Cyberscope

Audit Report

CryptoGame

May 2022

Github https://github.com/cryptogameavax/Contracts

Commit 4729b00b3e7964bcf72e004cf5c454631ac70efd

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	GAMETOKEN
Github	https://github.com/cryptogameavax/Contracts
Commit	4729b00b3e7964bcf72e004cf5c454631ac70efd
Testing Deploy	https://testnet.bscscan.com/address/0x006aFa96Ca5CC6C2b552F293DEed323a66b6ca21
Symbol	$CARD
Decimals	18
Total Supply	600,000,000
Domain	cgame.tech

Audit Updates

Initial Audit	2nd May 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/token/ERC20/ERC20.sol	f7831910f2ed6d32acff6431e5998baf50e4a00121303b27e974aab0ec637d79
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadc0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/math/SafeMath.sol	15941f3904992a62ed117e93d9e2d5c4c22bd09a7ff97fdd5f49273cf09703ac
contracts/Token.sol	9aea6c7d7fadd9da438499646fcd58f1480ffe782ba38785c0237d9510a77ecb

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L271,278,511

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `m_SellFeePercent` to a high value.

```
if (isSale) fee = m_SellFeePercent;
```

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `m_TxLimit` to zero.

```
if (
    (_sender == m_UniswapV2Pair &&
        _recipient != address(m_UniswapV2Router)) ||
    (_recipient == m_UniswapV2Pair &&
        _sender != address(m_UniswapV2Router))
) require(_amount <= m_TxLimit, "Amount is too big.");
```

The contract owner has the authority to stop the purcanges and transfers for all users excluding the owner. The owner may take advantage of it by setting the `m_MaxWalletSize` to zero.

```
if (
    _recipient != m_UniswapV2Pair &&
    _recipient != address(m_UniswapV2Router)
)
    require(
        ERC20.balanceOf(_recipient) <= m_MaxWalletSize,
        "The balance is too big"
    );
```

Recommendation

The contract could embody a check for not allowing setting the m_TxLimit and m_MaxWalletSize less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

Read more about the fees manipulation in the corresponding section.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L422,430

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setSellFeePercent` function with a high percentage value.

```
function setSellFeePercent(uint256 _sellFeePercent) external onlyOwner {
    m_SellFeePercent = _sellFeePercent;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L267

Description

The contract owner has the authority to massively stop contacts from transactions. The owner may take advantage of it by calling the `setBlackListMultiple` function.

```
require(!m_BlackList[_sender], "You are in block list.");
require(!m_BlackList[_recipient], "You are in block list.");
```

Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
Medium	MC	Missing Check
Minor	L01	Public Function could be Declared External
Minor	L02	State Variables could be Declared Constant
Minor	L04	Conformance to Solidity Naming Conventions
Minor	L06	Missing Events Access Control
Minor	L07	Missing Events Arithmetic

MC - Missing Check

Criticality	medium
Location	contract.sol#L535

Description

The contract is using the `m_SellFeePercent` or `m_BuyFeePercent` in order to determine the total fees. In the fee setter methods, it is not guaranteed that the sum of the underneath fees will be the same with the fee. As a result the contract may add or remove more tokens than it is required. This may cause prices to be inflated or deflated since the balances will be diverted from the total supply.

```
if (isSale) fee = m_SellFeePercent;
else if (isBuy) {
    fee = m_BuyFeePercent;
}

uint256 feeAmount = _amount.mul(fee).div(100);
uint256 feeforLP = _amount.mul(m_FeeforLP).div(200);
uint256 feeforRP = _amount.mul(m_FeeforRP).div(100);
uint256 feeforMarketing_Dev = _amount.mul(m_FeeforMarketing_Dev).div(1000);
uint256 feeforSalaryTeam = _amount.mul(m_FeeforSalaryTeam).div(1000);
uint256 feeforTreasury = _amount.mul(m_FeeforTreasury).div(1000);
uint256 feeforBurn = _amount.mul(m_FeeforBurn).div(1000);

if (feeAmount != 0) {
    _transfer(_sender, m_UniswapV2Pair, feeforLP);
    _transfer(_sender, rewardPool, feeforRP);
    _transfer(_sender, marketingWallet, feeforMarketing_Dev);
    _transfer(_sender, devWallet, feeforSalaryTeam);
    _transfer(_sender, treasuryWallet, feeforTreasury);
    _transfer(_sender, address(this), feeforLP);
    _burn(_sender, feeforBurn);
}

if (isSale) _payToll();
return _amount.sub(feeAmount);
```

Recommendation

The contract should guarantee that the sum of the underneath fees is the same with the amount that is going to be decreased from the sender.

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/Token.sol#L309

Description

Public functions that are never called by the contract should be declared external to save gas.

```
setBlackListMultiple
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contracts/Token.sol#L231,229,222,220,223,221,224,225,227,228,232,233

Description

Constant state variables should be declared constant to save gas.

```
treasuryWallet
marketingWallet
m_RewardPercent
m_LiquidityPercent
m_FeeforTreasury
m_FeeforSalaryTeam
m_FeeforRP
m_FeeforMarketing_Dev
m_FeeforLP
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/Token.sol#L24,25,42,78,293,301,305,309,318,322,326,334,342,354,355,356,414,418,422,430,438,446,540,553,218,219,220,221,222,223,224,225,226,227,228,229,236,238,239,241,242,244,246,247,250

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
m_SwapEnabled
m_UniswapV2Router
m_UniswapV2Pair
m_NumOfTokensForDisperse
m_MaxWalletSize
m_TxLimit
m_PublicTradingOpened
m_BlackList
m_IsSwap
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L06 - Missing Events Access Control

Criticality	minor
Location	contracts/Token.sol#L418

Description

Detected missing events for critical access control parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
m_UniswapV2Pair = _pairAddress
```

Recommendation

Emit an event for critical parameter changes.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contracts/Token.sol#L326,334,422,430,446

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
m_NumOfTokensForDisperse = _numOfTokensForDisperse.mul(10 ** 18)
m_BuyFeePercent = _buyFeePercent
m_SellFeePercent = _sellFeePercent
m_MaxWalletSize = _maxWalletSize.mul(10 ** 18)
m_TxLimit = _txLimit.mul(10 ** 18)
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
IUniswapV2Pair	Interface			

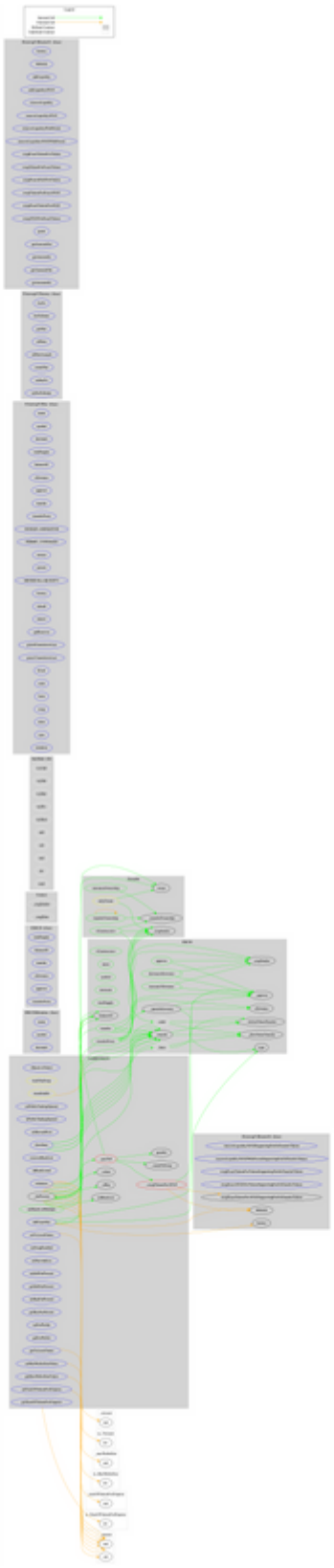
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-

	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WAVAX	External		-
	addLiquidity	External	✓	-
	addLiquidityAVAX	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityAVAX	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityAVAXWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactAVAXForTokens	External	Payable	-
	swapTokensForExactAVAX	External	✓	-
	swapExactTokensForAVAX	External	✓	-
	swapAVAXForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityAVAXSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityAVAXWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactAVAXForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForAVAXSupportingFeeOnTransferTokens	External	✓	-

GAMETOKEN	Implementation	Context, ERC20, Ownable		
	<Receive Ether>	External	Payable	-
	<Constructor>	Public	✓	ERC20
	setPublicTradingOpened	External	✓	onlyOwner
	isPublicTradingOpened	External		-
	setRewardPool	External	✓	onlyOwner
	setBlackList	Public	✓	onlyOwner
	setBlackListMultiple	Public	✓	onlyOwner
	removeBlackList	External	✓	onlyOwner
	isBlackListed	External		-
	setTxLimitToken	External	✓	onlyOwner
	getTxLimitToken	External		-
	setMaxWalletSizeToken	External	✓	onlyOwner
	getMaxWalletSizeToken	External		-
	transfer	Public	✓	transferable
	transferFrom	Public	✓	transferable
	addLiquidity	External	✓	onlyOwner
	setSwapEnabled	External	✓	onlyOwner
	setPairAddress	External	✓	onlyOwner
	setSellFeePercent	External	✓	onlyOwner
	getSellFeePercent	External		-
	setBuyFeePercent	External	✓	onlyOwner
	getBuyFeePercent	External		-
	setFeeWallet	External	✓	onlyOwner
	getFeeWallet	External		-
	setNumOfTokensForDisperse	External	✓	onlyOwner
	getNumOfTokensForDisperse	External		-
	_isBuy	Private		
	_isSale	Private		
	_swapTokensForAVAX	Private	✓	lockTheSwap
	_readyToSwap	Private		
	_payToll	Private	✓	
	_feeProcess	Private	✓	
	withdraw	External	✓	onlyOwner

	distribute	External	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	cgame.tech
Registry Domain ID	D274172382-CNIC
Creation Date	2022-02-08T21:37:25+00:00
Updated Date	2022-03-08T12:02:25+00:00
Registry Expiry Date	2023-02-08T23:59:59+00:00
Registrar WHOIS Server	whois.tucows.com
Registrar URL	http://www.tucows.com/
Registrar	Tucows.com Co.
Registrar IANA ID	69

The domain has been created 3 months before the creation of the audit. It will expire in 9 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

There are some functions that can be abused by the owner like stopping transactions, manipulating fees and massively blacklisting addresses. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io