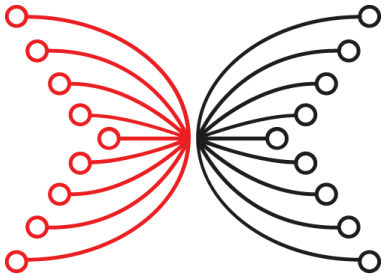
Security Issue Report



“Blocks from the far Future” bug

Severity: Low-Medium, potentially High if ignored for > 24h

Authors: Alexander Esgen

Date of Report: Feb 12, 2024

Table of Contents

1 Revision History

Version	Date	Change	Reason
1.0	2024-02-12	Initial version	

2 Summary of Issue

Date of Issue Ongoing. The issue has existed since the Shelley deployment.
Criticality Level Criticality: Medium (Overall); Potentially High (> 24h attack) If unfixed and exploited, the issue could have resulted in repeated network partitions, causing a significant loss of chain density (a Denial of Service attack). In case the exploit remains unanswered for more than ~24h, manual truncation of the ChainDB might be required. There is no evidence that the issue was ever exploited.
Context Blocks from the far future could cause nodes to disagree on whether they consider a block to be (in)valid, leading to a long-lasting network partition.
How was the Issue Detected The issue was discovered while investigating a DoS issue related to blocks from the *near* future, see this report.
What Action was Taken ● The issue was investigated by the Consensus and Networking teams ● A fix was produced and deployed in node version 8.8
Potential Effect Investigation showed that the issue could potentially be exploited by an adversarial actor to induce a network partition.

Actual Effect

None

Ongoing Mitigations Needed, if any

Release node 8.8 or later to mainnet, monitor uptake, and advise SPOs to upgrade.

In case this issue is actively exploited before node 8.8 or later is widely deployed, advise SPOs to restart their nodes as soon as they receive a block from the far future.

Responsibility for Mitigations

Core Tech/IOI

3 Fixing the “blocks from the far future” DoS vector

3.1 High-level context

In all cardano-node versions prior to the (upcoming) cardano-node 8.8, the Consensus layer’s treatment of “blocks from the far future” has undesirable consequences when such an “early” block that is on edge between being from the near and the far future is propagated through the network. Specifically, in such an event, the network might split into two distinct sets, disagreeing on whether the block is valid, causing a network partition. An attacker might repeat this process, partitioning the network into smaller and smaller sets, significantly reducing the chain density.

There is no evidence that this attack took place so far. It seems unlikely as affected SPOs would consider the mainnet chain to be invalid, and it seems very likely that they would raise this publicly.

However, there is a very simple mitigation: SPOs should restart their nodes as soon as they receive a block from the far future.

The purpose of this document is to describe how this attack vector has been resolved in the Consensus layer that has been integrated into the cardano-node 8.8.0-pre release.

3.2 Technical details of the DoS vector

Background on admissible clock skew

A node considers a block to be *from the future* if its slot is ahead of the current slot. Ouroboros Praos mandates that all chains containing blocks from the future (at that time) are ignored during chain selection.
As Praos assumes that all nodes have access to perfectly synchronized clocks, this will never cause nodes to disregard blocks that have been minted by other honest nodes. In an actual real-world deployment, this assumption is unrealistic due to the imperfections of protocols like NTP as well as leap seconds.
Even in the real world, the magnitude of clock skew between honest (in particular well-configured) nodes can reasonably be assumed to be bounded; for this, an *admissible clock skew* of 5 seconds is currently being used[1]. At a high level, the implementation distinguishes between blocks from the future to be either from the *near* or the *far* future.

- A block is from the *near* future if the onset of its slot is ahead of the wall clock, but only by at most the admissible clock skew. Despite being from the future, these blocks are assumed to potentially have been minted by honest nodes. These blocks are the cause of a different DoS vector, see this report.
- A block is from the *far* future if the onset of its slot is ahead of the wall clock by more than the admissible clock skew. By assumption, these blocks cannot have been minted by an honest node. They are the reason for the issue described in this document.

Prior to cardano-node 8.8, the node processed a new incoming block from the future like this:

1. The header and then block are downloaded via the ChainSync and BlockFetch mini-protocols.
2. The block is added to the ChainDB for chain selection.
    a. Before being validated, if the necessary preconditions are fulfilled (usually very likely), it is set as the tentative header for diffusion pipelining, and hence diffused through the network.
    b. *After* being validated, the in-future check is performed.
        i. If it is from the *near* future, it is not immediately selected, but still might be later on. The exact details do not matter for the purpose of this document.
        ii. If it is from the *far* future, it is considered as invalid, and its hash is stored in an in-memory cache of invalid blocks.
3. Now, if any peer sends a header/block *extending* a block that was previously judged to be from the far future, it can not be selected due to the aforementioned cache of invalid blocks, and the node will disconnect from such a peer.

[1] As advised by the Network team, we will soon reduce this value to 2 seconds.

The DoS dynamics

On Cardano mainnet, a diffused block reaches different nodes at different times, both due to different diffusion times and minor clock skew differences. Hence, if a block is diffused early by roughly the admissible clock skew (5s), nodes might disagree on whether it is from the near or far future. This can happen both by accident due to an insufficiently synchronized clock (although we haven't observed this so far, and it seems unlikely), as well as due to adversarial behavior. In particular, an adversary could leverage publicly available diffusion histograms (such as from pooltool.io) to inform their diffusion time offset. Even in the adversarial case, there is a significant portion of randomness involved, but it seems feasible to bring about this scenario with rather good chance.

Now consider what happens if a pool diffuses a block B such that some stake considers it to be from the near future, and the remaining stake considers it to be from the far future. Call the resulting disjoint sets of block-producing nodes N and F, respectively.

- The nodes in F won't consider B or any block extending it to be valid, hence disconnecting from all nodes in N, resulting in a network partition.
- Hence, the nodes in N and F will now mint on top of competing forks branching off before B of ever increasing length.

Unless there is some manual response, this fork will continue and eventually (~one day for a 50-50 split) result in different blocks being added to the immutable database, such that no reconciliation is possible without manual truncation/disaster recovery.

Note that the actual dynamics are complicated by the DoS vector due to blocks from the *near* future. This might make the issue described in this document more unlikely, but does not seem to be a fundamental obstacle.

3.3 Mitigation

In case this issue is actively exploited before node 8.8 or later is widely deployed, the network partition can be immediately resolved by advising SPOs that considered some block to be from the far future to restart their node (as this will clear the in-memory cache of invalid blocks).

3.4 The bug fix

The fix (implemented in #525) changes the handling of a block from the future like this:

- When receiving a header from the near future in ChainSync, an artificial delay is introduced until the header is no longer from the future.
- When receiving a header from the far future in ChainSync, we immediately disconnect from the corresponding peer.

The DoS is avoided as the entire handling of blocks from the (far) future in the chain selection logic is now effectively dead code[2] as we never download a block from the future (as we either introduced an artificial delay or disconnect upon receiving a *header* from the future). In particular, nodes won't record blocks from the far future as invalid, the root cause of the DoS.

See the corresponding paragraph in the report on the vector due to blocks from the near future for how we validated that the fix indeed causes the behavior just described.

[2] The Consensus team will remove the corresponding code in the future, but intentionally kept the bug fix as minimal as possible, in particular to avoid larger attention.