THE ARCADIA GROUP

AUDIT

arcadiamgroup.com

Table of Contents

Executive Summary

A Representative Party of the N3rd Finance ("N3rd") engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following N3rd smart contracts on the N3rd repo at Commit #`2b35940485d73ad7ef3c41b0197d9b13a708ff2a`.

NerdVault.sol

The scope of this audit was to review the logic inside the vault and its security permissions.

Gas optimization was not included in this report.

Findings

setFeeReciever() can't be called if devaddr is different from owner address.

NERD-1
Severity:
Impact: Low

Contract: NerdVault.sol
Category: Informational
Finding Type: Dynamic
Lines: 939-942

This function can only be called by the governance (assuming owner is the governance contract as described in the contract comments), in case devaddr is different from owner address, nobody will be able to update it because only the governance/owner can call it but with

```
require(devaddr == msg.sender, "only dev can change");
```

it will never be completed causing this function to be useless unless ownership transfer occurs.

If the goal is to only allow governance to update it, then the 'require' statement should be removed.

If the goal is to allow devaddr to update by itself then the onlyOwner modifier should be removed (not recommended)

The actual code implies that if the *devaddr* is different from the governance contract, the only way to update the *devaddr* is to transfer ownership of NerdVault.sol to *devaddr*.

The relevance of this issue exists in the fact that *devaddr* may need to be different from the onlyOwner so logic should be updated as needed.

```
function setDevFeeReciever(address _devaddr) public onlyOwner {
require(devaddr == msg.sender, "only dev can change");
tentativeDevAddress = _devaddr;
}
```

devaddr and tentativeDevAddress cannot be updated.

NERD-2
Severity: Medium
Impact: Medium

Contract: NerdVault.sol
Category: Informational
Finding Type: Dynamic
Lines: 944-948

Due to setDevFeeReceiver() it is impossible to update the tentativeDevAddress, no one can succeed in triggering this function unless the owner is devaddr.

Depending on the chosen behaviour while fixing finding #1, it will affect this function , if governance is the only one which can update tentativeDevAddress then 'require' statement should be removed and the modifier onlyOwner has to be added, as an extra step. While if this function is meant to be as a confirmation from the devs, can stay as it is. This issue does not impact key user functionality but should be updated depending on the intended developer and management functionality.

```
function confirmDevAddress() public {
require(tentativeDevAddress == msg.sender, "not tentativeDevAddress!");
devaddr = tentativeDevAddress;
tentativeDevAddress = address(0);
```

```
}
```

newSuperAdmin() function should be controlled by governance contract/owner

NERD-3
Severity: Medium
Impact: Medium

Contract: NerdVault.sol
Category: Informational
Finding Type: Dynamic
Lines: 967-970

This function updates the _superAdmin variable but should be available to be called by governance contract too. With the current code, if the superAdmin is burnt (only by itself for now), there will be no possibility to set a new _superAdmin. While it is unlikely that the superAdmin would be burnt, it is important to have the coverage for the unlikely event.

```
function newSuperAdmin(address newOwner) public virtual onlySuperAdmin {
require(
newOwner != address(0),
"Ownable: new owner is the zero address"
);
emit SuperAdminTransfered(_superAdmin, newOwner);
_superAdmin = newOwner;
}
```

SuperAdmin cannot be reassigned after burnSuperAdmin()

NERD-4
Severity: Medium
Impact: Medium

Contract: NerdVault.sol
Category: Informational
Finding Type: Dynamic
Lines: 972-979

This function assigns _superAdmin to the address zero, but there's no function to assign a new _superAdmin since the only function that allows that which is *newSuperAdmin()* has an onlySuperAdmin modifier which doesn't allow to update it if it has been burnt. If this is an expected behaviour, this function should allow the governance contract to do so too, or even better only governance/owner for security reasons.

```
function burnSuperAdmin() public virtual onlySuperAdmin {
emit SuperAdminTransfered(_superAdmin, address(0));
_superAdmin = address(0);
}
```

claimLPTokens() should be internal

NERD-5
Severity: Low
Impact: Low

Contract: NerdVault.sol
Category: Informational
Finding Type: Dynamic
Lines: 364

This function is called internally by other functions but doesn't seem to have a real use to require a 'public' identifier. If the design doesn't require it set it as internal. This issue does not impact users.

```
function claimLPToken() public {
```

```
if (!genesisLPClaimed) {
if (nerd.isLPGenerationCompleted()) {
genesisLPClaimed = true;
uint256 totalMinted = nerd.getReleasableLPTokensMinted();
poolInfo[0].token.safeTransferFrom(
address(nerd),
address(this),
totalMinted
);
}
}
}
```

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.

Document Information

Title	N3rd NerdVault Audit
Client	N3rd Finance
Auditor(s)	Andrea Burzi
Reviewed by	Joel Farris
Approved by	Rasikh Morani
Contact Details	Rasikh Morani (972) 543-3886 rasikh@arcadiamgroup.com https://t.me/thearcadiagroup