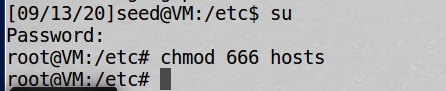
Cross-Site Scripting (XSS) Attack Lab

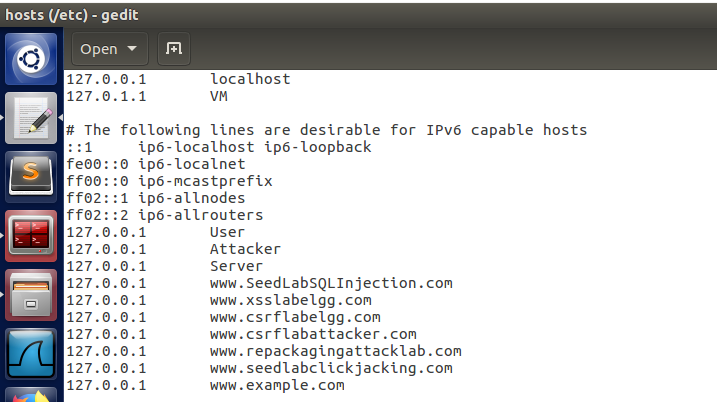
(Web Application: Elgg)

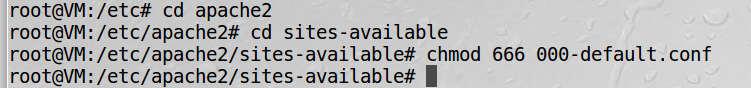
57118138 李嘉怡

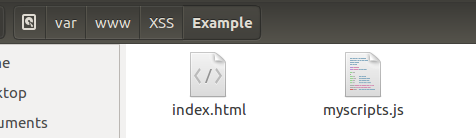
Task 1: Posting a Malicious Message to Display an Alert Window

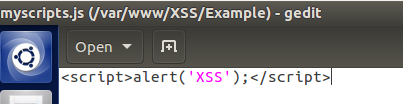
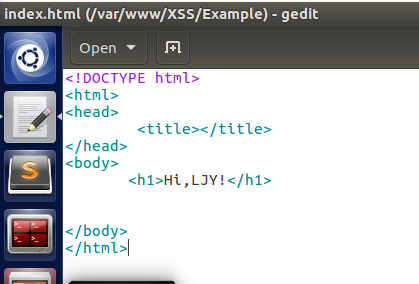
1. 配置DNS，即修改/etc/hosts文件，在最后一行添加网址www.example.com。在此之前需要修改这个文件的读写权限，使之可修改。



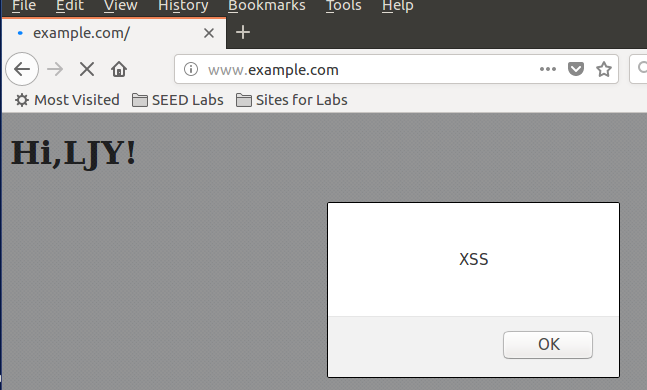
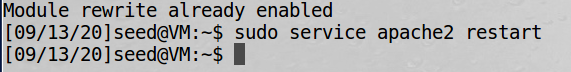


1. 配置Apache server，在/etc/apache2/site-available/000-default中添加代码，为www.example.com指定一个根目录。
2. 在这个根目录下添加index.html和myscripts.js文件，index.html用于测试www.example.com能否正常访问，myscripts.js用于显示警告窗口。

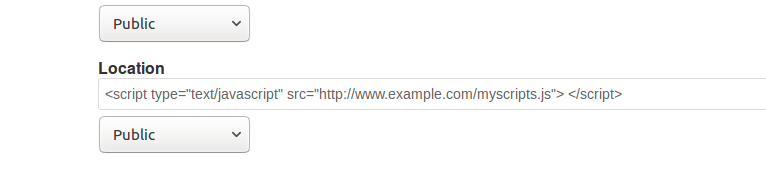




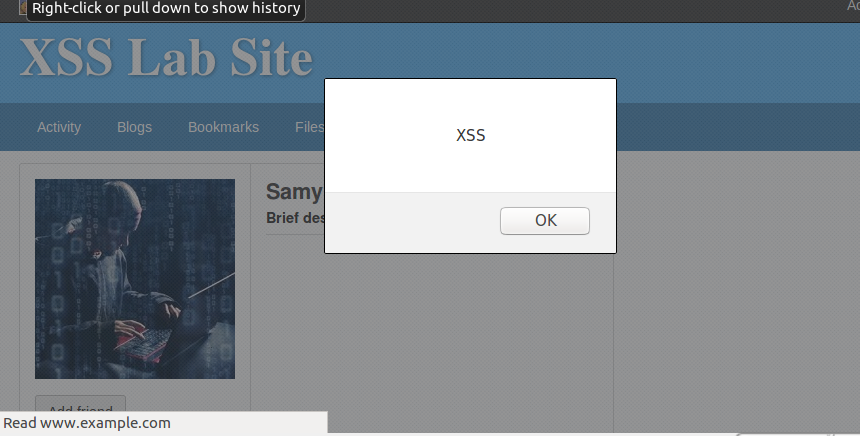
4.修改apache2后，测试www.example.com能否正常访问：



5.访问Elgg，以samy身份登入并修改samy的profile，在Location中添加如下代码：

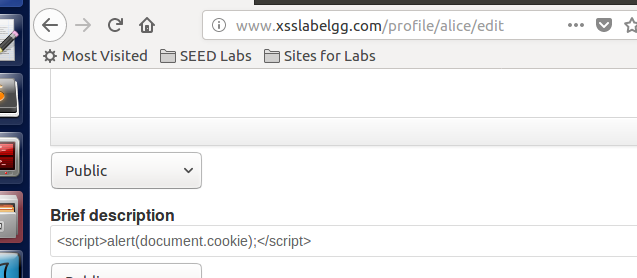


1. 访问Elgg,以alice身份登入并访问samy的profile。

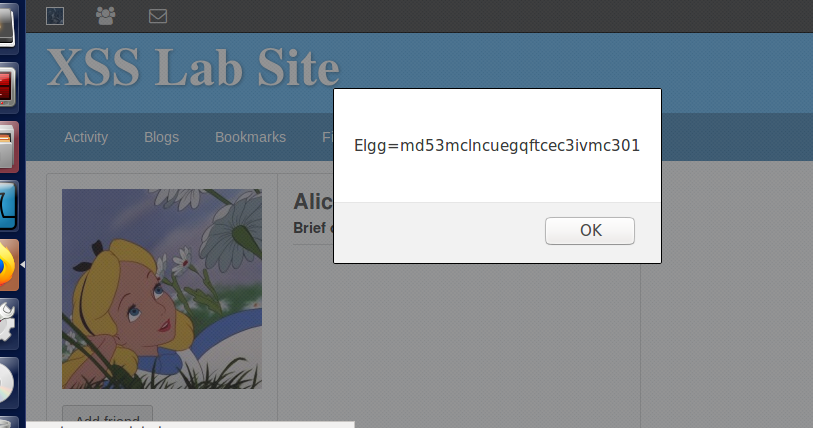


Task 2: Posting a Malicious Message to Display Cookies

1. 访问Elgg,并修改Alice的profile，添加如下代码:

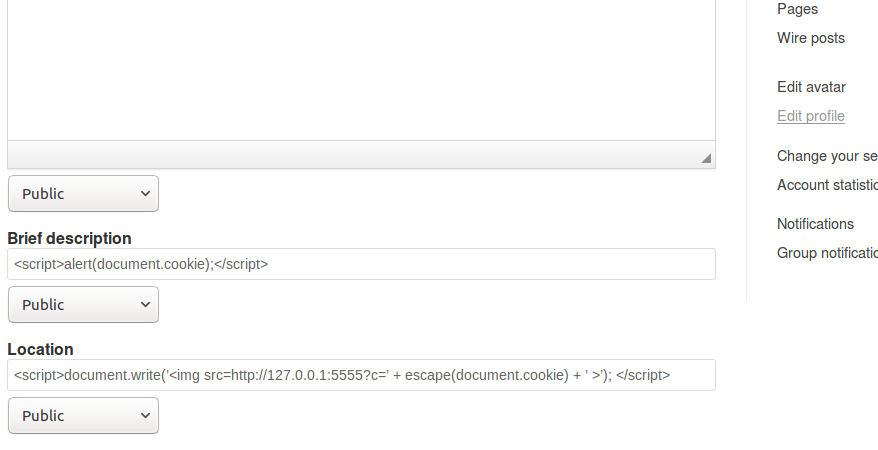


2.访问Elgg以samy身份登入，并访问alice的profile。

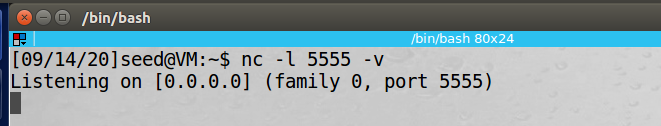


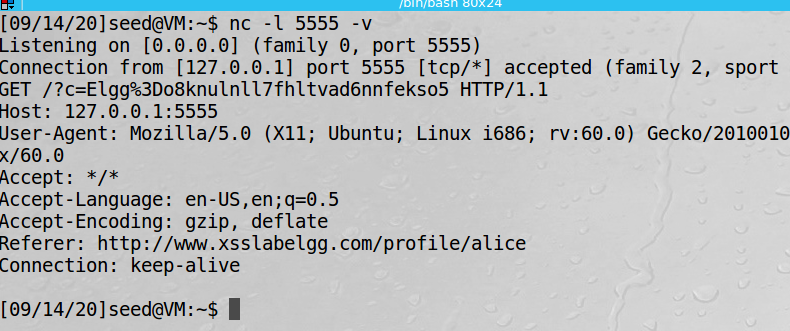
Task 3: Stealing Cookies from the Victim’s Machine

1. 访问Elgg,以alice身份登入并修改她的profile,在Location项中添加如下代码：



2.打开终端，输入下面的命令来监听端口5555:



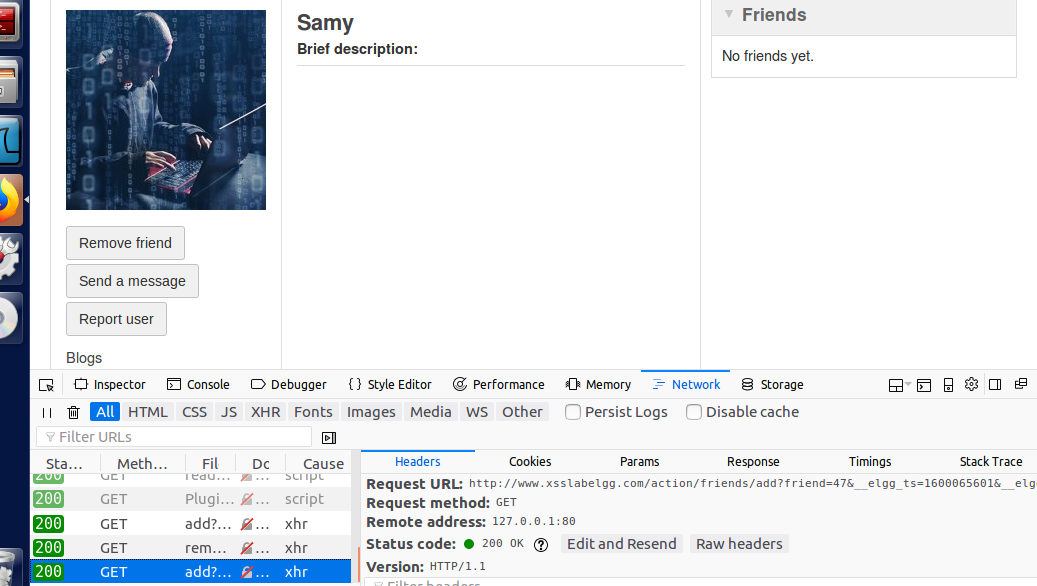
1. 访问Elgg，以samy身份登入并访问alice的profile。此时会发现cookie已经被捕获并输出到了命令行。

Task 4: Becoming the Victim’s Friend

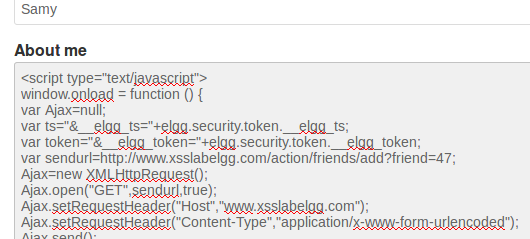
1. 以Charlie 登录，查看添加samy为好友的url：

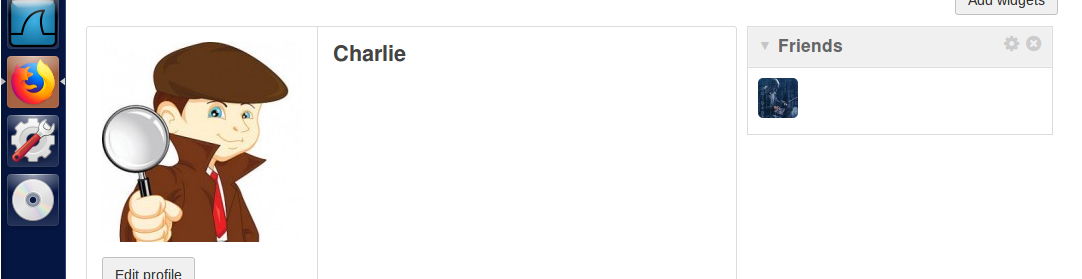
<http://www.xsslabelgg.com>/action/friends/add?friend=47&\_\_elgg\_ts=1600065601

&\_\_elgg\_token=dMWSdrpjhhRTdnFqcBL8cQ&\_\_elgg\_ts=1600065601&\_\_elgg\_token=dMWSdrpjhhRTdnFqcBL8cQ



1. 以samy登录，将上一步得到的url放到代码中，将代码放在edit profile的about me里面。



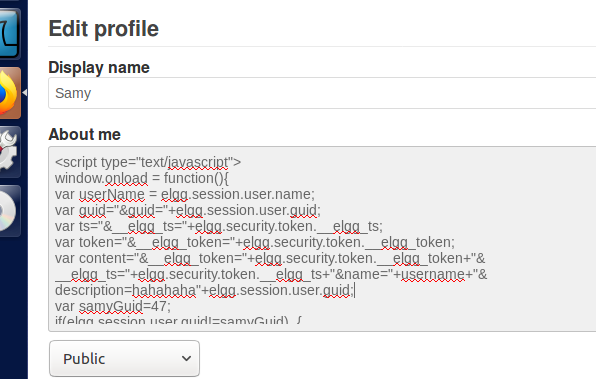
1. 以charlie登录，访问samy的profile，之后发现samy添加为了charlie的好友。

Question1：为了获取\_\_elgg\_ts和\_\_elgg\_token，用到新的url中。

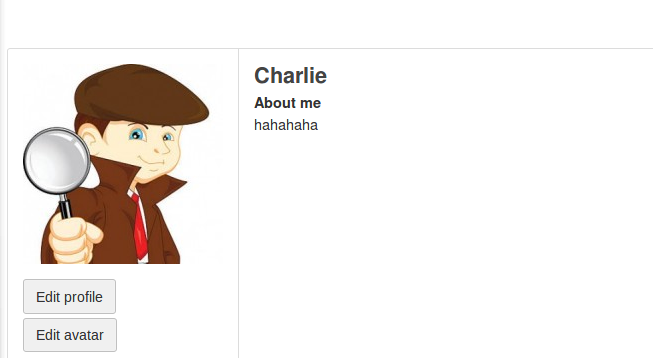
Question2：也可以。

Task 5: Modifying the Victim’s Profile

1.以samy登录，在代码中写入要修改的profile内容。



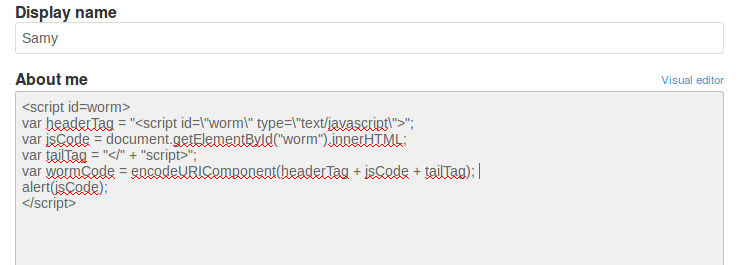
2.以charlie登录，访问samy的profile，之后发现profile被修改了。



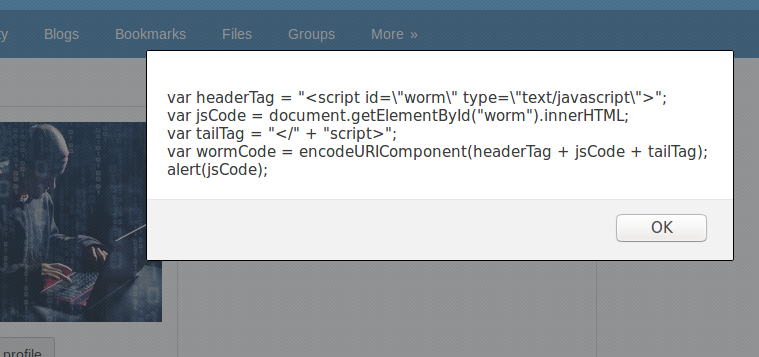
Question3：防止samy自己登录自己账户的时候profile自动被修改。

Task 6: Writing a Self-Propagating XSS Worm

1. 在samy的profile里加入使用DOM api的蠕虫代码，保存。



这段代码获得了自身的一个副本，并在一个警告窗口中显示。

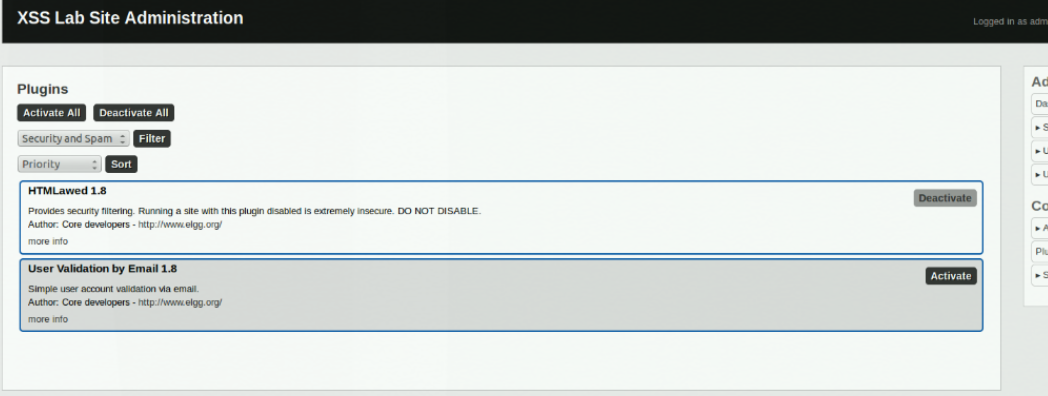


2.登录charlie的账号，访问samy的profile，发现samy被添加为了charlie的好友，而且chalie的profile中也被复制了这段代码，于是Charlie也成为了攻击者。

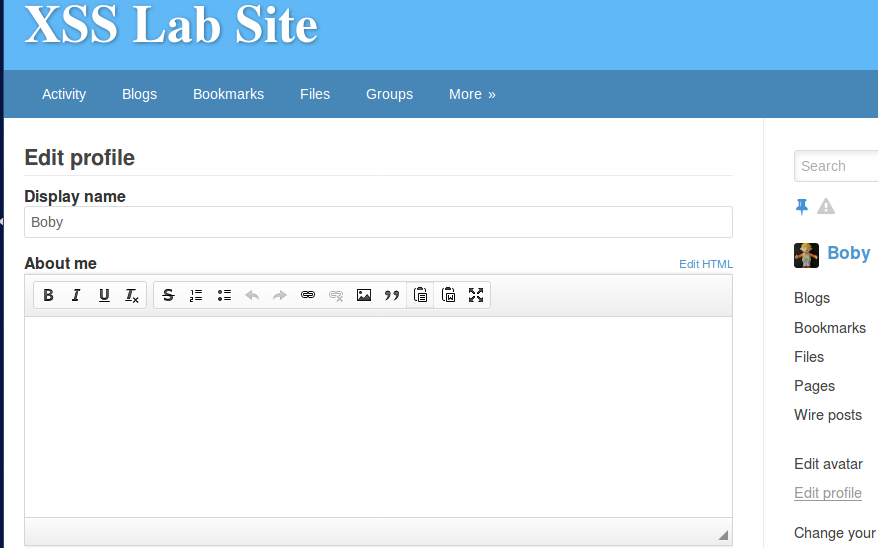


Task 7: Countermeasures

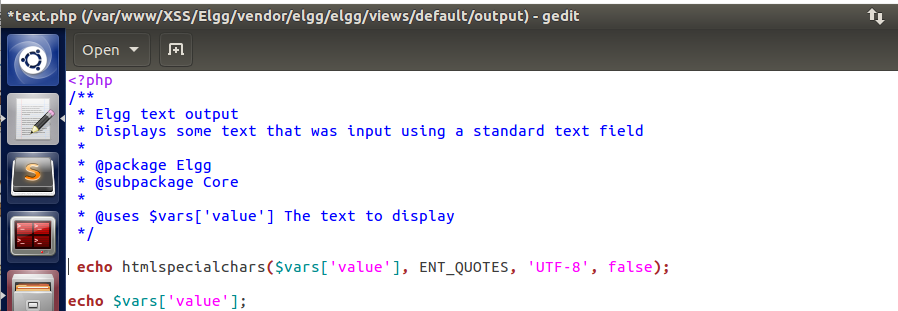
1. 以admin身份登录elgg，点击Plugins,激活HTMLawed 1.8。

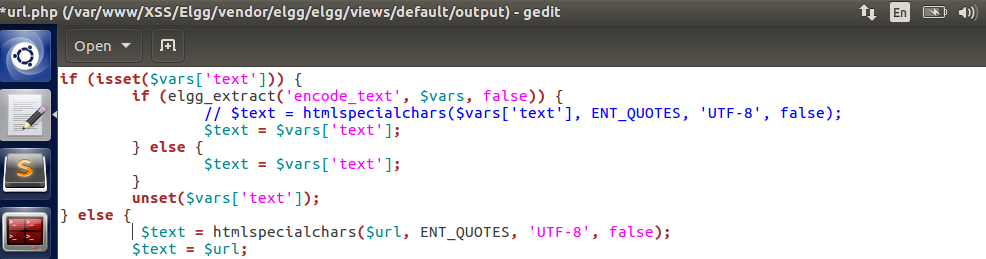


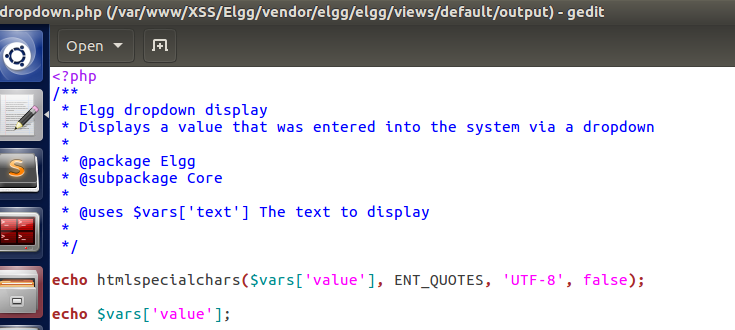
2.以boby身份登入Elgg,访问了boby和alice的profile，但是没有任何的修改，说明没有发生攻击。这种方式是把用户输入的script标签转换为a标签，避免了被动触发js代码造成的感染。

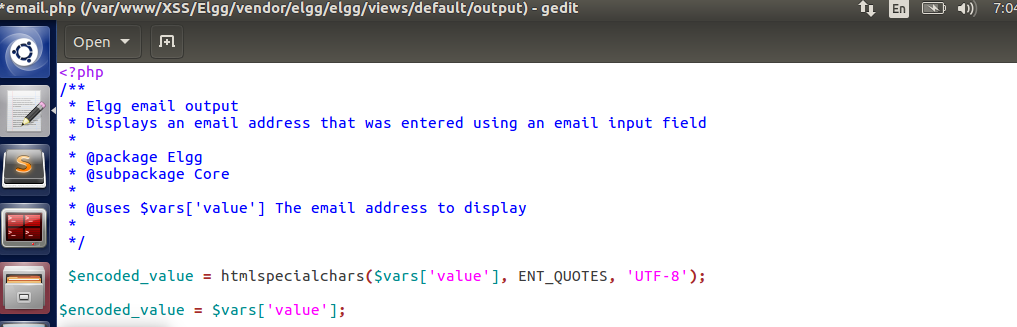


3.解除text.php，url.php，download.php，email.php等文件中的htmlspecialchars方法调用的注释。









4.以boby(xss worm的制造者)身份登入Elgg，访问他的profile,发现他的profile中Location项中的js代码消失了。再访问charlie(受害者)的profile, 蠕虫代码被显示了出来。这是将方括号转义为html的字符渲染形式，从而避免了被当作一个脚本去运行，阻止了攻击。

