Cyberscope

# Audit Report

# ACG

May 2023

SHA256 1e37940231b6ace85bf7bf10630ab580d1d9e7c6e315ac115c08b1df76ba4aad

Audited by © cyberscope

# 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 | Passed |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Passed |

# Diagnostics

Critical Medium Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| | CR | Code Repetition | Acknowledged |
| | RSML | Redundant SafeMath Library | Acknowledged |
| | L04 | Conformance to Solidity Naming Conventions | Unresolved |

# Table of Contents

# Review

| | |
|---|---|
| **Contract Name** | ACG |
| **Testing Deploy** | https://testnet.bscscan.com/address/0xc4da1dd95078941d3071ffb7bf2ed186251ec21c |
| **Symbol** | ACG |
| **Decimals** | 18 |
| **Total Supply** | 1,000,000,000 |

## Audit Updates

| | |
|---|---|
| **Initial Audit** | 23 May 2023 https://github.com/cyberscope-io/audits/blob/main/acg/v1/audit.pdf |
| **Corrected Phase 2** | 28 May 2023 |

## Source Files

| Filename | SHA256 |
| --- | --- |
| **contracts/acg.sol** | 1e37940231b6ace85bf7bf10630ab580d1d9e7c6e315ac115c08b1df76ba4aad |
| **contracts/contracts/access/Ownable.sol** | 42df7a70b8190e7c8e3aeb443aeacc2b23b389b18fa2ce00e9eb60a367a2bb20 |
| **contracts/contracts/interfaces/IUniswapV2Factory.sol** | 3dd4c1f051cee242d1c81b3868d19d983706f47dc6d4e61c83e8645dab7b190f |
| **contracts/contracts/interfaces/IUniswapV2Pair.sol** | d031a0cf0541e16cc08a0772453796dcbf77727976822ac038dbea47e16171cb |
| **contracts/contracts/interfaces/IUniswapV2Router01.sol** | 9e9232b0ab8af12bf698a622047a0057ab2b5b068360e24c8599576a40653601 |
| **contracts/contracts/interfaces/IUniswapV2Router02.sol** | add2f9ec336a24dfe0fcf25cd27fd11860fa09f8e303867f5188b2b1769b31e4 |
| **contracts/contracts/token/ERC20/ERC20.sol** | bce14c3fd3b1a668529e375f6b70ffdf9cef8c4e410ae99608be5964d98fa701 |
| **contracts/contracts/token/ERC20/extensions/IERC20Metadata.sol** | af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990 |
| **contracts/contracts/token/ERC20/extensions/IERC20Permit.sol** | 2919f8aa74c48a2fc38fff7875ebc9d1604e9180f8c57416ba1ee589fe0dde60 |
| **contracts/contracts/token/ERC20/IERC20.sol** | 94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5 |
| **contracts/contracts/token/ERC20/SafeERC20.sol** | 1d489ce3f5dd4966c090782c7547f51128968221f592301153c0644dfe862179 |
| **contracts/contracts/utils/Address.sol** | 8160a4242e8a7d487d940814e5279d934e81f0436689132a4e73394bab084a6d |

| | |
|---|---|
| **contracts/contracts/utils/Context.sol** | 1458c260d010a08e4c20a4a517882259a2<br>3a4baa0b5bd9add9fb6d6a1549814a |
| **contracts/contracts/utils/math/SafeMath.sol** | 0dc33698a1661b22981abad8e5c6f5ebca<br>0dfe5ec14916369a2935d888ff257a |

# Findings Breakdown


| Severity | Unresolved | Acknowledged | Resolved | Other |
|---|---|---|---|---|
| Critical | 0 | 0 | 0 | 0 |
| Medium | 0 | 0 | 0 | 0 |
| Minor / Informative | 0 | 4 | 0 | 0 |

## CR - Code Repetition

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/acg.sol#L719,762,982 |
| **Status** | Acknowledged |

### Description

The contract contains repetitive code segments. There are potential issues that can arise when using code segments in Solidity. Some of them can lead to issues like gas efficiency, complexity, readability, security, and maintainability of the source code. It is generally a good idea to try to minimize code repetition where possible.

```
// Update the total required buy fees to reflect the new fee configuration
totalRequiredBuyFees = totalNewFees;
// Update the total required fees to reflect the new fee configuration
totalRequiredFees = totalNewFees.add(totalRequiredSellFees).add(
    penaltyFee
);
// Update the total ERC20 fees to reflect the new fee configuration
// Subtract the liquidity fee as it is handled separately
totalERC20Fees = totalNewFees
    .add(totalRequiredSellFees)
    .add(penaltyFee)
    .sub(liquidityFee);
```

### Recommendation

The team is advised to avoid repeating the same code in multiple places, which can make the contract easier to read and maintain. The authors could try to reuse code wherever possible, as this can help reduce the complexity and size of the contract. For instance, the contract could reuse the common code segments in an internal function in order to avoid repeating the same code in multiple places.

### Team Update

The team replied with the following statement:

"*needed to be like this (no issues onlyOwner)*"

# TUU - Time Units Usage

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/acg.sol#L609 |
| **Status** | Acknowledged |

## Description

The contract is using arbitrary numbers to form time-related values. As a result, it decreases the readability of the codebase and prevents the compiler to optimize the source code.

```
uint256 _newSellTimeLimitInBlocks = _newLimit.mul(60).div(3);
```

## Recommendation

It is a good practice to use the time units reserved keywords like `seconds` , `minutes` , `hours` , `days` , `weeks` and `years` to process time-related calculations.

It's important to note that these time units are simply a shorthand notation for representing time in seconds, and do not have any effect on the actual passage of time or the execution of the contract. The time units are simply a convenience for expressing time in a more human-readable form.

## Team Update

The team replied with the following statement:

*"block.number is more acurate for the anti-bot mechanism, regarding RPCs[nodes]"*

## RSML - Redundant SafeMath Library

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/acg.sol |
| **Status** | Acknowledged |

### Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert to underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, overhead and increases gas consumption unnecessarily.

```
library SafeMath {...}
```

### Recommendation

The team is advised to remove the SafeMath library. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes.

### Team Update

The team replied with the following statement:

"*need a safeMath Library*"

## L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | contracts/acg.sol#L90,480,517,536,577,588,603,624,634,644,654,664,674,684,1121,1832,1854,1855,1879,1880,1905,1906,1907 |
| **Status** | Acknowledged |

### 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.

```
uint256 public constant maxTxAmount = 150000000 * 10 ** 18
address _token
uint256 _newLimit
bool _option
uint256 _newCount
address _newWallet
bool _enabled
uint256 _amount
address _to
```

### 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.

## Team Update

The team has acknowledged the finding.

# Functions Analysis

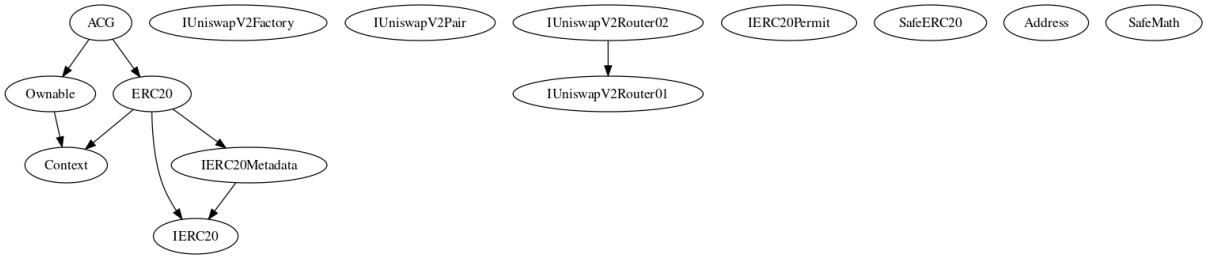
| Contract | Type | Bases | | |
|---|---|---|---|---|
| | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **ACG** | Implementation | ERC20, Ownable | | |
| | | Public | ✓ | ERC20 |
| | name | Public | | - |
| | symbol | Public | | - |
| | decimals | Public | | - |
| | totalSupply | Public | | - |
| | balanceOf | Public | | - |
| | | External | Payable | - |
| | _approve | Internal | ✓ | |
| | transfer | Public | ✓ | - |
| | allowance | Public | | - |
| | approve | Public | ✓ | - |
| | transferFrom | Public | ✓ | - |
| | increaseAllowance | Public | ✓ | - |
| | decreaseAllowance | Public | ✓ | - |
| | getBNBBalance | Public | | - |
| | getErc20TokenFeeBalance | Public | | - |
| | getErc20TokensBalance | Public | | - |

| | | | | |
|---|---|---|---|---|
| | setSwapRouter | External | ✓ | onlyOwner |
| | setSwapThresholdLimit | External | ✓ | onlyOwner |
| | setPenaltyTxAmount | External | ✓ | onlyOwner |
| | setErc20TokenAddress | External | ✓ | onlyOwner |
| | setAntiBotEnabled | External | ✓ | onlyOwner |
| | setMaxBotSellCount | External | ✓ | onlyOwner |
| | setBotSellTimeLimit | External | ✓ | onlyOwner |
| | setOperationWallet | External | ✓ | onlyOwner |
| | setMarketingWallet | External | ✓ | onlyOwner |
| | setPoolsLeaderboardWallet | External | ✓ | onlyOwner |
| | setCommunityWallet | External | ✓ | onlyOwner |
| | setTreasuryOneWallet | External | ✓ | onlyOwner |
| | setTreasuryTwoWallet | External | ✓ | onlyOwner |
| | setPenaltyWallet | External | ✓ | onlyOwner |
| | setOperationFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setMarketingFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setPoolsLeaderboardFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setCommunityFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setTreasuryOneFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setTreasuryTwoFeePercent | External | ✓ | onlyOwner feesNotBeingSet |

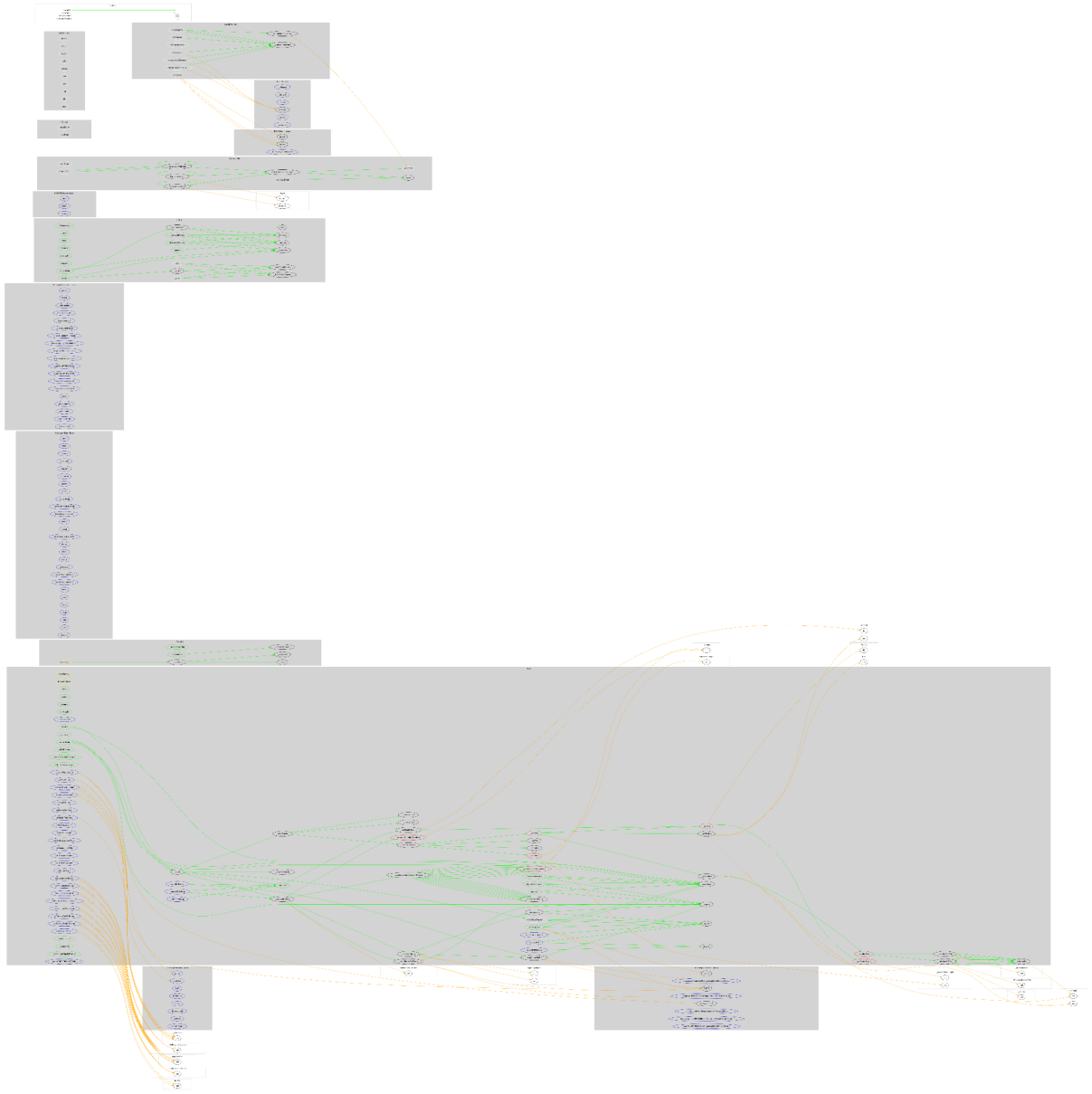
| | | | | |
|---|---|---|---|---|
| | setLiquidityFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | setPenaltyFeePercent | External | ✓ | onlyOwner feesNotBeingSet |
| | _removeAllFees | Private | ✓ | |
| | _restoreAllFees | Private | ✓ | |
| | excludeFromFee | Public | ✓ | onlyOwner |
| | includeInFee | Public | ✓ | onlyOwner |
| | setSwapAndLiquifyEnabled | Public | ✓ | onlyOwner |
| | _calculateFees | Private | | |
| | _calculateBuyFee | Private | | |
| | _calculateSellFee | Private | | |
| | _calculateFee | Private | | |
| | _getCurrentSupply | Private | | |
| | _getRate | Private | | |
| | tokenFromReflection | Public | | - |
| | _reflectFee | Private | ✓ | |
| | _getValues | Private | | |
| | _getTValues | Private | | |
| | _getRValues | Private | | |
| | _calculateERC20TokenFees | Private | | |
| | _calculateLiquidityTokenFees | Private | | |
| | _takeFees | Private | ✓ | |
| | _beforeTokenTransfer | Internal | | |

| | | | | |
|---|---|---|---|---|
| | _checkCanTransfer | Internal | | |
| | _transfer | Internal | ✓ | |
| | _tokenTransfer | Private | ✓ | |
| | _transferStandard | Private | ✓ | |
| | _reflectBot | Private | ✓ | |
| | _swapAndGetFees | Private | ✓ | lockTheSwap |
| | _swapAndLiquify | Private | ✓ | |
| | _calculateAvailableFeesAndTransfer | Private | ✓ | |
| | _transferFeesToWallet | Private | ✓ | |
| | _swapTokensForBnb | Private | ✓ | |
| | _addLiquidity | Private | ✓ | |
| | _swapTokensForTokens | Private | ✓ | |
| | manualBNBSwap | External | ✓ | onlyOwner |
| | manualERC20Swap | External | ✓ | onlyOwner<br>lockTheSwap |
| | autoERC20Swap | External | ✓ | onlyOwner |
| | recoverBNB | External | ✓ | onlyOwner |
| | recoverBNBToWallet | External | ✓ | onlyOwner |
| | recoverERC20Tokens | External | ✓ | onlyOwner |
| | recoverERC20TokensToWallet | External | ✓ | onlyOwner |

# Inheritance Graph



# Flow Graph



# Summary

ACG contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. acg is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error 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 15% buy fees and 25% if the transferred amount a specific threshold that is defined by the contract owner. Lastly, the contract has an antibot throttling mechanism that can prevent the transfers up to 100 blocks.

# 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