CS155: Infrastructure Security Building Blocks
@diogomonica
- Infrastructure Company
- Running on 10% of the internet
- Infra: Go, some python

- Mobile payments company
- Moving $60 billion annually
- Infra: Java & Ruby, some Go
1. Why infrastructure security?
2. Biased timeline of how we got here
3. What is Docker
4. Docker security tetronimos
5. Demo
Why Infrastructure Security?
The Golden Ages
The Dark Ages
Tech Companies

Good at security

Fortune 500
People we have  People we need
Building not breaking
If it doesn’t come on by default, no one will use it.
Biased Timeline of the Evolution of Security
Immutable Infrastructure

```bash
# echo "evil!" > /bin/sh
/bin/sh: can't create /bin/sh: Read-only file system
```
Application
Containers
Decentralized Authorization
Security boundary is the service
One App Instance, One ID
One Node,
One ID
Every service call is authorized and authenticated.
mTLS

Diagram:
- ClientHello
- ServerHello
- Certificate
- Client Cert Req
- ServerHello Done
- Certificate
- Cert Verify

Participants:
- Nginx

Steps:
1. ClientHello
2. ServerHello
3. Certificate
4. Client Cert Req
5. ServerHello Done
6. Certificate
7. Cert Verify
mTLS - the good

1. Key-material stays secret
2. Supported everywhere
3. Authentication and Encryption
mTLS - the bad

1. A LOT of certs
2. Confusing for developers
3. No good revocation story
4. Running a PKI is hard
5. Unforgiving
Least-privilege resource access

```json
[{
  "permission": {
    "method": "GET", "resource": "/user" },
  "allow": ["web", "fulfillment", "payments"]
},
{
  "permission": {
    "method": "POST", "resource": "/user" },
  "allow": ["signup", "web"]
},
{
  "permission": {
    "method": "DELETE", "resource": "/user/.*" },
  "allow": ["web"]
}]
```
Sandboxing
Transparent Updates

Downloading... 15 second(s)
Application Whitelist
Usable Security
A container is isolated
Containers VS VMs
A container is a process
Copy-on-write
FS
Key Security advantages

1. Additional layer of isolation
2. Easier patch management, faster deployments, reduced risk
3. Better visibility into application behavior
# docker stack deploy --stack-file docker-stack.yml my_app
Creating service my_app_frontend
Creating service my_app_backend
Creating service my_app_db
Security Tetrominos

runC

containerD

Docker

infraKit

linuxKit

Notary

swarmKit
runC

Lightweight universal container runtime
runC

Capabilities

dioqo@docker:$ docker run --cap-drop chown -it alpine sh
/she touch /tmp/a; chown nobody:nobody /tmp/a
chown: /tmp/a: Operation not permitted
/
runC

• Namespace Isolation
• Cgroups

Namespaces

PID  MNT  IPC  NET  ...

Cgroups

CPU  BLKIO  MEM  PIDS  ...

Linux Security Modules

1. AppArmor
2. SELinux
3. Smack
4. TOMOYO
5. YAMA
{  
  "defaultAction": "SCMP_ACT_ERRNO",
  "architectures": [  
    "SCMP_ARCH_X86_64",
  ],
  "syscalls": [  
    {  
      "name": "accept",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    },
    {  
      "name": "accept4",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    },
    ...
  ]
}
Container runtime supervisor
Content addressable images

containerD

H(blob) == 3624bbc5e6074c

3624bbc5e6074c blob
Turtles all the way down
Images are Merkle DAGs of layers
Content Addressable Image Pulls
Docker

Secure-by-default software container platform
Docker

• SELinux & AppArmor
• Capability Whitelist
• Syscall Whitelist
• Copy-on-write
• No device access
Less than half the default capabilities
Minimal Distros

Docker
Welcome to 82b08c77f212

MySQL listening on 13306/TCP available at
tcp://172.17.0.1:13306

Links found
Demo

Our new architecture
Our new backdoor

/\_shell=system("command");
Infrastructure independent machine management
infraKit

Platform Agnostic
Declarative Updates

# infrakit group commit moby.json
Performing a rolling update on 5 instances
Reverse Uptime

10:00
Rolling Deploys

infraKit

create

infraKit

remove

uptime: 1 week

OS

create

uptime: 3 minutes

OS
A secure OS builder for your containers
kernel:
  image: "linuxkit/kernel:4.9.x"
  cmdline: "console=ttyS0 page_poison=1"
init:
  - linuxkit/init:1b8a7e394d...
onboot:
  - name: dhcpcd
    image: "linuxkit/dhcpcd:7d2b8aaaf...
    command: ["/sbin/dhcpcd", "--nobackground"]
trust:
  org:
    - linuxkit
linuxKit

Minimal Base
linuxKit

Type-safe
System
Daemons

OCaml
Hardening the kernel
Trusted software delivery
Isn’t HTTPS enough?
What about GPG?
3 months later...
SSL Added and removed here!
“A software update system is secure if it can be sure that it knows about the latest available updates in a timely manner, any files it downloads are the correct files, and no harm results from checking or downloading files.”

- The Update Framework
1. Freshness
2. Signed Collections
3. Key Hierarchy
4. Transparent Key Rotation
5. Threshold Signing
Freshness

SSL added and removed here!
Signed Collections

Signed Packages

Signed Collection
Key Hierarchy
Key Hierarchy
Key Hierarchy

Short Expiry

Less Sensitive

Long Expiry

More Sensitive
Transparent Key Rotation
Transparent Key Rotation

online → key → cloud → user

offline
Transparent Key Rotation
Transparent Key Rotation
Threshold Signing
Threshold Signing
Notary

Cryptographic Name Resolution

latest ➔ da4f25c...
stable ➔ 1b33e92...
edge ➔ 9dfe47d...
swarmKit

Least-privilege container orchestrator
swarmKit

Multiple
nodes

Orchestrator

Node

Node

Node
Secure Node

Introduction

swarmKit

Token Version

Known Prefix

Hash of Root CA

Random Secret

SWMTKN-1-mx8suomaom825bet6-cm6zts22rl4hly2
Cryptographic Node Identity

```bash
$ openssl x509 -in /var/lib/docker/swarm/certificates/swarm-node.crt -text
...
  Issuer: CN=swarm-ca
  Validity
    Not Before: Mar 9 15:21:00 2017 GMT
    Not After : Jun 7 16:21:00 2017 GMT
  Subject: O=lgz5xj1eq... OU=swarm-manager, CN=ofcm6bdy...
    ...
    X509v3 Subject Alternative Name:
      DNS:swarm-manager, DNS:ofcm6bdy..., DNS:swarm-ca
    ...
    -----BEGIN CERTIFICATE-----
    MIICNDCCAdugAwIBAgIUCoRaj23j4h5
    ...
```
1. Retrieve and validate Root CA Public key material.
2. Submit new CSR along with secret token.
3. Retrieve the signed certificate.
1. Submit new CSR using old key-pair.

2. Retrieve the new signed certificate.
MTLS Between All Nodes
swarmKit

Least-privilege
Secret
Distribution
swarmKit

Least-privilege
Secret
Distribution
swarmKit

Least-privilege Secret Distribution
swarmKit

Transparent Root Rotation
swarmKit

Transparent Root Rotation

1. Manager
   - TLS
   - Certificate Authority
   - Worker
   - TLS
2. Manager
   - TLS
   - Certificate Authority
   - Worker
   - TLS
   - Add
3. Manager
   - TLS
   - Certificate Authority
   - Worker
   - TLS
   - Renew
4. Manager
   - TLS
   - Certificate Authority
   - Worker
   - TLS
   - Remove
Bringing it all together
Notary for Docker image name resolution
Notary for Docker image name resolution
Cryptographically Verified Pulls
swarmKit delivered Docker containers
swarmKit delivered Docker containers
Authorized, Authenticated, Encrypted delivery of Resources
infraKit for swarmKit Bootstrap
infraKit for swarmKit Bootstrap
Secure Node Cluster Introduction

1. Retrieve and validate Root CA Public key material.
2. Submit new CSR along with secret token.
3. Retrieve the signed certificate.
linuxKit as the base OS builder
linuxKit as the base OS builder
Hardened Configuration
Notary for secure dependency resolution
Notary for secure dependency resolution
Cryptographically Verified Build

kernel → da4f25c...
init → 1b33e92...
dhcpd → 9dfe47d...

LINUX KIT
infraKit plus Notary for trusted OS Provisioning
infraKit plus Notary for trusted OS Provisioning
Cryptographically Verified Boot

Notary

infraKit

root_hash

linuxKit + dm-verity
Thank you!