Cyberscope

# Audit Report

# Respan

February 2024

| | |
|---|---|
| Network | BSC |
| Address | 0x667C69bD295f0dc38b5b97DB06556129418514c6 |
| Audited by | © cyberscope |

# Analysis

Critical Medium Minor / Informative Pass

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | ST | Stops Transactions | Passed |
| ● | OTUT | Transfers User's Tokens | Passed |
| ● | ELFM | Exceeds Fees Limit | Passed |
| ● | MT | Mints Tokens | Passed |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Passed |

# Diagnostics

Critical Medium Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| | MEM | Misleading Error Messages | Unresolved |
| | L02 | State Variables could be Declared Constant | Unresolved |
| | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| | L07 | Missing Events Arithmetic | Unresolved |
| | L14 | Uninitialized Variables in Local Scope | Unresolved |
| | L16 | Validate Variable Setters | Unresolved |
| | L20 | Succeeded Transfer Check | Unresolved |

# Table of Contents

# Review

| | |
|---|---|
| **Contract Name** | Respan |
| **Compiler Version** | v0.8.19+commit.7dd6d404 |
| **Optimization** | 200 runs |
| **Explorer** | https://bscscan.com/address/0x667c69bd295f0dc38b5b97db06556129418514c6 |
| **Address** | 0x667C69bD295f0dc38b5b97DB06556129418514c6 |
| **Network** | BSC |
| **Symbol** | RSPN |
| **Decimals** | 9 |
| **Total Supply** | 2,000,000,000 |
| **Badge Eligibility** | Yes |

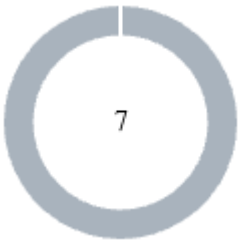
# Audit Updates

| | |
|---|---|
| **Initial Audit** | 24 Dec 2023<br>https://github.com/cyberscope-io/audits/blob/main/rspn/v1/audit.pdf |
| **Corrected Phase 2** | 08 Feb 2024 |

## Source Files

| Filename | SHA256 |
| --- | --- |
| **Respan.sol** | 0620a90c0d38b5309e5b4c5ac5a7bccc95ae003192a7018ce2df17455468aca1 |

# Findings Breakdown




| | |
|---|---|
| Critical | 0 |
| Medium | 0 |
| Minor / Informative | 7 |

| Severity | Unresolved | Acknowledged | Resolved | Other |
|---|---|---|---|---|
| Critical | 0 | 0 | 0 | 0 |
| Medium | 0 | 0 | 0 | 0 |
| Minor / Informative | 7 | 0 | 0 | 0 |

# MEM - Misleading Error Messages

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L344,345,378,503,513 |
| **Status** | Unresolved |

## Description

The contract is using misleading error messages. These error messages do not accurately reflect the problem, making it difficult to identify and fix the issue. As a result, the users will not be able to find the root cause of the error.

```
require(!tradingEnabled)
require(cInit != address(this) && aInit != address(this) && cInit !=
aInit)

require(account != address(this)
                && account != lpPair
                && account != DEAD)
require(gas < 750000)
require(allowedPresaleExclusion)
```

## Recommendation

The team is suggested to provide a descriptive message to the errors. This message can be used to provide additional context about the error that occurred or to explain why the contract execution was halted. This can be useful for debugging and for providing more information to users that interact with the contract.

## L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L176 |
| **Status** | Unresolved |

### Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
address public USDT = 0x55d398326f99059fF775485246999027B3197955
```

### Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L33,135,136,137,138,139,154,160,170,176,191,429,454,497 |
| **Status** | Unresolved |

### Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
function WETH() external pure returns (address);
uint256 constant private startingSupply = 2_000_000_000
string constant private _name = "Respan"
string constant private _symbol = "RSPN"
uint8 constant private _decimals = 9
uint256 constant private _tTotal = startingSupply * (10 ** _decimals)

Fees public _taxRates = Fees({
        buyFee: 900,
        sellFee: 900,
        transferFee: 0
    })

...
```

## Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

## L07 - Missing Events Arithmetic

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L465,477,487,504 |
| **Status** | Unresolved |

### Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

```
_maxTxAmount = (_tTotal * percent) / divisor
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
piSwapPercent = priceImpactSwapPercent
cashierGas = gas
```

### Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues with its arithmetic.

## L14 - Uninitialized Variables in Local Scope

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L687 |
| **Status** | Unresolved |

### Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
bool checked
```

### Recommendation

By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

## L16 - Validate Variable Setters

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L349 |
| **Status** | Unresolved |

### Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
lpPair = constructorLP
```

### Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

## L20 - Succeeded Transfer Check

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/Respan.sol#L783 |
| **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.

```
TOKEN.transfer(_owner, TOKEN.balanceOf(address(this)))
```

### Recommendation

The contract should check if the result of the transfer methods is successful. The team is advised to check the SafeERC20 library from the Openzeppelin library.

# Functions Analysis

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | decimals | External | | - |
| | symbol | External | | - |
| | name | External | | - |
| | getOwner | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| **IFactoryV2** | Interface | | | |
| | getPair | External | | - |
| | createPair | External | ✓ | - |
| | | | | |
| **IV2Pair** | Interface | | | |
| | factory | External | | - |

| | | | | |
|---|---|---|---|---|
| | getReserves | External | | - |
| | sync | External | ✓ | - |
| | | | | |
| **IRouter01** | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | addLiquidityETH | External | Payable | - |
| | addLiquidity | External | ✓ | - |
| | swapExactETHForTokens | External | Payable | - |
| | getAmountsOut | External | | - |
| | getAmountsIn | External | | - |
| | | | | |
| **IRouter02** | Interface | IRouter01 | | |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokensSupportingFeeOnTransferTokens | External | Payable | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactTokensForTokens | External | ✓ | - |
| | | | | |
| **Initializer** | Interface | | | |
| | setLaunch | External | ✓ | - |
| | getConfig | External | ✓ | - |
| | getInits | External | ✓ | - |
| | setLpPair | External | ✓ | - |

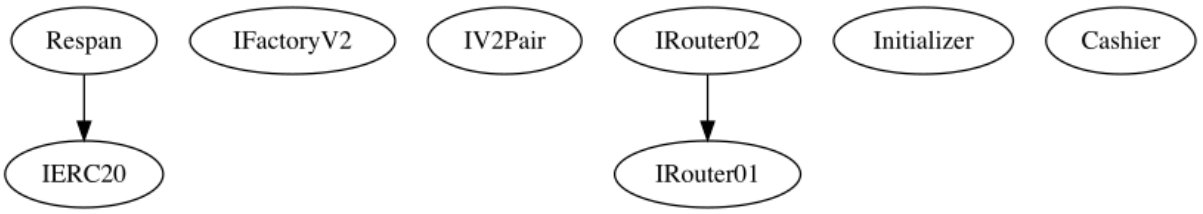
| | | | | |
|---|---|---|---|---|
| | checkUser | External | ✓ | - |
| | setProtections | External | ✓ | - |
| | removeSniper | External | ✓ | - |
| | removeBlacklisted | External | ✓ | - |
| | isBlacklisted | External | | - |
| | transfer | External | ✓ | - |
| | setBlacklistEnabled | External | ✓ | - |
| | setBlacklistEnabledMultiple | External | ✓ | - |
| | getRandom | External | ✓ | - |
| | | | | |
| **Cashier** | Interface | | | |
| | setRewardsProperties | External | ✓ | - |
| | tally | External | ✓ | - |
| | load | External | Payable | - |
| | loadManual | External | ✓ | - |
| | cashout | External | ✓ | - |
| | giveMeWelfarePlease | External | ✓ | - |
| | getTotalDistributed | External | | - |
| | getUserInfo | External | | - |
| | getUserRealizedRewards | External | | - |
| | getPendingRewards | External | | - |
| | initialize | External | ✓ | - |
| | getCurrentReward | External | | - |

| | | | | |
|---|---|---|---|---|
| **Respan** | Implementation | IERC20 | | |
| | | Public | ✓ | - |
| | transferOwner | External | ✓ | onlyOwner |
| | renounceOwnership | External | ✓ | onlyOwner |
| | | External | Payable | - |
| | totalSupply | External | | - |
| | decimals | External | | - |
| | symbol | External | | - |
| | name | External | | - |
| | getOwner | External | | - |
| | balanceOf | Public | | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | _approve | Internal | ✓ | |
| | approveContractContingency | Public | ✓ | onlyOwner |
| | transfer | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | setNewRouter | External | ✓ | onlyOwner |
| | setLpPair | External | ✓ | onlyOwner |
| | setInitializers | Public | ✓ | onlyOwner |
| | isExcludedFromFees | External | | - |
| | isExcludedFromDividends | External | | - |

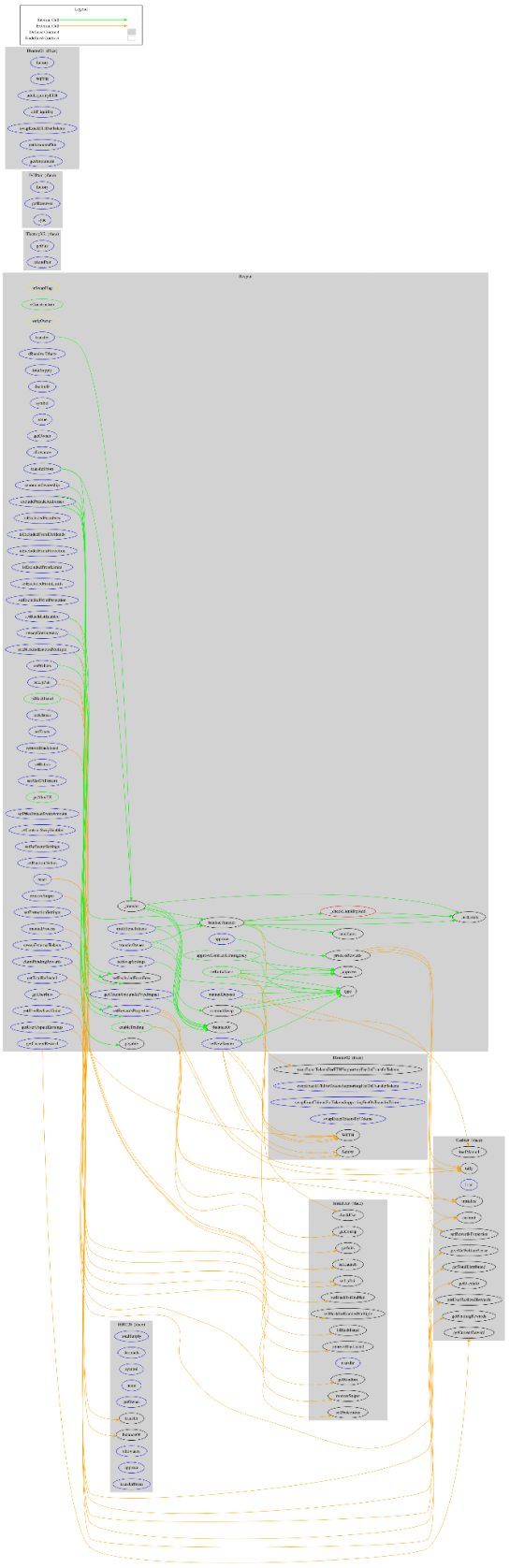
| | | | | |
|---|---|---|---|---|
| | isExcludedFromProtection | External | | - |
| | isExcludedFromLimits | External | | - |
| | setExcludedFromLimits | External | ✓ | onlyOwner |
| | setRewardsProperties | Public | ✓ | onlyOwner |
| | setExcludedFromFees | Public | ✓ | onlyOwner |
| | setExcludedFromProtection | External | ✓ | onlyOwner |
| | setBlacklistEnabled | External | ✓ | onlyOwner |
| | setBlacklistEnabledMultiple | External | ✓ | onlyOwner |
| | isBlacklisted | Public | | - |
| | removeBlacklisted | External | ✓ | onlyOwner |
| | removeSniper | External | ✓ | onlyOwner |
| | setProtectionSettings | External | ✓ | onlyOwner |
| | lockTaxes | External | ✓ | onlyOwner |
| | setTaxes | External | ✓ | onlyOwner |
| | setWallets | External | ✓ | onlyOwner |
| | setRatios | External | ✓ | onlyOwner |
| | setMaxTxPercent | External | ✓ | onlyOwner |
| | getMaxTX | Public | | - |
| | getTokenAmountAtPriceImpact | External | | - |
| | setSwapSettings | External | ✓ | onlyOwner |
| | setPriceImpactSwapAmount | External | ✓ | onlyOwner |
| | setContractSwapEnabled | External | ✓ | onlyOwner |
| | setRewardsProperties | External | ✓ | onlyOwner |

| | | | | |
|---|---|---|---|---|
| | setReflectorSettings | External | ✓ | onlyOwner |
| | setRandomValues | External | ✓ | onlyOwner |
| | excludePresaleAddresses | External | ✓ | onlyOwner |
| | _hasLimits | Internal | | |
| | _transfer | Internal | ✓ | |
| | contractSwap | Internal | ✓ | inSwapFlag |
| | react | External | ✓ | onlyOwner |
| | _checkLiquidityAdd | Private | ✓ | |
| | enableTrading | Public | ✓ | onlyOwner |
| | finalizeTransfer | Internal | ✓ | |
| | processRewards | Internal | ✓ | |
| | manualProcess | External | ✓ | - |
| | takeTaxes | Internal | ✓ | |
| | multiSendTokens | External | ✓ | onlyOwner |
| | manualDeposit | External | ✓ | onlyOwner |
| | sweepContingency | External | ✓ | onlyOwner |
| | sweepExternalTokens | External | ✓ | onlyOwner |
| | claimPendingRewards | External | ✓ | - |
| | getTotalReflected | External | | - |
| | getUserInfo | External | | - |
| | getUserRealizedGains | External | | - |
| | getUserUnpaidEarnings | External | | - |
| | getCurrentReward | External | | - |

# Inheritance Graph



# Flow Graph



# Summary

Respan contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Respan is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 20% in sales and 10% in buys.

# 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