Cyberscope

Audit Report
Web3Punks

December 2023

Network	BSC
Address	0xC97C6e167D8c1E1dC41BE7eA75E1B07108080Ecf
Audited by	© cyberscope

Table of Contents

Table of Contents **1**
Review **2**
Audit Updates 2
Source Files 2
Overview **4**
Findings Breakdown **5**
Diagnostics **6**
CCR - Contract Centralization Risks 7
Description 7
Recommendation 7
CO - Code Optimization 9
Description 9
Recommendation 9
Functions Analysis **10**
Inheritance Graph **17**
Flow Graph **18**
Summary **19**
Disclaimer **20**
About Cyberscope **21**

Review

Testing Deploy	https://testnet.bscscan.com/address/0xc97c6e167d8c1e1dc41be7ea75e1b07108080ecf

Audit Updates

Initial Audit	12 Dec 2023
Corrected Phase 2	19 Dec 2023

Source Files

Filename	SHA256
contracts/W3PContractResolved.sol	ea6ca8a9f8f5c483ea1cc1cdc1fc793ae2ceb0c18ed657c6eb213df2ffb0e84b
@openzeppelin/contracts/utils/Strings.sol	cb2df477077a5963ab50a52768cb74ec6f32177177a78611ddbbe2c07e2d36de
@openzeppelin/contracts/utils/Context.sol	b2cfee351bcafd0f8f27c72d76c054df9b571b62cfac4781ed12c86354e2a56c
@openzeppelin/contracts/utils/Address.sol	8b85a2463eda119c2f42c34fa3d942b61aee65df381f48ed436fe8edb3a7d602
@openzeppelin/contracts/utils/math/SignedMath.sol	420a5a5d8d94611a04b39d6cf5f02492552ed4257ea82aba3c765b1ad52f77f6
@openzeppelin/contracts/utils/math/Math.sol	85a2caf3bd06579fb55236398c1321e15fd524a8fe140dff748c0f73d7a52345
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5

@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol	77f0f7340c2da6bb9edbc90ab6e7d3eb8e2ae18194791b827a3e8c0b11a09b43
@openzeppelin/contracts/token/ERC721/IERC721.sol	c8d867eda0fd764890040a3644f5ccf5db92f852779879f321ab3ad8b799bf97
@openzeppelin/contracts/token/ERC721/ERC721.sol	7af3ff063370acb5e1f1a2aab125ceca457cd1fa60ff8afa37aabc366349d286
@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol	f16b861aa1f623ccc5e173f1a82d8cf45b678a7fb81e05478fd17eb2ccb7b37e
@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol	7bf559fad1068a1329517b56b1ecddefa67e79a03bb0801b9e6bf06bf73eb334
@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol	e04aa070ad6f111fae49b96a056671f36307a93dd79b27612e72560e4a9749b2
@openzeppelin/contracts/security/Pausable.sol	2072248d2f79e661c149fd6a6593a8a3f038466557c9b75e50e0b001bcb5cf97
@openzeppelin/contracts/interfaces/IERC721.sol	e3bcee0ce85a310031fcef279f963e73c12c676a66c5c562ab3945ccf10aecff
@openzeppelin/contracts/interfaces/IERC4906.sol	6b572852b6d6e1db371287a0eb443a724e9005e025025b9c82ebc8804433c0ff
@openzeppelin/contracts/interfaces/IERC165.sol	410e40cd79f1b82bb6bbab95fa4279252cae6e3962b0bff46ab4855f6de91d35

Overview

This document provides the overview of the smart contract audit conducted for the "Web3Punks" contract. This contract is designed for minting NFTs with various attributes and dynamic pricing mechanisms. It utilizes ERC721 standards and leverages OpenZeppelin libraries for enhanced security and functionality. The contract owner has the authority to pause/unpause the mint of NFTs, change price models, and change critical parameters, which pose several centralization risks that warrant attention.

Functionality

Mint

Users can mint NFTs by providing a token ID, URI, and attributes. Minting is subject to the contract not being paused and adheres to max supply limits.

Dynamic Pricing

The contract incorporates a dynamic pricing mechanism based on the token ID and attributes. It includes different base prices for various ranges of token IDs and attribute counts.

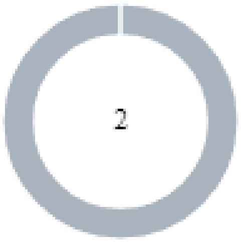
Mint Limit Enforcement

Implements a mint limit logic based on the token ID threshold, ensuring controlled minting activity.

Mint Limit Individually

Implements a mint limit logic for each user individually, where they can either mint 1 or 7 NFTs, based on how many total NFTs have been already minted.

Findings Breakdown



●	Critical	0
●	Medium	0
●	Minor / Informative	2

Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	0	0

Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor / Informative

Severity	Code	Description	Status
⚪	CCR	Contract Centralization Risks	Unresolved
⚪	CO	Code Optimization	Unresolved

CCR - Contract Centralization Risks

Criticality	Minor / Informative
Location	contracts/W3PContractResolved.sol#L119,208,229,244,263,272
Status	Unresolved

Description

The contract owner has the authority to pause/unpause the mint of NFTs, change price models, and change critical parameters like max supply. While this configuration can offer flexibility, it also poses several centralization risks that warrant attention. Centralization risks arising from the dependence on this type of configuration include Single Point of Control, Vulnerability to Attacks, Operational Delays, Trust Dependencies, and Decentralization Erosion.

```
function pause() public onlyOwner {
    _pause();
}
function updateBasePrice1e3(uint256 basePrice1k) external
onlyOwner {
    require(basePrice1k != basePrice1e3, "Previous price
provided");
    emit BasePrice1e3Updated(msg.sender, basePrice1e3,
basePrice1k);
    basePrice1e3 = basePrice1k;
}
function updateMaxSupply(uint256 newMaxSupply) external
onlyOwner {
    require(newMaxSupply != maxSupply, "Previous max supply
provided");
    emit MaxSupplyUpdated(msg.sender, maxSupply, newMaxSupply);
    maxSupply = newMaxSupply;
}
```

Recommendation

To mitigate these centralization risks, consider the following strategies:

- Implement a governance mechanism that allows NFT holders to vote on critical decisions.

- Transition control from a single owner to a multi-signature wallet.
- Implement time locks for critical functions.

CO - Code Optimization

Criticality	Minor / Informative
Location	contracts/W3PContractResolved.sol#L149
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. Specifically, the current way that the pricing and mint limit logic can be optimized, by integrating the mint limit logic inside the pricing conditions

```
// Pricing logic
if (tokenId < MINT_THRESHOLD) {
    if (tokenId < (MINT_THRESHOLD / 2)) {
        price = basePrice1e3;
    } else {
        price = BASE_PRICE_2E3;
    }
} else {
    price = calculatePrice(attributes);
}

// Mint limit logic
(tokenId < MINT_THRESHOLD) ? limit = 1 : limit = 7;
```

Recommendation

The team is advised to take these segments into consideration and rewrite them so the runtime will be more performant. That way it will improve the efficiency and performance of the source code and reduce the cost of executing it.

Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
W3PContractResolved	Implementation	ERC721URIStorage, Pausable		
		Public	✓	ERC721
	supportsInterface	Public		-
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	safeMint	Public	Payable	whenNotPaused
	calculatePrice	Internal		
	updateBasePrice1e3	External	✓	onlyOwner
	getBasePrice1e3	Public		-
	getBasePrice2e3	Public		-
	updateBasePriceAttributes	External	✓	onlyOwner
	getBasePriceAttributes	Public		-
	updateBasePriceZeroAttributes	External	✓	onlyOwner
	getBasePriceZeroAttributes	Public		-
	updateMaxSupply	External	✓	onlyOwner
	updateMintAmountReceiver	External	✓	onlyOwner
	getMintAmountReceiver	Public		-
	updateOwner	External	✓	onlyOwner

		External	Payable	-
		External	Payable	-
Strings	Library			
	toString	Internal		
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	equal	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
	_contextSuffixLength	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResultFromTarget	Internal		
	verifyCallResult	Internal		
	_revert	Private		
SignedMath	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	abs	Internal		
Math	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		
	mulDiv	Internal		
	sqrt	Internal		
	sqrt	Internal		

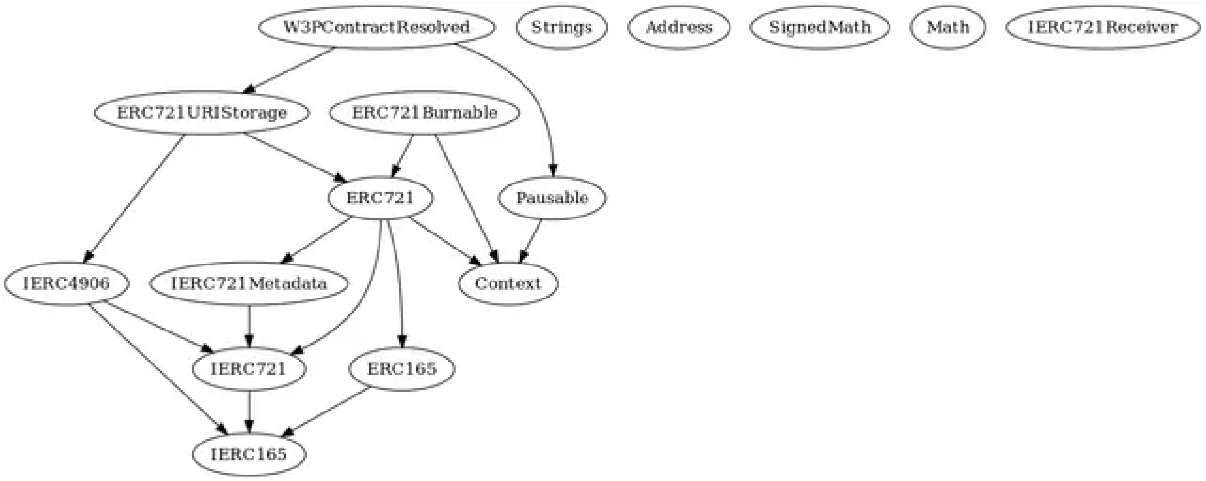
	log2	Internal		
	log2	Internal		
	log10	Internal		
	log10	Internal		
	log256	Internal		
	log256	Internal		
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-

	setApprovalForAll	External	✓	-
	getApproved	External		-
	isApprovedForAll	External		-
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata		
		Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-
	name	Public		-
	symbol	Public		-
	tokenURI	Public		-
	_baseURI	Internal		
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_ownerOf	Internal		

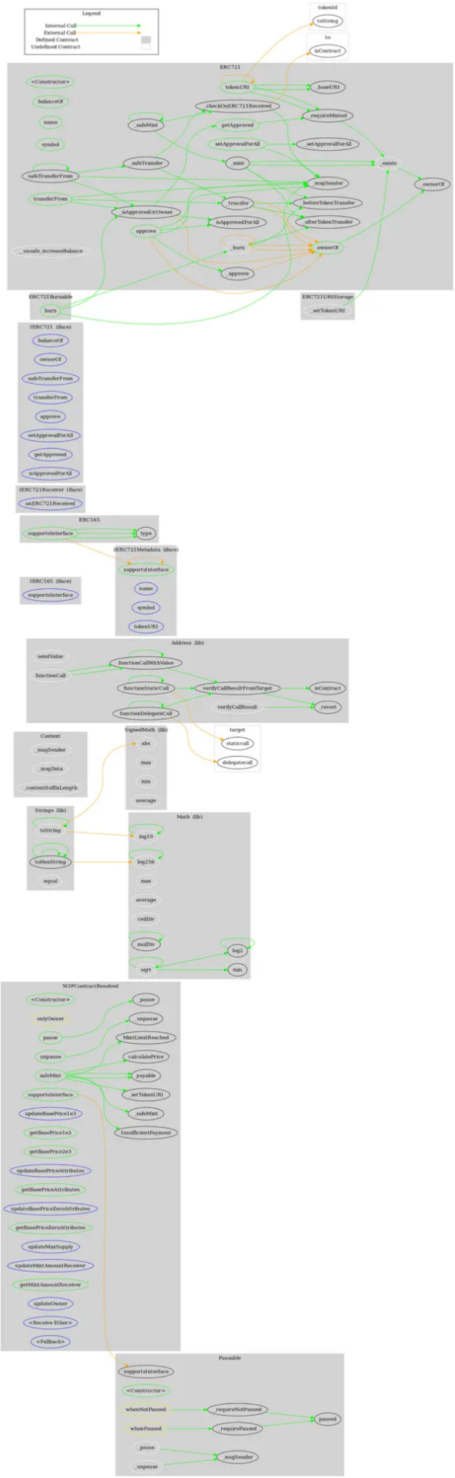
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_setApprovalForAll	Internal	✓	
	_requireMinted	Internal		
	_checkOnERC721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
	__unsafe_increaseBalance	Internal	✓	
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-
ERC721URIStorage	Implementation	IERC4906, ERC721		
	supportsInterface	Public		-
	tokenURI	Public		-

	_setTokenURI	Internal	✓	
	_burn	Internal	✓	
ERC721Burnable	Implementation	Context, ERC721		
	burn	Public	✓	-
Pausable	Implementation	Context		
		Public	✓	-
	paused	Public		-
	_requireNotPaused	Internal		
	_requirePaused	Internal		
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
IERC4906	Interface	IERC165, IERC721		

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
tokenId
toString
to
isContract
ERC721
<Constructor>
balanceOf
name
symbol
safeTransferFrom
transferFrom
__unsafe_increaseBalance
tokenURI
_baseURI
_checkOnERC721Received
_safeMint
getApproved
_requireMinted
setApprovalForAll
_setApprovalForAll
_safeTransfer
_mint
_exists
_msgSender
_ownerOf
_transfer
_beforeTokenTransfer
_isApprovedOrOwner
isApprovedForAll
_afterTokenTransfer
approve
_burn
ownerOf
_approve
ERC721Burnable
burn
ERC721URIStorage
_setTokenURI
IERC721 (iface)
balanceOf
ownerOf
safeTransferFrom
transferFrom
approve
setApprovalForAll
getApproved
isApprovedForAll
IERC721Receiver (iface)
onERC721Received
ERC165
supportsInterface
type
IERC721Metadata (iface)
supportsInterface
name
symbol
tokenURI
IERC165 (iface)
supportsInterface
Address (lib)
sendValue
functionCall
functionCallWithValue
functionStaticCall
verifyCallResultFromTarget
isContract
verifyCallResult
_revert
functionDelegateCall
target
staticcall
delegatecall
Context
_msgSender
_msgData
_contextSuffixLength
SignedMath (lib)
abs
max
min
average
Strings (lib)
toString
toHexString
equal
Math (lib)
log10
log256
max
average
ceilDiv
mulDiv
log2
sqrt
min
W3PContractResolved
<Constructor>
onlyOwner
pause
unpause
safeMint
supportsInterface
updateBasePrice1e3
getBasePrice1e3
getBasePrice2e3
updateBasePriceAttributes
getBasePriceAttributes
updateBasePriceZeroAttributes
getBasePriceZeroAttributes
updateMaxSupply
updateMintAmountReceiver
getMintAmountReceiver
updateOwner
<Receive Ether>
<Fallback>
_pause
_unpause
MintLimitReached
calculatePrice
payable
_setTokenURI
_safeMint
InsufficientPayment
Pausable
supportsInterface
<Constructor>
whenNotPaused
_requireNotPaused
paused
whenPaused
_requirePaused
_pause
_msgSender
_unpause

Summary

Web3Punks contract implements a nft mechanism. It allows users to mint NFTs with diverse attributes and dynamic pricing strategies. This audit investigates security issues, business logic concerns 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