How to avoid the Fabric CA beeing a single point of failure?

Asked 2 years, 5 months ago Active 2 years, 5 months ago Viewed 1k times

5

if I understood correctly, every peer in a fabric blockchain network (somehow interconnected through gossip) will only accept incoming connections from other peers if they use a HTTPS connection with a public key signed by the Fabric CA.

Is that correct?

3

So in my understanding, the Root-CA becomes the single point of failure because one could modify it and from then on modified Root-CA certificates will propagate to the nodes and eventually no node can connect to each other anymore.

Is this correct?

hyperledger hyperledger-fabric

asked Jul 25 '17 at 16:37

Romeo Kienzler
2,202 21 37

4 Answers

3

Let me try to answer the two questions also, perhaps a little more directly.

QUESTION1: if I understood correctly, every peer in a fabric blockchain network (somehow interconnected through gossip) will only accept incoming connections from other peers if they use a HTTPS connection with a public key signed by the Fabric CA. Is that correct?

ANSWER1: No, this is not correct. You said "the Fabric CA", but each fabric blockchain network has multiple trusted CAs where each may be a Fabric CA or another CA or a combination. There is no single trusted CA root in this model. Also, the connections from peers are over GRPC rather than HTTPS.

QUESTION2: So in my understanding, the Root-CA becomes the single point of failure because one could modify it and from then on modified Root-CA certificates will propagate to the nodes and eventually no node can connect to each other anymore. Is this correct?

ANSWER2: No, this is not correct. There is no SPoF (Single Point of Failure) because: a) a single Fabric CA can run in a cluster b) there are multiple Fabric CA clusters (or other CAs) in a blockchain network. c) the peers and orderers do not connect directly to a CA. They operate off of crypto material that is locally available from the file system or its copy of the ledger. There is also no SPoT (Single Point of Trust) because: a) their are multiple root CAs without a common root key, and b) configuration updates which affect who trusts whom may require signatures from multiple identities from different roots of trust. For example, changing a trust policy could require signature from an administrator from every organization in the blockchain (or in hyperledger terminology, in the channel).

answered Jul 27 '17 at 21:57

Keith Smith
276 1 5

1

Peers will accept incoming connections from other peers and orderers. You define which members are going to take part in a channel, i.e. who are going to take part in a mini Blockchain inside your network. Then, you create the artifacts for each member. You have more information about the channels and the artifacts that you should create, here. And more info about the tool that you will use here.

Once you have created the channel and joined the peers to it, the connections are controled by the MSP. When you

As you said, the Root-CA could be modified, but that could happen in any other system with any other Root-CA. The Fabric CA Server should be switched on when the members are requesting their keys, then, it would be stopped. Also, Hyperledger recomends to create intermediate CAs.

answered Jul 25 '17 at 17:15

Urko
1,257 1 10 18

1

The answers from varnit and Urko address the question in part. However, there are many facets to consider when determining whether the Fabric CA presents a SPoF. First, the Fabric CA can be made highly available as noted in the response from varnit. However, the Fabric CA is not required for operation of the blockchain network, it can be used by the SDK or by CLI to obtain certificates that are used to configure the peers and orderer(s) in the network and the channels over which transactions will be transacted. It is possible to create the certificates that you need when you configure the network without the Fabric CA entirely using the `cryptogen` tool. In the manual of the Fabric there is defined here. To configure the network you will use the `configtxgen` tool.

When configuring a network, the certificates representing each organization role are stored in the genesis block of the network, and when configuring a channel, in the channel's configuration block. Hence, each node, whether a peer or orderer, has access to all of the (root) certificates. The only way to change the root certificates of the various organizations would be to get a validated transaction to update the configuration of the network agreed per the endorsement policy defined for that network.

edited Jul 26 '17 at 11:44

Urko
1,257 1 10 18

answered Jul 25 '17 at 19:04

christo4ferris
3,843 1 12 29

0

First of all, I would like to say that the question is very interesting secondly I think your concerns are true about Hyperledger Composer but a solution I would say that because the all the Hyperledger Fabric components are container based they can be easily scaled so in the case of Docker swarm I would just use `

```
docker service scale hyper ledger-ca 5`
```

and it will scale it to 5 containers or different nodes i hope that answers your question please let me know if there is anything left to answer

edited Jul 25 '17 at 18:59

christo4ferris
3,843 1 12 29

answered Jul 25 '17 at 17:04

varnit
1,401 5 19

I think that @Romeo Kienzler didn't ask about hyperledger-composer or about how to scale them. I'm sorry if I hadn't understood your question. If that, tell me, please. – Urko Jul 25 '17 at 17:20

well he asked about single point of failure and my answer was to that question – varnit Jul 25 '17 at 17:25