



Cyberscope

Audit Report

NFTSport BetPools

November 2022

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

Commit [3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b](#)

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	3
Roles	5
Contract Diagnostics	6
MC - Missing Check	7
Description	7
Recommendation	7
IRD - Inconsistent Reward Distribution	8
Description	8
Recommendation	9
RSA - Rewards Sufficient Amount	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
Contract Functions	12
Contract Flow	16
Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	BetPools
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-upgradeable/access/AccessControlUpgradeable.sol	c5f290efee7b4156f4420d9f13b2caa990176495366b4edec4d8bc229ae1cb59
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	4d148e038344167b7506ee0efd58b38f8787c6229e43800fb1129a0d4215327f
@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol	31db49b5926d6f8eff496b0f4316a7f92f9b64274f7be94aa0b949f9d9c8112d
@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol	dabaab4d3d3f03e6bfb86eec1d54f31edf0429f4bfc4dff717d5776d5231c145
@openzeppelin/contracts-upgradeable/proxy/Initializable.sol	2c3a3edc2b1a4ac2c4a8645475b51f2668b1ad5ea22df074d0c0ebd3122ce2e7
@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	877bc9cb396d0f50330bb9c0057c029407e159739b6fab0b110f19451c8681e4
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	b9c1700bc8c28217952147b408dc67aa128eb2f71a45fceb4a8e73dff43fedac

@openzeppelin/contracts-upgradeable/utils/CountersUpgradeable.sol	5eaed54426f3286ef6ef62991c00c5c710833f12102b355dba2e8c3cda983ba4
@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol	634d70c2c44eda75e237be5a1f312c429475e8f3a0ab2b176aca3ae1a2d8f426
@openzeppelin/contracts-upgradeable/utils/ReentrancyGuardUpgradeable.sol	06e73664cf2eed972058697327c00d2595da4fe9a51398073bf8829e6307532a
contracts/bets/BetPools.sol	8ce6f7af9dc8bd545f0592f434a881b7ec64743ca4f2bb1945089f4b11380158
contracts/libraries/TransferHelper.sol	bf61f5798d83a34255cdd18d52a3fd51ea3f8e3983dd9418050d0d80b997920e

Roles

The contract has two Roles. The ADMIN_ROLE role and the UPDATER_ROLE.

ADMIN_ROLE has the authority to

- Create a betting pool with the corresponding bet options.
- Finalize a pool.
- Finalize a betting pool prior to the elapsed time.
- Give UPDATER_ROLE privilege.

UPDATER_ROLE has the authority to

- Create a betting pool with the corresponding bet options.
- Finalize a pool.

Users have the authority to

- Bet on any valid option with a minimum bet value of 0.1 ethers.
- Claim and view their refund balance.
- Claim and view their rewards.
- View transaction details.
- View the deposited balance on any betting option.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	MC	Missing Check	Unresolved
●	IRD	Inconsistent Reward Distribution	Unresolved
●	RSA	Rewards Sufficient Amount	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L165
Status	Unresolved

Description

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

```
function createPool(  
    string memory _title,  
    string memory _description,  
    string memory _thumbnail,  
    uint256 _startTime,  
    uint256 _endTime,  
    string[] memory _options  
) external {  
    //..  
    pool.startTime = _startTime;  
    pool.endTime = _endTime;  
    //..  
}
```

Recommendation

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

- The variable `_startTime` should be greater than the current timestamp.
- The variable `_endTime` should be greater than the `_startTime` variable.

IRD - Inconsistent Reward Distribution

Criticality	minor / informative
Location	contracts/bets/BetPools.sol
Status	Unresolved

Description

The contract distribute the redeem amounts during the finalization step. Users that placed a bet with the same amount in the same option may not receive the same award since the initial users are excluded from the deposit. We depict the **Case 1** where user A placed the bet initially and **Case 2** where user B placed the bet initially.

Case 1

User	Option	Bet
C	1	0.11
A	2	0.11
B	2	0.11

Finalization

User	Deposit	Refund
A	0	0.11
B	0.11	0

Case 2

User	Option	Bet
C	1	0.11

B	2	0.11
A	2	0.11

Finalization

User	Deposit	Refund
B	0	0.11
A	0.11	0

Recommendation

The redeem amount should be independent to the bet order.

RSA - Rewards Sufficient Amount

Criticality	minor / informative
Location	contract.sol#L275
Status	Unresolved

Description

The contract admin has the authority to create bets with unlimited amount of options. During the reward calculation phase the deposited amount is multiplied by the number of total options. If there are many options, the contract may not have sufficient balance to cover the redeem.

```
function pendingReward(uint256 _id, address _account) public view returns
(uint256) {
    if (getStatus(_id) != Status.Success || pools[_id].isClaimed[_account]) {
        return 0;
    }
    Pool memory pool = pools[_id];
    uint256 reward =
    pools[_id].options[pool.result].deposit[pool.result][_account].mul(pool.option
    Count);
    return reward;
}
```

Recommendation

The contract should guarantee during the finalization phase that the corresponding reward amount is sufficient to cover all the participants.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/bets/BetPools.sol#L167,159,160,275,257,143,189,166,169,168,144,264,146,284,251,158,134,246,171,88,170,145,102
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.

```
_description  
_optionId  
_account  
_id  
_title  
_startTime  
_thumbnail  
_end  
_options  
...
```

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 Upgradeable	Implementation	Initializable, ContextUpgradeable		
	__AccessControl_init	Internal	✓	initializer
	__AccessControl_init_unchained	Internal	✓	initializer
	hasRole	Public		-
	getRoleMemberCount	Public		-
	getRoleMember	Public		-
	getRoleAdmin	Public		-
	grantRole	Public	✓	-
	revokeRole	Public	✓	-
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Private	✓	
	_revokeRole	Private	✓	
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Ownable_init	Internal	✓	initializer
	__Ownable_init_unchained	Internal	✓	initializer
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
MathUpgradeable	Library			
	max	Internal		
	min	Internal		

	average	Internal		
SafeMathUpgradable	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		
Initializable	Implementation			
	_isConstructor	Private		
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	_verifyCallResult	Private		
ContextUpgradable	Implementation	Initializable		
	__Context_init	Internal	✓	initializer

	__Context_init_unchained	Internal	✓	initializer
	_msgSender	Internal		
	_msgData	Internal		
CountersUpgradeable	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
EnumerableSetUpgradeable	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		
ReentrancyGuardUpgradeable	Implementation	Initializable		

	__ReentrancyGuard_init	Internal	✓	initializer
	__ReentrancyGuard_init_unchained	Internal	✓	initializer
BetPools	Implementation	OwnableUp gradeable, AccessCont rolUpgrade able, Reentrancy GuardUpgra deable		
	initialize	External	✓	initializer
	getStatus	Public		-
	getPool	External		-
	getOptions	External		-
	getTransaction	External		-
	getDeposit	External		-
	createPool	External	✓	-
	bet	External	Payable	-
	_finalize	Internal	✓	nonReentrant
	finalize	External	✓	-
	forceFinalize	External	✓	-
	pendingRefund	Public		-
	claimRefund	External	✓	-
	pendingReward	Public		-
	claimReward	External	✓	-
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeTransferETH	Internal	✓	

Contract Flow



Summary

The BetPools contract implements a betting mechanism. This audit investigates potential vulnerabilities, improvements, and business logic concerns.

We state that admin and updater privileges are necessary and required for proper protocol operations. Thus, we emphasize the contract owner to be extra careful with the credentials.

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>



Cyberscope

Audit Report

NFTSport Lottery

November 2022

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

Commit [3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b](#)

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	3
Off Chain Winners Draw	5
Audit Context	5
Contract Diagnostics	6
CO - Code Optimization	7
Description	7
Recommendation	7
L09 - Dead Code Elimination	8
Description	8
Recommendation	8
Contract Functions	9
Contract Flow	14
Summary	15
Disclaimer	16
About Cyberscope	17

Contract Review

Contract Name	Lottery
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-upgradeable/access/OwnableUpgradeable.sol	4d148e038344167b7506ee0efd58b38f8787c6229e43800fb1129a0d4215327f
@openzeppelin/contracts-upgradeable/cryptography/MerkleProofUpgradeable.sol	38b7d647153f45309495a2713cd37536f6163ccac30717e3ecd9ac3e83200664
@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol	dabaab4d3d3f03e6bfb86eec1d54f31edf0429f4bfc4dff717d5776d5231c145
@openzeppelin/contracts-upgradeable/proxy/Initializable.sol	2c3a3edc2b1a4ac2c4a8645475b51f2668b1ad5ea22df074d0c0ebd3122ce2e7
@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol	3907de7006118eb83925d21fec1a435919366b545ef8fe6363f11cf9f6273501
@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	877bc9cb396d0f50330bb9c0057c029407e159739b6fab0b110f19451c8681e4
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	b9c1700bc8c28217952147b408dc67aa128eb2f71a45fceb4a8e73dff43fedac

@openzeppelin/contracts-upgradeable/Utils/CountersUpgradeable.sol	5eaed54426f3286ef6ef62991c00c5c710833f12102b355dba2e8c3cda983ba4
@openzeppelin/contracts-upgradeable/Utils/EnumerableSetUpgradeable.sol	634d70c2c44eda75e237be5a1f312c429475e8f3a0ab2b176aca3ae1a2d8f426
@openzeppelin/contracts-upgradeable/Utils/ReentrancyGuardUpgradeable.sol	06e73664cf2eed972058697327c00d2595da4fe9a51398073bf8829e6307532a
@openzeppelin/contracts/introspection/IERC165.sol	24d63fd063d0d9e954ce1a039404b4c01d2141f787143bbd3d5090a0220a2bcc
@openzeppelin/contracts/token/ERC721/IERC721.sol	07abc5d9ae593f0dc7b854cb476fbee9e9f0df1c8f864e061f61e1532fb16357
@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol	da6fa0593fd96281d88df725727540d0c61551ed756a31a2ef6e1e8ccfbbe59d
contracts/interfaces/INFTSport.sol	74cb5baaf50a6ed63c0dff5173c9fab90d6e1f9a61ddb59ca52300b675949efb
contracts/libraries/TransferHelper.sol	bf61f5798d83a34255cdd18d52a3fd51ea3f8e3983dd9418050d0d80b997920e
contracts/lottery/Lottery.sol	56a98e8bb8d6dcb9f490f456ada8e23cd916328d9362c1633aac2915b7087ca8

Off Chain Winners Draw

The winners draw mechanism is based on an off-chain logic. The contract owner is responsible for updating the in-chain “merkleRoot” in order to validate correctly amount of native tokens that the winners will claim. The verification algorithm is using the markle tree mechanism.

https://en.wikipedia.org/wiki/Merkle_tree

According to the markle algorithm, the off-chain mechanism pre-defines the amount of native tokens that each participant will claim per round.

Audit Context

This audit focuses on the in-chain functionality. The off-chain functionality is out of the audit context.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	CO	Code Optimization	Unresolved
●	L09	Dead Code Elimination	Unresolved

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The `unlockTime` and `timestamp` properties are assigned with the current timestamp. Both properties cannot be changed in the contract. As a result, they will always have the same value.

```
round.unlockTime = block.timestamp;  
round.result = _result;  
round.timestamp = block.timestamp;
```

Recommendation

The contract could remove the `timestamp` property since it always has the same value with `unlockTime`.

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.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
OwnableUpgradable	Implementation	Initializable, ContextUpgradable		
	__Ownable_init	Internal	✓	initializer
	__Ownable_init_unchained	Internal	✓	initializer
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
MerkleProofUpgradeable	Library			
	verify	Internal		
SafeMathUpgradeable	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		
Initializable	Implementation			

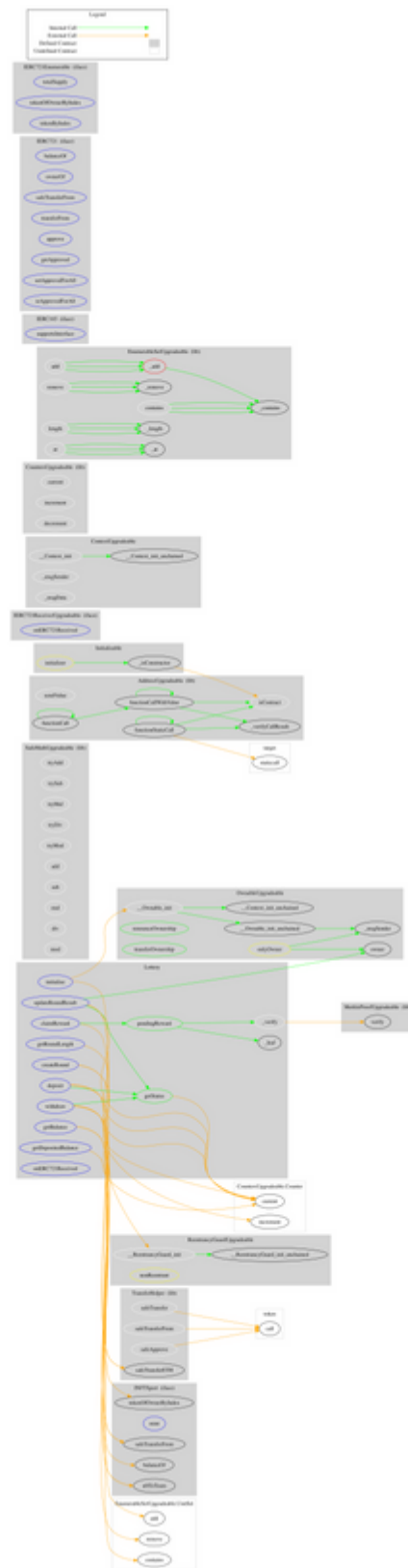
	_isConstructor	Private		
IERC721ReceiverUpgradeable	Interface			
	onERC721Received	External	✓	-
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	_verifyCallResult	Private		
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	initializer
	__Context_init_unchained	Internal	✓	initializer
	_msgSender	Internal		
	_msgData	Internal		
CountersUpgradeable	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
EnumerableSetUpgradeable	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		
ReentrancyGuardUpgradeable	Implementation	Initializable		
	__ReentrancyGuard_init	Internal	✓	initializer
	__ReentrancyGuard_init_unchained	Internal	✓	initializer
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	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
INFTSport	Interface	IERC721, IERC721Enumerable		
	nftToTeam	External		-
	mint	External	✓	-
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeTransferETH	Internal	✓	
Lottery	Implementation	IERC721ReceiverUpgradeable, OwnableUpgradeable, ReentrancyGuardUpgradeable		
	initialize	External	✓	initializer
	getStatus	Public		-
	getRoundLength	External		-
	createRound	External	Payable	onlyOwner
	updateRoundResult	External	✓	onlyOwner
	getBalance	External		-
	getDepositedBalance	External		-
	deposit	External	✓	-
	withdraw	External	✓	-
	_leaf	Internal		
	_verify	Internal		

	pendingReward	Public		-
	claimReward	External	✓	-
	onERC721Received	External	✓	-

Contract Flow



Summary

The Lottery contract implements a lottery mechanism based on NFTs contribution. 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>



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

Table of Contents	1
Contract Review	3
Audit Updates	3
Source Files	4
Contract Diagnostics	6
RV - Randomization Vulnerability	7
Description	7
Recommendation	7
TSI - Total Supply Inconsistency	8
Description	8
Recommendation	8
PIC - Property Initialization Check	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12
Recommendation	12
L15 - Local Scope Variable Shadowing	13
Description	13
Recommendation	13

Contract Functions	14
Contract Flow	20
Summary	21
Disclaimer	22
About Cyberscope	23

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` is 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

There 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	✓	-
IERC1155Meta dataURI	Interface	IERC1155		
	uri	External		-
IERC1155Rece iver	Interface	IERC165		
	onERC1155Received	External	✓	-
	onERC1155BatchReceived	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-

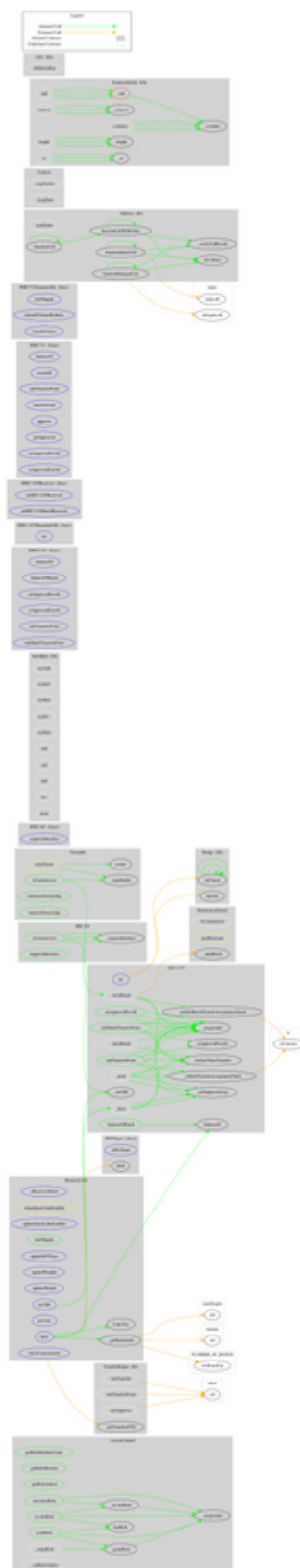
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	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, IERC1155M etadadataURI		
	<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>



Cyberscope

Audit Report

NFTSport

November 2022

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

Commit [3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b](#)

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	3
Contract Diagnostics	5
CO - Code Optimization	6
Description	6
Recommendation	6
INNM - Initial NFT Not Minted	7
Description	7
Recommendation	7
MT - Mints Tokens	8
Description	8
Recommendation	8
L15 - Local Scope Variable Shadowing	9
Description	9
Recommendation	9
Contract Functions	10
Contract Flow	16
Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	NFTSport
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/ERC721/ERC721.sol	1830e24292b045f3d44b14645cfce12d7652566d59f162186e42f668f0a024e0
@openzeppelin/contracts/token/ERC721/IERC721.sol	07abc5d9ae593f0dc7b854cb476fbee9e9f0df1c8f864e061f61e1532fb16357
@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol	da6fa0593fd96281d88df725727540d0c61551ed756a31a2ef6e1e8ccfbbe59d
@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol	17a75a430e00aa592ec076cecb7c1fee37b4b21c10cec9b84f57faac13fb3cb5
@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol	7e3d89b564e70918bc4e71e8346271f90dc3359d65b542baf24ce4de4e73d0a8

@openzeppelin/contracts/utils/Address.sol	11ad5e3e21434e00c4ceba1f5a977b7a68bdd7d16b849276ce4ff4495129eec7
@openzeppelin/contracts/utils/Context.sol	9a3d1e5be0f0ace13e2d9aa1d0a1c3a6574983983ad5de94fc412f878bf7fe89
@openzeppelin/contracts/utils/Counters.sol	8d78ab699ba115f01c6e1a213e6a696d49c7ed26c4f49f23c2b80cb2895e8853
@openzeppelin/contracts/utils/EnumerableMap.sol	26c7ec2df617e9420a3782d911dc6c339e83b02eac442de4c3c4bbbd18fe3273
@openzeppelin/contracts/utils/EnumerableSet.sol	c8b73a000476872a00f6153d66be31a4a99b7565068f05336129748bfad704ea
@openzeppelin/contracts/utils/Strings.sol	c3c3a9561de5e096929024e8a5476d6982dfa5c85065624fa94c358848c5285d
contracts/interfaces/INFTSport.sol	74cb5baaf50a6ed63c0dff5173c9fab90d6e1f9a61ddb59ca52300b675949efb
contracts/nfts/NFTSport.sol	3cce497d9abcc57af2e4ff9b1f8b059a41f29c2a421d20a20040d8c5b7ff34b7

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	CO	Code Optimization	Unresolved
●	INNM	Initial NFT Not Minted	Unresolved
●	MT	Mints Tokens	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L17
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The number of teams' maximum value is 32. A uint256 is used as a declaration type.

```
uint256 public constant NUMBER_OF_TEAMS = 32;  
function mint(address account, uint256 teamId);
```

Recommendation

The contract could use a uint8 type in order to handle the number of teams' maximum value and teamId.

INNM - Initial NFT Not Minted

Criticality	minor / informative
Location	contract.sol#L24
Status	Unresolved

Description

In the constructor of the contract, the `_tokenIds` counter is increased. As a result, the counter start counting from the number 1, and the number 0 is ignored.

```
constructor() public ERC721("NFT Sport", "NFTSport") {  
    _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);  
    _tokenIds.increment();  
}
```

Recommendation

The contract should mint the initial NFT to the number zero.

MT - Mints Tokens

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

Description

The role MINTER has the authority to mint tokens. The MINTER role may take advantage of it by calling the `mint` function.

```
function mint(address account, uint256 teamId) external override returns (uint256) {  
    require(hasRole(MINTER_ROLE, _msgSender()), "mint: only MINTER_ROLE");  
    require(teamId < NUMBER_OF_TEAMS, "mint: invalid teamId");  
    uint256 tokenId = _tokenIds.current();  
    _safeMint(account, tokenId);  
    _tokenIds.increment();  
    nftToTeam[tokenId] = teamId;  
    return tokenId;  
}
```

Recommendation

The MINTER role and contract 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.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contracts/nfts/NFTSport.sol#L27
Status	Unresolved

Description

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

```
baseURI
```

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		
ERC721	Implementation	Context, ERC165, IERC721, IERC721Me tadata, IERC721En umerable		
	<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	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
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		-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-

IERC721Receiver	Interface			
	onERC721Received	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		
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	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		
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		
Strings	Library			
	toString	Internal		

INFTSport	Interface	IERC721, IERC721Enumerable		
	nftToTeam	External		-
	mint	External	✓	-
NFTSport	Implementation	INFTSport, ERC721, Ownable, AccessControl		
	<Constructor>	Public	✓	ERC721
	setBaseURI	External	✓	onlyOwner
	mint	External	✓	-

Contract Flow



Summary

The NFTSport contract implements a pure NFT mechanism. 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>



Cyberscope

Audit Report

NFTSport

November 2022

Gitlab <https://gitlab.com/hola-tech1/worldcup-nft/nftsport-contracts>
Commit 3735ccf93cd73bcbb8f4857db4c215bf4f4ac09b
Audited by © cyberscope

Table of Contents

Table of Contents	1
Introduction	2
Corresponding Audits	3
NFTSport Audit	3
MysteryCube Audit	3
Lottery Audit	3
BetPools Audit	3
Disclaimer	4
About Cyberscope	5

Introduction

Cyberscope audited four sections of the NFTSport Project ecosystem. It consists of contacts NFTSport, MysteryCube, Lottery and Betpools. Each section of the NFTSport Project ecosystem consists of the corresponding independent audit.

Corresponding Audits

NFTSport Audit

<https://github.com/cyberscope-io/audits/blob/main/1-nfts/NFTSport.pdf>

MysteryCube Audit

<https://github.com/cyberscope-io/audits/blob/main/1-nfts/MysteryCube.pdf>

Lottery Audit

<https://github.com/cyberscope-io/audits/blob/main/1-nfts/Lottery.pdf>

BetPools Audit

<https://github.com/cyberscope-io/audits/blob/main/1-nfts/BetPools.pdf>

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>