Cyberscope

Audit Report

ByBet

October 2022

Github https://github.com/bybetdevelopment/smart-contract

Commit 9a922801009eda17b3690d9e7fa8d6d087ff8d36

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Transfers Contract's Tokens 5
Description 5
Recommendation 5
ULTW - Transfers Liquidity to Team Wallet 6
Description 6
Recommendation 6
BC - Blacklists Addresses 7
Description 7
Recommendation 7
Contract Diagnostics 8
US - Untrusted Source 9
Description 9
Recommendation 9
STC - Succeeded Transfer Check 10
Description 10
Recommendation 10
BLC - Business Logic Concern 11
Description 11
Recommendation 11
CR - Code Repetition 12
Description 12

Recommendation 12
L01 - Public Function could be Declared External 13
Description 13
Recommendation 13
L04 - Conformance to Solidity Naming Conventions 14
Description 14
Recommendation 14
L07 - Missing Events Arithmetic 15
Description 15
Recommendation 15
L09 - Dead Code Elimination 16
Description 16
Recommendation 16
L13 - Divide before Multiply Operation 17
Description 17
Recommendation 17
Contract Functions 18
Contract Flow 22
Domain Info 23
Summary 24
Disclaimer 25
About Cyberscope 26

Contract Review

Contract Name	BybetToken
Compiler Version	v0.8.11+commit.d7f03943
Testing Deploy	https://testnet.bscscan.com/token/0xD98A9340DC11030 2620f8Ff55bf26C05170a2f42
Symbol	BBET
Decimals	18
Total Supply	1,200,000,000
Domain	https://bybet.io

Source Files

Filename	**SHA256**
contract.sol	4086d231bc52b615d5bf58bc9efdca82959c7eff68d470c 617477fe8250e2e9b

Audit Updates

Initial Audit	10th October 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L1065
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `emergencySupport` function.

```
function emergencySupport(
      address token,
      address to,
      uint256 amount
  ) external onlyOwner {
      ERC20(token).transfer(to, amount);
  }
```

Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L1012
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `manualDistributeBuyFee` method.

```
function manualDistributeBuyFee() public onlyOwner returns (bool) {
        swapping = true;
        _swapBackToBUSD(balanceOf(address(this)));
        swapping = false;
        Fees memory fees = _sellFeesByTime(block.timestamp);
        distributeRewardBUSD(fees);
        return true;
    }
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BC - Blacklists Addresses

Criticality	critical
Location	contract.sol#L1030
Status	Unresolved

Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by enabling IPinkAntiBot interface.

```
function _transfer(
        address from,
        address to,
        uint256 amount
    ) internal override {
        super._beforeTokenTransfer(from, to, amount);
        require(from != address(0), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");
        if (antiBotEnabled) {
            pinkAntiBot.onPreTransferCheck(from, to, amount);
        }
```

Recommendation

The contract should initialize the pinkAntiBot with the trusted Pinksale antibot address. It should not allow changes after the initialization.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	US	Untrusted Source	Unresolved
	STC	Succeeded Transfer Check	Unresolved
	BLC	Business Logic Concern	Unresolved
	CR	Code Repetition	Unresolved
	L01	Public Function could be Declared External	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L09	Dead Code Elimination	Unresolved
	L13	Divide before Multiply Operation	Unresolved

US - Untrusted Source

Criticality	critical
Location	contract.sol#L852
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

```
IPinkAntiBot public pinkAntiBot;
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization. In addition, it is recommended to enclose every statement that uses external sources in try-catch statement. In order to avoid anomalies in the transaction flow.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L965
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.

```
function distributeRewardBUSD(Fees memory fees) internal returns (bool) {
        uint256 totalFee = fees.liquid + fees.marketing + fees.rewards + fees.team;
        uint256 amountBUSDToFee = ERC20(BUSD_ADDRESS).balanceOf(address(this));
        ERC20(BUSD_ADDRESS).transfer(LIQUID_ADDRESS, amountBUSDToFee * (fees.liquid *
1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(MARKETING_ADDRESS, amountBUSDToFee *
(fees.marketing * 1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(REWARDS_ADDRESS, amountBUSDToFee *
(fees.rewards * 1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(TEAM_ADDRESS, amountBUSDToFee * (fees.team * 1e3
/ totalFee) / 1e3);
        return true;
    }
```

Recommendation

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

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L965
Status	Unresolved

Description

The ratio distribution has an accuracy problem. All integer division rounds down to the nearest integer in solidity. As a result, residual fees are left to the contract.

```
function distributeRewardBUSD(Fees memory fees) internal returns (bool) {
        uint256 totalFee = fees.liquid + fees.marketing + fees.rewards + fees.team;
        uint256 amountBUSDToFee = ERC20(BUSD_ADDRESS).balanceOf(address(this));
        ERC20(BUSD_ADDRESS).transfer(LIQUID_ADDRESS, amountBUSDToFee * (fees.liquid *
1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(MARKETING_ADDRESS, amountBUSDToFee *
(fees.marketing * 1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(REWARDS_ADDRESS, amountBUSDToFee *
(fees.rewards * 1e3 / totalFee) / 1e3);
        ERC20(BUSD_ADDRESS).transfer(TEAM_ADDRESS, amountBUSDToFee * (fees.team *
1e3 / totalFee) / 1e3);
        return true;
    }
```

Recommendation

The ratio distribution may produce accuracy issues. All integer division rounds down to the nearest integer in Solidity. As a result, the remaining fees will remain in the contract.

CR - Code Repetition

Criticality	minor / informative
Location	contract.sol#L915,938
Status	Unresolved

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The following code segment is used in two functions. The `_sellFeesByTime` and `_buyFeesByTime` functions.

```
if (time >= launchTime + 180 days) {
    return sellFeesInfo[launchTime + 180 days];
}
if (time >= launchTime + 150 days) {
    return sellFeesInfo[launchTime + 150 days];
}
if (time >= launchTime + 120 days) {
    return sellFeesInfo[launchTime + 120 days];
}
if (time >= launchTime + 90 days) {
    return sellFeesInfo[launchTime + 90 days];
}
if (time >= launchTime + 60 days) {
    return sellFeesInfo[launchTime + 60 days];
}
if (time >= launchTime + 30 days) {
    return sellFeesInfo[launchTime + 30 days];
```

Recommendation

The contract could use an internal function that accepts the fee mapping as argument. As a result, the repetitive methods will be removed.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L975,542,275,479,589,462,505,892,570,885,283,1012,454,999,486,524,513
Status	Unresolved

Description

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

```
setAutomatedMarketMakerPair
transferFrom
renounceOwnership
decimals
decreaseAllowance
symbol
transfer
configLaunchTime
increaseAllowance
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L854,837,839,885,7,835,881,836,877,975,838
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.

```
_isFeeExempt
REWARDS_ADDRESS
BUSD_ADDRESS
PSAntibot
WETH
LIQUID_ADDRESS
_addr
MARKETING_ADDRESS
_pair
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L892
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.

```
launchTime = time
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L667
Status	Unresolved

Description

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

```
_burn
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L965
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
ERC20(BUSD_ADDRESS).transfer(MARKETING_ADDRESS,amountBUSDToFee * (fees.marketing * 1e3 / totalFee) / 1e3)
ERC20(BUSD_ADDRESS).transfer(REWARDS_ADDRESS,amountBUSDToFee * (fees.rewards * 1e3 / totalFee) / 1e3)
ERC20(BUSD_ADDRESS).transfer(TEAM_ADDRESS,amountBUSDToFee * (fees.team * 1e3 / totalFee) / 1e3)
ERC20(BUSD_ADDRESS).transfer(LIQUID_ADDRESS,amountBUSDToFee * (fees.liquid * 1e3 / totalFee) / 1e3)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

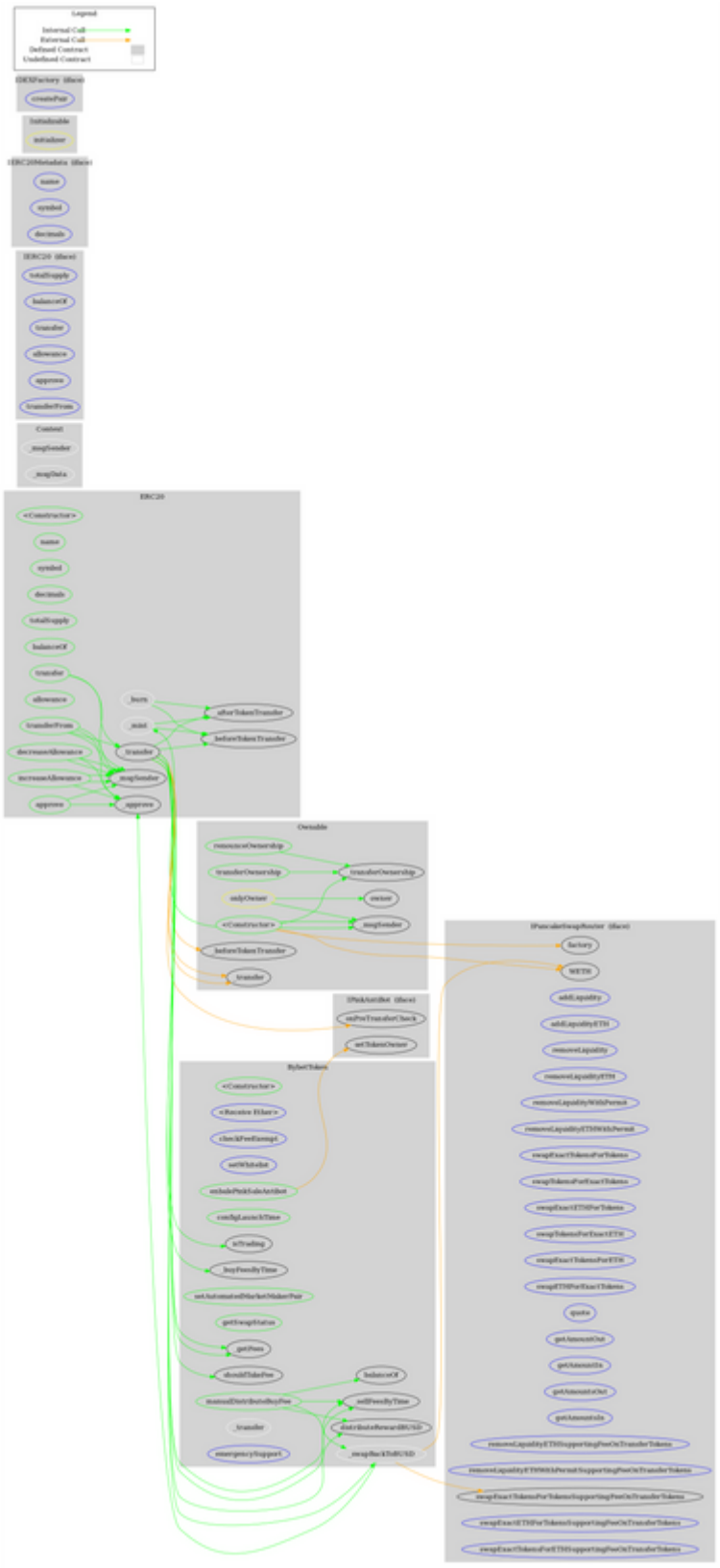
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IPancakeSwap Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	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	-
	swapExactTokensForETHSupporting	External	✓	-

	FeeOnTransferTokens			
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-

	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Initializable	Implementation			
IDEXFactory	Interface			
	createPair	External	✓	-
IPinkAntiBot	Interface			
	setTokenOwner	External	✓	-
	onPreTransferCheck	External	✓	-
BybetToken	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20 Ownable
	<Receive Ether>	External	Payable	-
	checkFeeExempt	External		-
	setWhitelist	External	✓	onlyOwner
	enbalePinkSaleAntibot	Public	✓	onlyOwner
	configLaunchTime	Public	✓	onlyOwner
	_sellFeesByTime	Internal		
	_buyFeesByTime	Internal		
	_getFees	Internal		

	distributeRewardBUSD	Internal	✓	
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_swapBackToBUSD	Internal	✓	
	getSwapStatus	Public		-
	shouldTakeFee	Public		-
	isTrading	Public		-
	manualDistributeBuyFee	Public	✓	onlyOwner
	_transfer	Internal	✓	
	emergencySupport	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	bybet.io
Registry Domain ID	c58c6bde1f0b46c4b5ee83b2eb58365e-DONUTS
Creation Date	2022-09-18T02:34:54Z
Updated Date	2022-09-28T11:15:58Z
Registry Expiry Date	2023-09-18T02:34:54Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 22 days before the creation of the audit. It will expire in 11 months.

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

Summary

There are some functions that can be abused by the owner like transferring tokens to the team’s wallet, transferring funds to the team’s wallet, and massively blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a max of 15% sell fees and a max of 5% buy fees. The fees are gradually decreasing after the first thirty days for the buy fees and after ninety days for the sell fees.

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