Cyberscope

Audit Report

BWORKER

September 2022

Type	BEP20
Network	BSC
Address	0x1061b0268acc6bcec91812c7e18b4cc6f59ada94
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BWORKER
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x1061b0268AcC6BCEC91812c7E18B4cc6F59Ada94
Symbol	BWP
Decimals	10
Total Supply	1,000,000
Domain	bworker.app

Source Files

Filename	**SHA256**
contract.sol	c7dd7908e17ee6d9496884623105629f09ea9dcb342e0effdda094657b123563

Audit Updates

Initial Audit	19th September 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

MT - Mints Tokens

Criticality	critical
Location	contract.sol#L1153
Status	Unresolved

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `manualMintProfit` function. As a result the contract tokens will be highly inflated.

```
function manualMintProfit() public {
     uint256 _currentEpoch = currentEpoch();
     require(
         address(nftContract) != address(0) &&
             _currentEpoch > lastMintEpoch.add(1)
     );
     mintProfit();
 }
```

Recommendation

The 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.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	TSED	Total Supply External Dependency	Unresolved
●	TSBD	Total Supply Balances Diversion	Unresolved
●	BSB	Buyback Sufficient Balance	Unresolved
●	ALBN	Add Liquidity Balance Normalisation	Unresolved
●	PER	Presale Execution Requirement	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L03	Redundant Statements	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L09	Dead Code Elimination	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L1066
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.

```
(bool success, ) = payable(insuranceReceiver).call{
    value: amount,
    gas: 30000
}("");
(success, ) = payable(treasuryReceiver).call{
    value: amountETH.sub(amount).sub(amount).sub(amount),
    gas: 30000
}("");
```

Recommendation

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

TSED - Total Supply External Dependency

Criticality	critical
Location	contract.sol#L1164
Status	Unresolved

Description

The total supply of the Bworker token depends on the external contract `nftContract.totalProfitAt(epoch);`. If the `nftContract` manipulated or have an issue, then it may affect the total supply of Bworker token unpredictable.

```
(uint256 lqTokenAmount, uint256 lqBNBAmount) = tokenPrice();
uint256 mintAmount = 0;
if (lqBNBAmount.mul(lqTokenAmount) > 0) {
    for (
        uint256 epoch = lastMintEpoch.add(1);
        epoch < _currentEpoch;
        epoch++
    ) {
        uint256 BNBAmount = nftContract.totalProfitAt(epoch);
        uint256 tokenAmount;
        tokenAmount = BNBAmount.mul(lqTokenAmount).div(lqBNBAmount);
        _mintProfit[epoch] = tokenAmount;
        _totalSupply = _totalSupply.add(tokenAmount);
        mintAmount = mintAmount.add(tokenAmount);
        _gameBalances[profitHolderAddress] = _gameBalances[
            profitHolderAddress
        ].add(tokenAmount);
    }
}
```

Recommendation

The contract could not rely on an external factor in order to mutate its total supply. If this is a business logic requirement, then it could sanitise the result of the external call so it cannot dramatically affect the internal total supply.

TSBD - Total Supply Balances Diversion

Criticality	critical
Location	contract.sol#L1178
Status	Unresolved

Description

According to the mintProfit requirements, the total supply is increased. Since the total supply is increased then a contract's address should receive the minted tokens. The contract does not mint these tokens to any address. As a result, the balances are diverse from the total supply.

```
tokenAmount = BNBAmount.mul(lqTokenAmount).div(lqBNBAmount);
_mintProfit[epoch] = tokenAmount;
_totalSupply = _totalSupply.add(tokenAmount);
mintAmount = mintAmount.add(tokenAmount);
_gameBalances[profitHolderAddress] = _gameBalances[
    profitHolderAddress
].add(tokenAmount);
```

Recommendation

The contract should not diverse the balances with the total supply.

BSB - Buyback Sufficient Balance

Criticality	minor / informative
Location	contract.sol#L1134
Status	Unresolved

Description

The contract should check if the contract native currency balance is sufficient to proceed with the swapExactETHForTokensSupportingFeeOnTransferTokens. In case of something unexpected produced by the bnbForBuyBack tracking, the contract should not allow the swap execution.

```
function _buyBack() internal swapping {
    (uint256 lqTokenAmount, uint256 lqBNBAmount) = tokenPrice();
    if (lqBNBAmount.mul(lqTokenAmount) == 0 || bnbForBuyBack == 0) {
        return;
    }
    uint256 amount = bnbForBuyBack;
    address[] memory path = new address[](2);
    path[0] = router.WETH();
    path[1] = address(this);
    router.swapExactETHForTokensSupportingFeeOnTransferTokens{
        value: amount
    }(0, path, buyBackReceiver, block.timestamp);
    bnbForBuyBack = 0;
}
```

Recommendation

The contract could check if the contract's native balance is more than the bnbForBuyBack in order to proceed with the swap.

ALBN - Add Liquidity Balance Normalisation

Criticality	minor / informative
Location	contract.sol#L1079
Status	Unresolved

Description

When the liquidity amount is more than the contract's balance, then the contract could liquidate the difference instead of preventing the liquitation.

```
if (
    lqBNBAmount == 0 ||
    lqTokenAmount == 0 ||
    bnbForAddLiquidity == 0 ||
    address(this).balance < bnbForAddLiquidity
) {
    return;
}
```

Recommendation

The contract could liquidate the difference of `bnbForAddLiquidity - address(this).balance` instead of preventing the execution.

PER - Presale Execution Requirement

Criticality	minor / informative
Location	contract.sol#L1531
Status	Unresolved

Description

The contract initialises the presaleNFTPrice with zero. As a result, if the presale begins with zero presaleNFTPrice, the investors will not be able to claim their tokens.

```
require(presaleNFTPrice > 0);
```

Recommendation

The contract should not depend on the presaleNFTPrice in order to proceed with the presale functionality..

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L486,499,504,789,1149,1298,1302,1306,1310,1324,1329,1339,1349,1359,1376,1389
Status	Unresolved

Description

Public functions that are never called by the contract should be declared external to save gas.

```
owner
renounceOwnership
transferOwnership
supportsInterface
manualMintProfit
name
symbol
decimals
totalSupply
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L883,886,887,900,916,904,898,902,896
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
DECIMALS
_name
_symbol
buyBackReceiver
feeDenominator
gameHolderAddress
insuranceReceiver
profitHolderAddress
treasuryReceiver
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L725
Status	Unresolved

Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

```
Context
```

Recommendation

Remove the redundant statements in order to decrease the code size.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L164,166,197,243,737,844,964,981,1262,1263,1264,1265,1273,1279,1284,1289,1521,1555,883,895,912,913,921,923,924,925,927,928
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.

```
DOMAIN_SEPARATOR
PERMIT_TYPEHASH
MINIMUM_LIQUIDITY
WETH
alphabet
IBWORKER_NFT
_flag
_flagAutoMintProfit
_flagAutoAddLiquidity
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor / informative
Location	contract.sol#L6
Status	Unresolved

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L968,985,1289,1555
Status	Unresolved

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
NFTPrice = nftPrice
NFTsCanBeSold = nums
presaleNFTPrice = _price
buyFee = _buyFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L705,614,621,629,643,683,695,660,673,600,103,97,89,93,32,758,771,739
Status	Unresolved

Description

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

```
_verifyCallResult
functionCall
functionCallWithValue
functionDelegateCall
functionStaticCall
sendValue
findDownnerBound
average
max
...
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
Math	Library			
	max	Internal		
	min	Internal		
	average	Internal		
Arrays	Library			
	findDownerBound	Internal		
IPancakeSwapPair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-

	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeSwap Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-

	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeSwapFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-

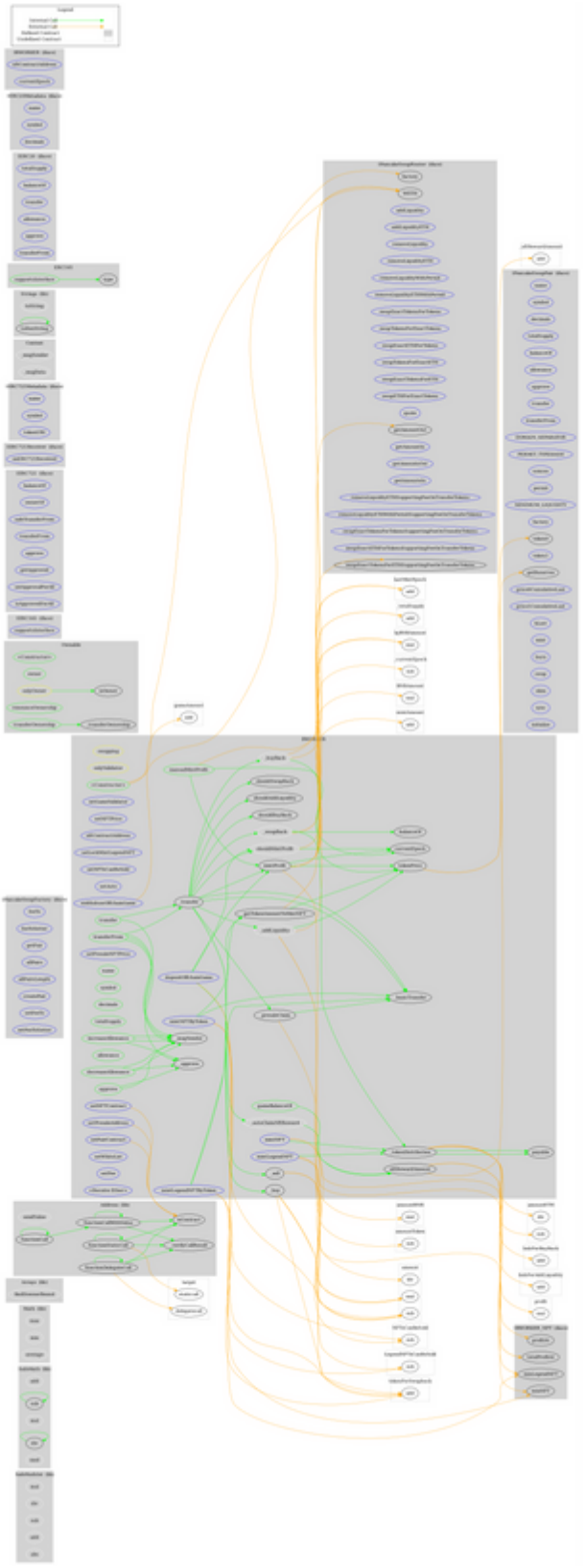
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC165	Interface			
	supportsInterface	External		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	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		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IBWORKER	Interface	IERC20Metadata		
	nftContractAddress	External		-
	currentEpoch	External		-

IBWORKER_NFT	Interface	IERC721Metadata		
	minNFT	External	✓	-
	minLegendNFT	External	✓	-
	profitAt	External		-
	totalProfitAt	External		-
BWORKER	Implementation	Context, IERC20, IERC20Metadata, Ownable, IBWORKER		
	<Constructor>	Public	✓	Ownable
	setGameValidator	External	✓	onlyOwner
	setNFTPrice	External	✓	onlyOwner
	nftContractAddress	External		-
	setLockMintLegendNFT	External	✓	onlyOwner
	setNFTsCanBeSold	External	✓	onlyOwner
	mintNFT	External	Payable	-
	mintNFTByToken	External	✓	-
	mintLegendNFT	External	Payable	-
	getTokenAmountToMintNFT	Public		-
	mintLegendNFTByToken	External	✓	-
	_tokenDistribution	Internal	✓	
	_addLiquidity	Internal	✓	swapping
	_swapBack	Internal	✓	swapping
	_buyBack	Internal	✓	swapping
	manualMintProfit	Public	✓	-
	mintProfit	Internal	✓	
	_autoClaimNftReward	Internal	✓	
	nftRewardAmount	Public		-
	tokenPrice	Public		-
	shouldMintProfit	Internal		
	shouldBuyBack	Internal		
	shouldAddLiquidity	Internal		

	shouldSwapBack	Internal		
	setAuto	External	✓	onlyOwner
	setPairContract	External	✓	onlyOwner
	setNFTContract	External	✓	onlyOwner
	setPresaleAddress	External	✓	onlyOwner
	setPresaleNFTPrice	External	✓	onlyOwner
	currentEpoch	Public		-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	gameBalanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_sell	Internal	✓	
	_buy	Internal	✓	
	depositOffchainGame	External	✓	-
	widthdrawOffchainGame	External	✓	onlyValidator
	setWhiteList	External	✓	onlyOwner
	_presaleClaim	Internal	✓	
	_basicTransfer	Internal	✓	
	setFee	External	✓	onlyOwner
	_approve	Internal	✓	
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	bworker.app
Registry Domain ID	4953EB28F-APP
Creation Date	2022-07-07T04:52:30Z
Updated Date	2022-07-12T04:52:30Z
Registry Expiry Date	2023-07-07T04:52:30Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	None
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain was created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

The Smart Contract analysis reported one critical severity issue. The contract owner has the authority to mint tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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