Special thanks to James Prestwich and @adlerjohn for discussions around this post and for the valuable feedback

Overview

In order to become a validator in ETH2.0, one has to deposit 32 ETH into adeposit contract and wait until the validators on the beacon chain process the set of deposits for a particular epoch. The main assumptions for this mechanism are that the validators on the beacon chain form an honest majority and that the PoW chain isn't under attack i.e. no 51% attack is occuring on ETH1.0 while this mechanism is in progress. These assumptions make it such that this bridge looks more like a federation of validators. This resemblance makes the current bridge design susceptible to attacks that affect federations. The point of this post is to show several attacks that are possible in this bridging mechanism and to provide some mitigations that could potentially help in deterring and recovering from these attacks.

Attacks

Collusion of validators on eth2

As it currently stands, there's no provisions for slashing validators on the ETH2.0 in case they misbehave. For the purposes of this post, misbehavior is defined as voting for anything that isn't prescribed in the current ETH2.0 specification. The fact that validators don't get slashed for misbehavior means that they can vote for any eth1 data to get included into the beacon chain. Of course, they can't do this naively. In order for a set of validators on the beacon chain to include invalid ETH1.0 data into the beacon chain, they will need to find a way to get enough computing power for a significant amount of time and vote invalid ETH1.0 history that includes arbitrarily-sized deposits. Several methods for this include the following:

- Renting hash power
- Colluding with miners
- Bribe miners
- If the validators themselves are also miners on the ETH1.0 chain and don't mind losing out of potential block rewards, then they can use their own hash power

One of the main ramifications of such an attack would be preventing the onboarding of potential validators to ETH2.0, a variation of the gatekeeping attack.

Collusion of miners on eth1

Alternatively, miners on the ETH1.0 can collude (or be bribed) to try to purposely lower the difficulty on the ETH1.0 in what is known as a difficulty depression attack

(See Braydon Fuller's paper).

Difficulty depression attacks can happen on the main chain or in privacy histories forked off the main chain. The main ramifications of these attacks is the using the disk space for unecessary data and exhausting CPU resources.

Another attack that colluding miners can attempt is to deposit a large of amount of ETH into the deposit contract and then attempt to rewrite this out of ETH1.0's history.

Both of these attacks would enable miners to manipulate the current ETH2.0 light client design. However, the major hurdle for these attacks by miners is coordination instead of cost (see crypto51.app for a ballpark figure)

Mitigations

There are a few potential mitigations to the above attacks that I'll go into.

Fault Attributable Light Clients

A simple way to help discentive validators from voting on any data would be to implement slashing conditions on the ETH2.0 end of the one-way peg. The slashing conditions would need to take into account the potential cost of colluding with other validators and bribery (the latter of which is difficulty and an in-progress research question). The particulars of such slashing conditions is an open question.

Preventing forks before a checkpoint

In order to prevent difficulty depression attacks, it suffices to prevent forks from being created that are set before a set checkpoint. As of this writing, this is the only known solution to such an attack.

Social Consensus

Obviously, in the case in which these attacks were to actually occure, the community should be able to detect this through various monitoring of block explorers, etc. Through social coordination, the Ethereum community can decide what to do such as ignore deposits after a certain block height and/or hard-forking. If the community acts within a reasonable window of time e.g. ~5 hrs of a purported attack, then recovering should be straightforward and the community should be able to revert to a previously agreed upon state.

References

- Braydon Fuller’s paper on Chain-width expansion and difficulty depression attacks
- Two-way bridge between ETH1.0 and ETH2.0
- Gatekeeping attack
- Private communication with James Prestwich and John Adler.

Feedback on this post is greatly appreciated!