Smart Contract Audit Report

DEXYNTH Smart Contract

20 Mar 2023

Numen Cyber Labs - Security Services

Table of Content

1 EXECUTIVE SUMMARY

Numen Cyber Technology was engaged by DEXYNTH 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 Medium severities findings is related to owner authority, centralized risk. One Information severities findings is related to logical judgment.One Low severities findings is related to Computational problems.

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: DEXYNTH

Project URL: https://github.com/Global-Repo/Synths

Audit Time: 2023/3.13 - 2023/3.20

Language: solidity

Commit Hash: f9780ec12e907d1d8036c21020fc964ca109b4ac

Contract Name	Source Code Link
GFarmNft1.sol	https://github.com/Global-Repo/Synths/contracts/GFarmNft1.sol
GFarmNft2.sol	https://github.com/Global-Repo/Synths/contracts/GFarmNft2.sol
GFarmNft3.sol	https://github.com/Global-Repo/Synths/contracts/GFarmNft3.sol
GFarmNft4.sol	https://github.com/Global-Repo/Synths/contracts/GFarmNft4.sol
GFarmNft5.sol	https://github.com/Global-Repo/Synths/contracts/GFarmNft5.sol
GFarmTradingStorageV5.sol	https://github.com/Global-Repo/Synths/contracts/GFarmTradingStorageV5.sol
GNSNftRewardsV6.sol	https://github.com/Global-Repo/Synths/contracts/GNSNftRewardsV6.sol
GNSPairInfosV6_1.sol	https://github.com/Global-Repo/Synths/contracts/GNSPairInfosV6_1.sol
GNSPairsStorageV6.sol	https://github.com/Global-Repo/Synths/contracts/GNSPairsStorageV6.sol
GNSPoolV5.sol	https://github.com/Global-

	Repo/Synths/contracts/GNSPoolV5.sol
GNSPriceAggregatorV6_2.sol	https://github.com/Global-Repo/Synths/contracts/GNSPriceAggregatorV6_2.sol
GNSPriceAggregatorV6_2_Bypassed.sol	https://github.com/Global-Repo/Synths/contracts/GNSPriceAggregatorV6_2_Bypassed.sol
GNSReferralsV6_2.sol	https://github.com/Global-Repo/Synths/contracts/GNSReferralsV6_2.sol
GNSStakingV6_2.sol	https://github.com/Global-Repo/Synths/contracts/GNSStakingV6_2.sol
GNSTradingV6_2.sol	https://github.com/Global-Repo/Synths/contracts/GNSTradingV6_2.sol
GNSTradingVaultV5.sol	https://github.com/Global-Repo/Synths/contracts/GNSTradingVaultV5.sol
V6.3	https://github.com/Global-Repo/Synths/contracts/V6.3

2.2 SUMMARY

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

2.3 KEY FINDINGS

One Medium severities findings is related to owner authority, centralized risk. One Information severities findings is related to logical judgment.One Low severities findings is related to Computational problems.

ID	Severity	Findings Title	Status	Confirm
NVE-001	Medium	Gov has higher authority	Ignore	Confirmed
NVE-002	Low	Computational problems	Ignore	Confirmed
NVE-003	Information	logical judgment	Ignore	Confirmed

Table 2.1: Key Audit Findings

3 Detailed Description of Findings

3.1 Gov has higher authority

ID: NVE-001 Location: GNSPoolV5.sol

Severity: Medium Category: Authority Issues

Likelihood: Medium

Impact: Medium

Description:

The Gov address in the GNSPoolV5 contract has the highest permissions, and functions such as setToken, setLp, and setBoostsP. These can set key parameters in the contract, and malicious parameter settings can affect the normal use of the contract.

```
function setToken(TokenInterfaceV5 _token) external onlyGov{
    require(address(_token) != address(0), "ADDRESS_0");
    require(address(token) == address(0), "ALREADY_SET");
    token = _token;
    emit AddressUpdated("token", address(_token));
}

function setLp(LpInterfaceV5 _lp) external onlyGov{
    require(address(_lp) != address(0), "ADDRESS_0");
    require(address(lp) == address(0), "ALREADY_SET");
    lp = _lp;
    emit AddressUpdated("lp", address(_lp));
}

function addAllowedContract(address c) external onlyGov{
    require(c != address(0), "ADDRESS_0");
    require(token.hasRole(MINTER_ROLE, c), "NOT_MINTER");
    allowedContracts[c] = true;
    emit ContractAllowed(c, true);
}
function removeAllowedContract(address c) external onlyGov{
    require(c != address(0), "ADDRESS_0");
    allowedContracts[c] = false;
    emit ContractAllowed(c, false);
}
function setBoostsP(uint _bronze, uint _silver, uint _gold, uint _platinum, uint _diamond) external onlyGov{
    require(_bronze < _silver && _silver < _gold && _gold < _platinum && _platinum < _diamond && _bronze > 0, "W
    boostsP = [_bronze, _silver, _gold, _platinum, _diamond];
    emit BoostsUpdated(boostsP);
}
```

Figure 1 Some functions that only gov can call

Recommendations:

In addition to the GNSPoolV5 contract, GFarmTradeStorageV5, GNSNftRewardsV6, and GNSPairInfosV6_ 1， GNSPairsStorageV6， GNSPriceAggregatorV6_ 2，GNSPriceAggregatorV6_ 2_ Bypassed，GNSReferralsV6_ 2，GNSStakingV6_ 2，GNSTradingV6_ 2. The highest permissions for GNSTradingVaultV5 contract are also the “gov" address.

Numen Cyber Lab recommends properly manage the gov address, and multiple signature or time lock contracts can be introduced to serve as the gov role.

Result: Pass

Fix Result:

Ignore.

3.2 COMPUTATIONAL PROBLEMS

ID: NVE-002

Location: GNSPoolV5.sol

Severity: Low

Category: Computational problems

Likelihood: Low

Impact: Low

Description:

Users can pass in the referral address when staking and will also allocate a portion to the referral when calculating the reward. the referral address is 0, the user will receive 94% of the reward. the referral address is not 0, the user will receive 97%, and the referrer will receive 3%. When the referral address is not 0, the user stake will get 3% more rewards.

```
function harvest() public{
    if(lpBalance == 0){ return; }

    User storage u = users[msg.sender];

    uint pendingTokens = pendingRewardToken();

    if(pendingTokens > 0){
        if(u.referral == address(0)){
            token.mint(msg.sender, pendingTokens - pendingTokens * referralP / 100); //94
        }else{
            uint referralReward = pendingTokens * referralP / 200;

            token.mint(msg.sender, pendingTokens - referralReward); //97
            token.mint(u.referral, referralReward);//3

            users[u.referral].referralRewardsToken += referralReward;
        }
    }

    u.debtToken = (u.provided + u.totalBoost) * accTokensPerLp / 1e18;
}

// Stake LPs
function stake(uint amount, address referral) external{
    User storage u = users[msg.sender];

    // 1. Transfer the LPs to the contract
    lp.transferFrom(msg.sender, address(this), amount);

    // 2. Harvest pending rewards
    harvest();

    // 3. Reset lp balance
    lpBalance -= (u.provided + u.totalBoost);

    // 4. Set user provided
    u.provided += amount;

    // 5. Set boosts and debt
    setBoosts();

    // 6. Update lp balance
    lpBalance += (u.provided + u.totalBoost);

    // 7. Set referral
    if(u.referral == address(0) && referral != address(0) && referral != msg.sender){
        u.referral = referral;
    }
```

Figure 2 function stake,function harvest

Result: Pass

Fix Result:

Ignore

3.3 LOGIC JUDGMENT

ID: NVE-003

Location: GNSPoolV5.sol

Severity: Information

Category: Logic Judgment

Likelihood: Information

Impact: Information

Description:

When calling stakeNft, there will be a structure record. The variable u.stakedNftsCount will record the user's mortgage quantity, starting from 0. UnstakeNft unlocks the pledge, requiring an nftIndex to be passed in to indicate the subscript. It should be added to determine whether the passed in nftindex is less than u.stakedNftsCount. When calling unstakeNft, you need to judge nftIndex < userNfts[msg.sender][u.stakedNftsCount].

```
function stakeNft(uint nftType, uint nftId) external notContract{
    User storage u = users[msg.sender];

    // 0. If didn't already stake NFT + nft type is either platinum or diamond
    require(u.stakedNftsCount < maxNftsStaked, "MAX_NFTS_ALREADY_STAKED");
    require(nftType >= 1 && nftType <= 5, "WRONG_NFT_TYPE");

    // 1. Transfer the NFT to the contract
    require(getnfts()[nftType-1].balanceOf(msg.sender) >= 1, "NOT_NFT_OWNER");
    getnfts()[nftType-1].transferFrom(msg.sender, address(this), nftId);

    // 2. Harvest pending rewards
    harvest();

    // 3. Reset lp balance
    lpBalance -= (u.provided + u.totalBoost);

    // 4. Store NFT info
    StakedNft storage stakedNft = userNfts[msg.sender][u.stakedNftsCount];
    stakedNft.nftType = nftType;
    stakedNft.nftId = nftId;
    u.stakedNftsCount ++;

    // 5. Set user boosts & debt
    setBoosts();

    // 6. Update lp balance
    lpBalance += (u.provided + u.totalBoost);
}

// Unstake NFT
function unstakeNft(uint nftIndex) external{
    User storage u = users[msg.sender];
    StakedNft memory stakedNft = userNfts[msg.sender][nftIndex];

    // 1. Harvest pending rewards
    harvest();

    // 2. Reset lp balance
    lpBalance -= (u.provided + u.totalBoost);

    // 3. Remove NFT from storage => replace by last one and remove last one
    userNfts[msg.sender][nftIndex] = userNfts[msg.sender][u.stakedNftsCount-1];
    delete userNfts[msg.sender][u.stakedNftsCount-1];
    u.stakedNftsCount -= 1;
```

Figure 3 function stakeNft,function unstakeNft

Recommendations:

Numen Cyber Lab recommends determine whether the incoming nftindex is less than u.stakedNftsCount.

Result: Pass

Fix Result:

Ignore.

4 CONCLUSION

In this audit, we thoroughly analyzed DEXYNTH 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