This post was co-authored with Nima Vaziri (@NimaVaziri). Special thanks to John Adler (@adlerjohn), Mustafa Al-Bassam (@musalbas), Ismail Khoffi (@ismail) and other members of the Celestia team for their reviews.

HackMD mirror: Spamming Attacks on Rollups - HackMD

Introduction

This document describes various attacks that target the publishing of rollup-specific messages on Celestia to prevent the safety and liveness of rollups. Inside parent chain blocks, these messages are held within shares reserved for namespace IDs which are unique to each rollup. Thus, in the attacks below, a wealthy client occupies all of the shares with its messages to prevent other users from posting their rollup-specific messages. The unifying theme of the attacks is the existence of a dedicated adversary with the financial means to spam the parent chain blocks with its messages.

Rollups on Celestia is prerequisite for a better understanding of this article as it assumes a basic knowledge of the rollups and namespaces.

Section Stakeholders and Assumptions lists the stakeholders of the attack and presents the security assumptions for the rollups which are subsequently violated by the attacks. Section Attack Description gives a detailed description of how the attacks work. Sections Costs, Discussion and Mitigation focus on the attack's costs, implications and mitigation respectively.

Stakeholders and Assumptions

Stakeholders that are affected by the attack are presented below:

1. Consensus (parent chain) nodes
2. Rollup full nodes under attack
3. Rollup light clients under attack
4. Attacker:

Any entity that can post messages to the parent chain can be an attacker, such as full nodes in malicious rollups.

As presented in Rollups on Celestia, following assumptions are required for the safety and liveness of the rollups. In the subsequent sections, we will refer to these assumptions to clarify which ones are violated by the attacker.

For liveness;

1. RU-Liveness-1:

Rollup has an honest aggregator or block producer.

1. RU-Liveness-2:

Rollup-specific messages, e.g rollup blocks and associated transactions, are published on the parent chain within a bounded time after they are created. This is necessary for a new rollup state to become finalized in the view of all nodes in a timely manner.

For the safety of optimistic rollups;

1. ORU-Safety-1:

There exist verifiers, which challenge invalid state roots appearing on the parent chain via fraud proofs or dispute resolution protocols. Fraud proofs and the messages exchanged within the dispute resolution protocol are created within bounded time.

1. ORU-Safety-2:

Fraud proofs or messages related to dispute resolution protocols are delivered to the rollup light clients over the peer-to-peer network and processed before the dispute time window

expires in their view.

1. ORU-Safety-3:

Fraud proofs or messages related to dispute resolution protocols are published on the parent chain in the view of the rollup light clients and processed before the dispute time window

expires in their view.

Attack Description

Attack on the Liveness of the Rollup State

Figure 1: Attack on the Liveness of the Rollup State. Attacker publishes blocks from rollup A on the parent chain so that there is no space left for the blocks from rollup B. Thus, rollup full nodes and light clients in rollup B cannot finalize the blocks B_{j+1}

, B_{j+2}

and their descendants for the duration of the attack.

In this version of the attack, RU-Liveness-2

is violated. Attacker spams the parent chain with its messages from rollup A for the duration of the attack by e.g. offering larger transaction fees to the consensus nodes (Figure 1). Then, block producers in other rollups, e.g rollup B, cannot publish their rollup-specific messages on the parent chain as all of the parent chain blocks would be occupied by the attacker's messages. This stops the rollup nodes in B from updating their states for the duration the attack as the new rollup blocks cannot be built before the older ones and their transactions are posted on the parent chain1

.

Attack on the Safety of the Rollup State as Viewed by Light Clients

[

attack2

893×408 26.8 KB

](https://forum.celestia.org/uploads/default/original/1X/c053230bf4c7f75683eda4906e9021572957bfd9.png)

Figure 2: Attack on the Safety of the Rollup State as Viewed by Light Clients. Malicious aggregator on rollup A produces A_j

with an invalid state root. It is posted on the parent chain and the verifiers on rollup A

attempt to notify the nodes in some other rollup B

, which act as light clients towards rollup A

, by creating fraud proofs. However, attacker again occupies the parent chain block with its messages. Thus, the fraud proofs cannot be published on the parent chain during the dispute time window following the appearance of the invalid block in the view of the light clients for rollup A

(e.g nodes in rollup B

). Hence, the invalid rollup block becomes finalized in their view of at the end of the dispute time window.

In this version of the attack, we consider optimistic rollups with applications depending on each others' states. Thus, the full nodes on each rollup act as light clients towards the other rollups, and they use the parent chain as a bridge to receive fraud proofs or related messages from these other rollups. Hence, the validity of the other rollup states observed by the light clients depends on ORU-Safety-3

, which is violated by the following attack.

A malicious aggregator on rollup A produces a block with an invalid state root Figure 2). Header of the invalid block is posted on the parent chain and the verifiers on rollup A attempt to notify the light clients from rollup B by creating fraud proofs or engaging in a dispute resolution protocol. However, attacker occupies the parent chain blocks with its messages as in Section Attack on the Liveness of the Rollup State Thus, the fraud proofs and messages related to the resolution protocol cannot be published on the parent chain during the dispute time window following the appearance of the invalid block in the view of the light clients from rollup B. Hence, the invalid rollup block becomes finalized in their view at the end of the dispute time window2

.

This attack concerns only the case where the light clients accept fraud proofs via the parent chain, which gives the attacker a limited space it can spam with its messages. Attacker cannot influence the validity of the finalized states to the same degree under ORU-Safety-2

when fraud proofs are disseminated over a peer-to-peer network, as these networks are harder to spam with sufficiently

many relay nodes.

Costs

Attacker's costs equal the fees it pays to occupy the portions of the parent chain that would have originally be consumed by the rollup blocks or fraud proofs for the duration of the attack. Note that the attacker has to buy all of the shares occupied by the attacked rollup as well as those that pay less. Suppose the rollup pays P

dollars per share in transaction fees to post rollup blocks and F

dollars per share to post fraud proofs. Furthermore, assume that each parent chain block consists of S

shares and the attack lasts for T_l

or T_s

consecutive parent chain blocks. Here, T_l

and T_s

correspond to the disruption time window for liveness or the dispute time window for state safety respectively. Thus, the costs of the attacks described in Sections Attack on the Liveness of the Rollup Stateand Attack on the Safety of the Rollup State as Viewed by Light Clients are upper bounded by P \cdot S \cdot T_l

and F \cdot S \cdot T_s

respectively.

If (i) there are 1024 data shares with namespaces in a Tendermint block, (ii) rollup pays $9 per share for both the blocks and the fraud proofs ($9 is an estimate on par with Ethereum transaction fees), and (iii) there is one block per minute, a rough estimate for the costs would be P \cdot S \cdot T_l = F \cdot S \cdot T_s = 1024 * 9 * (7*24*60)

, which is about 92

million USD for a dispute or disruption time window of one week. This is a reasonable cost for a wealthy adversary to potentially steal billions of dollars via an attack on state safety.

Discussion

A major question for the attacks above is whether the attacker's behavior actually constitutes an attack in the first place. After all, who is to claim that the rollup-specific messages pushed out of the parent chain are more important than the messages published by the attacker? In an economic framework where the importance of data is measured by the fees required to publish it, one can easily ignore the attacks above as artifacts of the users' preferences. However, this reasoning overlooks the importance of innovation by new applications. Financial constraints imposed by existing applications that buy up all available space on the parent chain could potentially stifle the development of new ones using Celestia for data availability. Hence, if such attacks become prevalent, instead of incentivizing a multi-chain world, Celestia could be strengthening the monopoly of a few by allowing them to push emerging ones out of the parent chain.

Mitigation

The most naive mitigation for the censored rollups is to offer higher fees than the one proposed by the attacker. If the attacker raises the fees, full nodes on the censored rollup can also run consensus nodes which would include the rollup-specific messages in their proposals. However, with stake that was insufficient to exceed the attacker's fees in the first place, these nodes would rarely be able to propose a parent chain block if any at all. Moreover, this mitigation highlights another attack vector for the censorship of the rollup-specific messages: Attackers can run their own validator nodes or bribe the existing ones outside the transaction fee mechanism to reject messages from the attacked rollup. Such a collusion happening off-chain is impossible to prevent via the native mechanisms of the protocol and highlight the need for a socially coordinated strategy for including messages from the censored rollups in the parent chain.

The strongest form of mitigation against a financially strong adversary is coordination on the social layer

to (i) identify the attacker, (ii) evaluate its messages, (iii) blacklist its messages if they do not serve any rollup or application. This endeavor could be helped by altruistic consensus nodes that accept messages from censored rollups. However, the parent chain should stop short of introducing any logic, e.g. in the Celestia app, to service censored messages as this would imply favoritism. Hence, mitigation should stay strictly in the realm of social layer as it requires a careful evaluation of the attacker's behaviors in terms of whether it constitutes an attack or warrants mitigation.

Footnotes

1: The same attack applies to rollups that are built on parent chains with a native execution engine and smart contracts. For instance, in Arbitrum which is built on Ethereum and where users send their transactions directly to an on-chain contract, the attacker can still prevent liveness by spamming the Ethereum blocks with its transactions up to the gas limit.

2: The same attack again applies to rollups on parent chains with smart contracts. For instance, in Arbitrum which relies on interactive dispute resolution protocols, the attacker can target the final step of the protocol which requires the participation of an on-chain contract to verify the execution of a single instruction. To avoid the contract from executing this step, attacker can again spam the Ethereum blocks with invalid transactions up to their gas limit during the dispute time window.