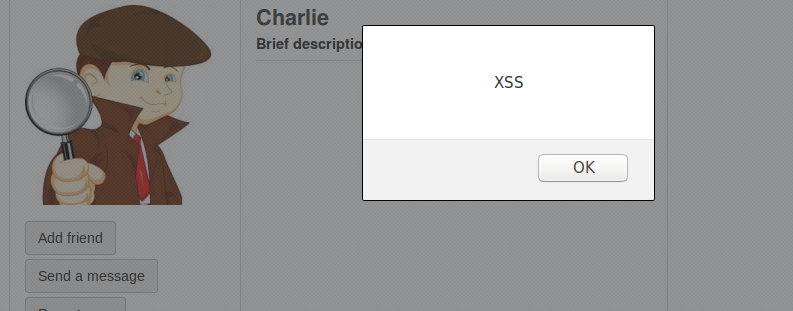
57118130 王嘉麟

Task1：

在charlie 的profile中的brief description中加入

<script>alert(‘XSS’)</script>

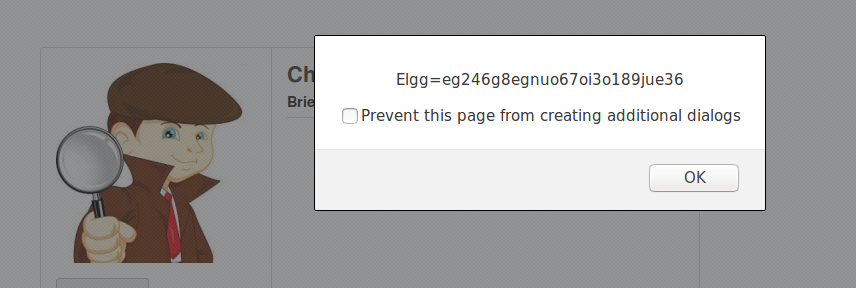
效果如下



网页注册后，在hosts文件中加入回环地址登记，无法通过浏览器打开界面

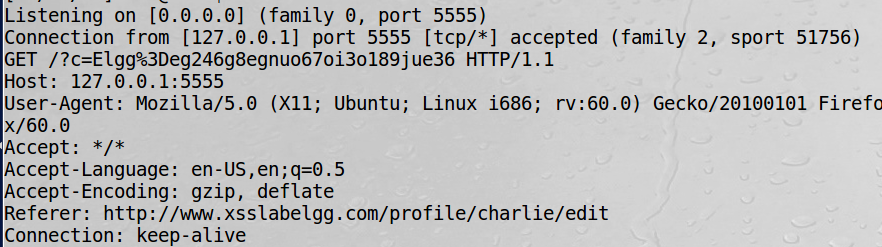
故接下来直接在浏览器上做

Task2：



Task3：

Charlie的cookie被发送到监听者客户端5555端口上，在攻击中只要填入自己的ip地址和想要监听的端口号即可，当有人访问Charlie profile界面时，他的cookie就会被发送



Task4：

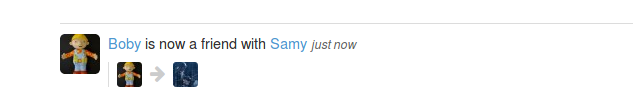
观测到加好友请求

<http://www.xsslabelgg.com/action/friends/remove?friend=47&__elgg_ts=1600330855&__elgg_token=IgXzdhCLtHf8OAl98tBCnQ&__elgg_ts=1600330855&__elgg_token=IgXzdhCLtHf8OAl98tBCnQ>

ts和token分别出现了两次，故构造表达式如下

在sendurl中填入"http://www.xsslabelgg.com/action/friends/add?friend=47" + ts + token + ts + token

然后登陆boby账号访问samy主页

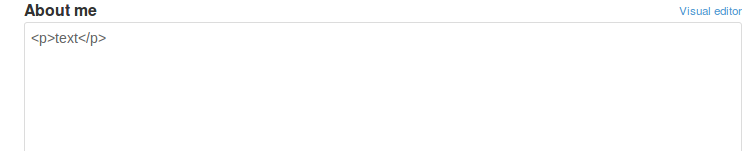


成功添加好友

Q1 ......why they are needed?

Elgg\_ts is used for time check and Elgg\_token is used for authoring the operation. They are turned on to defeat possible CSRF attacks. We definitely need them to trigger the attack and success.

Q2

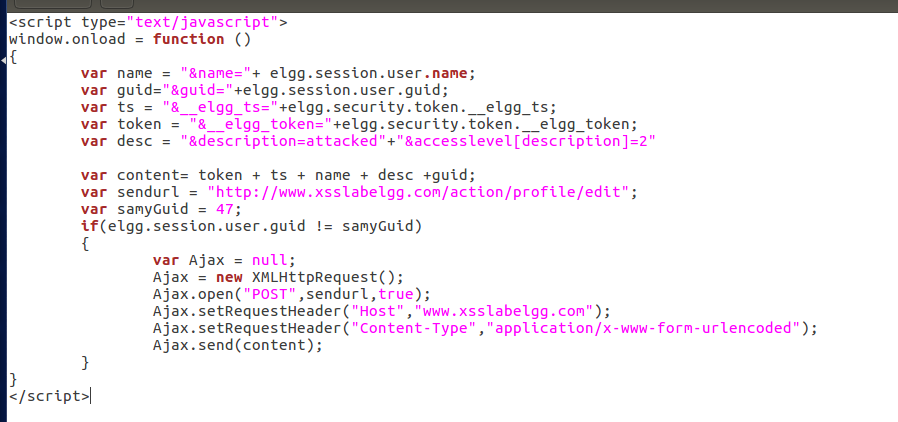


Yes we can trigger the attack by doing as follows:

“text”</p> <script> “malicious code” </script> <p>”text”

Task5:

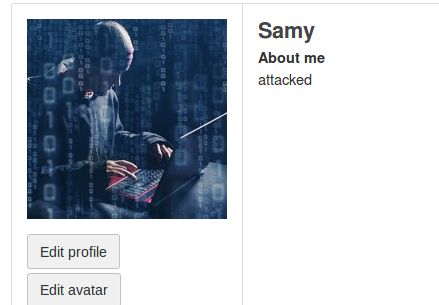
放到samy主页中



boby浏览samy主页中招如图



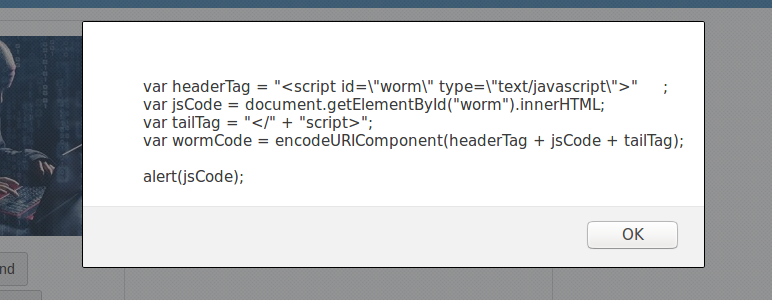
Q3



Samy自己也被攻击，移除的语句用于将samy本人排除在外

Task6：

测试wormCode如下



成功

将之前的代码拼接起来后，标签上添加id=worm一项，中间加入

var headerTag = "<script id=\"worm\" type=\"text/javascript\">" ;

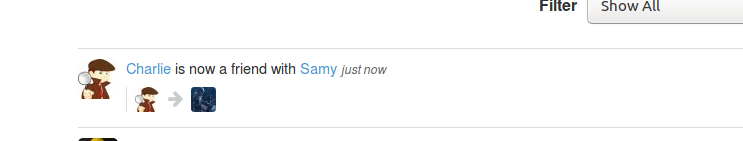
var jsCode = document.getElementById("worm").innerHTML;

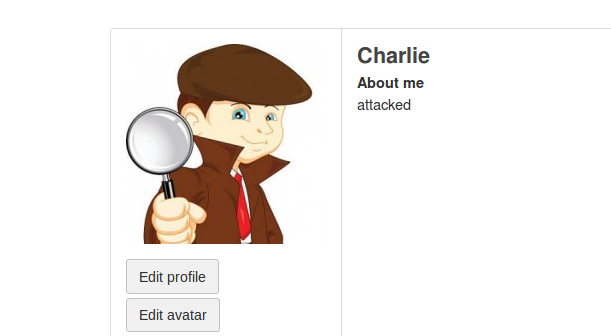
var tailTag = "</" + "script>";

var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

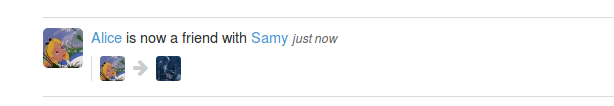
再将wormcode放到要修改的profile description中去。

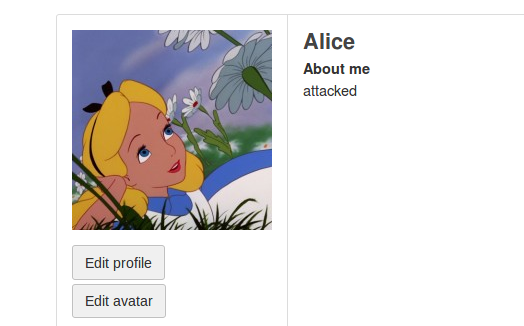
当Charlie访问samy主页时，出现以下情况



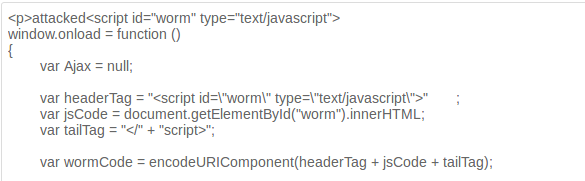


Alice时Charlie的朋友，无意中访问了Charlie的主页情况如下图





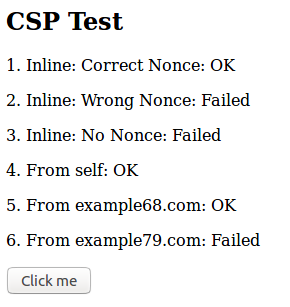
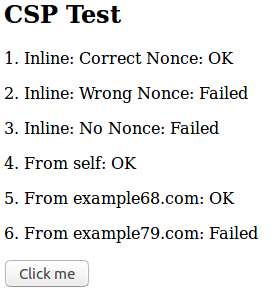
Alice和Charlie用text模式编辑About Me时发现恶意代码被复制到自己profile中



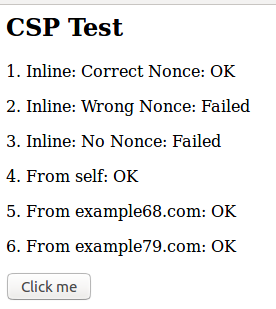
Task7

浏览器分别连接以下网址对应8000端口

Example32 example68



Example79

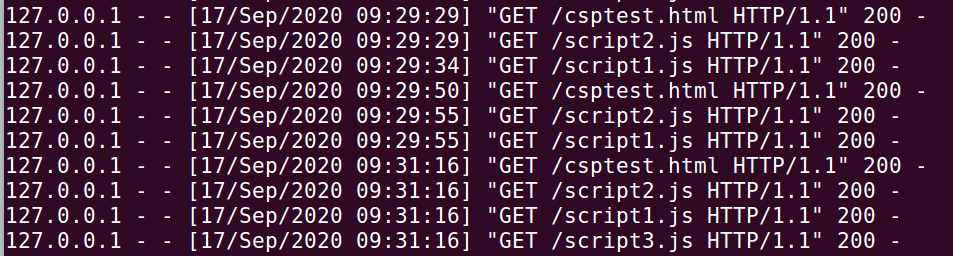


参考csptest.html 文件和py文件 1rA2345 是正确的nonce，在script-src可信源之列，故三个script都允许运行，显示OK，没有nonce和错误的nonce对服务器来说属于不可信的源，script不允许运行。

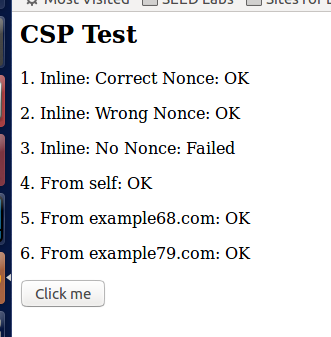
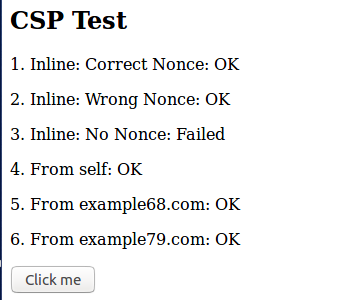
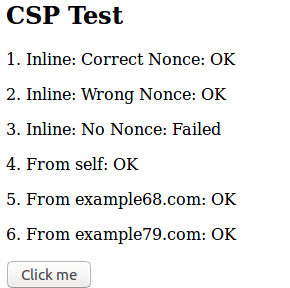
Csptest默认源是self，而script1.js在本目录下，允许运行

对于来自area5，即example68的script，py文件中添加了script-src可信源，允许运行。

Area6，即example79的script只有79自己能够运行。



更改服务器python文件后

三个网站都符合要求

具体更改如下

"script-src 'self' \*.example68.com:8000 'nonce-1rA2345' \*.example79.com:8000 'nonce-2rB3333'")