Securing the Deploy Pipeline

Felix Glaser
Production Security Engineer

shopify
curl | sudo bash
VMs

configure

install

database
Mutability is the enemy.
Mutability is no more!
Containerized infrastructure
Still allows manual changes

manual kubectl create, run, edit
Runs containers outside your org

manual kubectl create, run, edit

pull
The new curl | sudo bash

FROM Ubuntu:14.04
COPY executable /usr/bin
CMD ["/usr/bin/executable"]
The new curl | sudo bash

FROM Ubuntu:14.04
COPY executable /usr/bin
CMD ["/usr/bin/executable"]

apt-get install unattended-upgrades
How do we fix this?
Gate which images can run
When to make the decision
When to make the decision
Pre-computed signatures

```javascript
PGP.sign({
    "critical": {
        "identity": {
            "docker-reference": "gcr.io/some/where"
        },
        "image": {
            "docker-manifest-digest": "sha256:462205...28c9fd945a"
        },
        "type": "Google cloud binauthz container signature"
    }
})
```
Admission controller

deployment controller ➔ pod api ➔ admission controller

pod api ➔ pod ➔ pod ➔ pod

ok? ➔ yes/no
Kritis

github.com/grafeas/kritis
Kritis gating deploys

Pull

Apply

Ok?

Rogue access
Grafeas

github.com/grafeas/grafeas
## Kind, note and occurrence

<table>
<thead>
<tr>
<th>Kind</th>
<th>Note Summary</th>
<th>Occurrence Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTESTATION</td>
<td>A logical attestation role or authority, used as an anchor for specific attestations</td>
<td>An attestation by an authority for a specific property and resource</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>Package descriptions</td>
<td>Filesystem locations detailing where the package is installed in a specific resource</td>
</tr>
<tr>
<td>VULNERABILITY</td>
<td>CVE or vulnerability description and details including severity, versions</td>
<td>Affected packages/versions in a specific resource</td>
</tr>
</tbody>
</table>
Who creates the attestations?
Voucher

github.com/shopify/voucher
Voucher runs checks

Correct pipeline?

In our registry?

Vulnerable?

Tested?

root?

Vuln Scanner
Which attestations are required?
Policies

admissionWhitelistPatterns:
- namePattern: nginx/image:sha256...

defaultAdmissionRule:
  enforcementMode: ENFORCED_BLOCK_AND_AUDIT_LOG
  evaluationMode: REQUIRE_ATTESTATION
  requireAttestationsBy:
    - projects/binauthz/attestors/name
name: projects/shopify-security/policy
admissionWhitelistPatterns:
  - namePattern: nginx/image:sha256...

clusterAdmissionRules:
  us-east1-a.cluster:
    evaluationMode: REQUIRE_ATTESTATION
    enforcementMode: ENFORCED_BLOCK_AND_AUDIT_LOG
    requireAttestationsBy:
      - projects/name/attestors/name

defaultAdmissionRule: ...
Package vulnerability policy

packageVulnerabilityPolicy:
  maximumSeverity: HIGH
  whitelistCVEs:
    providers/vulnz/notes/CVE-2017-1000082
    providers/vulnz/notes/CVE-2017-1000082
Policies per project/cluster
But what about emergencies?

That require changes right now!
apiVersion: v1
kind: ReplicationController
spec:
template:
  metadata:
    annotations:
      alpha.image-policy.k8s.io/break-glass: "true"
spec:
  containers:
  - name: binary-authorization
    image: gcr.io/somewhere/image@sha256:digest
 Break glass

apply with annotation: break-glass  
ok?  
no!  
attestations  

still deploy

P1  P2
If everyone can just add break-glass…

… what is it good for?!
Page @cloudsec
Rollout hints

```javascript
PGP.sign({
    "identity": {
        "docker-reference": "gcr.io/some/where"
    },
    "image": {
        "docker-manifest-digest": "sha256:462205...28c9fd945a"
    }
})
```
admissionWhitelistPatterns:
- namePattern: gcr.io/project/*
- namePattern: gcr.io/project/helloworld
- namePattern: gcr.io/project/helloworld:tag
- namePattern: gcr.io/project/helloworld:v1.*
- namePattern: gcr.io/project/helloworld@sha256:123...abc
Rollout hints

`kubectl plugin resolve-tags -f <file with tags> --apply true`
What have we achieved?

And what’s left to do!
Do you have any questions?

Felix Glaser
@klautcomputing
felix.glaser@shopify.com
Resources:

- https://github.com/Shopify/voucher
- https://github.com/grafeas/grafeas
- https://github.com/grafeas/kritis
- https://cloud.google.com/binary-authorization/docs/
- https://codelabs.developers.google.com/codelabs/cloud-binauthz-intro/index.html#0