SMART CONTRACT AUDIT

Conducted for: Helldiver

Waleed Ahmed
VULSIGHT
15/4/2024

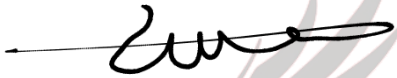
Contents

Executive Summary: 3
Project Background: 3
Audit Scope: 3
Audit Summary: 3
Severity Definitions: 4
Technical Checks: 5
Code Quality and Documentation: 5
Audit Findings: 5
Critical vulnerabilities: 5
Unstaking Logic Error ..... **Error! Bookmark not defined.**
High vulnerabilities: 5
Pricing Mismatch ..... **Error! Bookmark not defined.**
Minting Discrepancy: ..... **Error! Bookmark not defined.**
Medium vulnerabilities: 5
Low vulnerabilities: 6
Disclaimer: 6
Conclusion: 6

Executive Summary:

The client contacted our Agency to conduct a smart contract audit on their Solidity based smart contracts. One of the contract was the basis for their ERC-20 Token, While the other one provides the functionalities to facilitate the presale of the tokens. In our comprehensive audit of the contracts, we carefully examined several critical aspects including functionality, security vulnerabilities, best practices, and smart contract efficiency. The contract meticulously initiates a token presale process, utilizing phased pricing to incentivize early participation while implementing features like automatic staking upon purchase. The team didn't found any critical or high flag issue. Moreover, the analysis didn't revealed any mismatch between the Token description and Project contracts and structure. The audit was fully inspected all the potential issues and security presale process and safeguard participant contributions against vulnerabilities.

Audit conducted by: Waleed Ahmed



Project Background:

The smart contracts are written in solidity. The token contract is a simple ERC20 Contract with no additional functionalities. The presale contract has owner functionality within it. The contracts are deployed on the Binance testnet and after this audit in Binance Mainnet.

Audit Scope:

Deployed Addresses	**0x1364f0111b29bfcbbad854cd57c51cf560e8af7b** **0xf73a03c035ceedb155abe8beed5cf964be3a200b**
Chain	**Binance TestNet**
Language	**Solidity**
Audit Completion Date	**15/4/24**

Audit Summary:

- **Dynamic Pricing Mechanism:** The contract adopts a unique approach to pricing, utilising a predefined array of prices corresponding to distinct presale stages, influenced by the elapsed time since the presale launch. This strategy aims to reward early participants with lower prices.
- **Multi-currency Contributions:** Contributions can be made in the native currency (e.g., BNB/ETH) and the ERC20 stablecoins USDT and USDC, providing flexibility to participants. The conversion to the equivalent token amounts leverages a real-time price feed for the native currency, yet for

stablecoin contributions, it directly computes the number of tokens, showing an effective use of on-chain data and avoiding external dependency for stablecoin pricing.

- **Staking and Rewards:** The contract automatically stakes tokens purchased during the presale, with the APY and specific reward calculations encoded within. Impacting the token's economic model and overall presale integrity.

- **Ownership and Administrative Functions:** The contract vests huge security control for the presale early investors of the 15 stages of the presale. The presence of onlyOwner modifiers on sensitive functions mitigates unauthorized access risks.

- **Presale Status Management:** Dedicated functions to start and end the presale provide straightforward transitions between presale stages, including preparation for the open market.

- **Fund Withdrawal and Token Update Features:** Functionality dont allowing the owner to withdraw collected funds and update token addresses post-initial deployment introduces flexibility in managing and reallocating funds. While essential from an operational perspective and secury the early investors in Presale period.

Various tools such as Slither and different LLM scanners were used in the audit. Extensive manual code review was also done to ensure that any vulnerability if present could be detected in the audit.

We found:

- **0 Critical risk vulnerability**
- **0 High risk vulnerabilities**
- **0 Medium risk vulnerabilities**
- **0 Low risk vulnerabilities**

Severity Definitions:

Risk Level	Description
Critical	**Any kind of vulnerability that could lead to direct token or monetary loss**
High	**Any type of vulnerability that could disrupt the proper functioning of smart contract or indirectly lead to token or monetary loss.**
Medium	**Any type of vulnerability that could cause undesired actions but no serious disruption or monetary loss**

Low	Any type of vulnerability that doesn't have any significant impact on the execution of the proper functioning of contract

Technical Checks:

Checks	Result
Solidity version not specified	Passed
Solidity version too old	Passed
Integer overflow/underflow	Passed
Function input parameters check bypass	Passed
Insufficient randomness used	Passed
Fallback function misuse	Passed
Race Condition	Passed
Matching Project Specifications	Passed
Logical Vulnerabilities	Passed
Function visibility not explicitly declared	Passed
Use of deprecated keywords/functions	Passed
Out of Gas issue	Passed
Front Running	Passed
Insufficient Decentralization	Passed
Function input parameters lack of check	Passed
Proper function Access Control	Passed
Hardcoded Data	Passed

Code Quality and Documentation:

The audit scope included two smart contracts, which are written in Solidity programming language. The code is well indented and clear in nature. The code had sufficient commenting.

Audit Findings:

Critical vulnerabilities:

0 critical vulnerability was found in the smart contract.

High vulnerabilities:

0 high vulnerabilities were found in the smart contract.

Medium vulnerabilities:

Zero medium vulnerabilities were found in the smart contract.

Low vulnerabilities:

Zero low vulnerabilities were found in the smart contract.

Disclaimer:

The smart contract was tested on a best-effort basis. The smart contract was analyzed with the best security practices known at the time of writing this report. Due to the fact that the total number of test cases is unlimited and the fact that new vulnerabilities in technologies are discovered every day, the audit report makes no guarantees on the security of the contract. While we have done our best in testing this smart contract, it is recommended to not just rely on this audit report alone.

Conclusion:

The smart contract was tested extensively both automatically and manually. Zero Significant Vulnerabilities were discovered within it.