CSP Evaluator

CSP Evaluator allows developers and security experts to check if a Content Security Policy (CSP) serves as a strong mitigation against cross-site scripting attacks. It assists with the process of reviewing CSP policies, which is usually a manual task, and helps identify subtle CSP bypasses which undermine the value of a policy. CSP Evaluator checks are based on a large-scale study and are aimed to help developers to harden their CSP and improve the security of their applications. This tool (also available as a Chrome extension) is provided only for the convenience of developers and Google provides no guarantees or warranties for this tool.

Content Security Policy

Sample unsafe policy   Sample safe policy

```
default-src 'self';
script-src 'self' 'nonce-nRnXMpwEA2PkWLz9ogFUXA=='  'strict-dynamic';
style-src 'self' 'nonce-nRnXMpwEA2PkWLz9ogFUXA==';
font-src 'self'  https://fonts.gstatic.com;
img-src 'self' https://stage.luminousindia.com https://stgblobaccount.blob.core.windows.net
    https://videodelivery.net;
media-src 'self' https://stage.luminousindia.com https://stgblobaccount.blob.core.windows.net ;
connect-src 'self' https://uatapi.luminousindia.com https://stage-cms.luminousindia.com
    https://mpdev.luminousindia.com;
object-src 'none';
base-uri 'self';
frame-src 'self';
frame-ancestors 'none';
manifest-src 'self';
worker-src 'self' ;
```

CSP Version 3 (nonce based + backward compatibility checks)

CHECK CSP

Evaluated CSP as seen by a browser supporting CSP Version 3

expand/collapse all

default-src	
script-src	Consider adding 'unsafe-inline' (ignored by browsers supporting nonces/hashes) to be backward compatible with older browsers. Consider adding https: and http: url schemes (ignored by browsers supporting 'strict-dynamic') to be backward compatible with older browsers.
style-src	
font-src	
img-src	
media-src	
connect-src	
object-src	
base-uri	
frame-src	
frame-ancestors	
manifest-src	
worker-src	
upgrade-insecure-requests	
report-to	
report-uri	
require-trusted-types-for [missing]	Consider requiring Trusted Types for scripts to lock down DOM XSS injection sinks. You can do this by adding "require-trusted-types-for 'script'" to your policy.

Legend

- High severity finding
- Medium severity finding
- Possible high severity finding
- Directive/value is ignored in this version of CSP
- Possible medium severity finding
- Syntax error
- Information
- All good