Cyberscope

# Audit Report

# Betswamp

March 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC TESTNET |
| Address | 0x9E8128A37A04960b4a8F55A26CE988e6E5d60F9e |
| Audited by | © cyberscope |

# Table of Contents

# Contract Review

| | |
|---|---|
| **Contract Name** | BetSwampERC20Token |
| **Compiler Version** | v0.7.5+commit.eb77ed08 |
| **Optimization** | 200 runs |
| **Licence** | MIT |
| **Explorer** | https://bscscan.com/token/0x9E8128A37A04960b4a8F55A26CE988e6E5d60F9e |
| **Symbol** | BETS |
| **Decimals** | 9 |
| **Total Supply** | 1,000,000 |
| **Source** | contract.sol |
| **Domain** | betswamp.com |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 8th March 2022 |
| **Corrected** | |

# Contract Analysis

Critical Medium Minor Pass

| Severity | Code | Description |
|---|---|---|
| ● | ST | Contract Owner is not able to stop or pause transactions |
| ● | OCTD | Contract Owner is not able to transfer tokens from specific address |
| ● | OTUT | Owner Transfer User's Tokens |
| ● | ELFM | Contract Owner is not able to increase fees more than a reasonable percent (25%) |
| ● | ULTW | Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent |
| ● | MT | Contract Owner is not able to mint new tokens |
| ● | BT | Contract Owner is not able to burn tokens from specific wallet |
| ● | BC | Contract Owner is not able to blacklist wallets from selling |

# ST - Stop Transactions

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L1316,1362 |

## Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `tradingActive` to false.

```
if(!tradingActive){
    require(_isExcludedFromFees[from] || _isExcludedFromFees[to], "Trading is
not active.");
  }
```

Additionally the owner may increase the `burnFeeOnSell` to 100% and prevent any users from selling. This is the behaviour of a honeypot.

```
// on sell
   else if (automatedMarketMakerPairs[to] && burnFeeOnSell > 0){
        fees = amount.mul(burnFeeOnSell).div(100);
        tokensForBurn += fees;
```

## Recommendation

The contract could embody a check for not allowing setting the `burnFeeOnSell` less than a reasonable amount.

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.

# MT - Mint Tokens

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L1244 |

## Description

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

```
  function mint(address account_, uint256 amount_) external onlyVaultOrVested()
{
        if(msg.sender == _vested)
        amount_ == monthlyVestedAmount;
        _mint(account_, amount_);
    }
```

## 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.

# BC - Blacklisted Contracts

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L1284 |

## Description

The contract owner has the authority to stop contracts from selling. The owner may take advantage of it by calling the `blacklistAddress` function while the burn fee is very high value. This will result in burning the tokens instead of selling them.

```
    function blackListAddresses(address[] memory addrs) external onlyOwner
returns (bool) {
        for(uint256 i = 0; i < addrs.length; i++) {
            _blackListAddr[addrs[i]] = true;
        }
        return true;
    }
```

## 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.

# Contract Diagnostics

Critical Medium Minor

| Severity | Code | Description |
|---|---|---|
| | L01 | Public Function could be Declared External |
| | L02 | State Variables could be Declared Constant |
| | L05 | Unused State Variable |
| | L04 | Conformance to Solidity Naming Conventions |
| | L09 | Dead Code Elimination |
| | L07 | Missing Events Arithmetic |
| | L06 | Missing Events Access Control |
| | L15 | Local Scope Variable Shadowing |
| | L14 | Uninitialized Variables in Local Scope |

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L778,782,786,794,803,808,814,819,926,949 and 8 more |

## Description

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

```
setAutomatedMarketMakerPair
isExcludedFromFees
burnFrom
burn
vested
vault
renounceOwnership
owner
nonces
...
```

## Recommendation

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

# L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L1190,1191 |

## Description

Constant state variables should be declared constant to save gas.

```
totalVestedAmount
monthlyVestedAmount
```

## Recommendation

Add the constant attribute to state variables that never change.

# L05 - Unused State Variable

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L748 |

## Description

There are segments that contain unused state variables.

```
ERC20TOKEN_ERC1820_INTERFACE_ID
```

## Recommendation

Remove unused state variables.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L751,754,757,760,763,766,907,964,996,997 and 5 more |

## 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.

```
_isExcludedMaxTransactionAmount
deadAddress
_burnFrom
_fee
WETH
_vested
_vault
_owner
DOMAIN_SEPARATOR
...
```

## Recommendation

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

# L09 - Dead Code Elimination

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L555,515,525,540,550,462,489,879,33,109 and 38 more |

## Description

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

```
substractPercentage
sqrrt
quadraticPricing
percentageOfTotal
percentageAmount
bondingCurve
average
remove
length
...
```

## Recommendation

Remove unused functions.

# L07 - Missing Events Arithmetic

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L1303,1308,1312,1316 |

## 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.

```
burnFeeOnSell = newBurnFeeOnSell
maxWallet = newNum * (10 ** 18)
maxTransactionAmount = newNum * (10 ** 18)
blackListFee = _fee
```

## Recommendation

Emit an event for critical parameter changes.

# L06 - Missing Events Access Control

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L999,1005 |

## Description

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

```
_vested = vested_
_vault = vault_
```

## Recommendation

Emit an event for critical parameter changes.

# L15 - Local Scope Variable Shadowing

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L1227 |

## Description

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

```
totalSupply
```

## Recommendation

The local variables should have different names from the upper scoped variables.

# L14 - Uninitialized Variables in Local Scope

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L320,184,248 |

## Description

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

```
bytes4Array_
addressArray
```

## Recommendation

All the local scoped variables should be initialized.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| EnumerableSet | Library | | | |
| | _add | Private | ✓ | |
| | _remove | Private | ✓ | |
| | _contains | Private | | |
| | _length | Private | | |
| | _at | Private | | |
| | _getValues | Private | | |
| | _insert | Private | ✓ | |
| | add | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | contains | Internal | | |
| | length | Internal | | |
| | at | Internal | | |
| | getValues | Internal | | |
| | insert | Internal | ✓ | |
| | add | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | contains | Internal | | |
| | length | Internal | | |
| | at | Internal | | |
| | getValues | Internal | | |
| | insert | Internal | ✓ | |
| | add | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | contains | Internal | | |
| | length | Internal | | |
| | at | Internal | | |
| | getValues | Internal | | |

| | | | | |
|---|---|---|---|---|
| | insert | Internal | ✓ | |
| | add | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | contains | Internal | | |
| | length | Internal | | |
| | at | Internal | | |
| | add | Internal | ✓ | |
| | remove | Internal | ✓ | |
| | contains | Internal | | |
| | length | Internal | | |
| | at | Internal | | |
| | | | | |
| **Address** | Library | | | |
| | isContract | Internal | | |
| | sendValue | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | _functionCallWithValue | Private | ✓ | |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| **SafeMath** | Library | | | |
| | add | Internal | | |
| | sub | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |

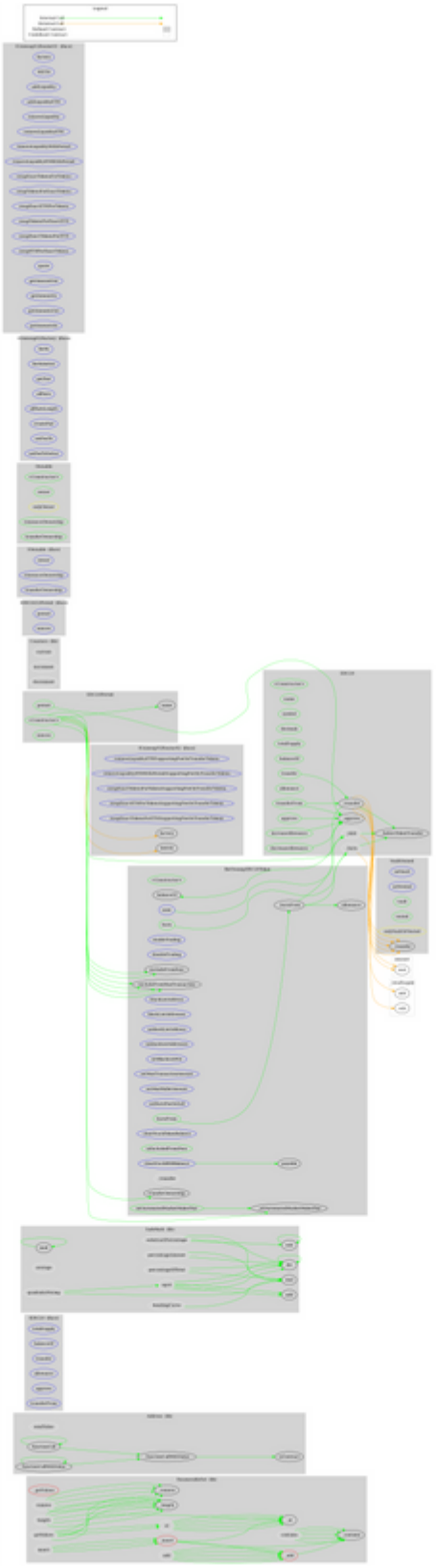
| | | | | |
|---|---|---|---|---|
| | div | Internal | | |
| | mod | Internal | | |
| | mod | Internal | | |
| | sqrrt | Internal | | |
| | percentageAmount | Internal | | |
| | substractPercentage | Internal | | |
| | percentageOfTotal | Internal | | |
| | average | Internal | | |
| | quadraticPricing | Internal | | |
| | bondingCurve | Internal | | |
| | | | | |
| **ERC20** | Implementation | IERC20 | | |
| | <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 | ✓ | |
| | | | | |
| **Counters** | Library | | | |
| | current | Internal | | |
| | increment | Internal | ✓ | |
| | decrement | Internal | ✓ | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **IERC2612Permit** | Interface | | | |
| | permit | External | ✓ | - |
| | nonces | External | | - |
| | | | | |
| **ERC20Permit** | Implementation | ERC20, IERC2612Permit | | |
| | <Constructor> | Public | ✓ | - |
| | permit | Public | ✓ | - |
| | nonces | Public | | - |
| | | | | |
| **IOwnable** | Interface | | | |
| | owner | External | | - |
| | renounceOwnership | External | ✓ | - |
| | transferOwnership | External | ✓ | - |
| | | | | |
| **Ownable** | Implementation | IOwnable | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| **VaultOwned** | Implementation | Ownable | | |
| | setVault | External | ✓ | onlyOwner |
| | setVested | External | ✓ | onlyOwner |
| | vault | Public | | - |
| | vested | Public | | - |
| | | | | |
| **IUniswapV2Factory** | Interface | | | |
| | feeTo | External | | - |
| | feeToSetter | External | | - |
| | getPair | External | | - |
| | allPairs | External | | - |
| | allPairsLength | External | | - |
| | createPair | External | ✓ | - |

| | | | | |
|---|---|---|---|---|
| | setFeeTo | External | ✓ | - |
| | setFeeToSetter | External | ✓ | - |
| | | | | |
| **IUniswapV2Router01** | 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 | | - |
| | | | | |
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 | | |
| | removeLiquidityETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokensSupportingFeeOnTransferTokens | External | Payable | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **BetSwampERC20Token** | Implementation | IERC20, ERC20Permit, VaultOwned | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | mint | External | ✓ | onlyVaultOrVested |
| | burn | Public | ✓ | - |
| | burnFrom | Public | ✓ | - |
| | enableTrading | External | ✓ | onlyOwner |
| | disableTrading | External | ✓ | onlyOwner |
| | excludeFromFees | Public | ✓ | onlyOwner |
| | excludeFromMaxTransaction | Public | ✓ | onlyOwner |
| | blackListAddress | External | ✓ | onlyOwner |
| | blackListAddresses | External | ✓ | onlyOwner |
| | unblackListAddress | External | ✓ | onlyOwner |
| | unblackListAddresses | External | ✓ | onlyOwner |
| | setBlackListFee | External | ✓ | onlyOwner |
| | setMaxTransactionAmount | External | ✓ | onlyOwner |
| | setMaxWalletAmount | External | ✓ | onlyOwner |
| | setBurnFeeOnSell | External | ✓ | onlyOwner |
| | clearStuckBNBBalance | External | ✓ | onlyOwner |
| | clearStuckTokenBalance | External | ✓ | onlyOwner |
| | isExcludedFromFees | Public | | - |
| | setAutomatedMarketMakerPair | Public | ✓ | onlyOwner |
| | _setAutomatedMarketMakerPair | Private | ✓ | |
| | _burnFrom | Public | ✓ | - |
| | _transfer | Internal | ✓ | |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | |
| **Registry Domain ID** | 2624723449_DOMAIN_COM-VRSN |
| **Creation Date** | 2021-07-06T10:54:28.00Z |
| **Updated Date** | 0001-01-01T00:00:00.00Z |
| **Registry Expiry Date** | |
| **Registrar WHOIS Server** | whois.namecheap.com |
| **Registrar URL** | http://www.namecheap.com |
| **Registrar** | NAMECHEAP INC |
| **Registrar IANA ID** | 1068 |

The domain has been created 8 months before the creation of the audit.

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

# Summary

Betswamp is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like minting tokens, stopping transactions and mass blacklisting wallets from selling. 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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io