

Disclaimer: this post is written on behalf of [\[axis\]](#) and not in my role as an Enforcer of the dYdX Operations Trust

I. Background and Context

dYdX recently experienced a significant security incident, where a targeted attack resulted in \$9 million in user liquidations, which was covered by the platform's insurance fund. This incident underscores the necessity for robust risk management strategies and mechanisms within dYdX's operational framework, particularly as the platform transitions to its V4 iteration. It is important to underscore the fact that this attempted attack was effected on dYdX V3; where both the orderbook & the matching engine are centrally controlled, administered & maintained. Naturally, this is not the case for dYdX V4 & thus, a proper risk identification & mitigation framework should ideally be in place given that there won't be arbitrary access to change the risk parameters so as to mitigate any prospective or actual security incidents.

II. Discussion Overview

This post aims to spearhead community discussion in relation to the dYdX Risk Parameter Recommendation Portal, developed by [@chaoslabs](#) and [@0xCLR](#), into dYdX V4. This portal is a dashboard that provided real-time parameter recommendations based on market liquidity and objective order book liquidity measures. Its aim was a simple one i.e. that of optimizing risk management within the dYdX ecosystem.

III. Benefits that were provided by the Risk Recommendation Portal

Enhanced Risk Management: The portal's well-calibrated margin risk parameters are designed to protect the dYdX protocol from bad debt resulting from market volatility or malicious activities, without unduly restricting standard trading operations.

Data-Driven Decision Making: Leveraging real-time and historical order book data, the portal allows for the computation of recommended baseline and maximum position sizes, effectively managing the risk of positions that cannot be liquidated in stressed market conditions.

Market Liquidity Monitoring: The portal displays Maximum Allowable Leverage at different position sizes, helping to maintain a balance between allowing traders to leverage up and protecting the exchange from overstressing the order book.

Objective Liquidity Metrics: By displaying observed liquidity at various depths and showing how consistent liquidity is in different markets, the portal provides an objective basis for functional analyses and decision-making.

IV. Further points on the Risk Parameter Portal

Adaptation of Current System: The portal was configured for dYdX V3 & naturally has adaptable data and algorithms that can be modified for use in V4.

Algorithmic Transparency: The portal's algorithm recommends margin risk parameters based on sampling order book liquidity in each live market, utilizing a structured approach for defining risk parameters based on observed liquidity levels.

Community Participation: The portal includes the dYdX community to monitor margin risk parameters in a transparent manner, fostering a more participatory approach to risk management within the ecosystem.

V. Tackling Community Involvement

One of the main hurdles in implementing the actual recommendations provided by the Portal is the actual implementation of these recommendations at the code-level in a non-centralised manner i.e. with this responsibility being disbursed/assigned to a class of persons for example.

The main issue with these recommendation implementations is efficiency. We cannot afford to have to go to a traditional Snapshot vote/on-chain vote (ex.) if we want to implement parameter recommendations because (as we have seen), some parameter recommendations (if not all) are pivotal in ensuring the product's well-functioning and also serve to mitigate and prevent attacks such as the V3 market manipulation attempt.

Potential solutions to cater for the above include using a chain-specific implementation of Lido's Easy-Track module (<https://github.com/lidofinance/lido-improvement-proposals/blob/develop/LIPS/lip-3.md>). The benefit of a potential implementation of this is the composability of the mechanism - more below:

A normal motion life cycle includes:

- **Motion creation.** A motion can be started by an address entitled to start the specific Easy Track motion type (this can be a member of a prospective risk management sub-dao)
- **The voting period.** A preset timeframe (motion duration) when objections can be submitted by the governance token holders. The motion cannot be enacted until this period expires (the periods can be adapted to the level/% of the risk parameter recommendations & the nature of the parameter recommendation).
- **Motion enactment.** If the objections threshold hasn't been exceeded within or after the voting period, the motion can be enacted by anyone. Before enactment, the motion can be canceled by the address that started it.

VI. Conclusion

The above is high-level & meant to initiate a discussion around the topic. In my personal opinion, it's high time that we start thinking of setting up a risk management sub-dao that will be encharged with overseeing the administration & implementation of dYdX V4's risk management program (however, this is merely a personal opinion of mine).

Some questions to think about:

How does one achieve efficiency without sacrificing decentralisation/causing asymmetry?

Is a risk management committee/subDAO an implementation that we should actively start thinking about?

What would be the necessary knowledge & experience of risk management subDAO members?

What would cause a risk parameter change to be classified as 'urgent' & merit a lower time period (imagine we are using the easy track mechanism)?

Looking forward to the discussion!

Kind regards,

IL, [Axis Advisory](#)