Coinsult

Request your audit at coinsult.net

Advanced Manual
Smart Contract Audit

December 11, 2022

CoinsultAudits

info@coinsult.net

coinsult.net

Audit requested by

Bitcoin V2

0x49816fEBcee3d5Bc370B60AB300dc88ECB5ac934

Table of Contents

Audit Summary

Project Name	**Bitcoin V2**
Website	https://bitcoin-v2.net/
Blockchain	Binance Smart Chain
Smart Contract Language	Solidity
Contract Address	0x49816fEBcee3d5Bc370B60AB300dc88ECB5ac934
Audit Method	Static Analysis, Manual Review
Date of Audit	11 December 2022

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identi ed.

Audit Scope

Source Code

Coinsult was comissioned by Bitcoin V2 to perform an audit based on the following code:

https://bscscan.com/token/0x49816febcee3d5bc370b60ab300dc88ecb5ac934

Note that we only audited the code available to us on this URL at the time of the audit. If the URL is not from any block explorer (main net), it may be subject to change. Always check the contract address on this audit report and compare it to the token you are doing research for.

SAFU by Trynos

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xa1dbf6af9e4f430b212dbf07d846935583f2048e	21,000,000	100.0000%

Audit Method

Coinsult's manual smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. This process is conducted to discover errors, issues and security vulnerabilities in the code in order to suggest improvements and ways to x them.

Automated Vulnerability Check

Coinsult uses software that checks for common vulnerability issues within smart contracts. We use automated tools that scan the contract for security vulnerabilities such as integer-overflow, integer-underflow, out-of-gas-situations, unchecked transfers, etc.

Manual Code Review

Coinsult's manual code review involves a human looking at source code, line by line, to nd vulnerabilities. Manual code review helps to clarify the context of coding decisions. Automated tools are faster but they cannot take the developer's intentions and general business logic into consideration.

Used Tools

- Slither: Solidity static analysis framework
- Remix: IDE Developer Tool
- CWE: Common Weakness Enumeration
  SWC: Smart Contract Weakness Classi cation and Test Cases
  DEX: Testnet Blockchains

Risk Classi cation

Coinsult uses certain vulnerability levels, these indicate how bad a certain issue is. The higher the risk, the more strictly it is recommended to correct the error before using the contract.

Vulnerability Level	Description
Informational	Does not compromise the functionality of the contract in any way
Low-Risk	Won't cause any problems, but can be adjusted for improvement
Medium-Risk	Will likely cause problems and it is recommended to adjust
High-Risk	Will de nitely cause problems, this needs to be adjusted

Coinsult has four statuses that are used for each risk level. Below we explain them briefly.

Risk Status	Description
Total	Total amount of issues within this category
Pending	Risks that have yet to be addressed by the team
Acknowledged	The team is aware of the risks but does not resolve them
Resolved	The team has resolved and remedied the risk

Disclaimer

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identi ed.

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any nancial losses. Nothing in this contract audit is nancial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

Global Overview

Manual Code Review

In this audit report we will highlight the following issues:

Vulnerability Level	Total	Pending	Acknowledged	Resolved
Informational	0	0	0	0
Low-Risk	3	2	0	0
Medium-Risk	0	0	0	0
High-Risk	0	0	0	0

Centralization Risks

Coinsult checked the following privileges:

Contract Privilege	Description
Owner can mint?	Owner cannot mint new tokens
Owner can blacklist?	Owner cannot blacklist addresses
Owner can set fees > 25%?	Owner cannot set the sell fee to 25% or higher
Owner can exclude from fees?	Owner can exclude from fees
Owner can pause trading?	Owner cannot pause the contract
Owner can set Max TX amount?	Owner cannot set max transaction amount

More owner priviliges are listed later in the report.

Maximum Fee Limit Check

Error Code	Description
CEN-01	Centralization: Operator Fee Manipulation

Coinsult tests if the owner of the smart contract can set the transfer, buy or sell fee to 25% or more. It is bad practice to set the fees to 25% or more, because owners can prevent healthy trading or even stop trading when the fees are set too high.

Type of fee	Description
Transfer fee	Owner cannot set the transfer fee to 25% or higher
Buy fee	Owner cannot set the buy fee to 25% or higher
Sell fee	Owner cannot set the sell fee to 25% or higher

Type of fee	Description
Max transfer fee	8%
Max buy fee	8%
Max sell fee	8%

Function

```
function setTaxes(uint16 buyFee, uint16 sellFee, uint16 transferFee) external onlyOwner {
require(!taxesAreLocked, "Taxes are locked.");
require(buyFee &amp;lt;= maxBuyTaxes
&amp;amp;&amp;amp; sellFee &amp;lt;= maxSellTaxes
&amp;amp;&amp;amp; transferFee &amp;lt;= maxTransferTaxes,
&amp;quot;Cannot exceed maximums.&amp;quot;);
_taxRates.buyFee = buyFee;
_taxRates.sellFee = sellFee;
_taxRates.transferFee = transferFee;
}
```

Ability To Mint Check

Error Code	Description
CEN-05	Centralization: Operator Increase Supply

Coinsult tests if the owner of the smart contract can mint new tokens. If the contract contains a mint function, we refer to the token's total supply as non- xed, allowing the token owner to "mint" more tokens whenever they want.

A mint function in the smart contract allows minting tokens at a later stage. A method to disable minting can also be added to stop the minting process irreversibly.

Minting tokens is done by sending a transaction that creates new tokens inside of the token smart contract. With the help of the smart contract function, an unlimited number of tokens can be created without spending additional energy or money.

Privilege Check	Description
Can owner mint?	Owner cannot mint new tokens

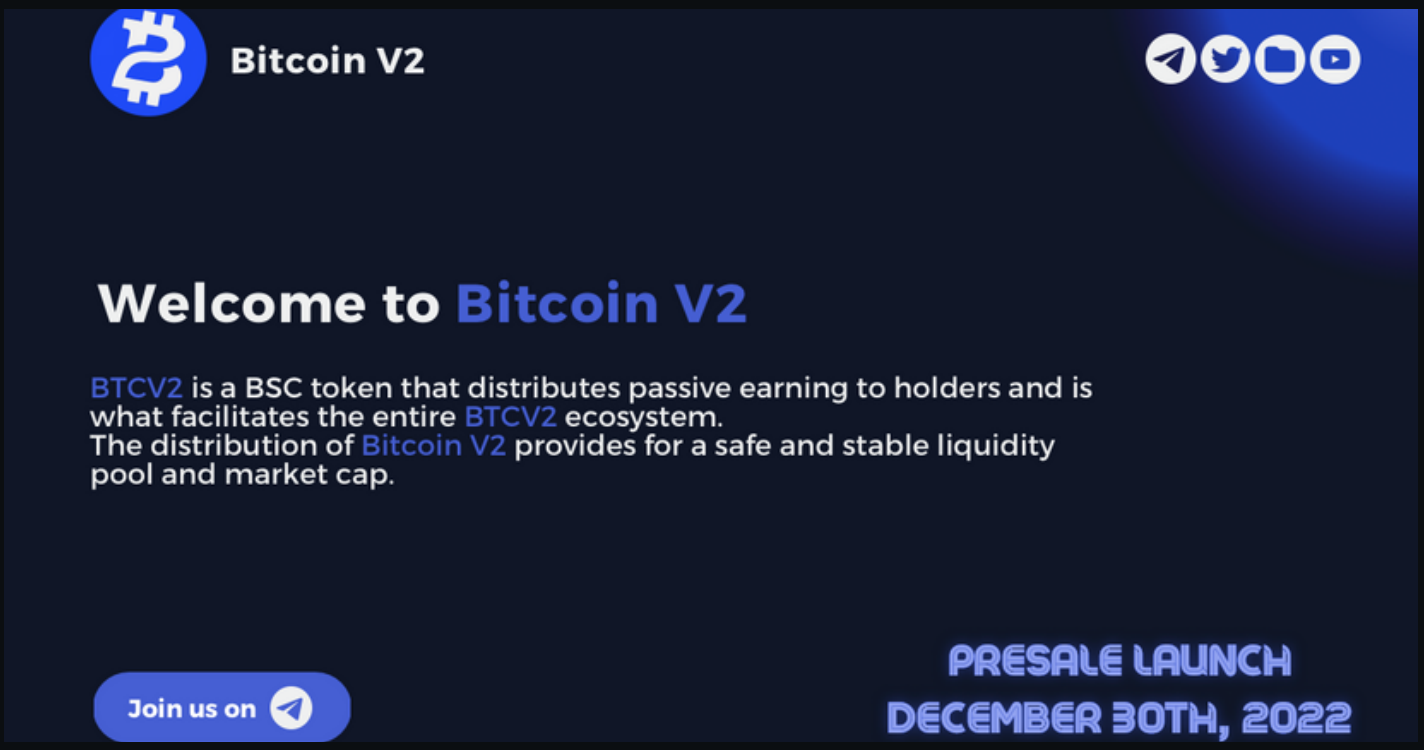
Contract Snapshot

This is how the constructor of the contract looked at the time of auditing the smart contract.

```
contract LitchiChain is IERC20 {
mapping (address =&gt; uint256) private _tOwned;
mapping (address =&gt; bool) lpPairs;
uint256 private timeSinceLastPair = 0;
mapping (address =&gt; mapping (address =&gt; uint256)) private _allowances;
mapping (address =&gt; bool) private _liquidityHolders;
mapping (address =&gt; bool) private _isExcludedFromProtection;
mapping (address =&gt; bool) private _isExcludedFromFees;
mapping (address =&gt; bool) private _isExcludedFromLimits;
mapping (address =&gt; bool) private presaleAddresses;
bool private allowedPresaleExclusion = true;

uint256 constant private startingSupply = 100_000_000;
string constant private _name = "Eternity Protocol";
string constant private _symbol = "$ETRNTY";
uint8 constant private _decimals = 18;
uint256 constant private _tTotal = startingSupply * 10**_decimals;

struct Fees {
uint16 buyFee;
        uint16 sellFee;
          uint16 transferFee;
}

Fees public _taxRates = Fees({
buyFee: 300,
sellFee: 300,
transferFee: 300
});

uint256 constant public maxBuyTaxes = 600;
uint256 constant public maxSellTaxes = 600;
uint256 constant public maxTransferTaxes = 600;
uint256 constant masterTaxDivisor = 10000;

bool  public  taxesAreLocked;
IRouter02 public dexRouter;
```

Website Review

Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.



Recommended to not have images used as text. This is bad for SEO reasons.

Type of check	Description
Mobile friendly?	The website is mobile friendly
Contains jQuery errors?	The website does not contain jQuery errors
Is SSL secured?	The website is SSL secured
Contains spelling errors?	The website does not contain spelling errors

Certi cate of Proof

Bitcoin V2

Audited by Coinsult.net

Date: 11 December 2022

Advanced Manual Smart Contract Audit

Coinsult

coinsult.net

End of report

Smart Contract Audit

CoinsultAudits

info@coinsult.net

coinsult.net