Warp Finance v2 - ToB Audit Fixes

Executive Summary

The Warp Finance v2 repository comprises an isolated lending pair implementation into which users can deposit collateral and borrow assets in a pair. More information can be found in the **Warp Finance repo.**

Trail of Bits reviewed these contracts and the audit report can be found **here**.

Findings Summary and Remediations

#	Title	Type	Severity
22	Borrow rate depends on approximation of blocks per year	Configuration	Informational
24	Malicious pairs can steal approved or deposited tokens	Access Controls	High
25	Flash loan rate lacks bounds and can be set arbitrarily	Data Validation	Low

Borrow Rate depends on approximation of blocks per year

The borrow rate formula uses an approximation of the number of blocks mined annually. This number can change across different blockchains and years. The current value assumes that a new block is mined every 15 seconds, but on Ethereum mainnet, a new block is mined every ~13 seconds. To calculate the base rate, the formula determines the approximate borrow rate over the past year and divides that number by the estimated number of blocks mined per year.

Exploit Scenario

The base rate formula uses an incorrect number of blocks per year, resulting in deviations from the actual borrow rate.

Remediation

We plan on providing proper documentation on how the blocks per year is estimated.

Malicious pairs can steal approved or deposited tokens

The LendingPair token flow may allow malicious pools to steal users' tokens when users are interacting with tokens deposited into the vault. To add funds from the vault, a user must deposit funds and then approve the LendingPair contract with which the user wants to interact.

This approval enables the LendingPair contract to deposit, withdraw, and transfer funds on behalf of the user.

Exploit Scenario

Alice creates a LendingPair contract with tokens A and B. Eve creates a malicious LendingPair pool with tokens C and D. Bob joins both lending pairs and allows his vault to transfer all of its tokens. Eve's pool steals all of Bob's A and B tokens.

Remediation

We now use signed messages to approve lending pairs access to vault funds with links included in the message to be signed that fully explains the risks involved in approving a lending pair access to funds.

Flash loan rate lack bounds and can be set arbitrarily

There are no lower or upper bounds on the flash loan rate implemented in the contract. The Blacksmith team could therefore set an arbitrarily high flash loan rate to secure higher fees. The Blacksmith team sets the _flashLoanRate when the Vault is first initialized.

However, because there is no check on either setter function, the flash loan rate can be set arbitrarily. A very high rate could enable the Blacksmith team to steal vault deposits.

Exploit Scenario

Bob, a member of the development team, tries to set the fee to 1%. However, he converts the rate to an incorrect normalized number, setting the flash loan rate to 1,000%. As a result, a user is forced to pay a higher-than-expected flash loan fee.

Remediation
We've added a maximum flashloan rate bound to the vault contract. The maximum flashloan rate is 10%

Ensure that the error messages for reverted transactions are meaningful so that users will know how to respond to a reversion

Remediation

We chose to use abbreviations because of contract code size.

ADDITIONAL CHANGES

Liquidation seizes entire collateral

We've made the change to seize the collateral value for amountOfDebt plus liquidation penalty instead of seizing entire collateral

Commit & Accept Vault transfer of Owner

We've introduced a commit & accept pattern to change vault admin ownership.

Add ability to compose multiple transactions

We've added the ability to compose multiple Vault & LendingPair actions in a single transaction.