What is etcd?

etcd is the key-value database that stores all Kubernetes cluster state, every object, configuration, and Secret. By default Secrets are stored only base64-encoded, not encrypted, so anyone who can read etcd can read every credential in the cluster. etcd listens on port 2379, and if it is reachable without client-certificate authentication, it is a full cluster compromise. Protect it with mutual TLS, network isolation, and encryption of Secrets at rest.

HOW IT WORKS

01 The abuse and payload

If etcd (port 2379) is reachable without client-cert auth:

- Read every key, including Secrets: `etcdctl --endpoints=https://NODE:2379 get / --prefix --keys-only` then fetch Secret values.
- The returned Secret data is base64, trivially decoded into real credentials, tokens, and TLS keys.
- With those, authenticate to the API server or downstream systems and take over the cluster and its cloud account.

A backup of etcd left unprotected is the same exposure. Documented techniques shown for defenders.

> **ETCD IS THE WHOLE CLUSTER**
>
> *Reading etcd means reading every Secret in the cluster (they are only base64-encoded by default). Unauthenticated etcd on 2379, or an unprotected etcd backup, is total compromise.*

HOW TO DEFEND

- Require mutual TLS (client certificates) for all etcd access; never allow anonymous connections.
- Network-isolate etcd so only the API server (control plane) can reach 2379.
- Enable encryption at rest for Secrets so etcd does not store them in recoverable form.
- Protect etcd backups with the same controls and encryption as the live store.
- Audit and test that etcd is unreachable from workloads and the network.

SOURCES

[1] Kubernetes docs: Securing a cluster

[2] MITRE ATT&CK: Containers Matrix

[3] NIST SP 800-190 Application Container Security Guide

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-etcd

Open online