



Nation3 / GoldmanDAO



Nation3 Launch - Report



A good auditor never makes mistakes

▼ Table of contents

[1 Report Summary](#)

[1.1 Introduction](#)

[1.2 Scope](#)

[2 Codebase overview](#)

[2.1 Description](#)

[2.2 Interactions / Integrations](#)

[3 Audit Summary](#)

[3.1 Test Approach & Methodology](#)

[3.2 Summary of findings](#)

[3.3 Summary of recommendations](#)

[Changes made](#)

[4 Findings & Tech Details](#)

[SOLVED - GOLDIS-001 || Controller has unlimited mint](#)

[SOLVED - GOLDIS-002 || Owner has unlimited mint](#)

[ACKNOWLEDGED - GOLDIS-003 || Airdrop claim has no deadline](#)

[Manual Testing](#)

[Automated Testing](#)

[Unit and Integration Tests](#)

[Static Analysis Report](#)

[Automated Security Scans Report](#)

1 Report Summary


1.1 Introduction

Nation3, it's time to re-invent the nation state.

Nation3DAO has contacted GoldmanDAO to audit their new token, plus the additions they've done to the Voting Escrow contract. @Carlos González Juárez conducted the

review of the source code provided. Details on the scope and findings are collected in this document.

The token is a version of Solmate's code, that has been audited before by Fixed Point and used by protocols like Olympus in their coin \$OHM. Here's the link to it:

 <https://github.com/Rari-Capital/solmate/blob/main/audits/v6-Fixed-Point-Solutions.pdf>

The additions made to this contract were to add the possibility to the DAO to mint new tokens based on proposals created by its members.

1.2 Scope

Code freeze done in commit:

<https://github.com/nation3/app/commit/435f6e03068095d5794b640bfe0c29f79eec1816>

Contracts of libraries called by these files will only be audited in the interaction between them, and for correct usage of the functions written by them. Code not in the "contracts" directory at that specific commit above will not be audited nor checked for correct functionality as it's considered out of the scope.

Fixed Point audit for the token was already listed in the heading of this document and should be followed for further investigation into the token contract.

2 Codebase overview

2.1 Description

Code tree **to be reviewed** consists of:

- Distributors:
 - MerkleDistributor.sol:
- Governance:
 - IVotingEscrow.sol
 - VotingEscrow.vy
- Tokens:
 - NATION.sol
- Utils
 - Controlled.sol

2.2 Interactions / Integrations

This code uses two main libraries:


- Solmate: Token already mentioned in the header of this document. Controlled is also inheriting from OpenZeppelin with minimal changes, no issues found in its modifications. This modifications allow the protocol to have one "owner" and one "controller", the first role higher than the second.
- Curve's VotingEscrow contract: To find the audits done to the contract, please follow the link: <https://curve.fi/audits>. The changes made to this contract were superfluous, and tested and checked in this audit.

3 Audit Summary

3.1 Test Approach & Methodology

Issues found by GoldmanDAO rank based on a risk assessment methodology based on the likelihood of the event happening as well as the impact the issue might have in the project.

Likelihood	Description
1	Unlikely
2	Low probability
3	Probable in the long term
4	High prob
5	This shit is happening

Impact	Description
1	Unnoticeable
2	Temporary impact
3	Partial or spread loss
4	Significant
5	

The final risk level will be calculated using a sum of these values, making a final value of 1 to 10.

3.2 Summary of findings

The table below summarizes the findings of the audit, arranged by impact and likelihood.

Critical	High	Medium	Low	Informational
0	0	0	2	1

Impact	5					
	4	G-001, G-002				
	3					
	2					
	1		G-003			
		1	2	3	4	5
		Likelihood				

3.3 Summary of recommendations

- Burn or transfer to a contract the NATION controller
- Setting the time limit for the TweetDrop in a Nation3DAO official message or changing the contract to have a limit for claims.

Changes made

The owner and controller of the token have been set to the Nation3 DAO (powered by Aragon) at the address → [0x7b81e8d4e82796c9b76284fa4d21e57b8b86a06c](#). Since the token distribution to founders is low, it's safe to assume that a take over is unlikely since a bad actor would need to vest their tokens in order to hack the DAO into it, making infeasible to be profitable while massively costly at the same time.

4 Findings & Tech Details

SOLVED - GOLDIS-001 || Controller has unlimited mint

Severity: Low

Contract: NATION.sol

Likelihood: 1

Impact: 4

The current implementation gives the controller of the NATION unlimited power to money printing. Unless controller is burned at some specified point, this is a dangerous functionality to have and rely on, since accounts can be lost, hack or owners can misbehave.

Recommendation is set it up in the deployment the burning of the controller or write down what's needed for the controller to be destroyed since right now there's no documentation about that. If this functionality is needed to give more tokens, it should be automatised by a contract/DAO and not being free for a person to tamper it.

SOLVED - GOLDIS-002 || Owner has unlimited mint

Severity: Low

Contract: NATION.sol

Likelihood: 1

Impact: 4

The current implementation gives the owner of the NATION unlimited power to money printing. Unless owner is burned at some specified point, this is a dangerous functionality to have and rely on, since accounts can be lost, hack or owners can misbehave. The way to do so for the owner is by changing the controller to an account controlled by him, and then printing infinite tokens.

Recommendation is set it up in the deployment the burning of the owner or write down what's needed for the controller to be destroyed since right now there's no documentation about that. If this functionality is needed to give more tokens, it should be automatised by a contract/DAO and not being free for a person to tamper it.

ACKNOWLEDGED - GOLDIS-003 || Airdrop claim has no deadline

Severity: Informational

Contract: MerkleDistributor.sol

Likelihood: 2

Impact: 1

Since there's no limit on when someone can claim the airdropped tokens, that liquidity is going to be locked there forever. You could un-approve and that would stop the airdropping, but might be a bit ugly and with bad contract UX.

Recommendation is set up a time limit or inform the users on when the unapproval will happen so you can put those tokens in good use.

Manual Testing

Manual testing was done by deploying the contract in testnet networks and seeing its functionality functioning in accordance with 3rd party systems like Etherscan and Metamask.

The code has a good unit and integration test coverage in its codebase, no further exploration was needed from one part in that sense. Find the test in its respective section of the repository:

```
https://github.com/nation3/app/tree/master/contracts/contracts/test
```

Automated Testing

Unit and Integration Tests

No further work was required in this section.

Static Analysis Report

The tool used for this report was the framework Slither, that runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses. Slither enables developers to find vulnerabilities.

No issues of significance were found by the usage of this tool.

Automated Security Scans Report

The framework used was MythX which allows finding possible vulnerabilities in the code.

No issues of significance were found by the usage of this tool.