RugFreeCoins Audit

Kakapo Token

Smart Contract Security Audit

December 3, 2021

Contents

Audit details

Audited project

Kakapo Token

Contract Address

0x0b8D7b793488319fB91A16dA6B97CE085379355D

Client contact

kakapotoken Token Team

Blockchain

Binance smart chain

Project website

https://www.kakapotoken.com/

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 Kakapo to perform an audit of the smart contract.

https://bscscan.com/address/0x0b8D7b793488319fB91A16dA6B97CE085379355D#code

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

kakapo Token is a token built on the Binance Smart Chain that protects endangered animals. Almost 4,000,000,000 species have gone extinct, since the beginning of life on Earth. kakapo token mission is to prevent that number from increasing. Each Kakapo token symbolizes the species that have gone extinct. The initial goal is to save the Kakapo from extinction, and many more animals will follow. Each transaction, purchase and sell incur a 10% fee.

Features

- ❖ **8% Charity fee** per transaction when buying and selling will be swapped to BNB and sent to the kakapo token charity wallet. This will help to protect a large variety of animals that are highly endangered starting from New Zealand based Kākāpō birds.

- ❖ The **sustainability fee of 1% for operations** is what allows kakapotoken to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a operation wallet per transaction. This way, kakapo token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- ❖ The additional component included under the sustainability section is a **liquidity fee of 1%,** which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.

The kākāpō

The kākāpō from the Māori: kākāpō, lit. 'night parrot'), also called owl parrot (Strigops habroptilus), is a species of large, flightless, nocturnal, ground-dwelling parrot of the super-family Strigopoidea, endemic to New Zealand.

It has finely blotched yellow-green plumage, a distinct facial disc, a large grey beak, short legs, large feet, and relatively short wings and tail. A combination of traits make it unique among parrots: it is the world's only flightless parrot, the world's heaviest parrot, nocturnal, herbivorous, visibly sexually dimorphic in body size, has a low basal metabolic rate, and no male parental care, and is the only parrot to have a polygynous lek breeding system. It is also possibly one of the world's longest-living birds, with a reported lifespan of up to 100 years.

Its anatomy typifies the tendency of bird evolution on oceanic islands with few predators and abundant food to exhibit island syndrome: a generally robust physique at the expense of flight abilities, resulting in reduced wing muscles and a diminished keel on the sternum. Like many other New Zealand bird species, the kākāpō was historically important to Māori, the indigenous people of New Zealand, appearing in many of their traditional legends and folklore; however it was also heavily hunted and used as a resource by Māori, both for its meat as a food source and for its feathers, which were used to make highly valued pieces of clothing. Kākāpō were also occasionally kept as pets.

The kākāpō is critically endangered; the total known adult population is 201 living individuals, all of which are named and tagged, confined to four small islands off the coast of New Zealand that have been cleared of predators. The introduction of predators such as cats, rats, ferrets, and stoats during British colonisation almost wiped out the kākāpō. Conservation efforts began in the 1890s, but they were not very successful until the implementation of the Kakapo Recovery Programme in 1995.

Most kākāpō are kept on two predator-free islands, Codfish / Whenua Hou and Anchor, where they are closely monitored, and Little Barrier / Hauturu Island is being trialled as a third home for the species.

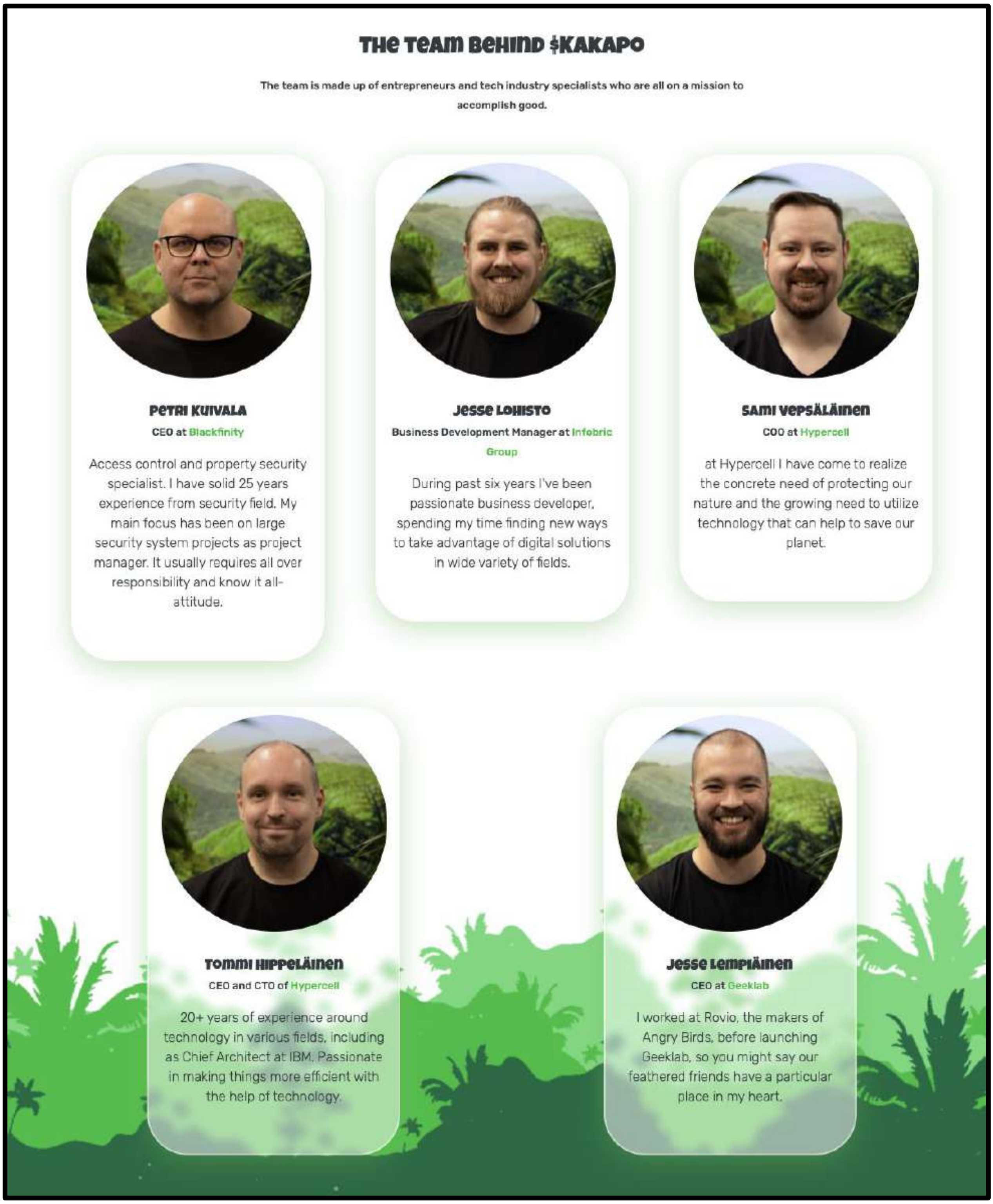
Genetics

Because kākāpō passed through a genetic bottleneck, in which their world population was reduced to 49 birds, they are extremely inbred and have low genetic diversity. This manifests in lower disease resistance and fertility problems: 40% of kākāpō eggs are infertile. Beginning in 2015, the Kākāpō 125 project aimed to sequence the genome of all living kākāpō, as well as some museum specimens – the first time an entire species has had its genome sequenced. The project is a collaboration between Duke University and the New Zealand Genomics lab in Dunedin.

Habitat

Before the arrival of humans, the kākāpō was distributed throughout both main islands of New Zealand. Although it may have inhabited Stewart Island / Rakiura before human arrival, it has so far not been found in the extensive fossil collections from there. Kākāpō lived in a variety of habitats, including tussocklands, scrublands and coastal areas. It also inhabited forests dominated by podocarps (rimu, matai, kahikatea, totara), beeches, tawa, and rata. In Fiordland, areas of avalanche and slip debris with regenerating and heavily fruiting vegetation – such as five finger, wineberry, bush lawyer, tutu, hebes, and coprosmas – became known as "kākāpō gardens".

The kākāpō is considered to be a "habitat generalist". Though they are now confined to islands free of predation, they were once able to live in nearly any climate present on the islands of New Zealand. They survived dry, hot summers on the North Island as well as cold winter temperatures in the sub-alpine areas of Fiordland. Kākāpō seem to have preferred broadleaf or mountain beech and Hall's tōtara forest with mild winters and high rainfall, but the species was not exclusively forest-dwelling. All kākāpō that were transferred to predator-free islands in the last decades have adapted well to any changes in environment and food plants.



The Team Behind $KAKAPO

The team is made up of entrepreneurs and tech industry specialists who are all on a mission to accomplish good.

Petri Kuivala

CEO at Blackfinity

Access control and property security specialist. I have solid 25 years experience from security field. My main focus has been on large security system projects as project manager. It usually requires all over responsibility and know it all-attitude.

Jesse Lohisto

Business Development Manager at Infobric Group

During past six years I've been passionate business developer, spending my time finding new ways to take advantage of digital solutions in wide variety of fields.

Sami Vepsäläinen

COO at Hypercell

at Hypercell I have come to realize the concrete need of protecting our nature and the growing need to utilize technology that can help to save our planet.

Tommi Hippeläinen

CEO and CTO of Hypercell

20+ years of experience around technology in various fields, including as Chief Architect at IBM. Passionate in making things more efficient with the help of technology.

Jesse Lempiäinen

CEO at Geeklab

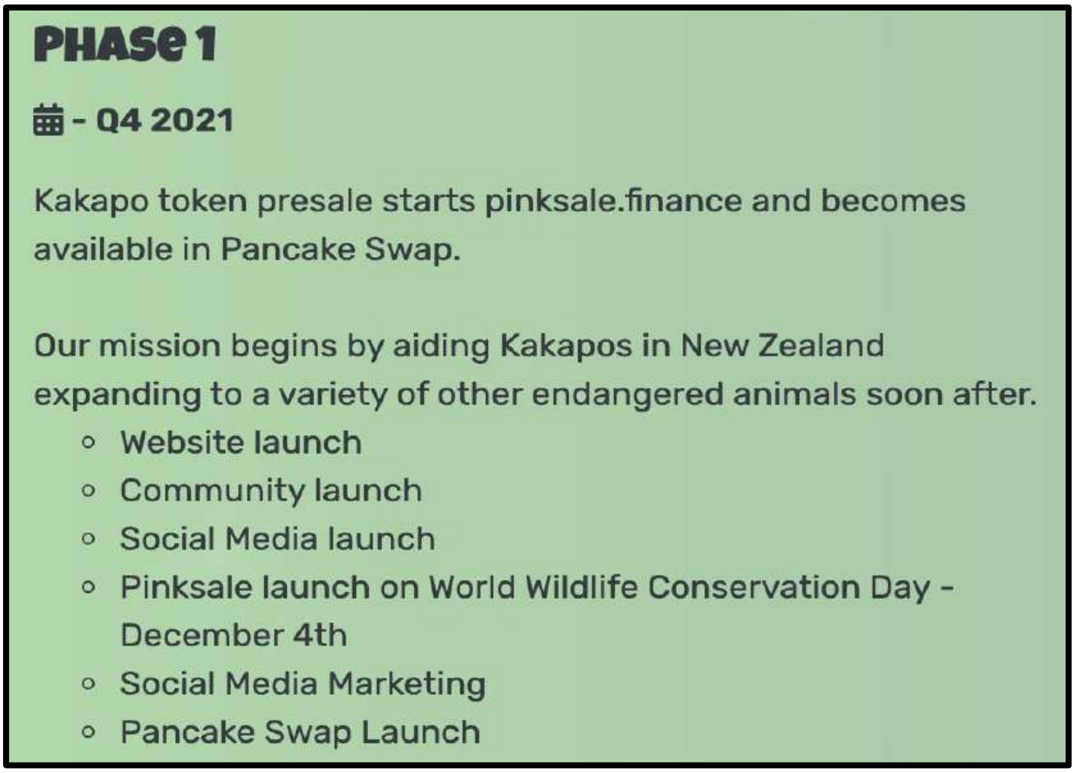
I worked at Rovio, the makers of Angry Birds, before launching Geeklab, so you might say our feathered friends have a particular place in my heart.

Tokenomics

10% fee when buy and sell

- 8% of trade goes to the charity wallet.
- 1% of trade goes to the operation wallet.
- 1% of trade goes to liquidity pool.

Roadmap



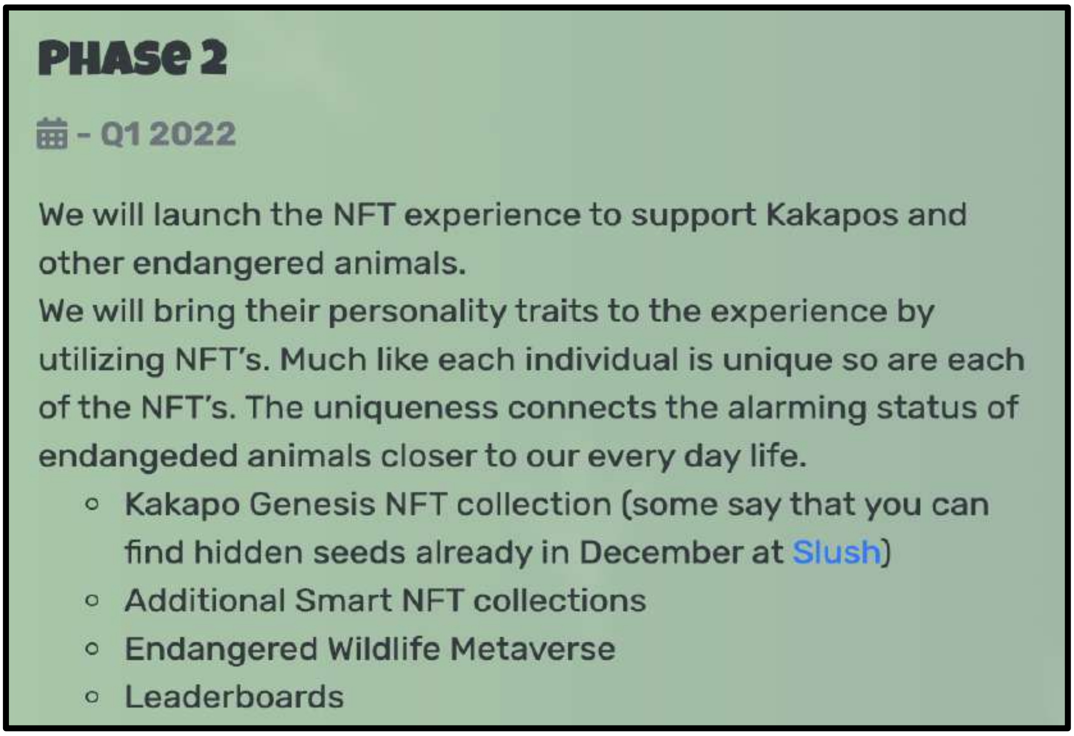
PHASE 1

📅 - Q4 2021

Kakapo token presale starts pinksale.finance and becomes available in Pancake Swap.

Our mission begins by aiding Kakapos in New Zealand expanding to a variety of other endangered animals soon after.

- Website launch
- Community launch
- Social Media launch
- Pinksale launch on World Wildlife Conservation Day - December 4th
- Social Media Marketing
- Pancake Swap Launch

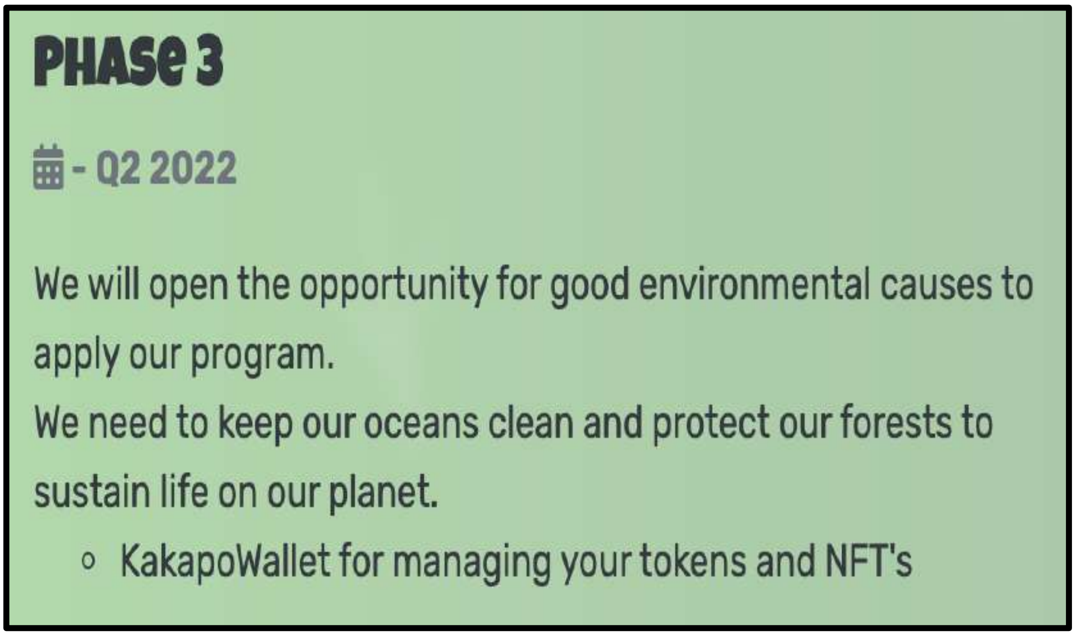


PHASE 2

📅 - Q1 2022

We will launch the NFT experience to support Kakapos and other endangered animals.
We will bring their personality traits to the experience by utilizing NFT's. Much like each individual is unique so are each of the NFT's. The uniqueness connects the alarming status of endangeded animals closer to our every day life.

- Kakapo Genesis NFT collection (some say that you can find hidden seeds already in December at Slush)
- Additional Smart NFT collections
- Endangered Wildlife Metaverse
- Leaderboards

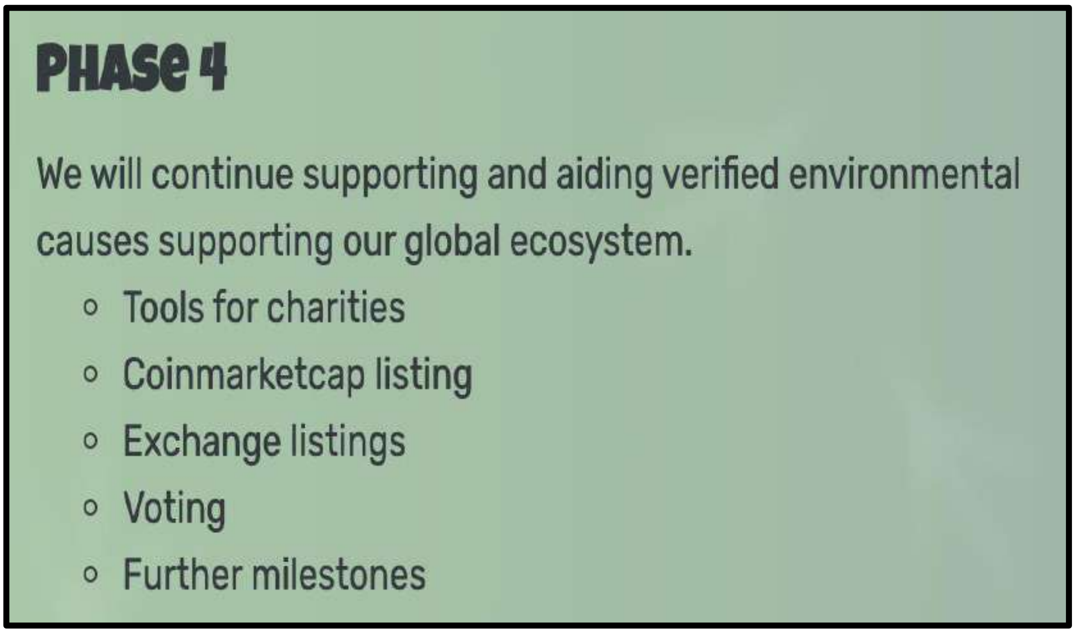


PHASE 3

📅 - Q2 2022

We will open the opportunity for good environmental causes to apply our program.
We need to keep our oceans clean and protect our forests to sustain life on our planet.

- KakapoWallet for managing your tokens and NFT's



PHASE 4

We will continue supporting and aiding verified environmental causes supporting our global ecosystem.

- Tools for charities
- Coinmarketcap listing
- Exchange listings
- Voting
- Further milestones

Target market and the concept

Target market

- Anyone who's interested in trading tokens.
- Anyone who's interested in supporting the good cause.
- Anyone who's interested in collecting NFTs or trading NFTs.
- Anyone who's interested in taking part with the future plans of the kakapo token.
- Anyone who's interested in making financial transactions with any other party using kakapo token as the currency.

Core concept

Good cause

8% Charity fee when buying and selling will be swapped to BNB and sent to the Kakapo Token charity wallet. This will help to protect a large variety of animals that are highly endangered starting from New Zealand based Kākāpō birds.

Sustainable mechanism

The **sustainability fee of 1% when selling for operations** is what allows kakapo Token to promote the token and use funds to further development of the platform. Tokens will be swapped into BNB and will be sent to a marketing and gaming pool wallet per transaction. This way, kakapo Token will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The liquidity fee of 1% when selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Future plan

Kakapo Token will launch the NFT experience to support Kakapos and other endangered animals. We will bring their personality traits to the experience by utilizing NFT's. Much like each individual is unique so are each of the NFT's. The uniqueness connects the alarming status of endangered animals closer to our everyday life.

- Kakapo Genesis NFT collection (some say that you can find hidden seeds already in December at Slush).
- Additional Smart NFT collections.
- Endangered Wildlife Metaverse.
- Leaderboards.

Potential to grow with score points

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

Contract details

Token contract details for 03rd December 2021

Contract name	Kakapo Token
Contract address	0x0b8D7b793488319fB91A16dA6B97CE085379355D
Token supply	3,998,434,323
Token ticker	KAKAPO
Decimals	18
Token holders	1
Transaction count	1
Charity wallet	0x7d959f34d10d8b94ece680e117ccf88c0c0df201
Marketing wallet	0x82d7a31732f98fb9f9879b0e09db0e4aba79fa60
Contract deployer address	0xfAc69aEcA3ad4ceC08e4D1F93f13a7356378a8e0
Contract’s current owner address	0xfac69aeca3ad4cec08e4d1f93f13a7356378a8e0

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

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❗
IERC20Metadata	**Interface**	**IERC20**		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
Context	**Implementation**			

└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
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 🔒	🛑	
SafeMathUint	**Library**			
└	toInt256Safe	Internal 🔒		
SafeMathInt	**Library**			
└	mul	Internal 🔒		

L	div	Internal 🔒		
L	sub	Internal 🔒		
L	add	Internal 🔒		
L	abs	Internal 🔒		
L	toUint256Safe	Internal 🔒		
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
IUniswapV2Pair	**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❗

└	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❗
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❗
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❗

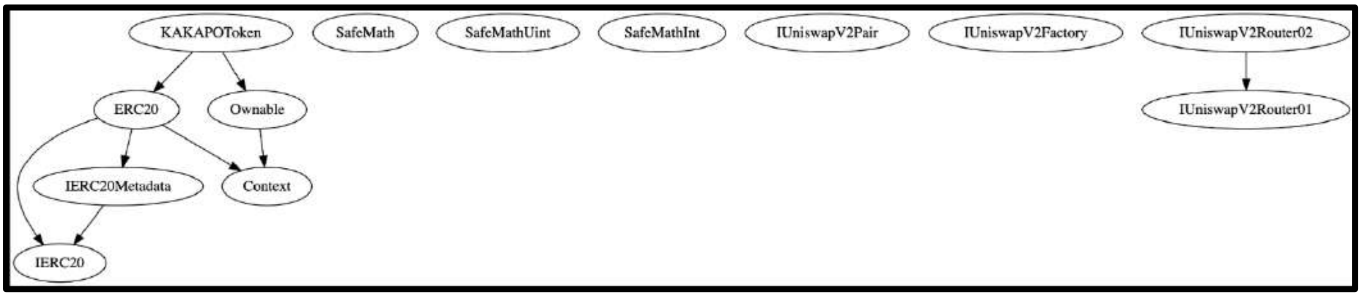
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
KAKAPOToken	**Implementation**	**ERC20, Ownable**		
L		Public ❗	🛑	ERC20
L		External ❗	💵	NO❗
L	updateUniswapV2Router	Public ❗	🛑	onlyOwner
L	excludeFromFees	Public ❗	🛑	onlyOwner
L	excludeMultipleAccountsFromFees	Public ❗	🛑	onlyOwner
L	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
L	_setAutomatedMarketMakerPair	Private 🔐	🛑	
L	updateGasForProcessing	External ❗	🛑	onlyOwner
L	updateswapTokensAtAmount	Public ❗	🛑	onlyOwner

└	updateMarketingWallet	Public ❗	🛑	onlyOwner
└	startTrading	Public ❗	🛑	onlyOwner
└	whitelistPresale	Public ❗	🛑	onlyOwner
└	updateCharityWallet	External ❗	🛑	onlyOwner
└	updateSwapPrecentage	External ❗	🛑	onlyOwner
└	updateTransactionFee	External ❗	🛑	onlyOwner
└	updateGasFeePercentage	External ❗	🛑	onlyOwner
└	updateAutomaticSwap	External ❗	🛑	onlyOwner
└	clearBNB	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	swapContractTokens	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	swapAndSendBnb	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	**Function can modify state**
💵	**Function is payable**

Inheritance Hierarchy



Security issue checking status

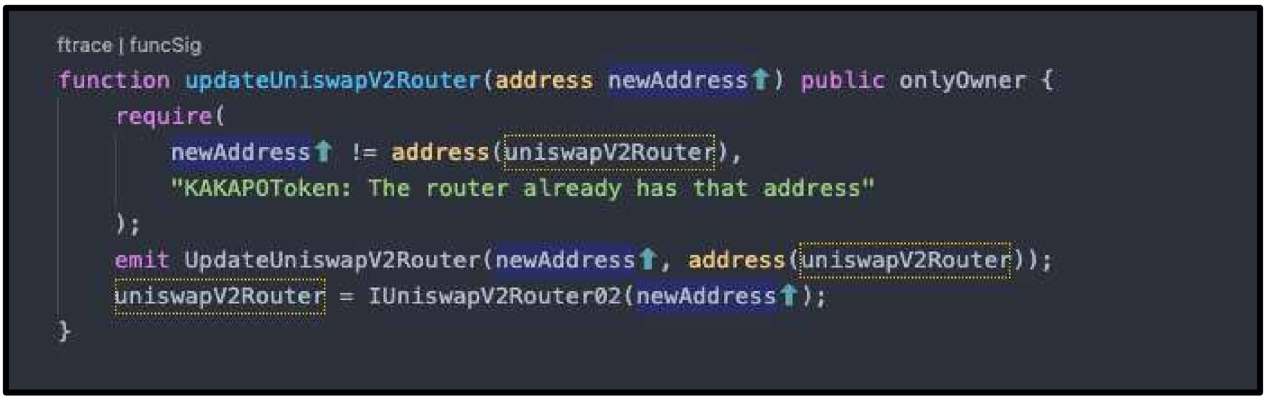
- **High severity issues**
  No high severity issues found.

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

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

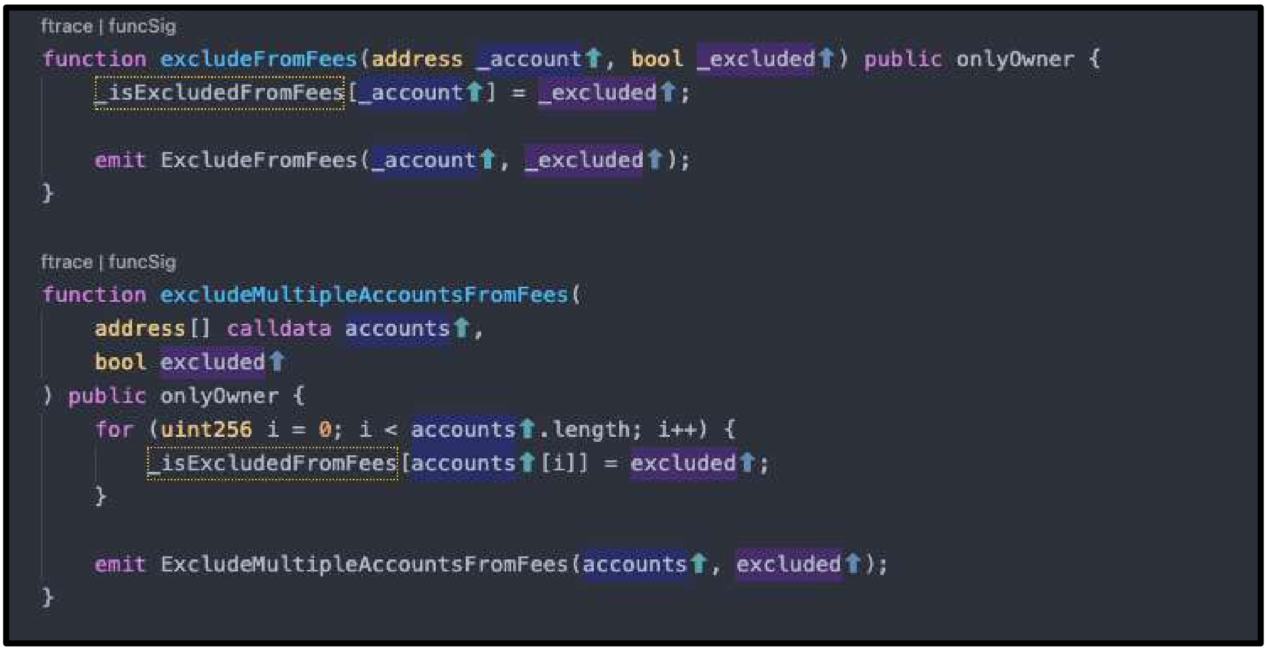
Owner privileges

- The owner can update the router address.



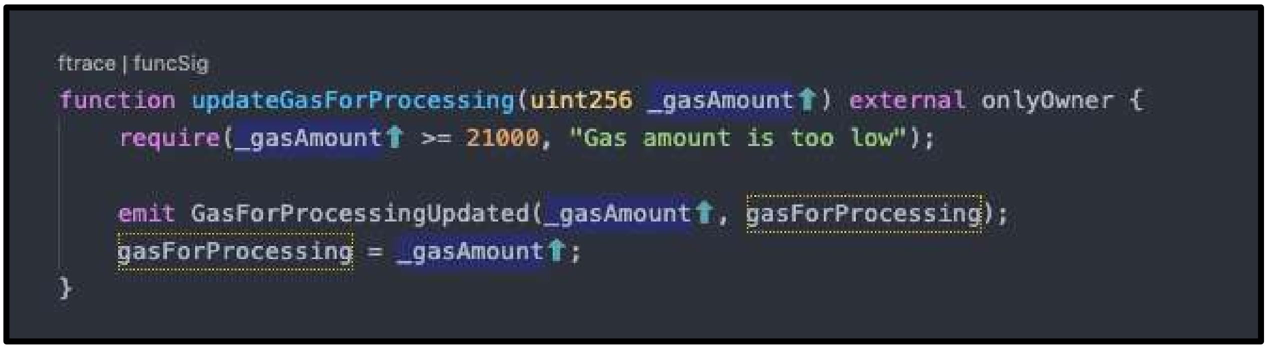
```
ftrace | funcSig
function updateUniswapV2Router(address newAddress) public onlyOwner {
    require(
        newAddress != address(uniswapV2Router),
        "KAKAPOToken: The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress);
}
```

- The owner can exclude wallets from the fees.



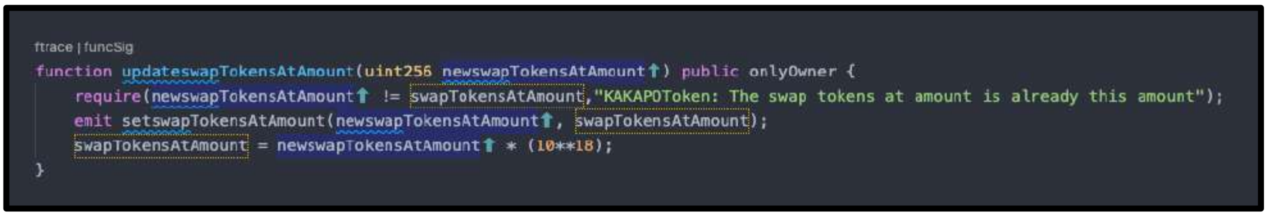
```
ftrace | funcSig
function excludeFromFees(address _account, bool _excluded) public onlyOwner {
    _isExcludedFromFees[_account] = _excluded;

    emit ExcludeFromFees(_account, _excluded);
}

ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

- The owner can update to process gas fee.



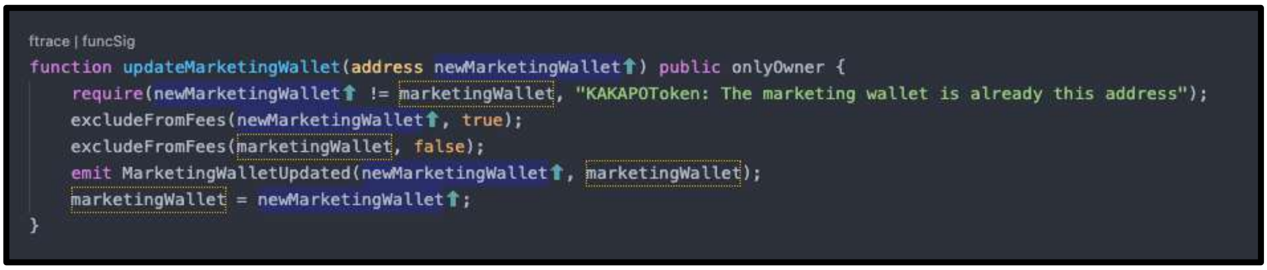
```
ftrace | funcSig
function updateGasForProcessing(uint256 _gasAmount) external onlyOwner {
    require(_gasAmount >= 21000, "Gas amount is too low");

    emit GasForProcessingUpdated(_gasAmount, gasForProcessing);
    gasForProcessing = _gasAmount;
}
```

- The owner can change the swap point.



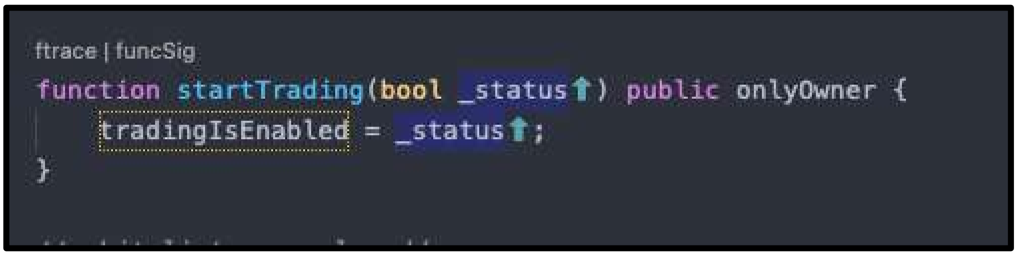
```
ftrace | funcSig
function updateswapTokensAtAmount(uint256 newswapTokensAtAmount) public onlyOwner {
    require(newswapTokensAtAmount != swapTokensAtAmount,"KAKAPOToken: The swap tokens at amount is already this amount");
    emit setswapTokensAtAmount(newswapTokensAtAmount, swapTokensAtAmount);
    swapTokensAtAmount = newswapTokensAtAmount * (10**18);
}
```

- The owner can update the marketing wallet.



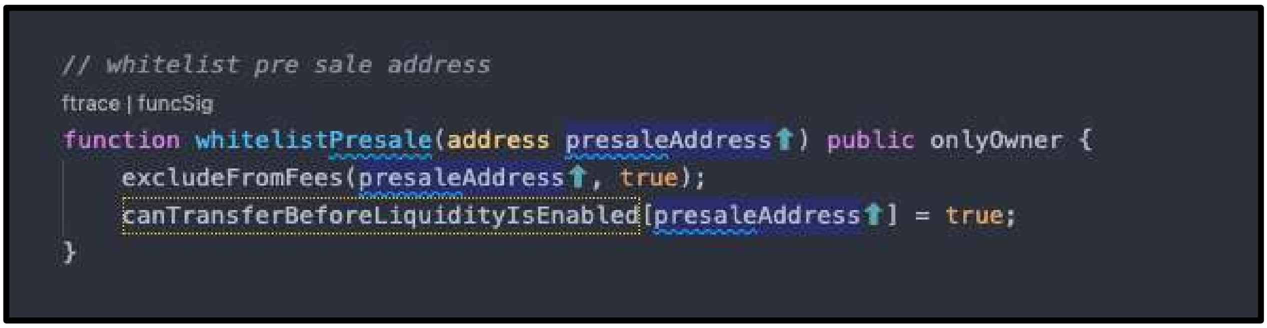
```
ftrace | funcSig
function updateMarketingWallet(address newMarketingWallet) public onlyOwner {
    require(newMarketingWallet != marketingWallet, "KAKAPOToken: The marketing wallet is already this address");
    excludeFromFees(newMarketingWallet, true);
    excludeFromFees(marketingWallet, false);
    emit MarketingWalletUpdated(newMarketingWallet, marketingWallet);
    marketingWallet = newMarketingWallet;
}
```

- The owner can enable and disable trading.



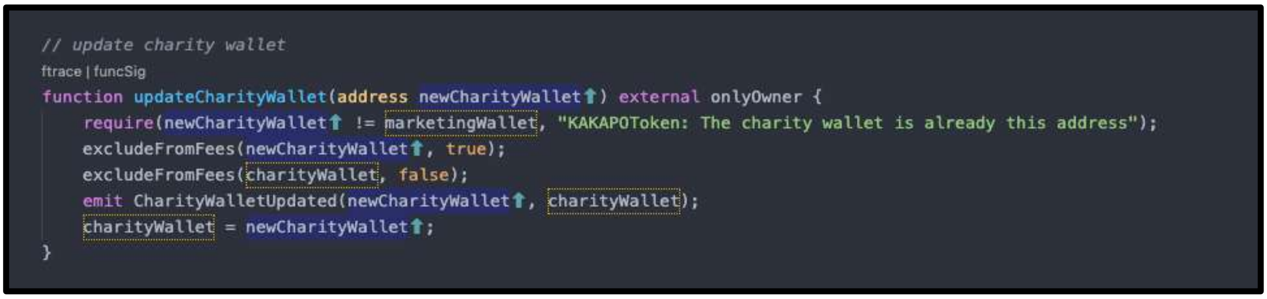
```
ftrace | funcSig
function startTrading(bool _status) public onlyOwner {
    tradingIsEnabled = _status;
}
```

- The owner can whitelist pre sale address.



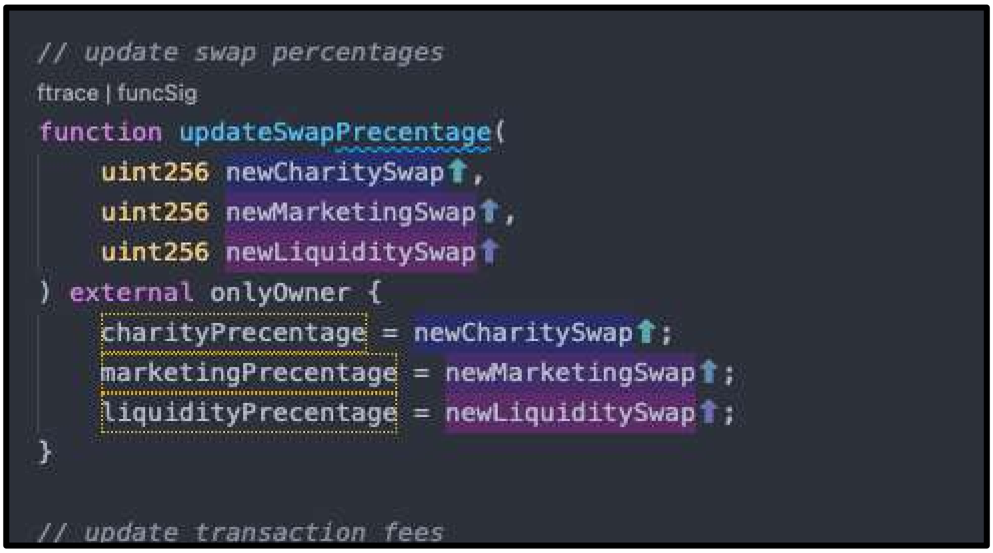
```
// whitelist pre sale address
ftrace | funcSig
function whitelistPresale(address presaleAddress) public onlyOwner {
    excludeFromFees(presaleAddress, true);
    canTransferBeforeLiquidityIsEnabled[presaleAddress] = true;
}
```

- The owner can update the charity wallet.



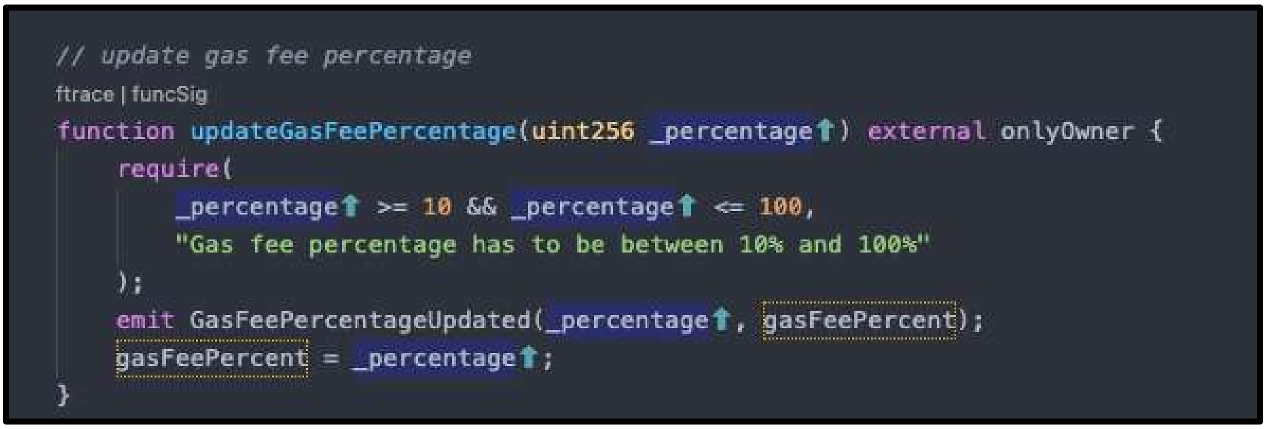
```
// update charity wallet
ftrace | funcSig
function updateCharityWallet(address newCharityWallet) external onlyOwner {
    require(newCharityWallet != marketingWallet, "KAKAPOToken: The charity wallet is already this address");
    excludeFromFees(newCharityWallet, true);
    excludeFromFees(charityWallet, false);
    emit CharityWalletUpdated(newCharityWallet, charityWallet);
    charityWallet = newCharityWallet;
}
```

- The owner can update the swap precentage.



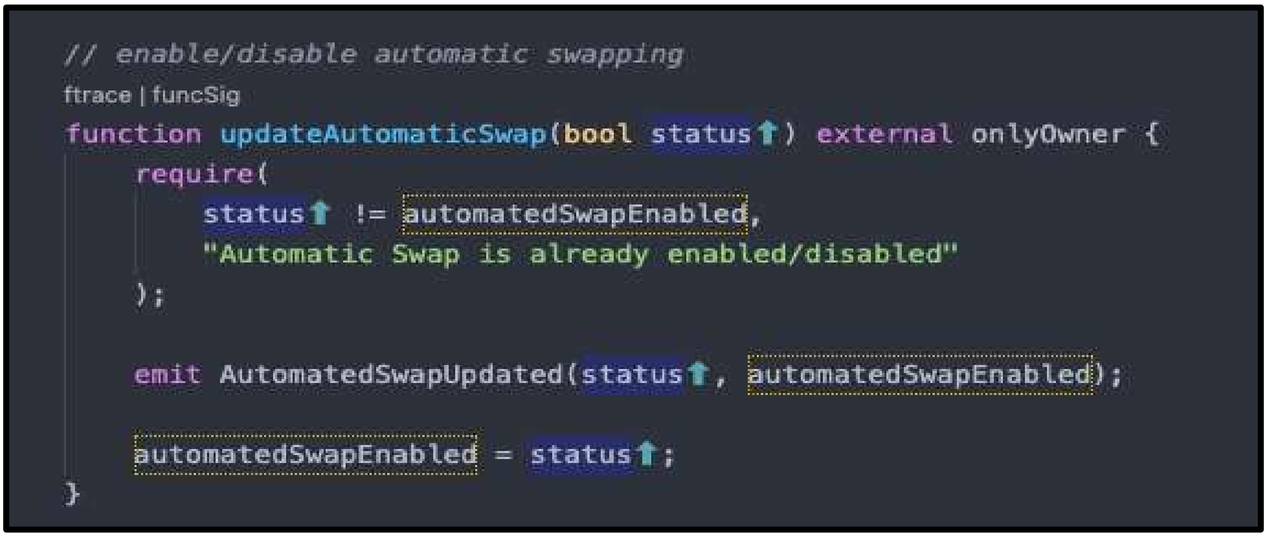
```
// update swap percentages
ftrace | funcSig
function updateSwapPrecentage(
    uint256 newCharitySwap,
    uint256 newMarketingSwap,
    uint256 newLiquiditySwap
) external onlyOwner {
    charityPrecentage = newCharitySwap;
    marketingPrecentage = newMarketingSwap;
    liquidityPrecentage = newLiquiditySwap;
}

// update transaction fees
```

- The owner can update the gas fee precentage.



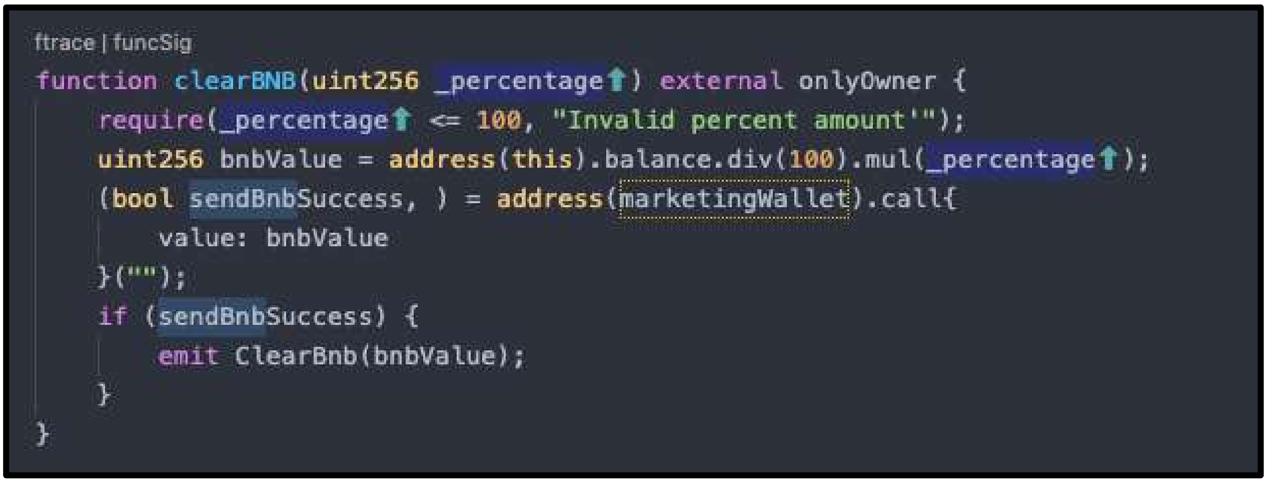
```
// update gas fee percentage
ftrace | funcSig
function updateGasFeePercentage(uint256 _percentage) external onlyOwner {
    require(
        _percentage >= 10 && _percentage <= 100,
        "Gas fee percentage has to be between 10% and 100%"
    );
    emit GasFeePercentageUpdated(_percentage, gasFeePercent);
    gasFeePercent = _percentage;
}
```

- The owner can enable and disable automatic swap.



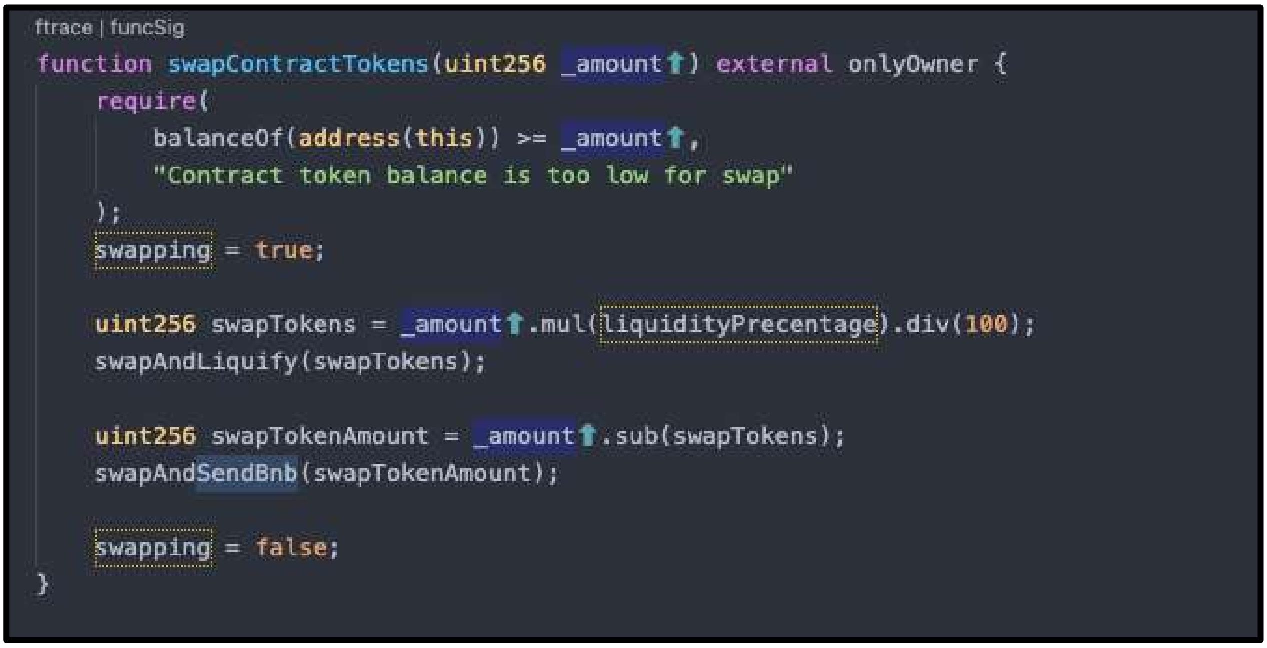
```
// enable/disable automatic swapping
ftrace | funcSig
function updateAutomaticSwap(bool status) external onlyOwner {
    require(
        status != automatedSwapEnabled,
        "Automatic Swap is already enabled/disabled"
    );

    emit AutomatedSwapUpdated(status, automatedSwapEnabled);

    automatedSwapEnabled = status;
}
```

- ❖ The owner can get stuck BNB balance in contract to the marketing wallet.



```
function clearBNB(uint256 _percentage) external onlyOwner {
    require(_percentage <= 100, "Invalid percent amount'");
    uint256 bnbValue = address(this).balance.div(100).mul(_percentage);
    (bool sendBnbSuccess, ) = address(marketingWallet).call{
        value: bnbValue
    }("");
    if (sendBnbSuccess) {
        emit ClearBnb(bnbValue);
    }
}
```

- ❖ The owner can swap tokens manually.



```
function swapContractTokens(uint256 _amount) external onlyOwner {
    require(
        balanceOf(address(this)) >= _amount,
        "Contract token balance is too low for swap"
    );
    swapping = true;

    uint256 swapTokens = _amount.mul(liquidityPrecentage).div(100);
    swapAndLiquify(swapTokens);

    uint256 swapTokenAmount = _amount.sub(swapTokens);
    swapAndSendBnb(swapTokenAmount);

    swapping = false;
}
```

Audit conclusion

While conducting the audit of the Kakapo Token smart contract, it was observed that there is nothing alarming with the code.