Revoluzion Audit

SEAL OF REVOLUZION AUDIT

INTEGRITY

Audit Report

Name	: Banana Labs Token
Symbol	: BLABS
Decimals	: 18
Address	: 0xcEf1A7b8f31acb73B10923461d7888b531C19042
Owner	: 0x85d220e311dc7cb9241f8023c2cef785e41ab4e7
Network	: Binance Smart Chain (Mainnet)
Type	: BEP20
Audited on	: 25 November 2022
Updated on	: 25 November 2022

Contents

Project Overview

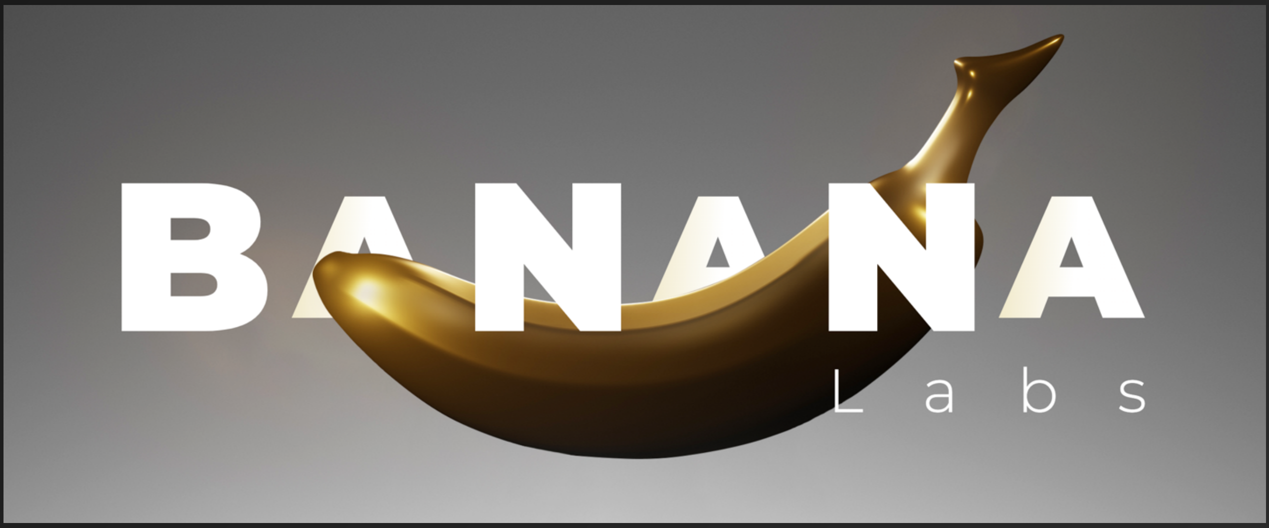
Name	Banana Labs Token
Symbol	BLABS
Decimals	18
Total Supply	1,000,000
Tax	Buy 7% | Sell 7% — (Fixed Tax)
Compiler Version	v0.8.16+commit.07a7930e
Optimization	No with 200 runs
License Type	MIT
Explorer Link	https://bscscan.com/address/0xcEf1A7b8f31acb73B10923461d7888b531C19042
Create Tx	0x45410a324b065c3e1323a9c39ccdf55d73606f90a9b58d1982f33e4203f239d0
Creator	0x85D220e311Dc7cb9241F8023C2CeF785e41ab4E7
Featured Address	Growth Wallet — 0x4f9fe5cd518b89ff10ec6fc28134cd4aa1b24c43 Distributor Smart Contract — 0x578669ac7fD735CA4b5Cc382564CA257e2745AF4
GitHub Link	N/A
Website	https://www.bananalabs.dev

Project Description

According to the website

Banana Labs is a team passionate about expressing their creativity through their work that focused on creating value in the metaverse. Banana Labs Token was developed to fund their growth. Holders will benefit the most from the ecosystem. The team have less of a focus on aggressive marketing and more of a focus on creating and providing value over time.

Release Date	: 4 December 2022
Category	: Utility Token



Online Presence

About Website

Registrar	: https://www.namecheap.com/
Domain Expiration	: 2023-11-14
SSL Certificate	: Issued by Let's Encrypt.

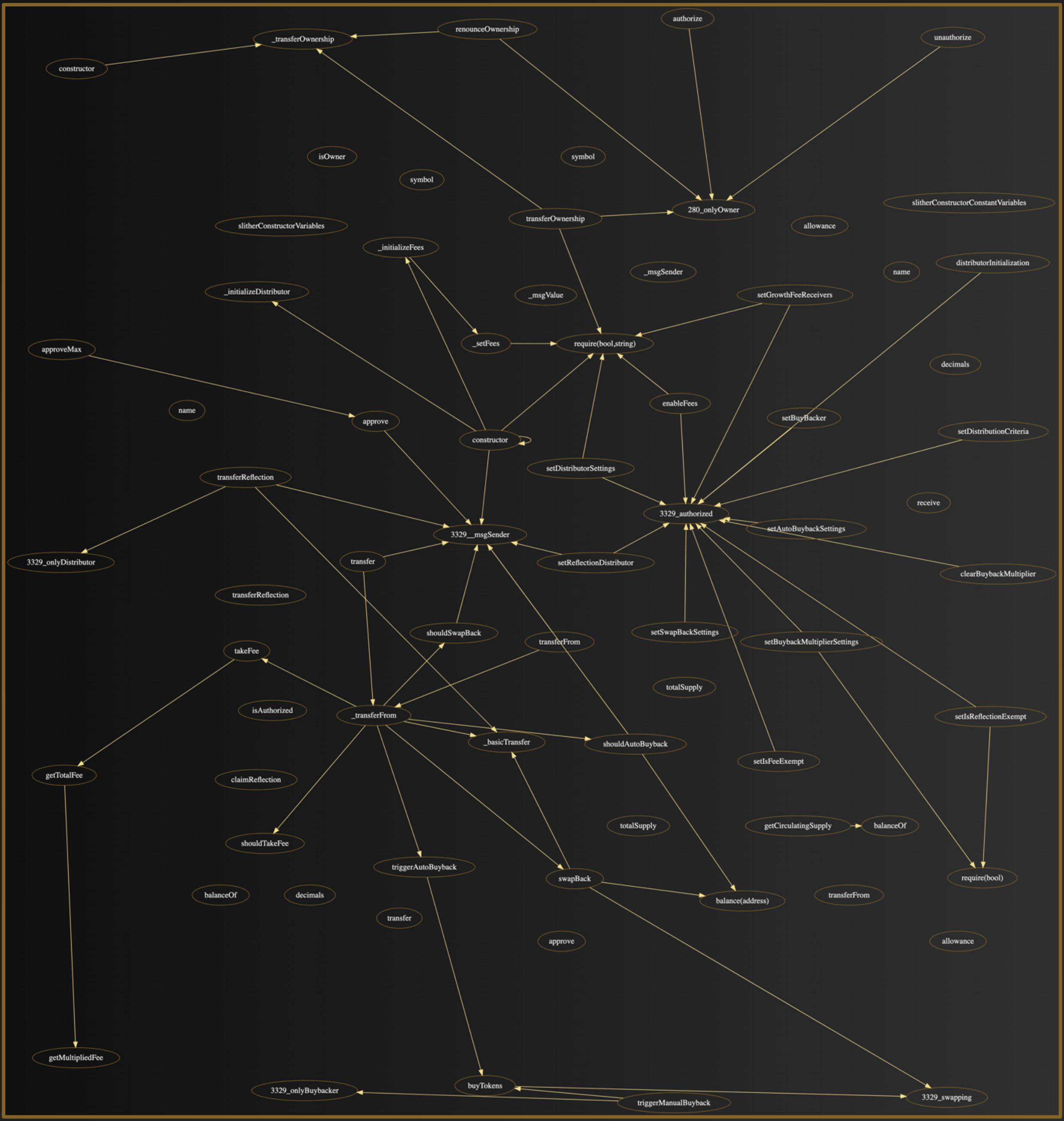
Official Links

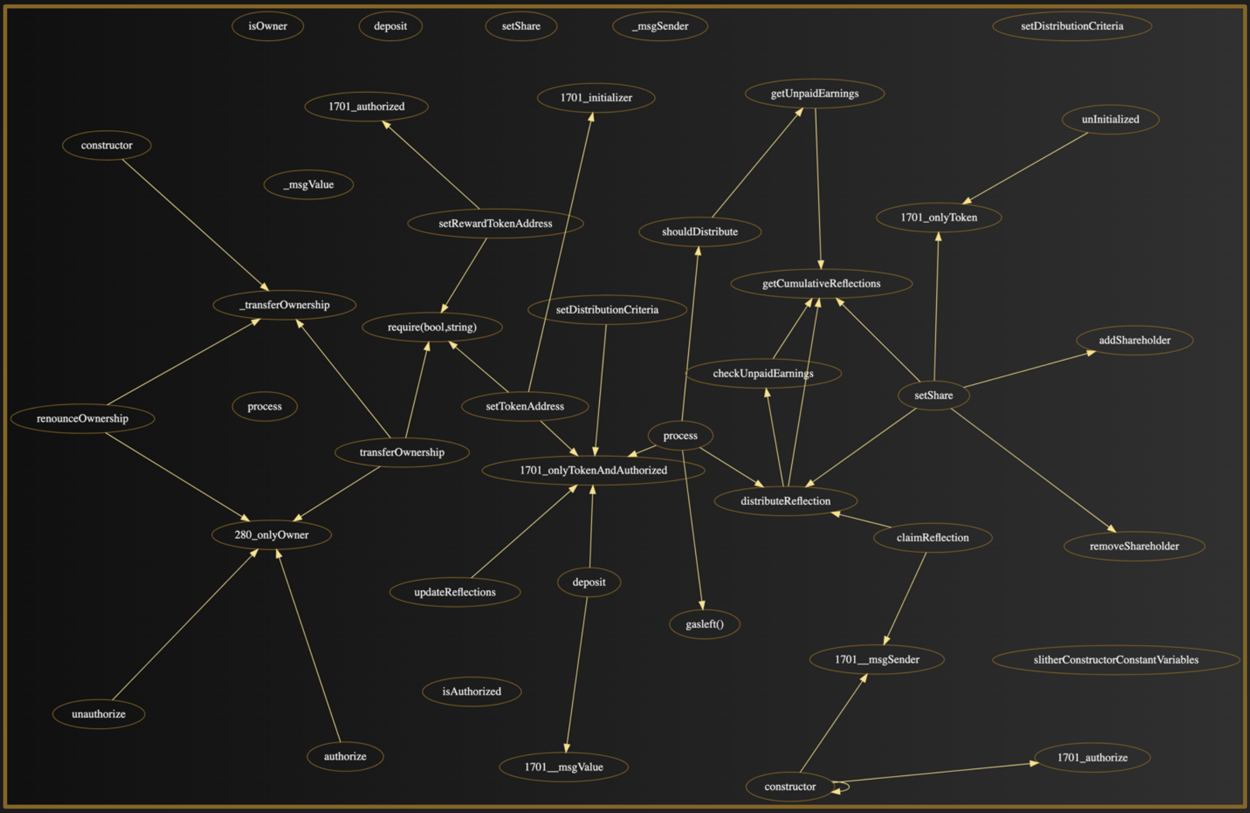
Website	https://bananalabs.dev
Telegram	https://t.me/BananaLabs

The Team

About	We only interacted with the owner for the audit. However, there are no KYC procedure being conducted by Revoluzion on any of Banana Labs Token's team members.
KYC Issuer	N/A
Member's KYC'd	N/A
KYC Date	N/A
Certificate Link	N/A
Task Completed	N/A

Contract Functions Interaction




isOwner
deposit
setShare
_msgSender
setDistributionCriteria
1701_initializer
getUnpaidEarnings
1701_authorized
unInitialized
constructor
_msgValue
1701_onlyToken
setRewardTokenAddress
shouldDistribute
getCumulativeReflections
_transferOwnership
setDistributionCriteria
require(bool,string)
addShareholder
checkUnpaidEarnings
setShare
process
setTokenAddress
renounceOwnership
process
transferOwnership
1701_onlyTokenAndAuthorized
distributeReflection
claimReflection
removeShareholder
280_onlyOwner
deposit
updateReflections
gasleft()
1701__msgSender
slitherConstructorConstantVariables
isAuthorized
unauthorize
authorize
1701__msgValue
1701_authorize
constructor

Audit Overview

Threat Level

When conducting audit on smart contract(s), we first look for known vulnerabilities and issues within the code because any exploitation on such vulnerabilities and issues by malicious actors could potentially result in serious financial damage to the projects. All the issues and vulnerabilities will be categorized into the categories as provided below.

Critical

This category provides issues and vulnerabilities that are critical to the performance/functionality of the smart contract and should be fixed by project creator before moving to a live environment.

Medium

This category provides issues and vulnerabilities that are not that critical to the performance/functionality of the smart contract but is recommended to be fixed by project creator before moving to a live environment.

Minor

This category provides issues and vulnerabilities that are minor to the performance/functionality of the smart contract and can remain unfixed by project creator before moving to a live environment.

Informational

This category provides issues and vulnerability that have insignificant effect on the performance/functionality of the smart contract and can remain unfixed by project creator before moving to a live environment. However, fixing them can further improve the efficacy or security for features with a risk-free factor.

Notable Information

- Contract Owner cannot stop or pause transactions.
- Contract Owner cannot transfer tokens from specific address.
- Contract Owner cannot mint new tokens after deploying smart contract.
- Contract Owner cannot burn tokens from specific wallet.
- Contract Owner cannot blacklist wallets from selling.
- There are no compiler warnings when compiling the smart contracts.
- Contract is using safe Zeppelin modules.
- Contract can be used to create presale and finalize pool (Tested on Pinksale).

Bugs and Optimizations Detection

This table is based on the result obtained from running the smart contract through Slither's Solidity static analysis.

What it detects	Impact	Confidence	Status
Storage abiencoderv2 array	High	High	Passed
transferFrom uses arbitrary from	High	High	Passed
Modifying storage array by value	High	High	Passed
The order of parameters in a shift instruction is incorrect.	High	High	Passed
Multiple constructor schemes	High	High	Passed
Contract's name reused	High	High	Passed
Detected unprotected variables	High	High	Passed
Public mappings with nested variables	High	High	Passed
Right-To-Left-Override control character is used	High	High	Passed
State variables shadowing	High	High	Passed
Functions allowing anyone to destruct the contract	High	High	Passed
Uninitialized state variables	High	High	Passed
Uninitialized storage variables	High	High	Passed
Unprotected upgradeable contract	High	High	Passed

transferFrom uses arbitrary from with permit	High	Medium	Passed
Functions that send Ether to arbitrary destinations	High	Medium	Moderated
Tainted array length assignment	High	Medium	Passed
Controlled delegatecall destination	High	Medium	Passed
Payable functions using delegatecall inside a loop	High	Medium	Passed
msg.value inside a loop	High	Medium	Passed
Reentrancy vulnerabilities (theft of ethers)	High	Medium	Moderated
Signed storage integer array compiler bug	High	Medium	Passed
Unchecked tokens transfer	High	Medium	Passed
Weak PRNG	High	Medium	Passed
Detects ERC20 tokens that have a function whose signature collides with EIP-2612's DOMAIN_SEPARATOR()	Medium	High	Passed
Detect dangerous enum conversion	Medium	High	Passed
Incorrect ERC20 interfaces	Medium	High	Passed
Incorrect ERC721 interfaces	Medium	High	Passed
Dangerous strict equalities	Medium	High	Passed
Contracts that lock ether	Medium	High	Passed

Deletion on mapping containing a structure	Medium	High	Passed
State variables shadowing from abstract contracts	Medium	High	Passed
Tautology or contradiction	Medium	High	Passed
Unused write	Medium	High	Passed
Misuse of Boolean constant	Medium	Medium	Passed
Constant functions using assembly code	Medium	Medium	Passed
Constant functions changing the state	Medium	Medium	Passed
Imprecise arithmetic operations order	Medium	Medium	Passed
Reentrancy vulnerabilities (no theft of ethers)	Medium	Medium	Passed
Reused base constructor	Medium	Medium	Passed
Dangerous usage of tx.origin	Medium	Medium	Passed
Unchecked low-level calls	Medium	Medium	Passed
Unchecked send	Medium	Medium	Passed
Uninitialized local variables	Medium	Medium	Passed
Unused return values	Medium	Medium	Passed
Modifiers that can return the default value	Low	High	Passed
Built-in symbol shadowing	Low	High	Passed

Local variables shadowing	Low	High	Passed
Uninitialized function pointer calls in constructors	Low	High	Passed
Local variables used prior their declaration	Low	High	Passed
Constructor called not implemented	Low	High	Passed
Multiple calls in a loop	Low	Medium	Moderated
Missing Events Access Control	Low	Medium	Passed
Missing Events Arithmetic	Low	Medium	Passed
Dangerous unary expressions	Low	Medium	Passed
Missing Zero Address Validation	Low	Medium	Passed
Benign reentrancy vulnerabilities	Low	Medium	Moderated
Reentrancy vulnerabilities leading to out-of-order Events	Low	Medium	Moderated
Dangerous usage of block.timestamp	Low	Medium	Moderated
Assembly usage	Informational	High	Passed
Assert state change	Informational	High	Passed
Comparison to boolean constant	Informational	High	Passed
Deprecated Solidity Standards	Informational	High	Passed

Un-indexed ERC20 event parameters	Informational	High	Passed
Function initializing state variables	Informational	High	Passed
Low level calls	Informational	High	Passed
Missing inheritance	Informational	High	Passed
Conformity to Solidity naming conventions	Informational	High	Moderated
If different pragma directives are used	Informational	High	Passed
Redundant statements	Informational	High	Passed
Incorrect Solidity version	Informational	High	Passed
Unimplemented functions	Informational	High	Passed
Unused state variables	Informational	High	Passed
Costly operations in a loop	Informational	Medium	Moderated
Functions that are not used	Informational	Medium	Passed
Reentrancy vulnerabilities through send and transfer	Informational	Medium	Moderated

Variable names are too similar	Informational	Medium	**Moderated**
Conformance to numeric notation best practices	Informational	Medium	**Passed**
State variables that could be declared constant	Optimization	High	**Passed**
Public function that could be declared external	Optimization	High	**Passed**

Contract Diagnostic

Link for initial smart contract commit being audited on GitHub:

https://github.com/RevoluzionToken/Revoluzion-Audits/commit/74891b6afc58756fec3b75558a984beb729003cf

CODE	SEVERITY	DESCRIPTION
SWC-103	**Minor**	A floating pragma is set.
SWC-120	**Minor**	Potential use of "block.number" as source of randomness.
CaL	**Minor**	Loops with multiple calls.
CoL	**Informational**	Loop with costly operations.
NC	**Informational**	Naming convention.
SN	**Informational**	Similar name.

SWC-103 — A floating pragma is set

SEVERITY	Minor
LOCATION(S)	BananaLabsToken.sol#L3
DESCRIPTION	The current pragma Solidity directive is set as ""^0.8.16"".
RECOMMENDATIONS	Project creator is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. It is important if the project rely on bytecode-level verification of the code.
STATUS	**N/A**

SWC-120 — Potential use of "block.number" as source of randomness

SEVERITY	Minor
LOCATION(S)	BananaLabsToken.sol#844, 1015, 1034
DESCRIPTION	The environment variable "block.number" looks like it might be used as a source of randomness to trigger a function.
RECOMMENDATIONS	We would recommend project owner to not use any of the environment variables like coinbase, gaslimit, block number and timestamp as sources of randomness since they are predictable and be aware that such usage could introduces a certain level of trust into miners. Keep in mind that malicious miner can manipulate the value of those variables and that any attackers could also predetermine the hashes of earlier blocks. However, based on our analysis, there's nothing to be done by project owner since in each of the "block.number" value was used as a means to keep track of time/epoch that relates to the trigger of a specific function.
STATUS	**N/A**

CaL — Loops with multiple calls

SEVERITY	Minor
LOCATION(S)	BananaLabsToken.sol#L513-529
DESCRIPTION	[ReflectionDistributor.distributeReflection] (#L513-529) has external calls inside a loop at line #L528
RECOMMENDATIONS	Project creator can choose to either make use of pull over push strategy for external calls or ignore the issues since the logic does require such function(s)
STATUS	**N/A**

CoL — Loop with costly operations

SEVERITY	Minor
LOCATION(S)	BananaLabsToken.sol#L478-504, 513-529
DESCRIPTION	[ReflectionDistributor.process] (#L478-504) has costly operations inside a loop at #L494 [ReflectionDistributor.distributeReflection] (#L513-529) has costly operations inside a loop at #L524
RECOMMENDATIONS	We would usually recommend the use of local variables instead to hold the loop computation result. However, project creator can ignore the issues since the logic does require such function(s)
STATUS	**N/A**

NC — Naming convention

SEVERITY	Informational — Minor
LOCATION(S)	BananaLabsToken.sol#L274
DESCRIPTION	[IUniswapV2Router01.WETH()] (#L274) is not in mixedCase..
RECOMMENDATIONS	Based on our analysis, the IUniswapRouter01 smart contract is a direct fork from Uniswap. Although the name doesn't conform to the standard convention, it's still okay to leave it be to avoid from potentially breaking any external function.
STATUS	**N/A**

SN — Similar name

SEVERITY	Informational — Minor
LOCATION(S)	BananaLabsToken.sol#L276
DESCRIPTION	[IUniswapV2Router01.addLiquidity().amountADesired] (#L276) is too similar to [IUniswapV2Router01.addLiquidity().amountBDesired] (#L276).
RECOMMENDATIONS	Based on our analysis, the IUniswapV2Router smart contract is a direct fork from Uniswap. Although their names are too similar, it's still okay to leave them be for the purpose of following the standard parameter declaration that is widely used as reference.
STATUS	**N/A**

Disclaimer

This report only shows findings based on our limited project analysis according to the good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall online presence and team transparency details of which are set out in this report. To get a full view of our analysis, **it is important for you to read the full report.** Under no circumstances did Revoluzion Audit receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. **Our team provides no guarantees against the sale of team tokens or the removal of liquidity by the project** audited in this document.

While **we have done our best to conduct thorough analysis to produce this report**, it is crucial to note that you should not rely solely on this report and use the content provided in this document as financial advice or a reason to buy any investment. The Our team disclaims any liability against us for the resulting losses based on the you decision made by relying on the content of this report. **You must conduct your own independent investigations before making any decisions** to protect yourselves from being scammed. We go into more detail on this in the disclaimer clause in the next page — please make sure to read it in full.

Full Disclaimer Clause

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 all copies of this report downloaded and/or printed by you. This report is provided for information purposes only, on a non-reliance basis and does not constitute to any investment advice.

No one shall have any right to rely on the report or its contents, and Revoluzion Audit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (collectively known as Revoluzion) owe no duty of care towards you or any other person, nor do we 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 Revoluzion 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, Revoluzion hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Revoluzion, 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, website, social media, and team.