Audit of Falcon Finance

Audit of Falcon Finance's Unipool Smart Contract

a report of findings by

Connor Martin

innovative fortuna iuvat

October 29th, 2020

Table of Contents

Document Info

Client	Falcon Finance
Title	Smart Contract Audit
Auditors	Connor Martin
Reviewed By	Joel Farris
Approved By	Rasikh Morani

Contact

For more information on this report, contact The Arcadia Media Group Inc.

Rasikh Morani
(972) 543-3886
rasikh@arcadiamgroup.com
https://t.me/thearcadiagroup

Executive Summary

A Representative Party of Falcon Finance engaged The Arcadia Group ("Arcadia"), a software development, research and security company, to conduct a review of the following Unipool smart contract on the Falcon Finance repo at Commit #aeeebb97269a6a958acc3b505e489c1871b48e15.

Unipool.sol

Arcadia completed the review using various methods primarily consisting of dynamic and static analysis. This process included a line by line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

Conclusion

No functional differences from the synthetix staking contracts were found apart from solc updates to address breaking changes in version 0.6.0. Additional code review and audits should be completed prior to launch in order to be maximally effective and to lower end-user risk.

Findings

1. A floating pragma is set.

- Unipool.sol
- Severity: Low
- Likelihood: Low
- Impact: Low

- Target: Unipool.sol
- Category: Low
- Finding Type: Dynamic
- Lines: 621

The current pragma Solidity directive is "^0.6.0". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

```
pragma solidity ^0.6.0;

contract LPTokenWrapper {
    using SafeMath for uint256;
    using SafeERC20 for IERC20;

    IERC20 public uni = IERC20(0x50e3D53b4a22e94ee1cE5c3A852D94d145d5852e);

    uint256 private _totalSupply;
    mapping(address => uint256) private _balances;

    function totalSupply() public view returns (uint256) {
        return _totalSupply;
    }

    function balanceOf(address account) public view returns (uint256) {
        return _balances[account];
    }

    function stake(uint256 amount) public virtual {
        _totalSupply = _totalSupply.add(amount);
        _balances[msg.sender] = _balances[msg.sender].add(amount);
        uni.safeTransferFrom(msg.sender, address(this), amount);
    }
```

```
    function withdraw(uint256 amount) public virtual {
        _totalSupply = _totalSupply.sub(amount);
        _balances[msg.sender] = _balances[msg.sender].sub(amount);
        uni.safeTransfer(msg.sender, amount);
    }
}
```

Action Recommended: Assign a fixed pragma ie “pragma solidity = 0.6.0;

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 for damages resulting from the use of the provided information. Additionally Arcadia would like to emphasize that 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 of time.