

**Smart Contract**

# **Security Assessment**

**For BarterCoin  
30 Nov 2022**



**Ascendant**

**Ascendant**

@ascendantproj  
www.ascendant.finance



**Ascendant**

# Table of Contents

**3** Disclaimer

---

**4** Executive Summary

---

**5** Overview

---

**6** Findings Summary & Legend

---

**8** Manual Review

- Issue Checking Status
- Audit Findings
- Functional Test Status
- Omitted Results

---

**21** Automated Review

- Unified Model Language

---

**23** Conclusion

---

# DISCLAIMER

Ascendant Finance (“Ascendant”) has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocacy for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Ascendant.

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. Information is provided ‘as is’, and Ascendant is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Ascendant or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team. Ascendant retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Ascendant is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. Ascendant may, at its discretion, claim bug bounties from third-parties while doing so.

# Executive Summary

Severity	Found
● High	0
● Medium	0
● Low	12
● Informational	24
Total	36

We performed an independent technical audit to identify Smart Contracts uncertainties. This shall protect the code from illegitimate authorization attempts or external & internal threats of any type. This also ensures end-to-end proofing of the contract from frauds. The audit was performed semi-manually. We analyzed the Smart Contracts code line-by-line and used an automation tool to report any suspicious code.

The following tools were used:

- Truffle
- Remix IDE
- Slither

# Overview

This report has been prepared for Bartercoin on the Binance network. Ascendant provides a user-centered examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

## Summary





<b>Project Name</b>	Bartercoin
<b>Platform</b>	Binance
<b>Language</b>	Solidity

## Contracts Assessed





Name	Location
BRX.sol	0x55F0DE86f0139F9226ba777C6Fc66BE7EC0f6800
SafeMath.sol	In BRX contract
Context.sol	In BRX contract
Address.sol	In BRX contract
Ownable.sol	In BRX contract
IBEP20.sol	In BRX contract

Name	Location
IUniswapV2Factory.sol	In BRX contract
IUniswapV2Pair.sol	In BRX contract
IUniswapV2Router01.sol	In BRX contract
IUniswapV2Router02.sol	In BRX contract

# Findings Summary

Severity	Found
 High	0
 Medium	0
 Low	12
 Informational	24
Total	36

## Classification of Issues

 High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
 Medium	Bugs or issues that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to 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.
 Informational	Consistency, syntax or style best practices, Generally pose a negligible level of risk, if any.

# Manual Review



Ascendant



# Issues Checking Status

Issue Description	Checking Status
Compiler errors	PASS
Race conditions and Reentrancy. Cross-function race conditions.	PASS
Possible delays in data delivery.	PASS
Oracle calls.	PASS
Front running.	PASS
Timestamp dependence.	PASS
Integer Overflow and Underflow.	PASS
DoS with Revert.	PASS
DoS with block gas limit.	PASS
Methods execution permissions.	PASS
Economy model of the contract.	PASS
The impact of the exchange rate on the logic.	PASS
Private user data leaks.	PASS
Malicious Event log.	PASS
Scoping and Declarations.	PASS
Uninitialized storage pointers.	PASS

Arithmetic accuracy.	PASS
Design Logic.	PASS
Cross-function race conditions.	PASS
Safe Open Zeppelin contracts implementation and usage.	PASS
Fallback function security.	PASS

# Audit Findings

Severity	Low
Contract	BRX.sol
Description	Important function calls lack event.
Code Snippet	BRX._setTaxFee(uint256) (Bartercoin.sol#1433-1436) should emit an event for: - _taxFee = taxFee (Bartercoin.sol#1435) BRX._setCharityFee(uint256) (Bartercoin.sol#1438-1444) should emit an event for: - _charityFee = charityFee (Bartercoin.sol#1443) BRX._setMaxTxAmount(uint256) (Bartercoin.sol#1453-1459) should emit an event for: - _maxTxAmount = maxTxAmount (Bartercoin.sol#1458)
Recommendation	The above function calls should emit an event for transparency purposes.
Status	

# Audit Findings

Severity	Lowx3
Contract	BRX.sol
Description	Functions lack zero-check.
Code Snippet	<pre>Ownable.constructor().msgSender (Bartercoin.sol#446) lacks a zero- check on :     - _owner = msgSender (Bartercoin.sol#447) BRX.constructor(address).charityW alletAddress (Bartercoin.sol#888) lacks a zero-check on :     - _charityWalletAddress = charityWalletAddress (Bartercoin.sol#889) BRX._setCharityWallet(address).ch arityWalletAddress (Bartercoin.sol#1446) lacks a zero- check on :     - _charityWalletAddress = charityWalletAddress (Bartercoin.sol#1450)</pre>
Recommendation	Require statements should be added to the above functions to prevent being set to the zero address.

# Audit Findings

Severity	Informational
Contract	BRX.sol
Description	Function should be declared external.
Code Snippet	<code>_getETHBalance()</code> should be declared external: - <code>BRX._getETHBalance()</code> (Bartercoin.sol#1429-1431)
Recommendation	Functions meant to be called from outside the contract should be declared external.
Status	

# Functional Test Status

Function Name	Type/Return Type	Score
IBEP20		
allowance	read/external	PASS
approve	write/external	PASS
balanceOf	read/external	PASS
totalSupply	read/external	PASS
transfer	write/external	PASS
transferFrom	write/external	PASS
SafeMath		
add	internal	PASS
div	internal	PASS
mod	internal	PASS
mul	internal	PASS
sub	internal	PASS
Context		
_msgData	internal	PASS
_msgSender	internal	PASS
Ownable		
owner	read/public	PASS
renounceOwnership	write/public	PASS

Function Name	Type/Return Type	Score
transferOwnership	write/public	PASS
IUniswapV2Factory		
allPairs	read/public	PASS
allPairsLength	read/external	PASS
createPair	write/external	PASS
feeTo	external	PASS
feeToSetter	external	PASS
getPair	read/external	PASS
setFeeTo	external	PASS
setFeeToSetter	write/external	PASS
IUniswapV2Pair		
DOMAIN_SEPARATOR	external	PASS
MINIMUM_LIQUIDITY	external	PASS
PERMIT_TYPEHASH	external	PASS
burn	write/external	PASS
factory	external	PASS
getReserves	read/external	PASS
initialize	write/external	PASS

Function Name	Type/Return Type	Score
swap	write/external	PASS
IUniswapV2Router02		
WETH	read/external	PASS
addLiquidity	write/external	PASS
getAmountIn	read/external	PASS
getAmountOut	read/external	PASS
getAmountsIn	read/external	PASS
getAmountsOut	read/external	PASS
quote	external	PASS
removeLiquidity	write/external	PASS
removeLiquidityETH	write/external	PASS
removeLiquidityETHWithPermit	write/external	PASS
removeLiquidityWithPermit	write/external	PASS
swapETHForExactTokens	write/external	PASS
swapExactETHForTokens	write/external	PASS
swapExactTokensForETH	write/external	PASS
swapExactTokensForTokens	write/external	PASS



swapTokensForExactEth	write/external	PASS
swapTokensForExactTokens	write/external	PASS
BRX		
_approve	internal	PASS
_getETHBalance	public	PASS
_getCurrentSupply	private	PASS
_getRValues	private	PASS
_getRate	private	PASS
_getTValues	private	PASS
_getValues	private	PASS
_reflectFee	private	PASS
_tokenTransfer	private	PASS
_transfer	private	PASS
_transferBothExcluded	private	PASS
_transferFromExcluded	private	PASS
_transferStandard	private	PASS
_transferToExcluded	private	PASS
_setCharityFee	write/external	PASS
_setCharityWallet	write/external	PASS
_setMaxTxAmount	write/external	PASS
_setTaxFee	write/external	PASS
decimals	read/public	PASS

decreaseAllowance	write/public	PASS
allowance	read/public	PASS
deliver	public	PASS
sendETHToCharity	private	PASS
excludeAccount	write/public	PASS
swapTokensForEth	private	PASS
includeAccount	write/external	PASS
increaseAllowance	write/public	PASS
isExcluded	read/public	PASS
isExcludedFromFee	read/public	PASS
name	read/public	PASS
restoreAllFee	private	PASS
receive	external	PASS
setExcludeFromFee	write/external	PASS
setSwapEnabled	write/external	PASS
reflectionFromToken	read/public	PASS
removeAllFee	private	PASS
restoreAllFee	private	PASS
setCharityFee	write/external	PASS
setCharityWallet	write/external	PASS
setMaxTxAmount	write/external	PASS
_takeCharity	private	PASS

manualSend	write/external	PASS
manualSwap	write/external	PASS
symbol	read/public	PASS
tokenFromReflection	read/public	PASS
totalFees	read/public	PASS
totalSupply	read/public	PASS
transfer	write/public	PASS
transferFrom	write/public	PASS

# Omitted Results

**Note: Any issues that have been omitted from this report have been deemed by the reviewing team as irrelevant, inapplicable, and/or negligible to the proper functioning of this contract. Thus, any omitted issues can be safely ignored.**

# Automated Review



Ascendant