December 8th 2023

Exploring Content Security Policies

What even is it?

Added layer of security for websites

Allowlist of website resources

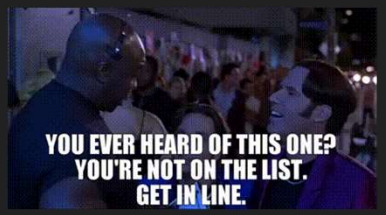
A W3C standard for browser implementations

How it works

How it works

How it works

In short… A CSP is a bouncer



Pros & Cons

- Benefits
  - Blocks malicious resources
  - Blocks browser extension modifications
  - Identifies external resources being used
  - Improves our security score
- Cons
  - Allowlist approach is a big ask
  - Potentially slow to roll out
  - Somewhat complicated for developers to understand
  - Can break your website!

The list… mostly

- Directives: Types of resources the website might load
  - script-src: Scripts (Javascript)
  - style-src: Styles (CSS)
  - Etc…
- Directive Values: What is allowed for the given directive
  - URL: https://fonts.google.com or https://fonts.google.com/family/font/style.css
  - 'self': Tells the browser the website is allowed to load resources from the website
  - data:: Tells the browser embedded data is allowed (Typically used for img-src directive)
  - blob:: Tells the browser blob data is allowed (Typically used for img-src directive)
  - 'none': Tells the browser there are no allowed resources for the directive
  - 'unsafe-inline': Tells the browser it's okay to render resources that are embedded in the html
  - 'unsafe-eval': Tells the browser it's okay to perform script evals
  - 'nonce-' or 'sha256-': Tells the browser it's okay to trust files or inline items if they match hashes
- report-uri & report-to: Where should the browser send violation reports?

The list… mostly

- Directives: Types of resources the website might load
  - script-src: Scripts (Javascript)
  - style-src: Styles (CSS)
  - Etc…
- Directive Values: What is allowed for the given directive
  - URL: https://fonts.google.com or https://fonts.google.com/family/font/style.css
  - 'self': Tells the browser the website is allowed to load resources from the website
  - data:: Tells the browser embedded data is allowed (Typically used for img-src directive)
  - blob:: Tells the browser blob data is allowed (Typically used for img-src directive)
  - 'none': Tells the browser there are no allowed resources for the directive
  - 'unsafe-inline': Tells the browser it's okay to render resources that are embedded in the html
  - 'unsafe-eval': Tells the browser it's okay to perform script evals
  - 'nonce-' or 'sha256-': Tells the browser it's okay to trust files or inline items if they match hashes
- report-uri & report-to: Where should the browser send violation reports?

This was intentional

Maybe use this sometimes…

Avoid this like the plague…

Let's make a CSP

- Options
  - Content-Security-Policy Header
  - Content-Security-Policy-Report-Only Header

Let's make a CSP

- Options
  - Content-Security-Policy Header
  - Content-Security-Policy-Report-Only Header

Let's make a CSP

- Options
  - Content-Security-Policy Header
  - Content-Security-Policy-Report-Only Header

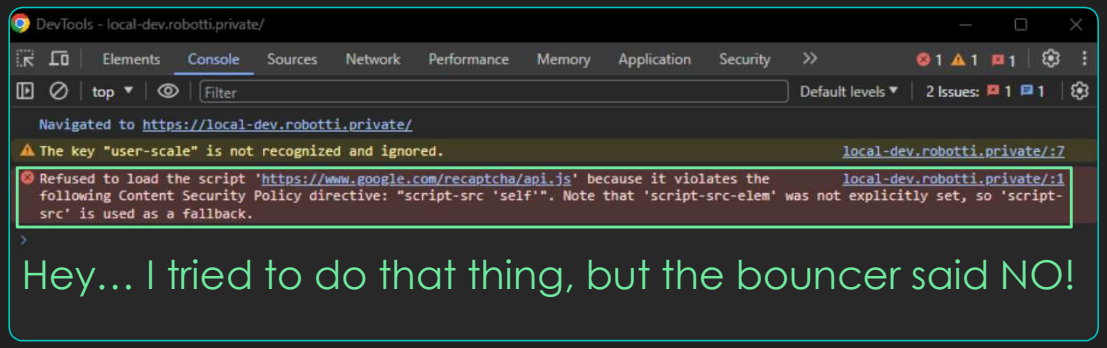
Let's make a CSP

- Options
  - Content-Security-Policy Header
  - Content-Security-Policy-Report-Only Header

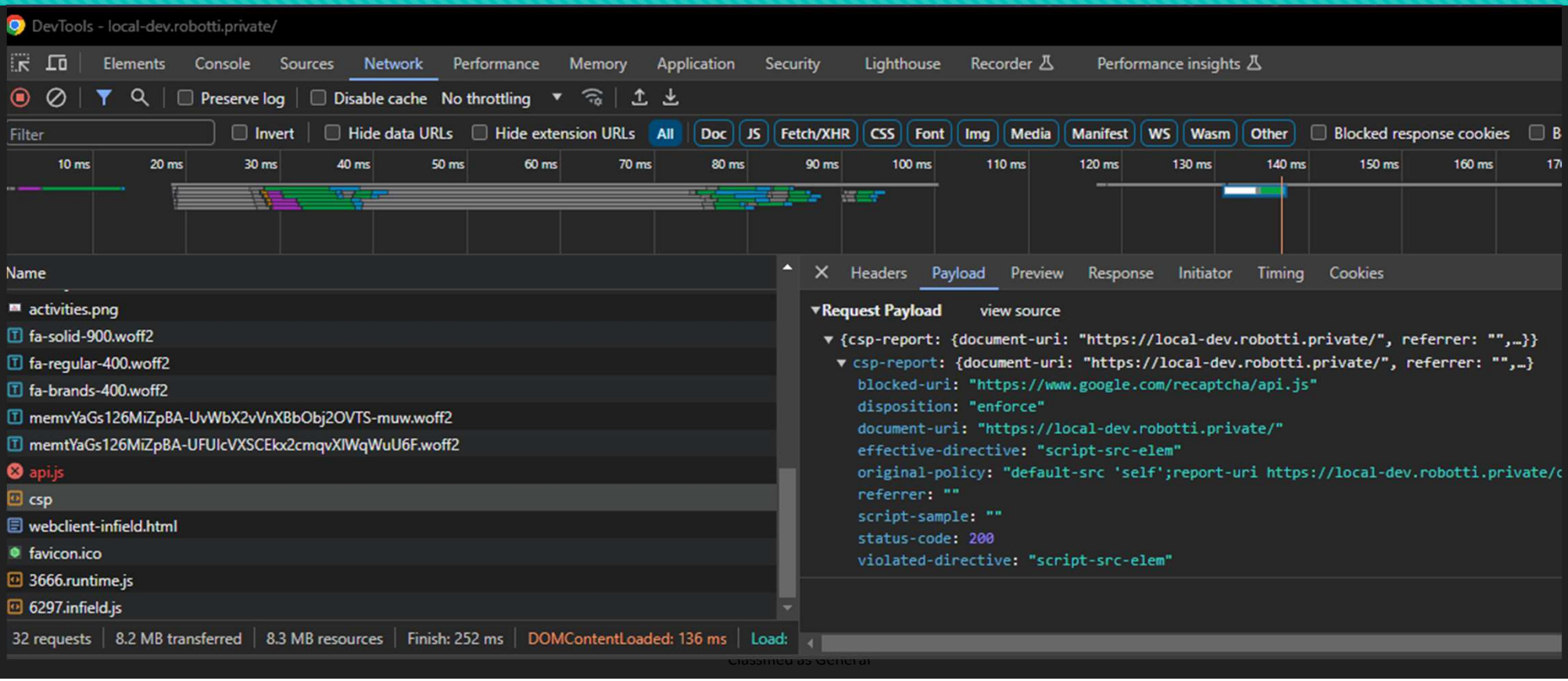
```
"helmet": {
  "csp": {
    "directives": {
      "default-src": [
        "'self'"
      ],
      "report-uri": [
        "https://local-dev.robotti.private/csp"
      ]
    },
    "reportOnly": false
  }
}
```

Let's make a CSP

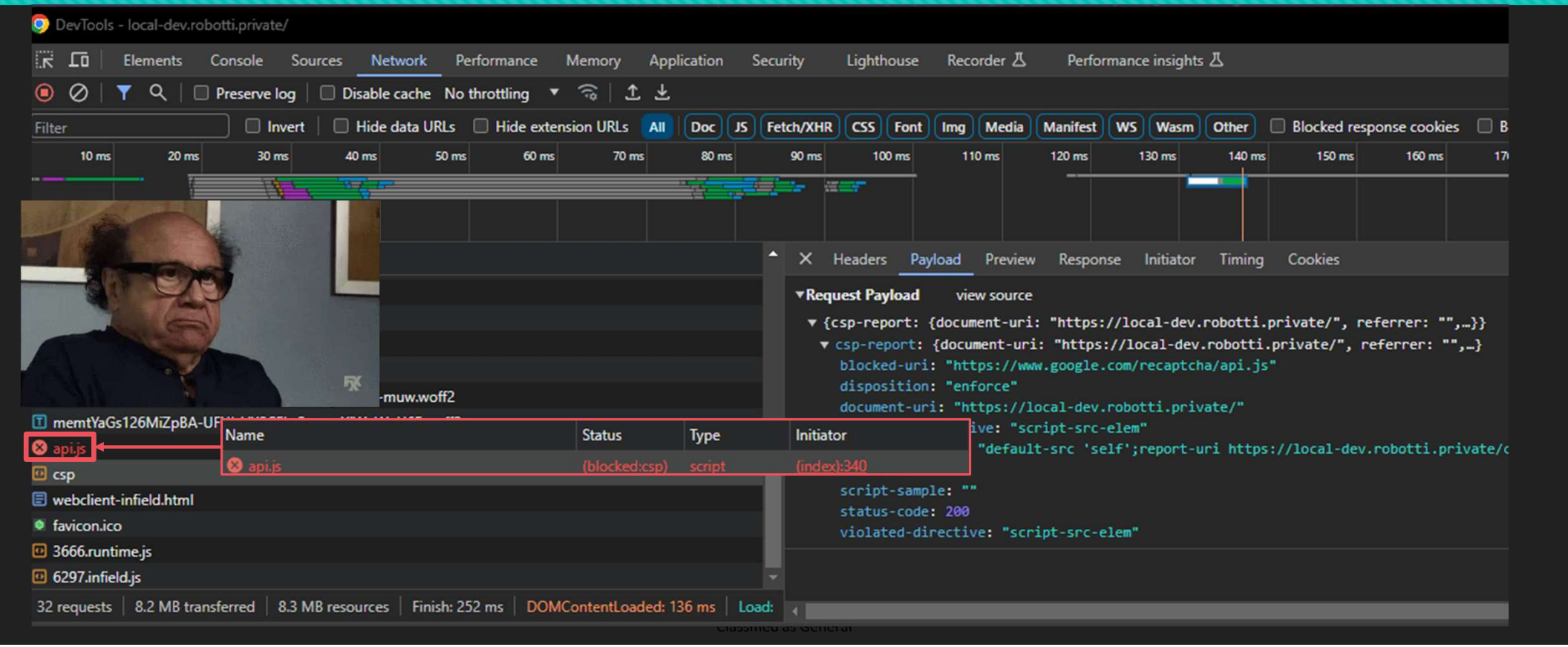
- Options
  - Content-Security-Policy Header
  - Content-Security-Policy-Report-Only Header



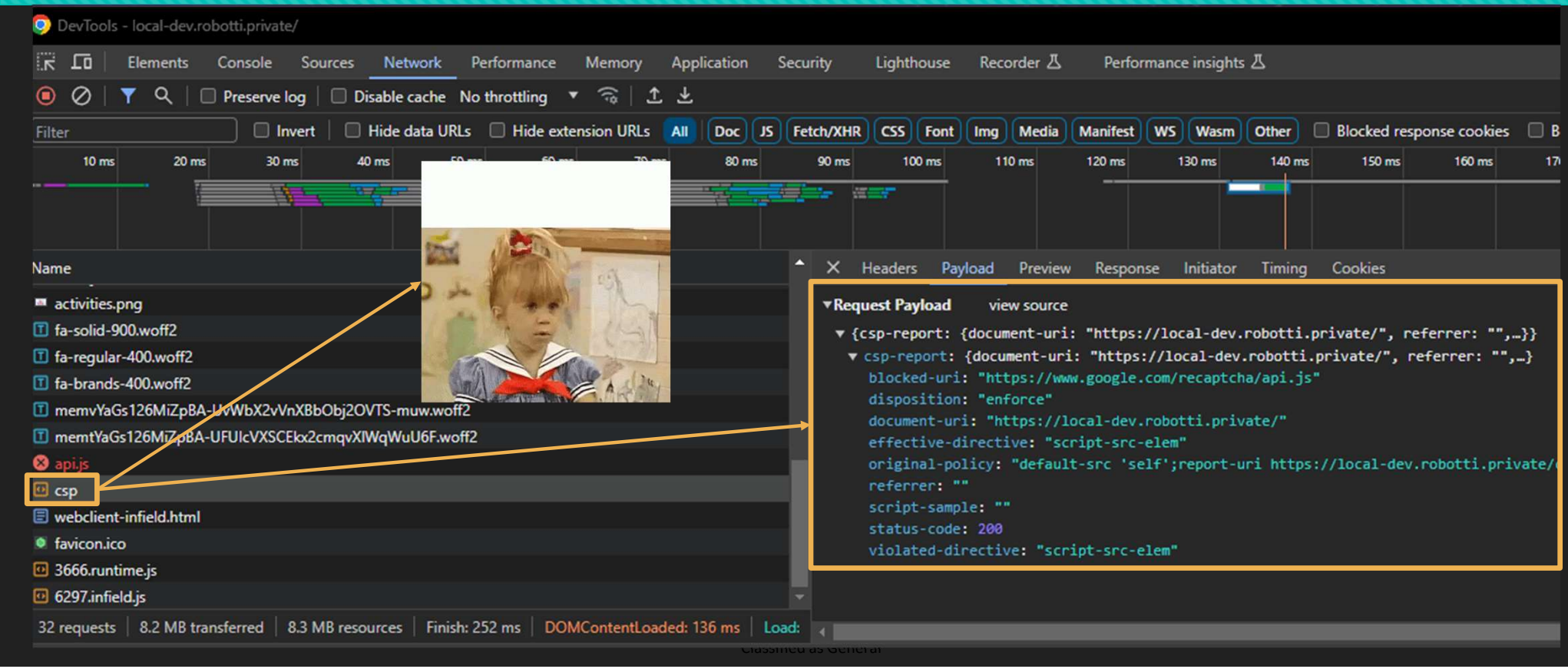
Reporting


DevTools - local-dev.robotti.private/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Performance insights
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All
Doc
JS
Fetch/XHR
CSS
Font
Img
Media
Manifest
WS
Wasm
Other
Blocked response cookies
10 ms
20 ms
30 ms
40 ms
50 ms
60 ms
70 ms
80 ms
90 ms
100 ms
110 ms
120 ms
130 ms
140 ms
150 ms
160 ms
Name
activities.png
fa-solid-900.woff2
fa-regular-400.woff2
fa-brands-400.woff2
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVTS-muw.woff2
memtYaGs126MiZpBA-UFUIcVXSCEkx2cmqvXlWqWuU6F.woff2
api.js
csp
webclient-infield.html
favicon.ico
3666.runtime.js
6297.infield.js
Headers
Payload
Preview
Response
Initiator
Timing
Cookies
Request Payload
view source
{csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}}
csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}
blocked-uri: "https://www.google.com/recaptcha/api.js"
disposition: "enforce"
document-uri: "https://local-dev.robotti.private/"
effective-directive: "script-src-elem"
original-policy: "default-src 'self';report-uri https://local-dev.robotti.private/
referrer: ""
script-sample: ""
status-code: 200
violated-directive: "script-src-elem"
32 requests
8.2 MB transferred
8.3 MB resources
Finish: 252 ms
DOMContentLoaded: 136 ms
Load:

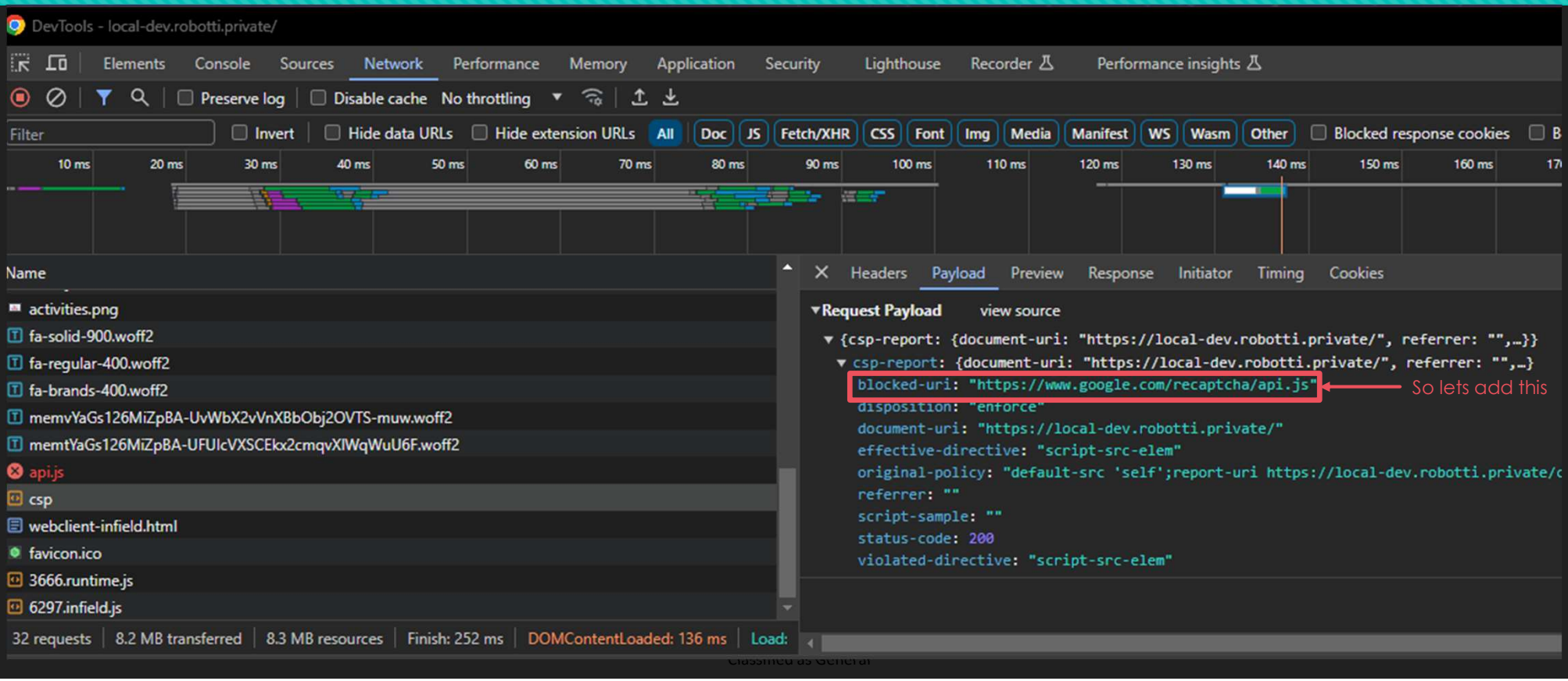
Reporting


DevTools - local-dev.robotti.private/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Performance insights
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All
Doc
JS
Fetch/XHR
CSS
Font
Img
Media
Manifest
WS
Wasm
Other
Blocked response cookies
10 ms
20 ms
30 ms
40 ms
50 ms
60 ms
70 ms
80 ms
90 ms
100 ms
110 ms
120 ms
130 ms
140 ms
150 ms
160 ms
Headers
Payload
Preview
Response
Initiator
Timing
Cookies
Request Payload
view source
{csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}}
csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}
blocked-uri: "https://www.google.com/recaptcha/api.js"
disposition: "enforce"
document-uri: "https://local-dev.robotti.private/"
violated-directive: "script-src-elem"
"default-src 'self';report-uri https://local-dev.robotti.private/
script-sample: ""
status-code: 200
violated-directive: "script-src-elem"
-muw.woff2
memtYaGs126MiZpBA-UF
api.js
csp
webclient-infield.html
favicon.ico
3666.runtime.js
6297.infield.js
Name
Status
Type
Initiator
api.js
(blocked:csp)
script
(index):340
32 requests
8.2 MB transferred
8.3 MB resources
Finish: 252 ms
DOMContentLoaded: 136 ms
Load:

Reporting


DevTools - local-dev.robotti.private/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Performance insights
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All
Doc
JS
Fetch/XHR
CSS
Font
Img
Media
Manifest
WS
Wasm
Other
Blocked response cookies
10 ms
20 ms
30 ms
40 ms
80 ms
90 ms
100 ms
110 ms
120 ms
130 ms
140 ms
150 ms
160 ms
Name
activities.png
fa-solid-900.woff2
fa-regular-400.woff2
fa-brands-400.woff2
memvYaGs126MiZpBA-UVWbX2vVnXBbObj2OVTS-muw.woff2
memtYaGs126MiZpBA-UFUIcVXSCEkx2cmqvXlWqWuU6F.woff2
api.js
csp
webclient-infield.html
favicon.ico
3666.runtime.js
6297.infield.js
Headers
Payload
Preview
Response
Initiator
Timing
Cookies
Request Payload
view source
{csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}}
csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}
blocked-uri: "https://www.google.com/recaptcha/api.js"
disposition: "enforce"
document-uri: "https://local-dev.robotti.private/"
effective-directive: "script-src-elem"
original-policy: "default-src 'self';report-uri https://local-dev.robotti.private/
referrer: ""
script-sample: ""
status-code: 200
violated-directive: "script-src-elem"
32 requests
8.2 MB transferred
8.3 MB resources
Finish: 252 ms
DOMContentLoaded: 136 ms
Load:

Reporting


DevTools - local-dev.robotti.private/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Performance insights
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All
Doc
JS
Fetch/XHR
CSS
Font
Img
Media
Manifest
WS
Wasm
Other
Blocked response cookies
10 ms
20 ms
30 ms
40 ms
50 ms
60 ms
70 ms
80 ms
90 ms
100 ms
110 ms
120 ms
130 ms
140 ms
150 ms
160 ms
Name
activities.png
fa-solid-900.woff2
fa-regular-400.woff2
fa-brands-400.woff2
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVTS-muw.woff2
memtYaGs126MiZpBA-UFUIcVXSCEkx2cmqvXlWqWuU6F.woff2
api.js
csp
webclient-infield.html
favicon.ico
3666.runtime.js
6297.infield.js
32 requests
8.2 MB transferred
8.3 MB resources
Finish: 252 ms
DOMContentLoaded: 136 ms
Load:
Headers
Payload
Preview
Response
Initiator
Timing
Cookies
Request Payload
view source
{csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}}
csp-report: {document-uri: "https://local-dev.robotti.private/", referrer: "",…}
blocked-uri: "https://www.google.com/recaptcha/api.js"
So lets add this
disposition: "enforce"
document-uri: "https://local-dev.robotti.private/"
effective-directive: "script-src-elem"
original-policy: "default-src 'self';report-uri https://local-dev.robotti.private/
referrer: ""
script-sample: ""
status-code: 200
violated-directive: "script-src-elem"

Wait… what?

DevTools - local-dev.robotti.private/

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | Recorder | Performance insights

Navigated to https://local-dev.robotti.private/

The key "user-scale" is not recognized and ignored. local-dev.robotti.private/:7

Refused to load the script '<URL>' because it violates the following Content Security Policy directive: "script-src-elem <URL>".

Refused to load the script 'https://local-dev.robotti.private/js/jquery.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/bootstrap.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/glightbox.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/aos.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/purecounter.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/swiper-bundle.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/main.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://www.gstatic.com/recaptcha/releases/-QbJqHfGOUB8nuVRLvzFLVed/recaptcha__en.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". api.js:1

So now, because we have a script-src-elem directive all our scripts are blocked?

Because we didn't include 'self' for that directive

Wait… what?

DevTools - local-dev.robotti.private/

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | Recorder | Performance insights

top | Filter | Default levels | 16 Issues: 8 7 1

Navigated to https://local-dev.robotti.private/

⚠ The key "user-scale" is not recognized and ignored. local-dev.robotti.private/:7

8 Refused to load the script '<URL>' because it violates the following Content Security Policy directive: "script-src-elem <URL>".

Refused to load the script 'https://local-dev.robotti.private/js/jquery.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/bootstrap.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/glightbox.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/aos.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/purecounter.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/swiper-bundle.min.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

Refused to load the script 'https://local-dev.robotti.private/js/main.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". local-dev.robotti.private/:1

▸ Refused to load the script 'https://www.gstatic.com/recaptcha/releases/-QbJqHfGOUB8nuVRLvzFLVed/recaptcha__en.js' because it violates the following Content Security Policy directive: "script-src-elem https://www.google.com/recaptcha/api.js". api.js:1

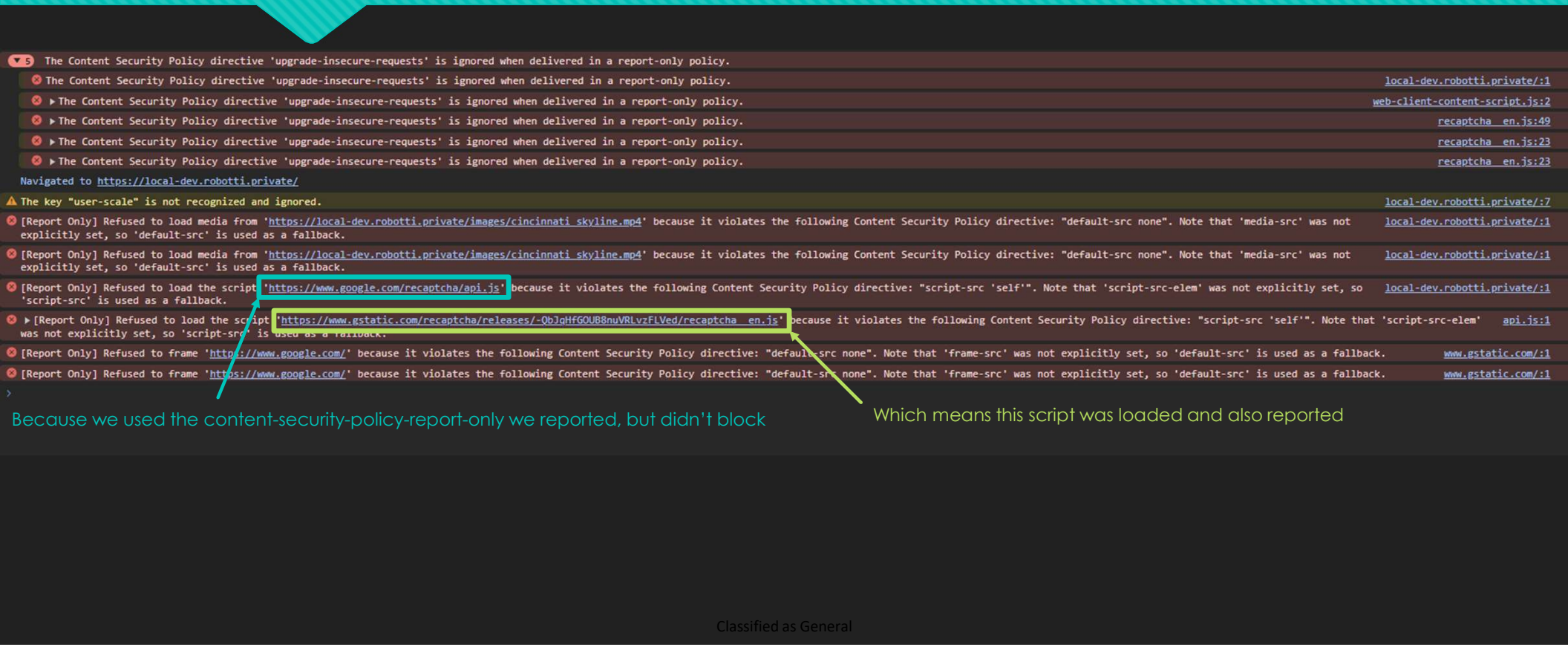
We also have this new script that is being blocked?

Because the script we allowed pulled it in

Lets be smarter about it

```
"helmet": {
  "csp": {
    "directives": {
      "default-src": [
        "'none'"
      ],
      "report-uri": [
        "https://local-dev.robotti.private/csp"
      ]
    },
    "reportOnly": true
  }
}
```

Lets be smarter about it



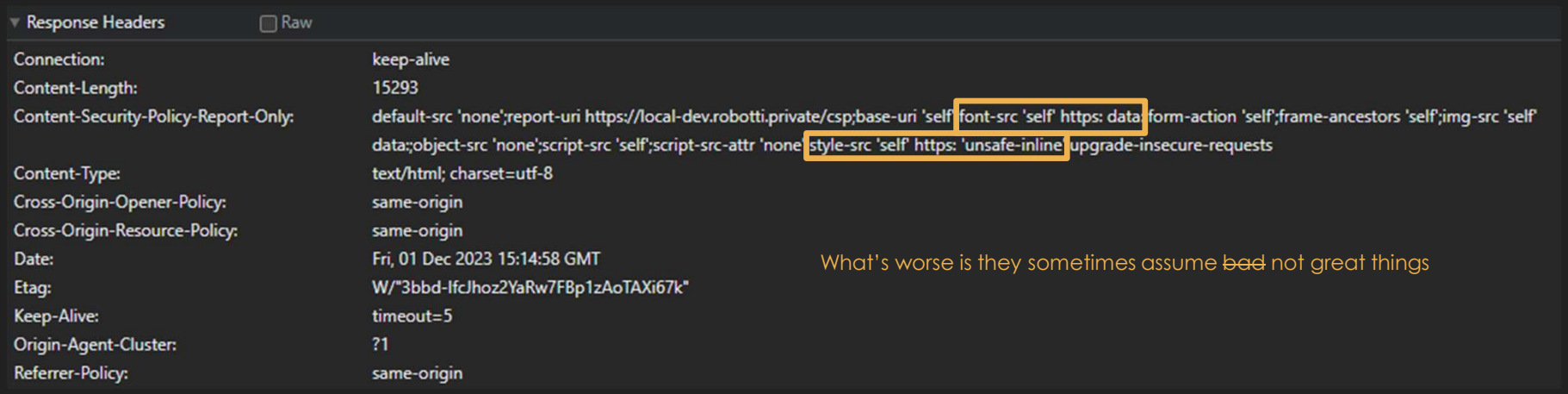
Lets be smarter about it

```
▼5 The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.
  The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.  local-dev.robotti.private/:1
  ▶The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.  web-client-content-script.js:2
  ▶The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.  recaptcha__en.js:49
  ▶The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.  recaptcha__en.js:23
  ▶The Content Security Policy directive 'upgrade-insecure-requests' is ignored when delivered in a report-only policy.  recaptcha__en.js:23
Navigated to https://local-dev.robotti.private/
⚠ The key "user-scale" is not recognized and ignored.  local-dev.robotti.private/:7
[Report Only] Refused to load media from 'https://local-dev.robotti.private/images/cincinnati_skyline.mp4' because it violates the following Content Security Policy directive: "default-src none". Note that 'media-src' was not explicitly set, so 'default-src' is used as a fallback.  local-dev.robotti.private/:1
[Report Only] Refused to load media from 'https://local-dev.robotti.private/images/cincinnati_skyline.mp4' because it violates the following Content Security Policy directive: "default-src none". Note that 'media-src' was not explicitly set, so 'default-src' is used as a fallback.  local-dev.robotti.private/:1
[Report Only] Refused to load the script 'https://www.google.com/recaptcha/api.js' because it violates the following Content Security Policy directive: "script-src 'self'". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.  local-dev.robotti.private/:1
▶[Report Only] Refused to load the script 'https://www.gstatic.com/recaptcha/releases/-QbJqHfGOUB8nuVRLvzFLVed/recaptcha__en.js' because it violates the following Content Security Policy directive: "script-src 'self'". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.  api.js:1
[Report Only] Refused to frame 'https://www.google.com/' because it violates the following Content Security Policy directive: "default-src none". Note that 'frame-src' was not explicitly set, so 'default-src' is used as a fallback.  www.gstatic.com/:1
[Report Only] Refused to frame 'https://www.google.com/' because it violates the following Content Security Policy directive: "default-src none". Note that 'frame-src' was not explicitly set, so 'default-src' is used as a fallback.  www.gstatic.com/:1
>
```

But there's something off about this…

I never added script-src 'self'

Well we assumed…



What's worse is they sometimes assume ~~bad~~ not great things

Some libraries that handle these headers assume you need certain directives set.

Directives:

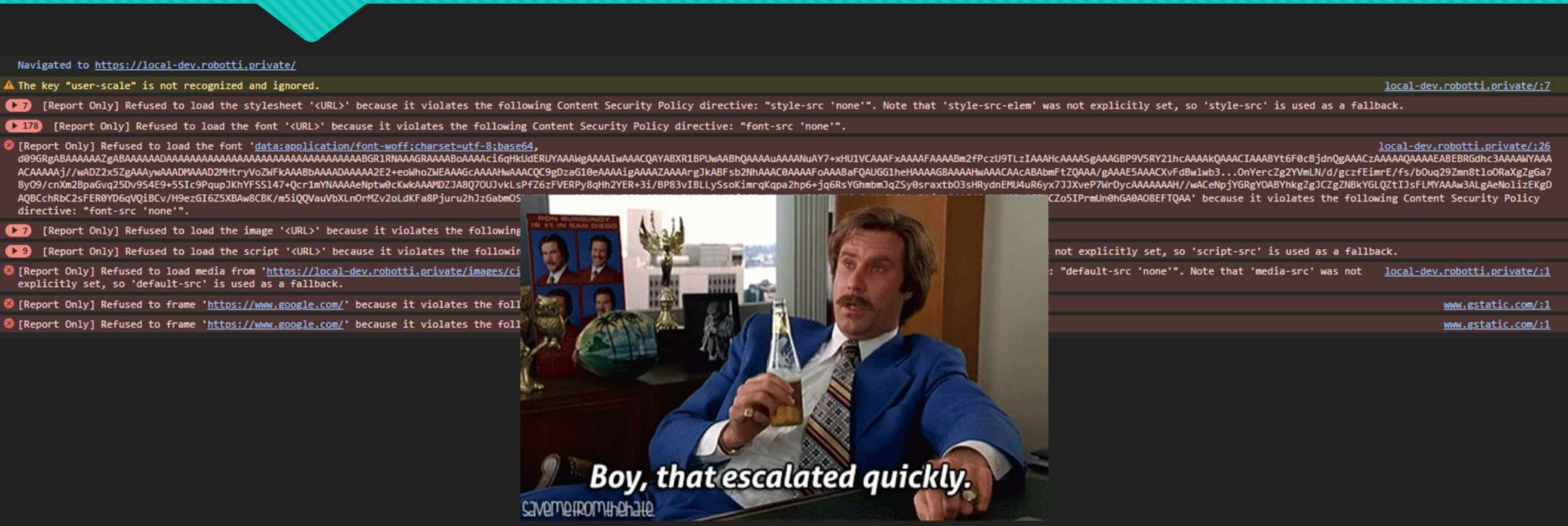
- base-uri
- font-src
- form-action
- frame-ancestors
- img-src
- script-src
- style-src

Okay… Big brain smart time

Let's be super explicit with our policy.
And set everything to 'none'

```
"helmet": {
  "csp": {
    "directives": {
      "default-src": [
        "'none'"
      ],
      "script-src": [
        "'none'"
      ],
      "object-src": [
        "'none'"
      ],
      "style-src": [
        "'none'"
      ],
      "font-src": [
        "'none'"
      ],
      "img-src": [
        "'none'"
      ],
      "connect-src": [
        "'none'"
      ],
      "frame-src": [
        "'none'"
      ],
      "report-uri": [
        "https://local-dev.robotti.private/csp"
      ]
    },
    "reportOnly": true
  }
}
```

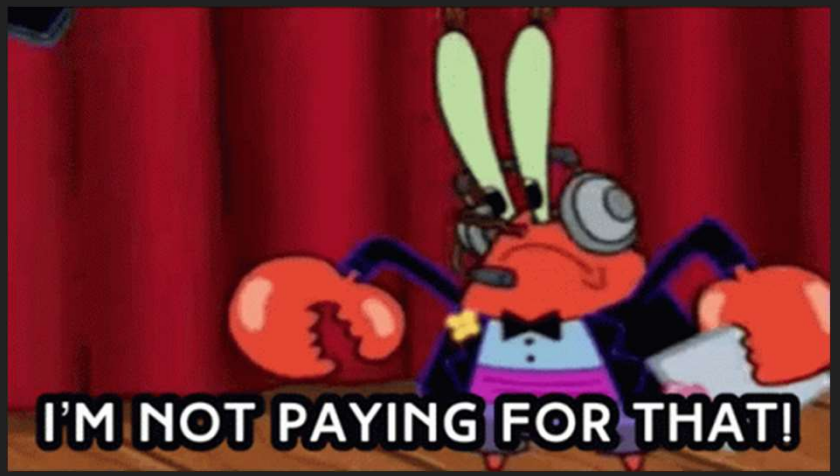
Okay… Big brain smart time



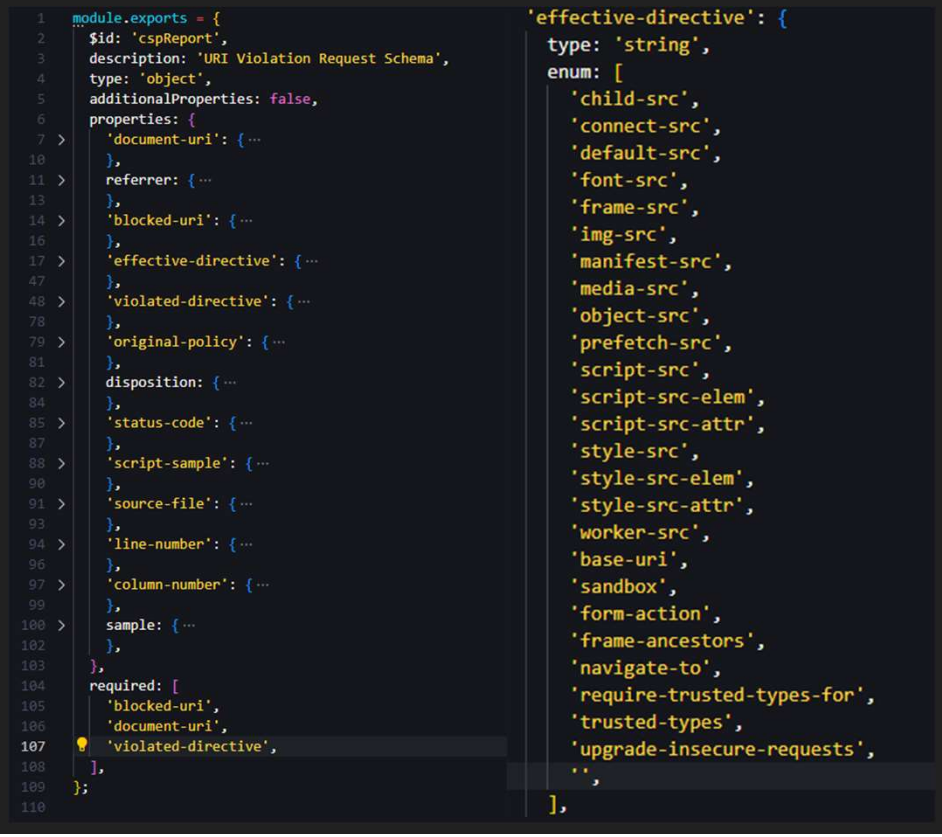
205 Violations for one page

Oh wait… We have report-uri

- Violation reporting services exist
  - Report-uri.com
  - Csper.com



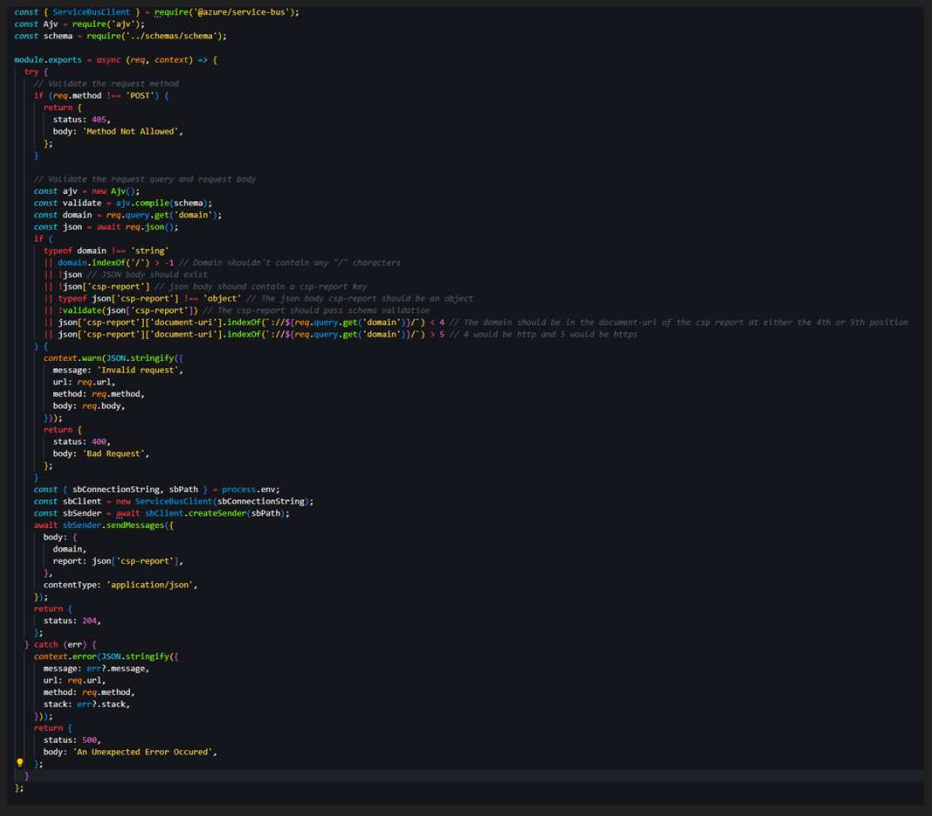
Oh wait… We have report-uri



```
module.exports = {
  $id: 'cspReport',
  description: 'URI Violation Request Schema',
  type: 'object',
  additionalProperties: false,
  properties: {
    'document-uri': {…
    },
    referrer: {…
    },
    'blocked-uri': {…
    },
    'effective-directive': {…
    },
    'violated-directive': {…
    },
    'original-policy': {…
    },
    disposition: {…
    },
    'status-code': {…
    },
    'script-sample': {…
    },
    'source-file': {…
    },
    'line-number': {…
    },
    'column-number': {…
    },
    sample: {…
    },
  },
  required: [
    'blocked-uri',
    'document-uri',
    'violated-directive',
  ],
};
```

```
'effective-directive': {
  type: 'string',
  enum: [
    'child-src',
    'connect-src',
    'default-src',
    'font-src',
    'frame-src',
    'img-src',
    'manifest-src',
    'media-src',
    'object-src',
    'prefetch-src',
    'script-src',
    'script-src-elem',
    'script-src-attr',
    'style-src',
    'style-src-elem',
    'style-src-attr',
    'worker-src',
    'base-uri',
    'sandbox',
    'form-action',
    'frame-ancestors',
    'navigate-to',
    'require-trusted-types-for',
    'trusted-types',
    'upgrade-insecure-requests',
    '',
  ],
```

Oh wait... We have report-uri

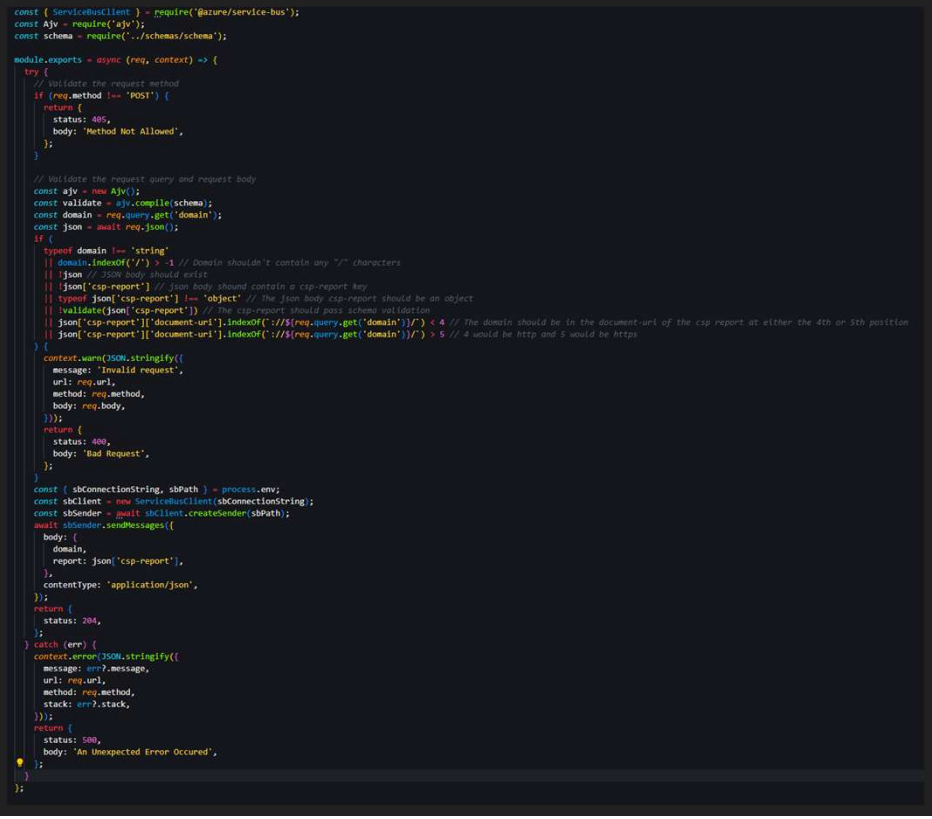


```
const { ServiceBusClient } = require('@azure/service-bus');
const Ajv = require('ajv');
const schema = require('../schemas/schema');

module.exports = async (req, context) => {
  try {
    // Validate the request method
    if (req.method !== 'POST') {
      return {
        status: 405,
        body: 'Method Not Allowed',
      };
    }

    // Validate the request query and request body
    const ajv = new Ajv();
    const validate = ajv.compile(schema);
    const domain = req.query.get('domain');
    const json = await req.json();
    if (
      typeof domain !== 'string'
      || domain.indexOf('/') > -1 // Domain shouldn't contain any "/" characters
      || !json // JSON body should exist
      || !json['csp-report'] // json body should contain a csp-report key
      || typeof json['csp-report'] !== 'object' // The json body csp-report should be an object
      || !validate(json['csp-report']) // The csp-report should pass schema validation
      || json['csp-report']['document-uri'].indexOf(`://${req.query.get('domain')}/`) < 4 // The domain should be in the document-uri of the csp report at either the 4th or 5th position
      || json['csp-report']['document-uri'].indexOf(`://${req.query.get('domain')}/`) > 5 // 4 would be http and 5 would be https
    ) {
      context.warn(JSON.stringify({
        message: 'Invalid request',
        url: req.url,
        method: req.method,
        body: req.body,
      }));
      return {
        status: 400,
        body: 'Bad Request',
      };
    }
    const { sbConnectionString, sbPath } = process.env;
    const sbClient = new ServiceBusClient(sbConnectionString);
    const sbSender = await sbClient.createSender(sbPath);
    await sbSender.sendMessages({
      body: {
        domain,
        report: json['csp-report'],
      },
      contentType: 'application/json',
    });
    return {
      status: 204,
    };
  } catch (err) {
    context.error(JSON.stringify({
      message: err?.message,
      url: req.url,
      method: req.method,
      stack: err?.stack,
    }));
    return {
      status: 500,
      body: 'An Unexpected Error Occured',
    };
  }
};
```

Oh wait… We have report-uri

Why all the validation?

1. Because this is open to all
2. I'm not hypocrite
3. It's javascript



```
const { ServiceBusClient } = require('@azure/service-bus');
const Ajv = require('ajv');
const schema = require('../schemas/schema');

module.exports = async (req, context) => {
  try {
    // Validate the request method
    if (req.method !== 'POST') {
      return {
        status: 405,
        body: 'Method Not Allowed',
      };
    }

    // Validate the request query and request body
    const ajv = new Ajv();
    const validate = ajv.compile(schema);
    const domain = req.query.get('domain');
    const json = await req.json();
    if (
      typeof domain !== 'string'
      || domain.indexOf('/') > -1 // Domain shouldn't contain any "/" characters
      || !json // JSON body should exist
      || !json['csp-report'] // json body should contain a csp-report key
      || typeof json['csp-report'] !== 'object' // The json body csp-report should be an object
      || !validate(json['csp-report']) // The csp-report should pass schema validation
      || json['csp-report']['document-uri'].indexOf(`://${req.query.get('domain')}/`) < 4 // The domain should be in the document-uri of the csp report at either the 4th or 5th position
      || json['csp-report']['document-uri'].indexOf(`://${req.query.get('domain')}/`) > 5 // 4 would be http and 5 would be https
    ) {
      context.warn(JSON.stringify({
        message: 'Invalid request',
        url: req.url,
        method: req.method,
        body: req.body,
      }));
      return {
        status: 400,
        body: 'Bad Request',
      };
    }
    const { sbConnectionString, sbPath } = process.env;
    const sbClient = new ServiceBusClient(sbConnectionString);
    const sbSender = await sbClient.createSender(sbPath);
    await sbSender.sendMessages({
      body: {
        domain,
        report: json['csp-report'],
      },
      contentType: 'application/json',
    });
    return {
      status: 204,
    };
  } catch (err) {
    context.error(JSON.stringify({
      message: err?.message,
      url: req.url,
      method: req.method,
      stack: err?.stack,
    }));
    return {
      status: 500,
      body: 'An Unexpected Error Occured',
    };
  }
};
```

Scaling Across Organization

- Training Developers
- Communication
- Pilot Groups
- Provide Tooling Access
- Phased Approach
- Progress Over Perfection

For science

- What do you do with the reports?
- Bandwidth?
- Other Pages?
- Recommended?
- Was it worth the effort?