

Security Audit Report

Mancium (MANC)

ERC20 on Ethereum

The 13th of August, 2022



Table of Contents

1. Overview

- 1.1 Project Summary
- 1.2 Audit Summary
- 1.3 Vulnerability Summary

2. Findings

- 2.1 Fully Sanity Checks
- 2.2 Source Code Analysis
- 2.3 Contract Ownership
- 2.4 Liquidity Ownership
- 2.5 Mint Function
- 2.6 Burn Function

3. Project Overview

- 3.1 Present Mode
- 3.2 Team Location
- 3.3 General Web Security

4. Disclaimer

5. About

Overview

Project Summary	
Project Name	Mancium (MANC)
Platform	Ethereum
Language	Solidity
Contract Type	ERC20
Contract Address	0xE0c05ec44775e4AD62CDC2eEcdF337aA7A143363
Contract Owner	0x4c7c7358e3dc071A7fc9A50424425068743db834
Block Explorer	https://etherscan.com/

Audit Summary	
Delivery Date	August 13th, 2022 GMT+0
Block Number	14958019
Static Analysis	Yes
Graphic Analysis	Yes
Logic Disassemble	Yes
Manual Review	Yes

Vulnerability Summary

Severity Level	Total	Acknowledged	Alleviated	Resolved
Critical	0	0	0	0
Major	0	0	0	0
Medium	0	0	0	0
Minor	0	0	0	0
Informational	0	0	0	0
Discussion	1	1	0	0

Fully Sanity Checks

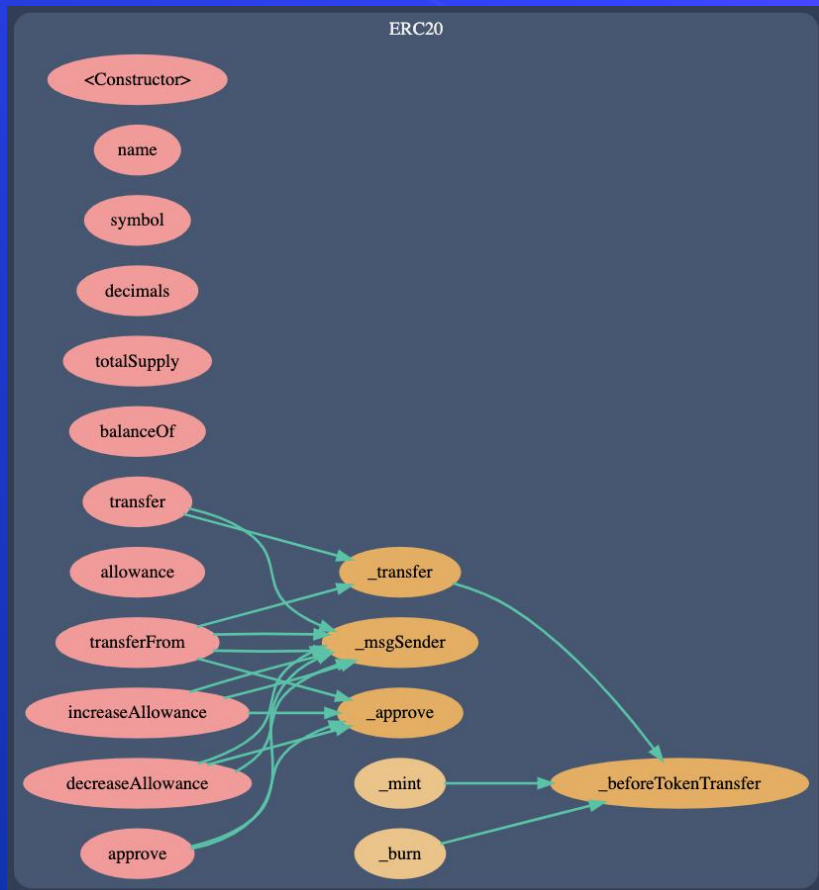
	Read	Write	AI Scanned	Human Reviewed	Result	Suggested	Resolved
name()	Yes		Completed	Completed	No Risk		
symbol()	Yes		Completed	Completed	No Risk		
balanceOf()	Yes		Completed	Completed	No Risk		
decimals()	Yes		Completed	Completed	No Risk		
totalSupply()	Yes		Completed	Completed	No Risk		
getBlacklist()	Yes		Completed	Completed	No Risk		
blacklistAddresses()	Yes		Completed	Completed	No Risk		
allowance()	Yes		Completed	Completed	No Risk		
pause()	Yes		Completed	Completed	No Risk		
approve()		Yes	Completed	Completed	No Risk		
decreaseAllowance()		Yes	Completed	Completed	✓ Low/No Risk		
increaseAllowance()		Yes	Completed	Completed	✓ Low/No Risk		
transfer()		Yes	Completed	Completed	✓ Low/No Risk		
transferFrom()		Yes	Completed	Completed	✓ Low/No Risk		
addAddressToBlacklist()		Yes	Completed	Completed	✓ Low/No Risk		
removeAddressFromBlacklist()		Yes	Completed	Completed	✓ Low/No Risk		

Source Code Analysis

```
contracts/8 Mancium.sol
505 function _burn(address account, uint256 amount) internal virtual {
506     require(account != address(0), "ERC20: burn from the zero address");
507     _beforeTokenTransfer(account, address(0), amount);
508     _burn(account, amount);
509     uint256 accountBalance = _balances[account];
510     require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
511     unchecked {
512         _balances[account] = accountBalance - amount;
513     }
514     _totalSupply -= amount;
515     emit Transfer(account, address(0), amount);
516     _afterTokenTransfer(account, address(0), amount);
517 }
518
519 function _approve(
520     address owner,
521     address spender,
522     uint256 amount
523 ) internal virtual {
524     require(owner != address(0), "ERC20: approve from the zero address");
525     require(spender != address(0), "ERC20: approve to the zero address");
526     _allowances[owner][spender] = amount;
527     emit Approval(owner, spender, amount);
528 }
529
530 function _spendAllowance(
531     address owner,
532     address spender,
533     uint256 amount
534 ) internal virtual {
535     uint256 currentAllowance = _allowance[owner, spender];
536     if (currentAllowance != type(uint256).max) {
537         require(currentAllowance >= amount, "ERC20: insufficient allowance");
538         unchecked {
539             _approve(owner, spender, currentAllowance - amount);
540         }
541     }
542 }
543
544 uint8 constant NUM_DECIMALS = 2;
545
546 //Total Supply: 1000000000, 100 million tokens
547 uint256 constant TOTAL_AMOUNT = 1000000000 * 10 ** 2;
548
549 contract Mancium is ERC20, Pausable, Burnable, Blacklist {
550     constructor() ERC20("Mancium", "MANC") {
551         _mint(owner(), TOTAL_AMOUNT);
552     }
553     function pause() public onlyOwner {
554         _pause();
555     }
556     function unpause() public onlyOwner {
557         _unpause();
558     }
559     function _beforeTokenTransfer(address from, address to, uint256 amount)
560     internal
561     whenNotPaused
562     override {
563         // This blocks transfer, transferFrom, burn and burnFrom calls from and
564         // to Blacklisted addresses
565         require(!blacklist[from], "from address is Blacklisted");
566         require(!blacklist[to], "to address is Blacklisted");
567         super._beforeTokenTransfer(from, to, amount);
568     }
569     function decimals() public view virtual override returns (uint8) {
570         return NUM_DECIMALS;
571     }
572 }
```

We've found 8 contracts in Mancium project source code and the partial screenshot of the contract code as left side shown.

- Mancium
 - ERC20
 - Context
 - IERC20
 - IERC20Metadata
 - Ownable
 - Blacklist
 - Pausable
- respectively.



ERC20 Contract

- name()
- symbol()
- decimals()
- totalSupply()
- balanceOf()
- allowance()

Read functions are running as expected while analyzing at the time of this writing.

- transferFrom()
- transfer()
- increaseAllowance()
- decreaseAllowance()
- approve()

Write functions are in no risk at the time of this writing

“ Contract Ownership

Contract Ownership Has Not Been Renounced at the Time of Audit.

The contract ownership is not currently renounced.

We just placed the contract of the owner address below for you to look up:

0x4c7c7358e3dc071A7fc9A50424425068743db834

Some feasible suggestions that would also mitigate the potential risk at a different level for privileged ownership.

- Time-lock with reasonable latency, e.g., 48 hours for awareness on privileged operations
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure, for example, due to the private key compromised



“ Liquidity Ownership

No Lock/Unlock Liquidity Logic for Owner 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 neither found on the project's website nor inside the contracts.

“ Mint Function

The Contract Cannot Mint New \$MANC Tokens.

We do understand that Mint functions are crucial to the functionality of the project, it's core related to its investors.

But a mint function was not found in the contract code.

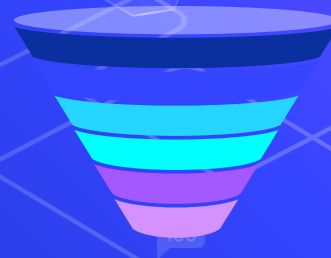


“ Burn Function

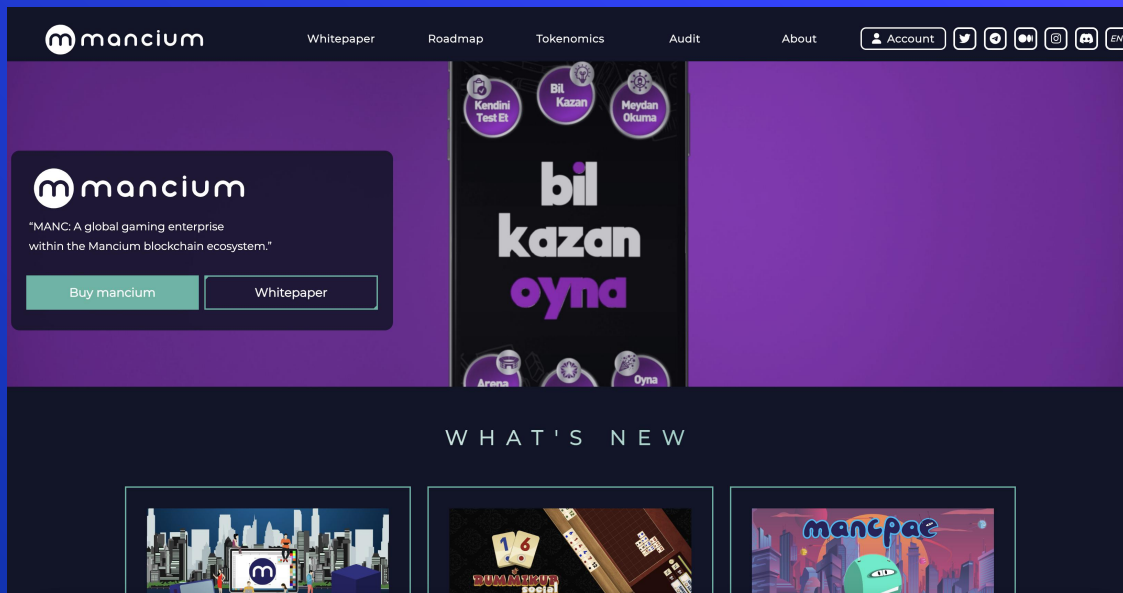
The Contract Cannot Burn Tokens at any cases.

No burn function has been found in the contract.

A burn function was inside the contract, but it's modified within internal which means no one is able to call to that.



Present Mode



The left image is an actual snapshot of the current live website.

The website was registered at the 18th of November 2021.

Team Location



Official Website: <https://www.mancium.io/>

General Web Security

DOMAIN

A valid domain hosted by
GoDaddy.com

Registered on 18-Nov-2021

mancium.io



Social Media Accounts

A bundle of social media accounts
was found.

Twitter: twitter.com/manciumtoken

Discord: discord.gg/RkggcR6GYq



A legal SSL certificate was found.
Issued at 18-Nov-2021

Signature Algorithm is
sha256WithRSAEncryption

SSL CERTIFICATE



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

Domain is marked clean by Google
and McAfee.

SPAM/MALWARE



Disclaimer



The opinions expressed in this document are for general informational purposes only and are not intended to provide specific advice or recommendations for any individual or on any specific investment. It is only intended to provide education and public knowledge regarding to this projects. This audit is only applied to the type of auditing specified in this report and the scope

of given in the results. Other unknown security vulnerabilities are beyond responsibility. MoonAudit only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, MoonAudit lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to MoonAudit or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, MoonAudit assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by MoonAudit still has the possibility that the entire risk cannot be completely detected. MoonAudit disclaims any liability for the resulting losses.

MoonAudit provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit scam. Please exercise caution when dealing with any cryptocurrency related platforms. The final interpretation of this statement belongs to MoonAudit.

MoonAudit highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.

About

MoonAudit has founded in 2021 by a squad of elite geeks on blockchain research and we analyze the loopholes in most smart contracts in ethereum-based chains. We, of course, offer the best-in-class report for your smart contracts auditing in non-evm public blockchains. Customer trusts smart contract, more trust security assessment report.





Thank You

MOONAUDIT.ORG. HAS BEEN COMPLETED FOR MANCIUM(\$MANC) AT BLOCK NUMBER: 14958019

THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://MOONAUDIT.ORG](https://moonaudit.org)

<https://MoonAudit.org>

<https://t.me/MoonAudit>