Revoluzion Audit

SEAL OF REVOLUZION AUDIT

• INTEGRITY •

Audit Report

Name	: Abitoken
Symbol	: ABIT
Decimals	: 9
Address	: 0xA666Bf6FC5813D4753DE4CdBB4174d6B4667E57B
Owner	: 0x85F060a3c0fDF6e73d5453ae08000BFF3Fa0eC9a
Network	: Binance Smart Chain (Mainnet)
Type	: BEP20
Audited on	: 24 December 2022

Contents

Project Overview

Name	Abitoken
Symbol	ABIT
Decimals	9
Total Supply	800,000,000
Tax	Buy 3% | Sell 3% — (Fixed Tax)
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	Yes with 200 runs
License Type	MIT
Explorer Link	https://bscscan.com/address/0xa666bf6fc5813d4753de4cdbb4174d6b4667e57b
Create Tx	0x142241d2e20a0e018d23d45ed88a4ae4d49c1cd6a9b72e4505001402a86679b3
Creator	0x85F060a3c0fDF6e73d5453ae08000BFF3Fa0eC9a
Featured Wallet	Marketing Wallet — 0xDB7dB3a1b4E2a75721bcDbbE18d11fD036AEe660
GitHub Link	N/A — Created as Pinksale Liquidity Generator Token
Website	https://www.abitoken.com

Project Description

According to their website

Abitoken was developed revolving around idea of providing anonymous private open-source cryptocurrency wallet to the users. A wallet that will be very convenient and easy to use that comes with many features.

Release Date	: TBA
Category	: Utility Token

Online Presence

About Website

Registrar	: https://www.wix.com
Domain Expiration	: 2023-08-18
SSL Certificate	: Issued by Sectigo Limited

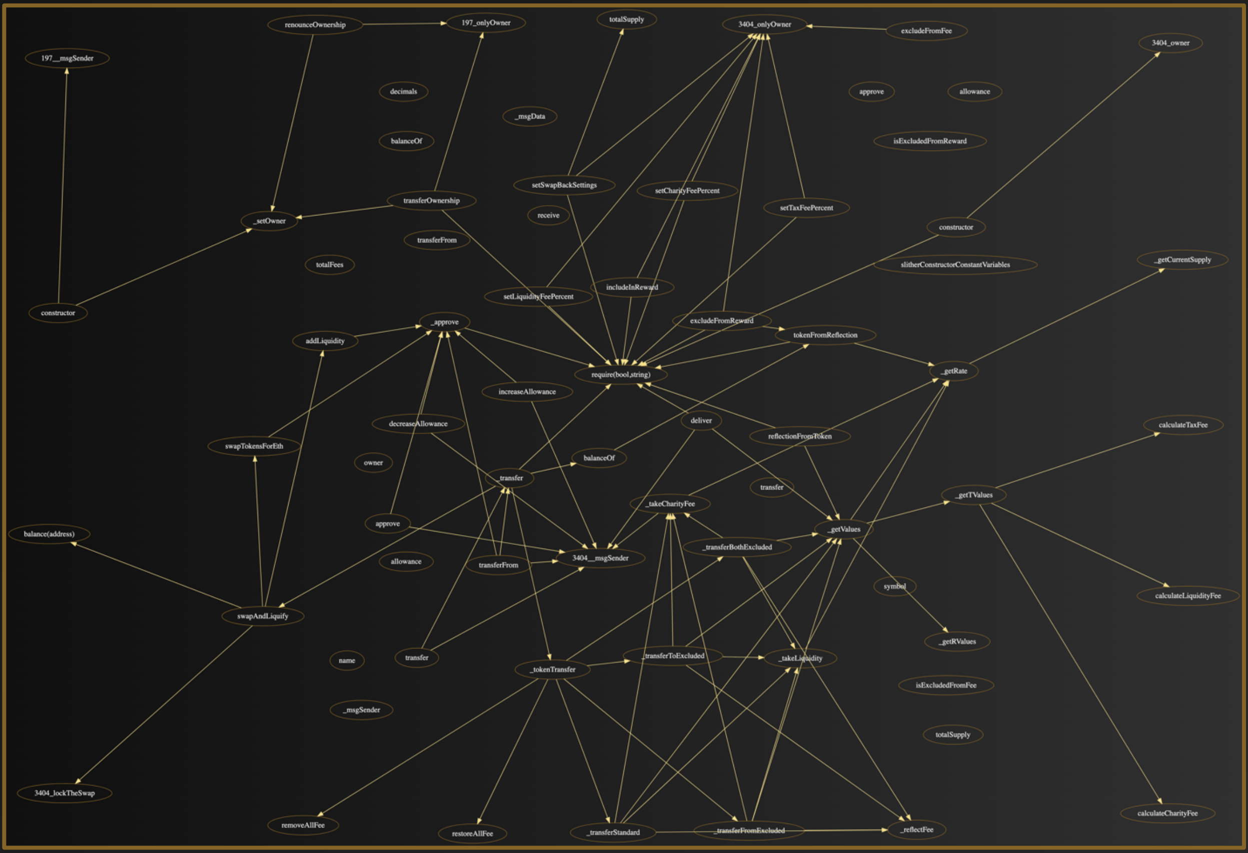
Official Links

Website	https://abitoken.com
Telegram	https://t.me/abittokenbep20

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 Abitoken' 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



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.
- Both buy and sell fees are hardcoded to be a total of 3%.
- Contract Owner cannot blacklist wallets from selling.
- There are no compiler warnings when compiling the smart contracts.
- Contract is using interface from safe Zeppelin modules.

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	Moderated
Modifiers that can return the default value	Low	High	Passed
Built-in symbol shadowing	Low	High	Passed

Local variables shadowing	Low	High	Moderated
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	Passed
Missing Events Access Control	Low	Medium	Passed
Missing Events Arithmetic	Low	Medium	Moderated
Dangerous unary expressions	Low	Medium	Passed
Missing Zero Address Validation	Low	Medium	Moderated
Benign reentrancy vulnerabilities	Low	Medium	Moderated
Reentrancy vulnerabilities leading to out-of-order Events	Low	Medium	Moderated
Dangerous usage of block.timestamp	Low	Medium	Passed
Assembly usage	Informational	High	Moderated
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	Moderated
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	Moderated
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	Moderated
Reentrancy vulnerabilities through send and transfer	Informational	Medium	Passed

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

CODE	SEVERITY	DESCRIPTION
SWC-108	Minor	State variable visibility is not set.
SWC-110	Unknown	Out of bounds array access.
EM	Informational	Function recommended to emit events.
CL	Informational	Costly loop.
DC	Informational	Dead code.
SV	Informational	Solidity compiler version.
NC	Informational	Naming convention.
UR	Informational	Unused return value(s).
SN	Informational	Similar name.
EF	Informational	Public function can be declared as external.

SWC-108 — State variable visibility is not set

SEVERITY	Minor
LOCATION(S)	Abitoken.sol#L959
DESCRIPTION	It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.
RECOMMENDATIONS	Project creator is recommended to set the visibility for " inSwapAndLiquify" parameter even if it is supposed to be internal.
STATUS	N/A

SWC-110 — Out of bounds array access

SEVERITY	Unknown
LOCATION(S)	Abitoken.sol#L1527
DESCRIPTION	The index access expression can cause an exception in case of use of invalid array index value.
RECOMMENDATIONS	This produces line of code could produce -1 index for the array. As long as project creator didn't include owner address, this should not produce any issue as the exclude array will not be an empty array at the start and in the case if there's no other address being excluded. No specific actions needed to be taken by project creator.
STATUS	**N/A**

EM — Function recommended to emit events

SEVERITY	Informational — Low
LOCATION(S)	Abitoken.sol#L1241-1247, 1249-1258, 1260-1266
DESCRIPTION	[LiquidityGeneratorToken.setTaxFeePercent] (#L1241-1247) should emits an event for L#1242 [LiquidityGeneratorToken.setLiquidityFeePercent] (#L1249-1258) should emits an event for L#1253 [LiquidityGeneratorToken.setCharityFeePercent] (#L1260-1266) should emits an event for L#1261
RECOMMENDATIONS	Project creator is recommended to emit events for these functions to facilitate better communication between smart contract and its user interfaces.
STATUS	N/A

CL — Costly loop

SEVERITY	Informational — Medium
LOCATION(S)	Abitoken.sol#L807-818
DESCRIPTION	[LiquidityGeneratorToken.includeInReward] (#L1200-1211) has costly operations inside a loop.
RECOMMENDATIONS	Project creator could further optimize this function by creating a better logic to search and remove the address from the array instead of doing it in a loop. We recommend using mapping to keep track of the index for the address within the address and use the value to update the array.
STATUS	N/A

DC — Dead code

SEVERITY	Informational — Medium
LOCATION(S)	Abitoken.sol#L110-112, 211-217, 224-229, 236-246, 253-258, 265-270, 340-342, 380-389, 406-415, 445-455, 473-478, 498-500, 508-514, 527-533, 541-552, 560-562, 570-579, 587-589, 597-606, 614-634
DESCRIPTION	[Context._msgData()] (#L110-112) is never used and should be removed. [SafeMath.tryAdd] (#L211-217) is never used and should be removed. [SafeMath.trySub] (#L224-229) is never used and should be removed. [SafeMath.tryMul] (#L236-246) is never used and should be removed. [SafeMath.tryDiv] (#L253-258) is never used and should be removed. [SafeMath.tryMod] (#L265-270) is never used and should be removed. [SafeMath.mod] (#L340-342) is never used and should be removed. [SafeMath.div] (#L380-389) is never used and should be removed. [SafeMath.mod] (#L406-415) is never used and should be removed. [Address.isContract] (#L445-455) is never used and should be removed. [Address.sendValue] (#L473-478) is never used and should be removed. [Address.functionCall] (#L498-500) is never used and should be removed.

	[Address.functionCall] (#L508-514) is never used and should be removed. [Address.functionCallWithValue] (#L527-533) is never used and should be removed. [Address.functionCallWithValue] (#L541-552) is never used and should be removed. [Address.functionStaticCall] (#L560-562) is never used and should be removed. [Address.functionStaticCall] (#L570-579) is never used and should be removed. [Address.functionDelegateCall] (#L587-589) is never used and should be removed. [Address.functionDelegateCall] (#L597-606) is never used and should be removed. [Address.verifyCallResult] (#L614-634) is never used and should be removed.
RECOMMENDATIONS	Based on our analysis, the Address, Context and SafeMath smart contracts is the standard that is a direct fork from Open Zeppelin and were used within the contract itself. However, it is recommended for project creator to remove those functions to further optimize the smart contract since they are not used anywhere at all. Doing so will reduce the amount gas required when deploying the smart contract.
STATUS	**N/A**

SV — Solidity compiler version

SEVERITY	Informational — High
LOCATION(S)	Abitoken.sol#L911
DESCRIPTION	Fixed pragma version =0.8.4 at L#911 despite all others being ^0.8.0
RECOMMENDATIONS	Due to this fixed pragma version, all the others will have a restriction to only support up to version 0.8.4. Project creator should choose either to use fixed version 0.8.4 or allow old version 0.8.0 support.
STATUS	**N/A**

NC — Naming convention

SEVERITY	Informational — Minor
LOCATION(S)	Abitoken.sol#L645, 946, 949, 952, 957, 1268, 1407, 1411, 1419
DESCRIPTION	[IUniswapV2Router01.WETH] (#L645) is not in mixedCase. [LiquidityGeneratorToken._taxFee] (#L946) is not in mixedCase. [LiquidityGeneratorToken._liquidityFee] (#L949) is not in mixedCase. [LiquidityGeneratorToken._charityFee] (#L952) is not in mixedCase. [LiquidityGeneratorToken._charityAddress] (#L957) is not in mixedCase. [LiquidityGeneratorToken.setSwapBackSettings] (#L1268) is not in mixedCase. [LiquidityGeneratorToken.calculateTaxFee] (#L1407) is not in mixedCase. [LiquidityGeneratorToken.calculateLiquidityFee] (#L1411) is not in mixedCase. [LiquidityGeneratorToken.calculateCharityFee] (#L1419) is not in mixedCase.
RECOMMENDATIONS	Based on our analysis, the IUniswapV2Router 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. However, for LiquidityGeneratorToken smart contract, it is okay for project creator to update the name of the parameters in those functions so that they conform to the standard naming convention.

STATUS	N/A

UR — Unused return value(s)

SEVERITY	Informational — Minor
LOCATION(S)	Abitoken.sol#L1541-1554
DESCRIPTION	[LiquidityGeneratorToken.addLiquidity] (#1541-1554) ignores the return value at [uniswapV2Router.addLiquidityETH] (#L1546-1553)
RECOMMENDATIONS	Based on our analysis, project creator doesn't need to do anything for this issue since it will be redundant.
STATUS	**N/A**

SN — Similar name

SEVERITY	Informational — Minor
LOCATION(S)	Abitoken.sol#L650-651
DESCRIPTION	[IUniswapV2Router01.addLiquidity] (#L650-651) has two parameters names that are too similar.
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

EF — Public function can be declared as external

SEVERITY	Informational — Medium
LOCATION(S)	Abitoken.sol#L169-171, 177-180, 1048-1050, 1052-1054, 1056-1058, 1069-1076, 1078-1085, 1087-1094, 1096-1111, 1113-1124, 1126-1140, 1142-1144, 11469-1148, 1150-1160, 1162-1175, 1190-1198, 1237-1239, 1444-1446
DESCRIPTION	[Ownable.renounceOwnership] (#L169-171) should be declared as external. [Ownable.transferOwnership] (#L177-180) should be declared as external. [LiquidityGeneratorToken.name] (#L1048-1050) should be declared as external. [LiquidityGeneratorToken.symbol] (#L1052-1054) should be declared as external. [LiquidityGeneratorToken.decimals] (#L1056-1058) should be declared as external. [LiquidityGeneratorToken.transfer] (#L1069-1076) should be declared as external. [LiquidityGeneratorToken.allowance] (#L1078-1085) should be declared as external. [LiquidityGeneratorToken.approve] (#L1087-1094) should be declared as external. [LiquidityGeneratorToken.transferFrom] (#L1096-1111) should be declared as external. [LiquidityGeneratorToken.increaseAllowance] (#L1113-1124) should be declared as external. [LiquidityGeneratorToken.decreaseAllowance] (#L1126-1140) should be declared as external. [LiquidityGeneratorToken.isExcludedFromReward] (#L1142-1144) should be declared as external.

	[LiquidityGeneratorToken.totalFees] (#L1146-1148) should be declared as external. [LiquidityGeneratorToken.deliver] (#L1150-1160) should be declared as external. [LiquidityGeneratorToken.reflectionFromToken] (#L1162-1175) should be declared as external. [LiquidityGeneratorToken.excludeFromReward] (#L1190-1198) should be declared as external. [LiquidityGeneratorToken.excludeFromFee] (#L1237-1239) should be declared as external. [LiquidityGeneratorToken.isExcludedFromFee] (#L1444-1446) should be declared as external.
RECOMMENDATIONS	Based on our analysis, it is best for project creator to change the visibility of these functions from public to external for the purpose of optimizing the smart contract since they are not used internally at all within any of the smart contract.
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.