Smart Contract Audit Report

Meteorn Run Smart Contract

22 May 2023

Numen Cyber Labs - Security Services

Table of Content

1 Executive Summary

Numen Cyber Technology was engaged by Meteorn Run 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.

Privileged roles can affect the overall operation of the contract, with some variables unrestricted.

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 Assessment	Apply Verification Control
	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: Meteorn Run

Audit Time: 2023/5/21 - 2022/5/22

Language: Solidity

Source Code Link:
https://sepolia.etherscan.io/address/0xc1F8251904A857EB829Abe79E5e6D9F546f6DfA5#code

2.2 Summary

Severity	Found	
Critical	0	
High	0	
Medium	1	
Low	1	
Informational	0	

2.3 Key Findings

There is a medium risk and a low risk.

ID	Severity	Findings Title	Status	Confirm
NVE-001	Medium	Owner Privileged Roles Can Modify Multiple Variables Of The Contract	Confirmed	Ignore
NVE-002	Low	Unrestricted Cooldown Time When Collecting Rewards	Confirmed	Ignore

Table 2.3: Key Audit Findings

3 Detailed Description of Findings

3.1 OWNER PRIVILEGED ROLES CAN MODIFY MULTIPLE VARIABLES OF THE CONTRACT

ID: NVE-001

Location: ConstantStaking.sol

Severity: Medium

Category: Business Issues

Likelihood: Low

Impact: High

Description:

The main function of ConstantStaking contract is collateral mining, the user will collateralize MTO to the contract, after that you can get the collateral reward of the contract, the reward Token is GMTO, the owner privileged role can modify multiple variables, such as collateral Token, reward Token, reward factor, if the privileged role is maliciously controlled, it may cause security risk. For example, if the user pledges the real MTO Token into the contract, the privileged role will set the MTO as a malicious Token, the initial pledged MTO Token will not be taken away by the user, and if the reward contract is modified, there will be no reward funds.

```
    function setAPY(uint256 _apy) external onlyOwner {
        require(_apy <= MAX_APY, "ConstantStaking: out of range");
        apy = _apy;
    }

    function setMTO(address _mto) external onlyOwner {
        MTO = _mto;
    }

    function setGMTO(address _gmto) external onlyOwner {
        GMTO = _gmto;
    }

    function setClaimInterval(uint256 _ts) external onlyOwner {
        claim_interval = _ts;
    }
```

Figure 1 owner privileged role

Recommendations:

It is recommended that owner privileged roles be managed using multiple signatures or time locks.

Result: **Pass**

Fix Result: Ignore

3.2 UNRESTRICTED COOLDOWN TIME WHEN COLLECTING REWARDS

ID: NVE-002

Location: ConstantStaking.sol

Severity: Low

Category: Business Issues

Likelihood: Low

Impact: Medium

Description:

applyClaim method is used to record the current user's balance and rewards and cooling time, during the cooling time, the user will not be able to take away the principal of the rewards, the default cooling time in the contract is 7 days, which is longer, and there is a setClaimInterval method in the contract to modify the cooling time, when the cooling time is modified to a larger value, the user's rewards and principal will not be taken out in time.

```
    function applyClaim() external {
        uint256 allMTO = calculateMTO(msg.sender);
        require(allMTO > 0, "ConstantStaking: no staking");
        // IERC20(MTO).transfer(msg.sender, allMTO);
        uint256 allRewards = calculateRewards(msg.sender);
        require(allRewards > 0, "ConstantStaking: no rewards");
        Claim storage claimInfo = claims[msg.sender];
        claimInfo.amount = allMTO;
        claimInfo.reward = allRewards;
        claimInfo.startTs = block.timestamp + claim_interval;
        delete stakings[msg.sender];
        emit ClaimApplied(msg.sender, allMTO, allRewards);
    }

    function claim() external {
        Claim memory claimInfo = claims[msg.sender];
        require(claimInfo.amount > 0, "ConstantStaking: no apply");
        require(block.timestamp >= claimInfo.startTs,"ConstantStaking: too early");

        IERC20(MTO).transfer(msg.sender, claimInfo.amount);
        if (claimInfo.reward > 0) {
            IGMTO(GMTO).mint(msg.sender, claimInfo.reward);
        }
        // clear data
        delete claims[msg.sender];
        total -= claimInfo.amount;
        emit Claimed(msg.sender, claimInfo.amount, claimInfo.reward);
    }
```

```
function setClaimInterval(uint256 _ts) external onlyOwner {
    claim_interval = _ts;
}
```

Figure 2 claim_interval variable

Recommendations:

It is recommended to set the maximum and minimum value limits for the claim_interval variable to avoid the variable value being too large and the principal and reward not being taken out in time.

Result: **Pass**

Fix Result: Ignore

4 Conclusion

In this audit, we thoroughly analyzed **Meteorn Run** smart contract implementation. The problems found are described and explained in detail in Section 3. The issues identified in the audit have been raised with project leaders, one medium risk and one low risk, and require timely adjustment.We therefore consider the audit result to be **Passed**.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