CSP Directives:

IAW: <https://content-security-policy.com/#source_list> and <https://blog.sucuri.net/2023/04/how-to-set-up-a-content-security-policy-csp-in-3-steps.html>

Common CSP Directives & Examples

|  |  |  |
| --- | --- | --- |
| **Directive** | **Example** | **Description** |
| default-src | default-src ‘self’ cdn.example.com; | Default policy, used in any case (JavaScript, Fonts, CSS, Frames etc.) except if overridden by a more precise directive. |
|  |  |  |
| script-src | script-src ‘self’ js.example.com; | Defines authorized sources for scripts ( JavaScript(JS) ) |
|  |  | example for cybercatalyst website: script-src 'self’; |
| style-src | style-src ‘self’ css.example.com; | Defines authorized sources for stylesheets (CSS) |
|  |  | example for cybercatalyst website: style-src 'self’; |
| img-src | img-src ‘self’ img.example.com; | Defines authorized sources for images, or link element related to an image type (ex: rel=”icon”) |
|  |  | example for cybercatalyst website: img-src 'self'; |
| connect-src | connect-src ‘self’; | Policy applies to connections from a XMLHttpRequest (AJAX) or a WebSocket |
|  |  | example for cybercatalyst website: connect-src ‘self’; ????? |
| font-src | font-src font.example.com; | Defines authorized sources where fonts files can be loaded from |
|  |  | example for cybercatalyst website: style-src 'self'’ 'fonts.googleapis.com'; <- is this correct???? |
| object-src | object-src ‘self’; | Defines authorized sources for plugins (ex: <object> or <embed>) |
|  |  | example for cybercatalyst website: object-src 'none'; |
| media-src | mia-src media.example.com; | Defines authorized sources for media elements (ex: <video>, <audio>) |
|  |  | example for cybercatalyst website: media-src 'self'; <- need to put YouTube??? |
| frame-src | frame-src ‘self’; | Defines authorized sources for loading frames (iframe or frame) |
|  |  | example for cybercatalyst website: frame-src 'self'; |
| sandbox | sandbox allow-forms allow-scripts; | Enables a sandbox for the requested resource similar to the iframe sandbox attribute. |
| report-uri | report-uri /some-report-uri; | Instructs a browser to create a report of policy failures. If a piece of content is blocked, the browser will send a report of the information to this URI.Alternatively, you can use Content-Security-Policy-Report-Only as the HTTP header name to receive the reports without blocking anything. |
|  |  | example for cybercatalyst website: ????? |
| child-src | child-src 'self' | Defines valid sources for web workers and nested browsing contexts loaded using elements such as <frame> and <iframe> |
|  |  | example for cybercatalyst website: child-src 'self' 'www.youtube.com'; <- is this correct???? |
| form-action | form-action 'self'; | Defines valid sources that can be used as an HTML <form> action. |
| frame-ancestors | frame-ancestors 'none'; | Defines valid sources for embedding the resource using <frame> <iframe> <object> <embed> <applet>. Setting this directive to 'none' should be roughly equivalent to X-Frame-Options: DENY |
| base-uri | base-uri 'self' | Defines a set of allowed URLs which can be used in the src attribute of a HTML base tag. |
|  |  | example for cybercatalyst website: base-uri 'self'; |
| report-to | report-to groupName; | Defines a reporting group name defined by a Report-To HTTP response header. |
| worker-src | worker-src 'none'; | Restricts the URLs which may be loaded as a Worker, SharedWorker or ServiceWorker. |
| manifest-src | manifest-src 'none'; | Restricts the URLs that application manifests can be loaded. |
| prefetch-src | prefetch-src 'none' | Defines valid sources for request prefetch and prerendering, for example via the link tag with rel="prefetch" or rel="prerender": |
| navigate-to | navigate-to example.com | Restricts the URLs that the document may navigate to by any means. For example when a link is clicked, a form is submitted, or window.location is invoked. If form-action is present then this directive is ignored for form submissions. |
| upgrade-insecure-requests | upgrade-insecure requests | Automatically Converts urls from http to https for links, images, javascript, css, etc. |
| block-all-mixed-content | block-all-mixed-content | Blocks requests to non secure http urls. |

Allows you to add header for the Content Security Policy (CSP).

|  |  |  |
| --- | --- | --- |
| Nginx Content-Security-Policy Header | |  |
|  | | In your server {} block add: |
|  | | add-header add\_header Content-Security-Policy “default-src ‘self’;”; |
| Apache Content-Security-Policy Header | | Add the following to your httpd.conf in your VirtualHost or in an .htaccess file: |
|  |  | Header set Content-Security-Policy "default-src 'self';" |

Overall CSP for Cybercatalyst in : Header set Content-Security-Policy "script-src 'self' bigin.zoho.com; style-src 'self' fonts.googleapis.com; img-src 'self'; frame-src 'self'; child-src 'self' www.youtube.com; base-uri 'self'; prefetch-src 'none'; frame-ancestors www.youtube.com forms.office.com; "

Here is a list of common source values for -src directives.

Common Source Values for -src Directives

|  |  |  |
| --- | --- | --- |
| **Value** | **Example** | **Description** |
| \* | img-src \* | Wildcard, allows any URL except data: blob: filesystem: schemes. |
| ‘none’ | object-src ‘none’ | Prevents loading resources from any source. |
| ‘self’ | script-src ‘self’ | Allows loading resources from the same origin (same scheme and domain name). |
| data: | img-src ‘self’ data: | Allows loading resources via the data scheme (eg Base64 encoded images). |
| domain.example.com | img-src domain.example.com | Allows loading resources from the specified domain name. |
| \*.example.com | img-src \*.example.com | Allows loading resources from any subdomain under example.com |
| https://cdn.com | img-src https://cdn.com | Allows loading resources only over HTTPS matching the given domain |
| https: | img-src https: | Allows loading resources only over HTTPS on any domain. |
| ‘unsafe-inline’ | script-src ‘unsafe-inline’ | Allows use of inline source elements such as style attribute, onclick, or script tag bodies and javascript: URIs |
| ‘unsafe-eval’ | script-src ‘unsafe-eval’ | Allows use of unsafe dynamic code evaluation like JavaScript eval() |

example in .html file: <meta http-equiv="Content-Security-Policy" content="default-src 'self'; img-src https://\*; child-src 'none';">

example in HTTP response header : Content-Security-Policy: default-src 'none'; script-src 'self'; img-src 'self'; style-src 'self';base-uri 'self';form-action 'self'

example in .htaccess file : Header set Content-Security-Policy-Report-Only "default-src 'self'; img-src \*" <- this delivers a CSP that is not enforced but is reported.