Security Audit

Email: contact@dwebox.com
Website: https://dwebox.com

Introduction

https://shibatoken.com/
https://t.me/ShibaInu_Dogecoinkiller
https://twitter.com/Shibtoken

SHIBA INU is a 100% decentralized community experiment which claims that 1/2 the tokens have been sent to Vitalik and the other half were locked to a Uniswap pool and the keys burned.

Purpose of the Audit:

The purpose of the audit is to identify potential security vulnerabilities in the smart contracts of Shiba Inu (SHIB). The audit aims to ensure that the smart contracts are free of exploitable vulnerabilities that could lead to the loss of user funds, manipulation of the platform, or other types of security risks.

Scope of the Audit:

The scope of the audit will cover all smart contracts used by the platform, including their interactions with external contracts or protocols. The audit will focus on the following areas:

1. **Security**: The audit will identify security vulnerabilities in the smart contracts, including potential attack vectors such as reentrancy attacks, integer overflows/underflows, and other types of malicious code.
2. **Functionality**: The audit will evaluate the functionality of the smart contracts to ensure that they operate as intended and that all user actions are handled appropriately.
3. **Architecture**: The audit will evaluate the overall architecture of the smart contracts to ensure that they are properly designed, modularized, and follow industry best practices.
4. **Gas Optimization**: The audit will evaluate the smart contracts for efficient use of gas to ensure that the platform can scale without incurring high transaction fees.
5. **Compliance**: The audit will evaluate the smart contracts for compliance with regulatory requirements and industry standards, where applicable.

The audit will be conducted by experienced security professionals who will use a combination of manual and automated testing techniques to identify potential vulnerabilities. The audit report documents any issues found and provides recommendations for remediation.

Background information on the smart contract(s)

Shiba Inu: 0x95ad61b0a150d79219dcf64e1e6cc01f0b64c4ce

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are those that can lead to complete compromise of the system or data, and/or have significant financial or reputational impacts. Examples of critical vulnerabilities include unauthorized access to sensitive information, remote code execution, and denial of service attacks. These vulnerabilities require immediate attention and remediation.
HIGH	High severity vulnerabilities are those that can cause significant damage to the system or data, but do not necessarily result in complete compromise. These vulnerabilities require immediate attention and remediation.
MEDIUM	Medium severity vulnerabilities are those that may result in damage to the system or data, but are not as severe as high or critical severity vulnerabilities.
LOW	Low severity vulnerabilities are those that have a low impact on the system or data and are considered minor security issues. Examples of low severity vulnerabilities include outdated software, lack of input validation, and missing security headers. These vulnerabilities should be addressed in a reasonable time frame.
OPTIMIZATION	Code style issues refer to coding practices that do not necessarily result in security vulnerabilities, but may impact the quality of the code or its maintainability.

Audit Report

Critical

No **critical** vulnerability(ies) found ✅

High

No **high** vulnerability(ies) found ✅

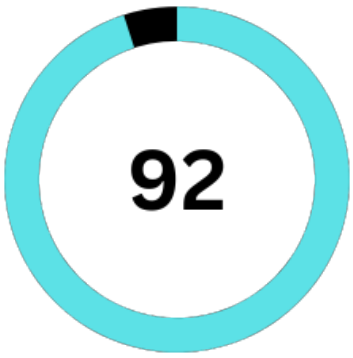
Medium

No **medium** vulnerability(ies) found ✅

Low

2 **low** vulnerability(ies) found and acknowledged ✅

RESULT



Vulnerability Information

Optimization related to Solidity Version.

The code uses v0.5.0+commit.1d4f565a which is an outdated solidity version. We recommend to upgrade to a higher version

Optimization related to “Burn”

When calling **Burn** function it calls the internal _burn method below. We recommend tracking in a separate variable the amounts of tokens burnt.

```
    function _burn(address account, uint256 value) internal {
        require(account != address(0), "ERC20: burn from the zero address");

        _totalSupply = _totalSupply.sub(value);
        _balances[account] = _balances[account].sub(value);
        emit Transfer(account, address(0), value);
    }
```

Approach

This report has been prepared to discover any issues or vulnerabilities within the source code of smart contracts. Our team uses both manual reviewing methodology and static analysis. This is a list of vulnerabilities that the smart contracts have been checked for:

- Honeypot vulnerability
- Reentrancy
- Business Logic
- Token Supply manipulation
- Kill-switch mechanisms
- User balance manipulation
- Unchecked external call
- Data consistency
- Any form of DoS
- Gas Limit
- Unchecked math
- Visibility level
- Integrity of digital assets

DISCLAIMER:

The cybersecurity report provided by Dwebox is intended to identify potential security vulnerabilities and provide recommendations for remediation. It is not intended to provide specific advice or recommendations for any individual or on any specific investment. Dwebox shall not be liable for any damage, loss or expense of any kind arising out of or resulting from any use or reliance on this report, including but not limited to, consequential, special, incidental, or indirect damages.

The report is based on the information and systems provided to Dwebox for review and analysis. While every effort has been made to ensure the accuracy of the report, Dwebox makes no representations or warranties, express or implied, as to the accuracy, completeness, reliability, suitability or availability of the report or the information contained therein.

Furthermore, Dwebox does not provide any warranty, express or implied, with respect to the servers used in the systems under review. The services are provided “AS IS” and “AS AVAILABLE” and with all faults and defects without warranty of any kind. Dwebox shall not be liable for any damage, loss or expense of any kind arising out of or resulting from any use or reliance on the servers.

Dwebox recognizes that blockchain technology and cryptographic assets entail ongoing risks. As such, it is the responsibility of each company and individual to conduct their own due diligence and maintain continuous security measures. Our objective is to assist in reducing attack vectors and minimizing the risks associated with utilizing new and continuously evolving technologies. We do not make any claims of guarantee regarding the security or functionality of the technology we analyze. Our role is to provide analysis and recommendations based on the information provided, with the understanding that technology is subject to ongoing change and development.

The report should not be considered as a substitute for professional advice or judgment, and users of the report should seek the advice of a professional if they have any specific concerns or questions regarding cybersecurity or investment decisions.

By using the report, users acknowledge that they have read this disclaimer, understood its terms, and agree to its terms and conditions.