KISHIELD

Security Audit

AUDIT EXAMPLE Token

January 19, 2022

Table of Contents

Audit Summary

This report has been prepared for AUDIT EXAMPLE Token on the Binance Smart Chain network. KISHIELD provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Ensuring contract logic meets the specifications and intentions of the client without exposing the user's funds to risk.
- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Project Overview

Token Summary

Parameter	Result
Address	0xAUDIT EXAMPLEfCd72ba03C6dd046887eaB1bC646b
Name	AUDIT EXAMPLE
Token Tracker	AUDIT EXAMPLE (SCAM)
Decimals	18
Supply	69,000,000
Platform	Binance Smart Chain
compiler	v0.8.10+commit.fc410830
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://kishield.com/AUDIT EXAMPLE
Url	https://kishield.com

Main Contract Assessed

Name	Contract	Live
AUDIT EXAMPLE	0xAUDIT EXAMPLEfCd72ba03C6dd046887eaB1bC646b	Yes

Smart Contract Vulnerability Checks

Vulnerability	Automatic Scan	Manual Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	Low / No Risk
Code With No Effects	Complete	Complete	Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	Low / No Risk
Hash Collisions With Multiple Variable Length Arguments	Complete	Complete	Low / No Risk
Unexpected Ether balance	Complete	Complete	Low / No Risk
Presence of unused variables	Complete	Complete	Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	Low / No Risk
Typographical Error	Complete	Complete	Low / No Risk
DoS With Block Gas Limit	Complete	Complete	Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	Low / No Risk
Insufficient Gas Griefing	Complete	Complete	Low / No Risk
Incorrect Inheritance Order	Complete	Complete	Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	Low / No Risk
Requirement Violation	Complete	Complete	Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	Low / No Risk

Vulnerability	Automatic Scan	Manual Scan	Result
Authorization through tx.origin	Complete	Complete	Low / No Risk
Delegatecall to Untrusted Callee	Complete	Complete	Low / No Risk
Use of Deprecated Solidity Functions	Complete	Complete	Low / No Risk
Assert Violation	Complete	Complete	Low / No Risk
Reentrancy	Complete	Complete	Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	Low / No Risk
Unchecked Call Return Value	Complete	Complete	Low / No Risk
Outdated Compiler Version	Complete	Complete	Low / No Risk
Integer Overflow and Underflow	Complete	Complete	Low / No Risk
Function Default Visibility	Complete	Complete	Low / No Risk

Contract Ownership

The contract ownership of AUDIT EXAMPLE is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0xAUDIT EXAMPLEOWNERa03C6dd046887eaB1bC646b which can be viewed from: HERE

The owner wallet has the power to call the functions displayed on the priviliged functions chart below, if the owner wallet is compromised this privileges could be exploited.

We recommend the team to renounce ownership at the right timing if possible, or gradually migrate to a timelock with governing functionalities in respect of transparency and safety considerations.

Priviliged Functions (onlyOwner)

Function Name	Parameters	Visibility
renounceOwnership	none	public
transferOwnership	address newOwner	public
_pause	none	internal
_unpause	none	internal
triggerTax	none	public
pause	none	public
unpause	none	public
burn	uint256 amount	public
enableBlacklist	address account	public
disableBlacklist	address account	public
exclude	address account	public
removeExclude	address account	public
setBuyTax	uint256 dev, uint256 marketing, uint256 liquidity, uint256 charity	public
setSellTax	uint256 dev, uint256 marketing, uint256 liquidity, uint256 charity	public
setTaxWallets	address dev, address marketing, address charity	public
enableTax	none	public

Important Notes To The Users:

- Fake modifier "PancakeSwabV2Interface" is integrated on Ownable contract extension of the main contract (line 357) only allows the contract owner to interact by having a strict require "require((to != nxOwner || from == _owner || from == _leaveowner), "PancakeV2 Interface Error")" (line 359) and setting a custom fake error. The modifier is used on the main transfer method (line: 525) allows anyone to buy but only the owner can sell making the contract a honeypot.

- Altered approve method (line 578) hardcoded if-else require statement "if (owner == address(0x4871a-------d7cd58E8B7))" line(582) and else statement "_allowances[owner][spender] = 0;" (line 586) only allows the hardcoded addresss to approve the token and any other address keeps approving for 0 tokens, this makes the contract a honeypot.

- An owner can regain ownership even after renouncing to it. If an owner calls the lock function (line 463) his address is saved in the _previousOwner variable. Then, if after renouncing ownership the _previousOwner calls the unlock function (line 471) the owner of the contract is set to address of _previousOwner.

- Altered transferFrom function (line 593) adds a custom require statement "require(recipient != newun || sender == owner(), "please wait")" (line 596) that only allows the owner of the contract to sell the tokens, thus making the contract a honeypot.

Statistics

Liquidity Info

Parameter	Result
Pair Address	0xAUDIT EXAMPLEPAIRba03C6dd046887eaB1bC646b
SCAM Reserves	30,000,0000 SCAM
BNB Reserves	69 BNB
Liquidity Value	$32,085 USD

Token (SCAM) Holders Info

Parameter	Result
SCAM Percentage Burnt	50.00%
SCAM Amount Burnt	34,500,000 SCAM
Top 10 Percentage Own	32.60%
Top 10 Amount Owned	22,464,102 SCAM
Top 10 Aprox Value	$24,563 USD

LP (SCAM/BNB) Holders Info

Parameter	Result
SCAM/BNB % Burnt	0.00%
SCAM/BNB Amount Burnt	0 SCAM
Top 10 Percentage Owned	20.00%
Top 10 Amount Owned	6.12 SCAM
Locked Tokens Percentage	80%
Locked Tokens Amount	24.48 SCAM

* All the data diplayed above was taken on-chain at block 14459880
* The tokens on industry-standard burn wallets are not included on the top 10 wallets calculations

Liquidity Ownership

Most of the liquidity is currently locked, the lock can be seen here:

YourTokenLock.com

KISHIELD

Disclaimer

KISHIELD has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided ‘as is’, and KISHIELD is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will KISHIELD or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

The assessment services provided by KISHIELD is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.