KlayBank Audit Report - Lending/Borrowing

Audit Info

- **delivered at** : 2022-02-03
- **auditor** : Creative Carrot 🥕
- **target**
  - **github** : https://github.com/klaybank/klaybank-audit-shared
  - **commit hash** : 3689b3f056f9ff847c83044dfc62b99959ff7584
  - **files** : `klaybank-protocol/**/*.sol`

IssueId	Description	Severity	Status
KBP-00	Double spending with front-running	Tips	Informed

Summary

Klaybank protocol contracts are a simple fork of aave/protocol-v2
As an auditor, I have checked if there is any breaking changes that can possibly break the protocol but was not able to find any breaking changes.

Issues

[KBP-00] - Double spending with front-running

Severity : Tips
Status : Informed

Description

`DebtTokenBase` contract has `approveDelegation()` function which delegates the right to borrow.
Which unfortunately vulnerable to front-running attacks.

Example

- User **B** is allowed to spend 100 borrow allowance from **A**.

- **A** wants to decrease this allowance to 50, so **A** calls `approveDeleagation(B, 50)` --- tx_1
- **B** notices this transaction and uses 100 borrow allowance before tx_1 gets mined
- tx_1 gets mined so allowance increases to 50 and **B** uses 50 borrow allowance
  As a result, **B** spends 150 borrow allowance, which is higher than the initial allowance

Recommendation

try adding `decreaseBorrowAllowance()` as external/public function and use that function when approved to malicious user.