CONTRACT WOLF

Blockchain Security - Smart Contract Audits

# Security Assessment

May 19, 2022

FREE FINANCE

# Disclaimer

**ContractWolf.io** audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

**ContractWolf** does not provide any warranty on its released reports.

**ContractWolf** should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

**ContractWolf** provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within it's **SMART CONTRACT**.

**ContractWolf** presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

# Scope of Work

**Free Finance's** team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

**ContractWolf** will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **Free Finance.**

## Network

Binance Smart Chain (BEP20)

## Contract link

https://bscscan.com/address/0xa2a471007C8955fc18813BFd7DfE15b2aa2A6BE5

## Website

https://freefinancetoken.com/

## Telegram

https://t.me/FreeFinancetoken

## Twitter

https://twitter.com/freefinancebusd

# Description

**Free Finance** is a revolutionary passive income reward token that focuses on ensuring every holder enjoys a generous BUSD reward collected from each of the buy and sell transactions. In total this results in a BUSD rewards payout across both the buys and sells. **Free Finance's** number one priority above anything else is to sustain a high-volume ecosystem of constantly growing investors, enabling the Free Finance reward system to provide passive income for all its holders.

# Logo

# Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.
Risk Level is computed based on CVSS Version 3.0

| Level | Value | Vulnerability |
|---|---|---|
| Critical | 9 - 10 | An Exposure that can affect the contract functions in several events that can risk and disrupt the contract |
| High | 7 - 8.9 | An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner |
| Medium | 4 - 6.9 | An opening that could affect the outcome in executing the contract in a specific situation |
| Low | 0.1 - 3.9 | An opening but doesn't have an impact on the functionality of the contract |
| Informational | 0 | An opening that consists of information's but will not risk or affect the contract |

# Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

# Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

    - Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
    - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

    - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

# Used Code from other Frameworks/Smart Contracts (Direct Imports)

## Imported Packages

- IERC20
- IERC20Metadata
- Context
- Ownable
- SafeMath
- SafeMathInt
- SafeMathUint
- DividendPayingTokenInterface
- DividendPayingTokenOptionalInterface
- DividendPayingToken
- DividendTracker
- IUniswapV2Router01
- IUniswapV2Router02
- IUniswapV2Factory
- ERC20
- IterableMapping
- FreeFinance

# Description

Optimization enabled: Yes

Decimal: 18

Symbol: FFToken

Max / Total supply: 1,000,000,000

# Capabilities

## Components

| Version | Contracts | Libraries | Interfaces | Abstract |
|---|---|---|---|---|
| 1.0 | 5 | 4 | 7 | 1 |

## Exposed Functions

| Version | Public | Private | External | Internal |
|---|---|---|---|---|
| 1.0 | 39 | 6 | 84 | 27 |

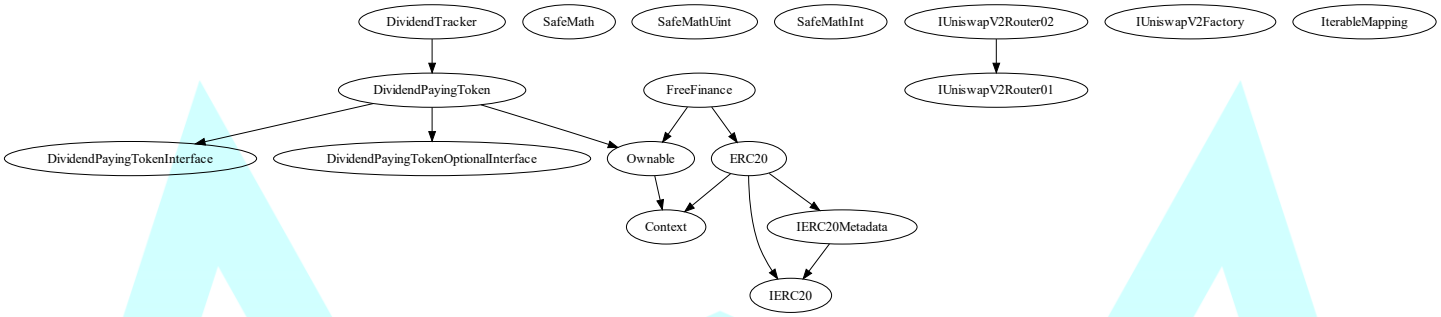
## State Variables

| Version | Total | Public |
|---|---|---|
| 1.0 | 53 | 41 |

## Capabilities

| Version | Solidity Versions Observed | Experimental Features | Can Receive Funds | Uses Assembly | Has Destroyable Contracts |
|---|---|---|---|---|---|
| 1.0 | v0.8.9 | | Yes | No | No |

# Inheritance Graph



## Correct implementation of Token Standard

| Tested | Verified |
|---|---|
| ✓ | ✓ |

## Overall Checkup (Smart Contract Security)

| Tested | Verified |
|---|---|
| ✓ | ✓ |

| Function | Description | Exist | Tested | Verified |
|---|---|---|---|---|
| TotalSupply | Information about the total coin or token supply | ✓ | ✓ | ✓ |
| BalanceOf | Details on the account balance from a specified address | ✓ | ✓ | ✓ |
| Transfer | An action that transfers a specified amount of coin or token to a specified address | ✓ | ✓ | ✓ |
| TransferFrom | An action that transfers a specified amount of coin or token from a specified address | ✓ | ✓ | ✓ |
| Approve | Provides permission to withdraw specified number of coin or token from a specified address | ✓ | ✓ | ✓ |

# Verify Claims

| Statement | Exist | Tested | Deployer |
|---|---|---|---|
| Renounce Ownership | ✓ | ✓ | ✓ |
| Mint | ✓ | ✓ | ✗ |
| Burn | ✓ | ✓ | ✗ |
| Block | ✓ | ✓ | ✓ |
| Pause | – | – | – |

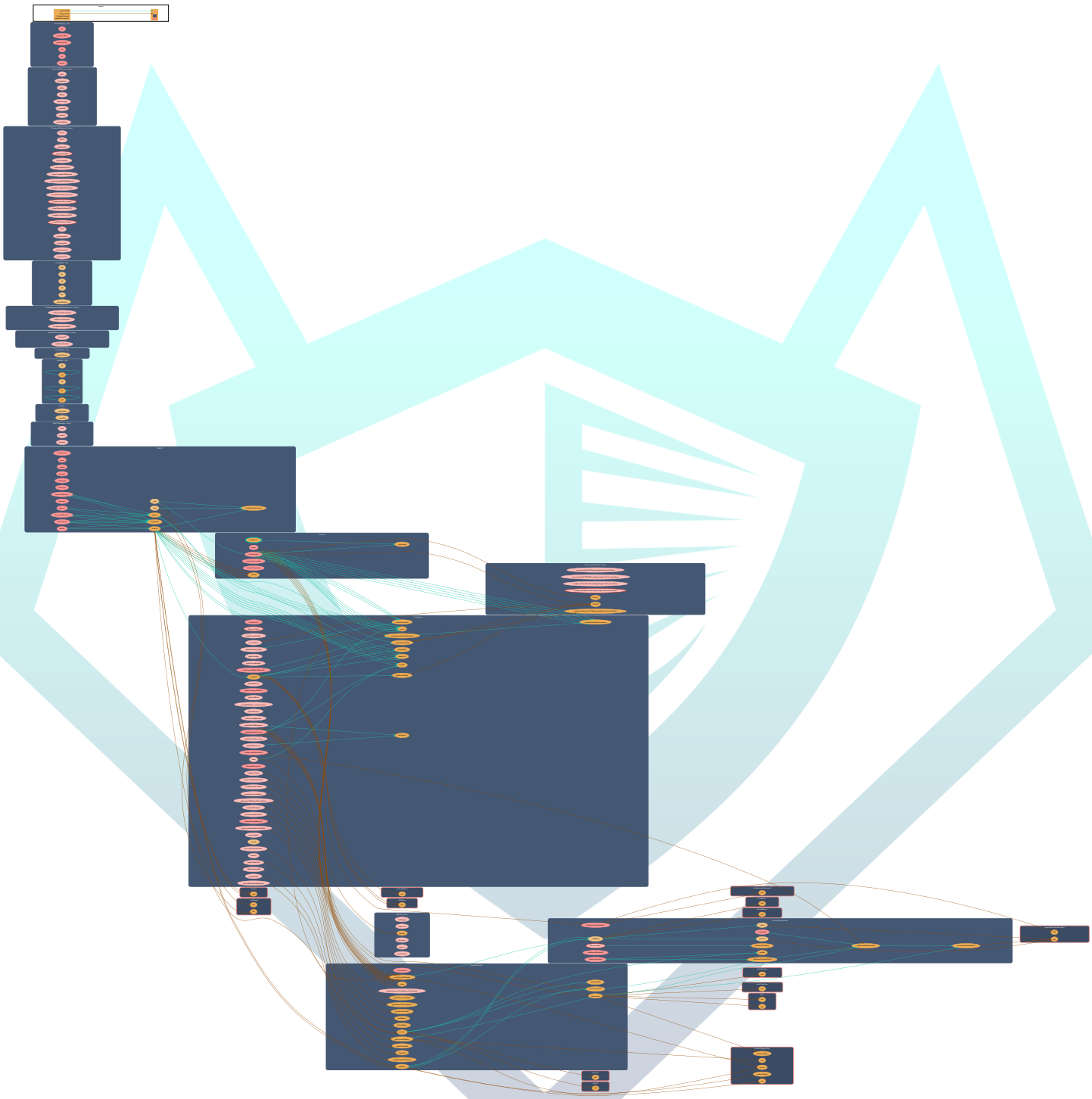
Legend

| Attribute | Symbol |
|---|---|
| Verified / Can | ✓ |
| Verified / Cannot | ✗ |
| Unverified / Not checked | ⚑ |
| Not Available | – |

# Write Functions of Contract

1. addToBlackList
2. approve
3. claim
4. decreaseAllowance
5. disableTransferDelay
6. enableTrading
7. excludeFromDividends
8. excludeFromFees
9. excludeFromMaxTransaction
10. excludeMultipleAccountsFromFees
11. includeInDividends
12. increaseAllowance
13. processDividendTracker
14. removeFromBlackList
15. removeLimits
16. renounceOwnership
17. rescueTokens
18. resetFee
19. setAutomatedMarketMakerPair
20. transfer
21. transferFrom
22. transferOwnership
23. updateBuyFees
24. updateClaimWait
25. updateDividendTracker
26. updateGasForProcessing
27. updateMarketingWallet
28. updateMaxAmount
29. updateMaxWalletAmount
30. updateSellFees
31. updateSwapEnabled
32. updateUniswapV2Router
33. withdrawStuckEth
34. withdrawToken

# Call Graph



# SWC Attacks

| ID | Title | Status |
|---|---|---|
| SWC-136 | Unencrypted Private Data On-Chain | PASSED |
| SWC-135 | Code With No Effects | PASSED |
| SWC-134 | Message call with hardcoded gas amount | PASSED |
| SWC-133 | Hash Collisions with Multiple Variable Length Arguments | PASSED |
| SWC-132 | Unexpected Ether balance | PASSED |
| SWC-131 | Presence of unused variables | PASSED |
| SWC-130 | Right-To Left Override control character (U+202E) | PASSED |
| SWC-129 | Typographical Error | PASSED |
| SWC-128 | DoS With Block Gas Limit | PASSED |
| SWC-127 | Arbitrary Jump with Function Type Variable | PASSED |
| SWC-126 | Insufficient Gas Griefing | PASSED |
| SWC-125 | Incorrect Inheritance Order | PASSED |
| SWC-124 | Write to Arbitrary Storage Location | PASSED |
| SWC-123 | Requirement Violation | PASSED |
| SWC-122 | Lack of Proper Signature Verification | PASSED |
| SWC-121 | Missing Protection against Signature Replay Attacks | PASSED |
| SWC-120 | Weak Sources of Randomness from Chain Attributes | LOW ISSUE |
| SWC-119 | Shadowing State Variables | PASSED |
| SWC-118 | Incorrect Constructor Name | PASSED |
| SWC-117 | Signature Malleability | PASSED |
| SWC-116 | Block values as a proxy for time | PASSED |
| SWC-115 | Authorization through tx.origin | PASSED |
| SWC-114 | Transaction Order Dependence | PASSED |
| SWC-113 | DoS with Failed Call | PASSED |
| SWC-112 | Delegate call to Untrusted Callee | PASSED |
| SWC-111 | Use of Deprecated Solidity Functions | PASSED |

| | | |
|---|---|---|
| SWC-110 | Assert Violation | **PASSED** |
| SWC-109 | Uninitialized Storage Pointer | **PASSED** |
| SWC-108 | State Variable Default Visibility | **PASSED** |
| SWC-107 | Reentrancy | **PASSED** |
| SWC-106 | Unprotected SELFDESTRUCT Instruction | **PASSED** |
| SWC-105 | Unprotected Ether Withdrawal | **PASSED** |
| SWC-104 | Unchecked Call Return Value | **PASSED** |
| SWC-103 | Floating Pragma | **PASSED** |
| SWC-102 | Outdated Compiler Version | **PASSED** |
| SWC-101 | Integer Overflow and Underflow | **PASSED** |
| SWC-100 | Function Default Visibility | **PASSED** |

# AUDIT PASSED

## Low Issues

| | |
|---|---|
| Potential use of “block.number” as source of randomness | L: 1619, 1916, 1919 |

# Audit Comments

- Deployer can transfer ownership
- Deployer can renounce ownership
- Deployer can distribute dividend tokens
- Deployer can disable transfer delay
- Deployer can exclude/include addresses from dividends
- Deployer can exclude/include addresses from max transaction amount
- Deployer can exclude/include addresses from fees
- Deployer can enable trading
- Deployer can disable/enable swap
- Deployer can set/update max transaction amount not greater than .5% of total supply
- Deployer can set/update ma wallet amount not greater than 1% of total supply
- Deployer can set/update total buy fees not greater than 20%
- Deployer can set/update total sell fees not greater than 30%
- Deployer can change address receivers
- Deployer can set/update gas settings between 200,000 and 500,000
- Deployer can set/update claim wait settings
- Deployer can withdraw tokens from contract
- Deployer can blacklist users
- Deployer can update minimum token balance for dividends with an indefinite amount
- Deployer can set balance
- Deployer cannot pause contract

# CONTRACTWOLF

Blockchain Security - Smart Contract Audits