CRACKEN

Smart Contract Security Audit

Project: Carbon Chain

Sep 08, 2022

Contract Address

0xa9BC4D368e356e209B368274e3477930e7142C12

Table of Contents

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 Carbon Chain 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	Carbon Chain
Token Symbol	CBC
Token Decimal	9
Total Supply	1,000,000,000
Platform	BSC
Buy Tax Fee	0%
Sell Tax Fee	0%
Contract Creation Date	Sep 07, 2022
Liquidity Status	Not available when Audit
Liquidity Lockup Time	Not available
Compiler Version	v0.8.16+commit.07a7930e
Optimization	Yes with 5000 runs
Contract Address	0xa9BC4D368e356e209B368274e3477930e7142C12
Deployer Address	0xb97234b783a28e52567eee47164d29a292e42ebf
Owner Address	0xb97234b783a28e52567eee47164d29a292e42ebf

Source Code

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

https://bscscan.com/address/0xa9BC4D368e356e209B368274e3477930e7142C12

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	0
🟠 Medium-Risk	0
🟡 Low-Risk	1
🔵 Informational	1
Total	2

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

Set outsider is able to create/change the smart contract

Description:

The smart contract can set an operator to allow someone other than the deployer to create/change things on it.

```
function setOperator(address newOperator) external {
        require(msg.sender == originalDeployer, "Can only be called by original
deployer.");
        address oldOperator = operator;
        if (oldOperator != address(0)) {
            _liquidityHolders[oldOperator] = false;
        }
        operator = newOperator;
        _liquidityHolders[newOperator] = true;
    }
```

🔵 **Informational: Implementation of certain corrective actions or accepting the risk.**

Set aniBot function is enabled

Description:

The aniBot function is enabled, bots will be killed by blocks.

```
function setProtectionSettings(bool _antiSnipe, bool _antiBlock) external onlyOwner {

            antiSnipe.setProtections(_antiSnipe, _antiBlock);

    }
```

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
enableTrading	None	Public
excludePresaleAddresses	address router, address presale	External
multiSendTokens	address[] memory accounts, uint256[] memory amounts	External
removeSniper	address account	External
renounceOriginalDeployer	None	External
renounceOwnership	None	Public
setExcludedFromLimits	address account, bool enabled	External
setExcludedFromProtection	setExcludedFromProtection	External
setInitializer	address initializer	External
setOperator	address newOperator	External
setProtectionSettings	bool _antiSnipe, bool _antiBlock	External
sweepContingency	None	External
transfer	address recipient, uint256 amount	External
transferFrom	address sender,address recipient,uint256 amount	Public
transferOwner	address newOwner	External

Contract Ownership

The contract ownership of Carbon Chain is not 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 0xb97234b783a28e52567eee47164d29a292e42ebf 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	0x9e650ab285ece62a7d0ea4cf0bd6fc278471e5a0
CBC Reserves	0.00 CBC
BNB Reserves	0.00 BNB
Liquidity Value	$0.00 USDT
Liquidity Ownership	The token does not have liquidity at the moment of the audit

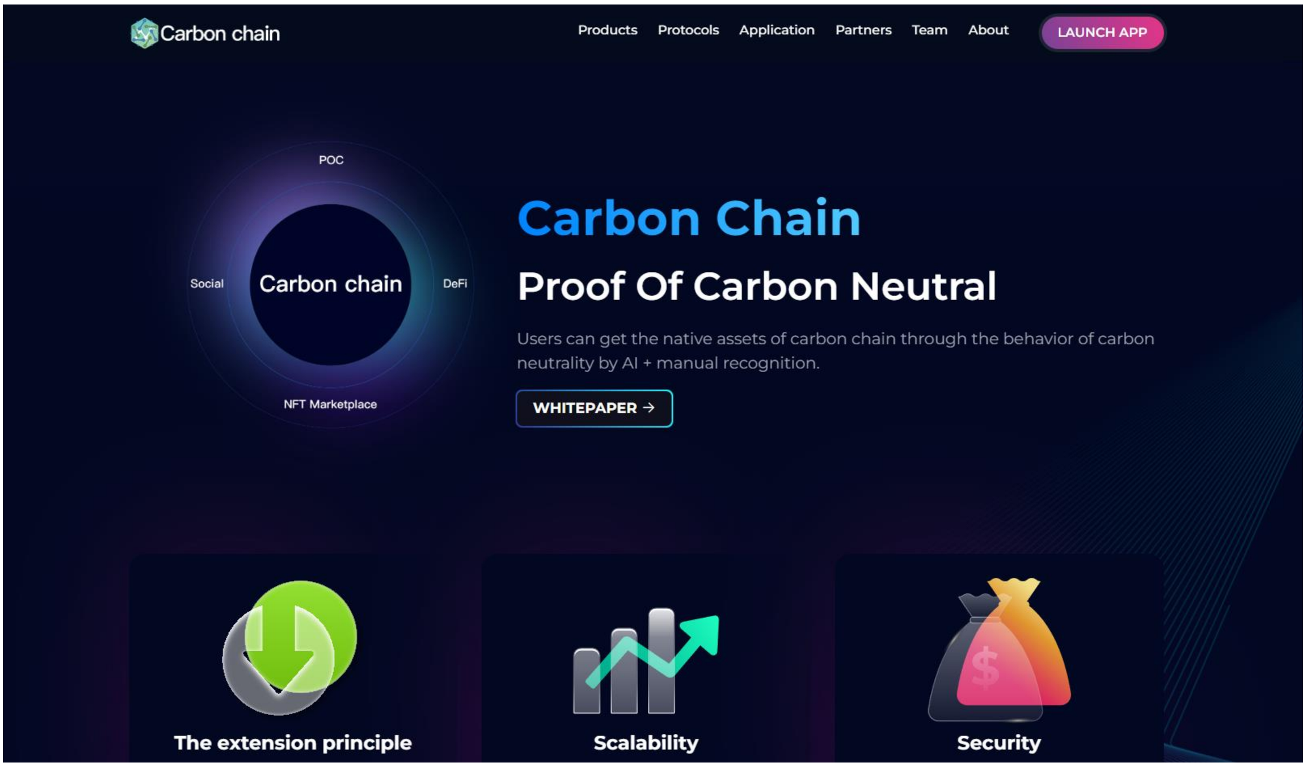
Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xb97234b783a28e52567eee47164d29a292e42ebf	1,000,000,000	100.0000%

Social Media Check

Social Media Type	Link	Result
Website	https://carbonchains.xyz/	Checked
Twitter	https://twitter.com/CarbonChains/	Checked
Telegram	https://t.me/CarbonChainEN/	Checked

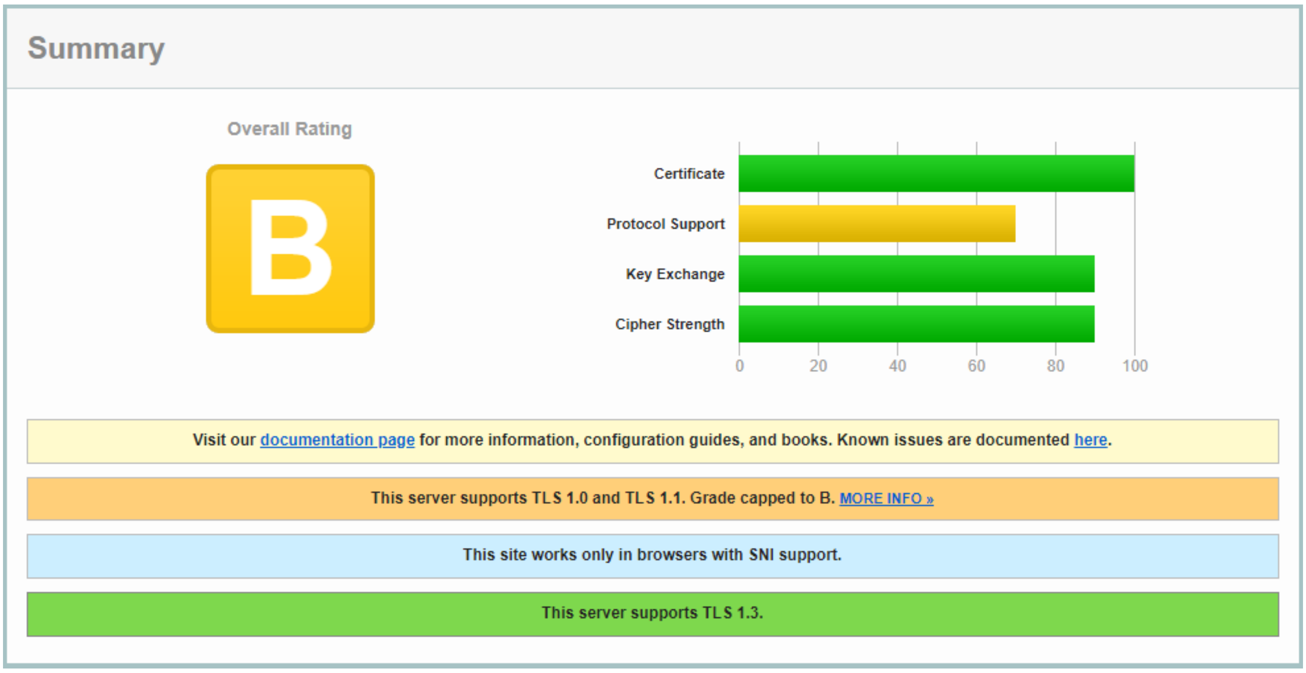
Website Review



Carbon Chain

Proof Of Carbon Neutral

Users can get the native assets of carbon chain through the behavior of carbon neutrality by AI + manual recognition.



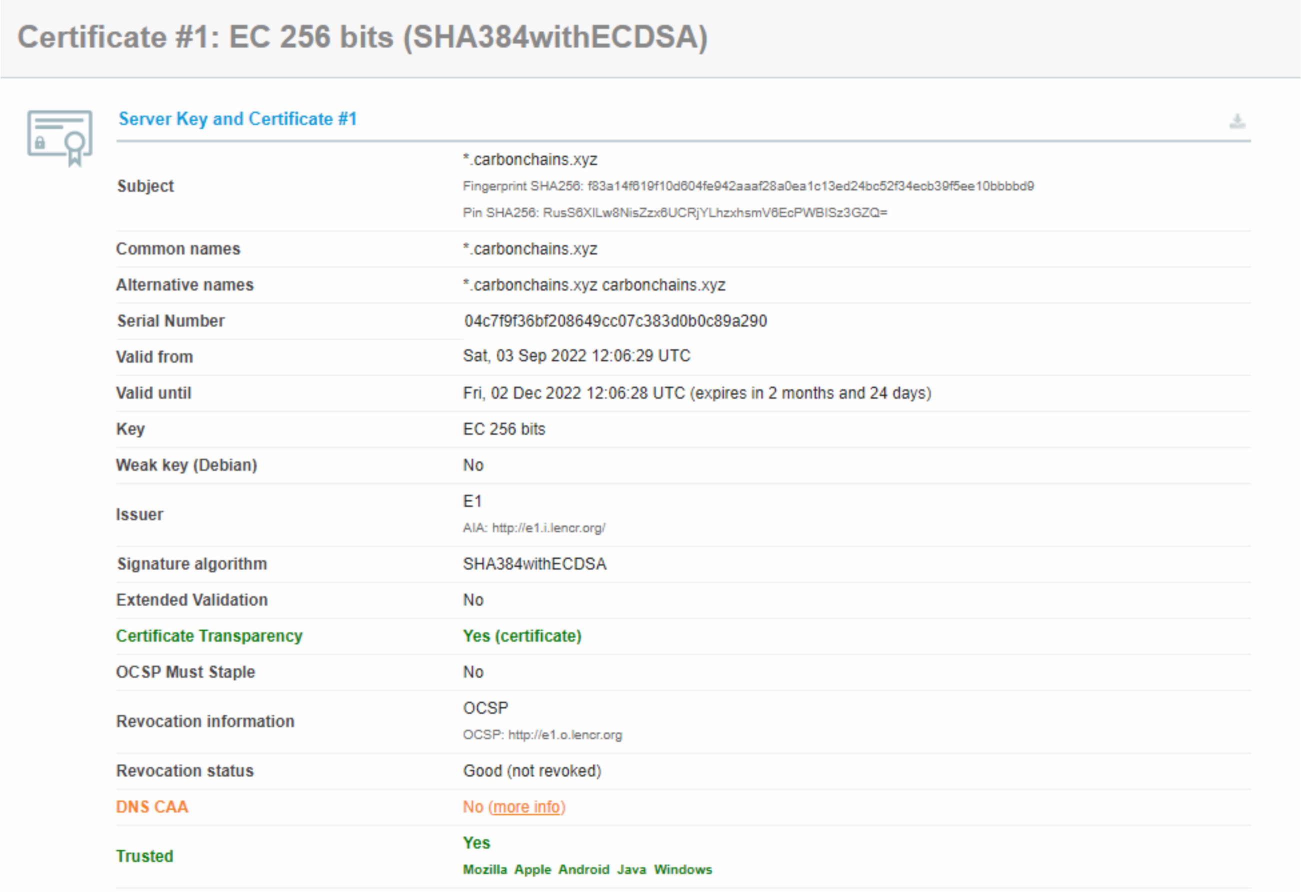
Summary

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. MORE INFO »

This site works only in browsers with SNI support.

This server supports TLS 1.3.



Certificate #1: EC 256 bits (SHA384withECDSA)

Server Key and Certificate #1

Subject	*.carbonchains.xyz Fingerprint SHA256: f83a14f619f10d604fe942aaaf28a0ea1c13ed24bc52f34ecb39f5ee10bbbbd9 Pin SHA256: RusS6XILw8NisZzx6UCRjYLhzxhsmV6EcPWBISz3GZQ=
Common names	*.carbonchains.xyz
Alternative names	*.carbonchains.xyz carbonchains.xyz
Serial Number	04c7f9f36bf208649cc07c383d0b0c89a290
Valid from	Sat, 03 Sep 2022 12:06:29 UTC
Valid until	Fri, 02 Dec 2022 12:06:28 UTC (expires in 2 months and 24 days)
Key	EC 256 bits
Weak key (Debian)	No
Issuer	E1 AIA: http://e1.i.lencr.org/
Signature algorithm	SHA384withECDSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://e1.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.
- The owner cannot set the max transaction amount.
- The owner cannot change the buy/sell fees up.
- The owner can antiBots by killing blocks
- The smart contract can set an operator to allow someone other than the deployer to create/change things on it.

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

AUDIT IS PASSED