Content Security Policy

作者：Vanessa
原文链接：https://ld246.com/article/1378266224085
来源网站：链滴

介绍
Content Security Policy(CSP) 是一个<a href="https://en.wikipedia.org
wiki/Computer_security">计算机安全的概念，用来阻止<a href="https://en.wikipedia.org/
iki/Cross-site_scripting">跨站点脚本（XSS）和相关的攻击。这是 W3C 工作组在 Web 应用
序

安全性的一个候选推荐标准。CSP 提供了一个标准的 HTTP 头，它允许网站所有者
明批准的内容来源，从而使浏览器允许重载页面——覆

盖类型有 JavaSc， CSS, HTML
rames， fonts， images 和嵌入对象，例如 Java applets， ActiveX，音频和视频文件。

现状
CSP 最初是由 Mozilla 基金会开发的，首次在 Firefox4
实现。

截至2012年 CSP 一直作为 W3C 的候选。下面的头名称为 CSP 试验中实现了
一部分：

 Content-Security-Policy - 由 W3C 文档提出的标准名称。Chrome 25
其进行了支持。在 2013 年 8 月 6 日发布的 Firefox 23 对

其进行了支持。

X-WebKit-CSP - 在 2011 年引入到 Chrome 和其它基于 WebKit 的浏览器（Safari浏览器）中
试验性质的头名称。
X-Content-Security-Policy - 基于 Gecko 的浏览器（Firefox 4 到 Firefox 22，Thunderbird 3.
，SeaMonkey 2.1）引进的试验性

质的头名称。

截至 2013 年发布的 Firefox 版本只支持实验性质的头名称 X
Content-Security-Policy。Firefox 使用“unsafe-inline”和“unsafe-

eval” 指令来
代废弃的语法: options inline-script eval-script。

Internet Explorer 10 使用实验性
的 X-Content-Security-Policy 头名称，可以支持沙箱指令。

W3C 正在开发新的 CSP 1
1 规范。

说明
如果服务器返回的头中包含 Content
Security-Policy，客户端将强制声明一个白名单策略。一个安全的例子就是为 JavaScript 的执行使<b
/>
用更严格的模式，以阻止某些跨站点脚本攻击。在实践中，这意味着一些默认的功能将被
用：

 内嵌的 JavaScript（例如：<script> </script>，DOM 中类似 onclick 等的
性事件，以及 A 标签中 href 属性值为“javascript”

开头的）被阻止 - 所有的脚本代
必须存放在独立的文件中，并且这些文件都来自白名单的域名中（可使用 unsafe-inline 来进行设置

 动态代码（通过 eval() 和字符串参数来使用 setTimeout 和 setInterval）被阻止（可使用
nsafe-eval 来进行设置）

为 CSP-compatible 的 Web 应用程序推荐的编码实践：使
外部源文件（<script src>）来加载代码，解析JSON，而不是使用动态代码及在其他函数段内使用内
方法。

除了限制执行 JavaScript 外，策略允许从指定的页面中加载特定的资源。这些
源包括 CSS，JavaScript，图像，frames，applets，Ajax等。

如果服务器响应头包含
Content-Security-Policy-Report-Only，兼容客户端的监测和报告仅不执行白名单策略。在开发过
中，这是非常有用。

报告
任何被请求的资源或脚
违反了执行政策，浏览器将会触发 POST 请求到 report-uri containing details 所指定的值中。

其他
 <a href="https://en.wikipedia.org/wiki/NoScr
pt">NoScript —— 防 XSS 保护及 Application Boundaries Enforcer（ABE）