P2P storms overview

submitted to security@interchain.io
10/2/2023

TL;DR:
The mempool is severely flawed because it is too chatty and this can result in both chain halts and attacks over IBC using IBC transfers with 50kb IBC memo fields.

This manifests first on chain A, and then later on chain B as the relayers move the packets and the same gossip conditions occur on B's P2P network.

The attack has very clear stages:

They combine, however, into a single attack that has a clearly defined sequence.

1. **Block Gossip (really only matters >10mb blocks):** tx payload begins, usually at a rate of about 30/s but this varies based on your equipment and how you deploy the payload. In general, it is possible to trigger P2P storms with a simple bash loop. once the first full 21mb block is proposed, P2P from block gossip increases to a few hundred times the normal level and will continue that way until the network is no longer proposing full blocks.

2. mempool: So the mempool fills up, and it has its own gossip issues. This portion also kicks off a few hundred times normal. Mempool traffic is about equal to block gossip traffic.

3. The mempool on the attack node fills and returns code 20 (but not reliably) and You can choose to multiplex the attack by filling the mempool's of multiple nodes, but we never needed to do this in order to have devastating effect

4. The tx stream stops, and as long as blocks are full and the mempool is processing, the network will have a few hundred times + a few hundred times normal P2P traffic. If there is another node attacking, the traffic will increase further.

5. when the mempool is empty, there will still be P2P problems that aren't yet explained. In our testing, these problems lasted for several days.

As of 6:12PM on 10/3/2023, the cosmos hub replicated security testnet is making a block once every eight minutes due to the submission of a transaction stream that took it offline.

Video

https://drive.google.com/file/d/1J3SYI9xt7Z7VjyECweUDI5Ub1IgPf6at/view?usp=drivesdk

Likely solution:

https://github.com/cometbft/cometbft/pull/1426

Original p2p storms doc is here:

https://docs.google.com/document/d/1oCjsVYMaV77etxOEbDxh58vkAQaXf7RAkhXvF_8GYis/edit?usp=drivesdk

Code for p2p storms is here:

GitHub.com/notional-labs/spammy

Scoping aligned with:

https://hackerone.com/cosmos/policy_scopes

Mitigations Doc

https://docs.google.com/document/d/1QcTae_LIUyGFuSn917S5NdM-Qbn-c_2CGikEUDrYBq4/edit?usp=sharing

Comet Scope Only

1) Banana king transactions violate maxtxbytes
    a) MaxTxBytes _is_ still used, but only on accepting txs into the mempool, on a per node config basis. It has no bearing on txs in blocks, and is irrelevant in our context. - joe
        i) Maxtxbytes ONLY affects the local node in question
2) Reaping mempool algo too slow
    a) Suggestors
        i) PFC
        ii) Dev Ojha

3) when 21mb blocks + full mempool, nodes consume 1gbps
4) default block sizes too large (comet)
    a) We should reduce to let's say 5mb
    b) Note: cosmos hub currently has 200kb blocks - Jerry Chong
        i) gaiad q params subspace baseapp BlockParams | jq

```
{
  "subspace": "baseapp",
  "key": "BlockParams",
  "value": "{\"max_bytes\":\"200000\",\"max_gas\":\"40000000\"}"
}
```

5) when the mempool is full the rpc doesn't reject txns
6) Validator set griefing via excessive p2p traffic
7) consensus issues resulting in an invalid proposer signature error
    a) This can be replicated by using the attack approach in the spammy repository
    b) Zaki mentioned that this could be as a result of a non-proposing validator not getting the block correctly over p2p
8) observed instances where transactions will simply remain in the mempool of validators only instead of being properly reaped after a p2p flood (credit to Clemens)
    a) Could be related to the lack of optimization in the mempool reaper
9) When we transmit block parts, every node broadcasts all of the block parts to all of its peers indiscriminately
10) Block size is likely being measured incorrectly, since the hub has 200kb blocks, and client updates are ~140kb and numerous blocks have multiple client updates
11) Massive attack: fill five nodes m no empools when they are not connected to the rest of the network, with different transactions each, then connect all at once

1) **IBC scope only No field length limitations Banana king transaction illustrate the need for this**
    a) **https://www.mintscan.io/osmosis/tx/D62F0F0354C4DEA0D9DFCA596D9BC3F2943DBA7D24818009DFD725F883088DD0**
    b) **https://www.mintscan.io/osmosis/tx/BE8ACBFAD7FB0B27F0886B879D14B33FCE01230382C51DEB1125D02FBBD56703**
    c) **https://www.mintscan.io/osmosis/tx/6DAF6899BB6EC16D7A55ED86DF47AC6A4F93DE449F9C9912751E27C091D4BB51**
    d) **https://www.mintscan.io/osmosis/tx/BB7192A30EF7A7CC9156B2C6A7C9F839250901888995DE370C2B8BBE3220ECC7 Multi Meg Banana King**
    e) **https://www.mintscan.io/osmosis/tx/F506105FAB8A7D26D950043B732C935A2896733C7AEB695F729CDC0631963252 Banana memo transactions illustrate the need for this**
    f) **https://explorer.rs-testnet.polypore.xyz/provider/blocks/3478248 Client update transactions are too large, so spamming them (even when valid) can be harmful Longer term fix Banana king and banana memo can be used**

to make chain a attack chain b This is proven between producer and pion Multiple attackers can create multiple mempool states resulting in increased mempool gossip

SDK scope only

1) Banana king txns show that message size limitations don't work
2) signature verification gas costs too low
3) We should increase the cost of bytes by 10x

Hub, Juno, Noble scope only

1. Max block size is probably not correctly enforced
2. ~~Testnet was not properly configured: resolved by notional and hypha with alert by jerry chong~~
    a. testnet had 22mb blocks while mainnet had 200kb blocks and this prevented real world testing
        i. https://explorer.rs-testnet.polypore.xyz/provider/blocks
        ii. With 200kb blocks and a mempool full of txns with 50kb ibc memos, the attack is different, but equally effective. The rs testnet is unusable, and unless validators intervene, it will be until late oct 5 or early oct 6
3. Global fee bypasses could lead to free attacks
    a. MsgAcknowledgement
    b. MsgReceive
    c. MsgUpdateClient
    d. MsgTimeout
    e. MsgTimeoutOnClose

Social Scope

There were very significant communications and procedural failures associated with reporting the p2p floods issue that meaningfully slowed both problem identification and disclosure of the issue.

- Allinbits, inc was contacted several times, they said they would have engineers look into it but didn't.
- We have received zero support from Amulet.
- Informal Systems and Amulet do not seem concerned with this issue. All other teams are quite concerned.

- We did goe support from
    - Cryptocrew
    - Hypha (all team members)
    - Dev Ojha (osmosis)
    - Zaki Manian (Iqlusoin)
    - Marko Baricevic (Binary Builders)
- Amulet published the vulnerability against the wishes of the Notional team, which increased the urgency of fixes across all of cosmos, and put all of cosmos at risk.
    - Ethan Buchman then put blame for a chaotic situation on Notional, calling our team unprofessional
        - In reality Cosmos is not very open to security research.
        - This limits the health and viability of cosmos overall.
    - When amulet published the vulnerability they were in possession of a video demonstrating how to use a P2P storm to disable the cosmos hub replicated security testnet.
- Amulet refused or ignored every offer of opportunities to directly communicate so that we could help them understand the p2p storms issue.
    - The lack of communication is the likely reason that amulet made the disastrous (So far, only for notionals conference schedule) choice to publish.
- Amulet claimed to at least two interchain foundation team members that our emails contained threatening language. No such threats existed, and all email comms with Amulet are in the same GitHub repository containing the code pursuant to the attack for others to review.
    - I encourage amulet to bring forth this threatening language
    - After review of our email comms no one at icf thinks there's any threats