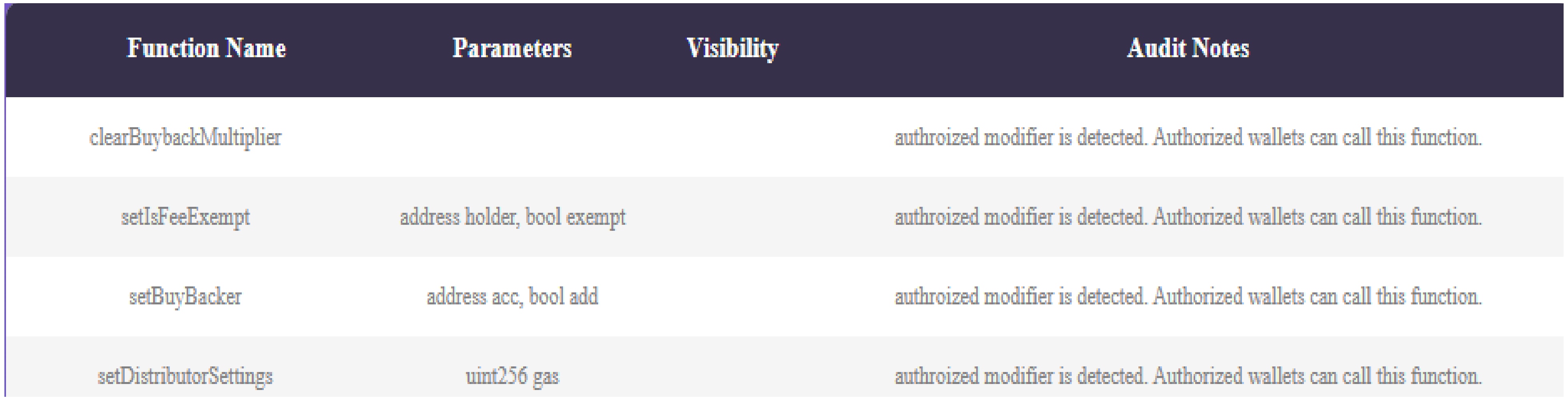
Contract Code Audit – Owner Accessible Functions



Function Name	Parameters	Visibility	Audit Notes
authorize	address adr	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
unauthorize	address adr	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
transferOwnership	address payable adr	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

The functions listed above can be called by the contract owner.
If contract ownership has been renounced there is no way for the above listed functions to be called.

Contract Code Audit – Authorized Accessible Functions



Function Name	Parameters	Visibility	Audit Notes
clearBuybackMultiplier			authroized modifier is detected. Authorized wallets can call this function.
setIsFeeExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setBuyBacker	address acc, bool add		authroized modifier is detected. Authorized wallets can call this function.
setDistributorSettings	uint256 gas		authroized modifier is detected. Authorized wallets can call this function.

The functions listed above can be called by authorized users.

Liquidity Ownership – Locked / Unlocked

No locked liquidity information has been found.

This page will contain links to locked liquidity for the project if we are able to locate that information.
Locked liquidity information was not found on the project's website.

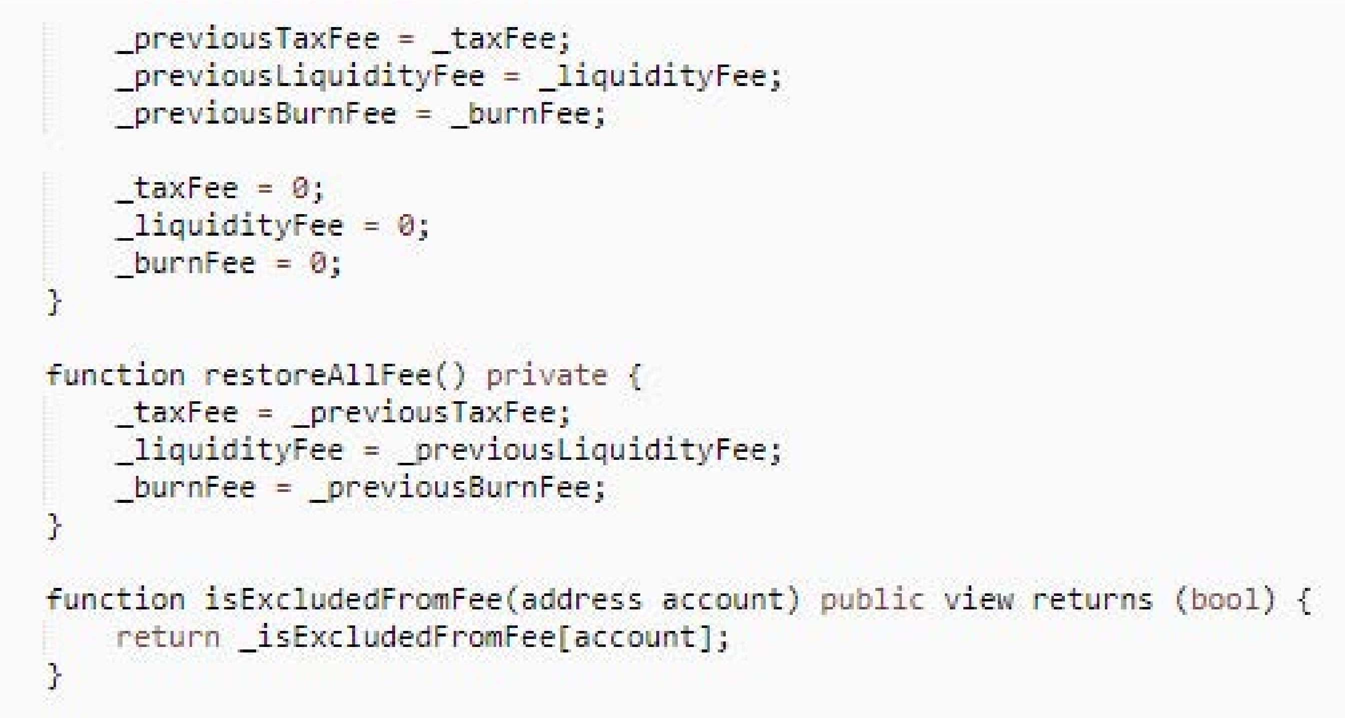
Contract Code Audit – Mint Functions

A mint function Was not found in the contract code.

We understand that sometimes mint function is very important for a project
but a project without mint function will have better trust for investors

Contract Transaction Fees

Fees: 0% Liquidity, 0% Distribution for holding (No Dev fees)



```
    _previousTaxFee = _taxFee;
    _previousLiquidityFee = _liquidityFee;
    _previousBurnFee = _burnFee;

    _taxFee = 0;
    _liquidityFee = 0;
    _burnFee = 0;
}

function restoreAllFee() private {
    _taxFee = _previousTaxFee;
    _liquidityFee = _previousLiquidityFee;
    _burnFee = _previousBurnFee;
}

function isExcludedFromFee(address account) public view returns (bool) {
    return _isExcludedFromFee[account];
}
```

t the time of Audit the transaction fees (“tax”) listed below are the fees associated with trading.
These fees are taken from every buy and sell transaction unless otherwise stated.

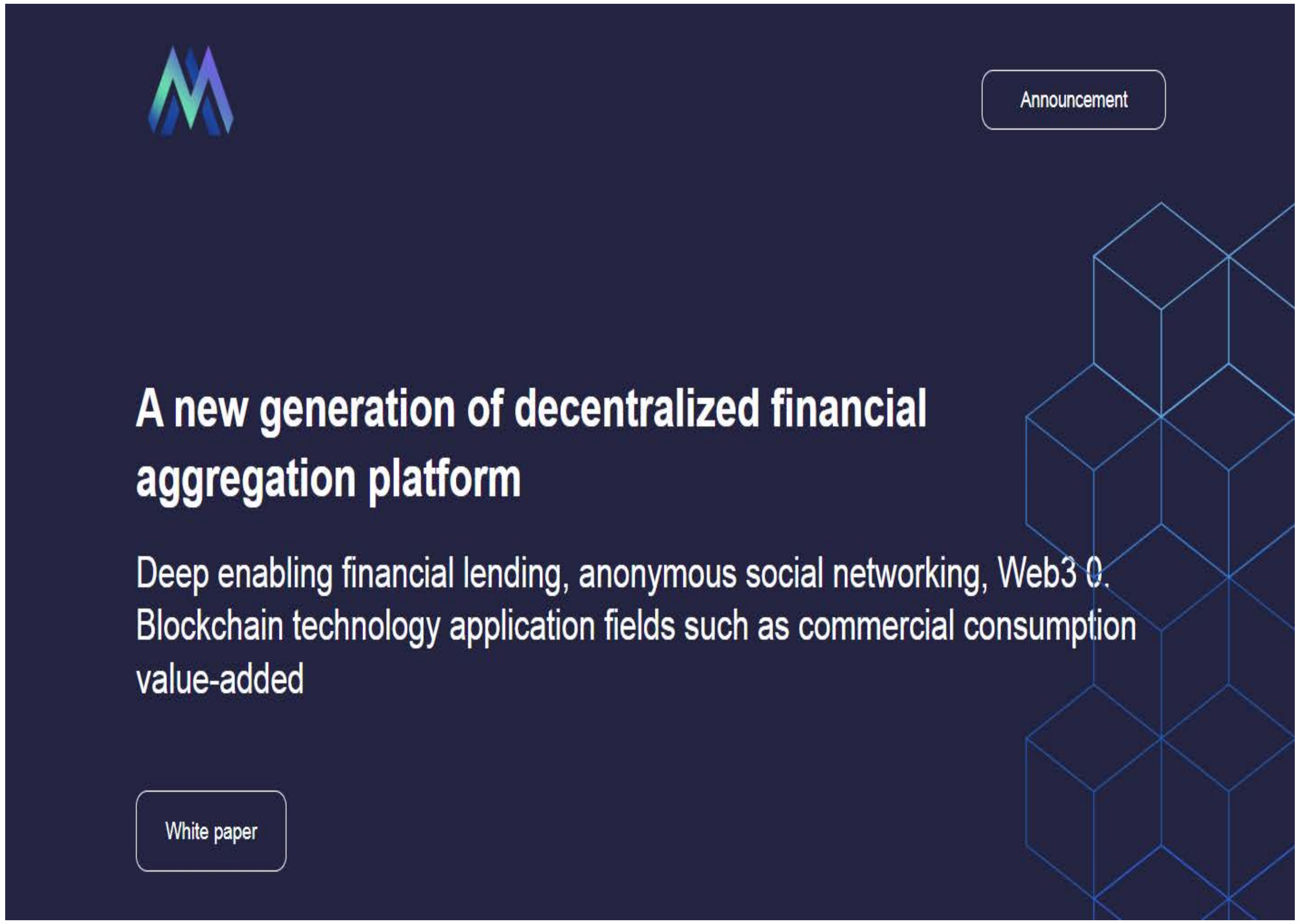
Website - Checklist

- ✓ Mobile Friendly
- ✓ No Javascript Error
- ✓ Spell Check
- ✓ SLL Certificate
- ✓ Website Loading

No additional issues were found on the website.

he website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

Website - Responsive HTML5 & CSS3



A new generation of decentralized financial aggregation platform

Deep enabling financial lending, anonymous social networking, Web3.0. Blockchain technology application fields such as commercial consumption value-added

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets.

Website - General Web Security

✓ SSL Certificate

A valid SSL certificate was found.
Issued by: TrustAsia TLS RSA CA Valid Until: 24/05/2023

✗ Contact Email

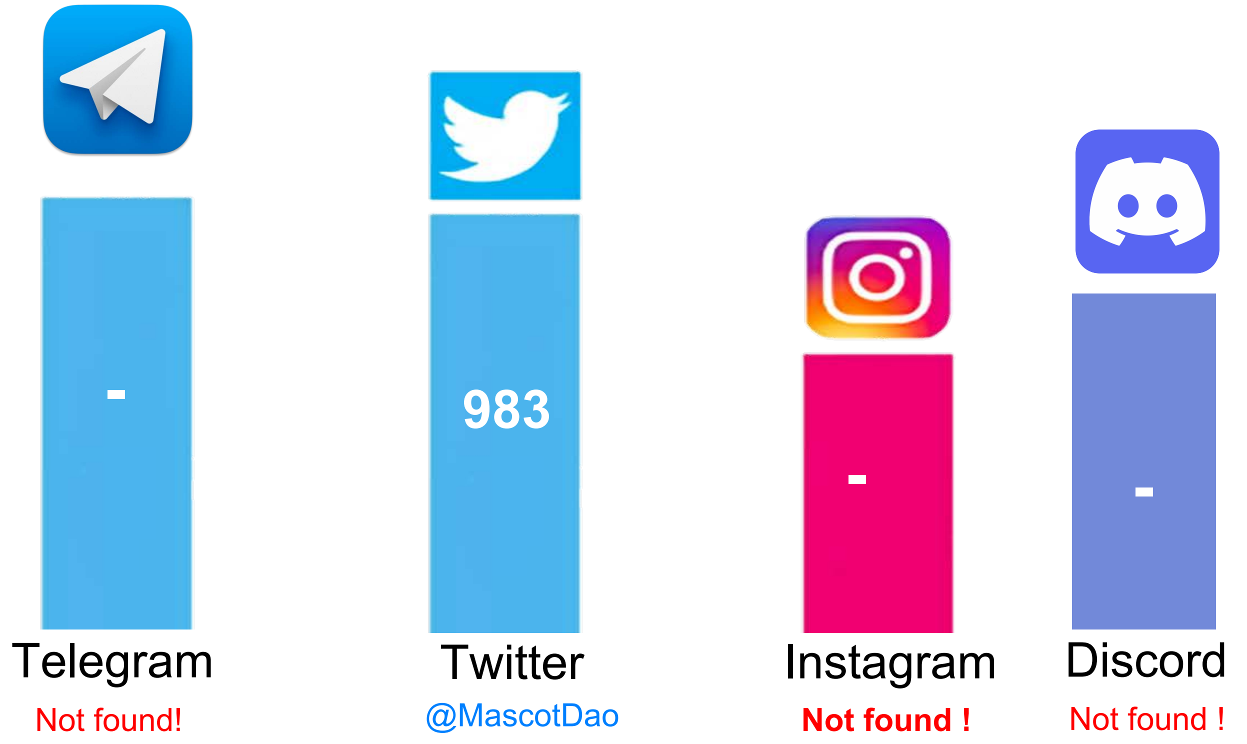
A valid contact email not found !

✓ Malware & Popups

No malware found
No injected spam found
No internal server errors
No popups found

Social media

Here are the social media stats for the project

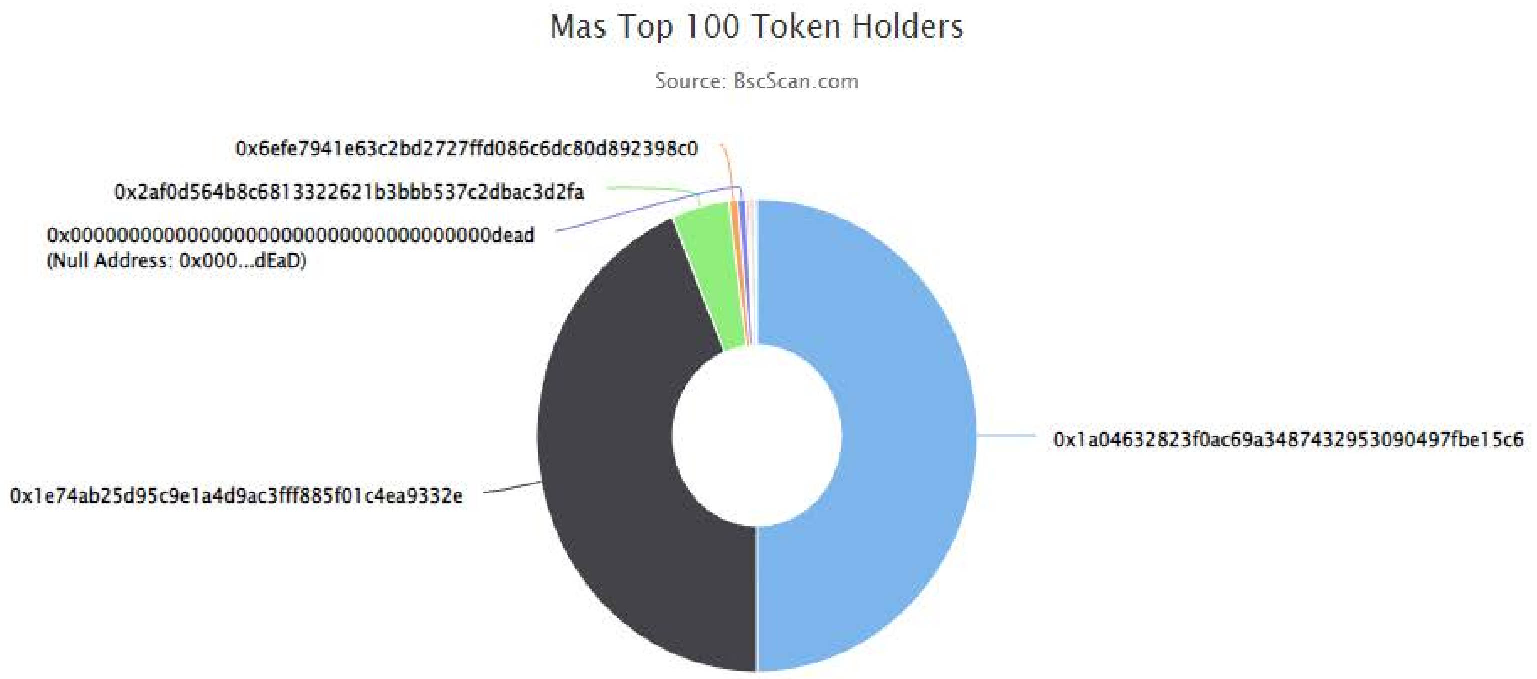


⊗ At least 3 social media networks we only found 1 active

Top Token Holders

The top token holders at the time of the audit are shown below.

Click here to view the most up-to-date list of holders



Mas Top 100 Token Holders

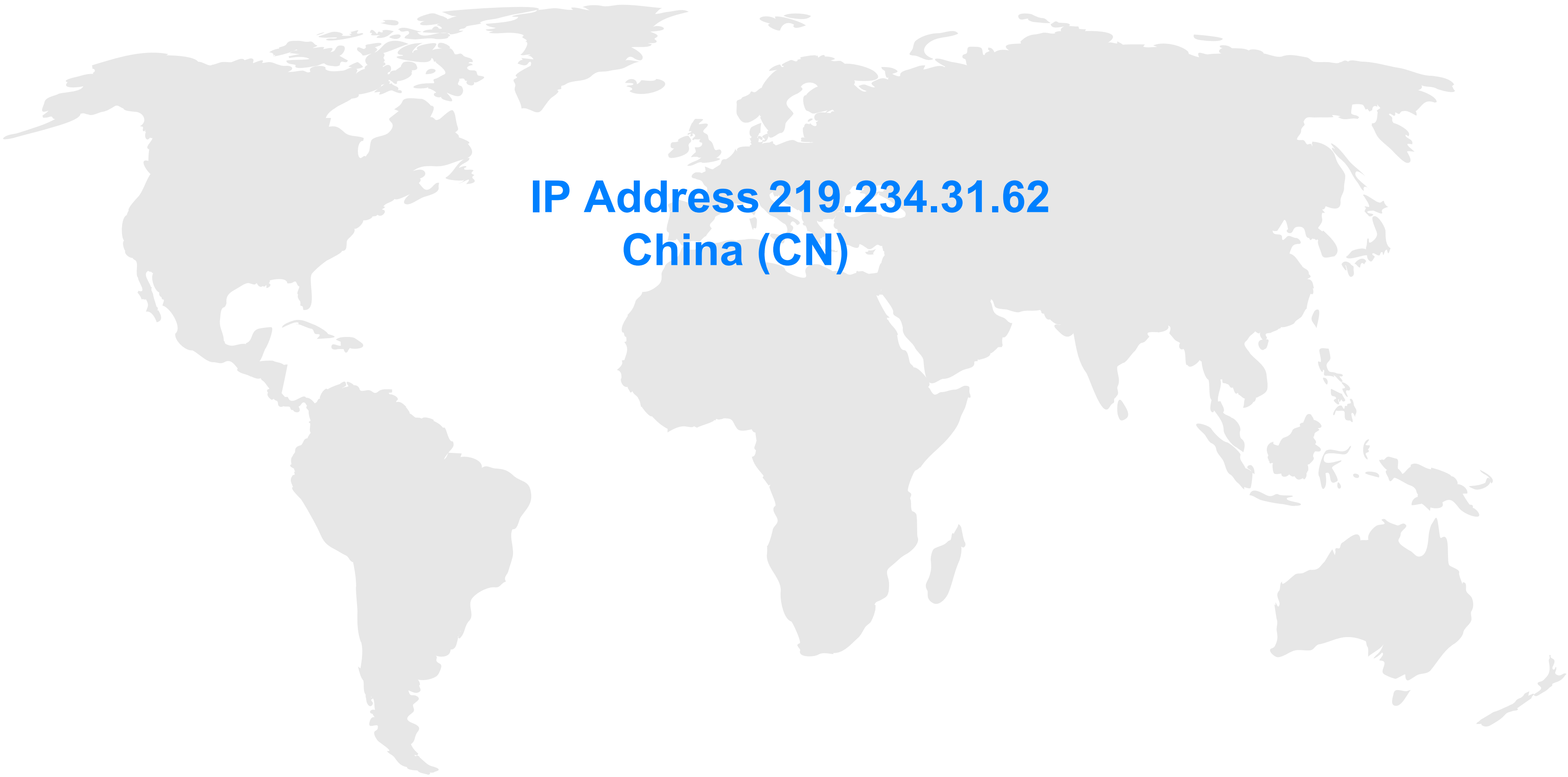
Source: BscScan.com

(A total of 1,896,235.64 tokens held by the top 100 accounts from the total supply of 1,899,822.34 token)

Rank	Address	Quantity (Token)	Percentage
1	0x1a04632823f0ac69a3487432953090497fbe15c6	950,000.0000987	50.0047%
2	0x1e74ab25d95c9e1a4d9ac3fff885f01c4ea9332e	830,681.902361	43.7242%
3	0x2af0d564b8c6813322621b3bbb537c2dbac3d2fa	80,668.7579	4.2461%
4	0x6efe7941e63c2bd2727ffd086c6dc80d892398c0	12,163.10694	0.6402%
5	0x000000000000000000000000000000000000dead	10,911.75981595	0.5744%

Location Audit

We were unable to identify a primary location for the project at this time or a location has not been declared.


IP Address 219.234.31.62
China (CN)

Team Overview

We didn't find any information about the team on the website at this time. Projects may choose to remain anonymous for a variety of reasons.

About Project

MASCOT NETWORK

is an open ecological platform, a comprehensive application ecosystem based on the principle of swap+defi+nft+dao+gamefi, which makes the construction more efficient, lower in cost, and provides all-round value in the market, traffic, resources, etc. MASCOT NETWORK uses VR, distributed storage, smart contracts and other technologies to build a decentralized linkage ecosystem with a high degree of freedom in the virtual world, creating a more free and efficient NFT asset swap agreement for players, providing high-quality diversified The combination method and the multi-modal products are perfectly integrated.

About the project found on the official website, we have conveniently placed it on this page for you to see.

Disclaimers

Trust Audits Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with Security vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit cannot guarantee the explicit security of the audited smart contracts.

TRUST AUDITS HAS BEEN COMPLETED AUDITS FOR COOKIE (CKE) AT BLOCK NUMBER: 16402753

THIS AUDITS IS ONLY VALID IF VIEWED ON HTTPS://WWW.TRUSTAUDITS.ORG

www.trustaudits.org

https://t.me/trustaudits