

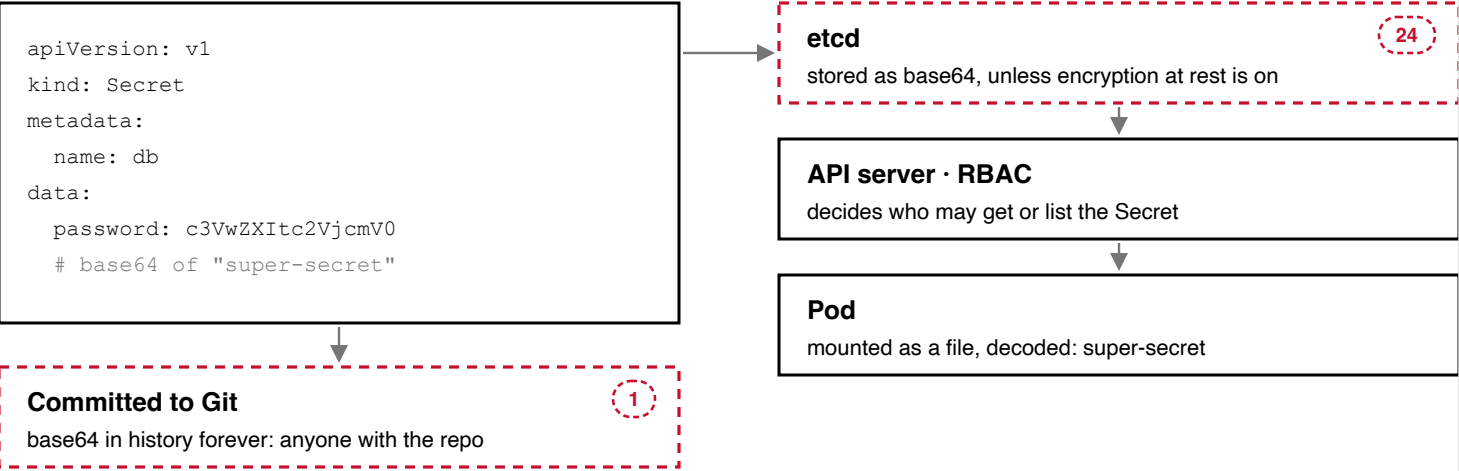
# KUBERNETES SECRET vs SVG BUNDLE — BASE64 IS A WRAPPER, NOT A LOCK

both carry encoded bytes inside a text format; only one of them can be locked

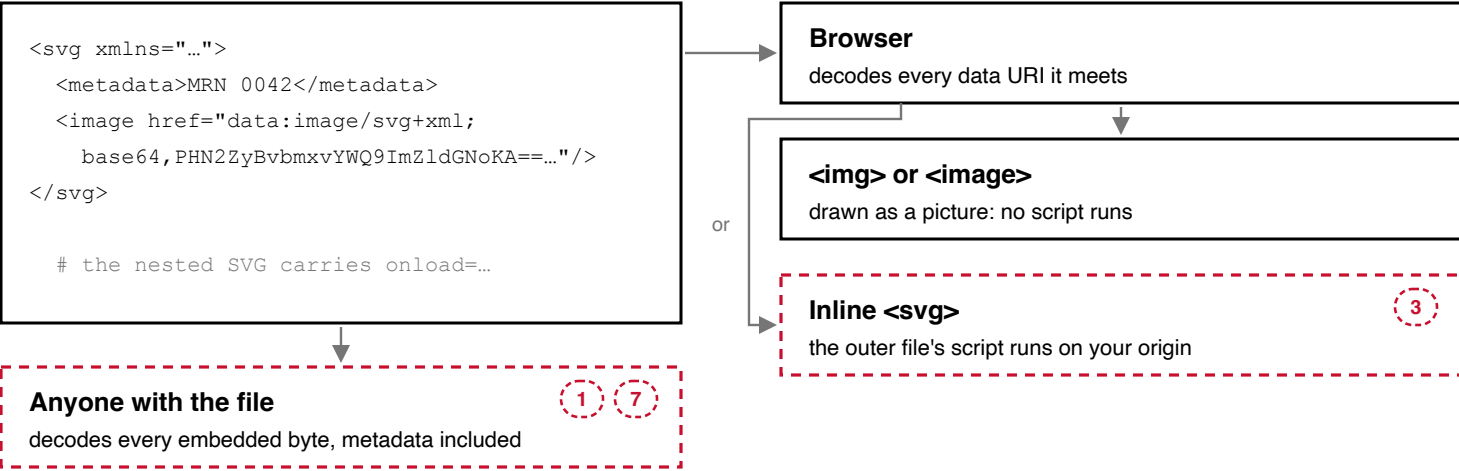
Red dashed = a gap. Numbers match the security gaps article.

## SAME WRAPPER, TWO PATHS

### Kubernetes Secret: configuration the cluster decodes



### SVG bundle: content the browser decodes



## WHERE THEY ARE ALIKE, AND WHERE THEY DIFFER

| CONCERN             | KUBERNETES SECRET                              | SVG BUNDLE                                 |
|---------------------|------------------------------------------------|--------------------------------------------|
| Text format         | YAML or JSON                                   | XML                                        |
| What's encoded      | Passwords, keys, certificates under data:      | Images, fonts or other SVGs in data: URIs  |
| Who decodes it      | The cluster, into a pod's files or environment | The browser, as it draws                   |
| Access control      | RBAC on get, list and watch                    | None once shipped: the file is the content |
| Encryption at rest  | Possible: etcd encryption with a KMS provider  | Not possible: the bytes are the content    |
| Can it execute      | No: it is configuration                        | Yes, when the outer SVG is inlined (gap 3) |
| A text scanner sees | Only the encoded value (gap 2)                 | Only the encoded payload (gap 2)           |

## THE SHARED LESSON

**Decode before you scan**  
Secret scanners must decode data: values;  
SVG sanitizers must open nested data URIs.

**Keep secrets out of both**  
A Secret YAML in Git and an SVG with an  
embedded payload leak the same way.

**Only one has a lock**  
Kubernetes: RBAC plus etcd encryption.  
An SVG: carry nothing private at all.

Base64 changes how bytes are written, not who can read them.