Cyberscope

Audit Report

PrivateSale

August 2022

SHA256 222d86edbbef91d39a37ef224b12331895458ba992c521c515232fa1abac4eee

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	PrivateSale
Compiler Version	v0.8.10+commit.fc410830
Testing Deploy	https://testnet.bscscan.com/token/0x6a2F07C1a952ceE878A6ea58a18a25487608614d
Domain	https://www.magnummeta.com

Audit Updates

Initial Audit	25th August 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	5af1771388b4fe634e0a566716e32c6d00a5372875099127b274d4cf8a94e9d2
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/security/Pausable.sol	2072248d2f79e661c149fd6a6593a8a3f038466557c9b75e50e0b001bcb5cf97
@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/Ad	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826

dress.sol	
@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/AbstractSaleRound.sol	a1165d30de9bab3c268763ef15152098d871acd85e153cd47b0354001c871daa
contracts/interfaces/IERC20Burn.sol	269d46bb4fcf77554fe054e673b49fbccaa1baf661fd3ab6e391aa4cb40cefa2
contracts/interfaces/ISaleRound.sol	25aae69be75186ce50ceea374539d6aeb4c5b8d3024dcc2b6e3c265ab21aed4d
contracts/interfaces/IUniswapV2Router02.sol	abe09b81ae0d88a2b8f1f79088a21c52eab8edbda3c8494241ccd3f93e659f51
contracts/PrivateSale.sol	222d86edbbef91d39a37ef224b12331895458ba992c521c515232fa1abac4eee
contracts/ReferralSystem.sol	9d2c1aaadf54d93959e646aae41eee229970ff4cb72996c7547c36229f97e367
contracts/Whitelist.sol	c114e0870ac00d35efc030784570924dd32b7bd08b0de00748a8124c2a951452

Introduction

The contract PrivateSale implements a mechanism for buying MGB tokens. During the buy process, the PrivateSale contract provides referral functionality with a reward system. The rewards are transferred directly to the referees on every buy. Users can buy MGB tokens providing either native or some predefined tokens.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	BLC	Business Logic Concern	Unresolved
●	RAV	Reentrancy Attack Vulnerability	Unresolved
●	MC	Missing Check	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

BLC - Business Logic Concern

Criticality	critical
Location	contract.sol:Whitelist#L23
Status	Unresolved

Description

The method `setWhiteList` is publicly available to any user. As a result, every user has the authority to mutate the whitelist array.

```
function setWhiteList(address[] memory account, bool[] memory status)
        external
    {
        uint256 len = account.length;
        if(len != status.length) {
            revert IncorrectArrayLength();
        }

        for (uint256 i; i < len; i++) {
            _whiteList[account[i]] = status[i];
        }
    }
```

Recommendation

The team should consider an access mechanism based on roles in order to avoid the arbitrary access of the whitelist.

RAV - Reentrancy Attack Vulnerability

Criticality	critical
Location	contract.sol#L42,74
Status	Unresolved

Description

The contract is vulnerable to reentrancy attack. The *buyMGB* method internally calls the *_distributeTheFee* method that internally calls the *payable(account).call{value: value}("");* method. If the user implements the receive call back, he will be able to execute the *buyMGB* again in the same execution thread.

```
function buyMGB(address referrer) external
        payable
        isWhiteList(msg.sender, referrer)
        isFinish
        whenNotPaused
    {
        .
        uint256 feeToReferrals = _distributeTheFee(msg.sender, amountMATIC, address(0));

function buyMGB( address usdAddr, uint256 usdAmount, address referrer )  external
        isWhiteList(msg.sender, referrer)
        isAvailableCurrency(usdAddr)
        isFinish
        whenNotPaused
    {
        .
        uint256 feeToReferrals = _distributeTheFee(msg.sender, usdAmount, usdAddr);

function _distributeTheFee( address referral, uint256 amount, address token)
       internal
       returns (
           uint256 feeToPeople
       )
    {
      .
     sendMATIC(newReferral, value);

function sendMATIC(address account, uint256 value) internal {
        payable(account).call{value: value}("");
```

```
}
```

Recommendation

The contract could embody a mutex pattern in order to avoid re-entrance issues.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L13
Status	Unresolved

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

```
_totalAmount = amount;
_percentDistributedImmediately = percentDistributedImmediately;
_MGBAddress = tokenAddr;
_vestingDuration = vesting;
_pricePerToken = pricePerToken;
_periodDuration = periodDuration * 1 days;
_tokenGenerationEvent = tokenGenerationEvent;
```

The contract should check if the `_maxContribution` is greater than `_minContribution`.

```
_maxContribution = contribuitionLimits[1];
_minContribution = contribuitionLimits[0];
```

The *percentReward* is used to distribute the fees to the referral addresses. The setter function could check if the *percentReward* array is summed to a specific threshold in order to avoid accidental huge distribution amounts.

Recommendation

The contract should properly check the variables according to the required specifications.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/AbstractSaleRound.sol#L40,34,22,35,21,31,23 contracts/ReferralSystem.sol#L14,17,15 contracts/Whitelist.sol#L8
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.

```
_stablecoin
_percentReward
_referralList
_receiveMATIC
_allReferralPercent
FACTOR
_receiveUSD
PRECISION
_MGBAddress
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contracts/ReferralSystem.sol#L55
Status	Unresolved

Description

Functions that are not used in the contract, and make the code's size bigger.

```
_setReferrer
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/AbstractSaleRound.sol#L340
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
month = (block.timestamp - _tokenGenerationEvent) / _periodDuration
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contracts/ReferralSystem.sol#L82,44 contracts/Whitelist.sol#L31 contracts/AbstractSaleRound.sol#L79
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
i
```

Recommendation

All the local scoped variables should be initialized.

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	✓	-
Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_requireNotPaused	Internal		

	_requirePaused	Internal		
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
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		

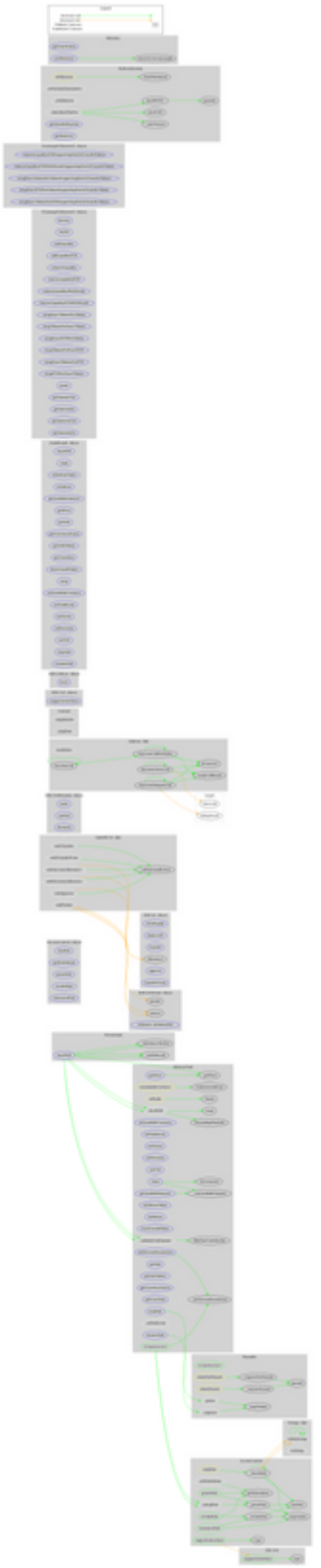
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
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		
AbstractSale	Implementation	ReferralSystem, Pausable, ISaleRound, AccessControl		
	<Constructor>	Public	✓	-
	setPercentParameters	External	✓	onlyRole
	setAvailableCurrency	External	✓	onlyRole
	setStablecoin	External	✓	onlyRole

	setFactor	External	✓	onlyRole
	setPrecision	External	✓	onlyRole
	setTGE	External	✓	onlyRole
	_buyMGB	Internal	✓	
	claim	External	✓	-
	withdrawToken	External	✓	onlyRole
	withdraw	External	✓	onlyRole
	burnUnsoldToken	External	✓	onlyRole whenPaused
	getAvailableAmount	External		-
	getPrice	External		-
	getInfo	External		-
	getInfoTokens	External		-
	getCurrencyStatus	External		-
	getUserData	External		-
	_validateUsdAmount	Internal		
	_setReferrals	Internal	✓	
	swap	Public		-
	stopSale	External	✓	onlyRole
	resumeSale	External	✓	onlyRole
	_calcAvailableAmount	Internal		
	_getPrice	Internal		
IERC20Burn	Interface			
	burn	External	✓	-
ISaleRound	Interface			
	buyMGB	External	Payable	-
	buyMGB	External	✓	-
	claim	External	✓	-
	withdrawToken	External	✓	-
	withdraw	External	✓	-
	getAvailableAmount	External		-
	getPrice	External		-
	getInfo	External		-

	getCurrencyStatus	External		-
	getInfoTokens	External		-
	getUserData	External		-
	burnUnsoldToken	External	✓	-
	swap	External		-
	setAvailableCurrency	External	✓	-
	setStablecoin	External	✓	-
	setFactor	External	✓	-
	setPrecision	External	✓	-
	setTGE	External	✓	-
	stopSale	External	✓	-
	resumeSale	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	✓	-
PrivateSale	Implementation	AbstractSale		
	<Constructor>	Public	✓	AbstractSale Pausable
	buyMGB	External	Payable	isWhiteList isFinish whenNotPaused
	buyMGB	External	✓	isWhiteList isAvailableCurrency isFinish whenNotPaused
ReferralSystem	Implementation	Whitelist		
	_setSystemParameters	Internal	✓	
	_setReferrer	Internal	✓	
	_distributeTheFee	Internal	✓	
	sendUSD	Internal	✓	
	sendMATIC	Internal	✓	
	_calcPercent	Internal		
	getDataRefSystem	External		-
	getReferrer	External		-
Whitelist	Implementation			
	getUserStatus	External		-
	setWhiteList	External	✓	-

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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The contract implements a buying mechanism with rewards for the referees. The contract is vulnerable for reentrance attack.

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