Description

Slashing conditions are used to ensure safety. However, the adversary can use the slashing condition to punish honest validators. The key idea of the attack is that the adversary deliberately violate slashing conditions to make honest validators violate slashing conditions too. Though EIP-3076 has been used to protect honest validators from violating slashing conditions, the adversary can utilize it to attack liveness and also punish honest validators using leaking.

Attack scenario

The attack starts at a special epoch. We denoted as epoch 1. Epoch 1 is not justified because of adversarial attestation delay and network problem. The last block of epoch 1 (block 63) is adversarial. This block contains enough adversarial attestations to justify epoch 1. This block is also withheld. The adversary is waiting for a proper time to release it. The detail is as follows:

[

figsl1

1596×657 46.8 KB

](https://ethresear.ch/uploads/default/original/2X/8/80645b7d1cc5f06885188bad185b209cd0b700b8.png)

1. For slot 64 to slot 77, epoch 1 is not justified. The honest validators and adversary vote on 0\rightarrow2

(denote as blue).

1. At slot 78, the block 63 is released. Epoch 1 is justified. The honest validators vote on 1\rightarrow2

(denote as purple). The adversary validators withhold their attestations.

1. At slot 95, the adversary deliberately violate slashing conditions. Some the attestations contains in the block 63 can not justify epoch 1. So the last justified epoch becomes epoch 0 again.

2. During epoch 3, the purple validators vote on 0\rightarrow3

. They violate slashing condition 2 and are punished.

Analysis

Finally, \frac{17}{32}

of honest validators violate the slashing condition 2. In reality, the beacon node uses EIP-3076 to prevent honest validators from violating slashing condition. But the adversary can use this implementation to punish validators. The validator will check the attestation and block before broadcast. During epoch 3, the purple validators try to release attestations that violate the slashing condition2. All of them are prevent because of EIP-3076. At most 64.4% validators can vote on epoch 3. Epoch 3 can not be justified. This scenario will last for many epochs. During the scenario, the system will enter inactive condition. The purple validator are punished because of inactive actions.

Split the Views

But we find that the current design has a flaw. The justified checkpoint only update when the the new justified checkpoint is higher than the old justified checkpoint. Suppose the epoch i

is justified at first. So the last justified epoch become epoch i

. Then the validators who vote on epoch i

are slashed. This lead to that some attestations are invalid. The rest attestations are not enough to justify epoch i

. But the last justified epoch will not become i-1

. It will still become epoch i

. By using this flaw, the adversary can split the views of honest validators using the following strategy: Let half of the validators receive the block that contains justified epoch i

first while the other half of validators receive the block that contains slashing imformation first. So the first half validators

denote epoch i

as last justified checkpoint while the last half denote epoch i-1

as last justified checkpoint. This leads to that the view of validators splits and causes a liveness attack.