June, 2021

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

Armageddon

Deployer address

0x6EDAf3e626cd4f5fd2317104485ddDE828622Dc5

Client contacts:

Armageddon team

Blockchain

Binance Smart Chain

Project website:

https://www.armageddontoken.com

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 Armageddon to perform an audit of smart contracts:
https://bscscan.com/address/0xd6bb3fc62083d369df178ea017c2e1c9f27d7b58#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.
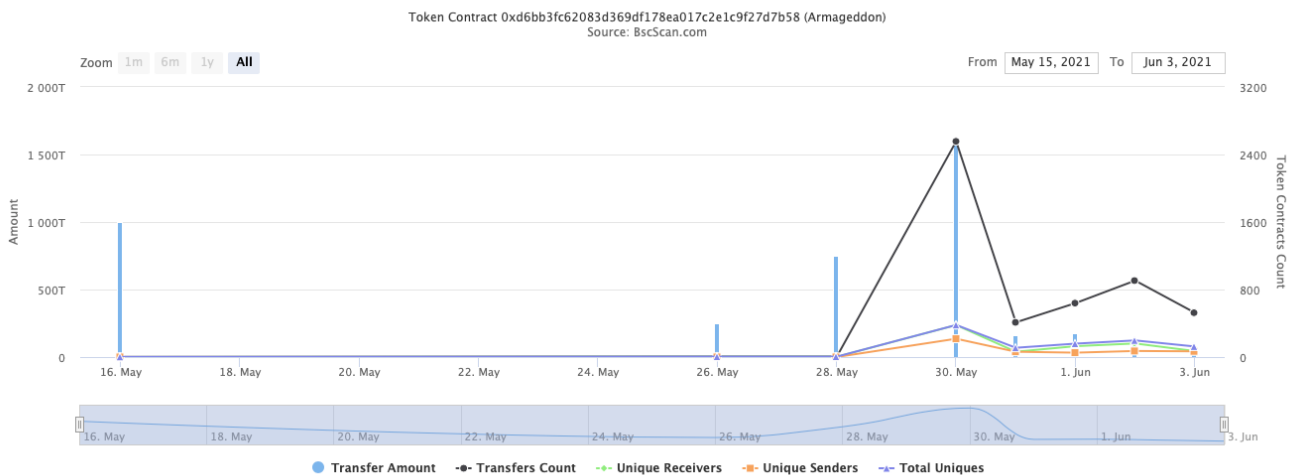
Contracts Details

Token contract details for 05.06.2021

Contract name	Armageddon
Contract address	0xD6bb3fc62083d369Df178ea017c2e1C9f27d7B58
Total supply	1,000,000,000,000,000
Token ticker	AMGD
Decimals	9
Token holders	684
Transactions count	5,356
Top 100 holders dominance	91.32%
Liquidity fee	4
Tax fee	4
Total fees	56770469765758348422979
Uniswap V2 pair	0xdb4400aaea1eed38d5ae747dc2ddb8b6b506c7c6
Contract deployer address	0x6EDAf3e626cd4f5fd2317104485ddDE828622Dc5
Contract's current owner address	0x6edaf3e626cd4f5fd2317104485ddde828622dc5

Armageddon Token Distribution

Armageddon Top 100 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

0x0000000000000000000000000000000000000dead

0xa803fc1c1e83d6389865e1248dc924ed4c6953de
0x6edaf3e626cd4f5fd2317104485ddde828622dc5
0x0000000000000000000000000000000000000000
0x7758ae53336a8b60db3bc3d2be2d476a8fff6f9e
0xaa55a16e7b3fd0dab5d795a3215798e622ae6dea
0x3fb5df2da721780484b0d578f3790b130ffd9cf6
0x4f5e06221e8642a793c360204dd16866b4bd74d7
0x2a653657157551a719316d2e4e059035743ae4d2

0xdb4400aaea1eed38d5ae747dc2ddb8b6b506c7c6
(PancakeSwap V2: AMGD 3)

(A total of 913,215,341,979,952.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Armageddon Contract Interaction Details

Time Series: Token Contract Overview   Sun 16, May 2021 - Thu 3, Jun 2021

Token Contract 0xd6bb3fc62083d369df178ea017c2e1c9f27d7b58 (Armageddon)
Source: BscScan.com

Zoom 1m 6m 1y **All**   From May 15, 2021   To   Jun 3, 2021

- Transfer Amount   - Transfers Count   - Unique Receivers   - Unique Senders   - Total Uniques

Armageddon Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x000000000000000000000000000000000000dead	402,007,089,387,207.707987262	40.2007%
2	⬚ PancakeSwap V2: AMGD 3	137,636,694,230,303.088981779	13.7637%
3	0x2a653657157551a719316d2e4e059035743ae4d2	48,136,459,677,312.058686191	4.8136%
4	0x4f5e06221e8642a793c360204dd16866b4bd74d7	38,945,762,832,753.223103515	3.8946%
5	0x3fb5df2da721780484b0d578f3790b130ffd9cf6	30,843,240,089,192.725571868	3.0843%
6	0xaa55a16e7b3fd0dab5d795a3215798e622ae6dea	21,379,441,986,967.587863432	2.1379%
7	0x7758ae53336a8b60db3bc3d2be2d476a8fff6f9e	18,257,077,766,027.206451408	1.8257%
8	⬚ 0x0000000000000000000000000000000000000000	14,899,617,418,738.705184585	1.4900%
9	0x6edaf3e626cd4f5fd2317104485ddde828622dc5	12,611,267,434,510.597831226	1.2611%
10	0xa803fc1c1e83d6389865e1248dc924ed4c6953de	11,398,998,207,969.353157292	1.1399%

Contract functions details

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

+ **[Lib]** SafeMath
  - **[Int]** add
  - **[Int]** sub
  - **[Int]** sub
  - **[Int]** mul
  - **[Int]** div
  - **[Int]** div
  - **[Int]** mod
  - **[Int]** mod

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

+ **[Lib]** Address
  - **[Int]** isContract
  - **[Int]** sendValue **#**
  - **[Int]** functionCall **#**
  - **[Int]** functionCall **#**
  - **[Int]** functionCallWithValue **#**
  - **[Int]** functionCallWithValue **#**
  - **[Prv]** _functionCallWithValue **#**

+ Ownable (Context)
  - **[Int]** <Constructor> **#**
  - **[Pub]** owner
  - **[Pub]** renounceOwnership **#**
    - modifiers: onlyOwner
  - **[Pub]** transferOwnership **#**
    - modifiers: onlyOwner
  - **[Pub]** geUnlockTime
  - **[Pub]** lock **#**
    - modifiers: onlyOwner
  - **[Pub]** unlock **#**

+ **[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 **#**

+ Armageddon (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]** isExcludedFromReward
- **[Pub]** totalFees
- **[Pub]** deliver **#**
- **[Pub]** reflectionFromToken
- **[Pub]** tokenFromReflection
- **[Pub]** excludeFromReward **#**
  - modifiers: onlyOwner
- **[Ext]** includeInReward **#**
  - modifiers: onlyOwner
- **[Prv]** _transferBothExcluded **#**
- **[Ext]** **<Fallback>** **($)**
- **[Prv]** _reflectFee **#**
- **[Prv]** _getValues
- **[Prv]** _getTValues
- **[Prv]** _getRValues
- **[Prv]** _getRate
- **[Prv]** _getCurrentSupply
- **[Prv]** _takeLiquidity **#**
- **[Prv]** calculateTaxFee
- **[Prv]** calculateLiquidityFee
- **[Prv]** removeAllFee **#**
- **[Prv]** restoreAllFee **#**
- **[Pub]** isExcludedFromFee
- **[Prv]** _approve **#**
- **[Prv]** _transfer **#**
- **[Prv]** swapAndLiquify **#**
  - modifiers: lockTheSwap
- **[Prv]** swapTokensForEth **#**
- **[Prv]** addLiquidity **#**
- **[Prv]** _tokenTransfer **#**
- **[Prv]** _transferStandard **#**
- **[Prv]** _transferToExcluded **#**
- **[Prv]** _transferFromExcluded **#**
- **[Pub]** excludeFromFee **#**
  - modifiers: onlyOwner
- **[Pub]** includeInFee **#**
  - modifiers: onlyOwner
- **[Ext]** enableAllFees **#**

- modifiers: onlyOwner
- **[Ext]** disableAllFees **#**
    - modifiers: onlyOwner
- **[Ext]** setmarketingWallet **#**
    - modifiers: onlyOwner
- **[Ext]** setMaxTxPercent **#**
    - modifiers: onlyOwner
- **[Pub]** setSwapAndLiquifyEnabled **#**
    - modifiers: onlyOwner

($) = 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.

```solidity
function includeInReward(address account↑) external onlyOwner() {
    require(_isExcluded[account↑], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account↑) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[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.

```solidity
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

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

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

- **Owner can disable and enable all fees.**

```
//Call this function after finalizing the presale
function enableAllFees() external onlyOwner() {
    _taxFee = 4;
    _previousTaxFee = _taxFee;
    _liquidityFee = 4;
    _previousLiquidityFee = _liquidityFee;
    _burnFee = 1;
    _previousBurnFee = _taxFee;
    _marketingFee = 1;
    _previousmarketingFee = _marketingFee;
    inSwapAndLiquify = true;
    emit SwapAndLiquifyEnabledUpdated(true);
}


function disableAllFees() external onlyOwner() {
    _taxFee = 0;
    _previousTaxFee = _taxFee;
    _liquidityFee = 0;
    _previousLiquidityFee = _liquidityFee;
    _burnFee = 0;
    _previousBurnFee = _taxFee;
    _marketingFee = 0;
    _previousmarketingFee = _marketingFee;
    inSwapAndLiquify = false;
    emit SwapAndLiquifyEnabledUpdated(false);
}
```

- **Owner can change marketing address.**

```
function setmarketingWallet(address newWallet) external onlyOwner() {
    marketingWallet = newWallet;
}
```

- **Owner can change the maximum transaction amount.**

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

- **Owner can exclude from the fee.**

```solidity
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}
```

- **Owner can lock and unlock. By the way, using these functions the owner could leave as owner even after the ownership was renounced.**

```solidity
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

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

Liquidity locking details provided by the team:
https://dxsale.app/app/pages/defipresale?saleID=2234&chain=BSC

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.

Techrate1   Techrate   Techrate_audits