> 以上示例使用了 `js-xss` 来实现， 可以看到在输出中保留了 `h1` 标签且过滤了 `script` 标签

2. CSP

> `CSP` 本质上就是建立白名单， 开发者明确告诉浏览器哪些外部资源可以加载和执行。我们只需要配置规则， 如何拦截是由浏览器自己实现的 。我们可以通过这种方式来尽量减少 `XSS` 攻击。

通常可以通过两种方式来开启CSP：

- 设置 `HTTP Header` 中的 `Content-Security-Policy`
- 设置 `meta` 标签的方式 `<meta http-equiv="Content-Security-Policy">`

这里以设置 `HTTP Header` 来举例

只允许加载本站资源

```
Content-Security-Policy: default-src ‘self’
```

只允许加载 HTTPS 协议图片

```
Content-Security-Policy: img-src https://*
```

允许加载任何来源框架

```
Content-Security-Policy: child-src 'none'
```

> 当然可以设置的属性远不止这些，你可以通过查阅文档 的方式来学习， 这里就不过多赘述其他的属性了。

> 对于这种方式来说， 只要开发者配置了正确的规则，那么即使网站存在漏洞，攻击者也不能执行它的攻击代码， 并且 `CSP` 的兼容性也不错。