Lock your swarm to protect its encryption key

Estimated reading time: 5 minutes

In Docker 1.13 and higher, the Raft logs used by swarm managers are encrypted on disk by default. This at-rest encryption protects your service's configuration and data from attackers who gain access to the encrypted Raft logs. One of the reasons this feature was introduced was in support of the new Docker secrets (https://docs.docker.com/engine/swarm/secrets/) feature.

When Docker restarts, both the TLS key used to encrypt communication among swarm nodes, and the key used to encrypt and decrypt Raft logs on disk, are loaded into each manager node's memory. Docker 1.13 introduces the ability to protect the mutual TLS encryption key and the key used to encrypt and decrypt Raft logs at rest, by allowing you to take ownership of these keys and to require manual unlocking of your managers. This feature is called *autolock*.

When Docker restarts, you must unlock the swarm (https://docs.docker.com/engine/swarm/swarm_manager_locking/#unlock-a-swarm) first, using a *key encryption key* generated by Docker when the swarm was locked. You can rotate this key encryption key at any time.

> **Note**: You don't need to unlock the swarm when a new node joins the swarm, because the key is propagated to it over mutual TLS.

Initialize a swarm with autolocking enabled

When you initialize a new swarm, you can use the `--autolock` flag to enable autolocking of swarm manager nodes when Docker restarts.

```
$ docker swarm init --autolock

Swarm initialized: current node (k1q27tfyx9rncpixhk69sa61v) is now
 a manager.

To add a worker to this swarm, run the following command:

    docker swarm join \
    --token SWMTKN-1-0j52ln6hxjpxk2wgk917abcnxywj3xed0y8vi1e5m9t3u
ttrtu-7bnxvvlz2mrcpfonjuztmtts9 \
    172.31.46.109:2377

To add a manager to this swarm, run 'docker swarm join-token manag
er' and follow the instructions.

To unlock a swarm manager after it restarts, run the `docker swarm
 unlock`
command and provide the following key:

    SWMKEY-1-WuYH/IX284+lRcXuoVf38viIDK3HJEKY13MIHX+tTt8
```

Store the key in a safe place, such as in a password manager.

When Docker restarts, you need to unlock the swarm (https://docs.docker.com/engine/swarm/swarm_manager_locking/#unlock-a-swarm). A locked swarm causes an error like the following when you try to start or restart a service:

```
$ sudo service docker restart

$ docker service ls

Error response from daemon: Swarm is encrypted and needs to be unl
ocked before it can be used. Use "docker swarm unlock" to unlock i
t.
```

Enable or disable autolock on an existing swarm

To enable autolock on an existing swarm, set the `autolock` flag to `true`.

```
$ docker swarm update --autolock=true

Swarm updated.
To unlock a swarm manager after it restarts, run the `docker swarm
 unlock`
command and provide the following key:

    SWMKEY-1-+MrE8NgAyKj5r3NcR4FiQMdgu+7W72urH0EZeSmP/0Y

Please remember to store this key in a password manager, since wit
hout it you
will not be able to restart the manager.
```

To disable autolock, set `--autolock` to `false`. The mutual TLS key and the encryption key used to read and write Raft logs are stored unencrypted on disk. There is a trade-off between the risk of storing the encryption key unencrypted at rest and the convenience of restarting a swarm without needing to unlock each manager.

```
$ docker swarm update --autolock=false
```

Keep the unlock key around for a short time after disabling autolocking, in case a manager goes down while it is still configured to lock using the old key.

Unlock a swarm

To unlock a locked swarm, use `docker swarm unlock`.

```
$ docker swarm unlock

Please enter unlock key:
```

Enter the encryption key that was generated and shown in the command output when you locked the swarm or rotated the key, and the swarm unlocks.

View the current unlock key for a running swarm

Consider a situation where your swarm is running as expected, then a manager node becomes unavailable. You troubleshoot the problem and bring the physical node back online, but you need to unlock the manager by providing the unlock key to read the encrypted credentials and Raft logs.

If the key has not been rotated since the node left the swarm, and you have a quorum of functional manager nodes in the swarm, you can view the current unlock key using `docker swarm unlock-key` without any arguments.

```
$ docker swarm unlock-key

To unlock a swarm manager after it restarts, run the `docker swarm
 unlock`
command and provide the following key:

    SWMKEY-1-8jDgbUNlJtUe5P/lcr9IXGVxqZpZUXPzd+qzcGp4ZYA

Please remember to store this key in a password manager, since wit
hout it you
will not be able to restart the manager.
```

If the key was rotated after the swarm node became unavailable and you do not have a record of the previous key, you may need to force the manager to leave the swarm and join it back to the swarm as a new manager.

Rotate the unlock key

You should rotate the locked swarm's unlock key on a regular schedule.

```
$ docker swarm unlock-key --rotate

Successfully rotated manager unlock key.

To unlock a swarm manager after it restarts, run the `docker swarm
 unlock`
command and provide the following key:

    SWMKEY-1-8jDgbUNlJtUe5P/lcr9IXGVxqZpZUXPzd+qzcGp4ZYA

Please remember to store this key in a password manager, since wit
hout it you
will not be able to restart the manager.
```

> **Warning**: When you rotate the unlock key, keep a record of the old key around for a few minutes, so that if a manager goes down before it gets the new key, it may still be unlocked with the old one.

swarm (https://docs.docker.com/glossary/?term=swarm), manager (https://docs.docker.com/glossary/?term=manager), lock (https://docs.docker.com/glossary/?term=lock), unlock (https://docs.docker.com/glossary/?term=unlock), autolock (https://docs.docker.com/glossary/?term=autolock), encryption (https://docs.docker.com/glossary/?term=encryption)