



# **RugFreeCoins Audit**



# **GEURO Token Smart Contract Security Audit**

**March 29<sup>th</sup>, 2023**

# Contents

Audit details	1
Disclaimer	2
Overview	3
Background	4
Roadmap	5
Target market and the concept	7
Potential to grow with score points	8
Total Points	8
Contract details	9
Contract code function details	10
Contract description table	12
Security issue checking status	18
Owner privileges	20
Audit conclusion	22



# Audit details



**Audited project**  
GEURO Token



**Contract Address**  
0x6f13b1fb6b2897bb40adbc09f7f6cfad181c0904



**Client contact**  
GEURO Team



**Blockchain**  
Binance smart chain



**Project website**  
<https://gameguru.gg>

# 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.

# Overview

- ✅ No mint function found for the owner, and the owner cannot mint tokens after initial deployment, but mint function is there for the holders.
- ✅ Owner can't set a max transaction limit
- ❌ Owner can pause trading.
- ❌ Owner can change the exit rate
- ✅ Owner can't set fees over 25%.
- ✅ Owner can't blacklist wallets.
- ✅ Owner can't set a max wallet limit
- ✅ Owner can't claim the contract's balance of its own token.

Note: The owner can pause trading and change the exit rate only through the contract or their own Dapp. None of these functions will be applied if the contract is listed on a DEX.

# Background

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

<https://bscscan.com/address/0x6f13b1fb6b2897bb40adb09f7f6cfad181c0904>

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, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

# Roadmap

## **Phase 1**

Safe e-Gift cards and digital gaming products via the first crypto gaming marketplace that connects only vetted wholesalers.

## **Phase 2**

Decentralized stabletoken pegged to the EURO via backed assets within the smart contract.

## **Phase 3**

Secure way for players to trade directly, with each other, in-game assets via its P2P platform

## **Phase 4**

Unique opportunity for players to build and participate in gaming tournaments/communities through its gaming social network.

# Tokenomics

**0% tax when buying & selling if listed on a DEX**

**3% tax when buying & selling (Through the contract)**

- 1% trade goes to the marketing wallet in tokens
- 2% trade goes to the liquidity pool. (Tokens are getting minted without considering this fee)



# 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 of the GEURO ecosystem.
- Anyone who's interested in taking part in the future plans of GEURO Token.
- Anyone who's interested in making financial transactions with any other party using GEURO Token as the currency.

# Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
9	Smart contract security	7/10
10	Smart contract functionality assessment	8/10
Total Points		<b>8.7/10</b>

# Contract details

## Token contract details for 29<sup>th</sup> of March 2023

Contract name	GEURO
Contract address	0x6F13b1Fb6B2897bb40AdBc09f7F6cfAd181C0904
Token supply	1,000,000
Token ticker	GEURO
Decimals	18
Token holders	2
Transaction count	2
GGT address	0xd2359c576632234d1354b20bf51b0277be20c81e
Marketing wallet	0x64fee1a00c2488e74c37368676845bcb16e6090e
Contract deployer address	0x7af19f142705ecfe5259bAb172e21b7ec1e2211B
Contract's current owner address	0x7af19f142705ecfe5259bab172e21b7ec1e2211b

# 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
		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
14	Centralised Risk		HIGH

# 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
<b>AggregatorV3 Interface</b>	<b>Interface</b>			
L	decimals	External !		NO !
L	description	External !		NO !
L	version	External !		NO !
L	getRoundData	External !		NO !
L	latestRoundData	External !		NO !
<b>SafeMath</b>	<b>Library</b>			
L	tryAdd	Internal 🔒		
L	trySub	Internal 🔒		
L	tryMul	Internal 🔒		
L	tryDiv	Internal 🔒		
L	tryMod	Internal 🔒		
L	add	Internal 🔒		
L	sub	Internal 🔒		





L	mul	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	sub	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
<b>ReentrancyGuard</b>	<b>Implementation</b>			
L		Public !	🔴	NO !
L	_nonReentrantBefore	Private 🔒	🔴	
L	_nonReentrantAfter	Private 🔒	🔴	
<b>Context</b>	<b>Implementation</b>			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
<b>Pausable</b>	<b>Implementation</b>	<b>Context</b>		
L		Public !	🔴	NO !
L	paused	Public !		NO !
L	_requireNotPaused	Internal 🔒		
L	_requirePaused	Internal 🔒		
L	_pause	Internal 🔒	🔴	whenNotPaused

L	_unpause	Internal 🔒	🔴	whenPaused
<b>Ownable</b>	<b>Implementation</b>	<b>Context</b>		
L		Public !	🔴	NO !
L	owner	Public !		NO !
L	_checkOwner	Internal 🔒		
L	renounceOwnership	Public !	🔴	onlyOwner
L	transferOwnership	Public !	🔴	onlyOwner
L	_transferOwnership	Internal 🔒	🔴	
<b>IERC20</b>	<b>Interface</b>			
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !	🔴	NO !
L	allowance	External !		NO !
L	approve	External !	🔴	NO !
L	transferFrom	External !	🔴	NO !
<b>IERC20Metadata</b>	<b>Interface</b>	<b>IERC20</b>		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !

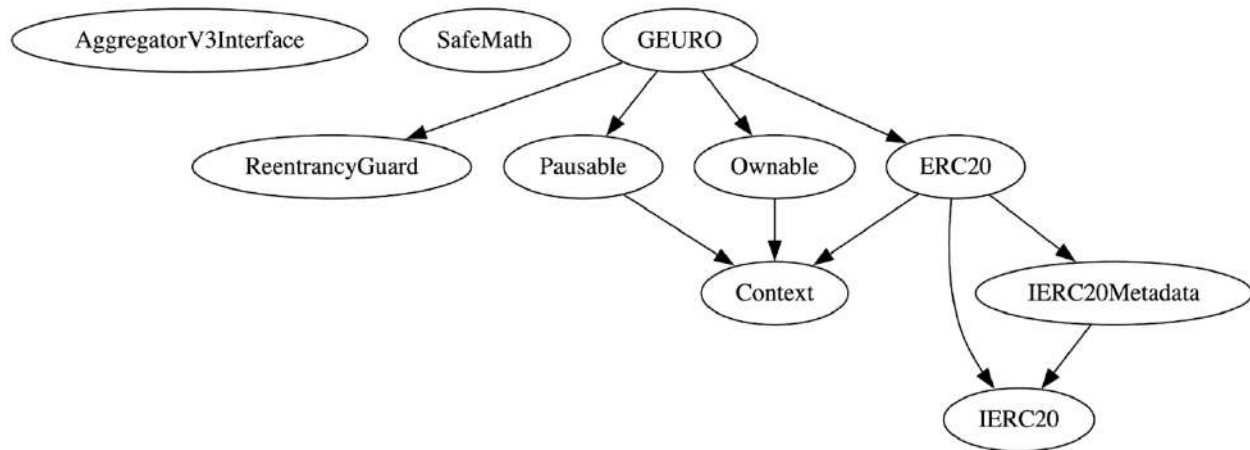
ERC20	Implementation	Context, IERC20, IERC20 Metadata		
L		Public !		NO !
L	name	Public !		NO !
L	symbol	Public !		NO !
L	decimals	Public !		NO !
L	totalSupply	Public !		NO !
L	balanceOf	Public !		NO !
L	transfer	Public !		NO !
L	allowance	Public !		NO !
L	approve	Public !		NO !
L	transferFrom	Public !		NO !
L	increaseAllowance	Public !		NO !
L	decreaseAllowance	Public !		NO !
L	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_approve	Internal 		
L	_spendAllowance	Internal 		
L	_beforeTokenTransfer	Internal 		

L	_afterTokenTransfer	Internal 		
<b>GEURO</b>	<b>Implementation</b>	<b>ERC20, Ownable, Reentrancy Guard, Pausable</b>		
L		Public !		ERC20
L	buy	Public !		whenNotPaused
L	redeem	Public !		whenNotPaused
L	redeemAll	Public !		whenNotPaused
L	pause	Public !		onlyOwner
L	unpause	Public !		onlyOwner
L	updatePriceFeedAddresses	Public !		onlyOwner
L		External !		NO !
L	updateGGTThreshold	External !		onlyOwner
L	getGGTThreshold	Public !		NO !
L	emergencyWithdrawAll	Public !		NO !
L	setEmergencyExchangeRate	External !		onlyOwner
L	toggleEmergencyStatus	External !		onlyOwner

### Legend

Symbol	Meaning
	Function can modify state
	Function is payable

## Inheritance Hierarchy



# Security issue checking status

## ❖ High severity issues

The contract relies on external price feeds (Chainlink oracles) for exchange rates. If these feeds were to be manipulated or compromised, it could result in incorrect exchange rates being used in the contract.

```
ftrace | funcSig
function updatePriceFeedAddresses(
    address newPriceFeedAddress↑,
    address newPriceFeedAddress2↑
) public onlyOwner {
    priceFeed = AggregatorV3Interface(newPriceFeedAddress↑);
    priceFeed2 = AggregatorV3Interface(newPriceFeedAddress2↑);
}
```

The owner can change emergency exchange rate to a very low value.

```
ftrace | funcSig
function setEmergencyExchangeRate(uint256 newRate↑) external onlyOwner {
    require(newRate↑ > 0, "Exchange rate must be greater than 0");
    emergencyExchangeRate = newRate↑;
}
```

The owner can pause the contract anytime.

```
ftrace | funcSig
function pause() public onlyOwner {
    _pause();
}
```

## ❖ Medium severity issues

No medium severity issues found



### ❖ Low severity issues

No input validation for updating the GGT threshold: The `updateGGTThreshold` function does not check if the new threshold provided is valid or not, which could result in an invalid GGT threshold being set.

```
ftrace | funcSig
function updateGGTThreshold(uint256 newThreshold↑) external onlyOwner {
    GGTMIN = newThreshold↑;
    GGT_THRESHOLD = GGTMIN * 10 ** 18; // 100,000 GGT
}
ftrace | funcSig
```

### ❖ Centralization Risk

No Centralization Risk found

# Owner privileges

- ❖ The owner can pause/unpause the contract

```
ftrace | funcSig
function pause() public onlyOwner {
    _pause();
}

ftrace | funcSig
function unpause() public onlyOwner {
    _unpause();
}
```

- ❖ The owner can change price feed address

```
ftrace | funcSig
function updatePriceFeedAddresses(
    address newPriceFeedAddress↑,
    address newPriceFeedAddress2↑
) public onlyOwner {
    _priceFeed = AggregatorV3Interface(newPriceFeedAddress↑);
    _priceFeed2 = AggregatorV3Interface(newPriceFeedAddress2↑);
}
```

- ❖ The owner can change GGTThershold value (GGTThershold value is minimum token amount to hold redeem )

```
ftrace | funcSig
function updateGGTThreshold(uint256 newThreshold↑) external onlyOwner {
    GGTMIN = newThreshold↑;
    GGT_THRESHOLD = GGTMIN * 10 ** 18; // 100,000 GGT
}
```

- ❖ The owner can change emergency exchange rate

```
ftrace | funcSig
function setEmergencyExchangeRate(uint256 newRate↑) external onlyOwner {
  require(newRate↑ > 0, "Exchange rate must be greater than 0");
  emergencyExchangeRate = newRate↑;
}
```

- ❖ The owner can enable/disable emergency exchange

```
ftrace | funcSig
function toggleEmergencyStatus() external onlyOwner {
  isEmergency = !isEmergency;
}
```

# 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: **PASS**

Number of risk issues: **3 High Severity issues associated with centralised ownership**  
**1 Low Severity issue**

Solidity code functional issue level: **PASS**

Number of owner privileges: **5**

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

Smart contract active ownership: **ACTIVE**

Note: The owner can pause trading and change the exit rate only through the contract or their own Dapp. None of these functions will be applied if the contract is listed on a DEX.