SolidSigma

HELPSTEPS (HSX)

Smart Contract Code Review and Security Analysis Report

Date

December 20 - 2023

Review

Contract Name: Helpsteps
Compiler Version: v0.8.19+commit.7dd6d404
Optimization: 200 runs
Platform/Chain : EVM | BSC
Language: Solidity
Timeline: 20/12/2023 - 21/12/2023

Review Scope

Repository	https://github.com/helpsteps/Helpsteps-Token-Contract
Commit	626fb71236f9b70c192baf44b68ab24532127d69

Audit Summary

9/10	10/10	8/10
Security Score	Code Quality Score	Documentation quality score

9/10

Total Score

3	0	3	0
Total Findings	Resolved	Acknowledged	Mitigated

Finding by severity

Critical	0
Major	0
Medium	2
Minor	1

Table of Contents

System Overview

Help Steps is a WEB3 Move2Earn application to promote health and charity. The application encourages users to help others while taking steps.

Welcome to HelpSteps, a groundbreaking platform that not only rewards users for embracing an active lifestyle but also empowers them to channel their physical activity towards meaningful causes. With our user-friendly mobile app, individuals can effortlessly monitor their daily movements and earn HS (HelpSteps) for every step they take. These HS can be directed towards supporting a range of charitable initiatives, sports clubs, individual campaigns, or exchanged for HSX (HelpSteps tokens) for increased flexibility and value extraction.

At HelpSteps, we believe that motivating people to lead healthier lives goes beyond the promise of incentives; it's about fostering a sense of community and purpose. Our platform transforms physical activity into an engaging game, motivating users to make positive choices every day. Whether your goal is personal wellness or contributing to a global cause, HelpSteps offers a unique and rewarding path to achieving it.

Name: Helpsteps
Symbol: $HSX
Decimals: 18
Total Supply: 1 billion tokens.

Executive Summary

This report offers a thorough examination and evaluation of the client's smart contract initiative

Documentation quality

The Documentation Quality score achieved 8 out of 10.

- Functional requirements can be found on the project's website at https://helpsteps.io
- Technical details are outlined in the project's whitepaper.

Code quality

The Code quality score achieved 10 out of 10.

Security score

Following the audit, no critical, no major, 2 medium and 1 minor severity issues were detected in the code, resulting in a flawless security score of 9 out of 10.
Comprehensive details regarding the identified issues can be found in the "Findings" section of this report.

Risks

- Initial Token Distribution - **Medium**
- Centralization Risks in Helpsteps.sol - **Medium**
- Improper Usage of public and external Type - **Minor**

Findings

Vulnerability Details

Initial Token Distribution - **Medium**

Description

Description:	During the contract deployment, all tokens are transferred to the one address. This presents a potential centralization risk as the one address holds sole control over token distribution without community consensus.
Location:	https://github.com/helpsteps/Helpsteps-Token-Contract/blob/626fb71236f9b70c192baf44b68ab24532127d69/Helpsteps.sol#L269
Status:	**Acknowledged**

Recommendations

Recommendation:	To mitigate this risk, it's advised that the team ensures transparency in the initial token distribution process. Additionally, efforts should be made to safeguard the private key access, thereby reducing the potential for centralized control.
Feedback	**[Helpsteps Team]:** After the contract is deployed, the tokens are sent to the Multisig Wallet. Here, multiple individuals have authority, and no single address can transact with the tokens held here. We will create vesting contract and most of the tokens will locked according to our tokenomics plan Check: https://bscscan.com/address/0xdcB6181E5B027583FCE32565250f620fCF3a7E87

Findings

Vulnerability Details

Centralization Risks in Helpsteps.sol - **Medium**

Description

Description: Within the Ownable contract, the _owner role holds control over the functions. Any breach of the _owner account could potentially empower a malicious actor to exploit this authority for transferring ownership.

Location: https://github.com/helpsteps/Helpsteps-Token-Contract/blob/626fb71236f9b70c192baf44b68ab24532127d69/Helpsteps.sol#L200

Status: Acknowledged

Recommendations

Recommendation: The risk assessment identifies areas in the project design needing security and decentralization enhancements. While full resolution may not be possible now, safeguarding privileged account keys is crucial to mitigate hacking risks. We recommend transitioning from centralized to decentralized mechanisms, such as smart-contract-based accounts like multisignature wallets, for improved security.

Feedback **[Helpsteps Team]:** We have utilized a Multisig wallet to safeguard the owner's private keys securely. The contract does not grant high owner privileges, adhering solely to the basic BEP20 token standard. The owner address can only execute transferOwnership and renounceOwnership functions if necessary in the future.

Findings

Vulnerability Details

Improper Usage of public and external Type - **Minor**

Description

Description: There are public functions in the contract that remain unused. These functions could be optimized by declaring them as external, as external functions offer greater efficiency compared to public functions.

Location: -

Status: Acknowledged

Recommendations

Recommendation: It is advisable to utilize the external attribute for public functions that are not invoked within the contract, thus enhancing efficiency and optimizing resource usage.

Feedback **[SolidSigma]:** The team acknowledged this issue and decided not to change the codebase this time.

Appendix

Severity	Description
Critical	Critical vulnerabilities are typically easy to exploit and can result in the loss of user funds or manipulation of contract state.
Major	Major vulnerabilities are more challenging to exploit, often requiring specific conditions, but they still pose a risk of user fund loss or contract state manipulation.
Medium	Medium vulnerabilities primarily involve state manipulations and rarely result in asset loss. This category also includes contradictions, requirements violations, and significant deviations from best practices.
Minor	Minor vulnerabilities involve major deviations from best practices or inefficient gas usage. While they don't significantly impact code execution or security scores, they can affect code quality scores.

Disclaimers

The smart contracts provided for audit have been thoroughly examined based on prevailing industry standards at the time of this report's drafting. The assessment encompasses cybersecurity vulnerabilities and issues within the smart contract source code, including details on compilation, deployment, and functionality, ensuring they perform their intended functions.

However, it's important to note that this report does not claim to identify all vulnerabilities or ensure the complete security of the code. It specifically pertains to the submitted and reviewed code and may become outdated after subsequent modifications.

Therefore, this report should not be considered a definitive or comprehensive assessment of the code's utility, safety, or bug-free status. We strongly advise supplementing this audit with independent assessments and a public bug bounty program to further ensure the security of the smart contracts.

It's essential to recognize that smart contracts operate within a blockchain ecosystem, which may have vulnerabilities in its platform, programming language, and related software. Consequently, the Consultant cannot guarantee the absolute security of the audited smart contracts.