There is one type of an attack on ETH PoS, where attacker is willing to sacrifice 32 ETH slashed to potentially have a much larger gain.

If an attacker shorts a significant amount of ETH, then the potential gain from even temporarily disruption of the network can outweigh the 32 ETH loss.

I do not know if the current ETH clients have been tested against attacks like that.

The simplest attack involves the malicious proposer creating a large number of non-unique but valid block proposals, and distributing them in such a way, that each client get a different proposal.

Then you essentially get a huge number of weak branches across the network where each client will have a different world view.

The clients should in theory adjust the fork choice rule once they become aware of another block version, but looking through the source code of some ETH clients I am not sure whether this case is well handled and the corresponding mechanisms are actually implemented.

For instance, the attacker can first release only a single unique block version, then wait until a long branch X is built on top of this block, and then flood with more non-unique versions later.

It is not clear to me how the current ETH specification handles this case, and whether the branch X needs to be invalidated in the fork choice rule after slashing, or whether the branch will stay valid and only the proposer needs to be slashed.