Slashing On Ethereum

Research Report

October 30, 2025

by Torran Green, Maksimjeet Chowdhary

Slashing on Ethereum

In blockchain staking systems, slashing serves as a critical safeguard —one of the protocol's last lines of defence. Despite being frequently misunderstood, it remains both a deterrent feared for its economic repercussions and a mechanism praised for strengthening network integrity. This article examines the concept of slashing, its broader implications, and key factors to consider when participating in staking on Ethereum.

Slashing vs penalties

A functioning Ethereum validator continuously signs attestations, or votes, for its view of the chain. If a validator's view of the chain (state) is in agreement with others, their attestations will match and be included in a block (attestations with invalid head votes may be included; however, they are not rewarded). In addition to regular attestations, validators are randomly assigned additional responsibilities: block proposals and participation in the sync committee. However, when a validator fails to fulfil its duties for an extended period of time, it is considered to be offline.

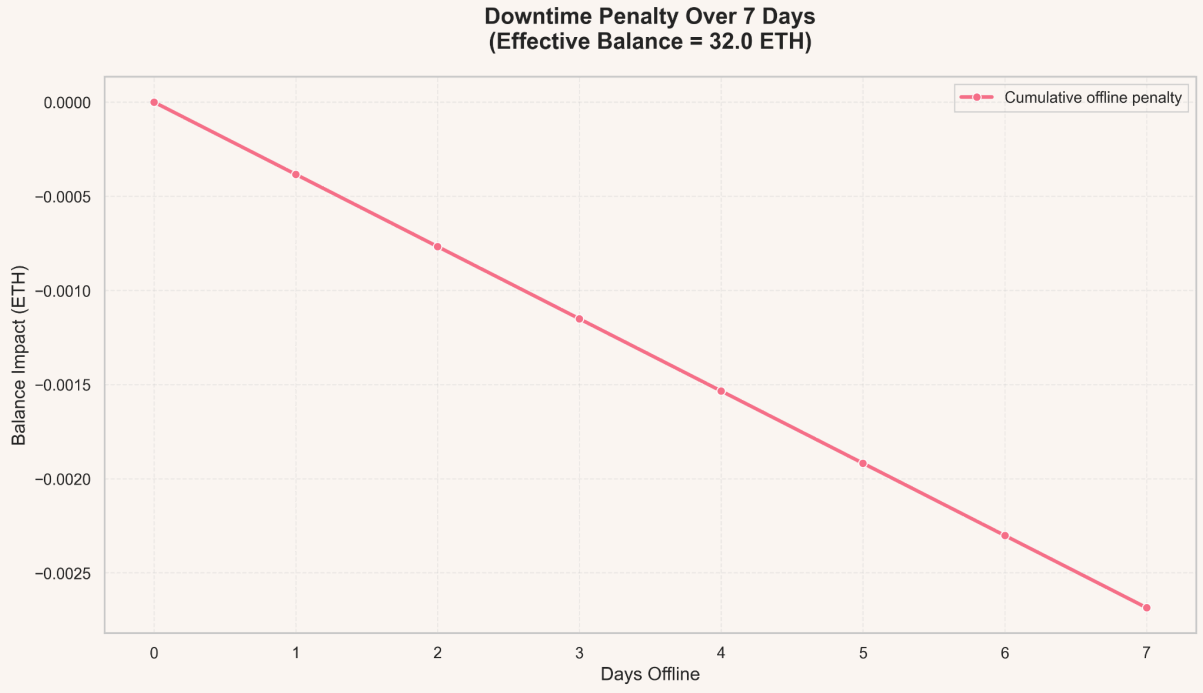
On Ethereum, being offline is not considered to be a major offence. Since Ethereum is designed with decentralisation in mind, being offline is treated leniently, and validators incur negligible downtime penalties. Any behaviour that acts against the security of the network and its decentralization, however, is treated harshly with slashing. We will dive deeper into that later.

Downtime Penalties

For the time a validator is offline, it is simply penalised by an amount almost equal to the amount of ETH it would have earned had it been online and functioning.

- For normal successful attestations per epoch, you might be rewarded with +0.000014 ETH for every 32 ETH staked. When the validator goes offline, it instead accrues a downtime penalty of around −0.000011 ETH per epoch for every 32 ETH staked (amounts vary with network conditions).
- For any block you fail to produce due to delinquency, the only penalty is the opportunity cost of not earning the block proposer rewards. Effectively, this means that for a day offline, the penalty in terms of negative ETH amounts to about 0.002 ETH.

As the chart below shows, the balance impact from a validator simply being offline is modest, and the protocol affords ample runway to diagnose configuration issues and bring the validator back online before losses meaningfully compound. The real step-change in risk comes from slashing and inactivity leaks —i.e., penalties for unsafe, conflicting signatures — which we turn to next.



Inactivity leaks

Apart from the negligible downtime penalties that validators may receive, mass network outages are discouraged with inactivity leaks. Finality on Ethereum is achieved when > 2/3 of the network agrees on the state of the network. If the chain goes more than 4 epochs (~ 25.6 minutes) without finalizing, the inactivity leak starts.

The leak uses per-validator "inactivity scores" which rise each epoch you miss a timely target vote and fall when you perform; the penalty per epoch is proportional to your score and your effective balance.

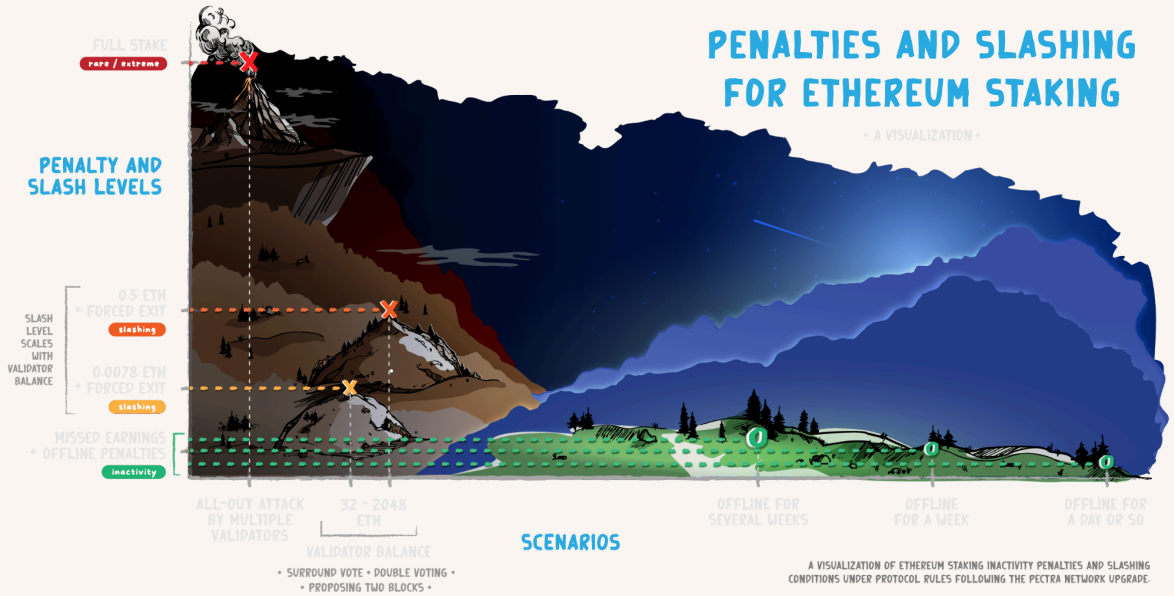
$$penalty \propto score * balance$$

The penalty applied hence keeps increasing as your score increases. Intuitively, if you stay offline, your loss grows roughly quadratically over time. This way, the leak reduces the effective balances of non-participating validators until the *remaining* online validators control ≥ 2/3 of the (now-smaller) total, at which point finality resumes.

Once the effective balance of an individual validator drops to 16 ETH, it is force-exited from the network. Simulations show that a continuously offline 32 ETH validator would be ejected by the leak in about 4,686 epochs (just under 3 weeks) if the leak persisted that long. If the exit queue is long, however, the validator may continue leaking balance below 16 ETH before fully exiting. It is also worth noting that if you run a high-balance validator and go offline during a leak, your absolute ETH loss scales with your effective balance (as shown in the formula above), though the *percentage* dynamics remain similar.

Empirically, leaks on mainnet are rare. There has only been one such incident in the entire history of Ethereum on May 12, 2023, and the leak lasted for 9 epochs (~1 hour) after which the network recovered. The possibility of inactivity leaks arises on when Ethereum clients are concentrated, and a client upgrade turns out to have a bug. Specifically for this, client diversity is emphasized, and validators are encouraged to use clients which are run by <1/3rd of the network.

Slashing



Source

A famous visualization (as shown above) on EthStaker website shows the spectrum of inactivity and slashing penalties on Ethereum. While inactivity penalties have green sceneries and clear skies, that of slashing is full of steep jagged mountains with dark shadows and smoke.

Slashing is a term used to describe the Ethereum network's response to a validator's violation of its rules. Validators are supposed to be slashed only when they act maliciously and harm the network. There are primarily four types of slashable events on Ethereum:

1. making two different attestations for the same slot
2. making an attestation whose source and target votes "surround" those in another attestation from the same validator
3. proposing more than one distinct block at the same height
4. attesting to different head blocks, with the same source and target checkpoints

Interestingly, all of the above slashing events are treated equally, and if proof of a validator committing a slashable event exists, the validator is slashed, which is a process of three steps:

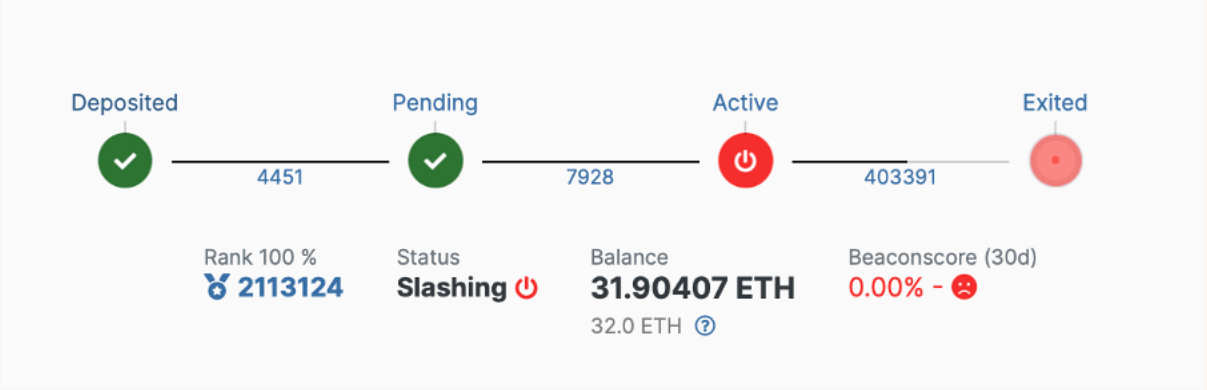
1. Initial Slashing
2. Subsequent inactivity penalties
3. Correlation penalties (if applicable)

Implementation of slashing

Initial Slashing

A slashable offence triggers an immediate, one-time penalty that scales linearly with the validator's staked balance. Specifically, **1/4096** of the validator's balance is slashed, roughly equivalent to **~3 days of staking rewards** at typical rates. After the initial penalty, the validator is **immediately exited** and marked **"slashed"** on the beacon chain. Its status transitions to **active_slashed**, as shown in the example image.

Once the initial slashing is applied, the validator is queued for exit and considered slashed. While awaiting exit, it appears as **active_slashed**; after exi,t it becomes **exited_slashed**. An example is shown in the image below.



Source

Subsequent Inactivity Penalties

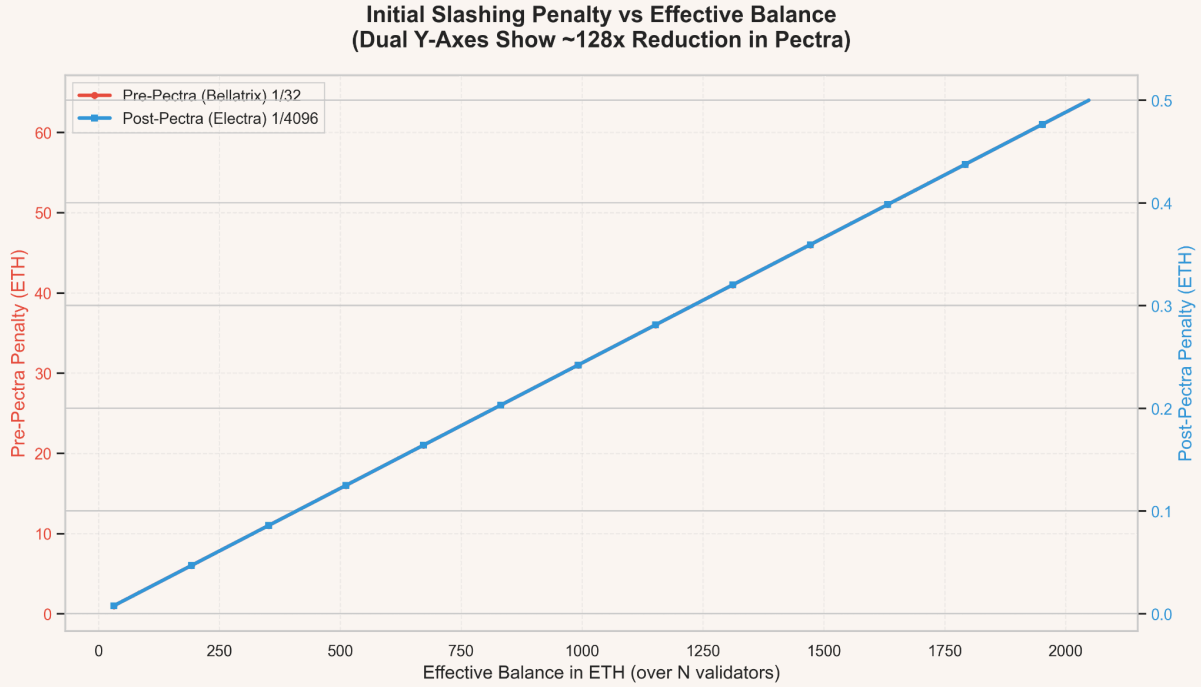
For about 36 days (8192 epochs), the validator is out of the active set and sits in the exit queue, earning no rewards and accruing inactivity penalties. Over this period, the validator loses an additional amount equivalent to ~62.5% of the potential consensus-layer rewards it would have earned in the same timeframe.

The chart below shows the initial penalty for a 32 ETH validator and a 2,048 ETH validator, highlighting how it scales with stake size.

Correlation Penalties

At about 18 days, when the midpoint of withdrawability is reached, the validator may incur a correlation penalty. This penalty scales with the share of the network slashed for misbehavior

during the ±18-day window around the event, lightly penalizing isolated incidents while severely escalating in mass-slashing scenarios (e.g., attempts to finalize conflicting blocks).



The calculation for the penalty is described on the Eth2Book website as:

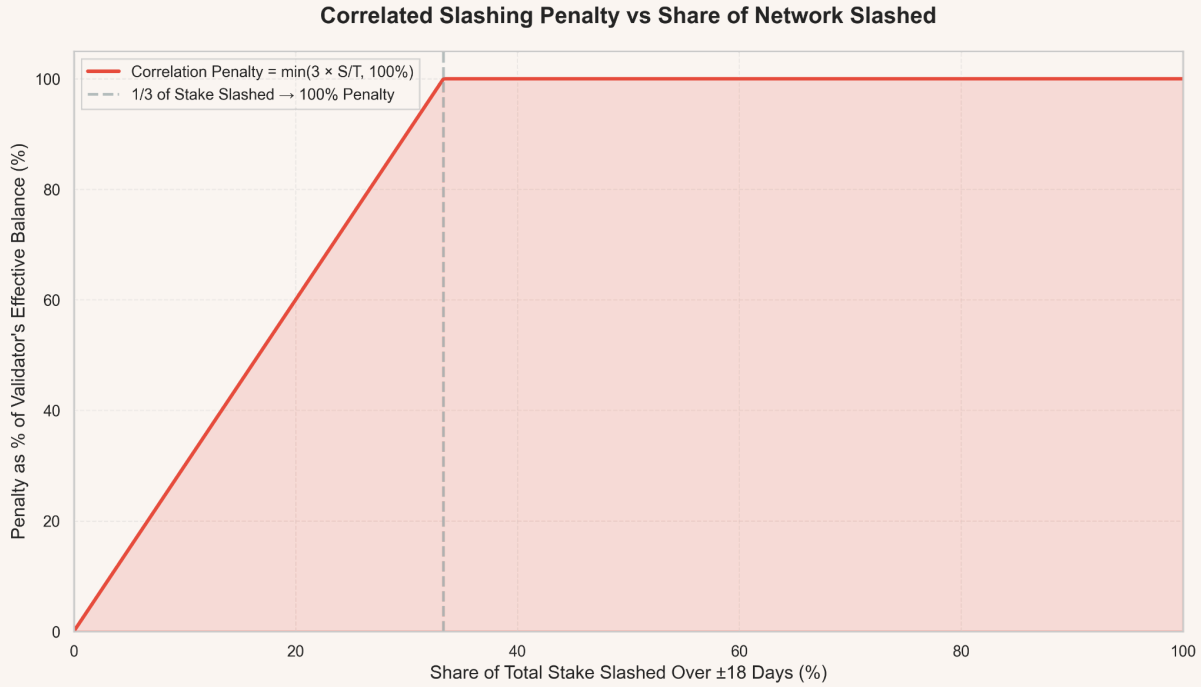
1. Compute the sum of the effective balances (as they were when the validators were slashed) of all validators that were slashed in the previous 36 days. That is, for the 18 days preceding and the 18 days following our validator's slashing.
2. Multiply this sum by 3, but cap the result at total_balance, the total active balance of all validators.
3. Multiply the slashed validator's effective balance by the result of #2 and then divide by the total_balance. This results in an amount between zero and the full effective balance of the slashed validator. That amount is subtracted from its actual balance as the penalty. Note that the effective balance could exceed the actual balance in odd corner cases, but decrease_balance() ensures the balance does not go negative.

$$\textit{Correlation penalty} = \min(B, SB/T)$$

where:

B - balance
S - total stake of validators slashed over the last 18 days
T - total ETH staked

In the extreme, the formula can penalize up to 100% of the slashed validator's stake. The penalty reaches 100% of the balance once ~1/3 of the network stake is slashed within the relevant window.



What you need to know as an ETH staker

Intentional vs non-intentional harm to the network

When token holders stake their assets on PoS chains, they should carefully research validator setups and put mechanisms in place to mitigate the risk of being slashed, since the cost can be extremely high. If the asset owner chooses to stake with a staking provider that runs validators on their behalf, they should enquire about and analyze, the precautions that the provider has in place to minimise the risk of slashing.

Beaconcha.in reports that, to date, 533 distinct validators have been slashed. While some operators have been slashed only once, others have had multiple incidents. The Ethereum network is designed so that slashing doesn't happen due to downtime, but only when network-harming activity is detected. However, network-harming activity doesn't necessarily indicate malicious intent, since validators may accidentally double-sign attestations and have their stake slashed. This usually occurs when you run your validator keys on two nodes at the same time (e.g., in a failover/redundancy setup) and your backup node comes online while your main node is still running. This can lead to double-signing blocks and attestations, which other validators will report to the network.

Preventing slashing at Twinstake

At Twinstake, we run validators through a remote signer that keeps a slashing-protection database, Dirk. Think of it like a logbook of everything a validator has already signed, including attestations. Before any new signature is sent, the system checks this log to ensure we never sign two different messages for the same duty. As this article explains, that's what triggers slashing. This matters most during restarts, upgrades, or failovers: without that history, a backup machine could accidentally "double-sign" while the primary is still active.

By centralizing signing and storing that history, we block conflicting signatures across machines and data centers. To support this, we implement enterprise-grade security, segregated key management, and strict change control. Twinstake also provides 24/7 monitoring and rapid incident response so institutional stakers maximize rewards while minimizing operational and slashing risks. In short, Twinstake's setup is designed to make the unsafe things hard and the safe path automatic.