



DevSecOps за 10 дней

Apsafe — безопасность, которая
не мешает разработке



Что сегодня обсудим

- ▶ Почему большинство команд страдает от DevSecOps
- ▶ Каким должен быть «идеальный» процесс
- ▶ Как Apsafe снимает ключевые боли
- ▶ Две реальные истории успеха
- ▶ Подробнее про Apsafe



Спикеры



Илья Новойдарский

Эксперт

Руководитель группы
инновационных решений
направления безопасной
разработки УЦСБ



Евгений Тодышев

Спикер

Руководитель
направления безопасной
разработки УЦСБ



Анастасия Камалова

Спикер

Пресейл инженер
направления безопасной
разработки УЦСБ



Виктор Тимашков

Спикер

AppSec-аналитик группы
инновационных решений
направления безопасной
разработки УЦСБ



эксперт в области DevSecOps

Илья Новойдарский

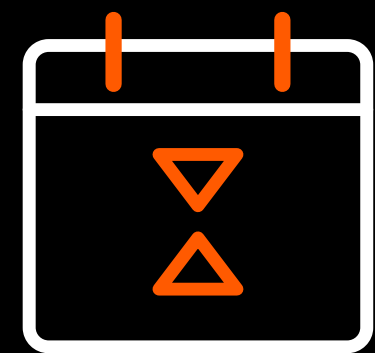
Руководитель группы инновационных решений
направления безопасной разработки УЦСБ



Почему команды страдают от DevSecOps



Долго, дорого сложно



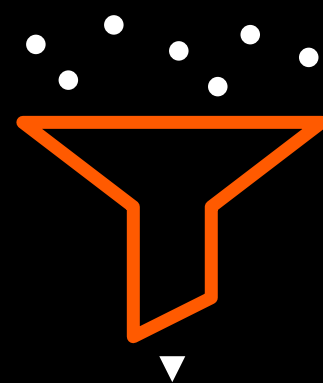
Долгое внедрение

3–6 месяцев на внедрение классических сканеров



Высокие затраты

Рост затрат пропорционален росту разработки



Ложные срабатывания

Сканеры генерируют тысячи ложных срабатываний за одно сканирование



Кадровый голод

Требуется команда из 4-х человек для работы процесса, включая редкого AppSec-аналитика



Проблема №1

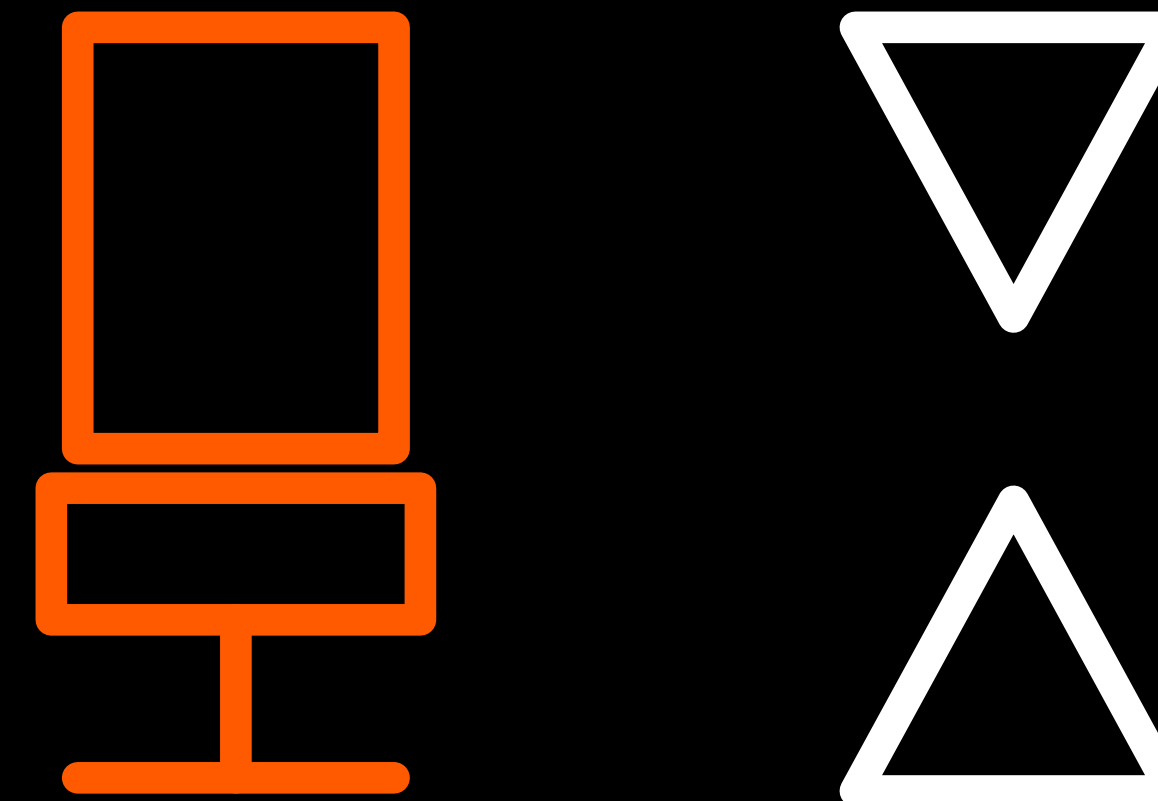
Нам некому и некогда

▶ Найм своей команды

От 4-х человек, срок найма от 6 месяцев, в среднем 6 млн руб. в год на человека

▶ Конкуренция за экспертов

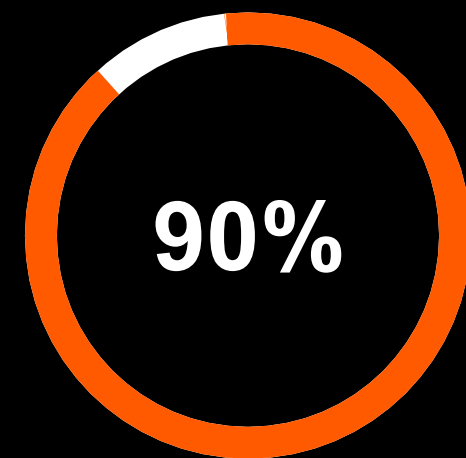
Опытные эксперты предпочитают работать в командах с уже высоким уровнем зрелости DevSecOps процессов, а их привлечение требует делать оффер на 30-50% выше текущего предложения



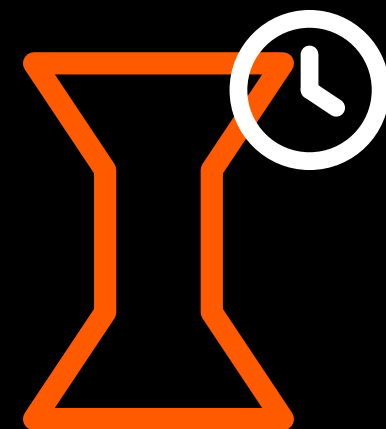


Проблема №2

Ложные срабатывания



До 90% срабатываний составляет шум



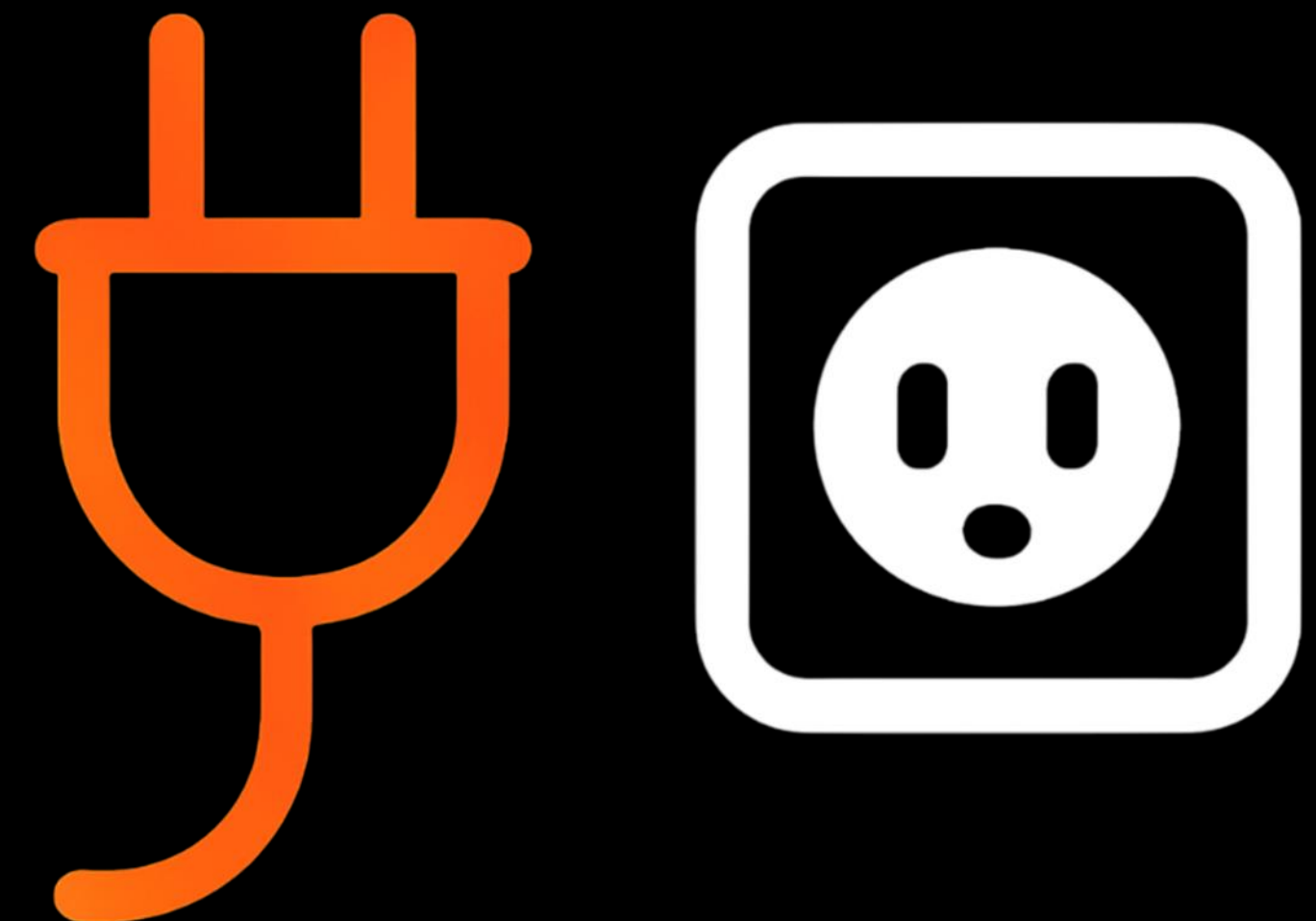
Потеря времени на поиск актуальных уязвимостей и риск задержки релиза или выпуска релиза с уязвимостями



Проблема №3

Отсутствие интеграций

- ▶ Результаты сканирования складываются вручную в Vulnerability Management систему
- ▶ Нет сквозного трекинга статуса уязвимости → дубли, просроченные уязвимости, потери времени на актуализации статуса
- ▶ Управление процессом затрудняется по мере увеличения масштаба разработки, аналогично растут финансовые затраты





Проблема №4

Комплаенс-давление



Нарушение требований = штрафы и риск приостановки деятельности



Требуется непрерывный журнал аудита и отчеты для комплаенса

Как выглядит идеальный процесс

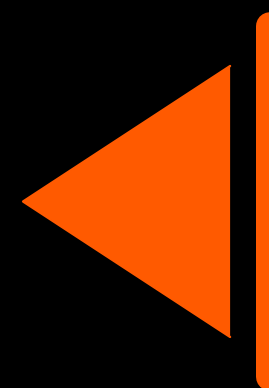


Идеальный процесс



Автономность AppSec

Работает «за кадром»
и не мешает разработке



Shift-left

Автоматические сканы
при запуске CI/CD
конвейера



Только актуальные уязвимости

Разработчики получают
уязвимости в виде тикета в
дефект-менеджмент системе



Ноль ручных действий

Разработчикам не нужно
самим заниматься
развитием безопасной
разработки



Отчетность быстро

Для сбора данных не
требуется привлекать
аналитика, данные доступны
по запросу



DevSecOps как услуга



Платформа берет на себя инфраструктуру

Аналитики вручную отсекают «шум»

Клиент получает только подтвержденные уязвимости

Вопросы





спикер

Пресейл инженер Arsafe

Анастасия Камалова

Пресейл инженер направления безопасной
разработки УЦСБ



Как Arsafe снимает ключевые боли



Представляем Apsafe

DevSecOps-as-a-Service

Предоставляет процесс, инструменты, кроме
устранения уязвимостей и Operate стадии DSO

Подключение

Первые актуальные
уязвимости через 10 дней
(или ранее)

Триаж сработок

Разработчики получают
только релевантные тикеты
в дефект-трекер (любой, у
которого есть API)

Понятный прайс

Прозрачная модель
оплаты — по числу KSLOC
(тыс. строк исходного кода),
месячных проверок и по
практике (SAST, DAST, SCA)

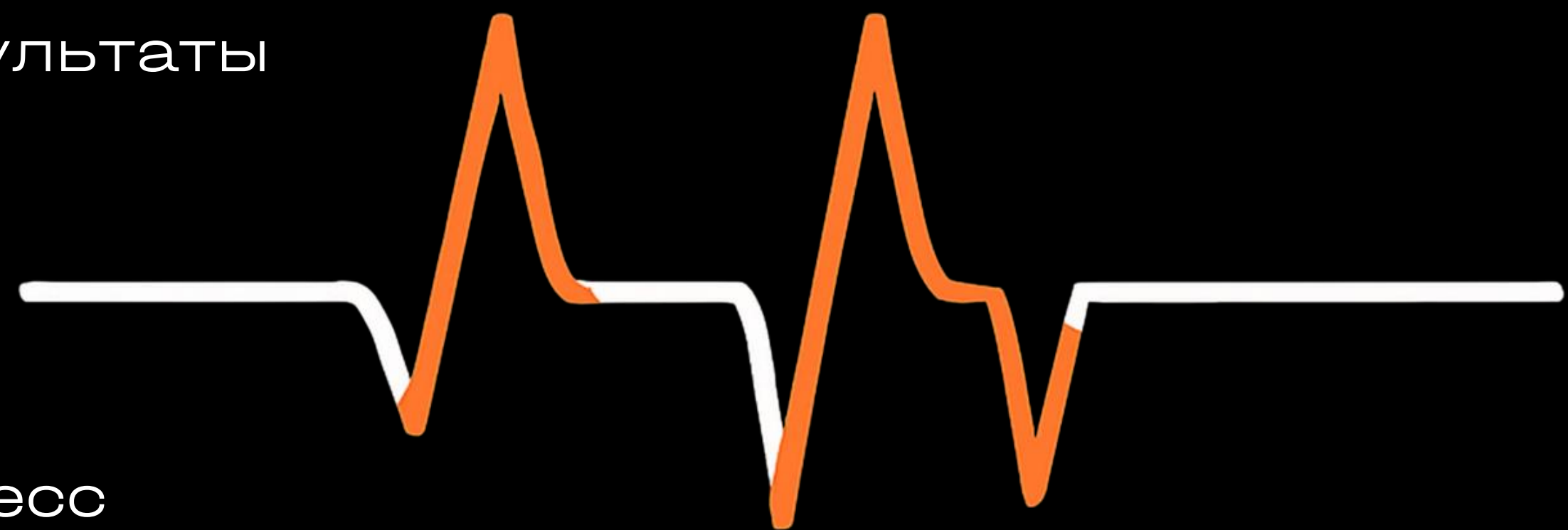
Сканирование

Код загружается в облако,
запускаются сканы SAST,
DAST, SCA



Как решаем проблему «Некому, некогда»

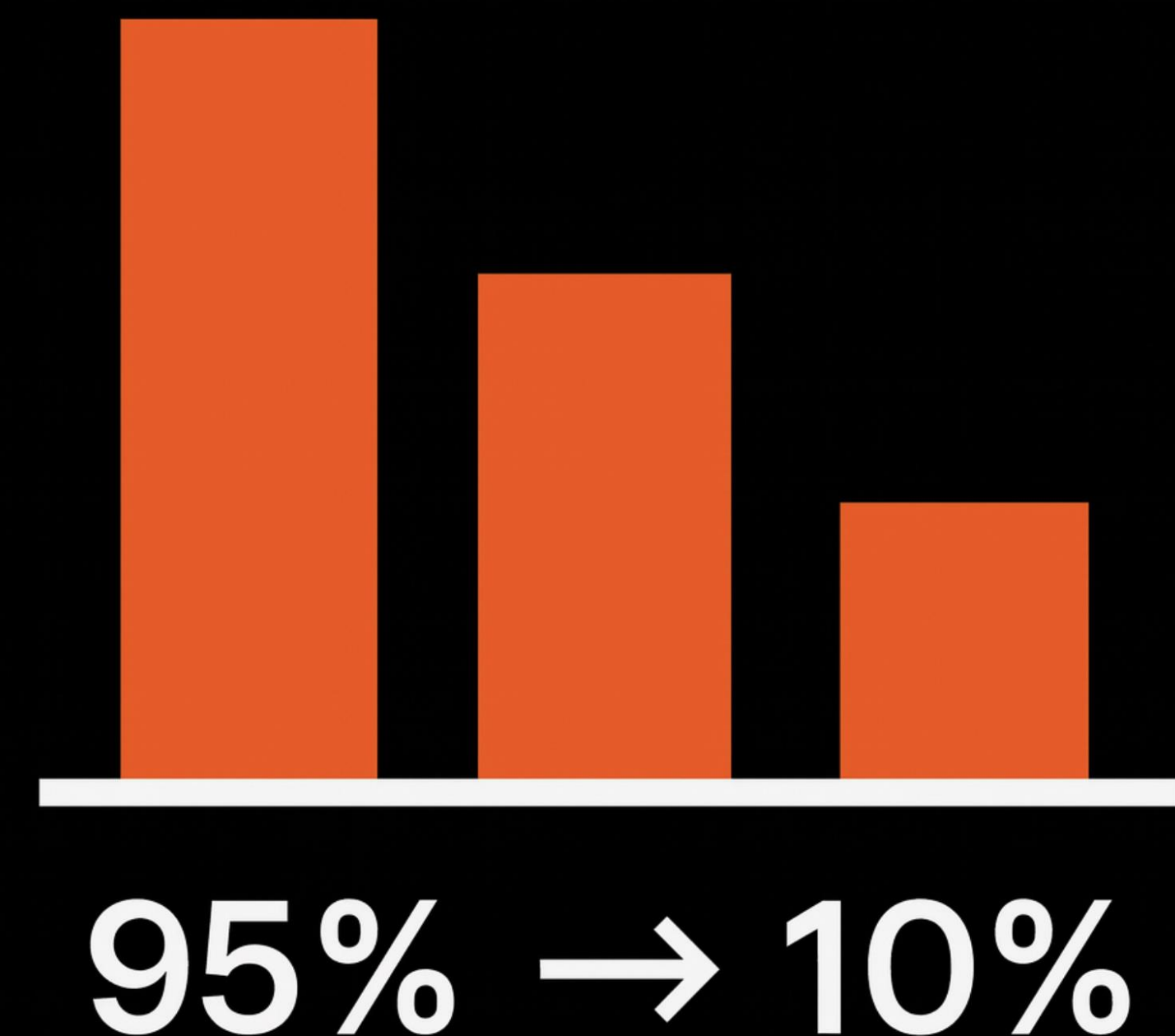
- ▶ Платформа + эксперты УЦСБ делают всё «под ключ»
- ▶ Сканы, анализ, приоритизация — готовые результаты
- ▶ Экспертный отчёт в удобном формате
- ▶ Можем интегрировать ваш сканер в наш процесс





Как решаем проблему «Ложные срабатывания»

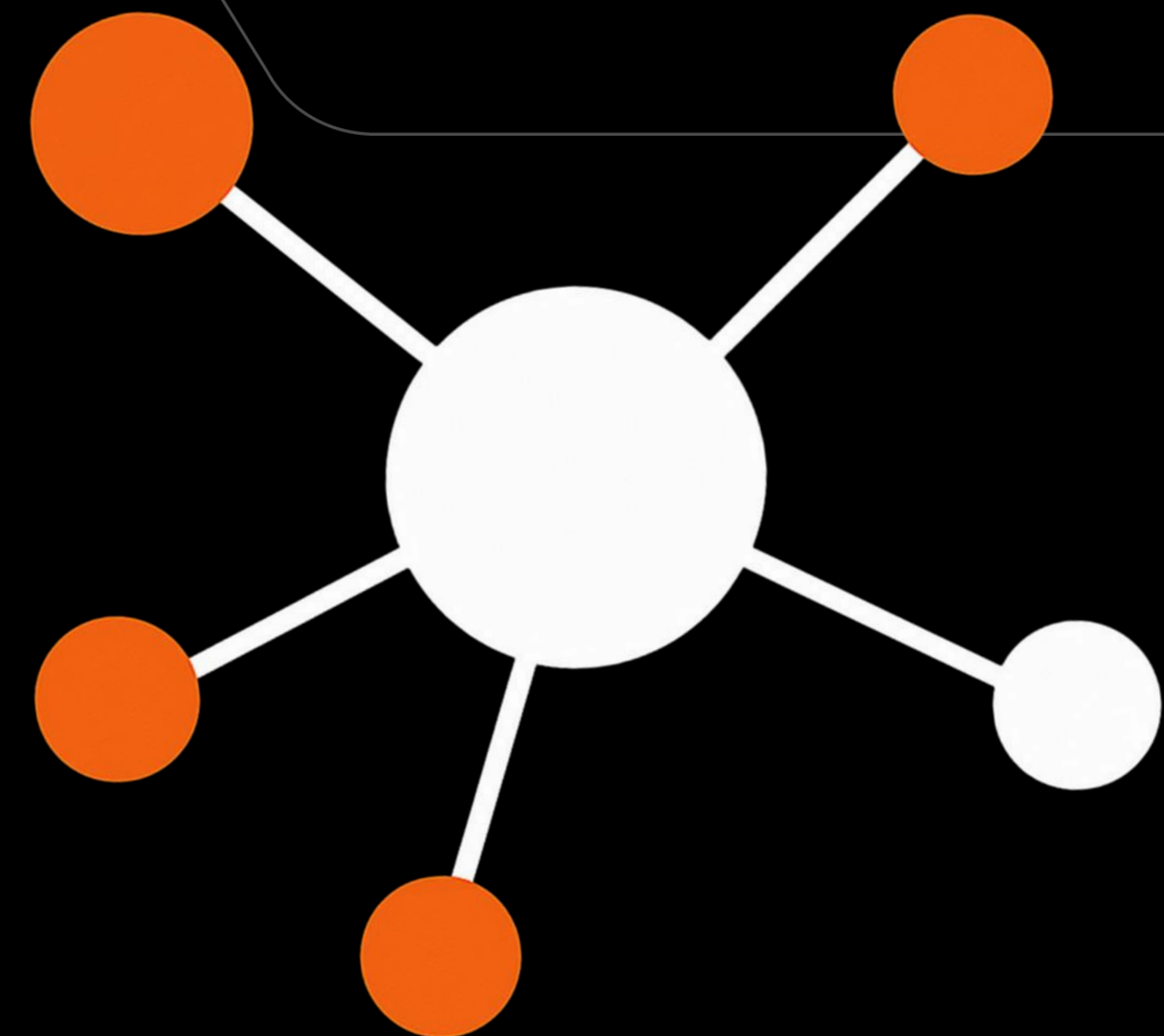
- ▶ Ложные срабатывания фильтруются в ручном режиме
- ▶ Подтвержденные уязвимости отправляются в таск-трекер только после ревью
- ▶ Разработчикам остаётся ~10 % актуальных уязвимостей → экономия времени





Как решаем проблему «Отсутствие интеграций»

- ▷ Нативные коннекторы: Jira, Redmine, YouTrack
(возможно подключение вашего трекера)
- ▷ Подключение к платформе меньше 10 дней
- ▷ Осуществляется интеграция со сканерами заказчика





Как решаем проблему «Комплаенс»

- ▶ Формирование отчётов под Приказ ФСТЭК №239, ГОСТ 56939, PCI DSS v4, Приказ ФСТЭК №117 (будет в силе с 01.03.2026)
- ▶ Передаем аудиторам описание нашей платформы для подтверждения соответствия





Базовые стандарты безопасной разработки

| СТАНДАРТ | | ЧТО ДЕЛАТЬ? |
|------------------------------|---|---|
| ГОСТ Р 56939-2024 | 1 | Внедрить статический и динамический анализ, SCA, проверку supply chain. Всё — как часть процесса разработки. |
| ГОСТ Р 71207-2024 | 2 | Использовать сертифицированный статический анализатор с нормальной точностью, API-интеграцией, фильтрацией ложных срабатываний. |
| ГОСТ Р 58412-2019 | 3 | Выявлять угрозы безопасности на всех этапах SDLC. Защищать сами процессы разработки. |
| ГОСТ Р ISO/IEC 27034 (серия) | 4 | Построить secure SDLC: определить уровни доверия и применять меры ASC. |
| ГОСТ Р 59793-2021 | 5 | Обязательное прохождение стадий создания АС. Безопасность должна быть включена в ТЗ и проектирование. |
| ГОСТ Р 57580.2-2018 | 6 | Зрелость процессов разработки с ИБ ≥ уровень 4. Периодический аудит и повышение зрелости. |

Что применять всем, как best practice



Обязательные стандарты для соответствия

| СТАНДАРТ | | ЧТО ДЕЛАТЬ? |
|------------------------------|---|---|
| ФЗ 152-ФЗ + Приказ ФСТЭК №21 | 1 | Анализ кода на НДВ, пентесты, защищённое программирование — при высокой угрозе (1–2 тип). |
| ФСТЭК 239 (КИИ) | 2 | Статический анализ, фаззинг, динамический анализ (1-я категория), устранение уязвимостей, информирование пользователей. |
| ФСТЭК 117 (ГИС) | 3 | Соблюдение ГОСТ 56939, контроль разработки (включая подрядчиков), регулярное сканирование кода, устранение уязвимостей. |

- 1: Операторы перс. данных**

2: Субъекты КИИ

3: Гос. ИС



Отраслевые стандарты для соответствия

| СТАНДАРТ | | ЧТО ДЕЛАТЬ? |
|------------------------------|---|--|
| ГОСТ 57580.1-2017 | 1 | Для ОУД-4: анализ кода, контроль уязвимостей, либо сертифицированное ПО. |
| Минэнерго 1215 | 2 | Обязательная система патч-менеджмента и оповещения об уязвимостях со стороны разработчиков. |
| ГОСТ 60880 / IEC 62859 (АЭС) | 3 | Проверка на скрытые функции и "unsafe code", особенно в критичных подсистемах. |
| PCI DSS v4.0 | 4 | SDLC с безопасными практиками: обучение, контроль среды, устранение топ-уязвимостей, обновления, защита веб-приложений (WAF/DAST). |

- 1: Финансовые организации

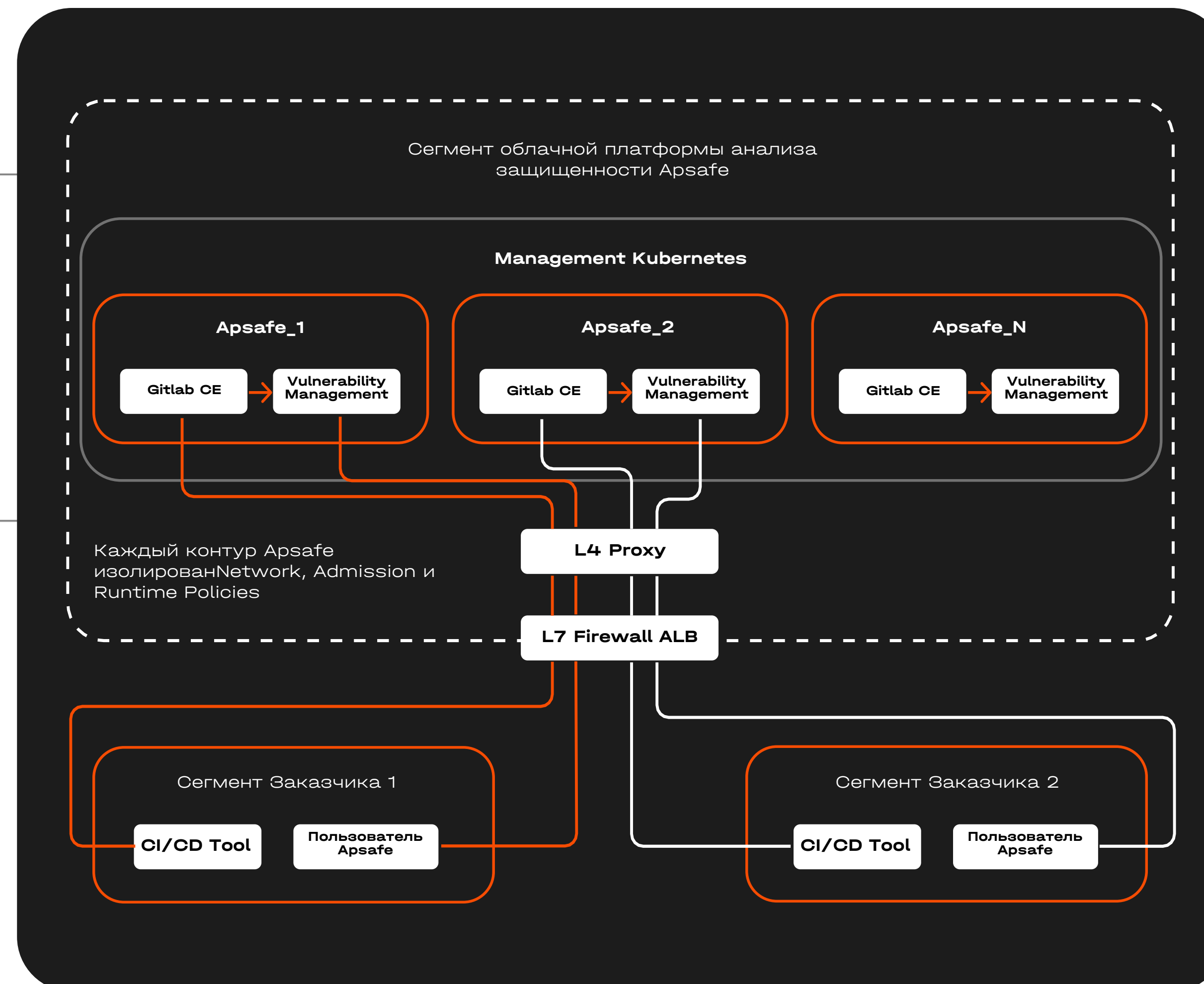
2: Субъекты электроэнергетики

3: Системы управления АЭС

4: Работа с платежными картами



Архитектура ПОД КАПОТОМ



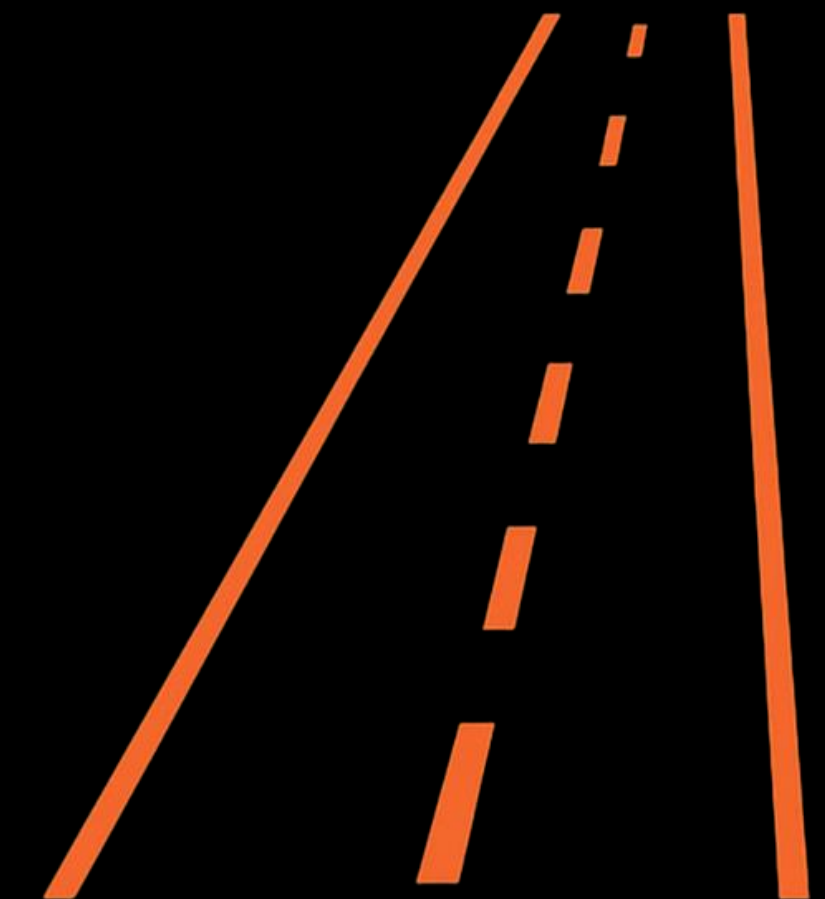


Финансовая выгода

- ▶ Сокращение времени на внедрение:
6 месяцев → 10 дней
- ▶ Снижение затрат на экспертов:
4 ставки → 0
- ▶ Убираем ложные срабатывания



6 месяцев,
4 эксперта



10 дней
0 экспертов



Container Security

безопасность в рантайме

▶ Container Security

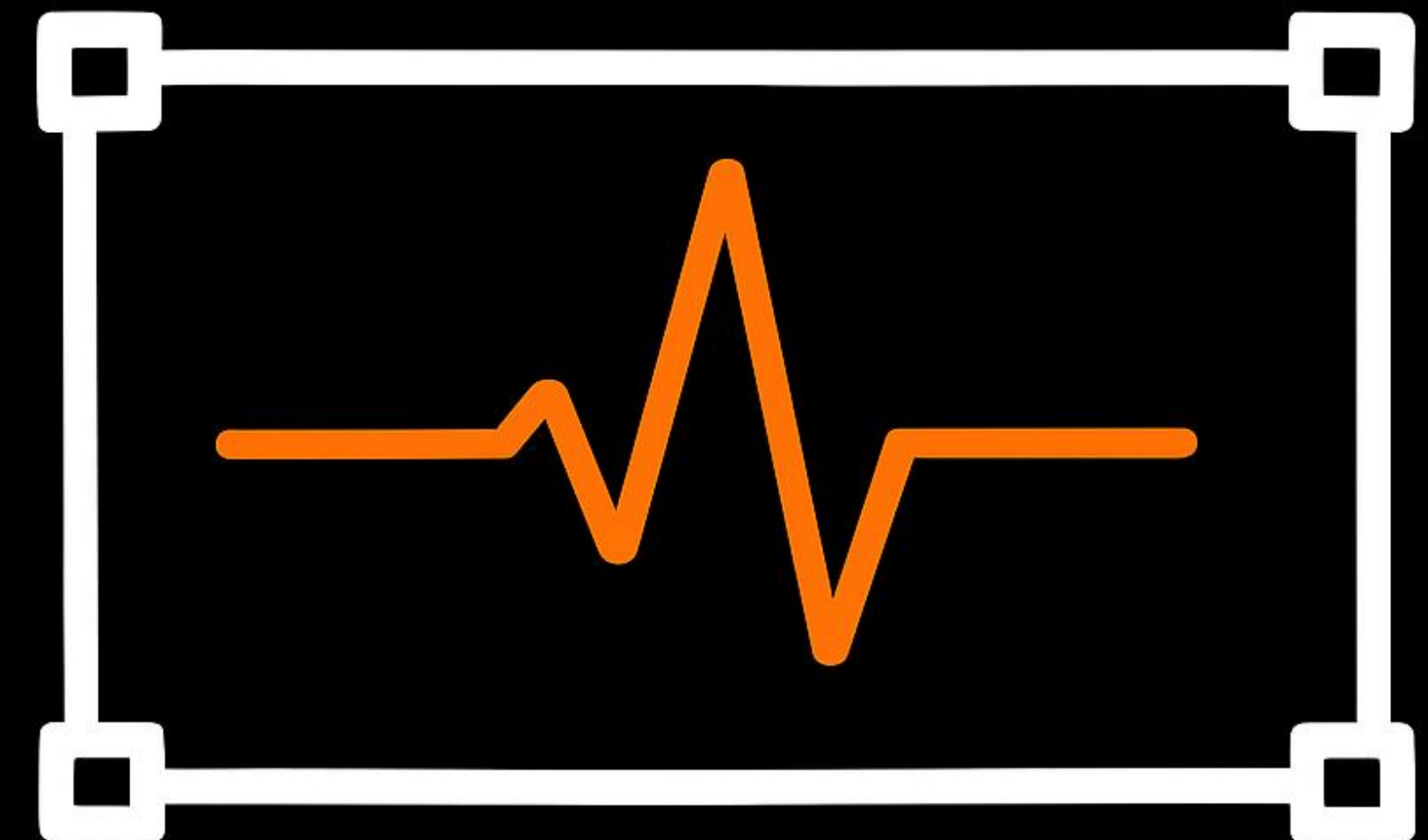
▶ Runtime Policy

 Под капотом Luntry

▶ Admission Policy

▶ Network Policy

 Подключаем к SOC





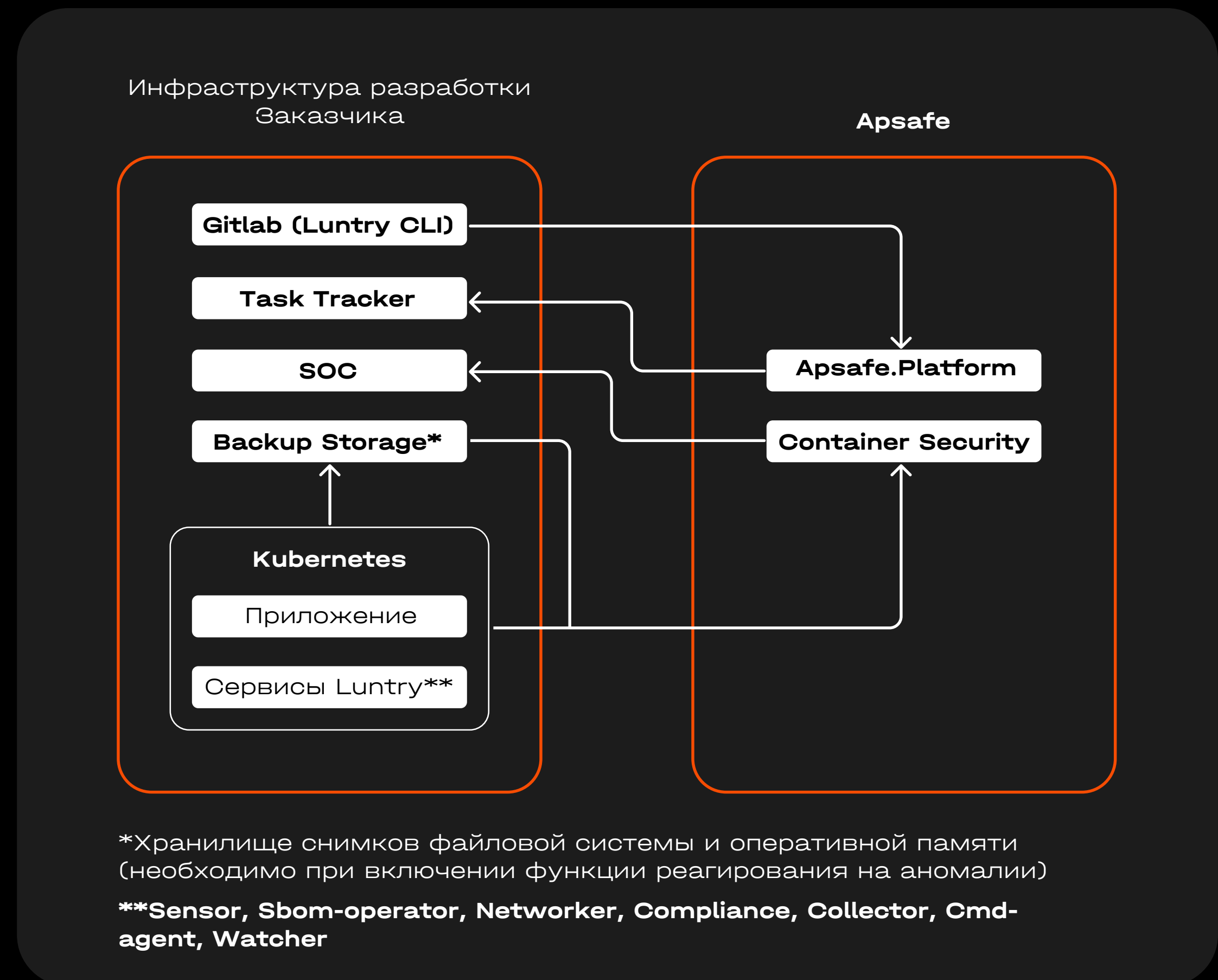
Container Security

процесс

Как это работает

Проводим обучение Luntry, после чего адаптируются все политики без блокировок. Далее возможно подключение к SOC или отправка событий.

По мере роста уровня зрелости процесса, перенесем инсталляцию Luntry в контур клиента.



Вопросы





спикер

AppSec-аналитик Apsafe

Виктор Тимашков

AppSec-аналитик группы инновационных решений
направления безопасной разработки УЦСБ



Два реальных кейса клиентов



СибКом Цифра кейс №1

- ▶ Провели все согласования и настроили доступы (6–13 марта)
- ▶ Интегрировали платформу безопасности приложения с Jira клиента — уязвимости сразу попадают в task-трекер
- ▶ На старте выявили 4 уязвимости высокого уровня критичности
- ▶ За несколько месяцев регулярных проверок найдены десятки уязвимостей
- ▶ Консультируем разработчиков по вопросам безопасной разработки



Банк Синара

кейс №2

- ▶ Интеграция прошла с 14 по 21 октября
- ▶ 29 октября — заказчик получил первый отчёт
- ▶ На старте выявлено:
 - уязвимые компоненты в продуктах
 - уязвимые конфигурации окружения
 - уязвимости в исходном коде
- ▶ За полгода работы платформы было выявлено множество уязвимостей разной критичности. На текущий момент большая часть уязвимостей устранена командой Заказчика
- ▶ Осуществлена интеграция с трекером, отправка уязвимостей автоматизирована
- ▶ Реализовали пожелания клиента в новой версии Apsafe Platform:
 - Улучшен интерфейс
 - Добавлены метрики и отчётность
 - Добавлена персонализированная аналитика для каждой команды разработки

Подробнее про Apsafe



Демонстрация Apsafe

Рассмотрим:

Как выглядит отправка кода на проверку из CI/CD, как это выглядит в платформе и в каком виде уязвимость попадает разработчикам.



Отправка кода из CI/CD

Collaborate with your team

We noticed that you haven't invited anyone to this group. Invite your colleagues so you can discuss issues, collaborate on merge requests, and share your knowledge.

Invite your colleagues

×

S

service

🔒

New subgroup

New project

⋮

Subgroups and projects

Shared projects

Archived projects

W

webapp

🔒

Репозиторий заказчика

★ 0

just now



Что происходит в платформе





Как разработчик ВИДИТ уязвимость

Jira Software

Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ Plans ▾ Create

Search

🔊 ? ⚙️ 👤

Доска DEF

Kanban board

QUICK FILTERS: Only My Issues Recently Updated

BACKLOG 22

SELECTED FOR DEVELOPMENT 0

IN PROGRESS 1

DONE 0 OF 5

Release...

DEF-5

Detected User Input Used to Manually Construct a SQL String. This Is Usually Bad Practice Because Manual Construction Could

⚡ ⚠️

DEF-7

Improper Neutralization of Special Elements in Data Query Logic

⚡ ⚠️

DEF-9

Отсутствие Атрибутов Безопасности В Конфигурации Контейнера

⚡ =

DEF-11

Improper Neutralization of Special Elements in Data Query Logic

⚡ ⚠️

DEF-12

Improper Neutralization of Special Elements in Data Query Logic

⚡ ⚠️

DEF-13

URL Redirection to Untrusted Site 'Open Redirect'

⚡ ⚠️

DEF-14

DEF-34

Внедрение Команд ОС (OS Command Injection)

⚡ ⚠️

We're only showing recently modified issues.

🔍 Looking for an older issue?

DEF-34

view this issue

Внедрение команд ОС (OS Command Injection)

Details

Status: IN PROGRESS (View Workflow)

Priority: ⚠️ Critical

Component/s: None

Labels: None

Affects Version/s: None

Fix Version/s: None

Epic Link: None

People

Reporter:

Assignee:

Dates

Created: 20/May/25 9:58 AM

Updated: 1 minute ago

Description

Title: Внедрение Команд ОС (OS Command Injection)

Add a comment...

Pro tip: press **m** to comment



Ключевые фишки Apsafe

▷ Сканирование

- Автоматический запуск сканирования
- Нет задержки разработки

```
1  apsafe-sync:
2  stage: build
3  rules:
4    - if: $CI_COMMIT_BRANCH == "main" && $CI_PIPELINE_SOURCE != "pipeline"
5  before_script:
6    - apk add git curl rsync
7    - git config --global user.email "security-check@asaas.ussc.ru"
8    - git config --global user.name "USSC_Security_Check"
9    # получение информации о текущем коммите
10   - current_branch=$(git rev-parse --abbrev-ref HEAD)
11   - commit_hash=$(git log -1 --pretty=format:"%H")
12   - commit_message=$(git log -1 --pretty=format:"%s")
13   - commit_author=$(git log -1 --pretty=format:"%an <%ae>")
14   - commit_history=$(git log -3 --pretty=format:"%H %an <%ae>\n" | tr "\n" ' ')
15  script:
16    - git clone -b ${APSAFE_SYNC_GITLAB_BRANCH} https://${GITLAB_ACCESS_TOKEN_NAME}:${GITLAB_ACCESS_TOKEN}@gitlab-apsafe.com/customer/${APSAFE_SYNC_GITLAB_REPOSITORY} /tmp/apsafe_sync
17    - find /tmp/apsafe_sync/ -maxdepth 1 -type f ! -name '.gitlab-ci.yml' -exec rm -rf {} +
18    - find /tmp/apsafe_sync -maxdepth 1 -type d ! -name '.git' ! -name '.' ! -name '..' ! -name 'apsafe_sync' -exec rm -rf {} +
19    # синхронизация файлов
20    - rsync -av ./ /tmp/apsafe_sync --exclude apsafe_sync --exclude .git --exclude .gitlab-ci.yml > /dev/null
21    - cd /tmp/apsafe_sync
22    - git add . *
23    - >
24    | git commit -m "Commit Hash: $commit_hash\nCommit Message: $commit_message\nCommit Author: $commit_author\nCurrent Branch: $current_branch\nCommit History:\n$commit_history"
25    # зеркалирование репозитория
26    - git remote set-url origin https://${GITLAB_ACCESS_TOKEN_NAME}:${GITLAB_ACCESS_TOKEN}@gitlab-apsafe.com/customer/${APSAFE_SYNC_GITLAB_REPOSITORY}
27    - git push -u origin HEAD -ff || true
28  after_script:
29    - rm -rf /tmp/apsafe_sync
```




Ключевые фишки Apsafe

- ▶ Управление уязвимостями
 - Полный цикл «Detected → Fixed»

The screenshot displays the 'All Findings' page in the Apsafe interface. It shows a table of three vulnerabilities, each with a status of 'Active, Unfixed'. An orange box highlights the 'Status' column for all three entries. To the right, a legend explains the status options:

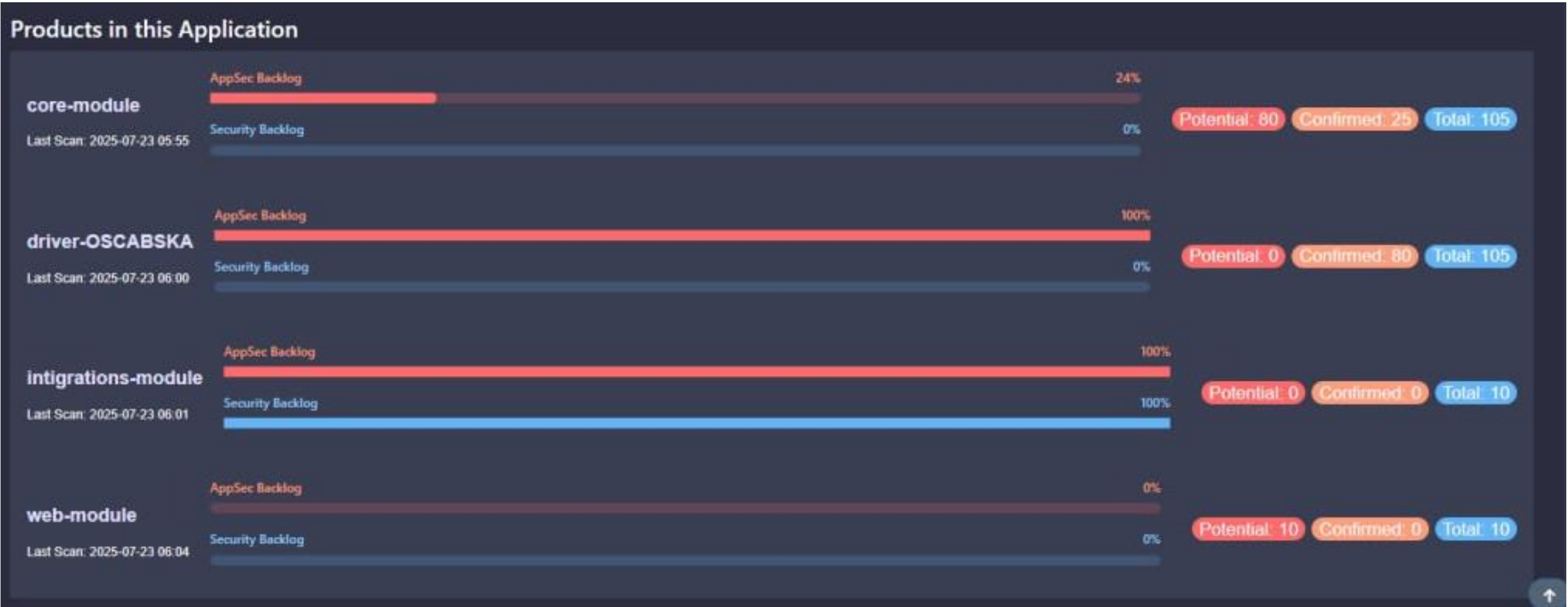
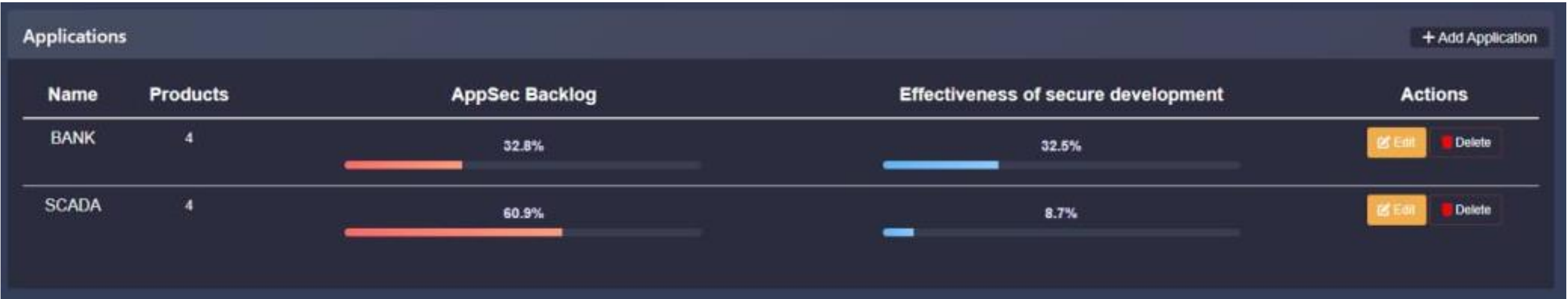
- ☐ Active ?
- ☐ Verified ?
- ☒ Fixed ? Indicates whether the vulnerability has been fixed.
- ☐ False Positive ?
- ☒ Duplicate ? [original: 156 : CVE-2023-45133 @Babel/Traverse]
- ☐ Out Of Scope ?

| | Severity | Name | CWE | Vulnerability Id | EPSS Score | EPSS Percentile | Date | Age | SLA | Reporter | Found By | Status |
|--------------------------|----------|---------------------------------------|------|------------------|------------|-----------------|---------------|-----|-----|--------------------|------------|-----------------|
| <input type="checkbox"/> | Critical | CVE-2023-45133 @Babel/Traverse 7.14.7 | 697 | CVE-2023-45133 | N.A. | N.A. | July 22, 2025 | 13 | 6 | Admin User (admin) | Trivy Scan | Active, Unfixed |
| <input type="checkbox"/> | High | CVE-2021-3807 Ansi-Regex 5.0.0 | 1333 | CVE-2021-3807 | N.A. | N.A. | July 22, 2025 | 13 | 17 | Admin User (admin) | Trivy Scan | Active, Unfixed |
| <input type="checkbox"/> | High | CVE-2022-25883 Semver 6.3.0 | 1333 | CVE-2022-25883 | N.A. | N.A. | July 22, 2025 | 13 | 17 | Admin User (admin) | Trivy Scan | Active, Unfixed |



Ключевые фишки Apsafe

- ▶ Управление уязвимостями
 - Объединение микросервисов в сущность «Приложение/команда разработки», общая статистика





Ключевые фишки

Apsafe

- ▶ Управление уязвимостями
 - Настраиваемые правила автотриажа

The screenshot displays the Apsafe interface with three panels:

- Автоматическое закрытие уязвимостей с устаревшими URL-адресами**
 - Этот сценарий полезен для автоматического закрытия уязвимостей, обнаруженных по URL-адресам, которые больше не существуют или не активны.
 - Scope: All
 - Matching Rules**
 - Rule Chain**
 - URL: old-domain.com
 - Description: 404 Not Found
 - Rule Chain**
 - Description: connection refused
 - URL: old-domain.com
 - Status Actions**
 - Out of Scope
 - Disabled (0 minutes ago)
- Автоматическая очистка тестовых каталогов**
 - Автоматическое закрытие уязвимостей, как "out of scope" в директориях test, QA, UT
 - Scope: Product
 - mega-bank
 - Matching Rules**
 - File Path: TEST
 - File Path: QA
 - File Path: doc
 - File Path: example
 - File Path: UT
 - Status Actions**
 - Out of Scope
 - Risk Accepted
 - Disabled (11 minutes ago)
- Объединение SQLi** (Auto Merge)
 - Этот кейс позволяет выявить и объединить одинаковые уязвимости, найденные двумя разными SAST-инструментами, что помогает избежать дубликатов в ASOC.
 - Scope: Application
 - BANK
 - Tools Mapping**
 - Semgrep OSS Scan (SARIF) → Semgrep JSON Report (false_p)
 - active
 - Merge Conditions**
 - Title
 - Source pattern: SQL Injection
 - Target pattern: SQL Injection
 - File Path
 - Fields must be equal
 - Enabled (3 minutes ago)



Ключевые фишки Apsafe

- ▶ Интеграции
 - Выгрузка дефектов в task-трекеры: уже есть поддержка Jira, Redmine YouTrack

Небезопасная Загрузка Файлов

Редактировать

Добавить комментарий

Назначить

Еще...

Нужно сделать

В работе

Бизнес-процесс

Детали задачи

Тип: Epic

Приоритет: Medium

Метки:

Имя эпика: Небезопасная Загрузка Файлов

Статус: В РАБОТЕ

Решение: Нет решения

Описание

Title: Небезопасная Загрузка Файлов

Defect Dojo link: finding/6742 (6742)

Severity: Medium

Due Date: Sept. 28, 2025

CWE: CWE-362

CVE: Unknown

Product/Engagement/Test: security-check#125

Branch/Tag: security-check

BuildID: 125

Commit hash: 4a10d846

Source File:

Source Line: 61

Description: Функция writeFile используется при инициализации ключей и сертификатов, но может создавать потенциальную входную точку для атак при внедрении произвольных файлов.

1. Path Traversal – возможность чтения произвольных файлов через ../ в пути.
2. Symlink-атаки – подмена файла через символические ссылки.
3. Инъекция NULL-байта – обход проверок через \0 в имени файла.

>_@...✎

БЗК x

kdbo-2803 Создал(a) svc_apsafe_yt 4 недели назад Обновил(a) Отображать для пользователи с доступом к чтению задачи

[Medium] Service 'Create-Buckets' Allows for Privilege Escalation via Setuid or Setgid Binaries. Add 'No-New-Privileges:true' in 'Security_opt' to [...]

Commit Hash:

Commit tag:

Commit Message: Merqe branch 'release/10' into release

Commit Author:

Current Branch: release

Commit URL:

Commit History:

Описание уязвимости

Сервис create-buckets позволяет повышение привилегий через setuid или setgid.

При компрометации приложения нарушитель потенциально получит возможность повышения привилегий в целях дальнейшего продвижения и закрепления.

Уязвимое место в коде (файл с номером строки или диапазоном строк)

[Apsafe] Potential for DLL Hijacking (LoadLibrary)

Добавил(a) User USSC 3 месяца назад.

Статус: Новая

Приоритет: Нормальный

Назначена: -

Версия: 04.2025

severity: High

line_number: 54

location: 6.3.0+b3.r113254/Server/externals/logger/inc/LogWrapper.h

Дата начала: 22.04.2025

Срок завершения:

Готовность: 0%

Оценка временных затрат:

status: Active, Verified

comment: Для исправления уязвимости рекомендуется заменить относительные пути загрузки DLL на абсолютные с использованием безопасных системных директорий (например, System32), применять флаги LOAD_LIBRARY_SEARCH_SYSTEM32 или SetDefaultDllDirectories для ограничения поиска библиотек, реализовать проверку цифровой подписи и хеша DLL перед загрузкой, а также рассмотреть возможность статической линковки или использования delay-load механизмов для критически важных библиотек. Дополнительно следует настроить политики AppLocker или Windows Defender ASR для блокировки несанкционированной загрузки DLL из ненадёжных мест.

Описание

Scanner: Semgrep
The 'LoadLibrary' function is used to load DLLs dynamically. Depending on the filepath parameter, the OS version, and the modes set for the process prior to calling LoadLibrary, DLL hijacking may be possible. Attackers can exploit this by placing DLL files with the same name in directories that are searched before the legitimate DLL is.



Roadmap Apsafe

- ▶ Проверка коммитов в релизах
- ▶ Автоматизированные отчеты под Приказ ФСТЭК №239, ГОСТ 56939, PCI DSS v4, Приказ ФСТЭК №117 (вступает в силу с 01.03.2026)
- ▶ Двусторонняя синхронизация статусов уязвимостей между Apsafe.Platform и дефект-трекером
- ▶ Улучшение качества сканирования за счет кастомных правил

Вопросы



СПАСИБО!

Получите обзор топ-угроз под вашу архитектуру

Заполните анкету по QR-коду и мы расскажем,
от чего вам следует защищаться



cybersec@ussc.ru