Content Security Policy

Christopher Christensen

CSP

Content Security Policy

- zusätzliche Sicherheitsschicht
- hilft, bestimmte Arten von Angriffen zu erkennen / entschärfen

Vor was schützt CSP?

- Cross-Site-Scripting (XSS)
- Data-Injection-Angriffe

Was ist das Hauptziel von CSP?

- XSS-Angriffe abschwächen und melden

Wie funktioniert CSP?

- Ermöglicht Server-Admins, die Vektoren, durch die XSS auftreten können, zu reduzieren und eliminieren
- Domäne vorschreiben, die Browser als gültige Quelle für ausführbare Skripte betrachten soll
- CSP-kompatibler Browser führt dann nur noch Skripte aus, die in Quelldateien von diesen Whitelist-Domänen geladen wurden
- Browser ignoriert restliche Daten (inkl. Inline-Skripts, etc.)
- Oder Skripts global verbieten

Konfiguration des CSP

- Hinzufügen des Content-Security-Policy-HTTP-Headers
- Hinzufügen von Angabe von Werten zur Kontrolle der Ressourcen, die User-Agent laden darf

CSP-HTTP-Header Richtlinie

- `Content-Security-Policy: policy`
- Wird durch eine Reihe von Direktiven beschrieben

CSP-HTTP-Header Direktiven

- beschreiben je die Politik für einen bestimmten Ressourcentyp
- Richtlinie sollte eine `default-src`-Direktive enthalten als Fallback, falls andere keine haben

Beispiele für CSP-HTTP-Header Direktiven

- `default-src`
- `script-src`
- `style-src`
- `img-src`
- `connect-src`
- `font-src`
- `object-src`
- `media-src`
- `frame-src`
- `sandbox`
- `report-uri`
- `child-src`
- `form-action`
- `frame-acestors`
- `plugin-types`

Mit Header Direktive alles verbieten

- `default-src 'none'`
- Firewall-Prinzip

Mit Header Direktive Skripts von definierten Domains erlauben

- `script-src 'self'` (von eigener Domain)
- `script-src ajax.googleapis.com`

Mit Header Direktive Kommunikation über Websockets erlauben

- `connect-src 'self' ws://dvwc.el.eee.intern;` (von eigener und anderer Domain)

Mit Header Direktive Laden von Ressourcen erlauben

- `img-src 'self'`
- `style-src 'self`
- etc.