Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

December, 2021

Audit Details

Audited project

Meta Hangry Games

Deployer address

0xa21edef9d421bc68caf8246651c62886da21b20b

Client contacts:

Meta Hangry Games team

Blockchain

Binance Smart Chain

Project website:

Not provided by Meta Hangry Games team

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (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.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by Meta Hangry Games to perform an audit of smart contracts:

https://bscscan.com/address/0x3025e70f87072e0442767b5cc9cb1f4cdd314bc3#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

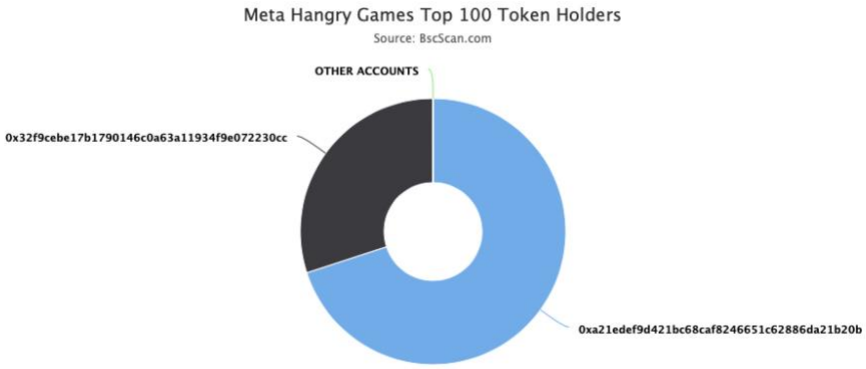
Token contract details for 11.12.2021

Contract name	Meta Hangry Games
Contract address	0x3025E70f87072E0442767b5Cc9Cb1f4cDD314Bc3
Total supply	1,000,000,000,000
Token ticker	MHG
Decimals	4
Token holders	2
Transactions count	2
Top 100 holders dominance	100.00%
Liquidity fee	100
Reflection fee	500
Team fee	700
Uniswap V2 pair	0xcd0eafb1dbc3174ad6fc991d47e38a1bd64564d6
Contract deployer address	0xa21edef9d421bc68caf8246651c62886da21b20b
Contract's current owner address	0xa21edef9d421bc68caf8246651c62886da21b20b

Meta Hangry Games Token Distribution

The top 100 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Meta Hangry Games

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 2

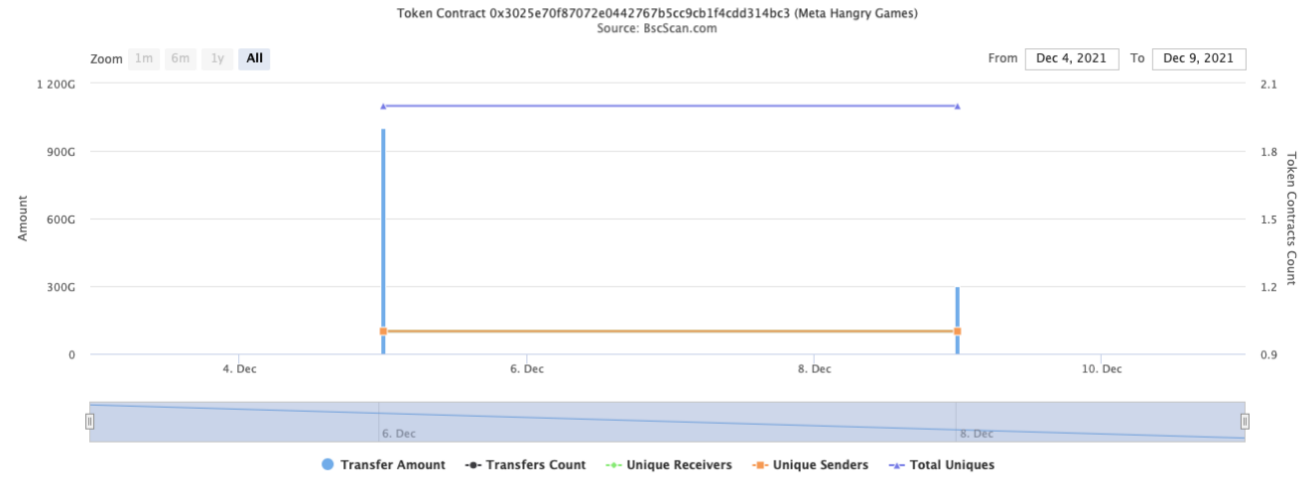


(A total of 1,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

Meta Hangry Games Contract Interaction Details

Time Series: Token Contract Overview

Sun 5, Dec 2021 - Thu 9, Dec 2021



Meta Hangry Games Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0xa21edef9d421bc68caf8246651c62886da21b20b	700,000,000,000	70.0000%
2	0x32f9cebe17b1790146c0a63a11934f9e072230cc	300,000,000,000	30.0000%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Int] functionStaticCall
  - [Int] functionStaticCall
  - [Int] functionDelegateCall #
  - [Int] functionDelegateCall #
  - [Prv] _verifyCallResult

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+ [Int] IUniswapV2Factory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transfer #
  - [Ext] transferFrom #

- [Ext] DOMAIN_SEPARATOR
- [Ext] PERMIT_TYPEHASH
- [Ext] nonces
- [Ext] permit #
- [Ext] MINIMUM_LIQUIDITY
- [Ext] factory
- [Ext] token0
- [Ext] token1
- [Ext] getReserves
- [Ext] price0CumulativeLast
- [Ext] price1CumulativeLast
- [Ext] kLast
- [Ext] mint #
- [Ext] burn #
- [Ext] swap #
- [Ext] skim #
- [Ext] sync #
- [Ext] initialize #

+ [Int] IUniswapV2Router01
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity #
- [Ext] addLiquidityETH ($)
- [Ext] removeLiquidity #
- [Ext] removeLiquidityETH #
- [Ext] removeLiquidityWithPermit #
- [Ext] removeLiquidityETHWithPermit #
- [Ext] swapExactTokensForTokens #
- [Ext] swapTokensForExactTokens #
- [Ext] swapExactETHForTokens ($)
- [Ext] swapTokensForExactETH #
- [Ext] swapExactTokensForETH #
- [Ext] swapETHForExactTokens ($)
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ MHG (Context, IERC20, Ownable)
- [Pub] <Constructor> #
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer #

- [Pub] allowance
- [Pub] approve #
- [Pub] transferFrom #
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Pub] totalFees
- [Pub] isExcludedFromFee
- [Pub] ___tokenInfo
- [Pub] ___feesInfo
- [Pub] ___wallets
- [Ext] Change_Wallet_Marketing #
  - modifiers: onlyOwner
- [Ext] Change_Wallet_Presale #
  - modifiers: onlyOwner
- [Ext] Change_Wallet_Team #
  - modifiers: onlyOwner
- [Ext] Change_Wallet_Burn #
  - modifiers: onlyOwner
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Pub] isExcludedFromReward
- [Pub] tradingStatus #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidityt #
  - modifiers: onlyOwner
- [Ext] setMaxTxPercent #
  - modifiers: onlyOwner
- [Ext] setMaxTxTokens #
  - modifiers: onlyOwner
- [Ext] setMaxWalletPercent #
  - modifiers: onlyOwner
- [Ext] setMaxWalletTokens #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] s_manageBlacklist #
  - modifiers: onlyOwner
- [Ext] s_excludeFromFee #
  - modifiers: onlyOwner
- [Ext] multitransfer #
  - modifiers: onlyOwner
- [Pub] convertLiquidityBalance #
  - modifiers: onlyOwner
- [Pub] purgeContractBalance #
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _getValues
- [Prv] _fees_to_bnb_process #
  - modifiers: lockTheSwap
- [Ext] fees_to_bnb_manual #

    - modifiers: onlyOwner
  - [Prv] _fees_to_bnb #
  - [Prv] _takeFee #
  - [Prv] _setAllFees #
  - [Ext] set_sell_multiplier #
    - modifiers: onlyOwner
  - [Ext] set_All_Fees_Triggers #
    - modifiers: onlyOwner
  - [Ext] set_All_Fees_Minimum_Balance #
    - modifiers: onlyOwner
  - [Ext] set_All_Fees #
    - modifiers: onlyOwner
  - [Prv] removeAllFee #
  - [Prv] restoreAllFee #
  - [Ext] burn_tokens_reduce_supply #
  - [Ext] burn_tokens_to_dead #
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] swapTokensForEthAndSend #
  - [Prv] addLiquidity #
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] _transferStandard #
  - [Ext] <Fallback> ($)

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function includeInReward(address account⬆) external onlyOwner {
    require(_isExcluded[account⬆], "Account is already included");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account⬆) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _balance_total[account⬆] = 0;
            _isExcluded[account⬆] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _supply_reflected;
    uint256 tSupply = _supply_total;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _balance_reflected[_excluded[i]] > rSupply ||
            _balance_total[_excluded[i]] > tSupply
        ) return (_supply_reflected, _supply_total);
        rSupply = rSupply - _balance_reflected[_excluded[i]];
        tSupply = tSupply - _balance_total[_excluded[i]];
    }
    if (rSupply < (_supply_reflected / _supply_total))
        return (_supply_reflected, _supply_total);
    return (rSupply, tSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

- The function s_manageBlacklist() uses the loop for blacklist addresses. It also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.

```
function s_manageBlacklist(address[] calldata addresses⬆, bool status⬆)
    external
    onlyOwner
{
    for (uint256 i; i < addresses⬆.length; ++i) {
        _isBlacklisted[addresses⬆[i]] = status⬆;
    }
}
```

- The function s_excludeFromFee() uses the loop for excluding addresses from the fee. It also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.

```
function s_excludeFromFee(address[] calldata addresses⬆, bool status⬆)
    external
    onlyOwner
{
    for (uint256 i; i < addresses⬆.length; ++i) {
        _isExcludedFromFee[addresses⬆[i]] = status⬆;
    }
}
```

- The function multitransfer() uses the loop for multitransfer tokens to addresses. It also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.

Recommendation:
Check that the array length is not too big.

Notes:

- _transfer function has required _fees_to_bnb with static values calling.

Owner privileges (In the period when the owner is not renounced)

- Owner can change marketing, preseller, team and burn wallets.
- Owner can change trading status.
- Owner can change number of tokens to sell to add to liquidity.
- Owner can change maximum transaction amount and maximum wallet token.
- Owner can manually swap and liquify.
- Owner can manually call _fees_to_bnb function with wrapper.
- Owner can change sell multiplier.
- Owner can change fees, fee limits and fee minimum balances.

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details NOT provided by the team.

TechRate note:

Please 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.