PEX Audit

GenesisX50

May 5th 2023

Audit Details

GenesisX50

Auditor's - PapaExchange

Website - www.Genesisx50.com

Blockchain - Ethereum Blockchain

Disclaimer

PapaExchange LLP will be referred to as PEX per this report

- **PEX** 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.
- **PEX** does not provide any warranty on its released reports. We should not be used as a decision to invest into an audited project please do your own research. **PEX** provides transparent reports to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its Smart Contract.
- Each company or project shall be liable for its own security flaws and functionalities. **PEX** presence is to analyze, audit and assess the client's smart contract's code.

Scope of Work

- The main focus of this report/audit, is to document an accurate assessment of the condition of the smart contract and whether it has any security flaws in the implementation of the contract.
GSX50 team agreed and provided us with the files that needed to be tested (Through Github, Etherscan, files, etc.). **PEX** will be focusing on contract issues/functionalities along with the projects claims from smart contract to their website, whitepaper and repository where available, which has been provided by the project. Code is reviewed manually and with the use of software using industry best practices.

Background

- **PEX** was commissioned by **GenesisX50** to perform this Audit

• Contract Address
0xD71562195A097905125a72B55791fb789E2a3855

The purpose of the audit was to achieve the following:
- **Ensure that the smart contract functions as intended.**
- **Identify potential security issues with the smart contract.**

The information in this report should be used to understand the risk exposure of the smart
contract, and as a guide to improve the security posture of the smart contract by remediating
the issues that were identified.

GenesisX50

GenesisX50 features a revolutionary burn mechanism designed to increase token price, improve stability, and add value to our holders. When most projects burn tokens, they are added to the burn wallet, which simply becomes another holder. Although those tokens can never be sold, they do nothing to help the project, nor the investors.

Social Media

Twitter - https://twitter.com/GSX50_

Telegram - https://t.me/+giubjljTtNkyODQ5

Facebook - https://www.facebook.com/profile.php?id=100087381440982

Contract Details

Project Name - GenesisX50

Token Description - Utility Token

Compiler Version - v0.8.19

Current Holders - 1 Addresses

Current Transaction Count - 1

Total Supply - 1,000,000,000,000 Tokens

Token Ticker - GSX50

Decimals - 9

Top 100 Holder % - 100%

LP Lock - Liquidity not added yet

Contract Address

0xD71562195A097905125a72B55791fb789E2a3855

Contract Deployer Address

0x58ffFC4540Ebb9f77Ea0a1C1ffDb736478855295

Contract Owner Address

0xb9f2DAe4F00fe40B1Ae2fA4D04a2007289c6B8B1

KYCd by - Audit Rate

Launch Type - N/A

Owner Privileges/Fees

Privileges

Ownership has NOT been renounced. The owner has privileges and has authority to make some changes now. Owner entitled to **change Buy/Sell fees, set true or false values and exclude from fees**.

Fees

Buy - 0% Sell - 0%

Owner must keep fees at 10% or lower. This is below our recommended upper percentage of 25%.

Adjustable Functions

(After Contract Deployment)

1. Contract_fees
2. Contract_limits
3. Contract_openTrade
4. Process_addPair
5. Process_autoProcess
6. Process_manualProcess
7. Process_rescueTokens
8. Process_triggerCount
9. Wallet_feeExempt
10. Wallet_limitExempt
11. Wallet_setLiquidityWallet
12. Wallet_setMarketingWallet
13. Wallet_whitelist
14. approve
15. decreaseAllowance
16. increaseAllowance
17. renounceOwnership
18. transfer
19. transferFrom
20. transferOwnership

Weakness/Vulnerabilities

SCAN RESULTS

SWC-129 —> Unencrypted Private Data On-Chain = PASSED

SWC-130 —> Code With No Effect = PASSED

SWC-131 —> Message Call with Hardcoded Gas Amount = PASSED

SWC-132 —> Hash Collisions with Multiple Variable Length Arguments = PASSED

SWC-133 —> Unexpected Ether Balance = PASSED

SWC-134 —> Presence of Unused Variables = PASSED

SWC-135 —> Righ-to-Left Override Control Character {U+202E} = PASSED

SWC-136 —> Typographical Error = PASSED

Weakness/Vulnerabilities

CONTINUED

SWC-119 —> Shadowing State Variables = PASSED

SWC=120 —> Weak Source of Randomness From Chain Attributes = PASSED

SWC-121 —> Missing Protection Against Signature Replay Attacks = PASSED

SWC-122 —> Lack of Proper Signature Verification = PASSED

SWC-123 —> Requirement Violation = PASSED

SWC-124 —> Write to Arbitrary Storage Location = PASSED

SWC-125 —> Incorrect Inheritance Order = PASSED

SWC-126 —> Insufficient Gas Griefing = PASSED

Weakness/Vulnerabilities

CONTINUED

SWC-127 —> Arbitrary Jump with Function Type Variable = PASSED

SWC=128 —> DoS with Block Gas Limit = PASSED

SWC-113 —> DoS with Failed Call = PASSED

SWC-114 —> Transaction Order Dependence = PASSED

SWC-115 —> Authorization Through Tx. Origin = PASSED

SWC-116 —> Block Values as a Value for Time = PASSED

SWC-117 —> Signature Malleability = PASSED

SWC-118 —> Incorrect Constructor Name = PASSED

Weakness/Vulnerabilities

CONTINUED

SWC-105 —> Unprotected Ether Withdrawal = PASSED

SWC=106 —> Unprotected SELF DESTRUCT Instruction = PASSED

SWC-107 —> Reentrancy = PASSED

SWC-108 —> State Variable Default Visibility = PASSED

SWC-109 —> Uninitialized Storage Pointer = PASSED

SWC-110 —> Assert Violation = PASSED

SWC-111 —> Use of Deprecated Solidity Functions = PASSED

SWC-112 —> Delegate Call to Untrusted Callee = PASSED

Weakness/Vulnerabilities

SWC-101 —> **Integer Overflow and Underflow = PASSED**

SWC=102 —> **Outdated Compiler Version = PASSED**

SWC-103 —> **Floating Pragma = PASSED**

SWC-104 —> **Unlocked Call Return Value = PASSED**

Low issue = Low-level weakness/vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution.

SOLHINT LINTER, Solidity Static Analysis using REMIX IDE **did not find** any serious issues.

Overall Assessment

Satisfactory

The **GenesisX50** has successfully passed the Pex Audit

Closing Notes

Whilst there are limitless ownable callable functions that have the potential to be dangerous, they are not overtly so. Trust in the team would mitigate many of these risks. Please make sure you do your own research. If in doubt please contact the project team.

Always make sure to inspect **all values and variables.**

This includes, but is not limited to: • Ownership • Proper Ownership Renouncement (if any) • Taxes • Transaction/Wallet Limits • Token Distributions • Timelocks • Liquidity Locks • Any other owner-adjustable settings or variables.