



**Rodeo Solutions**  
Develop - Audit - Coach

# **WOLFY Token Smart Contract Audit**



<b>WOLFY Token Smart Contract Audit</b>	<b>1</b>
Commision	3
Disclaimer	4
\$WOLFY Properties	5
Contract Functions	6
Public	6
View	6
Virtual	6
Executables	6
Owner Executables	6
Checklist	7
Potential Issues	8
Low severity warnings	8
Owner privileges	9
Conclusion	11



## Commision

Audited Project	WOLFY Token
Project website	<a href="https://token-wolf.com/">https://token-wolf.com/</a>
Contract Owner	<a href="#">0xa16188baa7406a2acbfa44c64335462538a932a5</a>
SmartContract Address	<a href="#">0x230Bd84540E40831bE73A2EeFD217B2d0D3C8015</a>
Blockchain	Binance Main Smart Chain

Rodeo Solutions was commissioned by WOLFY Token owners to perform an audit of their main smart contract.

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.



## 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 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 Rodeo Solutions and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rodeo Solutions) owe no duty of care towards you or any other person, nor does Rodeo Solutions 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 Rodeo Solutions 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, Rodeo Solutions hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rodeo Solutions, 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.



## \$WOLFY Properties

Contract name	WOLFY Token
Contract address	<a href="#">0x230Bd84540E40831bE73A2EeFD217B2d0D3C8015</a>
Total supply	1T
Token ticker	WOLFY
Decimals	6
Token holders	2
Transactions count	2
Top 100 holders dominance	100.00%
Liquidity fee	5%
Tax fee	4%
Total fees	9%
Mintable	No
Burnable	Manual burns only
Uniswap V2 pair	0x88e4f9ed39d774db4dbbf8d1d036e913189b3739
Contract deployer address	<a href="#">0xa16188baa7406a2acbfa44c64335462538a932a5</a>
Contract's current owner address	<a href="#">0xa16188baa7406a2acbfa44c64335462538a932a5</a>

As of 27/06/2021



## Contract Functions

### Public

#### View

```
name()
symbol()
decimals()
totalSupply()
allowance(address owner, address spender)
approve(address spender, uint256 amount)
isExcludedFromReward(address account)
totalFees()
reflectionFromToken(uint256 tAmount, bool deductTransferFee)
tokenFromReflection(uint256 rAmount)
isExcludedFromFee(address account)
```

#### Virtual

```
increaseAllowance(address spender, uint256 addedValue)
decreaseAllowance(address spender, uint256 subtractedValue)
```

### Executables

```
balanceOf(address account)
transfer(address recipient, uint256 amount)
transferFrom(address sender, address recipient, uint256 amount)
deliver(uint256 tAmount)
```

### Owner Executables

```
excludeFromReward(address account)
excludeFromFee(address account)
includeInReward(address account)
includeInFee(address account)
setTaxFeePercent(uint256 taxFee)
setLiquidityFeePercent(uint256 liquidityFee)
setMaxTxPercent(uint256 maxTxPercent)
setSwapAndLiquifyEnabled(bool _enabled)
setTaxEnable(bool _enable)
Extra standard from Ownable Contract
```



## Checklist

Compiler errors.	Passed
Possible delays in data delivery.	Passed
Timestamp dependence.	Passed
Integer Overflow and Underflow.	Passed
DoS with Revert.	Passed
DoS with block gas limit.	Low issues
Methods execution permissions.	Passed
Economy model of the contract.	Passed
Private user data leaks.	Passed
Malicious Event log.	Passed
Scoping and Declarations.	Passed
Uninitialized storage pointers.	Passed
Arithmetic accuracy.	Passed
Design Logic.	Passed
Cross-function race conditions.	Passed
Fallback function security.	Passed
Safe Open Zeppelin contracts implementation and usage.	Passed
Whitepaper-Website-Contract correlation.	Low issues



## Potential Issues

### Low severity warnings

The burning method used by the contract, being manual, renders the token as non-trustless. This means that it's up to the owners to follow through with the "Manual Burns" feature of the token.





## Owner privileges

- The owner can Exclude or Add an address from the rewards program. Not available on WOLFY.

```
function excludeFromReward(address account) public onlyOwner() {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router. ');
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

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 owner can Exclude or Add an address from being taxed.

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

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```



- The owner can change the Tax percentages at any time.

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
    emit SetTaxFeePercent(taxFee);
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
    emit SetLiquidityFeePercent(liquidityFee);
}
```

- The owner can enable or disable taxes at any time.

```
function setTaxEnable(bool _enable) public onlyOwner {
    _taxEnabled = _enable;
    emit SetTaxEnable(_enable);
}
```

- The owner can toggle the automatic liquidity swap at any time.

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

- The owner can set the maximum amount of a transaction at any time.

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

- Additionally the owner can execute the standard functions related to the Ownable Library



## Conclusion

The Smart Contract coded passed the audit successfully on the Binance Testnet.

There were only two warnings raised:

- The first warning should be made to clarify once again that the Token is not trustless as it depends on the owners to follow through the manual burns.
- The second warning is regarding the gas cost of the functions. Normally this doesn't represent any issues when working on the Binance chain but as the code is compatible with Ethereum, if it gets deployed on said chain, it should be tweaked for performance.

This new version verified the original code in the Testnet with the one pushed to the MainNet. Only the following changes were found:

1 /**	1 /**
2 *Submitted for verification at BscScan.com on 2021-06-28	2 *Submitted for verification at BscScan.com on 2021-06-24
3 */	3 */
4	4
5 /**	5 /**
761 IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5a578B854784E25	761 IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0xd99D1c3f9fC3444f810175408C46c52d16
762 6024E);	762 55001);
763 // Create a uniswap pair for this new token	763 // Create a uniswap pair for this new token

No changes need to be done to the Smart Contract as of 27/06/2021.