



# GOMEAT TOKEN SMART CONTRACT SECURITY AUDIT

Prepared By: **Block360**

**Date: July 2021**



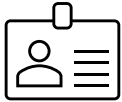
# AUDIT DETAILS



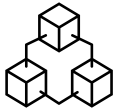
Project:  
QRC20Token



Deployer Address:  
HBATXqt28YaQbSKdNNRoV2N4wfBuJ8fjtC



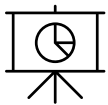
Client Contacts:  
GoMeat Team



Blockchain:  
Hydrachain



Start Date: 2021.06.17  
Completion Date: 2021.06.21



Overall Result:  
Needs to be updated.



Audit Team:  
Block360



# DISCLAIMER

The incumbent audit report is only limited to the type and scope of results specified in the table. Unknown security vulnerabilities are beyond the scope of this report. Block360 issued the said report by considering only existing attacks and/or vulnerabilities. Block360 is not responsible for any future changes and/or attacks on the security status of smart contracts. The security analysis and contents of the report are based on the information, documents and data provided by GoMeat Services EOOD. Except and only to the extent that it is prohibited by law, Block360 hereby excludes all liability and responsibility, and neither GoMeat Services EOOD nor any other entity shall have any claim against Block360, for any amount or kind of loss or damage that may result to GoMeat Services EOOD or any other entity (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

# INTRODUCTION

GoMeat Services EOOD commissioned Block360 to perform an audit of the QRC20Token smart contract, which uses SafeMath library.

Repository link for QRC20Token:

- <https://github.com/qtumproject/QRC20Token/blob/master/QRC20Token.sol>

Repository link for SafeMath:

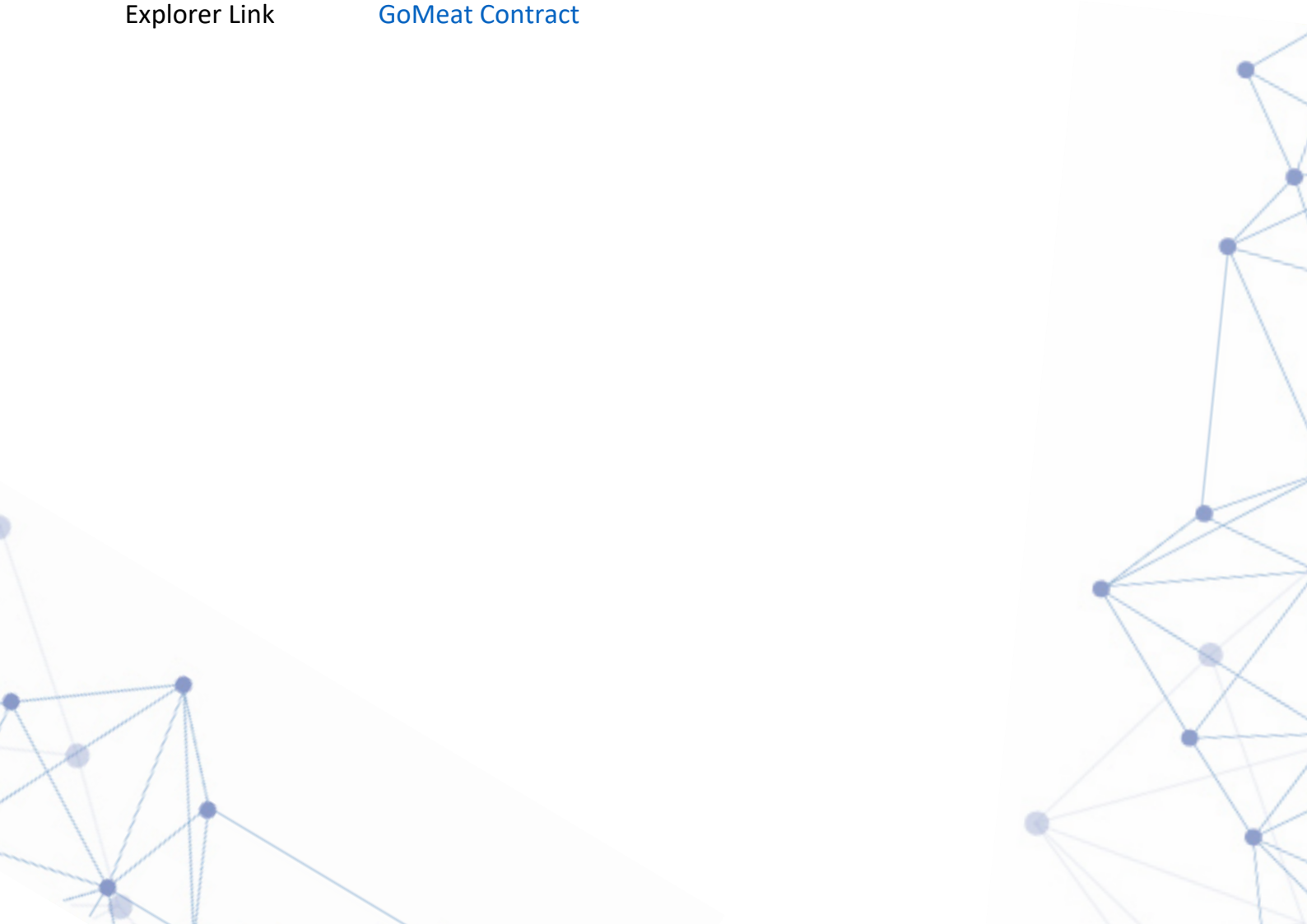
- <https://github.com/qtumproject/QRC20Token/blob/master/SafeMath.sol>

Commit Hash: 8797d4f5f0f706e87482d952dc4946a01d02882b

This audit report intends to expose any security vulnerabilities or design flaws in this contract that would hinder or otherwise affect its intended functionality. The information in this report is limited to the contents of the contract(s) as per the aforementioned commit and is meant to be used for informational purposes to understand the contract-level vulnerabilities in the existing code and to use its recommendations as guidelines to reduce the risk highlighted by this report.

# CONTRACT DETAILS

Contract name	QRC20Token
Contract address	7b8dd012ec9e1f11e008376ef21d29ba4cc78e24
Token name	GoMeat
Token ticker	GOMT
Decimals	8
Total Supply	5,000,000
Token holders	1
Transaction count	1
Contract owner	HBATXqt28YaQbSKdNNRoV2N4wfBuJ8fjtC
Contract deployer	HBATXqt28YaQbSKdNNRoV2N4wfBuJ8fjtC
Deployment block	156978
Explorer Link	<a href="#">GoMeat Contract</a>



# GoMeat TOP TOKEN HOLDERS

Wallet Names	GOMT Tokens 5M	% Of 5M	Hydra GOMT Token Public address
PRE-ICO Token (Private Sale)	400,000	8%	HPCgxxicdPe91TpS6BnEjXmGUs6g3DyN9R
ICO Token (Tier 1, 2, 3)	1,350,000	27%	HDKc65BQpvVKLrXBVKnk7skb2SHG8yWfKv
GM TEAM RESERVE FOR 3 YEARS VESTING	1,000,000	20%	HMjDTEVSTT3Tgw7fSWQ7Zqdru83p28YVbL
GoMeat Expansion Phases Tokens	1,250,000	25%	HJY6Yb75Myi1CMAGRNTibK7Es63NjPMsYK
Exchange Liquidity	250,000	5%	HHrnNPAs71omULDd5zG4mQbAL42DmDGEHE
Reserve for incentive Hydra & Staking	500,000	10%	H74jzp2dFr3599UjeUGoujGJS7jHCySmXs
Reward Management	250,000	5%	HVQkNPYMyQae7qHBirudcVS2YGK6eSobTz

# QRC20 TOKEN FUNCTION DETAILS

- **QRC20Token**

- QRC20Token() (public)
- approve(address,uint256) (public)
- fallback() (public)
- transfer(address,uint256) (public)
- transferFrom(address,address,uint256) (public)


- **SafeMath**

- SafeMath() (public)
- safeAdd(uint256,uint256) (internal)
- safeMul(uint256,uint256) (internal)
- safeSub(uint256,uint256) (internal)



# ISSUES CHECKING STATUS

1	Compiler errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Front running	Not applicable
4	Timestamp Dependence	Not applicable
5	Integer Overflow and Underflow	Passed
6	DoS with Revert	Not applicable
7	Design Logic	Medium issues
8	Assert Violation	Failed
9	Scoping and Declarations	Low
10	Arithmetic accuracy	Passed







# SECURITY ISSUES

## High Severity Issues:

No High severity issues found.

## Medium Severity Issues:

### 1. Assertion failure

Issue: `transfer(address,uint256)`, uses `SafeMath` to perform arithmetic operations, that uses `assert` statement to verify the inputs. Whereas the purpose of `assert` is to verify invariants<sup>1</sup>.

Recommendation: Replace **`assert`** with **`require`**.

## Low Severity Issues:

### 1. `transfer` function Incorrect visibility

Issues: The visibility of `transfer` function is `public`; that is only required if the contract needs to be accessed from both inside and outside the contract. This function is only accessed externally so the visibility should be changed.

Recommendation: Replace visibility from `Public` to `External`.

### 2. `Transfer from` function Incorrect visibility

Issues: The visibility of `transfer from` function is `public`; that is only required if the contract needs to be accessed from both inside and outside the contract. This function is only accessed externally so the visibility should be changed.

Recommendation: Replace visibility from `Public` to `External`.

### 3. `Approve` function Incorrect visibility

Issues: The visibility of `approve` function is `public`; that is only required if the contract needs to be accessed from both inside and outside the contract. This function is only accessed externally so the visibility should be changed.

---

<sup>1</sup> <https://github.com/OpenZeppelin/openzeppelin-contracts/issues/1120>

Recommendation: Replace visibility from Public to External.

#### 4. Fallback function Incorrect visibility

Issues: The visibility of fallback function is public; that is only required if the contract needs to be accessed from both inside and outside the contract. This function is only accessed externally so the visibility should be changed.

Recommendation: Replace visibility from Public to External.

#### 5. Total supply variable visibility

Issues: The total supply is a state variable that is never updated, so it should be constant.

Recommendations: Make totally supply as constant to save 21646 gas.

#### 6. Solidity version must be fixed

Issues: It should not be **pragma solidity ^0.4.18;**

Recommendations: It should be **pragma solidity 0.4.24;**

### Design Issues

The best practice is to always use the latest solidity compiler version to avoid any known security bugs. This contract uses version 0.4.18, which was released on Oct 18, 2017. The latest version supported by Hydra chain is 0.5.4 (as of writing this document). We strongly recommend using the latest version of solidity and updating the contracts as needed to support that version.





# CONCLUSION

QRC20Token, the smart contract under scrutiny, is a basic token contract which contains medium and low severity issues.



## RE-EVALUATION

After the code had been updated as per the recommendations in our original report, we re-evaluated the contract and present the following findings:

**Repository link for QRC20Token:**

- <https://github.com/huzaifa1100/QRC20Token-1/blob/master/QRC20Token.sol>

**Repository link for SafeMath:**

- <https://github.com/huzaifa1100/QRC20Token-1/blob/master/SafeMath.sol>

Commit Hash: 0fd3847474a1f4a1e171da6c238d6079c6733c23

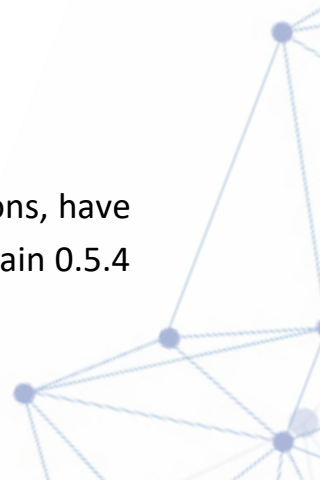


## ISSUES CHECKING STATUS

1	Compiler errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Front running	Not applicable
4	Timestamp Dependence	Not applicable
5	Integer Overflow and Underflow	Passed
6	DoS with Revert	Not applicable
7	Design Logic	Medium issues
8	Assert Violation	Passed
9	Scoping and Declarations	Passed
10	Arithmetic accuracy	Passed

## REMARKS

Most of the low-level issues, like fixing the access modifiers for functions, have been fixed. However, the latest solidity version supported by Hydra chain 0.5.4



(as of writing this document) has not been implemented. We also raised concerns previously over non-compliance with the universally implemented standard of ERC20 that will lead to third-party integration issues.

Updated – September 10, 2021:

September 10, 2021, “GoMeat Top Token Holders” address is updated, previously “HBATXqt28YaQbSKdNNRoV2N4wfBuJ8fjtC” holds the 5M supply, now the supply is distributed among 7 wallets, mentioned in “GoMeat Top Token Holders”

