



# Smart Contract Security Audit Report

---

## GameStation

October 2022

Security Status



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



# Audit Details



## Audited project

GameStation



## Deployer address

0x9c47DBE7d4B731353125C20485253e5CA75C1c8B



## Client contacts

GameStation Team



## Blockchain

Binance smart chain



## Website

<https://www.gamestation.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 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 GameStation to perform an audit of smart contracts:**

- <https://bscscan.com/address/0x3f6b3595ecf70735d3f48d69b09c4e4506db3f47#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 21.10.2022

Token Type	: BEP20
Contract name	: wGameStationToken
Contract address	: 0x3f6b3595ecF70735D3f48D69b09C4E4506DB3F47
Total supply	: 2,352,299.141676
Token ticker	: GAMER
Decimals	: 18
Token holders	: 461
Transactions count	: 14,037
Compiler version	: v0.8.9+commit.e5eed63a
Contract deployer address	: 0x9c47DBE7d4B731353125C20485253e5CA75C1c8B
Owner address	: 0x9c47dbe7d4b731353125c20485253e5ca75c1c8b

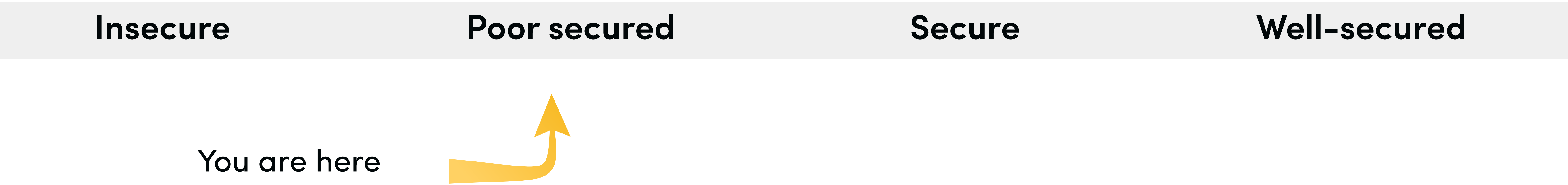


# Social profiles

Twitter profile	: <a href="https://twitter.com/gamestationio">https://twitter.com/gamestationio</a>
Coinmarketcap Profile	: <a href="https://coinmarketcap.com/currencies/gamestation/">https://coinmarketcap.com/currencies/gamestation/</a>
Telegram profile	: <a href="https://t.me/gamestationio_official">https://t.me/gamestationio_official</a>
Coingecko profile	: <a href="https://www.coingecko.com/en/coins/gamestation/">https://www.coingecko.com/en/coins/gamestation/</a>

# Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Poor 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.

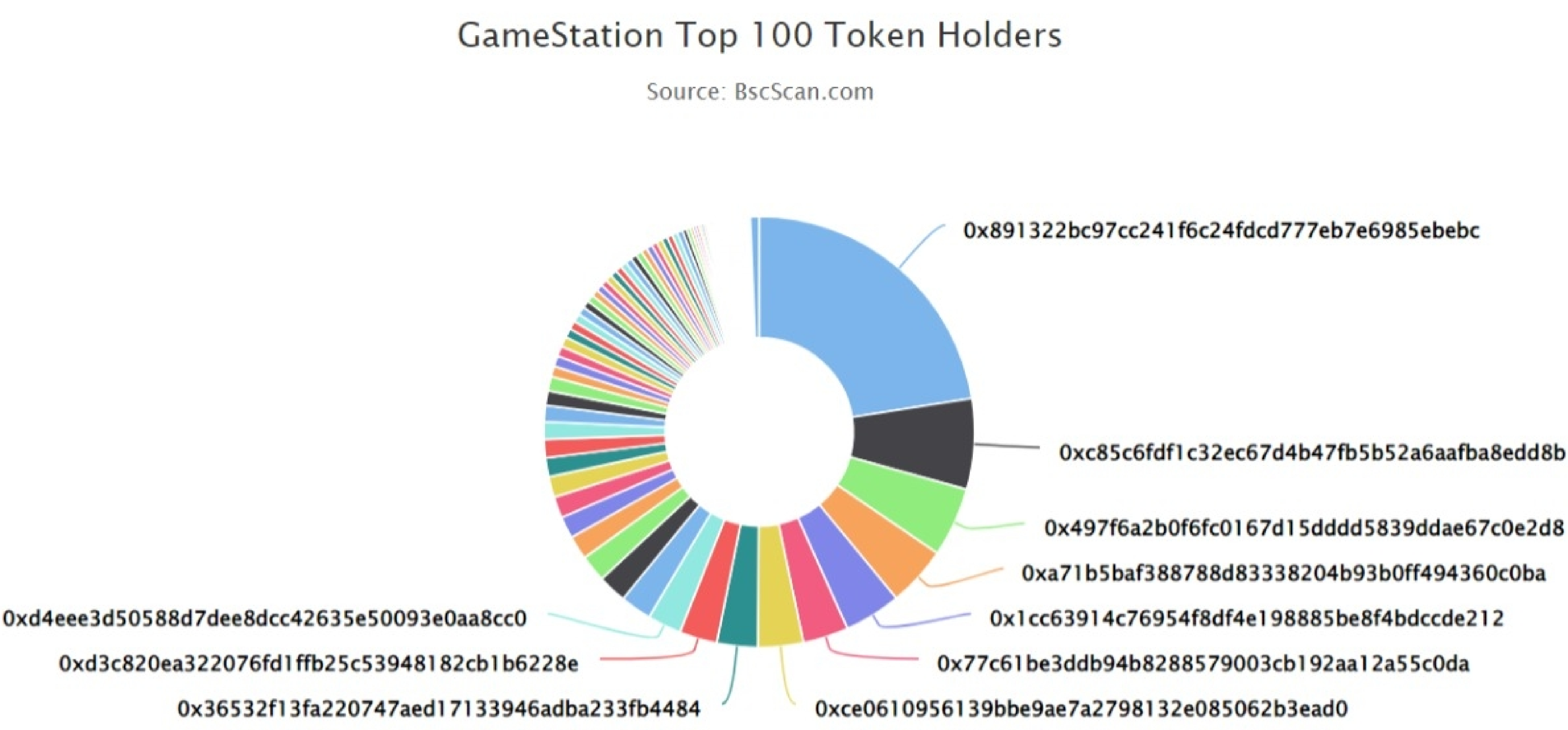
We found 0 critical, 1 high,0 medium and 0 low and some very low-level issues.



# GameStation Token Distribution


💡 The top 100 holders collectively own 99.33% (2,336,607.97 Tokens) of GameStation

💡 Token Total Supply: 2,352,299.14 Token | Total Token Holders: 461



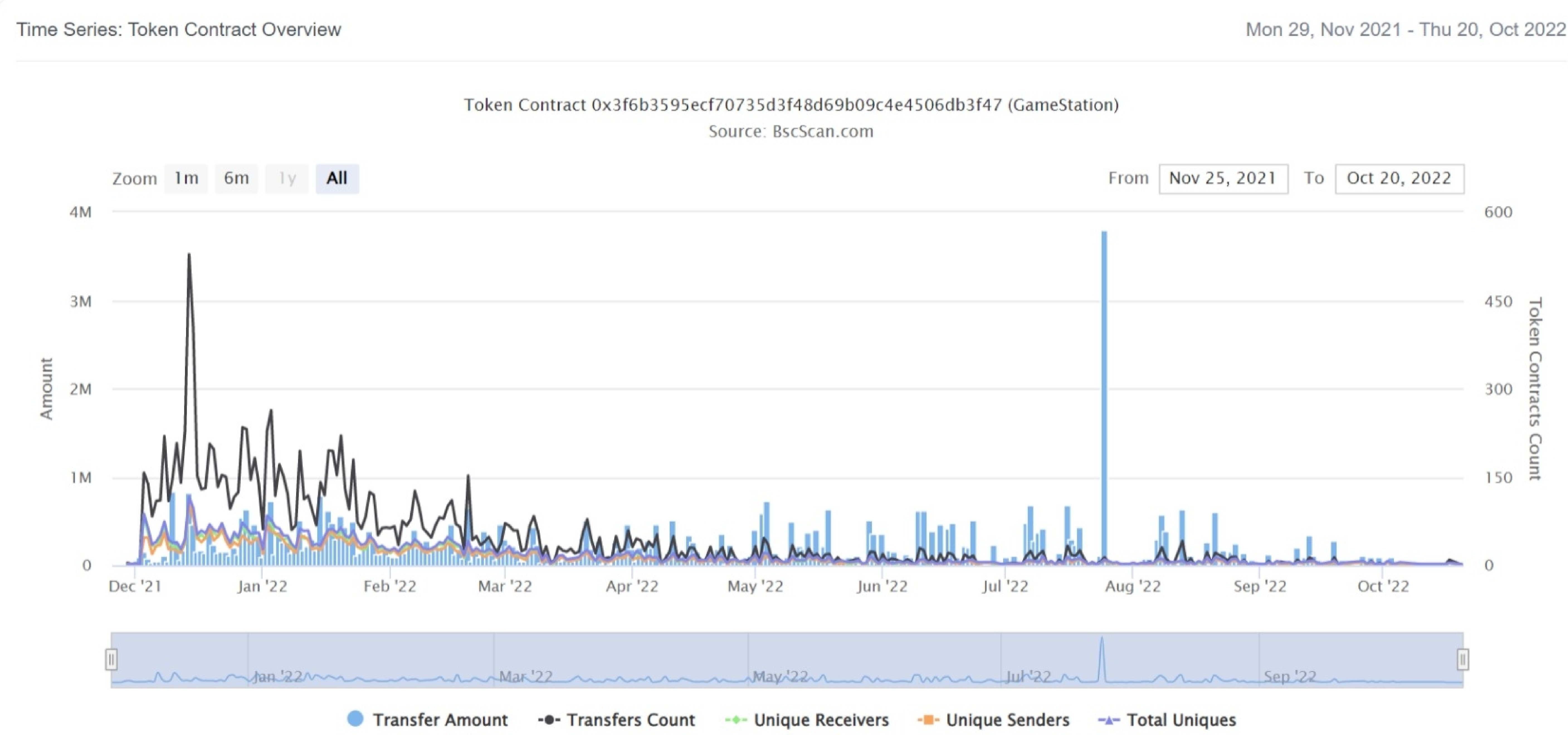
## GameStation Token 20 Token Holders

(A total of 2,336,607.97 tokens held by the top 100 accounts from the total supply of 2,352,299.14 token)

Rank	Address	Quantity (Token)	Percentage
1	0x891322bc97cc241f6c24fdcd777eb7e6985ebebc	529,630.157199522803509662	22.5154%
2	 0xc85c6fdf1c32ec67d4b47fb5b52a6aafba8edd8b	159,852.954778246145518174	6.7956%
3	0x497f6a2b0f6fc0167d15dddd5839ddae67c0e2d8	122,450.188156429985879842	5.2056%
4	0xa71b5baf388788d83338204b93b0ff494360c0ba	106,154.143099362258754665	4.5128%
5	0x1cc63914c76954f8df4e198885be8f4bdccde212	100,236.286850931667880913	4.2612%
6	0x77c61be3ddb94b8288579003cb192aa12a55c0da	80,572.008632045030005279	3.4252%
7	0xce0610956139bbe9ae7a2798132e085062b3ead0	80,099.260757552391339384	3.4051%
8	0x36532f13fa220747aed17133946adba233fb4484	71,996.736112827124721709	3.0607%
9	0xd3c820ea322076fd1ffb25c53948182cb1b6228e	66,792.050874001065820242	2.8394%
10	0xd4eee3d50588d7dee8dcc42635e50093e0aa8cc0	59,756.633529182978131282	2.5404%
11	0x1b6d1540524bbd7f38f88a5cb4dd957fbd0f2c78	55,622.219063459093462378	2.3646%
12	0x864100fd7cee870ccc8795295acdd2b0642e3bdc	50,337.900199654123601817	2.1399%
13	0x5b1a8a8039bf1dbf434b9e1b151bfc6086b7e86	47,562.334203277352920025	2.0220%
14	0xbde8f16afa055c5621aed747155b7c785af99fdb	42,000	1.7855%
15	0x21cc344708ae2ad114ac94f295a1cd60edf60fbe	38,484.986733200421123026	1.6361%
16	0x559e38ca4b12c791149cfdc42c4543d8dc1e8272	37,343.475483882825692132	1.5875%
17	0x26bd1a2d21d1d0f796067e751bd9145bd3bd40d3	37,024.46429774878613357	1.5740%
18	0xeb2941ae10dac00d8d82eb29d4732b8c8588aa41	33,228.53891234696216331	1.4126%
19	0xc2624c2926b24fdeb702e4488e6925a30ba75ef7	32,586.09805898678076107	1.3853%
20	0xa7e4aeac86fdc87a68c82cbf782e06540c4df16e	30,431.428981564363892811	1.2937%

# GameStation Token Distribution

## GameStation Contract Overview





# Contract functions details

wGameStationToken.sol

+ wGameStationToken (ERC20, Pausable, Ownable)

-<constructor>

-[Ext] getOwner

-[Ext] pause

-modifiers: onlyOwner

-[Ext] unpause

-modifiers: onlyOwner

-[Ext] setBridgeAddress

-modifiers: onlyOwner

-[Ext] mint

-modifiers: onlyBridge

-[Ext] burn

-[Ext] burnFrom

-[Ext] \_beforeTokenTransfer

Context.sol

+ Context

-[Int] \_msgSender

-[Int] \_msgData

IERC20Metadata.sol

+ [Int] IERC20Metadata (IERC20)

-[Ext] name

-[Ext] symbol

-[Ext] decimals

IERC20.sol

+ [Int] IERC20

-[Ext] totalSupply

-[Ext] balanceOf

-[Ext] transfer

-[Ext] allowance

-[Ext] approve

-[Ext] transferFrom

ERC20.sol

+ERC20 (Context, IERC20, IERC20Metadata)

-<constructor>

-[Pub] name

# Contract functions details

- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer
- [Pub] allowance
- [Pub] approve
- [Pub] transferFrom
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Int] \_transfer #
- [Int] \_mint #
- [Int] \_burn #
- [Int] \_approve #
- [Int] \_beforeTokenTransfer #
- [Int] \_afterTokenTransfer #

## Pausable.sol

### +Pausable (Context)

- <constructor>
- [Pub] paused
- [Int] \_pause #
  - modifiers: whenNotPaused
- [Int] \_unpause #
  - modifiers: whenNotPaused

## Ownable.sol

### +Ownable (Context)

- <constructor>
- [Pub] owner
- [Pub] renounceOwnership #
  - modifiers: onlyOwner
- [Pub] transferOwnership #
  - modifiers: onlyOwner
- [Pvt] \_setOwner #

(\$) = payable function

# = non-constant function



# Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
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.	High issue
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Passed

# 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

One high severity issues found.

### 1. Safe Open Zeppelin contracts implementation and usage.

- **Description**

The main contract has direct imported open zeppelin smart contracts.

- **Recommendation**

We advise you to not direct import any open zeppelin files in your contracts as any changes in their files can affect your code too.

## ✔ Medium Severity Issues

No medium severity issues found.

## ✔ Low Severity Issues

No low severity issues founds.

# Centralization

## Owner Privileges :

- GameStation Contract :
  - Owner can transfer and renounce ownership.
  - Owner can pause and unpaue transfers.
  - Owner can set bridge address.

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
- Renounceownership
- Setbridgeaddress
- Pause
- Unpause



# Conclusion

Smart contract contains high 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.