STAYSAFU AUDIT

SECURITY ASSESMENT: JANUARY 22ND, 2022

ROSS INU

TABLE OF CONTENTS

SUMMARY

This report has been prepared for ***Ross Inu*** *to discover issues and vulnerabilities in the source code of the* ***Ross Inu*** *project as well as any contract dependencies that were not part of an officially recognized library.*

The audit is based on the code of the following BSC smartcontract:

0x4645C1991Ca64D3f5C45050EaAF9894AeEF24e21

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and ***Ross Inu*** *Deployment techniques. The auditing process pays special attention to the following considerations:*

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Thorough line-by-line manual review of the entire codebase by industry experts

OVERVIEW

VULNERABILITY SUMMARY

UNDERSTANDING

The **Ross Inu** Protocol is a decentralized finance (**DeFi**) token deployed on the Binance smart chain (**BSC**).

Ross Inu mainly employs three features in its protocol: A LP (liquidity pool) acquisition mechanism, an auto-burn process and a marketing/development fee.

Each **Ross Inu** buy transaction is taxed **12%** and each sell is taxed **15%**. **6-8%** (**6%** for buys and **8%** for sales) are accumulated internally until a sufficient amount of capital has been amassed to perform an LP acquisition. When this number is reached, the total tokens accumulated are split with half being converted to **BNB** and the total being supplied to the **PANCAKESWAP** contract as liquidity. **5%** (for both buys and sales) are used for marketing/development and **1-2%** (**1%** for buys et **2%** for sales) are burnt.

PRIVILEGED FUNCTIONS

The contract contains the following privileged functions that are restricted by the **onlyOwner** modifier.
They are used to modify the contract configurations and address attributes. We grouped these functions below:

OWNERSHIP MANAGEMENT

-transferOwner

-renounceOwnership

ACCOUNTS MANAGEMENT

-setExcludedFromFees

-setBlacklistenabled

-includeInFee

-setStartingProtections

-setProtectionSettings

TAXES MANAGEMENT

-setTaxes

-setMaxtxPercent

-setBonusTaxTime

TRADING MANAGEMENT

-setNewRouter

-setLpPair

-setRatios

-setMaxWalletSize

-setSwapSettings

-enableTrading

-setGasPriceLimit

-setWallets

-setSwapAndLiquifyEnabled

-buybackAndBurn

OWNERSHIP

Here is a non-exhaustive list of what the smart-contract owner can and cannot do.

Feature	Able to modify / to do	Details
Transaction fees	**Partially**	Fees are under 20% of transaction
Max transaction	**Partially**	Max transaction is above 0,1% of total supply
Blacklist	**Partially**	Owner can enable/disable blacklist
Whitelist	**No**	
Mint	**No**	
Renounce	**Yes**	
Ownership	**Yes**	

FINDINGS

Unlocked compiler version

Severity: Minor

Ross Inu’s contract does not have locked compiler versions, meaning a range of compiler versions can be used. This can lead to differing bytecodes being produced depending on the compiler version, which can create confusion when debugging as bugs may be specific to a specific compiler version(s).

To rectify this, we recommend setting the compiler to a single version, the lowest version tested to be compatible with the code, an example of this change can be seen below.

Before	After
pragma solidity >=0.6.0<0.6.0;	pragma solidity 0.6.0;

Declaration of unused variables

Severity: Minor

Ross Inu's contract contains instances where a variable is declared but never used. Two of these instances have been identified: **_decimalsMul**, and **tFeeTotal**. There are no functions present in **Ross Inu's** contract code which references either of these variables, making them redundant.

It is best practice to only declare variables which will be used in the code, so we recommend removing any unused variables, as alongside their redundancy, they could create confusion in those reading the contract code. Where these unused declared variables have been identified are listed below.

Code	Line
tFeeTotal	291
_decimalsMul	289

Use of block.timestamp for comparison

Severity: Minor

The value of block.timestamp can be manipulated by the block's miner. This is a security problem since block.timestamp is used when exchanging token for LP acquisition. Moreover, conditions with strict equality are difficult to achieve. This problem can be avoided by not using block.timestamp.

Third-party dependencies

Severity : Minor

The contract is serving as the underlying entity to interact with third party **PancakeSwap** protocols. The scope of the audit would treat those third party entities as black boxes and assume they are fully functionnal. However in the real world, third parties may be compromised that led to assets lost or stolen.

We understand that the business logic of the **Ross Inu Protocol** requires the interaction **PancakeSwap** protocol for adding liquidity to **ROSS/BNB** pool and swap tokens. We encourage the team to constantly monitor the statuses of those third parties to mitigate the side effects when unexpected activities are observed.

Centralization of major privileges

Severity: Medium

The owner of the smart-contract has major privileges over it (they can modify fees, change marketing wallet and recover funds from the contract). This can be a problem, and we recommend at least to use a multi-sig wallet for the owner address, and at best to establish a community governance protocol to avoid such centralization. Overall, this problem is common and **Ross Inu** presents a satisfactory level of centralization.

Conclusion

No major issue has been found in the Ross Inu smart- contract. The findings we reported are low severity issues, and are common to the majority of rewards smart- contracts. The overall security of the smart-contract is very good, the only points that should be improved is the centralization of privileges, and the contract code’s abidance to best practices.

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement.

This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without **StaySAFU**'s prior written consent. This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts **StaySAFU** to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project.

This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.
Blockchain technology and cryptographic assets present a
high level of ongoing risk.

StaySAFU's position is that each company and individual are responsible for their own due diligence and continuous security. **StaySAFU**'s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.