SMART CONTRACT SECURITY AUDIT OF
Rocket Raccoon

Summary

Auditing Firm	InterFi Network
Architecture	InterFi "Echelon" Auditing Standard
Smart Contract Audit Approved By	Chris | Blockchain Specialist at InterFi Network
Project Overview Approved BY	Albert | Project Specialist at InterFi Network
Platform	Solidity
Audit Check (Mandatory)	Static, Software, Auto Intelligent & Manual Analysis
Project Check (Optional)	KYC, Website & Socials Analysis (Not Applicable)
Consultation Request Date	October 23, 2021
Report Date	October 26, 2021

Audit Summary

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

❖ **Rocket Raccoon's smart contract source code has <u>MEDIUM RISK SEVERITY.</u>**

❖ **Rocket Raccoon has PASSED the smart contract audit.**

For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit.

Table Of Contents

Project Overview

InterFi was consulted by Rocket Raccoon on October 23, 2021 to conduct a smart contract security audit of their token source code.

Rocket Raccoon automatically rewards $BTC to holders!

Project	Rocket Raccoon
Blockchain	Binance Smart Chain
Language	Solidity
Contract	0xFC111b40aD299572f74F1C119c036508C621BB19
Website	https://www.rocketraccoon.finance/
Telegram	https://t.me/RocketRaccoon_Chat
Twitter	https://twitter.com/RocketR_BTC

Public logo

Solidity Source Code On Blockchain (EtherScan Verified Contract Source Code)

https://bscscan.com/address/0xfc111b40ad299572f74f1c119c036508c621bb19#code

Rocket Raccoon's contract is generated using PinkSale's token generator

https://www.pinksale.finance/#/launchpad/token/create?chain=BSC-Test

Contract Name: Rocket Raccoon

Compiler Version: v0.7.6+commit.7338295f

Optimization Enabled: Yes with 200 runs

Solidity Source Code On InterFi GitHub

https://github.com/interfinetwork/audited-codes/blob/main/RocketRaccoon.sol

GitHub Commits

Solidity source code committed at: b0f77c609285678bcdd44a0cee52ae50f3ffc511

Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of Rocket Raccoon. The source code can be viewed in its entirety on

https://bscscan.com/address/0xfc111b40ad299572f74f1c119c036508c621bb19#code

InterFi has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category

Smart Contract Vulnerabilities	❖ Re-entrancy (RE) ❖ Unhandled Exceptions (UE) ❖ Transaction Order Dependency (TO) ❖ Integer Overflow (IO) ❖ Unrestricted Action (UA)
Source Code Review	❖ Ownership Takeover ❖ Gas Limit and Loops ❖ Deployment Consistency ❖ Repository Consistency ❖ Data Consistency ❖ Token Supply Manipulation
Functional Assessment	❖ Access Control and Authorization ❖ Operations Trail and Event Generation ❖ Assets Manipulation ❖ Liquidity Access

InterFi's Echelon Audit Standard

The aim of InterFi's "Echelon" standard is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by ECHELON-1 to assess the smart contract:

1. Solidity smart contract source code reviewal:
   ❖ Review of the specifications, sources, and instructions provided to InterFi to make sure we understand the size, scope, and functionality of the smart contract.
   ❖ Manual review of code, which is the process of reading source code line-byline to identify potential vulnerabilities.

2. Static, Manual, and Automated AI analysis:
   ❖ Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
   ❖ Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts

Automated 3P frameworks used to assess the smart contract vulnerabilities

❖ Slither

❖ Consensys MythX

❖ Consensys Surya

❖ Open Zeppelin Code Analyzer

❖ Solidity Code Complier

InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in ETH/BNB. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

Vulnerable: A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false-positive.

Exploitable: A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the "vulnerability" flagged by a tool is in a function which requires to own the contract, it would be vulnerable but not exploitable.

Exploited: A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.

| Risk severity	Meaning
! Critical	This level vulnerabilities could be exploited easily, and can lead to asset loss, data loss, asset manipulation, or data manipulation. They should be fixed right away.
! High	This level vulnerabilities are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to critical risk severity
! Medium	This level vulnerabilities are should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution.
! Low	This level vulnerabilities can be ignored. They are code style violations, and informational statements in the code. They may not affect the smart contract execution

Smart Contract – Overview

Contract information

Query	Result
Name	Rocket Raccoon
Symbol	ROC
Decimals	18
Total Supply	100,000,000,000,000
Total Fees	14
Owner	0x7980af1494b801895c45b84533596f3f58df152f
BTCB Token	0x7130d2a12b9bcbfae4f2634d864a1ee1ce3ead9c
Marketing Wallet	0x8fca876aae3396458190a0e6d5a56c3a59d074ac
Dividend Tracker	0x428136777363ecc5702ae9d664bbd5bb39b4854d
Implementation	0xf70a81d3f89025cfce1dcd98bd96091986d5f447
Pair	0xbfe27e64b7b11669b74cc99630cebf048e8b65c6
Router	0x10ed43c718714eb63d5aa57b78b54704e256024e

Smart Contract – Static Analysis

Symbol	Meaning
🛑	Function can be modified
💵	Function is payable
🔒	Function is locked
🔓	Function can be accessed
❗	Important functionality

```
| **Context** | Implementation |   |||
|  └ | _msgSender | Internal 🔒 |    | |
|  └ | _msgData | Internal 🔒 |    | |
||||||
| **IERC20** | Interface |   |||
|  └ | totalSupply | External ❗ |    |NO❗ |
|  └ | balanceOf | External ❗ |    |NO❗ |
|  └ | transfer | External ❗ | 🛑  |NO❗ |
|  └ | allowance | External ❗ |    |NO❗ |
|  └ | approve | External ❗ | 🛑  |NO❗ |
|  └ | transferFrom | External ❗ | 🛑  |NO❗ |
||||||
| **SafeMath** | Library |   |||
|  └ | tryAdd | Internal 🔒 |    | |
|  └ | trySub | Internal 🔒 |    | |
|  └ | tryMul | Internal 🔒 |    | |
|  └ | tryDiv | Internal 🔒 |    | |
|  └ | tryMod | Internal 🔒 |    | |
|  └ | add | Internal 🔒 |    | |
|  └ | sub | Internal 🔒 |    | |
|  └ | mul | Internal 🔒 |    | |
|  └ | div | Internal 🔒 |    | |
|  └ | mod | Internal 🔒 |    | |
|  └ | sub | Internal 🔒 |    | |
|  └ | div | Internal 🔒 |    | |
|  └ | mod | Internal 🔒 |    | |
||||||
| **ERC20** | Implementation | Context, IERC20 |||
|  └ | <Constructor> | 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 🔒 | 🔴  | |
||||||
| **AddressUpgradeable** | Library |  |||
| └ | isContract | Internal 🔒 |    | |
| └ | sendValue | Internal 🔒 | 🔴  | |
| └ | functionCall | Internal 🔒 | 🔴  | |
| └ | functionCall | Internal 🔒 | 🔴  | |
| └ | functionCallWithValue | Internal 🔒 | 🔴  | |
| └ | functionCallWithValue | Internal 🔒 | 🔴  | |
| └ | functionStaticCall | Internal 🔒 |    | |
| └ | functionStaticCall | Internal 🔒 |    | |
| └ | _verifyCallResult | Private 🔐 |    | |
||||||
| **Initializable** | Implementation |  |||
| └ | _isConstructor | Private 🔐 |    | |
||||||
| **ContextUpgradeable** | Implementation | Initializable |||
| └ | __Context_init | Internal 🔒 | 🔴 | initializer |
| └ | __Context_init_unchained | Internal 🔒 | 🔴 | initializer |
| └ | _msgSender | Internal 🔒 |    | |
| └ | _msgData | Internal 🔒 |    | |
||||||
| **IERC20Upgradeable** | Interface |  |||
| └ | totalSupply | External ❗ |    |NO❗ |
| └ | balanceOf | External ❗ |    |NO❗ |
| └ | transfer | External ❗ | 🔴  |NO❗ |
| └ | allowance | External ❗ |    |NO❗ |
| └ | approve | External ❗ | 🔴  |NO❗ |
| └ | transferFrom | External ❗ | 🔴  |NO❗ |
||||||
| **SafeMathUpgradeable** | Library |  |||
| └ | tryAdd | Internal 🔒 |    | |
| └ | trySub | Internal 🔒 |    | |
| └ | tryMul | Internal 🔒 |    | |
| └ | tryDiv | Internal 🔒 |    | |
| └ | tryMod | Internal 🔒 |    | |
```

```
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **ERC20Upgradeable** | Implementation | Initializable, ContextUpgradeable, IERC20Upgradeable |||
| └ | __ERC20_init | Internal 🔒 | 🔴 | initializer |
| └ | __ERC20_init_unchained | Internal 🔒 | 🔴 | initializer |
| └ | 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 🔒 | 🔴 | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🔴 |NO❗ |
| └ | owner | Public ❗ |   |NO❗ |
| └ | renounceOwnership | Public ❗ | 🔴 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🔴 | onlyOwner |
||||||
| **OwnableUpgradeable** | Implementation | Initializable, ContextUpgradeable |||
| └ | __Ownable_init | Internal 🔒 | 🔴 | initializer |
| └ | __Ownable_init_unchained | Internal 🔒 | 🔴 | initializer |
| └ | owner | Public ❗ |   |NO❗ |
| └ | renounceOwnership | Public ❗ | 🔴 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🔴 | onlyOwner |
||||||
| **Clones** | Library |  |||
| └ | clone | Internal 🔒 | 🔴 | |
| └ | cloneDeterministic | Internal 🔒 | 🔴 | |
| └ | predictDeterministicAddress | Internal 🔒 |   | |
| └ | predictDeterministicAddress | Internal 🔒 |   | |
||||||
| **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❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🔴 |NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🔴 |NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | 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❗
SafeMathInt	Library			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
└	toInt256Safe	Internal 🔒		
IterableMapping	Library			
└	get	Public ❗		NO❗
└	getIndexOfKey	Public ❗		NO❗
└	getKeyAtIndex	Public ❗		NO❗
└	size	Public ❗		NO❗
└	set	Public ❗	🔴	NO❗
└	remove	Public ❗	🔴	NO❗
IBabyToken	Interface			
└	initialize	External ❗	🔴	NO❗
DividendPayingTokenInterface	Interface			
└	dividendOf	External ❗		NO❗
└	withdrawDividend	External ❗	🔴	NO❗
DividendPayingTokenOptionalInterface	Interface			
└	withdrawableDividendOf	External ❗		NO❗
└	withdrawnDividendOf	External ❗		NO❗
└	accumulativeDividendOf	External ❗		NO❗
DividendPayingToken	Implementation	ERC20Upgradeable, OwnableUpgradeable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
└	__DividendPayingToken_init	Internal 🔒	🔴	initializer

```
|  └ | distributeCAKEDividends | Public ❗ | 🔴 | onlyOwner |
|  └ | withdrawDividend | Public ❗ | 🔴 |NO❗ |
|  └ | _withdrawDividendOfUser | Internal 🔒 | 🔴 | |
|  └ | dividendOf | Public ❗ |  |NO❗ |
|  └ | withdrawableDividendOf | Public ❗ |  |NO❗ |
|  └ | withdrawnDividendOf | Public ❗ |  |NO❗ |
|  └ | accumulativeDividendOf | Public ❗ |  |NO❗ |
|  └ | _transfer | Internal 🔒 | 🔴 | |
|  └ | _mint | Internal 🔒 | 🔴 | |
|  └ | _burn | Internal 🔒 | 🔴 | |
|  └ | _setBalance | Internal 🔒 | 🔴 | |
||||||
| **BABYTOKEN** | Implementation | ERC20Upgradeable, OwnableUpgradeable, IBabyToken |||
|  └ | <Constructor> | Public ❗ | 🔴 |NO❗ |
|  └ | initialize | External ❗ | 🔴 | initializer |
|  └ | <Receive Ether> | External ❗ | 💲 |NO❗ |
|  └ | setSwapTokensAtAmount | External ❗ | 🔴 | onlyOwner |
|  └ | updateDividendTracker | Public ❗ | 🔴 | onlyOwner |
|  └ | updateUniswapV2Router | Public ❗ | 🔴 | onlyOwner |
|  └ | excludeFromFees | Public ❗ | 🔴 | onlyOwner |
|  └ | excludeMultipleAccountsFromFees | Public ❗ | 🔴 | onlyOwner |
|  └ | setMarketingWallet | External ❗ | 🔴 | onlyOwner |
|  └ | setTokenRewardsFee | External ❗ | 🔴 | onlyOwner |
|  └ | setLiquiditFee | External ❗ | 🔴 | onlyOwner |
|  └ | setMarketingFee | External ❗ | 🔴 | onlyOwner |
|  └ | setAutomatedMarketMakerPair | Public ❗ | 🔴 | onlyOwner |
|  └ | blacklistAddress | External ❗ | 🔴 | onlyOwner |
|  └ | _setAutomatedMarketMakerPair | Private 🔐 | 🔴 | |
|  └ | updateGasForProcessing | Public ❗ | 🔴 | onlyOwner |
|  └ | updateClaimWait | External ❗ | 🔴 | onlyOwner |
|  └ | getClaimWait | External ❗ |  |NO❗ |
|  └ | getTotalDividendsDistributed | External ❗ |  |NO❗ |
|  └ | isExcludedFromFees | Public ❗ |  |NO❗ |
|  └ | withdrawableDividendOf | Public ❗ |  |NO❗ |
|  └ | dividendTokenBalanceOf | Public ❗ |  |NO❗ |
|  └ | excludeFromDividends | External ❗ | 🔴 | onlyOwner |
|  └ | getAccountDividendsInfo | External ❗ |  |NO❗ |
|  └ | getAccountDividendsInfoAtIndex | External ❗ |  |NO❗ |
|  └ | processDividendTracker | External ❗ | 🔴 |NO❗ |
|  └ | claim | External ❗ | 🔴 |NO❗ |
|  └ | getLastProcessedIndex | External ❗ |  |NO❗ |
|  └ | getNumberOfDividendTokenHolders | External ❗ |  |NO❗ |
|  └ | _transfer | Internal 🔒 | 🔴 | |
|  └ | swapAndSendToFee | Private 🔐 | 🔴 | |
|  └ | swapAndLiquify | Private 🔐 | 🔴 | |
|  └ | swapTokensForEth | Private 🔐 | 🔴 | |
|  └ | swapTokensForCake | Private 🔐 | 🔴 | |
|  └ | addLiquidity | Private 🔐 | 🔴 | |
|  └ | swapAndSendDividends | Private 🔐 | 🔴 | |
||||||
```

Smart Contract – Software Analysis

Function Signatures

```
16279055  =>  isContract(address)
39509351  =>  increaseAllowance(address,uint256)
43509138  =>  div(int256,int256)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
884557bf  =>  tryAdd(uint256,uint256)
a29962b1  =>  trySub(uint256,uint256)
6281efa4  =>  tryMul(uint256,uint256)
736ecb18  =>  tryDiv(uint256,uint256)
38dc0867  =>  tryMod(uint256,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
f43f523a  =>  mod(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
b745d336  =>  div(uint256,uint256,string)
71af23e8  =>  mod(uint256,uint256,string)
06fdde03  =>  name()
95d89b41  =>  symbol()
313ce567  =>  decimals()
a457c2d7  =>  decreaseAllowance(address,uint256)
30e0789e  =>  _transfer(address,address,uint256)
4e6ec247  =>  _mint(address,uint256)
6161eb18  =>  _burn(address,uint256)
104e81ff  =>  _approve(address,address,uint256)
61e9edb2  =>  _setupDecimals(uint8)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
c21d36f3  =>  functionStaticCall(address,bytes)
dbc40fb9  =>  functionStaticCall(address,bytes,string)
18c2c6a2  =>  _verifyCallResult(bool,bytes,string)
45549c1f  =>  _isConstructor()
f08d647e  =>  __Context_init()
ab96f671  =>  __Context_init_unchained()
```

```
678bd718  =>  __ERC20_init(string,string)
46753fdb  =>  __ERC20_init_unchained(string,string)
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
0142eb11  =>  __Ownable_init()
5ce29e24  =>  __Ownable_init_unchained()
8124b78e  =>  clone(address)
b86b2ceb  =>  cloneDeterministic(address,bytes32)
93a7e711  =>  predictDeterministicAddress(address,bytes32,address)
360d0fad  =>  predictDeterministicAddress(address,bytes32)
017e7e58  =>  feeTo()
094b7415  =>  feeToSetter()
e6a43905  =>  getPair(address,address)
1e3dd18b  =>  allPairs(uint256)
574f2ba3  =>  allPairsLength()
c9c65396  =>  createPair(address,address)
f46901ed  =>  setFeeTo(address)
a2e74af6  =>  setFeeToSetter(address)
c45a0155  =>  factory()
ad5c4648  =>  WETH()
e8e33700  =>  addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256)
f305d719  =>  addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
baa2abde  =>  removeLiquidity(address,address,uint256,uint256,uint256,address,uint256)
02751cec  =>  removeLiquidityETH(address,uint256,uint256,uint256,address,uint256)
2195995c  =>
removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes3
2,bytes32)
ded9382a  =>
removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,uint8,bytes32,byt
es32)
38ed1739  =>  swapExactTokensForTokens(uint256,uint256,address[],address,uint256)
8803dbee  =>  swapTokensForExactTokens(uint256,uint256,address[],address,uint256)
7ff36ab5  =>  swapExactETHForTokens(uint256,address[],address,uint256)
4a25d94a  =>  swapTokensForExactETH(uint256,uint256,address[],address,uint256)
18cbafe5  =>  swapExactTokensForETH(uint256,uint256,address[],address,uint256)
fb3bdb41  =>  swapETHForExactTokens(uint256,address[],address,uint256)
ad615dec  =>  quote(uint256,uint256,uint256)
054d50d4  =>  getAmountOut(uint256,uint256,uint256)
85f8c259  =>  getAmountIn(uint256,uint256,uint256)
d06ca61f  =>  getAmountsOut(uint256,address[])
1f00ca74  =>  getAmountsIn(uint256,address[])
af2979eb  =>
removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,uint256)
5b0d5984  =>
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,address,u
int256,bool,uint8,bytes32,bytes32)
5c11d795  =>
swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
b6f9de95  =>  swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint256)
```

```
791ac947  =>
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
3644e515  =>  DOMAIN_SEPARATOR()
30adf81f  =>  PERMIT_TYPEHASH()
7ecebe00  =>  nonces(address)
d505accf  =>  permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
ba9a7a56  =>  MINIMUM_LIQUIDITY()
0dfe1681  =>  token0()
d21220a7  =>  token1()
0902f1ac  =>  getReserves()
5909c0d5  =>  price0CumulativeLast()
5a3d5493  =>  price1CumulativeLast()
7464fc3d  =>  kLast()
6a627842  =>  mint(address)
89afcb44  =>  burn(address)
022c0d9f  =>  swap(uint256,uint256,address,bytes)
bc25cf77  =>  skim(address)
fff6cae9  =>  sync()
485cc955  =>  initialize(address,address)
bbe93d91  =>  mul(int256,int256)
adefc37b  =>  sub(int256,int256)
a5f3c23b  =>  add(int256,int256)
1b5ac4b5  =>  abs(int256)
744f7c7d  =>  toUint256Safe(int256)
e823b9bf  =>  toInt256Safe(uint256)
268d8e2e  =>  get(Map,address)
b45dad3d  =>  getIndexOfKey(Map,address)
7596720f  =>  getKeyAtIndex(Map,uint256)
b1b533f3  =>  size(Map)
6b06f325  =>  set(Map,address,uint256)
0eac8729  =>  remove(Map,address)
578b0250  =>  initialize(address[4],string,string,uint256,uint256,uint256,uint256,uint256)
91b89fba  =>  dividendOf(address)
6a474002  =>  withdrawDividend()
a8b9d240  =>  withdrawableDividendOf(address)
aafd847a  =>  withdrawnDividendOf(address)
27ce0147  =>  accumulativeDividendOf(address)
ee9358e8  =>  __DividendPayingToken_init(address,string,string)
ba72a955  =>  distributeCAKEDividends(uint256)
373de4aa  =>  _withdrawDividendOfUser(address)
ab86e0a6  =>  _setBalance(address,uint256)
afa4f3b2  =>  setSwapTokensAtAmount(uint256)
88bdd9be  =>  updateDividendTracker(address)
65b8dbc0  =>  updateUniswapV2Router(address)
c0246668  =>  excludeFromFees(address,bool)
c492f046  =>  excludeMultipleAccountsFromFees(address[],bool)
5d098b38  =>  setMarketingWallet(address)
4ed080c7  =>  setTokenRewardsFee(uint256)
adefd90c  =>  setLiquiditFee(uint256)
625e764c  =>  setMarketingFee(uint256)
```

SOCIAL @interfinetwork WEB interfi.network

```
9a7a23d6  =>  setAutomatedMarketMakerPair(address,bool)
455a4396  =>  blacklistAddress(address,bool)
a7f7b36f  =>  _setAutomatedMarketMakerPair(address,bool)
871c128d  =>  updateGasForProcessing(uint256)
e98030c7  =>  updateClaimWait(uint256)
a26579ad  =>  getClaimWait()
30bb4cff  =>  getTotalDividendsDistributed()
4fbee193  =>  isExcludedFromFees(address)
6843cd84  =>  dividendTokenBalanceOf(address)
31e79db0  =>  excludeFromDividends(address)
ad56c13c  =>  getAccountDividendsInfo(address)
f27fd254  =>  getAccountDividendsInfoAtIndex(uint256)
700bb191  =>  processDividendTracker(uint256)
4e71d92d  =>  claim()
e7841ec0  =>  getLastProcessedIndex()
64b0f653  =>  getNumberOfDividendTokenHolders()
a210621e  =>  swapAndSendToFee(uint256)
173865ad  =>  swapAndLiquify(uint256)
b28805f4  =>  swapTokensForEth(uint256)
dc0e347c  =>  swapTokensForCake(uint256)
9cd441da  =>  addLiquidity(uint256,uint256)
818c19dc  =>  swapAndSendDividends(uint256)
cd6dc687  =>  initialize(address,uint256)
09bbedde  =>  getNumberOfTokenHolders()
fbcbc0f1  =>  getAccount(address)
5183d6fd  =>  getAccountAtIndex(uint256)
77fdb837  =>  canAutoClaim(uint256)
e30443bc  =>  setBalance(address,uint256)
ffb2c479  =>  process(uint256)
bc4c4b37  =>  processAccount(address,bool)
```

Inheritance Graph
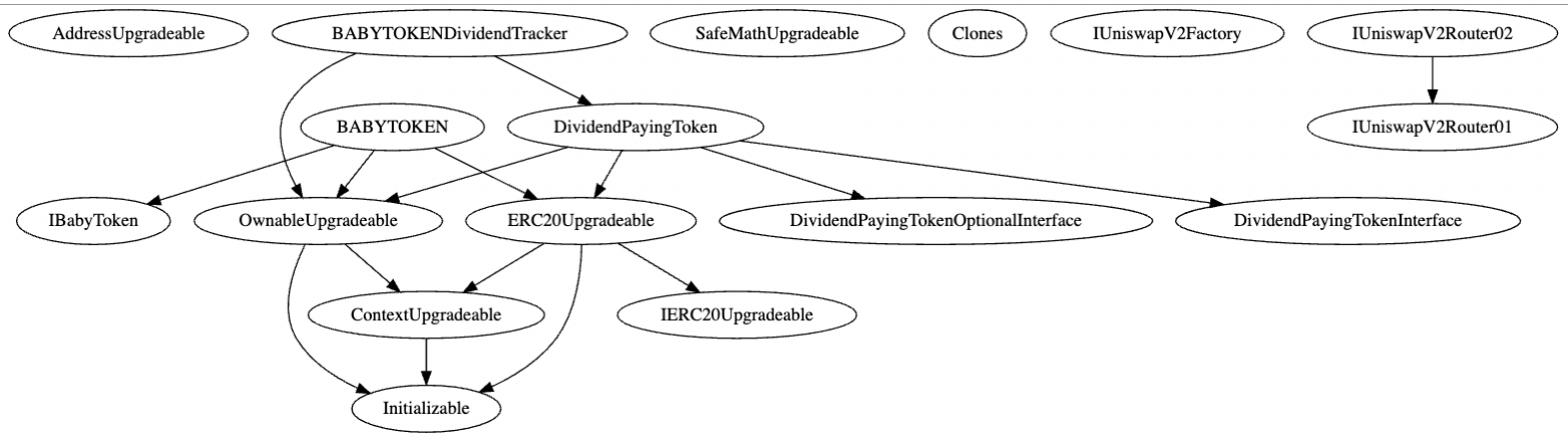
Smart Contract – Manual Analysis

Function	Description	Tested	Verdict
TotalSupply	provides information about the total token supply	Yes	**Passed**
BalanceOf	provides account balance of the owner's account	Yes	**Passed**
Transfer	executes transfers of a specified number of tokens to a specified address	Yes	**Passed**
Approve	allow a spender to withdraw a set number of tokens from a specified account	Yes	**Passed**
Allowance	returns a set number of tokens from a spender to the owner	Yes	**Passed**
burn	executes transfers of a specified number of tokens to a burn address	Yes	**Passed**

Note

- ❖ Active Owner: **0x7980af1494b801895c45b84533596f3f58df152f**
- ❖ When the smart contract has an active owner address, some of the smart contract functions can be edited, modified or altered.
- ❖ The smart contract uses blacklist function, the contract owner can stop any wallet address from trading the token. Trade with caution.
- ❖ Owner can mint tokens at token launch.
- ❖ Owner can-not lock or burn user assets.
- ❖ Owner can-not stop or pause the smart contract.

Important Information

1. **Rocket Raccoon Smart Contract utilizes "SafeMath" function to avoid common smart contract vulnerabilities.**

```solidity
library SafeMath {
function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    require(c >= a, 'SafeMath: addition overflow');

    return c;
}
function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    return sub(a, b, 'SafeMath: subtraction overflow');
}
uint256 c = a * b;
    require(c / a == b, 'SafeMath: multiplication overflow');

    return c;
}
```

2. **Rocket Raccoon smart contract has 1 low severity issue which may or may not create any functional vulnerability.**

```
{

    "resource": " /RocketRaccoon.sol",

    "owner": "_generated_diagnostic_collection_name_#0",

    "severity": 8, (! Low Severity)

    " Expected pragma, import directive or contract/interface/library definition",

    "source": "solc",

}
```

Smart Contract – SWC Attacks

| SWC ID	Description	Verdict
SWC-101	Integer Overflow and Underflow	Passed
SWC-102	Outdated Compiler Version	! Low
SWC-103	Floating Pragma	! Low
SWC-104	Unchecked Call Return Value	Passed
SWC-105	Unprotected Ether Withdrawal	Passed
SWC-106	Unprotected SELFDESTRUCT Instruction	Passed
SWC-107	Re-entrancy	Passed
SWC-108	State Variable Default Visibility	Passed
SWC-109	Uninitialized Storage Pointer	Passed
SWC-110	Assert Violation	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed
SWC-112	Delegate Call to Untrusted Callee	Passed
SWC-113	DoS with Failed Call	Passed
SWC-114	Transaction Order Dependence	Passed
SWC-115	Authorization through tx.origin	Passed
SWC-116	Block values as a proxy for time	Passed
SWC-117	Signature Malleability	Passed
SWC-118	Incorrect Constructor Name	Passed

SWC-119	Shadowing State Variables	Passed
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed
SWC-121	Missing Protection against Signature Replay Attacks	Passed
SWC-122	Lack of Proper Signature Verification	Passed
SWC-123	Requirement Violation	Passed
SWC-124	Write to Arbitrary Storage Location	Passed
SWC-125	Incorrect Inheritance Order	Passed
SWC-126	Insufficient Gas Griefing	Passed
SWC-127	Arbitrary Jump with Function Type Variable	Passed
SWC-128	DoS With Block Gas Limit	Passed
SWC-129	Typographical Error	Passed
SWC-130	Right-To-Left-Override control character (U+202E)	Passed
SWC-131	Presence of unused variables	Passed
SWC-132	Unexpected Ether balance	Passed
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed
SWC-134	Message call with hardcoded gas amount	Passed
SWC-135	Code With No Effects (Irrelevant/Dead Code)	! Low
SWC-136	Unencrypted Private Data On-Chain	Passed

Smart Contract - Risk Status & Radar Chart

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	**1 medium severity issue identified (Wallet Blacklist)**
! Low	**1 low severity issue identified**
Passed	**40 functions and instances verified and passed**
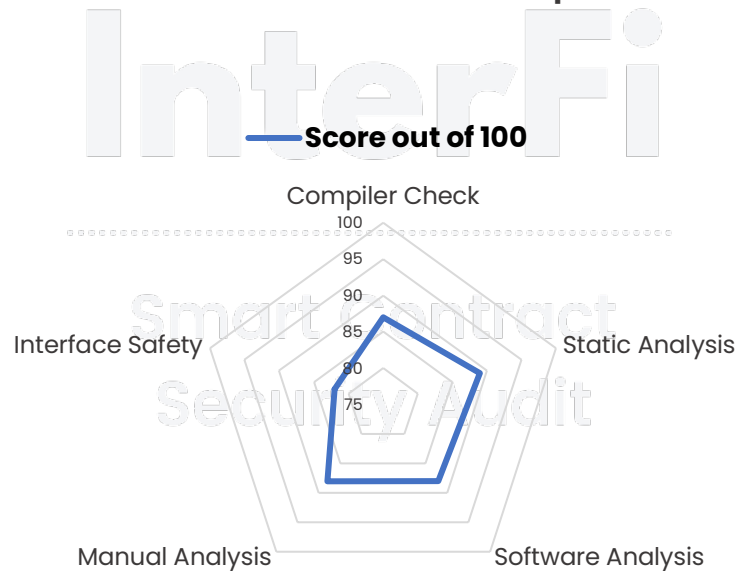
Compiler Check	87
Static Analysis	89
Software Analysis	88
Manual Analysis	88
Interface Safety	82

Auditor's Verdict

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

Rocket Raccoon's smart contract source code has <u>MEDIUM RISK SEVERITY.</u>

Rocket Raccoon has PASSED the smart contract audit.

Auditor's Note:

* ❖ Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security.

* ❖ The smart contract uses blacklist function, the contract owner can stop any wallet address from trading the token. Trade with caution.

* ❖ Owner or developer KYC isn't checked and verified due to out of scope.

* ❖ Project's liquidity pair isn't checked and verified due to out of scope.

* ❖ Project website is not checked due to out of scope. The website hasn't been reviewed for SSL and lighthouse report.

Important Disclaimer

InterFi Network provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project. **This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purposes without InterFi's prior written consent.**

InterFi provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as an enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. **Be aware that smart contracts deployed on a blockchain aren't resistant from external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, InterFi does not guarantee the explicit security of the audited smart contract.**

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.

This report should not be considered as an endorsement or disapproval of any project or team. The information provided on this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Do conduct your own due diligence and consult your financial advisor before making any investment decisions.

About InterFi Network

InterFi Network provides intelligent blockchain solutions. InterFi is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, Token Launchpad, NFT Marketplace, etc. **InterFi's mission is to interconnect multiple services like Blockchain Security, DeFi, Gaming, and Marketplace under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy-to-use.**

InterFi is built by a decentralized team of UI experts, contributors, engineers, and enthusiasts from all over the world. Our team currently consists of 6+ core team members, and 10+ casual contributors. **InterFi provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.**

To learn more, visit https://interfi.network

To view our audit portfolio, visit https://github.com/interfinetwork

To book an audit, message https://t.me/interfiaudits

@INTERFINETWORK

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN | MADE IN CANADA 🇨🇦