POLLENDAO

AUDIT

Distributed Lab

By Artem Chystiakov, AUGUST 2022

SUMMARY: SUMMARY: PRODUCTION-READY.

The audit was made regarding the inclusion of 2 features in the pollen governance contracts. The exact features are shorts and leagues. The former enables pollinators to short particular assets in portfolios in order to receive profit during the bear market. The latter enables pollinators to gather into clans or privileged groups that might provide perks in the future.

The reaudit of the system showed that every bug found during the formed audit session was fixed. Since there were no additional features added, we state that the system is safe to be brought live.

STRUCTURE AND ORGANIZATION OF DOCUMENT

crit **Critical** - The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

high **High** - The issue affects the ability of the contract to compile or operate in a significant way.

med **Medium** - The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.

low **Low** - The issue has minimal impact on the contract's ability to operate.

info **Informational** - The issue has no impact on the contract's ability to operate.

ANALYSIS

The audit hash:

Pollen DAO:
352d58cfe88f6cf34d569e12300ab98f540a6bac

The reaudit hash:

Pollen DAO:
4624451c15c6bf31b36aef9fe3fc86453765e667

01 PORTFOLIO.SOL

Private function is declared but never used

The function `getShortsValue()` is declared but never used or called in the contract.

Recommendation:

Remove the function from the contract.

Miscalculation in the unused function

The `shortsValue` gets miscalculated inside the unused `getShortsValue()` function. Basically, the resulting value gets multiplied by the precision twice.

```
function getShortsValue(
      uint256[] memory assetAmounts,
      uint256[] memory assetPrices,
      bool[] memory isShort
) private pure returns (uint256) {
      uint256 shortsValue = 0;
      for (uint256 i = 0; i < assetAmounts.length; i++) {
            if (assetAmounts[i] == 0 || assetPrices[i] == 0)
                  continue;
            if (isShort[i]) {
                  shortsValue += assetAmounts[i] * assetPrices[i];
            }
      }
      return shortsValue;
}
```

Recommendation:

Divide the `shortsValue` by the `10**18` in the end.

02 LEAGUES.SOL

Users can mint an arbitrary number of tokens due to reentrancy

The reentrancy happens in the `joinLeague()` function when internal `_mint()` function gets called. Because of the Openzeppelin `ERC1155` implementation, the `_mint()` function has a built-in callback that calls the token's receiver address, creating a positive environment for the reentrancy.

```
function joinLeague(uint256 id) external {
      require(whiteListed[id][msg.sender], "User not approved to
            join");
      _mint(msg.sender, id, 1, "");
      whiteListed[id][msg.sender] = false;
      emit JoinedLeague(msg.sender, id);
}

function _mint(
      address account,
      uint256 id,
      uint256 amount,
      bytes memory data
) internal virtual {
      . . .
      _doSafeTransferAcceptanceCheck(operator, address(0),
            account, id, amount, data);
}
```

Recommendation:

To prevent the reentracy possibility change the order of `_mint()` and `whiteListed[id][msg.sender] = false` lines in the `joinLeague()` function.

02 LEAGUES.SOL

The leagues can be created with the same naming

This possible bug does not impact the contract's ability to operate, however, the possibility of creating leagues with the same name might increase the room for scammers and spoofers.

Recommendation:

Forbid the creation of leagues that share namings.

The admin receives 2 league tokens upon league creation

When admin creates a league, he becomes whitelisted and receives 1 league token rightaway. Then he might call the `joinLeague()` method and receive another league token. The call will pass due to the whitelisted status of the admin.

Recommendation:

Either not whitelist the admin or not mint him the token upon league creation.

AFTERTHOUGHTS

- There is a problem with the `npm run coverage` command. For some reason some tests fail when executed via the coverage. The normal `npm run test` works just fine.

With great appreciation and respect.

Distributed Lab

By Artem Chystiakov, AUGUST 2022