DAudit

SMART CONTRACT REVIEW AND SECURITY REPORT

Audit

KYC

KYB
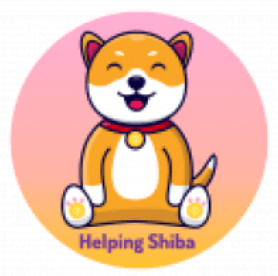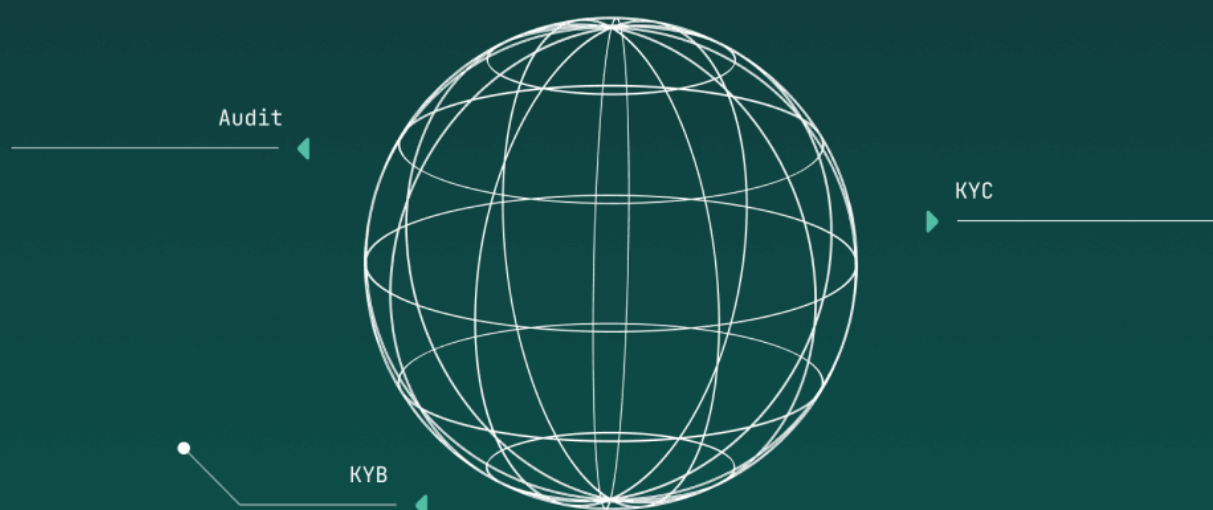
COMPLETED ON

JULY 14, 2022

Helping Shiba

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

OVERVIEW

This audit has been prepared for Helping Shiba to review their Smart Contract Code and Security. This audit report aims to help investors make an informative decision during the project research.

In this report, you will find a summarized review of the following key points:
✓ Contract's source code
✓ Contract's function
✓ Owner's wallets
✓ Important Technical Stats
✓ Good Practices
✓ Recommendation

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

▶ This Audit report DOES NOT guarantee nor reflect the outcome and goal of the project.

▶ DAudit's audit process only guarantees that the smart contract code has been verified not to have security breaches.

Table Of Content

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

CONTRACT INFORMATION

DAUDIT CONTRACT REVIEW PROCESS

Token Name
Helping Shiba

Symbol
HEBA

Contract Name
HELPINGSHIBA

Type
ERC-20

Website
https://helpingshiba.com/

Technical Documentation
https://helpingshiba.com/wp-content/
uploads/2022/07/Helping-Shiba-
Whitepaper-V1.pdf

Contract Address
0xe527cF9700aD4d7504572bfd3a7e418D
eb5c4fd8

Network
Binance Smart Chain

Language
Solidity

Compiler Version
v0.8.7+commit.e28d00a7

Optimization
Yes with 200 runs

Decimals
18

Total Supply
1,000,000,000

Smart Contract Code review process:
✓ Testing the smart contracts against both common and uncommon vulnerabilities.
✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
✓Ensuring contract logic meets the specifications and intentions of the client.
✓ Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
✓ Thorough line-by-line manual review of the entire codebase by industry experts.

DECENTRALAB PTE.LTD.

160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

PROJECT TECHNICAL INFORMATION

(AS OF JULY 14TH, 2022)

STATUS:

HAVEN'T LAUNCHED YET

Owner Address	0x318c95a9d555147574f093c9d97bfba0ab79a7a1
Marketing Wallet	0xa5221fd31465Eef07a5854Dd052F9808143aef5C
Advisor Wallet	0x7563eD5F97Af7b63000db05DD499632e5e0C346D
Ops Wallet	0x7FB31a56056C276E0Bb69a005467C4F58f954124
Charity Wallet	0x9aE5155F69Bc6070b89dDF5aD3f32Cf8E982A1F5
Dev Wallet	0x6985d4864221A4c4192D96f49062D537e9753333

LIQUIDITY

Status: Not added yet

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org
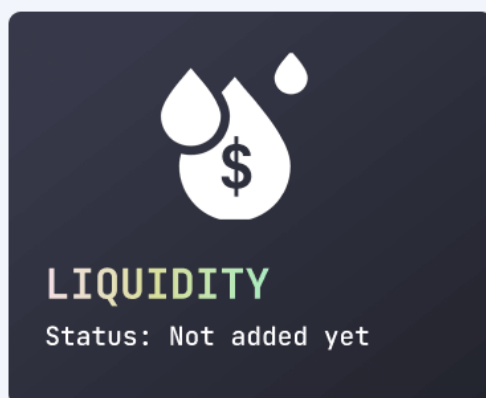
DAudit

DAudit

IMPORTANT STATS

TAX

Buy tax: 12%
Sell tax: 12%
Launch tax: 99%

OWNER CAN SET FEES

Buy up to 24%
Sell up to 24%

MAX TX AMOUNT

Owner can set max tx amount

OWNERSHIP

Owner can renounce or transfer ownership

MINT FUNCTION

No mint function found

PAUSE

Owner can only enable and disable trading pause once.

BLACKLIST

Owner can set blacklist

WHITELIST

Owner can set whitelist to avoid transaction fee
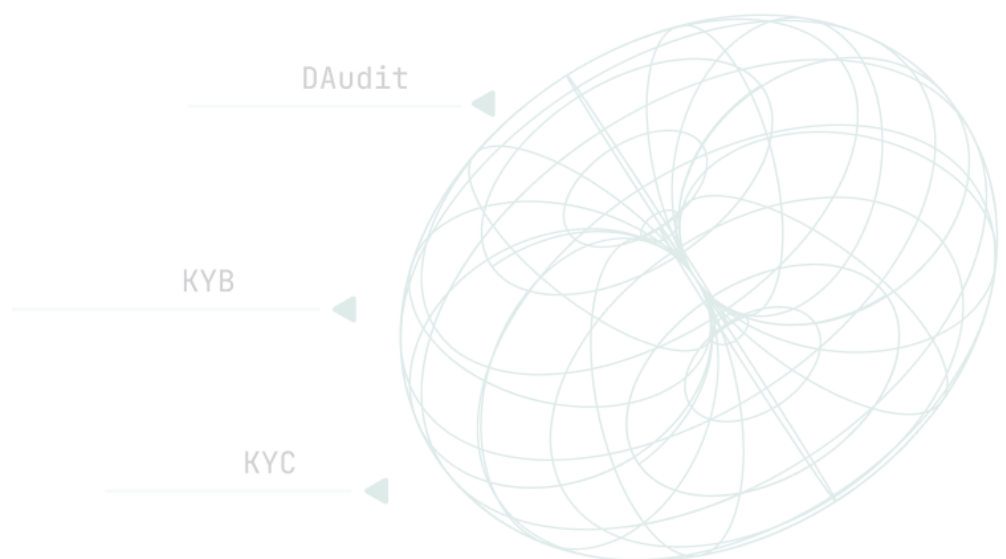
DAudit

VULNERABILITY CHECK

CODE REVIEW

Design Logic	Passed
Compiler Warnings	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle Calls	Passed
Front Running	Passed
DoS with block gas limit	Passed
DoS with Revert	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event Log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

DAudit

VULNERABILITY CHECK

FUNCTION REVIEW

Business Logics Review Functionality Checks	Passed
Access Control & Authorization	Passed
Escrow manipulation	Passed
Token Supply manipulation	Passed
Assets integrity	Passed
User Balances manipulation	Passed
Data Consistency manipulation	Passed
Kill - Switch Mechanism Operation Trails & Event Generation	Passed

DAudit

DAudit

KYB

KYC

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

DAudit

RISK LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

Critical

Critical vulnerabilities are usually straightforward to exploit and can lead to asset loss or data manipulations.

High

High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions

Medium

Medium-level vulnerabilities are important to fix; however, they can't lead to asset loss or data manipulations.

Low

Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

DAudit

RISK FOUND 01

CRITICAL _____

Owner can set blacklist user, through which any user can be prohibited from trading.

```solidity
function updateIsBot(address account, bool state) external onlyOwner {
    isBot[account] = state;
}

function bulkIsBot(address[] memory accounts, bool state) external onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        isBot[accounts[i]] = state;
    }
}
```

```solidity
require(
    !isBot[sender] && !isBot[recipient],
    "You can't transfer tokens"
);
```

Recommendation:

The blacklist function should not be used, this function can block the users' trading or transfer. Without this function, there could be more investors involved.

DAudit

RISK FOUND 02

CRITICAL

The owner can block trading/transfer, but wallets in the exemptFee list can still trade normally.

```solidity
if (!exemptFee[sender] && !exemptFee[recipient]) {
    require(tradingEnabled, "Trading not enabled");
}
```

```solidity
function updateExemptFee(address _address, bool state) external onlyOwner {
    exemptFee[_address] = state;
}

function bulkExemptFee(address[] memory accounts, bool state) external onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        exemptFee[accounts[i]] = state;
    }
}
```

Recommendation:

If you want to block trading sniperbot after listing, you should apply this function to all users.

DAudit

RISK FOUND 03

CRITICAL ──────────────

The contract uses the variable "deadline" to store the block number. If a user makes a transaction in between the starting block and the block specified in the "deadline" variable, then the following blocks will be charged 99%. The variable "deadline" is being defaulted to 1, but the owner can change the value of the variable before using the "enableTrading" function.

```
    } else if (useLaunchFee) {
        feeswap = launchtax;
        feesum = launchtax;
    }
```

```
uint256 private genesis_block;
uint256 private deadline = 1;
uint256 private launchtax = 99;
```

```
function updatedeadline(uint256 _deadline) external onlyOwner {
    require(!tradingEnabled, "Can't change when trading has started");
    deadline = _deadline;
}
```

Recommendation:

The variable "deadline" should be constant - cannot be changed.

DAudit

RISK FOUND 04

LOW _____

Owner can withdraw token and BNB from token contract

```
function rescueBNB(uint256 weiAmount) external onlyOwner {
    payable(devWallet).transfer(weiAmount);
}

function rescueBEP20Tokens(address tokenAdd, uint256 amount) external onlyOwner {
    IERC20(tokenAdd).transfer(devWallet, amount);
}
```

Recommendation:

This action should be performed automatically by smart contract and should not be intervened manually.

DAudit

HELPING SHIBA GOOD PRACTICES FOUND

1

The smart contract utilizes "SafeMath" to prevent overflows.

2

The owner cannot stop or pause the contract.

DECENTRALAB PTE.LTD.
160 Robinson Road, #14-04 Singapore
Singapore (068914)
support@daudit.org

DAudit

ABOUT DAUDIT

DAudit offers Smart Contract vulnerability and quality testing services at a rapid pace to ensure that projects do not fall behind the market.

Experienced	Fast	Careful	Affordable
A group of experienced blockchain developers built many successful DApp applications and are familiar with security flaws.	Within 6 hours, the audit report will be on your desk! We also have professional consultation and support staff available around the clock.	We deeply analyze the smart contracts line by line and cover the smart contracts with both automated and manual testing.	Affordable We provide the most competitive price in the industry, with audit reports ranging from $500 to $1,000, KYC services start at $1000, and KYB services start at $2,000

CONTACT US

Email

support@daudit.org

Support 24/7

@daudit (Mr.Drake)
@vietdn (Mr.Viet)

DAudit

DISCLAIMER

DAudit Disclaimer

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or prin and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

The smart contracts submitted for audit were examined in accordance with best industry practices at the time of this report in terms of cybersecurity vulnerabilities and issues in smart contract source code, which are detailed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no claims or guarantees about the code's security. It also cannot be deemed an adequate appraisal of the code's utility and safety, bug-free status, or any other contractual assertions. While we did our best in completing the study and generating this report, it is crucial to emphasize that you should not rely only on this report; we advocate doing many independent audits and participating in a public bug bounty program to assure smart contract security.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed

Technical Disclaimer

Smart Contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.