On April 21sh 2023 Justin Drake, samczsun, and myself received a disclosure from the user who performed the unbundling attack on April 3. They requested that they be called the term “low-carb-crusader” instead of “sandwich the ripper” or similar nomenclature in return for disclosing details on a unique block equivocation strategy that should be mitigated. The following post shares a timeline and details of this strategy. Flashbots relay logs confirm that the strategy was never used in production.

Timeline

- On April 21st, 2023 low carb crusader emails Justin Drake, samczsun, and Robert Miller indicating they were responsible for the previous block equivocations on the mev-boost relay leading to $20m+ lost by sandwich bots and had another disclosure to make. They requested confirmation that this was the appropriate channel, that they would be referred to as the neutral “low carb crusader,” and that their disclosure as well as the resulting mitigation details be made public as soon as possible.
- On April 22nd, 2023 low carb crusader signs a message from their EOA “0xF489Bd7bC0589Ae10C6dd7f39Eb429F3aACe746e” proving they were indeed who they said they were.
- On April 23nd, 2023 1:47am UTC low carb crusader disclosed details of a strategy whereby a proposer could have a structural advantage in a block equivocation race against a mev-boost relay, which would allow them to unbundle blocks from relays, thereby undermining user guarantees.
- On April 23, 2023 7:02am UTC Robert confirms receipt and reaches out to CL teams to confirm the details of the strategy.
- At 7:50am UTC details of the strategy are confirmed with Terence.
- Around 8:00am UTC a warroom is established at the Flashbots onsite to discuss potential mitigations.
- At 9:20am UTC the Flashbots team rollouts a mitigation on Goerli and Sepolia for testing.
- At 9:50am UTC other relays are notified of the details, the patch, and to prepare for a patch.
- Around 10:50am UTC the Flashbots team rolls out the patch to mainnet. Other relays roll out in the next 20 hours to few days.

Details

- Relays used to allow proposers to call getPayload before the start of the slot (denoted as t=0)
- For example, if a valid proposer called getPayload 1 second before their slot then the relay would respond.
- Due to an aspect in how beacon nodes work, as long as the block was valid it would be accepted by Prysm, but critically not propagated until the proposer’s slot starts. Lighthouse has slightly different behavior, discarding the block.
- This allowed proposers to receive payloads before the start of the slot. By doing so, the proposer could view the contents of the relay’s block, repackage the relay’s block’s transactions into a competing block, and attempt to race the relay’s block by equivocating.
- Moreover, the proposer could structurally gain an advantage in this race by directly sending their block across the p2p network in favorable positions while the relay’s block was still not propagating.
- In case of a successful race, the proposer-preferred block would land on-chain, and the relay’s block would be dismissed, although the proposer would get slashed.

Mitigation

- If getPayload is called early, wait until the proposer’s slot t=0 before sending the block to the relay’s beacon nodes or returning it to the proposer.
- See this commit for details.
- This mitigation is now live on all relays.

Impact

We were able to confirm through the Flashbots relay logs that this strategy has not been used.