Sentry Nodes

Make Nodes Resilient To DDoS Attacks

The Sentry Node Architecture is an infrastructure example for DDoS mitigation on Tendermint-based networks.

Sentry Node Architecture Overview Secret Nodes (Validators) are responsible for ensuring that the network can sustain denial of service attacks.

One recommended way to mitigate these risks is for validators to carefully structure their network topology in a so-called sentry node architecture.

Validator nodes should only connect to full-nodes they trust because they operate them themselves or are run by other validators they know socially. A validator node will typically run in a data center. Most data centers provide direct links the networks of major cloud providers. The validator can use those links to connect to sentry nodes in the cloud. This shifts the burden of denial-of-service from the validator's node directly to its sentry nodes, and may require new sentry nodes be spun up or activated to mitigate attacks on existing ones.

Sentry nodes can be quickly spun up or change their IP addresses. Because the links to the sentry nodes are in private IP space, an internet based attacked cannot disturb them directly. This will ensure validator block proposals and votes always make it to the rest of the network.

For those implementing Sentries on Validators who already have Public IP exposed. Currently any peer, be it a validator or full node, is given 16 attempts with exponential backoff, which in total amounts to around 35 hours, to connect. If the node remains unreachable then it is automatically removed from the address book. An unreachable validator node is not gossiped across the network i.e. all other nodes will each try to connect to the unreachable validator node before removing it from their address book.

Get Node Peer IDs

Log into your sentry node(s), and validator, then run the following commands to get the peer information:

Get node id

```

Copy secretdtendermintshow-node-id

```

```

Copy ipr

```

Save your peer information, be sure to remember which are for sentries and which is for your validator, you'll need it later:

```

Copy @:26656

```

Setup Sentry Node

To setup basic sentry node architecture you can follow the instructions below:

Sentry Nodes should edit their config.toml:

First follow theFull Node Guide

Edit the full nodes config file you want to use as a sentry node:

```

Copy nano/.secretd/config/config.toml

```

Proceed to add the peer id of your validator to the.secretd/config/config.toml :

```

Copy
```

Comma separated list of peer IDs to keep private (will not be gossiped to other peers)

Example ID: 3e16af0cead27979e1fc3dac57d03df3c7a77acc@1.4.7.7:26656

private_peer_ids="node_ids_of_private_peers"

```

Now proceed to restart your secret node with the following command.

```

Copy sudosystemctlrestartsecret-node

```

You now have a sentry node running!

Place Validator Behind Sentry Nodes

Validators nodes should add their sentry node peer information to their.secretd/config/config.toml :

```

Copy nano.secretd/config/config.toml

```

Proceed to add the peer id of your sentry nodes to the persistent_peers list and set pex to false:

```

Copy

Comma separated list of nodes to keep persistent connections to

Do not add private peers to this list if you don't want them advertised

persistent_peers=[list of sentry nodes]

Set true to enable the peer-exchange reactor

pex =false

```

Now proceed to restart your secret node with the following command.

```

Copy sudosystemctlrestartsecret-node

```

You're now running your validator behind a sentry node!

Resources:

https://github.com/cosmos/gaia/blob/master/docs/validators/security.md

Last updated1 year ago On this page *Make Nodes Resilient To DDoS Attacks* Get Node Peer IDs * Setup Sentry Node * Place
```

Validator Behind Sentry Nodes * Resources:

Was this helpful? Edit on GitHub Export as PDF