Marginal V1 Oracle Manipulation

Abstract

Explicitly derives cost of attack figures to manipulate the Uniswap V3 oracle. Adaption of the "Uniswap V3 TWAP Oracles in Proof of Stake" blogpost for the 12 hr TWAP used in Marginal V1. To manipulate the 12 hr TWAP by 50 bps in a PoS two block attack would require swapping ~ 8100x the Uniswap pool Y reserves. For a Uniswap pool with a total of \$1M of full range liquidity, this is about \$4B in upfront capital needed to successfully move the TWAP oracle value, with associated swap fees lost by the manipulator on a 5bps fee tier pool of \$4M.

1 Overview

Marginal V1 pools use the geometric TWAP offered by the Uniswap V3 oracle when assessing the safety of a position. The position is considered safe and not liquidatable if

$$(c_x + s_x) \cdot P_{t-\tau,t} \geq (1 + M) \cdot d_y \tag{1}$$

for a long X versus Y where $P_{t-\tau,t}$ is the geometric TWAP calculated from the Uniswap V3 oracle, s_x is position size, c_x is position margin, and d_y is the position debt.

Given the oracle is fully on-chain, a motivated actor with enough capital could move the Uniswap spot price enough to alter the TWAP and trigger liquidations of Marginal V1 positions for the rewards. We focus on the PoS $n + 1$ block attack [1] of an actor swapping capital all at once to move the Uniswap price and thus the TWAP. Then keeping price there for n blocks, without having to worry about arbitraguers before swapping back on the last block, only paying swap fees on the capital used.

2 Uniswap oracles

2.1 Calculating the geometric TWAP

Uniswap V3 [2] pools track snapshots of the `tickCumulative` tick accumulator

$$a_t = \sum_{j=1}^{t} \log_{1.0001}(P_j) \tag{2}$$

which is a running sum of the current pool tick i value

$$i = \log_{1.0001}(P) \tag{3}$$

for every second since pool creation. P_j is the price at time j. The oracle enables calculation of the geometric TWAP by comparing tick cumulative snapshots at different times. For the TWAP averaged over the last τ seconds, Marginal V1 pool contracts calculate the time weighted average tick raised to the tick base of 1.0001

$$\begin{aligned}
P_{t-\tau,t} &= 1.0001^{\frac{a_t - a_{t-\tau}}{\tau}} \\
&= \left(1.0001^{\sum_{j=t-\tau}^{t} \log_{1.0001}(P_j)}\right)^{\frac{1}{\tau}} \\
&= \left(\prod_{j=t-\tau}^{t} 1.0001^{\log_{1.0001}(P_j)}\right)^{\frac{1}{\tau}} \\
&= \left(\prod_{j=t-\tau}^{t} P_j\right)^{\frac{1}{\tau}} \qquad (4)
\end{aligned}$$

which is the geometric TWAP.

2.2 Manipulating the geometric TWAP

A motivated actor must swap capital through the Uniswap spot pool to move the associated TWAP price. The effect a spot change over several blocks has on the TWAP averaged over τ seconds can be found by deconstructing the geometric average. Take the actor to manipulate spot over the last n blocks of the TWAP averaging interval for a total time Δt from $t - \Delta t \to t$.

For the geometric average over the last τ seconds

$$\begin{aligned}
\log_{1.0001}(P_{t-\tau,t}) &= \frac{a_t - a_{t-\tau}}{\tau} \\
&= \frac{1}{\tau}\Big[a_t - a_{t-\Delta t} + a_{t-\Delta t} - a_{t-\tau}\Big] \\
&= \frac{a_t - a_{t-\Delta t}}{\tau} + \frac{\tau - \Delta t}{\tau} \cdot \frac{a_{t-\Delta t} - a_{t-\tau}}{\tau - \Delta t} \\
&= \frac{a_t - a_{t-\Delta t}}{\tau} + \left[1 - \frac{\Delta t}{\tau}\right] \cdot \log_{1.0001}(P_{t-\tau,t-\Delta t}) \qquad (5)
\end{aligned}$$

Or

$$\log_{1.0001}\left(\frac{P_{t-\tau,t}}{P_{t-\tau,t-\Delta t}}\right) = \frac{a_t - a_{t-\Delta t}}{\tau} - \frac{\Delta t}{\tau} \cdot \log_{1.0001}(P_{t-\tau,t-\Delta t}) \qquad (6)$$

The prices added to the running sum over the manipulated blocks add

$$i_{t-\Delta t} \cdot \Delta t \qquad (7)$$

to the accumulator, where we take $i_{t-\Delta t}$ to be the tick to which the attacker manipulates the Uniswap pool. This is due to the difference in the accumulator value from $t - \Delta t \to t$ being

$$\begin{aligned}
a_t - a_{t-\Delta t} &= \sum_{j=1}^{t} \log_{1.0001}(P_j) - \sum_{j=1}^{t-\Delta t} \log_{1.0001}(P_j) \\
&= \sum_{j=t-\Delta t}^{t} \log_{1.0001}(P_j) \\
&= \sum_{j=t-\Delta t}^{t} i_j \\
&= i_{t-\Delta t} \cdot \Delta t \qquad (8)
\end{aligned}$$

Plugging in above,

$$\log_{1.0001}\left(\frac{P_{t-\tau,t}}{P_{t-\tau,t-\Delta t}}\right) = \frac{\Delta t}{\tau} \cdot \Big[i_{t-\Delta t} - \log_{1.0001}(P_{t-\tau,t-\Delta t})\Big] \qquad (9)$$

The tick to which the attacker must move the Uniswap pool over the last block to change the TWAP oracle value a certain amount is then

$$\begin{aligned} i_{t-\Delta t} &= \frac{\tau}{\Delta t} \cdot \log_{1.0001}\left(\frac{P_{t-\tau,t}}{P_{t-\tau,t-\Delta t}}\right) + \log_{1.0001}(P_{t-\tau,t-\Delta t}) \\ &= \frac{\tau}{\Delta t} \cdot \delta + \log_{1.0001}(P_{t-\tau,t-\Delta t}) \qquad (10) \end{aligned}$$

which is the result from the Uniswap blogpost [1] if you define the TWAP deviation δ by

$$\frac{P_{t-\tau,t}}{P_{t-\tau,t-\Delta t}} = 1.0001^{\delta} \qquad (11)$$

To simplify things a bit, transform the remaining TWAP term into tick space

$$i_{t-\tau,t-\Delta t} = \log_{1.0001}(P_{t-\tau,t-\Delta t}) \qquad (12)$$

which is the time weighted average tick over the period prior to manipulation $t - \tau \to t - \Delta t$. Then,

$$i_{t-\Delta t} - i_{t-\tau,t-\Delta t} = \frac{\tau}{\Delta t} \cdot \delta \qquad (13)$$

Assuming Uniswap pool price is approximately the same over the period prior to manipulation simplifies expressions further, as $i_{t-\tau,t-\Delta t}$ is then approximately equal to the Uniswap pool tick prior to manipulation.

2.3 Upfront capital requirements to manipulate

Take the guidance of the Uniswap blogpost [1] and assume liquidity is incentivized to be distributed over the full tick range on the Uniswap v3 pool we are using as an oracle.

Full tick range positions obey the Uniswap v2 transformations

$$\begin{aligned} x &= L/\sqrt{P} \qquad (14) \\ y &= L\sqrt{P} \qquad (15) \end{aligned}$$

or

$$\begin{aligned} L &= \sqrt{xy} \qquad (16) \\ P &= \frac{y}{x} \qquad (17) \end{aligned}$$

To move the price from $P \to P'$, the swapper must send in to the pool

$$\begin{aligned}
\delta y &= y' - y \\
&= L \cdot \left[\sqrt{P'} - \sqrt{P}\right] \\
&= L\sqrt{P} \cdot \left[\sqrt{P'/P} - 1\right] \\
&= y \cdot \left[\sqrt{P'/P} - 1\right] \qquad (18)
\end{aligned}$$

in Y capital if moving price up, or

$$\begin{aligned}
\delta x &= x' - x \\
&= L \cdot \left[\frac{1}{\sqrt{P'}} - \frac{1}{\sqrt{P}}\right] \\
&= (L/\sqrt{P}) \cdot \left[\sqrt{P/P'} - 1\right] \\
&= x \cdot \left[\sqrt{P/P'} - 1\right] \qquad (19)
\end{aligned}$$

in X capital if moving price down. Transforming in to tick space

$$\begin{aligned}
(\delta y/y) &= 1.0001^{\Delta i/2} - 1 \qquad (20) \\
(\delta x/x) &= 1.0001^{-\Delta i/2} - 1 \qquad (21)
\end{aligned}$$

As mentioned at the end of the prior section, assume the geometric TWAP from $t - \tau \to t - \Delta t$ is approximately equal to the Uniswap v3 pool price prior to manipulation, such that

$$\Delta i \approx i_{t-\Delta t} - i_{t-\tau, t-\Delta t} = \frac{\tau}{\Delta t} \cdot \delta \qquad (22)$$

where the TWAP deviation $\delta > 0$ for Y in and $\delta < 0$ for X in on the initial manipulation swap to get to $i_{t-\Delta t}$.

We have as capital requirements on amount in (i.e. δx or δy) to manipulate the TWAP by deviation δ

$$\begin{aligned}
(\delta y/y) &= 1.0001^{(\tau/\Delta t)\cdot(|\delta|/2)} - 1 \qquad (23) \\
(\delta x/x) &= 1.0001^{(\tau/\Delta t)\cdot(|\delta|/2)} - 1 \qquad (24)
\end{aligned}$$

as fractions of the full tick range pool reserves.

2.4 Figures for the 12 hour TWAP

The Marginal v1 pool uses the 12 hour TWAP to determine whether a position is safe from liquidation, which means we take $\tau = 43200$ seconds. For 12 second block times on Ethereum mainnet, the PoS $n+1$ block attack will have the spot manipulation last over $\Delta t = n \cdot 12$ seconds. Capital requirement expressions become

$$(\delta y/y) = 1.0001^{(3600/n)\cdot(|\delta|/2)} - 1 \quad (25)$$
$$(\delta x/x) = 1.0001^{(3600/n)\cdot(|\delta|/2)} - 1 \quad (26)$$

Calculate for an attacker aiming to manipulate the TWAP by $\delta = 50$, producing a 50 bps deviation. For the two block attack ($n = 1$), the attacker would need to front capital into the pool equal to

$$(\delta y/y) = 1.0001^{(3600)\cdot(25)} - 1 \quad (27)$$
$$(\delta x/x) = 1.0001^{(3600)\cdot(25)} - 1 \quad (28)$$

or $\sim$ 8100x the pool Y reserves. For a Uniswap pool with \$1M of full range liquidity, this is about \$4B in upfront capital needed to successfully move the TWAP oracle value 50 bps. Swap fees lost by the attacker for a 5 bps Uniswap pool would total to ~\$4M, as they must execute the initial swap to move the price and the subsequent swap to return it to its original value.

Even to move the 12 hour TWAP only 25 bps, capital requirements would be 90x the Uniswap pool reserves, or \$45M in upfront capital with \$45K in fees lost. The three block attack $n = 2$ produces the same figures as the 25 bps case, but with the extra capital requirements coming from PoS consensus layer stake requirements to control three blocks in a row [3][4][5] – for the ability to execute the two block attack every $\sim$ 62 days is currently on the order of \$94M in ETH staked at the consensus layer.

References

[1] Austin Adams, Xin Wan, and Noah Zinsmeister. *Uniswap v3 TWAP Oracles in Proof of Stake.* 2022. URL: https://blog.uniswap.org/uniswap-v3-oracles.

[2] Hayden Adams et al. *Uniswap v3 Core.* 2021. URL: https://uniswap.org/whitepaper-v3.pdf.

[3] Chainsecurity. *Why is Oracle Manipulation after the Merge so cheap? Multi-Block MEV.* 2022. URL: https://chainsecurity.com/oracle-manipulation-after-merge/.

[4] Torgin Mackinga, Tejaswi Nadahalli, and Roger Wattenhofer. *TWAP Oracle Attacks: Easier Done than Said?* Cryptology ePrint Archive, Paper 2022/445. https://eprint.iacr.org/2022/445. 2022. URL: https://eprint.iacr.org/2022/445.

[5] Alvaro Revuelta. *Statistical analysis on Ethereum k-consecutive block proposal probabilities and case study.* 2022. URL: https://alrevuelta.github.io/posts/ethereum-mev-multiblock.