Cyberscope

Audit Report

MGB

August 2022

SHA256 c2b998096917a1a2357da12055b1a39b4fb77d03e0bda0909ac24d49794e8983

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	MGB
Compiler Version	v0.8.10+commit.fc410830
Testing Deploy	https://testnet.bscscan.com/address/0xa08dFF4285Ff9b18704475d8130bC0308cBCEEcC
Symbol	MGB
Decimals	18
Total Supply	Initialized on the constructor
Domain	https://www.magnummeta.com

Audit Updates

Initial Audit	25th August 2022 https://github.com/cyberscope-io/audits/tree/main/1-mgb/v1/mgb.pdf
Corrected	30th August 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	5af1771388b4fe634e0a566716e32c6d00a5372875099127b274d4cf8a94e9d2
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	fa36a21bd954262006d806b988e4495562e7b50420775e2aa0deecb596fd1902

@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/Strings.sol	34127ad0054df5963b0fd694c1b313d17e9114a2f426b85526d6d976210298ab
contracts/MGB.sol	c2b998096917a1a2357da12055b1a39b4fb77d03e0bda0909ac24d49794e8983
contracts/ReflectToken.sol	b880bbc8d781a735d8d6038a7740fe92946036ce95c84cdd3f4e9589c6e4d86a

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L105,44,55,63,74
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the withdrawToken, withdrawCommunityRewardPool, withdrawBuyback, withdrawProvideLiquidity, withdrawDistribute methods.

```
function withdrawToken(address token, uint256 amount)
    external
    onlyRole(ADMIN_ROLE)
  {
    ERC20(token).safeTransfer(msg.sender, amount);
  }

function withdrawCommunityRewardPool(address account)
    external
    onlyRole(DAO_ROLE)
  {
    _transfer(address(this), account, _communityRewardPool);
    _communityRewardPool = 0;
  }

  function withdrawBuyback(address account) external onlyRole(DAO_ROLE) {
    _transfer(address(this), account, _buyback);
    _buyback = 0;
  }

  function withdrawProvideLiquidity(address account)
    external
    onlyRole(DAO_ROLE)
  {
    _transfer(address(this), account, _provideLiquidity);
    _provideLiquidity = 0;
  }
```

```
function withdrawDistribute(address account) external onlyRole(ADMIN_ROLE) {
        uint256 distributedAmount = balanceOf(address(this)) -
            _provideLiquidity -
            _buyback -
            _communityRewardPool;
        _transfer(address(this), account, distributedAmount);
    }
```

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.

ELFM - Exceeds Fees Limit

Criticality	critical
Location	contract.sol#L30
Status	Unresolved

Description

The contract owner has the authority to initialize the fees over the allowed limit of 25%. The owner may take advantage of it by setting the _feePecent to 99%.

```
constructor(uint256 feePercent, uint256 initialSupply)
        ReflectToken("Magnumbits", "MGB", initialSupply)
    {
        if(feePercent > 100) {
            revert();
        }

        _owner = msg.sender;
        _feePecent = feePercent;

        _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _setupRole(ADMIN_ROLE, msg.sender);
        _setupRole(BURNER_ROLE, msg.sender);
    }
```

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.

BT - Burns Tokens

Criticality	critical
Location	contract.sol#L95
Status	Unresolved

Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the burn function. As a result the targeted contract address will lose the corresponding tokens.

```
function burn(address from, uint256 tAmount)
        external
        onlyRole(BURNER_ROLE)
    {
        _burn(from, tAmount);
    }
```

Recommendation

The owner should carefully manage the credentials of the owner’s account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor / Informative

Severity	Code	Description	Status
●	BLC	Business Logic Concern	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L164
Status	Unresolved

Description

The business logic seems peculiar. The implementation may not follow the expected behaviour.

The MGB contract overrides only _transfer from the ReflectToken contract. The fee logic is not applied on the transferFrom but the taxes are applied on the ReflectToken method.

```
function _transfer(
        address sender,
        address recipient,
        uint256 tAmount
    ) internal virtual override {
        _calcFees(sender, recipient, tAmount);
        super._transfer(sender, recipient, tAmount);
    }
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/ReflectToken.sol#L19,18
Status	Unresolved

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.

```
SYMBOL
NAME
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner

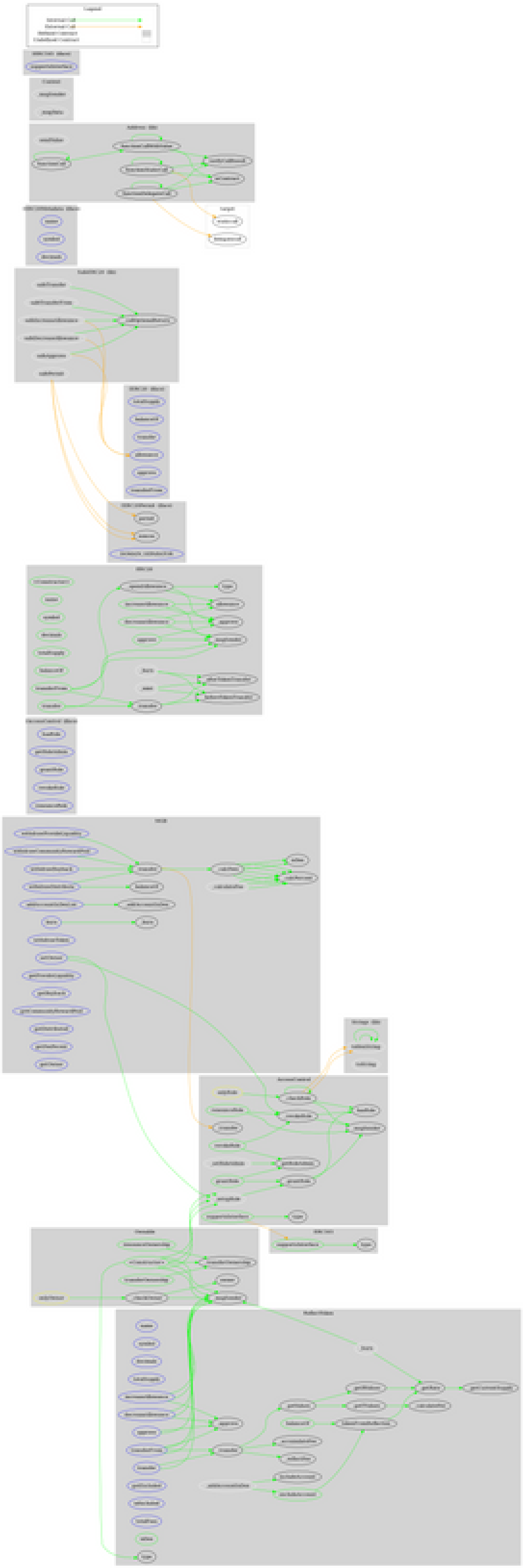
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-

	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	safePermit	Internal	✓	
	_callOptionalReturn	Private	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
MGB	Implementation	ReflectToken, AccessControl		
	<Constructor>	Public	✓	ReflectToken
	withdrawCommunityRewardPool	External	✓	onlyRole
	withdrawBuyback	External	✓	onlyRole
	withdrawProvideLiquidity	External	✓	onlyRole
	withdrawDistribute	External	✓	onlyRole
	addAccountInDexList	External	✓	onlyRole
	burn	External	✓	onlyRole
	withdrawToken	External	✓	onlyRole
	setOwner	External	✓	-
	getProvideLiquidity	External		-
	getBuyback	External		-
	getCommunityRewardPool	External		-
	getDistributed	External		-
	getFeePecent	External		-
	getOwner	External		-
	_transfer	Internal	✓	
	_calculateFee	Internal		
	_calcPercent	Internal		
	_calcFees	Internal	✓	

ReflectToken	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	_calculateFee	Internal		
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	getExcluded	External		-
	isExcluded	External		-
	totalFees	External		-
	tokenFromReflection	Public		-
	excludeAccount	Public	✓	onlyOwner
	includeAccount	Public	✓	onlyOwner
	_approve	Private	✓	
	_transfer	Internal	✓	
	_reflectFee	Private	✓	
	_burn	Internal	✓	
	_accumulateFee	Private	✓	
	_getValues	Private		
	isDex	Public		-
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_addAccountInDex	Internal	✓	

Contract Flow



Domain Info

Domain Name	magnummeta.com
Registry Domain ID	2658187410_DOMAIN_COM-VRSN
Creation Date	2021-11-29T06:24:46.00Z
Updated Date	2022-03-28T10:11:10.00Z
Registry Expiry Date	2023-11-29T06:24:46.00Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain was created 9 months before the creation of the audit. It will expire in over 1 year.

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 transferring tokens to the team’s wallet, manipulating fees and burning tokens. 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.

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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io