SAFUAUDIT
SMART CONTRACT AUDITING

TREASURY CASINO
SMART CONTRACT AUDIT

March 13, 2022

INTRODUCTION

Client	Treasury Casino (CCHIPS)
Language	Solidity
Contract address	0x6DCf2718c61f821183Be76334a867401728b1097
Decimals	9
Supply	250,000,000
Platform	Binance Smart Chain
Compiler	v0.8.9+commit.e5eed63a
Optimization	Yes, with 200 runs
Website	https://treasury-casino.online/
Telegram	https://t.me/treasurycasino
Twitter	https://twitter.com/fantasygirlnft

Description

The treasury casino is a crypto casino. The casino chips CCHIPS will give you access to the Casino's staking vault which will have an opening staking pool.

TABLE OF CONTENTS

01 INTRODUCTION

02 ABSTRACT

03 VULNERABILITIES TEST

04 MANUAL ANALYSIS

05 WEBSITE

06 CONCLUSIONS

Approach

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

,

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

Tools

- Remix IDE
- MythX, Mythril
- SWC Registry
- Open Zeppelin Code Analyzer
- Solidity Code Complier

RISK CLASSIFICATION

CRITICAL

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

MEDIUM

Issues on this level could potentially bring problems and should eventually be fixed.

MINOR

Issues on this level are minor details and warning that can remain unfixed but would be better fixed at some point in the future

INFORMATIONAL

Information level is to offer suggestions for improvement of efficacity or security for features with a risk free factor.

ABSTRACT

Fees	Ownership	Mint function
Buy Fees: 5% Sell Fees: 15%	Owned	Yes

Owner can set fees	Max Tx amount	Pause
Owner can set fees up to 10% (buy) / 15% (sell)	Owner can set max Tx amount, but not lower than 0.1%	Owner can't pause trading

Vulnerabilities Test

SWC ID	Description	
SWC-100	Function Default Visibility	Passed
SWC-101	Integer Overflow and Underflow	Passed
SWC-102	Outdated Compiler Version	Passed
SWC-103	Floating Pragma	Passed
SWC-104	Unchecked Call Return Value	Passed
SWC-105	Unprotected Ether Withdrawal	Passed
SWC-106	Unprotected SELF-DESTRUCT Instruction	Passed
SWC-107	Re-entrancy	Passed
SWC-108	State Variable Default Visibility	Minor
SWC-109	Uninitialized Storage Pointer	Passed
SWC-110	Assert Violation	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed
SWC-112	Delegate Call to Untrusted Callee	Passed
SWC-113	DoS with Failed Call	Passed
SWC-114	Transaction Order Dependence	Passed
SWC-115	Authorization through tx.origin	Passed

SWC-116	Block values as a proxy for time	**Passed**
SWC-117	Signature Malleability	**Passed**
SWC-118	Incorrect Constructor Name	**Passed**
SWC-119	Shadowing State Variables	**Passed**
SWC-120	Weak Sources of Randomness from Chain Attributes	**Passed**
SWC-121	Missing Protection against Signature Replay Attacks	**Passed**
SWC-122	Lack of Proper Signature Verification	**Passed**
SWC-123	Requirement Violation	**Passed**
SWC-124	Write to Arbitrary Storage Location	**Passed**
SWC-125	Incorrect Inheritance Order	**Passed**
SWC-126	Insufficient Gas Griefing	**Passed**
SWC-127	Arbitrary Jump with Function Type Variable	**Passed**
SWC-128	DoS With Block Gas Limit	**Passed**
SWC-129	Typographical Error	**Passed**
SWC-130	Right-To-Left-Override control character (U+202E)	**Passed**
SWC-131	Presence of unused variables	**Passed**
SWC-132	Unexpected Ether balance	**Passed**
SWC-133	Hash Collisions With Multiple Variable Length Arguments	**Passed**
SWC-134	Message call with the hardcoded gas amount	**Passed**
SWC-135	Code With No Effects (Irrelevant/Dead Code)	**Passed**
SWC-136	Unencrypted Private Data On-Chain	**Passed**

MANUAL ANALYSIS

The contract is verified to check if functions do and work as they should and malicious code is not inserted.

	Tested	Result
Transfer	Yes	**Passed**
Total Supply	Yes	**Passed**
Buy Back	Yes	**N/A**
Burn	Yes	**Passed**
Mint	Yes	**Passed**
Rebase	Yes	**N/A**
Pause	Yes	**N/A**
Blacklist	Yes	**N/A**
Lock	Yes	**N/A**
Max Transaction	Yes	**Passed**
Transfer Ownership	Yes	**Passed**
Renounce Ownership	Yes	**Passed**

MANUAL AUDIT

CONTRACT INSPECTION

Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
ERC20	Implementation	Context, IERC20		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗

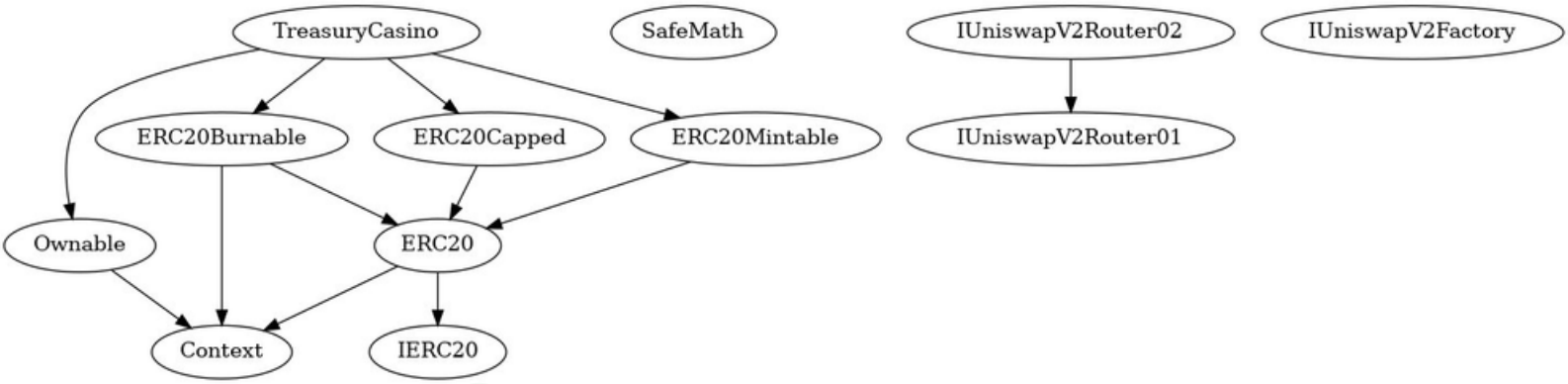
└	transfer	Public ❗	🛑		NO❗
└	allowance	Public ❗			NO❗
└	approve	Public ❗	🛑		NO❗
└	transferFrom	Public ❗	🛑		NO❗
└	increaseAllowance	Public ❗	🛑		NO❗
└	decreaseAllowance	Public ❗	🛑		NO❗
└	_transfer	Internal 🔒	🛑		
└	_mint	Internal 🔒	🛑		
└	_burn	Internal 🔒	🛑		
└	_approve	Internal 🔒	🛑		
└	_setupDecimals	Internal 🔒	🛑		
└	_beforeTokenTransfer	Internal 🔒	🛑		
ERC20Capped	Implementation	ERC20		
└	<Constructor>	Public ❗	🛑		NO❗
└	cap	Public ❗			NO❗
└	_beforeTokenTransfer	Internal 🔒	🛑		
ERC20Burnable	Implementation	Context, ERC20		
└	burn	Public ❗	🛑		NO❗
└	burnFrom	Public ❗	🛑		NO❗
ERC20Mintable	Implementation	ERC20		
└	mintingFinished	Public ❗			NO❗
└	mint	Public ❗	🛑	canMint
└	finishMinting	Public ❗	🛑	canMint
└	_finishMinting	Internal 🔒	🛑		
IUniswapV2Router01	Interface			
└	factory	External ❗			NO❗
└	WETH	External ❗			NO❗
└	addLiquidity	External ❗	🛑		NO❗
└	addLiquidityETH	External ❗	💵		NO❗
└	removeLiquidity	External ❗	🛑		NO❗
└	removeLiquidityETH	External ❗	🛑		NO❗
└	removeLiquidityWithPermit	External ❗	🛑		NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑		NO❗

└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
TreasuryCasino	Implementation	ERC20Capped, ERC20Mintable, ERC20Burnable, Ownable
└	<Constructor>	Public ❗	🛑	ERC20 ERC20Capped
└	<Receive Ether>	External ❗	💵	NO❗
└	excludeFromFees	Public ❗	🛑	onlyOwner
└	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	isExcludedFromFees	Public ❗		NO❗

└	_transfer	Internal 🔒	🛑	
└	swapTokensForBUSD	Private 🔐	🛑	lockTheSwap
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setBuyTax	External ❗	🛑	onlyOwner
└	setSellTax	External ❗	🛑	onlyOwner
└	setMaxTxAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	Public ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner
└	_mint	Internal 🔒	🛑	onlyOwner
└	_finishMinting	Internal 🔒	🛑	onlyOwner
└	_beforeTokenTransfer	Internal 🔒	🛑	

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable
🔐	Private function
🔒	Internal function
NO❗	Function has no modifier

INHERITANCE TREE



Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

Important Snippets

Owner can mint new tokens and also stop the minting

```
function _mint(address account, uint256 amount) internal override onlyOwner {
    super._mint(account, amount);
}

function _finishMinting() internal override onlyOwner {
    super._finishMinting();
}
```

Owner can set fees with a limit

```
function setBuyTax(uint256 _marketingBuyFee) external onlyOwner() {
    require(_marketingBuyFee <= 10, "buy tax must be less than or equal to 10");
    marketingBuyFee = _marketingBuyFee;
}

function setSellTax(uint256 _marketingSellFee) external onlyOwner() {
    require(_marketingSellFee <= 15, "buy tax must be less than or equal to 15");
    marketingSellFee = _marketingSellFee;
}
```

Owner can set max tx but not to 0

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    require(maxTxAmount > totalSupply().div(1000), "transaction amount must be greater than 0.1% of total supply");
    _maxTxAmount = maxTxAmount * (10**9);
}
```

Owner can exclude accounts from fees

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(_isExcludedFromFees[account] != excluded, "CCHIPS: Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

Owner can change swap settings

```
function setSwapEnabled(bool _enabled) public onlyOwner {
    swapEnabled = _enabled;
    emit SwapEnabledUpdated(_enabled);
}
```

GOOD PRACTICES

- The owner cannot stop or pause the smart contract
- The owner cannot set fees over a certain percentage
- The owner cannot set max Tx to 0
- The smart contract utilizes "SafeMath" to prevent overflows

```
library SafeMath {
    function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint2  , {
        unchecked {
            uint256 c = a + b;
            if (c < a) return (false, 0);
            return (true, c);
        }
    }

    function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b > a) return (false, 0);
            return (true, a - b);
        }
    }

    function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            // Gas optimization: this is cheaper than requiring 'a' not being zero, but
            // benefit is lost if 'b' is also tested.
            // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
            if (a == 0) return (true, 0);
            uint256 c = a * b;
            if (c / a != b) return (false, 0);
            return (true, c);
        }
    }

    function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b == 0) return (false, 0);
            return (true, a / b);
        }
    }

    function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b == 0) return (false, 0);
            return (true, a % b);
        }
    }
```

WEBSITE

Website	https://treasury-casino.online
Domain Registry	http://www.wordpress.com
Domain Expiry Date	2023-02-11
Response Code	200
SSL Checker and HTTPS Test	Passed
Deprecated HTML tags	Passed
Robots.txt	Passed
Sitemap Test	Passed
SEO Friendly URL	Low
Responsive Test	Passed
JS Error Test	Passed
Console Errors Test	Low
Site Loading Speed Test	2.38 seconds - Passed
HTTP2 Test	Passed
Safe Browsing Test	Passed

DISCLAIMER

SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

Accuracy of Information

SafuAudit will strive to ensure accuracy of information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only — we recommend proceeding with several independent audits Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. 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.

AUDIT RESULTS

CRITICAL

- Owner can mint new tokens - resulting in tokens being highly inflated. However, calling the _finishMinting() function will stop the minting. Also, renouncing the ownership can prevent minting functions ever being called again.

MEDIUM

No medium severity issues have been found.

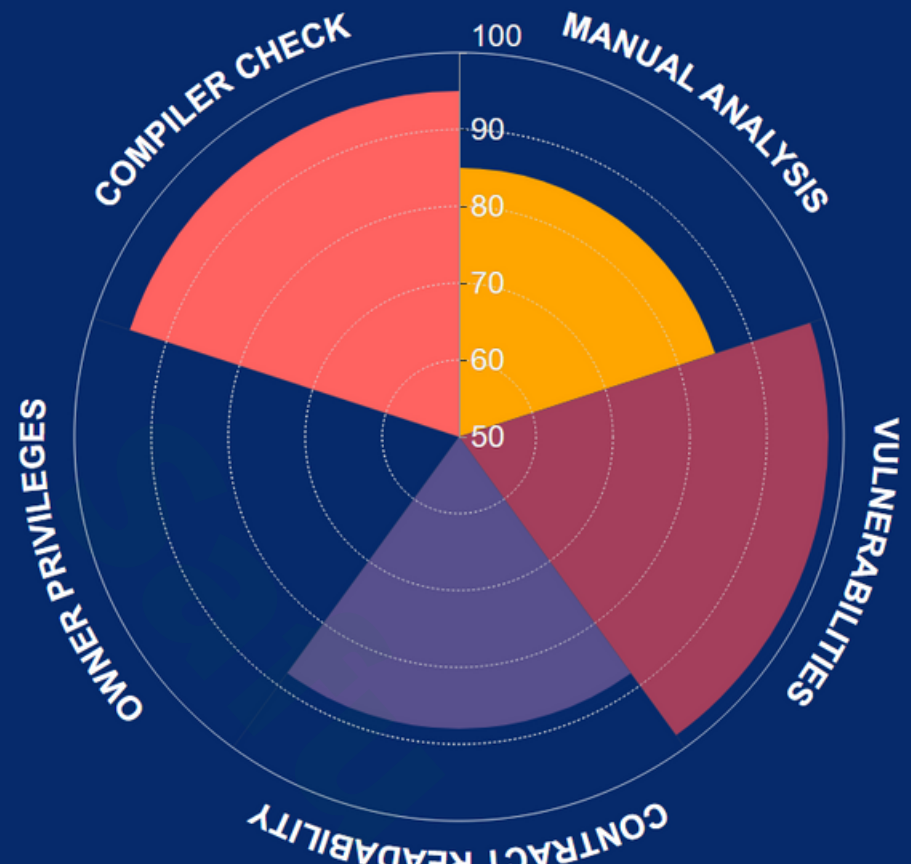
MINOR

- State variable visibility is not set. It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwap" is internal.

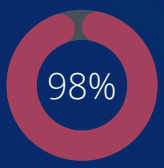
INFORMATIONAL

The standard audit model does not offer suggestions and consulting for improvements of efficacity.

SAFUSCORE



Manual Analysis



Vulnerabilities

Contract Readability

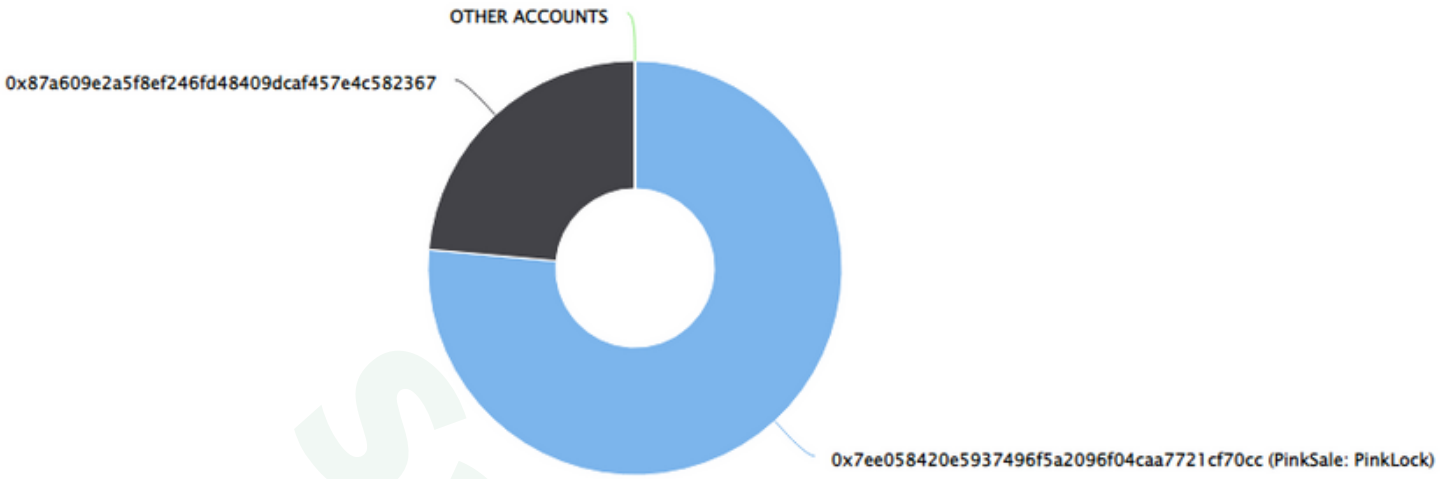
Owner Privileges

Compiler Check

Final Score:83.2

SUMMARY

Top 10 holders



Rank	Address	Quantity (Token)	Percentage
1	PinkSale: PinkLock	191,069,800	76.4279%
2	0x87a609e2a5f8ef246fd48409dcaf457e4c582367	58,930,200	23.5721%

Conclusion

Project Treasury Casino contains a risk characteristic. If the mint functionality is abused, tokens would be highly inflated.

SafuAudit has tested the security based on manual and automated tests. Please note that we don't offer any warranties for business model.

SafuAudit.com