跨站脚本攻击

W3.1. 使用 LiveHTTPHeaders 观察到，向 `www.example.com` 发送以下 GET 请求可以删除用户拥有的页面（只有页面所有者可以删除）。

```
http://www.example.com/delete.php?pageid=5

GET /delete.php?pageid=5
Host: www.example.com
...
```

请编写恶意 JavaScript。若该程序被注入受害者在 `www.example.com` 上的某个页面，它应能删除受害者拥有的页面。

W3.2. 使用 LiveHTTPHeaders 观察到，向 `www.example.com` 发送以下 POST 请求可以删除用户拥有的页面（只有页面所有者可以删除）。

```
http://www.example.com/delete.php

POST /delete.php HTTP/1.1
Host: www.example.com
...
Content-Length: 8
pageid=5
```

请编写恶意 JavaScript。若该程序被注入受害者在 `www.example.com` 上的某个页面，它应能删除受害者拥有的页面。

W3.3. 教材代码清单 C.2 中的 C 是 XSS 章节号，其实际值随教材版本而异。在发送修改个人资料的 Ajax 请求前，代码会检查目标是否为 Samy 自己。这个检查的主要目的是什么？如果去掉检查，攻击还能成功吗？为什么添加好友攻击（代码清单 C.1）不需要同样的检查？

W3.4. 为抵御 XSS，开发人员决定只在浏览器端过滤：每个页面都加入 JavaScript，在数据发送到服务器之前删除其中的所有 JavaScript 代码。假设过滤逻辑本身十分完善，这种方法能防止 XSS 吗？

W3.5. XSS 和 CSRF 攻击有什么区别？

W3.6. 防御 CSRF 的秘密令牌能否用于抵御 XSS？

W3.7. 防御 CSRF 的 SameSite Cookie 能否抵御 XSS？

W3.8. 为了从用户输入中过滤 JavaScript，是否只需查找并删除 `script` 标签？

W3.9. ★ ★
如果可以修改浏览器行为，你会加入什么机制来降低 XSS 风险？

W3.10. ★ ★ ★
程序通常有两种方式产生自身副本。一种是从外部取得自身内容，例如从文件、DOM 节点或网络读取；另一种不借助外部输入，仅由代码自身生成完整副本，称为 `quine`。Quine 是不接受输入、只输出自身源代码的非空程序，也常被称为自复制程序。代码清单 10.3 中的 JavaScript 蠕虫不是 quine，因为它通过 `document.getElementById()` 从 DOM 读取了自身内容。

请编写一个 quine，把它放入 `Elgg` 用户的个人资料。当其他人访问该资料时，代码会执行，并在警告框中输出自身副本。可参考各种语言的 quine 示例。

进一步的挑战是把代码清单 10.3 改写成 quine，同时保持原有行为：向受害者的个人资料添加一条语句以及蠕虫自身的副本。

W3.11. XSS 漏洞的根本原因是 HTML 允许 JavaScript 代码与数据混合。从安全角度看，代码与数据混合十分危险。除 XSS 外，请再举两个例子说明这种风险。

W3.12. 为什么 CSP（内容安全策略）能够有效抵御 XSS？这种方法有什么缺点？

W3.13. 能否使用 CSP（内容安全策略）来抵御 CSRF 攻击？为什么或为什么不呢？

W3.14. 以下 PHP 代码返回一个网页，并为页面中的 JavaScript 设置 CSP。哪些位置的 JavaScript 可以执行？

```
<?php
  $cspheader = "Content-Security-Policy:".
               "default-src 'self';".
               "script-src 'self' 'nonce-1rA2345' example.com".
               "";
  header($cspheader);
?>
<html>
<script type="text/javascript" nonce="1rA2345">
  ... JavaScript Code ...                                    ❶
</script>
```

```
<script type="text/javascript" nonce="2rB3333">
  ... JavaScript Code ...                                    ❷
</script>

<script type="text/javascript">
  ... JavaScript Code ...                                    ❸
</script>

<script src="script.js"> </script>                           ❹

<script src="https://example.com/script2.js"> </script>      ❺

<button onclick="alert('hello')">Click me</button>           ❻
</html>
```