22nd Mar, 23 - DKG issues in Lido

Status	Closed
Severity	Minor

Summary

Users in the lido discord server faced several issues while performing DKG ceremonies amongst themselves. This postmortem doc is a report on what went wrong and what we could do to mitigate scenarios like this.

Timeline

- START TIME: 21st Mar, 23
- END TIME: 22nd Mar, 23
- LENGTH of the disruption: 24 hrs

Incident Team

Report Author	Dhruv, Abhishek
Incident Reporter	Dhruv
Incident Commander	Dhruv, Abhishek

Impact

- What impacted
  - *charon dkg*
- Who impacted
  - Users in lido discord server trying to do DKG
- How impacted
  - Some users didn't receive *keystore* files at the end of the DKG. Others had file permissions issues. In short, unsuccessful DKGs for some users.

Detection

The incident was detected by Dhruv who was helping users in the lido discord server to conduct DKGs.

Resolution

We discussed the final solution internally. First, we recommended users to use v0.13.0 to perform the DKG as there were bugs in v0.14.0. But, this was a short-term solution as several users were still getting different kinds of errors. We proposed to patch release charon v0.14.3 with the fix. This was properly discussed, tested and then communicated to the users in lido discord.

Root Cause

The root cause can be attributed to a bug in charon latest release. This was exacerbated by bad communications between the core team and lido users about the fix.

Prevention Actions

Action	**Owner**	**Status**

Mitigation strategies:

1. **Before every DKG run**:
    a. Ensure all the users run the same version of charon, you can now use v0.14.3.
    b. Ensure you have only one file in *.charon/* directory i.e., only `*charon-enr-private-key*` and nothing else.
    c. Make sure to backup *charon-enr-private-key* in the .charon folder (without this key, we'll have to start from scratch).
    d. If you encounter a permissions error in DKG run on *linux*, then follow the post-installation steps of docker: https://docs.docker.com/engine/install/linux-postinstall/ . If the error still persists, do: `*chmod -R 777 .charon*`
    e. Ensure there is no other charon container/process running during the DKG ceremony. ***docker compose down*** all containers related to the current cluster in *charon-distributed-validator-node* directory.
    f. Don't use the same machine to run two DKG ceremonies at the same time with the same `charon-enr-private-key`. This can have side-effects.
    g. Avoid running two DKG ceremonies in the same machine even with different `*charon-enr-private-key*` files. This can lead to mismanagement of the *validator_keys* directory by the user. If you still do multiple DKG ceremonies on the same machine, please ensure they are on separate directories/locations.
2. **After every DKG run**:
    a. Make sure each participant has received *deposit-data.json*, *cluster-lock.json* and the required number of keystores in *validator_keys* directory in the .charon directory. The directory structure should look like this:
        i. .charon
        ii. ├── charon-enr-private-key
        iii. ├── cluster-lock.json
        iv. ├── deposit-data.json
        v. └── validator_keys
        vi.   ├── keystore-0.json
        vii.   └── keystore-0.txt
    b. The coordinator of the DKG **MUST** ensure that all the DKG participants have all the required keystore files, the cluster-lock and deposit data file.
    c. If there is any missing file for any participant, then the DKG has to be performed again.

d. Also note that the participants should wait for confirmation from the coordinator that the DKG is done correctly. Only then the participants should go ahead and do ***docker compose up****,* ie, run the charon stack.

DKG QA

As a side effect, we also made a list of QA items for charon dkg. We should perform diligent QA on our DKG process before every release. Here are some ways:

1. Long Wait DKG
    a. Purposefully delay a peer to perform DKG by like 4-5 minutes.
    b. This will test DKG in relay re-connections.
2. One peer per minute DKG
    a. Start a DKG peer at the end of every minute
3. Validator_Keys/ already exist
    a. Have a `.charon/validator_keys/` directory already existing and then do DKG.
    b. Have a `.charon/cluster-lock.json` file already existing and then do DKG.
4. Large DKG
    a. Perform DKG with a large set of users (>=10)
5. Multi machine DKG
    a. Perform DKG on multiple different machines with different OS/Arch.
    b. Linux, Windows, Mac, VM combinations
    c. Make sure we also test on windows.
6. Run charon dkg and charon run simultaneously on the same folder.
    a. First run "*docker compose up*" with existing keys. Basically, run the whole charon stack
    b. Then, delete *validator_keys/* directory.
    c. Then, run *"charon dkg"* again inside the same folder. See if it succeeds.

Events detailed timeline

A detailed list of events and actions that happened from the detection or reporting of the event until its resolution. Additional information or observations that might be relevant.

Relevant links

https://docs.google.com/document/d/1Pfcv3NUTlJpMF_tmOQbSwWHYN0Qv1GzgIsKG8XLkWUM/edit#