THE ARCADIA GROUP

AUDIT

arcadiamgroup.com

Table of Contents

Executive Summary

A Representative Party of the PolkaInsure Finance ("PolkaInsure") engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following PolkaInsure smart contracts on the PolkaInsure repo at Commit #`345cb5e72a6c3569079bf9f0bf43017565a80c91`.

FeeCalculator.sol

PISVault.sol

Conclusion

After presenting the audit of the codebase to the PolkaInsure Team, we also reviewed the deployed contracts and confirmed with them that findings #1, #3 ,#4 and #5 are not currently affecting the protocol security at the time of this report. #2 has been taken under advisement.

Findings

1. user.amount and user.rewardDebt should be reassigned before safeTransfer function

CODE-1
Severity: Theoretical
Impact: None

Contract: PISVault.sol
Category: Security
Finding Type: Dynamic
Lines: 606-617

In the emergencyWithdraw function, user.amount and user.rewardDebt should be reassigned and set to zero before transfering the token amount. Updating them after a transfer could lead to some reentrancy bug which can be prevented by reassigning them first.

This finding doesn't affect the actual deployed contract since the external call is made to a known and trusted contract. Care should be taken when adding new pools, reviewing the potential downside based on the actual code.

```
function emergencyWithdraw(uint256 _pid) public {
      PoolInfo storage pool = poolInfo[_pid];
      require(
          pool.emergencyWithdrawable,
          "Withdrawing from this pool is disabled"
      );
      UserInfo storage user = userInfo[_pid][msg.sender];
      pool.token.safeTransfer(address(msg.sender), user.amount);
      emit EmergencyWithdraw(msg.sender, _pid, user.amount);
      user.amount = 0;
      user.rewardDebt = 0;
```

```
  }
```

2. setDevFeeReciever() should be pointed to a multisig or a timelock instance

CODE-2
Severity: Low
Impact: Medium

Contract: PISVault.sol
Category: Governance
Finding Type: Dynamic
Lines: 647- 650

In the setDevFeeReciever() function, through intended functionality can change its own address to a different one. To prevent the risk of wrench attacks, this address should be pointed to an address that is not controlled by a single key (i.e. a multisig), or one that allows for action to be taken in the event of compromise (i.e. a timelock contract).

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

3. pending_DEV_rewards should be reassigned before if/else

CODE-3
Severity: Theoretical
Impact: None

Contract: PISVault.sol
Category: Security
Finding Type: Dynamic
Lines: 632- 644

pendin_DEV_rewards should be set to zero before any transfer, it would be better to reassign it to another variable, then set it to zero and use the new one to send the proper amount.

This finding is a theoretical issue in the code itself, it doesn't affect the actual deployed contract because we already know the PIS token contract so that this issue won't happen.

But if this contract were to be deployed with a different token contract, an issue could arise.

```
function transferDevFee() public {
       if (pending_DEV_rewards == 0) return;

       uint256 pisBal = pis.balanceOf(address(this));
       if (pending_DEV_rewards > pisBal) {
           pis.transfer(devaddr, pisBal);
           pisBalance = pis.balanceOf(address(this));
       } else {
           pis.transfer(devaddr, pending_DEV_rewards);
           pisBalance = pis.balanceOf(address(this));
       }

       pending_DEV_rewards = 0;
   }
```

4. feePercentX100 should be set to private

CODE-4
Severity: Low
Impact: None

Contract: FeeCalculator.sol
Category: Informational
Finding Type: Dynamic
Lines: 24

Considering set feePercentX100 to private and access or modify it with the corresponding function setFeeMultiplier().

Setting a variable to private ensures that it can only be manipulated via functions in the contract designed for that purpose and never by a successor contract that inherited it.

```
uint8 public feePercentX100;
```

5. noFeeList mapping should be set to private

CODE-5
Severity: Low
Impact: None

Contract: FeeCalculator.sol
Category: Informational
Finding Type: Dynamic
Lines: 26

Considering set noFeeList mapping to private and access or modify it only with the corresponding function editNoFeeList().

This mapping shouldn't be accessed or be updatable directly as public but should only be modified with its corresponding function.

Setting a mapping to private ensures that it can only be manipulated via functions in the contract designed for that purpose and never by a successor contract that inherited it.

```
mapping(address => bool) public noFeeList;
```

NOTE : Avoid leaving unused arguments in functions.

Severity: Informational
Impact: None

Contract: FeeCalculator.sol
Category: Informational
Finding Type: Dynamic
Lines: 26

Recipient argument is not used in this function, even if possibly needed in the future, it would still require an implementation so it would be better to remove it and make a reminder somewhere else, eventually the contract will have to be redeployed.

```
function calculateAmountsAfterFee(
      address sender,
      address recipient,
      uint256 amount
  )
      public
      returns (
          uint256 transferToAmount,
          uint256 transferToFeeDistributorAmount
      )
```

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	PolkaInsure Audit
Client	PolkaInsure 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