CRACKEN

Smart Contract Security Audit

Project: Astro Cash

Jul 05, 2022

Contract Address

0x1b24ebbEc03298576337B1805c733cD225C8a6BC

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 Astro Cash 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	Astro Cash
Token Symbol	ASTRO
Token Decimal	18
Total Supply	50,000,000
Platform	BSC
Buy Tax Fee	6%
Sell Tax Fee	6%
Contract Creation Date	Jun 19, 2022
Liquidity Status	Not Available
Liquidity Lockup Time	Not Available
Compiler Version	v0.8.15+commit.e14f2714
Optimization	No with 200 runs
Contract Address	0x1b24ebbEc03298576337B1805c733cD225C8a6BC
Deployer Address	0x4214B0a3bfC90Da0ED63b6330C22d5187953f503
6Owner Address	0xed33586034f87435592520482c65e5299c2fc104

Source Code

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

https://bscscan.com/address/0x1b24ebbEc03298576337B1805c733cD225C8a6BC

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 Buy / Sell Fees

Description:

Burn Fee, Liquidity Fee, Project Fee, Marketing Fee, and Development Fee can be changed by less than 10% separately.

```
* Set burn fee: Base 10000, ex.: 1.5% = 150
  **/
function setBurnFee(
    uint256 buy,
    uint256 sell,
    uint256 p2p
) external onlyOwner {
    require(
        buy <= maxIndividualFee &&
            sell <= maxIndividualFee &&
            p2p <= maxIndividualFee,
        "You must respect the maximum allowed fee"
    );
    burnFee[0] = buy;
    burnFee[1] = sell;
    burnFee[2] = p2p;
}

/**
  * Set liquidity fee: Base 10000, ex.: 1.5% = 150
  **/
function setLiquidityFee(
    uint256 buy,
    uint256 sell,
    uint256 p2p
) external onlyOwner {
    require(
        buy <= maxIndividualFee &&
            sell <= maxIndividualFee &&
            p2p <= maxIndividualFee,
        "You must respect the maximum allowed fee"
    );
    liquidityFee[0] = buy;
    liquidityFee[1] = sell;
```

```
        liquidityFee[2] = p2p;
    }

    /**
     * Set Project fee: Base 10000, ex.: 1.5% = 150
     **/
    function setProjectFee(
        uint256 buy,
        uint256 sell,
        uint256 p2p
    ) external onlyOwner {
        require(
            buy <= maxIndividualFee &&
                sell <= maxIndividualFee &&
                p2p <= maxIndividualFee,
            "You must respect the maximum allowed fee"
        );
        projectFee[0] = buy;
        projectFee[1] = sell;
        projectFee[2] = p2p;
    }

    /**
     *   Set Marketing fee: Base 10000, ex.: 1.5% = 150
     **/
    function setMarketingFee(
        uint256 buy,
        uint256 sell,
        uint256 p2p
    ) external onlyOwner {
        require(
            buy <= maxIndividualFee &&
                sell <= maxIndividualFee &&
                p2p <= maxIndividualFee,
            "You must respect the maximum allowed fee"
        );
        marketingFee[0] = buy;
        marketingFee[1] = sell;
        marketingFee[2] = p2p;
    }

    /**
     * Set Dev fee: Base 10000, ex.: 1.5% = 150
     **/
```

```
function setDevelopmentFee(
    uint256 buy,
    uint256 sell,
    uint256 p2p
) external onlyOwner {
    require(
        buy <= maxIndividualFee &&
            sell <= maxIndividualFee &&
            p2p <= maxIndividualFee,
        "You must respect the maximum allowed fee"
    );
    developmentFee[0] = buy;
    developmentFee[1] = sell;
    developmentFee[2] = p2p;
}
```

Recommendation: fees are set to a reasonable amount

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

Set max buy / sell amount

Description:

- Maximum sell amount can be set to one-thousandth of the total supply
- Minimum buy amount can be set to one-thousandth of the total supply

```
function setSaleTxAmount(uint256 amount) external onlyOwner {
    require(
        amount <= totalSupply() &&
            amount >= totalSupply().mul(minIndividualLimitTx).div(10000),
        "Limit needs to be between the individual minimum and the total supply"
    );
    maxSaleAmount = amount;
}

/**
  * Set max tx buy amount
  **/
function setBuyTxAmount(uint256 amount) external onlyOwner {
    require(
        amount <= totalSupply() &&
            amount >= totalSupply().mul(minIndividualLimitTx).div(10000),
        "Limit needs to be between the individual minimum and the total supply"
    );
    maxBuyAmount = amount;
}
```

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
decreaseAllowance	address spender, uint256 subtractedValue	Public
excludeFromLimitAmount	address account, bool excluded	Public
excludeMultipleAccountsFromFees	address[] calldata accounts,bool	External
increaseAllowance	address account, bool excluded	Public
renounceOwnership	Uint256 _amount	Private
setBurnFee	uint256 buy,uint256 sell,uint256 p2p	External
setBuyTxAmount	Unit256 amount	External
setDevelopmentFee	uint256 buy,uint256 sell,uint256 p2p	External
setLiquidityFee	uint256 buy,uint256 sell,uint256 p2p	External
setLpDestination	address newLpOwner	External
setMarketingFee	uint256 buy,uint256 sell,uint256 p2p	External
setMarketingWallet	address payable wallet	External
setMaxTxAmount	uint256 amount	External
setProjectFee	uint256 buy,uint256 sell,uint256 p2p	External
setProjectWallet	address payable wallet	External
setSaleTxAmount	uint256 amountuint256 amount	External
transferOwnership	address newOwner	Public
setSwapTokensAmount	uint256 amount	Public

Contract Ownership

The contract ownership of Astro Cash 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 0xed33586034f87435592520482c65e5299c2fc104 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	0xed33586034f87435592520482c65e5299c2fc104
ASTRO Reserves	0.00 ASTRO
BNB Reserves	0.00 BNB
Liquidity Value	$0 USD
Liquidity Ownership	The token does not have liquidity at the moment of the audit

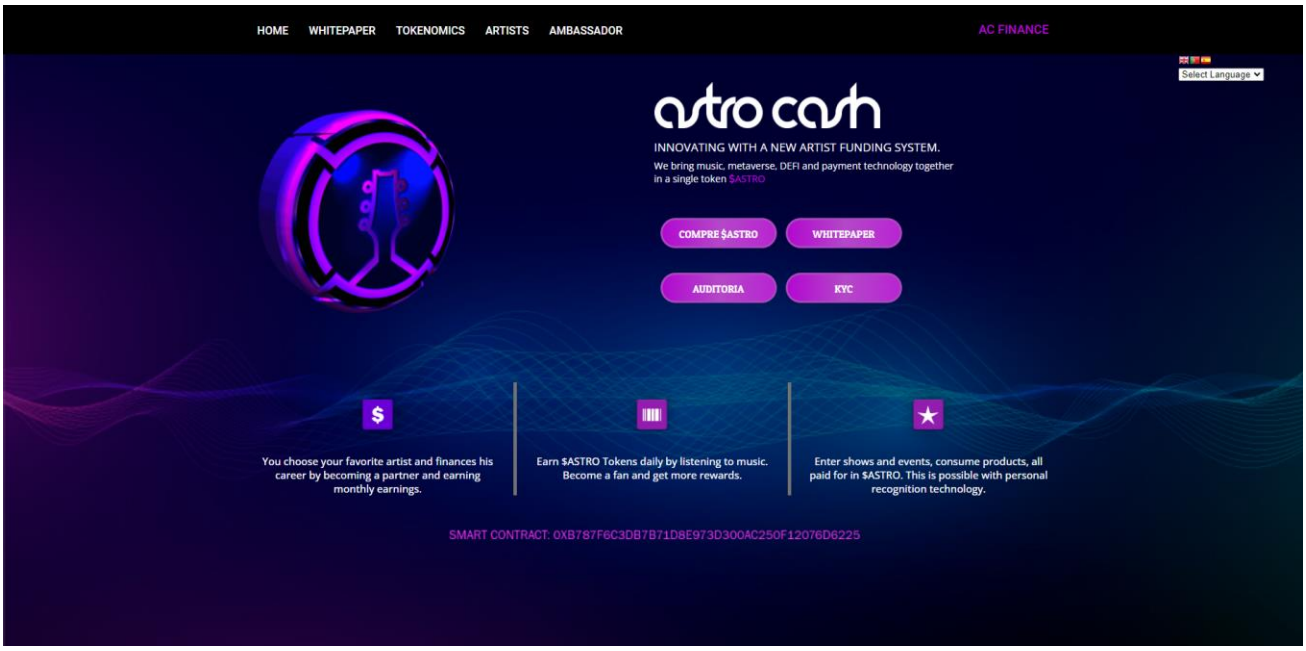
Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xed33586034f87435592520482c65e5299c2fc104	50,000,000	100.0000%

Social Media Check

Social Media Type	Link	Result
Website	https://www.astrocash.me/	Checked
Twitter	https://twitter.com/AstroCashCrypto/	Checked
Telegram	https://t.me/astrocasglobal	Checked
Facebook	https://www.facebook.com/astrocashcrypto	Checked
Instagram	https://www.instagram.com/astrocashcrypto	Checked

Website Review



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

Audit Conclusion

- The owner cannot pause trading
- The owner cannot mint new tokens
- The owner cannot blacklist users
- The owner can change the max tx amount to one-thousandth of the total supply

⚠ The owner can change buy/sell fees separately by less than 10%, which includes burn fee, liquidity fee, project fee, marketing fee, and development fee.

(The fees cannot be changed if the owner renounced the ownership)

AUDIT PASSED