BULLS PROTOCOL

Smart Contract security and compliance assessments

Mastery Of Monsters

BullsProtocol.com

July 23, 2023

INTRODUCTION

This audit has resulted from Smart Contract analysis and testing against common and uncommon vectors used in code attacks, analyzes from a cyber vulnerabilities perspective, as well as whether the code structure follows best practices in industry standards.

The works are developed through some methodologies, among them the revision of all lines of the source code.

Reviews result in ratings ranging from critical to approved. A rating is given as **CRITICAL** if it allows for potential exploitation of the problem by attackers, if the bug's manifestation causes financial loss or if it interrupts the execution of the code.

After the **CRITICAL** classification we also have the **HIGH**, **MEDIUM** and **LOW** classifications.

LOW severity ratings indicate points that do not compromise the code or its functioning.

The next classification would be the **INFORMATIVE**, these classifications are just simple indications and comments about the source code.

And finally, the **APPROVED**, which are all those that passed all analyzes and do not fit in the previous cases.

The indications of **NOT FOUND** refer to situations not found in the contract, which also mean that they have the status of **APPROVED**.

ÍTALO HONORATO
BLOCKCHAIN AUDITOR

OVERVIEW

The contract has only one file, **MasteryOfMonsters.sol**, which has a source code with 924 lines. We verified that all functions and variables were used and commented in accordance with the Ethereum EPI documentation and standards, the OpenZeppelin standard libraries and also in accordance with other technical standards.

Project name	Mastery Of Monsters
Source code	https://bscscan.com/address/0xa7d9d54a98adc16e6aa980e845c740053b5b7771#code
Contract address	0xA7d9D54A98aDC16E6aA980e845C740053b5B7771
ByteCode for Ethereum Virtual Machine	https://bscscan.com/bytecode-decompiler?a=0xa7d9d54a98adc16e6aa980e845c740053b5b7771
Contract fees	8.88% buy, 8.88% sell and 8.88% transfer
Website	https://masteryofmonsters.com
Telegram	https://t.me/MasteryOfMonsters

CHECKING STATUS

Table of audit analysis results:

Code accuracy according to expectations	APPROVED
Errors when compiling	APPROVED
Timestamp dependency	APPROVED
Code testability	APPROVED
Clean/lean code	APPROVED
DDOS attack by gas limit per block	APPROVED
DDOS attack due to failed function calls	APPROVED
DDOS attack by reversing transaction processing	APPROVED
Use of outdated functions in Solidity	APPROVED
Reentry attack/reentrancy attack	APPROVED
Poorly documented or unprotected SELFDESTRUCT statements	APPROVED
Not checking return value in Get functions	APPROVED
Integer overflow/underflow	APPROVED
Presence of unused variables	APPROVED
Code redundancy	APPROVED
Gas optimization in transactions	APPROVED
Optimization of contract execution	APPROVED
Prevention of loss of values within the contract	APPROVED
Violation of REQUIRE FUCTION	APPROVED
Typographical errors	APPROVED
Unsecured BNB withdrawal	APPROVED
Logical design	APPROVED
Arithmetic precision	APPROVED
Fallback function security	APPROVED
Compliance with EIP standards	APPROVED
Disclaimer of ownership	APPROVED
Implementation of standard and secure OpenZeppelin libraries	APPROVED

PRIVILEGED FUNCTIONS

The contract has functions that can only be modified by the owner of the contract, owner defined by the ***onlyOwner*** or ***OnlyAuth*** modifier. The following functions deserve attention:

- ***whiteList***, function to distribute wallet tokens to multiple addresses
- ***updateUniswapV2Router***, changes the Pancake V2 route address;
- ***except***, exempt addresses from fees, or place addresses on the fee collection list
- ***setMappingAuth***, includes addresses in an authorized list to call the ***transferAuthProject*** function;
- ***setBuy***, changes purchase rates;
- ***setSell***, changes the sales rates;
- ***setTransferFees***, changes transfer fees;
- ***setProjectWallets***, changes the address of the project's wallets;
- ***setMax***, changes the max buy and max wallet limits;
- ***transferAuthProject***, token transfer function for exclusive use of addresses in the authorization list;

COMMENTS IN THE CODE

Codes are punctually commented when more complex functions or mechanisms are discussed. Below are two examples of developer clarifications. In the first image there is a justification about the unnecessary need to make duplicate checks of mathematical calculations, with gas savings being one of the reasons. In the second, he comments on the ***transferAuthProject*** function, specially designed to send tokens with fee savings:

```
function swapTokens() internal {

    //Instruction unchecked helps to avoid gas expenses
    unchecked {

        _approve(address(this), address(addressPCVS2), triggerSwapTokensToUSD);

        uint256 _totalFees = totalFees + totalTransferFees;
        uint256 _totalFeesLiquidity = buy.liquidity + sell.liquidity + transferFees.liquidity;
        uint256 _fessToUsdt = _totalFees - _totalFeesLiquidity;

        //totalFees is greater than or equal to (buy.liquidity + sell.liquidity)
        //totalTransferFees is greater than or equal to (transferFees.liquidity)
        //So _totalFees >= _totalFeesLiquidity
        //So (_totalFees - _totalFeesLiquidity) >= 0
        //Never revert by errors
        uint256 tokensToSelltoUSDT = (_fessToUsdt * triggerSwapTokensToUSD) / _totalFees;
        uint256 tokensToSelltoLiquidity = (_totalFeesLiquidity * triggerSwapTokensToUSD) / _totalFees;
```

```
//Special function to be used by the game's backend
//Organized to avoid gas costs
function transferAuthProject(address to,uint256 amount) external onlyAuth() {
    require(amount > 0 && amount <= totalSupply() , "Invalid amount transferred");

    require(_balances[_msgSender()] >= amount, "ERC20: transfer amount exceeds balance");

    //Never overflow
    unchecked {

        _balances[_msgSender()] -= amount;
        _balances[to] += (amount);

    }

    emit Transfer(_msgSender(), to, amount);

}
```

RUG-PULL FUNCTIONS

There are no functions or mechanisms in the contract that can be used by the owner to perform rug-pulls. No harmful logic was found as in the examples below:

- deliberate token creation schemes (***mint*** function). There are no direct or hidden features for deliberate token creation in the contract. Tokens were created once on deploy

- functions that would deliberately change the fees to a high value (***setFees*** function). Rate change functions are limited to 10% of the value of each rate component

- functions to lock trades (***lock swap*** function). The max functions are limited to ranges. It is not possible to set the max wallet value to zero to revert to a token purchase..

- contract ***self-destruct*** mechanisms (***selfdestruct*** function) to make tracing of transactions more difficult. No contract self-destruction mechanisms were found

- the ***transferAuthProject*** function only transfers tokens from an authorized wallet, without creating tokens or without withdrawing tokens from another wallet other than the one sending the transaction

CONCLUSION

The audit performed manual line-by-line analysis and automated review of the smart contract. The contract and source code were analyzed mainly from the point of view of common vulnerabilities, exploits, manipulation hacks and also from the point of view of code structure optimizations.

The contract and source code have an **APPROVED** status in the audit reviews and discussions.

No issues were found that fall under the status of **CRITICAL**, **HIGH**, **MEDIUM**, or **LOW**.

LEGAL NOTICE

BULLS PROTOCOL and its technical team provide through this audit report only discussion and evaluation about project and contract, with the sole intention of helping to improve the quality of the code and to improve the level of variation used in smart contract technologies. Therefore, this audit report is not a legally binding document.

This technical audit report is not a buy or sell recommendation, approval or disapproval, or even a definitive answer on the points discussed here, but only expresses the technical and analytical opinion on the **Mastery Of Monsters** project.

The reproduction of this material is free and there is no need to request permission from the creators, but the source must be cited.