METATRUST

Security Assessment for

NodeDAO 3

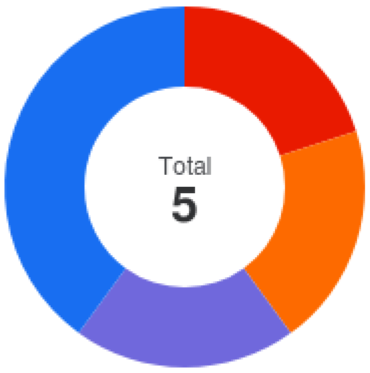
May 5, 2023

Executive Summary

Overview	
Project Name	NodeDAO 3
Codebase URL	https://github.com/NodeDAO/NodeDAO-Protocol
Scan Engine	Security Analyzer
Scan Time	2023/05/5 18:24:36
Commit Id	commit:3d0a9c42

Critical Issues	The issue can cause large economic losses, large-scale data disorder, loss of control of authority management, failure of key functions, or indirectly affect the correct operation of other smart contracts interacting with it.
High Risk Issues	The issue puts a large number of users' sensitive information at risk or is reasonably likely to lead to catastrophic impacts on clients' reputations or serious financial implications for clients and users.
Medium Risk Issues	The issue puts a subset of users' sensitive information at risk, would be detrimental to the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk Issues	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational Issue	The issue does not pose an immediate risk but is relevant to security best practices or Defence in Depth.

Total	
Critical Issues	1
High risk Issues	1
Medium risk Issues	0
Low risk Issues	1
Informational Issues	2



Critical Issues	20%	1
High risk Issues	20%	1
Medium risk Issues	0%	0
Low risk Issues	20%	1
Informational Issues	40%	2

Summary of Findings

MetaScan security assessment was performed on **May 5, 2023 18:24:36** on project **NodeDAO 3** with the repository **https://github.com/NodeDAO/NodeDAO-Protocol** on branch **main**. The assessment was carried out by scanning the project's codebase using the scan engine **Security Analyzer**. There are in total **5** vulnerabilities / security risks discovered during the scanning session, among which **1** critical vulnerabilities, **1** high risk vulnerabilities, **0** medium risk vulnerabilities, **1** low risk vulnerabilities, **2** informational issues.

ID	Description	Severity	Alleviation
MSA-001	The wrong value passed to validatorNumber of the _delaySlash function	Critical	Fixed
MSA-002	Lack of validate `number` in the `setNftExitBlockNumbers` function	High risk	Fixed
MSA-003	Missing emit event	Low risk	Fixed
MSA-004	Typo	Informational	Fixed
MSA-005	Gas Optimization	Informational	Fixed

Findings

Critical (1)

1. The wrong value passed to validatorNumber of the _delaySlash function — Critical — Security Analyzer

In the `OperatorSlash` contract, the `slashOfExitDelayed` function punishes operators who fail to exit on time. The `slashOfExitDeplayed` function gets `claimEthAmount` for the specified `requestId`, then calls the `_delaySlash` function to do the slash, but the value passed to the validatorNumber parameter of the `_delaySlash` function is `claimEthAmount % 32 ether`, instead of `claimEthAmount / uint256(32 ether)`. Per the definition of the `_delaySlash` function as shown below:

```
function _delaySlash(uint256 _operatorId, uint256 _startNumber, uint256 validatorNumber) internal {
    ...
}
```

We can also infer that the intended value passed to`validatorNumber` should not be `claimEthAmount % 32 ether`.

File(s) Affected

src/OperatorSlash.sol #134-147

```
        for (uint256 i = 0; i < _largeExitDelayedRequestIds.length; ++i) {
            uint256 requestId = _largeExitDelayedRequestIds[i];
            uint256 operatorId = 0;
            uint256 withdrawHeight = 0;
            uint256 claimEthAmount = 0;
            (operatorId, withdrawHeight,,, claimEthAmount,,) =
                withdrawalRequestContract.getWithdrawalOfRequestId(requestId);
            uint256 startNumber = withdrawHeight;
            if (largeExitDelayedSlashRecords[requestId] != 0) {
                startNumber = largeExitDelayedSlashRecords[requestId];
            }
            largeExitDelayedSlashRecords[requestId] = block.number;
            _delaySlash(operatorId, startNumber, claimEthAmount % 32 ether);
        }
```

Recommendation

Recommend passing the right value to the parameter validatorNumber of the `_delaySlash` function.

Alleviation Fixed

The development team resolved this issue in the commit https://github.com/NodeDAO/NodeDAO-Protocol/commit/418e010e63a996ca50c91b11abc9ae29539c56c2

High risk (1)

1. Lack of validate `number` in the `setNftExitBlockNumbers` function

High risk

Security Analyzer

In the `setNftExitBlockNumbers` function, the number will be stored in the `userNftExitBlockNumbers` map for `tokenId`, that exited. Meanwhile, the corresponding map `operatorExitButNoBurnNftCounts` for `operatorId` and `totalExitButNoBurnNftCounts` will increase by 1. Correspondingly, in the `whiteListBurn` function, the corresponding `operatorExitButNoBurnNftCounts` map for the `operatorId` and `totalExitButNoBurnNftCounts` need to be decreased by 1 if the burnt token is an exited token.

However, if the number assigned to `userNftExitBlockNumbers` for `tokenId` is zero with the `setNftExitBlockNumbers` function, it will still result in the increment of `operatorExitButNoBurnNftCounts` and `totalExitButNoBurnNftCounts`, but, as a result, the `whiteListBurn` function fails to decrease the `operatorExitButNoBurnNftCounts` and `totalExitButNoBurnNftCounts` since the `userNftExitBlockNumbers[_tokenId]` is zero.

File(s) Affected

src/tokens/VNFT.sol #460-460

```
460              if (number > block.number) revert InvalidBlockHeight();
```

Recommendation

Recommend adding a check to prevent the number to be zero in the setNftExitBlockNumbers function:

```
if (number > block.number || number == 0) revert InvalidBlockHeight();
```

Alleviation Fixed

The development team resolved this issue in the commit https://github.com/NodeDAO/NodeDAO-Protocol/commit/a0f7d8786c80dcc848b44aca6ac5c0c1bd8187e8

Medium risk (0)

No Medium risk vulnerabilities found here

Low risk (1)

1. Missing emit event

Low risk Security Analyzer

In the `_slash` function, an `OperatorArrearsIncrease` event needs to be emitted once the `operatorSlashAmountOwed` increases. However, in the final else branch, there is no OperatorArrearsIncrease event to be emitted.

File(s) Affected

src/registries/NodeOperatorRegistry.sol #641-646

```
        } else {
            operatorSlashAmountOwed[_operatorId] += _amount - pledgeAmounts;
            operatorPledgeVaultBalances[_operatorId] = 0;
            emit Slashed(_operatorId, pledgeAmounts);
            return pledgeAmounts;
        }
```

Recommendation

Recommend emitting the `OperatorArrearsIncrease` event in the else branch in the `_slash` function.

Alleviation Fixed

The development team resolved this issue in the commit https://github.com/NodeDAO/NodeDAO-Protocol/commit/8295963ffa703ce89ea0c9bb34f6ee187c77c93a

Informational (2)

1. Typo

Informational Security Analyzer

Per the meaning of the state variable `userActiceNftCounts` and its function `getUserActiveNftCountsOfOperator`, the name of the state variable `userActiceNftCounts` should be `userActiveNftCounts`.

File(s) Affected

src/tokens/VNFT.sol #531-533

```
    function getUserActiveNftCountsOfOperator(uint256 _operatorId) external view returns (uint256) {
        return userActiceNftCounts[_operatorId];
    }
```

Recommendation

Recommend fixing the typo.

Alleviation Fixed

The development team resolved this issue in the commit https://github.com/NodeDAO/NodeDAO-Protocol/commit/a725d04618274e04649067d18e8aed07182e60fd

2. Gas Optimization

Informational Security Analyzer

To avoid re-assigning **`isSettle`** to true, the **`isSettle`** variable could be checked in the **`if`** branch.

File(s) Affected

src/vault/VaultManager.sol #180-198

```
    function _elSettle(uint256[] memory _operatorIds) internal returns (uint256[] memory, bool) {
        uint256[] memory reinvestAmounts = new uint256[] (_operatorIds.length);
        uint256[] memory operatorElComissionRate;
        operatorElComissionRate = nodeOperatorRegistryContract.getOperatorComissionRate(_operatorIds);

        bool isSettle = false;
        for (uint256 i = 0; i < _operatorIds.length; ++i) {
            uint256 operatorId = _operatorIds[i];
            address vaultContractAddress = nodeOperatorRegistryContract.getNodeOperatorVaultContract(op

            uint256 _reinvest = _settle(operatorId, vaultContractAddress, operatorElComissionRate[i]);
            if (_reinvest > 0) {
                isSettle = true;
            }
            reinvestAmounts[i] = _reinvest;
        }

        return (reinvestAmounts, isSettle);
    }
```

Recommendation

Recommend checking the isSettle before assigning value to it.

```
            if (!isSettle && _reinvest > 0) {
                isSettle = true;
            }
```

Alleviation Fixed

The development team acknowledged this issue.

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 you ("Customer" or 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, nor may copies be delivered to any other person other than the Company, without MetaTrust's prior written consent in each instance.

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 MetaTrust 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. MetaTrust's position is that each company and individual are responsible for their own due diligence and continuous security. MetaTrust'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 MetaTrust is subject to dependencies and 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.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS Security Assessment AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, MetaTrust HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, MetaTrust SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, MetaTrust MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, MetaTrust PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER MetaTrust NOR ANY OF MetaTrust'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. MetaTrust WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT MetaTrust'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING Security Assessment MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST MetaTrust WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF MetaTrust CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST MetaTrust WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.