RugFreeCoins Audit

Predator Token Audit

Smart Contract Security Audit

August 22, 2021

Contents

Audit details

Audited project

Predator Token

Contract Address

0xe59046e1a4a83c11ccc578e26da4eeec8484ed8d

Client contact

Predator Token Team

Blockchain

Binance smart chain

Project website

https://predatortoken.org/

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.

Background

Rugfreecoins was commissioned by Predator Token to perform an audit of the smart contract.

https://bscscan.com/token/0xe59046e1a4a83c11ccc578e26da4eeec8484ed8d

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

Predator is a token built on the Binance Smart Chain. Each transaction, purchase incur a 15% fee, and sales incur an 13% fee. The Predator Token launched on 6th August 2021 on PancakeSwap.

The token aims to become the next big thing in the crypto world with its predatory approach. Offering a range of investment opportunities and services that could attract thousands of new traders to the ecosystem, Predator is accessible across multiple markets. Predator is confident in its superior tokenomics, as it focuses on extreme sports, and will become the industry's cryptocurrency leader.

Features

❖ 10% of each transaction when buying and 13% when selling gets converted to BUSD and is split amongst all holders. The holders will be eligible to receive tokens everyone hour and rewards are proportional to how many tokens each individual holds.

❖ The **sustainability fee of 3% marketing fee when buying and 4.5% when selling** is what allows Predator to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Predator will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

❖ **1.5% fee** from buying and selling is allocated to buyback tokens to save from massive dips in order to keep the token market price stable.

❖ **0.5% Charity fee** per transaction is allocated for charity. This will empower the Predator community in the long run and motivate more people to join in!

Tokenomics

15% fee when buying

- 10% of trade goes to holders pockets in BUSD.

5% to a private wallet
- 3% of trade will be allocated for marketing.
- 1.5% of trade will be allocated to buyback tokens.
- 0.5% of trade will be allocated for charity.

19.5% fee when selling

- 13% of trade goes to holders pockets in BUSD.

6.5% to a private wallet
- 4.5% of trade will be allocated for marketing.
- 1.5% of trade will be allocated for buyback tokens.
- 0.5% of trade will be allocated for charity.

Team

Roadmap

Roadmap

August 2021

Presale and PancakeSwap Launch
Contract Audit
Global Marketing Launch

Q3 2021

CoinGecko Listing
CoinMarketCap Listing
PredFit Launch
Fitness Partnerships Start
Charity Partnerships Start
UniSwap Bridge and Listing
PredNFT Launch
Exchange Listing #1
Exchange Listing #2
Blockfolio, Delta, Stocktwits Listing

Q4 2021

PredatorTV Launch
NFT Tickets Launch
Exchange Listings #3 and #4
Tier-1 Exchange Listing

2022

Anonymous Payment Launch
Cross-Chain Migration

Target market and the concept

- ❖ Anyone who's interested in Crypto space with long term investment plans.
- ❖ Anyone who's ready to earn a passive income in BNB by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ All BNB investors and fans out there.
- ❖ Anyone who's interested in taking part with the games, NFT marketplace and NFT ticketing system that's going to be built by the Predator team.
- ❖ Anyone who's interested in taking part with Predator TV and other future plans.
- ❖ Anyone who's interested in making financial transactions with any other party using Predator token as the currency.

Core concept

The BUSD reward system

10% of each transaction when buying and 13% when selling gets converted to BUSD and is split amongst all holders. The rewards are sent to holders that have at least 1 Predator tokens, holders will be eligible to receive tokens everyone hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The marketing fee of 3% when buying and 4.5% when selling is what allows Predator 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 wallet per transaction. This way, Predator 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 controlled buyback collects 1.5% tax on each transaction, which is stored inside a private wallet. It is to save from massive dips in order to keep the token market price stable.

Good cause

0.5% Charity fee will be used for selective charity causes.

Whale Protection

- ❖ Sell regulation: 90 sec buy and sell delay to stop bots.
- ❖ Max transaction amount of 0.5% of the total supply.

The future plan

The predator will launch its own gym-wear and fitness clothing collections, introduce crypto pay-TV, and set up an NFT marketplace and an NFT ticketing system for sports events.

PredNFT

PredNFT comprises two main projects:

NFT Collectibles

Users will be able to collect NFTs representing their favourite athletes and moments from selected sports.

We will then establish a platform where users can buy, sell, and trade their NFTs. Profits can be made by users, since athletes at the height of their careers will be in higher demand, and iconic moments will be more valuable.

Fees from this platform will be used to further the development of the project.

NFT Tickets

Since we are targeting extreme sports, we think a natual step is to help those sports grow. Therefore, Pred will be launching an NFT ticketing system to selected sports and events, allowing a wider audience to participate, as well as facilitating the ownership tracking to help fight ticket scalping.

Tickets will also come with unique perks which will be elaborated on at a later date.

PredTV

Many of our users will be sports fans...

So a natural step is to expand the Pred ecosystem to create a platform for our user to enjoy their favourite sports, with exclusive rewards.

PredTV will go hand-in-and with the PredNFT project and wil bring further adoption and marketing opportunities to the Predator project.

The launch of PredTV will be a significant milestone in the Predator project, and as such will also present a unique challenge.

At first, the sports and events available will be limited to a smaller selection, but this will expand with time.

We are excited to reveal more as time goes on.

Private Payment System

The technological themes of this decade are decentralisation and privacy.

Never before has it been so important to own and protect your data – both from hostile entities, and legal ones.

We believe that people should own their own data, and should have the right to financial privacy.

That is why, in 2022, we will begin developing and testing a private payment system based on the Predator ecosystem. Users will be able to confidently manage their assets without the fear of third-parties gaining access to their private data.

We will consult with cryptography and data security experts to ensure that we develop a robust, secure, and easy-to-use payment system which puts (and keeps) the user in control.

More information will be given as the project progresses.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	810
5	System quality	8/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	8/10
Total Points		**8.13/10**

Contract details

Token contract details for 22nd August 2021

Contract name	Predator Token
Contract address	0xe59046e1a4a83c11ccc578e26da4eeec8484ed8d
Token supply	1,000,000,000
Token ticker	PRED
Decimals	9
Token holders	864
Transaction count	6,319
Top 100% holders dominance	44.83%
Dividend Tracker	0x9c0270f3df9b26c366258f9f5291c629efc34162
Buyback wallet address	0xbde1c5ab2b603449e1ed1b6bffcd43c64834787d
Contract deployer address	0x4f09AF61a587e3d16aB6cF6389D0130dCB1b71e2
Contract's current owner address	0x4f09af61a587e3d16ab6cf6389d0130dcb1b71e2

Top token holders

Top 10 Token Holders

The top 10 holders collectively own 44.83% (448,311,718.90 Tokens) of Predator Token   Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 864

Predator Token Top 10 Token Holders
Source: BscScan.com

- 0x2d045410f002a95efcee67759a92518fa3fce677
- 0xdd2f504a5b94122921c1833d6522def3853e5ee9 (PancakeSwap V2: PRED 5)
- 0xca9905b03183ff153ff2fb0abb9c5c683e7866d3a
- 0x7ae55ea31ab4b030423f5cafa33a625c805babb2
- 0x58a6acec87fc4bb3547c69ae7f66af1015197fa8
- 0x78938e96dfcc96c6e57e92fba6678e1ba4cabed0
- 0x20c46aa4d397e308be810f7c57cf778f927fe067
- 0x81e4a7abb2c3cf304ac9371b5a2ed88fd52450f4
- 0x727731f1ab4a13b9caf992c220bf04be5aa27094

OTHER ACCOUNTS

(A total of 448,311,718.90 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x2d045410f002a95efcee67759a92518fa3fce677	197,052,000	19.7052%
2	PancakeSwap V2: PRED 5	102,928,557.230193216	10.2929%
3	0xca9905b03183ff153ff2fb0abb9c5c683e7866d3a	41,865,250	4.1865%
4	0x7ae55ea31ab4b030423f5cafa33a625c805babb2	24,993,750	2.4994%
5	0x58a6acec87fc4bb3547c69ae7f66af1015197fa8	17,834,341.710712126	1.7834%
6	0x78938e96dfcc96c6e57e92fba6678e1ba4cabed0	15,068,800	1.5069%
7	0x20c46aa4d397e308be810f7c57cf778f927fe067	14,969,209.627830042	1.4969%
8	0x81e4a7abb2c3cf304ac9371b5a2ed88fd52450f4	13,481,258.289209568	1.3481%
9	0x727731f1ab4a13b9caf992c220bf04be5aa27094	10,291,301.05	1.0291%
10	0x42db01e03d59641608422384be1a1cfe30aa1990	9,827,250.993261849	0.9827%

Token distribution

Tokens are distributed as follows:

- ❖ 15% Private sale for early investors and partners that funded the project
- ❖ 17.8% Burn wallet (locked and will be burned gradually)
- ❖ 5% Marketing and development wallet
- ❖ 62.2% public funds (DxSale) split into:
- ❖ 30% Presale (Whitelist)
- ❖ 25% Public sale (PancakeSwap)
- ❖ 7.2% Dxsale burn

Top 100 Token Holders

The top 100 holders collectively own 78.22% (782,167,718.30 Tokens) of Predator Token | Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 864

Predator Token Top 100 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

0x2d045410f002a95efcee67759a92518fa3fce677

0xdd2f504a5b94122921c1833d6522def3853e5ee9
(PancakeSwap V2: PRED 5)

0xca9905b03183f153ff2fb0abb9c5c683e7866d3a

0x7ae55ea31ab4b030423f5cafa33a625c805babb2

0x58a6acec87fc4bb3547c69ae7f66af1015197fa8

0x78938e96dfcc96c6e57e92fba6678e1ba4cabed0

0x20c46aa4d397e308be810f7c57cf778f927fe067

0x81e4a7abb2c3cf304ac9371b5a2ed88fd52450f4

0x727731f1ab4a13b9caf992c220bf04be5aa27094

0x0e23840f0dc0fba9082f168dbabdc5b96b9eee1c

0x562219e7129cb4e4ceacc2041120d3b6aa9d3410

0x41555247d294be0ef26814ae3cd99f439c5b5719

(A total of 782,167,718.30 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

15

Contract interaction details
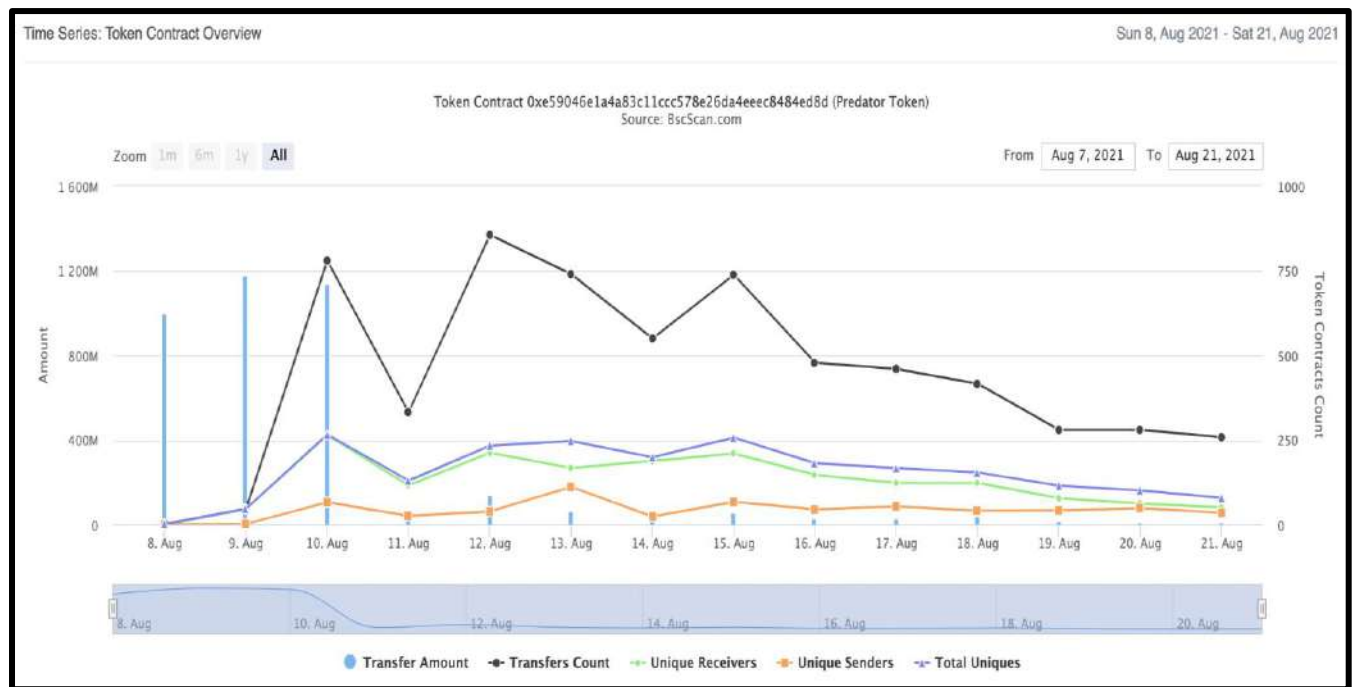
Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	low issue
		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	informational
		Business logics	pass
		Business implementations	low issue
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
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	**Implementation**	**Context**		
└		Public ❗	⬢	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	⬢	onlyOwner
└	transferOwnership	Public ❗	⬢	onlyOwner
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	⬢	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	⬢	NO❗
└	transferFrom	External ❗	⬢	NO❗
ERC20	**Implementation**	**Context, IERC20**		
└		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 🔒	●	
└	_setupDecimals	Internal 🔒	●	
└	_beforeTokenTransfer	Internal 🔒	●	
IDividendPayingToken	**Interface**			
└	dividendOf	External		NO
└	distributeDividends	External	🔲	NO
└	withdrawDividend	External	●	NO
IDividendPayingToken Optional	**Interface**			
└	withdrawableDividendOf	External		NO
└	withdrawnDividendOf	External		NO
└	accumulativeDividendOf	External		NO

DividendPayingToken	Implementation	ERC20, IDividendPayingToken , IDividendPayingToken Optional		
L		Public 〚	⬟	ERC20
L		External 〚	⧈	NO〚
L	distributeDividends	Public 〚	⧈	NO〚
L	distributeDividends	Public 〚	⬟	NO〚
L	withdrawDividend	Public 〚	⬟	NO〚
L	setDividendTokenAddress	Public 〚	⬟	NO〚
L	_withdrawDividendOfUser	Internal 🔒	⬟	
L	dividendOf	Public 〚		NO〚
L	withdrawableDividendOf	Public 〚		NO〚
L	withdrawnDividendOf	Public 〚		NO〚
L	accumulativeDividendOf	Public 〚		NO〚
L	_transfer	Internal 🔒	⬟	
L	_mint	Internal 🔒	⬟	
L	_burn	Internal 🔒	⬟	
L	_setBalance	Internal 🔒	⬟	
IUniswapV2Factory	**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〚

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⬚
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	mint	External ⬚	⬤	NO⬚
L	burn	External ⬚	⬤	NO⬚
L	swap	External ⬚	⬤	NO⬚
L	skim	External ⬚	⬤	NO⬚
L	sync	External ⬚	⬤	NO⬚
L	initialize	External ⬚	⬤	NO⬚

IUniswapV2Router01	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▯
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ▯	●	NO▯
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ▯	●	NO▯
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ▯	●	NO▯

L	swapExactETHFor TokensSupporting FeeOnTransferTo kens	External		NO
L	swapExactTokens ForETHSupporting FeeOnTransferTo kens	External	⬢	NO
IterableMapping	**Library**			
L	get	Public		NO
L	getIndexOfKey	Public		NO
L	getKeyAtIndex	Public		NO
L	size	Public		NO
L	set	Public	⬢	NO
L	remove	Public	⬢	NO
SafeMath	**Library**			
L	tryAdd	Internal 🔒		
L	trySub	Internal 🔒		
L	tryMul	Internal 🔒		
L	tryDiv	Internal 🔒		
L	tryMod	Internal 🔒		
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	sub	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		

SafeMathInt	Library			
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	sub	Internal 🔒		
L	add	Internal 🔒		
L	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
L	toInt256Safe	Internal 🔒		
PredatorToken	Implementation	ERC20, Ownable		
L		Public ▮	⬤	ERC20
L		External ▮	▦	NO▮
L	whitelistDxSale	Public ▮	⬤	onlyOwner
L	setMaxBuyTransaction	External ▮	⬤	onlyOwner
L	setMaxSellTransaction	External ▮	⬤	onlyOwner
L	setMaxWalletToken	External ▮	⬤	onlyOwner
L	setSellTransactionMultiplier	External ▮	⬤	onlyOwner
L	setMarketingDivisor	External ▮	⬤	onlyOwner
L	prepareForPreSale	External ▮	⬤	onlyOwner
L	afterPreSale	External ▮	⬤	onlyOwner
L	setTradingIsEnabled	Public ▮	⬤	onlyOwner
L	setBuyBackEnabled	Public ▮	⬤	onlyOwner
L	setBuyBackRandomEnabled	Public ▮	⬤	onlyOwner
L	triggerBuyBack	Public ▮	⬤	onlyOwner
L	updateDividendTracker	Public ▮	⬤	onlyOwner
L	updateDividendRewardFee	Public ▮	⬤	onlyOwner
L	updateMarketingFee	Public ▮	⬤	onlyOwner

L	updateUniswapV2Router	Public ◖	⬢	onlyOwner
L	excludeFromFees	Public ◖	⬢	onlyOwner
L	excludeMultipleAccountsFromFees	Public ◖	⬢	onlyOwner
L	setAutomatedMarketMakerPair	Public ◖	⬢	onlyOwner
L	_setAutomatedMarketMakerPair	Private 🔐	⬢	
L	updateBuyBackWallet	Public ◖	⬢	onlyOwner
L	updateGasForProcessing	Public ◖	⬢	onlyOwner
L	updateClaimWait	External ◖	⬢	onlyOwner
L	getClaimWait	External ◖		NO◖
L	getTotalDividendsDistributed	External ◖		NO◖
L	isExcludedFromFees	Public ◖		NO◖
L	withdrawableDividendOf	Public ◖		NO◖
L	dividendTokenBalanceOf	Public ◖		NO◖
L	getAccountDividendsInfo	External ◖		NO◖
L	getAccountDividendsInfoAtIndex	External ◖		NO◖
L	processDividendTracker	External ◖	⬢	NO◖
L	claim	External ◖	⬢	NO◖
L	getLastProcessedIndex	External ◖		NO◖
L	rand	Public ◖		NO◖
L	getNumberOfDividendTokenHolders	External ◖		NO◖
L	isBlackListed	Public ◖		NO◖
L	blacklistUpdate	Public ◖	⬢	onlyOwner
L	_transfer	Internal 🔒	⬢	
L	swapTokensForBNB	Private 🔐	⬢	
L	swapBNBForTokens	Private 🔐	⬢	
L	swapTokensForDividendToken	Private 🔐	⬢	
L	swapAndSendDividends	Private 🔐	⬢	
L	swapAndSendDividendsInBNB	Private 🔐	⬢	

PredatorTokenDividend Tracker	Implementation	DividendPayingToken, Ownable		
L	transferToBuyBackWallet	Private 🔐	⬤	
PredatorTokenDividend Tracker	**Implementation**	**DividendPayingToken, Ownable**		
L		Public ⫯	⬤	DividendPaying Token
L	_transfer	Internal 🔒		
L	withdrawDividend	Public ⫯		NO⫯
L	excludeFromDividends	External ⫯	⬤	onlyOwner
L	updateClaimWait	External ⫯	⬤	onlyOwner
L	getLastProcessedIndex	External ⫯		NO⫯
L	getNumberOfTokenHolders	External ⫯		NO⫯
L	getAccount	Public ⫯		NO⫯
L	getAccountAtIndex	Public ⫯		NO⫯
L	canAutoClaim	Private 🔐		
L	setBalance	External ⫯	⬤	onlyOwner
L	process	Public ⫯	⬤	NO⫯
L	processAccount	Public ⫯	⬤	onlyOwner

Legend

Symbol	Meaning
🔴	**Function can modify state**
🔡	**Function is payable**

Inheritance Hierarchy
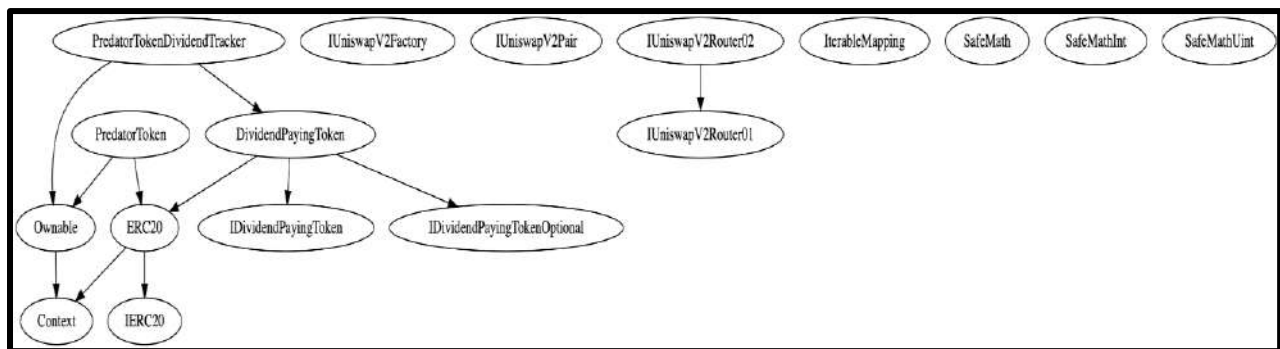
Security issue checking status

❖ **High severity issues**

- **No high severity issues found.**

❖ **Medium severity issues**

- **No medium severity issues found**.

❖ **Low severity issues**

- **Misleading variable (buyBackWallet)**

```
);
swapTokensForBNB(swapTokens);
transferToBuyBackWallet(
    payable(buyBackWallet),
    address(this).balance.div(10**2).mul(marketingDivisor)
);
```

Using the variable name as buyBackWallet for the marketing wallet, it should be renamed as marketing wallet or a similar one.

- **Liquidity is not getting added to the liquidity pool against all the trades, which might be a reason to lead to a market and token price imbalance. Owners can manually add liquidity from time to time but, still not a good feasible solution.**

- **Out of gas issue**

```
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↑);
}
```

In the excludeMultipleAccountsFromFees function, if they use a long wallet list there can be an OUT_OF_GAS issue, better to use a small array list at once.

❖ **Informational**

- **The owner can enable and disable trading at any time.**

```
ftrace | funcSig
function setTradingIsEnabled(bool _enabled↑) public onlyOwner {
    tradingIsEnabled = _enabled↑;
    emit SwapAndLiquifyEnabledUpdated(_enabled↑);
}
```

Suggestion:
Function should change for the owner to only enable the trading not to disable.

Owner privileges

❖ The owner can change the max buying and selling amount.

```
ftrace | funcSig
function setMaxBuyTransaction(uint256 maxTxn↑) external onlyOwner {
    maxBuyTranscationAmount = maxTxn↑ * (10**9);
}


ftrace | funcSig
function setMaxSellTransaction(uint256 maxTxn↑) external onlyOwner {
    maxSellTransactionAmount = maxTxn↑ * (10**9);
}
```

❖ The owner can change the max wallet hold amount.

```
ftrace | funcSig
function setMaxWalletToken(uint256 maxToken↑) external onlyOwner {
    maxWalletToken = maxToken↑ * (10**9);
}
```

❖ The owner can change the sell fee up to 2X.

```
ftrace | funcSig
function setSellTransactionMultiplier(uint256 multiplier↑)
    external
    onlyOwner
{
    require(
        sellFeeIncreaseFactor >= 100 && sellFeeIncreaseFactor <= 200,
        "PredatorToken: Sell transaction multipler must be between 100 (1x) and 200 (2x)"
    );
    sellFeeIncreaseFactor = multiplier↑;
}
```

❖ The owner can change marketing divisor up to 100%.

```
ftrace | funcSig
function setMarketingDivisor(uint256 divisor↑) external onlyOwner {
    require(
        marketingDivisor >= 0 && marketingDivisor <= 100,
        "PredatorToken: Marketing divisor must be between 0 (0%) and 100 (100%)"
    );
    sellFeeIncreaseFactor = divisor↑;
}
```

❖ The owner can enable/disable trading.

```
ftrace | funcSig
function setTradingIsEnabled(bool _enabled↑) public onlyOwner {
    tradingIsEnabled = _enabled↑;
    emit SwapAndLiquifyEnabledUpdated(_enabled↑);
}
```

❖ The owner can enable/disable buy back.

```
ftrace | funcSig
function setBuyBackEnabled(bool _enabled↑) public onlyOwner {
    buyBackEnabled = _enabled↑;
    emit BuyBackEnabledUpdated(_enabled↑);
}
```

❖ The owner can manually run the buy back.

```
ftrace | funcSig
function triggerBuyBack(uint256 amount↑) public onlyOwner {
    require(
        !swapping,
        "PredatorToken: A swapping process is currently running, wait till that is complete"
    );

    uint256 buyBackBalance = address(this).balance;
    swapBNBForTokens(buyBackBalance.div(10**2).mul(amount↑));
}
```

❖ The owner can update the dividend coin.

```
ftrace | funcSig
function updateDividendTracker(address newAddress↑) public onlyOwner {
    require(
        newAddress↑ != address(dividendTracker),
        "PredatorToken: The dividend tracker already has that address"
    );

    PredatorTokenDividendTracker newDividendTracker = PredatorTokenDividendTracker(
            payable(newAddress↑)
        );

    require(
        newDividendTracker.owner() == address(this),
        "PredatorToken: The new dividend tracker must be owned by the FLOKIBUSD token contract"
    );

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));

    emit UpdateDividendTracker(newAddress↑, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```

❖ The owner can update reward fee and marketing fee up to 10%.

```
ftrace | funcSig
function updateDividendRewardFee(uint8 newFee↑) public onlyOwner {
    require(
        newFee↑ >= 0 && newFee↑ <= 10,
        "PredatorToken: Dividend reward tax must be between 0 and 10"
    );
    dividendRewardsFee = newFee↑;
}

ftrace | funcSig
function updateMarketingFee(uint8 newFee↑) public onlyOwner {
    require(
        newFee↑ >= 0 && newFee↑ <= 10,
        "PredatorToken: Dividend reward tax must be between 0 and 10"
    );
    marketingFee = newFee↑;
}
```

❖ The owner can change v2 router address.

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

❖ The owner can exclude accounts from the fee.

```
ftrace | funcSig
function excludeFromFees(address account↑, bool excluded↑) public onlyOwner {
    require(
        isExcludedFromFees[account↑] != excluded↑,
        "PredatorToken: Account is already the value of 'excluded'"
    );
    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 change the buy back wallet.

```
ftrace | funcSig
function updateBuyBackWallet(address newBuyBackWallet↑) public onlyOwner {
    require(
        newBuyBackWallet↑ != buyBackWallet,
        "PredatorToken: The liquidity wallet is already this address"
    );
    excludeFromFees(newBuyBackWallet↑, true);
    buyBackWallet = newBuyBackWallet↑;
    emit BuyBackWalletUpdated(newBuyBackWallet↑, buyBackWallet);
}
```

❖ The owner can change the claim wait.

```
ftrace | funcSig
function updateClaimWait(uint256 claimWait↑) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait↑);
}
```

Audit conclusion

While conducting the audit of the Predator token smart contract, it was observed that there is nothing alarming with the code and the contract contains only low severity issues.