Smart Contract Audit Report

CGC Smart Contract

22 NOV 2022

Numen Cyber Labs - Security Services

Table of Content

1 Executive Summary 2

Methodology 2

2 Findings Overview 5

2.1 Project info and Contract address 5

2.2 Summary 6

2.3 Key Findings 7

3 Detailed Description of Findings 7

3.1 Transaction Blacklist Check 8

3.2 Token Lockup Information 10

3.3 Redundant Code 12

4 Conclusion 13

5 Appendix 14

5.1 Basic Coding Assessment 14

5.2 Advanced Code Scrutiny 15

6 Disclaimer 17

References 17

1 EXECUTIVE SUMMARY

Numen Cyber Technology was engaged by CGC to review smart contract implementation. The assessment was conducted in accordance with our systematic approach to evaluate potential security issues based upon customer requirement. The report provides detailed recommendations to resolve the issue and provide additional suggestions or recommendations for improvement.

One high security finding is related to blacklist check during transaction, In addition, there are also Two Informational findings.

The outcome of the assessment outlined in chapter 3 provides the system's owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the underlying technical issue.

METHODOLOGY

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology [10] which is the gold standard in risk assessment using the following risk models:

- Likelihood: represents how likely a particular vulnerability is to be uncovered and exploited in the wild.
- Impact: measures the technical loss and business damage of a successful attack.
- Severity: determine the overall criticality of the risk.

Likelihood and impact are categorized into three ratings: High, Medium and Low. Severity is determined by likelihood and impact and can be classified into four categories accordingly, Critical, High, Medium, Low shown in table 1.1.

Table 1.1: Overall Risk Severity

To evaluate the risk, we will be going through a list of items, and each would be labelled with a severity category. The audit was performed with a systematic approach guided by a comprehensive assessment list carefully designed to identify known and impactful security issues. If our tool or analysis does not identify any issue, the contract can be considered safe regarding the assessed item. For any discovered issue, we might further deploy contracts on our private test environment and run tests to confirm the findings. If necessary, we would additionally build a PoC to demonstrate the possibility of exploitation. The concrete list of check items is shown in Table 1.2.

- Basic Coding Bugs: We first statically analyze given smart contracts with our proprietary static code analyzer for known coding bugs, and then manually verify (reject or confirm) all the issues found by our tool.

- Code and business security testing: We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs.

- Additional Recommendations: We also provide additional suggestions regarding the coding and development of smart contracts from the perspective of proven programming practices.

Category	Assessment Item
Basic Coding	Apply Verification Control

Assessment	Authorization Access Control
	Forged Transfer Vulnerability
	Forged Transfer Notification
	Numeric Overflow
	Transaction Rollback Attack
	Transaction Block Stuffing Attack
	Soft fail Attack
	Hard fail Attack
	Abnormal Memo
	Abnormal Resource Consumption
	Secure Random Number
Advanced Source Code Scrutiny	Asset Security
	Cryptography Security
	Business Logic Review
	Source Code Functional Verification
	Account Authorization Control
	Sensitive Information Disclosure
	Circuit Breaker
	Blacklist Control
	System API Call Analysis

	Contract Deployment Consistency Check
Additional Recommendations	Semantic Consistency Checks
	Following Other Best Practices

Table 1.2: The Full List of Assessment Items

To better describe each issue we identified, we categorize the findings with Common Weakness Enumeration (CWE-699) [14], which is a community-developed list of software weakness types to better delineate and organize weaknesses around concepts frequently encountered in software development.

2 FINDINGS OVERVIEW

2.1 PROJECT INFO AND CONTRACT ADDRESS

Project Name: CGC

Project URL: NULL

Audit Time: 2022/11.17 - 2022/11.22

Language: solidity

Contract Name	Smart Contract Address
CarGaiaToken.sol	https://etherscan.io/address/0x5913dce2041a2607d9ee7d0374b88ad00bec2dc0#code

Token Info:

Token Name	CarGaia Coin
Token Symbol	CGC
Decimals	18
TotalSupply	1 billion(total supply is constant, the current circulation is 167,500,000)
LockTotal	9,832,500,000
Token Type	ERC20

2.2 SUMMARY

Severity	Found	
Critical	0	
High	1	

Medium	0	
Low	0	
Informational	2	

2.3 KEY FINDINGS

One high security finding is related to blacklist check during transaction, in addition, there are also Two Informational findings.

ID	Severity	Findings Title	Status	Confirm
NVE-001	High	Transaction Blacklist Check	Fixed	Confirmed
NVE-003	Informational	Token Lockup Information	Ignore	Confirmed
NVE-004	Informational	Redundant Code	Fixed	Confirmed

Table 2.1: Key Audit Findings

3 DETAILED DESCRIPTION OF FINDINGS

3.1 TRANSACTION BLACKLIST CHECK

ID: NVE-001 Location: CarGaiaToken.sol

Severity: High Category: Business Issues

Likelihood: High

Impact: High

Description:

As shown in figure 1 below, The modifier isBlackListed only checks whether the caller is a blacklist address.

- transfer

As shown in figure 2 below, When the user calls the transfer function to transfer tokens, it only checks whether the caller is a blacklist address, However there is not check on the "_to" address.

- transferFrom

As shown in figure 3 below，When the user calls the transferFrom function to transfer tokens, it only checks whether the caller is a blacklist address, However there is no check on the "_from" address and "_to" address. Even if _form is a blacklist address, tokens can be transferred out.

```
    }

    modifier isBlackListed() {
        require(!blackListed[msg.sender]);
        _;
    }
}
```

Figure 1 modifier function

```
function transfer(address _to, uint256 _value)
    public
    isBlackListed()
    whenNotPaused
    returns (bool)
{
    balances[msg.sender] -= _value;
    balances[_to] += _value;
    emit Transfer(msg.sender, _to, _value);
    return true;
}
```

Figure 2 transfer function

```
    function transferFrom(
        address _from,
        address _to,
        uint256 _value
    ) public isBlackListed whenNotPaused returns (bool) {
        allowed[_from][msg.sender] -= _value;
        balances[_from] -= _value;
        balances[_to] += _value;
        emit Transfer(_from, _to, _value);
        return true;
    }
}
```

Figure 3 transferFrom function

Recommendations:

Numen Cyber Lab recommends to modify the code logic.

Result: PASS

Fix Result: Fixed(After communicating with the project party, the “from” address blacklist check in the transferFrom function has been fixed and the blacklist address can receive tokens as normal logic).

The fixed code is as follows:

```
function transferFrom(address _from, address _to, uint _value) public isBlackListed(_from) whenNotPaused returns (bool) {
    allowed[_from][msg.sender] -= _value;
    balances[_from] -= _value;
    balances[_to] += _value;
    emit Transfer(_from, _to, _value);
    return true;
}
```

3.2 TOKEN LOCKUP INFORMATION

ID: NVE-002 Location:CarGaiaToken.sol

Severity: Informational Category: Business Issues

Likelihood: Informational

Impact: Informational

Description:

As shown in Figure 4 below, the total amount of CGC tokens is 1 billion, which 16.75% are allocated directly, and the remaining 83.25% are locked.

The lock information is as follows:

Lock Address	Percent	Unlock Info
sale(0x875f474417E6f2d393B57E032a0c4397a207C6d6)	18%	After one year, 20% of the locked total amount will be released directly, and remaining 80% daily unlock for 48 months.
community(0x46D83a3e67140F1B080B713da78d884c2076faB8)	11.4%	weekly unlock for 24 months
advisor(0xA26632155fCCA6E855D0364D48E36C59EF04A706)	5%	yearly unlock for 5 years

ecosystem(0xE55d2a665970cEB46a7160846729Aa62119CD919)	19.6%	daily vesting for 24 months
foundation(0x24aD61058d4243535FeC91366283192fBB1Fb9Bc)	14.25%	weekly unlock for 24 months
team(0x09B34152e7cB2Ec41D35B7FB240b307E445D2bCf)	15%	yearly unlock for 5 years

```
    initBalances(_initialSupply, _sale, 3300, 1500);
    //Total: 6% + 6% = 12%   Unlock: 6%*5% + 6%*5% = 0.6%
    initBalances(_initialSupply, _community, 1200, 60);
    //Total: 5%   Unlock: 0
    initBalances(_initialSupply, _advisor, 500, 0);
    //Total: 20%   Unlock: 20%*2% = 0.4%
    initBalances(_initialSupply, _ecosystem, 2000, 40);
    //Total: 15%   Unlock: 15%*5%  = 0.75%
    initBalances(_initialSupply, _foundation, 1500, 75);
    //Total: 15%   Unlock: 0
    initBalances(_initialSupply, _team, 1500, 0);
}

function initBalances(
    uint256 _initialSupply,
    address _owner,
    uint256 _tn,
    uint256 _uln
) private {
    uint256 _total = (_initialSupply * _tn) / 10000;     //12%
    uint256 _value = (_initialSupply * _uln) / 10000;   //15%
    uint256 _lock = _total - _value;//18%
    cliffVesting[_owner].lockTotal = _lock;
    if (_value > 0) {
        balances[_owner] = _value;
        emit Transfer(address(0x0), _owner, _value);
    }
}
```

Figure 4 initBalances function

Recommendations:

Null

Result: Pass

3.3 REDUNDANT CODE

ID: NVE-003 Location:CarGaiaToken.sol

Severity: Informational Category: Business Issues

Likelihood: Informational

Impact: Informational

Description:

As shown in Figure 5 below, the DestroyedBlackFunds event declared in the contract is not used.

```
contract BlackList is Ownable {
    mapping(address => bool) public blackListed;

    event DestroyedBlackFunds(address _blackListedUser, uint256 _balance);
    event AddedBlackList(address _user);

    event RemovedBlackList(address _user);
```

Figure 5 DestroyedBlackFunds event

Recommendations:

Numen Cyber Lab recommends to delete code DestroyedBlackFunds event.

Result: Pass

Fix Result: Fixed

The fixed code is as follows:

```
contract BlackList is Ownable {

    mapping (address => bool) public blackListed;

    event AddedBlackList(address _user);

    event RemovedBlackList(address _user);
```

4 CONCLUSION

In this audit, we thoroughly analyzed **CGC**'s smart contract implementation. The problems found are described and explained in detail in Section 3. The problems found in the audit have been brought up to the project party, ignored issues are in line with the project design, and permissions are only used for the project to properly function. We therefore deem the audit result to be a **PASS.** To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, audit findings, or potential gaps in scope/coverage.

5 APPENDIX

5.1 BASIC CODING ASSESSMENT

5.1.1 Apply Verification Control

- Description: The security of apply verification
- Result: Not found
- Severity: Critical

5.1.2 Authorization Access Control

- Description: Permission checks for external integral functions
- Result: Not found
- Severity: Critical

5.1.3 Forged Transfer Vulnerability

- Description: Assess whether there is a forged transfer notification vulnerability in the contract
- Result: Not found
- Severity: Critical

5.1.4 Transaction Rollback Attack

- Description: Assess whether there is transaction rollback attack vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.5 Transaction Block Stuffing Attack

- Description: Assess whether there is transaction blocking attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.6 soft fail Attack Assessment

- Description: Assess whether there is soft fail attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.7 hard fail Attack Assessment

- Description: Examine for hard fail attack vulnerability
- Result: Not found
- Severity: Critical

5.1.8 Abnormal Memo Assessment

- Description: Assess whether there is abnormal memo vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.9 Abnormal Resource Consumption

- Description: Examine whether abnormal resource consumption in contract processing.
- Result: Not found
- Severity: Critical

5.1.10 Random Number Security

- Description: Examine whether the code uses insecure random number.
- Result: Not found
- Severity: Critical

5.2 Advanced Code Scrutiny

5.2.1 Cryptography Security

- Description: Examine for weakness in cryptograph implementation.
- Results: Not Found
- Severity: High

5.2.2 Account Permission Control

- Description: Examine permission control issue in the contract
- Results: Not Found
- Severity: Medium

5.2.3 Malicious Code Behaviour

- Description: Examine whether sensitive behaviour present in the code
- Results: Not found
- Severity: Medium

5.2.4 Sensitive Information Disclosure

- Description: Examine whether sensitive information disclosure issue present in the code.
- Result: Not found
- Severity: Medium

5.2.5 System API

- Description: Examine whether system API application issue present in the code
- Results: Not found
- Severity: Low

6 DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without Numen'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 considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Numen to perform a security assessment. This report 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 in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. Numen's position is that each company and individual are responsible for their own due diligence and continuous security. Numen'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.

REFERENCES

[1] MITRE. CWE- 191: Integer Underflow (Wrap or Wraparound).
https://cwe.mitre.org/data/ definitions/191.html.

[2] MITRE. CWE- 197: Numeric Truncation Error.
https://cwe.mitre.org/data/definitions/197. html.

[3] MITRE. CWE-400: Uncontrolled Resource Consumption.
https://cwe.mitre.org/data/ definitions/400.html.

[4] MITRE. CWE-440: Expected Behavior Violation.
https://cwe.mitre.org/data/definitions/440. html.

[5] MITRE. CWE-684: Protection Mechanism Failure.
https://cwe.mitre.org/data/definitions/ 693.html.

[6] MITRE. CWE CATEGORY: 7PK - Security Features.
https://cwe.mitre.org/data/definitions/ 254.html.

[7] MITRE. CWE CATEGORY: Behavioral Problems.
https://cwe.mitre.org/data/definitions/438. html.

[8] MITRE. CWE CATEGORY: Numeric Errors.
https://cwe.mitre.org/data/definitions/189.html.

[9] MITRE. CWE CATEGORY: Resource Management Errors.
https://cwe.mitre.org/data/ definitions/399.html.

[10] OWASP. Risk Rating Methodology.
https://www.owasp.org/index.php/OWASP_Risk_ Rating_Methodology

Numen Cyber Technology Pte. Ltd.

11 North Buona Vista Drive, #04-09,

The Metropolis, Singapore 138589

Tel: 65-63555555

Fax: 65-63666666

Email: sales@numencyber.com

Web: https://numencyber.com