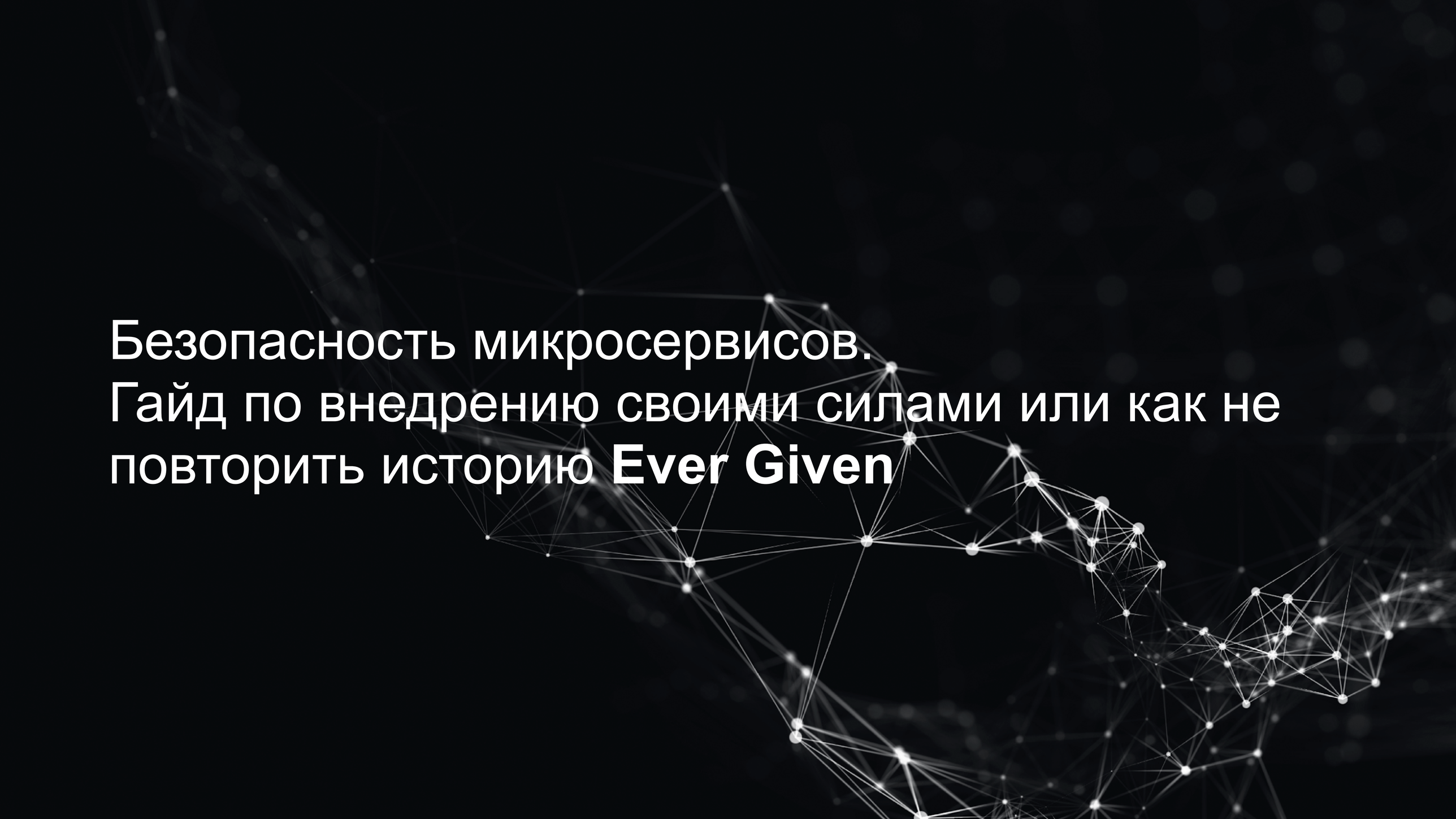
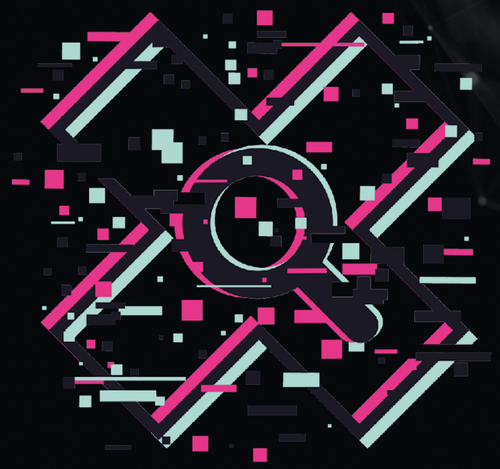
Безопасность микросервисов. Гайд по внедрению своими силами или как не повторить историю **Ever Given**



Agenda

* Темная сторона безопасности Cloud Native.
* Подходы к решению
* SSDLC: с чего и как начать выстраивать
* DevSecOps: от принципов к действиям.
* Безопасный конвейер разработки.



Темная сторона Cloud Native. Угрозы и предпосылки к внедрению

Предпосылки для внедрения DevSecOps

- Облака туманы и особенно их безопасность
- Cloud-Native несет в себе большой пул угроз
- Нет достаточного количества инструментов для повышения Security & Observability
- 90% + инцидентов это дефекты ПО и мисконфиги в микросервисах
- Профессиональная деформация = 0 - trust

Риски облаков

Риски провайдера Cloud

- Данные нельзя потрогать
- Общие ресурсы
- Появляются договорные ограничения/SLA/соглашения о КБ
- Ограничения со стороны провайдера (можно сделать только то что разрешает провайдер)
- Непонятен уровень защищенности и вовлеченности провайдера в процессы КБ

Непривычный PaaS

- Программно-определяемое всё
- Не традиционные средства ЗИ
- Высокая сложность и динамичность
- Высокие риски неправильной настройки
- Требует DevSecOps и автоматизации
- Требует понимания специфичных угроз для сред оркестрации и контейнеризации

Угрозы облаков

1. Утечки данных
2. Неверная конфигурация и недостаточный контроль изменений
3. Отсутствие паттерн безопасной архитектуры и стратегии облачной безопасности.
4. Недостаточная уровень контроля управлением идентификацией, учетными данными, доступом и ключами
5. Взлом аккаунта
6. Внутренние угрозы – угрозы утечки информации
7. Небезопасные интерфейсы взаимодействия и API
8. Непрозрачность использования облачных сервисов - Shadow IT
9. Злоупотребление и неправомерное использование облачных сервисов

НАСКОЛЬКО ОБЛАКА ЗАЩИЩЕНЫ?

РОВНО НАСТОЛЬКО НАСКОЛЬКО ВЫ В НИХ ПОГРУЖАЕТЕСЬ

PoC Hosts

Host details

Hostname	preprod	Cluster	control-plane
OS distribution	Ubuntu 20.04.1 LTS		
OS release	focal		
Scan time	Jul 11, 2023 11:49:25 AM		
Docker version	19.03.5		
Kubernetes version	1.21.14		

Vulnerabilities | Compliance | Runtime | Package info | Environment

Filter vulnerabilities by keywords and attributes

Type	Highest severity	Description
OS	high	systemd (used in libnss-systemd, libsystemd0, udev, libpam-systemd, systemd-sysv, libudev1, systemd-timesyncd, systemd) version 245.4-4ubuntu3.2 has 3 vulnerabilities. Affected service: systemd-journald.
OS	high	sudo version 1.8.31-1ubuntu1.1 has 2 vulnerabilities.
OS	high	snapd version 2.45.1+20.04.2 has 5 vulnerabilities.
OS	high	policykit-1 (used in libpolkit-agent-1-0, libpolkit-gobject-1-0, policykit-1) version 0.105-26ubuntu1 has 4 vulnerabilities. Affected service: polkit.
OS	high	openssl (used in libssl1.1, openssl) version 1.1.1f-1ubuntu2 has 8 vulnerabilities. Affected service: ssh.
OS	high	networkd-dispatcher version 2.0.1-1 has 2 vulnerabilities.
OS	high	linux (used in linux-modules-extra-5.4.0-42-generic, linux-modules-5.4.0-42-generic) version 5.4.0-42.46 has 312 vulnerabilities.
OS	high	intel-microcode version 3.20200609.0ubuntu0.20.04.2 has 9 vulnerabilities.

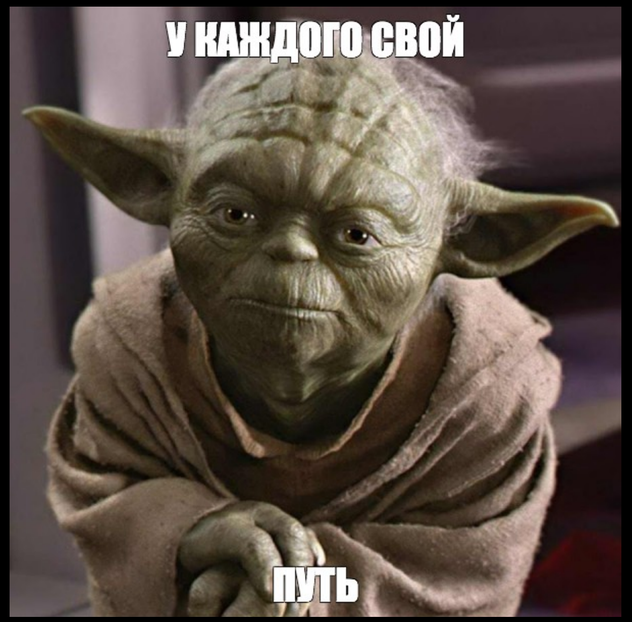
БЕЗОПАСНОСТЬ ОБЛАКА ЗАВИСИТ НЕ ТОЛЬКО ОТ MSCP НО И ОТ ВАС

PoC Images

Tag	Hosts	Clusters	Apps	Vulnerabilities	Risk factors	Collections
preprod	preprod	control-plane		383 381 84	10	-
preprod	preprod	control-plane		287 257 55	10	-
preprod	preprod	control-plane		279 242 45	10	-
preprod	preprod	control-plane		185 180 35 ⚠	9	-
1.16.1	preprod	control-plane		33 37 25	10	-
1.19	preprod	control-plane		19 22 25	10	-
1.21.4	preprod	control-plane		11 11 15	9	-
preprod	preprod	control-plane		11 11 15	9	-
3.6.2-debian-10-r156	preprod	control-plane		27 42 11	10	-
2.7.0-debian-10-r68	preprod	control-plane		27 40 11	10	-
3.6.1	preprod	control-plane		28 39 11	10	-
preprod	preprod	control-plane		8 34 8	9	-
preprod	preprod	control-plane		8 34 8	9	-
latest	preprod	control-plane		1 12 8	9	-
3.7.0-debian-10-r295	preprod	control-plane		12 23 7	10	-
2.8.1-debian-10-r132	preprod	control-plane		7 23 7	10	-
3.0.0.5-7e7a22e	preprod	control-plane		19 21 7	9	-

Подходы к решению

Подходы



Путь джедая

Пригласить друга

Путь джедая

Кто нужен?

Что нужно?

Инструменты SCA/SAST/DAST/FAST/ASOC

Стратегия, фреймворк, гайд по внедрению

Поддержка руководства и желание команды

Союзники - Security Champions

Помощь друга

Кто нужен?

Инструменты SCA/SAST/DAST/FAST/ASOC

Потребность в DevSecOps

Поддержка руководства

Ресурсы

Что нужно?

Плюсы/минусы подходов

Подход	Плюсы	Минусы
Путь джедая	• На старте минимальные вложения • Развитие внутренней экспертизы • Рост зрелости продукта вместе с ростом зрелости команд • Возможность тестирования различных гипотез и инструментов • Возможность создания процессов управления изменениями под конкретные команды и проекты • Повышение осведомленности команд • Глубокое погружение в процессы/архитектуру/код	• Дефицит кадров • ФОТ немаленький • Зависимость от людей и команд • Трудно масштабировать на больших проектах • Ограничения в опенсорсе • Отсутствие или чрезмерная сложность кастомизаци (допилить напильником придется все) • Высокий уровень False Negative или False Positive • Нет интеграций с существующими инструментами • Отсутствие экспертизы на рынке • Несколько раз заваленные пайплайны и Prod
Помощь друга	• Быстрый старт • Экономия времени и ресурсов • Опыт реализации разных по масштабу проектов • Возможность кастомизации под конкретные требования заказчика • Поддержка и сопровождение • Экономия сил и времени • Формализация процессов	• Большие расходы на старте • Возможно увеличение сроков по реализации проекта • Возможны ошибки при реализации проекта в виду низкой осведомленности о внутренних процессах разработки

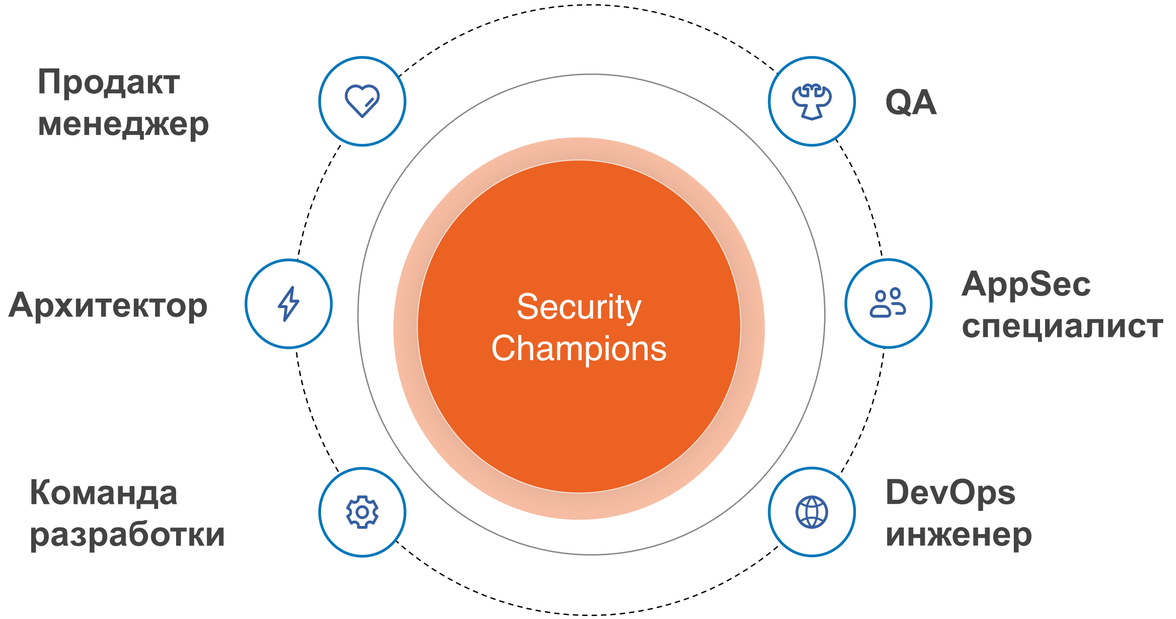
Наш опыт

Инструменты SCA/SAST/DAST/FAST/ASOC

Стратегия, фреймворк, гайд по внедрению

Поддержка руководства и желание команды

Союзники - Security Champions



Старт:

- 15 проектов
- Частые релизы
- 0 понимания что вообще происходит команда ИБ из 2-х человек
- куча опенсорса и гипотез
- Огромное желание делать свервисы безопасно

Итого:

- 6 продуктовых команд
- 500 проектов
- 2 AppSec + 1 QA
- Стек из опенсорса
- OPEX - затраты на КТС
- ФОТ - 820 000 месяц

MVP:

- 6 месяцев боли
- Внедрение SAST/DAST/FAST/CVA
- Оркестрация уязвимостей
- Борьба с фолзами, дедупликацией, надежностью
- Оценка команд по SVS
- Создание своего проекта и настройка пайплайнов
- Подведение итогов

PROD:

- Работа над оптимизацией
- Формирование требований безопасности
- Формирование метрик
- Повышение осведомленности
- Курсы для Dev
- Создание QG
- Масштабирование на другие проекты
- Управление SSDLC

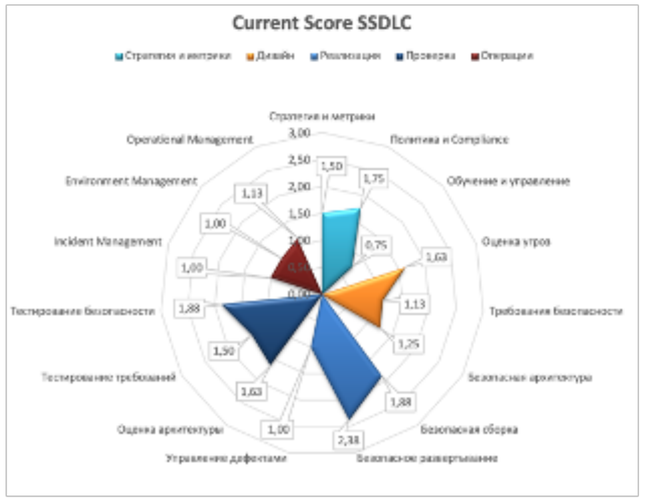
SSDLC:
с чего и как начали выстраивать

С чего начать?

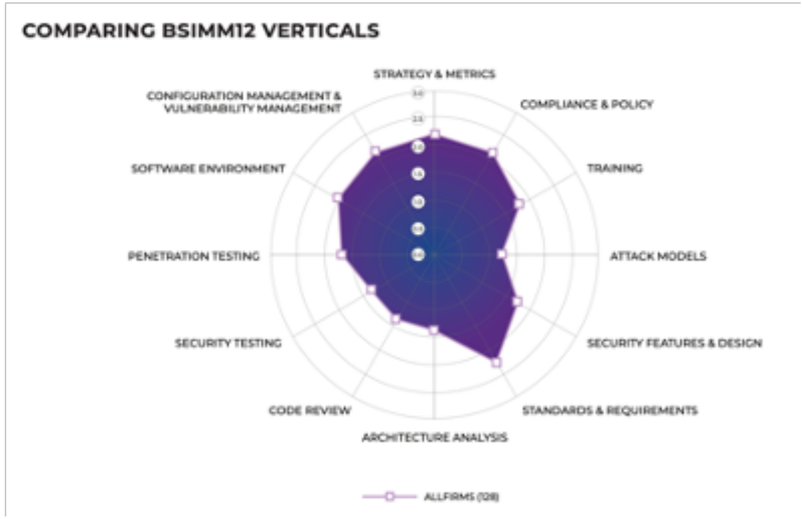
Самооценка наше все!

Предписывающая модель

Описательная модель



Сравнение тут



Рекомендации

- Обеспечьте себя людьми (AppSec)
- Начните с изучения существующих процессов
- Начните с определения наименее зрелых областей в процессе SSDLC и DevSecOps (SAMM/BSIMM)
- Оцените уровень соответствия Best Practices (CIS Benchmarks)
- Начните с инвентаризации сервисов и определения их владельцев
- Определите критичность сервисов вместе с командой
- Сформируйте технологический ландшафт продукта
- Определите какие инструменты безопасности вы готовы внедрять
- Сформулируйте требования к разрабатываемым сервисам (ASVS/MASVS/CSVS/CIS Benchmarks/QG)
- Найдите общие точки соприкосновения с остальными участниками команд (CI/CD/IaC/ARCHaC/DaC/Observability)
- Стройте безопасность шаг за шагом, опираясь на уровень критичности разрабатываемых сервисов
- Соберите всех лидов и сформируйте понятную для всех политику безопасной разработки (Политика безопасной разработки / RFC)
- Сделайте понятный для бизнеса калькулятор критичности уязвимостей и дефектов безопасности (Risk Calulator)
- Определите сроки устранения уязвимостей (SLO)
- Формируйте культуру DevSecOps и SSDLC (Continuing education)
- Определите требования по мониторингу
- Определите порядок и группу реагирования на инциденты (CERT/CSIRT)
- Покройте основные процессы метриками

Стратегия внедрения SSDLC

Шаг 1

Формирование политики и оценка текущих мер и практик SDLC

(SAMM/BSIMM)

Шаг 2

Создание RoadMap

по внедрению практик и инструментов SSDLC

поиск союзников и value

Шаг 3

Внедрение инструментов и авто-тестов безопасности в пайплайн

(Security pipeline SCA/SAST/ DAST)

Шаг 4

Внедрение безопасности на стадии проектирования и моделирование угроз

(SecArch паттерны ASVS/ MASVS/CSVS/NIST)

Шаг 5

Ручное ревью безопасности продукта

(WSTG/MSTG)

Шаг 6

Анализ выявленных уязвимостей и подготовка рекомендаций по их устранению

(PoC/Risk Calculator)

Шаг 7

Управление дефектами

(Defect Dojo/Jira/SLO)

Шаг 8

Обучение Best Practices (Secure Coding Practices/ Cheat Sheats)

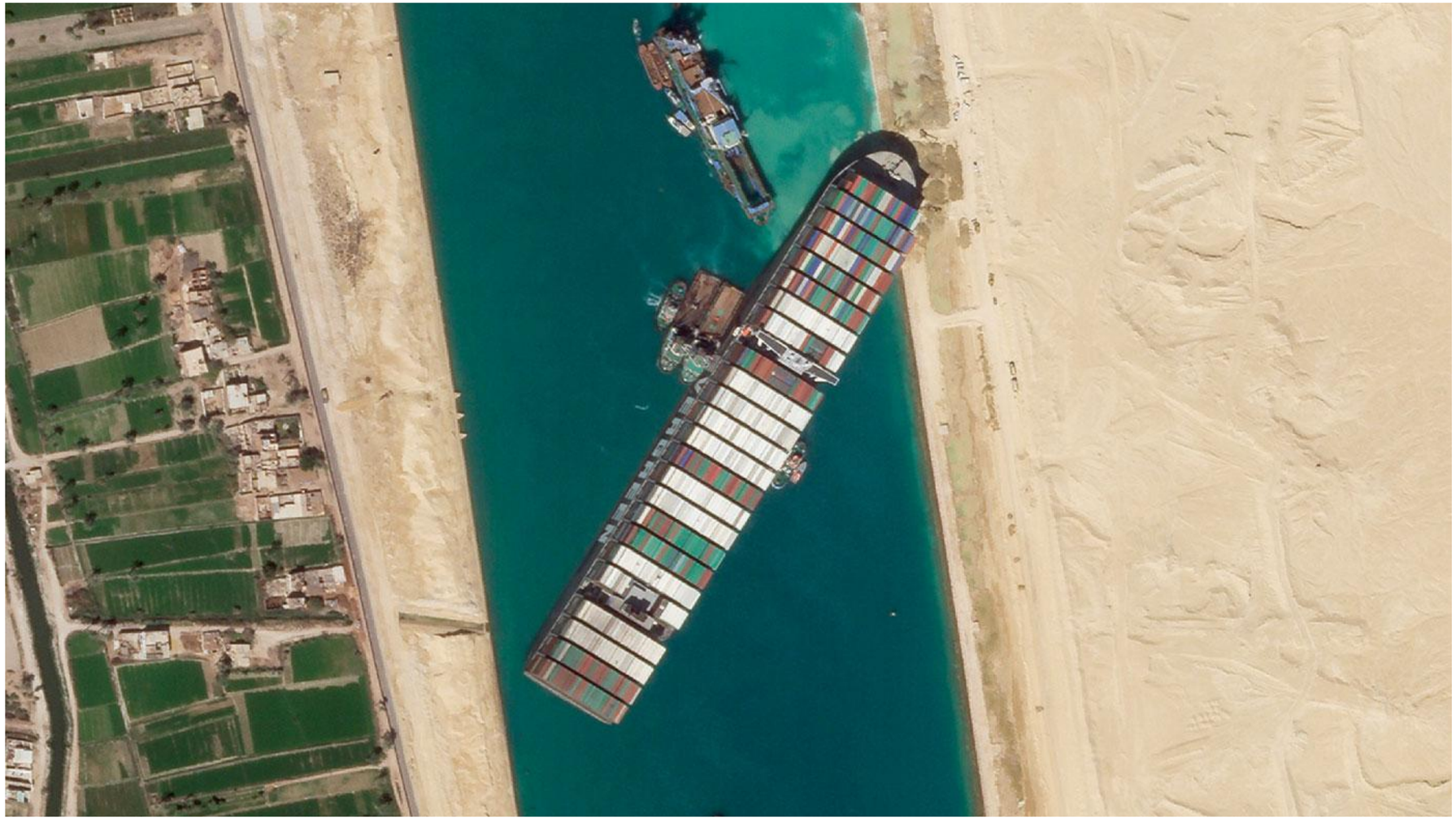
Шаг 9

Обеспечение непрерывности

(Runtime Security/Network Policy/Security Policy/Service Mesh/KSM/RBAC/IAM/RASP/ WAF/IDS/TI/ Vulnerability management)

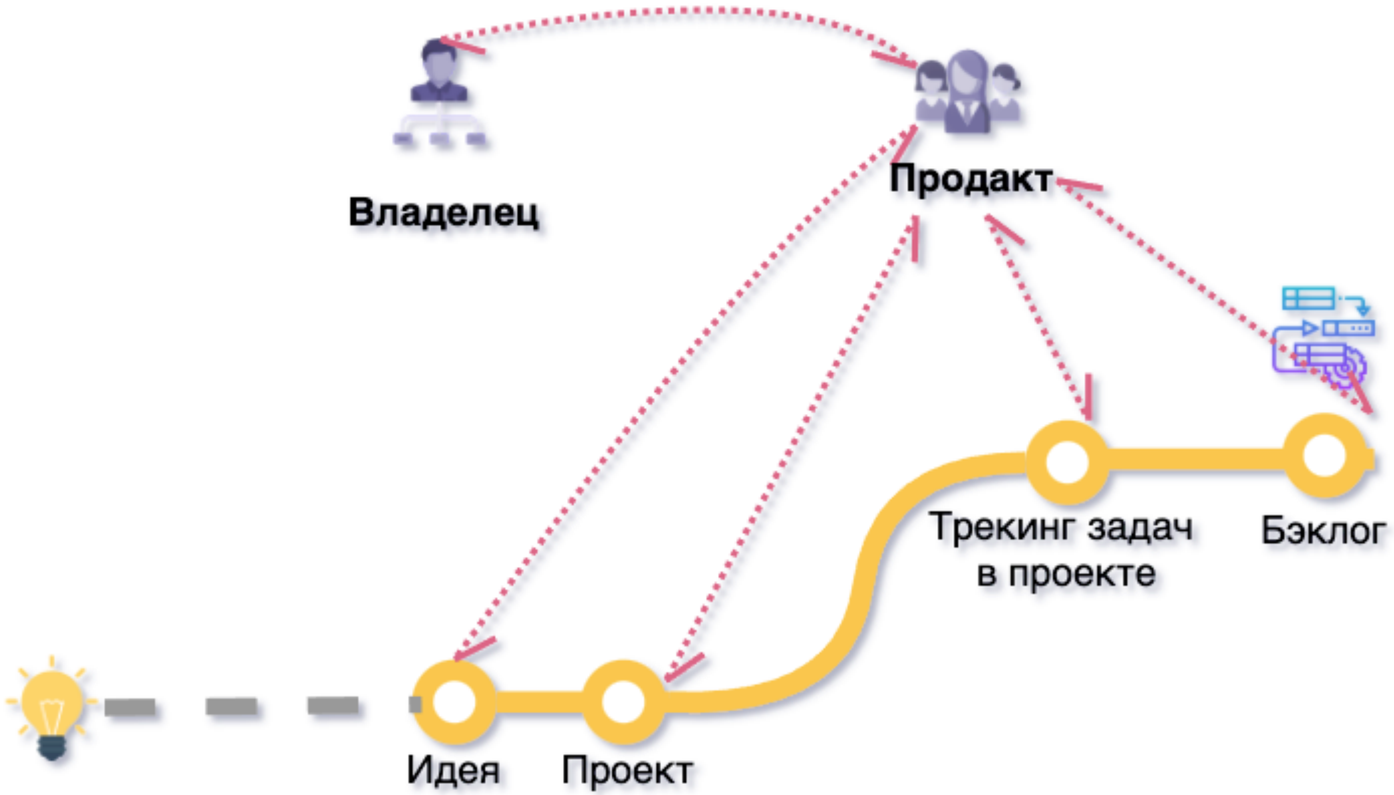
Причем тут Ever Given?

Когда решил внедрить SEC в DevOps




Процессы
ASVS/MASVS
SAMM/BSIMM
SSDLC
CSV
DAST/FAST
SAST
SCA
Технологии
DEV
QA
PM
DevOps
AppSec
Люди

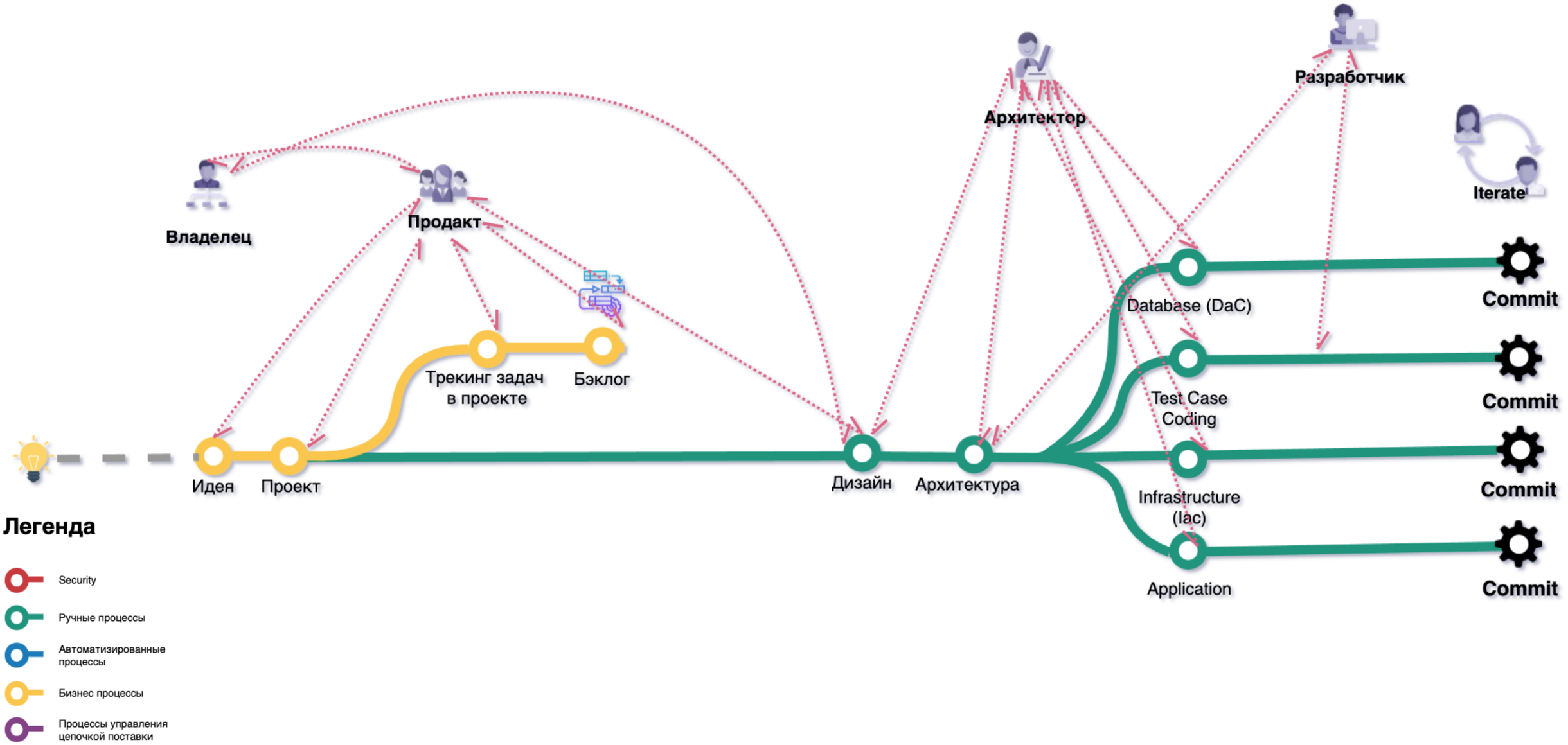
Step by step



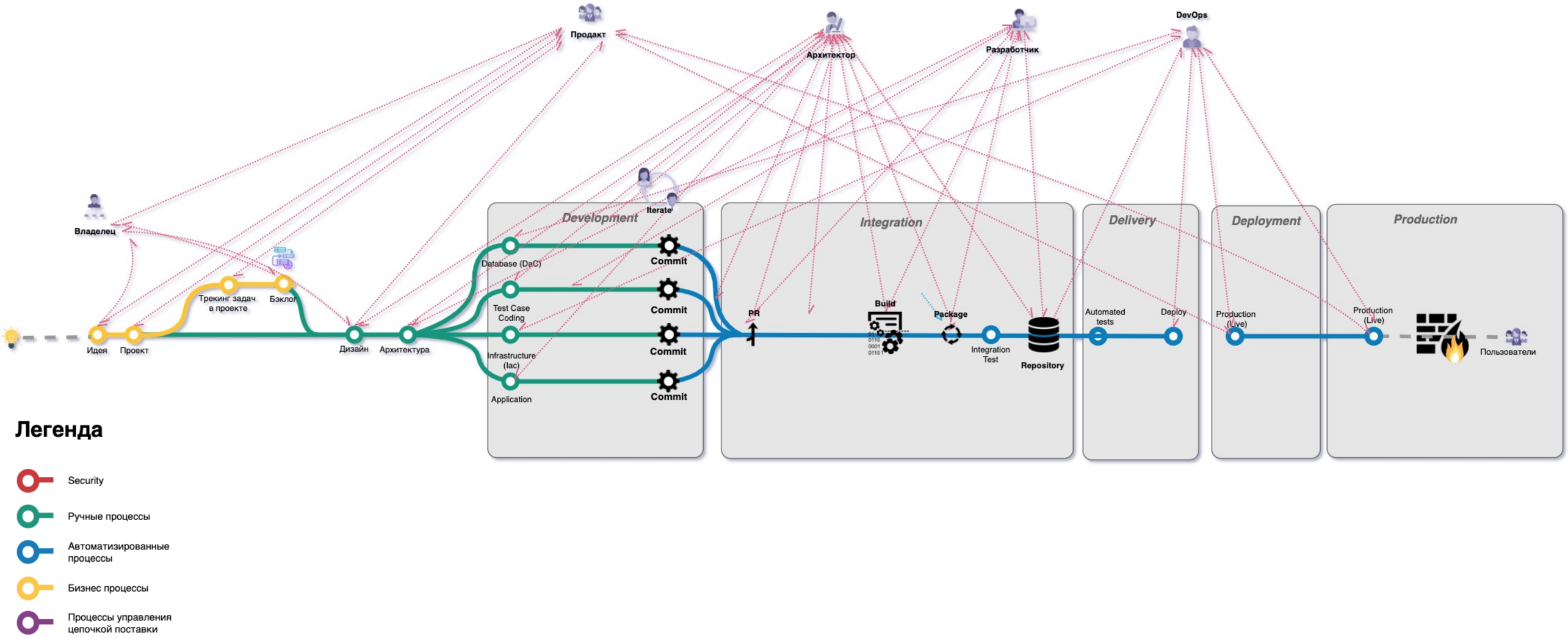
Легенда

- Security
- Ручные процессы
- Автоматизированные процессы
- Бизнес процессы
- Процессы управления цепочкой поставки

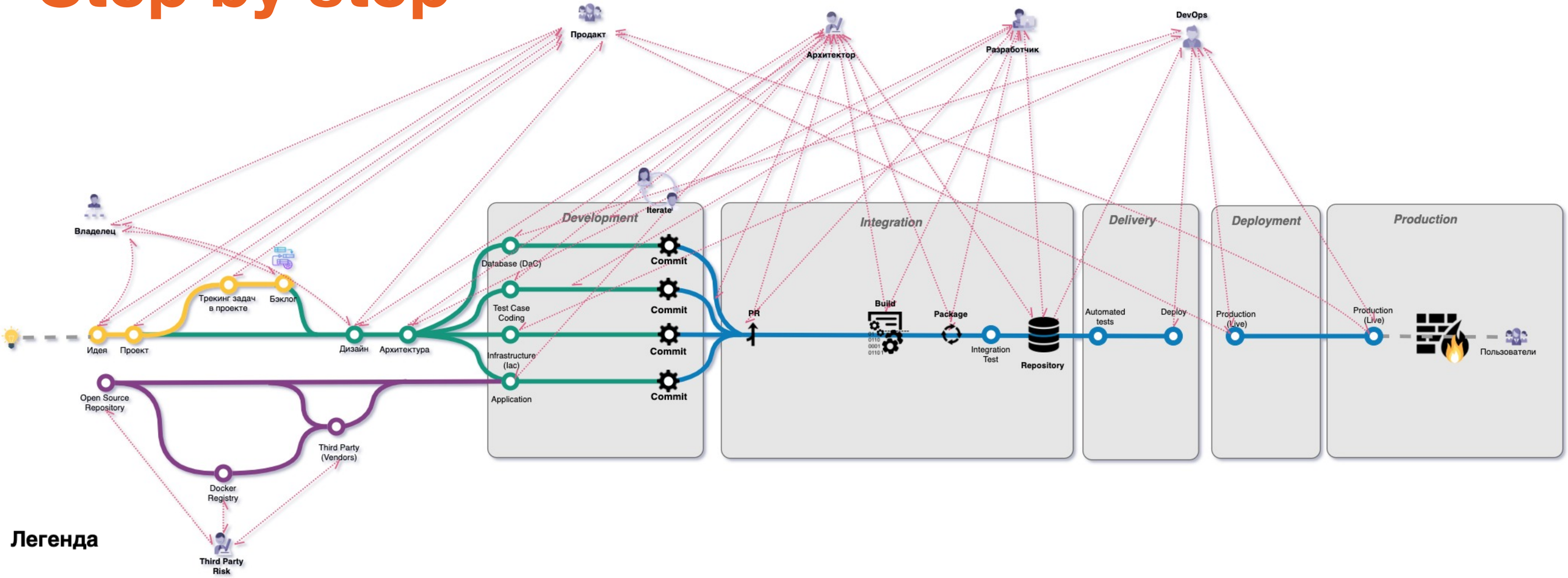
Step by step


Владелец
Продакт
Архитектор
Разработчик
Iterate
Трекинг задач в проекте
Бэклог
Database (DaC)
Test Case Coding
Infrastructure (Iac)
Application
Commit
Commit
Commit
Commit
Идея
Проект
Дизайн
Архитектура
Легенда
Security
Ручные процессы
Автоматизированные процессы
Бизнес процессы
Процессы управления цепочкой поставки

Step by step


Продакт
Архитектор
Разработчик
DevOps
Владелец
Iterate
Development
Integration
Delivery
Deployment
Production
Идея
Проект
Трекинг задач в проекте
Бэклог
Дизайн
Архитектура
Database (DaC)
Test Case Coding
Infrastructure (Iac)
Application
Commit
Commit
Commit
Commit
PR
Build
Package
Integration Test
Repository
Automated tests
Deploy
Production (Live)
Production (Live)
Пользователи
Легенда
Security
Ручные процессы
Автоматизированные процессы
Бизнес процессы
Процессы управления цепочкой поставки

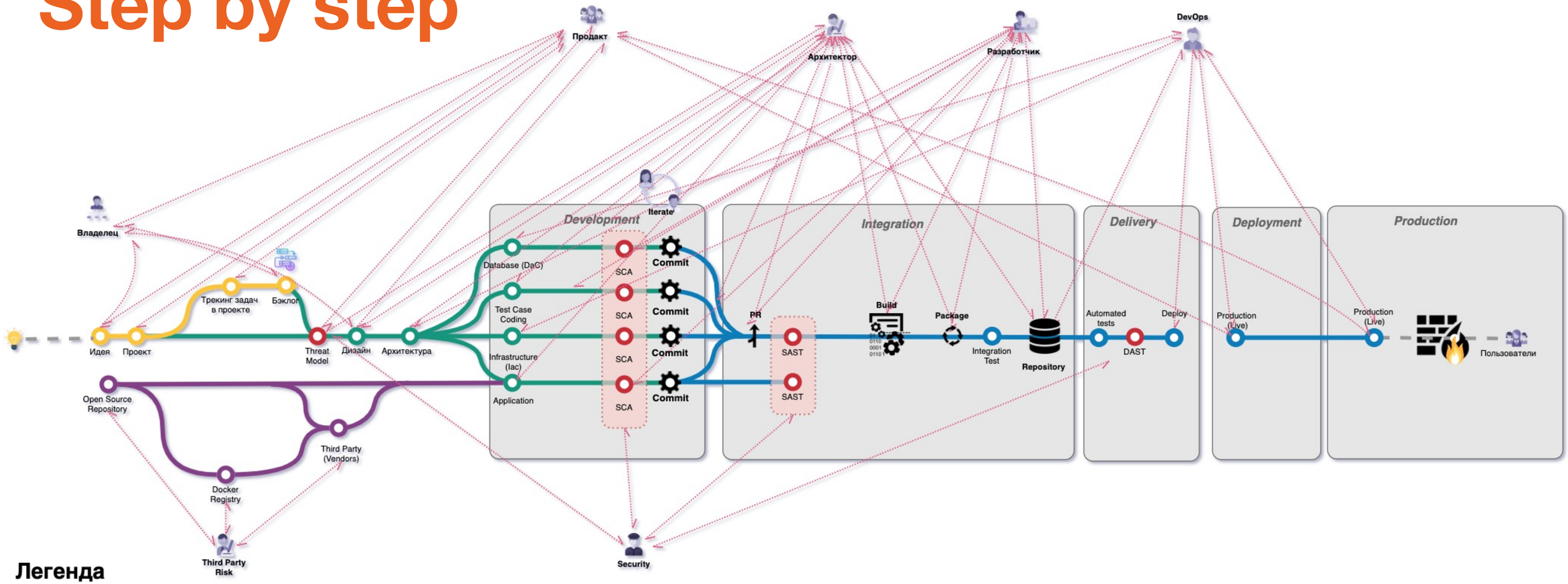
Step by step



Легенда

- Security
- Ручные процессы
- Автоматизированные процессы
- Бизнес процессы
- Процессы управления цепочкой поставки

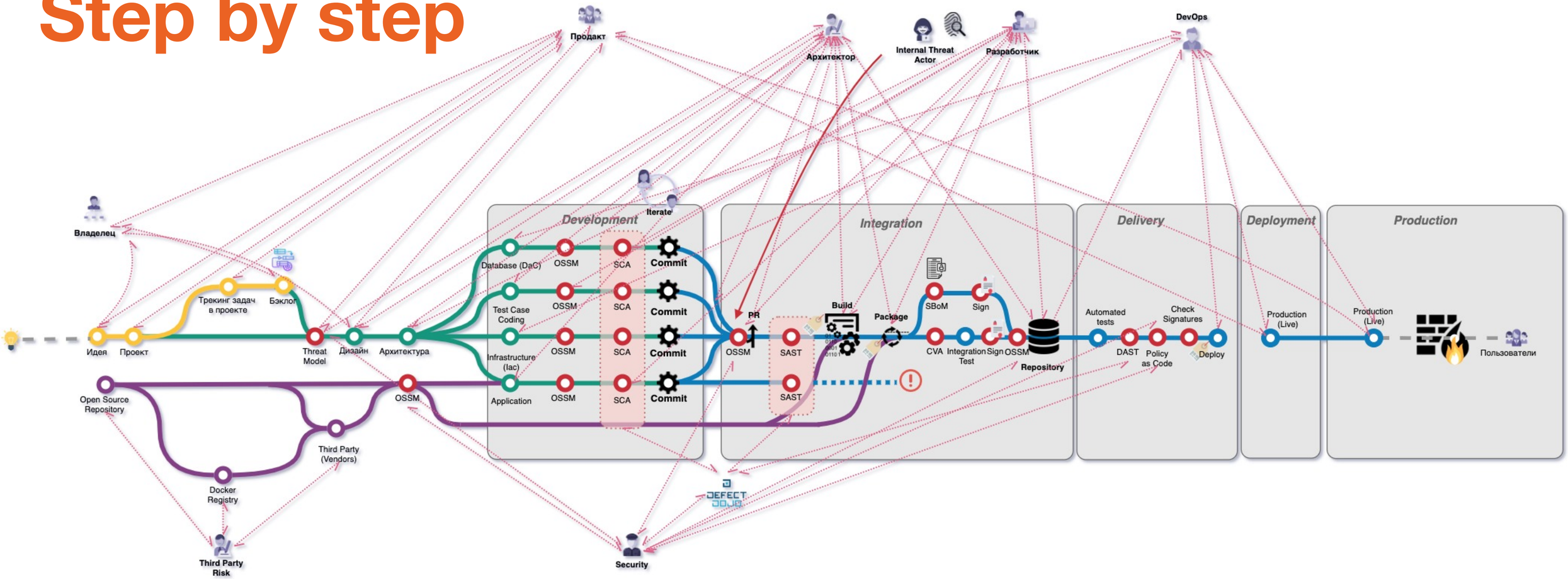
Step by step



Легенда

- Security
- Ручные процессы
- Автоматизированные процессы
- Бизнес процессы
- Процессы управления цепочкой поставки

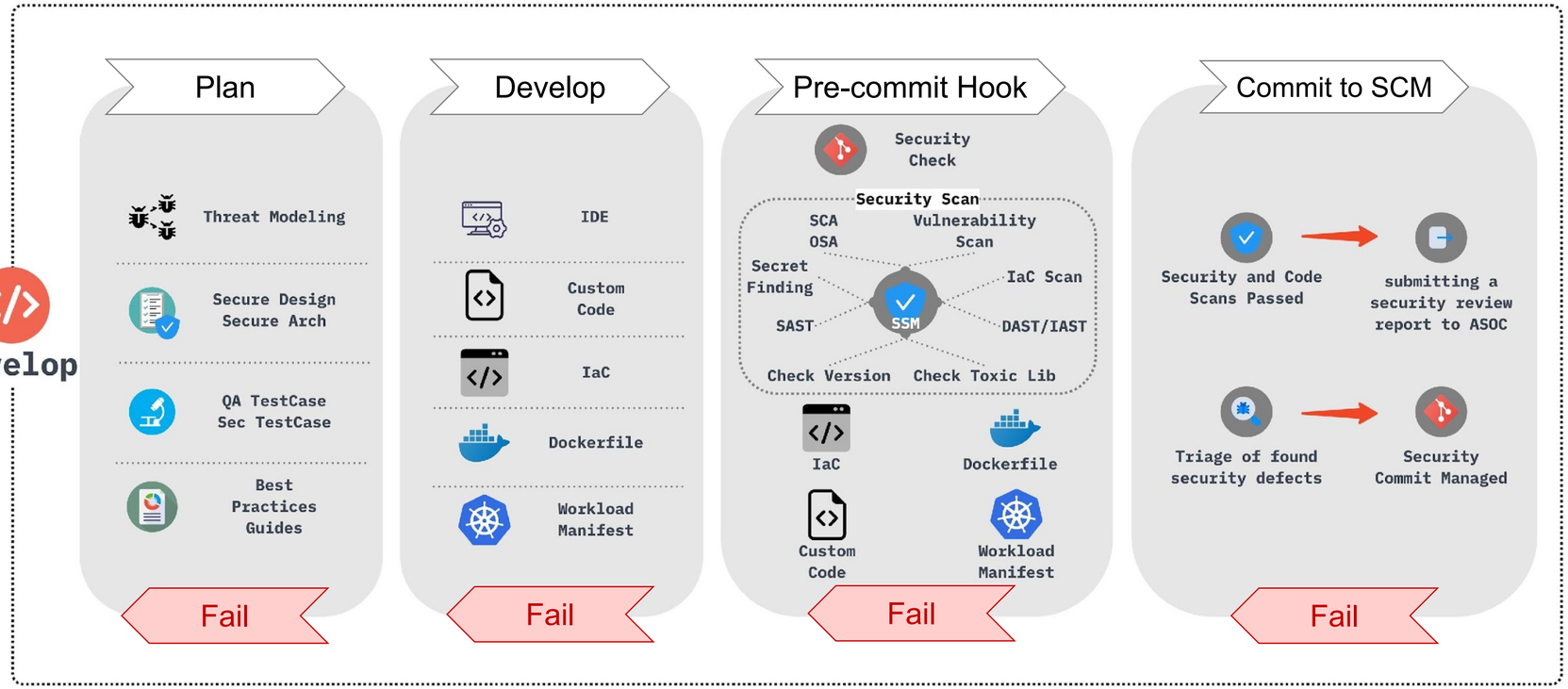
Step by step



Легенда

- Security
- Ручные процессы
- Автоматизированные процессы
- Бизнес процессы
- Процессы управления цепочкой поставки

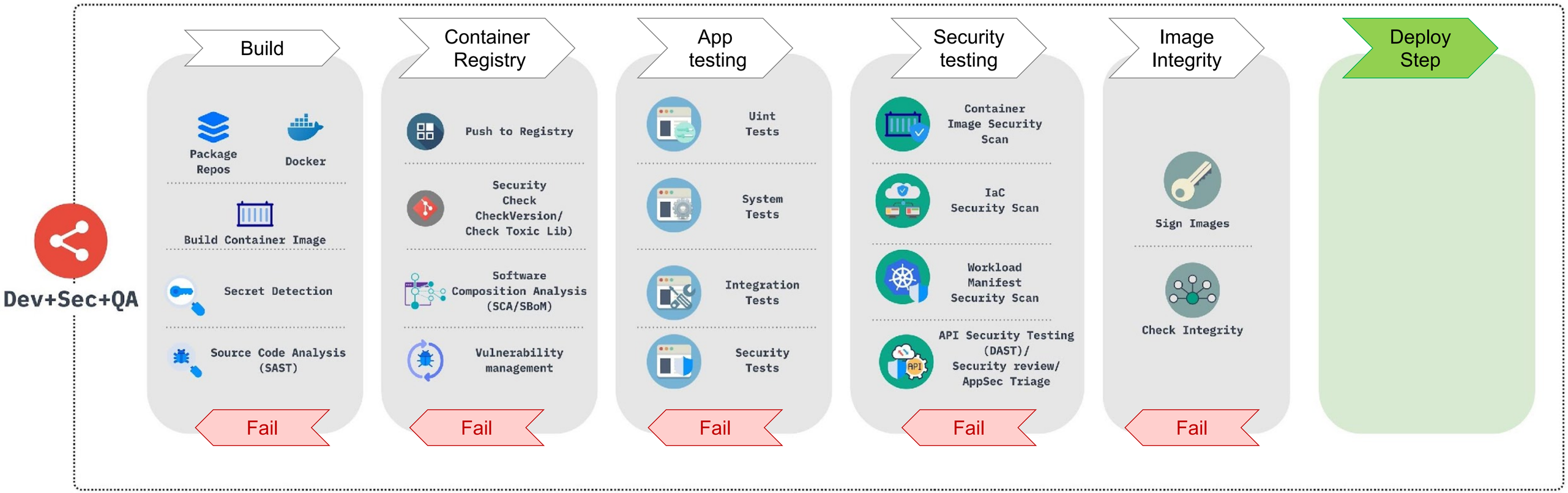
Dev+Sec



Best Practices: NIST SP 800-190. OWASP: ASVS/MASVS/CSVS/CheatSheats/OWASP Secure Coding Practices.

- Моделирование угроз
- Определение объектов воздействия
- Проектирование безопасной архитектуры
- Улучшение безопасности инфраструктуры
- Определение инструментов, контролей и этапов их внедрения
- Создание тест-кейсов по безопасности
- Обучение команды
- Проведение триажа найденных уязвимостей
- Проведение автоматизированного и ручного тестирования
- Повышение Observability
- Формирование SLA
- Измерение эффективность принятых мер

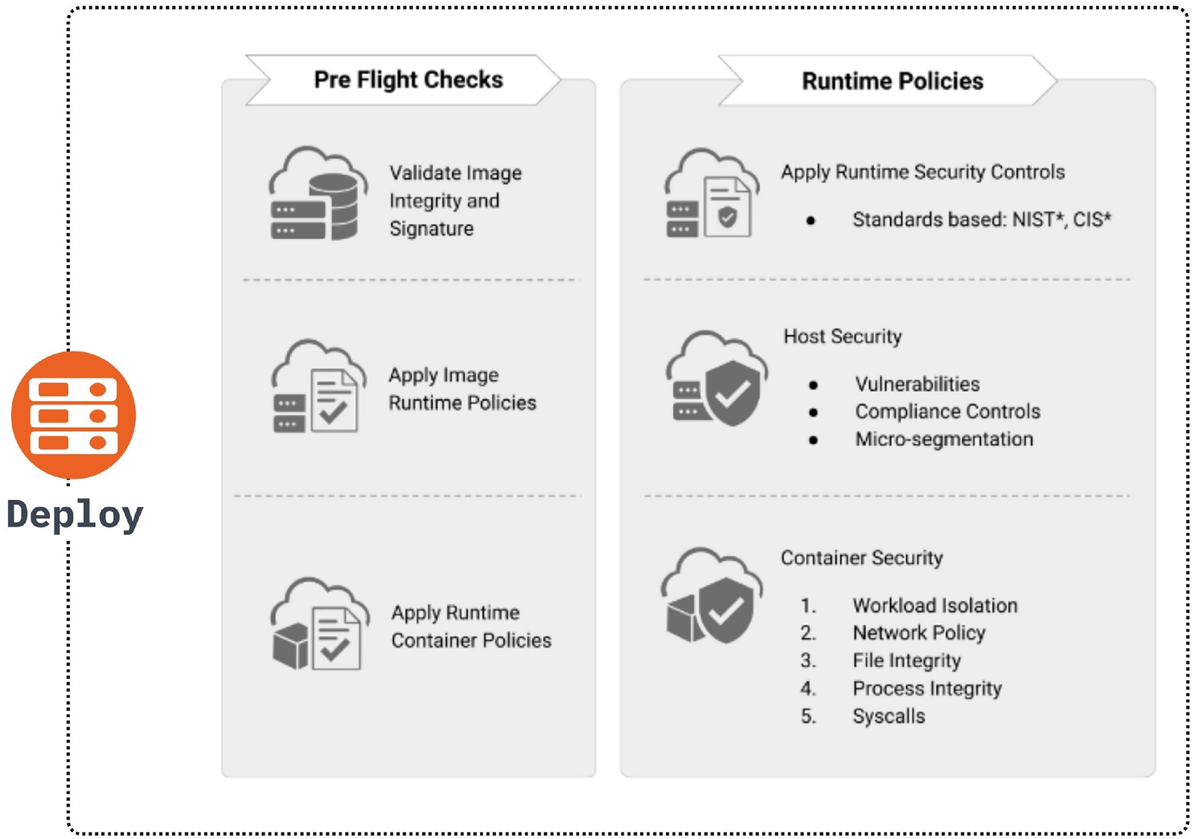
Dev+Sec+QA



Best Practices

OWASP: SAMM,WSTG,MSTG,ASVS,MASVS / Synopsys: DevSecOps Reference Architecture / CIS Benchmark / NIST

Sec+OPS

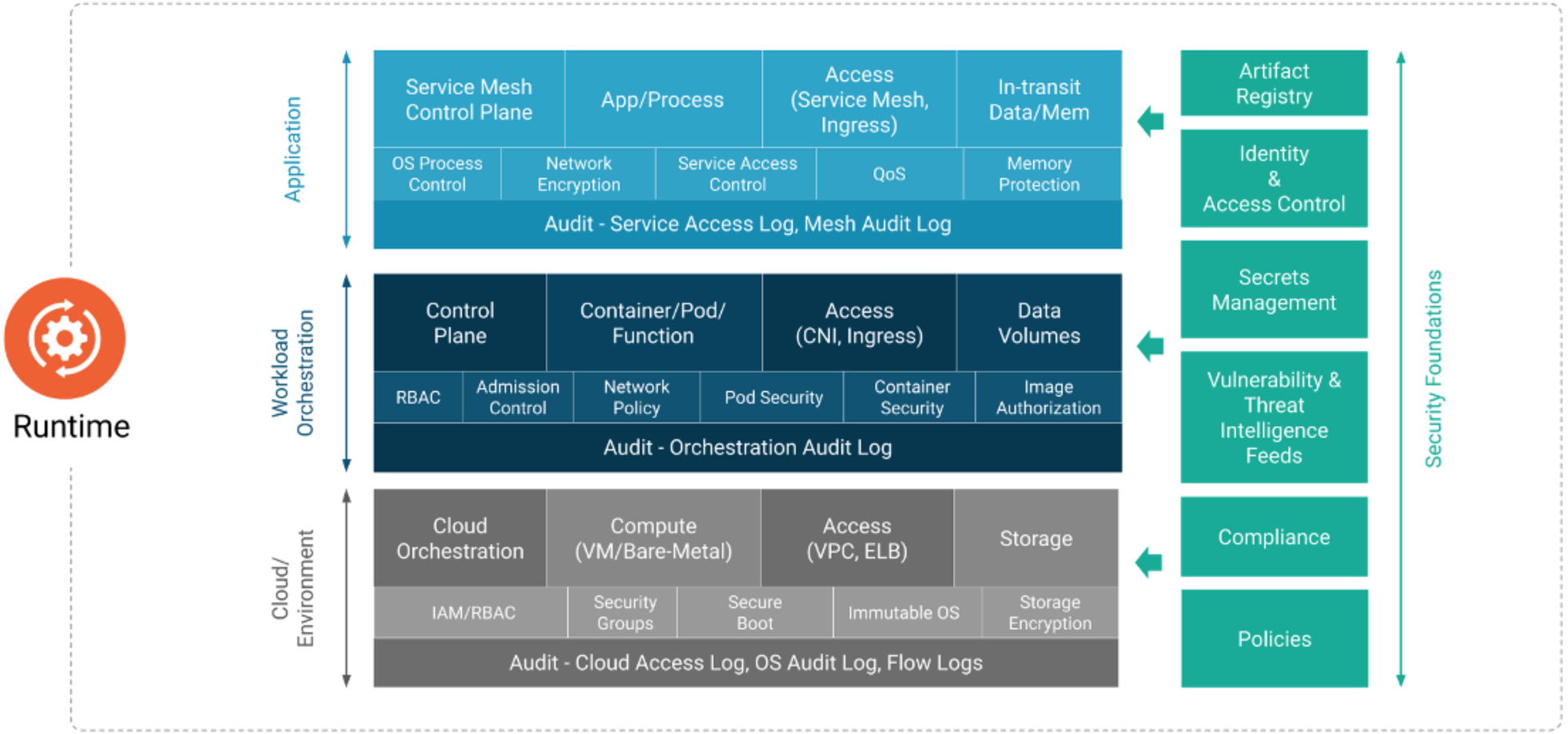


До развертывания образа контейнера, следует проверить наличие, применимость и текущее состояние:

- **Подписи образа и ее целостности**
- **Image Runtime Policies (например, отсутствие критических уязвимостей)**
- **Container Runtime Policies (например, отсутствие чрезмерных привилегий)**
- **Уязвимости хоста**
- **Соответствия Compliance**
- **Настройки микросегментации**
- **Изоляции Workerload**
- **Сетевых политик**
- **Системных вызовов**
- **Безопасной доставки секретов**

Best Practices: CIS Benchmark/CISA Kubernetes Hardening Guidance

Security Run-Time



Security Pipeline

Plan

- Требования безопасного проектирования
- Требования обеспечения конфиденциальности
- Управление активами
- Требования безопасности архитектуры
- Требование к безопасной разработке
- Моделирование угроз

Code

- Управление учетными данными и секретами
- Лучшие практики безопасной разработки
- Генерация тестов безопасности
- Использование Secure OSS
- Ручная проверка кода
- Безопасность API

Build

- Безопасность IaC
- Безопасность DaC
- Контроль OSS
- Безопасность конфигурации App

Internal Testing

- DAST
- SAST
- Функциональное тестирование безопасности
- Внутренний анализ защищенности
- Управление уязвимостями
- Автотесты по безопасности
- Модульное тестирование на безопасность

Release/ Deployment

- Безопасность контейнеров и Kubernetes
- Безопасный deploy
- UAT с привлечением безопасности
- Автоматизация проверки безопасности релиза
- Проверка правил WAF
- Проверка правил мониторинга
- Защита и контроль физической и виртуальной среды
- Безопасность транспортного уровня
- Безопасность поставщика (третьей стороны)

External Testing

- Тесты на проникновение
- Bug Bounty
- Управление рисками

Monitoring

- Мониторинг операций безопасности
- Мониторинг событий безопасности
- Управление инцидентами безопасности
- Учения по выявлению угроз

Run/Operate

- Исправление ошибок
- Patch менеджмент
- Новые обновления безопасности
- Управление инцидентами и реагирование
- Совместное обучение по вопросам безопасности
- Контроль версий и метаданных
- Compliance
- Безопасность контроля доступа (для внутренней команды, внешнего поставщика, потребителей)

ПОЛЕЗНЫЕ ШТУКИ

Misconfiguration и недостаточный контроль за изменениями

Причины возникновения инцидентов:

- Незащищенные элементы хранения данных или контейнеры
- Чрезмерные разрешения
- Учетные данные по умолчанию и параметры конфигурации оставлены без изменений
- Стандартные средства управления безопасностью отключены

Влияние на бизнес

Влияние неверно настроенного элемента облачного сервиса на бизнес может быть серьезным в зависимости от характера неправильной конфигурации и того, насколько быстро она обнаруживается и устраняется. Наиболее частый эффект - это **раскрытие данных**, хранящихся в облачных репозиториях и ФИР.

Ответственность

- ✅ Владелец услуги
- ❌ Поставщик облачных услуг
- ❌ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ❌ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ❌ Elevation of Privilege

Отсутствие паттерн безопасной архитектуры облака

Причины возникновения инцидентов:

Одна из самых больших проблем во время перехода в облака - реализация соответствующей архитектуры безопасности, способной противостоять современным кибератакам и угрозам, характерным для облачной инфраструктуры.

Влияние на бизнес

Независимо от размера предприятия, правильная архитектура и стратегия безопасности являются необходимыми элементами для безопасного перемещения, развертывания и работы в облаке. Успешные кибератаки могут иметь серьезные последствия для бизнеса, включая финансовые потери, репутационный ущерб, юридические последствия и штрафы.

Ответственность

- ✅ Владелец услуги
- ❌ Поставщик облачных услуг
- ❌ Оба

Сервис модель

- ❌ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ✅ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ✅ Elevation of Privilege

Недостаточный уровень контроля IAM
Key Managment

Причины возникновения инцидентов:

Неадекватная защита учетных данных

Отсутствие регулярной автоматической ротации криптографических ключей, паролей и сертификатов

Отсутствие масштабируемых систем управления идентификацией, учетными данными и доступом.

Невозможность использования многофакторной аутентификации.

Отсутствие надежных паролей.

Влияние на бизнес

Злоумышленники, маскирующиеся под законных пользователей, администраторов или разработчиков, могут:

- Читать, изменять и удалять данные
- Получить доступ к управлению и изменению инфраструктурой
- Перехватывать данные
- Релизить вредоносное программное обеспечение, под видом легитимного.

Ответственность

- ✅ Владелец услуги
- ❌ Поставщик облачных услуг
- ❌ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ❌ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ❌ Elevation of Privilege

Взлом аккаунта

Влияние на бизнес

Захват учетной записи подразумевает полный доступ: контроль над учетной записью, ее службами и данными внутри.

В таком сценарии бизнес-логика, функции, данные и приложения, зависящие от учетной записи, подвергаются риску.

Последствия подверженности риску угона аккаунта имеют очень серьезное влияние на бизнес.

В большинстве случаев угона аккаунта имели место значительные сбои в работе и остановке бизнес-процессов, включая примеры полного уничтожения активов и данных организации.

Последствия кражи аккаунта включают риск утечки данных, который приводит к репутационному ущербу, снижению стоимости бренда, юридической ответственности, раскрытию персональных данных, конфиденциальной и служебной информации.

Ответственность

- ✅ Владелец услуги
- ✅ Поставщик облачных услуг
- ✅ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ✅ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ✅ Elevation of Privilege

Риск утечки

Влияние на бизнес

Внутренние угрозы могут привести к потере конфиденциальной информации и интеллектуальной собственности. Простои системы, связанные с атаками, могут негативно сказаться на производительности компании. Кроме того, потеря данных или другой вред клиентам могут снизить доверие к услугам компании.

Работа с инцидентами внутренней безопасности включает локализацию, устранение последствий, реагирование на инциденты, расследование, анализ после инцидентов, эскалацию, мониторинг и наблюдение. Эти действия могут значительно увеличить рабочую нагрузку компании и увеличить бюджет безопасности.

Ответственность

- ✅ Владелец услуги
- ❌ Поставщик облачных услуг
- ❌ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ✅ Spoofing Identity
- ✅ Tampering with Data
- ❌ Repudiation
- ✅ Information Disclosure
- ❌ Denial of Service
- ✅ Elevation of Privilege

Небезопасные интерфейсы интеграции (API)

Влияние на бизнес

Несмотря на то что большинство провайдеров стремятся обеспечить, интеграцию безопасности в их модели сервисов, для потребителей этих сервисов критически важно понимать последствия безопасности, связанные с использованием, управлением, оркестровкой и мониторингом облачных сервисов.

Использование не безопасно настроенных интерфейсов и API ставит организации перед множеством проблем, связанных с конфиденциальностью, целостностью, доступностью и невозможностью отказа от совершенного действия.

Ответственность

- ❌ Владелец услуги
- ❌ Поставщик облачных услуг
- ✅ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ❌ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ✅ Elevation of Privilege

Непрозрачность использования облака

Влияние на бизнес

• Отсутствие управления: когда сотрудники не знакомы или не обучены надлежащему контролю доступа и управления в облачных сервисах, часто можно увидеть конфиденциальные корпоративные данные, размещенные в общем доступе, без учета требований защиты от НСД.

• Неправильно настроенные функции безопасности:
когда сотрудник неправильно настраивает облачный сервис, и он может стать доступным не только для компании, но и для злоумышленника, который в свою очередь может внедрить в сервисы вредоносные программы, бот-сети, вредоносное ПО для майнинга криптовалют и многое другое, что подставит под угрозу контейнеры и среду оркестрации.

Ответственность

- ❌ Владелец услуги
- ❌ Поставщик облачных услуг
- ✅ Оба

Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

Угрозы

- ✅ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ✅ Elevation of Privilege

Злоупотребление и неправомерное использование облачными сервисами

Примеры злоупотребления или неправомерного использования облачными сервисами:

- Запуск DDoS-атак.
- Спам по электронной почте и фишинговые кампании.
- «Майнинг» цифровой валюты.
- Крупномасштабное автоматизированное мошенничество с кликами.
- Атаки методом перебора украденных баз учетных данных.
- Размещение вредоносного или пиратского контента.0

Влияние на бизнес

Если злоумышленник скомпрометировал уровень управления облачной инфраструктурой или CI/CD, то он может использовать облачную инфраструктуру в незаконных целях таких как добыча криптовалюты или в качестве альтернативы злоумышленники также могут использовать облако для хранения и распространения вредоносных программ или фишинговых атак.

Ответственность

- ❌ Владелец услуги
- ❌ Поставщик облачных услуг
- ✅ Оба

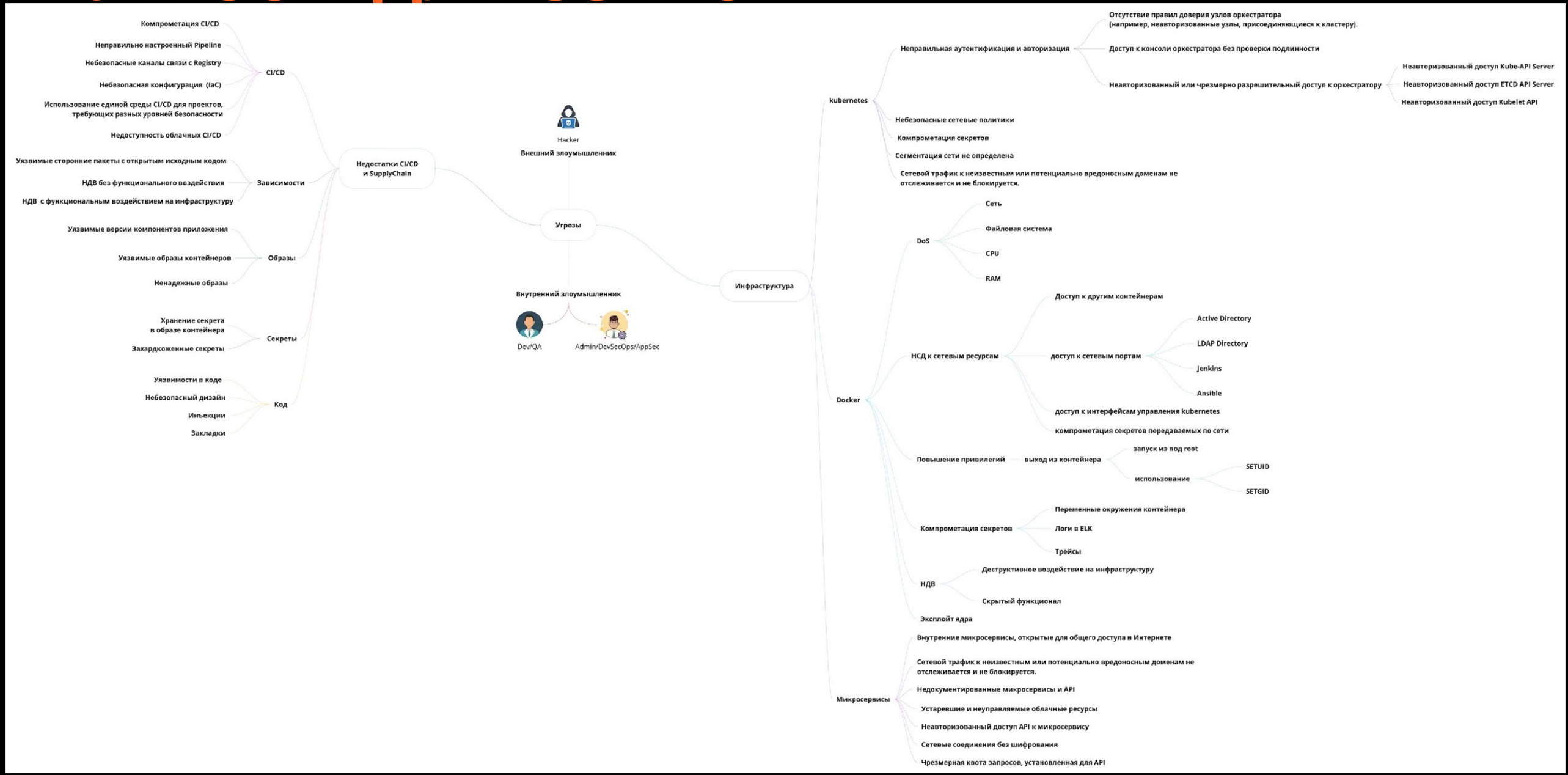
Сервис модель

- ✅ Software as a service (SaaS)
- ✅ Platform as a service (PaaS)
- ✅ Infrastructure as a service (IaaS)

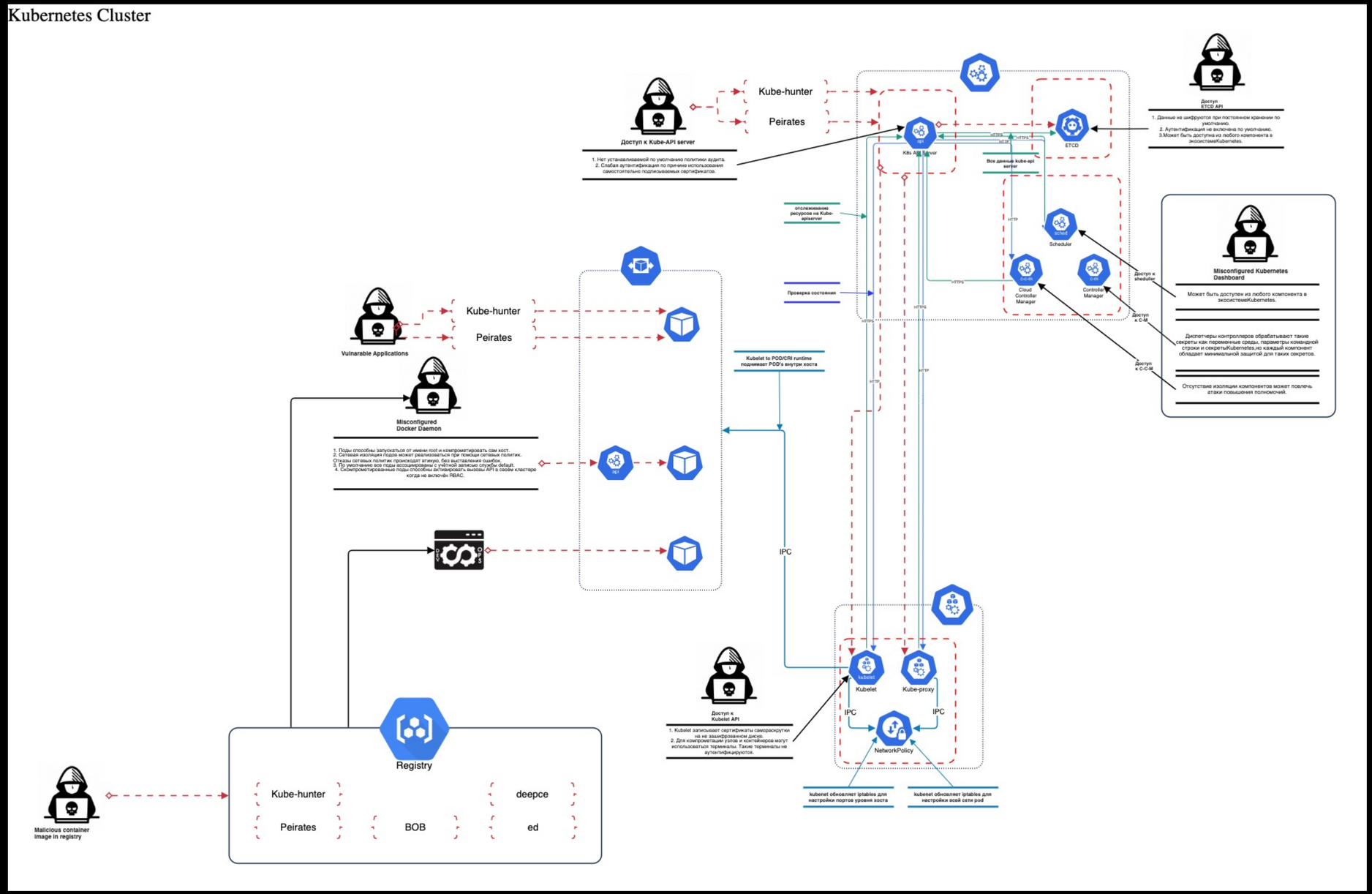
Угрозы

- ✅ Spoofing Identity
- ✅ Tampering with Data
- ✅ Repudiation
- ✅ Information Disclosure
- ✅ Denial of Service
- ✅ Elevation of Privilege

УГРОЗЫ ДЛЯ SSDLC



УГРОЗЫ ДЛЯ K8S



УГРОЗЫ ДЛЯ DOCKER