Cyberscope

Audit Report

SoccerNStake

November 2022

SHA256 4882e071b27bdcf5abe3f907286af5310051623311145e2585ca42050cd95e299

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	SoccerNStake
Compiler Version	v0.8.14+commit.80d49f37
Explorer	https://testnet.bscscan.com/token/0x795aCc04bAFB25B62f6F897b28b1a34E80A7AB22
Domain	https://soccern.xyz

Audit Updates

Initial Audit	18th November 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	5af1771388b4fe634e0a566716e32c6d00a5372875099127b274d4cf8a94e9d2
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC721/IERC721.sol	fde830ac73ef320f7e3ce977b8cf567173f1e479ba86d584498f8362a67a5dc0
@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol	77f0f7340c2da6bb9edbc90ab6e7d3eb8e2ae18194791b827a3e8c0b11a09b43
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a

@openzeppelin/contracts/utils/Strings.sol	34127ad0054df5963b0fd694c1b313d17e9114a2f426b85526d6d976210298ab
@openzeppelin/contracts/utils/structs/EnumerableSet.sol	778d5305652c4eb562b12880cb6cf023d67df24844c15783a0b80fac2e715585
contracts/SoccerNLib.sol	a3d9f32a8e3f63e2302af870a910c96b3f87896f5213b0eda20c72b884942aff
contracts/SoccerNStake.sol	4882e071b27bdcf5abe3f907286af531005162331145e2585ca42050cd95e299

Introduction

The SoccerNStake contract implements an NFT staking mechanism. The contract has two tiers. The standard and the premium tier. The premium tier triples the APY of the staking pool. The tier depends on the holder's NFT passcard.

Staking Funds

The contract owner is responsible for providing funds to cover the staking rewards. The contract does not keep reserves in relation to the pools and the stacked amount. Hence, if the contract owner does not manually transfer the reward funds to the staking contract, the users will not be able to claim their amount.

Roles

The contract has a stake admin and an owner role.

Owner

The owner role has the authority to

- Give Stake Admin privileges.
- Change staking pool status.

Stake Admin

The owner role has the authority to create staking pools.

Users

The users have the authority to

- Stake in a stake pool.
- Claim stake rewards.
- View active stake pools.
- View staking pools.
- Get staking pool NFT reward amount.
- Get PassCardRewards.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	CO	Code Optimization	Unresolved
●	MC	Missing Check	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L223
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
erc20Token.transfer(_stake.staker, _stake.rewardAmount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L113,152
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 `remove()` and `add()` methods of the Set library will remove or add an item only if it exists/not exist. Thus, an additional check is redundant.

```
if (status == Pool_Status_Closed) {
    if (_activePoolIds.contains(poolId)) {
        _activePoolIds.remove(poolId);
    }
} else {
    if (!_activePoolIds.contains(poolId)) {
        _activePoolIds.add(poolId);
    }
}
```

The reward calculation is based on the variables `baseTokenAmount`, `apy` and `REWARD_RATE` that are statically defined and never changed. Thus, the value `baseTokenAmount * apy / REWARD_RATE` could be calculated once during the initialization. Additionally the `baseTokenAmount` and `apy` could be removed.

```
uint256 _rewardAmount =
_pool.config.baseTokenAmount.mul(_pool.config.apy).div(REWARD_RATE).mul(apyTim
es);
```

Recommendation

The contract could directly call the `remove()` and `add()` methods since the Set structure implements the existence check internally.

The construct could calculate the `baseTokenAmount * apy / REWARD_RATE` once.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L84,130
Status	Unresolved

Description

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

```
    constructor(IERC20 _erc20Token, ISoccerNFT _nft) Ownable() {
        address _owner = msg.sender;
        _setupRole(DEFAULT_ADMIN_ROLE, _owner);
        _setupRole(STAKE_ADMIN_ROLE, _owner);

        erc20Token = _erc20Token;
        nft = _nft;
    }

    function _createPool(PoolConfig calldata _config) private {
        lastPoolId++;
        Pool memory _pool;
        _pool.status = Pool_Status_Opened;
        _pool.id = lastPoolId;
        _pool.config = _config;
        _pool.rewardsRemain = _config.rewardsSupply;

        pools[lastPoolId] = _pool;

        _activePoolIds.add(_pool.id);

        emit PoolCreated(lastPoolId, _config.rewardsSupply, _config.duration,
_config.stakeOpenAt, _config.stakeCloseAt);
    }
...
    function setPoolStatus(uint256 poolId, uint8 status) external onlyOwner {
        pools[poolId].status = status;
        ...
    }
```

Recommendation

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

- _erc20Token and _nft should not be set to zero address.
- _config.apy should be greater than zero.
- _config.duration should be greater than zero.
- _config.stakeOpenAt should be greater than the current.timestamp.
- _config.stakeCloseAt should be greater than the stakeOpenAt.
- _config.baseTokenAmount should be greater than zero.
- _config.rewardsSupply should be greater than zero.
- pools[poolId].status should be either Pool_Status_Opened or Pool_Status_Closed.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/SoccerNLib.sol#L5,6 contracts/SoccerNStake.sol#L21,22,124
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.

```
PassCard_Standard
Pool_Status_Opened
PassCard_Premium
Pool_Status_Closed
_configs
```

Recommendation

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

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/SoccerNStake.sol#L150
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
_rewardAmount =
_pool.config.baseTokenAmount.mul(_pool.config.apy).div(REWARD_RATE).mul(apyTimes)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contracts/SoccerNStake.sol#L183,132
Status	Unresolved

Description

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

```
_stake
_pool
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contracts/SoccerNStake.sol#L85
Status	Unresolved

Description

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

```
_owner
```

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, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner

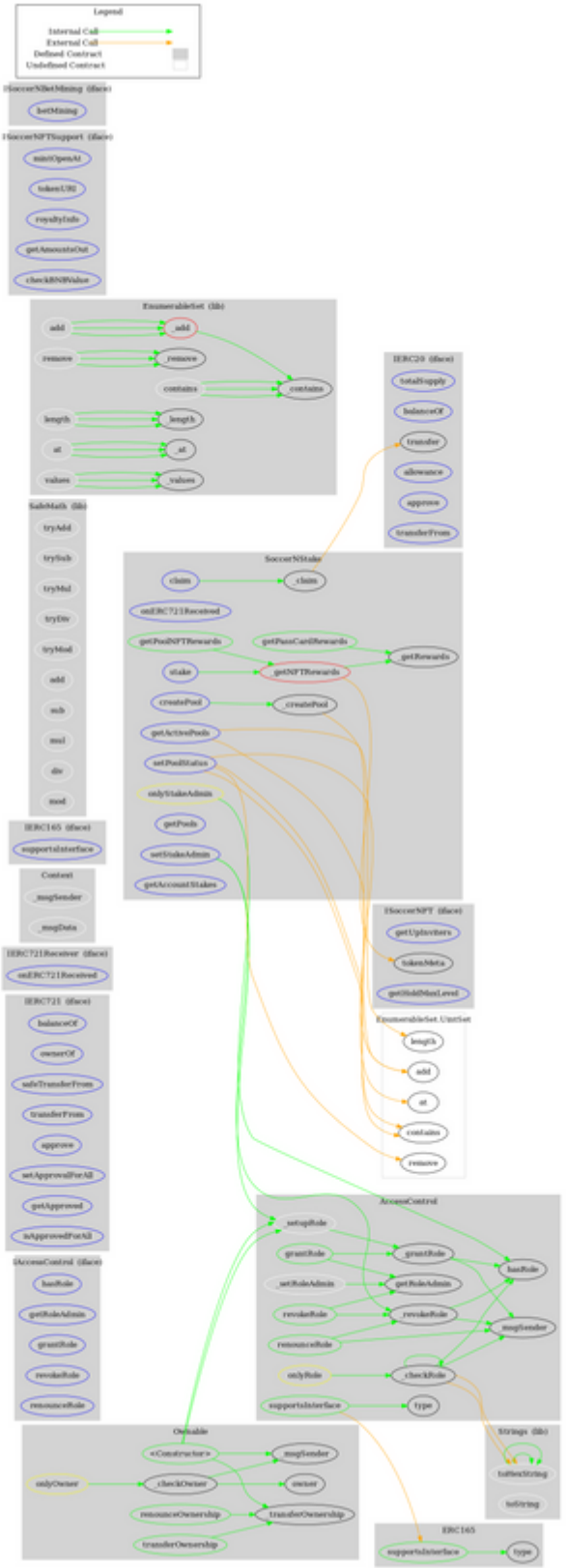
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	setApprovalForAll	External	✓	-
	getApproved	External		-
	isApprovedForAll	External		-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
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		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_values	Private		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		

	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
SoccerNLib	Library			
ISoccerNFTSupport	Interface			
	mintOpenAt	External		-
	tokenURI	External		-
	royaltyInfo	External		-
	getAmountsOut	External		-
	checkBNBValue	External		-
ISoccerNFT	Interface			
	getUpInviters	External		-
	tokenMeta	External		-
	getHoldMaxLevel	External		-
ISoccerNBetMining	Interface			
	betMining	External	✓	-
SoccerNStake	Implementation	Ownable, AccessControl, IERC721Re		

		ceiver		
	<Constructor>	Public	✓	Ownable
	onERC721Received	External		-
	setStakeAdmin	External	✓	onlyOwner
	setPoolStatus	External	✓	onlyOwner
	createPool	External	✓	onlyStakeAdmin
	_createPool	Private	✓	
	getPassCardRewards	Public		-
	_getRewards	Internal		
	_getNFTRewards	Private		
	getPoolNFTRewards	Public		-
	stake	External	✓	-
	claim	External	✓	-
	_claim	Private	✓	
	getPools	External		-
	getActivePools	External		-
	getAccountStakes	External		-

Contract Flow



Domain Info

Domain Name	soccern.xyz
Registry Domain ID	D333457734-CNIC
Creation Date	2022-11-15T06:18:12.0Z
Updated Date	2022-11-15T06:53:42.0Z
Registry Expiry Date	2023-11-15T23:59:59.0Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com/
Registrar	Go Daddy, LLC
Registrar IANA ID	146

The domain was created 3 days before the creation of the audit. It will expire in 12 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The SoccerNStake contract operates as an NFT staking contract. The Smart Contract analysis reported no compiler errors or critical issues. This audit focused on investigating possible security issues and potential improvements.

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