



Smart Contract Security Audit

Project: BNBhare

Mar 06, 2023



Contract Address

0x6f8768ae9273473fc067943c55974cbfd2a8b6be

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 BNBhare 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	BNBhare
Token Symbol	BNBhare
Token Decimal	9
Total Supply	10,000,000,000,000,000,000,000
Platform	BSC
Buy Tax Fee	9%
Sell Tax Fee	9%
Contract Creation Date	Jan 09, 2022
Liquidity Status	75% of liquidity is locked
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	No with 200 runs
Contract Address	0x6f8768ae9273473fc067943c55974cbfd2a8b6be
Deployer Address	0x413a3789fbcbcd03346feef2c60d6f7fd2b0dd87
Owner Address	0x00

Source Code

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

<https://bscscan.com/address/0x6F8768Ae9273473fC067943C55974cbfd2A8b6BE>





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.
 Informational	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
 High-Risk	3
 Medium-Risk	0
 Low-Risk	0
 Informational	0
Total	3

● High-Risk: functions make cause the rug or scam project. **Must be fixed.**

Set max buy / sell tax fee

Description:

The owner can change the buy & sell fees up to 100%

[HIGH-RISK]

```
function setBuyTaxes(uint256 newLiquidityTax, uint256 newMarketingTax, uint256
newTeamTax) external onlyOwner() {
    _buyLiquidityFee = newLiquidityTax;
    _buyMarketingFee = newMarketingTax;
    _buyTeamFee = newTeamTax;

    _totalTaxIfBuying = _buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee);
}
```

```
function setSellTaxes(uint256 newLiquidityTax, uint256 newMarketingTax, uint256
newTeamTax) external onlyOwner() {
    _sellLiquidityFee = newLiquidityTax;
    _sellMarketingFee = newMarketingTax;
    _sellTeamFee = newTeamTax;

    _totalTaxIfSelling = _sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee);
}
```

Recommendation:

We recommend adding a requirement to limit the max fee amount.

● High-Risk: functions make cause the rug or scam project. **Must be fixed.**

Set Max Tx Amount

Description:

The owner can set max tx amount without limited [HIGH-RISK]

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {  
  
    _maxTxAmount = maxTxAmount;  
  
}
```

Recommendation:

We recommend removing the trading function for safety.

- High-Risk: functions make cause the rug or scam project. **Must be fixed.**

The Retrieve Ownership exists

Description:

The owner can get the ownership back from fake renouncement
[HIGH-RISK]

```
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:

We recommend that the owner should disable the fake renouncement function.

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
approve	address spender, uint256 amount	External
changeRouterVersion	address newRouterAddress	Public
decreaseAllowance	address spender, uint256 subtractedValue	Public
enableDisableWalletLimit	bool newValue	External
increaseAllowance	address spender, uint256 addedValue	Public
lock	uint256 time	Public
setBuyTaxes	uint256 newLiquidityTax, uint256 newMarketingTax, uint256 newTeamTax	External
setDistributionSettings	uint256 newLiquidityShare, uint256 newMarketingShare, uint256 newTeamShare	External
setIsExcludedFromFee	address account, bool newValue	Public
setIsTxLimitExempt	address holder, bool exempt	External
setIsWalletLimitExempt	address holder, bool exempt	External
setMarketPairStatus	address account, bool newValue	Public
setMarketingWalletAddress	address newAddress	External
setMaxTxAmount	uint256 maxTxAmount	External
setNumTokensBeforeSwap	uint256 newLimit	External
setSellTaxes	uint256 newLiquidityTax, uint256 newMarketingTax, uint256 newTeamTax	External
setSwapAndLiquifyByLimitOnly	bool newValue	Public
setSwapAndLiquifyEnabled	bool _enabled	Public



setTeamWalletAddress	address newAddress	External
setWalletLimit	uint256 newLimit	External
transfer	address recipient, uint256 amount	Private
transferFrom	address sender,address recipient,uint256 amount	External
transferOwnership	address newOwner	Public
unlock	None	Public
waiveOwnership	None	Public

Contract Ownership

The contract ownership of BNBhare is currently being 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 0x00 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 time 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	0xa0c9d5fca26528ad0253e3f281510828ea84a35e
BNBhare Reserves	804983784442576.50 BNBhare
BNB Reserves	82.08 BNB
Liquidity Value	\$46.50K USDT
Liquidity Ownership	The token does not have liquidity at the moment of the audit

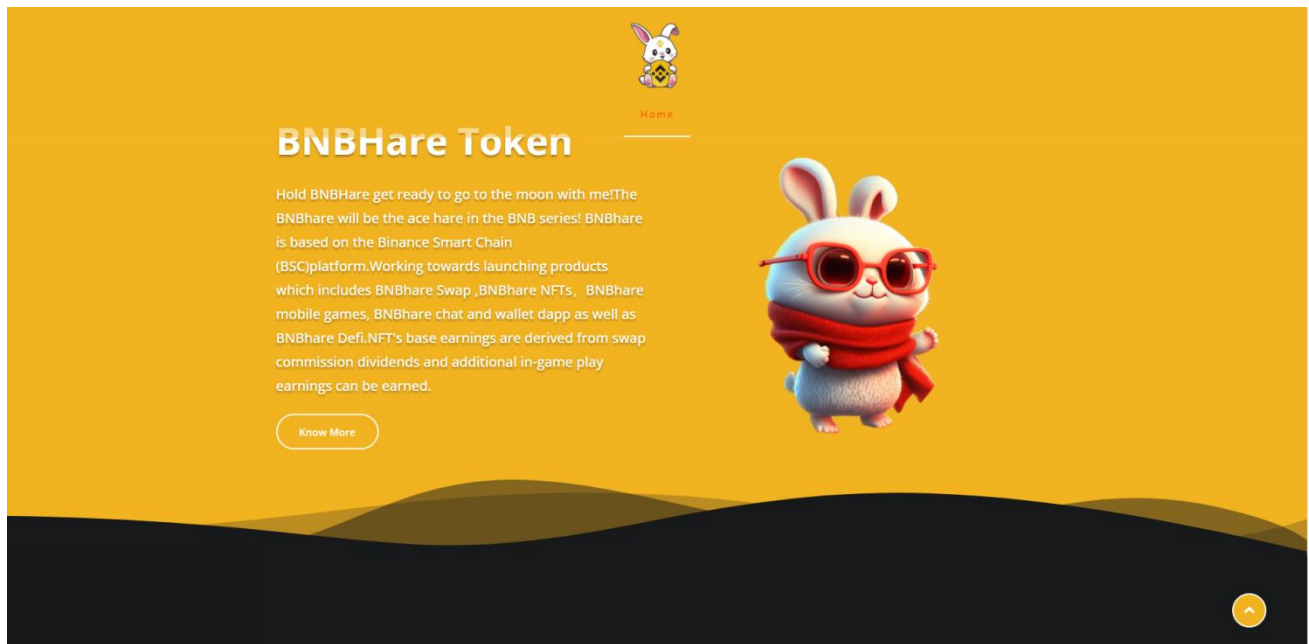
Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	Null: 0x000...dEaD	5,789,572,254,220,6 42,704,923,200.1404 57528	57.8957%
2	PancakeSwap V2: BNBhare 2	811,441,769,319,585 ,212,259,266.570921 874	8.1144%

Social Media Check

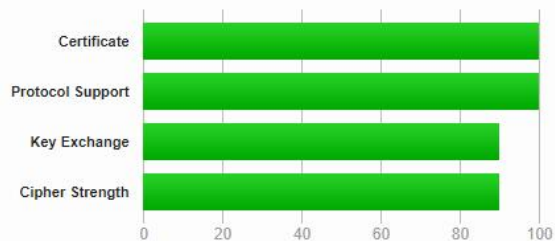
Social Media Type	Link	Result
Website	https://BNBhare.net/	Checked
Twitter	https://twitter.com/bnbhare_bsc/	Checked
Telegram	https://t.me/BNBhare_EN/	Checked

Website Review



Summary

Overall Rating



Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

This site works only in browsers with SNI support.

This server supports TLS 1.3.

Certificate #1: RSA 2048 bits (SHA256withRSA)



Server Key and Certificate #1



Subject	bnbhare.net Fingerprint SHA256: fdf40f375a721af58bfaad50ce61dac83f73eef2d9fdf2698febdbb35136ea0f Pin SHA256: soJetyplorJN14XxtVWlkuIE+EtFpPvGVqXQRNxoY=
Common names	bnbhare.net
Alternative names	bnbhare.net www.bnbhare.net
Serial Number	0434c4b7ff1d4a4813e61315eac79979fe1
Valid from	Fri, 03 Mar 2023 13:26:01 UTC
Valid until	Thu, 01 Jun 2023 13:26:00 UTC (expires in 2 months and 26 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	R3 AIA: http://r3.i.lencr.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Revocation information	OCSP OCSP: http://r3.o.lencr.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

- Mobile Friendly
- Contains no code errors
- SSL is secured
- No spelling errors

Audit Conclusion

- The owner cannot pause trading.
- The owner cannot mint new tokens.
- The owner cannot add blacklist users.
- 25.4% of the PancakeSwap liquidity is unlocked.
- **The owner can get ownership back from fake renouncement. [High-Risk]**
- **The owner can set the max tx amount without limit. [High-Risk]**
- **The owner can change the buy/sell fee up to 100%. [High-Risk]**

(All functions cannot be used if the ownership is renounced)