

# Advanced Manual **Smart Contract Audit**

September 14, 2022

Audit requested by



**Gingr Bricks**

0x854212e9fce8c713e951199ebcee9ba8efbd92bb

# Table of Contents

## 1. Audit Summary

- 1.1 Audit scope
- 1.2 Tokenomics
- 1.3 Source Code

## 2. Disclaimer

## 3. Global Overview

- 3.1 Informational issues
- 3.2 Low-risk issues
- 3.3 Medium-risk issues
- 3.4 High-risk issues

## 4. Vulnerabilities Findings

## 5. Contract Privileges

- 5.1 Maximum Fee Limit Check
- 5.2 Contract Pausability Check
- 5.3 Max Transaction Amount Check
- 5.4 Exclude From Fees Check
- 5.5 Ability to Mint Check
- 5.6 Ability to Blacklist Check
- 5.7 Owner Privileges Check

## 6. Notes

- 6.1 Notes by Coinsult
- 6.2 Notes by Gingr Bricks

## 7. Contract Snapshot

## 8. Website Review

## 9. Certificate of Proof

# Audit Summary

## Audit Scope

Project Name	Gingr Bricks
Website	<a href="https://www.gingr.world/">https://www.gingr.world/</a>
Blockchain	Binance Smart Chain
Smart Contract Language	Solidity
Contract Address	0x854212e9fce8c713e951199ebcee9ba8efbd92bb
Audit Method	Static Analysis, Manual Review
Date of Audit	14 September 2022

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

## Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0x6d5afc2a52a086e64035cdd1001a4b64ac9f5e4c	2,500,000	100.0000%

## Source Code

Coinsult was commissioned by Gingr Bricks to perform an audit based on the following code:

<https://www.bscscan.com/address/0x854212e9fce8c713e951199ebcee9ba8efbd92bb#code>

# Disclaimer

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

# Global Overview

## Manual Code Review

In this audit report we will highlight the following issues:

Vulnerability Level	Total	Pending	Acknowledged	Resolved
● Informational	0	0	0	0
● Low-Risk	1	1	0	0
● Medium-Risk	1	1	0	0
● High-Risk	0	0	0	0

## Privilege Overview

Coinsult checked the following privileges:

Contract Privilege	Description
Owner can mint?	● Owner can mint new tokens
Owner can blacklist?	● Owner cannot blacklist addresses
Owner can set fees > 25%?	● Owner cannot set the sell fee to 25% or higher
Owner can exclude from fees?	● Owner cannot exclude from fees
Owner can pause trading?	● Owner can pause the smart contract
Owner can set Max TX amount?	● Owner cannot set max transaction amount

More owner privileges are listed later in the report.

● **Low-Risk:** Could be fixed, will not bring problems.

## Too many digits

Literals with many digits are difficult to read and review.

```
_mint(msg.sender, 2500000 * 10 ** decimals());
```

## Recommendation

Use: Ether suffix, Time suffix, or The scientific notation

## Exploit scenario

```
contract MyContract{
    uint 1_ether = 1000000000000000000;
}
```

While `1_ether` looks like 1 ether, it is 10 ether. As a result, it's likely to be used incorrectly.

● **Medium-Risk:** Should be fixed, could bring problems.

## Owner can mint new tokens

```
function mint(address to, uint256 amount) public onlyOwner {  
    _mint(to, amount);  
}
```

## Recommendation

Remove the mint function



# Contract Privileges

## Maximum Fee Limit Check

Coinsult tests if the owner of the smart contract can set the transfer, buy or sell fee to 25% or more. It is bad practice to set the fees to 25% or more, because owners can prevent healthy trading or even stop trading when the fees are set too high.

Type of fee	Description
Transfer fee	● Owner cannot set the transfer fee to 25% or higher
Buy fee	● Owner cannot set the buy fee to 25% or higher
Sell fee	● Owner cannot set the sell fee to 25% or higher

Type of fee	Description
Max transfer fee	0%
Max buy fee	0%
Max sell fee	0%


## Contract Pausability Check

Coinsult tests if the owner of the smart contract has the ability to pause the contract. If this is the case, users can no longer interact with the smart contract; users can no longer trade the token.

Privilege Check	Description
Can owner pause the contract?	● Owner can pause the smart contract


## Max Transaction Amount Check

Coinsult tests if the owner of the smart contract can set the maximum amount of a transaction. If the transaction exceeds this limit, the transaction will revert. Owners could prevent normal transactions to take place if they abuse this function.

Privilege Check	Description
Can owner set max tx amount?	 Owner cannot set max transaction amount

## Exclude From Fees Check

Coinsult tests if the owner of the smart contract can exclude addresses from paying tax fees. If the owner of the smart contract can exclude from fees, they could set high tax fees and exclude themselves from fees and benefit from 0% trading fees. However, some smart contracts require this function to exclude routers, dex, cex or other contracts / wallets from fees.


Privilege Check	Description
Can owner exclude from fees?	 Owner cannot exclude from fees

## Ability To Mint Check

Coinsult tests if the owner of the smart contract can mint new tokens. If the contract contains a mint function, we refer to the token's total supply as non-fixed, allowing the token owner to "mint" more tokens whenever they want.

A mint function in the smart contract allows minting tokens at a later stage. A method to disable minting can also be added to stop the minting process irreversibly.


Minting tokens is done by sending a transaction that creates new tokens inside of the token smart contract. With the help of the smart contract function, an unlimited number of tokens can be created without spending additional energy or money.

Privilege Check	Description
Can owner mint?	 Owner can mint new tokens

## Ability To Blacklist Check

Coinsult tests if the owner of the smart contract can blacklist accounts from interacting with the smart contract. Blacklisting methods allow the contract owner to enter wallet addresses which are not allowed to interact with the smart contract.

This method can be abused by token owners to prevent certain / all holders from trading the token. However, blacklists might be good for tokens that want to rule out certain addresses from interacting with a smart contract.

Privilege Check	Description
Can owner blacklist?	 Owner cannot blacklist addresses

## Other Owner Privileges Check

Coinsult lists all important contract methods which the owner can interact with.

⚠ Interactors with the contract can burn from an external address as long as they have allowance for it.

# Notes

## Notes by Gingr Bricks

We understand that the possibility of minting additional tokens is a point of attention for tokens that are intended as a means of payment. We think it is important to further clarify why our smart contract provides this option.

GinGr.World, the metaverse project in which GGBricks (\$GGB) will circulate, provides an innovative DeFi engine that facilitates the creation and consumption of building materials such as \$GGB.

Within the tokenomics of GinGr.World, bricks are necessary in combination with a building cube NFT to build. This can be compared to bricks in the real world. The platform will automatically burn \$GGB when building with it.

New \$GGB tokens will be created by the play-to-earn management contract – the supply of tokens in combination with their burning can be monitored at all times. The health status of the ecosystem is related to the rational management of tokens.

Other projects, with which we compare, have created many times more tokens when creating their smart contract. We believe that a burn & mint model is better for our purpose; introducing building materials in the metaverse.

Making bricks ultra-rare doesn't make sense to us, our model will distribute new bricks to those who play (play-to-earn). Our DeFi balancer will ensure that the amounts of tokens are in accordance with the building possibilities. We are convinced that our choice to mint-burn tokens is a valuable addition to the governance; that is precisely how our model works.

The value of the token will be determined by the market; we are convinced that our fair and transparent model is better than keeping a large amount of tokens in a team wallet. The team is known (KYC), GinGr is building a regulated business and we believe that transparency and openness is crucial to succeed with this project.

Finally this; we are considering transferring our building materials smart contracts to a governance contract that will also have a DAO (Decentralized Autonomous Organization) aspect. It is the people who walk around in the Metaverse that determine how the ecosystem will evolve. This means that in the future the minting and burning of tokens will not be done by a person or a team, but by a DeFi contract.

## Notes by Coinsult

 No notes provided by Coinsult



# Contract Snapshot

This is how the constructor of the contract looked at the time of auditing the smart contract.

```
contract GGBrick is ERC20, ERC20Burnable, ERC20Snapshot, Ownable, Pausable {
    constructor() ERC20("GG Brick", "GGB") {
        _mint(msg.sender, 2500000 * 10 ** decimals());
    }

    function snapshot() public onlyOwner {
        _snapshot();
    }

    function pause() public onlyOwner {
        _pause();
    }

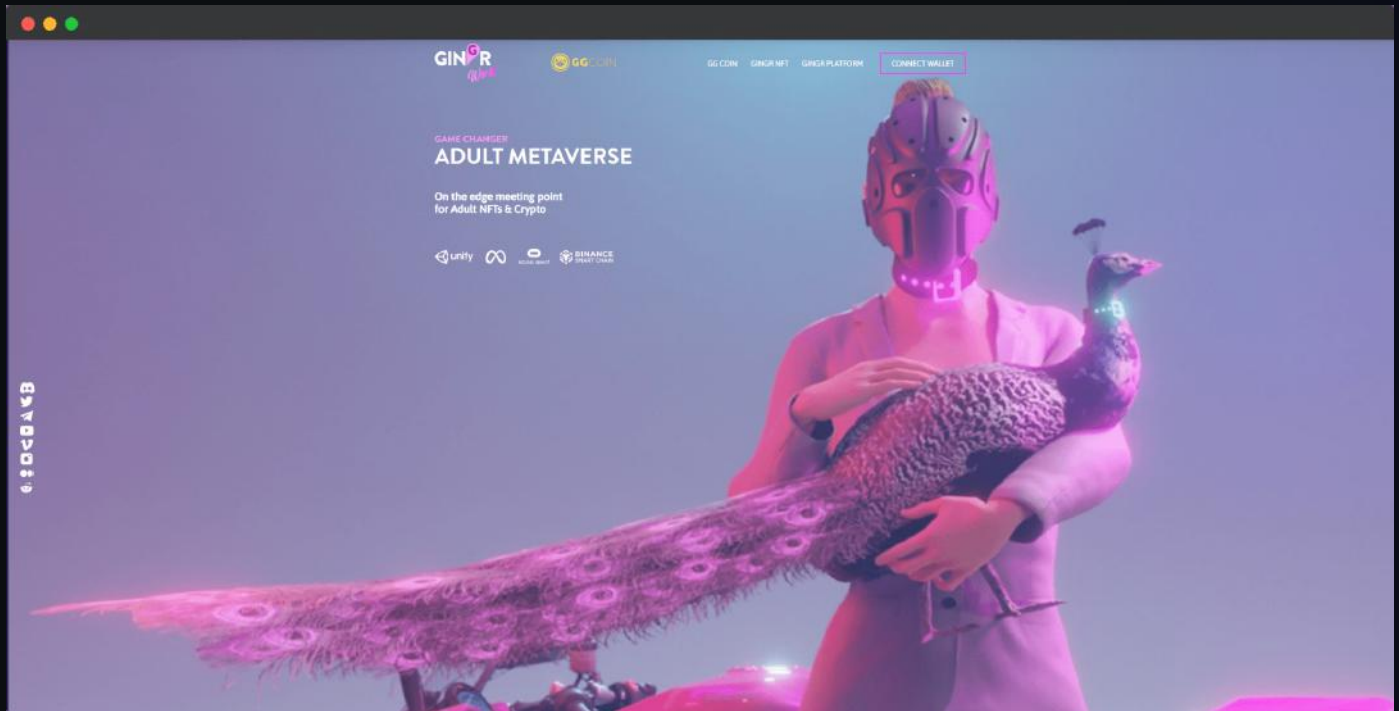
    function unpause() public onlyOwner {
        _unpause();
    }

    function mint(address to, uint256 amount) public onlyOwner {
        _mint(to, amount);
    }

    function _beforeTokenTransfer(address from, address to, uint256 amount)
        internal
        whenNotPaused
        override(ERC20, ERC20Snapshot)
    {
        super._beforeTokenTransfer(from, to, amount);
    }
}
```

# Website Review

Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.



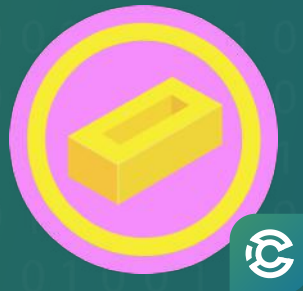
Type of check	Description
Mobile friendly?	● The website is mobile friendly
Contains jQuery errors?	● The website does not contain jQuery errors
Is SSL secured?	● The website is SSL secured
Contains spelling errors?	● The website does not contain spelling errors

# Certificate of Proof

● Not KYC verified by Coinsult

## Gingr Bricks

Audited by Coinsult.net



Date: 14 September 2022

✓ Advanced Manual Smart Contract Audit

# End of report **Smart Contract Audit**

Request your smart contract audit / KYC

**[t.me/coinsult\\_tg](https://t.me/coinsult_tg)**