Audit Report

Bernard lottery

January 2022

Type	BEP20
Network	BSC
Address	0x189396f720a3C31E7650cC3eD066797B4D7fC0CF
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	Lottery
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/address/0x189396f720a3C31E7650cC3eD066797B4D7fC0CF
Source	contract.sol

Audit Updates

Initial Audit	30th January 2022
Corrected	

Contract Analysis

Contract Owner Drain Lottery Amount

Criticality	medium
Location	contract.sol#L332

Description

The users that buy lottery tickets are transferring tokens to the contract. These tokens can be drained by the contract owner anytime by calling the *adminWithdraw()* function. As a result the winners may not be able to claim their rewards.

```
// Withdraw without caring about rewards. EMERGENCY ONLY.
function adminWithdraw(uint256 _amount) public onlyAdmin {
    bern.safeTransfer(msg.sender, _amount);
    emit DevWithdraw(msg.sender, _amount);
}
```

Recommendation

There could be some limitation regarding the contract owner's access. For instance, the contract owner should be allowed to drain tokens only on legacy lotteries.

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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CR	Code Repetition
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic
	L06	Missing Events Access Control

CR - Code Repetition

Criticality	minor
Location	contract.sol#L1

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The functions *buy()* and *multiBuy()* share the same statements internally.

```
for (uint i = 0; i < 4; i++) {
    require (_numbers[i] <= maxNumber, 'exceed number scope');
}
uint256 tokenId = lotteryNFT.newLotteryItem(msg.sender, _numbers, _price,
issueIndex);
lotteryInfo[issueIndex].push(tokenId);
if (userInfo[msg.sender].length == 0) {
    totalAddresses = totalAddresses + 1;
}
userInfo[msg.sender].push(tokenId);
totalAmount = totalAmount.add(_price);
lastTimestamp = block.timestamp;
uint64[keyLengthForEachBuy] memory userNumberIndex =
generateNumberIndexKey(_numbers);
for (uint i = 0; i < keyLengthForEachBuy; i++) {

userBuyAmountSum[issueIndex][userNumberIndex[i]]=userBuyAmountSum[issueIndex][us
erNumberIndex[i]].add(_price);
}
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L2609,L2604,L2599 and 19 more

Description

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

```
adminWithdraw
setLotteryNFT
setAdmin
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L2249,L2625,L2620 and 30 more

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.

```
keyLengthForEachBuy
_allcation3
_allcation2
...
```

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
Location	contract.sol#L352,L362,L387 and 37 more

Description

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

```
trySub
tryMul
tryMod
...
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L2620,L2615

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.

```
maxNumber = _maxNumber
minPrice = _price
```

Recommendation

Emit an event for critical parameter changes.

L06 - Missing Events Access Control

Criticality	minor
Location	contract.sol#L2599

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.

```
adminAddress = _adminAddress
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC165	Interface			
	supportsInterface	External		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-

IERC721Receiver	Interface			
	onERC721Received	External	✓	-
ERC165	Implementation	IERC165		
	<Constructor>	Internal	✓	
	supportsInterface	Public		-
	_registerInterface	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		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	

	_verifyCallResult	Private		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
EnumerableMap	Library			
	_set	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_tryGet	Private		
	_get	Private		
	_get	Private		
	set	Internal	✓	

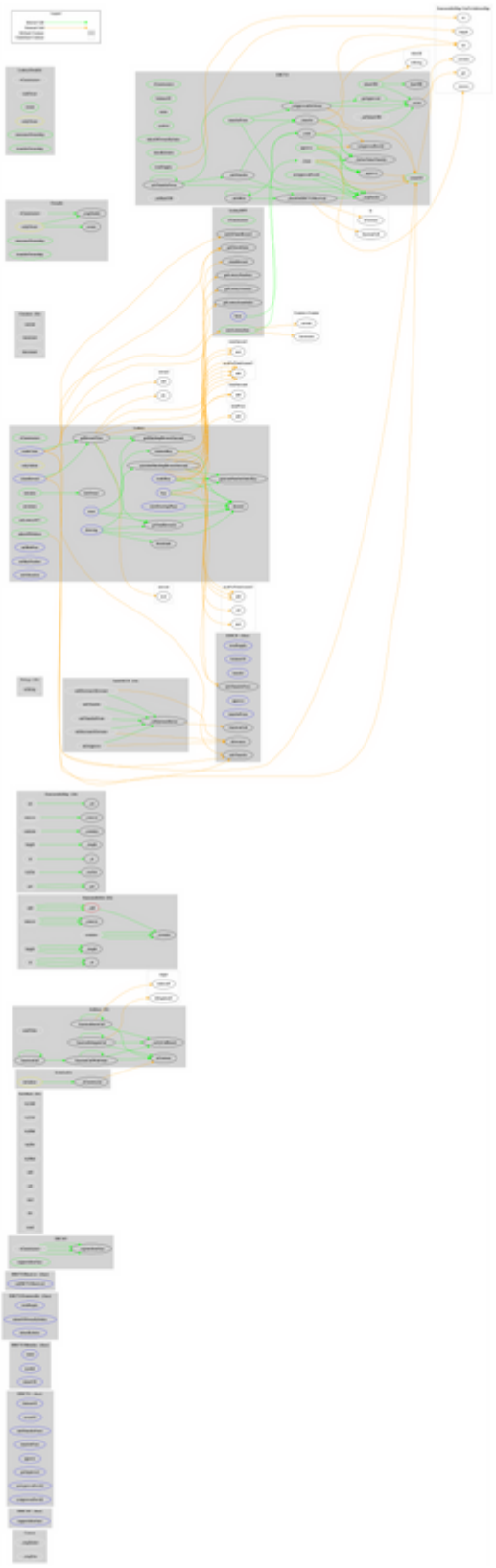
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	tryGet	Internal		
	get	Internal		
	get	Internal		
Strings	Library			
	toString	Internal		
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata, IERC721Enumerable		
	<Constructor>	Public	✓	-
	balanceOf	Public		-
	ownerOf	Public		-
	name	Public		-
	symbol	Public		-
	tokenURI	Public		-
	baseURI	Public		-
	tokenOfOwnerByIndex	Public		-
	totalSupply	Public		-
	tokenByIndex	Public		-
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_exists	Internal		

	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_transfer	Internal	✓	
	_setTokenURI	Internal	✓	
	_setBaseURI	Internal	✓	
	_checkOnERC721Received	Private	✓	
	_approve	Private	✓	
	_beforeTokenTransfer	Internal	✓	
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
LotteryNFT	Implementation	ERC721, Ownable		
	<Constructor>	Public	✓	ERC721
	newLotteryItem	Public	✓	onlyOwner
	getLotteryNumbers	External		-
	getLotteryAmount	External		-
	getLotteryIssueIndex	External		-
	claimReward	External	✓	onlyOwner
	multiClaimReward	External	✓	onlyOwner
	burn	External	✓	onlyOwner
	getClaimStatus	External		-
LotteryOwnabl	Implementation			

e				
	<Constructor>	Internal	✓	
	initOwner	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
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	✓	
	_callOptionalReturn	Private	✓	
Initializable	Implementation			
	_isConstructor	Private		
Lottery	Implementation	LotteryOwnable, Initializable		
	<Constructor>	Public	✓	-
	initialize	Public	✓	initializer
	drawed	Public		-
	reset	External	✓	onlyAdmin
	enterDrawingPhase	External	✓	onlyAdmin
	drawing	External	✓	onlyAdmin
	internalBuy	Internal	✓	

	buy	External	✓	-
	multiBuy	External	✓	-
	claimReward	External	✓	-
	multiClaim	External	✓	-
	generateNumberIndexKey	Public		-
	calculateMatchingRewardAmount	Internal		
	getMatchingRewardAmount	Public		-
	getTotalRewards	Public		-
	getRewardView	Public		-
	setAdmin	Public	✓	onlyOwner
	setLotteryNFT	Public	✓	onlyOwner
	adminWithdraw	Public	✓	onlyAdmin
	setMinPrice	External	✓	onlyAdmin
	setMaxNumber	External	✓	onlyAdmin
	setAllocation	External	✓	onlyAdmin

Contract Flow



Summary

Bernard lottery is a lottery application that is built in the BSC chain. The lottery is proceeding in rounds. Users are buying their lottery tickets. In the lottery ticket, the users choose the amount that they want to supply and the numbers that they are guessing. Then, the contract owner draws the random numbers. The users that guess correctly can claim their proportional rewards.

Even though the blockchain is a deterministic environment, the randomization function that is setting the winning numbers is well implemented. The algorithm combines fee costs, block information and an external number. This combination produces quite random numbers and it is very difficult to be manipulated.

There are some functions that can be abused by the contract owner. We state that the owner privileges are necessary and required for proper operation of the lottery application. Thus, we emphasise the contract owner to be extra careful with the credentials.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

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

The Coinscope.co team

https://www.coinscope.co