



SMART CONTRACT AUDIT REPORT

24 December 2022

Audited by
Solidity CheckUp




Table of Contents

Disclaimer	03
Audit Details	04
Contract Details	05
Token Details	06
Token Analysis & Top 10 Holders	07
Audit Overview	08
Conclusion & Analysis	11



Disclaimer

The information provided on this report document is only for general information and should not be used as a reason to invest. Solidity CheckUp Team will take no payment for manipulating the results of this audit. Solidity CheckUp Team does not guarantees that a project will not sell off team supply, or any other scam strategy.

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

Audit Details



Audited project

Merit Circle



Deployer Address

0x3cB580c041Cce953adfc2148e5BE6c1c893CCa9E



Client Contacts

Merit Circle Team



Website

www.meritcircle.io/



Blockchain

Ethereum



Contract Details

Token Contract Details for 24.12.2022	
Contract Name	MeritToken
Contract Address	0x949D48EcA67b17269629c7194F4b727d4Ef9E5d6
Total Supply	661,473,482.159355716992816509 MC
Token Ticker	MC
Decimals	18
Token Holders	8,002
Transections Count	187,556
Liquidity Fee	N/A
Tax Fee	N/A
Uniswap V2 Pair	N/A



Token Details

Details

Buy Fees: N/A

Sell Fees: N/A

Max TX: N/A

Max Sell: N/A

Honeypot Risk

Ownership: Renounced or N/A

Blacklist: Not Detected

Modify Max TX: Not Detected

Modify Max Sell: Not Detected

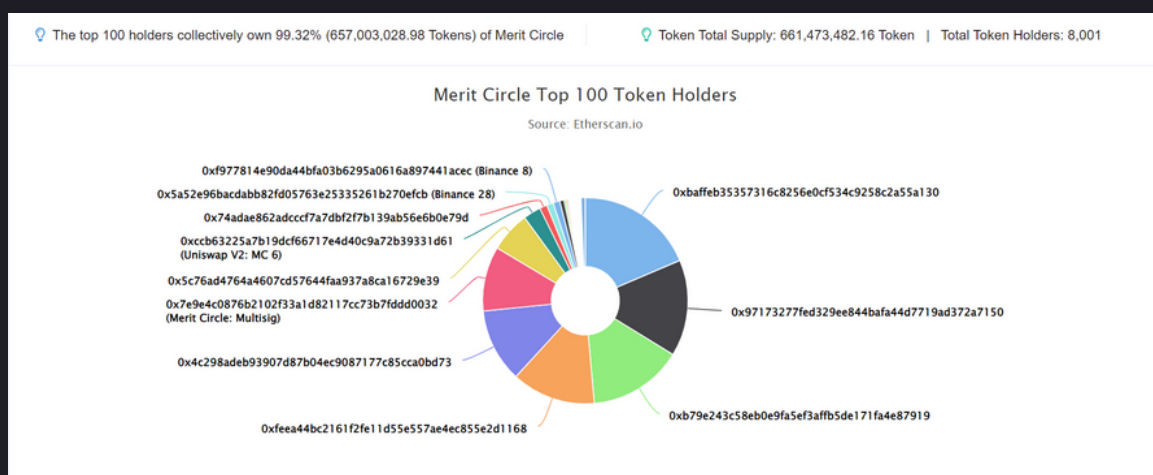
Disable Trading: Not Detected

Rug Pull Risk

Liquidity: Not Detected

Holders: Not Detected

Token Analysis & Top10 Holders



Merit Circle Token Contract and Distribution Chart (etherscan.io)

Rank	Address	Quantity (Token)	Percentage
1	0xbaffeb35357316c8256e0cf534c9258c2a55a130	123,128,600.905367758434529901	18.6143%
2	0x97173277fed329ee844bafa44d7719ad372a7150	100,000,000	15.1178%
3	0xb79e243c58eb0e9fa5ef3affb5de171fa4e87919	98,503,670	14.8916%
4	0xf6ea44bc2161f2fe1d55e557ae4ec855e2d1168	87,357,792.626860949319562086	13.2065%
5	0x4c298adeb93907d87b04ec9087177c85cca0bd73	76,646,466.706567852669487359	11.5872%
6	Merit Circle: Multisig	67,054,936.277972943054314916	10.1372%
7	0x5c76ad4764a4607cd57644faa937a8ca16729e39	42,251,841.493907515153975354	6.3875%
8	Uniswap V2: MC 6	18,290,455.134342934237427769	2.7651%
9	0x74adae862adcccf7a7dbf2f7b139ab56e6b0e79d	7,754,722.063140169009149636	1.1723%
10	Binance 28	7,000,000	1.0582%



Audit Overview

Security Issue

8 High

4 Medium

6 Low

0 Critical



Audit Overview

High Severity Issues

ERC20.sol

1. TransferFrom function: needs multiple require statements that assert that the sender and recipient are not zero addresses, the sender has the minimum amount in their balance, the caller has the minimum allowance of sender's tokens
2. _beforeTokenTransfer & _afterTokenTransfer: functions are incomplete and therefore pose extremely high security issue as they are used in key functions that involve the transfer of real money
3. _mint & _burn functions: These functions must be limited to specific role holders like admin role otherwise anyone can call the function and inflate the supply and profit from it

Would suggest modifier be included in contract to limit access to functions efficiently instead of using require statements for access.

IAccessControl.sol

4. hasRole function: needs an if else statement to return true or false to the function. This a key function so it is important that it works and is completed.
5. getRoleAdmin function: The function is incomplete and is a key function.
6. grantRole function: The caller must have the admin role and they must also emit grantedRole event
7. revokeRole function: The caller must have admin role and it must emit roleRevoked event
8. renounceRole function: The caller must have the role they want to revoke and it must emit roleRevoked event

Would suggest modifier be included in contract to limit access to functions efficiently instead of using require statements

draft-IERC20Permit.sol

9. All functions inside interface contract not completed



Audit Overview

Medium Severity Issues

ERC20.sol

1. Transfer function: needs 2 require statements to be added asserting that the recipient address is not a zero address and that the caller has the minimum amount in their balance
2. Approve function: needs require statement to assert that the spender is not a zero address
3. increaseAllowance function: needs require statement to assert that spender is not a zero address
4. decreaseAllowance function: needs 2 require statements that assert spender is not zero address and that spender has the minimum allowance for the caller

Low severity issues

ERC20.sol

1. TransferFrom function: needs to emit Approval event of updated allowance
2. increaseAllowance function: needs to emit Approval event of updated allowance
3. decreasedAllowance function: needs to emit Approval event of updated allowance

IERC20.sol

4. transferFrom function: needs to emit a transfer event
5. Approve function: needs to emit an Approval event
6. transfer function: needs to emit a transfer event

Conclusion & Analysis

- Smart Contracts within the scope were manually reviewed and analyzed with static tools.
- Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.
- Found no Critical issue during the first review.
- Liquidity pair contract security is not checked due to out of scope



Kindly check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.



Solidity checkUp

soliditycheckup.com