Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

WolfGirl

Deployer address

0xce8B73df4234e62f1ffAA2f7CCdc341c54A9306A

Client contacts:

WolfGirl team

Blockchain

Binance Smart Chain

Project website:

www.thewolfgirl.io

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 WolfGirl to perform an audit of smart contracts:

https://bscscan.com/address/0x7a3e66dad59b99404dc28d48767b7528301318f6#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

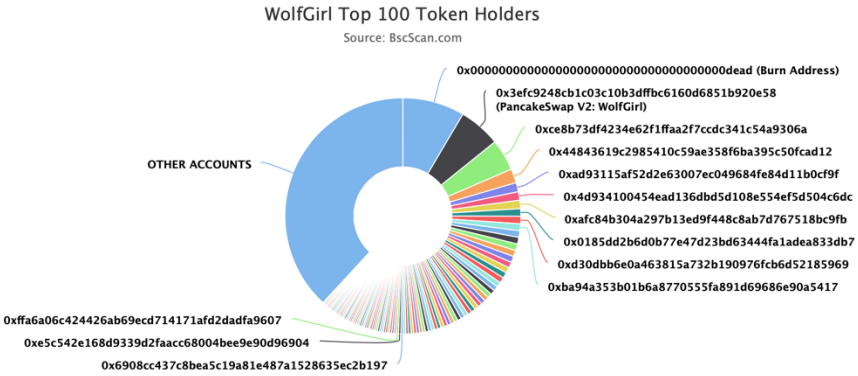
Token contract details for 09.11.2021

Contract name	WolfGirl
Contract address	0x7A3e66DAD59b99404dC28D48767B7528301318F6
Total supply	100,000,000,000,000
Token ticker	WolfGirl
Decimals	18
Token holders	2,096
Transactions count	12,232
Top 100 holders dominance	61.86%
Buy fee	10
Sell fee	10
Uniswap V2 pair	0x3EFc9248Cb1c03C10b3Dffbc6160d6851B920e58
Contract deployer address	0xce8B73df4234e62f1ffAA2f7CCdc341c54A9306A
Contract's current owner address	0xce8B73df4234e62f1ffAA2f7CCdc341c54A9306A

WolfGirl Token Distribution

The top 100 holders collectively own 61.86% (61,861,566,437,127.50 Tokens) of WolfGirl

Token Total Supply: 100,000,000,000,000.00 Token | Total Token Holders: 2,096

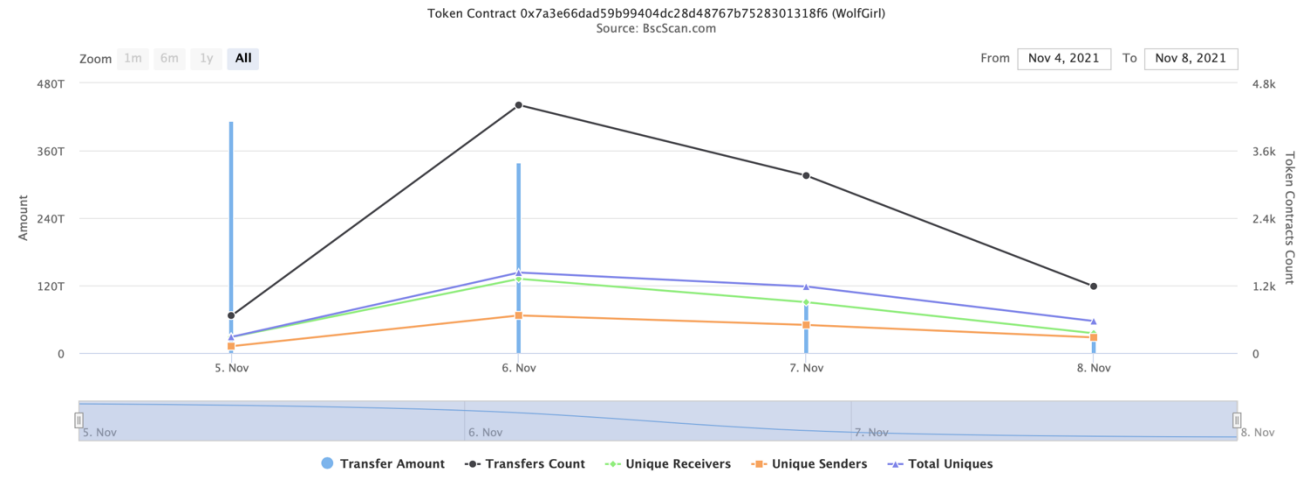


(A total of 61,861,566,437,127.50 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000.00 token)

WolfGirl Contract Interaction Details

Time Series: Token Contract Overview

Fri 5, Nov 2021 - Mon 8, Nov 2021



WolfGirl Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	Burn Address	8,500,000,000,000	8.5000%
2	PancakeSwap V2: WolfGirl	5,726,712,468,041.984020194731315668	5.7267%
3	0xce8b73df4234e62f1ffaa2f7ccdc341c54a9306a	4,319,442,476,038.288525433185876861	4.3194%
4	0x44843619c2985410c59ae358f6ba395c50fcad12	1,923,202,685,011.739836576318939883	1.9232%
5	0xad93115af52d2e63007ec049684fe84d11b0cf9f	1,247,709,410,137.752180393948480785	1.2477%
6	0x4d934100454ead136dbd5d108e554ef5d504c6dc	1,171,115,892,241.764953865785403224	1.1711%
7	0xafc84b304a297b13ed9f448c8ab7d767518bc9fb	1,147,280,720,562.708844622706343706	1.1473%
8	0x0185dd2b6d0b77e47d23bd63444fa1adea833db7	1,025,772,034,711.209480209086716311	1.0258%
9	0xd30dbb6e0a463815a732b190976fcb6d52185969	1,013,266,645,616.614069094197850552	1.0133%
10	0xba94a353b01b6a8770555fa891d69686e90a5417	951,000,000,000	0.9510%

WolfGirl Top LP Token Holders

Rank	Address	Quantity	Percentage
1	0xae7e6cabad8d80f0b4e1c4dde2a5db7201ef1252	23,000,789.076644904744265215	99.5849%
2	0xb1b9b4bbe8a92d535f5df2368e7fd2ecfb3a1950	94,970.65332525276436164	0.4112%
3	0x0627cca705f29d34997cb642328d28aa0b52984f	892.895311213740936563	0.0039%
4	0x0000000000000000000000000000000000000000	0.000000000000001	0.0000%

Contract functions details

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

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

+ 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] renounceOwnership #
    - modifiers: onlyOwner
  - [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] 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 #

+ WolfGirl (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] excludeFromFee #
    - modifiers: onlyOwner
  - [Pub] includeInFee #
    - modifiers: onlyOwner
  - [Ext] _set_Fees #
    - modifiers: onlyOwner
  - [Pub] Wallet_Update_Dev #
    - modifiers: onlyOwner
  - [Pub] set_Swap_And_Liquify_Enabled #
    - modifiers: onlyOwner
  - [Pub] set_Number_Of_Transactions_Before_Liquify_Trigger #
    - modifiers: onlyOwner
  - [Ext] <Fallback> ($)
  - [Ext] blacklist_Add_Wallets #
    - modifiers: onlyOwner
  - [Ext] blacklist_Remove_Wallets #
    - modifiers: onlyOwner
  - [Pub] blacklist_Switch #
    - modifiers: onlyOwner
  - [Ext] set_Transfers_Without_Fees #
    - modifiers: onlyOwner
  - [Ext] set_Max_Transaction_Percent #
    - modifiers: onlyOwner
  - [Ext] set_Max_Wallet_Percent #
    - modifiers: onlyOwner
  - [Prv] removeAllFee #
  - [Prv] restoreAllFee #
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] sendToWallet #
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Pub] process_Tokens_Now #
    - modifiers: onlyOwner
  - [Prv] swapTokensForBNB #
  - [Pub] remove_Random_Tokens #
    - modifiers: onlyOwner
  - [Pub] set_New_Router_and_Make_Pair #
    - modifiers: onlyOwner
  - [Pub] set_New_Router_Address #
    - modifiers: onlyOwner

- [Pub] set_New_Pair_Address #
  - modifiers: onlyOwner
- [Prv] _tokenTransfer #
- [Prv] _transferTokens #
- [Prv] _getValues

($) = 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.	Passed
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

No low severity issues found.

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

- Owner can include in and exclude from fees.

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

// Set a wallet address so that it has to pay transaction fees
function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

- Owner can change sell and buy fees.

```
function _set_Fees(uint256 Buy_Fee, uint256 Sell_Fee) external onlyOwner() {

    require((Buy_Fee + Sell_Fee) <= maxPossibleFee, "Fee is too high!");
    _sellFee = Sell_Fee;
    _buyFee = Buy_Fee;

}
```

- Owner can change dev wallet address.

```
// Update main wallet
function Wallet_Update_Dev(address payable wallet) public onlyOwner() {
    Wallet_Dev = wallet;
    _isExcludedFromFee[Wallet_Dev] = true;
}
```

- Owner can enable / disable swap and liquify.

```
// Toggle on and off to auto process tokens to BNB wallet
function set_Swap_And_Liquify_Enabled(bool true_or_false) public onlyOwner {
    swapAndLiquifyEnabled = true_or_false;
    emit SwapAndLiquifyEnabledUpdated(true_or_false);
}
```

- Owner can change number of transaction before liquify trigger.

```
// This will set the number of transactions required before the 'swapAndLiquify' function triggers
function set_Number_Of_Transactions_Before_Liquify_Trigger(uint8 number_of_transactions) public onlyOwner {
    swapTrigger = number_of_transactions;
}
```

- Owner can add / remove addresses from blacklist.

```
// Blacklist - block wallets (ADD - COMMA SEPARATE MULTIPLE WALLETS)
function blacklist_Add_Wallets(address[] calldata addresses) external onlyOwner {

    uint256 startGas;
    uint256 gasUsed;

for (uint256 i; i < addresses.length; ++i) {
    if(gasUsed < gasleft()) {
    startGas = gasleft();
    if(!_isBlacklisted[addresses[i]]){
    _isBlacklisted[addresses[i]] = true;}
    gasUsed = startGas - gasleft();
}
}
}



// Blacklist - block wallets (REMOVE - COMMA SEPARATE MULTIPLE WALLETS)
function blacklist_Remove_Wallets(address[] calldata addresses) external onlyOwner {

    uint256 startGas;
    uint256 gasUsed;

for (uint256 i; i < addresses.length; ++i) {
    if(gasUsed < gasleft()) {
    startGas = gasleft();
    if(_isBlacklisted[addresses[i]]){
    _isBlacklisted[addresses[i]] = false;}
    gasUsed = startGas - gasleft();
}
}
}
```

- Owner can enable / disable blacklist restrictions.

```
// Blacklist Switch - Turn on/off blacklisted wallet restrictions
function blacklist_Switch(bool true_or_false) public onlyOwner {
    noBlackList = true_or_false;
}
```

- Owner can enable / disable transfer fees.

```
function set_Transfers_Without_Fees(bool true_or_false) external onlyOwner {
    noFeeToTransfer = true_or_false;
}
```

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

```
// Set the Max transaction amount (percent of total supply)
function set_Max_Transaction_Percent(uint256 maxTxPercent_x100) external onlyOwner() {
    _maxTxAmount = _tTotal*maxTxPercent_x100/10000;
}
```

- **Owner can change maximum tokens amount per wallet.**

```
// Set the maximum wallet holding (percent of total supply)
 function set_Max_Wallet_Percent(uint256 maxWallPercent_x100) external onlyOwner() {
    _maxWalletToken = _tTotal*maxWallPercent_x100/10000;
}
```

- **Owner can manually process swap and liquify.**

```
// Manual Token Process Trigger - Enter the percent of the tokens that you'd like to send to process
function process_Tokens_Now (uint256 percent_Of_Tokens_To_Process) public onlyOwner {
    // Do not trigger if already in swap
    require(!inSwapAndLiquify, "Currently processing, try later.");
    if (percent_Of_Tokens_To_Process > 100){percent_Of_Tokens_To_Process == 100;}
    uint256 tokensOnContract = balanceOf(address(this));
    uint256 sendTokens = tokensOnContract*percent_Of_Tokens_To_Process/100;
    swapAndLiquify(sendTokens);
}
```

- **Owner can purge random tokens.**

```
// Remove random tokens from the contract and send to a wallet
function remove_Random_Tokens(address random_Token_Address, address send_to_wallet, uint256 number_of_tokens) public onlyOwner returns(bool _sent){
    require(random_Token_Address != address(this), "Can not remove native token");
    uint256 randomBalance = IERC20(random_Token_Address).balanceOf(address(this));
    if (number_of_tokens > randomBalance){number_of_tokens = randomBalance;}
    _sent = IERC20(random_Token_Address).transfer(send_to_wallet, number_of_tokens);
}
```

- **Owner can change uniswapV2pair and router.**

```
// Set new router and make the new pair address
function set_New_Router_and_Make_Pair(address newRouter) public onlyOwner() {
    IUniswapV2Router02 _newPCSRouter = IUniswapV2Router02(newRouter);
    uniswapV2Pair = IUniswapV2Factory(_newPCSRouter.factory()).createPair(address(this), _newPCSRouter.WETH());
    uniswapV2Router = _newPCSRouter;
}

// Set new router
function set_New_Router_Address(address newRouter) public onlyOwner() {
    IUniswapV2Router02 _newPCSRouter = IUniswapV2Router02(newRouter);
    uniswapV2Router = _newPCSRouter;
}

// Set new address - This will be the 'Cake LP' address for the token pairing
function set_New_Pair_Address(address newPair) public onlyOwner() {
    uniswapV2Pair = newPair;
}
```

Conclusion

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

Liquidity locking details provided by the team:
https://www.mudra.website/?certificate=yes&type=0&lp=0x3efc9248cb1c03c10b3dffbc6160d6851b920e58

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.