QuillAudits

AUDIT REPORT

January 2025

For

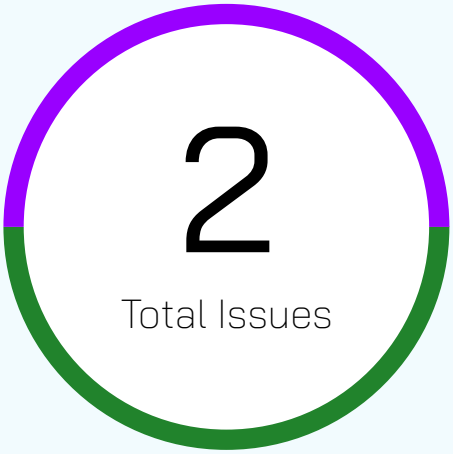
SocialGrowAI

Table of Content

Executive Summary

Project name	SocialGrowAI
Overview	SocialGrowAI is an ERC20 token with burnable capabilities, allowing token holders to reduce supply voluntarily. It's launched with an initial supply of 1 billion tokens, all minted to the owner at creation. The project includes a unique feature for the owner to recover ERC20 tokens accidentally sent to the contract, enhancing asset recovery and reducing token loss risks.
Project Website	https://socialgrowai.io
Audit Scope	The scope of this audit was to analyse the SocialGrowAI Token Contract for quality, security, and correctness.
Source Code	https://ether-scan.io/token/0x1C9B158e71BC274EA5519ca57A73E337Cac72b3A#c
Blockchain	Ethereum
Method	Manual Review, Functional Testing, Automated Testing, etc. All the raised flags were manually reviewed and re-tested to identify any false positives.
Review 1	13th January 2025
Review 2	NA
Fixed In	NA

Number of Issues per Severity



High	0 (0.00%)
Medium	0 (0.00%)
Low	1 (50.00%)
Informational	1 (50.00%)

Severity

Issues	High	Medium	Low	Informational
Open	0	0	0	0
Resolved	0	0	0	0
Acknowledged	0	0	**1**	**1**
Partially Resolved	0	0	0	0

Checked Vulnerabilities

- ☑ Re-entrancy
- ☑ Timestamp Dependence
- ☑ DoS with Block Gas Limit
- ☑ Transaction-Ordering Dependence
- ☑ Use of tx.origin
- ☑ Exception Disorder
- ☑ Balance Equality
- ☑ Compiler Version Not Fixed
- ☑ Transfer Forwards All Gas
- ☑ Send Instead of Transfer
- ☑ Style Guide Violation
- ☑ Unchecked External Call
- ☑ Unchecked Math
- ☑ Access Management
- ☑ Implicit Visibility Level
- ☑ Centralization of Control
- ☑ Logical Issues and Flaws
- ☑ Arithmetic Computations Correctness
- ☑ Race Conditions/Front Running
- ☑ Revert/Require Functions
- ☑ Private Modifier
- ☑ Missing Zero Address Validation

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.
- Implementation of ERC standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools And Platforms Used For Audit

Remix IDE, Foundry, Solhint, Mythril, Slither, Solidity statistical analysis.

Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational Issues

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

Security vulnerabilities identified that must be resolved and are currently unresolved.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/ impact of the security issue, but are not completely resolved.

Low Severity Issues

No Event Emission When a Critical Function is Called

Acknowledged

Path

https://etherscan.io/token/0x1C9B158e71BC274EA5519ca57A73E337Cac72b3A#code

Function

widthdrawToken()

Description

The withdrawToken function allows the owner to transfer tokens from the contract without emitting an event, making it difficult to track this critical operation.

Recommendation

Make Sure to emit an event when the state changes

Informational Severity Issues

Ownership Transfer should be a Two-way Process

Acknowledged

Description

Due to the importance of the contract owner being responsible for mint, and burn; it is important to stress the need for the use of a two-way process on the transfer of ownership. When the current owner invokes the transferOwnership function, passing the parameter of the new address, this sets the new address immediately, supposing it is not an address zero; hence, it would revert. However, the issue arises when the address passed was that of a wrong address, and this would not be redeemable anymore.

Recommendation

Use the Openzeppeling Ownable2StepUpgradable to remedy the issue of instantaneous transfer to the wrong address. This way, the assigned address would claim ownership first before the completion of the ownership transfer.

Functional Tests

Some of the tests performed are mentioned below:

- ✔ Should Get the Name of the Contract
- ✔ Should get the Symbol
- ✔ Should get the Owner of the Contract
- ✔ Should get the Decimal
- ✔ Should get the Total Supply
- ✔ Should be able to Transfer
- ✔ Should be able to do transfer from
- ✔ Should be able to RenounceOwnership
- ✔ Should be able to Transfer Ownership
- ✔ Should be able to approve
- ✔ Should be able to Burn
- ✔ Should be able to burn from
- ✔ OnlyOwner should be able to withdraw Token

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the SocialGrowAI Token contract. We performed our audit according to the procedure described above.

One issues of Low and informational severity were found, Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in SocialGrowAI smart contract. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of SocialGrowAI smart contract. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of the SocialGrowAI to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With six years of expertise, we've secured over 1000 projects globally, averting over $30 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

6+ Years of Expertise	**1M+** Lines of Code Audited
$30B+ Secured in Digital Assets	**1K+** Projects Secured

Follow Our Journey

AUDIT REPORT

January 2025

For

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com