OpenShift-Anwendertreffen - Breakout-Session „Security“

Dominik Pataky, Yannic Ahrens // openbastei.de

2019-11-19

Was bedeutet Security für uns?

- *Welche Stichworte verbinden Sie mit „Security“?*

Was bedeutet Security für uns?

- *Welche Stichworte verbinden Sie mit „Security“?*
- *Welches Interesse verfolgen wir alle, wenn von „Security“ gesprochen wird?*

Was bedeutet Security für uns?

- *Welche Stichworte verbinden Sie mit „Security“?*
- *Welches Interesse verfolgen wir alle, wenn von „Security“ gesprochen wird?*
- *Wie präsent ist das Thema in Ihrer Organisation?*

Security im Kontext von OpenShift

- *Was versprechen Sie sich von OpenShift zum Thema Security?*

Security im Kontext von OpenShift

- *Was versprechen Sie sich von OpenShift zum Thema Security?*
- *Sind die Security-Features von OpenShift Auslöser für eine Migration, weg von einem vorherigen Software-Stack?*

Security im Kontext von OpenShift

- *Was versprechen Sie sich von OpenShift zum Thema Security?*
- *Sind die Security-Features von OpenShift Auslöser für eine Migration, weg von einem vorherigen Software-Stack?*
- *Erfahrungswerte: Umsetzung von Security? Vorfälle? Besondere Erlebnisse?*

Schutzziele in der Informationssicherheit

- Security ist ein Prozess. Die Ziele müssen klar sein.
- Schutzziele, mit denen wir arbeiten:
  - Vertraulichkeit (Confidentiality)
  - Integrität (Integrity)
  - Verfügbarkeit (Availability)
  - Authentizität (Authenticity)
  - Autorisierung (Authorisation)

Vertraulichkeit

- „Wer darf Inhalte und Kommunikationsbeziehungen sehen?“
- Ziel: Informationen nur für berechtigten Empfangskreis einsehbar machen
- Methoden
  - Verschlüsselung von Daten
  - Verschleierung von Kommunikation
  - Schutz vor Seitenkanälen

Integrität

- „Sind Änderungen an meinen Daten nachvollziehbar?“
- Ziel: Modifikationen an Daten, gewollt oder ungewollt, überprüfbar machen
- Methoden
  - Prüfsummen und Hashwerte für Daten
  - Nummerierung von Paketen zur Erkennung von Verlusten
  - Container: immutable Image-Layer

Verfügbarkeit

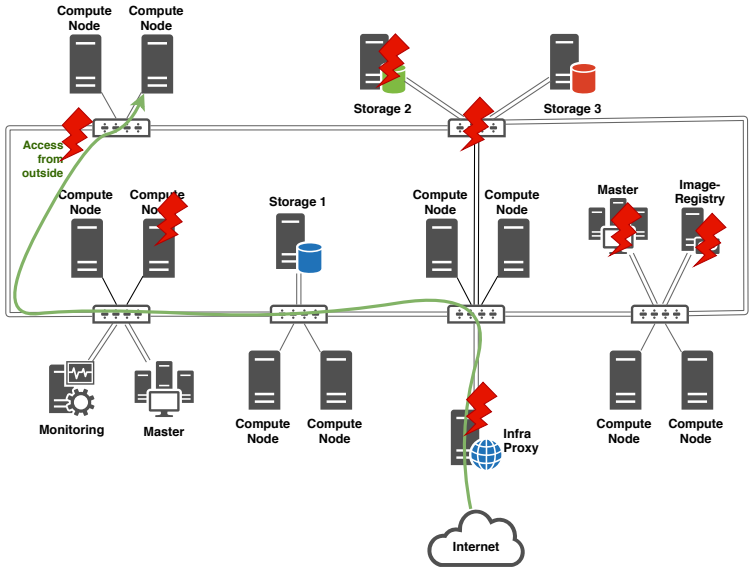
- „Sind meine Systeme, wenn ich sie brauche, auf jeden Fall verfügbar?“
- Ziel: Sicherstellen, dass Systeme zuverlässig laufen und bei Ausfall Ersatz bereit steht
- Methoden
  - Redundanz und Autoscaling
  - Monitoring
  - Load-Balancing und Lastverteilung

Authentifizierung

- „Ist mein Gegenüber der, der er vorgibt zu sein?“
- Ziel: (kryptographischer) Beweis der Identität eines Kommunikationspartners
- Methoden
  - Multi-Faktor-Authentifizierung: Wissen, Haben, Sein
  - Zertifikate (z. B. HTTPS zwischen Server und Clients), API-Token
  - Identitätsprovider als Vermittler

Autorisierung

- „Darf der Account eine Aktion durchführen?“
- Ziel: Entscheiden, ob ein Befehl/API-Call/Programm ausgeführt wird
- Methoden
  - Access Control Lists (ACL).
    Beispiele: Dateiberechtigungen, Paketfilter
  - Rollenbasiertes Rechtesystem (RBAC)
    „Hat Account die Rolle ’Sysadmin’?“
  - Attributbasiertes Rechtesystem (ABAC)
    „Hat Account das Attribut ’Top-Secret’?“


Compute Node
Compute Node
Storage 2
Storage 3
Access from outside
Compute Node
Compute Node
Storage 1
Compute Node
Compute Node
Master
Image-Registry
Monitoring
Master
Compute Node
Compute Node
Infra Proxy
Compute Node
Compute Node
Internet

Umsetzung der Schutzziele in OpenShift

- Von der Theorie jetzt in die Praxis:
  welche Features eignen sich für welche Schutzziele?
- Beispiele zufällig ausgewählt, nicht vollständige Liste
- *Welche Schutzmaßnahmen kennen Sie?*

Vertraulichkeit: HTTPS, IPsec, Secure Routes

- Verbindungen zum Master-API werden mit HTTPS gesichert (Server- und Client-Zertifikate)
- Zwischen Nodes sind IPsec-Tunnel Best Practice (in OKD, z. B. für Overlay-Netzwerk)
- Traffic von Proxy zur App: Secure Routes mit TLS

Integrität: Container, Images, HTTPS

- Container erlauben nur Zugriff auf eigene Dateien.
  Umgesetzt mit Docker, podman und Kubernetes
- Zertifizierte Images mit Hashes und Signatur.
  Beispiel: Docker Hub, quay.io, interner Image Registry Operator
- Calls zum Master-API nur über HTTPS-Kanal
  mit internen Zertifikaten

Verfügbarkeit: Scheduling, Monitoring, Scaling, Failover

- Scheduler platziert Pods an geeigneter Stelle
- Health-Checks vergleichen dauerhaft Ist- mit Soll-Zustand (Kubernetes reconciliation loop)
- Rechtzeitige Bereitstellung von Ressourcen durch Autoscaling
- Failover durch abstrahierte Services

Authentifizierung: Tokens, Identity-Provider

- Web-UI: Login via Passwort oder OAuth
- User und Service Accounts (Bots) eindeutig identifizierbar durch API-Tokens

```
$ oc policy add-role-to-user admin system:serviceaccount:mypr:bot
$ oc serviceaccounts get-token bot
eyJhbGciOiJSUzI1NiIsInR5c...
$ curl -H ''Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5c...'' ...
```

- Einbindung externer Identitätsprovider
  OpenID, LDAP, Google/GitHub/GitLab/Keystone

Autorisierung: Roles, RBAC, SCCs, NetworkPolicies

- Zuordnung von Accounts zu Rollen (Bindings)
  `oc adm policy add-role-to-user testqa anton`
- Rollenbasiertes Rechtesystem mit Regeln (Rules),
  Cluster- oder Projekt-beschränkt (Cluster-weit oder in Namespace)
  `$ oc adm policy who-can get pods`
- Kontrolle von Pods mit Security Context Constraints (SCCs)
  Beispiel: priviligierte Pods, SELinux, `capabilities: add: ["NET_ADMIN"]`
- Netzwerk-Regeln: isolierte Overlays, feingranulare NetworkPolicys,
  Ingress Controllers

Perspektive & Innovation

- Bedeutung von Security wächst stetig (Leaks, Hacks, Compliance)
- Integration von weiteren Security-Features in OpenShift absehbar
- Breites Spektrum an Zusatzmodulen (Addons) für OpenShift bzw. Kubernetes zur Erweiterung von Schutzmaßnahmen …
- Werbeblock: OpenBastei-Plattform = „Kubernetes: more secure, by design“

Fragen, Erfahrungen, Anmerkungen

- Inhaltliche Fragen zum Thema?
- Austausch zu Erfahrungen mit Features
- Generelle Anmerkungen? Feedback zur Präsentation?

Stichworte aus der ersten Runde

- Wie kann man Leute überzeugen, dass Software auch die gleiche Isolation wie Hardware leisten kann? Viel Wissen basiert noch auf „Hardware-Welt“
- Bedenken von (externen) Auditoren sollte ernstgenommen werden, Übersetzung von Anforderungen an Technologie nötig. Rolle des BSI?
- Frage bzgl. Update-Policy in Containern, „Überblick verloren durch tausende Abhängigkeiten“
- OpenShift dient als sichere Grundlage, ist aber auch (Security-)Baukasten
- Neues Problem durch geteilte Maschinen: Hosts können ggf. nicht gepatcht werden, wenn kritische Systeme auf der gleichen Node laufen

Stichworte aus der zweiten Runde

- Applikationen werden von mehr und mehr Herstellern als Docker-Image verschickt, Paket-Verwaltungen haben Nachteil. Wer liefert Updates für das Base-Image?
- Problem daraus: „Irgendwo kommt ein Container her“. Deployment auf Containerplattform weckt auch falsche Erwartungshaltung, das „System ist ja sicher, kann nichts passieren“
- „Generationenkonflikt“: Sysadmins mit überschaubaren Packages aus vertrauter Quelle versus (plakatives Beispiel) npm/nodejs mit tausenden, ungetesteten Abhängigkeiten von Github
- Verweis auf den TrailOfBits Kubernetes-Audit