Bug bounty submission

Read more about the mStable Bug Bounty:
docs.mstable.org/protocol/security/mstable-bug-bounty

Reporter: Kevin Foesenek; kevin.foes@live.nl
Date: 04th Nov 2020

Response: Alex Scott | mStable; alex@mstable.org
Date: 05th Nov 2020

Context

mStable has a liquidation mechanism for reward tokens ($COMP, $LEND) accrued in the protocol, as described in MIP2. Through the liquidator mechanism tokens may be liquidated and the value of these tokens realised by SAVE, through an external function with no access control.

Flow

- Liquidator sells $COMP for USDC (or other) on Uniswap once per week (up to trancheAmount)
- Sell USDC for mUSD on Curve and send to SavingsManager
- SavingsManager streams mUSD to SAVE, second by second over the course of a week

Reported Issue

Sent via email on 04th Nov

Based on my evaluation of the protocol I found a potential risk in the liquidator.sol implementation, specifically the triggerLiquidation() function. It is possible for an attacker to influence the uniswap price (for example using a flashloan) then call triggerLiquidation() to sell the COMP at a worse then market price. After the sell the attacker can return the price to normal. The difference in market price and the attacked price is received by the uniswap pool and withdrawn by the attacker when returning the price to normal.

Detailed description / Example
This example uses COMP - USDC, for simplification directly using uniswaps COMP to USDC path. Steps attacker, the steps are executed as one transaction on chain:

1. Influence price COMP - USDC by depositing COMP and withdrawing USDC, using flashloan. Example: initial uniswap Pool 10.000 COMP - 850.000 USDC, after sell attacker COMP 100.000 COMP - 85.000 USDC.
2. Call triggerLiquidation() mStable. This function can be called by anyone at intervals. To make sure to be first at an interval the attacker can use a high gasprice.
    a. triggerLiquidation() - calculates the amountIn / sellAmount based on liquidation.trancheAmount (example 5.000 USDC). With the example attack price of 85.000/100.000 ~= 0.85 this results in selling 5.000 / 0.85 ~= 5.850 COMP.
        i. The market price would be 850.000/10.000 ~= 85 resulting in selling only 5.000 / 85 ~= 58 COMP.
        ii. NOTE: these are example simplified calculations. The real Uniswap price will differ a little as a result of the sells themselves changing the price.
    b. The triggerLiquidation() function will further exchange the 5.000 USDC received from uniswap for mUSD on Curve.
3. The attacker will repay the Uniswap pool, returning the price to normal. Example pool after attack (before payback attacker) 105.850 COMP - 80.000 USDC. After payback 765.000 USDC from step 2: ~10.100 COMP - 845.000 USDC.
    a. The attacker receives the 105.850 - 10.100 = 95.750 COMP.
    b. The attacker pays back the 90.000 COMP loan from step 2. Profit of attack = 5.750 COMP.

The attack can withdraw all the liquidation.sellToken that are set in the Liquidator.sol and hold by the integration. In this example the COMP tokens in the integration contract. If other assets are added as liquidation.sellToken, the same attack is possible and all these tokens can be withdrawn from the integration. Worst case is adding a bAsset to increase the rewards, making it possible to withdraw all funds from the integration. This currently to my knowledge is not possible. I am only aware of COMP rewards set as liquidation.sellToken.

mStable Response

There is an attack vector there, if performed correctly by an attacker with a flash loan, although it is not as simple as described in the above description, or as profitable.
The reasons for this are:

- The Liquidator uses the COMP > ETH > USDC path for selling
    - COMP > ETH has ~$1.6m total liquidity

- **Obtaining sufficient amount of COMP to move the COMP/ETH price as far as detailed is improbable** (COMP is not able to be flash loaned, and liquidity on secondary markets may not be sufficient)
- Using the 90k as detailed would move the price 90% (incurring 270 COMP fee)
    - Given price reduction, the attack would not be cost efficient after fees, and would require specific precision by the attacker

Profitability analysis

Assume current COMP price of ~$85

Step 1: Getting COMP
To my knowledge, it's not possible to flash loan COMP. Flash loaning ETH and then buying on the secondary market is the method then. Currently the only place with sufficient liquidity is either the Uniswap or Compound market. Therefore it could be feasible to borrow $20m ETH, supply to Compound and then borrow 100k COMP at 200%.

Step 2: Attack

1. Drop 100k COMP into the Uniswap market, returns ~1918.24 ETH and costs 300 COMP in fees. Price drops by 90%, causing 1 COMP to be ~= $8.5
2. Call trigger liquidation. $5000 would mean that mStable puts up ~600 COMP for sale for $5k
3. Sell ~1918.24 ETH back to the market and retrieve the 100k COMP + 600. Costs ~5 ETH in fees

Total cost: 300 COMP + 5 ETH = $25.5k + $2k = $27.5k
Total gain: 600 COMP = $51k

Risk analysis

Topic	Analysis
Motive	Moderate
Opportunity	Moderate
Ease of discovery	High
Ease of exploit	Moderate
Scope of affected users	High - Savers and LPs would be equally affected

Financial damage	Moderate-Low
Reputation damage	Moderate

Resolution

Immediate fix: `trancheAmount` was set to 0 on 4th Nov by system governors

Ideal solution: Time weighted oracle over the past X minutes/hours/days

Implemented solution:

- Disable function from being called by a contract, by adding `require(tx.origin == msg.sender)`
- Add a constant floor price to the `liquidation` struct, to determine the minimum amount of `buyToken` that should be purchased with each `sellToken`. Use this to calculate the `minAmountOut` for communications with Uniswap

Implemented in PR 114 here

Conclusion

This vulnerability could surely be exploited, although is shown not to be exceptionally profitable for the attacker. No user funds were at risk, and financial damage to the project would not be severe. If left unpatched, it's likely that this would have been exploited at some stage and the COMP held by the liquidator would have been liquidated at little benefit to the system.

Likelihood	Very low	Low	Moderate	High	Critical
Almost certain	**$500**	**$1,250**	**$4,000**	**$10,000**	**$25,000+**
Likely	$250	**$500**	**$1,250**	**$4,000**	**$10,000**
Possible	$100	$250	**$500**	**$1,250**	**$4,000**
Unlikely	$100	$100	$250	**$500**	**$1,250**
Almost possible	$50	$100	$100	$250	**$500**

Severity

Severity: Moderate

Likelihood: Likely

Bounty: **$1,250**