OFTExtended Audit Report

Version 1.0

Windhustler

OFTExtended Audit Report

Windhustler

September 3, 2024

Introduction

A time-boxed security review of the OFTExtended Contract was done by **Windhustler**, focusing on the security aspects of the smart contracts.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource, and expertise-bound effort where I try to find as many vulnerabilities as possible. I can not guarantee 100% security after the review or even if the review will find any vulnerabilities. Subsequent security reviews, bug bounty programs, and on-chain monitoring are recommended.

About Windhustler

Windhustler is an independent smart contract security researcher. Having extensive experience in developing and managing DeFi projects holding millions in TVL, he is putting his best efforts into security research & reviews. Check his previous work here or reach out on X @windhustler.

About OFTExtended

OFTExtended is a contract that extends the functionality of the OFT standard from LayerZero by adding extra features:

- Possibility to pause bridge transactions
- Limit bridging rate
- Hourly limit rate
- Possibility to enable fees

Severity classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

Impact - The technical, economic, and reputation damage from a successful attack

Likelihood - The chance that a particular vulnerability gets discovered and exploited

Severity - The overall criticality of the risk

Informational - Findings in this category are recommended changes for improving the structure, usability, and overall effectiveness of the system.

Security Assessment Summary

review commit hash **- 2f9fba2cd3602f5880c644770289ca2fda34b3c5**

mitigation review commit hash **- ad6f39c641a2902e4ea2a8a276677b35abd14cde**

Scope

The following smart contracts were in the scope of the audit:

- OFTExtended

Findings Summary

ID	Title	Severity	Status
[L-01]	`unpauseBridge` should have `onlyGuardian` modifier	Low	-
[L-02]	Replace < with <= in `minBalanceLimit` and > with >= in `hourlyLimit` checks according to the whitepaper	Low	-
[L-03]	`hourlyLimit` configuration change leads to incorrect `slidingHourlyLimitUtilization` calculation	Low	-

Detailed Findings

[L-01] `unpauseBridge` should have `onlyGuardian` modifier

Context

- OFTExtended.sol#L147

Description

Based on the `comment above the unpauseBridge function` it should be restricted to the guardian address only. It's however restricted to the owner address.

Recommendation

Adjust the comment to reflect the correct modifier or change the modifier to `onlyGuardian`.

Resolution

Fixed by adjusting the comment.

[L-02] Replace < with <= in `minBalanceLimit` and > with >= in `hourlyLimit` checks according to the whitepaper

Context

- OFTExtended.sol#L209
- OFTExtended.sol#L229

Description

According to the whitepaper the send function should revert if:

- `balanceUpdate` is less or equal to the `minBalanceLimit`
- `slidingHourlyLimitUtilization` is higher or equal to the `hourlyLimit`

In the implementation the checks are done with < and >, instead of <= and >=.

Recommendation

Recommendation: Adjust the checks to <= and >= respectively.

```diff
- if (balanceUpdate_ < eidToConfigPtr.minBalanceLimit) {
+ if (balanceUpdate_ <= eidToConfigPtr.minBalanceLimit) {

- if (BridgeUtilizationPtr.slidingHourlyLimitUtilization >
    eidToConfigPtr.hourlyLimit) {
+ if (BridgeUtilizationPtr.slidingHourlyLimitUtilization >=
    eidToConfigPtr.hourlyLimit) {
```

Resolution

Acknowledged.

[L-03] `hourlyLimit` configuration change leads to incorrect `slidingHourlyLimitUtilization` calculation

Context

- OFTExtended.sol#L111

Description

Changing the value for `hourlyLimit` leads to incorrect `slidingHourlyLimitUtilization` calculation. Consider the following scenario:

- `hourlyLimit` is 20 ether
- Someone debits/sends 10 ether
- Half an hour passes
- The owner changes the `hourlyLimit` to 14 ether
- Now the maximum you can debit is 11 ether, as the `slidingHourlyLimitUtilization` is 10 ether - 7 ether + 11 ether = 14 ether.

`slidingHourlyLimitUtilization` hasn't taken into account the configuration change.

Recommendation

A potential solution could be to update the `slidingHourlyLimitUtilization` with the previous `hourlyLimit` value before the configuration changes.

Resolution

Fixed.