



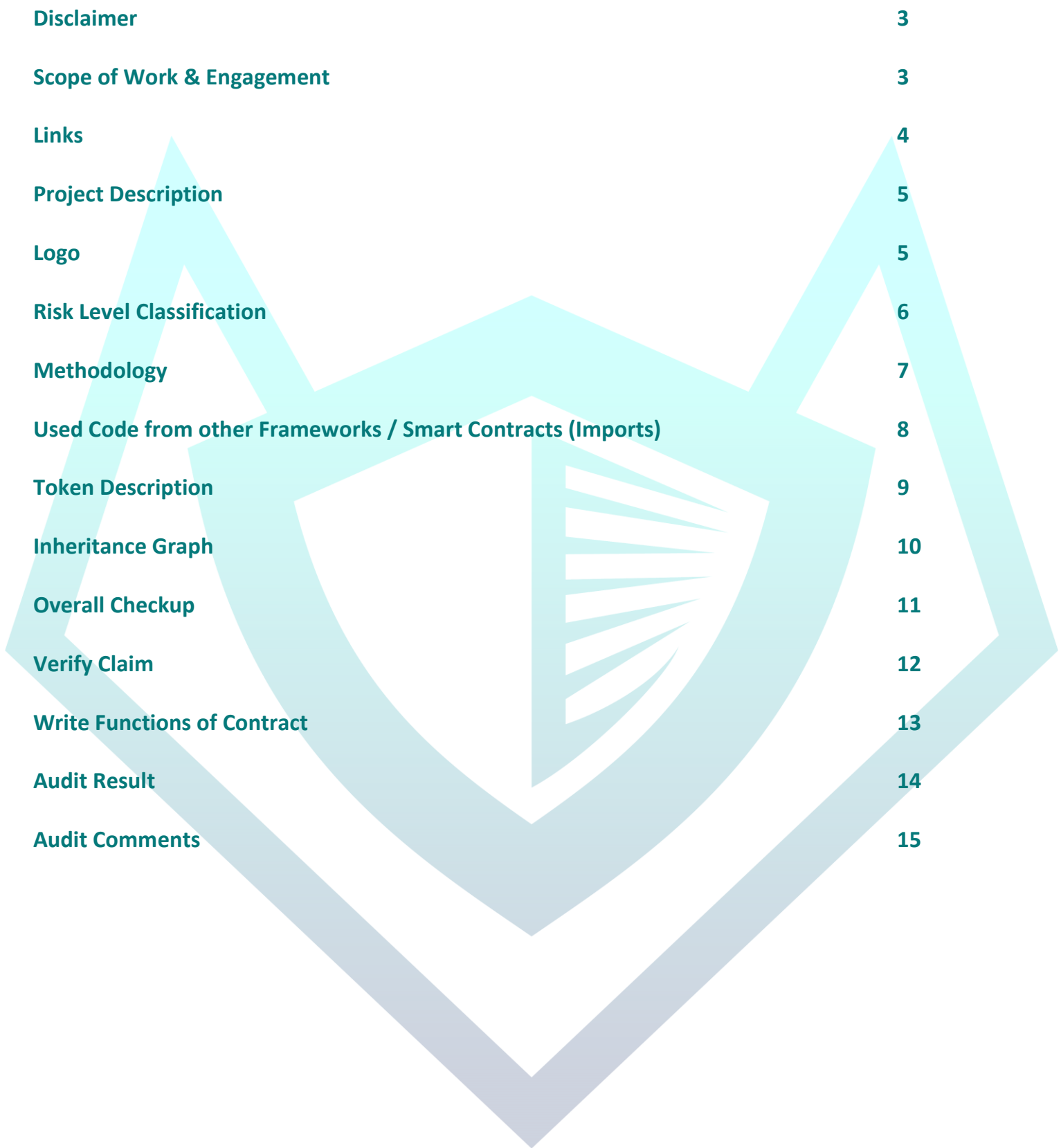
CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

April 13, 2022





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 its **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

Koala Hero 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 **Koala Hero**.

Network

Binance Smart Chain (BEP20)

Contract link

<https://bscscan.com/address/0x1C8b6c2465d97648e7aA1E4F4f9F3A6d414F6741>

Website

<https://www.koalahero.live>

Telegram

<https://t.me/koalaheroofficial>

Twitter

<https://twitter.com/KoalaHerobsc>

Description

Koala Hero is a deflationary token designed for the BSC network. Koala Hero owners will earn from 1% to 6%% Rewards in BUSD. 4% of each purchase and each sale is sent to the auto lp wallet. From 6% to 1% on marketing which is important to US.

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

- BABYTOKEN
- BABYTOKENDividendTracker
- BaseToken
- Clones
- Context
- ContextUpgradeable
- DividendPayingToken
- DividendPayingTokenInterface
- DividendPayingTokenOptionalInterface
- ERC20
- ERC20Upgradeable
- IERC20
- IERC20Metadata
- IERC20MetadataUpgradeable
- IERC20Upgradeable
- IUniswapV2Factory
- IUniswapV2Pair
- IUniswapV2Router01
- IUniswapV2Router02
- Initializable
- IterableMapping
- Ownable
- OwnableUpgradeable
- SafeMath
- SafeMathInt
- SafeMathUint

Description

Optimization enabled: Yes

Decimal: 18

Symbol: KoHe

Max / Total supply: 90,000,000,000

Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	5	5	10	6

Exposed Functions

Version	Public	Private	External	Internal
1.0	56	10	106	54

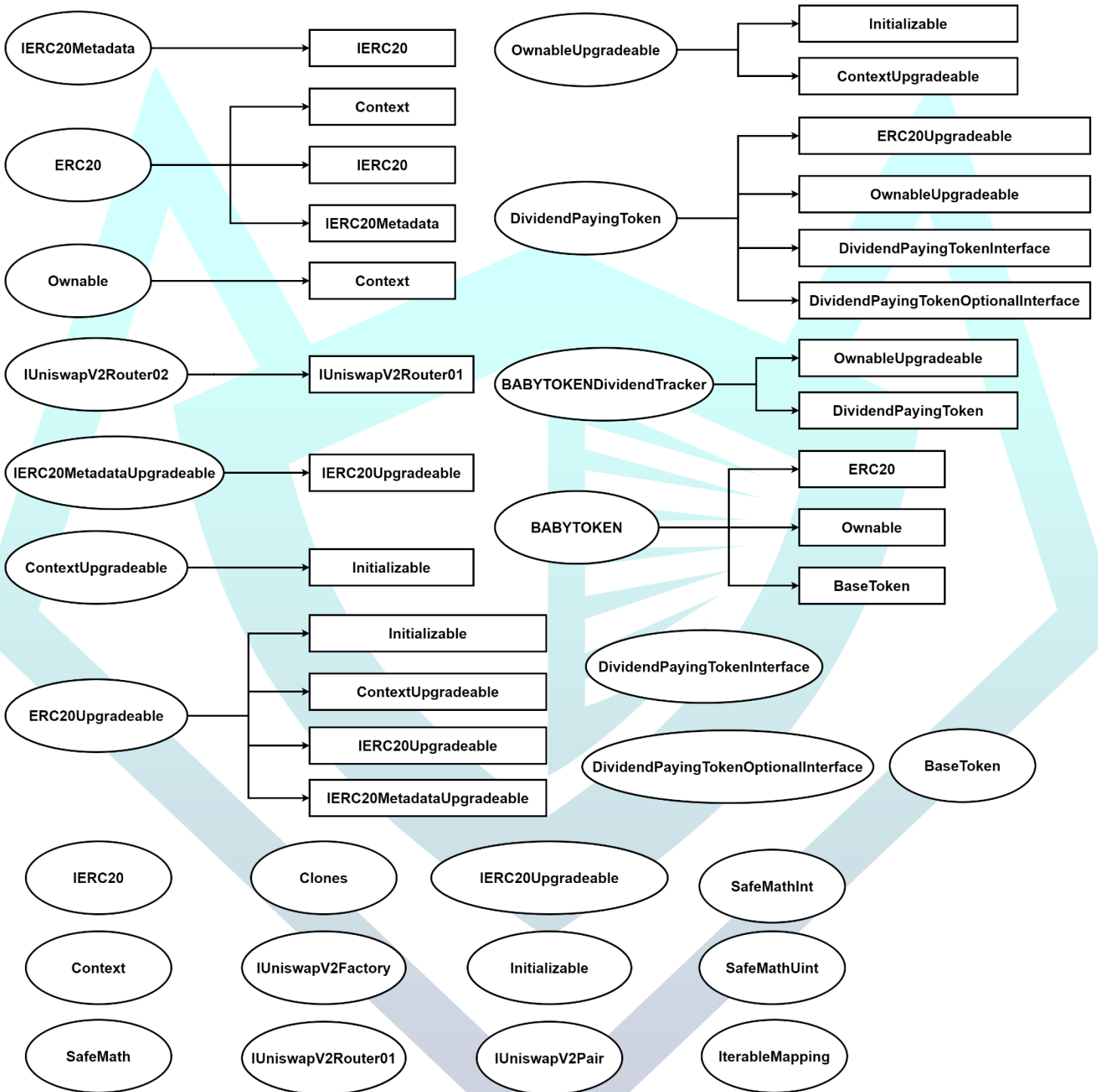
State Variables

Version	Total	Public
1.0	42	20

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.4		Yes	Yes	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. approve	13. setMarketingWallet
2. claim	14. setSwapTokensAtAmount
3. decreaseAllowance	15. setTokenRewardsFee
4. excludeFromDividends	16. transfer
5. excludeFromFees	17. transferFrom
6. excludeMultipleAccountsFromFees	18. transferOwnership
7. increaseAllowance	19. updateClaimWait
8. processDividendTracker	20. updateDividendTracker
9. renounceOwnership	21. updateGasForProcessing
10. setAutomatedMarketMakerPair	22. updateMinimumTokenBalanceForDividends
11. setLiquiditFee	23. updateUniswapV2Router
12. setMarketingFee	

AUDIT PASSED

Low Issues

Use of “tx.origin” as part of authorization control (SWC-115)	L: 2952 C: 12, L: 3084 C: 20
---	------------------------------

Audit Comments

- Deployer cannot mint after initial deployment
- Deployer cannot burn
- Deployer cannot pause contract
- Deployer cannot block user
- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can set fees not greater than 25%



CONTRACTWOLF

Blockchain Security - Smart Contract Audits