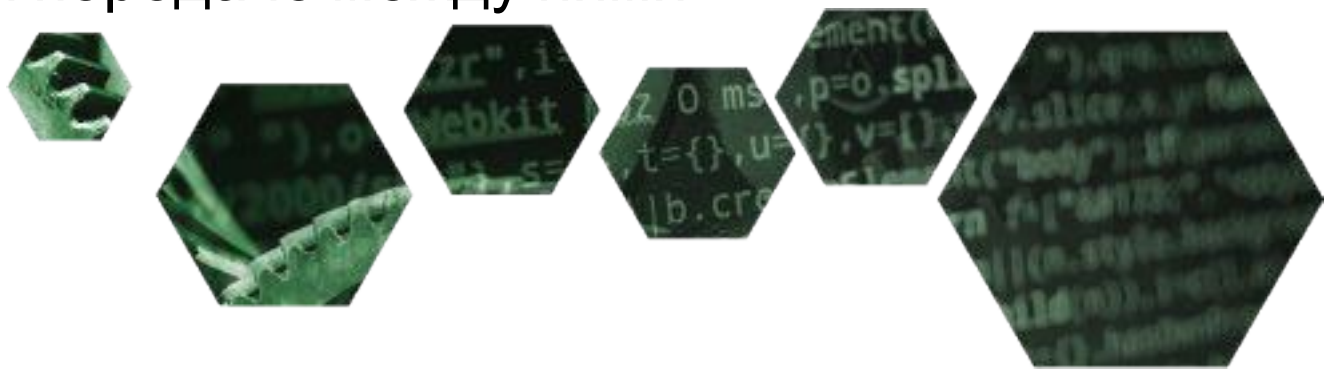
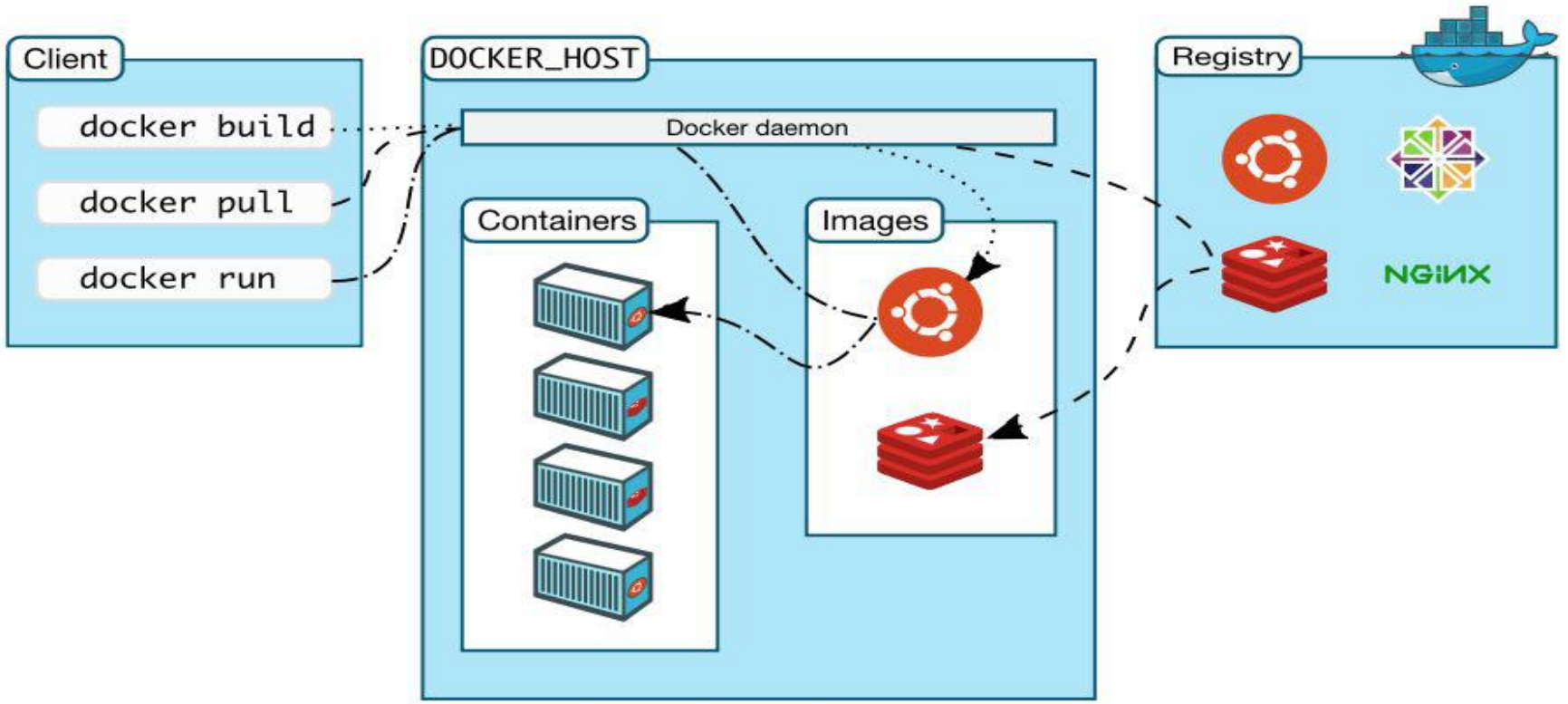
Безопасность в среде docker kublr k8s

Безопасность в среде kublr docker k8s

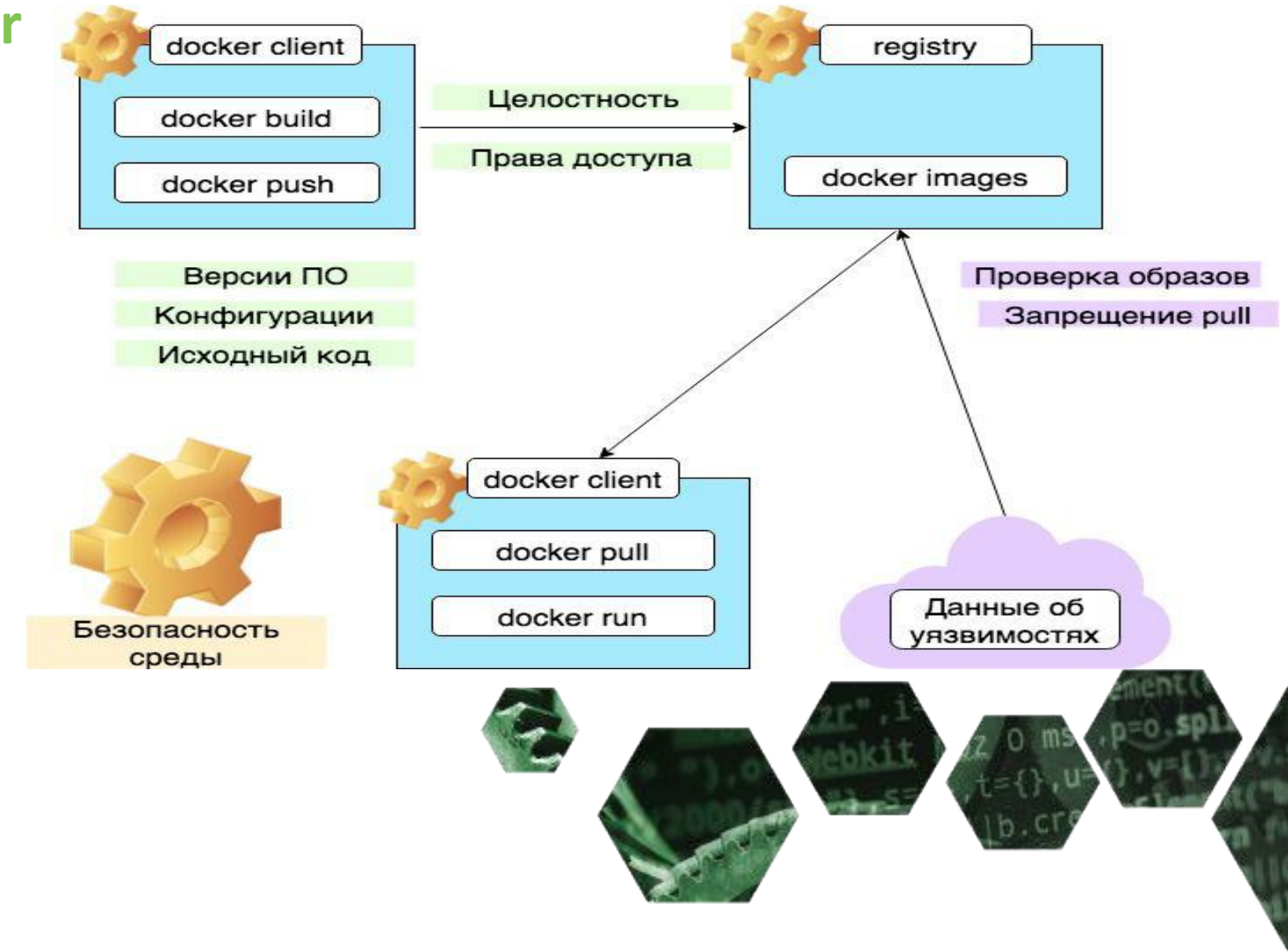
- Оценка безопасности распределенной системы
  - оценка безопасности каждого компонента
  - самый незащищенный компонент определяет общий уровень
- Обеспечение безопасности - это:
  - защита компонент
  - защита данных при передаче между ними



Клиент-серверная архитектура docker



Безопасность клиент-серверной архитектуры docker

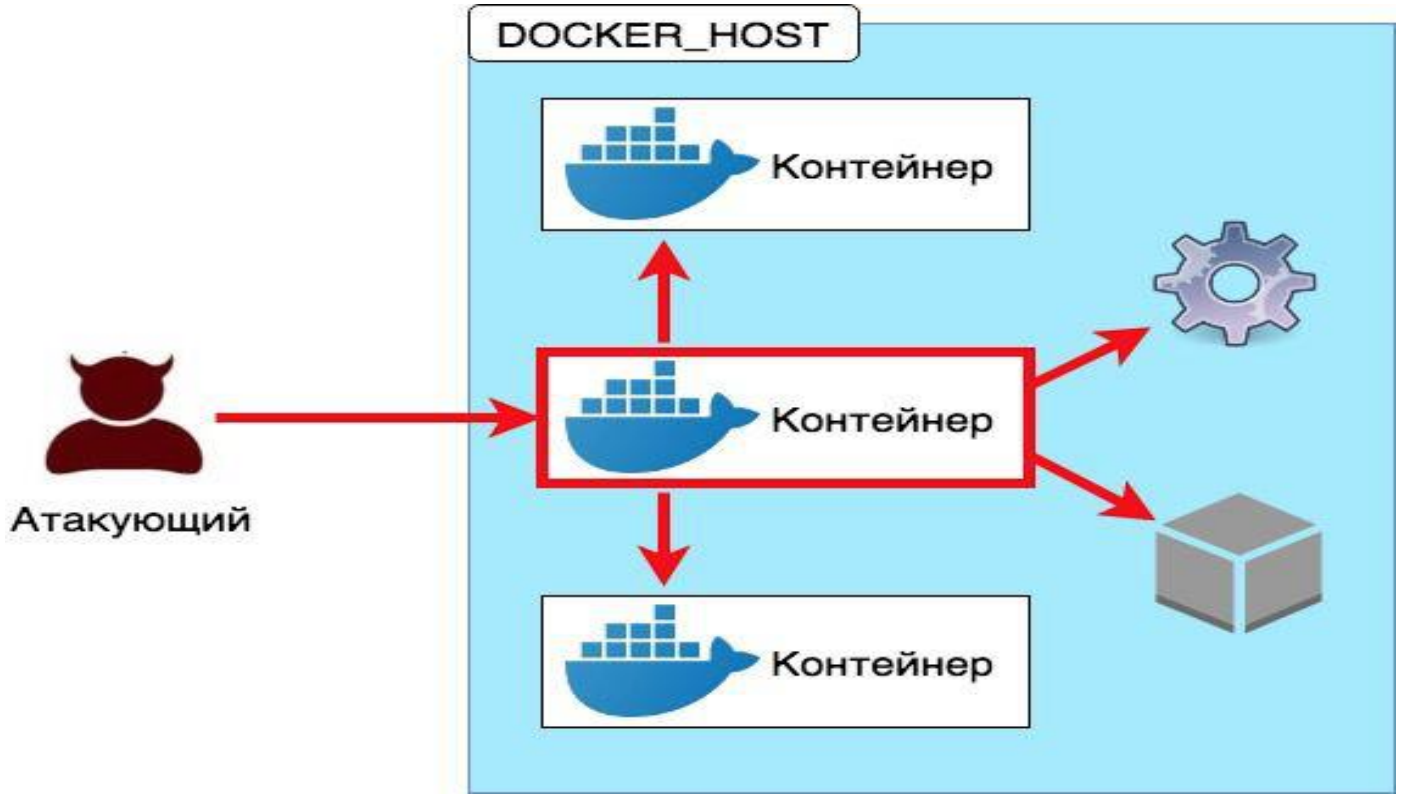


Безопасность среды docker и k8s

- Изоляция контейнеров (валидные конфигурации)
  - друг от друга
  - контейнеров и ОС
  - ограничение используемых ресурсов

- Контроль Runtime - среды
- Безопасный деплоймент контейнеров

Атака в среде docker



Механизмы изоляции контейнеров

- системные механизмы
  - пространства имен (namespaces)
  - cgroups
  - seccomp
  - ограничение linux capabilities
  - MAC в SeLinux (RedHat) / AppArmor (Debian)
  - непривилегированные пользователи ОС
- контроллер допуска PodSecurityPolicy

Безопасная конфигурация k8s

- Проблемы:
  - k8s - распределенная очень сложная система
- Решение:
  - готовые решения проверки конфигурации среды k8s CIS security benchmark
    - kube-bench (Aqua Security)
    - docker-bench-security (docker)
  - разработка тестов для собственных контейнеров

Пример использования kube-bench

```
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --allow-privileged argument is set to false (Scored)
[FAIL] 1.1.2 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 1.1.3 Ensure that the --basic-auth-file argument is not set (Scored)
[PASS] 1.1.4 Ensure that the --insecure-allow-any-token argument is not set (Scored)
[FAIL] 1.1.5 Ensure that the --kubelet-https argument is set to true (Scored)
[PASS] 1.1.6 Ensure that the --insecure-bind-address argument is not set (Scored)
[PASS] 1.1.7 Ensure that the --insecure-port argument is set to 0 (Scored)
[PASS] 1.1.8 Ensure that the --secure-port argument is not set to 0 (Scored)
[FAIL] 1.1.9 Ensure that the --profiling argument is set to false (Scored)
[FAIL] 1.1.10 Ensure that the --repair-malformed-updates argument is set to false (Scored)
[PASS] 1.1.11 Ensure that the admission control policy is not set to AlwaysAdmit (Scored)
[FAIL] 1.1.12 Ensure that the admission control policy is set to AlwaysPullImages (Scored)
[FAIL] 1.1.13 Ensure that the admission control policy is set to DenyEscalatingExec (Scored)
[FAIL] 1.1.14 Ensure that the admission control policy is set to SecurityContextDeny (Scored)
[PASS] 1.1.15 Ensure that the admission control policy is set to NamespaceLifecycle (Scored)
[FAIL] 1.1.16 Ensure that the --audit-log-path argument is set as appropriate (Scored)
[FAIL] 1.1.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Scored)
[FAIL] 1.1.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Scored)
[FAIL] 1.1.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Scored)
[PASS] 1.1.20 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)
[PASS] 1.1.21 Ensure that the --token-auth-file parameter is not set (Scored)
[FAIL] 1.1.22 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Scored)
```

Пример использования d

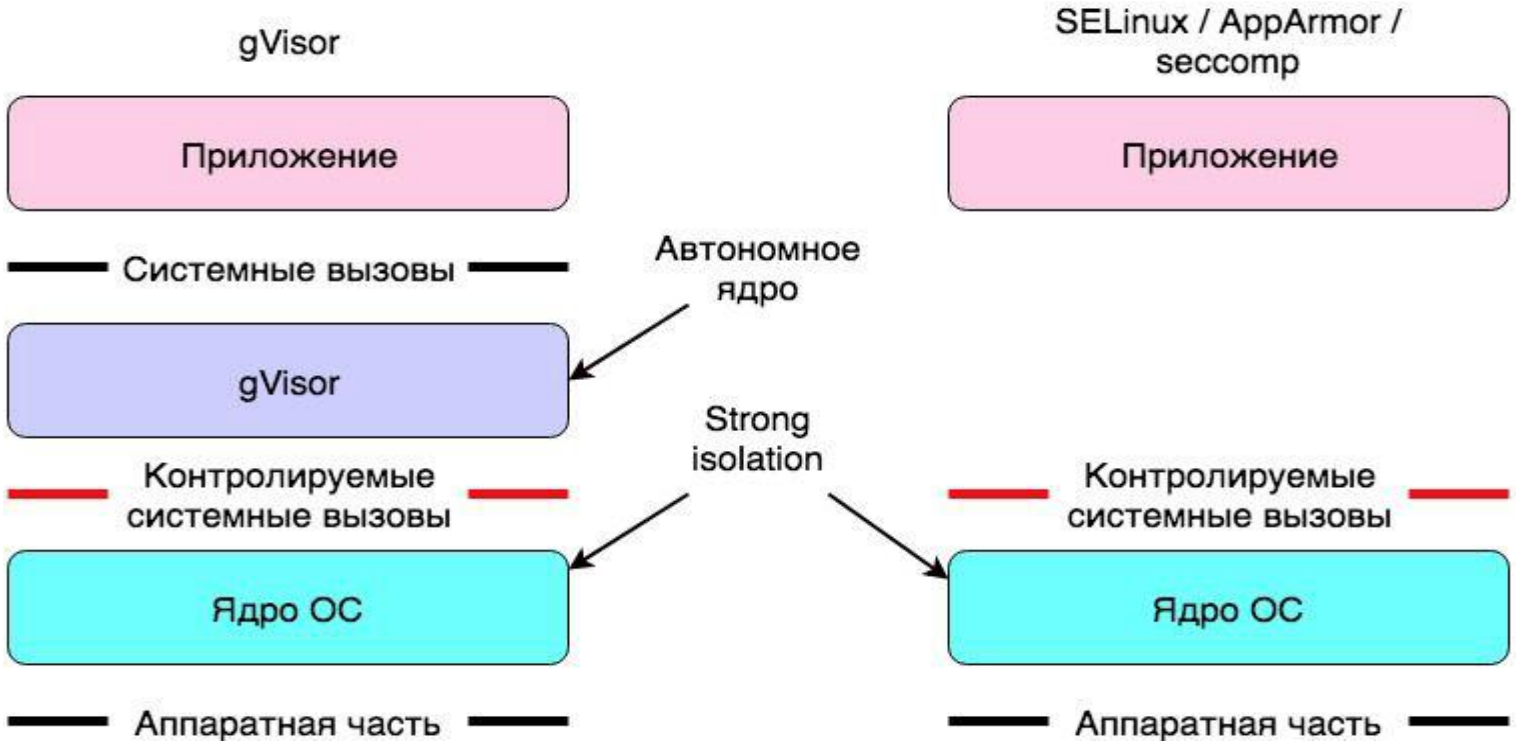
```
[INFO] 1 - Host Configuration
[WARN] 1.1  - Ensure a separate partition for containers has been created
[NOTE] 1.2  - Ensure the container host has been Hardened
[PASS] 1.3  - Ensure Docker is up to date
[INFO]      * Using 17.06.0 which is current
[INFO]      * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4  - Ensure only trusted users are allowed to control Docker daemon
[INFO]      * docker:x:992:vagrant
[WARN] 1.5  - Ensure auditing is configured for the Docker daemon
[WARN] 1.6  - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7  - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8  - Ensure auditing is configured for Docker files and directories - docker.service
[INFO] 1.9  - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO]      * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO]      * File not found
[INFO] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[INFO]      * File not found
[WARN] 1.12 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
[WARN] 1.13 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-runc


[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge
[PASS] 2.2  - Ensure the logging level is set to 'info'
[PASS] 2.3  - Ensure Docker is allowed to make changes to iptables
```

Hard multi-tenancy

- Проблемы:
  - ни один из контейнеров не доверяет другому
  - механизмы изоляции контейнеров потенциально уязвимы
- Решение:
  - двойной контроль доступа в механизмах изоляции

gvisor


gVisor
SELinux / AppArmor / seccomp
Приложение
Приложение
Системные вызовы
Автономное ядро
gVisor
Strong isolation
Контролируемые системные вызовы
Контролируемые системные вызовы
Ядро ОС
Ядро ОС
Аппаратная часть
Аппаратная часть

Сетевые политики

- традиционный Firewall
  - использование статических IP адресов
  - использование статических диапазонов портов, в том числе для SNAT / DNAT
- Firewall для среды k8s
  - быстрая смена IP адресов
  - сервис “service discovery” для определения IP адресов и портов
  - состояние системы хранится в etcd

Реализация сетевых политик

- CNI плагины
  - calico
  - cilium
  - kube-router
- правила сетевых политик
  - ingress и egress правила
  - фильтрация по портам, ip адресам, пространствам имен namespaces, именам pod
  - правила stateful

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: network-policy-db
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
    - namespaceSelector:
        matchLabels:
          project: dbproject
    - podSelector:
        matchLabels:
          role: frontend
    ports:
    - protocol: TCP
      port: 6379
  egress:
  - to:
    ports:
    - protocol: TCP
      port: 5978
```

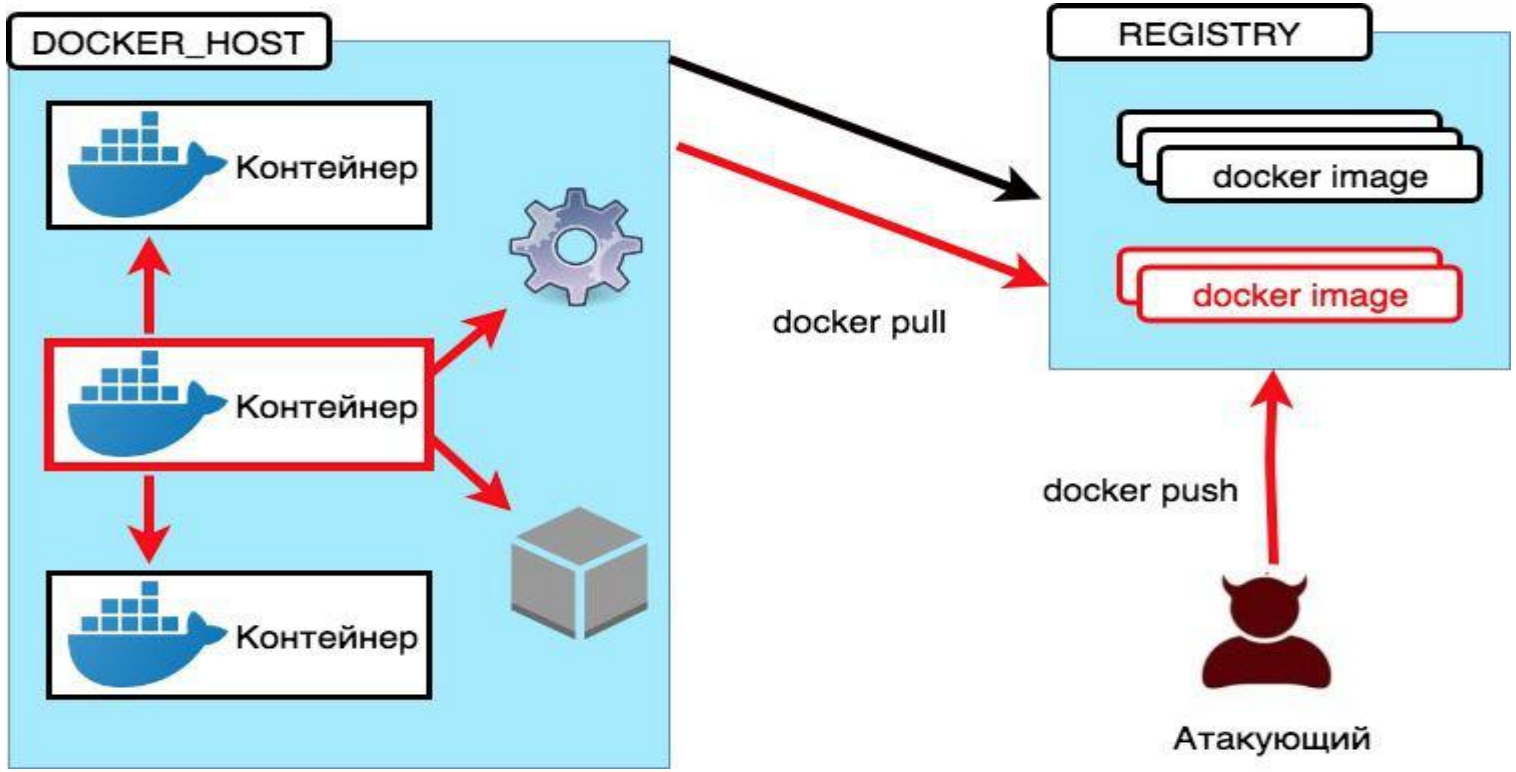
Деплоймент контейнеров

- Защита от MiTM атак
- Ограничение доступа к образам посредством RBAC
- Безопасность образов
  - образы без известных уязвимостей
  - образы без вредоносного кода

Content trust

- стандартный механизм, реализован в docker
- проверка сертификата издателя
- проверка целостности образа при загрузке в репозиторий
- проверка целостности образа при загрузке демоном docker
- управление через переменную среды DOCKER_CONTENT_TRUST

Атака через репозиторий



Функции репозитория

- управление образами
- авторизация пользователей
- RBAC
- статическое сканирование образов
- проверка целостности

Docker trusted registry

- авторизация и RBAC
- поддержка ContentTrust
- статическое сканирование с использованием БД CVE
  - идентификация ПО каждого слоя
  - сканирование бинарных файлов
  - поиск уязвимостей в статистически слинкованных модулях
  - поиск уязвимостей при сборке с нестандартными именами

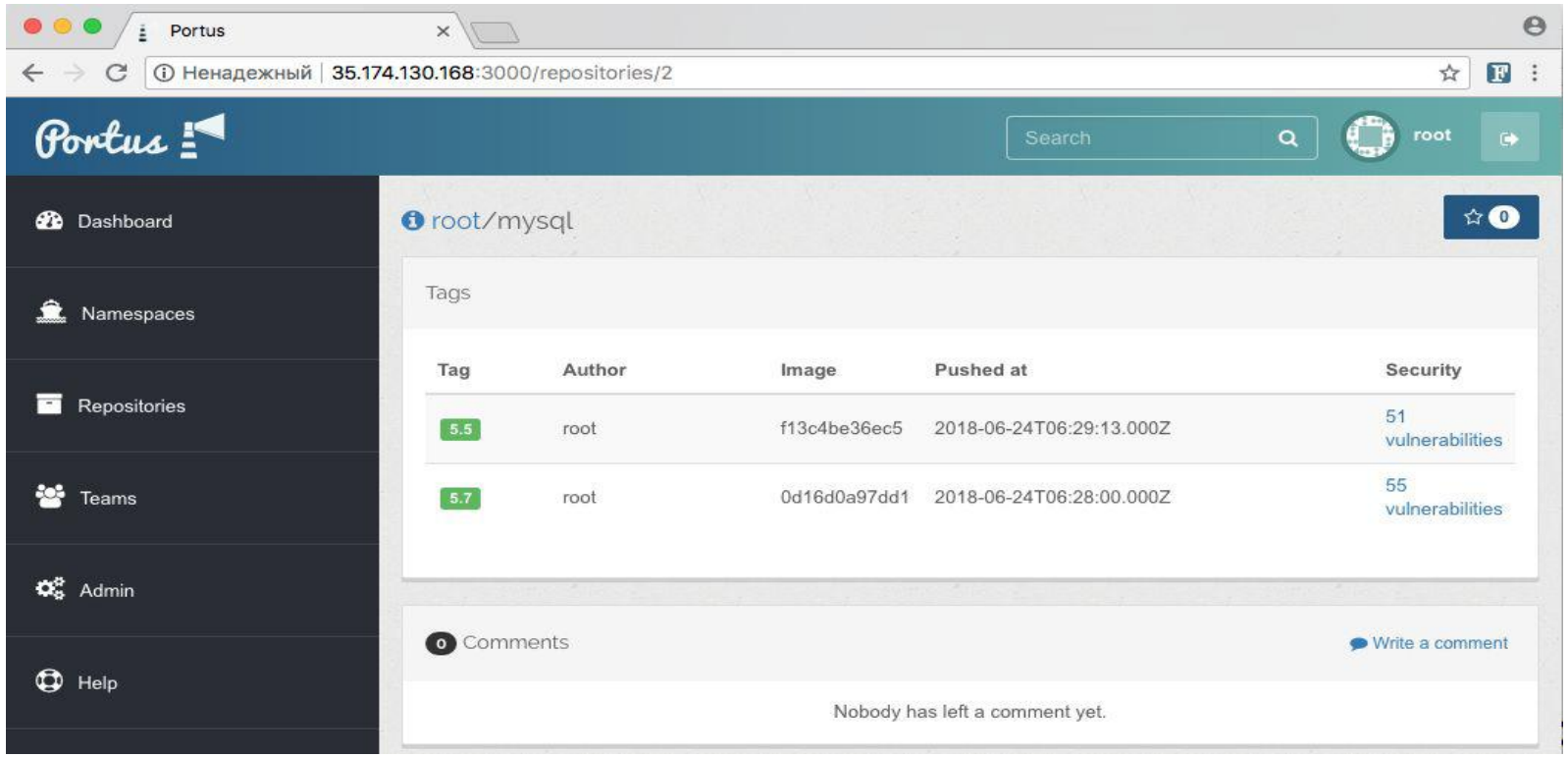
Сканер clair

- статическое сканирование образов
  - идентификация ПО каждого слоя
- обновление БД CVE
  - alpinelinux, debian, linux-oracle, redhat, ubuntu
- ограничение повторного сканирования
  - индексирование и хранение результатов в собственной БД
  - повторное сканирование после обновления БД CVE

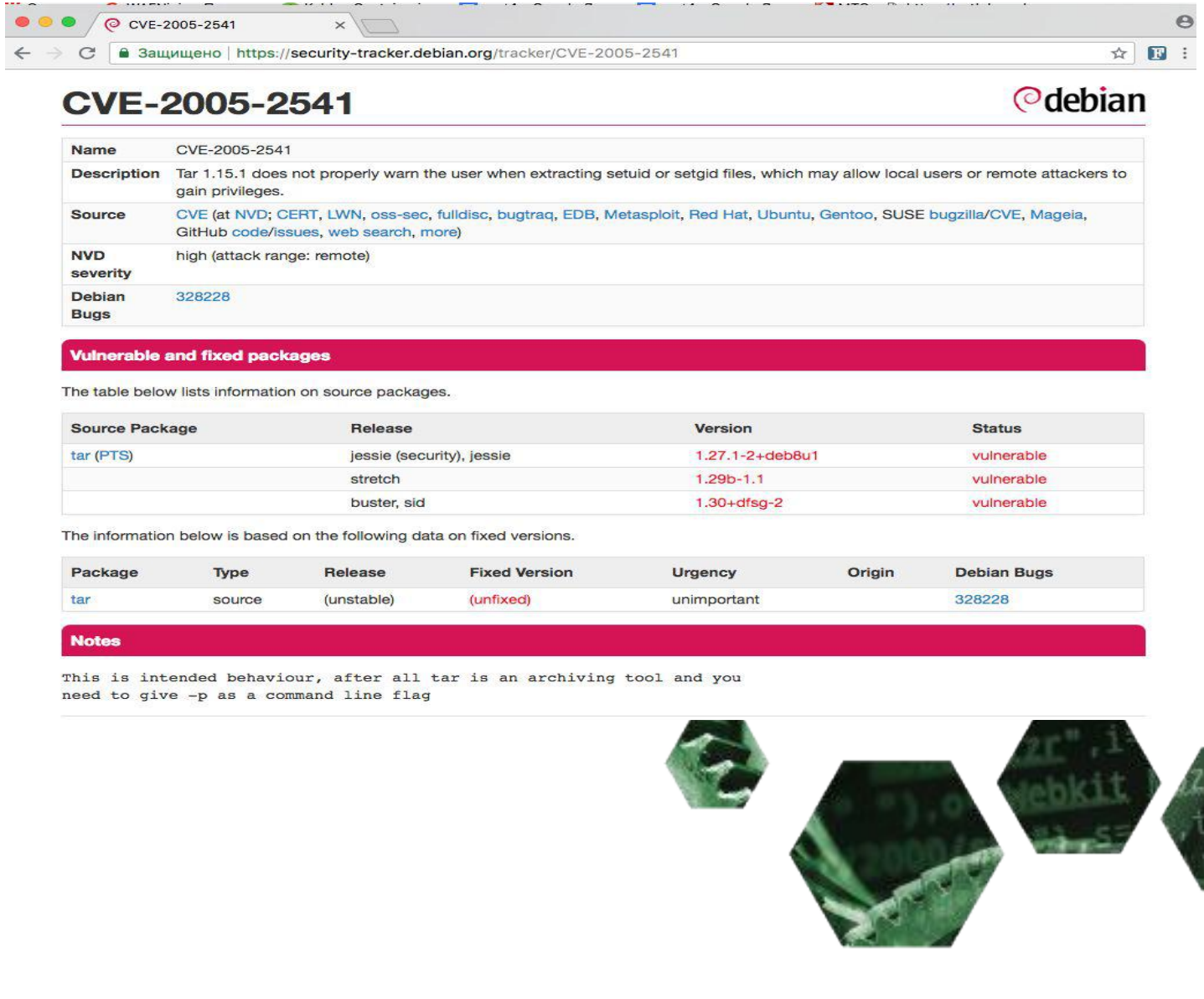
Portus

- дружелюбный веб-интерфейс для docker репозитория
- функции поиска и работа с тэгами
- сервис авторизации
  - LDAP
  - OAuth
  - OpenID
- собственный RBAC
- мониторинг и аудит действий пользователей
- поддержка clair

Portus: пример деплоймента



Portus: пример деплоймента



dagda

- статический анализ уязвимостей
  - анализ версий ПО
  - CVE, BID, RHSA, RHBA
- обнаружения подозрительного ПО
  - статический анализ контейнеров
  - антивирус ClamAV
- интегрирован в Sysdig Falco
  - как сканер образов
  - как мониторинг контейнеров во время выполнения

kublr: как мы строили безопасную среду

Ключевые моменты в обеспечении безопасности:

- безопасность хостовых ОС
- безопасный деплоймент контейнеров
- выявление и обновление уязвимого ПО

Проблемы безопасности:

- сложная конфигурация системы
- отсутствие универсальных средств оценки конфигурации
- использование недоверенных образов и/или уязвимого ПО

kublr

Спасибо!