Вебмониторэкс
защита веб-приложений и API

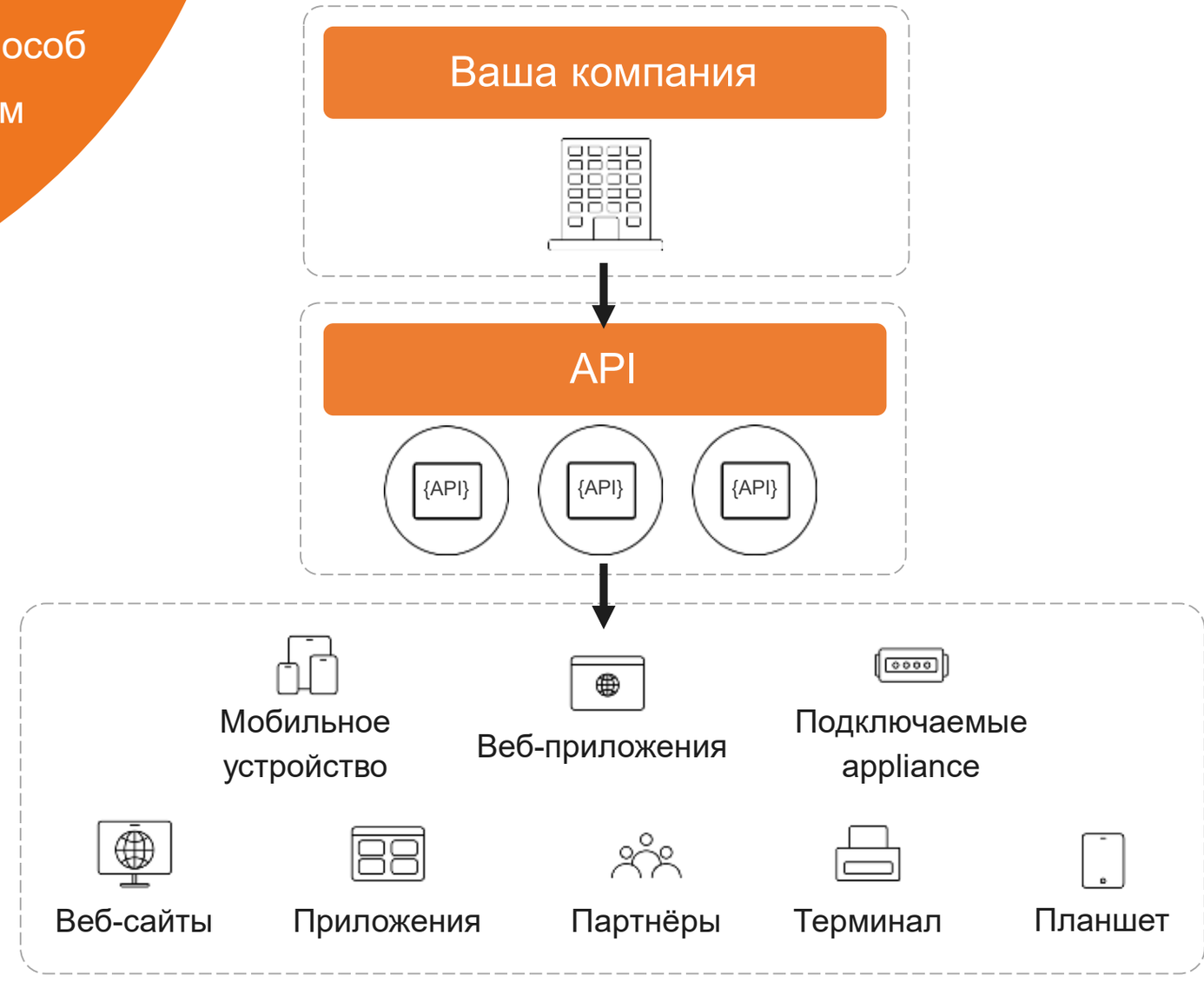
Методология построения и защиты API

Лев Палей
Директор по информационной безопасности

Что такое API?

Application Programming Interface (программный интерфейс приложения) – способ взаимодействия компьютерных программ

Вебмониторэкс
защита веб-приложений и API



API: приватные, партнёрские, внешние

- REST
- SOAP
- RPC
- Web Socket
- GraphQL

Почему API — это важно?

Вебмониторэкс
защита веб-приложений и API

на 50%

Каждый год увеличивается количество API-интерфейсов

83%

Доля API-вызовов в интернет трафике

5x

Рост пользователей Open API в период с 2020 по 2024 год

Почему API стало много?

Вебмониторэкс
защита веб-приложений и API

Веб-приложения и API являются значимой частью бизнеса

Потребность в поддержке современного стека технологий

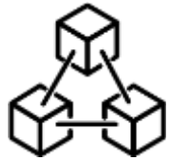
В инфраструктуре большое количество веб-приложений

Необходимо защищать микросервисы

API важнее WEB

Вебмониторэкс
защита веб-приложений и API

Структурированный обмен данными

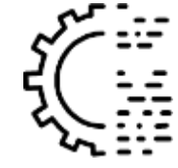


Коммуникация между сервисами

Масштабируемость и эффективность

Гранулярный контроль

Стандартизация и взаимодействие

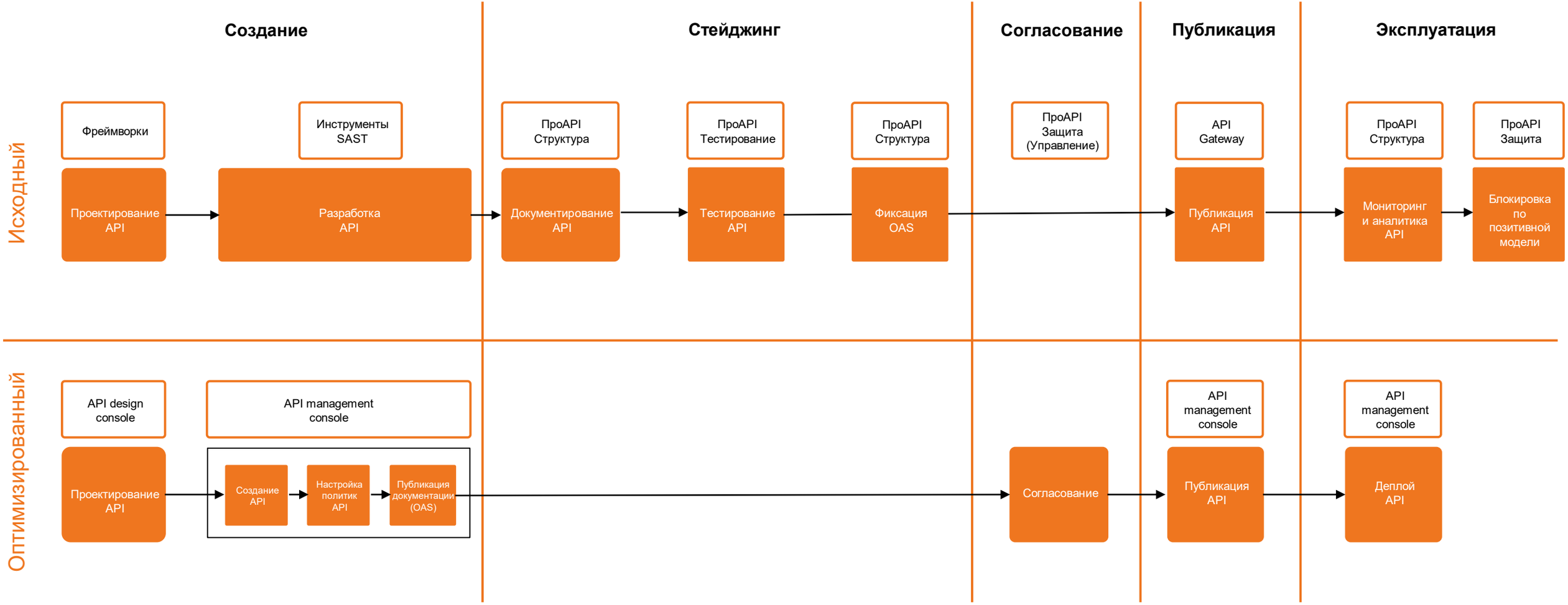


Версионирование и управление жизненным циклом

Монетизация и бизнес-возможности

Схема процессов

Вебмониторэкс
защита веб-приложений и API



Этапы реализации защиты API

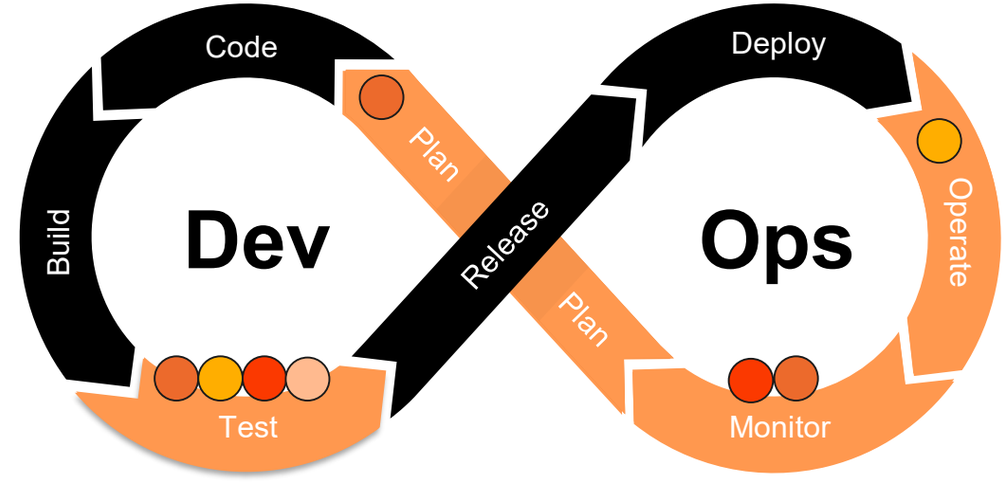
Вебмониторэкс
защита веб-приложений и API

Исходный процесс

- Отсутствует проверка соответствия того, что согласовывалось, и того, что опубликовано.
- Отсутствует мониторинг изменений. Нет верификации трафика через спецификацию.
- Требует ручной настройки политик безопасности (например, rate limit, IP Filtering)
- Требует дополнительных инструментов для DAST

Оптимизированный процесс

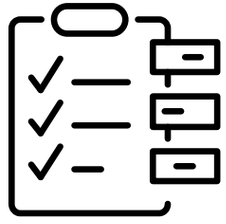
- Требует инструментов для автоматизации проектирования и публикации API
- Увеличение длительности и трудоёмкости процесса создания API на этапах «Стэйджинг» и «Согласование»



Этапы реализации защиты API

Вебмониторэкс
защита веб-приложений и API

Этап I: Знать



- Что есть в текущих API, из чего они состоят
- Какие данные передаются, какие точки используются
- Какие уязвимости и угрозы в API
- Когда происходит что-то аномальное на endpoint

Этап II: Не допускать

- Контролировать расхождения с согласованной спецификацией (политикой)
- Заблокировать те endpoint, которые несут угрозу
- Автоматизировать реагирование при обнаружении угрозы (BOLA, обнаружение PII)

Этап III: Защищать

- Фиксировать «нормальный» трафик запросов к API
- Блокировать все что не описано в OAS
- Блокировать все запросы с признаком утечки секретов
- Выявлять проблемные endpoint в API (Shadow API)

Первый пример

Вебмониторэкс
защита веб-приложений и API

Описание проблемы

API-интерфейсы программы лояльности ритейл компании подвергаются атакам на бизнес-логику.

Механика предполагала использование разных IP-адресов для прохождения аутентификации и последующих запросов, реализующих атаку типа BOLA.

Существующий WAF работал с трафиком, в котором присутствовало большое количество API соединений с автоматически сгенерированными параметрами, что вместе с отсутствием более тонкой настройки правил фильтрации вызывало перегрузку WAF.

Результат

WAF стал обрабатывать кратно меньше трафика и стабилизировался.

Решение

- Формирование актуальной документации по API (OAS-спецификации)
- Публикация API-интерфейсов через специальный компонент
- Фиксация спецификации
- Настройка мониторинга
- Корректировка документации по итогам мониторинга
- Создание правил отслеживания сессий, меняющих IP-адреса и нарушающих бизнес-логику
- Интеграция с SIEM для дальнейшей блокировки этих сессий

Второй пример

Вебмониторэкс
защита веб-приложений и API

Описание проблемы

Сторонним подрядчиком разработано приложение, которое впоследствии было интегрировано с другими приложениями внутри инфраструктуры компании.

Новое приложение было принято без зафиксированной на этапе разработки документации (OAS-спецификации), поэтому отсутствует представление о коммуникациях между API-интерфейсами приложений.

Требуется оценка и минимизация рисков каскадной компрометации связанных приложений.

Результат

Недостатки спецификаций существующих приложений собраны и исправлены.

Решение

- Формирование документации по API (OAS-спецификации)
- Применение инструментов DAST для проверки по полученной документации приложения, созданного сторонним подрядчиком
- Отправка комментариев по уязвимостям и недостаткам OAS-спецификации разработчикам
- Фиксация исправленной разработчиками или самостоятельно скорректированной документации (OAS-спецификации) на специализированном компоненте (МЭ API)

Вебмониторэкс

защита веб-приложений и API

webmonitorx.ru

info@webmonitorx.ru

+7 495-740-35-44

Habr

Телеграм

ВКонтакте