Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

August, 2021

Audit Details

Audited project

Art of War

Deployer address

0x7d9c0d1262ea97374bf72968442cc83fd4798c2d

Client contacts:

Art of War team

Blockchain

Binance Smart Chain

Project website:

https://artofwar.land

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 Art of War to perform an audit of smart contracts:

https://bscscan.com/address/0xdd5efe248322d6bd3b1954e1c6ddb1d18b1a96e8#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

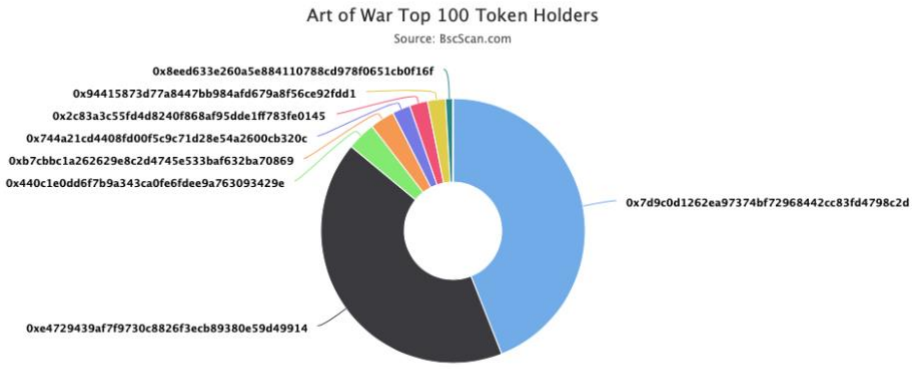
Token contract details for 27.08.2021

Contract name	Art of War
Contract address	0xDd5eFe248322D6BD3b1954E1c6Ddb1d18B1A96e8
Total supply	10,000,000,000
Token ticker	$AOW
Decimals	8
Token holders	8
Transactions count	8
Top 100 holders dominance	100.00%
Tax/Burn/Marketing&Team	2/2/2
Total fees	0
Contract deployer address	0x7d9c0d1262ea97374bf72968442cc83fd4798c2d
Contract's current owner address	0x0000000000000000000000000000000000000000

Art of War Token Distribution

The top 100 holders collectively own 100.00% (10,000,000,000.00 Tokens) of Art of War

Token Total Supply: 10,000,000,000.00 Token | Total Token Holders: 8

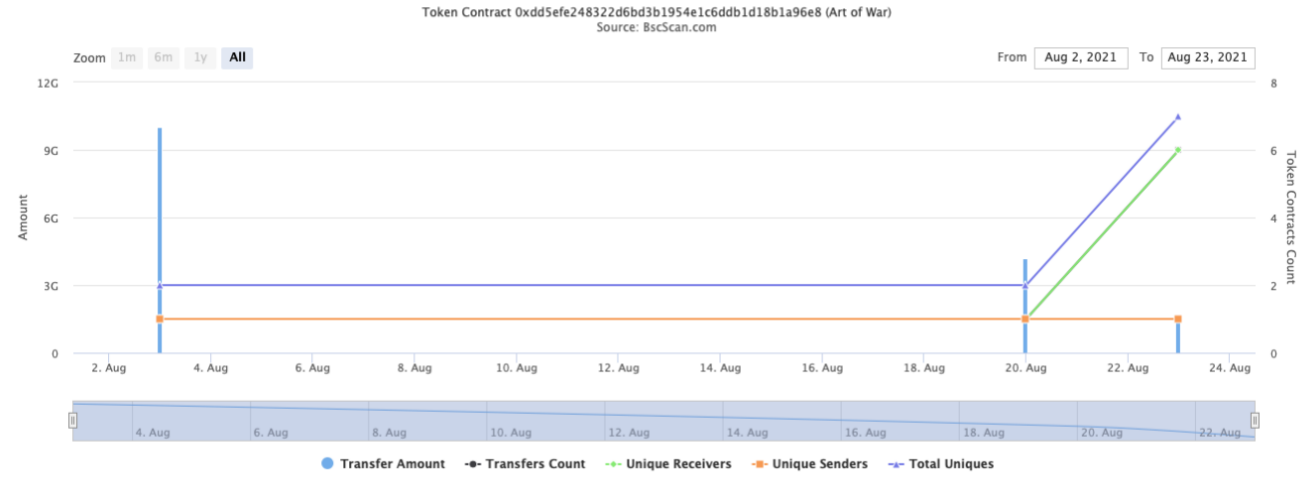


(A total of 10,000,000,000.00 tokens held by the top 100 accounts from the total supply of 10,000,000,000.00 token)

Art of War Contract Interaction Details

Time Series: Token Contract Overview

Tue 3, Aug 2021 - Mon 23, Aug 2021



Art of War Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x7d9c0d1262ea97374bf72968442cc83fd4798c2d	4,400,000,000	44.0000%
2	0xe4729439af7f9730c8826f3ecb89380e59d49914	4,200,000,000	42.0000%
3	0x440c1e0dd6f7b9a343ca0fe6fdee9a763093429e	350,000,000	3.5000%
4	0xb7cbbc1a262629e8c2d4745e533baf632ba70869	300,000,000	3.0000%
5	0x744a21cd4408fd00f5c9c71d28e54a2600cb320c	220,000,000	2.2000%
6	0x2c83a3c55fd4d8240f868af95dde1ff783fe0145	220,000,000	2.2000%
7	0x94415873d77a8447bb984afd679a8f56ce92fdd1	220,000,000	2.2000%
8	0x8eed633e260a5e884110788cd978f0651cb0f16f	90,000,000	0.9000%

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)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] team
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Pub] transferTeam #
    - modifiers: onlyTeam

+ ArtOfWar (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] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] calculateTaxFee
- [Prv] calculateBurnFee
- [Prv] calculateMarketingAndTeamFee
- [Prv] calculateGameFundFee
- [Pub] isExcludedFromFee
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Pub] getBalanceAddress ($)
  - modifiers: onlyTeam
- [Pub] transferOtherToken ($)
  - modifiers: onlyTeam
- [Pub] getBalanceOfToken
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] getBalance
- [Pub] setMarketingAndTeamAddress #
  - modifiers: onlyTeam
- [Pub] setGameFundAddress #
  - modifiers: onlyTeam
- [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.

```
ftrace | funcSig
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];
            _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.

```
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)

- Team address can withdraw ERC20 tokens.

```
ftrace | funcSig
function transferOtherToken(address _token⬆, address _owner⬆, uint _amount⬆) public payable onlyTeam {
    IERC20(_token⬆).transfer(_owner⬆, _amount⬆);
}
```

- Team address can withdraw BNB balance.

```
ftrace | funcSig
function getBalanceAddress(address payable _to⬆) public payable onlyTeam {
    // Call returns a boolean value indicating success or failure.
    // This is the current recommended method to use.
    (bool sent, ) = _to⬆.call{value: address(this).balance}("");
    require(sent, "Failed to send Ether");
}
```

- Team address can change marketing&team and gameFund addresses.

```
ftrace | funcSig
function setMarketingAndTeamAddress(address _market_team_address⬆) public onlyTeam {
    _marketingAndTeam = _market_team_address⬆;
}

ftrace | funcSig
function setGameFundAddress(address _game_fund_address⬆) public onlyTeam {
    _gameFundAddress = _game_fund_address⬆;
}
```

- Owner can exclude from the fee.

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

Conclusion

Smart contracts contain low severity issues!

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.