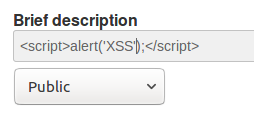
**网络空间安全实训Lab[5]**

**实验报告**

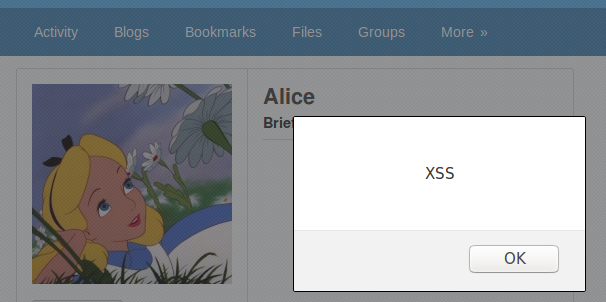
**57118117 谌雨阳**

**Task 1: Posting a Malicious Message to Display an Alert Window**

1、向登录账户的brief description框中键入如下语句：

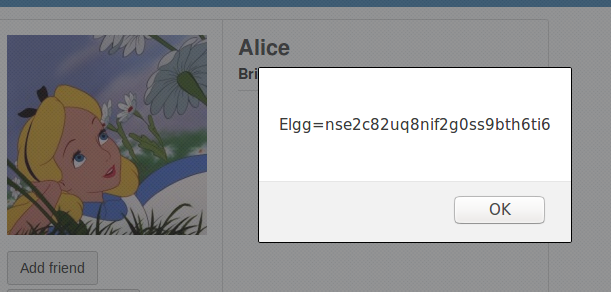


2、查看该用户的profile会弹出如下警告框：



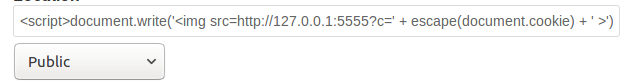
**Task 2: Posting a Malicious Message to Display Cookies**

1、在task1的相同位置将输入语句改为：<script>alert(document.cookie);</script>，访问其profile可以得到cookie，显示在下图中的警告框中：

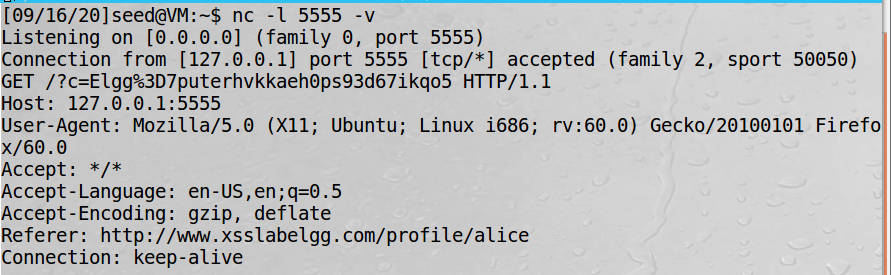


**Task 3: Stealing Cookies from the Victim’s Machine**

1、在profile中输入如下语句：



2、打开终端并执行nc -l 5555 -v命令：



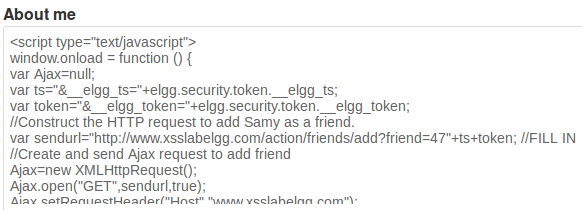
可以从受害人的机器中盗取cookie。

**Task 4: Becoming the Victim’s Friend**

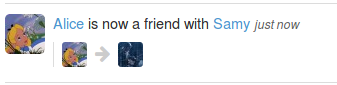
1、从其他账户登录，打开HTTP Header Live工具，了解添加Samy为好友的请求语句和Samy的GUID：



2、以Samy的账号登录，切换html模式后在about me栏中输入如下代码：



3、从其他用户端访问Samy的profile，被自动添加为好友：

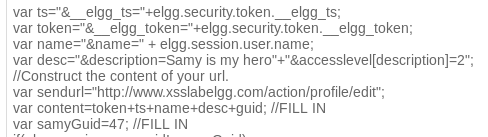


Question1：这两行语句的作用是从JavaScript变量中获取timestamp和secret token。

Question2：不能。XSS攻击的核心就是混淆文本与代码，在文本编辑模式下无法形成攻击。

**Task 5: Modifying the Victim’s Profile**

1、在Samy的about me栏中以html模式输入如下代码：



2、从其他账户访问Samy的profile，受害者的about me栏遭到更改：

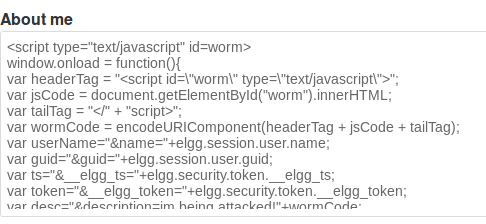


Question3：

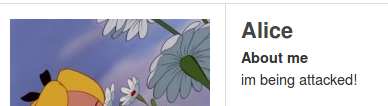
A：该行语句为判断语句，判断guid是否为攻击者本人，需要该语句的原因是为了防止在攻击者的profile中复写同样的恶意代码。

**Task 6: Writing a Self-Propagating XSS Worm**

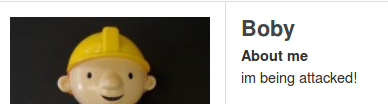
1、为了实现蠕虫的自动传播，需要对上一个task中输入的代码进行修改：



2、从alice的账户访问samy，遭到攻击：



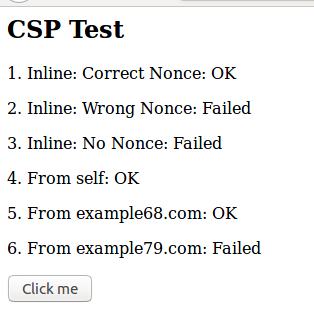
3、再以boby的账户访问alice，同样遭到攻击：

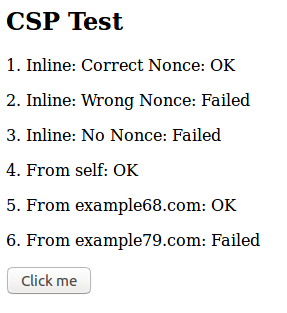


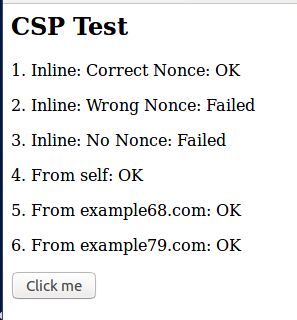
证明该蠕虫病毒实现了自动传播。

**Task 7: Defeating XSS Attacks Using CSP**

1、运行http\_server.py文件，启动服务器后，用浏览器制定三个域名：







观察：发现对于三个域名分别有不同的条目显示为Failed。

2、将python的http服务器代码修改成如下代码：

#!/usr/bin/env python3

from http.server import HTTPServer, BaseHTTPRequestHandler

from urllib.parse import \*

class MyHTTPRequestHandler(BaseHTTPRequestHandler):

def do\_GET(self):

o = urlparse(self.path)

f = open("." + o.path, ’rb’)

self.send\_response(200)

self.send\_header(’Content-type’, ’text/html’)

self.end\_headers()

self.wfile.write(f.read())

f.close()

httpd = HTTPServer((’127.0.0.1’, 8000), MyHTTPRequestHandler)

httpd.serve\_forever()

即去除CSP对XSS的抵御之后，各条目皆显示为OK：

