



RUGFREECOINS



Paw Stars Token

RugfreeCoins Verified on October 12th, 2023

Overview

- ✓ No mint function found, the owner cannot mint tokens after initial deployment.
- ✓ The owner can't set a max transaction limit
- ✓ The owner can't pause trading once it's enabled
- ✗ The owner can't change fees over 20%.
- ✓ The owner can't blacklist wallets.
- ✓ The owner can't set a max wallet limit
- ✓ The owner can't claim the contract's balance of its own token.

! HIGH SEVERITY ISSUES

The owner can set buy and sell fees total fees maximum of up-to 50%(buy 25% and sell 25%) and can not change marketing fee

```
function setBuyFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + buyMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidBuyFees();
    buyBurning = _burnFee;
    buyCharity = _chyFee;
    buyLp = _lpFee;
    totalBuyFee = totalFee;
    emit SetBuyFees(totalFee, _chyFee, _burnFee, _lpFee);
}

function setSellFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + sellMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidSellFees();
    sellBurning = _burnFee;
    sellCharity = _chyFee;
    sellLp = _lpFee;
    totalSellFee = totalFee;
    emit SetSellFees(totalFee, _chyFee, _burnFee, _lpFee);
}
```

Contents

Overview.....	2
Contents.....	3
Audit details.....	4
Disclaimer.....	5
Background.....	6
Tokenomics.....	7
Target market and the concept.....	8
Potential to grow with score points.....	9
Contract details.....	10
Contract code function details.....	11
Contract description table.....	12
Inheritance Hierarchy.....	17
Security issue checking status.....	18
Owner privileges.....	19
Audit conclusion.....	22

Audit details



Audited project
PawStars Token



Contract Address
0xFC914eCB4e4cbEea1Fcf5315129C6cdB398cd465



Client contact
PawStars Team



Blockchain
Binance Smart chain



Project website
<https://pawstar.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 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 **RugfreeCoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (RugfreeCoins) owe no duty of care towards you or any other person**, nor does RugfreeCoins 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 RugfreeCoins 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, RugfreeCoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against RugfreeCoins, 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

RugfreeCoins was commissioned by the **Paw Stars Token Team** to perform an audit of the smart contract.

<https://bscscan.com/token/0xFC914eCB4e4cbEea1Fcf5315129C6cdB398cd465>

This audit focuses on verifying that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

Tokenomics

▲ 5.5% tax when buying & selling (12/10/2023)

3% of trade goes to the marketing wallet in BNB

1% of trade goes to the marketing wallet in BNB

1% of trade goes to the Liquidity Pool

0.5% of trade goes to the burning wallet












▲ 2% tax when transfer from wallet to wallet

2% of trade goes to the burning wallet

Target market and the concept

- ▶ Anyone who's interested in the Crypto space with long-term investment plans.
- ▶ Anyone who's ready to earn a passive income by holding tokens.
- ▶ Anyone who's interested in trading tokens.
- ▶ Anyone who's interested in taking part in the Paw Stars token ecosystem.
- ▶ Anyone who's interested in taking part in the future plans of Paw Stars Token.
- ▶ Anyone who's interested in making financial transactions with any other party using Paw Stars Token as the currency.

Potential to grow with score points

 Project efficiency	8 / 10
 Project uniqueness	8 / 10
 Information quality	8 / 10
 Service quality	8 / 10
 System quality	8 / 10
 Impact on the community	8 / 10
 Impact on the business	9 / 10
 Preparing for the future	8 / 10
 Smart contract security	9 / 10
 Smart contract functionality assessment	10 / 10
 Total Score	8.4 / 10

Contract details

Token contract details for 12th of October 2023

Contract name	PawStars
Contract address	0xFC914eCB4e4cbEea1Fcf5315129C6cdB398cd465
Token supply	100,000,000,000
Token ticker	PAWS
Decimals	18
Token holders	1
Transaction count	1
Contract deployer address	0xAf72552eA03857dccE90E921005FBddA2d406afe
Contract's current owner address	0x54C83626c9A1190dE1829Cc4b141b0B65FCc36f0
Marketing wallet	0xa8bA202EA9039684821D4E5FAd18eaA88353646e
Charity wallet	0x54C83626c9A1190dE1829Cc4b141b0B65FCc36f0

Contract code function details








Nº	Category	Item	Result
1	Coding conventions	ERC20 Token standards	PASS ▾
		Compile errors	PASS ▾
		Compiler version security	PASS ▾
		Visibility specifiers	PASS ▾
		Gas consumption	PASS ▾
		SafeMath features	PASS ▾
		Fallback usage	PASS ▾
		tx.origin usage	PASS ▾
		Deprecated items	PASS ▾
		Redundant code	PASS ▾
2	Function call audit	Overriding variables	PASS ▾
		Authorization of function call	PASS ▾
		Low level function (call/delegate call) security	PASS ▾
		Returned value security	PASS ▾
3	Business security & centralisation	Self destruct function security	PASS ▾
		Access control of owners	HIGH ▾
		Business logics	PASS ▾
4	Integer overflow/underflow	Business implementation	PASS ▾
5	Reentrancy		PASS ▾
6	Exceptional reachable state		PASS ▾
7	Transaction ordering dependence		PASS ▾
8	Block properties dependence		PASS ▾
9	Pseudo random number generator (PRNG)		PASS ▾
10	DoS (Denial of Service)		PASS ▾
11	Token vesting implementation		PASS ▾
12	Fake deposit		PASS ▾
13	Event security		PASS ▾











Contract description table

The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !	⛔	NO !
L	allowance	External !		NO !
L	approve	External !	⛔	NO !
L	transferFrom	External !	⛔	NO !
IERC20 Metadata	Interface	IERC20		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
Context	Implementation			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
ERC20	Implementation	Context, IERC20, IERC20 Metadata		
L		Public !	⛔	NO !



L	name	Public !		NO !
L	symbol	Public !		NO !
L	decimals	Public !		NO !
L	totalSupply	Public !		NO !
L	balanceOf	Public !		NO !
L	transfer	Public !	⛔	NO !
L	allowance	Public !		NO !
L	approve	Public !	⛔	NO !
L	transferFrom	Public !	⛔	NO !
L	increaseAllowance	Public !	⛔	NO !
L	decreaseAllowance	Public !	⛔	NO !
L	_transfer	Internal 🔒	⛔	
L	_mint	Internal 🔒	⛔	
L	_burn	Internal 🔒	⛔	
L	_approve	Internal 🔒	⛔	
L	_spendAllowance	Internal 🔒	⛔	
L	_beforeTokenTransfer	Internal 🔒	⛔	
L	_afterTokenTransfer	Internal 🔒	⛔	
Ownable	Implementation	Context		
L		Public !	⛔	NO !
L	owner	Public !		NO !
L	_checkOwner	Internal 🔒		
L	renounceOwnership	Public !	⛔	onlyOwner
L	transferOwnership	Public !	⛔	onlyOwner
L	_transferOwnership	Internal 🔒	⛔	
IUniswapV2 Factory	Interface			
L	feeTo	External !		NO !

L	feeToSetter	External !		NO !
L	getPair	External !		NO !
L	allPairs	External !		NO !
L	allPairsLength	External !		NO !
L	createPair	External !		NO !
L	setFeeTo	External !		NO !
L	setFeeToSetter	External !		NO !
IUniswapV2 Pair	Interface			
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	allowance	External !		NO !
L	approve	External !		NO !
L	transfer	External !		NO !
L	transferFrom	External !		NO !
L	DOMAIN_SEPARATOR	External !		NO !
L	PERMIT_TYPEHASH	External !		NO !
L	nonces	External !		NO !
L	permit	External !		NO !
L	MINIMUM_LIQUIDITY	External !		NO !
L	factory	External !		NO !
L	token0	External !		NO !
L	token1	External !		NO !
L	getReserves	External !		NO !
L	price0CumulativeLast	External !		NO !
L	price1CumulativeLast	External !		NO !

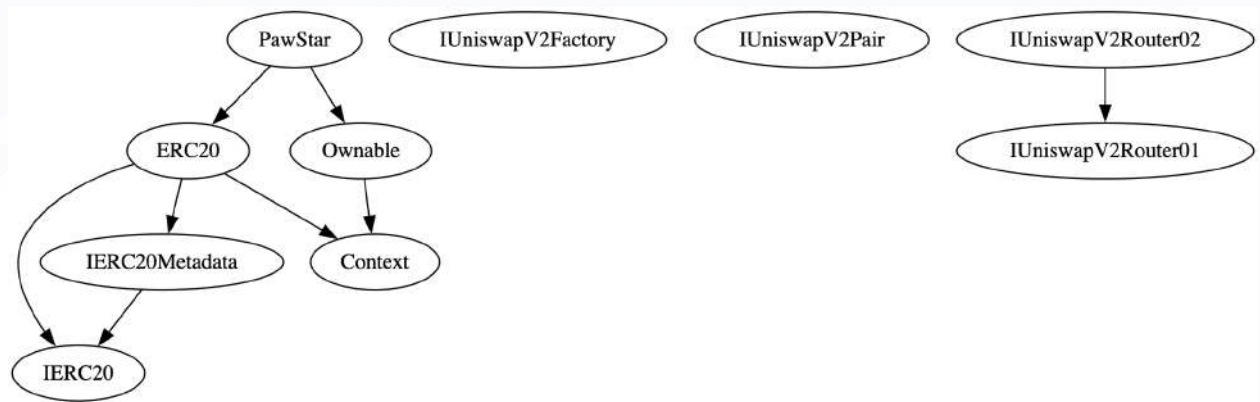
L	kLast	External !		NO !
L	burn	External !		NO !
L	swap	External !		NO !
L	skim	External !		NO !
L	sync	External !		NO !
L	initialize	External !		NO !
IUniswapV2 Router01	Interface			
L	factory	External !		NO !
L	WETH	External !		NO !
L	addLiquidity	External !		NO !
L	addLiquidityETH	External !		NO !
L	removeLiquidity	External !		NO !
L	removeLiquidityETH	External !		NO !
L	removeLiquidityWithPermit	External !		NO !
L	removeLiquidityETHWithPermit	External !		NO !
L	swapExactTokensForTokens	External !		NO !
L	swapTokensForExactTokens	External !		NO !
L	swapExactETHForTokens	External !		NO !
L	swapTokensForExactETH	External !		NO !
L	swapExactTokensForETH	External !		NO !
L	swapETHForExactTokens	External !		NO !
L	quote	External !		NO !
L	getAmountOut	External !		NO !
L	getAmountIn	External !		NO !
L	getAmountsOut	External !		NO !
L	getAmountsIn	External !		NO !

IUniswapV2 Router02	Interface	IUniswapV2 Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
PawStar	Implementation	ERC20, Ownable		
L		Public !		ERC20
L		External !		NO !
L	swapAndDistributeBNB	Public !		non Reentrant
L	_transfer	Internal 		
L	distributeBuy	Internal 		
L	distributeSell	Internal 		
L	setBuyFees	External !		onlyOwner
L	setSellFees	External !		onlyOwner
L	setThreshold	External !		onlyOwner
L	setTransferFee	External !		onlyOwner
L	setExcludedStatus	External !		onlyOwner
L	setCharityWallet	External !		onlyOwner

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

❖ High severity issues

The owner can set buy and sell fees total fees maximum of up-to 50%(buy 25% and sell 25%) and can not change marketing fee

```
function setBuyFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + buyMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidBuyFees();
    buyBurning = _burnFee;
    buyCharity = _chyFee;
    buyLp = _lpFee;
    totalBuyFee = totalFee;
    emit SetBuyFees(totalFee, _chyFee, _burnFee, _lpFee);
}

function setSellFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + sellMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidSellFees();
    sellBurning = _burnFee;
    sellCharity = _chyFee;
    sellLp = _lpFee;
    totalSellFee = totalFee;
    emit SetSellFees(totalFee, _chyFee, _burnFee, _lpFee);
}
```

❖ Medium severity issues

No medium severity issues found

❖ Low severity issues

No low severity issues found

Owner privileges

- ❖ Anyone can manually swap the tokens before swap point met

```
function swapAndDistributeBNB() public nonReentrant {
    // get the LP balance and swap half for BNB.
    uint bnbBalance = address(this).balance;
    uint lpBalance = totalTokensLp;
    uint amountToSwap = lpBalance / 2;
    lpBalance -= amountToSwap;

    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = WETH;
    // Liquify lp allocated tokens
    if (lpBalance > 0) {
        router.swapExactTokensForETHSupportingFeeOnTransferTokens(
            amountToSwap,
            0,
            path,
            address(this),
            block.timestamp
        );
        // add liquidity with the bnb and lp
        amountToSwap = address(this).balance - bnbBalance;
        router.addLiquidityETH{value: amountToSwap}(
            address(this),
            lpBalance,
            0,
            0,
            DEAD,
            block.timestamp
        );
        totalTokensLp = 0;
    }
    bnbBalance = balanceOf(address(this));
    amountToSwap = totalTokensCharity + totalTokensMarketing;
    if (amountToSwap > 0) {
        // swap rest of tokens for BNB entirely
        router.swapExactTokensForETHSupportingFeeOnTransferTokens(
            bnbBalance,
            0,
            path,
            address(this),
            block.timestamp
        );
        // send BNB to marketing and charity wallets
        bnbBalance = address(this).balance;
        uint marketingAmount = (bnbBalance * totalTokensMarketing) /
            amountToSwap;
        uint charityAmount = bnbBalance - marketingAmount;
        bool succ;
        if (marketingAmount > 0) {
            (succ, ) = marketingWallet.call{value: marketingAmount}("");
            if (!succ) emit FailTransfer(marketingWallet, marketingAmount);
        }
        if (charityAmount > 0) {
            (succ, ) = charityWallet.call{value: charityAmount}("");
            if (!succ) emit FailTransfer(charityWallet, charityAmount);
        }
        totalTokensMarketing = 0;
        totalTokensCharity = 0;
    }
}
```

- ❖ The owner can set buy and sell fees total fees maximum of up-to 50%(buy 25% and sell 25%) and can not change marketing fee

```
function setBuyFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + buyMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidBuyFees();
    buyBurning = _burnFee;
    buyCharity = _chyFee;
    buyLp = _lpFee;
    totalBuyFee = totalFee;
    emit SetBuyFees(totalFee, _chyFee, _burnFee, _lpFee);
}

function setSellFees(
    uint8 _chyFee,
    uint8 _lpFee,
    uint8 _burnFee
) external onlyOwner {
    uint8 totalFee = _chyFee + sellMarketing + _lpFee + _burnFee;
    if (totalFee > PERCENTAGE / 4) revert PAWS__InvalidSellFees();
    sellBurning = _burnFee;
    sellCharity = _chyFee;
    sellLp = _lpFee;
    totalSellFee = totalFee;
    emit SetSellFees(totalFee, _chyFee, _burnFee, _lpFee);
}
```

- ❖ The owner can change the swap threshold

```
function setThreshold(uint128 _threshold) external onlyOwner {
    threshold = _threshold;
    emit SetThreshold(_threshold);
}
```

- ❖ The owner can change transfer fees maximum of up-to 10%

```
function setTransferFee(uint8 _transferFee) external onlyOwner {  
    if (_transferFee > PERCENTAGE / 10) revert PAWS__InvalidTransferFee();  
    walletToWallet = _transferFee;  
    emit SetTransferFee(_transferFee);  
}
```

- ❖ The owner can include/exclude wallets from fees

```
function setExcludedStatus(address user, bool status) external onlyOwner {  
    require(  
        isExcludedFromFee[user] != status,  
        "The wallet already has that value!"  
    );  
    isExcludedFromFee[user] = status;  
    emit SetExclusionStatus(user, status);  
}
```

- ❖ The owner can change the charity wallet address

```
function setCharityWallet(address payable _cty) external onlyOwner {  
    require(_cty != address(0), "setCharityWallet: ZERO");  
    emit SetCharityWallet(charityWallet, _cty);  
    charityWallet = _cty;  
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testing, line-by-line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status:	PASS ▾
Smart contract security Status:	HIGH ISSUES ▾
Number of risk issues:	1
Solidity code functional issue level:	PASS ▾
Number of owner privileges:	6
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾