Description

The reward/penalty mechanism in Proof of Stake is designed to encourage validators to vote and deter validators from misbehaving. It prevents honest validators from being framed and ensures that validators who engage in malicious behavior (e.g., wrong voting) are detectable and gain punishments. However, under our attack, we found that this mechanism in state-of-the-art implementations is not functioned well, where an attacker has a high probability of preventing honest validators from receiving rewards or even being punished. This may lead to potential security issues, as an honest validator may leave the ecosystem due to unfair treatment or no profit.

Attack scenario

Overview

Under the design of current implementations (e.g., in the case of Prysm), a validator modifies the justification state upon receipt of a block from the preceding epoch (see Prysm Code). In our attack, an attacker uses this update mechanism to deceive honest validators, causing them to transition from one chain to another, and further make those who cast their votes prior be considered to have made incorrect votes, resulting in penalties.

Assumptions

We assume that more than 2/3 validators are honest actors who will act in the best interest of the network. It means adversaries can, at most, control 1/3 of validators. We also assume that our attack works under the partially synchronous network model. The attack begins from a situation where epoch i

just ended and all validators now reach Global Stabilization Time (GST).

How to launch our attack?

Initially, our attack requests the justification of epoch i

is delayed due to the adversary withholds some votes in epoch i

. The key point of attack is described as follows.

1. At first, the adversary withholds a block (dashed reddish square) in epoch i

containing enough votes to justify checkpoint in epoch i

. Once all validators now reach GST at the end of epoch i

, chain A is considered the canonical chain for all validators.

1. In epoch i+1

, the victim validator proposes on chain A.

1. Then, the adversary block (with the hidden votes) controlled by the attack is released. This triggers the update of justification to change the candidate chain for chain A to chain B.
2. After that, the victim validators who vote with checkpoint on chain A as target are punished.

[

fig1

2855×2175 154 KB

](https://ethresear.ch/uploads/default/original/2X/8/8e7f219a648e5e72bc9fe686e974908bd038b427.png)

Long-time attack

We present out our attack against the Ethereum Proof-of-Stake. The begin from a situation where epoch i

just ended and all validators now reach GST. The justification of epoch i

is delayed since the adversary withholds some votes in epoch i

. And chain A is consideration of the canonical chain for all validators.

1. At first, the adversary also withholds a block in epoch i

containing enough votes to justify checkpoint in epoch i

.

1. In epoch i+1

, adversary propose blocks and cast votes on chain B whereas honest validators do so on chain A.

1. Then, the adversary block (with the hidden votes) is released at the moment when half of the honest validators have already cast their vote for chain A. This adversary block triggers the update of justification to change the rest of honest validator view of the candidate chain from chain A to chain B.
2. Lastly, the adversary withholds some votes in the last few slots of epoch i+1

.

1. Repeat the process.

Reward

Penalty

54\cdot base_reward

40\cdot base_reward

Impact

As the attacks are continuously launched, some validators may incur more penalties than rewards. This significantly discourages the validators from participating in the staking process, eventually leading to their withdrawal from the system.

Proposed fixes

The penalty rule right now is too strict in partial synchronous network. Considering that slashing conditions and inactivity leak have already played their roles in maintaining safety and liveness, it is suggested that a checkpoint vote with mismatched target or source should not result in penalties for the validators.