CHAIN AUDIT
PROFESSIONAL AUDIT

SMART CONTRACT AUDIT

BNB ALL-STARS

AUGUST 29, 2023

Prepared by
Francis K.

Approved by
Chain Audit Team

INTRODUCTION

AUDITING FIRM	CHAIN AUDIT
CLIENT FIRM	BNB ALL STARS
METHODOLOGY	MANUAL CODE REVIEW AND AUTOMATED ANALYSIS
LENGUAGE	SOLIDITY
CONTRACT	0x964C71Be189bc93A75aCF05179B5899558C58DC6
BLOCKCHAIN	BINANCE SMART CHAIN
CENTRALIZATION	YES
WEBSITE	https://bnb-allstars.com/

You can check the authenticity of this audit on our website.

EXECUTIVE SUMMARY

Objective and Scope

The purpose of this audit was to evaluate and ensure the integrity, security, and functionality of the "BNBAllStars" smart contract developed in Solidity for the Ethereum platform. A thorough review of the code was conducted, encompassing both manual and automated testing, with the aim of identifying and mitigating potential vulnerabilities and risks associated with the contract.

TestNet:
https://testnet.bscscan.com/address/0x964C71Be189bc93A75aCF05179B5899558C58DC6

Methodology

A combined review approach was adopted:

- **Manual Review:** Our team of Solidity experts carried out a detailed and systematic review of the source code, paying particular attention to common vulnerabilities and insecure code patterns. Each function and modularity of the contract, its internal logic, and interactions amongst them were assessed.

- **Automated Review:** We utilized top-tier automated audit tools to scan the code for known vulnerabilities, compilation errors, and other potential security issues.

STATUS	CRITICAL 🔴	MAJOR 🟠	MEDIUM 🟡	MINOR 🟢	UNKNOWN 🟤
OPEN	0	0	0	0	0
ACKNOWLEDGED	0	0	0	0	0
RESOLVED	0	0	0	0	0

IMPORTANCE OF SMART CONTRACT AUDITS

Smart contracts have ushered in a new era of trustless and decentralized operations on the blockchain. While they have the potential to revolutionize numerous sectors, from finance to supply chain, their immutable nature means that any vulnerability or flaw in their code is permanent once deployed. This underscores the crucial importance of smart contract audits.

Why are Audits Necessary?

1. **Immutability:** Once a contract is deployed, it cannot be changed. A bug or vulnerability can be exploited repeatedly unless it's fixed in a new version of the contract.
2. **Financial Implications:** Smart contracts often handle and manage valuable assets. Vulnerabilities can lead to substantial financial losses.
3. **Reputation:** A flawed contract can tarnish the reputation of a project, leading to a loss of trust and confidence among its users.
4. **Complexity:** Solidity, the primary language for Ethereum contracts, has its quirks. Even experienced developers might overlook subtleties that can become vulnerabilities.

Types of Smart Contract Attacks:

Smart contracts are vulnerable to a variety of attacks. Some of the most common include reentrancy attacks, overflow and underflow attacks, timestamp dependence attacks, and more. Each of these attacks exploits specific vulnerabilities in contract code, and a comprehensive audit aims to safeguard against all known vulnerabilities.

RISK CATEGORIES

Risk Type	Definition
Critical 🔴	A vulnerability that, if exploited, could have a catastrophic impact, potentially leading to substantial financial loss or irreversible damage to the contract's operations.
Major 🟠	A significant vulnerability that might not lead to total loss but can hamper the contract's functionality and compromise its objectives.
Medium 🟡	Issues that are of concern but might require specific conditions to be exploited. They can pose risks if combined with other vulnerabilities.
Minor 🟢	These vulnerabilities pose a limited threat and have a lower probability of being exploited. Often, they relate to best practices rather than direct exploitable flaws.
Unknown 🟤	Risks that haven't been fully understood or classified yet. They could be new or unique to the contract's specific design or context.

Status of Identified Risks

Status Type	Definition
Open	Vulnerabilities that have been identified but have not yet been addressed or rectified by the development team.
Acknowledged	The development team has recognized the issue but might be in the process of determining the best solution or mitigation strategy.
Resolved	The vulnerability has been effectively addressed and resolved by the development team, eliminating the risk it posed.

AUDITING IS AN ESSENTIAL STEP IN THE DEVELOPMENT AND DEPLOYMENT OF SMART CONTRACTS. IT ENSURES NOT ONLY THE SECURITY AND RELIABILITY OF THE CONTRACT BUT ALSO BUILDS TRUST AMONG ITS USERS AND STAKEHOLDERS. AS SMART CONTRACTS CONTINUE TO GROW IN COMPLEXITY AND IMPORTANCE, ROBUST AUDITING MECHANISMS WILL REMAIN A CORNERSTONE OF THE BLOCKCHAIN ECOSYSTEM.

IMPORTANT CONTRACT DETAILS

- **CHAIN:** BSC
- **TOKEN:** BNB
- **DAPP TYPE:** ROI DAPP
- **DAILY ROI PERCENTAGE:** 1.5%
- **MINIMUM INVESTMENT:** 0.1 BNB
- **MAXIMUM INVESTMENT:** 25 BNB
- **MAXIMUM WITHDRAWAL:** 200 BNB
- **MINIMUM AIRDROP:** 0.1 BNB
- **ACCUMULATION REWARDS LIMIT:** 200 BNB
- **MANDATORY REINVESTMENT:** 3 TIMES (STATIC - IRREVERSIBLE)
- **REINVESTMENT BONUS:** 1%

OWNER PRIVILEGES

THE SMART CONTRACT INCLUDES A SET OF FUNCTIONS THAT GRANT EXCLUSIVE PRIVILEGES TO THE CONTRACT OWNER. THESE FUNCTIONS ENABLE THE OWNER TO PERFORM CRITICAL ACTIONS THAT CAN AFFECT THE OPERATION OF THE CONTRACT AND ITS USERS. BELOW ARE THE FUNCTIONS AND THEIR RESPECTIVE ACTIONS:

1. CHANGE_OWNERSHIP

- DESCRIPTION: CHANGES THE ADDRESS OF THE CONTRACT OWNER.
- ACTION: ALLOWS THE OWNER TO TRANSFER CONTRACT OWNERSHIP TO A SPECIFIC ADDRESS.
- RESTRICTION: CAN ONLY BE EXECUTED BY THE CURRENT CONTRACT OWNER

2. CHANGE_PROJECT_WALLET

- DESCRIPTION: CHANGES THE PROJECT WALLET ADDRESS.
- ACTION: ENABLES THE OWNER TO CHANGE THE ADDRESS TO WHICH PROJECT FUNDS ARE SENT.
- RESTRICTION: CAN ONLY BE EXECUTED BY THE CURRENT CONTRACT OWNER.

3. ENABLE_AIRDROP

- DESCRIPTION: ENABLES THE AIRDROP FUNCTIONALITY.
- ACTION: ALLOWS THE OWNER TO ACTIVATE THE ABILITY TO PERFORM AIRDROPS IN THE CONTRACT (IRREVERSIBLE).
- RESTRICTION: CAN ONLY BE EXECUTED BY THE CURRENT CONTRACT OWNER.

THESE FUNCTIONS PROVIDE THE CONTRACT OWNER WITH A SIGNIFICANT LEVEL OF CONTROL OVER ITS OPERATION AND FEATURES. IT'S IMPORTANT FOR THE OWNER TO USE THESE FUNCTIONS WITH CARE AND RESPONSIBILITY, AS THEIR ACTIONS CAN DIRECTLY IMPACT USERS AND THE OVERALL FUNCTIONING OF THE CONTRACT. THE CONTRACT CLIENT IS ADVISED TO BE AWARE OF THESE FUNCTIONS AND THEIR IMPLICATIONS BEFORE DEPLOYING THE CONTRACT IN A PRODUCTION ENVIRONMENT.

CONTRACT OVERVIEW CHECKLIST

VULNERABILITY/PROBLEM DESCRIPTION	STATUS
Reentrancy Attack	Pass
Integer Overflow/Underflow	Pass
Delegatecall Vulnerability	Pass
Front-Running	Pass
Visibility of functions	Pass
Fallback Function Vulnerability	Pass
State Variable Mutable	Pass
Erroneous External Calls	Pass
Immutable Keyword Misuse	Pass
Storage Layout & Proxy Contracts	Pass
Timestamp Dependence	Pass
Compiler error	Pass
Short Address Attack	Pass
Using inline assembly	Pass
Weak sources of randomness	Pass
Gas limit and loops	Pass
Use of tx.origin	Pass

CONTRACT OVERVIEW CHECKLIST

VULNERABILITY/PROBLEM DESCRIPTION	STATUS
Oracle security	Pass
Malicious libraries	Pass
Missing event emission	Pass
Uninitialised Storage Pointers	Pass
Under-optimized Code	Pass
Inadequate Testing	Pass
Magic Numbers	Pass
Inadequate Permissions & Governance	Pass
Presence of unused code	Pass
Self-destruct interaction	Pass
User balance manipulation	Pass
Access Control and Authorization	Pass
Ownership Control	Pass
Assets Manipulation	Pass
Liquidity Access	Pass
Stop and Pause Trading	Pass
Missing zero address validation	Pass

CONTRACT OVERVIEW CHECKLIST

VULNERABILITY/PROBLEM DESCRIPTION	STATUS
Race Conditions	Pass
Sybil Attack	Pass
Data consistency	Pass
Divide before multiply	Pass
Unnecessary use of SafeMath	Pass
Solidity Naming Guides	Pass
Signature unique id	Pass
Optimize code & gas fee	Pass
Phishing with contract addresses	Pass
Array Length Manipulation	Pass
Unchecked Return Values	Pass
Forced Ether Reception	Pass
Transfer Block	Pass
Floating pragma	Pass
Deprecated solidity functions	Pass
Lack of Arbitrary limits	Pass
Incorrect Inheritance Order	Pass

CONTRACT OVERVIEW CHECKLIST

VULNERABILITY/PROBLEM DESCRIPTION	STATUS
Typographical Errors	Pass
Requirement Violation	Pass
Coding Style Violations	Pass
Third-Party Dependencies	Pass
Dos with revert Passed	Pass

CRITICAL VULNERABILITY FINDINGS

Vulnerability	Risk Type	Vulnerable	Reason	Affected Code
-	-	-	-	-

FINDINGS AND RECOMMENDATIONS

THE FOLLOWING ARE THE KEY FINDINGS IDENTIFIED DURING THE AUDIT, ALONG WITH CORRESPONDING RECOMMENDATIONS:

OVERFLOW AND UNDERFLOW HANDLING:

- THE CONTRACT IMPLEMENTS MEASURES TO PREVENT OVERFLOW AND UNDERFLOW IN CRITICAL MATHEMATICAL OPERATIONS, WHICH MITIGATES RISKS ASSOCIATED WITH THESE ISSUES.

2. INPUT VALIDATION:

- THE CODE INCLUDES VALIDATIONS TO PREVENT INVALID USER INPUTS, WHICH IS ESSENTIAL TO ENSURE THE INTEGRITY OF THE SYSTEM.

3. REFERRAL SYSTEM SECURITY:

- THE IMPLEMENTATION OF THE REFERRAL SYSTEM APPEARS ROBUST AND DOES NOT PRESENT EVIDENT ABUSE RISKS.

4. ADMINISTRATIVE FUNCTIONS:

- THE CONTRACT INCLUDES ADMINISTRATIVE FUNCTIONS THAT ALLOW THE OWNER TO CHANGE CONTRACT PROPERTIES AND FEATURES. THESE FUNCTIONS SHOULD BE USED WITH CAUTION AND TRANSPARENCY.

CONCLUSION

BASED ON THE REVIEW OF THE SMART CONTRACT, IT IS CONCLUDED THAT IT PRESENTS A SOLID AND SECURE IMPLEMENTATION. NO SIGNIFICANT VULNERABILITIES THAT COULD BE EXPLOITED THROUGH THE ANALYZED ATTACK VECTORS WERE IDENTIFIED. HOWEVER, IT IS RECOMMENDED TO CONTINUE PRACTICING A RIGOROUS SECURITY APPROACH AND CONDUCTING THOROUGH TESTING TO ENSURE THE INTEGRITY OF THE SYSTEM UNDER REAL-WORLD USAGE CONDITIONS.

SECURITY RATING 9/10

THE RATING ASSIGNED TO THIS SMART CONTRACT IS 9 ON A SCALE OF 0 TO 10. THIS RATING REFLECTS A ROBUST IMPLEMENTATION WITH PROPER SECURITY PRACTICES. FURTHER AUDITS ARE ENCOURAGED IN CASE OF FUTURE MODIFICATIONS TO ENSURE THE SECURITY AND OPERATION OF THE SYSTEM

THIS AUDIT REPORT IS BASED ON THE ANALYSIS OF THE PROVIDED SMART CONTRACT AND IMPLEMENTED SECURITY PRACTICES

CHAINAUDIT
FRANCIS K.
AUDITOR FOR CHAINAUDIT