Cyberscope

Audit Report

NFTSport MysteryCube

November 2022

Gitlab https://gitlab.com/hola-tech1/worldcup-nft/nftsport-contracts

Commit 3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	MysteryCube
Gitlab	https://gitlab.com/hola-tech1/worldcup-nft/nftsport-contracts
Commit	3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b

Audit Updates

Initial Audit	13th November 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	dcebb99daefb7b6c2b5ddb1052f670cf9986240e5549da4ad47b5072857c620e
@openzeppelin/contracts/access/Ownable.sol	b9f957b42bdcf3d3499be4c94558152e91658e34a1fe5a5e8f0972ce20e15ed7
@openzeppelin/contracts/introspection/ERC165.sol	e6a3cba0775773bd92c8de6ac14d0614ca443ad63464a4e0241ca345940ea973
@openzeppelin/contracts/introspection/IERC165.sol	24d63fd063d0d9e954ce1a039404b4c01d2141f787143bbd3d5090a0220a2bcc
@openzeppelin/contracts/math/SafeMath.sol	4a04d0a20a19e3ef1dcabae9cad9ba006430a4e7eec4d9b519db87999722c98a
@openzeppelin/contracts/token/ERC1155/IERC1155.sol	321a0373e1d05812d82d966cc805eade59b1b0ab17be9455e2da699f11828d1f
@openzeppelin/contracts/token/ERC1155/IERC1155MetadataURI.sol	e46b551826c4497fe033f841b920a21a6be33543c0f9fedc3d2f9bcdee96ee14
@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol	8073b9327f6d52a357154d24923a80591f493dde019d98459d0389be4c3c8e60
@openzeppelin/contracts/token/ERC721/IERC721.sol	07abc5d9ae593f0dc7b854cb476fbee9e9f0df1c8f864e061f61e1532fb16357
@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol	da6fa0593fd96281d88df725727540d0c61551ed756a31a2ef6e1e8ccfbbe59d

@openzeppelin/contracts/utils/Address.sol	11ad5e3e21434e00c4ceba1f5a977b7a68bdd7d16b849276ce4ff4495129eec7
@openzeppelin/contracts/utils/Context.sol	9a3d1e5be0f0ace13e2d9aa1d0a1c3a6574983983ad5de94fc412f878bf7fe89
@openzeppelin/contracts/utils/EnumerableSet.sol	c8b73a000476872a00f6153d66be31a4a99b7565068f05336129748bfad704ea
@openzeppelin/contracts/utils/ReentrancyGuard.sol	3fc7968f4a1937caf3c96dffbac350398f86faad96288502e02c3a2b9f245e39
contracts/interfaces/INFTSport.sol	74cb5baaf50a6ed63c0dff5173c9fab90d6e1f9a61ddb59ca52300b675949efb
contracts/libraries/Strings.sol	88966c4fbc953d57f64b87264d245eef46864c08bfb9e9a1396c2f9f256d9558
contracts/libraries/TransferHelper.sol	bf61f5798d83a34255cdd18d52a3fd51ea3f8e3983dd9418050d0d80b997920e
contracts/libraries/Utils.sol	f2b5ae6f7eaa3ee650fb500e813080d9609bd486272895f0d6a7cd3d41a62259
contracts/nfts/ERC1155.sol	16530e7acad2db53be1808f5114b1416298cc719306ccce3834902bb1910b03d
contracts/nfts/MysteryCube.sol	a7353a6d7ba7a88453213ead9e4a295e0a8164e646ff4ce2291fdae1cd699ea8

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	RV	Randomization Vulnerability	Unresolved
●	TSI	Total Supply Inconsistency	Unresolved
●	PIC	Property Initialization Check	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

RV - Randomization Vulnerability

Criticality	minor / informative
Location	contract.sol#L141
Status	Unresolved

Description

The contract is using an on-chain technique in order to determine random numbers. The blockchain runtime environment is fully deterministic, as a result, the pseudo-random numbers could be predicted.

```
function _getRandomId(CubeInfo memory _cubeInfo) internal view returns (uint256)
{...}
```

Recommendation

The contract could use an advanced randomization technique that guarantees an acceptable randomization factor. For instance, the Chainlink VRF (Verifiable Random Function). https://docs.chain.link/docs/chainlink-vrf/

TSI - Total Supply Inconsistency

Criticality	minor / informative
Location	contract.sol
Status	Unresolved

Description

The contract keeps track of the total supply by updating it in the mintBatch method. The total supply may be changed from other methods as well, like transferFrom. If these methods are called by the users, then an inconsistency between the actual totalSupply will be produced.

```
function _mintBatch(
  address _to,
  uint256[] memory _ids,
  uint256[] memory _amounts,
  bytes memory data
) internal virtual override {
  super._mintBatch(_to, _ids, _amounts, data);
  for (uint256 i = 0; i < ids.length; i += 1) {
    tokenSupply[ids[i]] = amounts[ids[i]];
  }
}

function totalSupply(uint256 _id) public view returns (uint256) {
  return tokenSupply[_id];
}
```

Recommendation

The totalSupply should always be updated according to the actual totalSupply.

PIC - Property Initialization Check

Criticality	minor / informative
Location	contract.sol#L1
Status	Unresolved

Description

The contract contains methods that require the nftInfo property to be initialized. The contractor does not initialize this variable. If these methods are called prior to the nftInfo initialization, then they will produce unexpected results.

```
function open(uint256 _id) external whenOpenCubeEnabled nonReentrant {}
function _getRandomId(CubeInfo memory _cubeInfo) internal view returns (uint256)
{}
```

Recommendation

The contract should guarantee that the nftInfo in initialized for the methods that use the nftInfo property.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contracts/nfts/MysteryCube.sol#L59
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
NUMBER_OF_BOXES
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/nfts/MysteryCube.sol#L59,115,89,109,125,121,93
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.

```
NUMBER_OF_BOXES
_weights
_startTime
_tiers
_id
_i
_weight
```

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/libraries/TransferHelper.sol#L27,7,17
Status	Unresolved

Description

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

```
safeTransferFrom
safeApprove
safeTransfer
```

Recommendation

Remove unused functions.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contracts/nfts/MysteryCube.sol#L73
Status	Unresolved

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_uri
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AccessControl	Implementation	Context		
	hasRole	Public		-
	getRoleMemberCount	Public		-
	getRoleMember	Public		-
	getRoleAdmin	Public		-
	grantRole	Public	✓	-
	revokeRole	Public	✓	-
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Private	✓	
	_revokeRole	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
ERC165	Implementation	IERC165		
	<Constructor>	Internal	✓	
	supportsInterface	Public		-
	_registerInterface	Internal	✓	
IERC165	Interface			
	supportsInterface	External		-
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		
IERC1155	Interface	IERC165		
	balanceOf	External		-
	balanceOfBatch	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
	safeBatchTransferFrom	External	✓	-
IERC1155MetadataURI	Interface	IERC1155		
	uri	External		-
IERC1155Receiver	Interface	IERC165		
	onERC1155Received	External	✓	-
	onERC1155BatchReceived	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-

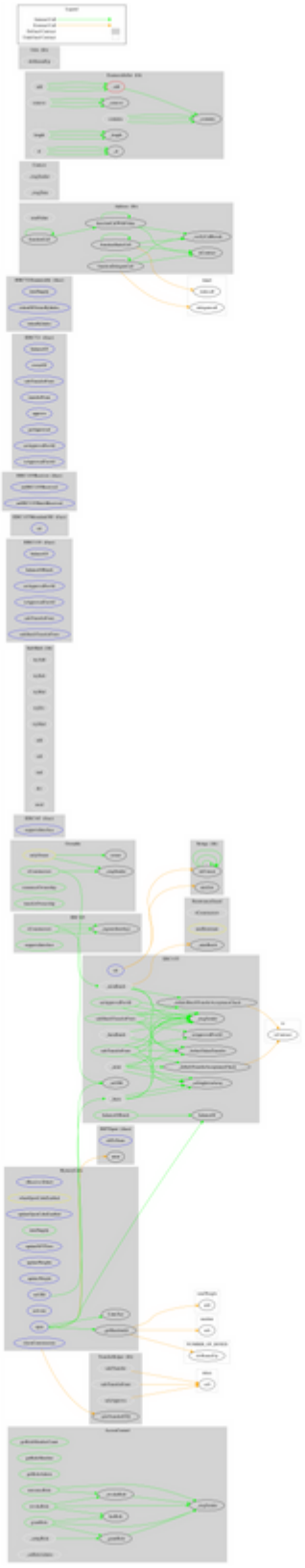
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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		
ReentrancyGuard	Implementation			
	<Constructor>	Internal	✓	
INFTSport	Interface	IERC721, IERC721Enumerable		
	nftToTeam	External		-
	mint	External	✓	-
Strings	Library			
	strConcat	Internal		
	strConcat	Internal		
	strConcat	Internal		
	strConcat	Internal		
	uint2str	Internal		
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	

	safeTransferETH	Internal	✓	
Utils	Library			
	divRoundUp	Internal		
ERC1155	Implementation	Context, ERC165, IERC1155, IERC1155MetadataURI		
	<Constructor>	Public	✓	-
	uri	External		-
	balanceOf	Public		-
	balanceOfBatch	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	safeTransferFrom	Public	✓	-
	safeBatchTransferFrom	Public	✓	-
	_setURI	Internal	✓	
	_mint	Internal	✓	
	_mintBatch	Internal	✓	
	_burn	Internal	✓	
	_burnBatch	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_doSafeTransferAcceptanceCheck	Private	✓	
	_doSafeBatchTransferAcceptanceCheck	Private	✓	
	_asSingletonArray	Private		
MysteryCube	Implementation	ERC1155, Ownable, AccessControl, ReentrancyGuard		
	<Constructor>	Public	✓	ERC1155
	<Receive Ether>	External	Payable	-
	updateOpenCubeEnabled	External	✓	onlyOwner
	totalSupply	Public		-

	_mintBatch	Internal	✓	
	updateNFTTiers	External	✓	onlyOwner
	updateWeights	External	✓	onlyOwner
	updateWeight	External	✓	onlyOwner
	open	External	✓	whenOpenCubeEnabled nonReentrant
	_getRandomId	Internal		
	activate	External	✓	-
	claimCommissions	External	✓	-
	setURI	External	✓	onlyOwner

Contract Flow



Summary

The MysteryCube contract implements an NFT mechanism with rewards in teams and a referral method. This audit investigates potential vulnerabilities, improvements, and business logic concerns.

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