



Smart Contract Security Audit

Project: ShinobiVerse

May 20, 2022



Contract Address

0x1532C74250DE406a83FEc3ACc8030Da4159e4cbB

Table of Contents

- 1 Disclaimer**
- 2 Audit Review**
- 3 Project Review**
- 4 Smart Contract Vulnerability Checks**
- 5 Manual Code Review**
- 6 Owner Privileges**
 - 6.1 Contract Ownership
 - 6.2 Liquidity Overview
- 7 Tokenomics**
- 8 Social Media Check**
- 9 Website Review**
- 10 Audit Conclusion**

Disclaimer

The contents of this report reflect only the CRACKEN TECH audit team's understanding of the current progress and status of the security of the code audited, to verify the integrity of the code provided for the scope of this audit. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Given the size of the project, the findings detailed here are not to be considered exhaustive, and further testing and audit are recommended after the issues covered are fixed. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report.

The review does not address the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. If the audited source files are smart contract files, risks or issues introduced by using data feeds from off-chain sources are not extended by this review either.

Audit Review

The source code of the ShinobiVerse was audited in order to acquire a clear impression of how the project was implemented. The Cracken Tech audit team conducted in-depth research, analysis, and scrutiny, resulting in a series of observations. A detailed list of each issue found, and vulnerabilities in the source code will be included in the audit report. The problems and potential solutions are given in this report, we will identify common sources for such problems and comments for improvement.

The auditing process will follow a routine as special considerations by Cracken:

- Review of the specifications, sources, and instructions provided to Cracken to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Cracken describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analyzing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

Project Review

Token Summary

Parameter	Result
Token Name	ShinobiVerse
Token Symbol	SHINO
Token Decimal	9
Total Supply	45,000,000,000
Platform	BSC
Buy Tax Fee	6%
Sell Tax Fee	6%
Contract Creation Date	May 8, 2022
Liquidity Status	Locked
Liquidity Lockup Time	365 Days after listing
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	Yes with 200 runs
Contract Address	0x1532C74250DE406a83FEc3ACc8030Da4159e4cbB
Deployer Address	0x8A911e1afF89a0A58E224Da43E8E4D8A4d756614
Owner Address	0x4eaf5492838f34aaf6a5e1c603872da94baedc7d

Source Code

CRACKEN was commissioned by ShinobiVerse to perform an audit based on the following smart contract:

<https://bscscan.com/address/0x1532C74250DE406a83FEc3ACc8030Da4159e4cbB>





Smart Contract Vulnerability Checks

Vulnerability	Auto-Scan	Manual-Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	Low / No Risk
Code With No Effects	Complete	Complete	Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	Low / No Risk
Hash Collisions with Multiple Variable Length Arguments	Complete	Complete	Low / No Risk
Unexpected Ether balance	Complete	Complete	Low / No Risk
Presence of unused variables	Complete	Complete	Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	Low / No Risk
Typographical Error	Complete	Complete	Low / No Risk
DoS With Block Gas Limit	Complete	Complete	Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	Low / No Risk
Insufficient Gas Grieving	Complete	Complete	Low / No Risk
Incorrect Inheritance Order	Complete	Complete	Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	Low / No Risk
Requirement Violation	Complete	Complete	Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	Low / No Risk
Authorization through tx. origin	Complete	Complete	Low / No Risk
Delegate call to Untrusted Callee	Complete	Complete	Low / No Risk



Vulnerability	Auto-Scan	Manual-Scan	Result
Use of Deprecated Solidity Functions	Complete	Complete	Low / No Risk
Assert Violation	Complete	Complete	Low / No Risk
Reentrancy	Complete	Complete	Low / No Risk
Unprotected SELF-DESTRUCT Instruction	Complete	Complete	Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	Low / No Risk
Outdated Compiler Version	Complete	Complete	Low / No Risk
Integer Overflow and Underflow	Complete	Complete	Low / No Risk
Function Default Visibility	Complete	Complete	Low / No Risk

Manual Code Review

Classification of Issues

Severity	Description
 High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
 Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
 Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
 Information	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
 High-Risk	2
 Medium-Risk	0
 Low-Risk	1
 Information	0
Total	3

● **High-Risk:** Implementation of corrective actions as soon as possible.

Buy/Sell Max Tx Amount

Description:

Can set maximum buy & sell amount, will cause token sell unsuccessfully.

- ```
function setBuyMaxTxAmount(uint256 bMaxTxAmount) external onlyOwner {
 _bMaxTxAmount = bMaxTxAmount;
 require(_bMaxTxAmount > _tTotal.div(1000), "Too less limit");
}
```

```
function setSellMaxTxAmount(uint256 sMaxTxAmount) external onlyOwner {
 _sMaxTxAmount = sMaxTxAmount;
 require(_sMaxTxAmount > _tTotal.div(1000), "Too less limit");
}
```

**Recommendation: set maximum limit with a reasonable amount**

- **High-Risk: Implementation of corrective actions as soon as possible.**

## Lock & Unlock

Description:

The contract owner can use the lock function to transfer the ownership to the dead address with a set time, and then use the unlock function to retrieve it after the set time is reached.

- ```
function lock(uint256 time) public virtual onlyOwner {  
    _previousOwner = _owner;  
    _owner = address(0);  
    _lockTime = block.timestamp + time;  
    emit OwnershipTransferred(_owner, address(0));  
}
```



```
function unlock() public virtual {  
    require(_previousOwner == msg.sender, "You don't have permission to unlock");  
    require(block.timestamp > _lockTime, "Contract is locked until 7 days");  
    emit OwnershipTransferred(_owner, _previousOwner);  
    _owner = _previousOwner;  
}
```

Recommendation: remove the function

- **Low-Risk:** Implementation of certain corrective actions or accepting the risk.

Set Buy / Sell Fees

Description:

Tax Fee, Vault Fee, Reward Fee, Liquidity Fee, Marketing Fee, and Development Fee are less than 25%.

- ```
function setAllBuyAndTxFees(uint256 taxFee, uint256 vaultFee, uint256
 rewardFee, uint256 liquidityFee,
 uint256 marketingFee, uint256 developmentFee) external onlyOwner
{
 _buyTaxFee = taxFee;
 _buyVaultFee = vaultFee;
 _buyRewardFee = rewardFee;
 _buyLiquidityFee = liquidityFee;
 _buyMarketingFee = marketingFee;
 _buyDevelopmentFee = developmentFee;
 _buyAllSwapableFees =
 _buyRewardFee+_buyLiquidityFee+_buyMarketingFee+_buyDevelopmentFee;
 uint256 _totalFees = _buyTaxFee+_buyVaultFee+_buyAllSwapableFees;
 require(_totalFees<=25, "Too High Fee");
}
```
- ```
function setAllSellFees(uint256 taxFee, uint256 vaultFee, uint256 rewardFee,
    uint256 liquidityFee,
    uint256 marketingFee, uint256 developmentFee) external onlyOwner
{
    _sellTaxFee = taxFee;
    _sellVaultFee = vaultFee;
    _sellRewardFee = rewardFee;
    _sellLiquidityFee = liquidityFee;
    _sellMarketingFee = marketingFee;
    _sellDevelopmentFee = developmentFee;
    _sellAllSwapableFees =
    _sellRewardFee+_sellLiquidityFee+_sellMarketingFee+_sellDevelopmentFee;
    uint256 _totalFees = _sellTaxFee+_sellVaultFee+_sellAllSwapableFees;
    require(_totalFees<=25, "Too High Fee");
}
```

Recommendation: fees are set to a reasonable amount

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
removeLiquidity	None	External
setBuyMaxTxAmount	Uint256 bMaxTxAmount	External
setSellMaxTxAmount	Uint256 sMaxTxAmount	External
setAllBuyAndTxFees	Uint256 taxfee, vaultFee, rewardFee, liquidityFee, marketingFee, developmentFee	External
setAllSellFees	Uint256 taxfee, vaultFee, rewardFee, liquidityFee, marketingFee, developmentFee	External
excludedFromFee	_isExcludedFromFee	Private
minimumTokenbeforeSwap	None	Private
tokenFromReflection	Uint256 rAmount	Public
removeAllFee	None	Private
restoreAllFee	None	Private

Contract Ownership

The contract ownership of ShinobiVerse is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address `0x4eaf5492838f34aaf6a5e1c603872da94baedc7d` which can be viewed: [HERE](#)

The owner wallet has the power to call the functions displayed on the privileged functions list above, if the owner wallet is compromised these privileges could be exploited.

We recommend the team renounce ownership at the right timing if possible, or gradually migrate to a time lock with governing functionalities in respect of transparency and safety considerations.

Liquidity Overview

Liquidity Information

Parameter	Result
Pair Address	Not available
SHINO Reserves	0.00 SHINO
BNB Reserves	0.00 BNB
Liquidity Value	\$0 USD
Liquidity Ownership	The token does not have liquidity at the moment of the audit

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	PinkSale: PinkLock	22,500,000,000	50.0000%
2	0xc5175b363763047d456abcfefd06198d6e32fee0	18,178,200,000	40.3960%
3	0xc4cb187c2b0f1c691ad4127cfbc9a69dcb4afb1	3,984,300,000	8.8540%

Social Media Check

Social Media Type	Link	Result
Website	https://shinobiverse.io/	Checked
Twitter	https://twitter.com/Shinobiverse_io	Checked
Telegram	https://t.me/shinobiversegame	Checked
Facebook	https://twitter.com/Shinobiverse_io	Checked
Discord	https://discord.com/invite/gZCHRqhKuV	Checked
Reddit	https://www.reddit.com/r/shinobiversegame/	Checked

Website Review



- Mobile Friendly
- Contains no code errors
- SSL Secured
- No spelling errors

Audit Conclusion

- The owner can change router and pair addresses
- The owner can change buy/sell Max Tx Amount
- The owner can lock and unlock the function to fake renounce the ownership of the contract
- The owner can exclude from the fee
- The owner can change fees
- The owner can withdraw BNBs
- The owner can manually swap and liquify
- The owner can disable fees
- The owner can change the minimum number of tokens before the swap
- The owner can change marketing, development, busd, and vault addresses