DPA ANALYTICS

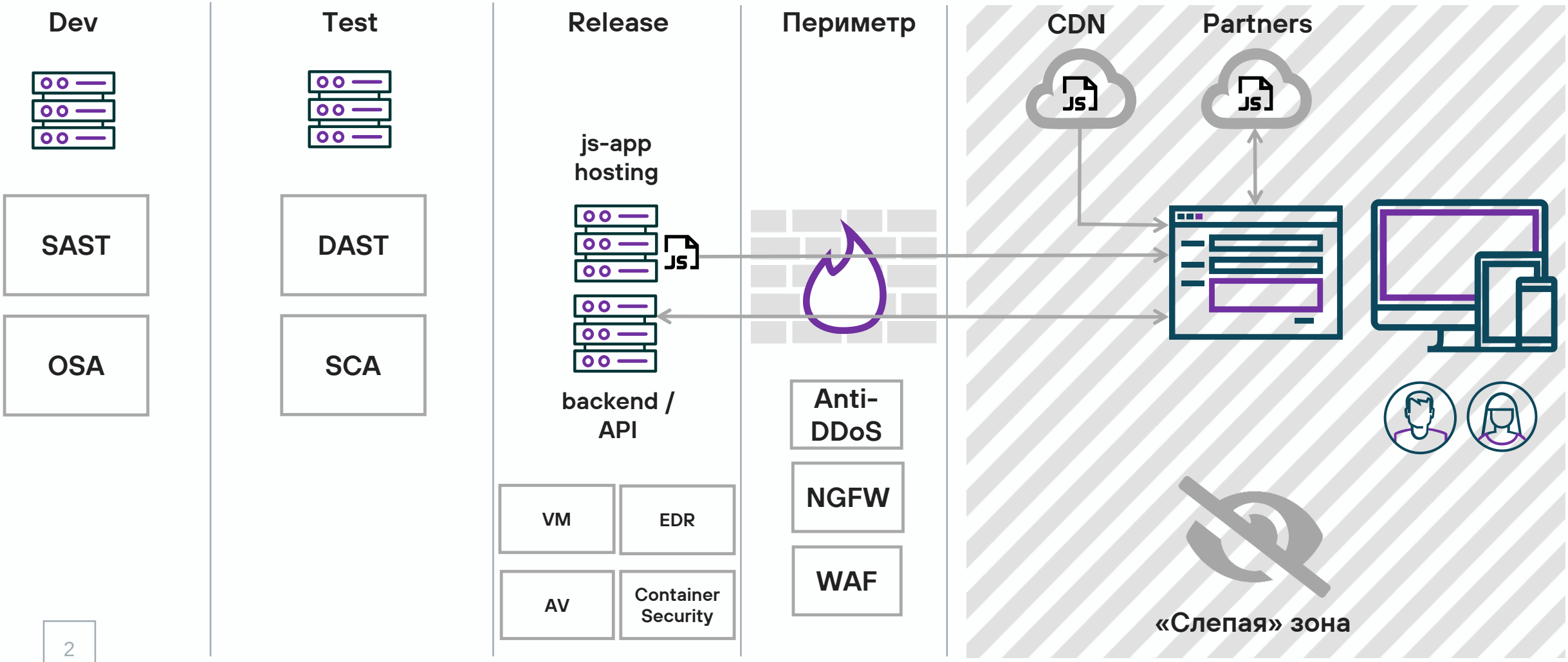
Браузер пользователя – идеальное место для кражи данных и атак. Как обнаружить утечки там, где WAF не видит?

Михаил Парфенов
Главный архитектор по ИБ

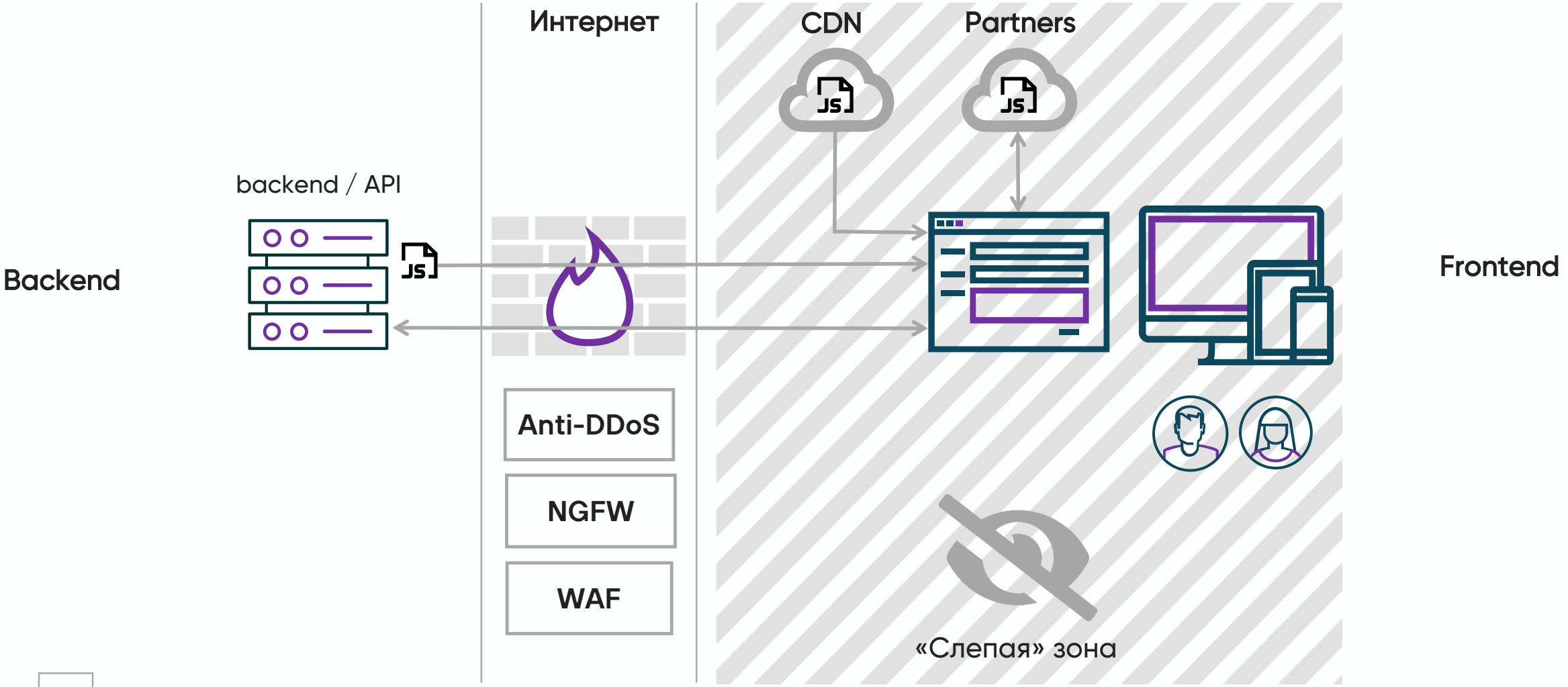
dpa-analytics.ru

Безопасность веб-приложений

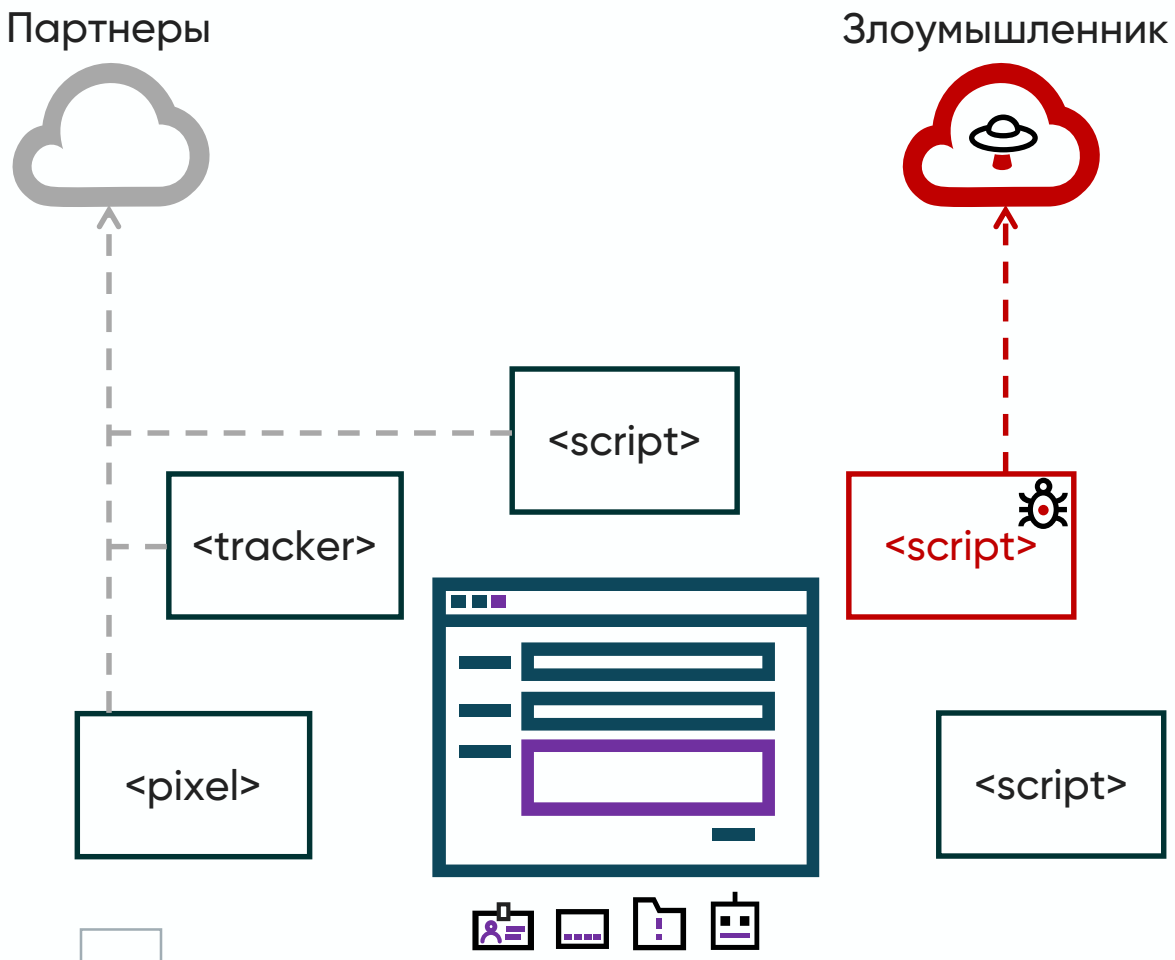
DPA ANALYTICS



Backend или frontend?



Ценность frontend-приложений для злоумышленника



- Персональные данные, данные банковских карт, коммерческая тайна, учетные данные, коды OTP и т. д.
- Снятие профиля пользователя / установка cookie сетей обмена трафиком для показа рекламы конкурентов либо атак на пользователей через сторонние сайты
- Выполнение действий от имени пользователя веб-приложения
- Показ пользователю мошеннических баннеров от имени компании для последующей кражи денег / данных
- Майнинг криптовалюты в браузере пользователя либо использование браузера в DDoS-атаках на другие ресурсы
- Заражение устройства пользователя через уязвимости браузера

Основные компоненты frontend-приложения

DPA ANALYTICS

JS-приложение и его зависимости

Код фреймворка

Собственный код

Прямые зависимости

Транзитивные зависимости

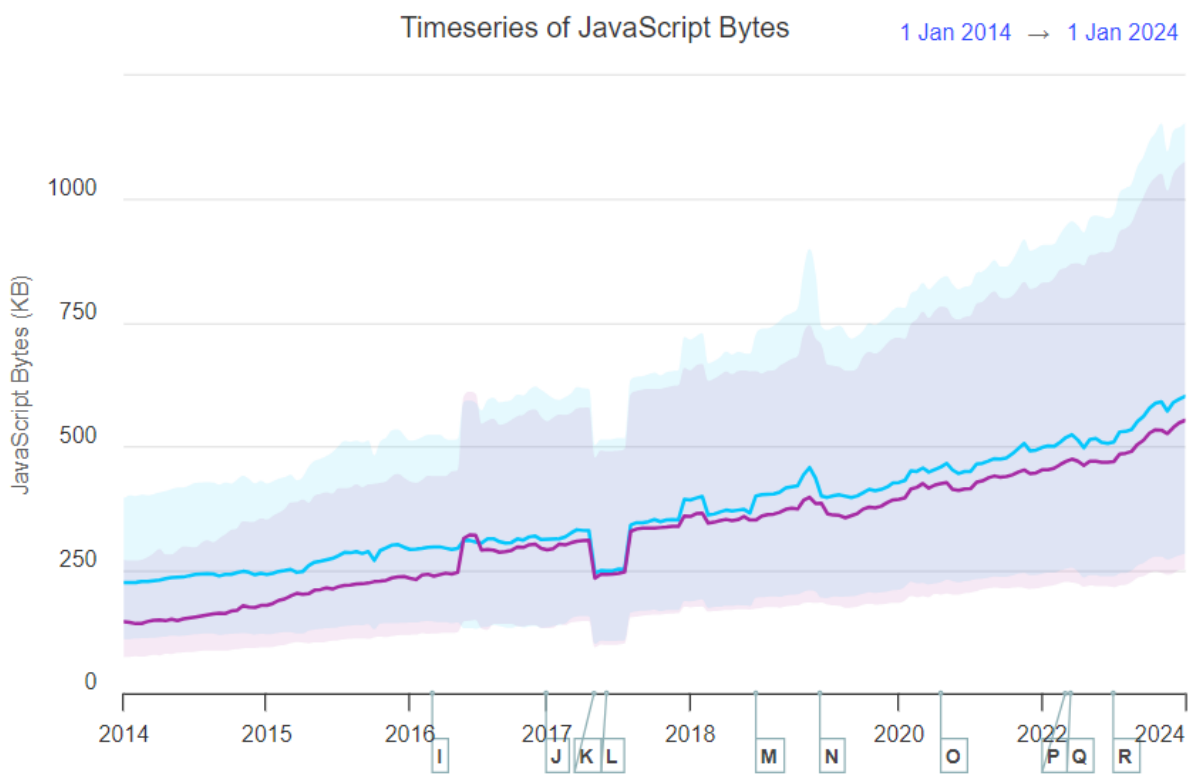
Как правило, перед публикацией приложения собираются в единый файл-bundle

Сторонние JS-сервисы

- Сервисы веб-аналитики
- Интернет-счетчики
- Маркетинговые системы
- Платформы контекстной рекламы
- Captcha
- Онлайн-чаты
- Онлайн-карты
- JS-библиотеки во внешних CDN
- И другие

Размер JavaScript-приложений

Веб-приложение	Размер JS-файлов
Jira Cloud	50 МБ
mail.google.com	20 МБ
1Password.com	13 МБ
gitlab.com	13 МБ
YouTube	12 МБ
Google.com	9 МБ
ChatGPT	7 МБ
Npmjs.com	4 МБ
StackOverflow	3,5 МБ
wikipedia.org	0,2 МБ

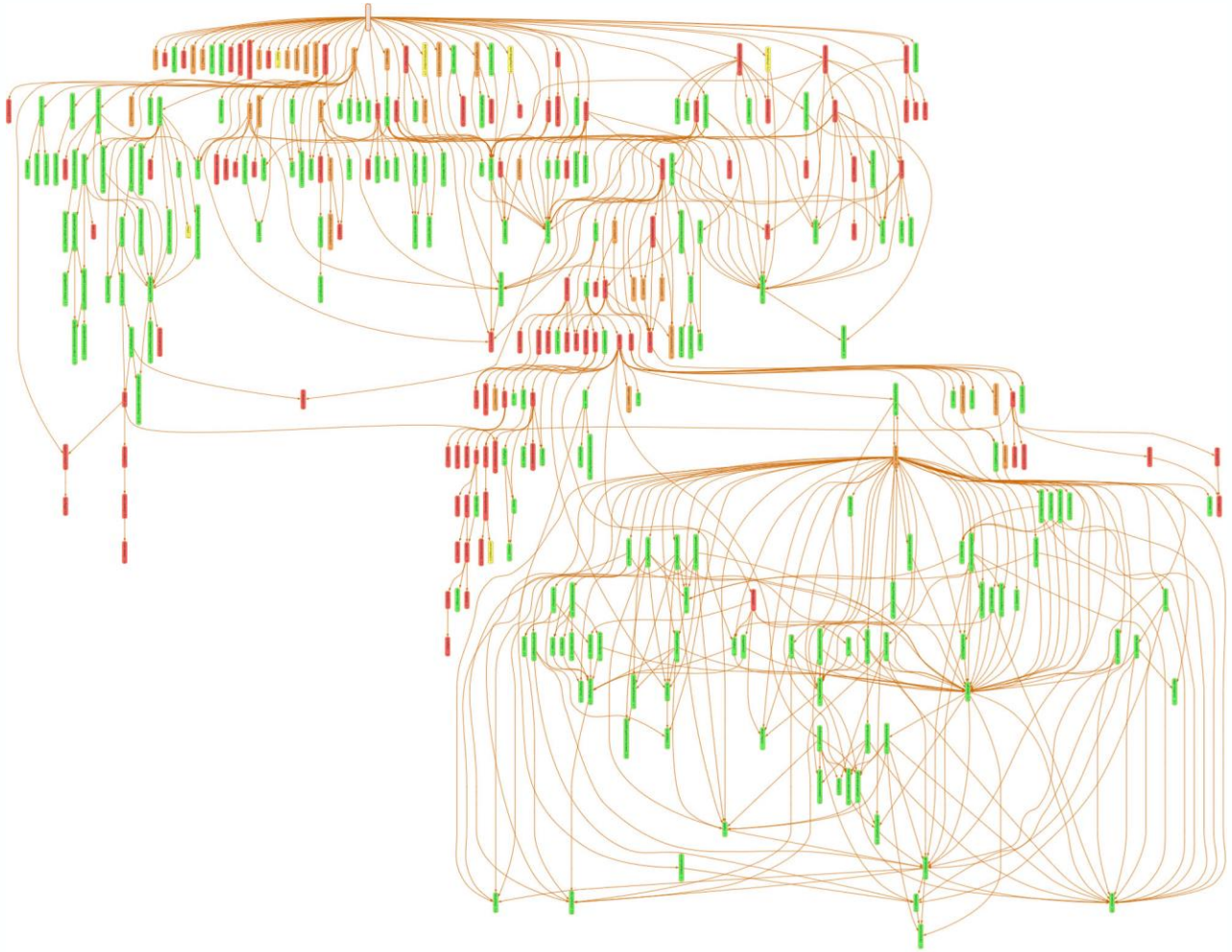


https://habr.com/ru/companies/ruvds/articles/796595/

https://httparchive.org

Зависимости в JavaScript-приложениях

DPA ANALYTICS



Количество

94

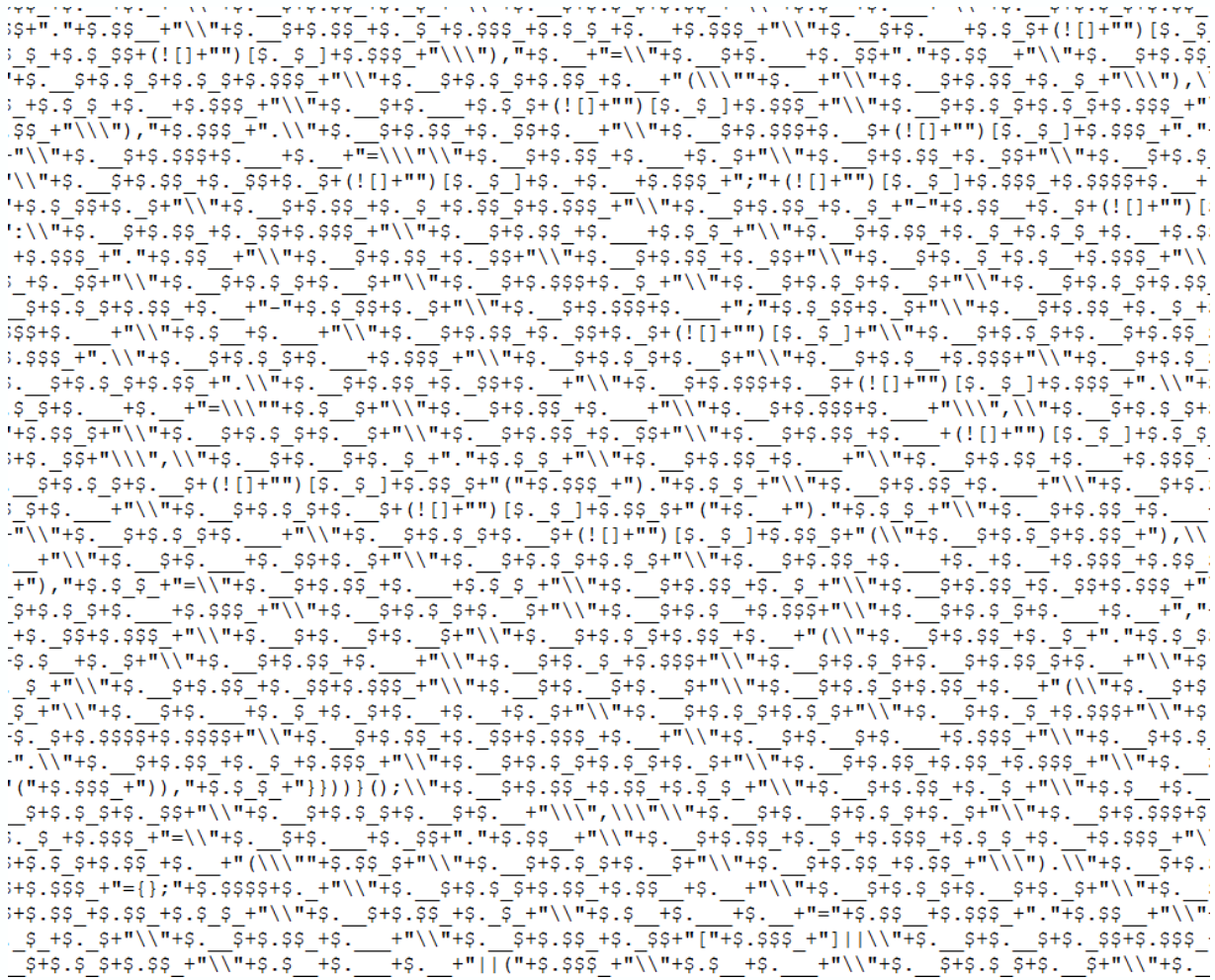
Глубина

15

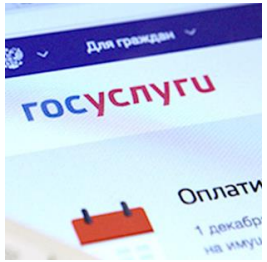
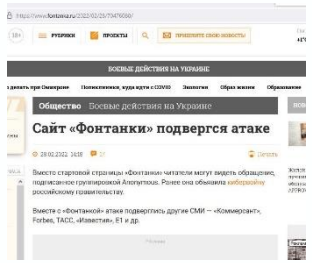
Размер (МБ)

12

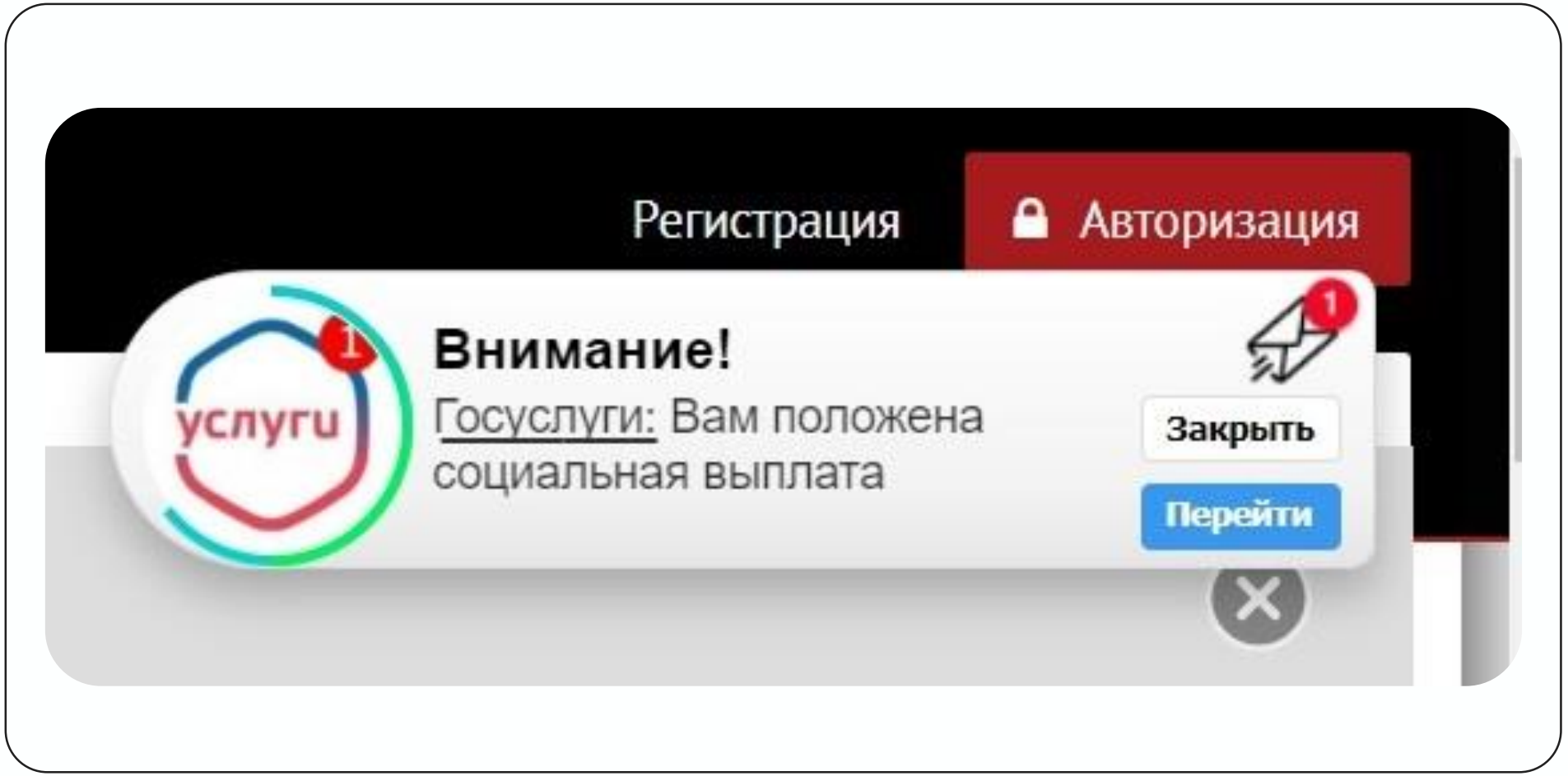
Минификация и обфускация



Инциденты

			Magento	Google Tag Manager		Polyfill.io
Год	**2017**	**2018**	**2019**	**2021**	**2022**	**2024**
Инцидент	Размещены iframe с неизвестными доменами в Нидерландах	Злоумышленник встроил в одну из js-библиотек js-сниффер	В 100 000+ интернет-магазинов встроен js-сниффер	В 316 интернет-магазинах обнаружен js-сниффер, скрытый в Google Tag Manager	Внедрен код на сайты СМИ «Коммерсантъ», Forbes, РБК, ТАСС, «Известия» и других крупных российских компаний	Внедрен вредоносный код в библиотеку Polyfill.js. Код выполнялся на > 350 000 веб-приложений.
Вектор	N/A	Взлом через уязвимость	Взлом через уязвимость в CMS Magento	Уязвимости CMS: WordPress, Shopify, BigCommerce	Взломан внешний сервис статистики onthe.io, изменен код js-скрипта	Supply chain attack. Код внедрен владельцами библиотеки.
Время присутствия	N/A	15 дней	5 месяцев	N/A	1-3 дня	> 4 месяцев
Последствия	N/A	Похищены данные банковских карт 380 000 клиентов	Похищены персональные данные клиентов (1.5 млн посетителей / день), банковские карты 500 000 клиентов.	Похищены данные банковских карт	Неработоспособность ресурсов. Политические лозунги на страницах.	Редирект пользователей мобильных устройств на сайты онлайн-букмекеров.
Ущерб	N/A Устранено через 4 часа после публикации статьи Dr. Web	2 280 000 000 £ + штраф 20 000 000 £ по GDPR		N/A	N/A	N/A Неработоспособность сайтов после блокировки домена.

Инциденты


Регистрация
Авторизация
услуги
Внимание!
Госуслуги: Вам положена социальная выплата
Закрыть
Перейти

Frontend-приложения

DPA ANALYTICS

1. Работают в «слепой» зоне для ИБ
2. Средства защиты и анализаторы ИБ не обнаруживают актуальные угрозы
3. Время присутствия вредоносного кода – недели / месяцы в известных инцидентах
4. Часто игнорируются ИБ-специалистами
5. Максимальная монетизация для злоумышленника

Что необходимо контролировать?

DPA ANALYTICS

1

Скрипты и другие активные элементы

2

Сетевые запросы, выполняемые js-приложением в браузере

3

API-браузера, используемое js-приложением

DPA ANALYTICS

“Единственное место, где можно обнаружить изменения и признаки вредоносной активности – это браузер пользователя, где страница полностью собрана и выполнен весь JavaScript-код”

PCI DSS 4.0.1

Способы контроля

DPA ANALYTICS

1

Ручной анализ

- Долго
- Необходимо выполнять все Use Case при каждом анализе
- Ограниченная видимость

2

Решения класса Frontend Sandbox (Frontend Application Security Testing - FAST)

- Анализатор
- Автоматизированное выполнение Use Case
- Глубокий анализ на контентном слое браузера

3

Решения класса Frontend Observability

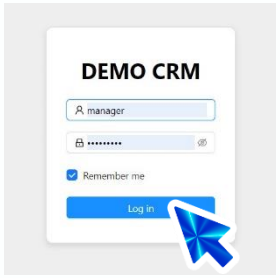
- JS-агент работает на страницах приложения в браузерах пользователей
- Контроль в реальном времени

Frontend Application Security Testing (FAST)

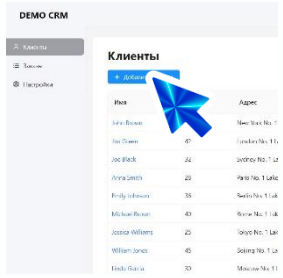
DPA ANALYTICS

Открытие страницы

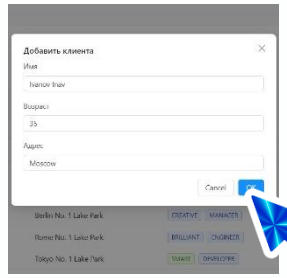
Ввод логина, пароля, клик по кнопке



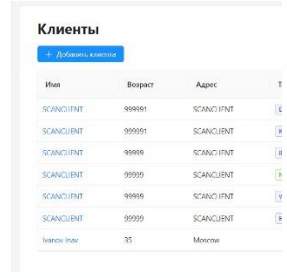
Успешная аутентификация, редирект



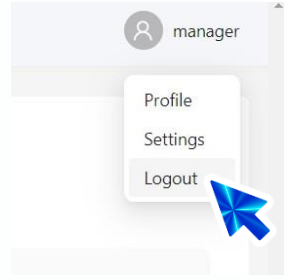
Открытие формы создания клиента, заполнение данных



Клик по кнопке, клиент успешно добавлен



Клик по кнопке Logout, закрытие страницы



Автоматизированное выполнение E2E-сценария (Use Case)

Элементы

script, iframe, embed, form и др.

Запросы

xhr, fetch, img, websocket и др.

API браузера

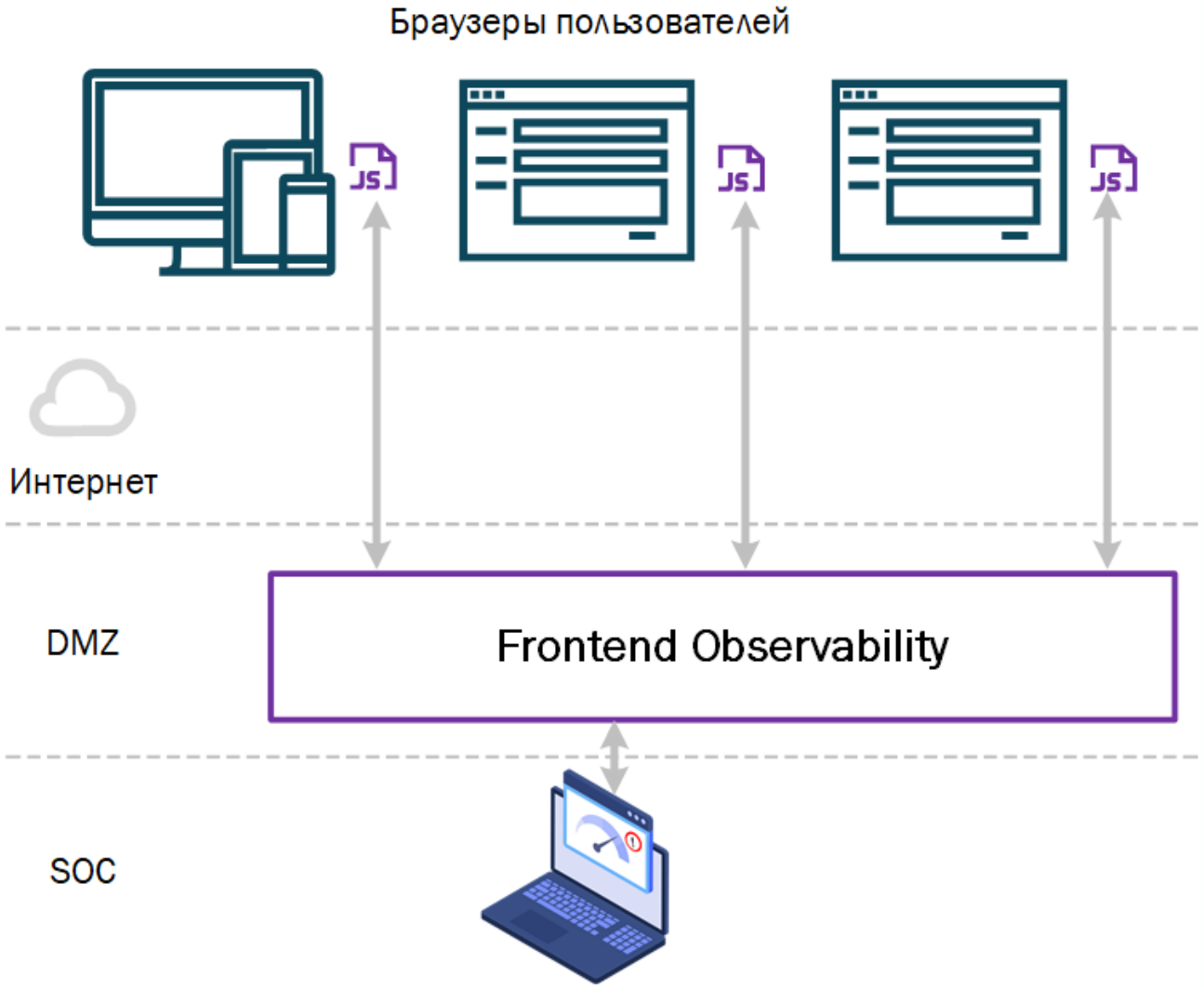
eval, clipboard, geolocation, cookie, notification и др.

Software Bill of Behavior (SBOB)

Контентный слой браузера

Frontend Observability Platform (FOP)

- JS-агент, подключенный на страницы
- Инвентаризация скриптов и других активных элементов в реальном времени
- Мониторинг всех сетевых запросов js-приложения
- Выявления фактов появления на страницах конфиденциальных данных
- Применение правил корреляции
- Выявление критичных инцидентов
- Отправка событий в SIEM / SOC



Модель зрелости обеспечения ИБ frontend-приложений

- Ручной анализ
- Периодическая инвентаризация всех скриптов

- Регулярный анализ приложения в frontend-sandbox (FAST-анализаторе), круглосуточно / каждый час.
  - Инвентаризация всех скриптов / активных элементов
  - Карта сетевых запросов
  - Используемые API-браузера
- Реагирование на любые отклонения от доверенной политики
- Особое внимание после новых релизов приложения

- Мониторинг поведения frontend-приложения в реальном времени:
  - Скрипты / активные элементы
  - Все сетевые запросы браузера
- Контроль объема / состава данных, передаваемых сторонним сервисам, контроль трансграничной передачи данных
- Выявление аномалий
- Реагирование на срабатывания правил корреляции (инциденты, затрагивающие значительное число пользователей)

Безопасное frontend-приложение

Требования регуляторов

НКЦКИ «Рекомендации по повышению уровня защищенности российских web-приложений» № ALRT-20220311.1 от 11 марта 2022 г.

19. **Перед использованием** на web-ресурсах JavaScript-кода, подгружаемого со сторонних ресурсов, **осуществлять его проверку на предмет вредоносного воздействия** на отображаемую в браузерах пользователя информацию и возможность кражи аутентификационных данных и файлов-cookie пользователей.

20. Осуществлять периодическую проверку хэш-сумм, используемых JavaScript. В случае изменения хэш-сумм отключать использование JavaScript на сайте и **выполнять повторную проверку функциональности**.

PCI DSS 4.0.1 (Требования вступают в силу 31.03.2025)

6.4.3 Все скрипты платежных страниц, которые загружаются и выполняются в браузере пользователя, управляются следующим образом:

- Реализован метод подтверждения **авторизации каждого скрипта**.
- Реализован метод, обеспечивающий **целостность каждого скрипта**.
- Актуальная **инвентаризация всех скриптов** с письменным обоснованием необходимости каждого из них.

11.6.1 Обнаружение и реагирование на несанкционированное изменение платежных страниц:

- Контроль **изменений на платежных страницах**
- Контроль **изменений HTTP-заголовков**
- Оповещение персонала о несанкционированных изменениях

Задачи средств защиты веб-приложений

DPA ANALYTICS

WAF

- Обнаружение признаков атак во входящих HTTP-запросах пользователей по правилам / сигнатурам

FAST / FOP

- Обнаружение утечек информации в frontend-приложениях в браузере
- Обнаружение НДВ / вредоносного кода в js-зависимостях
- Обнаружение злоупотреблений доступом системами аналитики и другими внешними js-сервисами

NGFW

- Обнаружение сетевых атак

Сканер уязвимостей

- Обнаружение пакетов ОС, ПО с известными уязвимостями
- Обнаружение уязвимостей backend-приложения

Что делать?

DPA ANALYTICS

- Ответить на вопрос: «Я знаю/уверен, что делает frontend-приложение прямо сейчас? Куда отправляет данные?»
- Выполнять мониторинг поведения frontend-приложений
- Использовать средства автоматизации (FAST/FOP)
- Согласование с ИБ:
  - изменения js-кода;
  - новые хосты, с которыми взаимодействует js-приложение;
  - новые сторонние js-сервисы;
  - новые API-браузера, используемые js-кодом.

Telegram-канал FrontSecOps

DPA ANALYTICS

- Разбор инцидентов
- Лучшие практики
- Frontend Observability
- DevSecOps для frontend-приложений
- Обзоры инструментов

@FRONTSECOPS

Спасибо!

Михаил Парфенов

Главный архитектор по ИБ

dpa-analytics.ru

info@dpa-analytics.ru

https://t.me/FrontSecOps

@FRONTSECOPS