Михаил Парфенов
Application Security Architect

Взгляд в темноту... Безопасность frontend-приложений от модели угроз до secure by design

PHDAYS FEST от positive technologies

- 10 лет – в ИБ (управление рисками, vulnerability management, безопасность веб-приложений)
- 5 лет - Application Security Architect, DevSecOps
- Исследую методы поведенческого анализа frontend-приложений в DevSecOps (FAST, frontend-sandbox, frontend observability)
- Управляю разработкой FAST-анализатора в DPA Analytics
- Telegram-канал @FrontSecOps

Михаил Парфенов

Application Security Architect
DPA Analytics

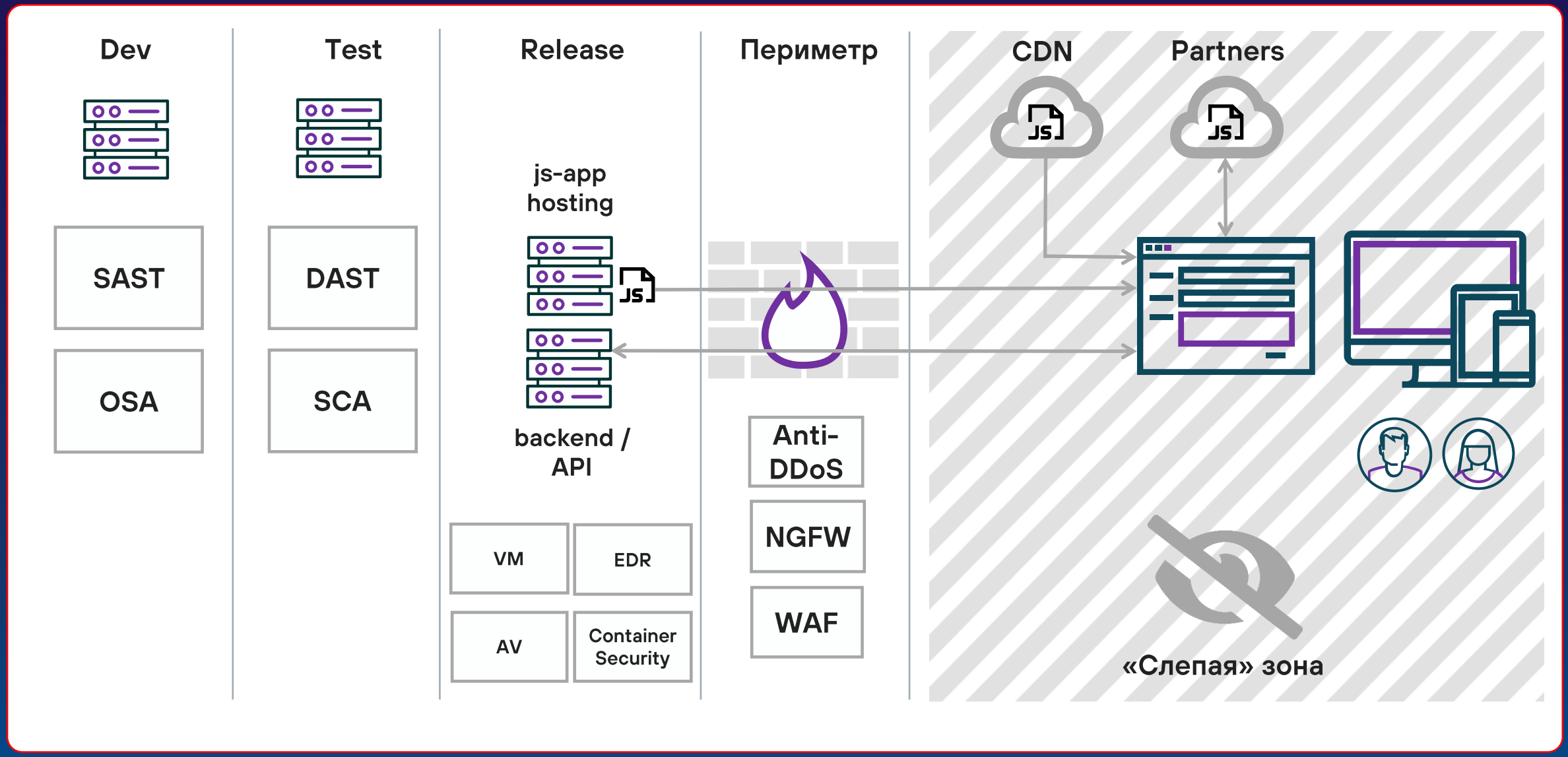
План доклада

1. Frontend-приложения всегда в «слепой» зоне. Обзор инцидентов
2. Результаты исследования безопасности российских frontend-приложений
3. Модель угроз для frontend-приложений
4. Анализ поведения frontend-приложения в DevSecOps – путь к Secure by Design
5. Что делать?

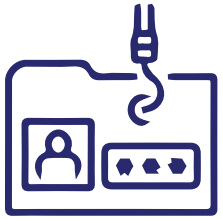
01

Frontend – всегда в «слепой» зоне

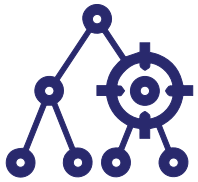
Безопасность веб-приложений



В чем выгода злоумышленника от внедрения вредоносного кода в frontend-приложение?



Сбор и кража критичных данных со страниц web-приложения



Загрузка на страницу других скриптов злоумышленника

Выполнение действий от имени пользователя

Показ пользователю фишинговых баннеров

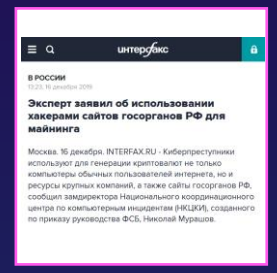
Майнинг криптовалюты в браузере пользователя

Заражение устройства пользователя через уязвимости браузера

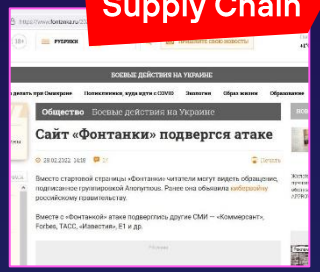
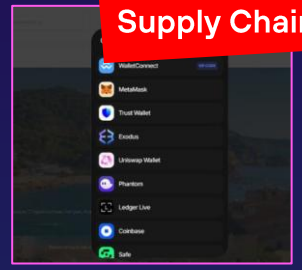
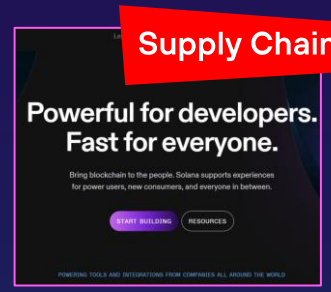
Как вредоносный код может попасть в frontend-приложение?

Вектор	Примеры инцидентов
Зависимости js-приложения / сторонние библиотеки с новыми версиями	2024 – вредоносный код в библиотеке Polyfill.js 2024 - вредоносный код в библиотеке lottie-player (LottieFiles) 2024 - вредоносный код в библиотеке solana/web3.js
Компрометация внешнего js-сервиса, подключенного на веб-страницах (системы аналитики, счетчики, онлайн-чаты и т.п.)	2022 - Взлом сервиса onthe.io, в js-скрипт добавлен вредоносный код, затронуло сайты СМИ «Коммерсантъ», Forbes, РБК, ТАСС, Известия и других крупных российских компаний 2022 – Взлом виджета Минэкономразвия РФ. Дефейс сайтов, использующих виджет: ФСИН, ФССП, ФАС, Минкульта, Минэнерго, Росстата и других
Компрометация учетной записи от Google Tag Manager (или его аналогов)	2021 - В 316 интернет-магазинах обнаружен js-сниффер, скрытый в Google Tag Manager
Компрометация веб-сервера	2018 – British Airways – размещен js-сниффер, похищающий данные банковских карт
Умышленное размещение кода сотрудником / подрядчиком (разработчик, маркетолог, администратор сайта)	2019 – по информации НКЦКИ, были обнаружены факты размещения js-майнеров на сайтах государственных и муниципальных организаций
Копирование кода из недоверенных источников, генерация кода «плохой» нейросетью	-

Инциденты

						
Год	**2017**	**2017**	**2018**	**2019**	**2019**	**2021**
Инцидент	Ticketmaster – js-сниффер на странице с платежной формой	Размещены iframe с неизвестными доменами в Нидерландах	Злоумышленник встроил в одну из js-библиотек js-сниффер	В 100 000+ интернет-магазинов встроен js-сниффер	По информации НКЦКИ на сайтах гос. организаций обнаружены js-майнеры	В 316 интернет-магазинах обнаружен js-сниффер, скрытый в Google Tag Manager
Вектор	Взломан внешний сервис Inbenta	N/A	Взлом через уязвимость	Взлом через уязвимость в CMS Magento	N/A	Уязвимости CMS: WordPress, Shopify, BigCommerce
Время присутствия	> 8 месяцев	N/A	15 дней	5 месяцев	N/A	N/A
Последствия	Похищены данные банковских карт > 40 000 клиентов	N/A	Похищены данные банковских карт 380 000 клиентов	Похищены данные банковских карт 500 000 клиентов (1.5 млн посетителей / день)	N/A	Похищены данные банковских карт
Ущерб	N/A	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

Инциденты

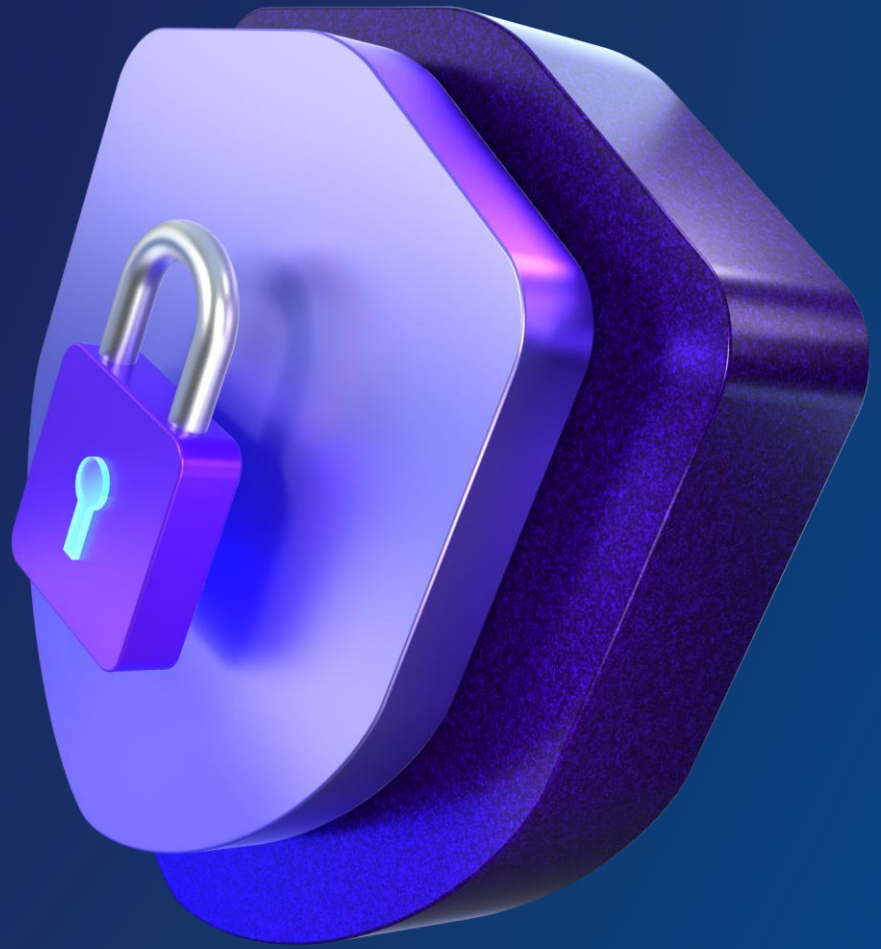
					
Год	**2022**	**2022**	**2024**	**2024**	**2024**
Инцидент	Внедрен код на сайты СМИ «Коммерсантъ», Forbes, РБК, ТАСС, «Известия» и других крупных компаний	Внедрение кода в виджет Минэкономразвития Госмониторинг	Внедрен вредоносный код в библиотеку Polyfill.js. Код выполнялся на > 350 000 веб-приложений	Вредоносный код в библиотеке lottie-player	Вредоносный код в библиотеке solana/web3.js
Вектор	Взломан внешний сервис статистики onthe.io, изменен код js-скрипта	N/A	Supply chain attack. Код внедрен владельцами библиотеки	Компрометация npm-библиотеки / фишинг атака на разработчика	Компрометация npm-библиотеки / фишинг атака на разработчика
Время присутствия	1-3 дня	1 день	> 4 месяцев	3 дня в NPM	1 день в NPM
Последствия	Неработоспособность ресурсов. Политические лозунги на страницах	Политические лозунги на страницах сайтов ведомств, использующих виджет	Редирект пользователей мобильных устройств на сайты онлайн-букмекеров	Показ фишинг окна с предложением подключить криптовалютный кошелек -> вывод $	Кража приватных ключей, вывод денежных средств
Ущерб	N/A	N/A	N/A	> 700 000 $	> 160 000 $

Frontend-приложения

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

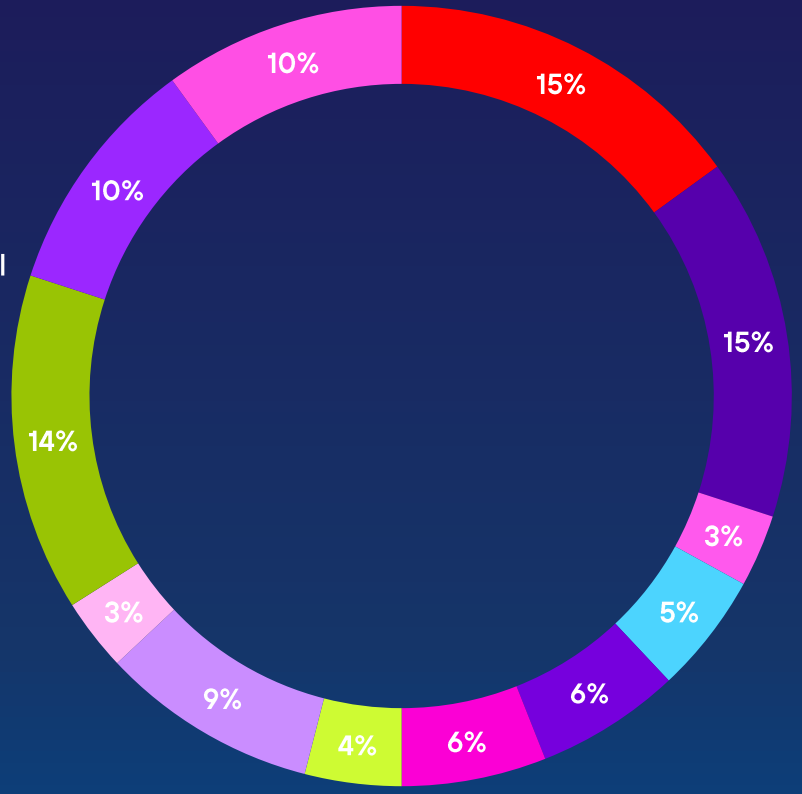
02

Исследование безопасности российских frontend-приложений



Объект исследования

- Промышленность
- ИТ и сервисы
- Услуги
- Медиа
- E-commerce / маркетплейсы
- Личные кабинеты
- Телекоммуникации
- Медицина
- Транспорт
- Банки / финансовый сектор
- Интернет-банки / ДБО
- Вендоры и интеграторы по ИБ



3100 российских frontend-приложений

- Публично доступные
- Коммерческие компании
- Анализ 1 страницы в браузере (frontend-sandbox) по простейшему E2E-сценарию
- Без влияния/нагрузки на приложения
- Q2 2025

Основные показатели

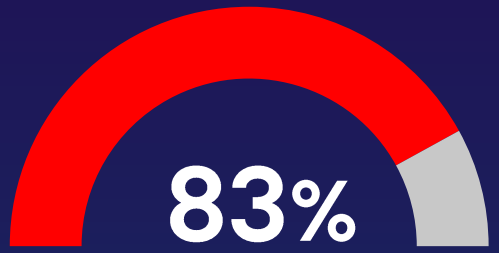
49

File 27 | Inline 22

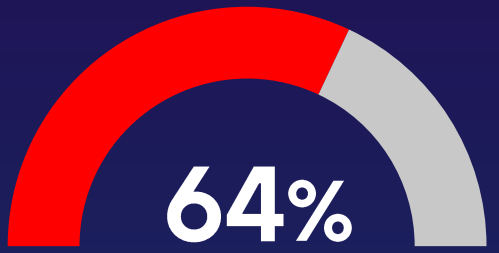
Среднее количество скриптов на странице

3.3МБ

Средний объем js-кода



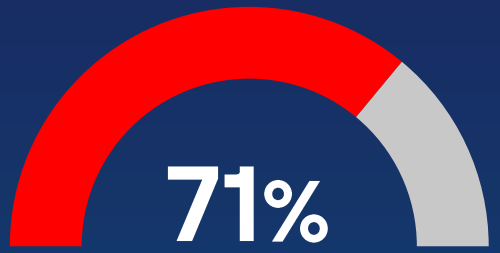
Наличие скриптов со сторонних хостов



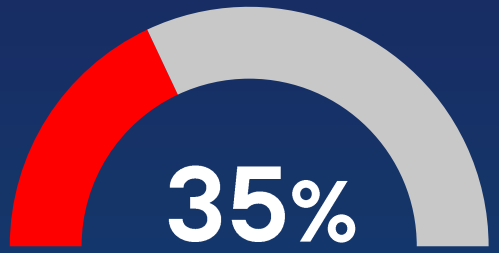
Наличие скриптов с зарубежных хостов

14

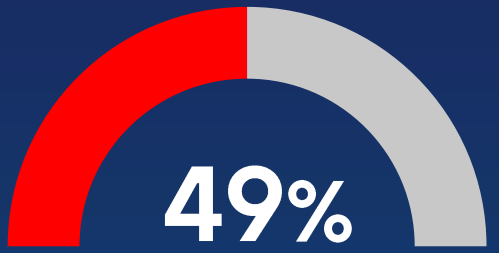
Среднее количество хостов, на которые отправляются запросы



Наличие сетевых запросов на зарубежные хосты

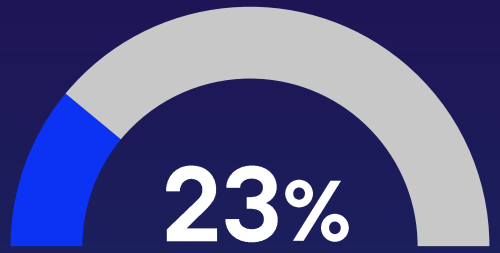


Наличие iframe с зарубежных хостов



Наличие вызовов функции eval()

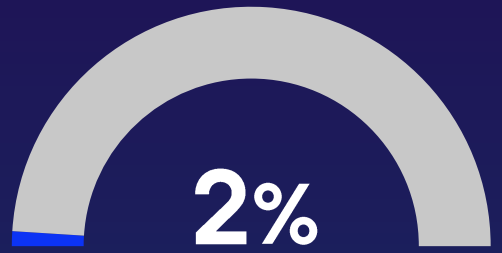
Основные показатели



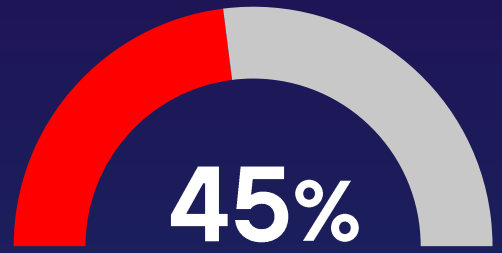
Наличие заголовка Content Security Policy (CSP)

7 / 100

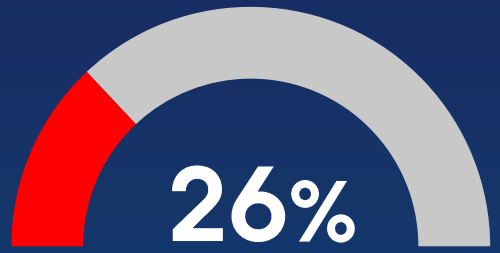
Оценка конфигурации CSP



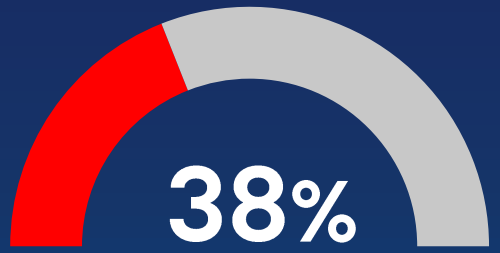
Использование Subresource Integrity (SRI)



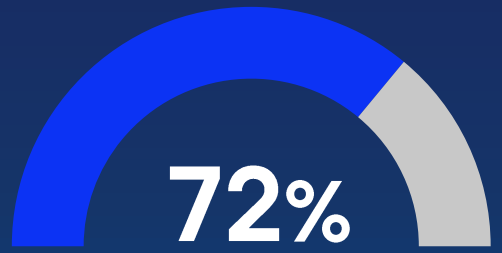
Наличие ошибок в консоли браузера



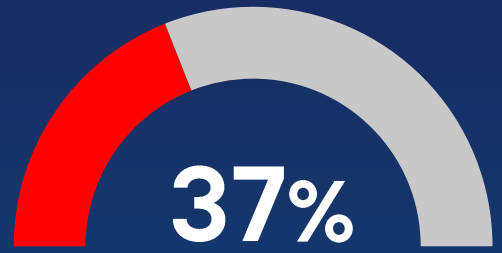
Наличие Google Tag Manager (GTM)



Наличие Google Analytics

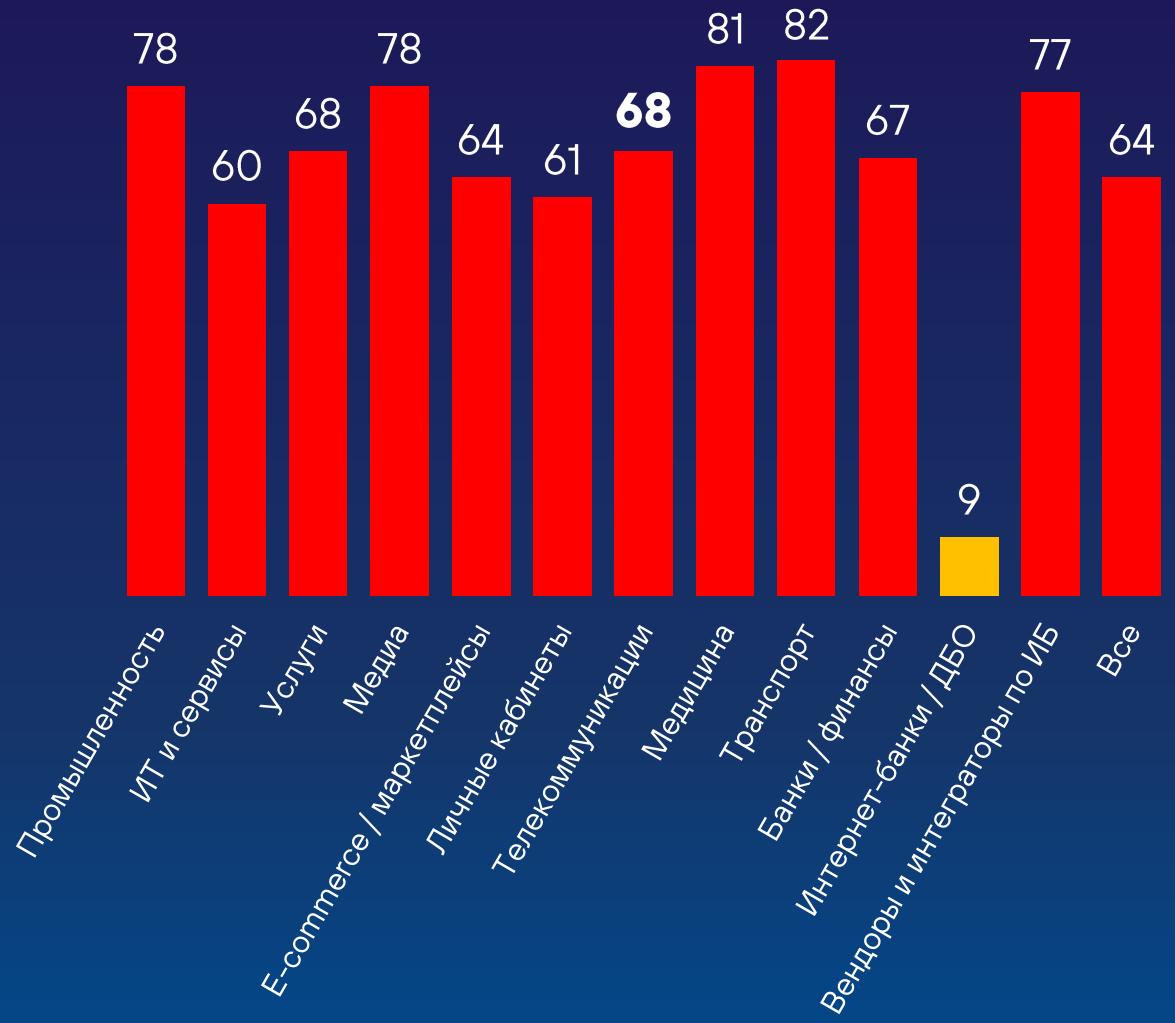


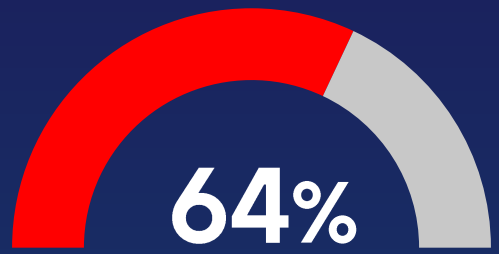
Наличие Яндекс Метрики



Наличие Яндекс Метрики с включенным Вебвизор

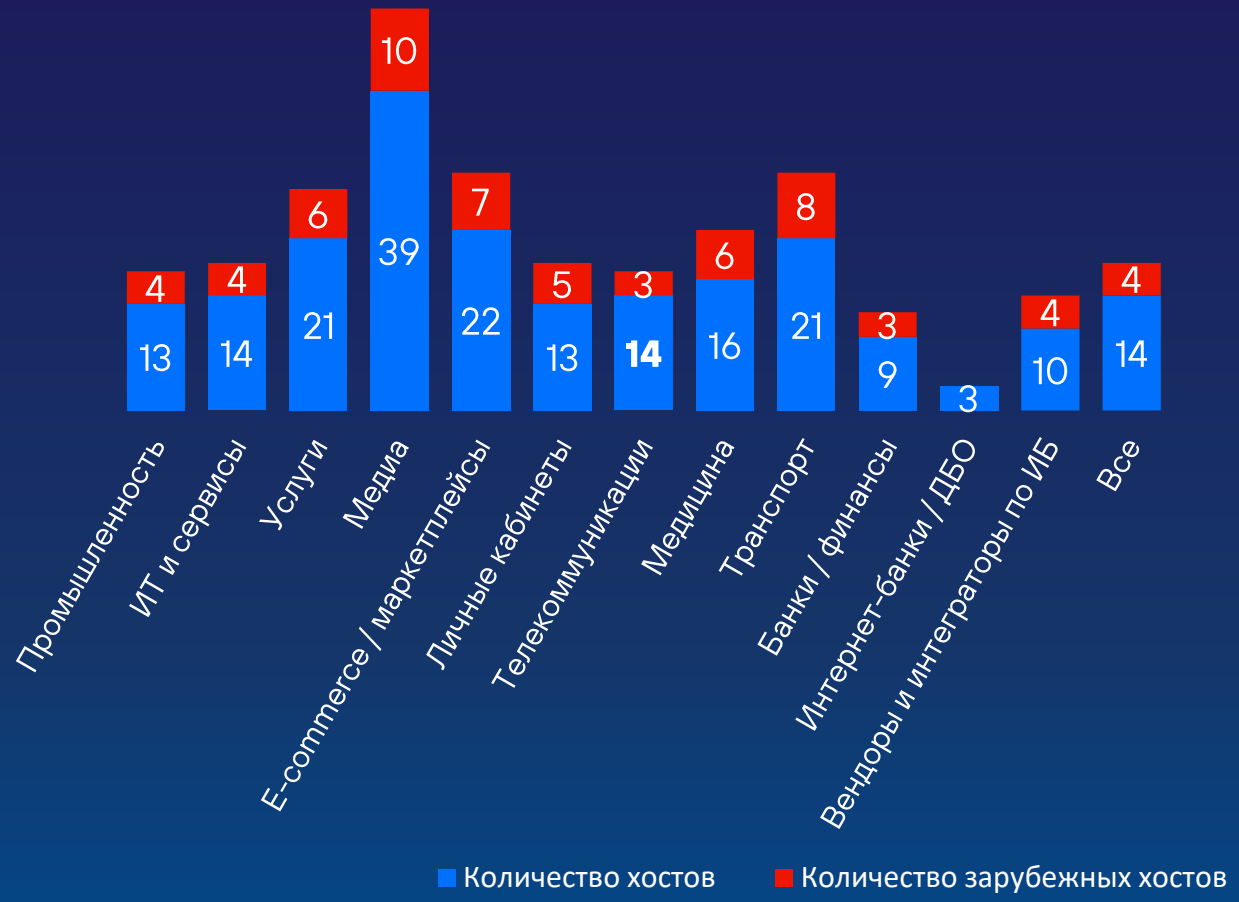
Скрипты с зарубежных хостов





Наличие скриптов с зарубежных хостов по всем категориям

Количество хостов, на которые отправляются сетевые запросы



14

Среднее количество хостов, на которые отправляются запросы

4

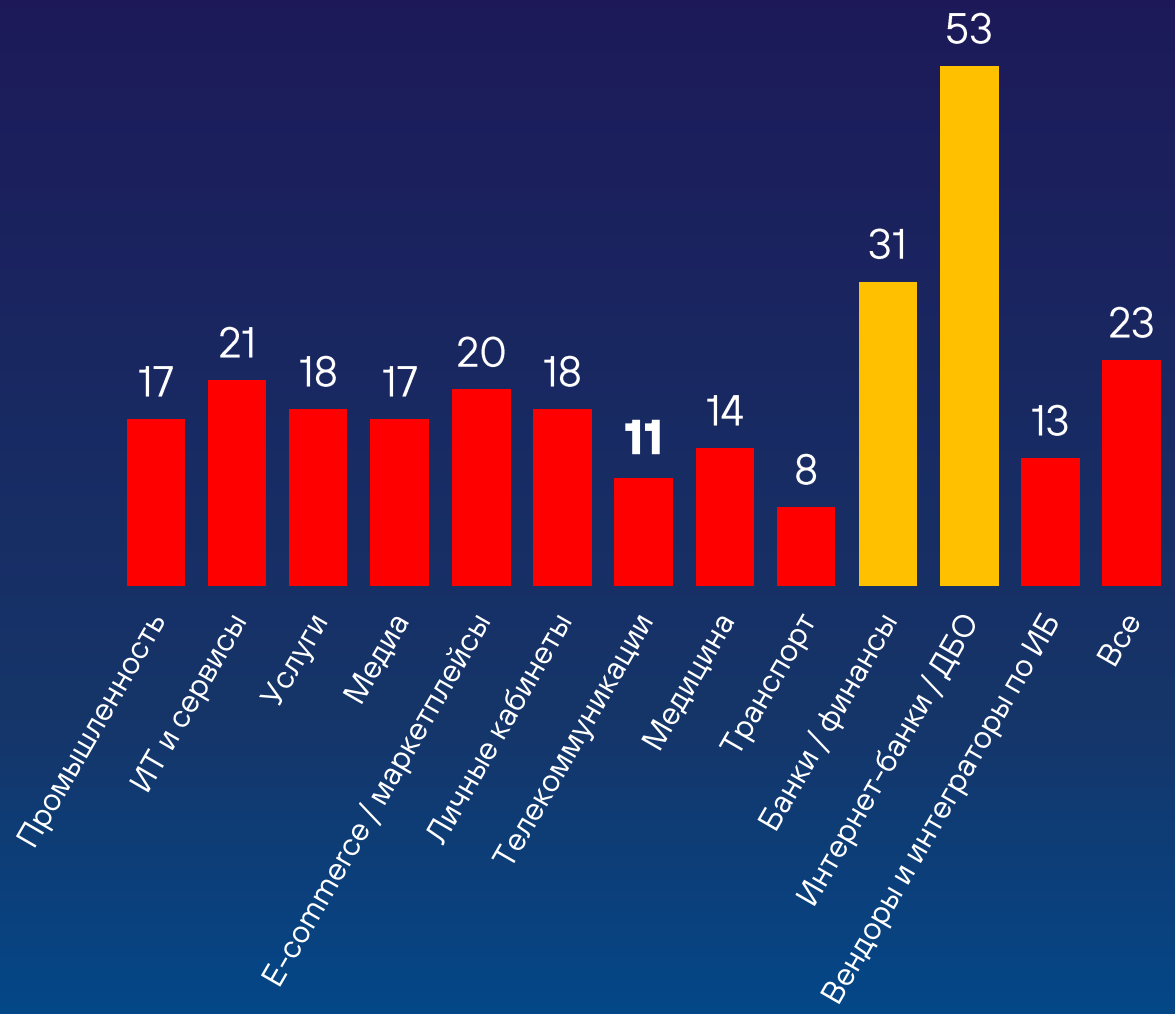
Среднее количество зарубежных хостов, на которые отправляются запросы

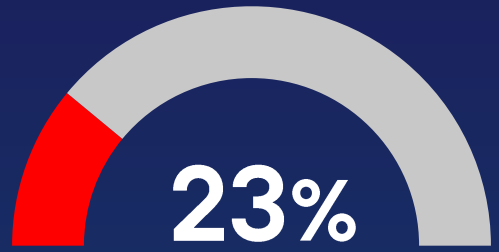
Content Security Policy (CSP)

- «CSP не предназначена для использования в качестве основной защиты от инъекций контента (в том числе XSS), а подходит для снижения последствий от таких атак» Спецификация CSPv2
- Не защищает от кражи данных со страницы через механизм навигации.
- Почти всегда ослабляется по требованиям бизнеса.
- Ослабляется из-за плохих практик написания кода (например, инлайн-скрипты или код в атрибутах событий)

Статья «Content Security Policy (CSP) защитит от js-снифферов и утечек?»

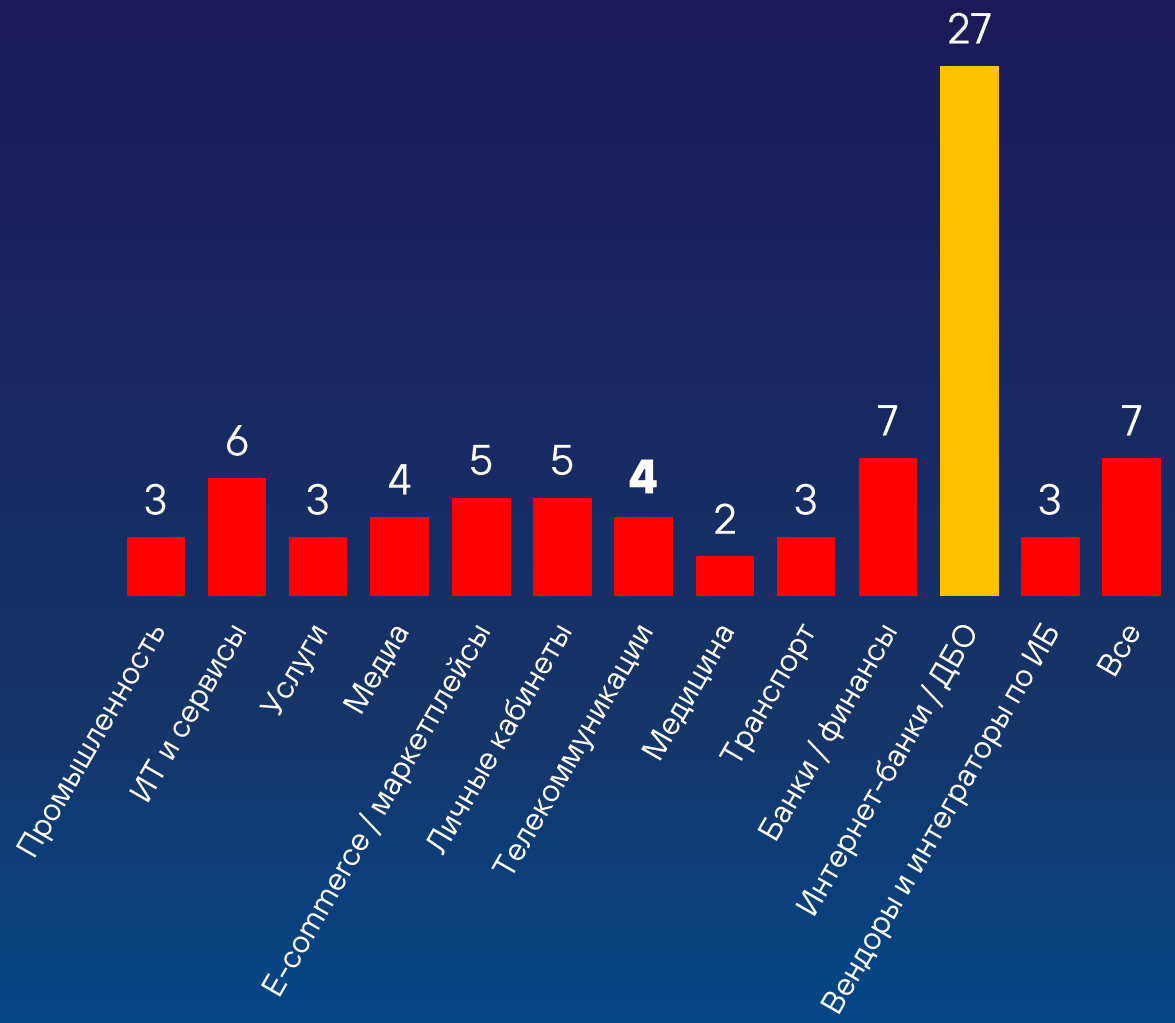
Content Security Policy (CSP)
Наличие заголовка





Наличие заголовка CSP по всем категориям

Content Security Policy (CSP)
Оценка конфигурации



7 / 100

Средняя оценка конфигурации CSP по всем категориям

Google Tag Manager (GTM)

- GTM – скрипт, который позволяет добавлять на страницы frontend-приложения другие скрипты / теги по определенным триггерам.
- Примеры триггеров: конкретный URL страницы, время, тип устройства, ОС глубина скролла, таймер и т.п.
- Используется маркетологами, например, чтобы добавлять на разные страницы код разных счетчиков или выполнять скрипты достижения целей и т.п., при этом не отвлекая разработчиков по каждому изменению.
- Все скрипты хранятся на серверах Google и управляются из интерфейса аккаунта Google.

Другими словами:

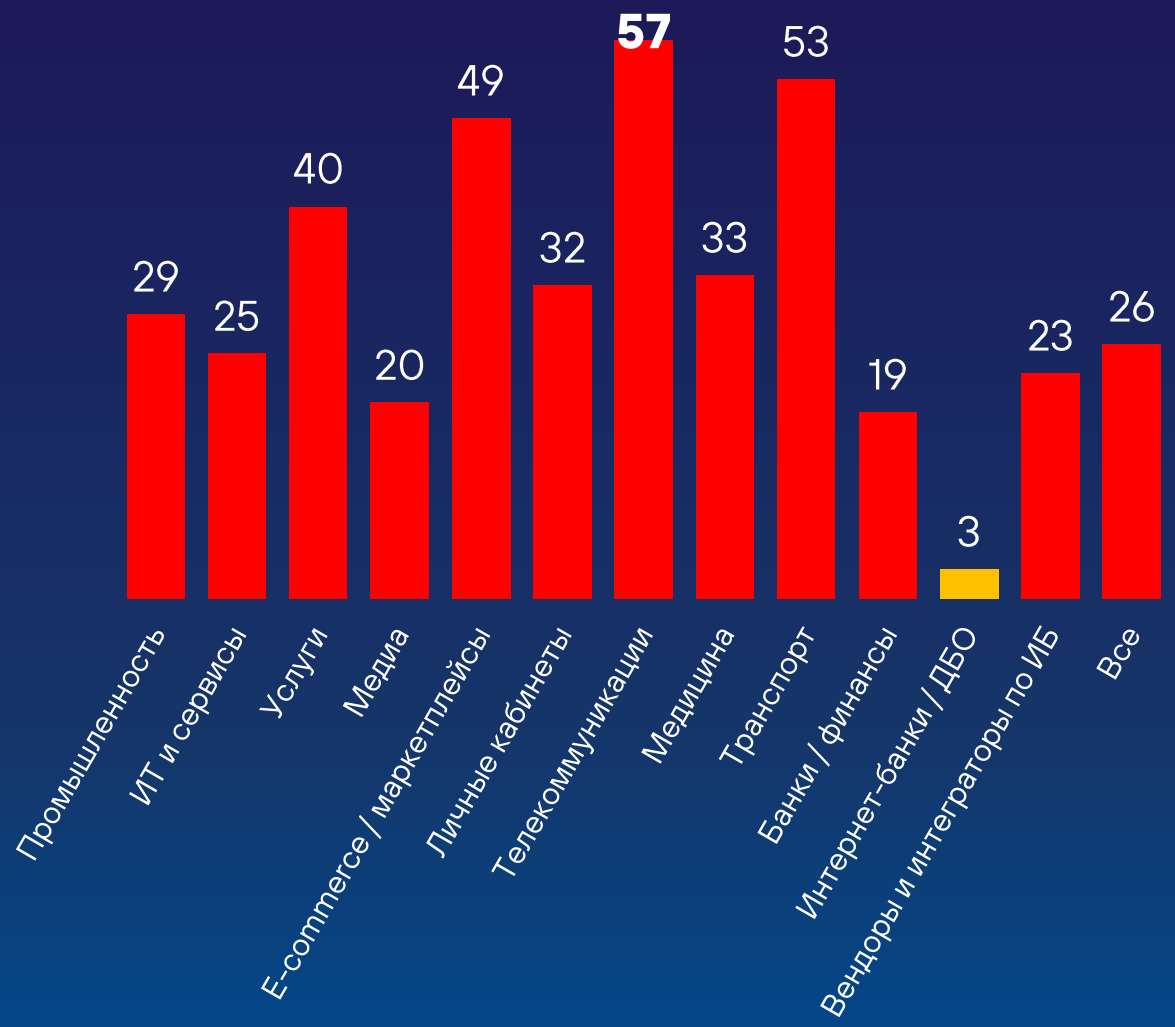
frontend-бэкдор

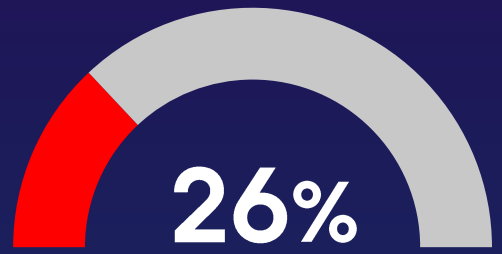
frontend-шелл

В 2019 злоумышленники добавляли на страницы взломанных интернет-магазинов скрипт GTM, который при инициализации добавлял js-сниффер.

Это позволяло «скрыться» от поверхностного ручного анализа безопасности

Google Tag Manager (GTM)





Использование GTM по всем категориям

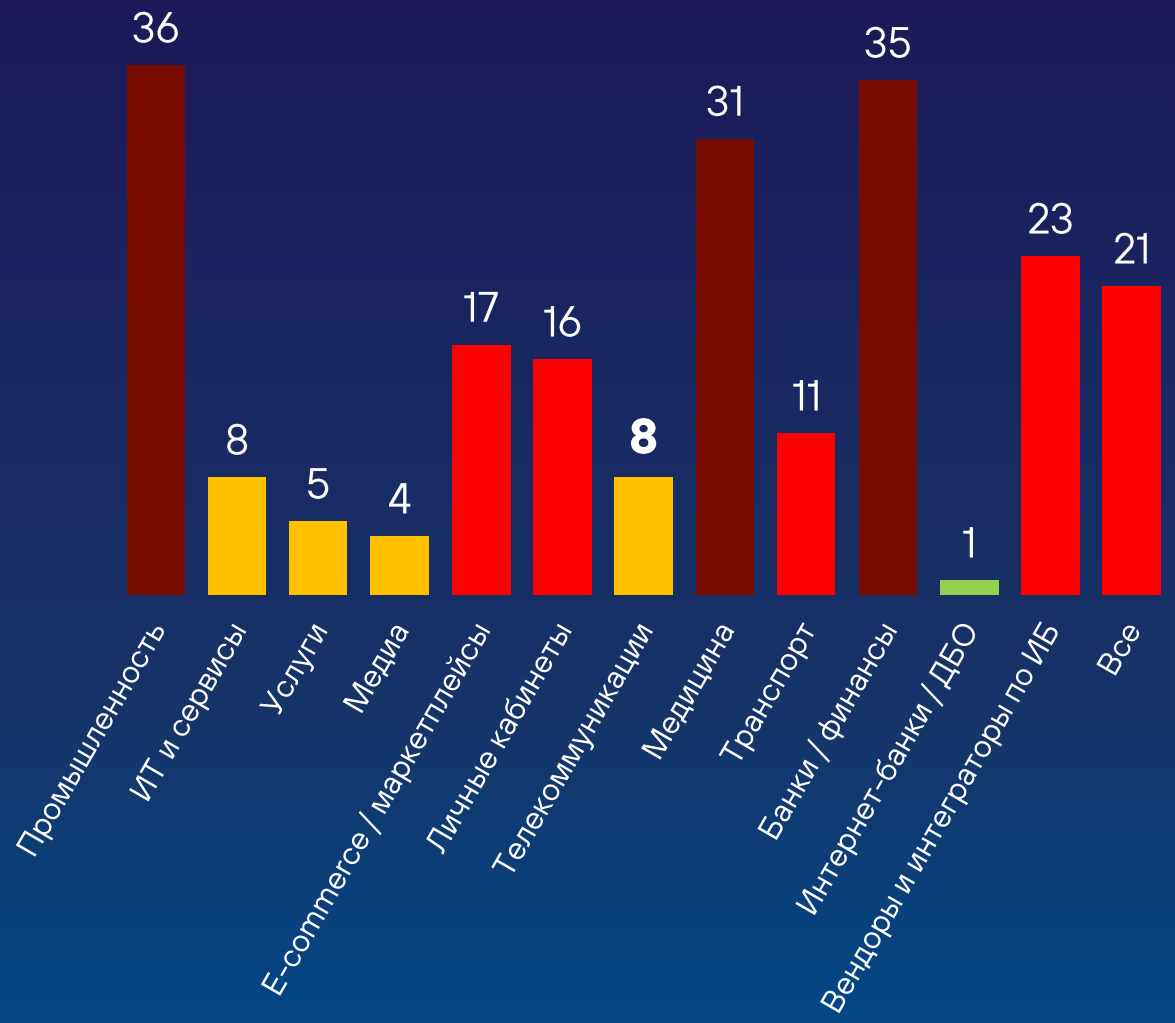
Риски:

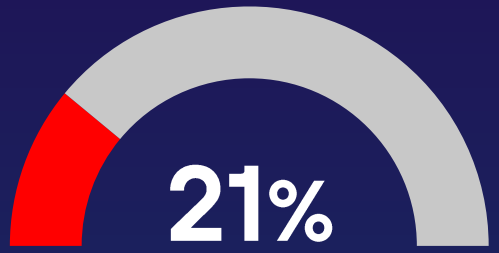
- Обычно управлением GTM занимается маркетолог, а не ИБ
- Компрометация Google Аккаунта
- Возможна блокировка IP сервиса
- Зарубежный сервис

Google Tag Manager (GTM) Как снизить риск?

- Отказаться от использования GTM, если возможно (либо перейти на российские аналоги)
- У ИБ / AppSec должен быть доступ к Google аккаунту управления GTM
- Контроль / согласования / управление изменениями в конфигурации GTM
- Анализ поведения приложения для оперативного обнаружения вредоносных действия и реагирования (например, с помощью FAST-анализаторов)

Битрикс Веб-Аналитика





Используют Битрикс Веб-Аналитика

> 650 frontend-приложений

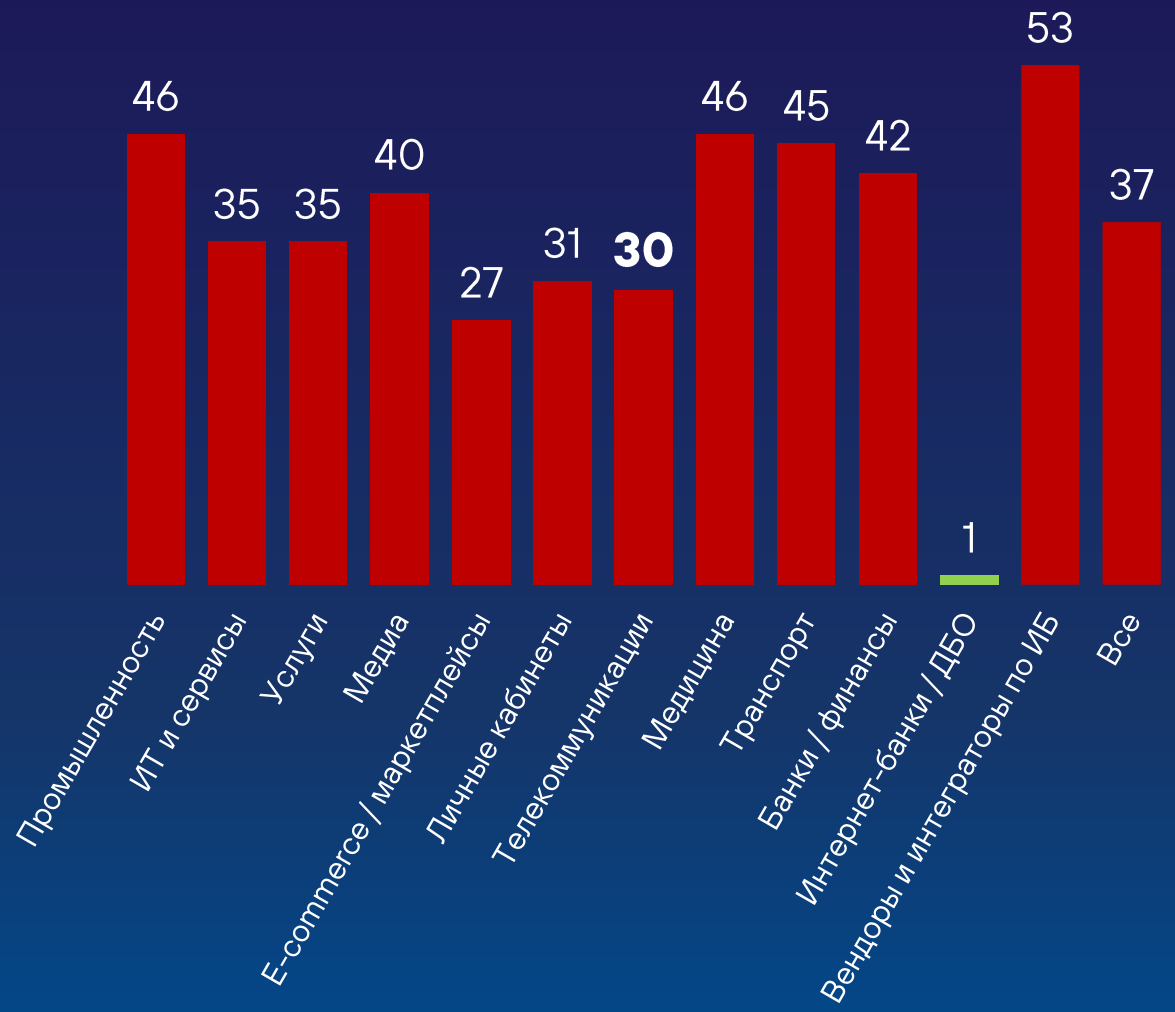
Скрипт: https://bitrix.info/ba.js
Эндпойнт: https://bitrix.info/bx_stat

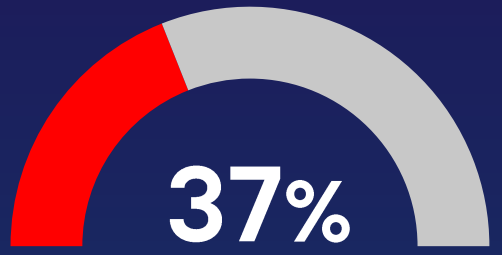
AS number	AS16509
AS name (ISP)	Amazon.com, Inc.
IP-range/subnet	54.154.0.0/16
Location	Dublin, Leinster, Ireland (IE)

Битрикс Веб-Аналитика

- Компании, использующие CMS, знают, что данные о поведении их клиентов/пользователей (персональные данные) передаются в Ирландию? РКН
- Описано ли это в документации CMS?
- Является ли это недекларированной возможностью (НДВ)?
- Данная функциональность включена по умолчанию? (Privacy by Default?)
- Что будет, если IP-адреса Amazon попадут под блокировку РКН? РКН
- Вам известно, что с 2023 года трансграничная передача ПД без уведомления Роскомнадзора запрещена? РКН
- Это сбор ПД, с использованием БД, находящихся за пределами РФ? РКН
- Это сбор ПД, с использованием иностранных программ и сервисов? РКН
- В согласии/политике конфиденциальности на сайте указано, что данные передаются за границу? РКН
- Должны ли вендоры ПО проверять frontend-часть своих продуктов, чтобы знать о таком поведении?

Яндекс Метрика с включенным Вебвизор





Наличие Яндекс Метрики с включенным Вебвизор

Сбор избыточных данных: все данные со страниц, нажатия клавиш и т.д.

HTML-комментарии

phd x pt

Бывают «безобидные»

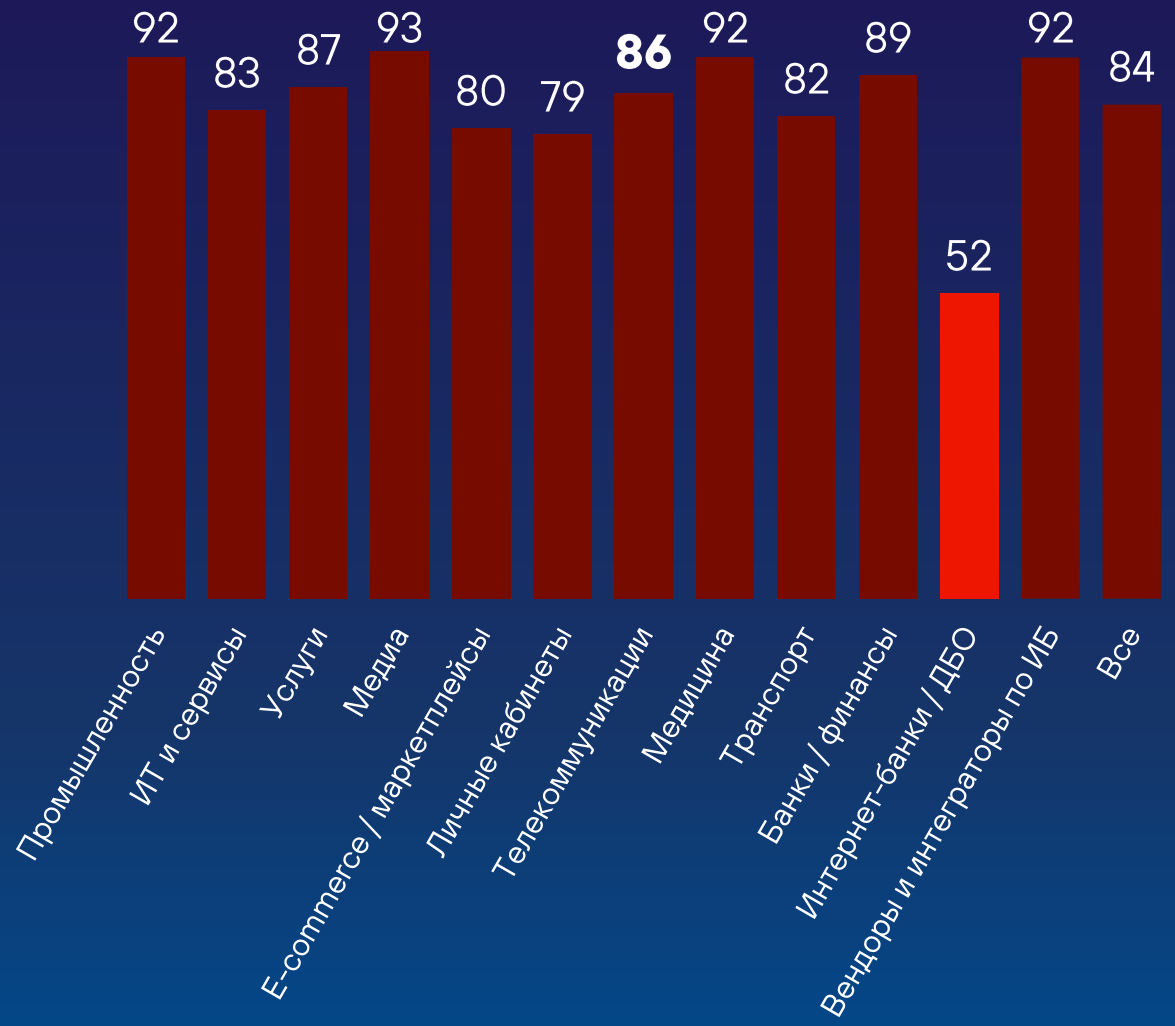
```
304
305 <!-- Yandex.Metrika counter -->
306 <script type="text/javascript" >
307     (function(m,e,t,r,i,k,a){m[i]=m[i]||
308     m[i].l=1*new Date();
309     for (var j = 0; j < document.scripts
310     k=e.createElement(t),a=e.getElements
311     (window, document, "script", "https:
312
313     ym(26993514, "init", {
314          clickmap:true,
315          trackLinks:true,
316          accurateTrackBounce:true,
317          webvisor:true
318     });
319 </script>
320 <noscript><div><img src="https://mc.yan
321 <!-- /Yandex.Metrika counter -->
322
323
```

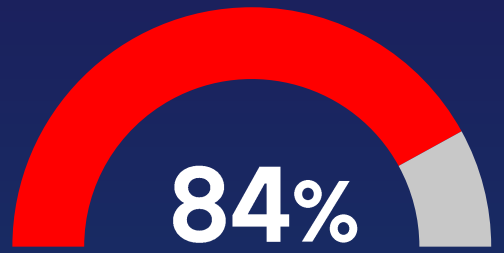
Но бывают «не очень»

```
10 <span class="DDKf1c">
11 <div class="eqA2re UnOTSe" style="height:26px;width:26px">
12 </div>
13 </span>
14
15 <!-- Показания датчиков загрязнения воздуха -
16 замена нулей случайными числами
17 в пределах заданного диапазона - просьба заказчика  -->
18
19 <div class="CA5RN"><div><span class="VuuXrf">
20 </span>
21 </div>
22 <div class="byrV5b"><cite class="qLRx3b tjvcx GvPZzd cHaqb"
```

```
31 </div>
32 </div>
33 <!-- Проверка второго фактора аутентификации
34 временно отключена по требованию заказчика.
35 Учетная запись manager3412 / Qwerty123(@
36 -->
37 </div>
38 </div>
```

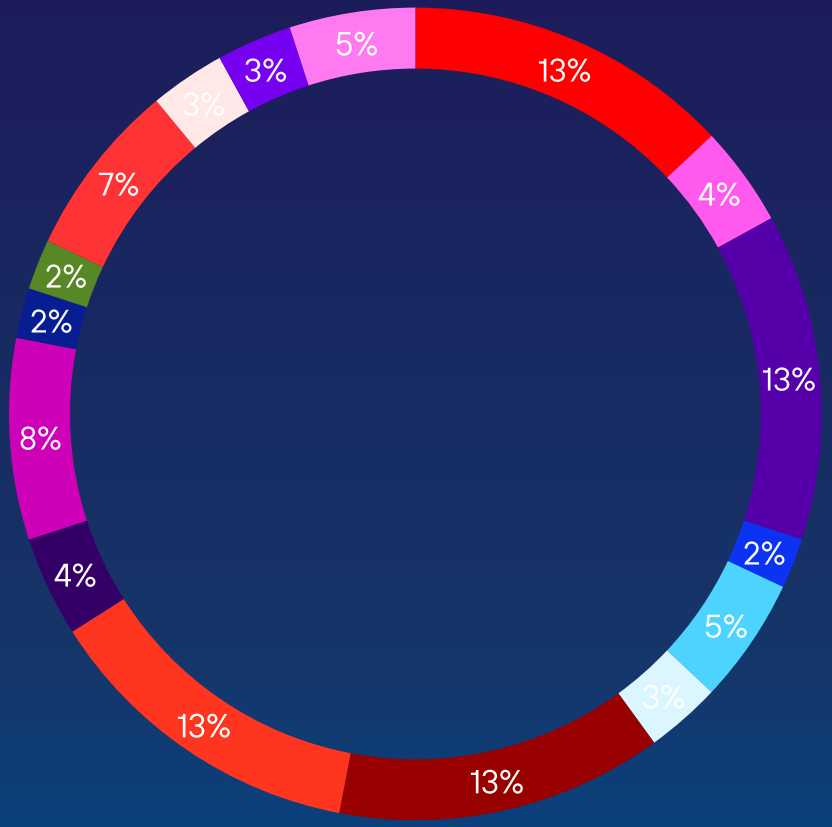
HTML-комментарии





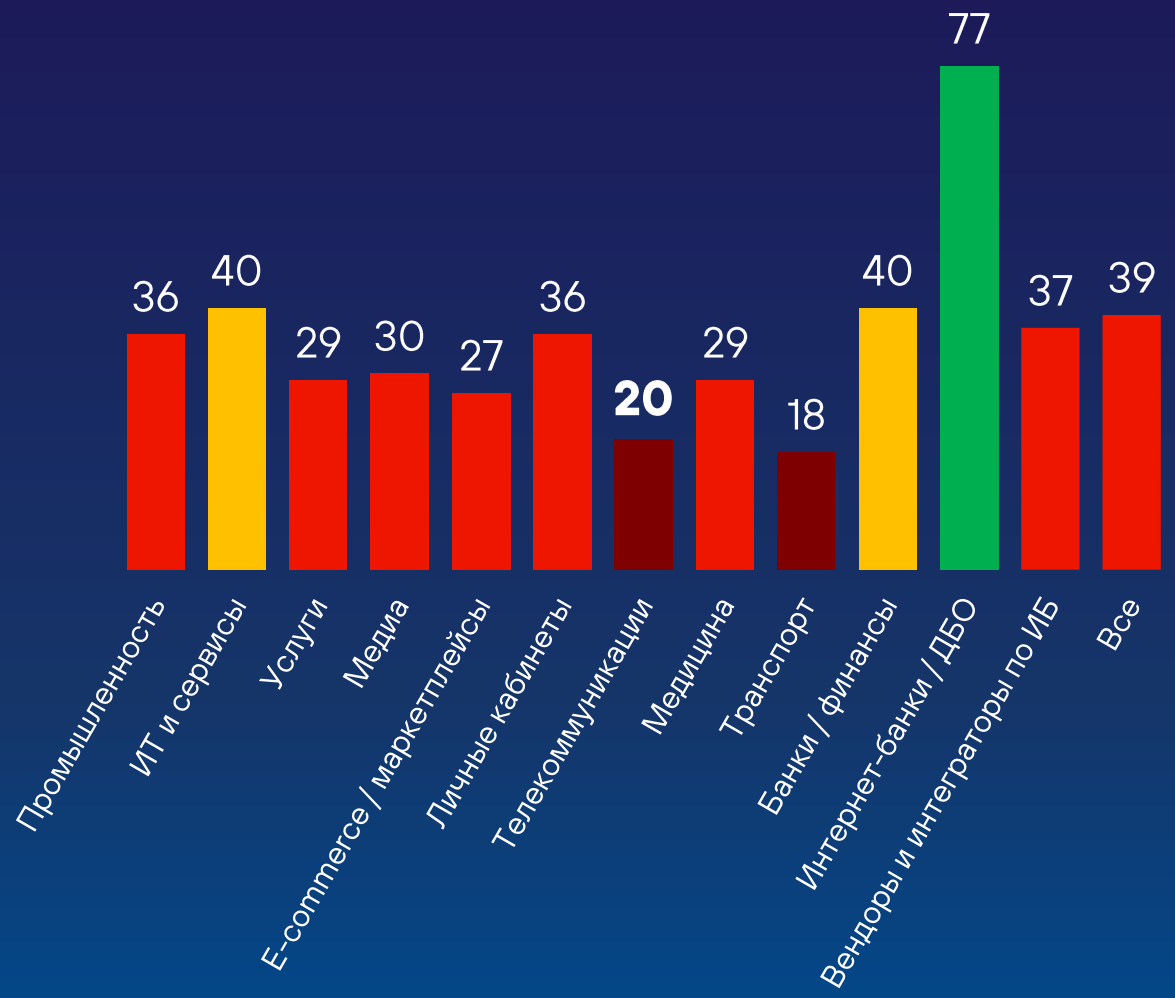
Наличие HTML-комментариев

Расчет общего показателя безопасности



- Конфигурация Content Security Policy (CSP)
- Использование SRI
- Тег-менеджеры
- Скрипты аналитики
- Скрипты аналитики (зарубежные)
- Избыточный сбор данных
- Скрипты с зарубежных серверов
- Передача данных на зарубежные хосты
- Наличие инлайн скриптов
- Фреймы с зарубежных хостов
- Наличие HTML-комментариев
- Наличие ошибок в консоли
- Использование eval()
- Чтение cookie из js-кода
- Показ нотификаций ОС
- Другое

Общий показатель безопасности по отраслям



39 / 100

Средний по всем категориям

Интерпретация общего показателя безопасности

- Само по себе наличие запросов на зарубежные хосты, наличие GTM и другое не является инцидентом, если вы знаете об их наличии.
- «Знаем конечно, у нас разрешено» – не снижает риск.
- Если вы проводите регулярный автоматизированный глубокий анализ поведения frontend-приложения, контролируете изменения, получаете оповещения персонала о несанкционированных изменениях – это снижает риски – можете увеличить оценку своего показателя безопасности на +40.
- Для решения данной задачи можно, например, использовать FAST-анализатор с запуском сканирования каждые 6-12 часов.

Интерпретация общего показателя безопасности

Примеры:

1. Скрипт корпоративной антифрод-системы легитимно использует функцию eval(). После обновления версий зависимостей, скомпрометированная транзитивная зависимость начинает использовать eval() для запуска вредоносного кода. При глубоком анализе выявляется количество вызовов, инициаторы, аргументы функции и т.д.
2. В приложении легитимно используется GTM. Сервер был взломан, злоумышленники добавили еще один скрипт GTM, к которому они имеют доступ. При ручном анализе это может быть пропущено. При глубоком анализе будет обнаружен второй скрипт GTM и его вредоносные действия, запросы и т.д.
3. Злоумышленник добавляет в frontend-часть личного кабинета еще один скрипт Яндекс Метрики, к которой он имеет доступ, с включенным Вебвизор. Далее извлекает собранные ПД/логины/пароли клиентов из UI Яндекс Метрики.

03

Модель угроз для frontend-приложений

Что можно применить из готовых методик, каталогов, инструментов?

PASTA
TARA
...

Про процессы и методики

Универсальные, не про технические угрозы

STRIDE
DREAD
...

Классификации и оценки

Слабо применимы к frontend-приложениям

MITRE CAPEC
OWASP
WASC
БДУ ФСТЭК (частично)
...

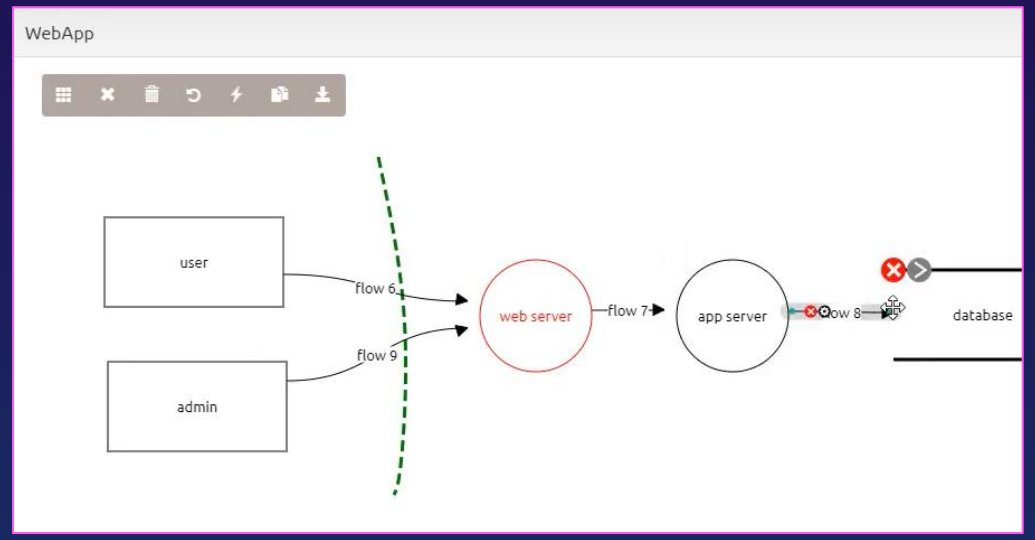
Про приложения, посмотрим подробнее

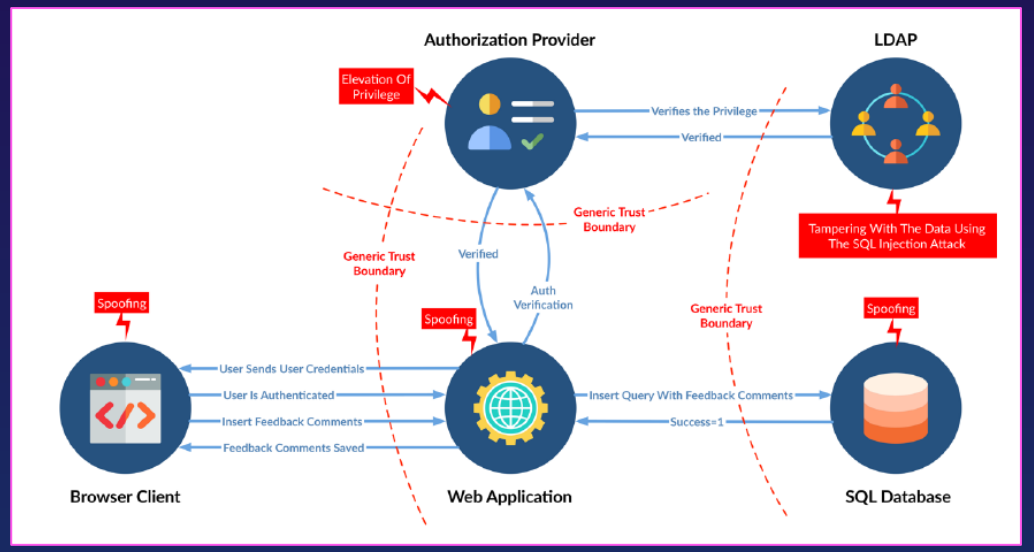
MS Threat Modeling Tool
OWASP Threat Dragon
...

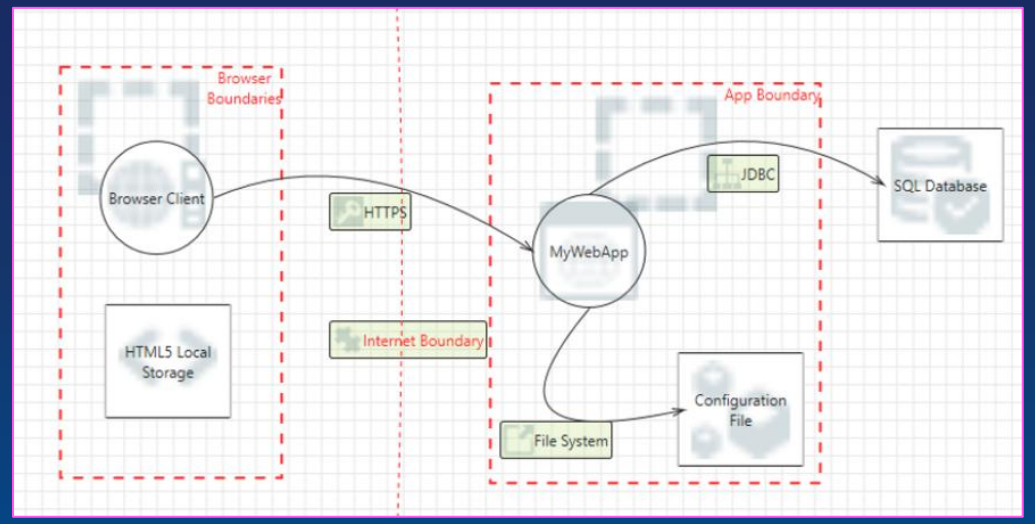
Инструменты DFD-based

Слабо применимы к frontend-приложениям

Инструменты







- Подходят для ИС (архитектура, интеграции, взаимодействия и т.д.)
- Не применимы к браузерному frontend-приложению

MITRE CAPEC

frontend

CAPEC-588: DOM-Based XSS
CAPEC-103: Clickjacking
CAPEC-148: Content Spoofing
CAPEC-472: Browser Fingerprinting
CAPEC-85: AJAX Footprinting
CAPEC-37: Retrieve Embedded Sensitive Data

frontend + backend

CAPEC-591: Reflected XSS
CAPEC-592: Stored XSS
CAPEC-18: XSS Targeting Non-Script Elements
CAPEC-32:XSS Through HTTP Query Strings
CAPEC-86:XSS Through HTTP Headers
CAPEC-198: XSS Targeting Error Pages
CAPEC-199: XSS Using Alternate Syntax
CAPEC-243: XSS Targeting HTML Attributes
CAPEC-244: XSS Targeting URI Placeholders
CAPEC-245: XSS Using Doubled Characters
CAPEC-247: XSS Using Invalid Characters
CAPEC-62: Cross Site Request Forgery
CAPEC-107: Cross Site Tracing
CAPEC-593: Session Hijacking
CAPEC-31: Accessing / Intercepting / Modifying HTTP Cookies

универсальные

CAPEC-186: Malicious Software Update
CAPEC-89: Pharming
CAPEC-98: Phishing
CAPEC-443: Malicious Logic Inserted Into Product by Authorized Developer
CAPEC-445: Malicious Logic Insertion into Product Software via Configuration Management Manipulation
CAPEC-446: Malicious Logic Insertion into Product via Inclusion of Third-Party Component
CAPEC-523: Malicious Software Implanted
CAPEC-558: Replace Trusted Executable
CAPEC-569: Collect Data as Provided by Users
CAPEC-637: Collect Data from Clipboard
CAPEC-648: Collect Data from Screen Capture

БДУ ФСТЭК

phd x pt

frontend

СП.22.22 Подмена действий пользователя (Clickjacking)

СП.22.4 Межсайтовый скриптинг (XSS) без участия сервера

СП.22.16 Получение чувствительной информации со страниц веб-приложения или из ответов сервера

СП.4.3 Внедрение вредоносного программного обеспечения через посещение сайтов

frontend + backend

СП.22.3 Межсайтовый скриптинг (XSS) с запросами через сервер

СП.22.7 Нелегитимная отправка (подделка) запросов от имени пользователя (CSRF)

СП.22.9 Раскрытие чувствительной информации о пользователях

СП.5.8 Внедрение закладок в код веб-приложения

СП.22.25 Межсайтовая трассировка (XST-атака)

универсальные

СП.11.1 Применение скрытых каналов по времени

СП.4.10 Внедрение вредоносного программного обеспечения через скомпрометированные обновления программного обеспечения или операционной системы

СП.22.10 Раскрытие чувствительной информации о приложении и инфраструктуре

СП.22.30 Нарушение логики работы приложения

СП.1.1 Эксплуатация известных уязвимостей

СП.1.2 Эксплуатация уязвимостей "нулевого дня"

СП.12.7 Снятие звука с микрофона

Проблемы

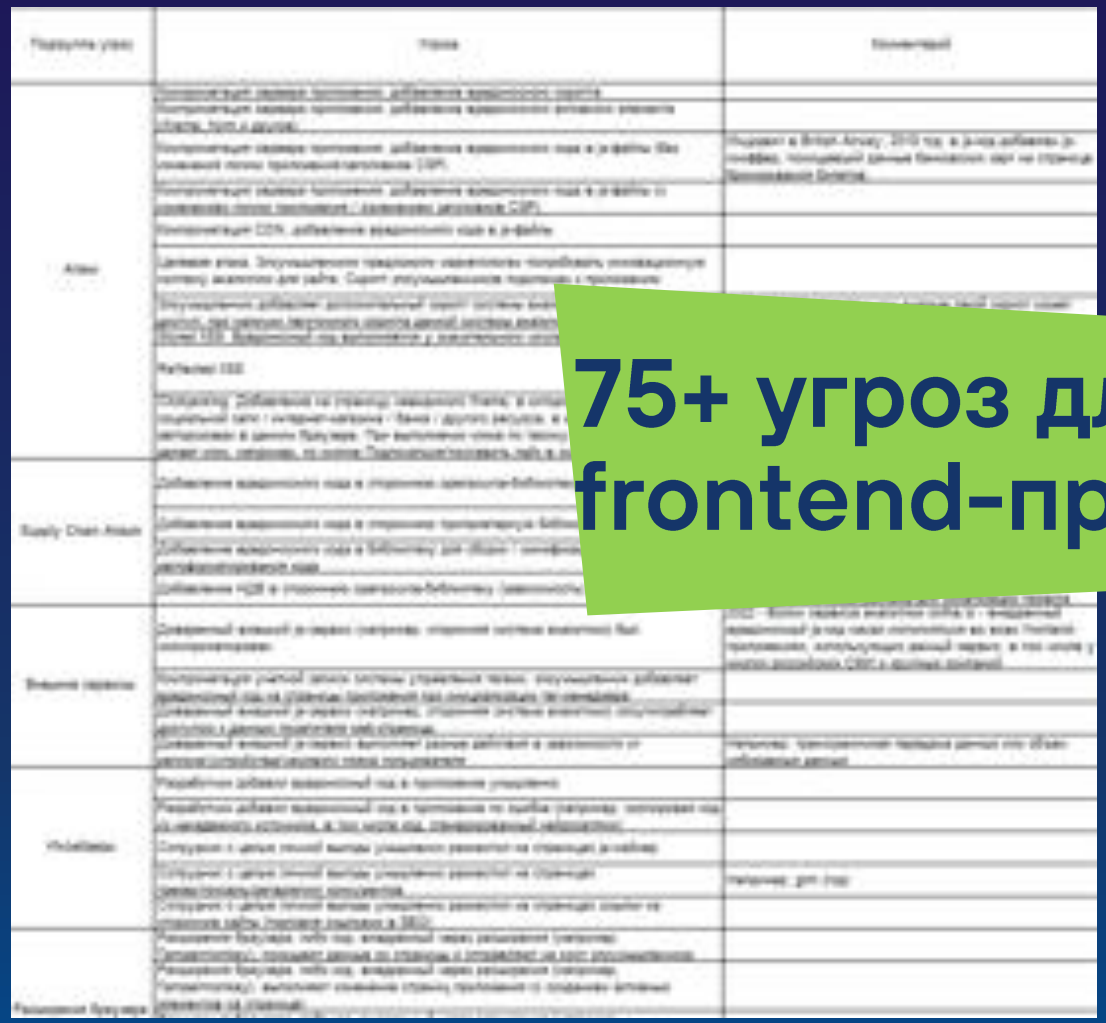
1. Непонимание последствий

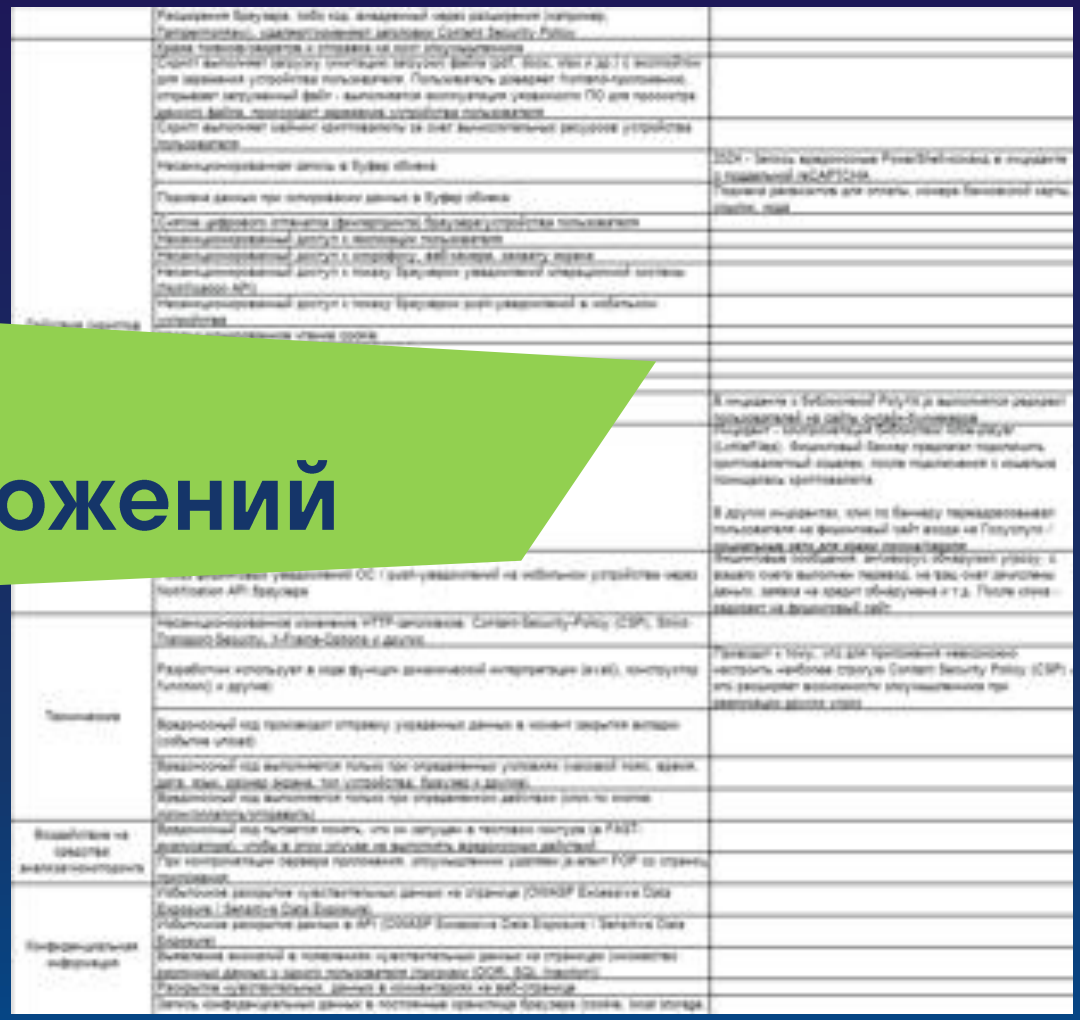
2. Непонимание векторов

“А чего там ломать то на фронтенде? И так всё на стороне клиента...

Ну может XSS, CSRF и то не критично...”

Построим свой фреймворк с рисками и митигациями





75+ угроз для frontend-приложений

По последствиям и ущербу

Технический вектор

- Внедрен вредоносный js-код
- Внедрен активный элемент (iframe, form...)

По последствиям и ущербу

Технический вектор	Последствия	Ущерб
Внедрен вредоносный js-код	Утечка персональных данных	Финансовый ущерб клиентам
Внедрен активный элемент (iframe, form...)	Frontend-фишинг	Снижение прибыли
	Js-майнинг	Участие в уголовном деле
	Действия от имени пользователя	Репутационный ущерб
	Заражение устройства пользователя	Штрафы по 152-ФЗ
	“Чёрное” SEO	Санкции регуляторов (ЦБ, GDPR и т.д.)

По последствиям и ущербу

Технический вектор	Способ реализации	Последствия	Ущерб
Внедрен вредоносный js-код	Внедрение js-сниффера, кража ПД со страниц	Утечка персональных данных	Финансовый ущерб клиентам
Внедрен активный элемент (iframe, form...)	Подмена данных при копировании в буфер обмена	Frontend-фишинг	Снижение прибыли
	Показ фишинговых нотификаций ОС	Js-майнинг	Участие в уголовном деле
	Кража cookie / токенов	Действия от имени пользователя	Репутационный ущерб
	Запуск вредоносных действий по триггеру (время, язык...)	Заражение устройства пользователя	Штрафы по 152-ФЗ
	Имитация загрузки .docx файла с эксплойтом MS Word	“Чёрное” SEO	Санкции регуляторов (ЦБ, GDPR и т.д.)
	Доступ к микрофону/камере/ геолокации пользователя		

25+ способов

По последствиям и ущербу

Технический вектор	Способ реализации	Последствия	Ущерб
Внедрен вредоносный js-код	Внедрение js-сниффера, кража ПД со страниц	Утечка персональных данных	Финансовый ущерб клиентам
Внедрен активный элемент (iframe, form...)	Подмена данных при копировании в буфер обмена	Frontend-фишинг	Снижение прибыли
Небезопасная конфигурация	Показ фишинговых нотификаций ОС	Js-майнинг	Участие в уголовном деле
Невыполнение требований регуляторов	Кража cookie / токенов	Действия от имени пользователя	Репутационный ущерб
	Запуск вредоносных действий по триггеру (время, язык...)	Заражение устройства пользователя	Штрафы по 152-ФЗ
	Имитация загрузки .docx файла с эксплойтом MS Word	“Чёрное” SEO	Санкции регуляторов (ЦБ, GDPR и т.д.)
	Доступ к микрофону/камере/ геолокации пользователя		

25+ способов

По векторам

Тип вектора	Вектор	Результат
Зависимости js-приложения	Добавление вредоносного кода в стороннюю opensource-библиотеку (зависимость)	Внедрен вредоносный js-код
Внешние js-сервисы	Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)	
Тег-менеджеры	Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)	
Добавление кода сотрудником	Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер	Внедрен активный элемент (iframe, form...)
Компрометация веб-сервера	Компрометация сервера приложения, добавление вредоносного скрипта	
Атаки	Stored XSS. Вредоносный код выполняется у значительного числа пользователей	

30+ способов

Соединяем...

Зависимости js-приложения

Внешние js-сервисы

Тег-менеджеры

Добавление кода сотрудником

Компрометация веб-сервера

Атаки

Вектор

Добавление вредоносного кода в стороннюю opensource-библиотеку (зависимость)

Добавление вредоносного кода в стороннюю opensource-библиотеку (зависимость)

Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)

Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)

Использование функции системы аналитики, собирающей полный код web-страниц / нажатия клавиш пользователем

Использование функции системы аналитики, собирающей полный код web-страниц / нажатия клавиш пользователем

Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)

Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)

Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер

Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер

Компрометация сервера приложения, добавление вредоносного скрипта

Компрометация сервера приложения, добавление вредоносного скрипта

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Добавление вредоносного кода в стороннюю opensource-библиотеку (зависимость)

Добавление вредоносного кода в стороннюю opensource-библиотеку (зависимость)

Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)

Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)

Использование функции системы аналитики, собирающей полный код web-страниц / нажатия клавиш пользователем

Использование функции системы аналитики, собирающей полный код web-страниц / нажатия клавиш пользователем

Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)

Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)

Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер

Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер

Компрометация сервера приложения, добавление вредоносного скрипта

Компрометация сервера приложения, добавление вредоносного скрипта

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Stored XSS. Вредоносный код выполняется у значительного числа пользователей

Внедрен вредоносный js-код

Внедрен активный элемент (iframe, form...)

Способ реализации

Внедрение js-сниффера, кража ПД со страниц

Внедрение js-сниффера, кража ПД со страниц

Внедрение js-сниффера, кража ПД со страниц

Внедрение js-сниффера, кража ПД со страниц

Подмена данных при копировании в буфер обмена

Подмена данных при копировании в буфер обмена

Подмена данных при копировании в буфер обмена

Подмена данных при копировании в буфер обмена

Показ фишинговых нотификаций ОС

Показ фишинговых нотификаций ОС

Показ фишинговых нотификаций ОС

Показ фишинговых нотификаций ОС

Кража cookie /токенов

Кража cookie /токенов

Кража cookie /токенов

Кража cookie /токенов

Запуск вредоносных действий по триггеру (время, язык..)

Запуск вредоносных действий по триггеру (время, язык..)

Запуск вредоносных действий по триггеру (время, язык..)

Запуск вредоносных действий по триггеру (время, язык..)

Имитация загрузки docx файла с эксплойтом MS Word

Имитация загрузки docx файла с эксплойтом MS Word

Имитация загрузки docx файла с эксплойтом MS Word

Имитация загрузки docx файла с эксплойтом MS Word

Доступ к микрофону/камере/ геолокации пользователя

Доступ к микрофону/камере/ геолокации пользователя

Доступ к микрофону/камере/ геолокации пользователя

Доступ к микрофону/камере/ геолокации пользователя

Утечка персональных данных

Frontend-фишинг

Js-майнинг

Действия от имени пользователя

Заражение устройства пользователя

“Чёрное” SEO

Финансовый ущерб клиентам

Снижение прибыли

Участие в уголовном деле

Репутационный ущерб

Штрафы по 152-ФЗ

Санкции регуляторов (ЦБ, GDPR и т.д.)

Примеры с известными инцидентами

Компрометация веб-сервера

Компрометация сервера приложения, добавление вредоносного скрипта

Внедрен вредоносный js-код

Внедрение js-сниффера, кража ПД со страниц

Утечка персональных данных

Финансовый ущерб клиентам

Снижение прибыли

Участие в уголовном деле

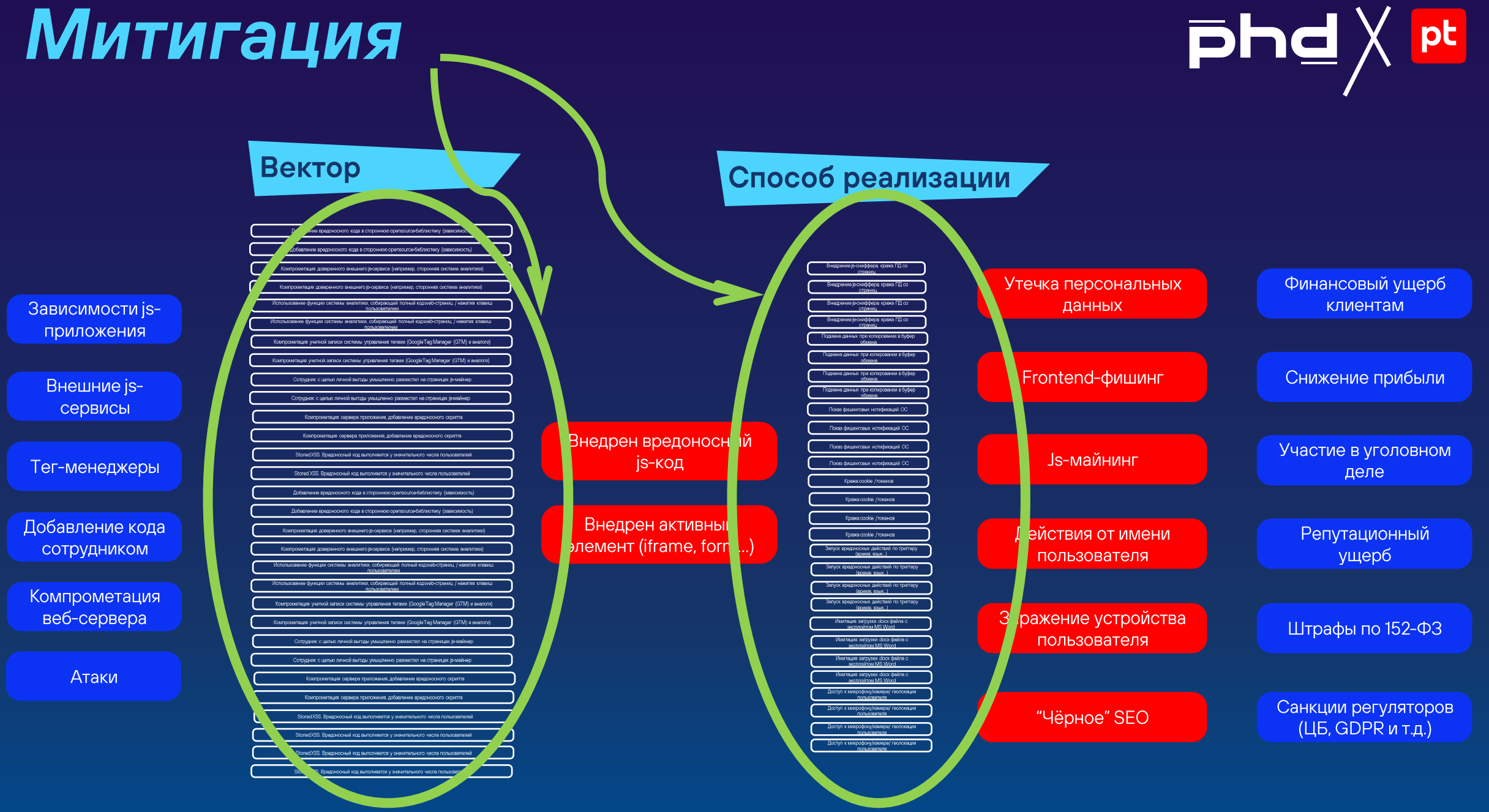
Репутационный ущерб

Санкции регуляторов (ЦБ, GDPR и т.д.)

2018

Js-сниффер на сайте British Airways

Митигация


Вектор
Способ реализации
Зависимости js-приложения
Внешние js-сервисы
Тег-менеджеры
Добавление кода сотрудником
Компрометация веб-сервера
Атаки
Добавление вредоносного кода в сторонюю opensource-библиотеку (зависимость)
Компрометация доверенного внешнего js-сервиса (например, сторонняя система аналитики)
Использование функции системы аналитики, собирающей полный код web-страниц / нажатия клавиш пользователем
Компрометация учетной записи системы управления тегами (Google Tag Manager (GTM) и аналоги)
Сотрудник с целью личной выгоды умышленно разместил на страницах js-майнер
Компрометация сервера приложения, добавление вредоносного скрипта
Stored XSS. Вредоносный код выполняется у значительного числа пользователей
Внедрен вредоносный js-код
Внедрен активный элемент (iframe, form ...)
Внедрение js-сниффера, кража ПД со страниц
Подмена данных при копировании в буфер обмена
Показ фишинговых нотификаций ОС
Кража cookie /токенов
Запуск вредоносных действий по триггеру (время, язык..)
Имитация загрузки docx файла с эксплойтом MS Word
Доступ к микрофону/камере/ геолокации пользователя
Утечка персональных данных
Frontend-фишинг
Js-майнинг
Действия от имени пользователя
Заражение устройства пользователя
“Чёрное” SEO
Финансовый ущерб клиентам
Снижение прибыли
Участие в уголовном деле
Репутационный ущерб
Штрафы по 152-ФЗ
Санкции регуляторов (ЦБ, GDPR и т.д.)

Митигация (90%) frontend-угроз через обнаружение / observability

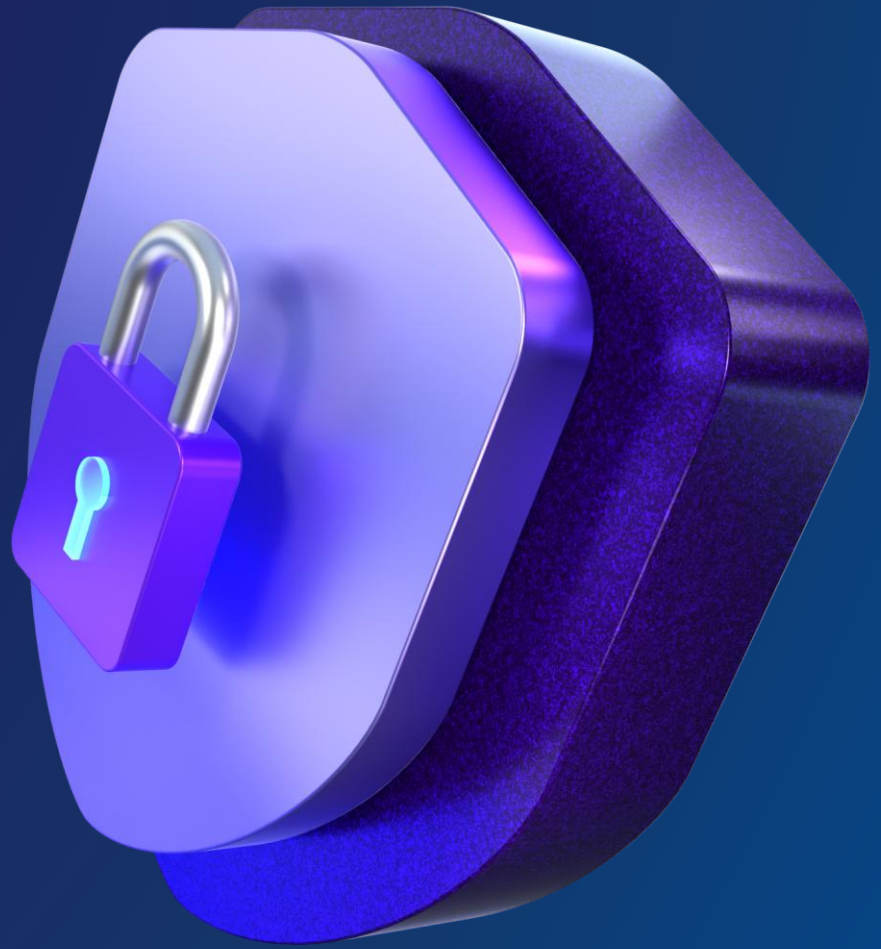
- До релиза
- После релиза (минуты/часы)
- Реагирование

Фреймворк МУ

1. Митигирующие меры применяются для Векторов и Способов реализации
2. Для удобства работы с МУ объединили Векторы и Способы реализации в одной таблице
3. Добавили оценку рисков
4. Добавили митигирующие меры с оценкой эффективности
5. План обработки рисков
6. Представлю в виде онлайн-сервиса в июне 2025 в telegram-канале @FrontSecOps

04

Анализ поведения frontend-приложения в DevSecOps – путь к Secure by Design



Применимость классических анализаторов безопасности

SAST

- JavaScript-код может быть выполнен «на лету» из зашифрованного текста, что не выявляется SAST.
- Также из анализа обычно исключаются сторонние библиотеки, что снижает покрытие кода анализом до 95%.

DAST

- Обычно DAST анализирует frontend-приложения для определения API-эндпойнтов бэкенда и нескольких видов XSS.
- Поведение приложения не анализируется, т. к. оно не отличимо от нормальных бизнес-функций.

SCA / OSA

- Анализ зависимостей обнаруживает известные уязвимости, но не НДВ.
- JS-код может динамически подгружать модули прямо в браузере.
- Анализ внешних js-сервисов не производится.

phd x pt

“В сторонний JavaScript-код в любое время могут быть добавлены новые функции. Риск возникает т.к. сторонний код редко анализируется на безопасность.
Любое тестирование, проведенное до ввода в эксплуатацию, теряет достоверность (в т.ч. IAST, SAST, DAST)”

OWASP Third Party JavaScript Management Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/Third_Party_Javascript_Management_Cheat_Sheet.html

“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

Frontend Application Security Testing (FAST)

SAST

- JavaScript-код может быть выполнен «на лету» из зашифрованного текста, что не выявляется SAST.
- Также из анализа обычно исключаются сторонние библиотеки, что снижает покрытие кода анализом до 95%.

DAST

- Обычно DAST анализирует frontend-приложения для определения API-эндпойнтов бэкенда и нескольких видов XSS.
- Поведение приложения не анализируется, т. к. оно не отличимо от нормальных бизнес-функций.

SCA / OSA

- Анализ зависимостей обнаруживает известные уязвимости, но не НДВ.
- JS-код может динамически подгружать модули прямо в браузере.
- Анализ внешних js-сервисов не производится.

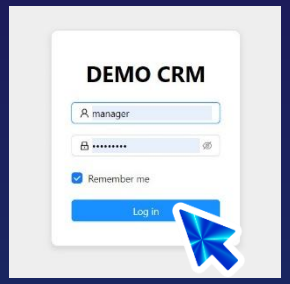
FAST

- Выполняет весь js-код в реальном браузере, включая динамически загруженный.
- Анализирует поведение js-кода во время выполнения реальных Use Case.
- Сравнивает полученный профиль поведения с разрешенным (whitelist).

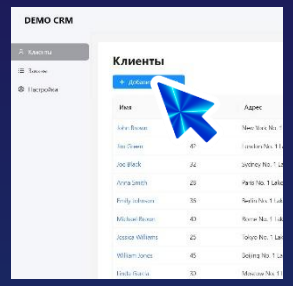
Frontend Application Security Testing (FAST)

Открытие страницы

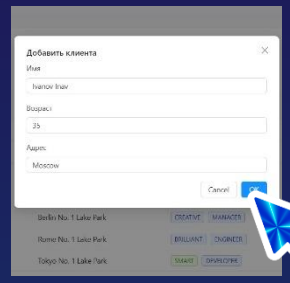
Ввод логина, пароля, клик по кнопке



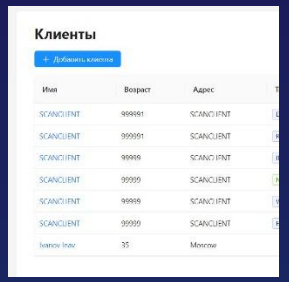
Успешная аутентификация, редирект



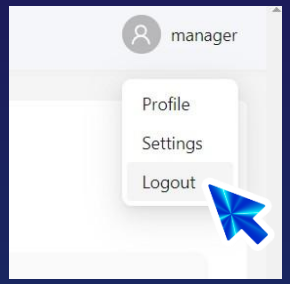
Открытие формы создания клиента, заполнение данных



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.

Sensitive Data

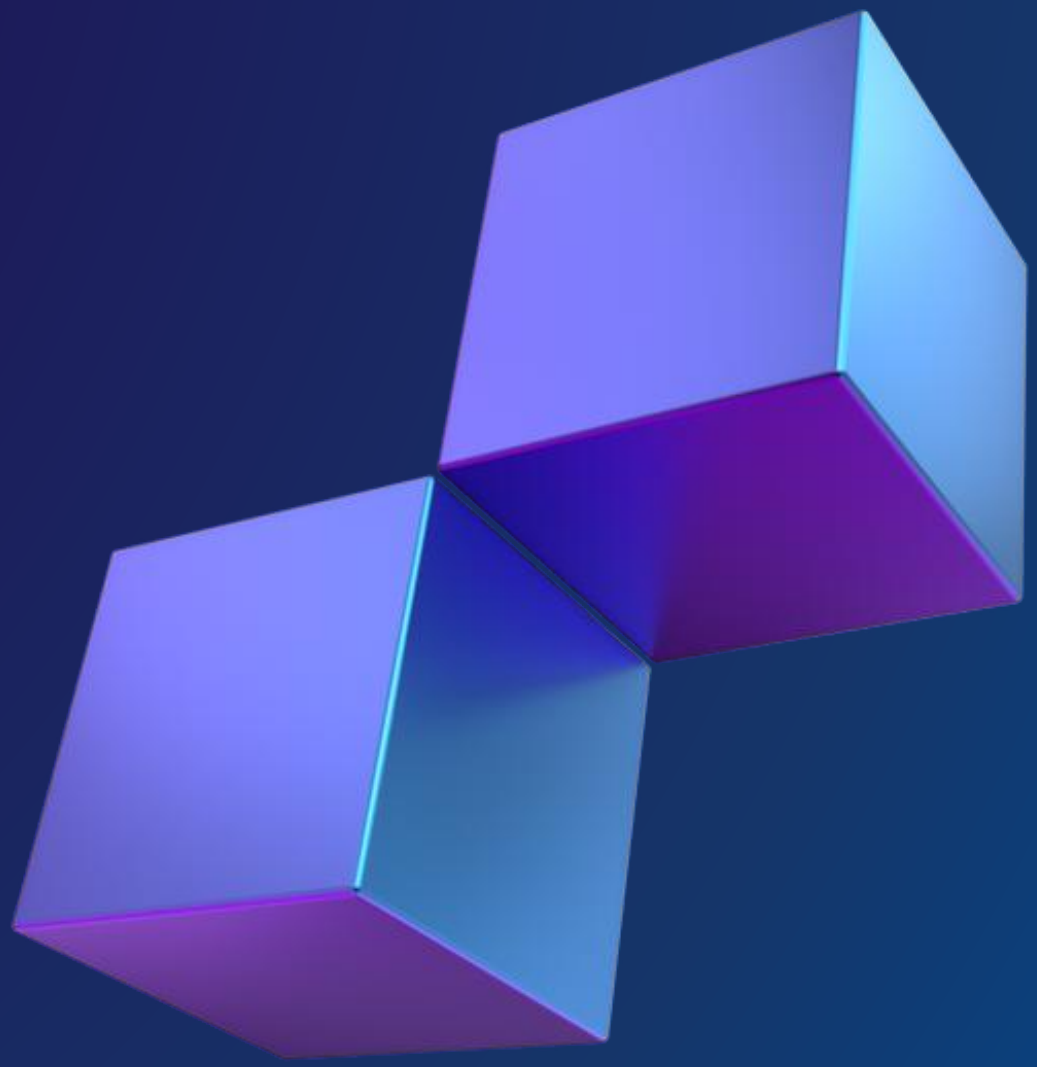
В сетевых запросах, на странице, в постоянных хранилищах браузера

Software Bill of Behavior (SBOB)

Контентный слой браузера

05

Что делать?



Что делать?

- Понять, что frontend-приложения – важная цель для злоумышленников, дающая гарантированную монетизацию
- Ответить на вопрос: «Я знаю/уверен, что делает frontend-приложение прямо сейчас? Куда отправляет данные?»
- Создать модель угроз для frontend-приложений (опубликую онлайн сервис в telegram-канале @FrontSecOps в июне 2025)
- Выполнять мониторинг/контроль поведения frontend-приложений в DevSecOps
- Использовать средства автоматизации (например, FAST-анализатор) для глубокого анализа, контроля изменений и оповещения о несанкционированных изменениях


MLSecOps
AppSec
backend
frontend

Ссылки

Доклад “Frontend application security testing (FAST). Задачи подхода и место в DevSecOps”

Доклад “Анализ поведения как способ контроля безопасности frontend-приложений в DevSecOps”

Исследование безопасности российских frontend-приложений

Telegram-канал @FrontSecOps, публикация в июне 2025

Сервис для моделирования frontend-угроз

Telegram-канал @FrontSecOps, публикация в июне 2025

Telegram-канал FrontSecOps

- Разбор инцидентов
- DevSecOps для frontend-приложений
- Онлайн-сервис для моделирования угроз (июнь 2025)
- Отчет об исследовании безопасности российских frontend-приложений (июнь 2025)

@FRONTSECOPS

Оценить доклад

Спасибо!

Михаил Парфенов
Application Security Architect
DPA Analytics
TG: @mkparfenov

@FRONTSECOPS