Protecting Sensitive Code with Encrypted Container Images on Kubernetes

Harshal Patil, Brandon Lum
IBM
Docker Hub hack exposed data of 190,000 users

Docker Hub usernames, hashed passwords, GitHub and Bitbucket access tokens exposed in the hack.

By Catalin Cimpanu for Zero Day | April 27, 2019 -- 09:11 GMT (02:11 PDT) | Topic: Security

Recommended Content:
White Papers: Illustrated Guide to Container Security
Download this Illustrated Guide to Container Security that provides visuals and easy-to-digest diagrams that answer questions about cloud security. ...
What does this mean for your images?

Image Signing with Notary & Portieris will ensure the integrity of your deployment images!

But...

Private Images’ sensitive content will be exposed!
Container Image Encryption

Build
- Build as normal
- **Encrypt**
- Push

Encrypt
- Encrypted image stored
- Cannot be read

Run
- Pull
- **Decrypt**
- Run
Benefits?

● Image Confidentiality & Deprivileged Registry

● Execution Boundary Control

“If my code is running, I know it’s in my cluster”

○ Encrypted Containers Images + Key management could provide guarantees about where an image can run.

○ i.e. Image X can only run in the EU nodes.
docker build --encrypt --keys User.pub ...
1. `docker build --encrypt --keys User.pub ...

2. Push to Registry
1. `docker build --encrypt --keys User.pub ...`

2. Push to Registry

Key Material

Registry

BUILD

CI / CD

Image

Enc. Image

K8s Master

K8s Worker

runtime

kubelet

DEPLOY
1. `docker build --encrypt --keys User.pub ...`

2. Push to Registry

3. Create secret

Key Material

K8s Master

K8s Worker

runtime

kubelet
1. `docker build --encrypt --keys User.pub ...`

2. Push to Registry

3. Create secret

4. Create encrypted pod.yaml
1 docker build --encrypt --keys User.pub ...

2 Push to Registry

3 Create secret

4 Create Pod with runtime and kubelet
1. `docker build --encrypt --keys User.pub ...`
2. Push to Registry
3. Create secret
4. Create Pod with `ImageDecryptSecret`
5. Pull & Decrypt
DEMO
Demo Link/Materials

KEP  https://github.com/kubernetes/enhancements/issues/1067

OCI SPEC Issue
https://github.com/opencontainers/image-spec/issues/747

OCI SPEC PR
https://github.com/opencontainers/image-spec/pull/775

POC and Demo
https://github.com/harche/kubernetes/commit/8a0ef9898a55110d7e41f8e1a846c66cfc2fa691

https://youtu.be/S3FK4y5McOk
Conventional Knowledge

Encryption $\rightarrow$ **Bad** deduplication

**Claim:**
With container images, this doesn’t have to be the case!
Encryption Primer

https://www.outsystems.com/blog/posts/how-to-teach-child-about-asymmetric-cryptography/
This is a secret message.

Fast, good for large data

Key needs to be securely shared somehow
Encryption Primer - Assym Enc.

This is a secret message.

Addresses key sharing: Each user has a Public-Private key pair, where Public Key is not secret, can be published.

Slow
Encrypt and distribution flow

```
docker build --encrypt --keys Alice.pub ...
```
Encrypt and distribution flow

```
docker build --encrypt --keys Alice.pub ...
```

Image + Generate Sym. Key = Enc. Image
Encrypt and distribution flow

docker build --encrypt --keys Alice.pub ...

For Alice
(Wrapped Key)

Wrap sym. key with public key of Alice to create a Wrapped Key.
Encrypt and distribution flow

```
docker build --encrypt --keys Alice.pub ...
```

For Alice

(Wrapped Key)

Wrap sym. key with public key of Alice to create a Wrapped Key.

Image Registry

docker push
Encrypt and distribution flow

For Alice (Wrapped Key)

Unwrap wrapped key with private key of Alice to get the sym. key.
Spec Details
OCI Proposal

Image Spec

```
{
  "schemaVersion": 2,
  "config": {
    "mediaType": "...image.config.v1+json",
    "size": 7023,
    "digest": "sha256:b5b2b2c507..."
  },
  "layers": [
    {
      "mediaType": "...tar+gzip",
      "size": 32654,
      "digest": "sha256:9834876d..."
    },
    {
      "mediaType": "...tar+gzip",
      "size": 16724,
      "digest": "sha256:3c3a4604..."
    },
    {
      "mediaType": "...tar+gzip",
      "size": 73109,
      "digest": "sha256:ec4b8955..."
    }
  ]
}
```
Container Encryption Features

- Encrypt on layers means images can still benefit from deduplication of non-sensitive layers
- Encrypt once for multiple recipients. Registry deduplication on large encrypted data blob.
Future Work

Encryption in OCI Image Spec
Issue: https://github.com/opencontainers/image-spec/issues/747
PR: https://github.com/opencontainers/image-spec/pull/775

Hardware Encryption & FIPS compliance
Hardware Encryption: With TPMs, HSMs, TEEs and Key Management Key Wrapping Services
Support Enterprise FIPS: Ability to plugin FIPS compliant components for enterprise use