Cloud-Native Microservices Security

https://github.com/andifalk/cloud-native-microservices-security

Andreas Falk

Agenda

1. OWASP Top 10
2. Defense against SQL injection, XSS, and CSRF
3. Authentication
   - Basic Auth, Form Login
   - Encryption and password hashing
   - Mutual TLS
4. Authorization
5. Security Testing
6. Container- and Kubernetes Security

Practice

https://andifalk.gitbook.io/cloud-native-microservices-security

https://github.com/andifalk/cloud-native-microservices-security

Why Security ?

#securityfails

Cayla Doll - The Spy Toy

Troy Hunt
@troyhunt
Folge ich

Germany not mucking around with spying toys "Germany Issues Kill Order for a Domestic Spy—Cayla the Toy Doll"

Data from connected CloudPets teddy bears leaked and ransomed, exposing kids' voice messages

28 FEBRUARY 2017

Germany Issues Kill Order for a Domestic Spy—Cayla the Toy Doll

On a campaign to promote digital privacy, authorities warn that "My Friend Cayla" makes children vulnerable to malicious surveillance. They've ordered parents to d...

wsj.com

NOVATEC

Ashley Madison - Serious Life Changer

Troy Hunt @troyhunt · 10. Dez.

Good insight into the human impact: "Scared, dead, relieved: How the **Ashley Madison** hack changed its victims' lives" fusion.net/story/242502/a...

NOVATEC

Car Hacking - Life Threatening

ANDY GREENBERG SECURITY 07.21.15 6:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

I WAS DRIVING 70 mph on the edge of downtown St. Louis when the exploit began to take hold.

NOVATEC

Facebook Leak - Big Data Exposure

Over 540 million Facebook records found on exposed AWS servers

Leak originated at two third-party companies that had collected Facebook data on their own servers.

By Catalin Cimpanu for Zero Day | April 3, 2019 -- 18:32 GMT (19:32 BST) | Topic: Security

MORE FROM CATALIN CIMPANU

Windows 10
Microsoft changes how Windows 10 disconnects USB storage devices

Developer
After Chrome, Firefox will also support off-screen image lazy loading

Security
FBI criticized for delaying breach notifications, including insufficient details

Security
IoT botnet targeting your enterprise? Nope. Just a kid with an ExploitDB account

NEWSLETTERS

ZDNet Security

Your weekly update on security around the globe, featuring research, threats, and more.

NOVATEC

Crypto Mining Via K8s Dashboard

Source: https://blog.heptio.com

On Securing the Kubernetes Dashboard

Joe Beda Follow

Feb 28, 2018 · 13 min read

Recently Tesla (the car company) was alerted, by security firm RedLock, that their Kubernetes infrastructure was compromised. The attackers were using Tesla's infrastructure resources to mine cryptocurrency. This type of attack has been called "cryptojacking".

The vector of attack in this case was a Kubernetes Dashboard that was exposed to the general internet with no authentication and elevated privileges. Not only this, but core AWS API keys and secrets were visible. How do you prevent this from happening to you?

Shodan.io - Google for “Hackers”

Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.

See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!

Monitor Network Security

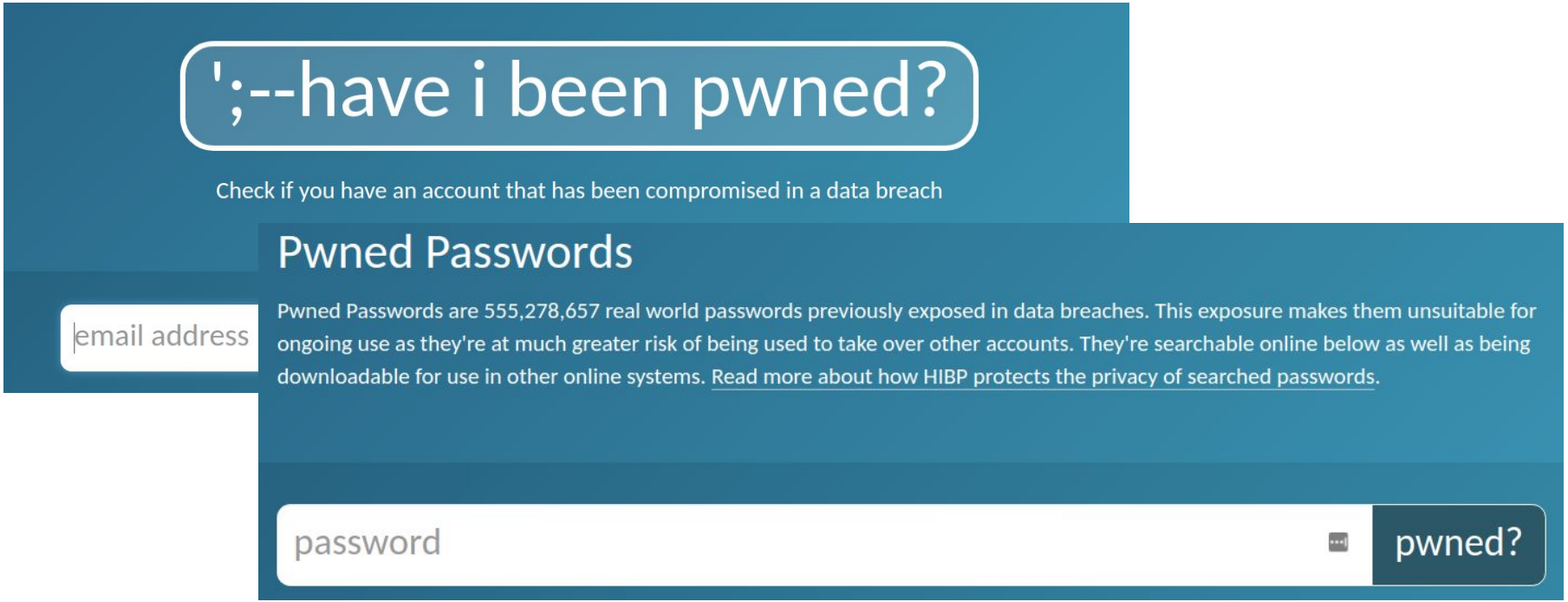
Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.

Get a Competitive Advantage

Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence.

NOVATEC

haveibeenpwned.com - Exposed Accounts & Passwords



NOVATEC

Pay attention on what you copy from stackoverflow.com

This is some what simple

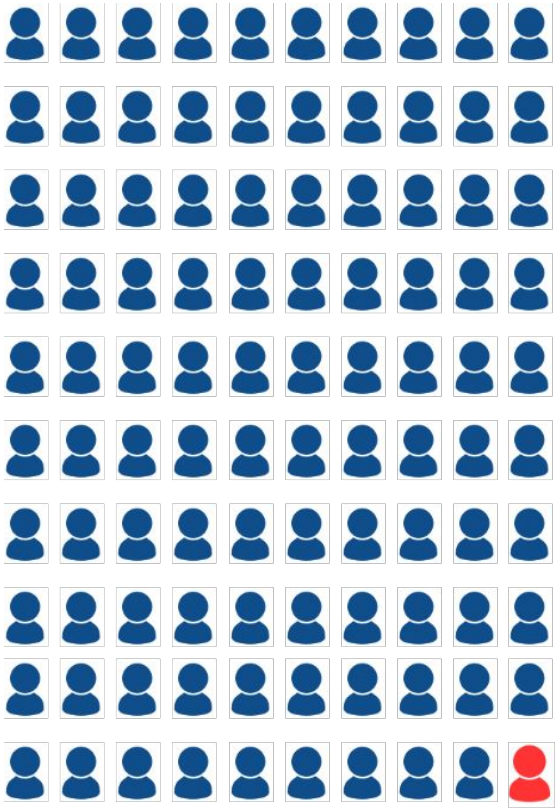
```
string inp = "hai";
StringBuilder strb = new StringBuilder();
foreach (char s in inp)
{
    int sin = s + 5;
    char newch = (char)sin;
    strb.Append(newch);
}
string output = strb.ToString();
```

Now the output contains the encrypted string "mfn" (ie., 5 letters away from the original)in it....

Security is a really comprehensive topic

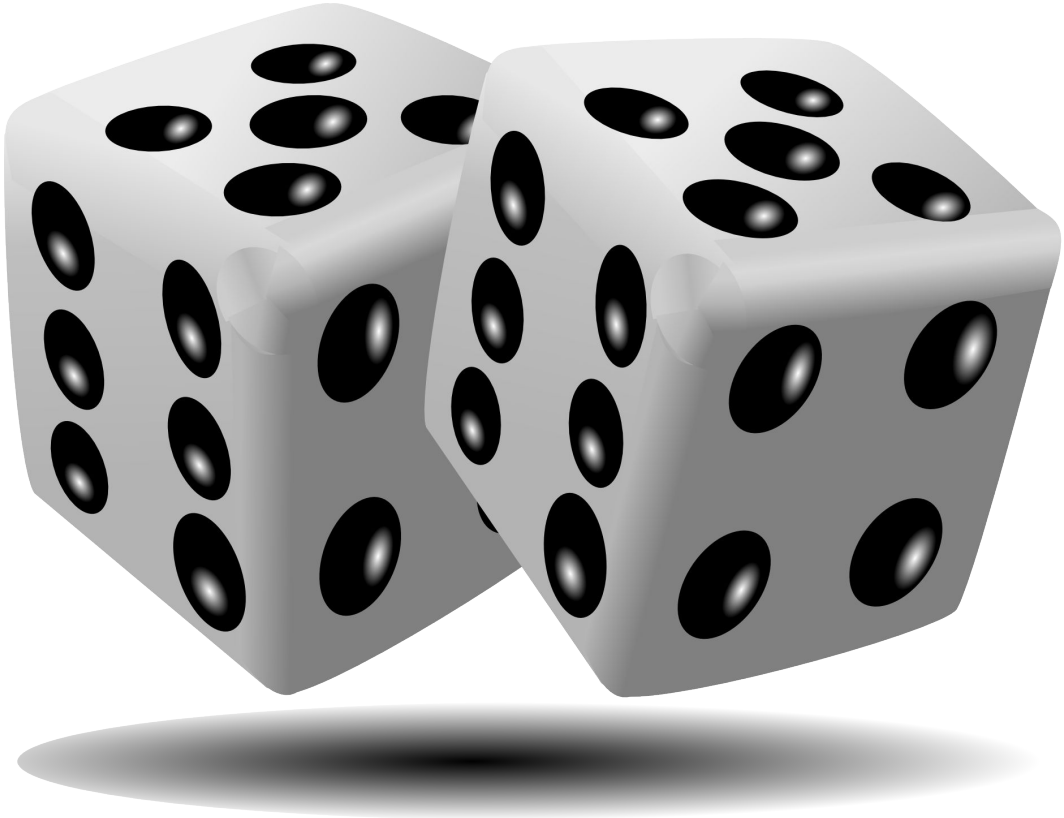
NOVATEC

1 Security-Professional for 100 Developers



Source:
https://www.sonatype.com/devops-survey-report

Security as a game of chance?



NOVATEC

Open ~~Web~~ Application Security Project

https://www.owasp.org

OWASP Top 10 - 2017

OWASP
Open Web Application Security Project

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

https://github.com/OWASP/Top10

NOVATEC

Other OWASP Offerings

OWASP Pro Active CONTROLS

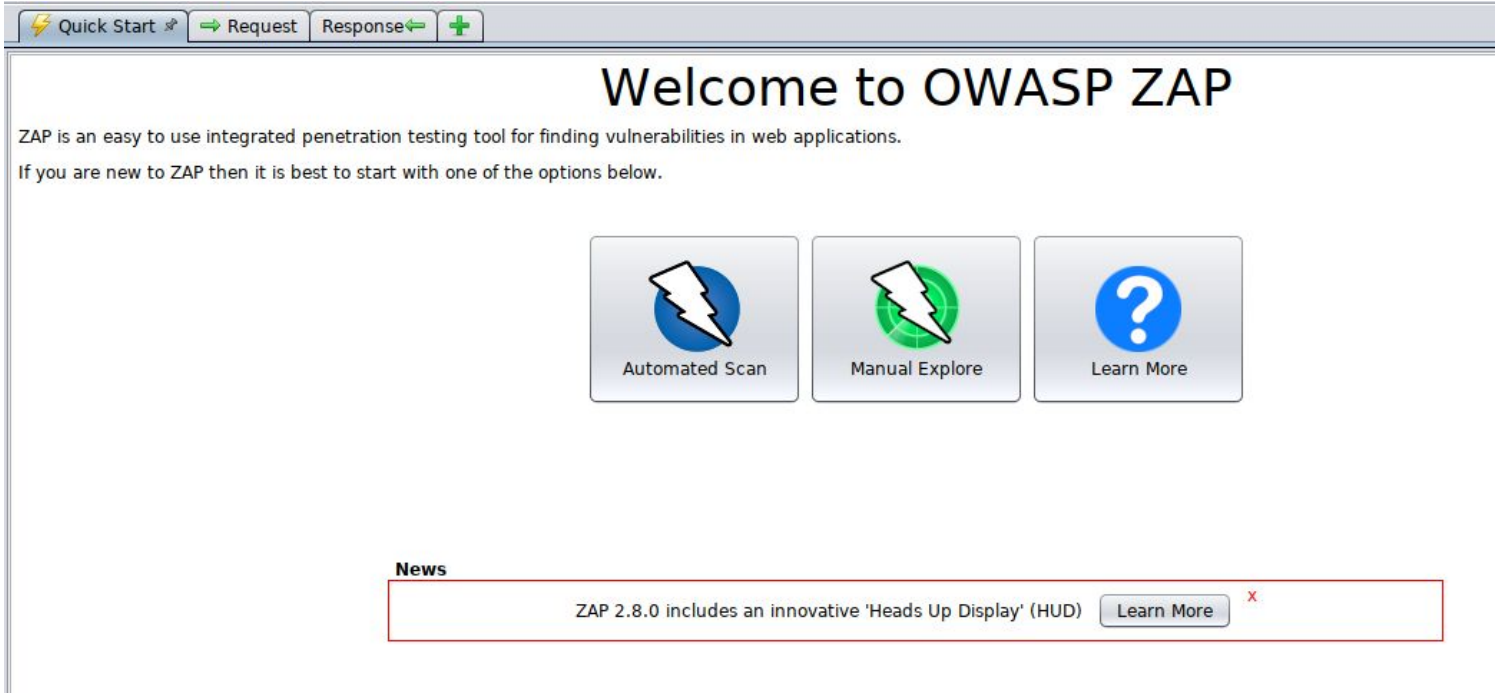
https://www.owasp.org/index.php/OWASP_Proactive_Controls

Application Security Verification Standard

https://github.com/OWASP/ASVS

NOVATEC

OWASP Zap - Open Source Penetration Testing Tool



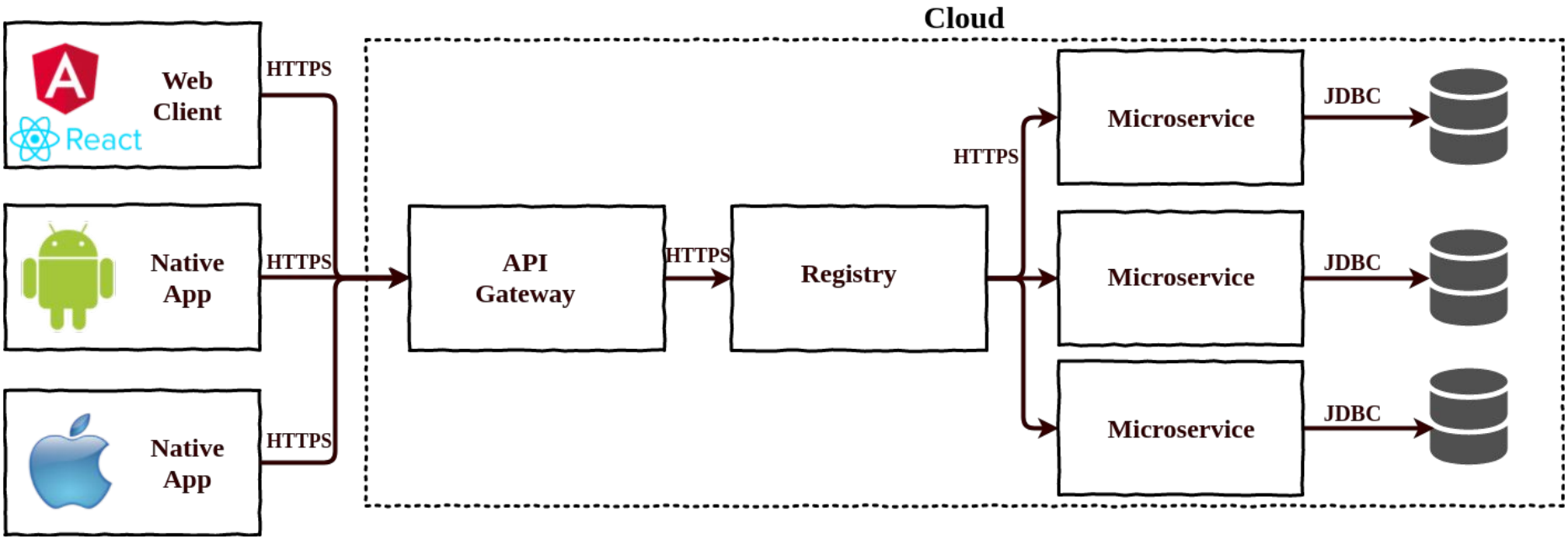
https://www.zaproxy.org/

NOVATEC

Cloud-Native Security

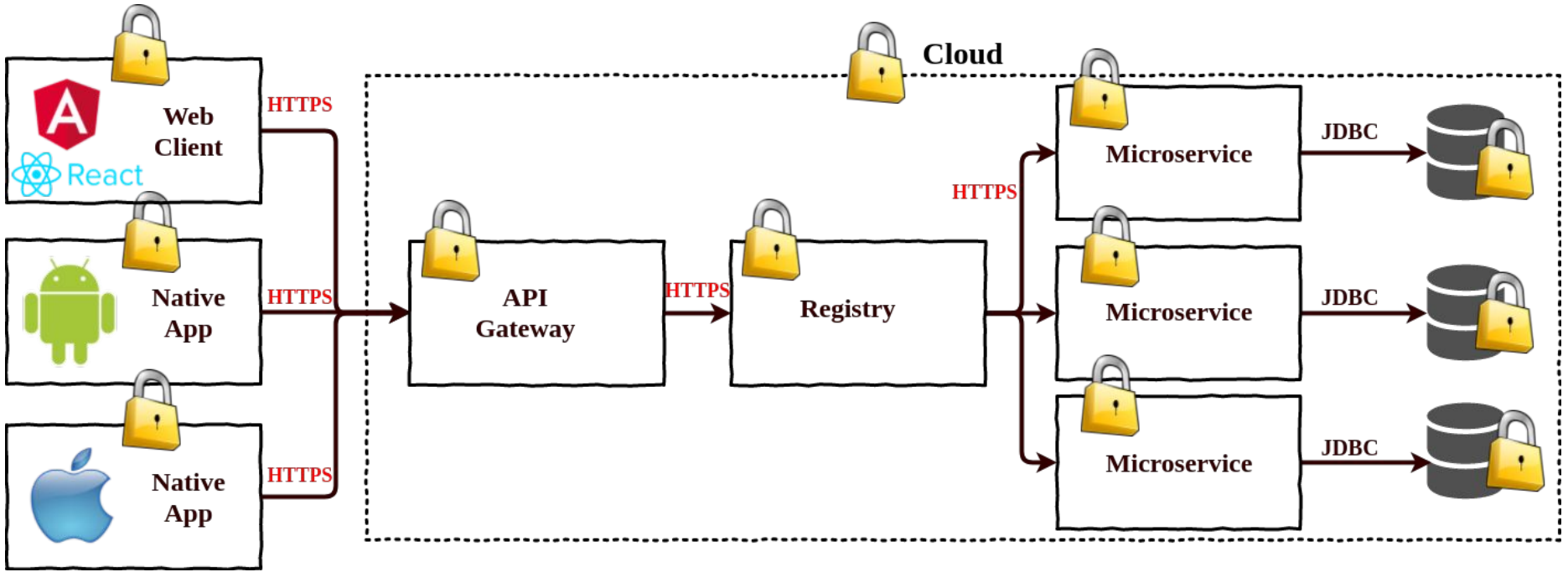
Cloud / CloudFoundry / Kubernetes

Cloud-Native Architecture



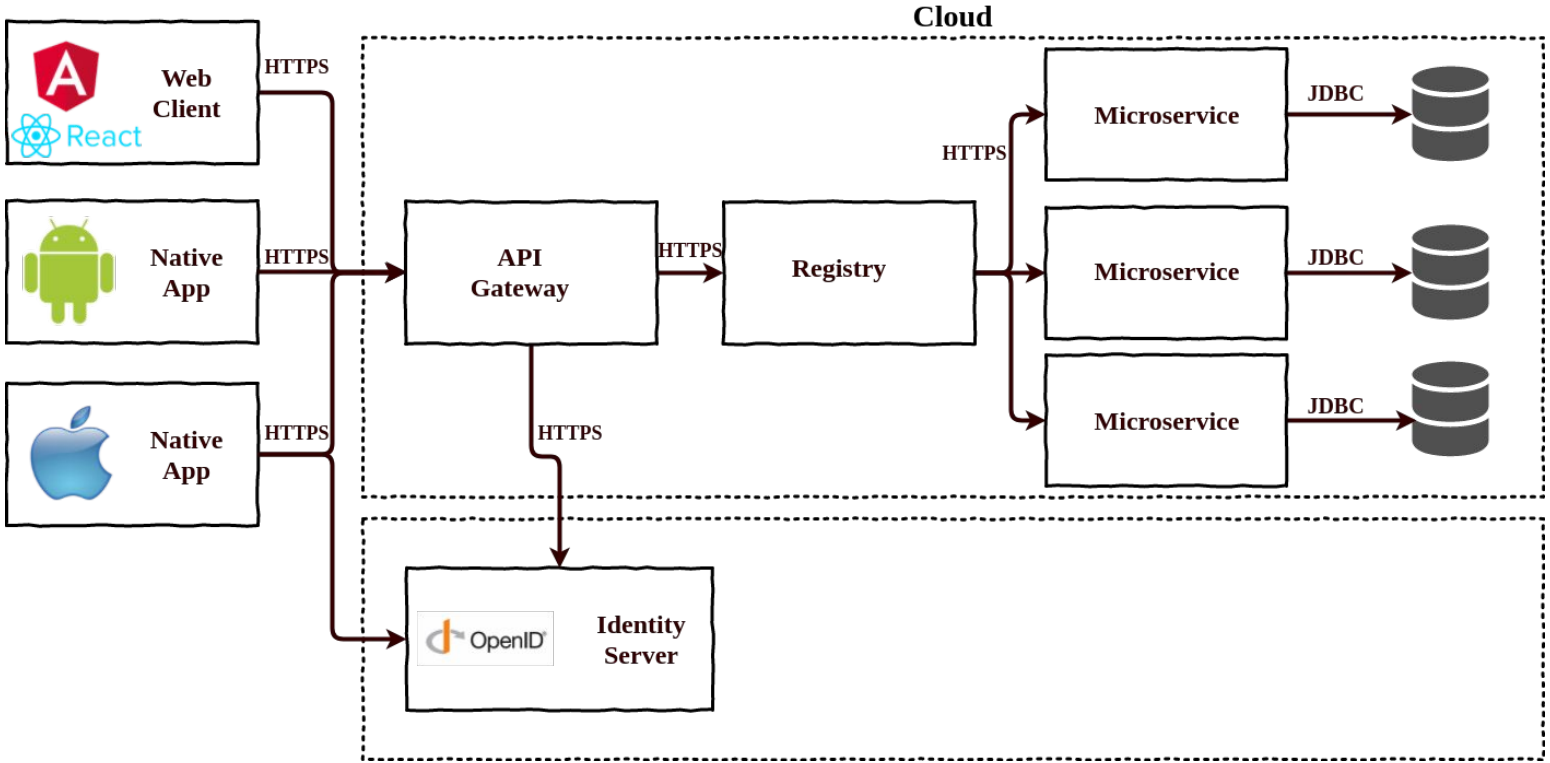
NOVATEC

Secure Cloud-Native Architecture



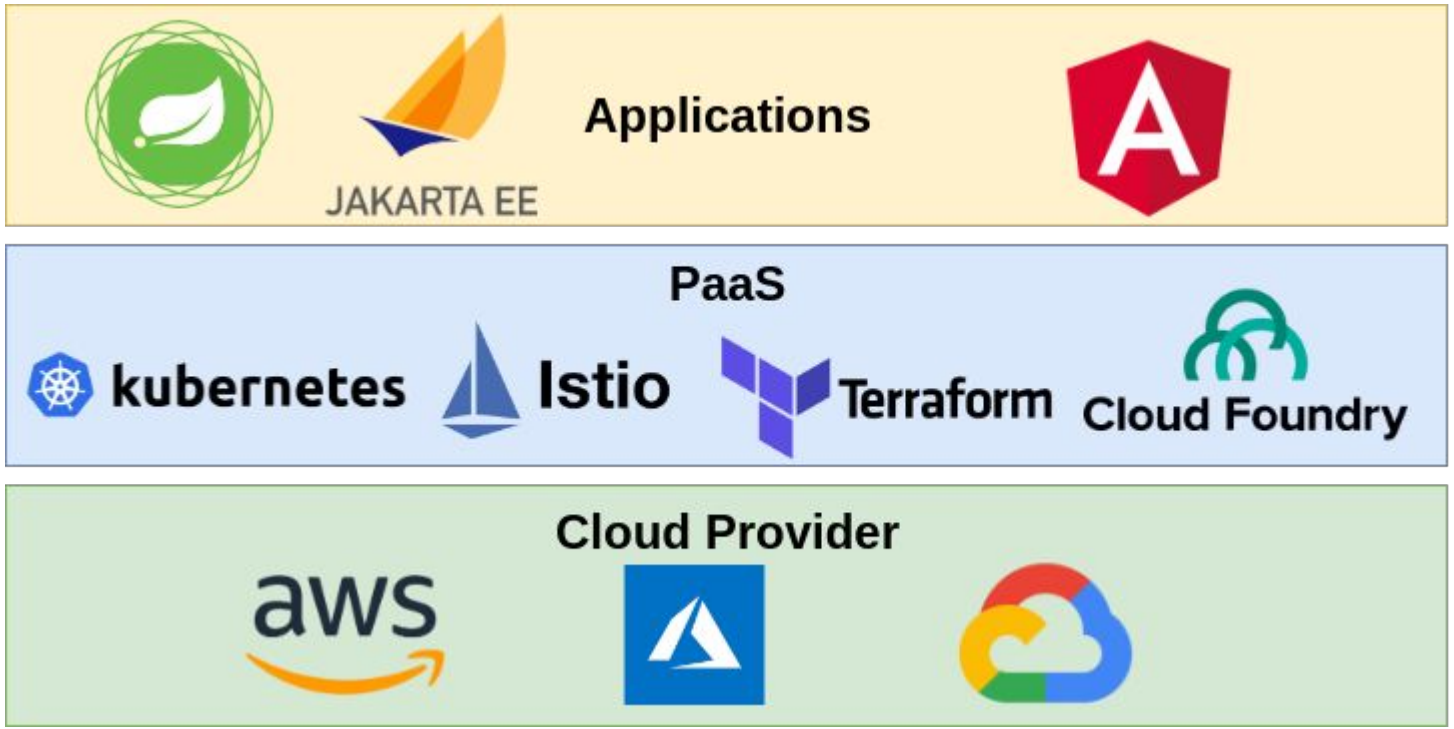
NOVATEC

Cloud-Native Architecture with Identity-Management



NOVATEC

Technology Stacks



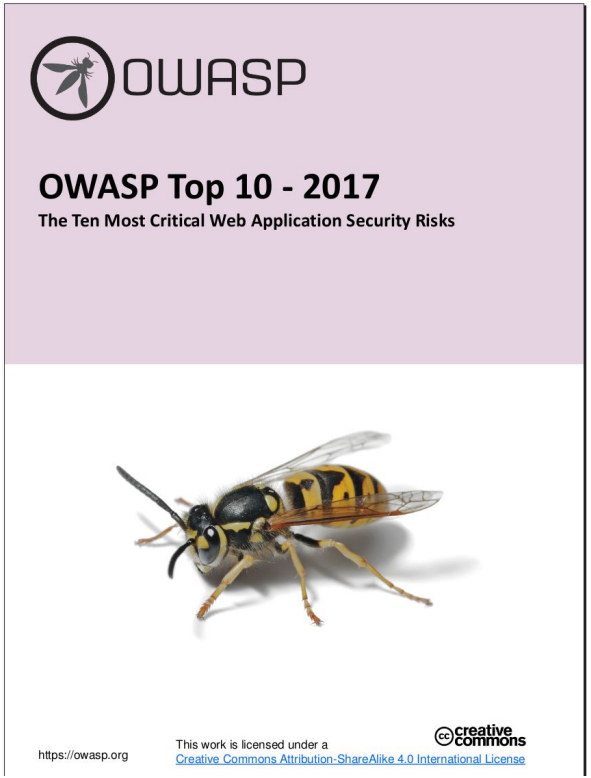
Server Side Security

Java/Kotlin Backends

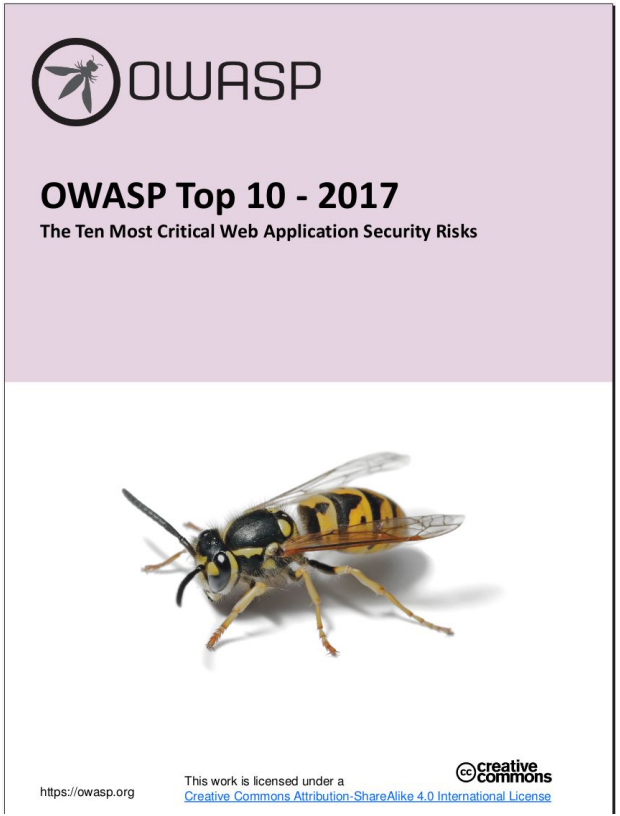
Java / Kotlin Server-Side Applications

OWASP Top 10 on Server-Side

A1: Injection
A2: Broken Authentication
A3: Sensitive Data Exposure
A5: Broken Access Control
A6: Security Misconfiguration
A9: Using Components With Known Vulnerabilities



A1: Injection



NOVATEC

SQL Injection

```
SELECT balance FROM user WHERE username =
            + "'" + request.getParameter('userName')
             + "'";
```

→ Parameter *userName*: Tom

```
SELECT balance FROM user WHERE username = 'Tom';
```

→ Parameter userName: Tom' or 1=1; --

```
SELECT balance FROM user WHERE username = 'Tom' or 1=1
```

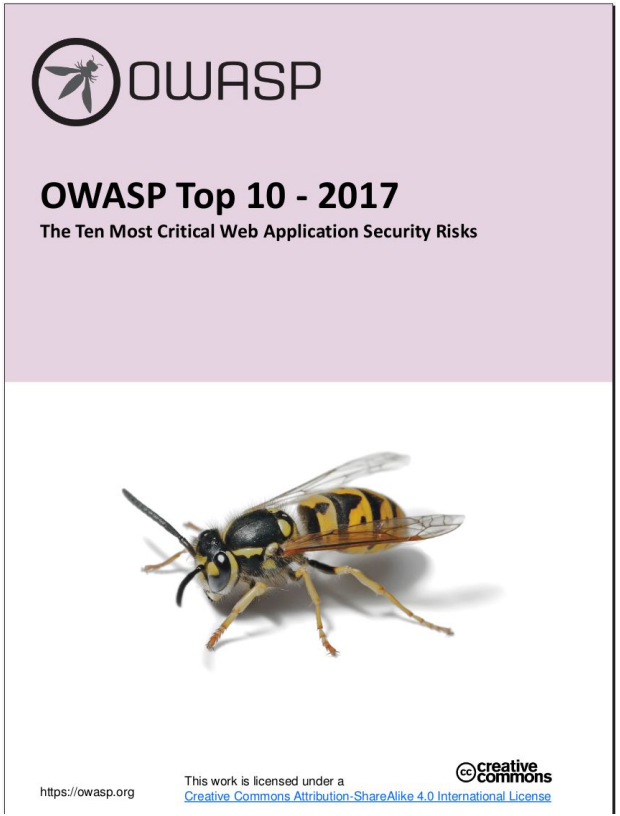
SQL Injection Defensive - Use Input Validation

```
@Entity
public class Person extends AbstractPersistable<Long> {

    @NotNull
    @Pattern(regexp = "^[A-Za-z0-9- ]{1,30}$")
    private String lastName;

    @NotNull
    @Enumerated(EnumType.STRING)
    private GenderEnum gender;
    ...
}
```

SQL Injection Defensive - Prepared Statements

```
@Query(
"select u from User u where u.username = "
+ " :username and u.password = :password")
User findByUsernameAndPassword(
     @Param("username") String username,
     @Param("password") String password);
```

A9: Using Components With Known Vulnerabilities



NOVATEC

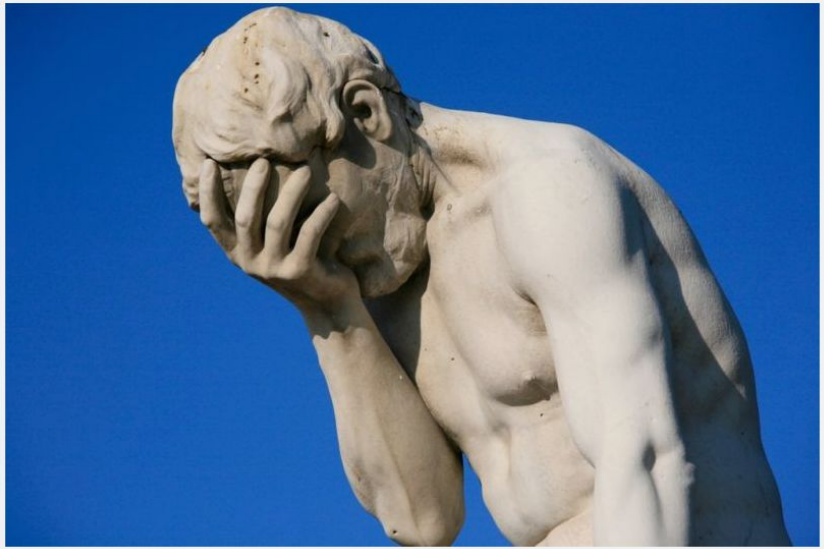
Equifax Struts Vulnerability

BIZ & IT —

Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/14/2017, 5:12 AM



wikimedia commons/alex e. proimos

Enlarge

237

The Equifax breach that exposed sensitive data for as many as 143 million US consumers was accomplished by exploiting a Web application vulnerability that had been patched more than two months earlier, officials with the credit reporting service said Thursday.

NOVATEC

OWASP Dependency Check

- Detects Vulnerabilities in Project Dependencies
- Supports Java and .NET applications
- Experimental: Python, Ruby, PHP, Node.js
- Command line, Ant, Maven, Gradle, Jenkins, SBT

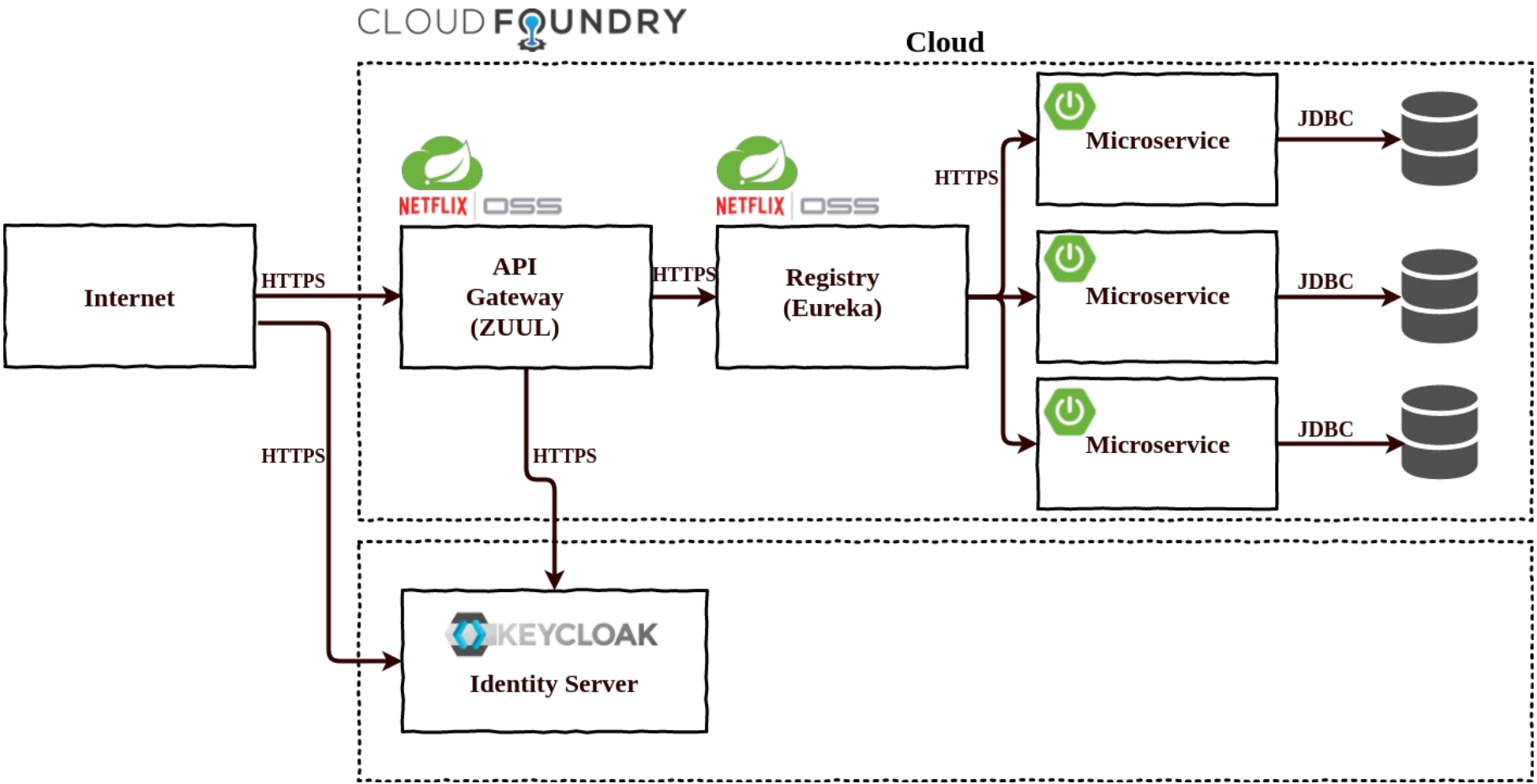
https://github.com/jeremylong/DependencyCheck

Demo Time

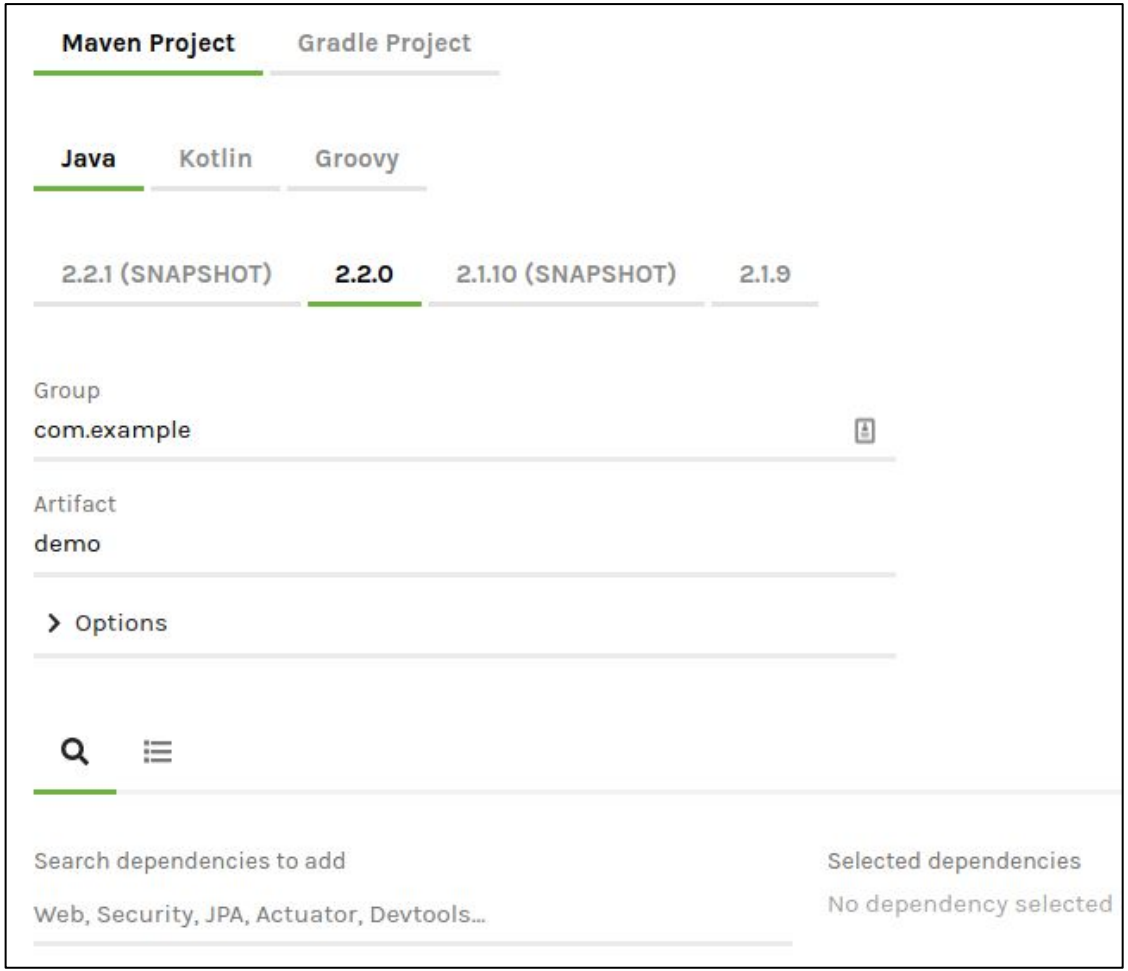
Sample Application with Base Security Defense

The Spring Platform

https://spring.io

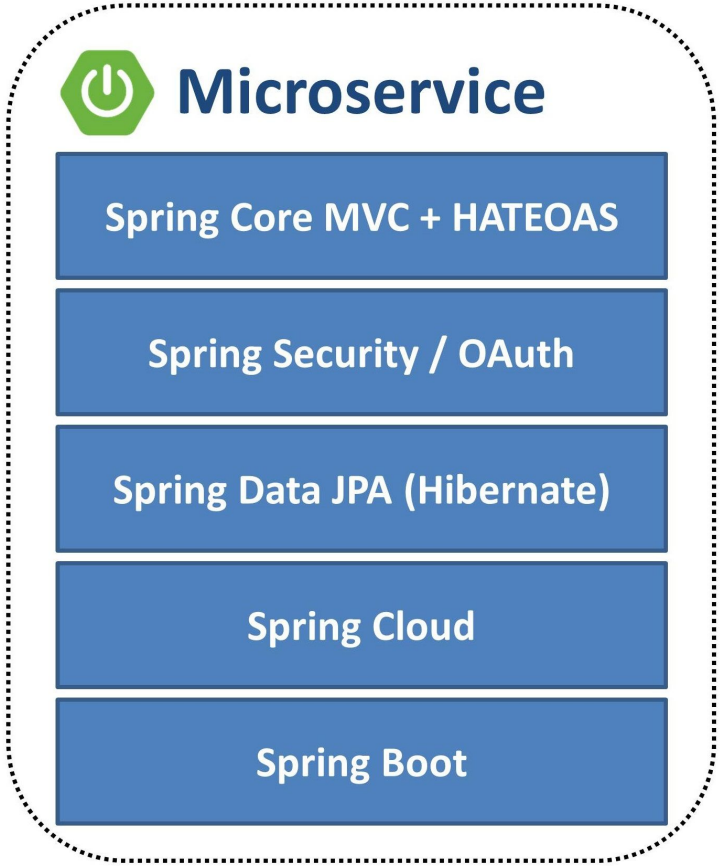

CLOUD FOUNDRY
Cloud
Internet
HTTPS
API
Gateway
(ZUUL)
NETFLIX OSS
HTTPS
Registry
(Eureka)
NETFLIX OSS
HTTPS
Microservice
JDBC
Microservice
JDBC
Microservice
JDBC
HTTPS
HTTPS
KEYCLOAK
Identity Server

NOVATEC



https://start.spring.io

The Spring Microservice Tech Stack



Spring Security 5 Basics

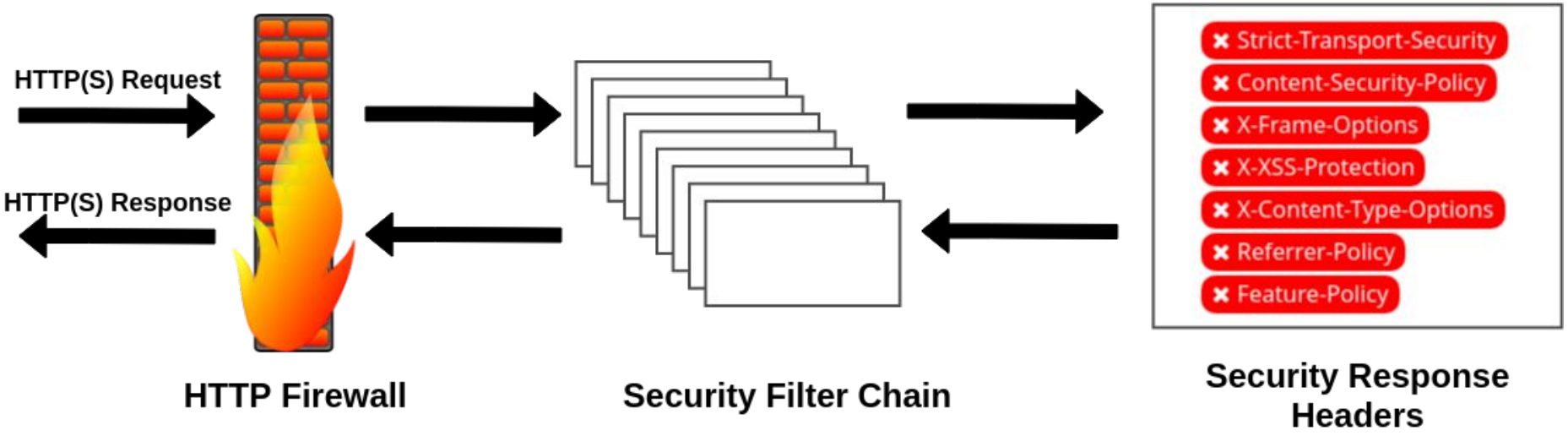
Spring Security 5

- Authentication & Authorization
- Password Encoding
- Support for Servlet & Reactive Web Applications
- Support for OAuth 2.0
- Support for OpenID Connect 1.0, JWT and JOSE (JWS/JWE/JWK)
- Testing Support for Auth/Authz/JWT

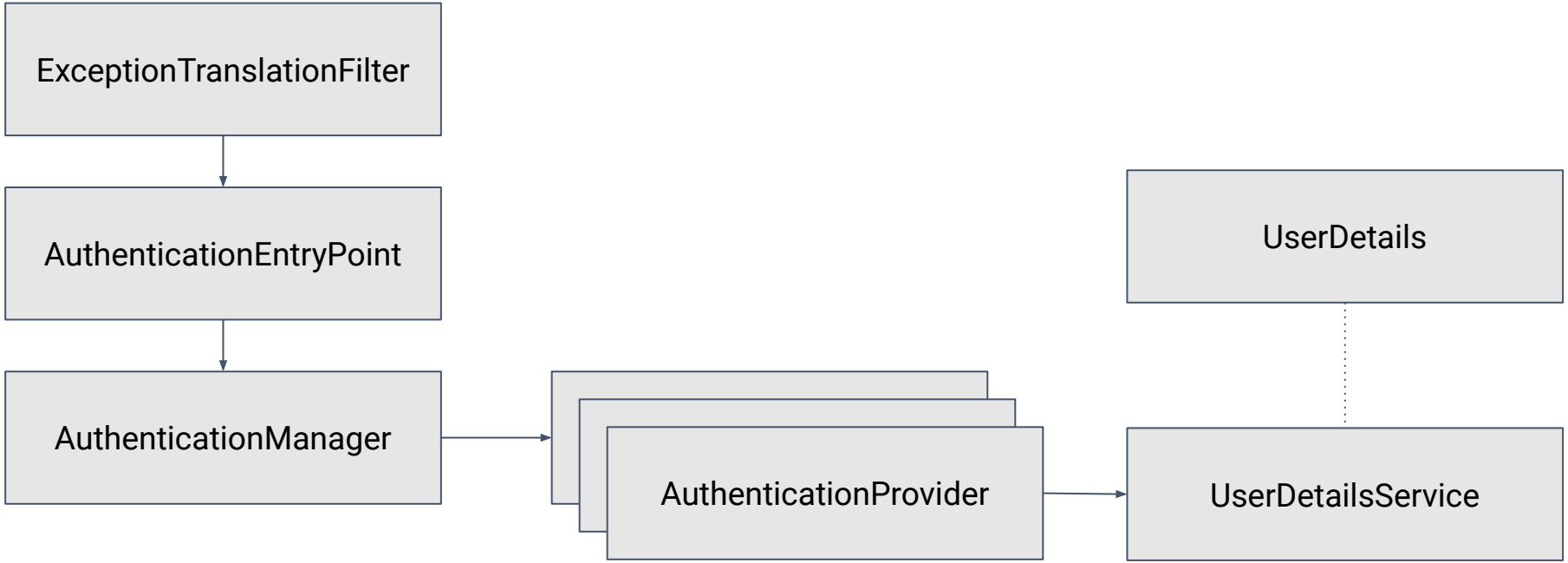
Spring Security 5 - Secure by Default

- Authentication required for all HTTP endpoints
- Session Fixation Protection
- Session Cookie (HttpOnly, Secure)
- CSRF Protection
- Security Response Header

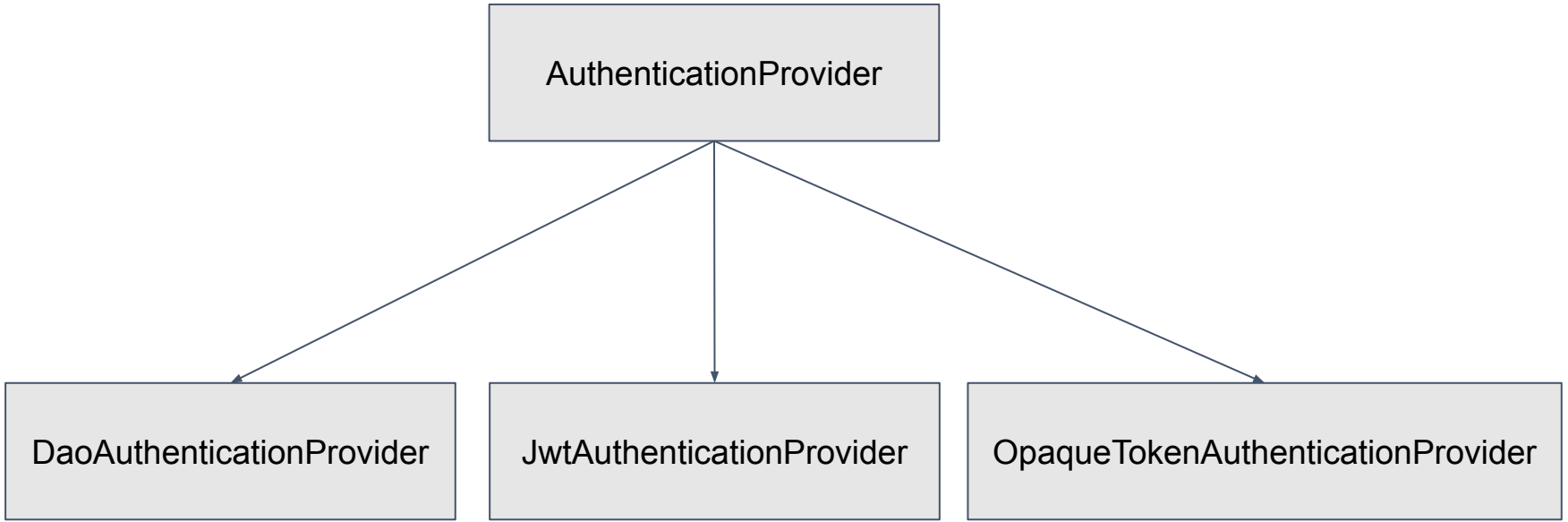
Spring Security High Level View



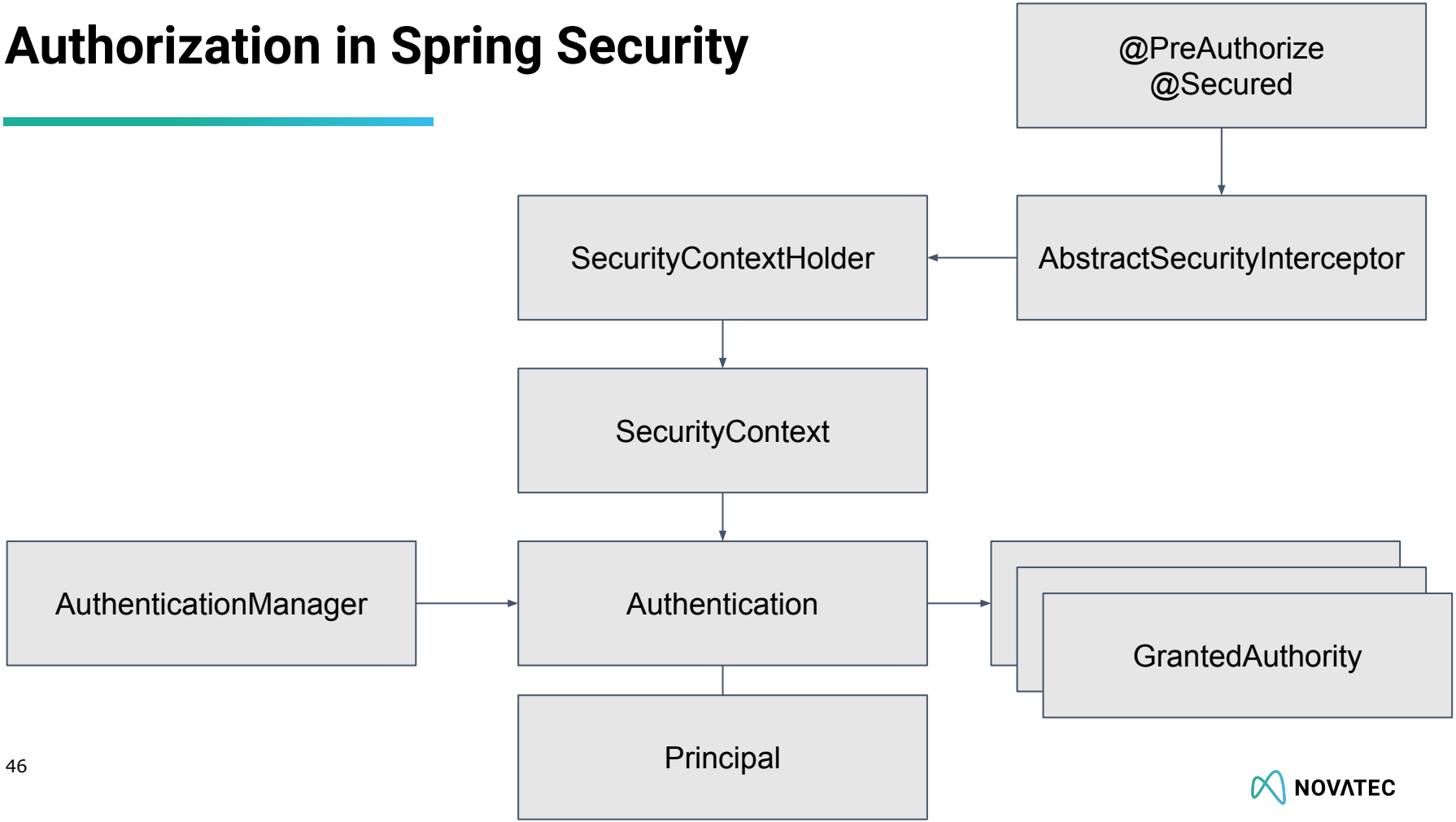
Authentication in Spring Security



Authentication in Spring Security



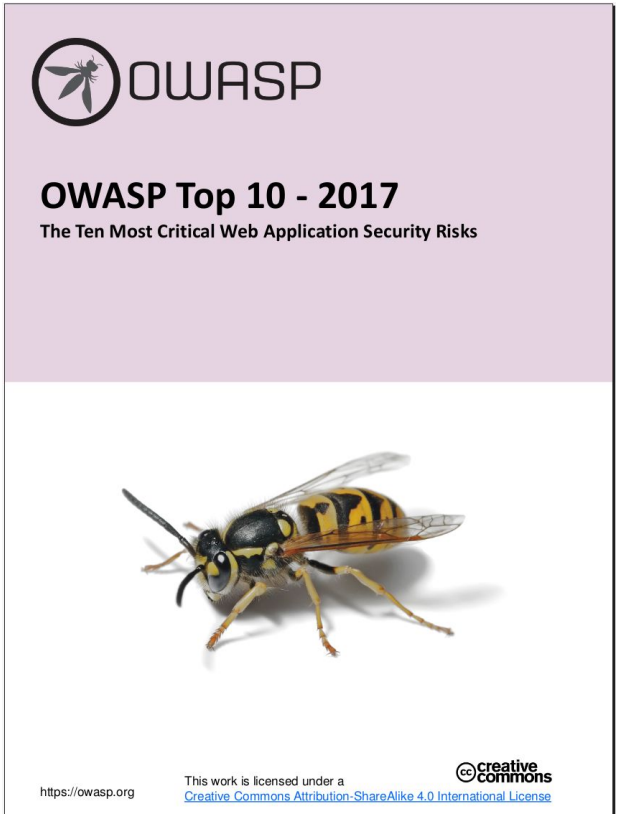
Authorization in Spring Security



NOVATEC

Authentication

A2: Broken Authentication



NOVATEC

Authentication - Who am I?

HTTP 401 - UNAUTHORIZED

Authentication

Knowledge Factor (something the user knows):
Password, PIN, security question,...

Ownership Factor (something the user has):
ID card, security token, cell phone holding a software token,...

Inherence Factor (something the user is):
Fingerprint, retinal pattern,...

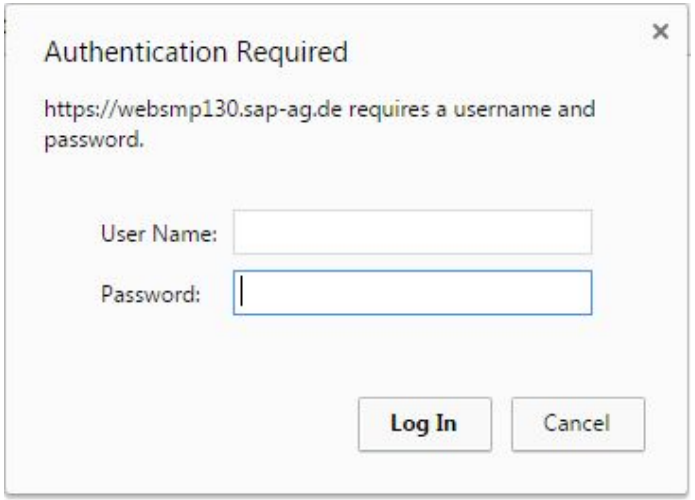
Authentication

- **Single-Factor Authentication**
- **Multi-Factor Authentication**

Common Authentication Mechanisms

- Basic Authentication / Digest Access Authentication
- Form-based Authentication (i.e. using Session Cookies)
- Client-Certificates (Mutual TLS)
- Kerberos Tickets
- Proprietary mechanisms like API-Tokens, Siteminder etc.
- SAML Assertion Tokens
- JSON Web Tokens
- OAuth 2.0 & OpenID Connect 1.0
- WebAuthn / FIDO2

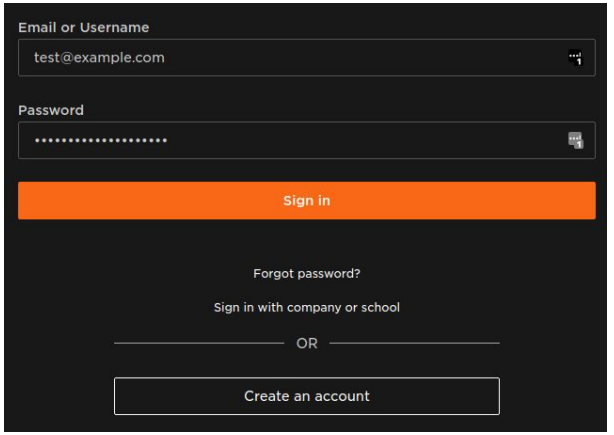
Basic Authentication




```
GET / HTTP/1.1
Host: localhost:8080

HTTP/1.1 401
WWW-Authenticate: Basic realm="hello"

GET / HTTP/1.1
Host: localhost:8080
Authorization: Basic dXNlcjpzZWNyZXQ=
```

https://tools.ietf.org/html/rfc7235
https://tools.ietf.org/html/rfc7617

Form-Based Authentication




```
POST /login HTTP/1.1
Host: localhost:8080
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=14965E3A995DA1973F42F308D59727D4
username=user&password=secret&submit=Login

HTTP/1.1 302
Set-Cookie: JSESSIONID=49C63238780031602 1BE;Path=/; HttpOnly

GET / HTTP/1.1
Host: localhost:8080
Cookie: JSESSIONID=49C63238780031602 1BE
```

Bearer Token Authentication

- Used for
  - OAuth 2.0
  - OpenID Connect

```
GET / HTTP/1.1
Host: localhost:8080
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI
```

https://tools.ietf.org/html/rfc6750

Authentication - Stateful vs. Stateless

Session Cookie	Token
With each Request (on same domain)	Manually set as Header
Potential CSRF!	No CSRF possible
One domain	Cross domain (Cross-Origin Resource Sharing)
Sensitive Info (HTTPS)	Sensitive Info (HTTPS)

Practice Time

Lab 1: Auto-Configured Security

Password Encoding

Cryptography

- Encryption
  - Symmetric
  - Asymmetric (Private/Public Key)
- Hashing

Encryption

- **Symmetric Encryption**
  - ~~Data Encryption Standard~~ (~~DES~~)
  - Advanced Encryption Standard (AES)
- **Asymmetric Encryption (Private/Public Key)**
  - RSA
  - Diffie-Hellman (DH)
  - Elliptic Curve Cryptography (ECC)

Hashing

- **Standard Hashing**
  - ~~MD5~~, ~~SHA-1~~
  - SHA-3, SHA-256
- **Password Hashing (Encoding)**
  - BCrypt
  - SCrypt
  - PBKDF2
  - Argon2

Weak Passwords

- **Weak passwords like**
  - 123456, password, secret, qwertz
  - https://github.com/danielmiessler/SecLists
- **Known passwords from data breaches**
  - Pwned Passwords
    https://haveibeenpwned.com/Passwords

Password Policies

- **NIST Guideline 800-63B**
  - At least 8 characters in length.
  - Up to 64 characters in length.
  - Accept all printing ASCII characters
  - Accept Unicode characters
  - No truncation
  - Offer guidance (password-strength meter)

https://pages.nist.gov/800-63-3/sp800-63b.html **(section 5.1.1.2)**

Practice Time

Lab 2: Customized Authentication

HTTPS & Mutual TLS (MTLS)

HTTPS (HTTP Secure)

- HTTP over a TLS Connection
- Secure Socket Layer (SSL)
- Transport Layer Security (TLS)

HTTPS (HTTP Secure)

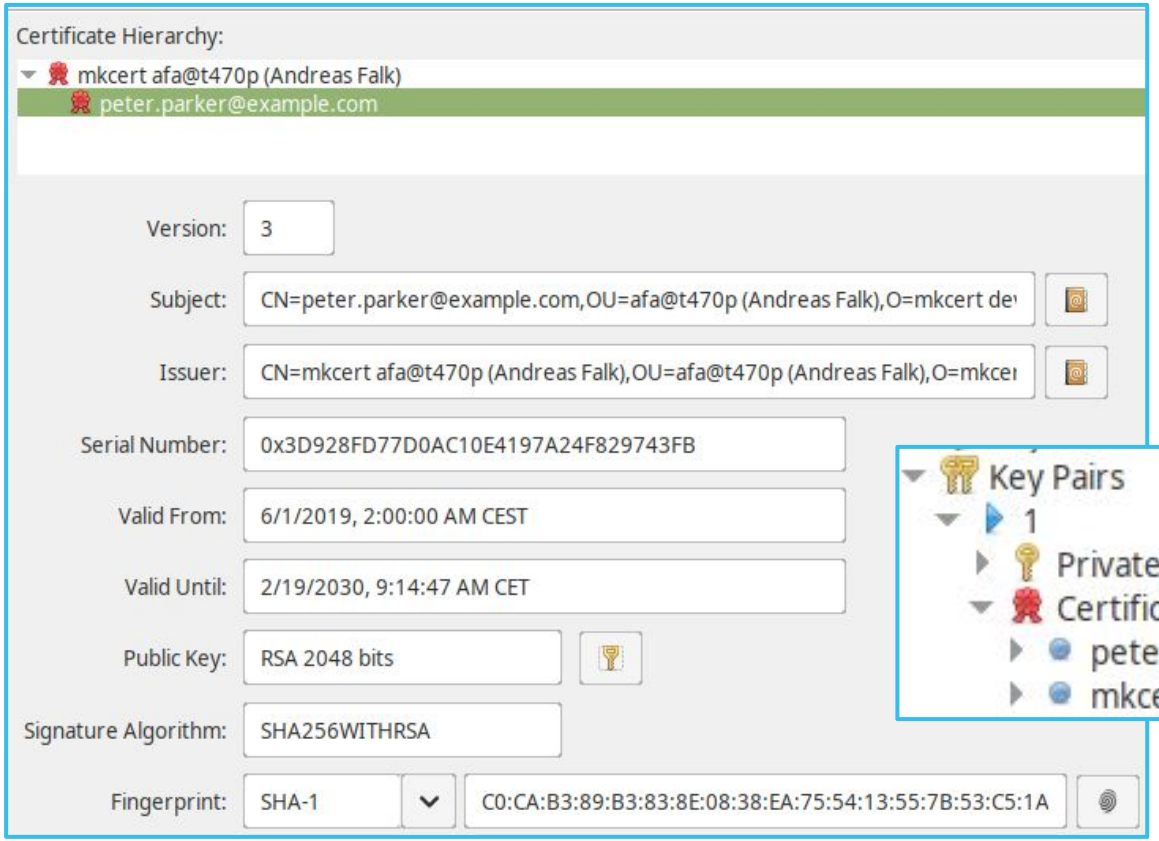
- Validation (Destination Server)
- Data Confidentiality (Encryption)
- Data Integrity (Hashing)

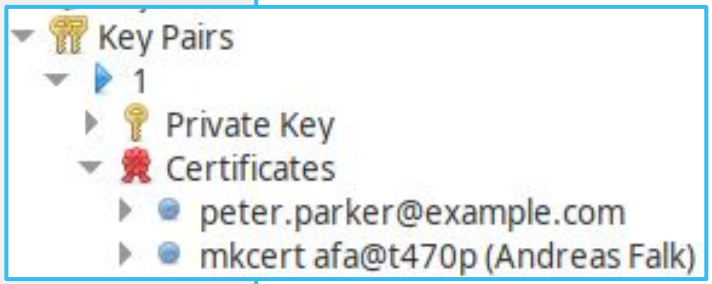
HTTPS (HTTP Secure)

- Secure Socket Layer (SSL)
  - SSL 1.0 
  - SSL 2.0 
  - SSL 3.0 
- Transport Layer Security (TLS)
  - TLS 1.0 
  - TLS 1.1 
  - TLS 1.2 
  - TLS 1.3 

https://security.googleblog.com/2018/10/modernizing-transport-security.html
https://webkit.org/blog/8462/deprecation-of-legacy-tls-1-0-and-1-1-versions

X509v3 Certificates



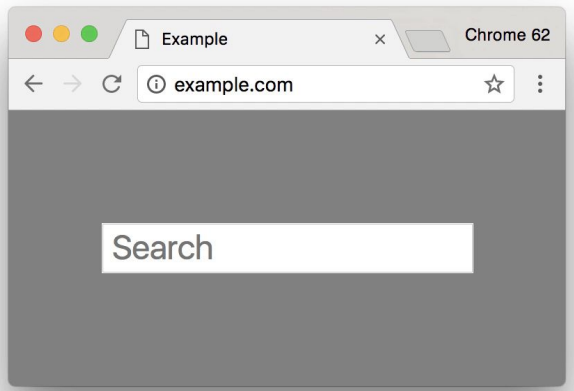


https://tools.ietf.org/html/rfc5280

NOVATEC

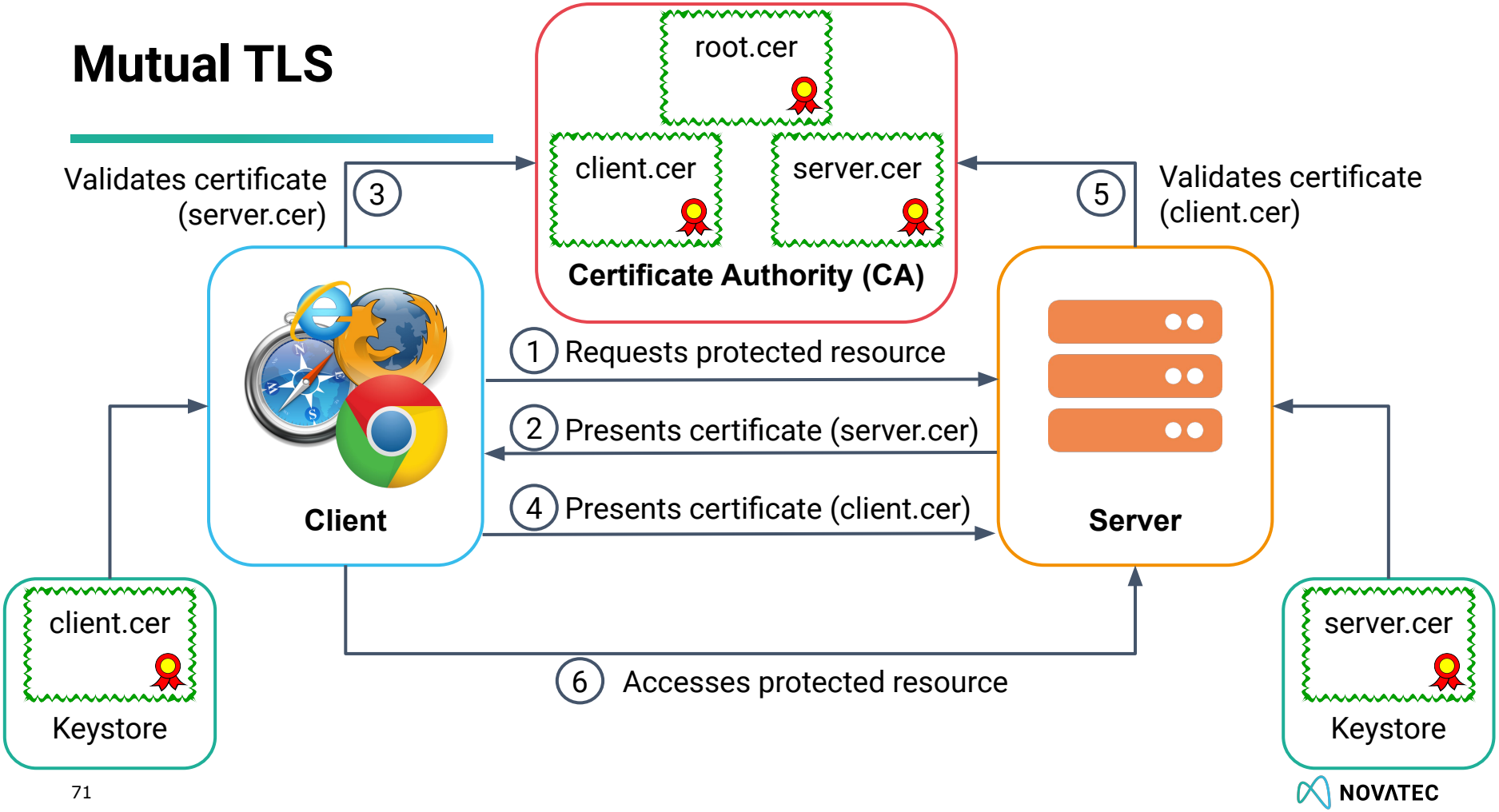
HTTPS and HTTP/2

- Let's Encrypt
- CloudFlare
- HTTP/2



NOVATEC

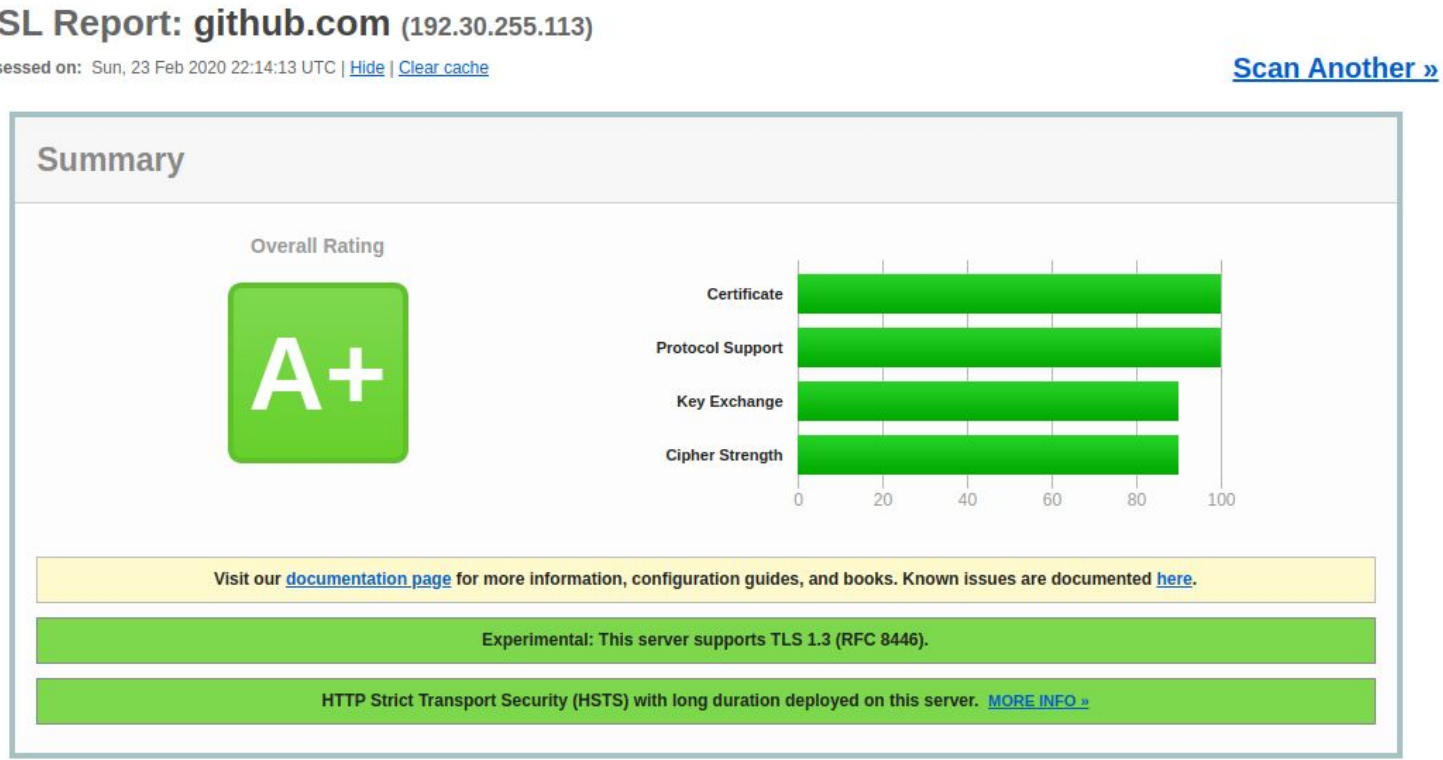
Mutual TLS



NOVATEC

Testing TLS Configuration

https://www.ssllabs.com/ssltest



NOVATEC

Practice Time

Lab 3: Mutual TLS

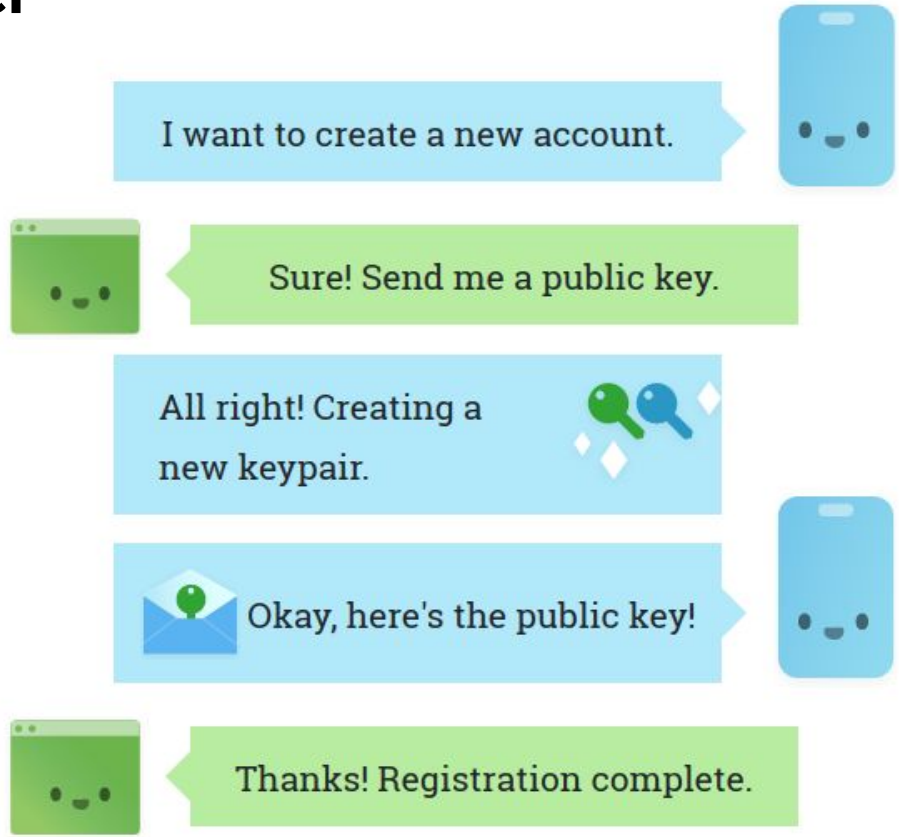
WebAuthn (Web Authentication API)

https://w3c.github.io/webauthn

WebAuthn (Web Authentication API)

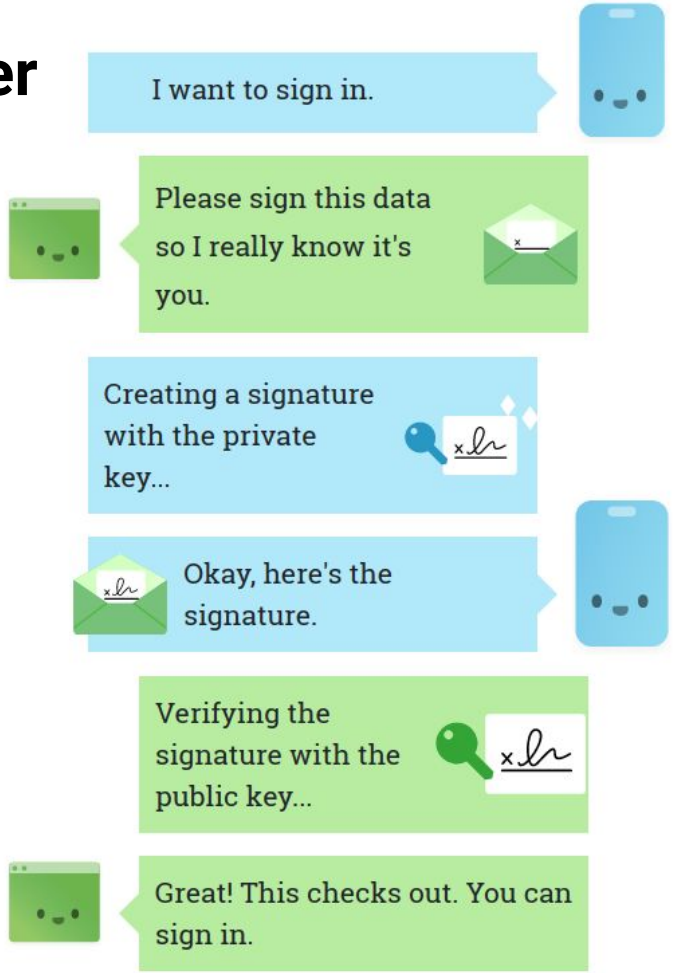
- Specification by W3C and FIDO
- Support by Google, Mozilla, Microsoft, Yubico
- Uses private/public key cryptography to
  - register users
  - authenticate users
- No password required any more!

WebAuthn - Register User



Source: https://webauthn.guide

WebAuthn - Authenticate User

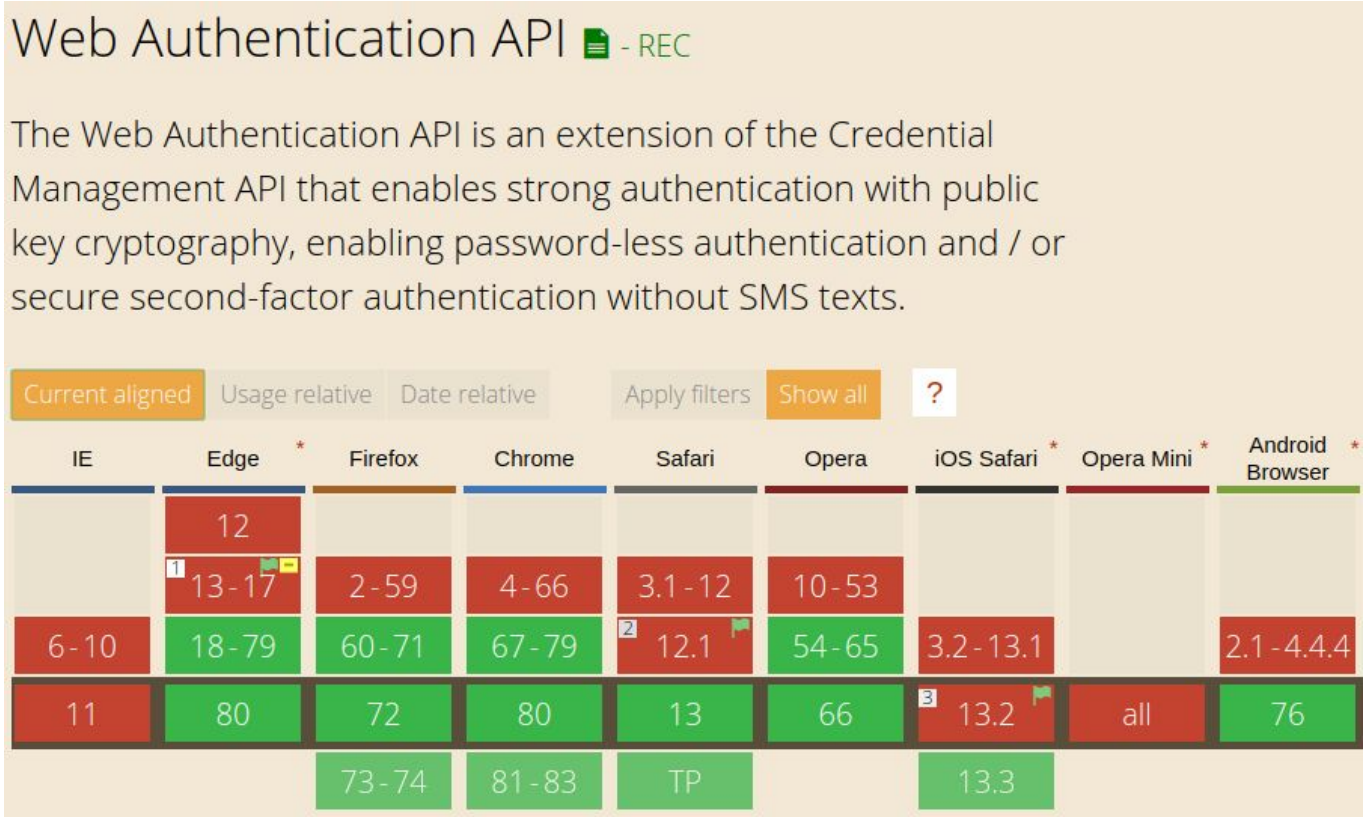


Source: https://webauthn.guide

NOVATEC

WebAuthn - Web Browser Adoption

Source: https://caniuse.com/#search=webauthn



NOVATEC

WebAuthn (Web Authentication Libraries)

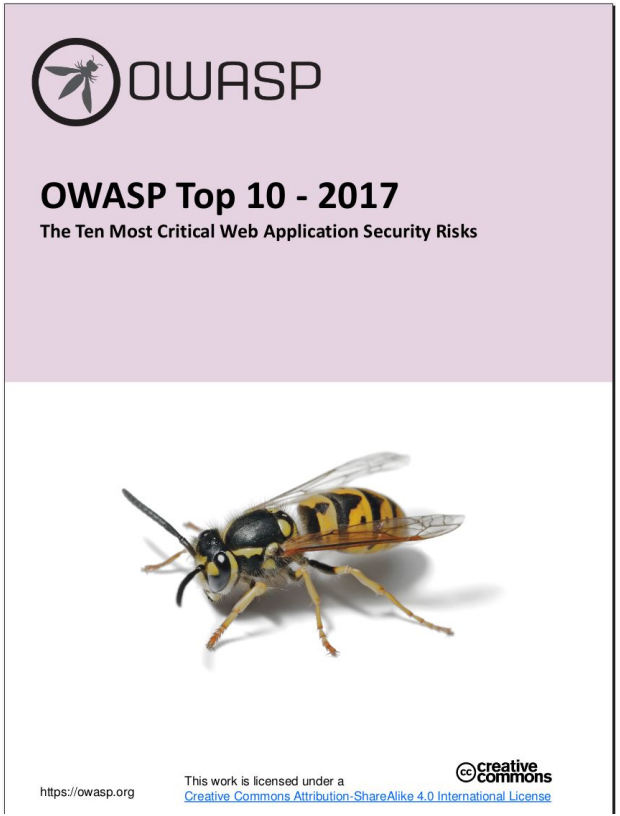
- https://github.com/webauthn4j/webauthn4j (Java)
- https://github.com/abergs/fido2-net-lib (.Net)
- https://github.com/duo-labs/py_webauthn (Python)
- https://github.com/Yubico/java-webauthn-server (Java)
- https://github.com/cedarcode/webauthn-ruby (Ruby)
- https://github.com/duo-labs/webauthn (Go)
- ...

Demo Time

https://github.com/rwinch/spring-security-webauthn

Authorization

A5: Broken Access Control

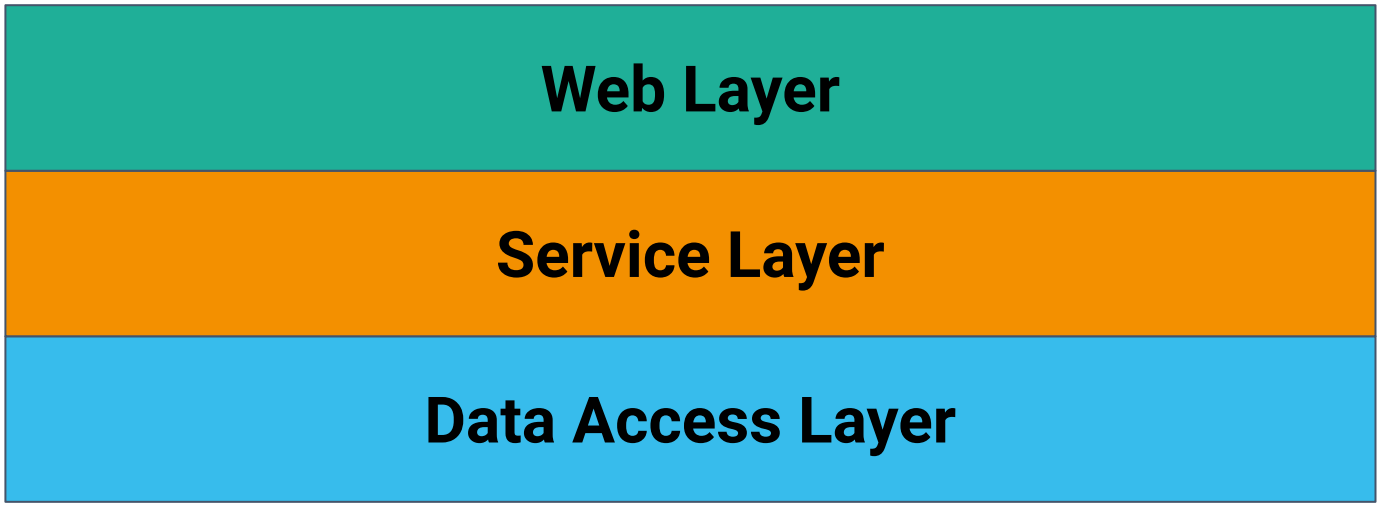


NOVATEC

Authorization - What can I access?

HTTP 403 - FORBIDDEN

Authorization Layers



Authorization on Web Layer

```
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests(
            authorizeRequests ->
                authorizeRequests
                    .mvcMatchers("/users", "/users/{userIdentifier}")
                    .hasRole("LIBRARY_ADMIN")
                    .anyRequest()
                    .authenticated())
    ...
}
```

Authorization on Service Layer (Static Roles)

```
public class UserBoundaryService {

    @PreAuthorize("hasRole('ADMIN')")
    public List<User> findAllUsers() {...}
}
```

Authorization on Service Layer (Dynamic Permissions)

```
public class TaskBoundaryService {

    @PreAuthorize("hasPermission(#taskId, 'TASK', 'WRITE')")
    public Task findTask(UUID taskId) {...}
}
```

Authorization on Data Access Layer

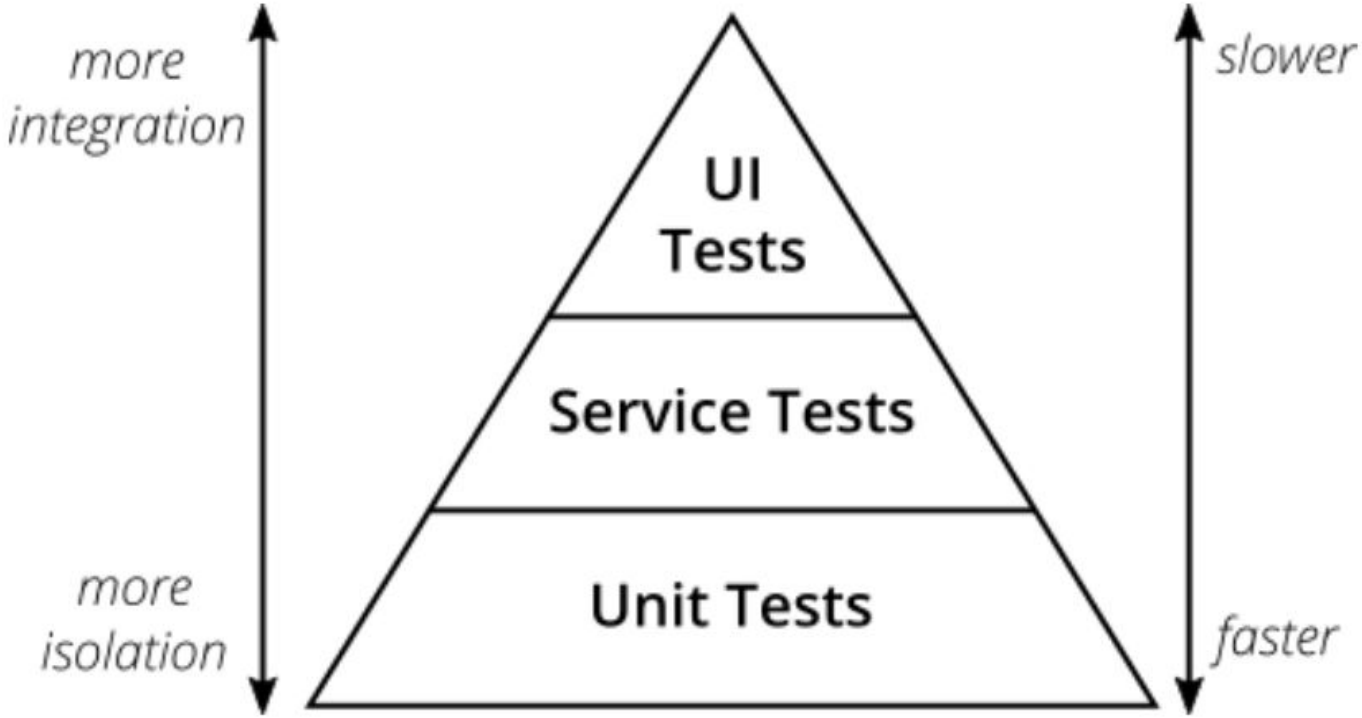
```
@Query(
"UPDATE AppUser u SET u.lastLogin=:lastLogin WHERE"
  + " u.username = ?#{ principal?.username }")
public void updateLastLogin (Date lastLogin);
```

Practice Time

Lab 4: Authorization

Security Testing

Testing Pyramid



Source: Succeeding with Agile (Mike Cohn)

Testing Done Right!

Reading Tip:

The Practical Test Pyramid (Martin Fowler)
https://martinfowler.com/articles/practical-test-pyramid.html

Source: Succeeding with Agile (Mike Cohn)

Automated Authorization Testing

```
public class AuthorizationIntegrationTest {

    @WithMockUser(roles = "ADMIN")
    @Test
    public void verifyFindAllUsersAuthorized() {...}

    @WithMockUser(roles = "USER")
    @Test(expected = AccessDeniedException.class)
    public void verifyFindAllUsersUnauthorized() {...}
}
```

Security Testing

- Security can be tested on EVERY level
  - Static Code Analysis (SAST)
  - Unit Test
  - Integration Test
  - Dynamic Testing (DAST), e.g. OWASP Zap
  - Explorative Testing (Security Charters)

Spring Security Testing Support

- Spring Security can support in
  - Testing Authentication
  - Testing Authorization
  - Testing setting Security Headers

Practice Time

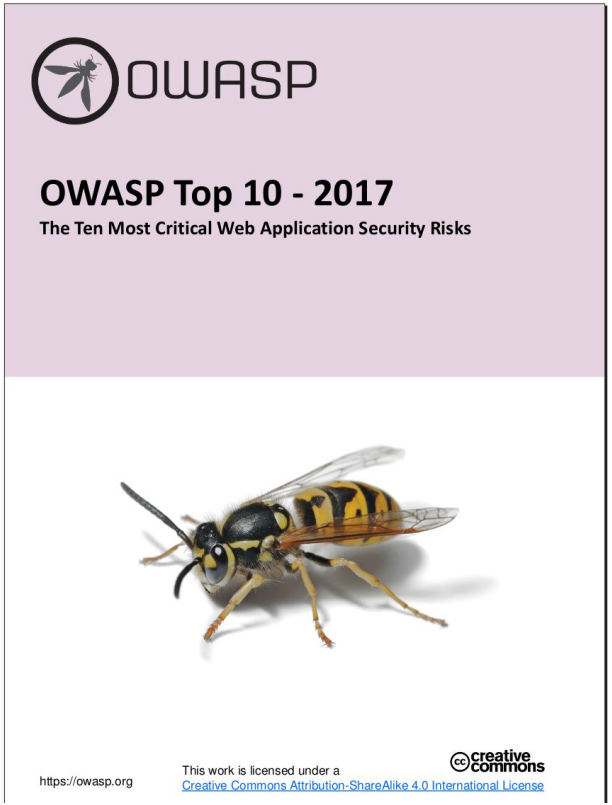
Lab 5: Security Testing

What about the Cloud?

“Good old friends” + more

- CSRF, XSS, SQL Injection, ...
- Distributed DoS
- Economic DoS

A3: Sensitive Data Exposure



NOVATEC

EU General Data Protection Regulation (GDPR)

“...should adopt internal policies and implement measures which meet in particular the principles of **data protection by design** and **data protection by default**”

http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679

Key Management

NOVATEC

Key Management

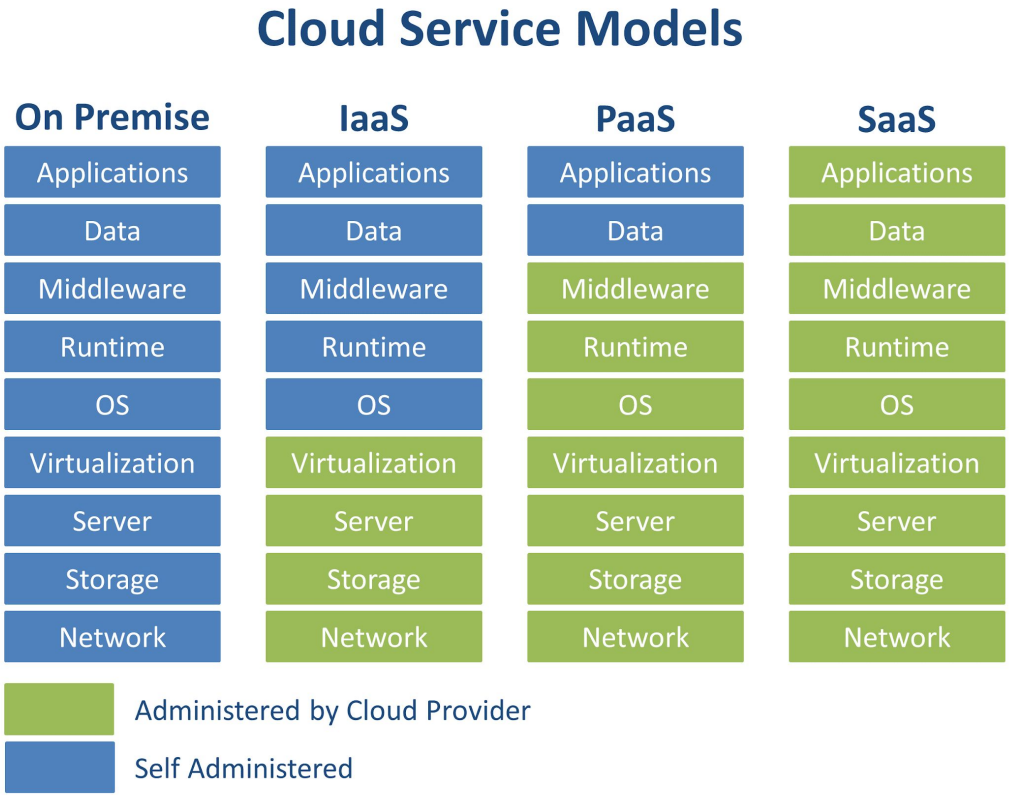
- Azure Key Vault
- AWS Secrets Manager
- Google Cloud HSM
- CloudFoundry CredHub
- Hashicorp Vault

Rotate, Repair, Repave

“What if every server inside my data center had a maximum lifetime of two hours? This approach would frustrate malware writers...”
Justin Smith (Chief Security Officer at Pivotal)

https://thenewstack.io/cloud-foundrys-approach-security-rotate-repair-repave

Cloud Service Models

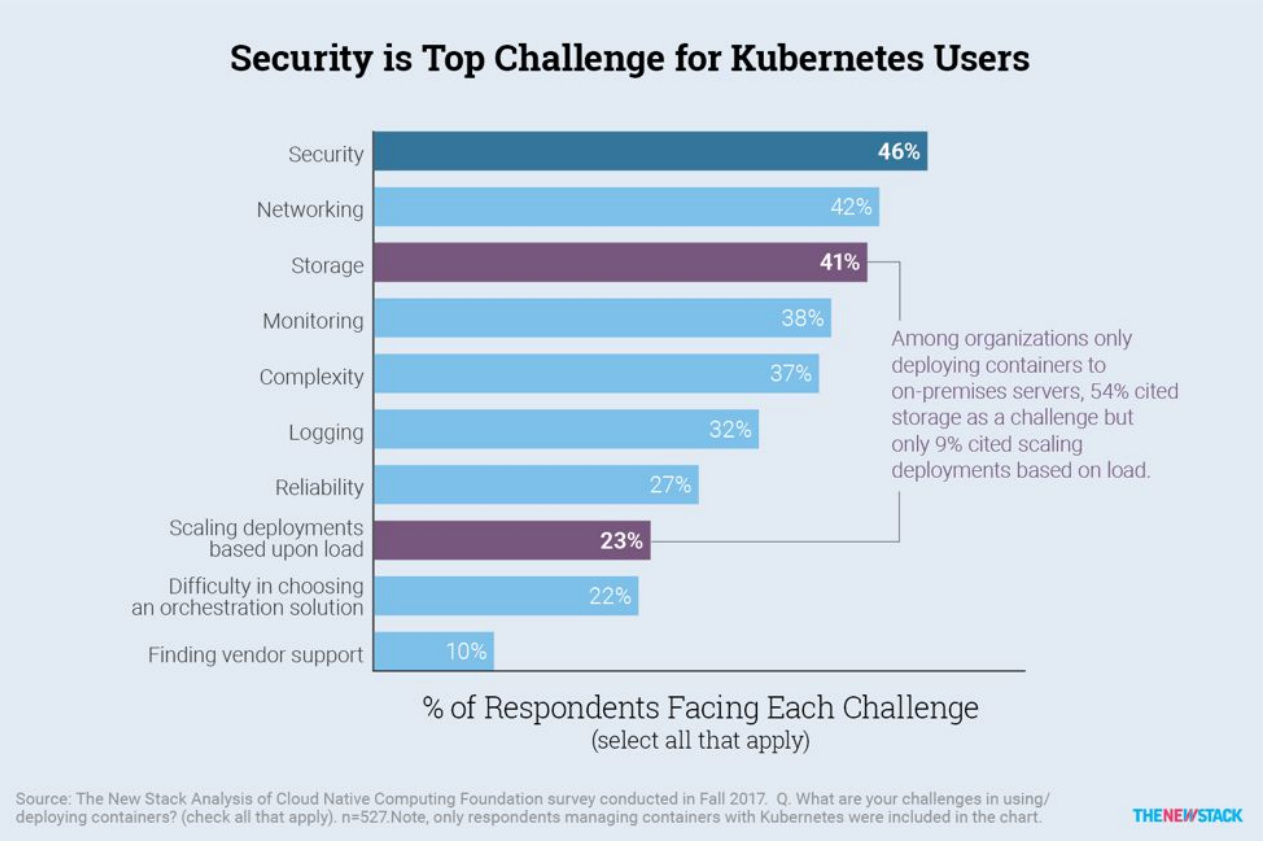


NOVATEC

Container & Kubernetes Security

Top Challenges in Kubernetes

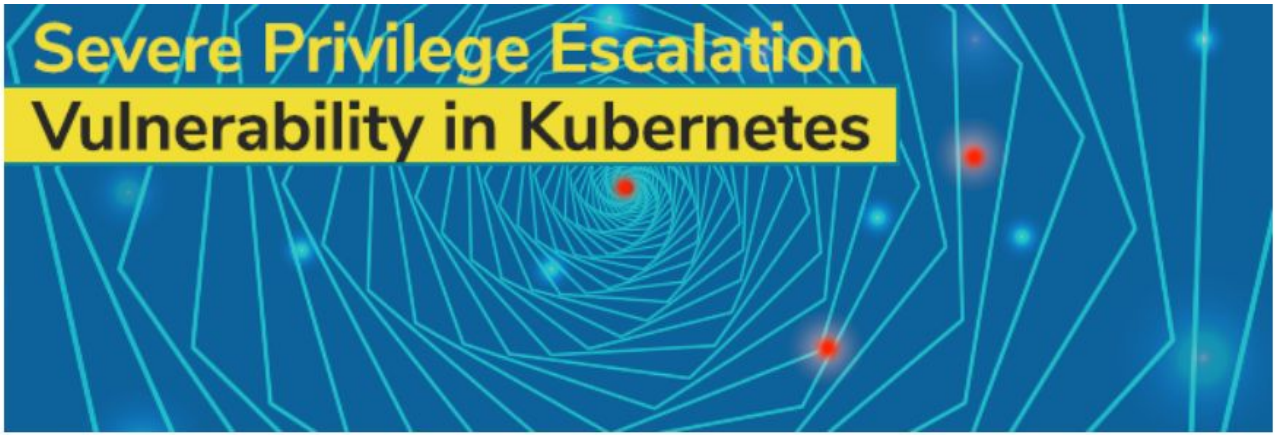
Source: https://thenewstack.io



NOVATEC

Severe Vulnerability in Kubernetes

Source: https://blog.aquasec.com



Ariel Shuper • December 06, 2018

Severe Privilege Escalation Vulnerability in Kubernetes (CVE-2018-1002105)

Earlier this week, a severe vulnerability in Kubernetes (CVE-2018-1002105) was disclosed that allows an unauthenticated user to perform privilege escalation and gain full admin privileges on a cluster. The CVE was given the high severity score of 9.8 (out of 10) and it affects all Kubernetes versions from 1.0 onwards, but fixes are available for recent versions.

Crypto Mining Via K8s Dashboard

Source: https://blog.heptio.com

On Securing the Kubernetes Dashboard

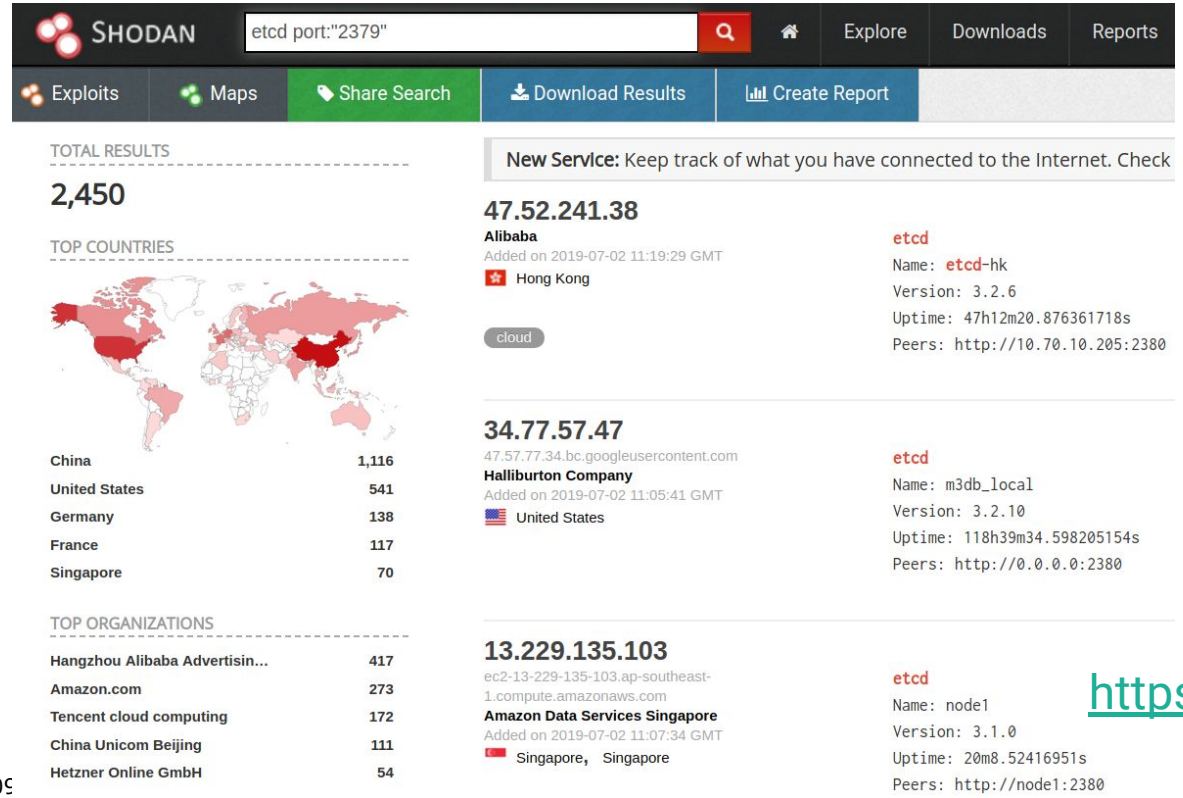
Joe Beda Follow

Feb 28, 2018 · 13 min read

Recently Tesla (the car company) was alerted, by security firm RedLock, that their Kubernetes infrastructure was compromised. The attackers were using Tesla's infrastructure resources to mine cryptocurrency. This type of attack has been called "cryptojacking".

The vector of attack in this case was a Kubernetes Dashboard that was exposed to the general internet with no authentication and elevated privileges. Not only this, but core AWS API keys and secrets were visible. How do you prevent this from happening to you?

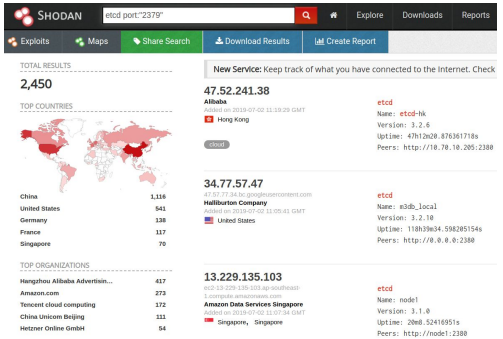
Open ETCD Ports in Kubernetes (1)



https://shodan.io

NOVATEC

Open ETCD Ports in Kubernetes (2)



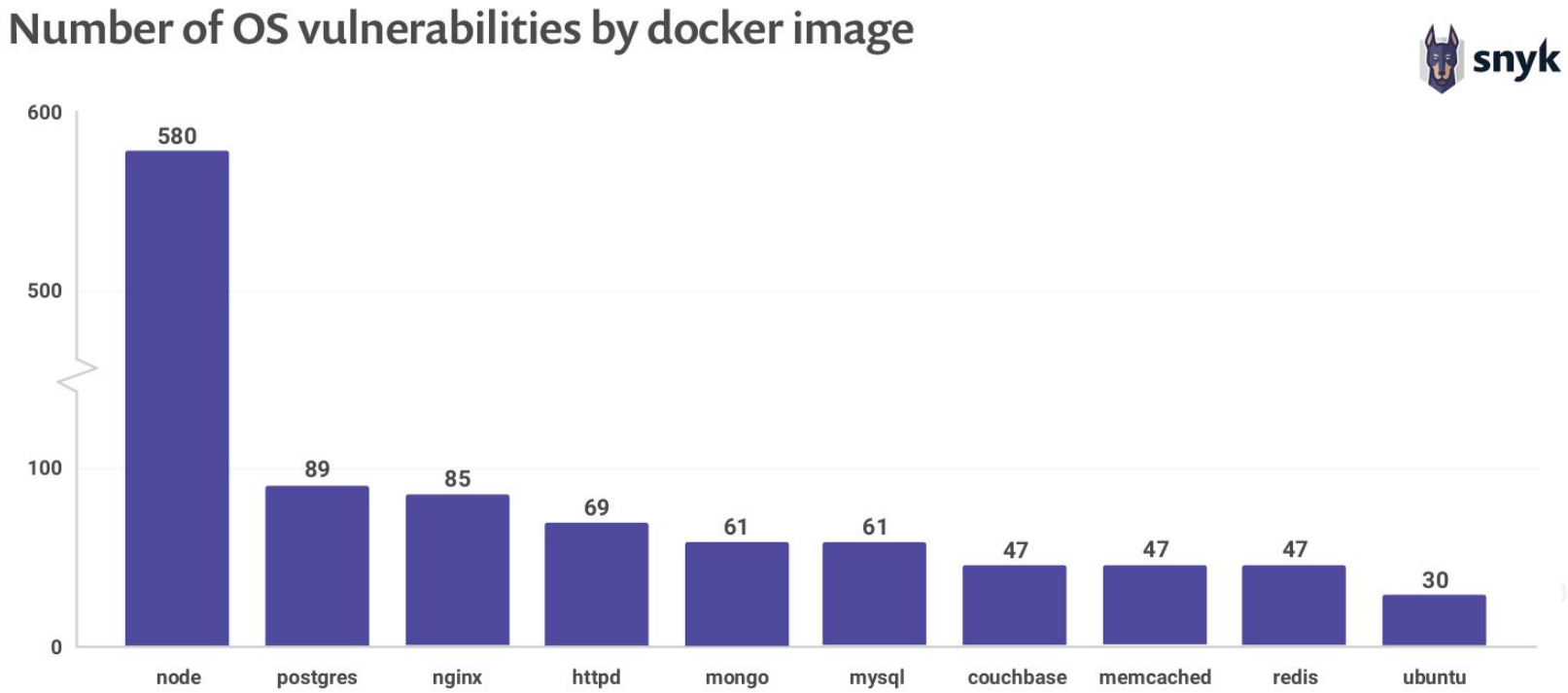

```
$ etcdctl --endpoints=http://xx.xx.xx.xx:2379
cluster-health

member b97ee4034db41d17 is healthy: got healthy
result
from http://xx.xx.xx.xx:2379
cluster is healthy
```

https://github.com/etcd-io/etcd/releases

NOVATEC

Vulnerable Docker Images

Source: The state of open source security report (snyk.io)



NOVATEC

All is Root

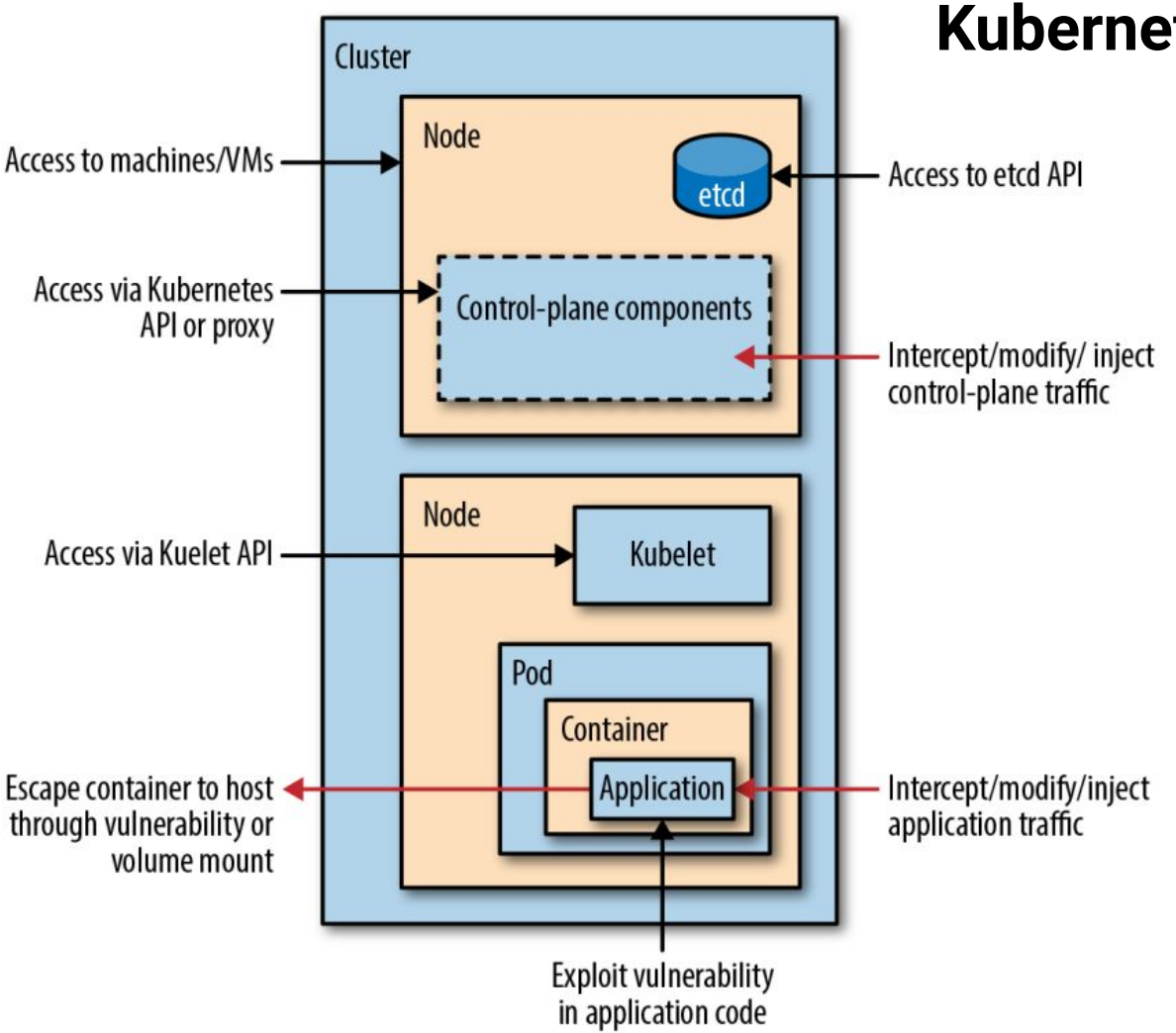
CZnative @ home
@pczarkowski

Welcome to Kubernetes where everything runs as root and the security doesn't matter!

14:22 - 8. Mai 2019

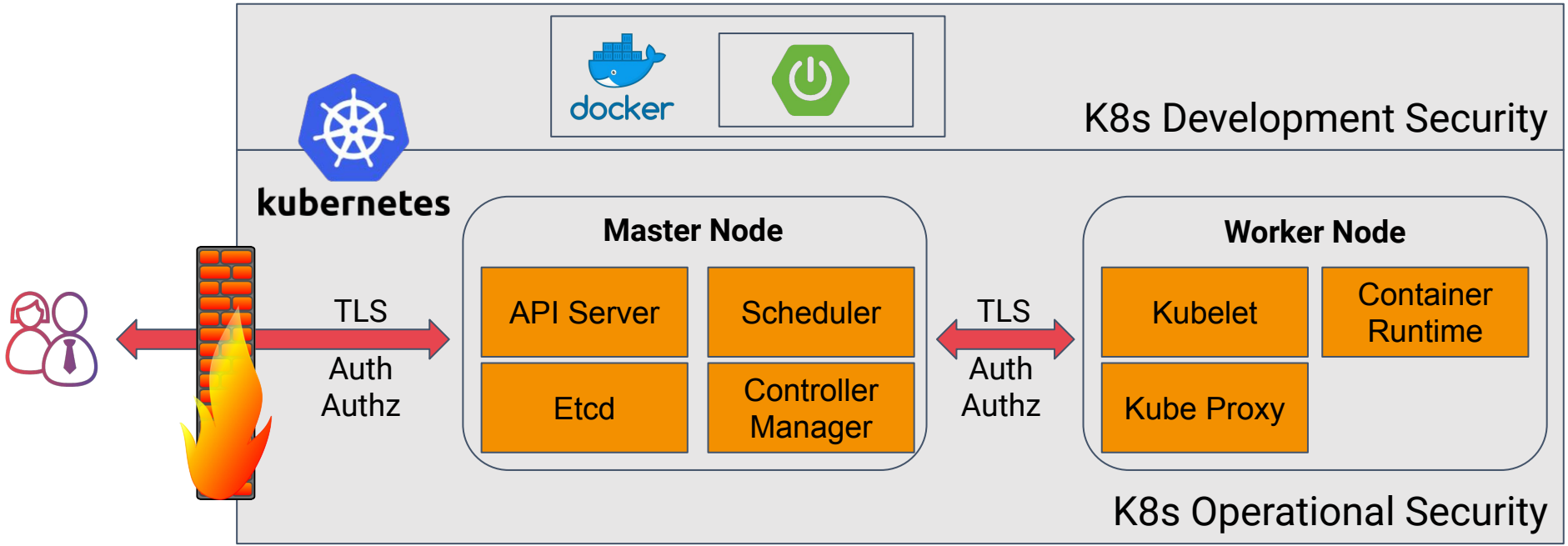
Kubernetes attack vectors

Source:
Kubernetes Security, O'Reilly, 2018



NOVATEC

Operational / Development Kubernetes Security



https://kubernetes.io/docs/concepts/security/overview/#the-4c-s-of-cloud-native-security
https://learnk8s.io/production-best-practices/

So what can we do as developers?

Application- / Docker- / K8s-Security

The Path for Secure Development on K8s

The Path for Secure Development on K8s

Application Security

Authentication

Authorization

SQL Injection

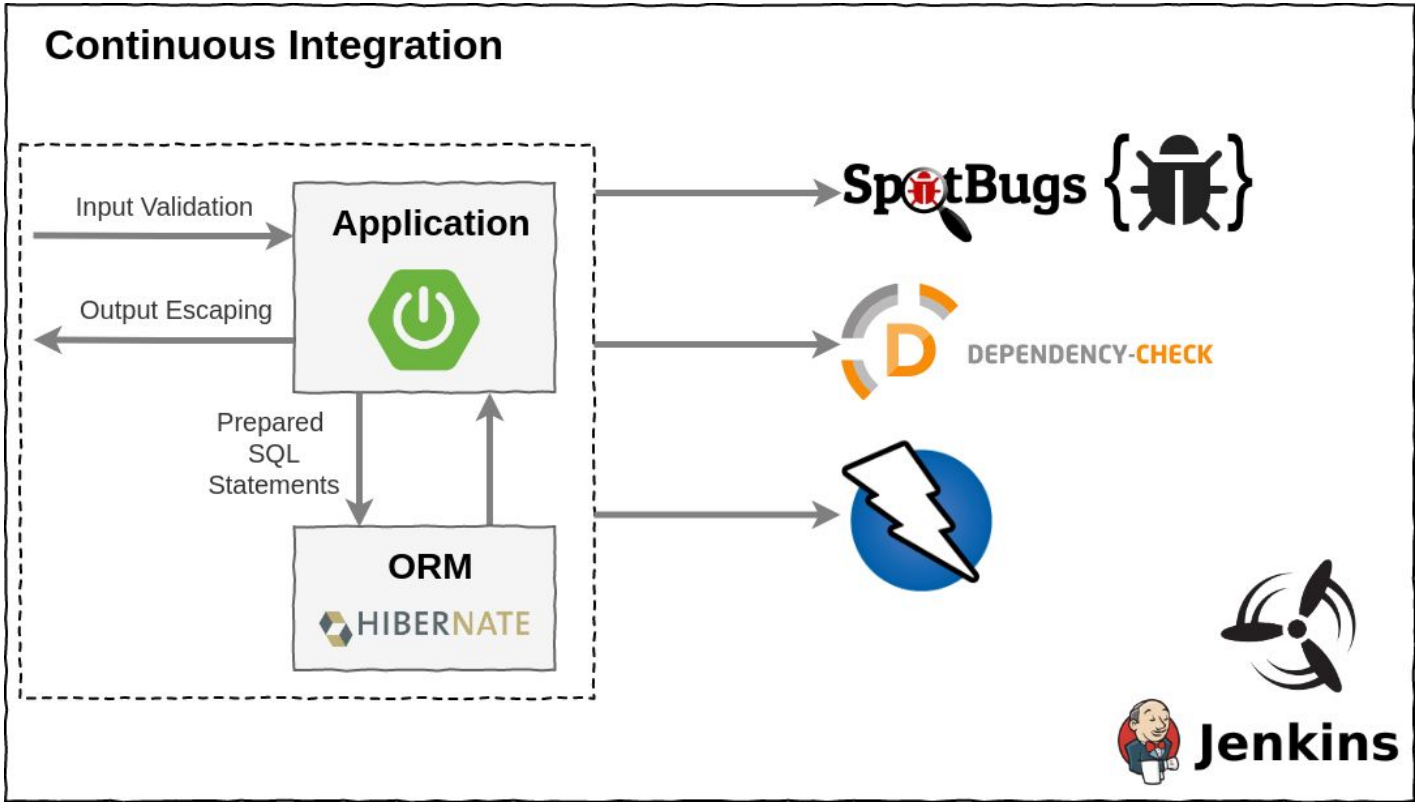
Cross Site Scripting (XSS)

Cross Site Request Forgery (CSRF)

Data Protection (Crypto)

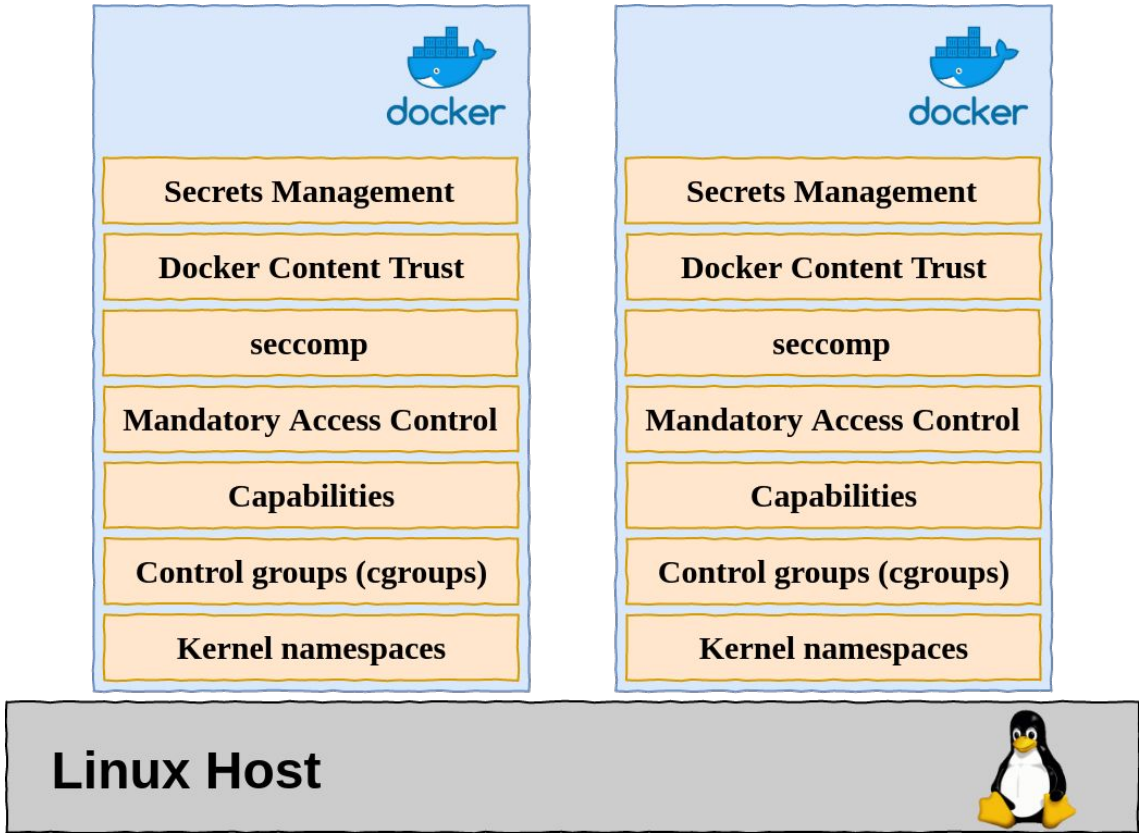
...

Application Security



The Path for Secure Development on K8s

Docker Security Basics



Linux Kernel Namespaces

- Process ID (pid)
- Network (net)
- Filesystem/mount (mnt)
- Inter-Process Communication (ipc)
- User (user)
- UTS (hostname)

Linux Control Groups (CGroups)

- Resource Limits
  - CPU
  - Memory
  - Devices
  - Processes
  - Network

For Java this only works with container aware JDK versions as of **OpenJDK 8u192** or above

Linux Capabilities

- Break up root privileges into smaller units
  - CAP_SYS_ADMIN
  - CAP_NET_ADMIN
  - CAP_NET_BIND_SERVICE
  - CAP_CHOWN
  - ...

```
$ docker run --cap-drop=ALL --cap-add=NET_BIND_SERVICE
```

http://man7.org/linux/man-pages/man7/capabilities.7.html

Mandatory Access Control (MAC)

- AppArmor
- Security Enhanced Linux (SELinux)

https://gitlab.com/apparmor/apparmor/wikis/home
https://github.com/SELinuxProject

Secure Computing Mode (SecComp)

- Deny critical system calls by default
  - reboot
  - mount
  - swapon
  - ...

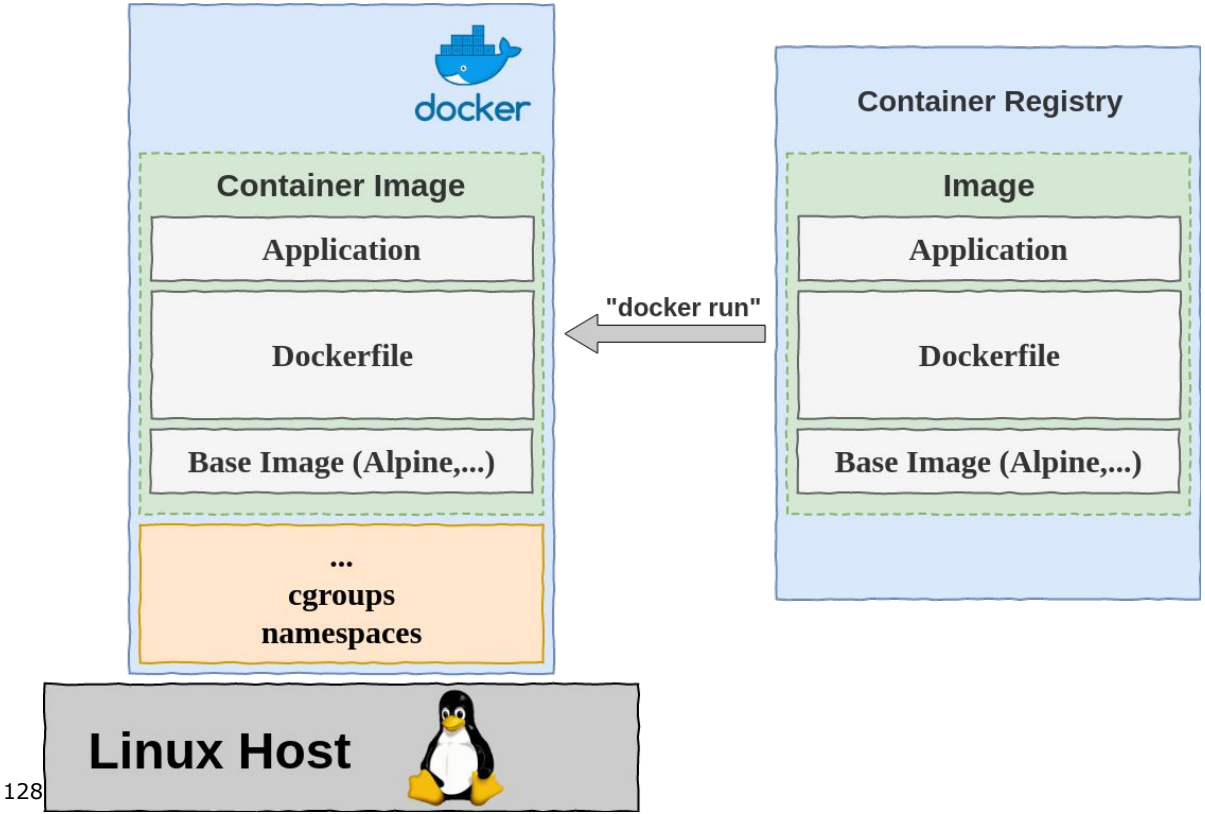
http://man7.org/linux/man-pages/man2/seccomp.2.html
https://docs.docker.com/engine/security/seccomp

OWASP Docker Top 10

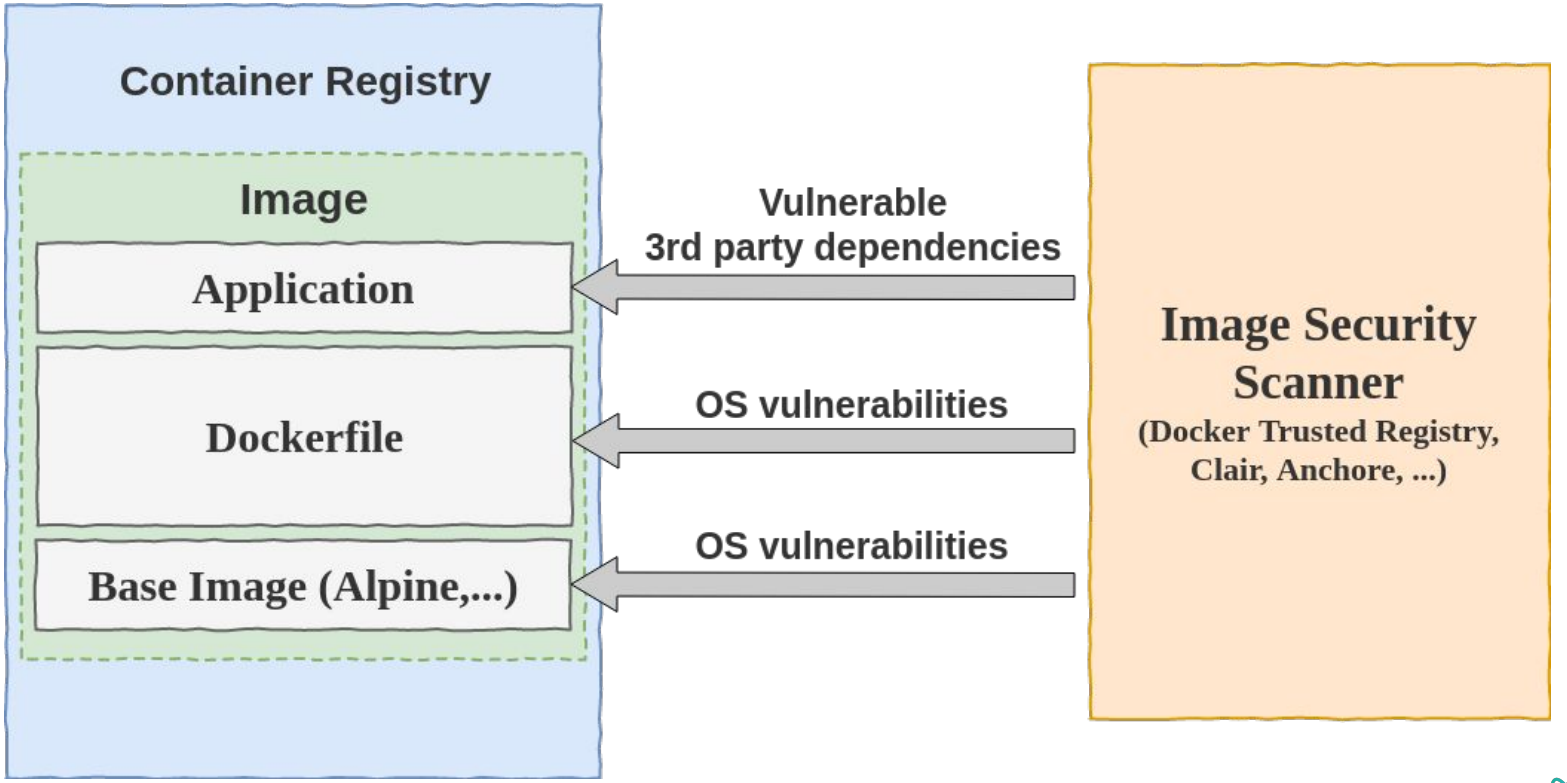
1. Secure User Mapping
2. Patch Management Strategy
3. Network Segmentation and Firewalling
4. Secure Defaults and Hardening
5. Maintain Security Contexts
6. Protect Secrets
7. Resource Protection
8. Container Image Integrity and Origin
9. Follow Immutable Paradigm
10. Logging

https://github.com/OWASP/Docker-Security

Docker Images



Docker Image Security



Say No To Root!

USER directive in Dockerfile

```
FROM openjdk:11-jre-slim
COPY hello-spring-kubernetes-1.0.0-SNAPSHOT.jar app.jar
EXPOSE 8080
RUN addgroup --system --gid 1002 app && adduser
    --system --uid 1002 --gid 1002 appuser
USER 1002
ENTRYPOINT java -jar /app.jar
```

https://opensource.com/article/18/3/just-say-no-root-containers

Say No To Root!

Use JIB and Distroless Images

```
plugins {
  id 'com.google.cloud.tools.jib' version '...'
}

jib {
 container {
   user = 1002
 }
}
```

https://github.com/GoogleContainerTools/jib

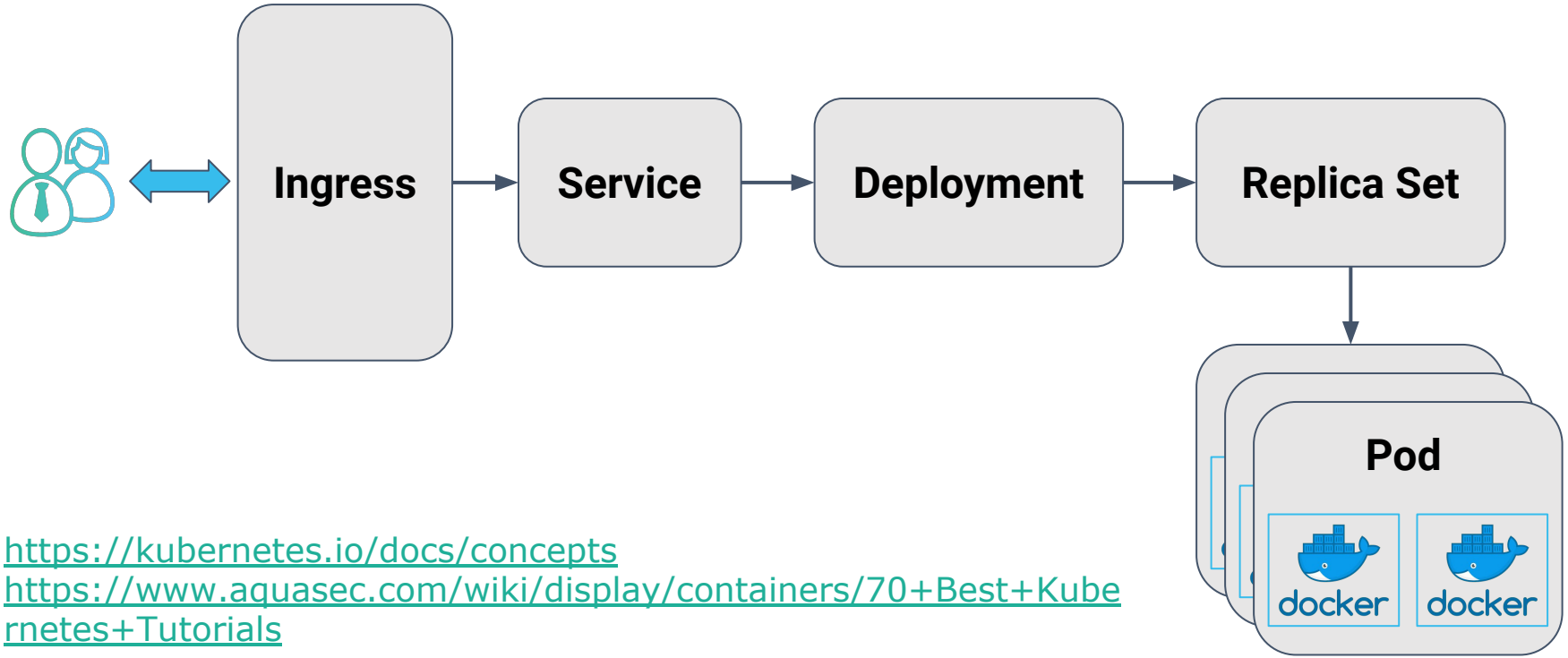
Keep Being Secure

- Perform Image Scanning
  - Anchore
  - Clair
  - Trivy
- Regularly Update Base Images

https://anchore.com/opensource/
https://github.com/coreos/clair
https://github.com/aquasecurity/trivy

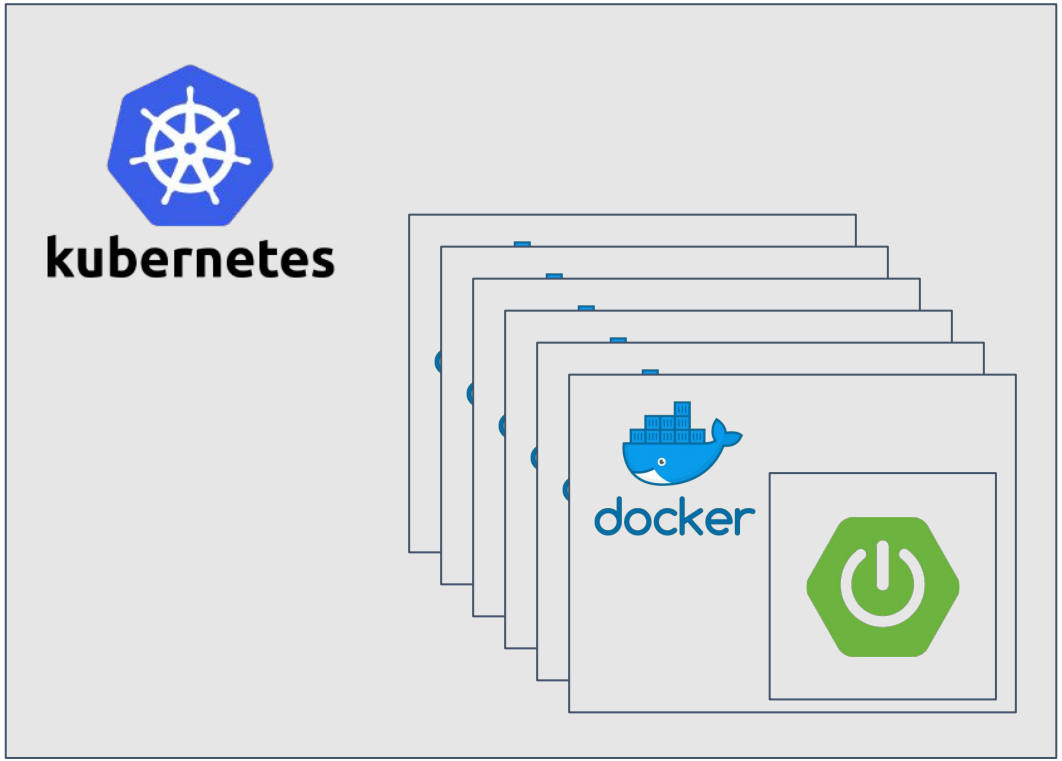
The Path for Secure Development on K8s

Kubernetes Basics



https://kubernetes.io/docs/concepts
https://www.aquasec.com/wiki/display/containers/70+Best+Kubernetes+Tutorials

Kubernetes Security



- Kubernetes Auditing
- Network Policies
- Role Based Access Control (RBAC)
- Resource Limits
- Pod Security Context
- Pod Security Policy

Resource Limits

```
spec:
  ...
  containers:
    resources:
      limits:
        cpu: "1"
        memory: "512Mi"
      requests:
        cpu: 500m
        memory: "256Mi"
  ...
```

https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource
https://kubernetes.io/docs/tasks/configure-pod-container/assign-memory-resource

Pod/Container Security Context

```
spec:
  securityContext:
    runAsNonRoot: true
  containers:
    securityContext:
      allowPrivilegeEscalation: false
      privileged: false
      runAsNonRoot: true
      readOnlyRootFilesystem: true
      capabilities:
        drop:
          - ALL
```

https://kubernetes.io/docs/tasks/configure-pod-container/security-context

Pod Security Policy (Still In Beta!)

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: no-root-policy
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  runAsUser:
    rule: 'MustRunAsNonRoot'
  ...
```

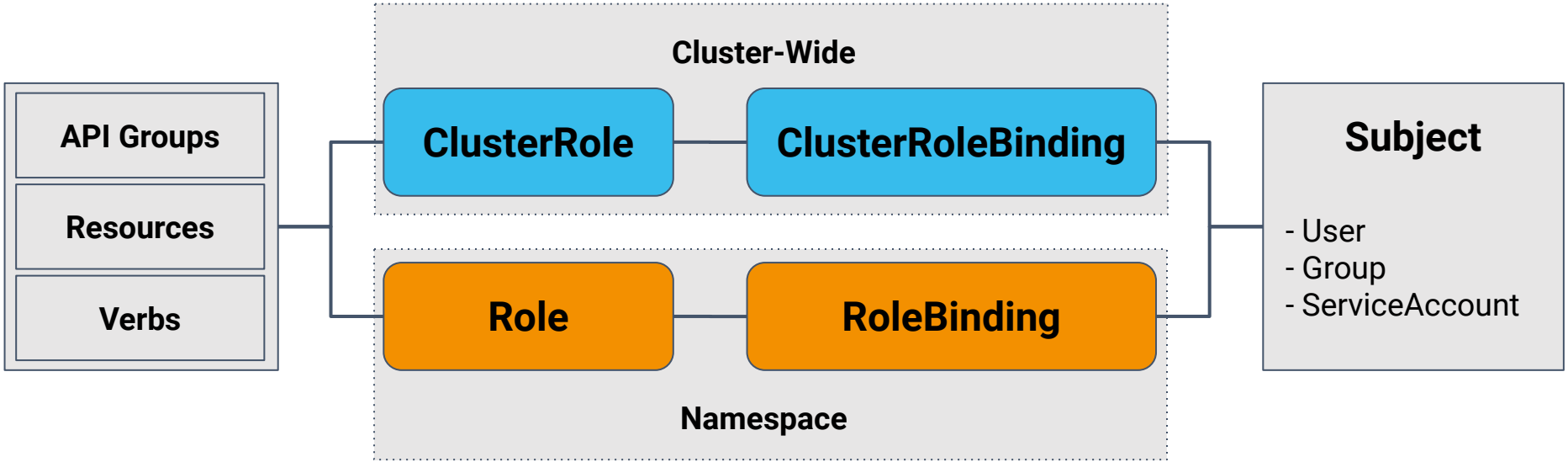
https://kubernetes.io/docs/concepts/policy/pod-security-policy

Pod Security Policy (Policy Order)

```
Policy order selection criteria:

1.  Policies which allow the pod as-is are preferred
2.  If pod must be defaulted or mutated, the first
    policy (ordered by name) to allow the pod is
    selected.
```

https://kubernetes.io/docs/concepts/policy/pod-security-policy/#policy-order
https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers

Kubernetes Role Based Access Control (RBAC)



https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Kubernetes Role Based Access Control (RBAC)

apiGroups	extensions, apps, policy, ...
resources	pods, deployments, configmaps, secrets, nodes, services, endpoints, podsecuritypolicies, ...
verbs	get, list, watch, create, update, patch, delete, use, ...

https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Service Account

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: deploy-pod-security-policy
  namespace: default
```

https://kubernetes.io/docs/concepts/policy/pod-security-policy/#authorizing-policies

Pod Security Policy Role

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: no-root-policy-role
  namespace: default
rules:
  - apiGroups: ['policy']
    resources: ['podsecuritypolicies']
    verbs:     ['use']
    resourceNames:
      - no-root-policy
```

https://kubernetes.io/docs/concepts/policy/pod-security-policy/#authorizing-policies

Pod Security Policy Role Binding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: deploy-pod-security-policy
  namespace: default
roleRef:
  kind: Role
  name: no-root-policy-role
  apiGroup: rbac.authorization.k8s.io
subjects:
  - kind: ServiceAccount
    name: deploy-pod-security-policy
    namespace: default
```

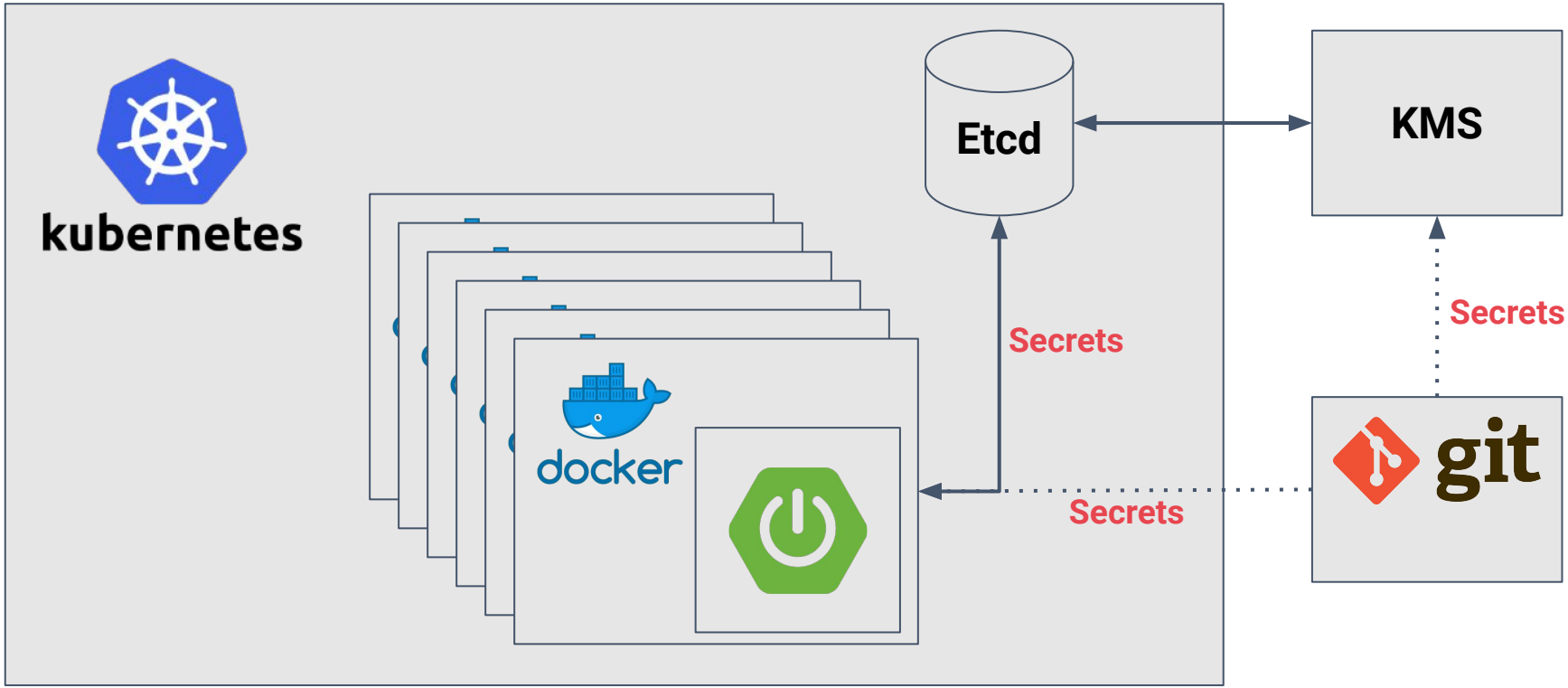
Helm 3 Is Here! 😀

https://v3.helm.sh
https://helm.sh/docs/faq/#removal-of-tiller

Lab 6: Container/K8s Security

The Path for Secure Development on K8s

Kubernetes Secrets



Kubernetes Secrets

```
apiVersion: v1
kind: Secret
metadata:
  name: hello-spring-cloud-kubernetes
  namespace: default
type: Opaque
data:
  user.username: dXNlcg==
  user.password: azhzX3VzZXI=
  admin.username: YWRtaW4=
  admin.password: azhzX2FkbWlu
```

https://kubernetes.io/docs/concepts/configuration/secret

Kubernetes Secrets - Best Practices

- Encrypt Secret Data at Rest
  Only Base64 Encoded by Default!
- Applications interacting with secrets API should be limited using RBAC
- Mount secrets instead of ENV Mapping

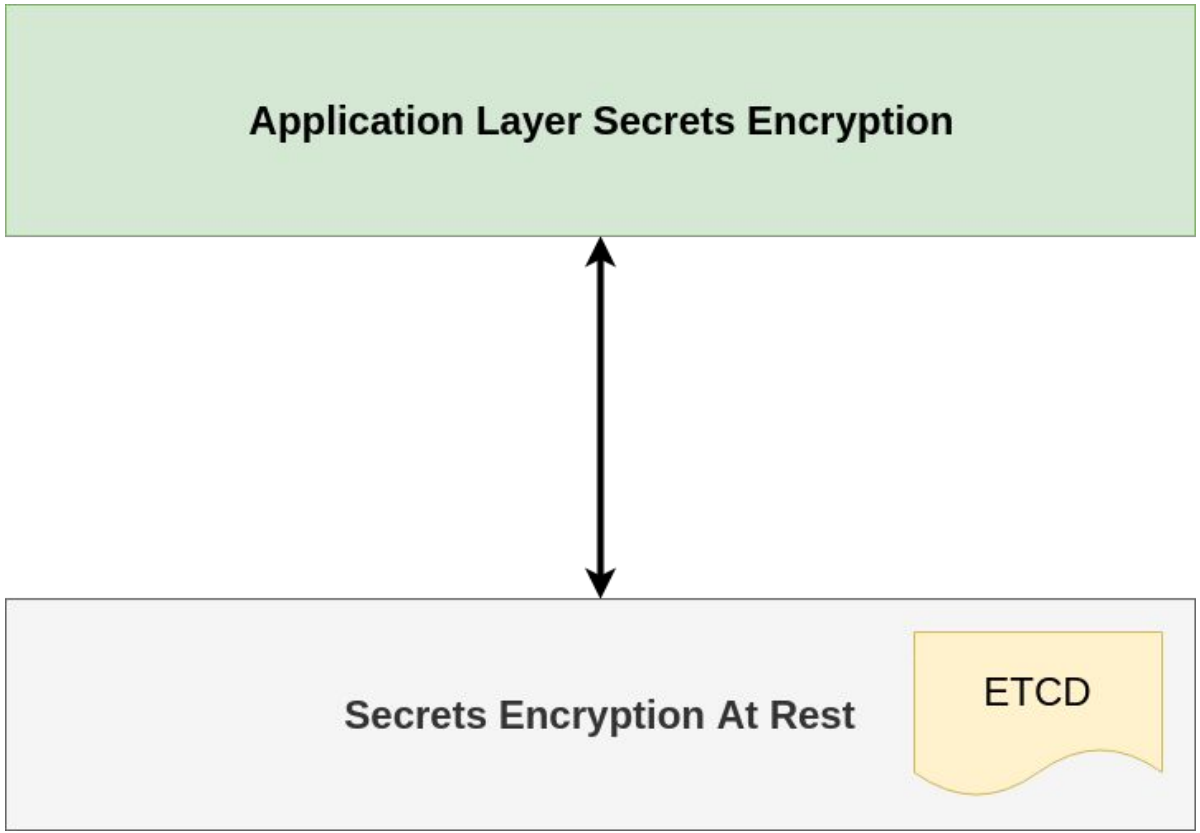
https://kubernetes.io/docs/concepts/configuration/secret/#best-practices
https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data

Pay Attention to Spring Boot Actuator

```
{
      "name": "applicationConfig: ...",
      "properties": {
        "greet.my-sec": {
          "value": "geheim",
          "origin": "class path resource ..."
        },
        "greet.password": {
          "value": "******",
          "origin": "class path resource ..."
        }
}
```

http://localhost:8080/actuator/env

Encryption Layers



Envelope Encryption On Kubernetes

https://cloud.google.com/kms/docs/envelope-encryption
https://kubernetes.io/docs/tasks/administer-cluster/kms-provider

Key Management System (KMS) Cloud Providers

- Azure Key Vault (Key Vault FlexVolume)
- Google Cloud KMS
- AWS KMS
- ...

https://github.com/Azure/kubernetes-kms
https://github.com/Azure/kubernetes-keyvault-flexvol
https://cloud.google.com/kms
https://aws.amazon.com/de/kms

What about Secrets in 

- Sealed Secrets
- Helm Secrets
- Kamus
- Sops
- Hashicorp Vault

https://learnk8s.io/kubernetes-secrets-in-git
https://github.com/bitnami-labs/sealed-secrets
https://github.com/futuresimple/helm-secrets
https://github.com/Soluto/kamus
https://github.com/mozilla/sops
https://www.vaultproject.io

Summary

Summary / Key Insights

- Containers use Linux Namespaces+Caps
- Say NO to root on K8s
- “Least privilege” for service accounts
- Keep K8s up-to-date and scan for security
- Ensure your secrets are encrypted in K8s
- Keep K8s and container images up-to-date

Andreas Falk
Managing Consultant

Mobil: +49 151 46146778
E-Mail: andreas.falk@novatec-gmbh.de

Novatec Consulting GmbH
Dieselstraße 18/1
D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700
info@novatec-gmbh.de
www.novatec-gmbh.de