Безопасная разработка, как сделать первые шаги

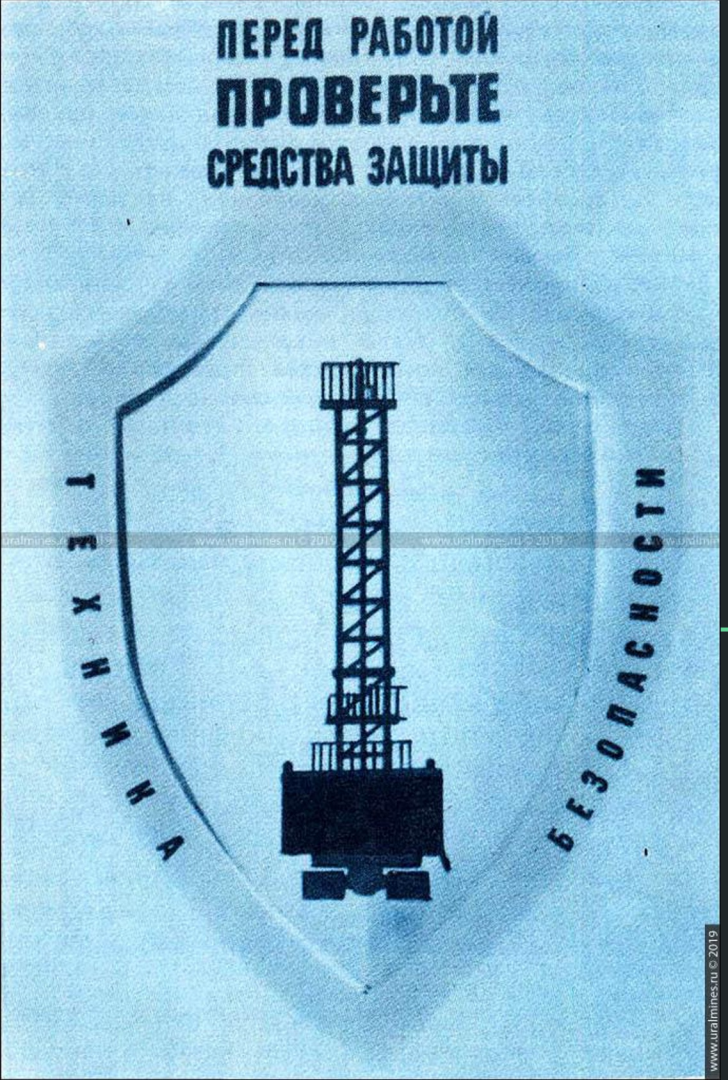
Гибкий и безопасный Software Development Life Cycle

Александр Киверин, CTO

12+ лет в ИТ-менеджменте
B2G, B2B, B2C продукты FinTech, E-commerce, HealthCare, Oil&Gas
«Ак Барс» Банк, «Газпром нефть»

О чем поговорим

- Зачем защищаться?
- От чего защищаться?
- Secure Software Development Lifecycle
- Что такое OWASP SAMM
- Пример внедрения SSDLC



Техника безопасности

Зачем и от чего защищаться?

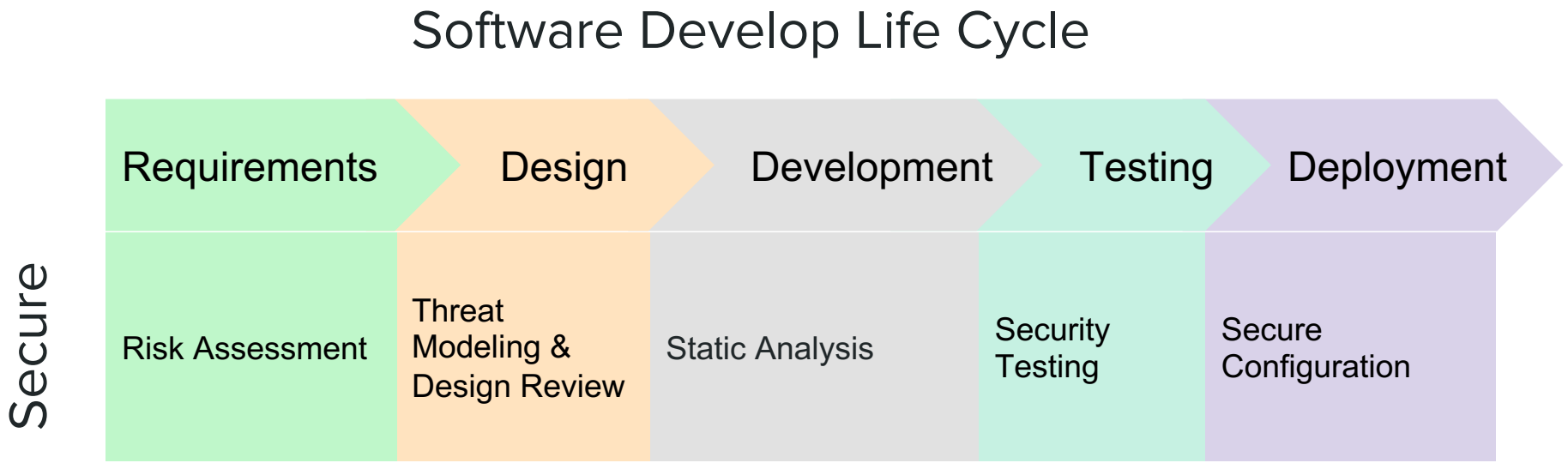
Косяки

- Инъекции (SQL, LDAP, XPath) и Cross-Site Scripting (XSS)
- Слабая криптография
- Небезопасная передача информации
- Некорректная обработка ошибок
- Раскрытие информации
- Cross Site Request Forgery (CSRF)
- Ошибки в контроле доступов
- Некорректная аутентификация и управление сеансами
- Небезопасные компоненты третьих лиц

SSDLC

Думайте о безопасности на всех этапах жизненного цикла разработки ПО, с самого начала

SSDLC?


Software Develop Life Cycle
Requirements
Design
Development
Testing
Deployment
Secure
Risk Assessment
Threat Modeling & Design Review
Static Analysis
Security Testing
Secure Configuration

OWASP SAMM

Систематизация по Software Assurance Maturity Model

Что такое OWASP?

Open Web Application Security Project (OWASP) – это открытый проект обеспечения безопасности веб-приложений.

OWASP top 10

OWASP top 10
API Security

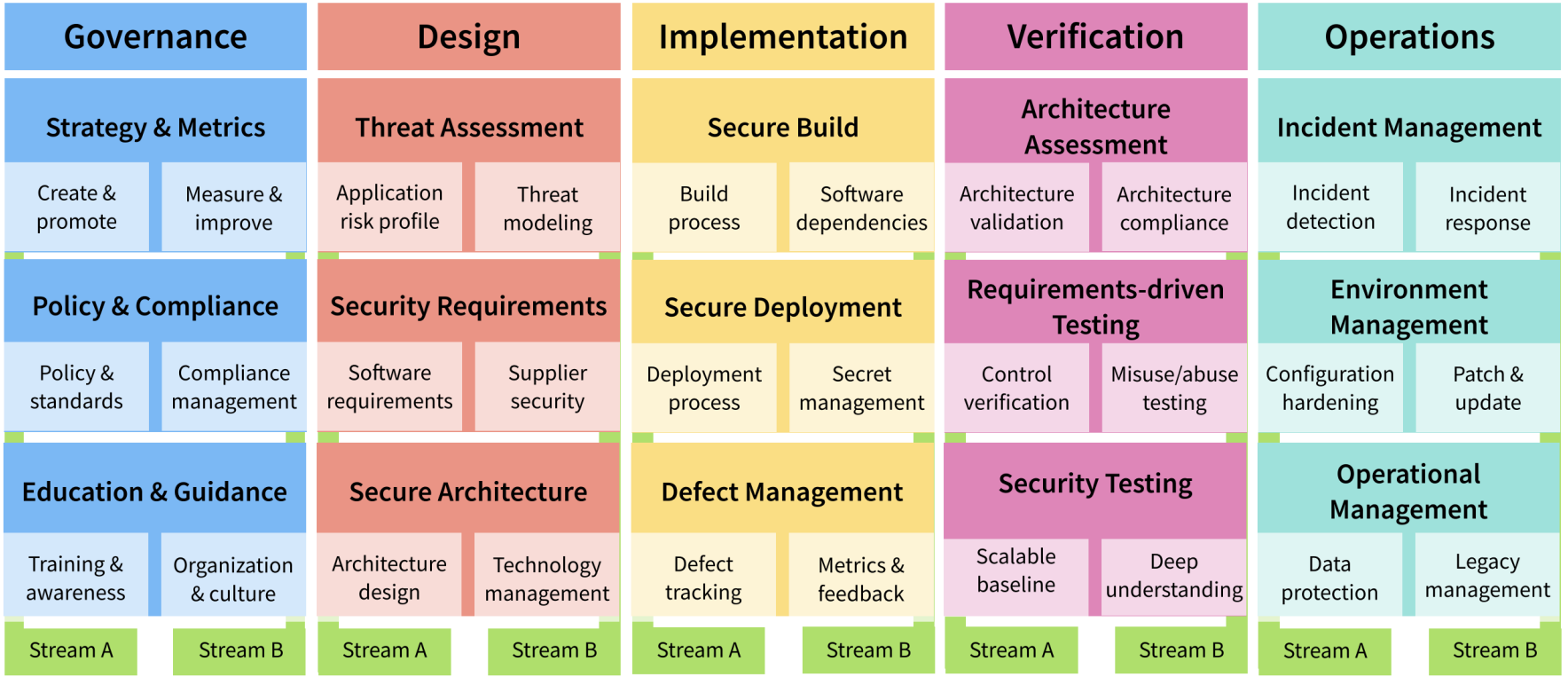
OWASP top 10
Proactive Controls

OWASP
cheat sheet

OWASP ASVS

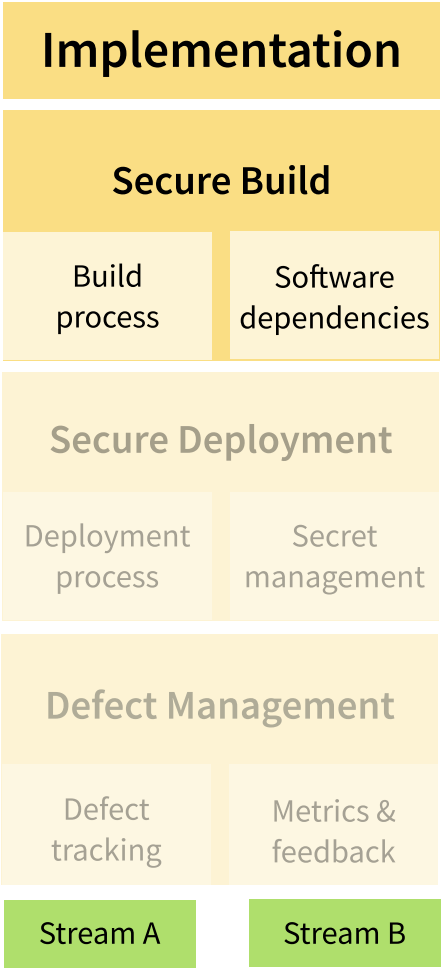
OWASP SAMM

OWASP SAMM


Governance
Strategy & Metrics
Create & promote
Measure & improve
Policy & Compliance
Policy & standards
Compliance management
Education & Guidance
Training & awareness
Organization & culture
Stream A
Stream B
Design
Threat Assessment
Application risk profile
Threat modeling
Security Requirements
Software requirements
Supplier security
Secure Architecture
Architecture design
Technology management
Stream A
Stream B
Implementation
Secure Build
Build process
Software dependencies
Secure Deployment
Deployment process
Secret management
Defect Management
Defect tracking
Metrics & feedback
Stream A
Stream B
Verification
Architecture Assessment
Architecture validation
Architecture compliance
Requirements-driven Testing
Control verification
Misuse/abuse testing
Security Testing
Scalable baseline
Deep understanding
Stream A
Stream B
Operations
Incident Management
Incident detection
Incident response
Environment Management
Configuration hardening
Patch & update
Operational Management
Data protection
Legacy management
Stream A
Stream B

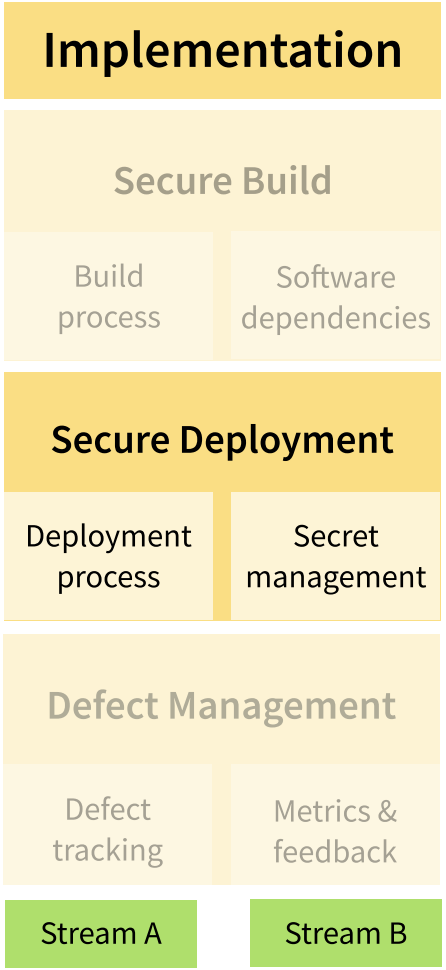
OWASP SAMM

1. Процесс сборки является повторяемым и последовательным.
2. Процесс сборки оптимизирован и полностью интегрирован в рабочий процесс.
3. Процесс сборки позволяет предотвратить попадание известных дефектов в производственный контур.



OWASP SAMM

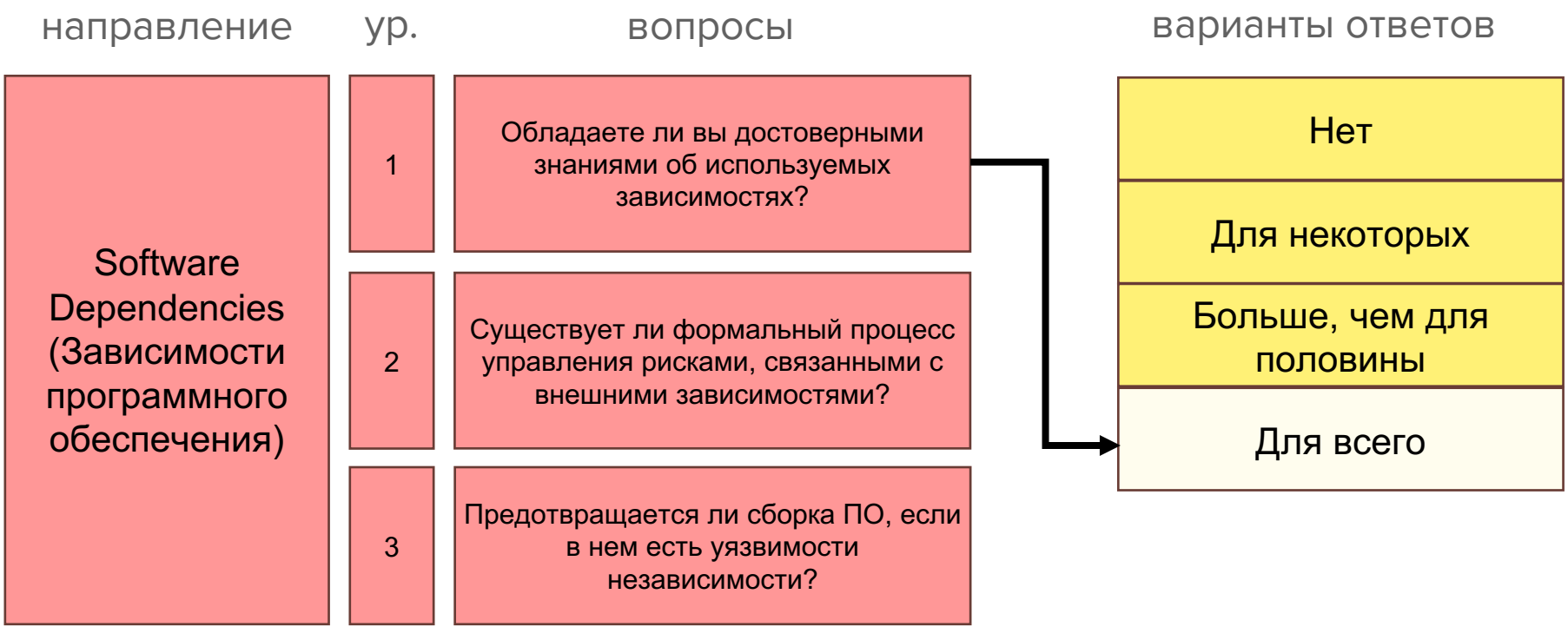
1. Процессы развертывания полностью документированы.
2. Процессы развертывания включают этапы проверки безопасности.
3. Процесс развертывания полностью автоматизирован и включает в себя автоматическую проверку всех важных этапов.



OWASP SAMM

1. Все дефекты отслеживаются в рамках каждого проекта.
2. Отслеживание дефектов позволяет влиять на процесс развертывания.
3. Отслеживание дефектов по нескольким источникам позволяет сократить количество новых дефектов.

Оценка начальной зрелости



Сборка с учетом требований безопасности

1

Stream A

Создать формальное определение процесса сборки, чтобы он стал последовательным и повторяемым.

Stream B

Создайте SBOM ваших приложений и проводите их выборочный анализ.

Сборка с учетом требований безопасности

Stream A

1 Создать формальное определение процесса сборки, чтобы он стал последовательным и повторяемым.

2 Добавление проверок безопасности в конвейер сборки.

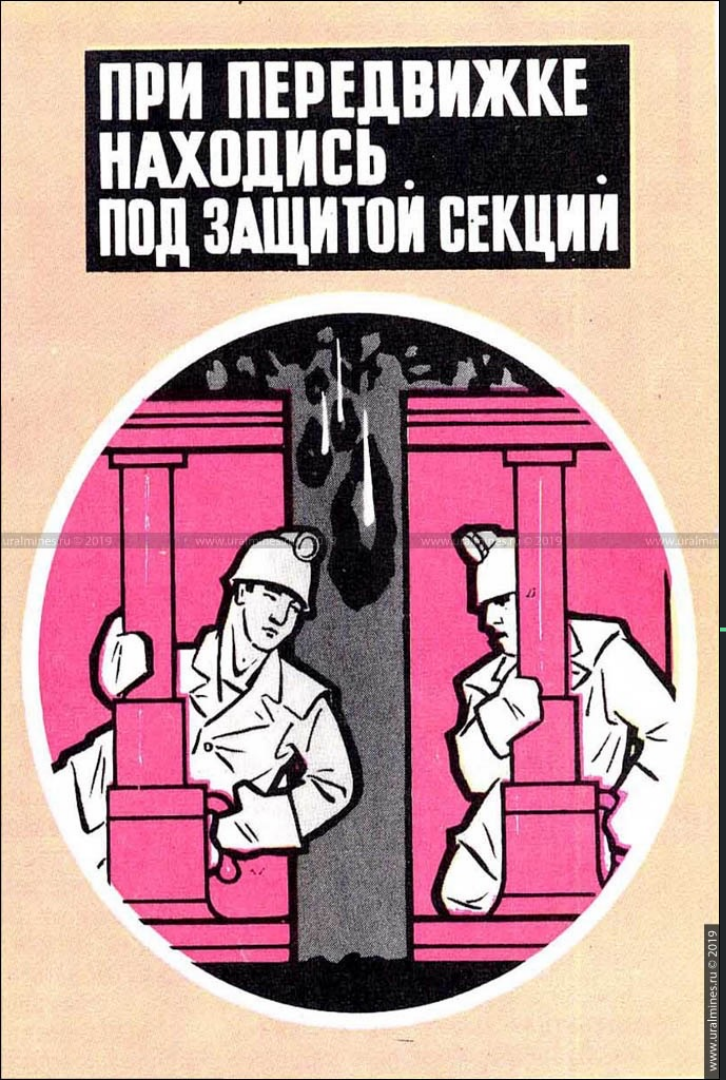
Stream B

Создайте SBOM ваших приложений и проводите их выборочный анализ.

Оценка используемых зависимостей и своевременное реагирование на ситуации, представляющие риск для ваших приложений.

Сборка с учетом требований безопасности

	Stream A	Stream B
1	Создать формальное определение процесса сборки, чтобы он стал последовательным и повторяемым.	Создайте SBOM ваших приложений и проводите их выборочный анализ.
2	Добавление проверок безопасности в конвейер сборки.	Оценка используемых зависимостей и своевременное реагирование на ситуации, представляющие риск для ваших приложений.
3	Определите обязательные проверки безопасности в процессе сборки и заблокируйте сборку артефактов, не соответствующих требованиям.	Внедрить структурированное отслеживание дефектов безопасности и принимать на основе этой информации грамотные решения.



Внедрение SSDLC

SSDLC в гибкой итеративной разработке

SSDLC и итерации

Консультация

Проработка бэклога (PBR)

Разгребайте уязвимости по аналогии с багами и техдолгом каждую итерацию, занесите это в метрики команды

Проект* [redacted]

Тип запроса* Уязвимость

Основное | Дополнительно

Тема* XSS

Приоритет Средний

Критичность* Medium

Уровни серьезности по стандартам ИБ

Риск*
* Medium = 4.7
* CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N = 5.4
* BIS-IMP:1.0/FD:M/RD:M/NC:M/PV:HT = 4

Описание риска

Исполнитель Автоматически

Назначить мне

Описание

Стиль B I U A ªA

В функционале [redacted] есть возможность сохранить исполняемый код.

Визуальный | Текстовый

Шаги воспроизведения

Стиль B I U A ªA

В поле "Name" ввести

```
<svg onload=alert(1)>
```

Визуальный | Текстовый

Компоненты **Нет**

Вложение Перетащите файлы, чтобы прикрепить, или выберите с диска.

Метки

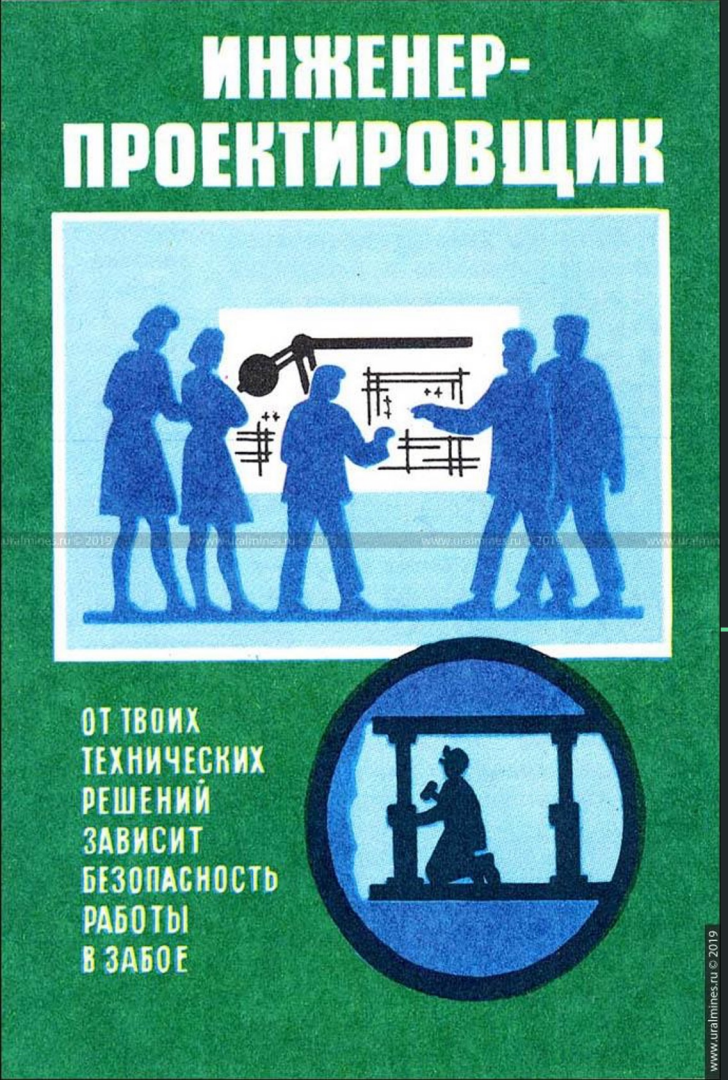
Начните вводить для поиска и создания меток или нажмите вниз, чтобы выбрать предложенную метку.

Рекомендация*

Стиль B I U A ªA

- Фильтрация входных данных от пользователей.
- [redacted]/security/XSS

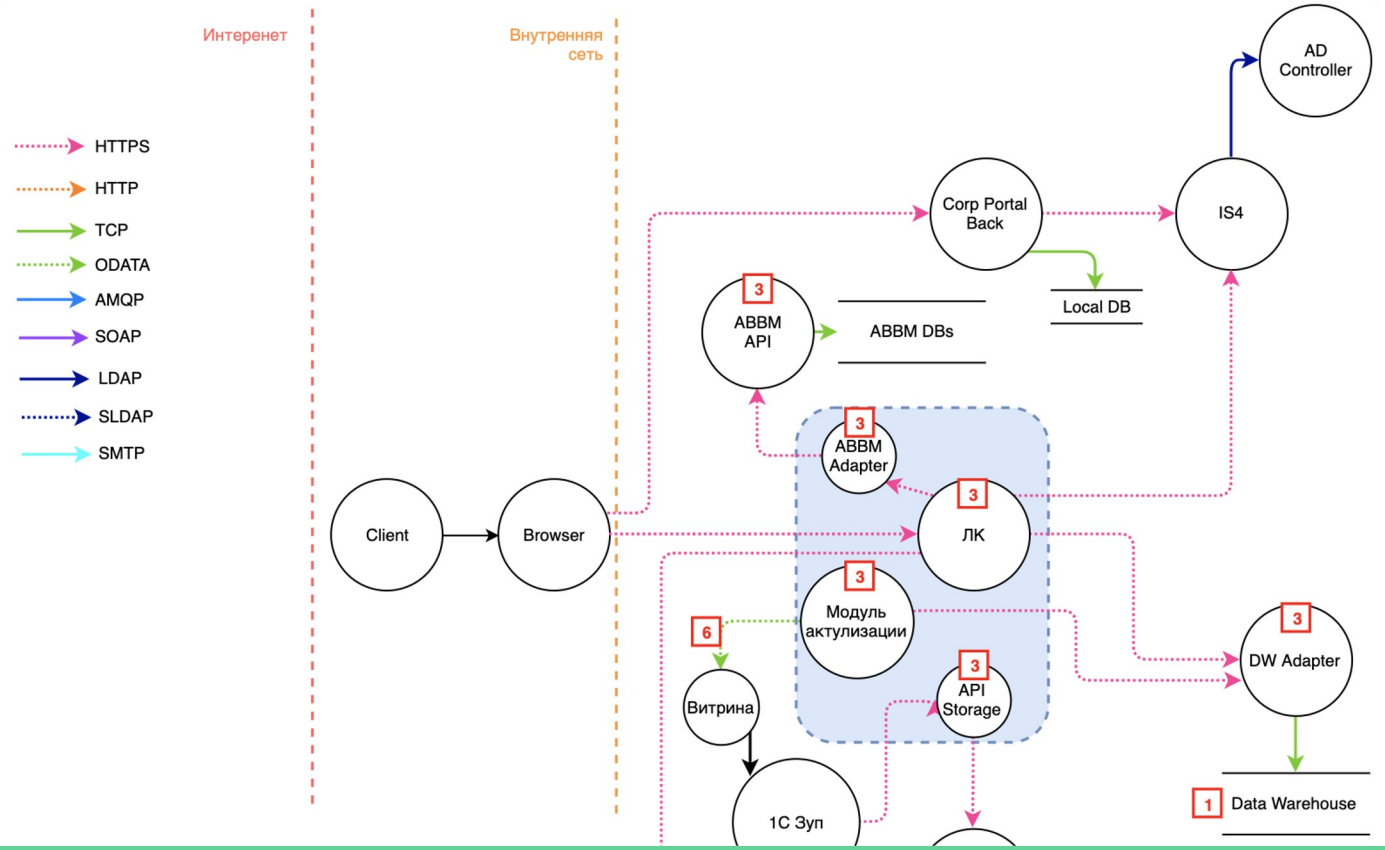
Визуальный | Текстовый



Анализ

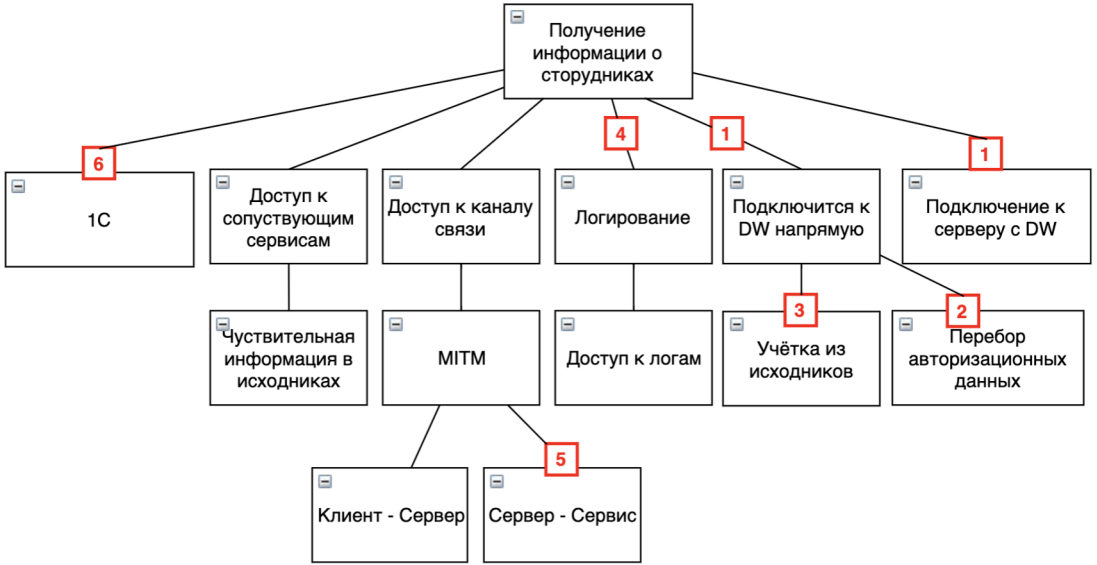
Анализируйте риски при проектировании решений и при проработке функциональных и нефункциональных требований

Модель угроз



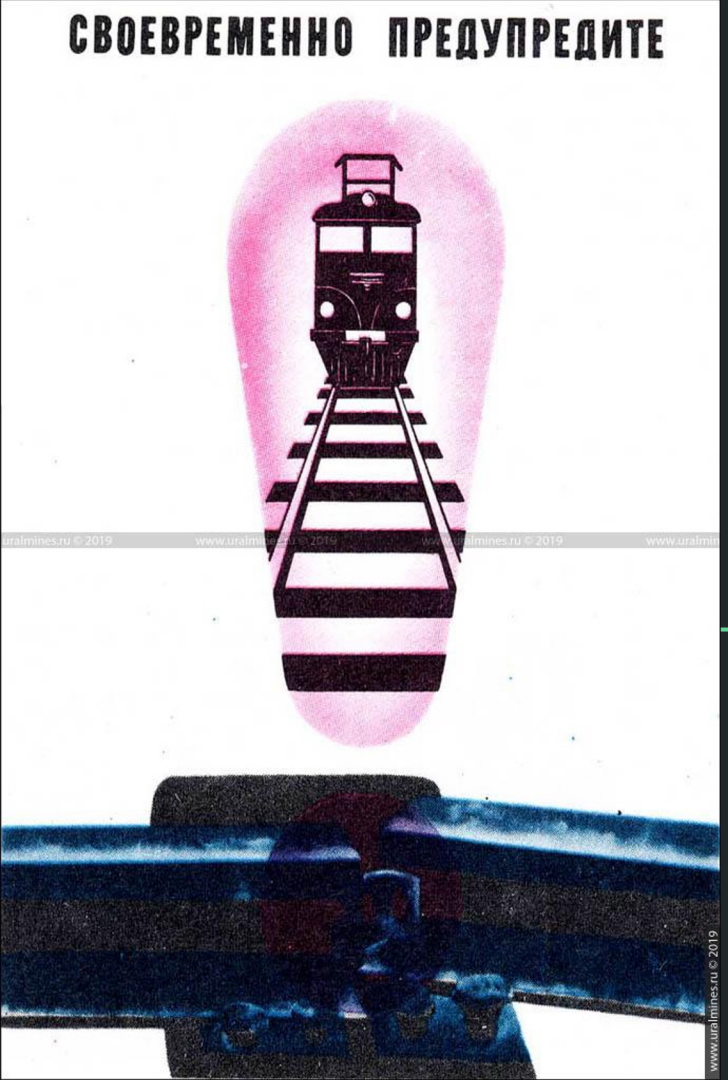
Угрозы и векторы атак

Дерево угроз



Меры безопасности

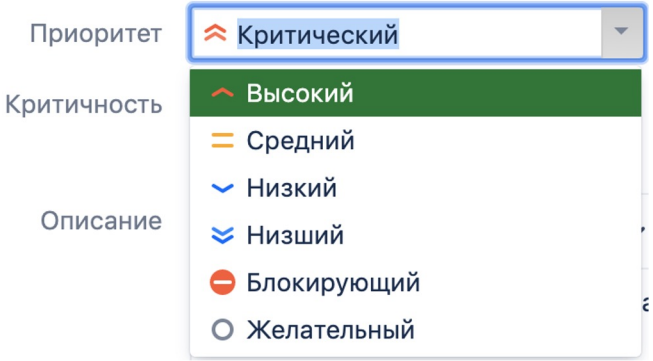
- ✓ 1 Доступ к серверу с DW через AD
- ✗ 2 Сложные учётные данные для доступа в DW
  Соответствие ГОСТ Р 57580.1-2017
- ✓ 3 SAST
- ✓ 4 Логирование только необходимой информации.
  Только AD учетки. + Хранение на локальном сервере
- ✓ 5 HTTPS внутри контура
- ✓ 6 Исключение прямого доступа до 1С



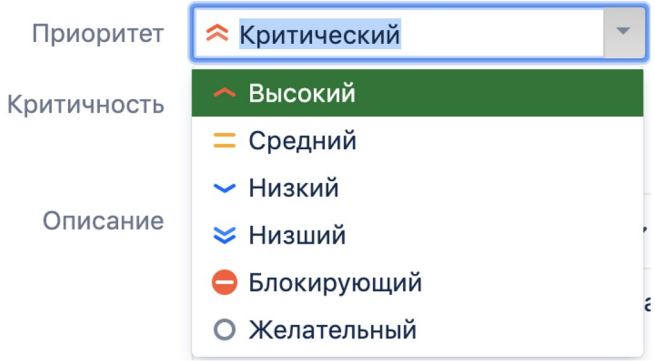
Планирование итерации

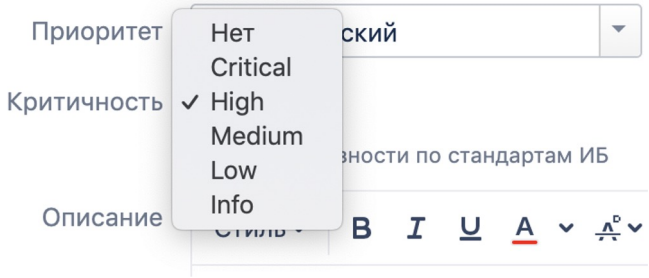
Поднимайте приоритеты для критичных уязвимостей, “продавайте” технические задачи

Priority (приоритет)



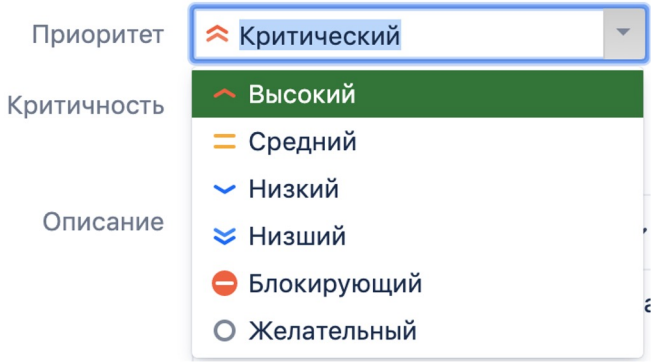
Priority (приоритет), Severity (серьезность)

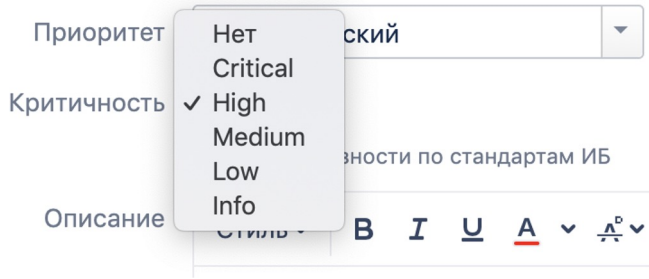


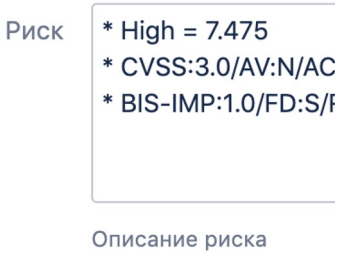


Критическая	3 дня
Высокая	14 дней
Средняя	45 дней
Низкая	90 дней
Инфо	>90 дней

Priority (приоритет), Severity (серьезность), Risk (риск)







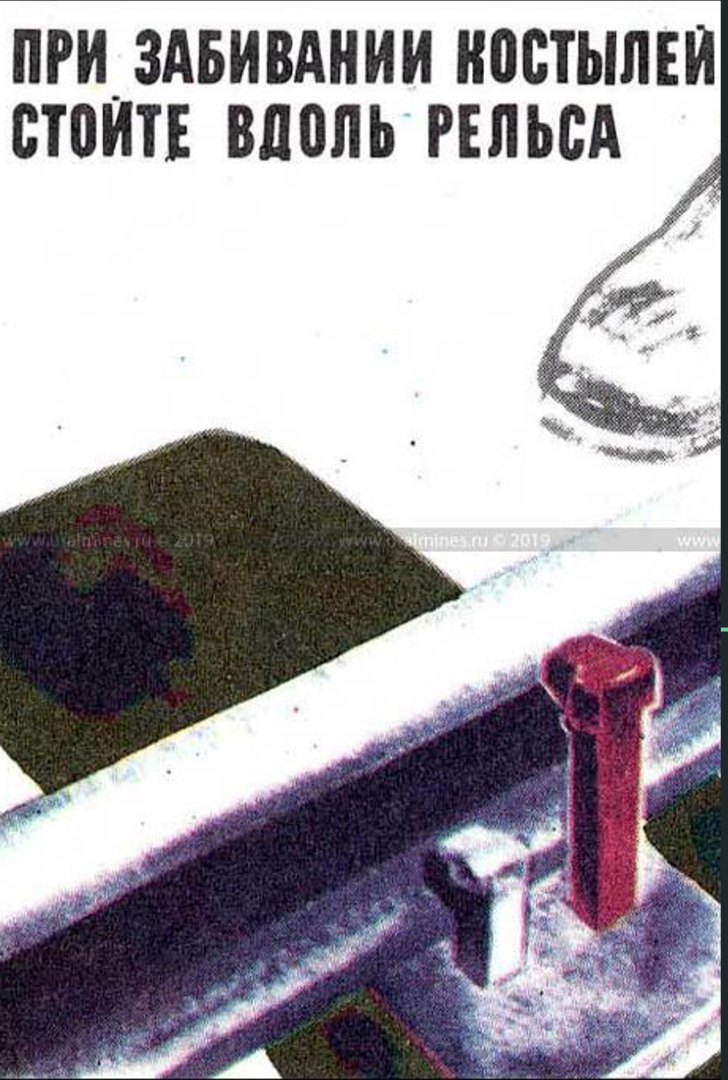
Risk = Severity score = (Application score + Business Impact)/2

Application score

= InCode Severity score или Nessus Severity score или SonarQube Severity score

Business Impact

= (Financial Damage + Reputation Damage + Non-Compliance + Privacy Violation)/4



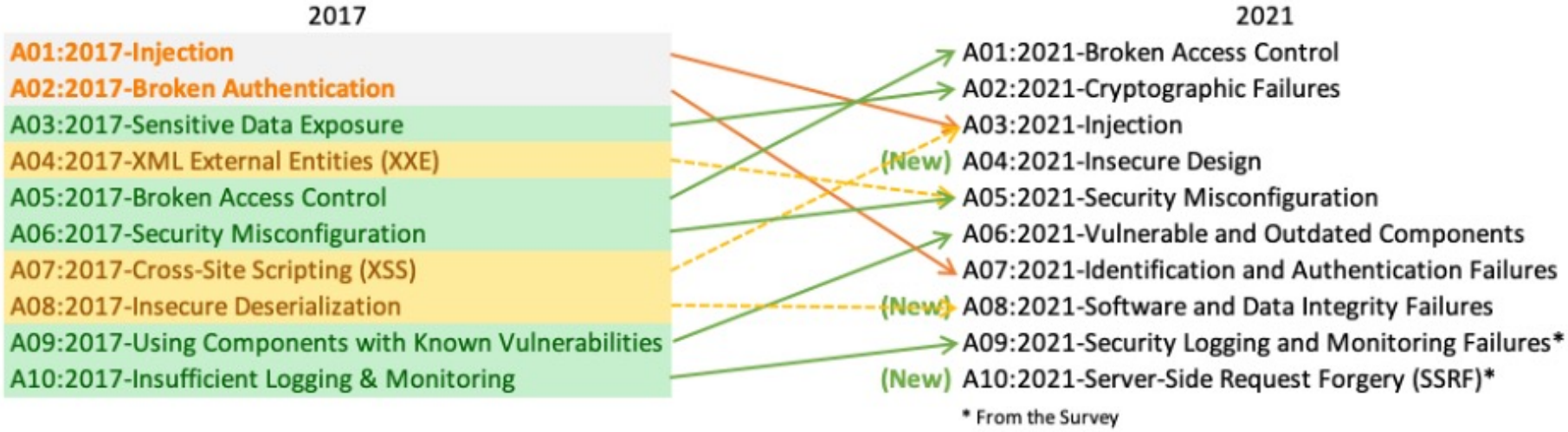
Разработка

Следите за всеми временными решениями в приложении, смотрите на все типы угроз по тому же OWASP

OWASP Top Ten

(A1) Broken Access Control
(A2) Cryptographic Failures
(A3) Injection
(A4) Insecure Design
(A5) Security Misconfiguration
(A6) Vulnerable and Outdated Components
(A7) Identification and Authentication Failures
(A8) Software and Data Integrity Failures
(A9) Security Logging and Monitoring Failures
(A10) Server-Side Request Forgery (SSRF)

OWASP Top Ten



(A1) Нарушенный контроль доступа

- Механизмы контроля доступа
- CORS
- Инвалидация сессии (для JWT-токенов)
- Логирование нарушенного контроля доступа
- Rate limit

(A3) Инъекции и Cross-Site Scripting (XSS)

- Файрволы веб-приложений (WAF)
- Объектно-реляционное преобразование (ORM)
- Валидация пользовательского ввода
- Экранирование данных
- Параметризованные запросы SQL
- Content Security Policy (CSP)
- X-XSS-Protection
- Cookies: HTTP-only, Secure, SameSite
- Библиотеки / фреймворки (например, Microsoft AntiXSS)

(A5) Ошибки в конфигурировании

- Удалять тестовый функционал в релизе
- Отключать неиспользуемые компоненты
- Отключать неиспользуемую функциональность

(A06) Vulnerable and Outdated Components

- Удалять неподдерживаемые, уязвимые, старые компоненты
- Подписаться на уведомления об уязвимостях
- Проверять версии сторонних компонентов
- Удалять неиспользуемые компоненты, файлы

- Использовать локальные репозитории (Artifactory, Nexus)
- Регулярно сканировать репозитория на уязвимости
- Регулярно обновлять библиотеки и фреймворки или фризить апдейты
- Обновляться только с официальных источников + тщательное ревью

(A7) Некорректная аутентификация

- Алгоритм хеширования не слабее SHA256
- Логирование неудачной аутентификации
- Сессионный токен
- Защита от перебора
- Сложность вводимого пароля
- 2FA

Небезопасное
(A9) ~~Недостаточное~~ логирование

- PAN, CVV / CVC
- PIN карты
- Пароли и OTP
- Токены и cookie
- ФИО, персональные данные
- Серия и номер паспорта

(A10) Server-Side Request Forgery (SSRF)

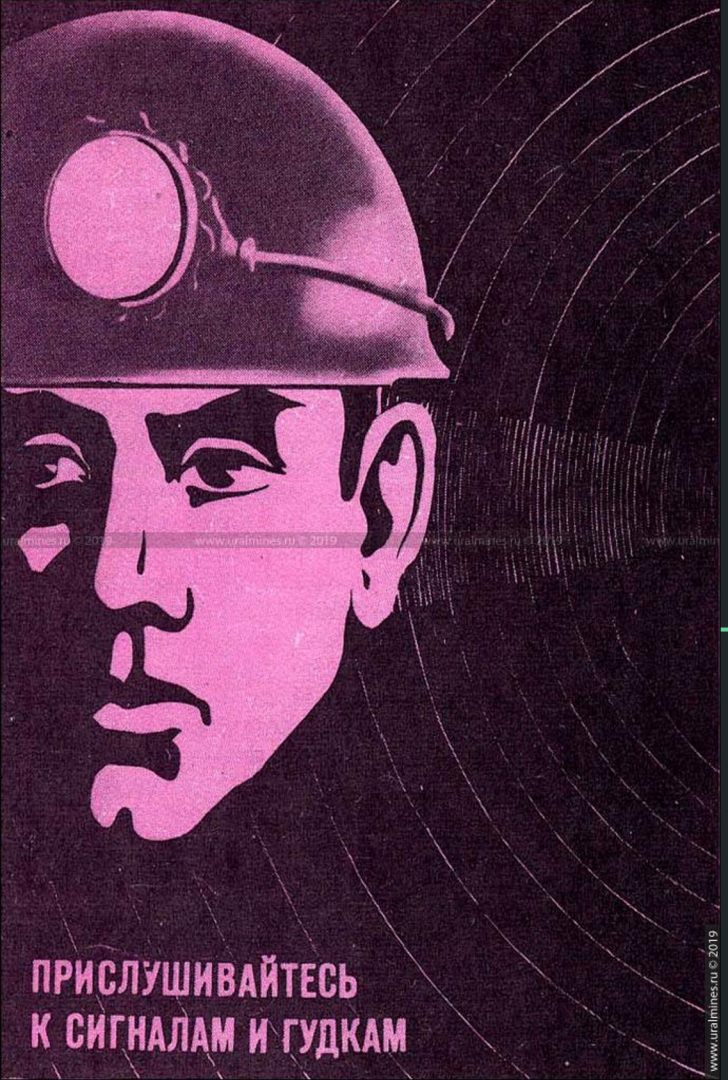
- Валидировать входные запросы (ограничить поддерживаемые схемы)
- Валидировать доменные имена
- Блокировать запросы на внутренние подсети
- Ввести межсервисную аутентификацию
- Отключить поддержку перенаправлений или проверять каждый шаг

Code Review

Ревьюйте код не только на соблюдения правил написания кода и архитектуры, а также на наличие потенциальных дыр

Такого быть не должно

```
if (!_cardService.IsUnique(request.Card.Number) && request.Name != "test") {
        return ErrorResponse("Возможно вы уже зарегистрированны в системе");
}
```



Тестирование и Feature Freeze

Встройте в проверки QA анализаторы SAST и DAST, следите за алертами по уязвимостям, атакам, логам

Проверка при сборке

Static Application Security Testing (SAST) – это анализ кода или его части на наличие уязвимостей без реального запуска исследуемого приложения.

Solar APPscreener. Поддержка большого количества языков. Самые востребованные: C#, JS, Java, Python.

SonarQube. Оценка качества кода. free

Dependency Check. Нахождение уязвимых зависимостей. free

Trivy. Проверка контейнеров на уязвимые компоненты. free

GitLeaks. Инструмент по поиску секретов. free

Проверка при деплое

Dynamic Application Security Testing (DAST) – это анализ на наличие уязвимостей выполняемого приложения. DAST делает тест черного ящика.

ZAP. Расширяемый, с большим количеством проверок. Автоматизируемый и вшиваемый в CI/CD. free

Burp Suite. Инструмент для ручного тестирования веб-приложений.

Nessus™

Nessus. Сканер уязвимостей. Альтернатива **OpenVAS** free

WFuzz

WFuzz. Инструмент для фаззинга веб-приложений. free

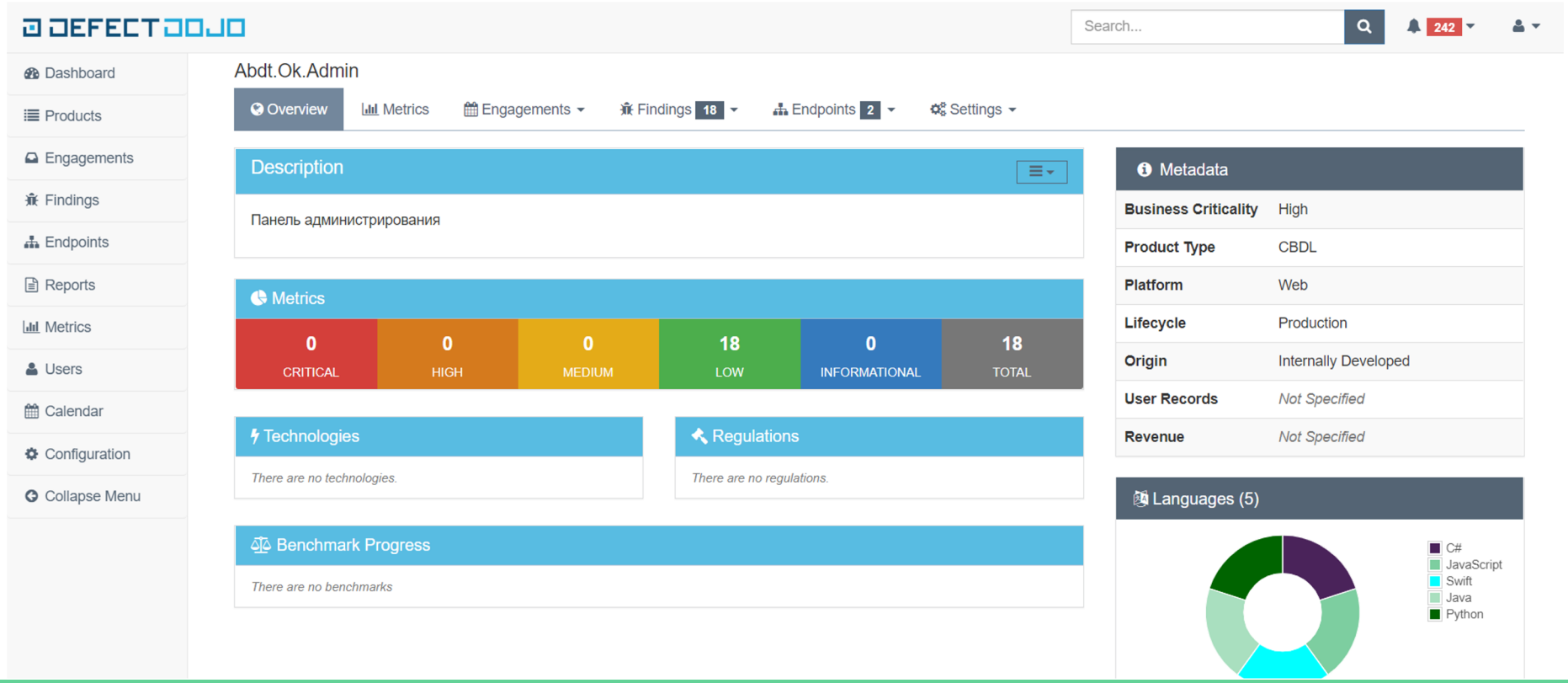
Готовность к релизу

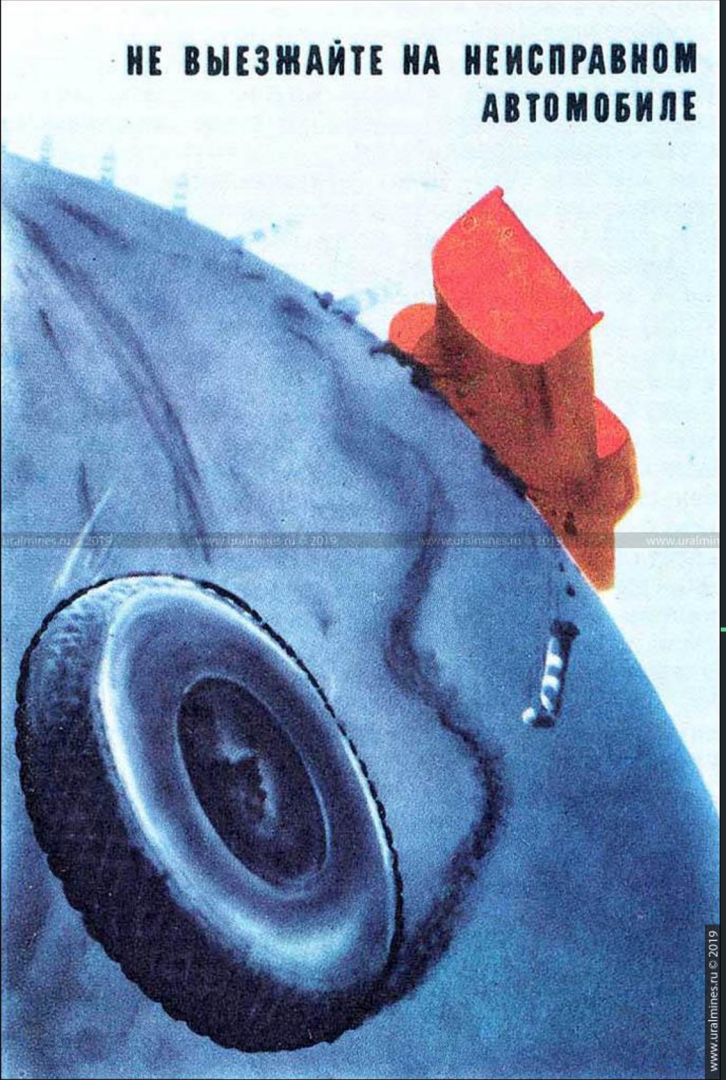
Релизьте только после апрува QA при отсутствии критичных дефектов и уязвимостей

Артефакты

Фиксируйте факт релиза, прикладывайте результаты проверки – у нас заявка с артефактами на безопасников

DefectDojo


DEFECTDOJO
Search...
242
Dashboard
Products
Engagements
Findings
Endpoints
Reports
Metrics
Users
Calendar
Configuration
Collapse Menu
Abdt.Ok.Admin
Overview
Metrics
Engagements
Findings 18
Endpoints 2
Settings
Description
Панель администрирования
Metrics
0 CRITICAL
0 HIGH
0 MEDIUM
18 LOW
0 INFORMATIONAL
18 TOTAL
Technologies
There are no technologies.
Regulations
There are no regulations.
Benchmark Progress
There are no benchmarks
Metadata
Business Criticality High
Product Type CBDL
Platform Web
Lifecycle Production
Origin Internally Developed
User Records Not Specified
Revenue Not Specified
Languages (5)
C#
JavaScript
Swift
Java
Python



Релиз

Релизьте только при отсутствии критичных дефектов и уязвимостей – дефекты S1/S2 с P1 являются блокерами для деплоя

Серьезность (Severity)

- S1 Блокирующая (Blocker)
- S2 Критическая (Critical)
- S3 Значительная (Major)
- S4 Незначительная (Minor)
- S5 Тривиальная (Trivial)

Серьезность (Severity) и Приоритет (Priority)

- S1 Блокирующая (Blocker)
- S2 Критическая (Critical)
- S3 Значительная (Major)
- S4 Незначительная (Minor)
- S5 Тривиальная (Trivial)

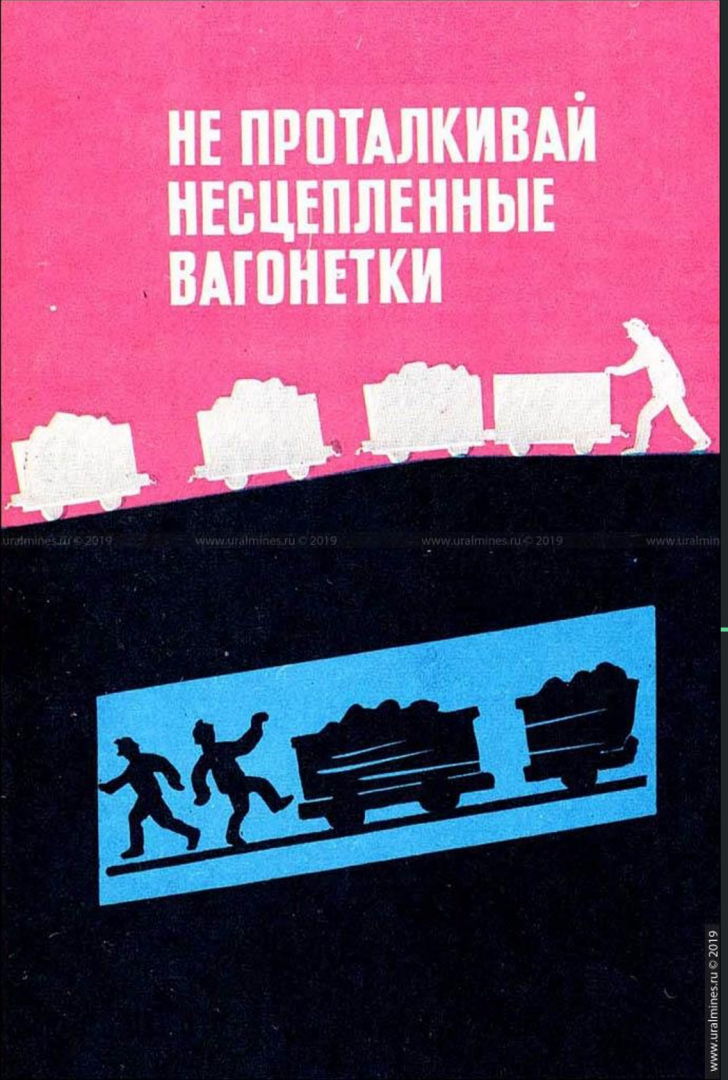
- P1 Highest
- P2 High
- P3 Medium
- P4 Low
- P5 Lowest

Серьезность (Severity) и Приоритет (Priority)

- S1 Блокирующая (Blocker)
- S2 Критическая (Critical)
- S3 Значительная (Major)
- S4 Незначительная (Minor)
- S5 Тривиальная (Trivial)

- P1 Highest
- P2 High
- P3 Medium
- P4 Low
- P5 Lowest

Наличие багов S1/S2 и P1 является блокером для деплоя версии на прод.

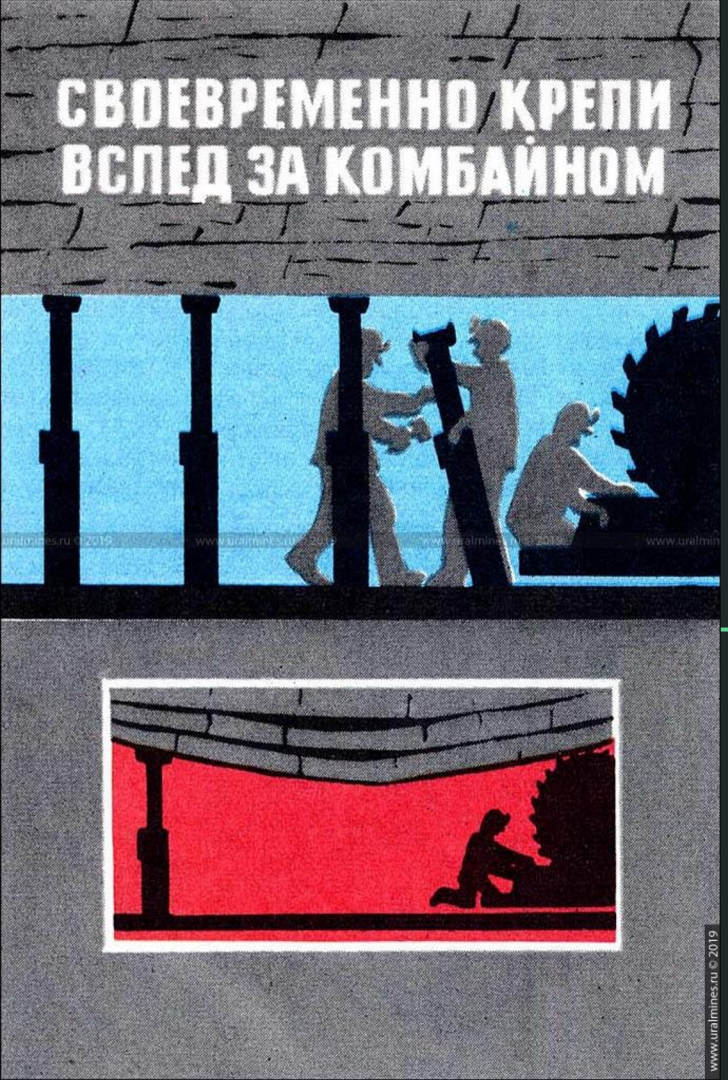


Автоматизация проверок

Проверяйте уязвимости в сборках и контейнерах, лучше с возможностью автоматически заблокировать релиз

CI/CD pipeline

Проверки не должны создавать простой в деплое, апрув заявок не должен создавать простой в запуске

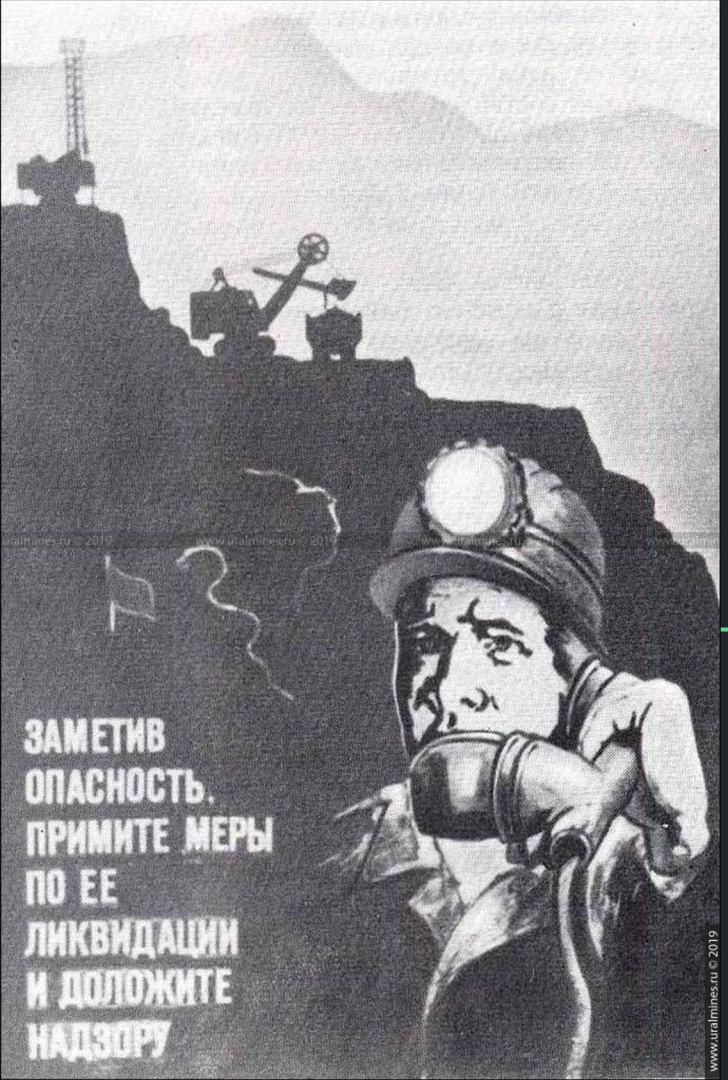


Архитектура и инфраструктура

В целом, следите, где и как размещать сервисы, какие контуры безопасности строить, делайте это своевременно

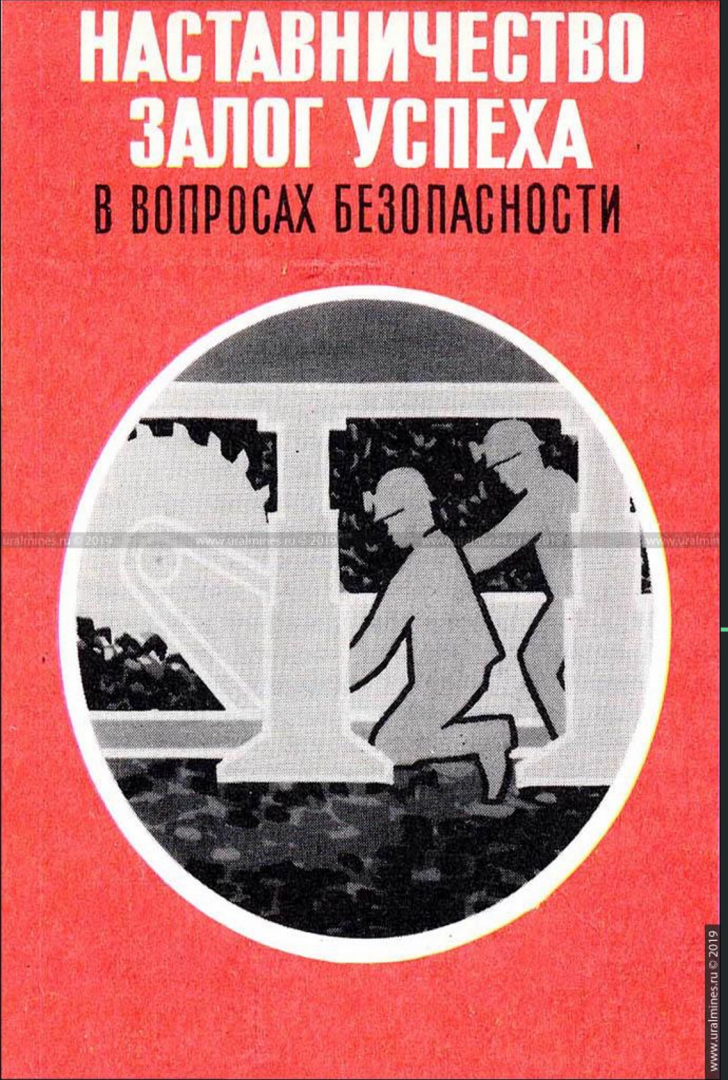
AppSec Engineers

Выстраивайте процесс, развивайте производственные практики. Круто, если есть AppSec инженеры.



Security Champions

Не обязательно AppSec инженер должен все делать, команда может иметь амбассадора. Security Champions.



Агитируем и обучаем

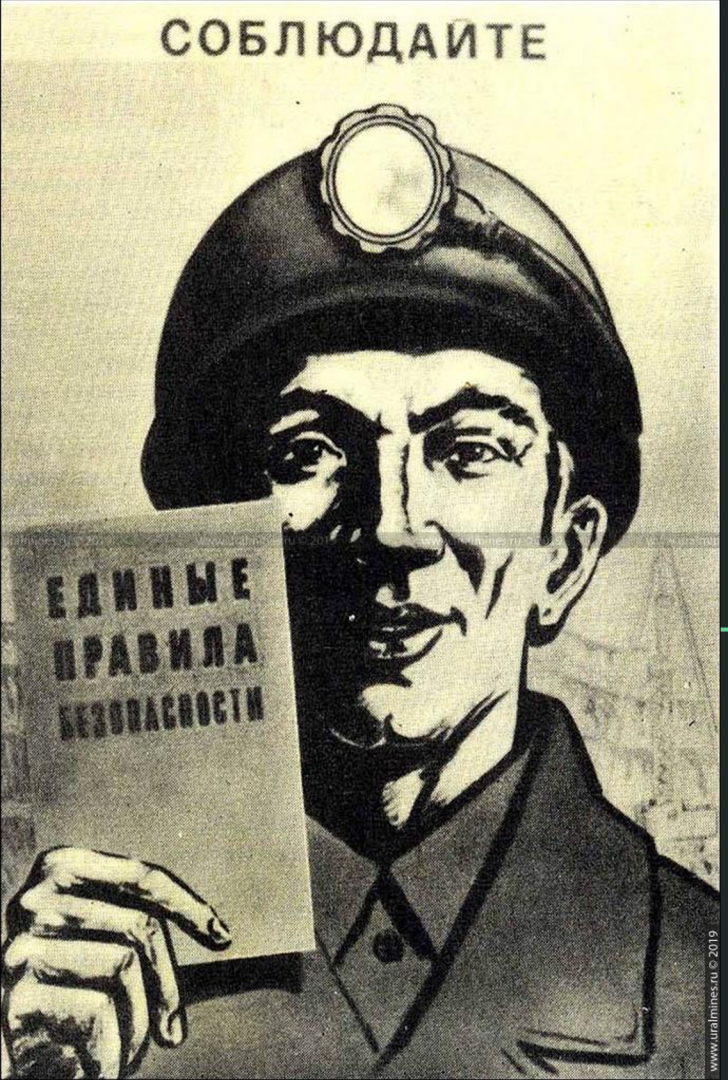
Выискивайте Security Champions из разработчиков, аналитиков или QA, обучайте, дайте требования и обязанности

Компетенции и знания Security Champion

- Организация SSDLC
- Законодательные и регуляторные требования безопасности
- OWASP Top 10 Web and Mobile
- Основные виды угроз ИБ, понимание рисков ИБ

Дополнительные навыки SecChamp

- Безопасная разработка – уязвимый код, способы решения
- Проведение тестов безопасности – ручное тестирование инъекций
- Проведение тестов безопасности – запуск сканеров безопасности
- Написание кейсов для фаззинга приложений
- Криптография – виды, лучшие практики, применение в приложениях
- Тестирование новых инструментов для тестирования безопасности



Пишем правила

Ведите knowledge base в confluence с правилами и требованиями, обязательное соблюдение командами

Правила

- Правила безопасной работы на ПК/ноутбуках и в сети
- Методология безопасной разработки (Web, Mobile)
- Криптография
- Как закрывать уязвимости в продукте
- Оценка уровня критичности уязвимости
- Создание тикета "Уязвимость" в JIRA
- Расчет метрик качества продукта со стороны безопасности
- Релизная политика, порядок деплоя сервисов в боевую среду

Инструкции и гайды

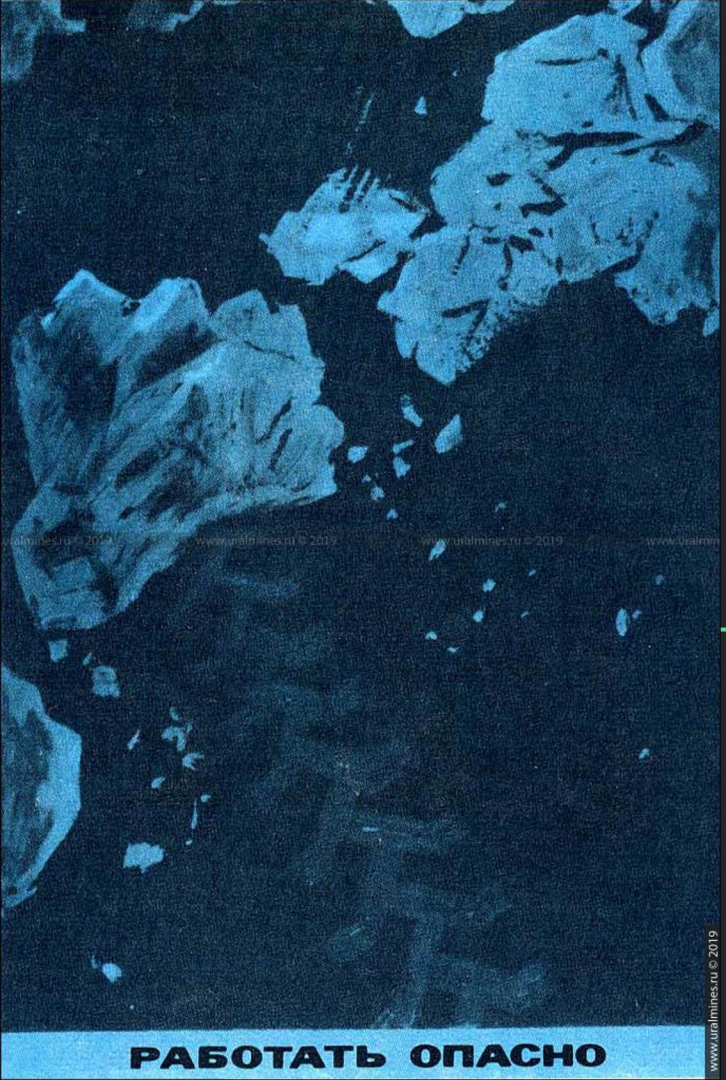
Пишите инструкции как для команды в целом, так и для Security Champions в частности

Инструкции и гайды

- Welcome guide
- Рекомендации к проведению контроля исходного кода
- Рекомендации к проведению оценки защищенности
- Требования безопасности к системам логирования
- Методика моделирования угроз
- Основные нормативно-правовые документы
- Внедренные системы безопасности и их настройка
- Обучающие материалы по информационной безопасности

Покрытие всех команд

Таким образом, покрываем все команды



Enterprise

В энтерпрайзе больше ответственности – работать опасно, поэтому уделяем безопасности много внимания

Касается всех

Но безопасность касается всех, любые приложения подвержены рискам и могут повлиять на потери компании

Защищайтесь и будьте здоровы!

Спасибо за внимание