Who am I?

• Valentinas Bakaitis
• Security consultant at Aura Information Security
• @vbakaitis on twitter
What is XSS?

- User Supplied Text: `<script>alert('xss');</script>`

- Image with user supplied title `<img title='<> onerror='alert('xss');' />'`

- User supplied URL: `<img src='javascript:alert("xss")' />'`

- User input passed to eval: `<script>eval('userParam=1; alert("xxx");')</script>`
Preventing XSS:

• Escape `<>`

• Escape `'` or `"` in attributes (depending on which one is used).

• Escape space if the attribute is not quoted.

• Also `\` should be escaped as it can double-escape `"` to `\"` which will defeat escaping.

• Check that URLs are using HTTP or HTTPS schema and not javascript:.

• Don’t pass user input to eval or SetTimeout

• Don’t allow users to upload html files to the same domain

• When returning any user controllable resource (e.g. json, image, files, etc) ensure that an appropriate content type is set (don’t use text/html)

• OWASP describes over 80 different common XSS vectors
WHAT IF I TOLD YOU XSS COULD BE PREVENTED
CSP to the rescue!

• Content Security Policy is a security standard introduced to prevent XSS.

• It allows the browser to restrict where scripts can originate from.
Enabling CSP

- CSP is enabled by returning Content-Security-Policy header.
  - nginx.conf add_header
  - apache .htaccess mod_headers
  - IIS web.config <customHeaders>
  - Or return it programmatically
- E.g.: Content-Security-Policy: default-src ‘none’
Configuring CSP

• Start with default-src ‘none’;
  • or default-src ‘self’

• Specify other rules to make your web application work: script-src, style-src, other attributes as necessary.

• CSP encourages you to avoid inline JS and eval() - unsafe-inline and unsafe-eval

• Specify report-uri for reports
Deploying CSP

- Deploy as Content-Security-Policy-Report-Only first
- Review reports, refine it, deploy as Content-Security-Policy
- Make it stricter, keeping your old Content-Security-Policy deploy the new rules under Content-Security-Policy-Report-Only to test it.
This slide is for non devs

• BAs / Prod Owners: make CSP a requirement
• Testers: suggest CSP as improvement
• DevOps: apply CSP to your staging environment and watch people flip out
CSP 2.0

- Frame-ancestors (X-Frame-Options)
- Form-action
- Plugin-types
- Nonces + Hashes
Nonces + Hashes

• CSP: script-src ‘nonce-d41d8cd98' 'sha256-1DCfk1NYWuHM8DgTq1k0ta97gzk +oBDDv4s7woGaPIY='

• <script nonce='d41d8cd98'>alert('1')</script>
Browser support

Content Security Policy 1.0

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

<table>
<thead>
<tr>
<th>Current aligned</th>
<th>Usage relative</th>
<th>Show all</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE 8</td>
<td>Edge 12</td>
<td>Safari 9</td>
</tr>
<tr>
<td>IE 9</td>
<td>Firefox 42</td>
<td>Opera 34</td>
</tr>
<tr>
<td>IE 10</td>
<td>Chrome 47</td>
<td>iOS Safari 8.4</td>
</tr>
<tr>
<td>IE 11</td>
<td>Opera Mini 8</td>
<td>Android Browser 4.3</td>
</tr>
</tbody>
</table>

Global 79.83% + 8.35% = 88.18%
**Browser support**

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources. CSP 2 adds hash-source, nonce-source, and five new directives.

### Content Security Policy Level 2

- **Current aligned**:
  - IE: 8
  - Edge: 12
  - Firefox: 42
  - Chrome: 45
  - Safari: 47
  - Opera: 9
  - iOS Safari: 34
  - Opera Mini: 9
  - Android Browser: 4.3
  - Chrome for Android: 47

- **Usage relative**:
  - IE: 9
  - Edge: 13
  - Firefox: 43
  - Chrome: 48
  - Safari: 9
  - Opera: 34
  - iOS Safari: 35
  - Opera Mini: 8
  - Android Browser: 4.4
  - Chrome for Android: 47

- **Global**:
  - 47.21% + 8.46% = 55.66%
Important note

• CSP is not a replacement for data validation/escaping

• It is a defence-in-depth mechanism
Questions?
Links

• http://www.cspplayground.com/

• https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

• https://www.w3.org/TR/2012/CR-CSP-20121115/

• https://www.w3.org/TR/CSP2/

• https://w3c.github.io/webappsec-csp/

• http://tobias.lauinger.name/papers/csp-raid2014.pdf