



# Smart Contract Security Audit Report

---

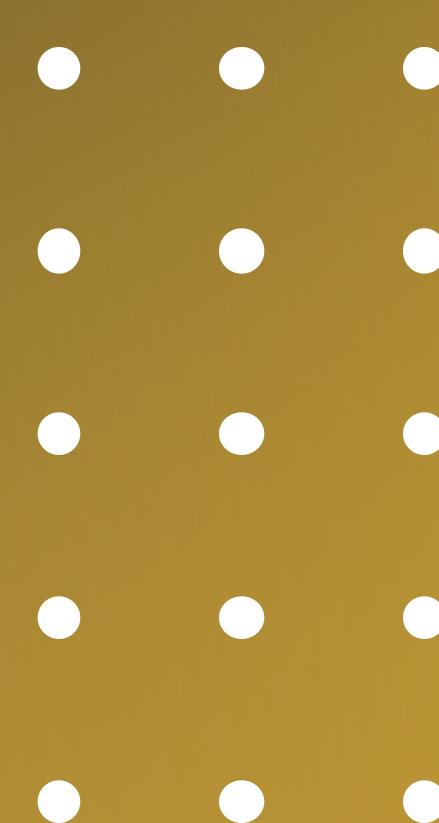
**SHIRO INU**

November 2022

Security Status



[www.hacksafe.io](https://www.hacksafe.io)



# Audit Details



## Audited project

SHIRO INU



## Deployer address

0x77E17431BF806e7eD2e98dd8a56AF45C637BB6c9



## Client contacts

SHIRO INU Team



## Blockchain

Binance smart chain



## Website

<https://shirotoken.com/>



# 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 below 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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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.



# Procedure

## **Step 1 - In-Depth Manual Review**

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

## **Step 2 - Automated Testing**

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

## **Step 3 – Leadership Review**

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

## **Step 4 - Resolution of Issues**

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

## **Step 5 - Published Audit Report**

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

# Background

HackSafe was commissioned by SHIRO INU to perform an audit of smart contracts:

- <https://bscscan.com/address/0x4ddba615a7f6ee612d3a23c6882b698dfbbef7e7#code>

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understood to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

# Contract Details

## Token contract details for 04.11.2022

Token Type	: BEP20
Contract name	: CoinToken
Contract address	: 0x4ddba615a7F6ee612d3a23C6882B698dFBbef7E7
Total supply	: 80,400,000,000
Token ticker	: SHIR
Decimals	: 18
Token holders	: 5,819
Transactions count	: 47,849
Compiler version	: v0.4.24+commit.e67f0147
Contract deployer address	: 0x77E17431BF806e7eD2e98dd8a56AF45C637BB6c9
owner address	: 0x77e17431bf806e7ed2e98dd8a56af45c637bb6c9

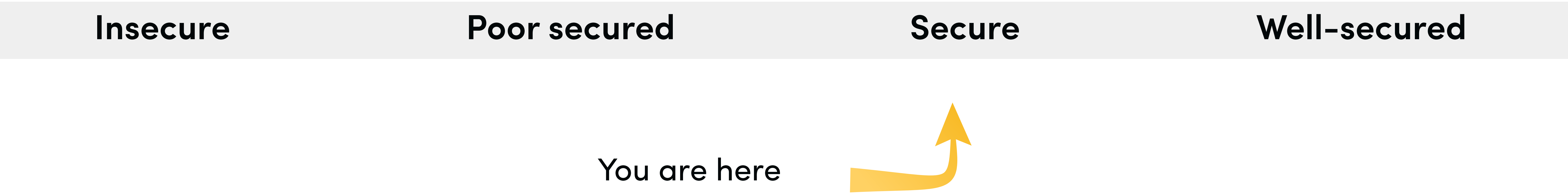


# Social profiles

Twitter Profile	: <a href="https://www.twitter.com/shirotokens">https://www.twitter.com/shirotokens</a>
Telegram profile	: <a href="https://t.me/shirotokens">https://t.me/shirotokens</a>
Coinmarketcap profile	: <a href="https://coinmarketcap.com/currencies/shiro-inu/">https://coinmarketcap.com/currencies/shiro-inu/</a>
Facebook profile	: <a href="https://www.facebook.com/shirotokens">https://www.facebook.com/shirotokens</a>

# Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Secure”**. This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.



We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 2 low.



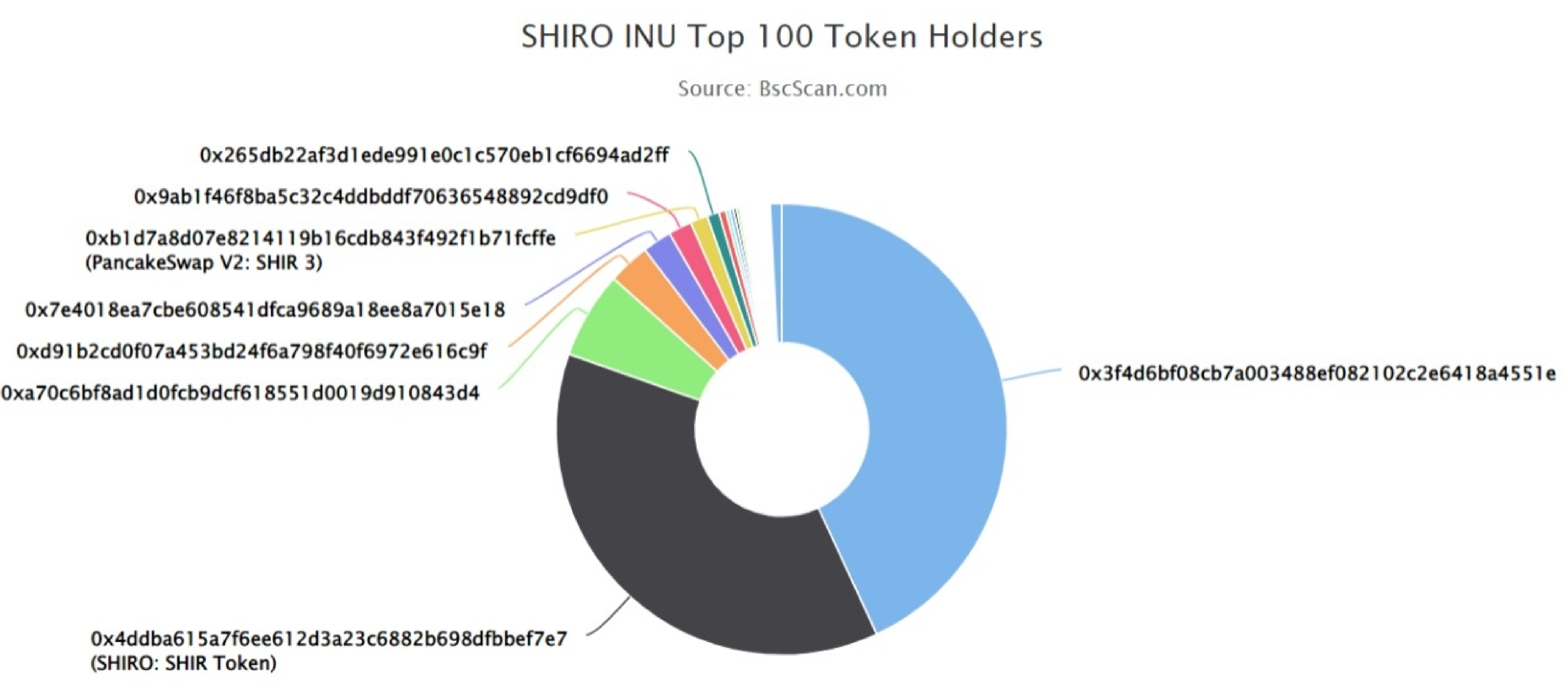
# SHIRO INU Token Distribution

 The top 100 holders collectively own 99.19% (79,747,985,122.42 Tokens) of SHIRO INU

 Token Total Supply: 80,400,000,000.00 Token








|

Total Token Holders: 5,819



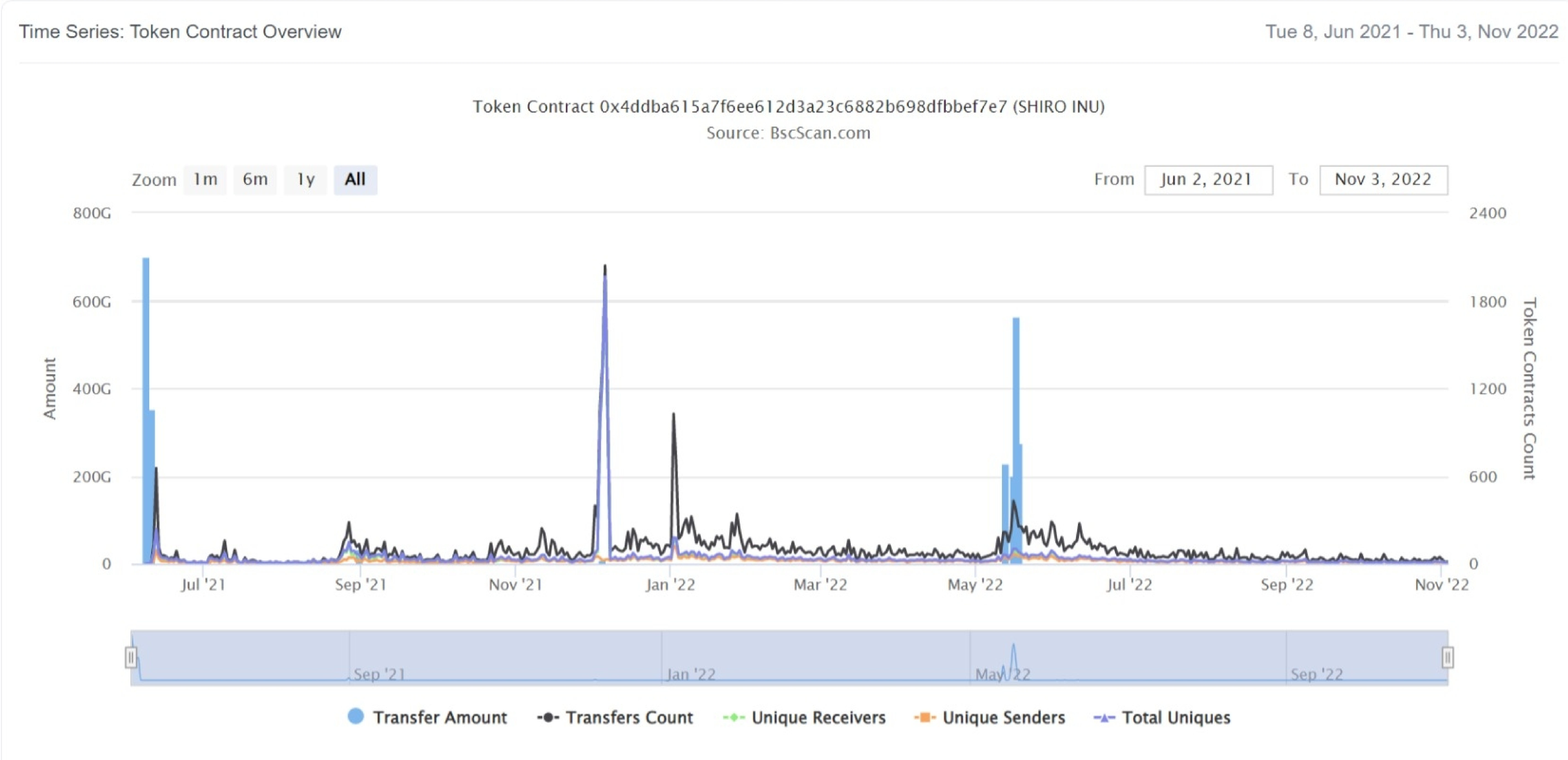
## SHIRO INU Token Top 20 Token Holders

(A total of 79,747,985,122.42 tokens held by the top 100 accounts from the total supply of 80,400,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	 0x3f4d6bf08cb7a003488ef082102c2e6418a4551e	34,647,000,000	43.0933%
2	 SHIRO: SHIR Token	30,000,011,167.367221271960030197	37.3134%
3	0xa70c6bf8ad1d0fcb9dcf618551d0019d910843d4	5,000,000,000	6.2189%
4	 0xd91b2cd0f07a453bd24f6a798f40f6972e616c9f	2,412,911,290.543375762273861891	3.0011%
5	0x7e4018ea7cbe608541dfca9689a18ee8a7015e18	1,680,472,156	2.0901%
6	0x9ab1f46f8ba5c32c4ddbdf70636548892cd9df0	1,342,631,074	1.6699%
7	 PancakeSwap V2: SHIR 3	1,009,341,243.309628488197216186	1.2554%
8	0x265db22af3d1ede991e0c1c570eb1cf6694ad2ff	690,498,351.443670786712077555	0.8588%
9	0x47afae878132835ff1063154a379f71b332fab2e	400,000,000	0.4975%
10	 0xd237e95f06095ee7cec2b7d276a5920c3ac2d620	232,173,318.816776479594409467	0.2888%
11	0xe1af680317e25a53ff20939cce1e06ce533427bd	218,994,372.790062456496637773	0.2724%
12	 0x5763092fe4b763213a600e797f9ff06f449a14b4	173,500,713.8250627483719195	0.2158%
13	 0xacc2adbeb89b6c6273acf13a64ef95efa90107d5	163,936,850.699026224483875953	0.2039%
14	0x5e1e5e551afa75a90730d2e7b8d766ba095bcb3e	97,252,404.397759185205441915	0.1210%
15	0x8ea314a37acbe56401b8fb1e9cc1693db9501595	91,919,008.856382285610731773	0.1143%
16	0xc4bd113e47c9ac60275ee614ca522c5929a0da76	88,452,802.784109930973998581	0.1100%
17	0xf6169615710a80db6394f65ee1c37f07bc514c50	84,484,698.546400633201307944	0.1051%
18	0xa084acde8f9bd102beeba0d07a4c2b84b9cd01af	78,339,443.2	0.0974%
19	0x446805c2cc888c69fef8d066e547fab28ca2c90c	70,000,000.597638446767200467	0.0871%
20	0x3ee608bbd38fa135c943f317578415bd7fd051b9	62,941,413.303877755969472761	0.0783%

# SHIRO INU Token Distribution

## SHIRO INU Token Contract Overview





# Contract functions details

## + [Lib] SafeMath

- [Int] mul
- [Int] div
- [Int] sub
- [Int] add

## +Ownable

- [Pub] transferOwnership #
- modifiers: onlyOwner

## +Pausable (Ownable)

- [Pub] pause #
- modifiers: onlyOwner, whenNotPaused
- [Pub] unpause #
- modifiers: onlyOwnerwhenPaused

## +ERC20Basic

- [Pub] balanceOf
- [Pub] transfer

## +ERC20 (ERC20Basic)

- [Pub] allowance
- [Pub] transferFrom
- [Pub] approve

## +StandardToken (ERC20)

- [Pub] transfer #
- [Pub] balanceOf
- [Pub] transferFrom #
- [Pub] approve #
- [Pub] allowance
- [Pub] increaseApproval #
- [Pub] decreaseApproval #
- [Int] \_blackList #

## +PausableToken (StandardToken, Pausable)

- [Pub] transfer #
- modifiers: whenNotPaused
- [Pub] transferFrom #

# Contract functions details

- modifiers: whenNotPaused
- [Pub] approve #
- modifiers: whenNotPaused
- [Pub] increaseApproval #
- modifiers: whenNotPaused
- [Pub] decreaseApproval #
- modifiers: whenNotPaused
- [Pub] blacklistAddress #
- modifiers: whenNotPaused, onlyOwner

+CoinToken (PausableToken)

- [Pub] <constructor> \$
- [Pub] burn #
- [Int] \_burn #
- [Pub] mint #
- modifiers: onlyOwner

(\$) = payable function

# = non-constant function



# Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Low issue

# Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.



# Security Issues

## ✔ Critical Severity Issues

No critical severity issue found.

## ✔ High Severity Issues

No high severity issues found.

## ✔ Medium Severity Issues

No medium severity issues found.

## ✔ Low Severity Issues

Two low severity issue founds.

### 1. Too old compiler version found.

- **Description**

The smart contract has been deployed using too old compiler version.

- **Recommendation**

It is advisable to deploy contract using any of the latest version of solidity.

### 2. Unlocked Compiler Version.

- **Description**

The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version 0.4.24 the contract should contain the following line:

```
pragma solidity 0.4.24;
```

# Centralization

## Owner Privileges:

- Owner can transfer ownership.
- Owner can pause and unpause transfers.
- Owner can add blacklist address.
- Owner can mint new tokens.

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble as smart contract ownership has not been renounced. Following are Admin functions:

- Transferownership
- Pause
- Unpause
- Blacklistaddress
- Mint



# Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.