UNIVERSITY OF
Southampton

UNIVERSITY OF Southampton

Content Security Policy

COMP3220 Web Infrastructure

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

Cross-Site Scripting (XSS)

The Same-Origin Policy doesn't protect against all cross-site exploits

XSS attacks are a form of code injection

User input used to insert XSS vector (malicious client code) into a web page

- via a GET method form that encodes user input into a URI (non-persistent)
- via a form that changes the state of a resource (persistent)

Browser trusts malicious code and executes it with user credentials

Content Security Policy

Server sends a header that says:

- what a resource is allowed to access
- what should be done if the policy is violated

`Content-Security-Policy:` contains:

- Fetch directives, that control the places from which resources can be loaded
- Document directives, that control the properties of a document or worker environment
- Navigation directives, that control where a user can navigate to
- Reporting directives, that controls where violations are reported to

Fetch directives

`font-src`
: Valid sources for fonts loaded with `@font-face`

`frame-src`
: Valid sources for frames and iframes

`img-src`
: Valid sources for images

`media-src`
: Valid sources for media loaded with `<audio>`, `<video>` and `<track>`

`object-src`
: Valid sources for resources loaded with `<object>`, `<embed>` and `<applet>`

`script-src`
: Valid sources for scripts loaded with `<script>`

`style-src`
: Valid sources for stylesheets

`worker-src`
: Valid sources for `Worker`, `ServiceWorker`

`default-src`
: Fallback rule – default for fetch directive

Document directives

`base-uri`

Valid URIs for <base> elements

`plugin-types`

Valid plugins which can be embedded into a document
i.e. `plugin-types application/x-shockwave-flash application/x-java-applet`

`block-all-mixed-content`

Prevent loading any resources using HTTP when the page is loaded with HTTPS

(defined in West, M. (2016))

Navigation directives

`form-action`

URIs which can be used as the target of form submissions

`frame-ancestors`

Valid parents which may embed a page using `<frame>` or `<iframe>`

`navigate-to`

Valid URIs to which a document can initiate navigation
i.e. `<a>` , `window.location`, etc

UNIVERSITY OF Southampton

Reporting directives

`report-uri` (from CSP Level 2)

- Instructs the user to report violation attempts
- Sent as a JSON document via a POST request

`report-to` (from CSP Level 3)

- Raises a `SecurityPolicyViolationEvent` within the page

`report-uri` has been deprecated in favour of `report-to`

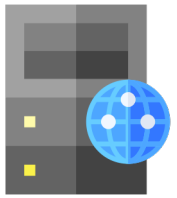
`report-to` is experimental (relies on Reporting API)

Web standards are unnecessarily hard!

CSP example

```
GET /page.html HTTP/1.1
Host: example.org
Referer: http://example.org/
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Security-Policy: default-src 'self';
        img-src image.example.org;
        media-src video.example.org audio.example.org;
        script-src js.example.org;
        report-uri http://example.org/csp-report/
```



CSP violation report

```
{
  "csp-report": {
    "document-uri": "http://example.org/page.html",
    "referrer": "http://example.org/",
    "original-policy": "default-src 'self'; img-src image.example.org;
media-src video.example.org audio.example.org; script-src
js.example.org; report-uri http://example.org/csp-report/"
    "violated-directive": "script-src js.example.org",
    "blocked-uri": "http://malware4u.com/nasty.js",
  }
}
```

Further reading

West, M. et al (2015) *Content Security Policy Level 2*. W3C Recommendation.
https://www.w3.org/TR/CSP2

West, M. et al (2018) *Content Security Policy Level 3*. W3C Working Draft.
https://www.w3.org/TR/CSP3/

West, M. (2016) Mixed Content. W3C Candidate Recommendation.
https://www.w3.org/TR/mixed-content/

Creager, D. et al (2018) *Reporting API*. W3C Working Draft.
https://www.w3.org/TR/reporting-1/

Creager, D. et al (2020) *Reporting API*. Editor's Draft.
https://w3c.github.io/reporting/

UNIVERSITY OF Southampton

Next Lecture: Memento