RugFreeCoins Audit

Asta Inu Token

Smart Contract Security Audit

April 13, 2022

Contents

Audit details

Audited project
Asta Inu

Contract Address
0x25007F5F3083Fae36EdD5a7a460B5b324F2BA3A6

Client contact
Asta Inu Team

Blockchain
Binance smart chain

Project website
https://astainu.io/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

Rugfreecoins was commissioned by the Asta Inu Team to perform an audit of the smart contract.

https://bscscan.com/token/0x25007F5F3083Fae36EdD5a7a460B5b324F2BA3A6

The focus of this audit is to verify that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long-term sustainability, and as a guide to improving the security posture of the smart contract by remediating the issues that were identified.

About the project

Asta Inu Token is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources and heading towards building an even greater Community, with safe tokenomics. Each transaction, purchase, and sale incur a 12% fee.

Features

- The **Asta Inu reflections** will be distributed in tokens among every holder proportional to how many tokens each individual holds in values of **5% when buying and selling.**

- The additional component included under the sustainability section is a **liquidity fee of 5% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- **Asta Inu** has a burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Asta Inu more and more valuable. 2% fee goes to a wallet from buying and selling, which allocates for the manual burn.

Roadmap

Phase 1 - Launch

- 1,000 Holders
- 2,000 Twitter Followers
- $10,000 Airdrop Event
- Website Launch
- CoinGecko Listing
- CoinMarketCap Listing
- 2,000 Telegram Members
- Whitepaper v1
- Listing AstaInu

Phase 2 - Growth

- 10,000 Telegram Members
- 3,000 Holders
- 5,000 Twitter Followers
- Redesign Website
- First Charity Give-Away $20,000

Phase 3 - Expansion

- $20,000 Airdrop Event
- ASTA Swap on Asta Inu website (Decentralized Exchange)
- CEX Listings (Hotbit, Bilaxy, CoinTiger)
- 30,000 Telegram Members
- 10,000 Holders
- 15,000 Twitter Followers
- Second Charity Give-Away $30,000

Phase 4 - Jump to the moon

- $30,000 Airdrop Event
- 20,000 Holders
- 50,000 Twitter Followers
- 100,000 Telegram Members
- CEX Listings (Gate, MEXC, Kucoin)
- Listing AstaInu NFT on Opensea
- Release NFT Marketplace, NFT collection, and game for Asta Inu
- Third Charity Give-Away $40,000
- Launch AstaInu Wallet • Multichain-Bridge Feature

Tokenomics

12% fee when buying and buying

- 5% of trade goes to holders’ pockets in tokens.
- 5% of trade goes to the liquidity pool.
- 2% of trade goes to a wallet for manual burn.

Target market and the concept

Target market

- Anyone who’s interested in the Crypto space with long-term investment plans.
- Anyone who’s ready to earn a passive income by holding tokens.
- Anyone who’s interested in trading tokens.
- Anyone who’s interested in taking part in the future plans of the Asta Inu.
- Anyone who’s interested in making financial transactions with any other party using Asta Inu as the currency.

Core concept

The Asta Inu reward system

5% of each transaction when buying and selling is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The liquidity fee of 5% when buying and selling, is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Sonic Bytes has a manual burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making tokens sent to the burn wallet manually that making Asta Inu price more and more valuable.

Potential to grow with score points

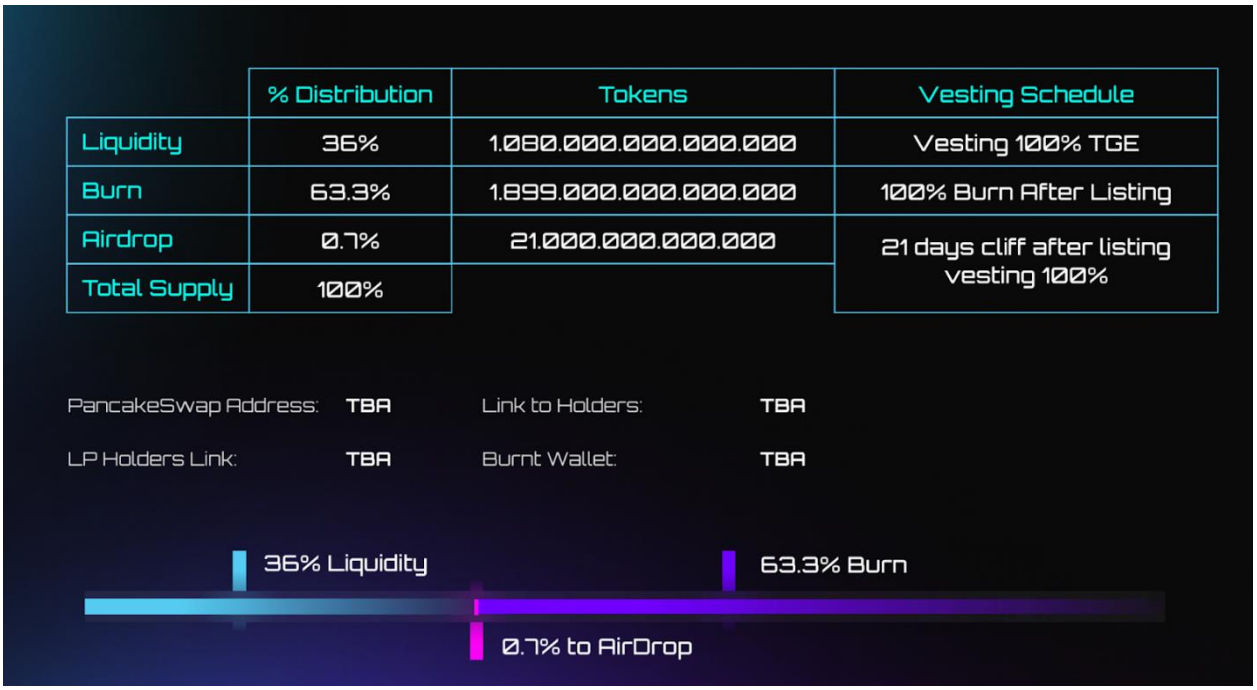
1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	9/10
4	Service quality	8/10
5	System quality	8/10
6	Impact on the community	9/10
7	Impact on the business	7/10
8	Preparing for the future	8/10
Total Points		**8.125/10**

Contract details

Token contract details for 13th April 2022

Contract name	Asta Inu
Contract address	0x25007F5F3083Fae36EdD5a7a460B5b324F2BA3A6
Token supply	3,000,000,000,000,000
Token ticker	ASTA
Decimals	18
Token holders	56
Transaction count	153
Contract deployer address	0xc6bf94065EC808134AbE13686eA932E649e60C33
Advertisement wallet	0xb5352e6c79d896abfd1cd39f508e266dd62f4478
Contract's current owner address	0xc6bf94065ec808134abe13686ea932e649e60c33

Token Distribution



Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	High severity centralization risk
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass

9	**Pseudo random number generator (PRNG)**		**pass**
10	**DoS (Denial of Service)**		**pass**
11	**Token vesting implementation**		**pass**
12	**Fake deposit**		**pass**
13	**Event security**		**pass**

Contract description table

The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

Contract	**Type**	**Bases**		
L	Function Name	Visibility	Mutability	Modifiers
IERC20	**Interface**			
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		

└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	

└	_functionCallWithValue	Private 🔐	🛑	
Ownable	**Implementation**	**Context**		
└		Internal 🔒	🛑	
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	geUnlockTime	Public ❗		NO❗
└	lock	Public ❗	🛑	onlyOwner
└	unlock	Public ❗	🛑	NO❗
IUniswapV2 Factory	**Interface**			
└	createPair	External ❗	🛑	NO❗
IUniswapV2 Router01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
AstaInu	**Implementation**	**Context, IERC20, Ownable**		
└		Public ❗	🛑	NO❗

└	name	Public ❗		NO❗
└	changeAdvestisementWallets	Public ❗	🛑	onlyOwner
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	isExcludedFromReward	Public ❗		NO❗
└	totalFees	Public ❗		NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFee	Public ❗	🛑	onlyOwner

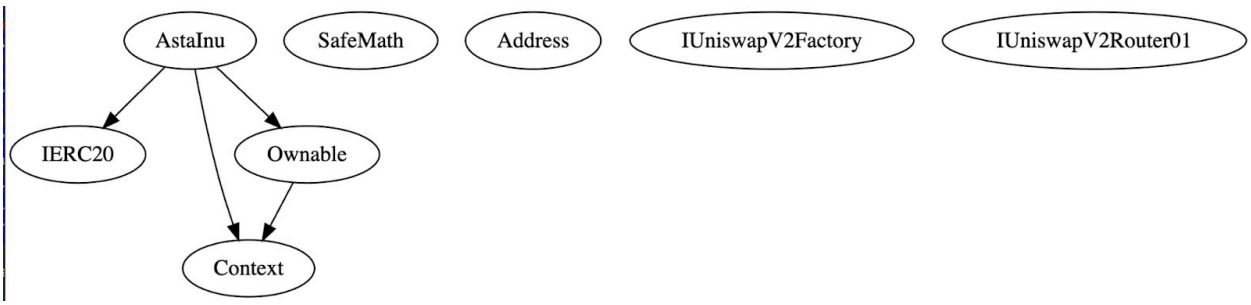
└	manageAmmPairs	Public ❗	🛑	onlyOwner
└	includeInFee	Public ❗	🛑	onlyOwner
└	setTaxFeePercent	External ❗	🛑	onlyOwner
└	setBurnFee	External ❗	🛑	onlyOwner
└	setAdvestisementFeePercent	External ❗	🛑	onlyOwner
└	setMaxTxPercent	External ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeAdvertisement	Private 🔐	🛑	
└	_takeBurn	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateAdvestisementFee	Private 🔐		

└	calculateBurnFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔐	🛑	
└	turnOffAntibotMode	Public ❗	🛑	onlyOwner
└	setAirdropContract	Public ❗	🛑	onlyOwner
└	setAntibotModeWhitelist	Public ❗	🛑	onlyOwner
└	antibotModeCheck	Private 🔐		
└	_transfer	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

No High severity issues found.

- **Medium severity issues**

No medium severity issues found

- **Low severity issues**

No Low severity issues found

- **Centralization risk**

No Centralization issues found

Owner privileges

- The owner can change advertisement wallet

```
ftrace | funcSig
function changeAdvestisementWallets(address wallet⬆) public onlyOwner {
    advertisementWallet = wallet⬆;
}
```

- The owner can include/exclude wallets from rewards

```
ftrace | funcSig
function excludeFromReward(address account⬆) public onlyOwner {
    require(!_isExcluded[account⬆], "Account is already excluded");
    if (_rOwned[account⬆] > 0) {
        _tOwned[account⬆] = tokenFromReflection(_rOwned[account⬆]);
    }
    _isExcluded[account⬆] = true;
    _excluded.push(account⬆);
}

ftrace | funcSig
function includeInReward(address account⬆) external onlyOwner {
    require(_isExcluded[account⬆], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account⬆) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account⬆] = 0;
            _isExcluded[account⬆] = false;
            _excluded.pop();
            break;
        }
    }
}
```

❖ The owner can include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

ftrace | funcSig
function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

❖ The owner can change all fees

```
function setTaxFeePercent(uint256 buyTaxFee, uint256 sellTaxFee)
    external
    onlyOwner
{
    _buyTaxFee = buyTaxFee;
    _sellTaxFee = sellTaxFee;
}

ftrace | funcSig
function setBurnFee(uint256 fee) external onlyOwner {
    _burnFee = fee;
}

ftrace | funcSig
function setAdvestisementFeePercent(
    uint256 buyAdvestisementFee,
    uint256 sellAdvestisementFee
) external onlyOwner {
    _sellAdvestisementFee = sellAdvestisementFee;
    _buyAdvestisementFee = buyAdvestisementFee;
}
```

❖ The owner can change the max transaction percentage

```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(10**3);
}
```

- The owner can turn off the antibot (when the antibot is enabled, only owner whitelisted wallets can make transactions)

```
ftrace | funcSig
function turnOffAntibotMode() public onlyOwner {
    isAntibotModeEnabled = false;
}
```

- The owner can set airdrop address

```
ftrace | funcSig
function setAirdropContract(address _airdropContract⬆) public onlyOwner {
    airdropContract = _airdropContract⬆;
}
```

- The owner can add/remove wallets from antibot whitelist

```
ftrace | funcSig
function setAntibotModeWhitelist(
    address[] memory toAddAddesses⬆,
    address[] memory toRemoveAddesses⬆
) public onlyOwner {
    for (uint256 i = 0; i < toAddAddesses⬆.length; i++)
        antibotModeWhitelist[toAddAddesses⬆[i]] = true;
    for (uint256 i = 0; i < toRemoveAddesses⬆.length; i++)
        antibotModeWhitelist[toRemoveAddesses⬆[i]] = false;
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testings, line by line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status: **PASSED**

Number of risk issues: **1**

Solidity code functional issue level: **LOW SEVERITY**

Number of owner privileges: **8**

Centralization risk correlated to the active owner: **HIGH**

Smart contract active ownership: **YES**