Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

Alpha Wolf

Deployer address

0xaAdF69389faD9319a1CccC7B1B6CAa5cDE186EC9

Client contacts:

Alpha Wolf team

Blockchain

Ethereum

Project website:

https://www.alphawolf.finance

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 Alpha Wolf to perform an audit of smart contracts:

https://etherscan.io/token/0x0342ac5dfad866985dde477caa85027a3c01a334

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	Alpha Wolf
Contract address	0x0342ac5dfaD866985Dde477caa85027A3C01a334
Total supply	992,250,320,845,354.7176
Token ticker	AWF
Decimals	18
Token holders	3,510
Transactions count	10,056
Top 100 holders dominance	99.13%
Total fees	774968712967609313897693487052 8
Contract deployer address	0xaAdF69389faD9319a1CccC7B1B6CAa5cDE186EC9
Contract’s current owner address	0xaadf69389fad9319a1cccc7b1b6caa5cde186ec9

Alpha Wolf Token Distribution

The top 100 holders collectively own 99.13% (983,578,985,939,544.00 Tokens) of Alpha Wolf

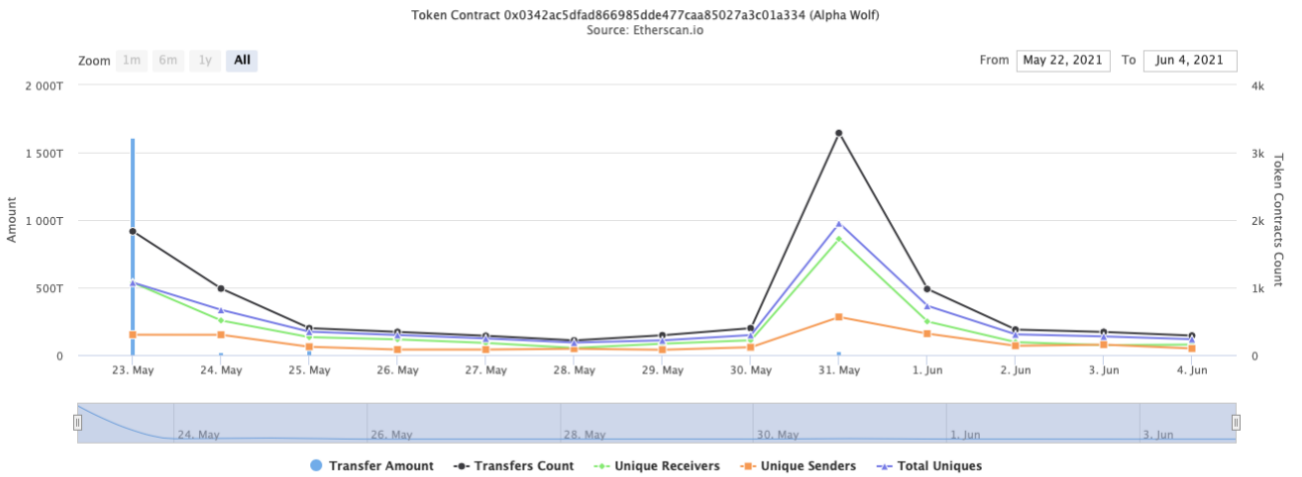
Token Total Supply: 992,250,312,870,323.91 Token | Total Token Holders: 3,512

(A total of 983,578,985,939,544.00 tokens held by the top 100 accounts from the total supply of 992,250,312,870,323.91 token)

Alpha Wolf Contract Interaction Details

Time Series: Token Contract Overview

Sun 23, May 2021 - Fri 4, Jun 2021



Alpha Wolf Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Alpha Wolf: Deployer	450,833,163,704,968.080237746905228822	45.4354%
2	Burn Address	386,026,375,626,954.131748721586491153	38.9041%
3	0x7a920d41bb3f0040d21f47c7f435ab8ac9743048	10,863,234,211,367.13545883472779094	1.0948%
4	0x3078f86478e5e70f1a61aedd086c84f8d3328c17	10,862,960,615,690.612565779839909266	1.0948%
5	0xfdd18716afab41acdfeb490d37bb1a571382a37e	10,860,626,243,096.458416504197117658	1.0945%
6	0xc0aaab10c1e01c867f639e4e86d4323ffa8cd5d6	10,859,330,596,668.405589159715788521	1.0944%
7	0x47db40d4b1564ffff5c05494ffcc9d3ee10c1923	10,858,254,137,746.822970570018134325	1.0943%
8	0xb9e75ff6465d19dca69458d643828e5739b05330	10,856,624,050,967.797891241430981175	1.0941%
9	0x4fe7cf7f736f27b3e96707b51cbe88be053f54f6	10,852,812,870,097.476265251977582308	1.0938%
10	0x2fb8a177abcb1ba25165f8c0512bc7ae422ecc99	10,849,567,780,109.784836028051129083	1.0934%

Contract functions details

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

+ [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

+ [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

+ AlphaWolf (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] isExcluded
- [Pub] totalFees
- [Pub] totalBurn
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Ext] excludeAccount #
  - modifiers: onlyOwner
- [Ext] includeAccount #
  - modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _getTaxFee
- [Prv] _getMaxTxAmount
- [Ext] _setTaxFee #
  - modifiers: onlyOwner
- [Ext] _setMaxTxAmount #
  - 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 includeAccount() 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 includeAccount(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is not 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.

```
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.

2. Known vulnerabilities of BEP-20 token

Issue:

- Lack of transaction handling mechanism issue. More details here.

Recommendation:

Add the following code to the `_transfer(address sender, ...)` function:

```
require( recipient != address(this) );
```

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

- Owner can change tax fee.

```
function _setTaxFee(uint256 taxFee⬆) external onlyOwner() {
    require(taxFee⬆ >= 1 && taxFee⬆ <= 10, 'taxFee should be in 1 - 10');
    _taxFee = taxFee⬆;
}
```

- Owner can change the maximum transaction amount.

```
function _setMaxTxAmount(uint256 maxTxAmount⬆) external onlyOwner() {
    require(maxTxAmount⬆ >= 1000000000000000e18 , 'maxTxAmount should be greater than 1000000000000000e18');
    _maxTxAmount = maxTxAmount⬆;
}
```

Notes:

- `_getTaxFee` is private, so nobody can know current tax fee.

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.