In this quick note we analyze the special case of ex-anti and sandwitch attacks on ePBS vs the current implementation. We show that with the proposed values for proposer and builder's boosts as in Payload boosts in ePBS the situation is actually an improvement over the status quo.

This short note contains the raw numbers and it's meant to be a quick update, no fancy diagrams, for such I recommend looking at the design notes in ePBS specification notes - HackMD or even the forkchoice implementation notes inePBS Forkchoice annotated spec - HackMD

Ex-anti reorgs, the need for proposer boost

The classical 1-slot ex-anti reorg goes like this. The proposer of slot N

plans to reorg the block of N+1

. For this they withhold their block during their time. After the proposer of N+1

reveals his block (based on N-1

) the attacker reveals their block N

together with \beta

attestations for it. The attack is succesful if

\beta > PB

.

Which in the current situation makes us resilient to these attacks up to a 40% adversary.

Ex anti on ePBS

On ePBS the situation for an ex-anti attack changes due to the (block, slot) voting nature of fork choice. The attack goes as follows.

- Since the proposer of N

wants to get their payload included, they can't simply reveal their block after N+1

does. They have to have a timely payload so that the PTC votes for it.

- They therefore reveal their consensus block targeting a split view of the attesters at 1/4 of a slot. 1-x

of the committee votes for N-1

, as they didn't see the block on time, and x - \beta

vote for N

(the adversary withholds their attestations).

- The builder of N

reveals on time and the PTC attests to the builder's presence.

- The proposer of N+1

will reveal a block based on N-1

only if

1 - 2x > RB - \beta

where RB is the reveal boost that the builder of N

received.

- The attacker now reveals their attestations for N

.

The attack is successful if RB > PB + 1 - 2x

But given the above inequality this implies RB > PB + RB - \beta

. Therefore we obtain as in the current status quo \beta > PB

.

Since in ePBS the proposer boost PB is set to 20%

. One is inclined to think that we have ex-anti reorg protections only up to 20%, a considerable downgrade from the current implementation. But notice that the payload of N+1 is not reorged

. In fact, the builder of N+1 will not reveal their block since the head is N when the attack is successful, and because of the builder withholding safety

, their bid payment will not be necessary. In fact, in order to reorg the payload as well in ePBS, we would require a sandwich attack.

Sandwich attacks the classical case

A sandwich attack is very similar to an ex-anti one, but now the adversary is proposing slots N and N+2 and plans to reorg the block N. Their setup is just as in the ex-anti attack: they reveal the block N late, together with \beta

attestations for it. Block N+1

is early and receives 1 - \beta

attestations (the attacker votes for N

during N+1. The attacker then reveals N+2 based on N, obtaining proposer boost and attempting to reorg N+1. The attack is successful if

2 \beta + PB > 1 - \beta \Leftrightarrow 3\beta > 1 - PB

From where with the current values we obtain protection against this attack against validators up to \beta = 20\%

.

Sandwich attack in ePBS

In ePBS the sandwich attack starts also as an ex-anti setup. In particular, to get the proposer of N+1 to base their block on N-1, the setup requires

1 - 2x > RB - \beta

as above. The consensus block of N+1 receives 1 - \beta

votes just as in the current implementation, and the builder of N+1 reveals timely obtaining a builder’s boost: this is the main difference, the builder’s boost makes this sandwich attack much more difficult

.

The attacker then reveals their N+2 block based on N. Obtaining proposer boost. The attacker’s branch then has weight PB + \beta + x

. While the canonical branch has weight RB + 1 - \beta + 1 - x

. The attack is successful then if

PB + 2\beta > RB + 1 + (1 - 2x)

Which according to the inequality above implies PB + 2 \beta > 2RB + 1 - \beta

, from where

\beta > \frac{2RB + 1 - PB}{3}

Which with the proposed values of RB = 40% and PB = 20% gives protection against this attack by an attacker up to 50%, a significant improvement over the current situation.

Multiple slot post-anti reorgs become worse in ePB. To give some numbers, in the current implementation we are resistant to 60% attackers for 1 slot post-anti reorgs and 53% for 2 slots post-anti-reorgs. On ePBS these numbers become 40% and

37%.