RugFreeCoins Audit

Drago Land Token

Smart Contract Security Audit

January 07, 2022

Contents

Audit details

Audited project

Drago Land Token

Contract Address

0x4D10E2E1E157962b50CEE968B35B4392dE0a9EA5

Client contact

Drago Land Team

Blockchain

Binance smart chain

Project website

https://dragoland.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 Drago Land Token to perform an audit of the smart contract.

https://bscscan.com/address/0x4D10E2E1E157962b50CEE968B35B4392dE0a9EA5

The focus of this audit is to verify 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, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Drago Land is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, which in turn, powers play to earn game. Each transaction, purchase incurs 6% fee, and sale incurs a 7% fee.

Features

- The **sustainability fee of 2% when buying and selling for dev and 4% when buying and selling for marketing** is what allows Drago Land to hold the aforementioned promise. Tokens will be swapped into BUSD and will be sent to a marketing wallet and dev wallet. This way, Drago Land will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

Tokenomics

6% fee when buying and selling

- 2% of trade goes to the Dev wallet.
- 4% of trade goes to the marketing wallet.

Roadmap

SEP-NOV 2021 - PHASE 1

- NFT Characters Design (BEP20)
- Smart Contracts Development
- Play2Earn Ecosystem Development
- NFT Marketplace Development
- Website Development
- Beta Browser Game Development

DEC-JAN 2022 - PHASE 2

- Introduce Dragoland To The Public
- Community Growth
- Social Media Marketing
- Strategic Partnerships
- Private Sales & Presale
- Dex Listing (Pancakeswap)
- Dragoland Browser Game Release (Dragons VS Monsters, NFT Shop, NFT Marketplace, Staking/Farming)

Q1 2022 - PHASE 3

- Android & IOS Game App
- Branded Swap
- Additional Mini-Games
- Dragon Breeding
- Additional Dragon Evolutions
- CEX Listings
- Cross-Chain Dragoland to Solana

Q2 2022 - PHASE 4

- Leaderboard Release
- Guilds & Guild Wars Release
- Additions Cross-Chain Integration

Target market and the concept

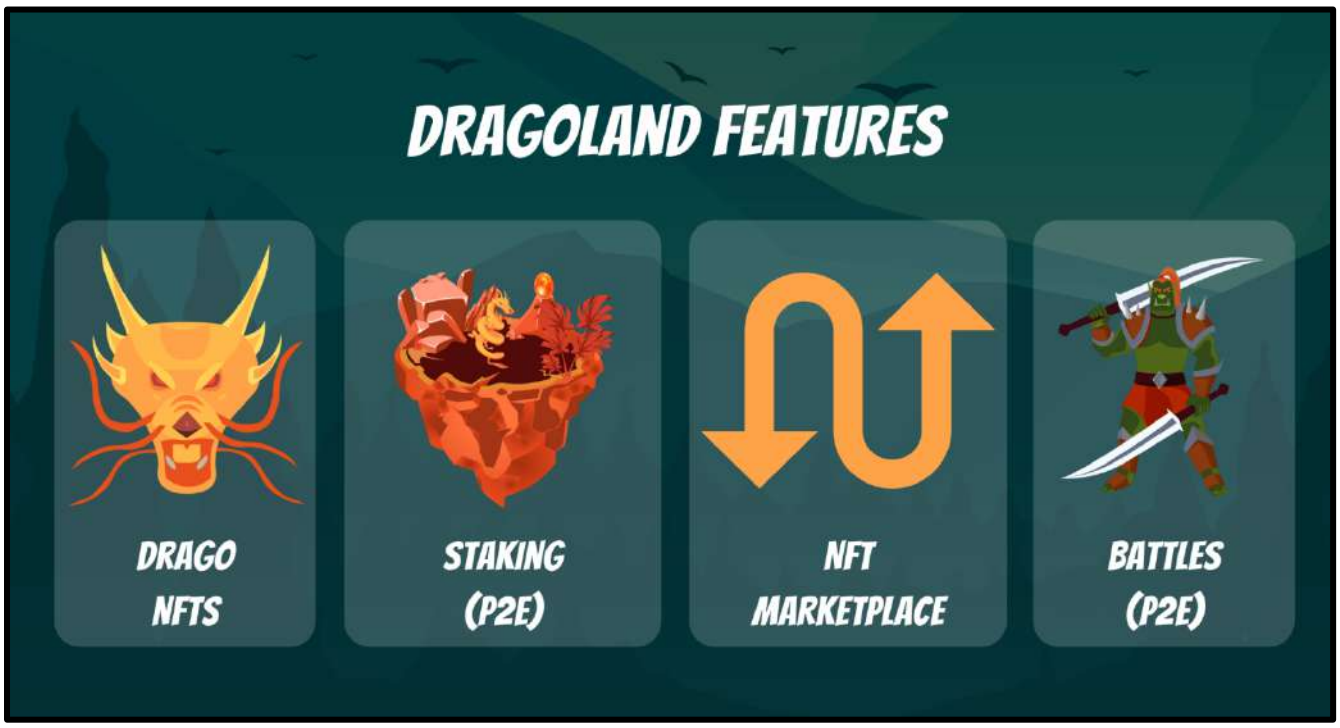
Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's interested in trading tokens.
- Anyone who's interested in owning NFTs.
- Anyone who's interested in doing daily tasks in the game and getting rewards
- Anyone who's interested in collecting NFTs or trading NFTs.
- Anyone who's interested in taking part with the future plans of the Drago Land token.
- Anyone who's interested in making financial transactions with any other party using Drago Land as the currency.

Core concept

Sustainable mechanism

The **sustainability fee of 2% when buying and selling for dev and 4% when buying and selling for marketing** is what allows Drago Land to promote the token and use funds to further the development of the platform. Tokens will be swapped into BUSD and will be sent to a marketing wallet and dev wallet. This way, Drago Land will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.




ECOSYSTEM MAP
EXCHANGE
NFT MARKET
$DRAGO TOKEN
REWARDS
GOVERNANCE
STAKING
BATTELING
P2E
CRAFT ITEMS
DRAGON KINGDOM
EVOLVED DRAGON
MYSTERY EGG
DRAGON EGG


HOW TO GET A DRAGO EGG?
75,000
MYSTERY EGG
2,500
DRAGON BREATH
DARKNESS
ENERGY
LIGHT
EARTH
WATER
FIRE


DARKNESS
9% Rarity
ENERGY
13% Rarity
LIGHT
3% Rarity
EACH DRAGO EGG HAS A DIFFERENT RARITY
EARTH
15% Rarity
WATER
25% Rarity
FIRE
35% Rarity


ANY EGG
30,000
LIFE ELIXIR
EVOLVE YOUR EGGS INTO DRAGOS
FIRE
WATER
EARTH
ENERGY
DARKNESS
LIGHT


CRAFT ITEMS
150
ROCK
250
LAVA
100
ICE
50
DRAGON FLAME
450
CRYSTALS
250
DRAGON BREATH
175
LIGHTNING BOLT
30,000
LIFE ELIXIR


DRAGO KINGDOMS
PURCHASE CRAFT ITEMS AND CRAFT DIFFERENT DRAGO KINGDOMS. EACH DRAGO KINGDOM ALLOWS YOU TO SET 3 DRAGOS TO PROTECT IT (STAKING) THE MORE KINGDOMS YOU HAVE AND THE MORE DRAGOS YOU HAVE PROTECTING IT, THE HIGHER YOUR REWARD WILL BE.
KEEP IN MIND, THAT IN ORDER TO EVOLVE YOUR DRAGO EGG INTO A DRAGO YOU MUST STAKE THE EGG FOR AT LEAST 1 DAY IN ORDER TO QUALIFY IT TO BE EVOLVED INTO A DRAGO.
400
DARKNESS
300
EARTH
350
ENERGY
999
LIGHT
90
FIRE
160
WATER

Potential to grow with score points

1.	Project efficiency	10/10
2.	Project uniqueness	9/10
3	Information quality	10/10
4	Service quality	10/10
5	System quality	10/10
6	Impact on the community	9/10
7	Impact on the business	10/10
8	Preparing for the future	10/10
Total Points		**9.75/10**

Contract details

Token contract details for 07th January 2022

Contract name	Drago
Contract address	0x4D10E2E1E157962b50CEE968B35B4392dE0a9EA5
Token supply	1,000,000,000
Token ticker	Drago
Decimals	0
Token holders	1
Transaction count	1
Dev wallet	0x8d96e9678d2fae750f4e0c50a82160359e31ef00
Marketing wallet	0xcd45fad7f03067d3d03ea4fbfc73fe1c09d25d57
Contract deployer address	0x1F357b095a9667d74f28724D7A5E7E3Af701076A
Contract's current owner address	0x1f357b095a9667d74f28724d7a5e7e3af701076a

Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 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
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		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

Token distribution

Tokens are distributed as follows:

Token Distribution	Token Supply	Vesting Period	Total Supply
In-Game Rewards	700,000,000	**Lifetime lock** inside the in-game rewards smart contract.	70%
Presale (PinkSale)	200,000,000	The liquidity pool generated by the presale is **locked for one year.**	20%
Marketing Wallet	50,000,000	**Locked for 13 months**: after the first 3 months, 10% will be released every 30 days.	5%
Team Wallet	30,000,000	**Locked for 13 months**: after the first 3 months, 10% will be released every 30 days.	3%
Private Sale	20,000,000	Any allocation above 2 BNB will be vested for **2 weeks** after the launch day.	2%

Contract description table

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 its visibility and mutability.

Contract	**Type**	**Bases**		
└	Function Name	Visibility	Mutability	Modifiers
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IUniswapV2Router01	**Interface**			
└	factory	External ❗		NO❗

└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🔘	NO❗
└	addLiquidityETH	External ❗	💷	NO❗
└	removeLiquidity	External ❗	🔘	NO❗
└	removeLiquidityETH	External ❗	🔘	NO❗
└	removeLiquidityWithPermit	External ❗	🔘	NO❗
└	removeLiquidityETHWithPermit	External ❗	🔘	NO❗
└	swapExactTokensForTokens	External ❗	🔘	NO❗
└	swapTokensForExactTokens	External ❗	🔘	NO❗
└	swapExactETHForTokens	External ❗	💷	NO❗
└	swapTokensForExactETH	External ❗	🔘	NO❗
└	swapExactTokensForETH	External ❗	🔘	NO❗
└	swapETHForExactTokens	External ❗	💷	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗

IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💷	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
IUniswapV2Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🔘	NO❗

└	setFeeTo	External ❗	🔘	NO❗
└	setFeeToSetter	External ❗	🔘	NO❗
IUniswapV2Pair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔘	NO❗
└	transfer	External ❗	🔘	NO❗
└	transferFrom	External ❗	🔘	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🔘	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗

└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🔘	NO❗
└	burn	External ❗	🔘	NO❗
└	swap	External ❗	🔘	NO❗
└	skim	External ❗	🔘	NO❗
└	sync	External ❗	🔘	NO❗
└	initialize	External ❗	🔘	NO❗
IERC20Metadata	**Interface**	**IERC20**		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗

Ownable	**Implementation**	**Context**		
└		Public ❗	🔘	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🔘	onlyOwner
└	transferOwnership	Public ❗	🔘	onlyOwner
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
ERC20	**Implementation**	**Context, IERC20, IERC20Metadata**		
└		Public ❗	🔘	NO❗
└	name	Public ❗		NO❗

└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
Drago	**Implementation**	**ERC20, Ownable**		
└		Public ❗	🛑	ERC20

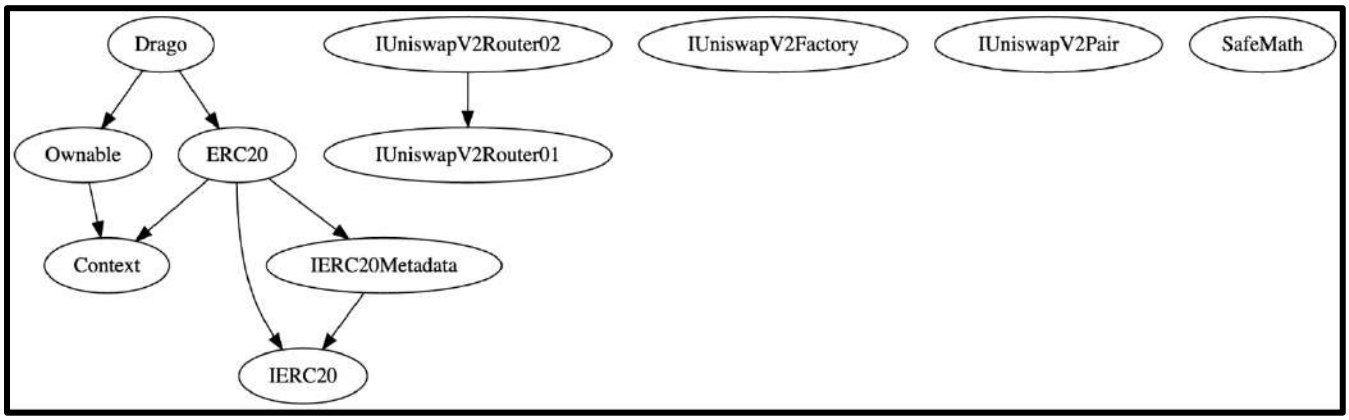
└		External ❗	💸	NO❗
└	updateUniswapV2Router	Public ❗	🛑	onlyOwner
└	excludeFromFees	Public ❗	🛑	onlyOwner
└	setswapTokensAtAmount	External ❗	🛑	onlyOwner
└	excludeMultipleAccountsFromFees	Public ❗	🛑	onlyOwner
└	setDevelopmentWallet	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner
└	setDevelopmentFee	External ❗	🛑	onlyOwner
└	setMarketingFee	External ❗	🛑	onlyOwner
└	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
└	removeMaxWallet	Public ❗	🛑	onlyOwner
└	setMaxBuyTransaction	External ❗	🛑	onlyOwner
└	setMaxSellTransaction	External ❗	🛑	onlyOwner
└	setMaxWalletToken	External ❗	🛑	onlyOwner
└	whitelistPinkSale	Public ❗	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	isExcludedFromFees	Public ❗		NO❗
└	_transfer	Internal 🔒	🛑	

L	swapAndSendToFee	Private		
L	swapTokensForBUSD	Private		

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**
  No high severity issues found.

- **Medium severity issues**
  No medium severity issues found.

- **Low severity issues**
  No low severity issues found.

- **Issues informed and fixed**

  This contract is not adding liquidity, but all functions that want to add liquidity is there (addLiquidity,swapTokensForEth,swapAndLiquify), can reduce contract gas fees by removing the unwanted functions.

```
ftrace | funcSig
function addLiquidity(uint256 tokenAmount⬆, uint256 ethAmount⬆) private {

    // approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount⬆);

    // add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount⬆}(
        address(this),
        tokenAmount⬆,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        address(0),
        block.timestamp
    );

}
```

Status: Fixed

Fess will deduct in wallet-to-wallet transfers

In the documentation they mentioned there are no fees on in-game transactions, but fees will deduct on every transaction.

```
if(takeFee) {
    uint256 fees = amount⬆.mul(totalFees).div(100);
    if(automatedMarketMakerPairs[to⬆]){
        fees += amount⬆.mul(1).div(100);
    }
    amount⬆ = amount⬆.sub(fees);

    super._transfer(from⬆, address(this), fees);
}
```

> *There are no fees on in-game transactions.*

Status: Fixed

More than one swap happening

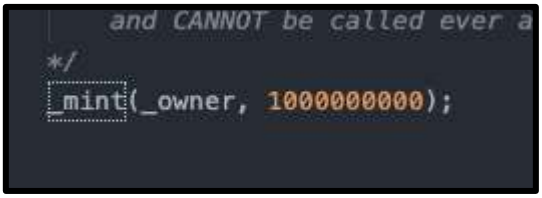
To send bnb to marketing and dev wallet contract swapping them separately this will increase the gas fee twice, so contract can swap both tokens at once and send swapped bnb according to the ratio.

```
swapping = true;
uint256 marketingTokens = contractTokenBalance.mul(marketingFee).div(totalFees);
swapAndSendToFee(_marketingWalletAddress, marketingTokens);

uint256 developmentTokens = contractTokenBalance.mul(developmentFee).div(totalFees);
swapAndSendToFee(_developmentWalletAddress, developmentTokens);

swapping = false;
```

Status: Fixed

Undeclared variable

_owner variable not declared in the contract, because of that contract will not deploy, can replace _owner with owner ()



Status: Fixed

Unused address

deadWallet address not use in the contract, so removing it can reduce gas fee.

Status: Fixed

Owner privileges

- The owner can update the router address.

```
ftrace | funcSig
function updateUniswapV2Router(address newAddress⬆) public onlyOwner {
    require(newAddress⬆ != address(uniswapV2Router), "Drago: The router already has that address");
    emit UpdateUniswapV2Router(newAddress⬆, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress⬆);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

- The owner can exclude wallets from fee.

```
ftrace | funcSig
function excludeFromFees(address account⬆, bool excluded⬆) public onlyOwner {
    require(_isExcludedFromFees[account⬆] != excluded⬆, "Drago: Account is already the value of 'excluded'");
    _isExcludedFromFees[account⬆] = excluded⬆;

    emit ExcludeFromFees(account⬆, excluded⬆);
}
```

- The owner can change token swap amount.

```
ftrace | funcSig
function setswapTokensAtAmount (uint256 value⬆) external onlyOwner{
    swapTokensAtAmount = value⬆;
}
```

- The owner can change the dev and marketing wallet.

```
ftrace | funcSig
function setDevelopmentWallet(address payable wallet⬆) external onlyOwner{
    _developmentWalletAddress = wallet⬆;
}

ftrace | funcSig
function setMarketingWallet(address payable wallet⬆) external onlyOwner{
    _marketingWalletAddress = wallet⬆;
}
```

❖ The owner can change the dev and marketing fee.

```
ftrace | funcSig
function setDevelopmentFee(uint256 value⬆) external onlyOwner{
    developmentFee = value⬆;
    totalFees = developmentFee.add(marketingFee);
}

ftrace | funcSig
function setMarketingFee(uint256 value⬆) external onlyOwner{
    marketingFee = value⬆;
    totalFees = developmentFee.add(marketingFee);

}
```

❖ The owner can remove max wallet token limitation.

```
ftrace | funcSig
function removeMaxWallet() public onlyOwner {
    maxWalletTokens = 1000000000;
}
```

❖ The owner can change max buy,sell and wallet token amount.

```
ftrace | funcSig
function setMaxBuyTransaction(uint256 maxTxn⬆) external onlyOwner {
    maxBuyTranscationAmount = maxTxn⬆;
}

ftrace | funcSig
function setMaxSellTransaction(uint256 maxTxn⬆) external onlyOwner {
    maxSellTransactionAmount = maxTxn⬆;
}

ftrace | funcSig
function setMaxWalletToken(uint256 maxToken⬆) external onlyOwner {
    maxWalletTokens = maxToken⬆;
}
```

- The owner can whitelist pinksale address.

Audit conclusion

While conducting the audit of the Drago Land smart contract, it was observed that there is nothing alarming with the code and the issues have already communicated and fixed.