Content Security Policy (CSP)

Christine Koppelt
19.02.2013

XSS Lücken

- Immer noch unter den häufigsten Sicherheitslücken in Webanwendungen
  - Platz 2 bei den OWASP Top 10
- Prävention trotz einiger Unterstützung durch Frameworks immer noch aufwendig und fehleranfällig
  - Yahoo!Mail (Januar 2013)
  - Ebay (November 2012)
  - Apple (November 2011)

Content Security Policy

- Im Wesentlichen
  - Möglichkeit um die Ausführung von Inline JavaScript Code zu verhindern
  - Möglichkeit eine Whitelist für externe Skripte festzulegen
  - Möglichkeit Quellen für weitere Ressourcen einzuschränken
    - CSS, Bilder, Schriften, (i)Frames, XSLT, Video, Audio
- Im Browser implementiert
- Kommunikation über HTTP Header

Aktueller Stand

- W3C Candidate Recommendation
- Ca. 56% aller Benutzer verfügen über einen Browser der CSP grundsätzlich unterstützt (caniuse.com)
  - Firefox ab 4.0, Chrome ab 14, Safari ab 6.0, iOS Safari ab 6.0
  - teilweise Unterstützung: IE 10
  - Keine Unterstützung: Android Browser, Opera, Opera Mini, Blackberry Browser

HTTP Header

- Offizieller Header Key: `Content-Security-Policy`
  - Firefox: `X-Content-Security-Policy`
  - Chrome (bis Version 24): `X-WebKit-CSP`
- Header Value
  - Besteht wiederum aus Key-Value-Paaren
  - Legen fest, für welchen Ressourcentyp welche Einschränkungen gelten

Beispiele

- `Content-Security-Policy: default-src 'self'`
  - Ressourcen dürfen nur von derselben Domain geladen werden, inline Deklarationen sind ausgeschlossen (auch Subdomains werden ausgeschlossen)
- `Content-Security-Policy: default-src https: 'unsafe-inline'`
  - Externe Ressourcen dürfen nur über https geladen werden, inline Deklarationen sind zulässig
- `Content-Security-Policy: default-src 'self'; img-src *; script-src trusted.example.com`
  - Bilder dürfen von überall geladen werden, Skripte von trusted.example.com, alle anderen Ressourcen nur von der derselben Domain

Demo

Google Go + Chrome/Firefox

Header Werte

- Typ der Ressource
  - default-src
  - script-src, object-src, style-src, img-src, media-src, frame-src, font-src, connect-src
- Restriktionen
  - Schema
    - z.B. https
  - Host
    - Kombinationen aus Schema, Wildcard, Hostname und Port
  - Keyword
    - none, self, unsafe-inline, * (Wildcard)

Probleme

- Altanwendungen enthalten häufig Unmengen an Inline JavaScript und Styles
- Einige (JavaScript)-Webframeworks verwenden inline JavaScript und Styles oder laden Bilder nach
- Einige Browser-Plugins verwenden ebenfalls inline JavaScript und Styles