

Yandex

Yandex Security

Content Security Policy.
How to implement on an
industrial scale

\$ whois

- › Product security team lead in [Yandex](#)
- › [OWASP Russia](#) chapter leader
- › Yet another security blogger [oxdef.info](#)

| Does anybody use CSP?

| < 1% of all sites :- (But ...



Empty slide about XSS

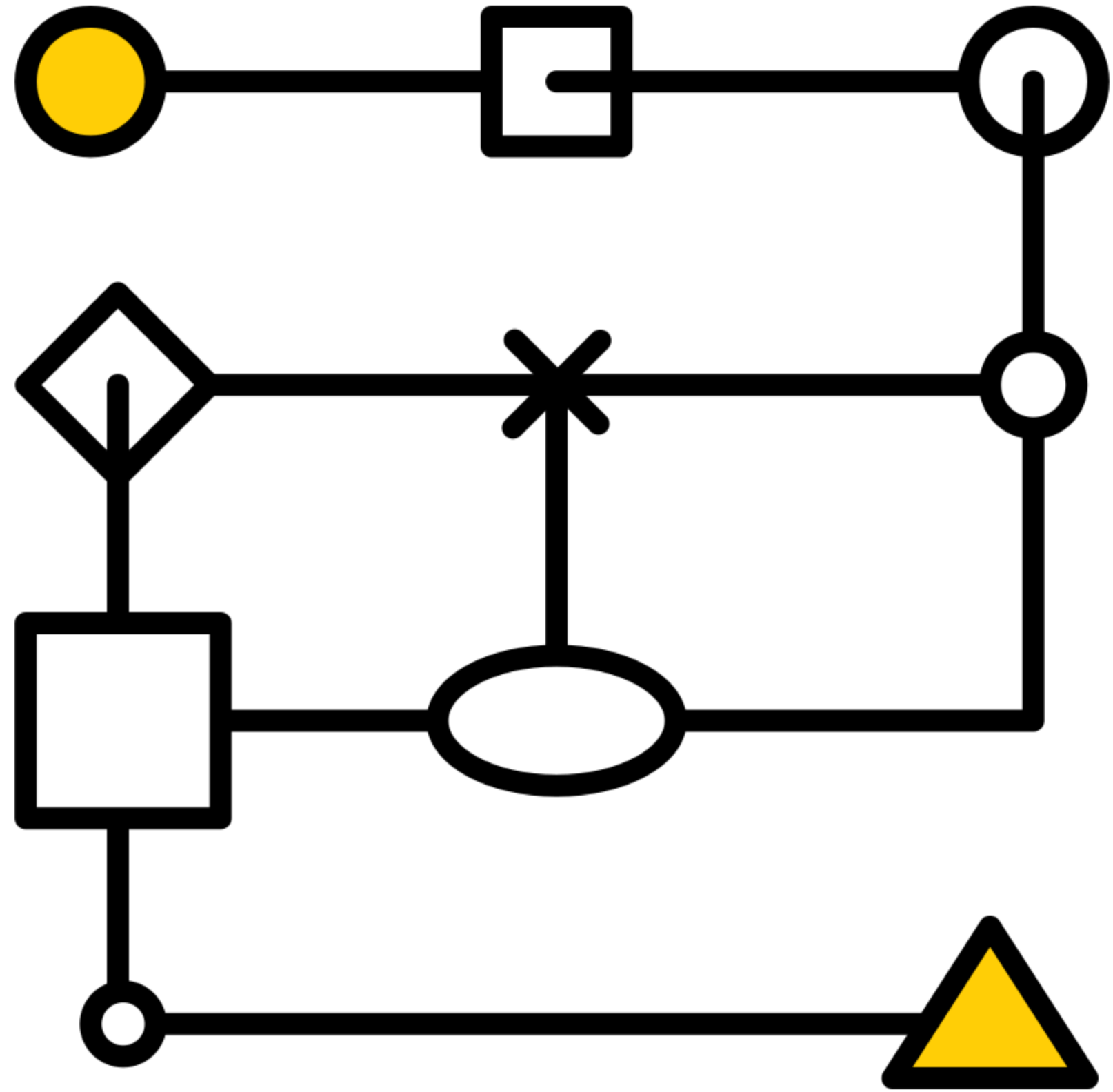
Because no more slides about XSS

Content security policy

Content security policy



- › Browser side mechanism to mitigate XSS attacks
- › Open live standard www.w3.org/TR/CSP
- › Source whitelists and signatures for client side code and resources of web application
- › `Content-Security-Policy` and `Content-Security-Policy-Report-Only` HTTP headers
- › HTML `meta` element



In a nutshell

Policy

```
default-src 'none'; script-src 'nonce-Nc3n83cnSAd' static.example.com
```

HTML

```
<!doctype html><html><head>  
<meta charset="utf-8">  
<script src="//static.example.com/jquery.js"></script>  
<script nonce="Nc3n83cnSAd"></script>  
<script src="//evil.net/evil.js"></script>
```

unsafe-inline and unsafe-eval

unsafe-inline

- › Inline scripts and styles
- › `onclick="..."`
- › `javascript:`

unsafe-eval

- › `eval()`
- › `new Function`
- › `setTimeout`, `setInterval` with string as a first argument

Other directives

- › `style-src` - CSS styles
- › `media-src` – audio and video
- › `object-src` - plugin objects (e.g. Flash)
- › `frame-src` – iframe sources
- › `font-src` – font files
- › `connect-src` - XMLHttpRequest, WebSocket

When CSP protects against XSS

In order to protect against XSS, web application authors SHOULD include:

- › both the `script-src` and `object-src` directives, or
- › include a `default-src` directive, which covers both scripts and plugins.

In either case, authors SHOULD NOT include either `'unsafe-inline'` or `data:` as valid sources in their policies. Both enable XSS attacks by allowing code to be included directly in the document itself; they are best avoided completely.

www.w3.org/TR/CSP2/

When CSP stronger protects against XSS

- › Default value should be `'none'`
- › Strictly specify `base-uri`
- › Avoid wildcard sources in source lists of directives
- › Minimize source lists
- › Exam `script-src` sources against JSONP endpoints
- › Use strict `style-src` value

Reporting

Policy

```
Content-Security-Policy-Report-Only: ...; report-uri /csp-log
```

Log contents

```
"csp-report": {  
  "violated-directive": "img-src data: ...*.example.com",  
  "blocked-uri": "https://static.doubleclick.net",  
  "document-uri": "https://example.com/foo",  
  "original-policy": "default-src ...; report-uri /csp-log"  
}
```

What is current version?

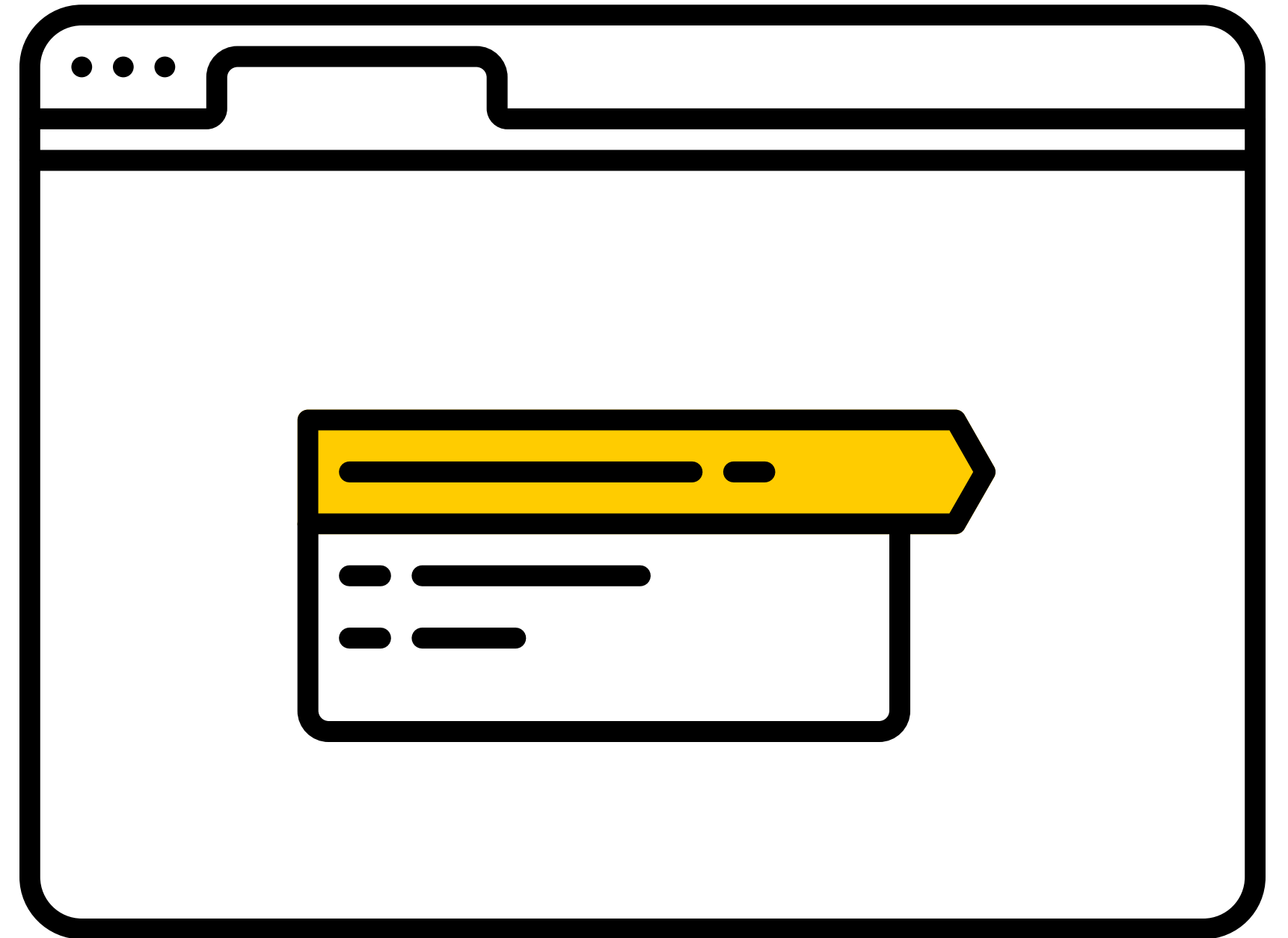
- › CSPv2: [W3C Candidate Recommendation](#)
- › CSPv3: [W3C Working Draft](#)

CSPv2 vs. CSPv3

- › The specification has been rewritten from the ground up in terms of the FETCH specification
- › The `strict-dynamic` source expression will now allow script which executes on a page to load more script via non-“parser-inserted” `script` elements.
- › `report-uri` → `report-to`
- › More directives: `manifest-src`, `disown-opener`

Browser support

- › Google Chrome 25+
- › Mozilla Firefox 23+
- › Yandex Browser 1.7+
- › Safari 10.2+
- › MS Edge 14+



Bypass?!

Bypass ways

- › Manipulating HTTP response headers
- › Implementation bugs
- › JSONP
- › Script gadgets
- › XSS without JS

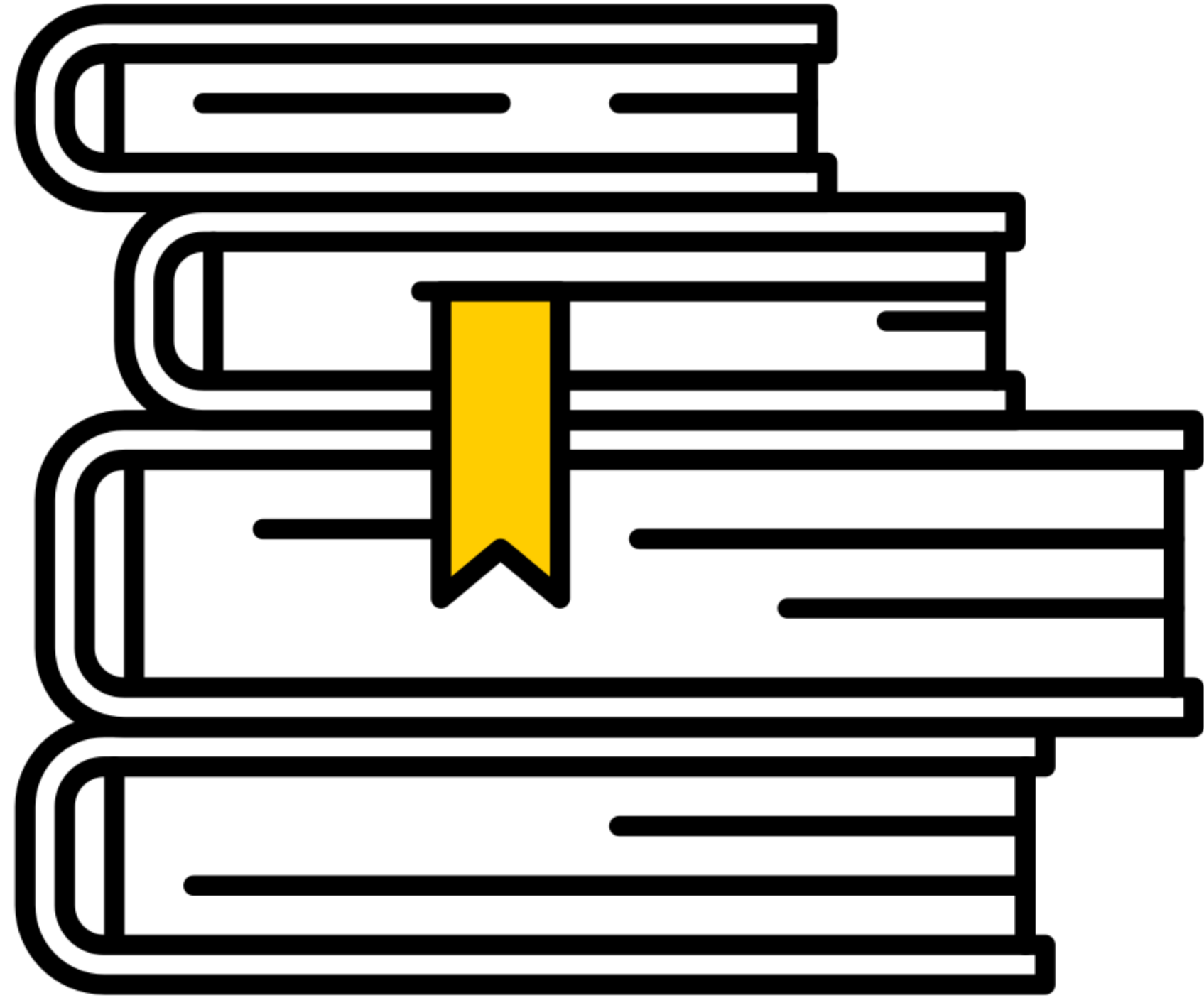
~~Pain~~ Experience

Yandex scale

- › At least 50+ public complex services yandex.ru/all
- › Thousands of developers
- › Lots of client side code and hosts to communicate with
- › Error in policy can cause problems for millions of users
- › Many 3rd party JavaScript libraries
- › Legacy code

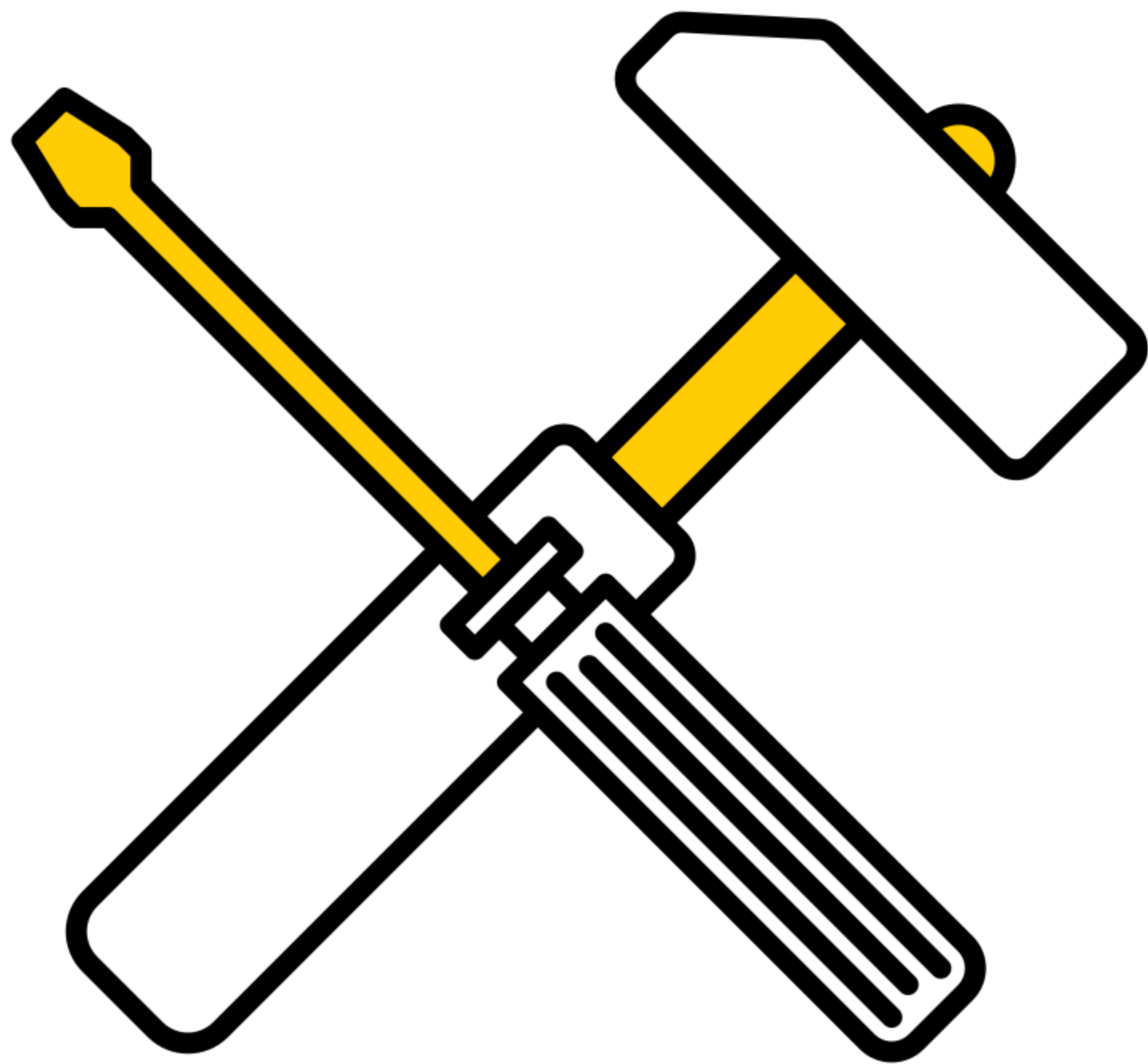


Main goal is to
implement CSP into all
services.



Education

- › Detailed guide at internal security portal
 - › What is CSP
 - › How to prepare service to implement CSP
 - › Policy best practices
 - › Tools
 - › FAQ
 - › Contact form
- › Internal activities and talks
- › Hype ^_-



CSP Tester as extension

- › Extension for Chromium based browsers and Mozilla Firefox
- › Simple and Advanced modes
- › Help links for directives
- › github.com/yandex/csp-tester

CSP Tester in action

CSP Tester

URL Pattern

default-src

script-src

style-src

img-src

frame-ancestors

connect-src

report-uri

base-uri

Active Report Only

[Advanced Mode](#)

CSP Tester as service

- › Self-checking service
- › Education
- › Part of global automated security control Molly
- › Approximately 50 tests
- › API

CSP Tester service in action

☰ CSP tester

```
default-src 'none';
connect-src 'self' yastatic.net mc.yandex.ru mc.yandex.ua mc.yandex.by
mc.yandex.kz mc.yandex.com.tr mc.yandex.com mc.webvisor.org
mc.webvisor.com;
font-src yastatic.net;
frame-src mc.yandex.ru mc.yandex.ua mc.yandex.by mc.yandex.kz
mc.yandex.com.tr mc.yandex.com mc.webvisor.org mc.webvisor.com;
img-src 'self' data: www.tns-counter.ru *.captcha.yandex.net
yastatic.net mc.yandex.ru mc.yandex.ua mc.yandex.by mc.yandex.kz
```

Проверить

Обнаруженные ошибки CSP-политики

В директиве 'style-src' включена опция 'unsafe-inline'. Необходимо подписывать инлайн-код (использовать нонсы или хэши).

В директиве 'script-src' указаны опасные опции эквивалентные 'unsafe-eval' (blob: or filesystem:). Обнаружено: 'unsafe-eval'. Рассмотрите возможность отказа от соответствующих языковых конструкций в коде.

В списке источников критичной директивы 'script-src' содержится либо значение 'self', либо один из 17 доменов: 'self'. Это увеличивает риски от влияния XSS на другом сервисе на этом домене. Старайтесь избегать такой практики

Other stuff

- › Collector for CSP logs from all services
- › Support and modules for core front-end components, e.g. middleware for Express/NodeJS
- › CSP log parser [CSP Reporter](#)



Manage and control

- › High level tickets to implement CSP for all services
- › Mandatory requirement for all new services
- › [Final security review and robots](#)
- › CSP checks are integrated into automation security scanning process by Molly

Public JavaScript API changes

Let's make our public JavaScript API more friendly to CSP


- › [Yandex Metrica counter](#)
- › [Yandex Maps API](#)

Problems and solutions

- › 3rd party JavaScript components
- › 3rd party services without built-in CSP support
- › Wildcard sources
- › Big size of policy
- › JSONP
- › Legacy code

Summary

- › Teach your front-end developers
- › Implement CSP into existing services
- › Add CSP as security requirement for new ones
- › Don't forget about mobile versions
- › Research your core front-end components to support CSP
- › Keep your CSP policy clean, minimal and strict



It could be difficult but
you should try it...

| to make your users safer!

Q&A

Contacts

Taras Ivashchenko

Product security team



oxdef@yandex-team.ru



[oxdef](https://github.com/oxdef)



[@oxdef](https://twitter.com/oxdef)