Cyberscope

Audit Report

NFTSport

November 2022

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

Table of Contents

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

The 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, 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	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