Security Audit Report

HappyLand

11/25/2021

Table of Contents

Executive Summary

A Representative Party of **HappyLand**("**CLIENT**") engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following **HappyLand** smart contracts on the **happyland-finance/hpl-contracts** github repository at Commit **#b0cd86ef79ce7d44765885ba52a730b8927dbfb3**

The scope of this audit included the following files:

1. **HPL**.sol
2. **HPLBase**.sol

Arcadia completed this security 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.

There were **7** of issues found, **0** of which were deemed to be 'critical', and **0** of which were rated as 'high'.

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	0	0
HIGH	0	0
MEDIUM	0	0
LOW	0	0
INFORMATIONAL	7	0

Findings

1. Convert `public` to `external` function to reduce gas

Issue: **HPL-1**   Target: **HPLBase.sol**
Severity: **INFORMATIONAL**   Finding Type: **DYNAMIC**

https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L18

This function can be external function

Action Recommended:

Change public keyword to external to save gas

2. Check zero address before set

Issue: **HPL-2**   Target: **HPLBase.sol**
Severity: **INFORMATIONAL**   Finding Type: **DYNAMIC**

https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L36
https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L44

Address zero address checker before setting is good practice and will avoid unexpected issues.

Action Recommended:

Add zero address checker

3. Emit an event after changing important things

Issue: **HPL-3**

Severity: **INFORMATIONAL**

Target: **HPLBase.sol**

Finding Type: **DYNAMIC**

https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L29
https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L36
https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L44

Emit events after changing important things to make easy to track changes.

Action Recommended:

Add events after changing important things.

4. Use `calldata` keyword instead of `memory` in external function arguments

Issue: **HPL-4**

Severity: **INFORMATIONAL**

Target: **HPLBase.sol**

Finding Type: **DYNAMIC**

https://github.com/happyland-finance/hpl-contracts/blob/master/contracts/token/hpl/HPLBase.sol#L39

Use `calldata` keyword instead of `memory` in external function arguments to reduce gas.

Action Recommended:

Change `memory` to `calldata` keyword

5. withdrawEther function is not necessary

Issue: **HPL-5**   Target: **HPLBase.sol**
Severity: **INFORMATIONAL**   Finding Type: **DYNAMIC**

Token contract does not have receive() function or any payable function, so unable to receive the ether.

So withdrawEther function won't be used.

Action Recommended:

Remove withdrawEther function

6. `payable` keyword is not required

Issue: **HPL-6**   Target: **HPLBase.sol**
Severity: **INFORMATIONAL**   Finding Type: **DYNAMIC**

In withdrawERC20, withdrawERC721 functions, `payable` keywords are not required for the receiver.

Action Recommended:

Remove `payable` keyword

7. Reentrancy issue

Issue: **HPL-7**   Target: **HPLBase.sol**
Severity: **INFORMATIONAL**   Finding Type: **DYNAMIC**

When adding liquidity to a pancake, the recipient is a pancake pair.

So when the recipient is a pancake pair, we shouldn't call tokenHook.addLiquidity.

But now, we don't add liquidity when the sender is a pancake router.

For now, HPLHook.sol has a `lockTheSwap` modifier which prevents adding liquidity several times.

Action Recommended:

Change `!pancakePairs[sender] && address(tokenHook) != address(0)` to `!pancakePairs[recipient] && address(tokenHook) != address(0)`

Conclusion

Arcadia identified issues that occurred at hash
#b0cd86ef79ce7d44765885ba52a730b8927dbfb3.

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.