Cyberscope

Audit Report
2023 Mega Lotto

December 2022

Type	BEP20
Network	BSC
Address	0x2adecfc853b6d5bc9f07ff3d7691a4be0be9e627
Audited by	© cyberscope

Table of Contents

Review

Contract Name	MegaLotto
Compiler Version	v0.8.12+commit.f00d7308
Optimization	200 runs
Explorer	https://bscscan.com/address/0x2adecfc853b6d5bc9f07ff3d7691a4be0be9e627
Address	0x2adecfc853b6d5bc9f07ff3d7691a4be0be9e627
Network	BSC
Symbol	2023
Decimals	18
Total Supply	10,000,000

Audit Updates

Initial Audit	19 Dec 2022 https://github.com/cyberscope-io/audits/tree/main/1-2023/v1/audit.pdf
Corrected Phase 2	21 Dec 2022 https://github.com/cyberscope-io/audits/tree/main/1-2023/v2/audit.pdf
Corrected Phase 3	22 Dec 2022

Source Files

Filename	SHA256
@chainlink/contracts/src/v0.8/interfaces/LinkTokenInterface.sol	918f6de793c6af9b880ab389ad62b16e5f28f5f7719b8501224c40186f9a4837
@chainlink/contracts/src/v0.8/VRFConsumerBase.sol	4090337843498ac18bbfc9dda04e023d3c6e33ba6c4364795a437b8d197b09a0
@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol	3fd22ee3613205ee2e48052363123fd80aabc29c4638490d6096ca10e0df5e83
contracts/interfaces/IDealer.sol	8a271ec44d4c83ca0a4eeaeddd66507d298093c05a55b7db42aef8a9511c2eb1
contracts/interfaces/IUniswapV2Factory.sol	51d056199e3f5e41cb1a9f11ce581aa3e190cc982db5771ffeef8d8d1f962a0d
contracts/interfaces/IUniswapV2Pair.sol	29c75e69ce173ff8b498584700fef76bc81498c1d98120e2877a1439f0c31b5a
contracts/interfaces/IUniswapV2Router01.sol	4c12c0f98671beaa0c53ec9210a12bcd40dcbf097305aac757452b24e2853b96
contracts/interfaces/IUniswapV2Router02.sol	1641ff55f44aaefca1712c3503598fa167685a09923a598f5d0c4c8cd4e926cd
contracts/tokens/2023Mega.sol	acb58af59647623650bda6807b1970bf505cba3334f1e843ddb13c2fda93e655
contracts/tokens/Dealer.sol	2683700977f6733d1d0dbc39e6777c361fb9e9805b907f77088dc0f9f0aba731

Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔵	ST	Stops Transactions	Passed
🔵	OCTD	Transfers Contract's Tokens	Passed
🔵	OTUT	Transfers User's Tokens	Passed
🔵	ELFM	Exceeds Fees Limit	Passed
🔵	ULTW	Transfers Liquidity to Team Wallet	Passed
🔴	MT	Mints Tokens	Acknowledged
🔴	BT	Burns Tokens	Acknowledged
🔵	BC	Blacklists Addresses	Passed

BT - Burns Tokens

Criticality	Critical
Location	contracts/tokens/2023Mega.sol#L437
Status	Acknowledged

Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the `distributeReward` function. As a result, all the winner contract addresses will lose the corresponding tokens.

```
_balances[_selectedWinner] = 0;
```

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.

Team Update

The `distributeReward` can be called by everyone after called the `randomSeedforWinner` function completely and got the random Seed from chainlink in the `fulfillRandomness` function. The burning token process is one of process to provide rewards. When the address is picked as a winner, the token of the winner will be burned to avoid duplicated winners for the next picking winner in the contract.

MT - Mints Tokens

Criticality	Critical
Location	contracts/tokens/2023Mega.sol#L416
Status	Acknowledged

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `distributeReward` function, which mints tokens equal to 1 million percent of the `totalSupply`. As a result, the contract tokens will be highly inflated.

```
_mint(address(this), 100000000000 * 10 ** decimals());
```

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.

Team Update

The `distributeReward` can be called by everyone after called the `randomSeedforWinner` function completely and got the random Seed from chainlink in the `fulfillRandomness` function. As our feature of this token, the minting token process is one of process to drain the liquidity and send it as rewards to the 10 winners.

Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor / Informative

Severity	Code	Description	Status
🔴	PVC	Price Volatility Concern	Acknowledged
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L13	Divide before Multiply Operation	Unresolved

PVC - Price Volatility Concern

Criticality	Critical
Location	contracts/tokens/2023Mega.sol#L416,417
Status	Acknowledged

Description

The contract mints tokens by calling the `distributeReward` function and swaps them. The swap amount is way higher than the `totalSupply`. It is important to note that the price of the token representing it, can be highly volatile. This means that the value of a price volatility swap involving Ether could fluctuate significantly at the triggered point, potentially leading to significant price volatility for the parties involved.

```
_mint(address(this), 100000000000 * 10 ** decimals());
_swapSell(_balances[address(this)], dealer);
```

Recommendation

The contract could ensure that it will not sell more than a reasonable amount of tokens in a single transaction. A suggested implementation could check that the maximum amount should be less than a fixed percentage of the total supply. Hence, the contract will guarantee that it cannot accumulate a huge amount of tokens in order to sell them.

Team Update

As we explained previously, the `distributeReward` can be called by everyone after called the `randomSeedforWinner` function completely and got the random Seed from Chainlink in the `fulfillRandomness` function. The minting token and swap it to sell is a process to drain the liquidity and distribute it to winners from our contract that randomized by using Chainlink. The trading of the token will be ended after the random and reward's distribution happened, so the price means nothing.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/tokens/Dealer.sol#L115,115,115 contracts/tokens/2023Mega.sol#L106,110,325,330,336,342,348,354,425,446,461
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of your Solidity code, making it easier for others to understand and work with.

The followings are few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of your code.
7. Keep lines short (around 120 characters) to improve readability.

```
address _to
address _token
uint256 _amount
mapping(address => uint256) internal _balances
uint256 internal _totalSupply
address _adr
uint256 _fee
uint256 _fee
uint256 _fee
address _wallet
bool _status
address _whitelist
uint256 _seed
uint256 _initialRandomNumber

...
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

You can find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L13 - Divide before Multiply Operation

Criticality	Minor / Informative
Location	contracts/tokens/2023Mega.sol#L421
Status	Unresolved

Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause lose of prediction.

```
uint256 rewardAmount = WBNBAmount / 1000000 * (1000000/winnerNumber)
```

Recommendation

To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
LinkTokenInterface	Interface			
	allowance	External		-
	approve	External	✓	-
	balanceOf	External		-
	decimals	External		-
	decreaseApproval	External	✓	-
	increaseApproval	External	✓	-
	name	External		-
	symbol	External		-
	totalSupply	External		-
	transfer	External	✓	-
	transferAndCall	External	✓	-
	transferFrom	External	✓	-
VRFConsumerBase	Implementation	VRFRequestIDBase		
	fulfillRandomness	Internal	✓	
	requestRandomness	Internal	✓	
		Public	✓	-
	rawFulfillRandomness	External	✓	-
VRFRequestIDBase	Implementation			
	makeVRFInputSeed	Internal		
	makeRequestId	Internal		

IDealer	Interface			
	sendToken	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
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		-

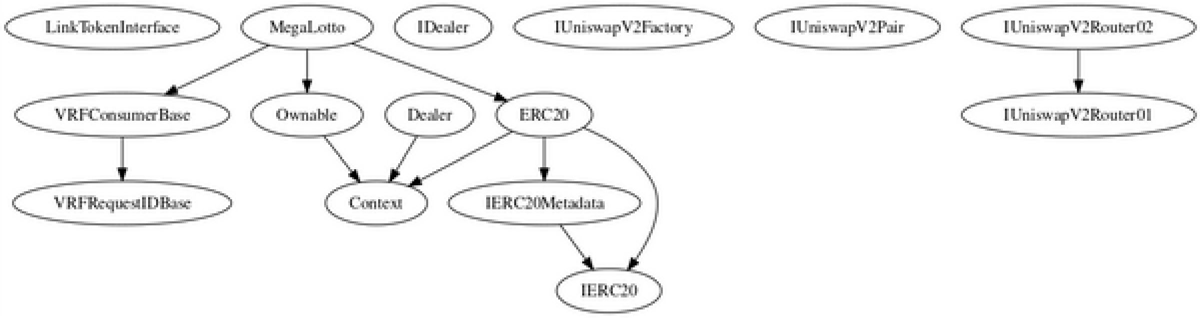
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-

	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-

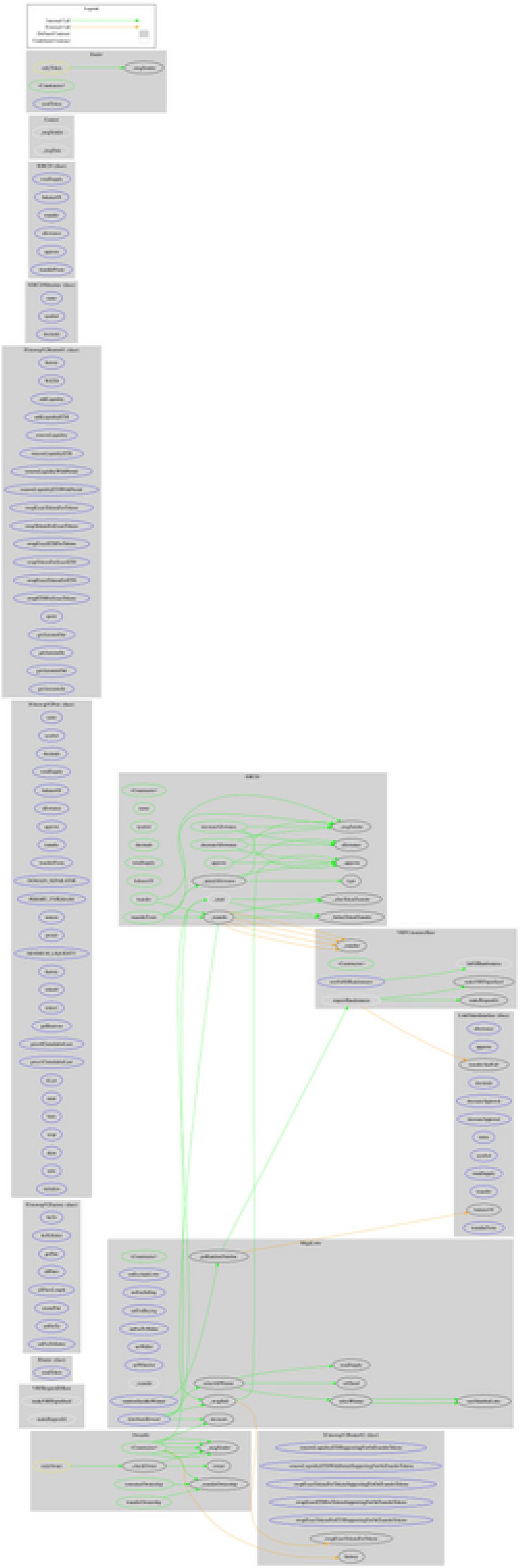
ERC20	Implementation	Context, IERC20, IERC20Metadata		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
MegaLotto	Implementation	ERC20, Ownable, VRFConsumerBase		
		Public	✓	ERC20 VRFConsumerBase
	setExcludeLotto	External	✓	onlyOwner
	setFeeSelling	External	✓	onlyOwner
	setFeeBuying	External	✓	onlyOwner
	setFeeToWallet	External	✓	onlyOwner

	setWallet	External	✓	onlyOwner
	setWhitelist	External	✓	onlyOwner
	_transfer	Internal	✓	
	_swapSell	Internal	✓	
	randomSeedforWinner	External	✓	-
	distributeReward	External	✓	-
	selectAllWinner	Internal	✓	
	selectWinner	Internal		
	userNumberLotto	Internal		
	selfSeed	Internal		
	getRandomNumber	Internal	✓	
	fulfillRandomness	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Dealer	Implementation	Context		
		Public	✓	-
	sendToken	External	✓	onlyToken

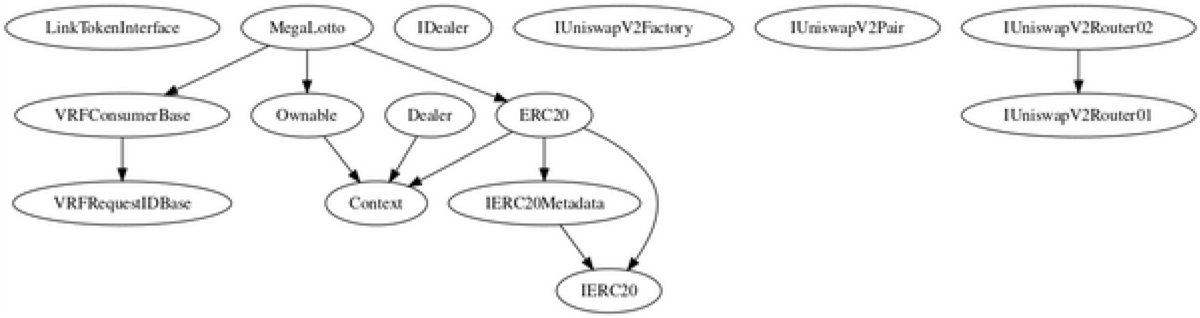
Inheritance Graph



Flow Graph



Contract Inheritance



Summary

There are some functions that can be abused by the owner like mint tokens and burn tokens from any address. If the contract owner abuses the mint functionality, then the contract will be highly inflated. If the contract owner abuses the burn functionality, then the users could lost their tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 20% sell fees and 5% buy fees.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors’ funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io