



AUDIT REPORT

SecureWise

SMART CONTRACT AUDIT



 <https://github.com/securewise>
 <https://t.me/securewise>
 <https://securewise.info/>



Table of Contents

03

Disclaimer

04

Overview

05

Quick Result

06

Auditing
Approach and
Methodologies

07

Automated
Analysis

09

Inheritance
Graph

10

Contract
Summary

11

Manual Review



Disclaimer

SecureWise provides the smart contract audit of solidity. Audit and report are for informational purposes only and not, nor should be considered, as an endorsement to engage with, invest in, participate, provide an incentive, or disapprove, criticise, discourage, or purport to provide an opinion on any particular project or team.

This audit report doesn't provide any warranty or guarantee regarding the nature of the technology analysed. These reports, in no way, provide investment advice, nor should be used as investment advice of any sort. Investors must always do their own research and manage their risk.

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 printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and SecureWise and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) SecureWise owe no duty of care towards you or any other person, nor does SecureWise make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SecureWise hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SecureWise hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SecureWise, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

Overview

Token Name: SLEEPZ (**SLEEPZ**)

Methodology: Automated Analysis, Manual Code Review

Language: Solidity

Contract Address: 0xbA0F64792ff4bfFB7084E7b8da9C0Ac3b4784F37

ContractLink: <https://bscscan.com/address/0xbA0F64792ff4bfFB7084E7b8da9C0Ac3b4784F37>

Network: Binance Smart Chain (BSC)

Decimals: 18

Supply: 10,000,000

Website: <https://sleepz.ai/>

Twitter: -

Telegram: https://t.me/sleepz_official

Report Date: September 13, 2022

Quick Result

SecureWise has applied the automated and manual analysis of Smart Contract and were reviewed for common contract vulnerabilities and centralized exploits

Owner Privileges



Trading must be enabled by the owner



The owner can exclude accounts from fees




The owner can set fees with limit up to 5%



The owner can change swap settings



The owner can withdraw any token from the contract but can't withdraw native token



SLEEPZ (SLEEPZ) has succesfully **PASSED** the smart contract audit with **LOW** severity issue

Auditing Approach and Methodologies

SecureWise has performed starting with analyzing the code, issues, code quality, and libraries. Reviewed line-by-line by our team. Finding any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

Methodology

- Understanding the size, scope and functionality of your project's source code
- Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Testing and automated analysis of the Smart Contract to determine proper logic has been followed throughout the whole process
- Deploying the code on testnet using multiple live test
- Analyzing a program to determine the specific input that causes different parts of a program to execute its functions.
- Checking whether all the libraries used in the code are on the latest version.

Goals

Smart Contract System is secure, resilient and working according to the specifications and without any vulnerabilities.



Risk Classification






























































High: Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

Medium: Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fixed as soon as possible.

Low: Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.

Automated Analysis

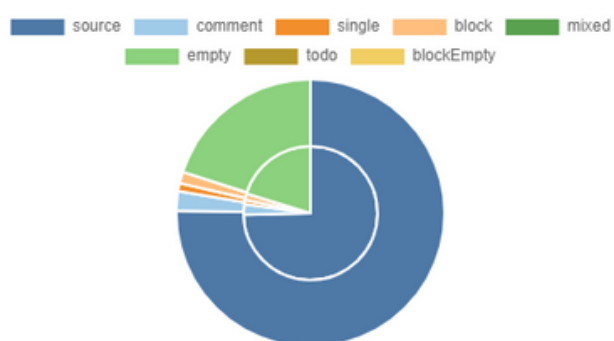
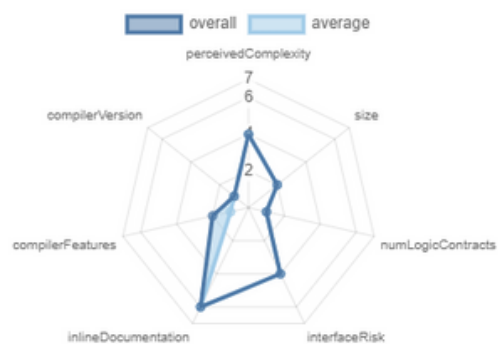
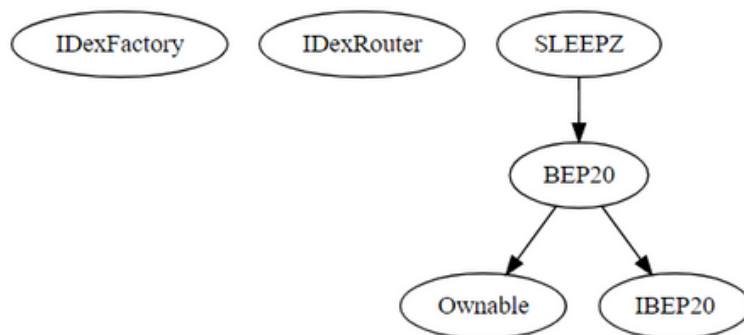
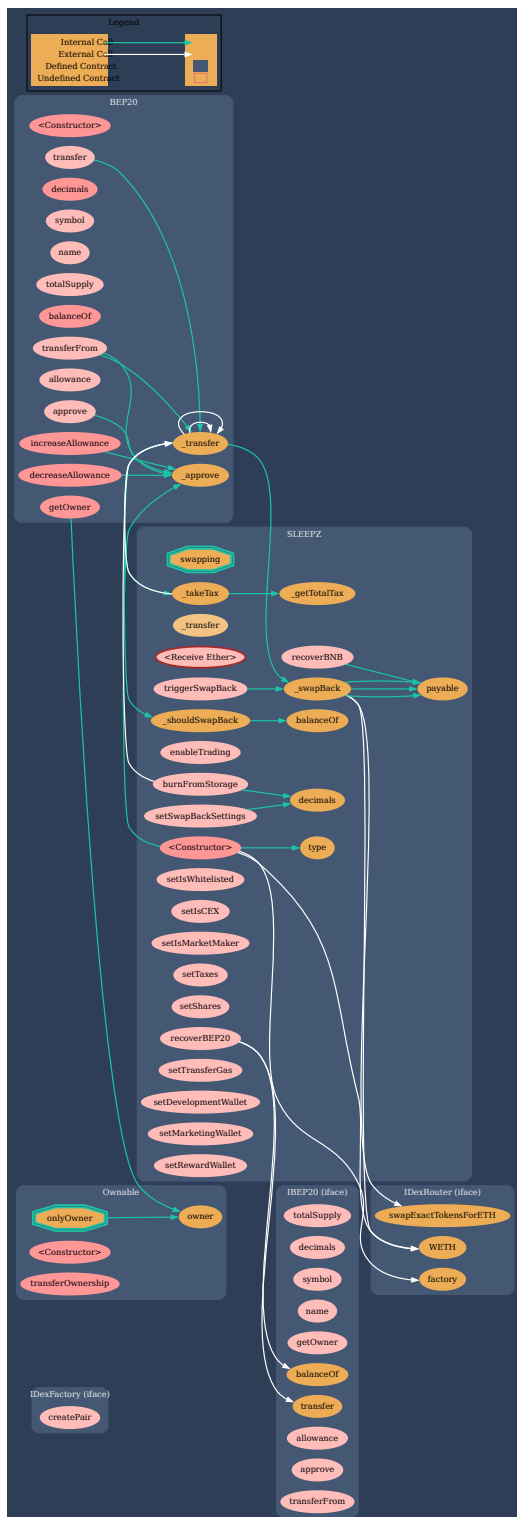
| Symbol | Meaning |
|---|---------------------------|
|  | Function can modify state |
|  | Function is payable |

| | | | | |
|--------------------|-----------------------|--|---|--|
| IDexFactory | Interface | | | |
| L | createPair | External  |  | NO  |
| IDexRouter | Interface | | | |
| L | factory | External  | | NO  |
| L | WETH | External  | | NO  |
| L | swapExactTokensForETH | External  |  | NO  |
| Ownable | Implementation | | | |
| L | | Public  |  | NO  |
| L | transferOwnership | Public  |  | onlyOwner |
| L | owner | Internal  | | |
| IBEP20 | Interface | | | |
| L | totalSupply | External  | | NO  |
| L | decimals | External  | | NO  |
| L | symbol | External  | | NO  |
| L | name | External  | | NO  |
| L | getOwner | External  | | NO  |
| L | balanceOf | External  | | NO  |
| L | transfer | External  |  | NO  |
| L | allowance | External  | | NO  |
| L | approve | External  |  | NO  |
| L | transferFrom | External  |  | NO  |
| BEP20 | Implementation | IBEP20, Ownable | | |
| L | | Public  |  | Ownable |
| L | getOwner | Public  | | NO  |
| L | decimals | Public  | | NO  |
| L | symbol | External  | | NO  |
| L | name | External  | | NO  |
| L | totalSupply | External  | | NO  |
| L | balanceOf | Public  | | NO  |
| L | transfer | External  |  | NO  |
| L | allowance | External  | | NO  |
| L | approve | External  |  | NO  |



Automated Analysis

| | | | | |
|---------------|----------------------|------------|---|-----------|
| L | transferFrom | External ! | ● | NO ! |
| L | increaseAllowance | Public ! | ● | NO ! |
| L | decreaseAllowance | Public ! | ● | NO ! |
| L | _transfer | Internal 🔒 | ● | |
| L | _approve | Internal 🔒 | ● | |
| | | | | |
| SLEEPZ | Implementation | BEP20 | | |
| L | | Public ! | ● | BEP20 |
| L | _transfer | Internal 🔒 | ● | |
| L | | External ! | 🟢 | NO ! |
| L | _takeTax | Private 🔒 | ● | |
| L | _getTotalTax | Private 🔒 | | |
| L | _shouldSwapBack | Private 🔒 | | |
| L | _swapBack | Private 🔒 | ● | swapping |
| L | enableTrading | External ! | ● | onlyOwner |
| L | triggerSwapBack | External ! | ● | onlyOwner |
| L | burnFromStorage | External ! | ● | onlyOwner |
| L | recoverBNB | External ! | ● | onlyOwner |
| L | recoverBEP20 | External ! | ● | onlyOwner |
| L | setIsWhitelisted | External ! | ● | onlyOwner |
| L | setIsCEX | External ! | ● | onlyOwner |
| L | setIsMarketMaker | External ! | ● | onlyOwner |
| L | setTaxes | External ! | ● | onlyOwner |
| L | setShares | External ! | ● | onlyOwner |
| L | setSwapBackSettings | External ! | ● | onlyOwner |
| L | setTransferGas | External ! | ● | onlyOwner |
| L | setDevelopmentWallet | External ! | ● | onlyOwner |
| L | setMarketingWallet | External ! | ● | onlyOwner |
| L | setRewardWallet | External ! | ● | onlyOwner |

Inheritance Graph



Contract Summary


| Logic Contracts | Interfaces | Lines | nLines | nSLOC | Comment Lines | Complex. Score | Capabilities |
|-----------------|------------|-------|--------|-------|---------------|----------------|---|
| 3 | 3 | 415 | 380 | 291 | 10 | 283 |  |
| 3 | 3 | 415 | 380 | 291 | 10 | 283 |  |

Components

| | | | |
|---|---|--|--|
|  Contracts |  Libraries |  Interfaces |  Abstract |
| 2 | 0 | 3 | 1 |

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.





| | |
|--|---|
|  Public |  Payable |
| 43 | 1 |







| External | Internal | Private | Pure | View |
|----------|----------|---------|------|------|
| 37 | 38 | 4 | 6 | 13 |


StateVariables

| | |
|-------|--|
| Total |  Public |
| 28 | 17 |

Capabilities

| | | | | |
|----------------------------|---|---|---|---|
| Solidity Versions observed |  Experimental Features |  Can Receive Funds |  Uses Assembly |  Has Destroyable Contracts |
| 0.8.13 | | yes | | |

| | | | | | |
|---|---|--|---|--|--|
|  Transfers ETH |  Low-Level Calls |  DelegateCall |  Uses Hash Functions |  ECRecover |  New/Create/Create2 |
| yes | | | | | |

| | |
|--|-------------|
|  TryCatch | Σ Unchecked |
| | |

Manual Review

The owner can exclude accounts from fees

```
349 function setIsWhitelisted(address account, bool value) external onlyOwner {
350     isWhitelisted[account] = value;
351     emit SetWhitelisted(account, value);
352 }
353
354 function setIsCEX(address account, bool value) external onlyOwner {
355     isCEX[account] = value;
356     emit SetCEX(account, value);
357 }
```

Recommendation

Authorizing privileged roles to exclude accounts from fees. These cause can affect decentralization.

The owner can change swap settings

```
385 function setSwapBackSettings(bool enabled, uint256 amount) external onlyOwner {
386     uint256 tokenAmount = amount * 10**decimals();
387     swapEnabled = enabled;
388     swapThreshold = tokenAmount;
389     emit SetSwapBackSettings(enabled, amount);
390 }
```

Recommendation

Authorizing privileged roles to enable or disable the swap. These cause can affect decentralization.

The owner can withdraw any token from the contract

```
335 function recoverBNB() external onlyOwner {
336     uint256 amount = address(this).balance;
337     (bool sent,) = payable(marketingWallet).call{value: amount, gas: transferGas}("");
338     require(sent, "Tx failed");
339     emit RecoverBNB(amount);
340 }
341
342 function recoverBEP20(IBEP20 token, address recipient) external onlyOwner {
343     require(address(token) != address(this), "Can't withdraw SLEEPZ");
344     uint256 amount = token.balanceOf(address(this));
345     token.transfer(recipient, amount);
346     emit RecoverBEP20(address(token), amount);
347 }
```




Recommendation

Authorizing privileged roles to withdraw any token from the contract. Except native token. These cause can affect decentralization.

AUDIT REPORT

SecureWise

SMART CONTRACT AUDIT

 <https://github.com/securewise>
 <https://t.me/securewise>
 <https://securewise.info/>

