Written in collaboration with @Izzy and @ujenjt

Motivation

Currently the Lido protocol only collects Beacon chain staking rewards, but after the Merge there will be two new types of rewards: transaction fees and extracted MEV. A good option for MEV extraction could be using third-party solutions that the community could trust. A promising solution that aims to be ready together with the merge is MEV-Boost by Flashbots, and in this post we want to share research on how this solution might be integrated. Discussions around if or how Lido will extract MEV will take place in a separate post.

MEV-Boost

MEV-Boost architecture allows validators to outsource the task of block construction to third-party block builders. Builders construct blocks using the transactions or bundles available to them and send their bids to relays. A bid contains a value and block header without transactions. The MEV-Boost client running next to the validator collects possible bids from relays, chooses the bid with highest value and blindly signs it. The signed block is sent to the builder, and the builder reveals the contents of the payload and propagates the block.

Relay-based attack vectors

The architecture of MEV-Boost assumes extra participants in the network: the MEV-Boost itself, relays, builders and searchers. MEV-Boost has open source code and runs directly next to the validator, so it can be considered trusted. MEV-Boost only interacts directly with relays and attacks on validators by builders must go through relays and exploit defects in validation mechanics. The current architecture does not provide builder moderation on the validator side and only provides a white list of relays. Since we can only control the choice of relays, it seems we can simplify things and say that relays are responsible for any misbehavior of all participants behind them. So, the main attack vectors come from the relays, let's look at the likely scenarios.

Invalid payload or no block reveal

A relay can provide an invalid payload that will not be included in the chain or can not reveal a block after it's signed by a validator. In this case the validator does not profit and is penalized by the network for the missed block.

Mitigation:

1. MEV-Boost assumes built-in protection against fraud and probably this misbehavior can be caught in it.

*Edit Jul 1, 2022: after conferring with flashbots we understand that this mechanism is not in place and instead practical solutions via relay-attested correctness should be used instead (and thus relay reputation becomes increasingly important).

1. Using whitelisted relays to build blocks.
2. Observing the status of the slots allows us to see an increase in the frequency of missed blocks, and collecting information about which relays suggested a payload with a maximum value for a slot allows us to identify relays that with some probability may have been used to build missed blocks. If investigation shows that this is caused by a malicious relay or builder and the relay does nothing to penalize the builder or improve the verification process on their side, such a relay can be excluded from the whitelist.

Inaccurate value

A builder can provide a payload with an inaccurate payment to a fee recipient or no payment at all.

Mitigation:

1. The MEV-Boost design assumes proof of payment, which is verified on the MEV-Boost side so such a payload should not be chosen.
2. Using whitelisted relays to build blocks.
3. Observing the extracted value for each block and collecting information about what payloads and value were suggested by relays allows to identify such cases.

Value withholding

If there is insufficient competition on the market or collusion, builders behind relays may offer Lido blocks with zero or low value. A builder with exclusive access to a popular private mempool, colluding with a large staking pool, can hold transactions and bundles with good value for validators from that pool. This can give a competitive advantage to the pool, and thus economic motivation to the builder.

Such collusion is possible only among large participants and should lead to reputational losses and community reaction. Nevertheless, observing the distribution of extracted value among validators can highlight problems in the early stages.

Mitigation:

1. Using at least one relay that looks into a shared mempool allows to extract MEV in at least a publicly available amount.

2. Observing the average extracted value among Lido and the rest validators allows to see the economic preferences in some groups of validators (all Lido or individual node operators).

Transaction censoring

Relay can censor transactions in the network. We do not believe that it's realistic to censor an individual user for a long time, because it would be very expensive for the attacker if there are at least two competing block builders. However, censoring for a short time can also be effective in some scenarios. We believe that this problem can be solved either at the level of attracting new relays and block builders, or at the level of PBS and we would be happy to participate in this discussion.

Mitigation:

1. Using at least one relay that looks into a shared mempool doesn't solve the problem, but might make censoring for the attacker more expensive in some scenarios.

Validator misbehavior

To extract MEV and receive transaction fees, it's required to specify the fee_recipient address in the validator client, to which the rewards will be collected. If an approved policy requires a smart contract controlled by the Lido protocol for collecting rewards, then this address must be specified by all node operators as a fee_recipient. A node operator may be tempted to specify another address as a fee_recipient and receive the rewards bypassing the Lido protocol.

The MEV extraction policy may contain a list of possible payloads sources. Validator may deviate from the policy to achieve his own goals and use other sources.

Mitigation:

1. Transparent MEV extraction policy, prearranged penalties for node operators for non-compliance.

2. Validator monitoring that tracks fee_recipient addresses and payload sources.

Monitoring

To track the availability of all services, configuration problems and misbehavior by relays or validators, we assume the presence of monitoring. Monitoring should collect the following data from whitelisted relays and trusted CL and EL nodes for each slot:

- Was the block proposed in the slot.
- Which node operator's validator proposed the block.
- Were the whitelisted relays available at the time of the propose.
- What were suggested payloads by the whitelisted relays.
- Which relay was used to build the block.
- Was there a transfer from the builder, to what address and what the value.
- How much the fee_recipient balance has increased.
- How much Lido MEV Vault balance has increased.

Based on this data, the following analysis can be done:

- Understand which validators do not use MEV-Boost.
- Find validators that extract MEV to unknown addresses.
- Understand how often blocks are missed in slots and what relays may have been selected by MEV-Boost for this slots.
- Find validators using a local payload, the possible reasons for using it and the frequency of use.
- Calculate and compare the average extracted value over long distances and see if censoring of Lido validators could

be present.

In some cases, it’s not immediately possible to detect obvious malicious behavior of a validator, since it’s always possible that all relays are unavailable and the validator has to build a block using a local payload. But in large samples it’s visible if validators of one of the node operators have regular misbehavior.

Detailed MEV monitoring design

Whitelisting

We assume to use a white list of relays. The limited list simplifies monitoring and trusted relays in the list protects against a number of attacks. The current design of MEV-Boost suggests running with a white list of trusted relays. In the process, relays check each other for fraud, and if fraud is detected, relays are removed from the list (1, 2).

Integration

Node operators setup MEV-Boost and specify the Lido MEV Vault address as a fee_recipient

. The DAO then chooses a list of trusted relays by voting and node operators use these addresses in the MEV-Boost. Dev team monitors blocks suggested by Lido validators and inspects them for relays or validators misbehavior. We suppose to open source the monitoring code and develop a UI for the collected statistics.

Completed steps

- Lido joins the Flashbots Eth2 Working Group.
- Published ADR: Rewards distribution after The Merge.
- Published LIP: On-chain part of the rewards distribution after the Merge.
- Prepared Merge-ready protocol service pack.
- The final audit of the contracts passed.
- Deployed Lido protocol on Kiln and launched validators and oracle.
- Developed the prototype of a monitoring service.
- Lido protocol updated on Goerli.
- Lido protocol updated on Mainnet.

Required steps

- Test the usage of MEV-Boost with a relay on Kiln.
- Check which versions of the clients used by Lido node operators are compatible with MEV-Boost.
- Organize a MEV-Boost testnet for Lido node operators.
- Develop and run the monitoring service.