LIDO on Ethereum RockLogic GmbH Slashing Incident - April 13, 2023

Detailed Report of the Incident

Rocklogic GmbH, April 18, 2023

1. Incident Summary

Background

On April 11th, 2023, we discovered that one of our node's client databases had become corrupted, necessitating the transfer of keys to a new node. To do this, we removed the keys using the validator key API and imported them into a new node. We then proceeded to resynchronize the old node, which had been up and fully synced for a few days without showing any signs of the keys being imported in the client, which could have led to slashing.

On April 13th, we restarted and performed the Shapella upgrade on the old node running Prysm and Nethermind, which unfortunately led to a bug that caused the keys to regenerate within the validator key store. This is considered a slashable offense, and we reacted quickly as soon as we noticed it happening.

Despite our efforts, 11 of the validators we operate were slashed. We deeply regret this outcome and have taken steps to address the issue and prevent similar incidents in the future.

2. Statement

We sincerely apologize for the incident and the damage it has caused to all those directly involved. We now understand that immediately 'nuking' or destroying the node in question could have prevented this damage, and it was our mistake not to take that measure. Although 'nuking a node' or destroying it was recommended during the early days of Ethereum development, the excellent work of the ETH client teams over the years has made it an extreme measure in our experience.

As we have extensively and regularly used the clients, we had confidence in their work and chose not to pursue that option, instead attempting to resolve the problem in a less intrusive manner. However, we believe that it is a positive outcome for Ethereum and its community that this problem occurred to RockLogic operating on LIDO, rather than anyone else.

Our analysis, in collaboration with LIDO and Prysm, revealed that the bug that caused the incident could potentially affect any staker using the Prysm client, even those who do not use our Stereum software. Identifying and disabling the bug as quickly as possible, making it reproducible for further in-depth analysis, and doing so in a joint effort with LIDO, Prysm, and RockLogic, demonstrates the

commitment of the Ethereum community to constantly improving the security of the Ethereum blockchain for the benefit of all users and contributors.

The RockLogic team is proud to be part of this community and is committed to continuing our efforts in securing Ethereum development, even in the face of challenges. We want to express our sincere thanks to LIDO for enabling us to continue providing our services and contributing to Ethereum, and for the professionalism of their team in turning this unfortunate incident into a valuable learning experience and another opportunity to make Ethereum better. We also extend our appreciation to the Prysm team for their assistance in confirming and immediately resolving the issue.

We have taken key learnings from this incident and are implementing measures to prevent similar incidents in the future, as outlined in section 4.

3. Incident Description

Timeline

11.04.2023 - 7:30 UTC	500 Offline Vals	The Corruption of a Nethermind Database caused 500 Validators to be offline, so the NO migrated the keys onto another machine. (remove keys from cluster A and import keys into cluster B)
11.04.2023 - 9:30 UTC	Resync Nethermind	The resync of nethermind was initiated to get a backup-node up.
12.04.2023 - late morning	est. time of Nethermind finish syncing	At this point Nethermind should definitely finish syncing. If the keys were not removed, it would have started staking right away.
13.04.2023 - 12:27 UTC	Update Prysm	Prysm Docker image was updated from 4.0.1 to 4.0.2 for Consensus and Validator Client. The containers were restarted afterwards to apply the update.
13.04.2023 - 12:50 UTC	First slashings	Slot 6213852 with the first 2 slashings

13.04.2023 - 1:13 UTC	Shutdown VC	Shutdown of the Prysm validator of the node that was causing the slashing.
13.04.2023 - couple minutes later	Nuke node	Complete vanish of the node that caused the slashing.

Root Cause

The root cause was double votes of validators imported on 2 different nodes. This duplication was due to an image version update followed by a reboot of Consensus and Validator Client (Prysm) to apply the update (4.0.1 -> 4.0.2). It seems that this process caused some kind of re-import of the previously deleted keys. However, destroying the node beforehand would have prevented this issue in the first place.

4. Key Learnings

Encountering a severe problem, despite precautions taken to avoid it, and being held responsible is never easy.

Still, this it the kind of incident we all can learn from, and the most beautiful key learning for us was that the community works perfectly in the way how all parties concerned acted immediately and jointly to prevent any more damage.

In this sense, these are the measures planned for the future:

Action Points

- Further investigations to verify the root cause of the key re-import.
- Reproduction of the bug causing the failure (see section 6)
- Expand internal monitoring
- Security checks of client configurations (eg. if doppelgänger is enabled)
- Documented and clear instructions for the migration of keys.
- Review other processes to see if bugs like this could possibly cause similar outcomes.
- Schedule additional automized tests for such cases
- Tighten up the process of moving keys and configs to eliminate the risk of running into a bug

5. Detailed Report By LIDO

Isidoros Passadis provides us we a thorough report of incidents, the impact it caused, recommended action points and a list of the slashed validators here:

https://blog.lido.fi/loe-rocklogic-gmbh-slashing-incident/

6. Reproduction Of The Causing Bug

Reproduce Prysm Validator Key not correctly Imported / Deleted via Keymanager-API

1. Install Prysm Node

In this exampe we'll install Prysm via Docker. You can find a guide for this here.

2. Create Wallet

You can find an easy guide for this here.

3. Check Requirements for the next steps

After the step above you should have

- a running Prysm Beacon Client
- a running Prysm validator Client
- a wallet and access to following file: 'wallet-dir/accounts/all-accounts.keystore.json'
- a validator key keystore to test Import / Delete

4. Change persmissions of all-accounts.keystore.json

Make sure you're only changing the permissions of this file and no other directories:

```
cd /data/wallets/direct/accounts
chmod 700 all-accounts.keystore.json
```

In the commands above we direct to the **all-accounts.keystore.json** file which should be located somewhere In the specified wallet directory. We then change the permission to **700.**
This file should normally have **600** permissions.

5. Import Validator

So if you have your validator port exposed you can simply use curl. If not, you could use the docker curl image curlimages/curl to reach the VC docker container, but you have to make sure that these containers are in the same network. You'll also need the API-Token that is created by the Prysm. The import curl command would look something like this:

```
curl http://<VC-Endpoint>:7500/eth/v1/keystores -X PUT -H 'Content-
Type: application/json' -H 'Authorization: Bearer <TOKEN>' -d '{
"keystores": ["<KeystoreContent>"],
"passwords": ["<Password>"],
"slashing_protection": "<SlashingProtectionContent>"}'
```

If the permission of **all-accounts.keystore.json** was set to 700 previously, then the result should be something like this:

{"message":"Could not import keystores: could not write accounts: file already exists without proper 0600 permissions ","code":500}

6. List Validator

This message inidactes that something went wrong. Of course because the permission of **all-accounts.keystore.json** was not set correctly. But if we now list the keys:

```
curl http://<VC-Endpoint>:7500/eth/v1/keystores -X GET -H 'Content-Type:
application/json' -H 'Authorization: Bearer <TOKEN>'
```

You'll then see that your key is listed in there although it seamed to fail to import previously:

{"data":[{"validating_pubkey":"<pubkey>","derivation_path":""}]}

If you now restart your VC docker container (docker restart <docker-container>) and list the keys again you'll receive an empty array:

{"data":[]}

7. Import and Delete Validator

We can see the same behaviour the other way around. To reproduce this we first need to sucessfully import a validator. The first thing is to change the permission of **all-accounts.keystore.json** to **600**, then import the key successful and then change the permission of **all-accounts.keystore.json** back to **700.** If we now try to delete the key like this:

```
curl http://<VC-Endpoint>:7500/eth/v1/keystores -X DELETE -H 'Content-
Type: application/json' -H 'Authorization: Bearer <TOKEN>' -d '{ "pubkeys":
["<pubkey>"] }'
```

We get something like this:

{"message":"Could not delete keys: could not write keystore file for accounts: could not write accounts: file already exists without proper 0600 permissions","code":500}

But if you list them again you'll see that the key was deleted:

{"data":[]}

Restart your VC docker container, list the keys and you'll find an array with a key in it:

{"data":[{"validating_pubkey":"<pubkey>","derivation_path":""}]}