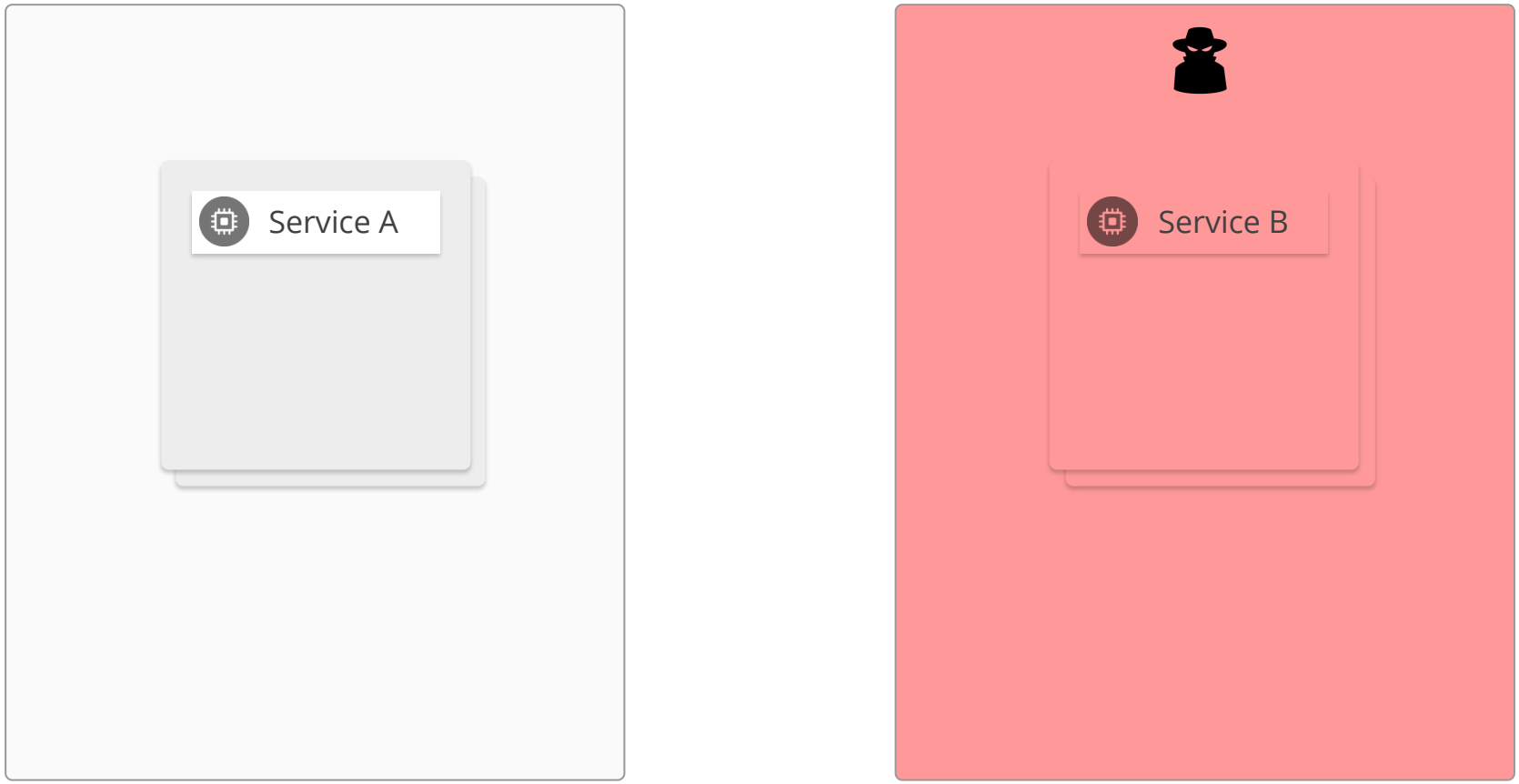
Addressing security in containerized applications with Istio

SecAppDev 2019

about:me

Erlend Oftedal

- @ Blank, Oslo, Norway
- Developer, security architect, security tester, bug bounty hunter
- Builds open source security tools like Retire.js
- Head of the OWASP Norway chapter
- @webtonull

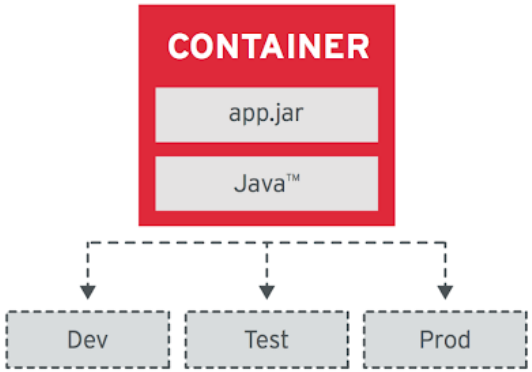

Service A
Service B

Building the image

*"Nearly **half of all attacks came through the application layer**, mimicking the sentiment of other annual data breach reports. For example the Verizon Data Breach Investigations Report also shows that **the Application Layer is the number one attack vector**."*

https://www.templarbit.com/blog/2019/01/10/we-analyzed-data-breaches-and-cyber-attacks-of-2018-here-are-the-key-insights/

Image Immutability Principle

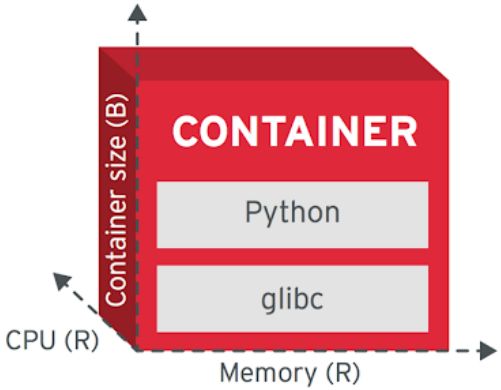


High Observability Principle

Process Disposability Principle

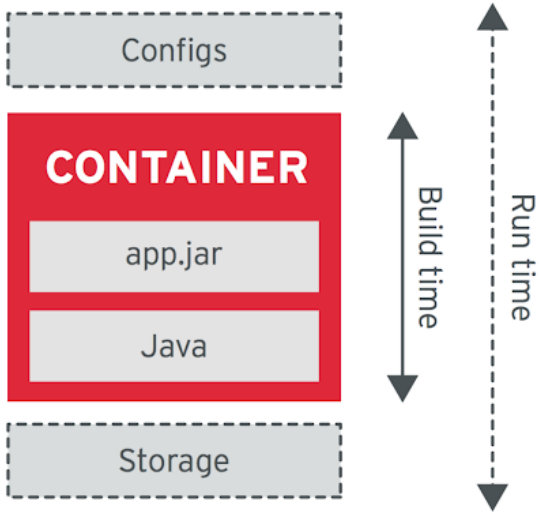
Lifecycle Conformance Principle

Runtime Confinement Principle



Single Concern Principle

Self-Containment Principle



https://kubernetes.io/blog/2018/03/principles-of-container-app-design/

8 Principles of Secure Development & Deployment

1. Secure development is everyone's concern
2. Keep your security knowledge sharp
3. Produce clean & maintainable code
4. Secure your development environment
5. Protect your code repository
6. Secure the build and deployment pipeline
7. Continually test your security
8. Plan for security flaws

The National Cyber Security Centre: https://github.com/ukncsc/secure-development-and-deployment

Securing the materials

- Source code repositories
- Base images
- Dependencies

Source code repositories

- Access to read code
- Access to commit code
  - Review requirements (pull requests)
- Access to source code servers

Base images

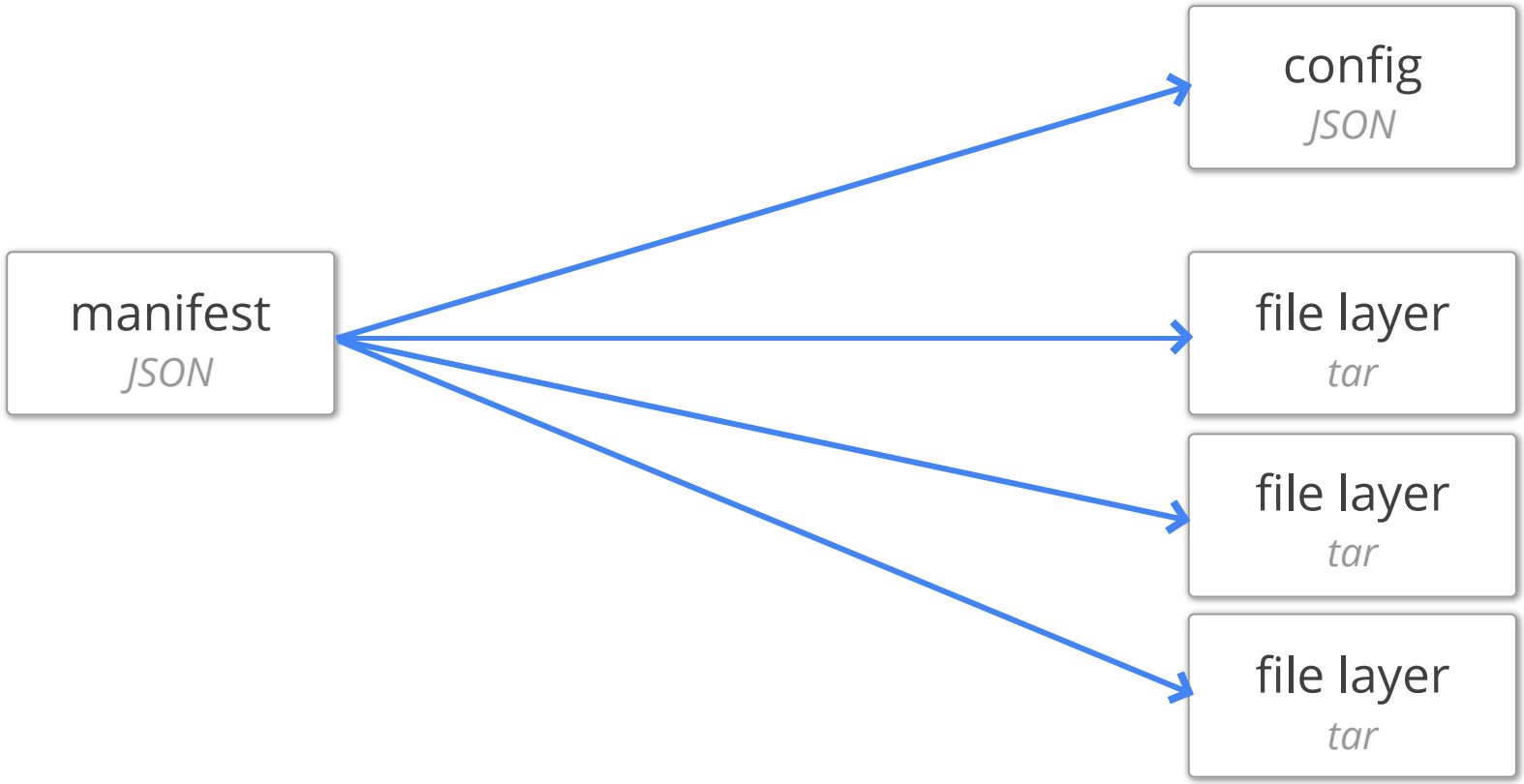
- Outdated/vulnerable images
- Doppelganger images
- Verified/signed images
- Keep in local/protected registry

Backdoored images

- Backdoors in MySQL and Tomcat images
- Included:
  - Python Reverse Shell
  - Bash Reverse Shell
  - Attacker's SSH key
  - embedded cryptocoin mining software
  - and more

https://neuvector.com/docker-security/backdoored-images-removed/

Image format in registry



manifest.json

https://registry-1.docker.io/v2/library/node/manifests/10


```
[
  {
    "Config": "358d031737783911 46fb55e9a77af6b7b5c5965c47033e127c2352aa7cfcc8fd.json",
    "RepoTags": [
      "node:10"
    ],
    "Layers": [
      "eeb4dbc820db170d1aa2221c74f33b150d29dddc2c0124b4ee509042ea27afd9/layer.tar",
      "d3fba4067805f00019a490b2d9aed6bef3551cc13afcf335340416029ac25461/layer.tar",
      "bfb2fc512f6ea96cb97cdf5e51a06c21be2c2abddf5e496f08a298d0f0035c4a/layer.tar",
      "764a1beace5dbda3ed2fd3bb46eaffdf081443671b8b77b45cdba152127a0b42/layer.tar",
      "b600a745aeb6bb2ff8b457b6007a350b8bd53314b4c165168df49bc498c9fc7f/layer.tar",
      "b3e79b146747d8f8b94a8a0df46c819bed572622f4c29280ba6b8ad972f3c417/layer.tar",
      "d4f2b8ed531aaf5e341e667767126f828651e3edb60e6c7adee65c3301e76d90/layer.tar",
      "45b3ddf7563396ca306d4f9bf78cff2c6b3d9301eb7718d86361ac523679e987/layer.tar"
    ]
  }
]
```


Config and layers are identified by SHA256 hashes

Inspecting an image

Using docker:

```
docker save image:tag -o filename.tar
```

Using a download-frozen-image :

```
./download-frozen-image-v2.sh some/folder image:tag
```

Using dive:

```
dive node:10-alpine
```

- https://github.com/moby/moby/blob/master/contrib/download-frozen-image-v2.sh
- https://github.com/wagoodman/dive

Depending on tags vs hashes

```
image:latest
image:sha256@e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

- Hashes are immutable, tags arent
- Depend on hashes externally
  - Pull into your own repository
- Optionally depend on tags internally

Dependencies

- OWASP Top 10 A9 - Using components with known vulnerabilities
- Outdated/vulnerable dependencies
- Doppelganger dependencies
- Verified/signed dependencies
- Keep in local/protected repository

Image storage

- Protect the registry!
- Optionally sign the images on the (protected) build server

Handling secrets

- Don't bake secrets into the image
  - Violates Image Immutability principle
  - Leaks to everyone with access to the image repository
- Env variables
  - Can be inspected from linked containers
- Using secrets/volumes/mounts
  - Ok, but no very well protected
- Secure key-value store
  - vault
  - keywhiz

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y curl
ENV pw=password123
WORKDIR /app
RUN curl -vvv -u user:${pw} https://repo/artifact -o /app/artifact
ENTRYPOINT export
```

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y curl
ENV pw=password123
WORKDIR /app
RUN curl -vvv -u user:${pw} https://repo/artifact -o /app/artifact
ENV pw=redacted
ENTRYPOINT export
```

Image config

{"architecture":"amd64","config":{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":fa
lse,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","pw=r
edacted"],"Cmd":null,"ArgsEscaped":true,"Image":"sha256:2991dad8dbba54090e90914b4108c8c032560b9f42492a8370a6599752812803","Volumes":
null,"WorkingDir":"/app","Entrypoint":["/bin/sh","-c","export"],"OnBuild":null,"Labels":null},"container":"b86ae27ccf54b348396487c06
a15137948a3f42111cdfc054f1e8389a4f7e008","container_config":{"Hostname":"b86ae27ccf54","Domainname":"","User":"","AttachStdin":fals
e,"AttachStdout":false,"AttachStderr":false,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["PATH=/usr/local/sbin:/usr/local/
bin:/usr/sbin:/usr/bin:/sbin:/bin","pw=redacted"],"Cmd":["/bin/sh","-c","#(nop) ","ENTRYPOINT [\"/bin/sh\" \"-c\" \"export\"]"],"Arg
sEscaped":true,"Image":"sha256:2991dad8dbba54090e90914b4108c8c032560b9f42492a8370a6599752812803","Volumes":null,"WorkingDir":"/ap
p","Entrypoint":["/bin/sh","-c","export"],"OnBuild":null,"Labels":{}},"created":"2019-01-13T20:23:41.458568799Z","docker_version":"1
8.06.0-ce","history":[{"created":"2017-12-14T20:59:45.307546564Z","created_by":"/bin/sh -c #(nop) ADD file:f5a2d04c3f3cafada15eb32e4
e8d971e48ef11724939c399a8664bf498111e67 in / "},{"created":"2017-12-14T20:59:46.089739785Z","created_by":"/bin/sh -c set -xe \t\t\u0
026\u0026 echo '#!/bin/sh' \u003e /usr/sbin/policy-rc.d \t\u0026\u0026 echo 'exit 101' \u003e\u003e /usr/sbin/policy-rc.d \t\u0026\u
0026 chmod +x /usr/sbin/policy-rc.d \t\t\u0026\u0026 dpkg-divert --local --rename --add /sbin/initctl \t\u0026\u0026 cp -a /usr/sbi
n/policy-rc.d /sbin/initctl \t\u0026\u0026 sed -i 's/^exit.*/exit 0/' /sbin/initctl \t\t\u0026\u0026 echo 'force-unsafe-io' \u003e /
etc/dpkg/dpkg.cfg.d/docker-apt-speedup \t\t\u0026\u0026 echo 'DPkg::Post-Invoke { \"rm -f /var/cache/apt/archives/*.deb /var/cache/a
pt/archives/partial/*.deb /var/cache/apt/*.bin || true\"; };' \u003e /etc/apt/apt.conf.d/docker-clean \t\u0026\u0026 echo 'APT::Upda
te::Post-Invoke { \"rm -f /var/cache/apt/archives/*.deb /var/cache/apt/archives/partial/*.deb /var/cache/apt/*.bin || true\"; };' \u
003e\u003e /etc/apt/apt.conf.d/docker-clean \t\u0026\u0026 echo 'Dir::Cache::pkgcache \"\"; Dir::Cache::srcpkgcache \"\";' \u003e\u0
03e /etc/apt/apt.conf.d/docker-clean \t\t\u0026\u0026 echo 'Acquire::Languages \"none\";' \u003e /etc/apt/apt.conf.d/docker-no-langu
ages \t\t\u0026\u0026 echo 'Acquire::GzipIndexes \"true\"; Acquire::CompressionTypes::Order:: \"gz\";' \u003e /etc/apt/apt.conf.d/do
cker-gzip-indexes \t\t\u0026\u0026 echo 'Apt::AutoRemove::SuggestsImportant \"false\";' \u003e /etc/apt/apt.conf.d/docker-autoremove
-suggests"},{"created":"2017-12-14T20:59:46.735863349Z","created_by":"/bin/sh -c rm -rf /var/lib/apt/lists/*"},{"created":"2017-12-1
4T20:59:47.411048575Z","created_by":"/bin/sh -c sed -i 's/^#\\s*\\(deb.*universe\\)$/\\1/g' /etc/apt/sources.list"},{"created":"2017
-12-14T20:59:48.062144391Z","created_by":"/bin/sh -c mkdir -p /run/systemd \u0026\u0026 echo 'docker' \u003e /run/systemd/containe
r"},{"created":"2017-12-14T20:59:48.244728969Z","created_by":"/bin/sh -c #(nop) CMD [\"/bin/bash\"]","empty_layer":true},{"create
d":"2019-01-13T20:21:27.917018919Z","created_by":"/bin/sh -c apt-get update \u0026\u0026 apt-get install -y curl"},{"created":"2019-
01-13T20:21:28.391135794Z","created_by":"/bin/sh -c #(nop) ENV pw=password123","empty_layer":true},{"created":"2019-01-13T20:21:28.7
07683671Z","created_by":"/bin/sh -c #(nop) WORKDIR /app"},{"created":"2019-01-13T20:23:40.929918392Z","created_by":"/bin/sh -c curl
-vvv -u user:${pw} https://erlend.oftedal.no/artifact?${pw} -o /app/artifact"},{"created":"2019-01-13T20:23:41.201051927Z","created_
by":"/bin/sh -c #(nop) ENV pw=redacted","empty_layer":true},{"created":"2019-01-13T20:23:41.458568799Z","created_by":"/bin/sh -c #(n
op) ENTRYPOINT [\"/bin/sh\" \"-c\" \"export\"]","empty_layer":true}],"os":"linux","rootfs":{"type":"layers","diff_ids":["sha256:48e0
baf45d4dfd028eab0a7d444309b436cac01ae879805ac0cfdf5aaf1ff93a","sha256:d2f8c05d353b2d55e191eccba65b72d72fd230c6fbef9cac97d5be5499bcd9
39","sha256:5a876f8f1a3d04b1f9735e38e5111e6f11cbf6cad2a44dc3ca6d394dd93caa5f","sha256:6458f770d435ace5bfba5b99460059c08e7dba66e8606e
70ecf941dc1f705077","sha256:f17fc24fb8d0efef646c75c2977fd41e6816922322944fa3683eeac3aaeec80f","sha256:1e6a12334a4424b7916c8090e65b4a
f6f7210067aee198aef2cf3432a82cd4eb","sha256:124ff862c2b30650acf42166e82ed61db98cc82cfb9ebe56e1daa0b56d4b4e5a","sha256:9bc152489c1702
9a690bfb096a0145cb3d458d332798d7dc25c942b05e749f7b"]}}

Image config


```
{"architecture":"amd64","config":{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":fa
lse,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","pw=r
edacted"],"Cmd":null,"ArgsEscaped":true,"Image":"sha256:2991dad8dbba54090e90914b4108c8c032560b9f42492a8370a6599752812803","Volumes":
null,"WorkingDir":"/app","Entrypoint":["/bin/sh","-c","export"],"OnBuild":null,"Labels":null},"container":"b86ae27ccf54b348396487c06
a15137948a3f42111cdfc054f1e8389a4f7e008","container_config":{"Hostname":"b86ae27ccf54","Domainname":"","User":"","AttachStdin":fals
e,"AttachStdout":false,"AttachStderr":false,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["PATH=/usr/local/sbin:/usr/local/
bin:/usr/sbin:/usr/bin:/sbin:/bin","pw=redacted"],"Cmd":["/bin/sh","-c","#(nop) ","ENTRYPOINT [\"/bin/sh\" \"-c\" \"export\"]"],"Arg
sEscaped":true,"Image":"sha256:2991dad8dbba54090e90914b4108c8c032560b9f42492a8370a6599752812803","Volumes":null,"WorkingDir":"/ap
p","Entrypoint":["/bin/sh","-c","export"],"OnBuild":null,"Labels":{}},"created":"2019-01-13T20:23:41.458568799Z","docker_version":"1
8.06.0-ce","history":[{"created":"2017-12-14T20:59:45.307546564Z","created_by":"/bin/sh -c #(nop) ADD file:f5a2d04c3f3cafada15eb32e4
e8d971e48ef11724939c399a8664bf498111e67 in / "},{"created":"2017-12-14T20:59:46.089739785Z","created_by":"/bin/sh -c set -xe \t\t\u0
026\u0026 echo '#!/bin/sh' \u003e /usr/sbin/policy-rc.d \t\u0026\u0026 echo 'exit 101' \u003e\u003e /usr/sbin/policy-rc.d \t\u0026\u
0026 chmod +x /usr/sbin/policy-rc.d \t\t\u0026\u0026 dpkg-divert --local --rename --add /sbin/initctl \t\u0026\u0026 cp -a /usr/sbi
n/policy-rc.d /sbin/initctl \t\u0026\u0026 sed -i 's/^exit.*/exit 0/' /sbin/initctl \t\t\u0026\u0026 echo 'force-unsafe-io' \u003e /
etc/dpkg/dpkg.cfg.d/docker-apt-speedup \t\t\u0026\u0026 echo 'DPkg::Post-Invoke { \"rm -f /var/cache/apt/archives/*.deb /var/cache/a
pt/archives/partial/*.deb /var/cache/apt/*.bin || true\"; };' \u003e /etc/apt/apt.conf.d/docker-clean \t\u0026\u0026 echo 'APT::Upda
te::Post-Invoke { \"rm -f /var/cache/apt/archives/*.deb /var/cache/apt/archives/partial/*.deb /var/cache/apt/*.bin || true\"; };' \u
003e\u003e /etc/apt/apt.conf.d/docker-clean \t\u0026\u0026 echo 'Dir::Cache::pkgcache \"\"; Dir::Cache::srcpkgcache \"\";' \u003e\u0
03e /etc/apt/apt.conf.d/docker-clean \t\t\u0026\u0026 echo 'Acquire::Languages \"none\";' \u003e /etc/apt/apt.conf.d/docker-no-langu
ages \t\t\u0026\u0026 echo 'Acquire::GzipIndexes \"true\"; Acquire::CompressionTypes::Order:: \"gz\";' \u003e /etc/apt/apt.conf.d/do
cker-gzip-indexes \t\t\u0026\u0026 echo 'Apt::AutoRemove::SuggestsImportant \"false\";' \u003e /etc/apt/apt.conf.d/docker-autoremove
-suggests"},{"created":"2017-12-14T20:59:46.735863349Z","created_by":"/bin/sh -c rm -rf /var/lib/apt/lists/*"},{"created":"2017-12-1
4T20:59:47.411048575Z","created_by":"/bin/sh -c sed -i 's/^#\\s*\\(deb.*universe\\)$/\\1/g' /etc/apt/sources.list"},{"created":"2017
-12-14T20:59:48.062144391Z","created_by":"/bin/sh -c mkdir -p /run/systemd \u0026\u0026 echo 'docker' \u003e /run/systemd/containe
r"},{"created":"2017-12-14T20:59:48.244728969Z","created_by":"/bin/sh -c #(nop) CMD [\"/bin/bash\"]","empty_layer":true},{"create
d":"2019-01-13T20:21:27.917018919Z","created_by":"/bin/sh -c apt-get update \u0026\u0026 apt-get install -y curl"},{"created":"2019-
01-13T20:21:28.391135794Z","created_by":"/bin/sh -c #(nop) ENV pw=password123","empty_layer":true},{"created":"2019-01-13T20:21:28.7
07683671Z","created_by":"/bin/sh -c #(nop) WORKDIR /app"},{"created":"2019-01-13T20:23:40.929918392Z","created_by":"/bin/sh -c curl
-vvv -u user:${pw} https://erlend.oftedal.no/artifact?${pw} -o /app/artifact"},{"created":"2019-01-13T20:23:41.201051927Z","created_
by":"/bin/sh -c #(nop) ENV pw=redacted","empty_layer":true},{"created":"2019-01-13T20:23:41.458568799Z","created_by":"/bin/sh -c #(n
op) ENTRYPOINT [\"/bin/sh\" \"-c\" \"export\"]","empty_layer":true}],"os":"linux","rootfs":{"type":"layers","diff_ids":["sha256:48e0
baf45d4dfd028eab0a7d444309b436cac01ae879805ac0cfdf5aaf1ff93a","sha256:d2f8c05d353b2d55e191eccba65b72d72fd230c6fbef9cac97d5be5499bcd9
39","sha256:5a876f8f1a3d04b1f9735e38e5111e6f11cbf6cad2a44dc3ca6d394dd93caa5f","sha256:6458f770d435ace5bfba5b99460059c08e7dba66e8606e
70ecf941dc1f705077","sha256:f17fc24fb8d0efef646c75c2977fd41e6816922322944fa3683eeac3aaeec80f","sha256:1e6a12334a4424b7916c8090e65b4a
f6f7210067aee198aef2cf3432a82cd4eb","sha256:124ff862c2b30650acf42166e82ed61db98cc82cfb9ebe56e1daa0b56d4b4e5a","sha256:9bc152489c1702
9a690bfb096a0145cb3d458d332798d7dc25c942b05e749f7b"]}}
```

Docker build secrets (experimental)

`Dockerfile`:

```
# syntax = docker/dockerfile:experimental
FROM ...
...
RUN --mount=type=bind,src=/some/file,target=/tmp/env eval (cat /tmp/env) && <some command>
...
```

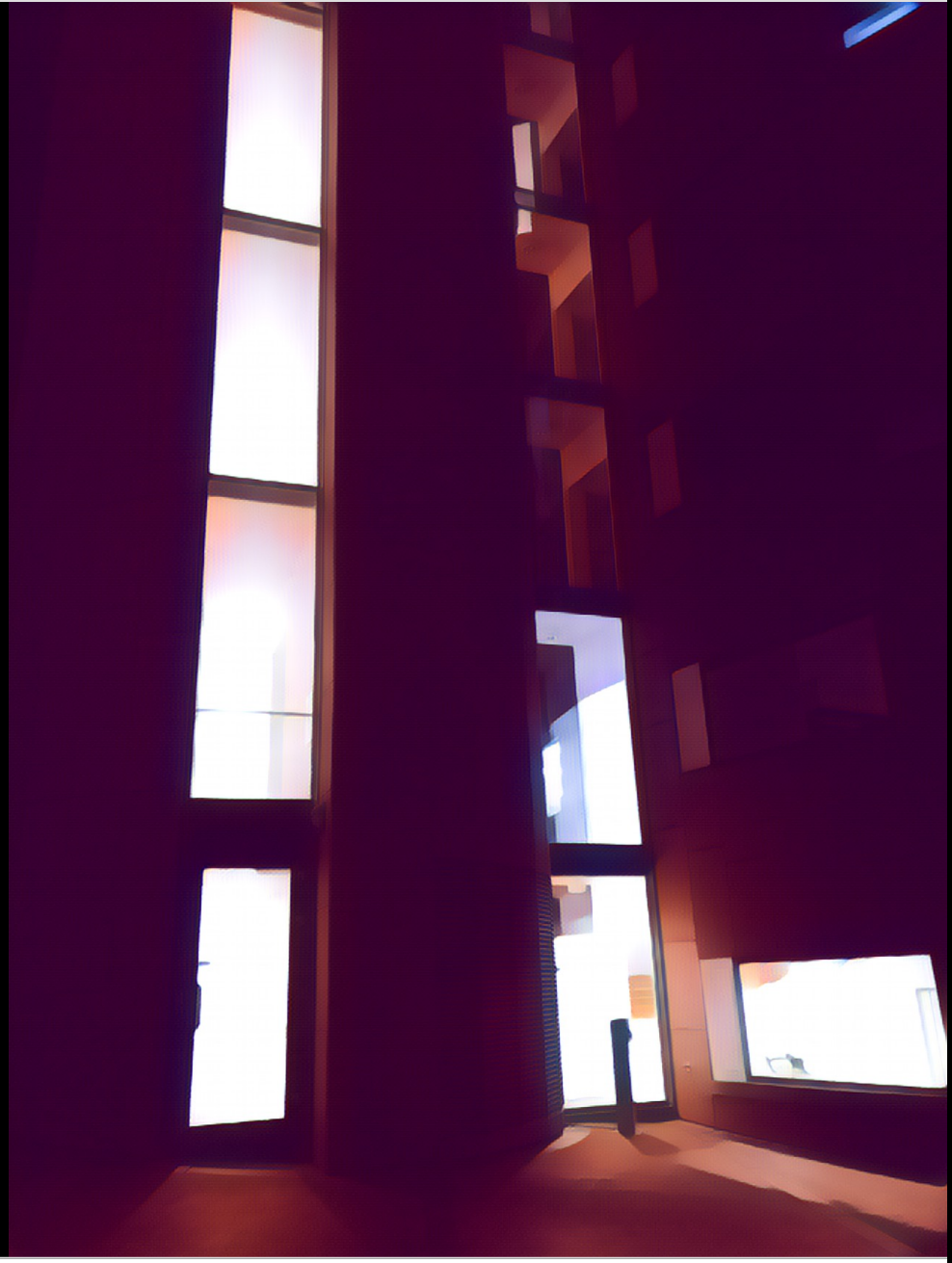
`/some/file`:

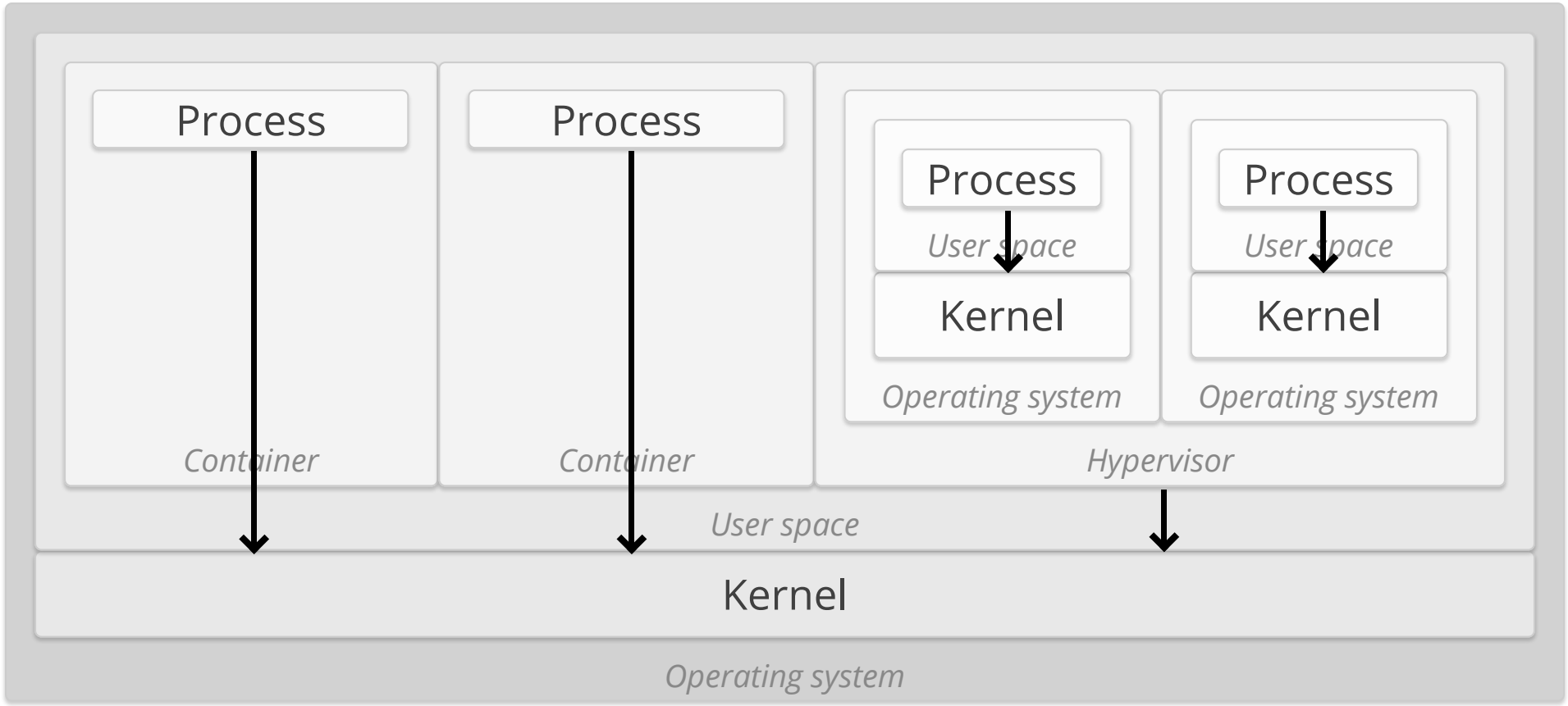
```
SECRET=password123
```

Command:

```
$ DOCKER_BUILDKIT=1 docker build -t demo:latest .
```

Runtime




Process
Process
Process
Process
User space
Process
User space
Kernel
Kernel
Container
Container
Operating system
Operating system
Hypervisor
User space
Kernel
Operating system

Container security

- Namespaces
- Control groups
- Capabilities
- Linux Security modules (LSMs)
- Seccomp

Namespaces

Namespace	Constant	Isolates
Cgroup	CLONE_NEWCGROUP	Cgroup root directory
IPC	CLONE_NEWIPC	System V IPC, POSIX message queues
Network	CLONE_NEWNET	Network devices, stacks, ports, etc.
Mount	CLONE_NEWNS	Mount points
PID	CLONE_NEWPID	Process IDs
User	CLONE_NEWUSER	User and group IDs
UTS	CLONE_NEWUTS	Hostname and NIS domain name

Container escapes - examples

- CVE-2016-5195 - Dirty Cow
- CVE-2017-1000405
- CVE-2017-1002101
- CVE-2019-5736

Don't run as `root`

- No `root` user
- No `privileged` containers
- Kuberenetes - PodSecurityPolicy
  - Limit what can be deployed

RCE

- What is your threat model?
- Image attack surface
- Drop to shell
- Skilled attacker does not need shell
  - Full force of the application runtime

Running with read-only file-system

- `docker run --read-only <some-image>`
- Steps to read-only:
  1. Run in normal (read-write) mode
  2. After some time: `docker diff <container-name>`
- Example:

```
docker run --read-only --tmpfs /tmp --tmpfs /run <some-image>
```

Docker scanning

- Haskell Dockerfile linter
- Lynis
- Docker Bench
- CoreOS Clair
- Banyanops Collector
- Anchore
- TwistLock

Application level DOS

- Network
- CPU
- Disk
- Pids
- Files
- Others like `/dev/random`

Built-in mitigations

- Memory limits
- CPU shares
- Disk
  - Separate partition(s)
  - Quotas: `--storage-opt size=5G`
- Pids - pidlimit
- Files - ulimit etc.

Mitigations

- Do not run as root
- Mount drives as ro if possible
- Least privilege (seccomp, capabilities, LSMs)
- Use minimal containers with only what is really needed to run
- Update containers
- Content trust

Locking down the platform

Auditing your docker hosts

- CIS Docker Community Edition Benchmark
  - https://www.cisecurity.org/cis-benchmarks/
  - 105 checkpoints

docker/docker-bench-security

```
# ------------------------------------------------------------------------
# Docker Bench for Security v1.3.4
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers
# Inspired by the CIS Docker Community Edition Benchmark v1.1.0.
# ------------------------------------------------------------------------

Initializing Fri Jan 18 19:39:53 UTC 2019


[INFO] 1 - Host Configuration
[WARN] 1.1  - Ensure a separate partition for containers has been created
[NOTE] 1.2  - Ensure the container host has been Hardened
[INFO] 1.3  - Ensure Docker is up to date
[INFO]       * Using 18.09.1, verify is it up to date as deemed necessary
[INFO]       * Your operating system vendor may provide support and security mai
```

Remapping the user namespace

Enable in kernel, if CentOS/Redhat:

```
grubby --args="namespace.unpriv_enable=1" --update-kernel=/boot/vmlinuz-$(uname -r)
```

Enable for docker

```
adduser --system --no-create-home --group dockremap
echo "dockremap:30000:65536"  >> /etc/subuid
echo "dockremap:30000:65536"  >> /etc/subgid
echo '{
  "userns-remap" : "dockremap"
}' > /etc/docker/daemon.json
systemctl restart docker
```

Warning! Can be problematic in some cases

Locking down your Kubernetes cluster

- Enabling Role Based Access Controls
- Protecting `etcd`
- Protecting the API (disabling `insecure port` etc.)
- PodSecurityPolicy
- Admission controllers
- NetworkPolicy
- etc.

Istio

Connect, secure, control, and observe services.

Istio

- Recently hit the version 1.0 milestone
  - Current version 1.0.6
- https://github.com/istio
- Built and maintained by Google, Lyft, Redhat, IBM, VMware and Cisco ++

Service mesh

"The network of microservices and the interactions between them"

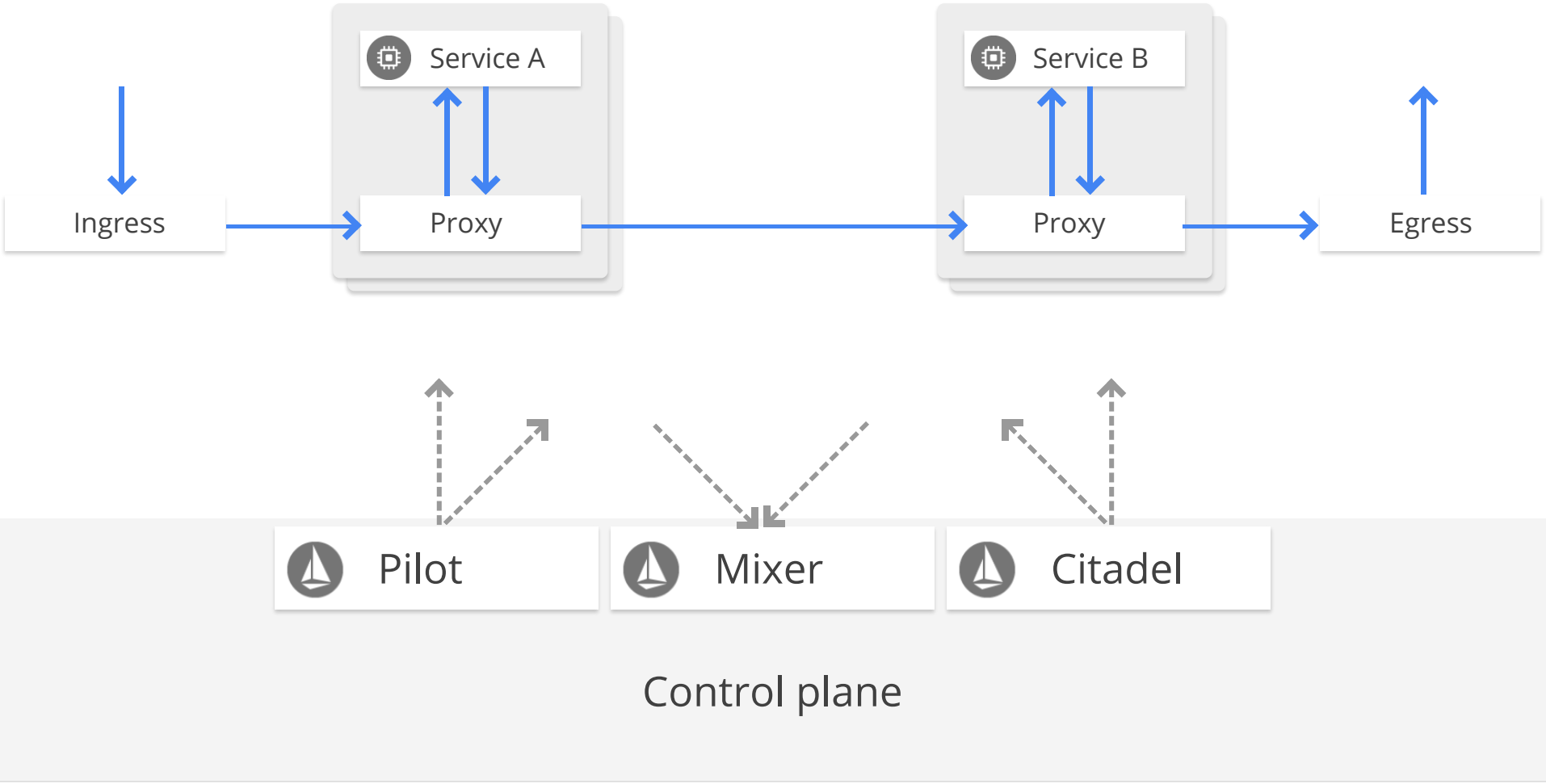
Typically provides

- Discovery
- Load balancing
- Failure recovery
- Metrics
- Monitoring

May also also include

- A/B testing and canary releases
- Rate limiting
- Access control
- End-to-end authentication

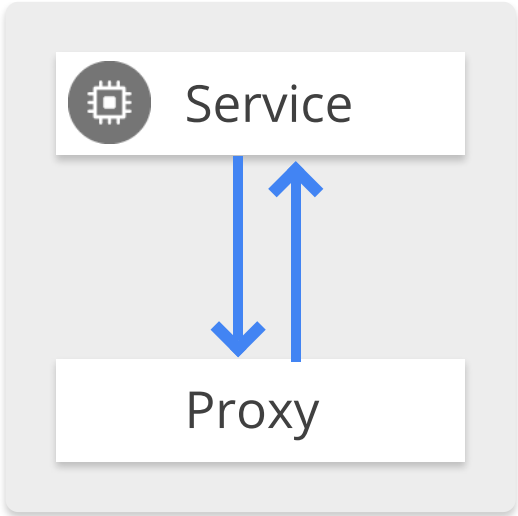
Istio Security Goals


Service A
Service B
Ingress
Proxy
Proxy
Egress
Pilot
Mixer
Citadel
Control plane

Back to the proxy

- The applications are unaware of the proxy...
- ...so how does it "hijack" the traffic?

- Containers in a pod share a network namespace
- `docker run --net=container:<container-name> ...`



istioctl kube-inject apply -f deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
...
spec:
  template:
    ...
    spec:
      containers:
      ...
      - args:
        ...
        - --controlPlaneAuthPolicy
        - MUTUAL_TLS
        env:
        ...
        - name: ISTIO_META_INTERCEPTION_MODE
          value: REDIRECT
        image: gcr.io/istio-release/proxyv2:1.0.1
        imagePullPolicy: IfNotPresent
        name: istio-proxy
        resources:
          requests:
            cpu: 10m
        securityContext:
          readOnlyRootFilesystem: true
          runAsUser: 1337
        volumeMounts:
        - mountPath: /etc/istio/proxy
          name: istio-envoy
        - mountPath: /etc/certs/
          name: istio-certs
          readOnly: true
...
```

istioctl kube-inject apply -f deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
...
spec:
   template:
      ...
      spec:
         containers:
         ...
         initContainers:
         - args:
           - -p
           - "15001"
           - -u
           - "1337"
           - -m
           - REDIRECT
           - -i
           - '*'
           - -x
           - ""
           - -b
           - 8000,
           - -d
           - ""
           image: gcr.io/istio-release/proxy_init:1.0.1
           imagePullPolicy: IfNotPresent
           name: istio-init
           resources: {}
           securityContext:
             capabilities:
               add:
               - NET_ADMIN
             privileged: true
...
```

istio-1.0.1/tools/deb/istio-iptables.sh

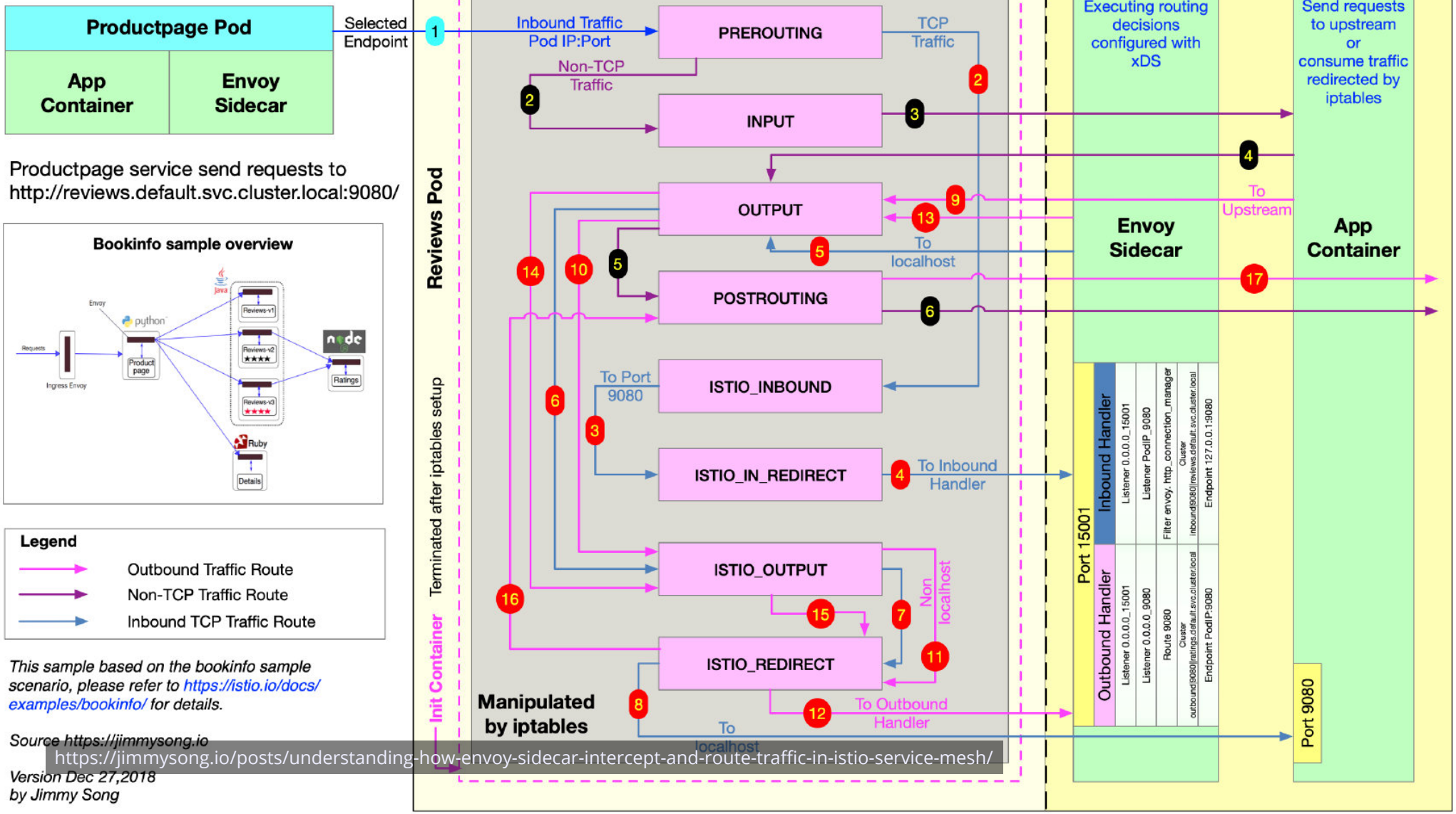
```
...
# Create a new chain for redirecting outbound traffic to the common Envoy port.
# In both chains, '-j RETURN' bypasses Envoy and '-j ISTIO_REDIRECT'
# redirects to Envoy.
iptables -t nat -N ISTIO_REDIRECT
iptables -t nat -A ISTIO_REDIRECT -p tcp -j REDIRECT --to-port ${PROXY_PORT}

# Use this chain also for redirecting inbound traffic to the common Envoy port
# when not using TPROXY.
iptables -t nat -N ISTIO_IN_REDIRECT
iptables -t nat -A ISTIO_IN_REDIRECT -p tcp -j REDIRECT --to-port ${INBOUND_CAPTURE_PORT}

# Handling of inbound ports. Traffic will be redirected to Envoy, which will process and forward
# to the local service. If not set, no inbound port will be intercepted by istio iptables.
if [ -n "${INBOUND_PORTS_INCLUDE}" ]; then
  if [ "${INBOUND_INTERCEPTION_MODE}" = "TPROXY" ] ; then
    # When using TPROXY, create a new chain for routing all inbound traffic to
    # Envoy. Any packet entering this chain gets marked with the ${INBOUND_TPROXY_MARK} mark,
    # so that they get routed to the loopback interface in order to get redirected to Envoy.
    # In the ISTIO_INBOUND chain, '-j ISTIO_DIVERT' reroutes to the loopback
    # interface.
    # Mark all inbound packets.
    iptables -t mangle -N ISTIO_DIVERT
    iptables -t mangle -A ISTIO_DIVERT -j MARK --set-mark ${INBOUND_TPROXY_MARK}
    iptables -t mangle -A ISTIO_DIVERT -j ACCEPT

    # Route all packets marked in chain ISTIO_DIVERT using routing table ${INBOUND_TPROXY_ROUTE_TABLE}.
    ip -f inet rule add fwmark ${INBOUND_TPROXY_MARK} lookup ${INBOUND_TPROXY_ROUTE_TABLE}
    # In routing table ${INBOUND_TPROXY_ROUTE_TABLE}, create a single default rule to route all traffic to
    # the loopback interface.
    ip -f inet route add local default dev lo table ${INBOUND_TPROXY_ROUTE_TABLE} || ip route show table all
...
```

How Envoy Proxy Working As Sidecar Proxies to Intercept and Route Traffic in Istio Service Mesh



Productpage service send requests to http://reviews.default.svc.cluster.local:9080/

Legend

- Outbound Traffic Route
- Non-TCP Traffic Route
- Inbound TCP Traffic Route

This sample based on the bookinfo sample scenario, please refer to https://istio.io/docs/examples/bookinfo/ for details.

Source https://jimmysong.io

Version Dec 27,2018
by Jimmy Song

https://jimmysong.io/posts/understanding-how-envoy-sidecar-intercept-and-route-traffic-in-istio-service-mesh/

```c
#include <sys/socket.h>

int getsockopt(int sockfd, int level, int optname, void *optval, socklen_t *optlen);
```

```
...
getsockopt(fd, SOL_IP, SO_ORIGINAL_DST, ref, dstlen);
...
```

Side note on cap NET_ADMIN

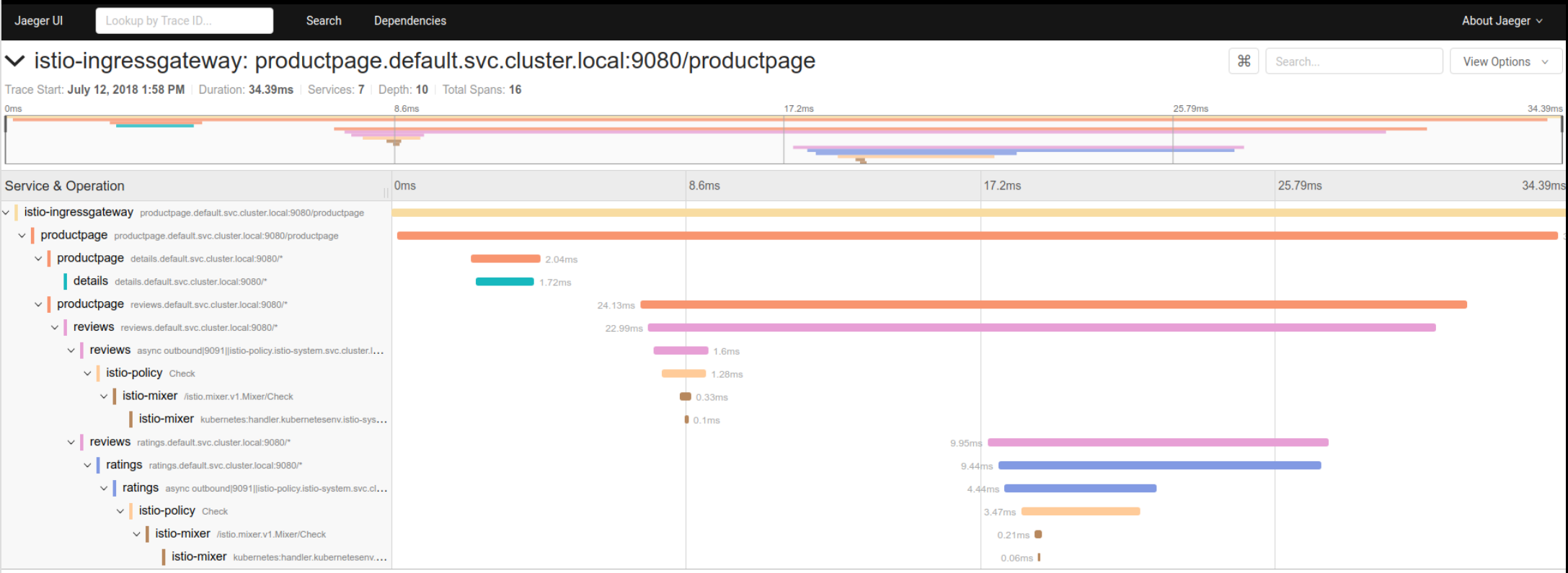
- Good paper: Understanding and Hardening Linux Containers - NCC Group
- Provides larger attack surface
- Lots of CVEs tied this capability

PodSecurityPolicy vs Istio

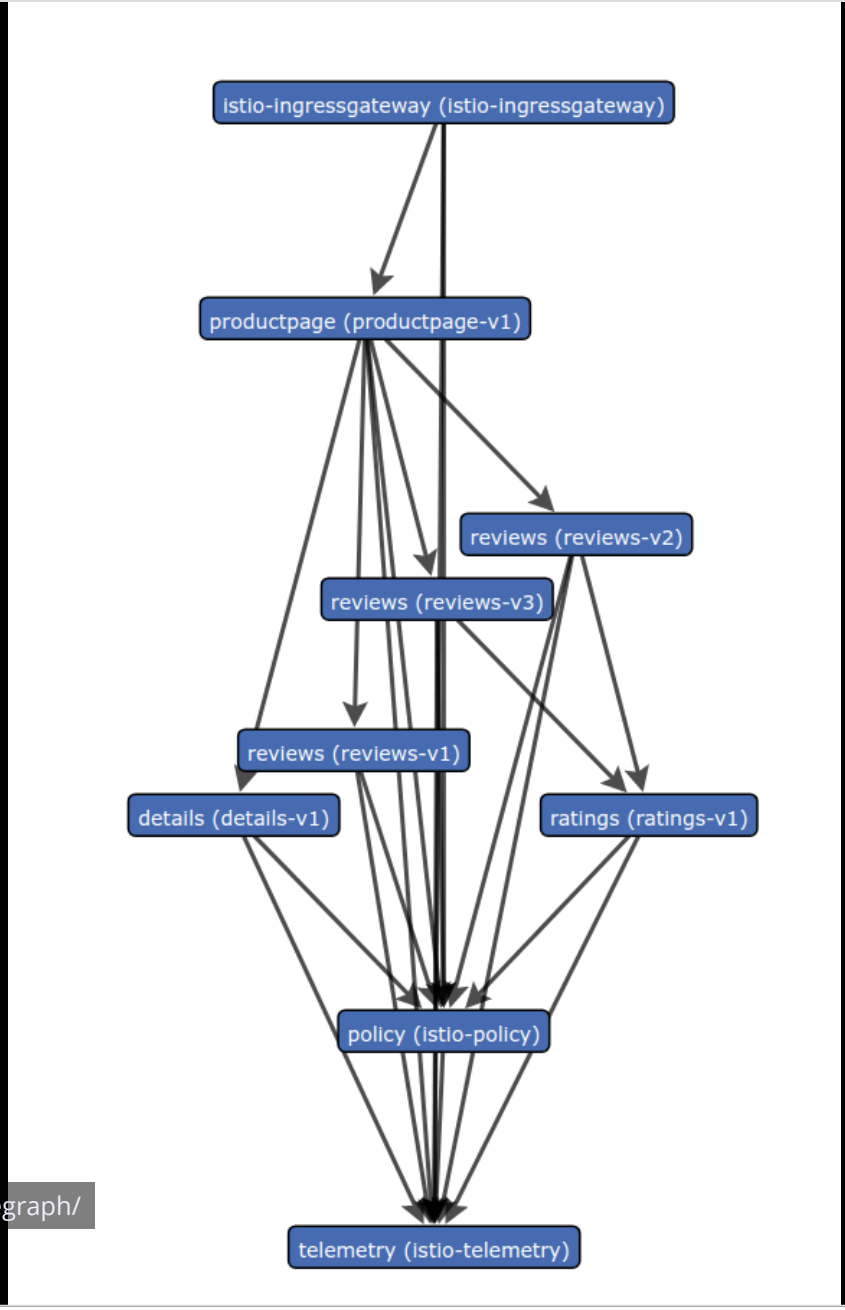
- What if the PodSecurityPolicy blocks privileged containers or `NET_ADMIN`?
- Will hopefully be fixed by a CNI plugin in a future release

Tracing

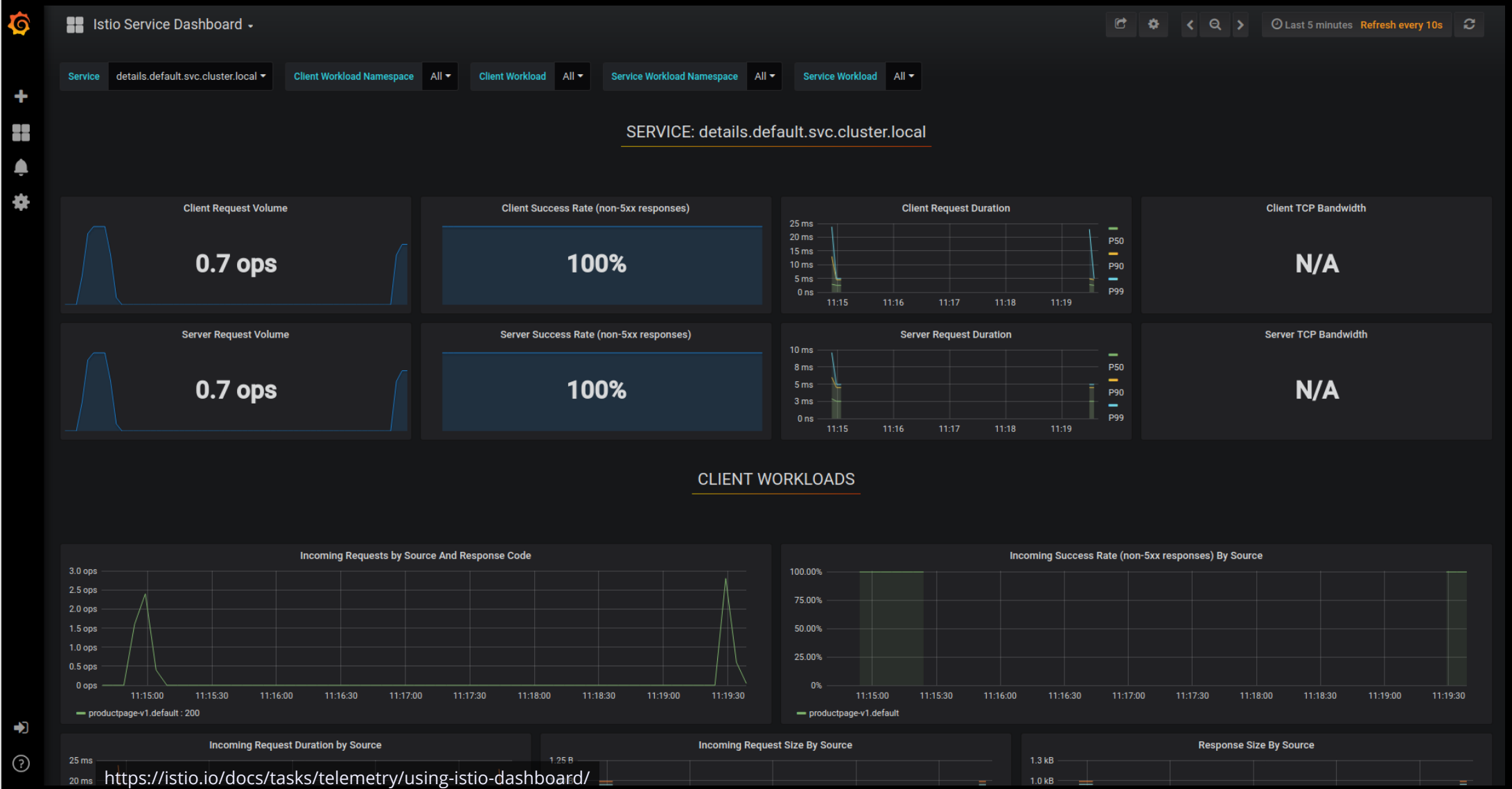
- ServiceGraph and Jaeger
- Application needs to forward:
  - `x-request-id`
  - `x-b3-traceid`
  - `x-b3-spanid`
  - `x-b3-parentspanid`
  - `x-b3-sampled`
  - `x-b3-flags`
  - `x-ot-span-context`



https://istio.io/docs/tasks/telemetry/distributed-tracing/



https://istio.io/docs/tasks/telemetry/servicegraph/



https://istio.io/docs/tasks/telemetry/using-istio-dashboard/

Istio Egress filtering

- Can stop Layer 7 connections
  - reverse shells
  - SSRF
  - etc.
- Will not stop more advanced exfil like DNS
- Calico + Network Policy
  - Filter on layer3/4

Routing and policies

- Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic
- Fine-grained control
  - Routing rules
  - Retries
  - Failovers
- Policies
  - Quotas
  - Rate limits
  - Access control
- Secure service-to-service communication through mutual TLS
  - Strong identity-based authentication and authorization

Rate limiting

```
apiVersion: config.istio.io/v1alpha2
kind: memquota
metadata:
  name: handler
  namespace: istio-system
spec:
  quotas:
  - name: requestcount.quota.istio-system
    maxAmount: 500      # Defaults
    validDuration: 1s   #
    overrides:
    - dimensions:
        destination: reviews
      maxAmount: 1
      validDuration: 5s
```

https://istio.io/docs/tasks/policy-enforcement/rate-limiting/">https://istio.io/docs/tasks/policy-enforcement/rate-limiting/

SPIFFE

- **S**ecure **P**roduction **I**dentity **F**ramework **F**or **E**veryone
- https://spiffe.io/
- Inspired by Google's ALTS - Application Layer Transport Security
- Workload/service identification vs. host identification

SPIFFE ID

spiffe://**example.com**/*some/service*

issuer *identifier*

SVID

- **S**PIFFE **V**erifiable **I**dentity **D**ocument
  - A SPIFFE ID
  - A public key
  - A valid signature
- X.509 Certificate
- Short-lived

SPIRE - **SPI**FFE **R**untime **E**nvironment

Istio Auth

- SPIFFE implementation
- Relies on Kubernetes for attestation
- More comprehensive than SPIRE
  - authentication
  - authorization
  - auditing

Istio Citadel

- Formerly known as Istio CA
- Runs as a k8s pod
- CA certificate
  - Self-signed or plugin

Istio Citadel rights

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: istio-citadel-istio-system
rules:
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["create", "get", "watch", "list", "update", "delete"]
- apiGroups: [""]
  resources: ["serviceaccounts"]
  verbs: ["get", "watch", "list"]
- apiGroups: [""]
  resources: ["services"]
  verbs: ["get", "watch", "list"]
```

Istio SVIDs

- Kubernetes secrets created by the Istio Citadel
- Per service account
- Valid for 1 hour (default)
- Refreshed every 30 minutes (default)
- Only contain the SPIFFE ID
  - No DNS names by default

Istio 0.5.1

3 security/pkg/pki/ca/controller/secret.go View

```
@@ -280,7 +280,8 @@ func (sc *SecretController) scrtUpdated(oldObj, newObj interface{}) {
280 280         certLifeTimeLeft := time.Until(cert.NotAfter)
281 281         certLifeTime := cert.NotAfter.Sub(cert.NotBefore)
282 282         // TODO(myidpt): we may introduce a minimum gracePeriod, without making the config too complex.
283     -       gracePeriod := time.Duration(sc.gracePeriodRatio) * certLifeTime
    283 +       // Because time.Duration only takes int type, multiply gracePeriodRatio by 1000 and then divide it.
    284 +       gracePeriod := time.Duration(sc.gracePeriodRatio*1000) * certLifeTime / 1000
284 285         rootCertificate := sc.ca.GetRootCertificate()
285 286
286 287         // Refresh the secret if 1) the certificate contained in the secret is about
```

Mutual TLS

- Binds non-replayable service identities to TLS channels
- Can be enforced using policies on:
  - Cluster level
  - Namespace level
  - Service level

TLS Destination rules

```
apiVersion: "networking.istio.io/v1alpha3"
kind: "DestinationRule"
metadata:
  name: "default"
  namespace: "default"
spec:
  host: "*.local"
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

https://istio.io/docs/tasks/security/authn-policy/

Istio 0.5 example of authorization

```
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: denyproductpage
spec:
  match: destination.labels["app"] == "details"
   && source.user == "cluster.local/ns/default/sa/bookinfo-productpage"
  actions:
  - handler: denyproductpagehandler.denier
    instances: [ denyproductpagerequest.checknothing ]
```

ServiceRole - Mutual TLS Authorization

```
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: tester
  namespace: default
spec:
  rules:
  - services: ["test-*"]
    methods: ["*"]
  - services: ["bookstore.default.svc.cluster.local"]
    paths: ["*/reviews"]
    methods: ["GET"]
    constraints:
    - key: request.headers[version]
      values: ["v1", "v2"]
```

https://istio.io/docs/concepts/security/#authorization-policy

ServiceRoleBinding - Mutual TLS Authorization

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRoleBinding
metadata:
  name: test-binding-products
  namespace: default
spec:
  subjects:
  - user: "spiffe://cluster.local/ns/demo/sa/product-viewer"
  roleRef:
    kind: ServiceRole
    name: "products-viewer"
```

https://istio.io/docs/concepts/security/#authorization-policy

ServiceRoleBinding - Binding to user and service

```
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRoleBinding
metadata:
  name: role-binding-user-admin
  namespace: demo
spec:
  subjects:
  - user: "spiffe://cluster.local/ns/demo/sa/user-admin-ui"
    properties:
      request.auth.claims[email]: "erlend@oftedal.no"
  roleRef:
    kind: ServiceRole
    name: "user-admin"
```

End user authentication

- JWT
- Istio checks signature
- Auth0, Firebase Auth, Google Auth, and custom auth

ServiceRole - End user authentication

```
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
  name: "jwt-example"
spec:
  targets:
  - name: the-micro-service
  peers:
  - mtls: {}
  origins:
  - jwt:
      issuer: "demo"
      jwksUri: "https://login.demo.no/.well-known/jwks.json"
      audiences:
      - demo-microservice
  principalBinding: USE_ORIGIN
```

https://istio.io/docs/concepts/security/#authorization-policy

Istio authorization vs service authorization

- Coarse grained
- Horizontal authorization must be handled by services/application
  - Between users
  - Between clients (services)
- Istio does not currently forward SPIFFE-identities to the application:
  - https://github.com/istio/old_issues_repo/issues/165
  - But may be included in 1.1: https://github.com/istio/istio/issues/8263

Why a sidecar is helpful for mutual TLS

- Support for short-lived certs is uncommon
  - Java
    - Keystores - Read on start
  - Nginx/Apache
    - Read cert/key files on start
    - Can be re-read if signalled
  - Node.js and others
    - Normally reads cert/key-files on start

Using Istio certs without the sidecar

- Certs are just k8s secrets
- Can be mounted as volumes into containers
- Volume secrets are auto-updated in running pods
- Our application can watch (inotify) and reload

node.js: Watching for new certs

```
...
fs.watch('/secrets/', {}, (eventType, filename) => {
  if (filename == '..data') {
   //We have a new certificate
  }
});
...
```

BASH: Reloading nginx on new certs

```
...
nginx -g "daemon off;" &
inotifywait -e moved_to -m /secrets/ |
while read -r directory events filename; do
    echo "Change detected: $filename $events"
    echo "Reloading nginx..."
    nginx -s reload
done
...
```

```
SubjectAltName: spiffe://cluster.local/ns/demo/sa/webcert
NotBefore: Wed Feb 20 2019 11:18:46 GMT+0000 (UTC)
NotAfter: Tue May 21 2019 11:18:46 GMT+0000 (UTC)
SerialNumber: 41cb23376e888f0c6b5f33376f7cfb8f
Issuer: k8s.cluster.local
```

A service mesh caution

- Misconfiguration (OWASP Top 10 A6)
- Broken access control (OWASP Top 10 A5)
  - Vertical vs horizontal authorization
- Impedance mismatch
  - istio check vs micro service use
- Istio bugs
- Automated verification tests

Example: Misconfiguration

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
  labels:
    app: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - name: backend
        image: authz-backend:3
        imagePullPolicy: Never
        # ports:
        # - containerPort: 8081
```

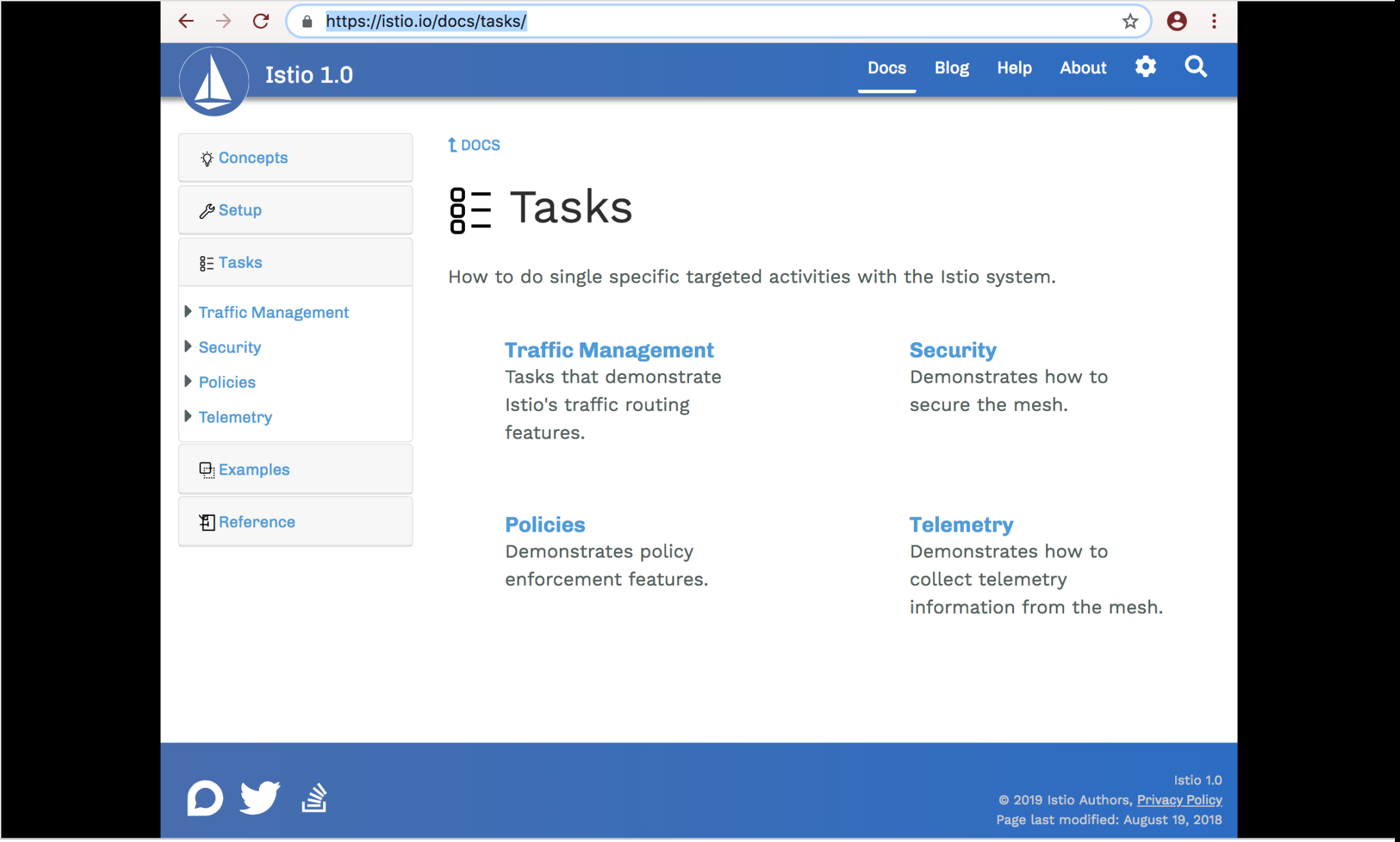
Side note on bugs

- Istio 1.0.2 was released September 2018
- "Fixed bug in Envoy where the sidecar would crash if receiving normal traffic on the mutual TLS port."

Are service meshes the new app servers?

Summary: Istio security in one slide

- Service-to-service authentication through mutual TLS
- End-user authentication through JWT
- Authorization through policies
- Rate limiting and other features through policies


https://istio.io/docs/tasks/
Istio 1.0
Docs
Blog
Help
About
Concepts
Setup
Tasks
Traffic Management
Security
Policies
Telemetry
Examples
Reference
DOCS
Tasks
How to do single specific targeted activities with the Istio system.
Traffic Management
Tasks that demonstrate Istio's traffic routing features.
Security
Demonstrates how to secure the mesh.
Policies
Demonstrates policy enforcement features.
Telemetry
Demonstrates how to collect telemetry information from the mesh.
Istio 1.0
© 2019 Istio Authors, Privacy Policy
Page last modified: August 19, 2018

Resources

- "Docker Security" - O'Reilly - Adrian Mouat
- "Docker Security Quick Reference" - Kim Carter
- https://istio.io/docs/concepts/security/
- Istio: Defense in Depth for Modern Production Environments (Cloud Next '18) https://www.youtube.com/watch?v=eOI2aM9P7-c
- https://www.youtube.com/watch?v=346WmxQ5xtk
- DevSecOps London Gathering videos from Control Plane: https://www.youtube.com/channel/UCR4oVMkRjNN2OQaWMiBcfJA/videos

Thank you for listening!

Erlend Oftedal
@webtonull
eo@blank.no
Blank AS