**[Introduction](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md" \l "introduction)**

The challenges cover each OWASP topic given below:

[1) Injection](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-1-injection)

[2) Broken Authentication](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-2-broken-authentication)

[3) Sensitive Data Exposure](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-3-sensitive-data-exposure)

[4) XML External Entity](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-4-xml-external-entity)

[5) Broken Access Control](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-5-broken-access-control)

[6) Security Misconfiguration](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-6-security-misconfiguration)

[7) Cross-site Scripting](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-7-cross-site-scripting)

[8) Insecure Deserialization](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-8-insecure-deserialization)

[9) Components with Known Vulnerabilities](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-9-components-with-known-vulnerabilities)

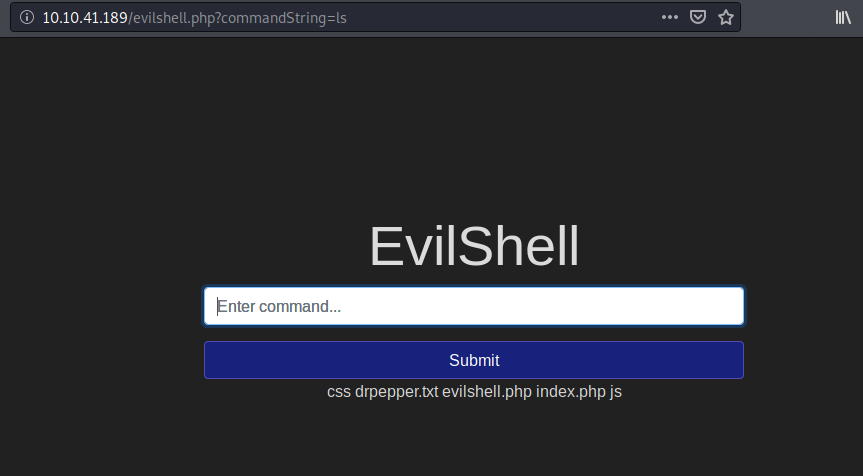
[10) Insufficent Logging & Monitoring](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/OWASP.md#day-10-insufficient-logging--monitoring)

[**Topic 1: Injection**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-1-injection)

Deployed machine IP: 10.10.41.189

1. What strange text file is in the website root directory?

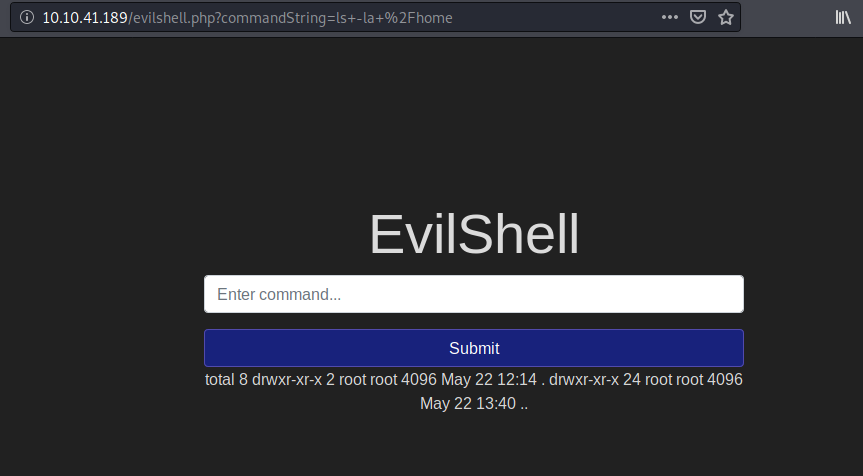
Command: ls

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/ls.png?raw=true)

drpepper.txt

1. How many non-root/non-service/non-daemon users are there?

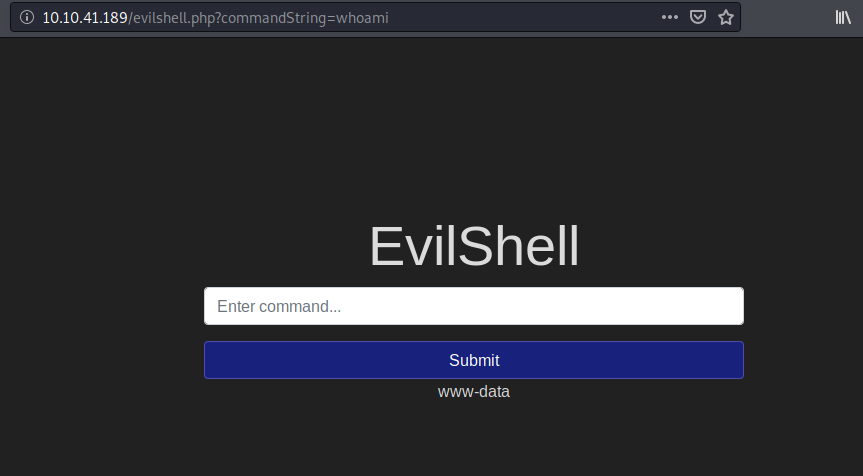
I check the number of users in the */home* directory.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/home.png?raw=true)

0

1. What user is this app running as?

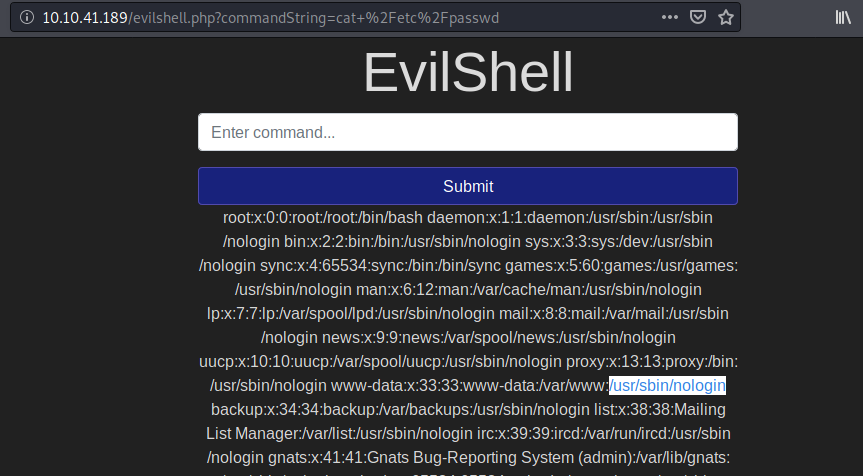
Use the whoami command.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/whoami.png?raw=true)

www-data

1. What is the user's shell set as?

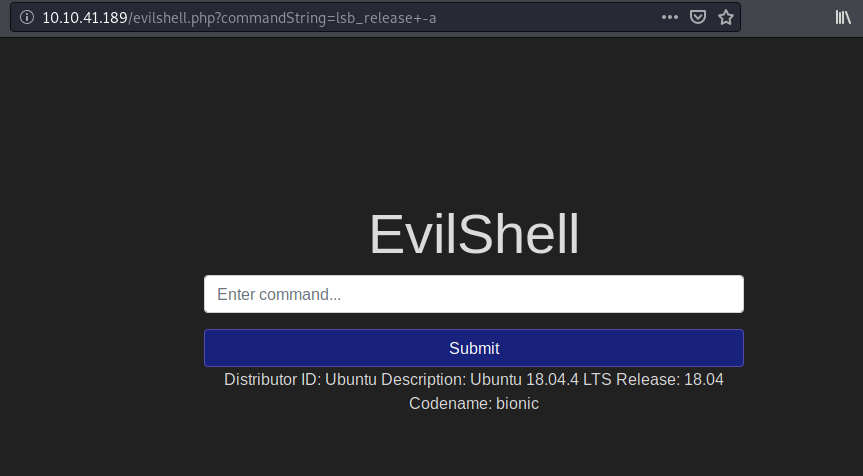
Check the */etc/passwd* file.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/shell.png?raw=true)

/usr/sbin/nologin

1. What version of Ubuntu is running?

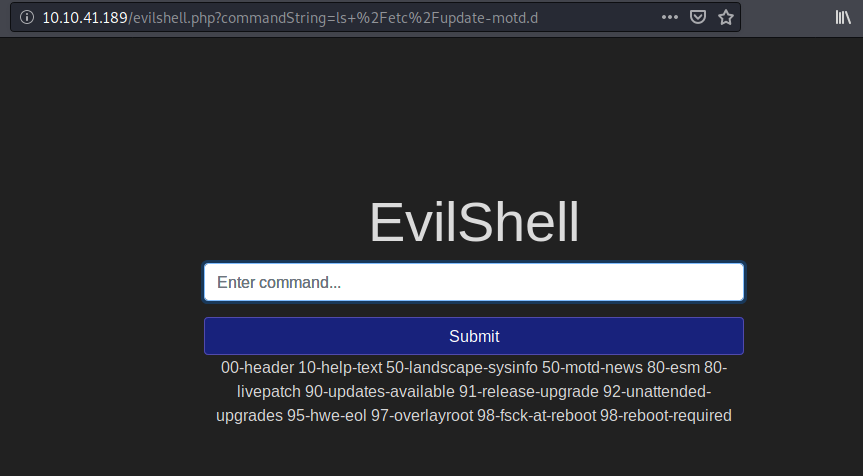
Command: lsb\_release -a

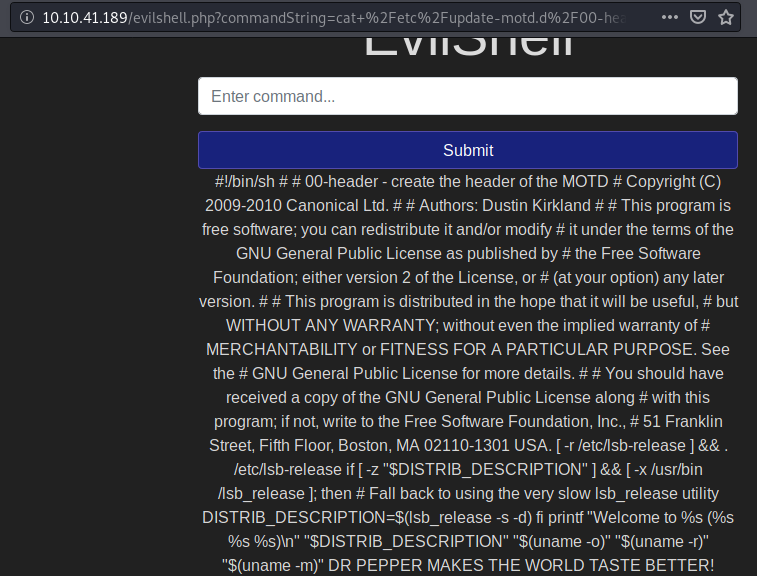
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/release.png?raw=true)

18.04.4

1. Print out the MOTD. What favorite beverage is shown?

I had to search Google for this one. MOTD stands for Message Of The Day, and is a directory (/etc/update-motd.d). More on it [here](http://manpages.ubuntu.com/manpages/eoan/en/man5/update-motd.5.html)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/motd_1.png?raw=true)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day01/motd_2.png?raw=true)

DR PEPPER

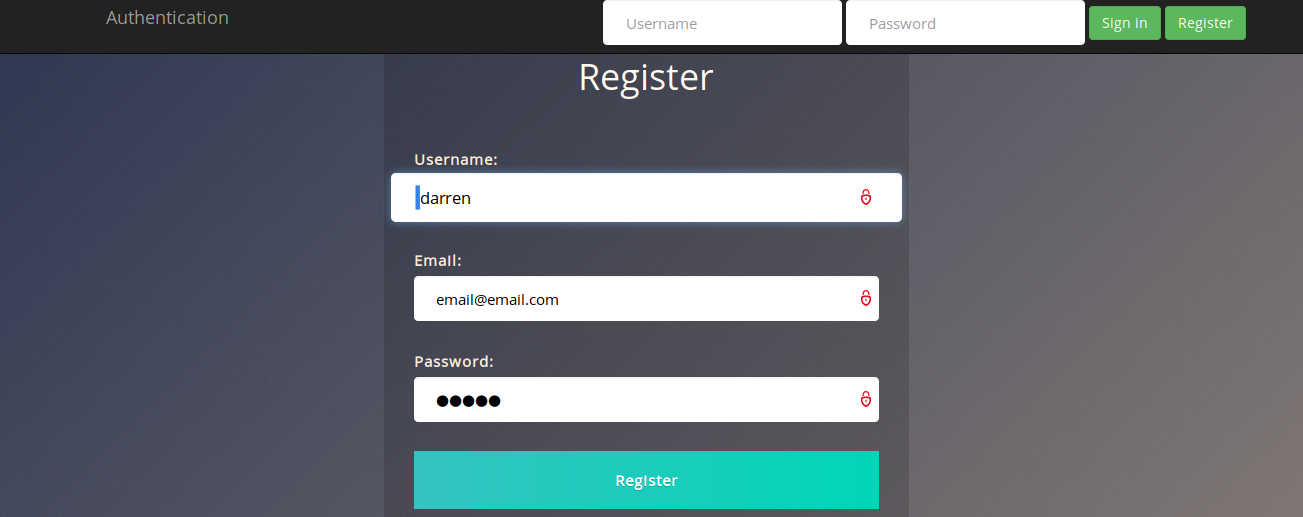
[**Topic 2: Broken Authentication**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-2-broken-authentication)

Machine IP: 10.10.50.243

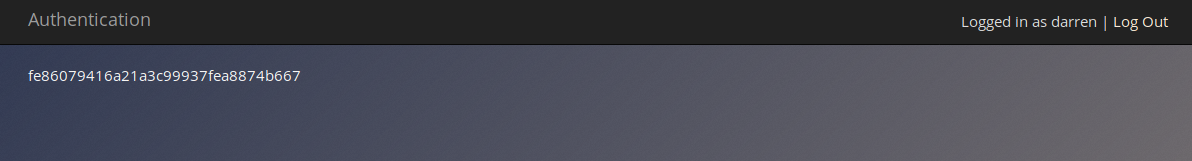
1. What is the flag that you found in darren's account?

For today's task, all I had to do was follow the steps the task describes.

First, I made an account as darren, with a slight modification to the username.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day02/darren.png?raw=true)

Notice the space in front of "darren". After registering as darren-not-darren, I logged in using my registration credentials and voila, we have the flag found in darren's account.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day02/flag.png?raw=true)

fe86079416a21a3c99937fea8874b667

1. Now try to do the same trick and see if you can login as arthur.

I did the same thing for arthur as for darren.

No answer needed

1. What is the flag that you found in arthur's account?

d9ac0f7db4fda460ac3edeb75d75e16e

[**Topic 3: Sensitive Data Exposure**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-3-sensitive-data-exposure)

Machine IP: 10.10.156.59

1. Have a look around the webapp. The developer has left themselves a note indicating that there is sensitive data in a specific directory.

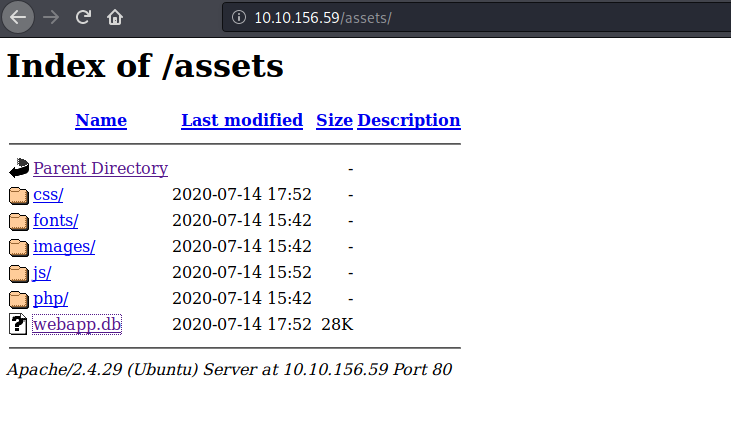
What is the name of the mentioned directory?

Going to the source code of the login page, I found the folder */assets*

/assets

1. Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?

I went to the directory and this is what I found:

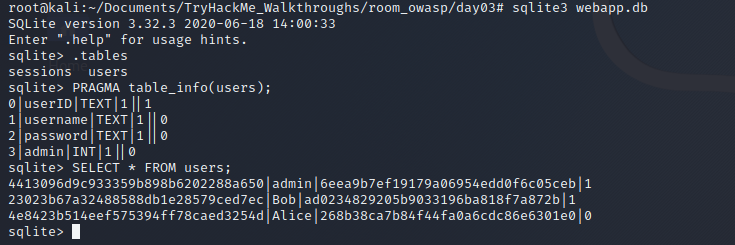
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day03/assets.png?raw=true)

Seems *webapp.db* is the file of interest. I downloaded it to my machine.

webapp.db

1. Use the supporting material to access the sensitive data. What is the password hash of the admin user?

I just followed the supporting material to get the password hash of the admin user.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day03/commands.png?raw=true)

6eea9b7ef19179a06954edd0f6c05ceb

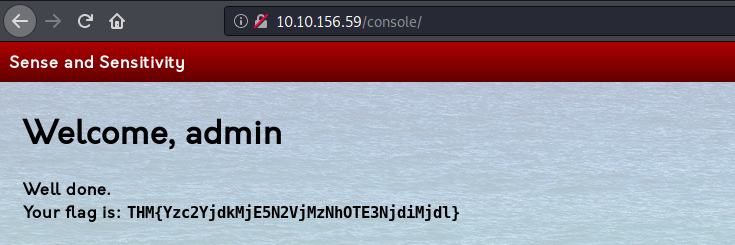
1. Crack the hash. What is the admin's plaintext password?

I used hashcat for this one. Command: hashcat -m 0 "6eea9b7ef19179a06954edd0f6c05ceb" /usr/share/wordlists/rockyou.txt

qwertyuiop

1. Login as the admin. What is the flag?

Just log into the deployed machine through the login page (Username: admin; Password: qwertyuiop)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day03/flag.png?raw=true)

THM{Yzc2YjdkMjE5N2VjMzNhOTE3NjdiMjdl}

[**Topic 4: XML External Entity**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-4-xml-external-entity)

Machine IP: 10.10.84.233

[**XML External Entity - eXtensible Markup Language**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#xml-external-entity---extensible-markup-language)

For this part, the questions can be answered by reading the explanations from the TryHackMe room.

1. Full form of XML.

extensible markup language

1. Is it compulsory to have XML prolog in XML documents?

No

1. Can we validate XML documents against a schema?

Yes

1. How can we specify XML version and encoding in XML document?

XML Prolog

[**XML External Entity - DTD**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#xml-external-entity---dtd)

1. How do you define a new ELEMENT?

!ELEMENT

1. How do you define a ROOT element?

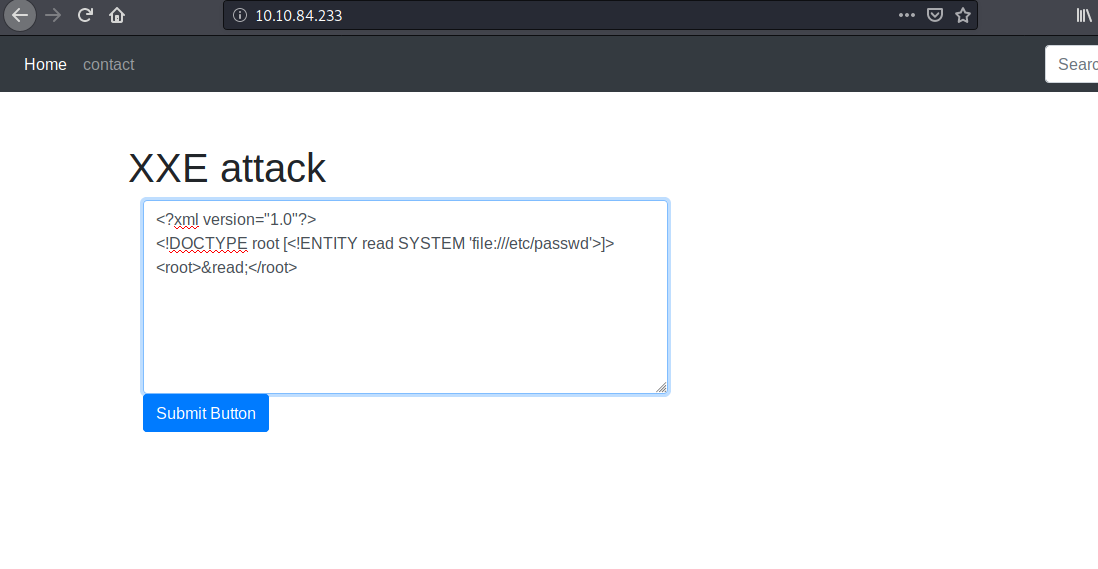
!DOCTYPE

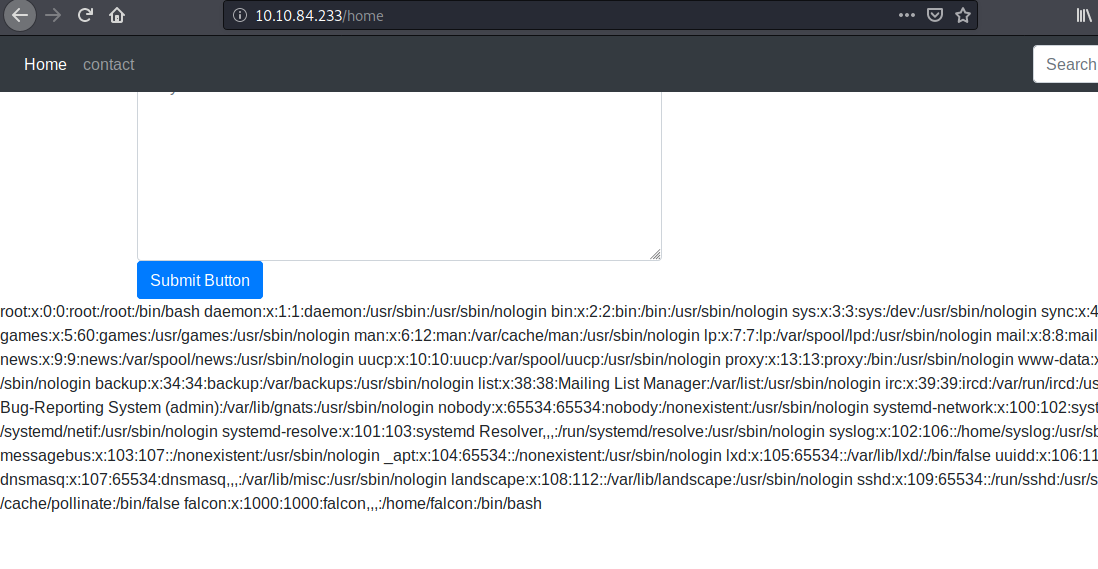
1. How do you define a new ENTITY?

!ENTITY

[**XML External Entity - XXE Payload**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#xml-external-entity---xxe-payload)

1. Try the payload mentioned in description on the website.

So, this is stright forward, just use the second payload in the text box and submit it. After submitting, the webpage should display the contents of */etc/passwd*. [](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day04/xxe_1.png?raw=true)

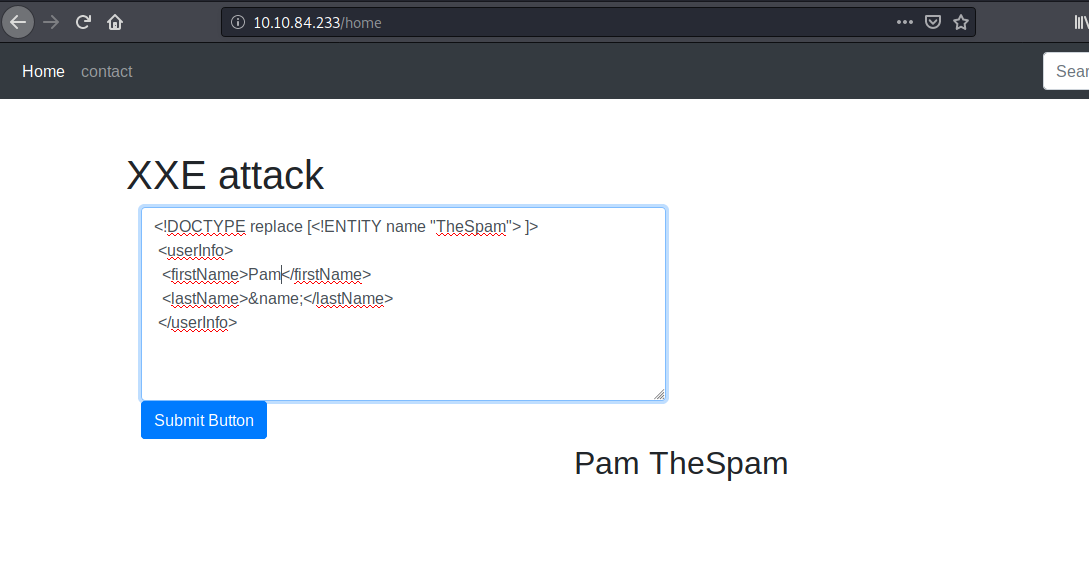
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day04/xxe_2.png?raw=true)

No answer needed

[**XML External Entity - Exploiting**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#xml-external-entity---exploiting)

1. Try to display your own name using any payload.

Just submit the first payload shown in the previous task and replace accoringly.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day04/name.png?raw=true)

No answer needed

1. See if you can read the /etc/passwd.

This part was done in the previous task.

No answer needed

1. What is the name of the user in /etc/passwd?

falcon

1. Where is falcon's SSH key located?

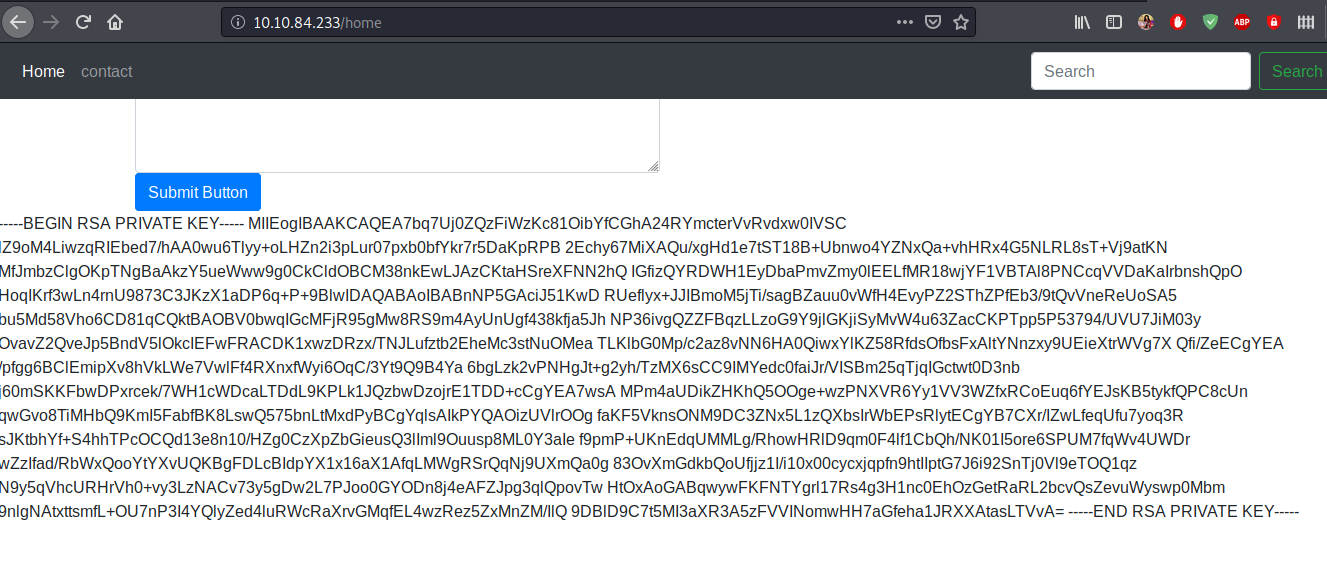
Payload:

<?xml version="1.0"?>

<!DOCTYPE root [<!ENTITY read SYSTEM 'file:///home/falcon/.ssh/id\_rsa'>]>

<root>&read;</root>

/home/falcon/.ssh/id\_rsa

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day04/ssh.png?raw=true)

1. What are the first 18 characters for falcon's private key?

MIIEogIBAAKCAQEA7bq

[**Topic 5: Broken Access Control**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-5-broken-access-control)

Machine IP: 10.10.240.167

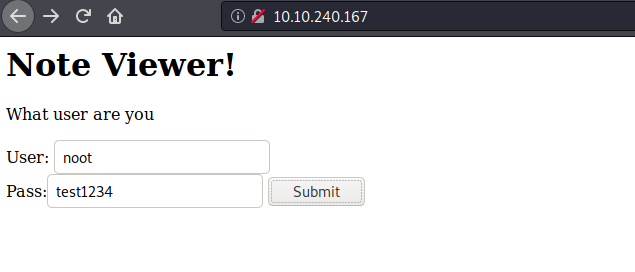
1. Read and understand how IDOR works.

IDOR = Insecure Direct Object Reference

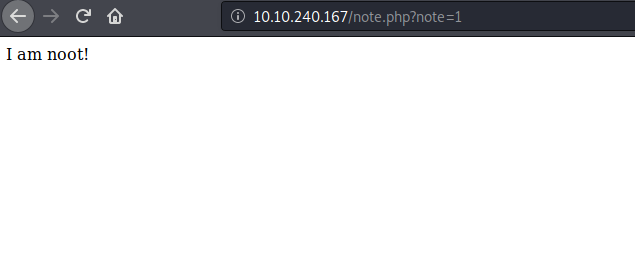
Read more about it by accessing the room.

No answer needed

1. Deploy the machine and go to [http://10.10.240.167](http://10.10.240.167/) - Login with the username being noot and the password test1234.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day05/login.png?raw=true)

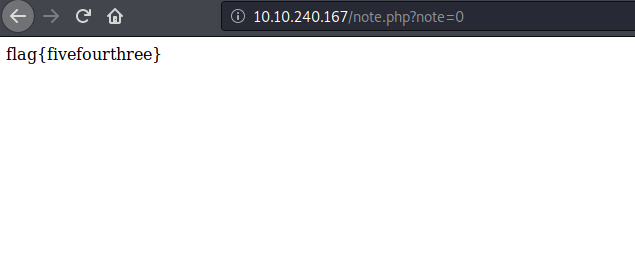
After loggin in with the username and password given to us, look at the URL. Notice the note id.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day05/logged.png?raw=true)

No answer needed

1. Look at other users notes. What is the flag?

I changed the note id from 1 to 0 and got the flag. Since the site is incorrectly configured, anyone can access any note by simply changing the note id.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day05/flag.png?raw=true)

flag{fivefourthree}

[**Topic 6: Security Misconfiguration**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-6-security-misconfiguration)

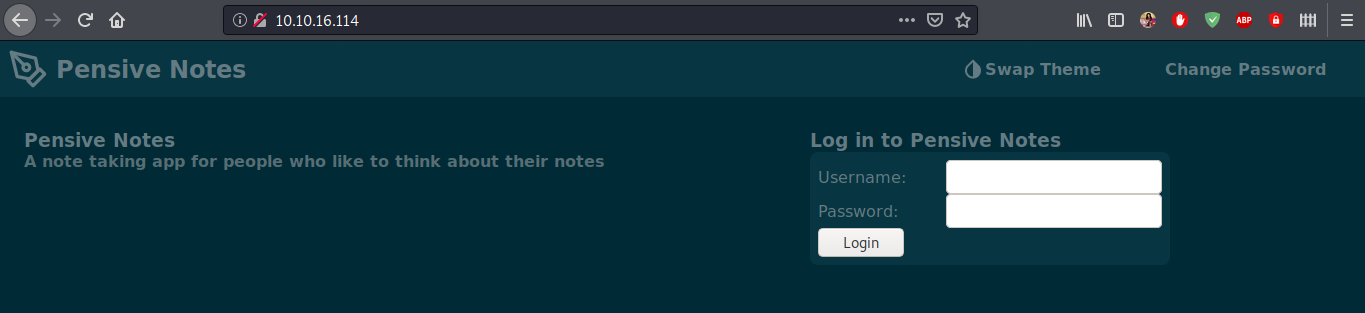
1. Deploy the VM.

Machine IP: 10.10.16.114

No answer needed

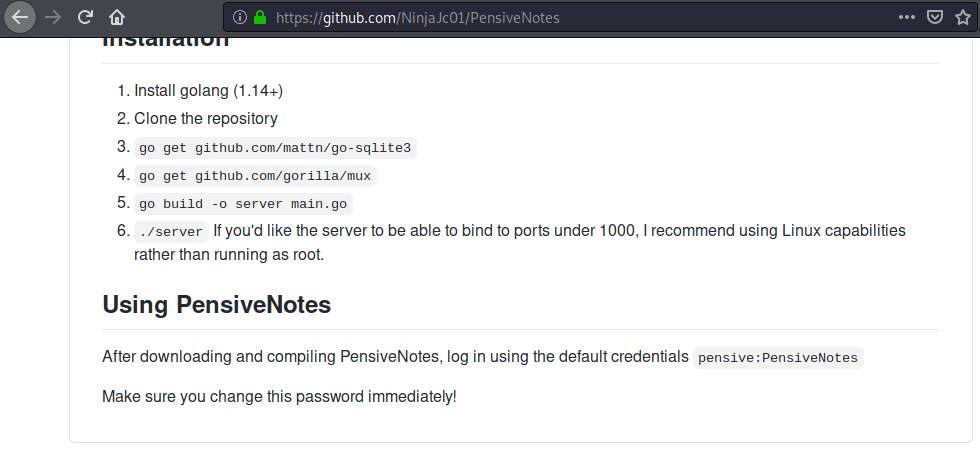
1. Hack into the webapp, and find the flag!

I accessed the page, which looks like this:

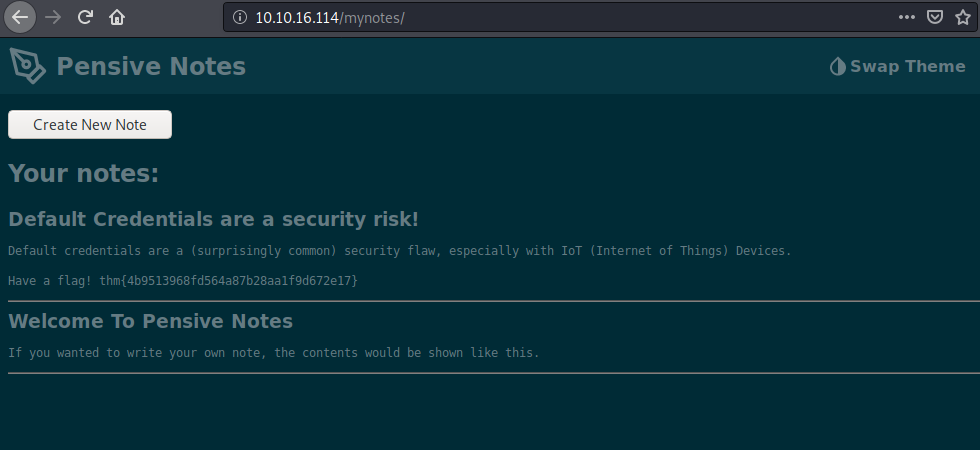
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day06/notes.png?raw=true)

I used gobuster (gobuster dir -u http://10.10.16.114/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt) to see if there were any hidden directories. It didn't find any, so I quickly went through the */js* directory to see if I could find anything there.

The next thing I did was a Google search to see if the webapp is open source. It is! The first result is the github page of the app, containing the source code of the app. The README.md file contained the default credentials.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day06/github.png?raw=true)

I used these default credentials to login to the app and obtained the flag.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day06/flag.png?raw=true)

thm{4b9513968fd564a87b28aa1f9d672e17}

[**Topic 7: Cross-site Scripting**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-7-cross-site-scripting)

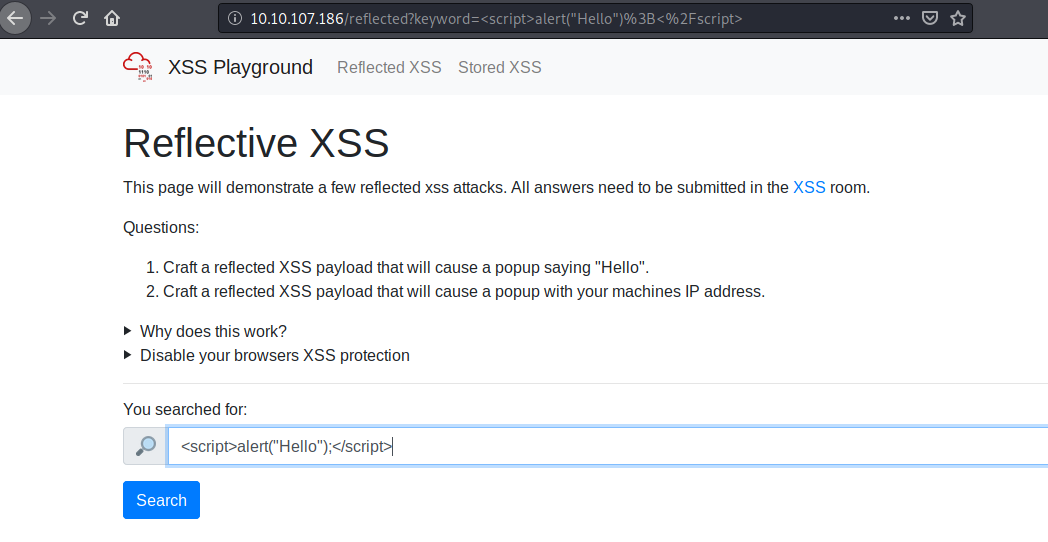
1. Deploy the VM.

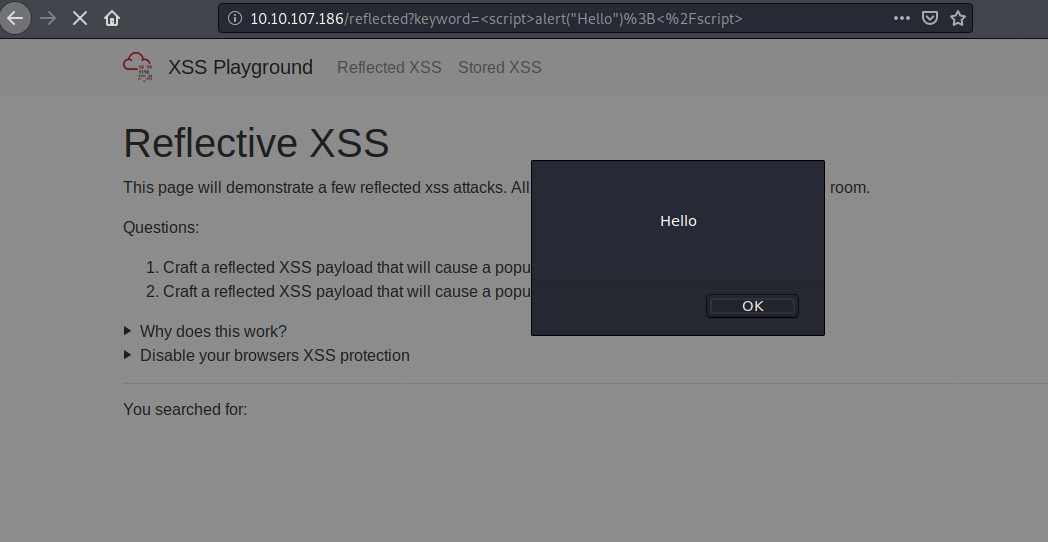
Machine IP: 10.10.107.186

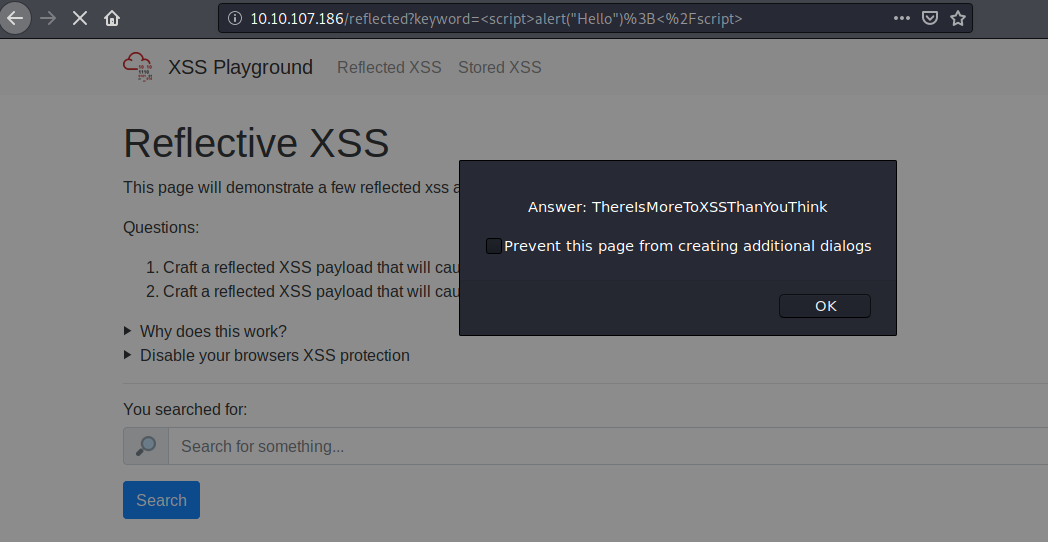
No answer needed

1. Go to <http://10.10.107.186/reflected> and craft a reflected XSS payload that will cause a popup saying "Hello".

To create a payload that will cause a popup saying "Hello", simply create a script tag containing the javascript alert function (<script>alert("Hello");</script>). Then hit *search*. The page will reload, with two popups appearing: the first says "Hello", the second the flag for this question.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective11.png?raw=true)

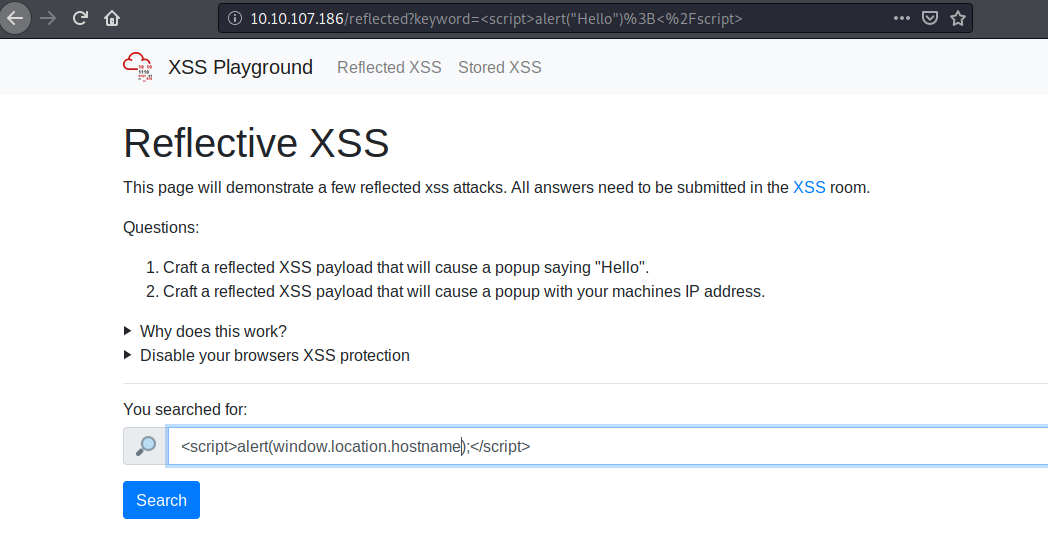
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective12.png?raw=true)

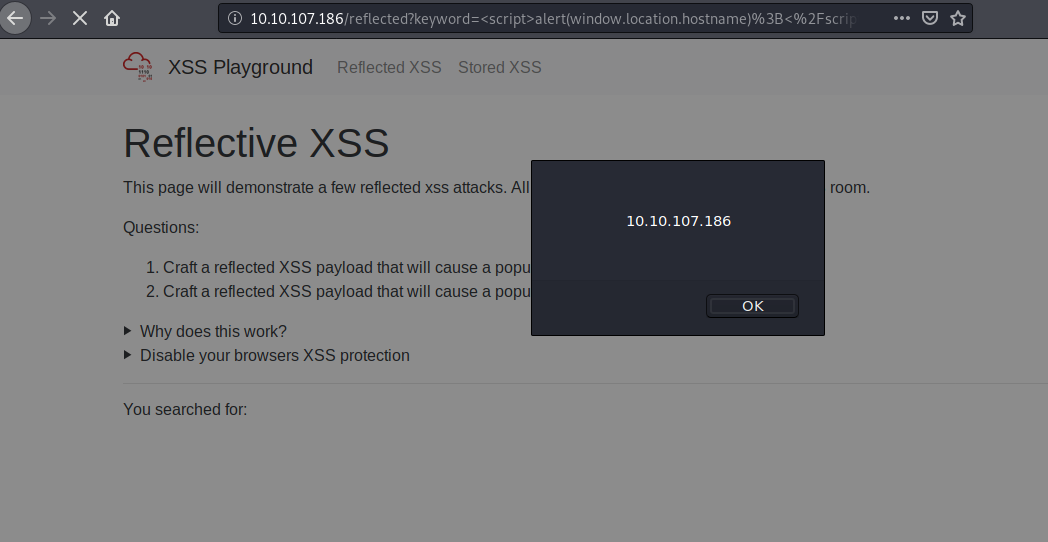
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective13.png?raw=true)

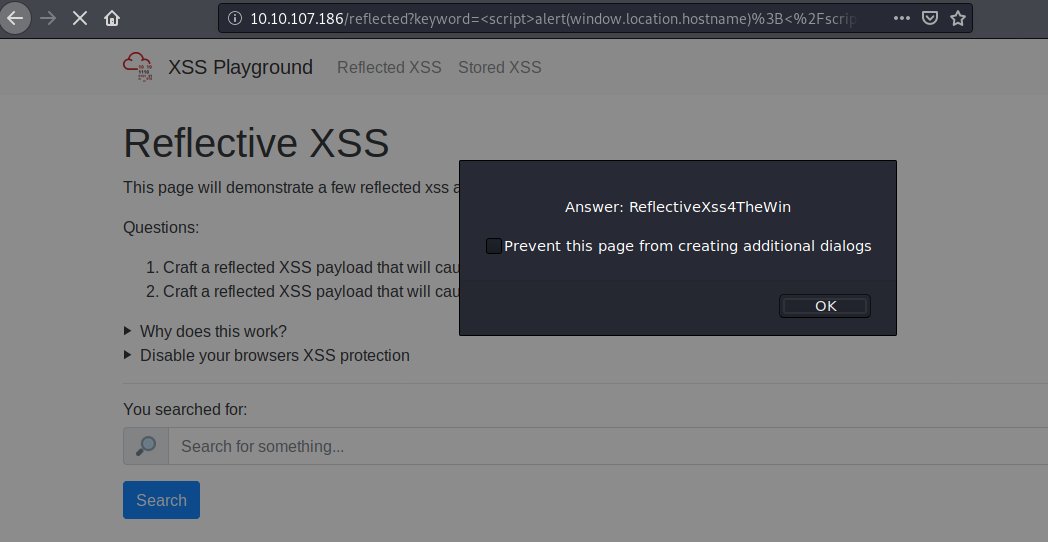
ThereIsMoreToXSSThanYouThink

1. On the same reflective page, craft a reflected XSS payload that will cause a popup with your machines IP address.

This one is solved similary to the previous one, except, instead of using the string "Hello" as an argument for the *alert* function, I used *window.location.hostname*. Just like in the previous question, after hitting *search*, the page reloads and displays 2 popups: first with the deployed machine's IP, the second with the flag for this question.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective21.png?raw=true)

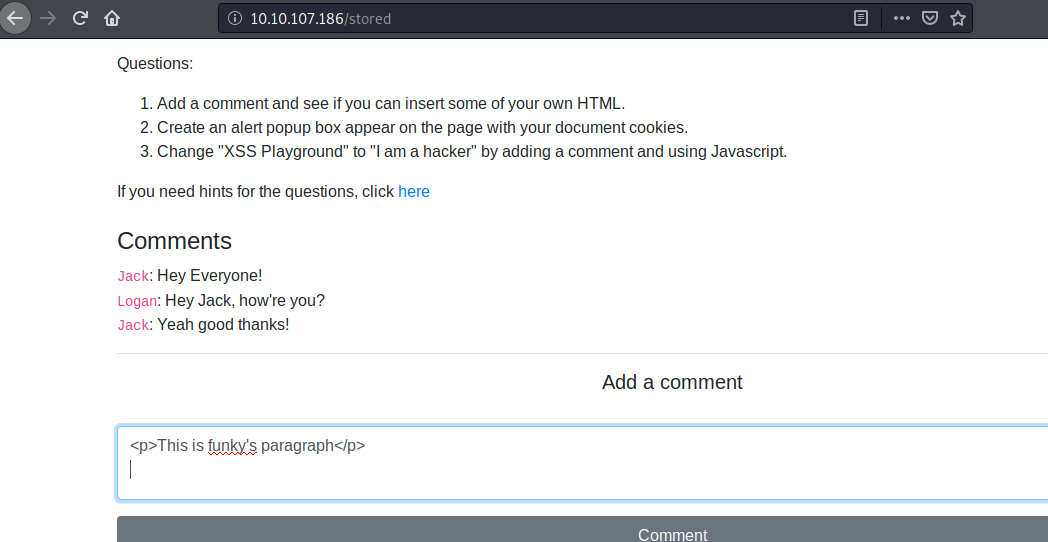
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective22.png?raw=true)

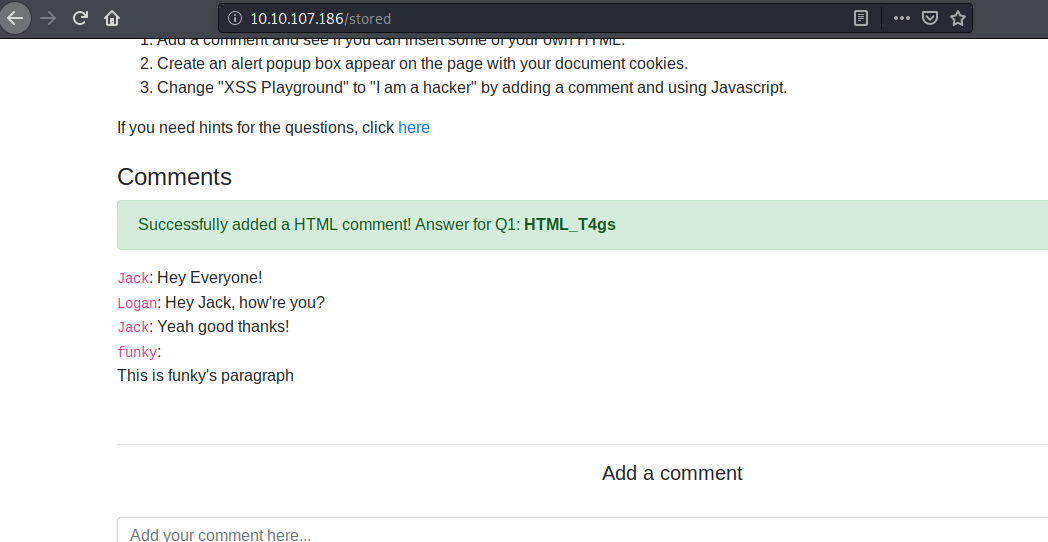
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/reflective23.png?raw=true)

ReflectiveXss4TheWin

1. Now navigate to <http://10.10.107.186/stored> and make an account. Then add a comment and see if you can insert some of your own HTML.

I made and account, after which I inserted some HTML code (a paragraph). After submitting my comment, which appeared as a paragraph (since I used the p tag), alongside the flag.

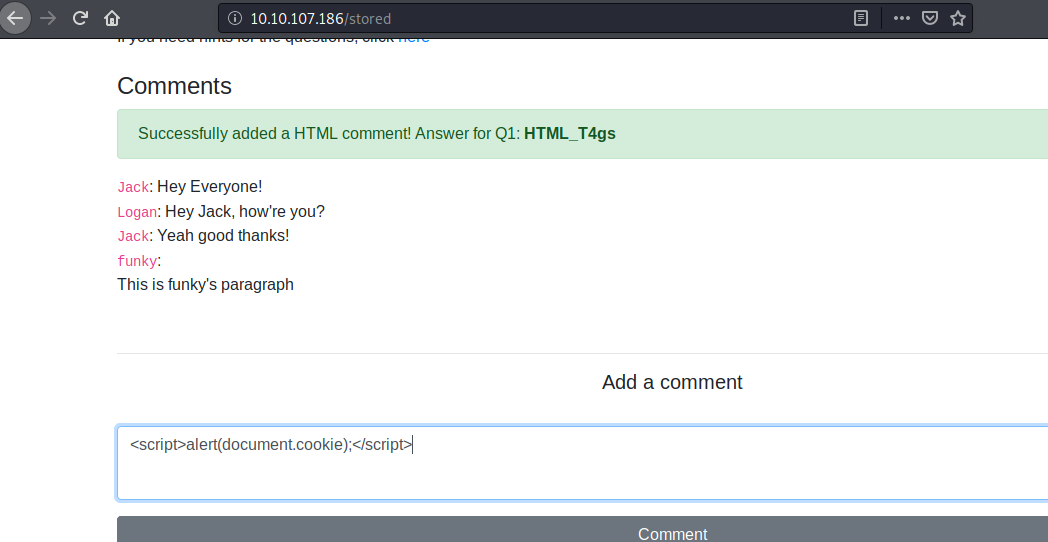
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored11.png?raw=true)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored12.png?raw=true)

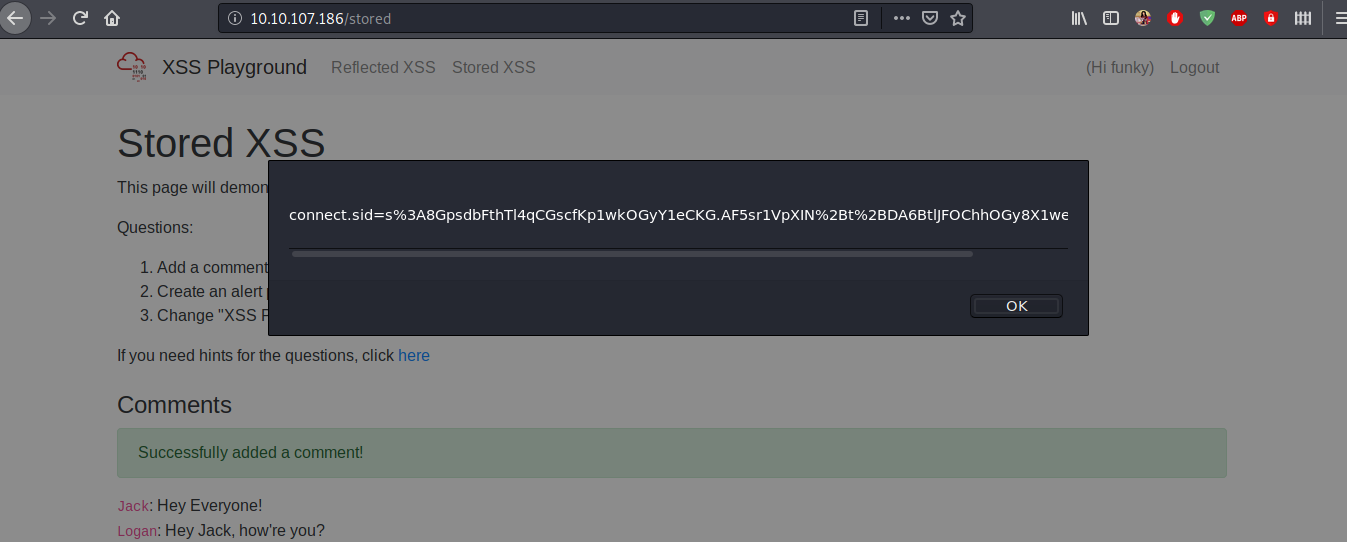
HTML\_T4gs

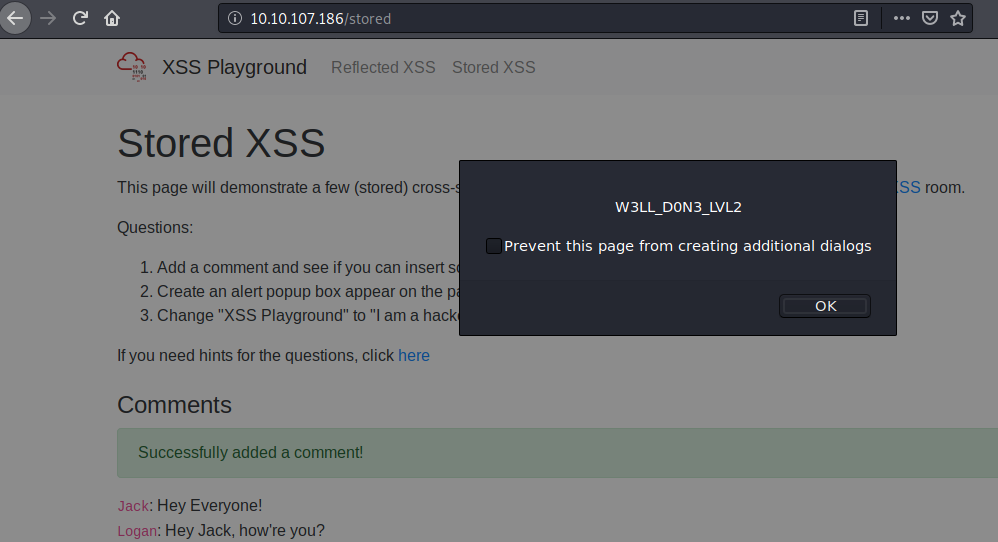
1. On the same page, create an alert popup box appear on the page with your document cookies.

Just like in questions 2 and 3, submit the alert function with document.cookie as its argument in a script tag.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored21.png?raw=true)

Upon submitting the comment, the page reloads, the document's cookie is displayed in a popup, after which the flag for this question is displayed in a second popup.

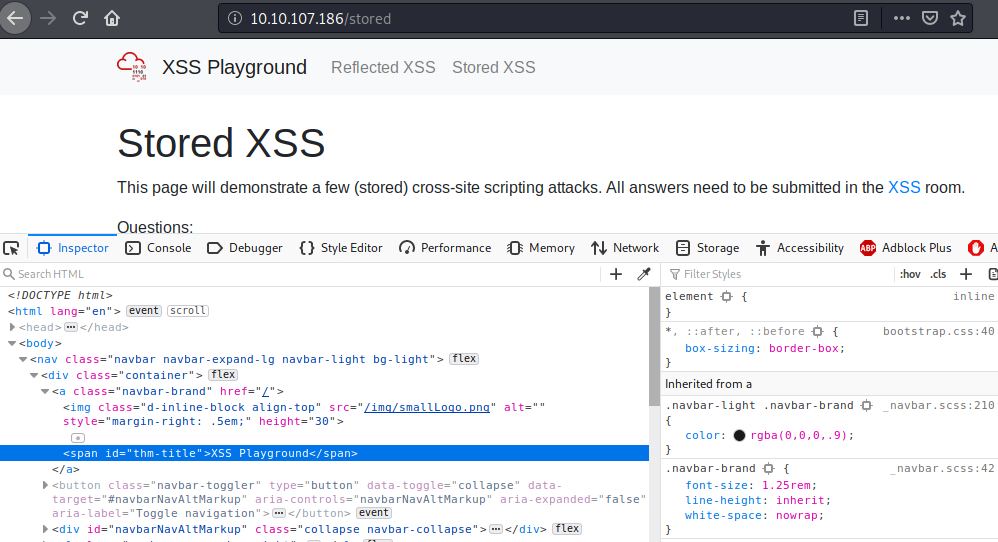
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored22.png?raw=true)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored23.png?raw=true)

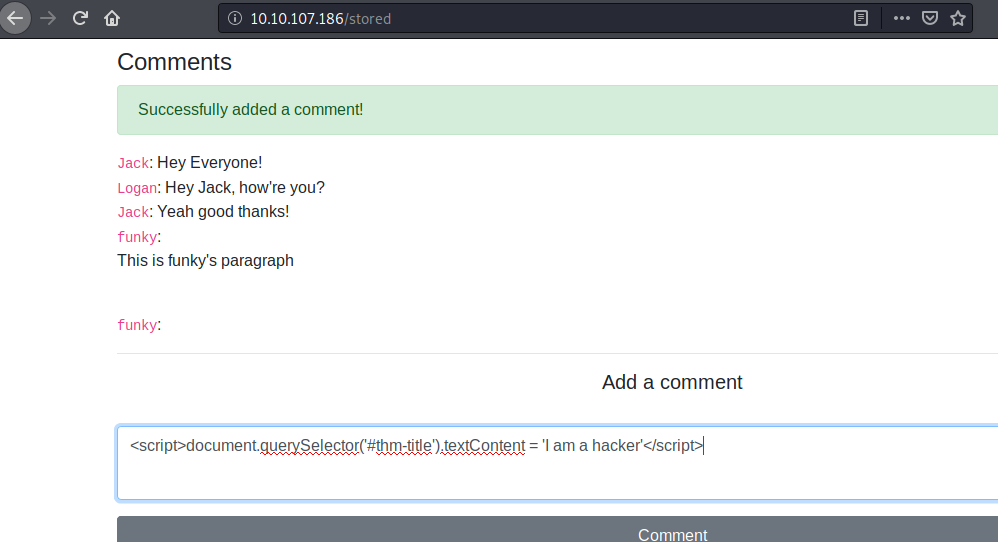
W3LL\_D0N3\_LVL2

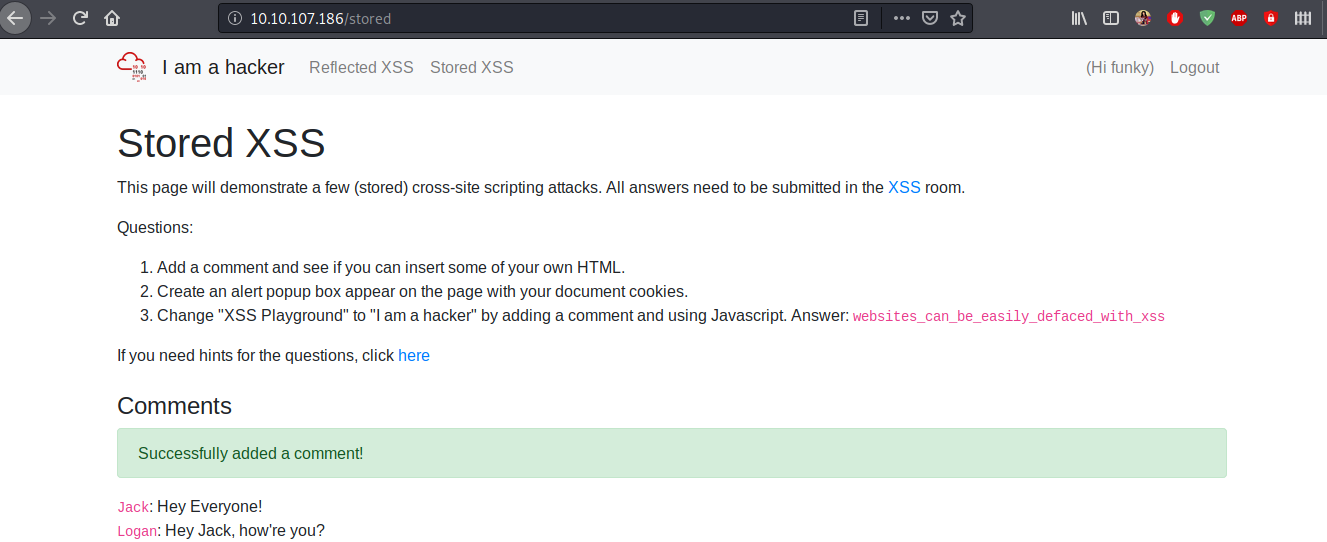
1. Change "XSS Playground" to "I am a hacker" by adding a comment and using Javascript.

For this question, the HTML code needed to be manipulated. I check the id of the **XSS Playground** element of the page.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored31.png?raw=true)

Then, using javascript, I replaced the text content of the element with the id *#thm-title* with the required text. After submitting the comment, observed the text has changed from **XSS Playground** to **I am a hacker**.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored32.png?raw=true)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day07/stored33.png?raw=true)

websites\_can\_be\_easily\_defaced\_with\_xss

[**Topic 8: Insecure Deserialization**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-8-insecure-deserialization)

*Insecure Deserialization is a vulnerability which occurs when untrusted data is used to abuse the logic of an application" (Acunetix., 2017)*

Machine IP: 10.10.61.119

1. Who developed the Tomcat application?

The Apache Software Foundation

1. What type of attack that crashes services can be performed with insecure deserialization?

Denial of Service

[**Insecure Deserialization - Objects**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#insecure-deserialization---objects)

1. Select the correct term of the following statement:

*if a dog was sleeping, would this be:*

A) A State

B) A Behaviour

a behaviour

[**Insecure Deserialization - Deserialization**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#insecure-deserialization---deserialization)

1. What is the name of the base-2 formatting that data is sent across a network as?

binary

[**Insecure Deserialization - Cookies**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#insecure-deserialization---cookies)

1. If a cookie had the path of *webapp.com/login* , what would the URL that the user has to visit be?

webapp.com/login

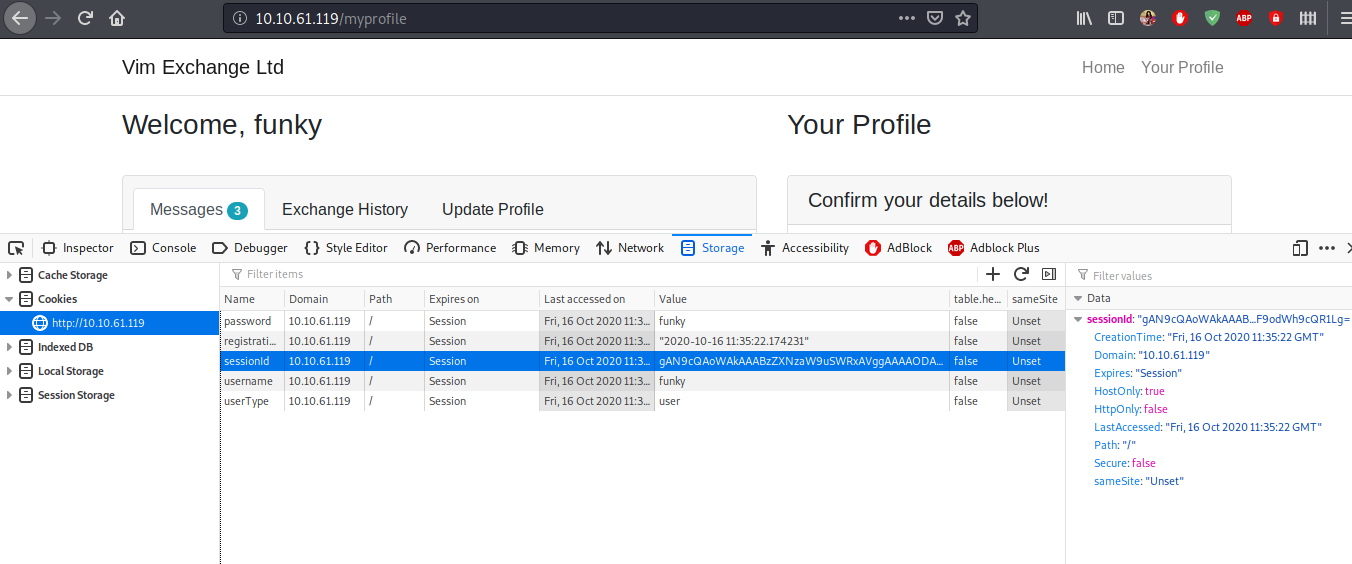
1. What is the acronym for the web technology that Secure cookies work over?

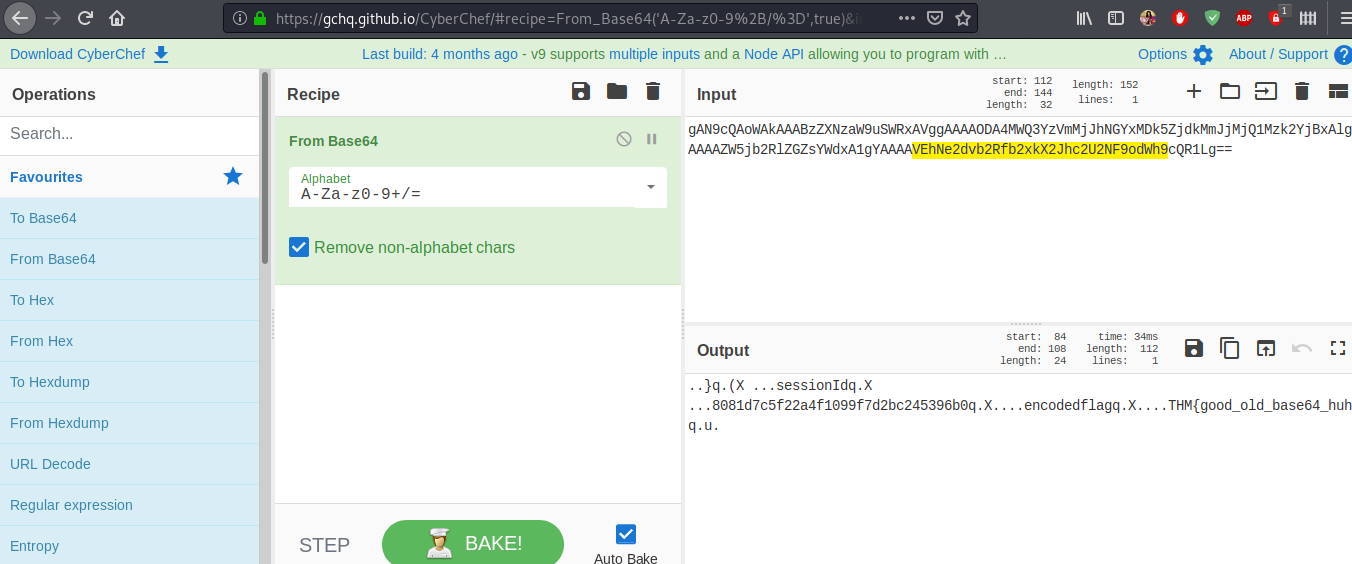
HTTPS

[**Insecure Deserialization - Cookies Practical**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#insecure-deserialization---cookies-practical)

1. 1st flag (cookie value)

In the *Inspect Element* option, check the Cookie values. I decoded the sessionId value using [CyberChef](https://gchq.github.io/CyberChef/) and obtained the flag.

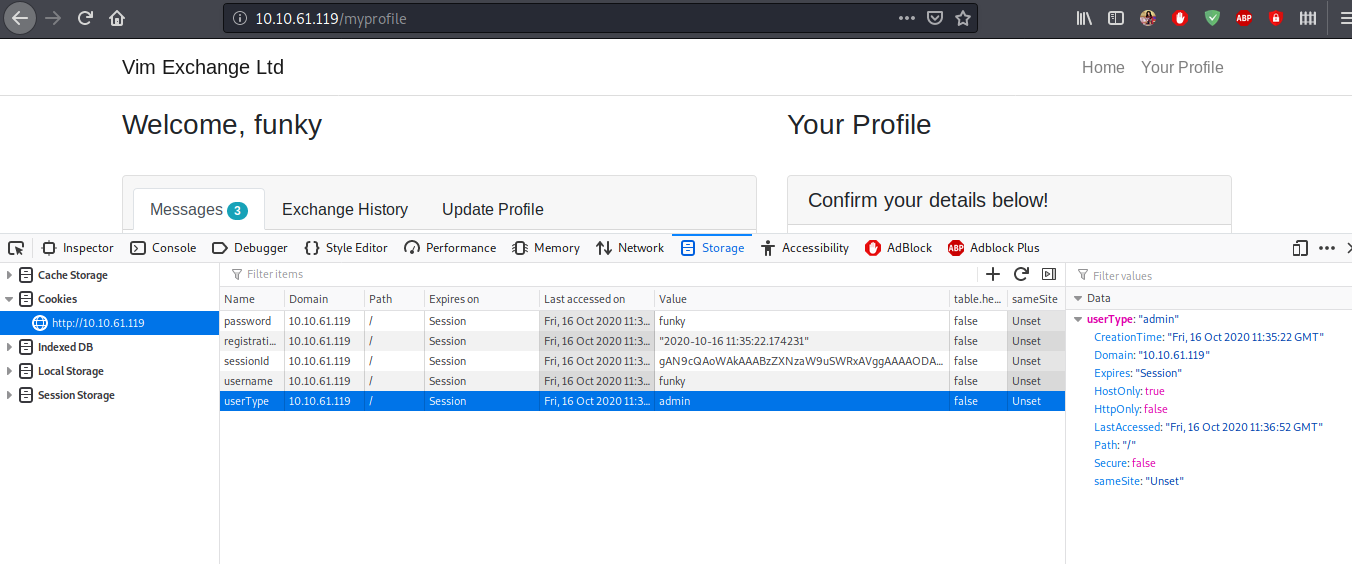
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day08/cookie1.png?raw=true)

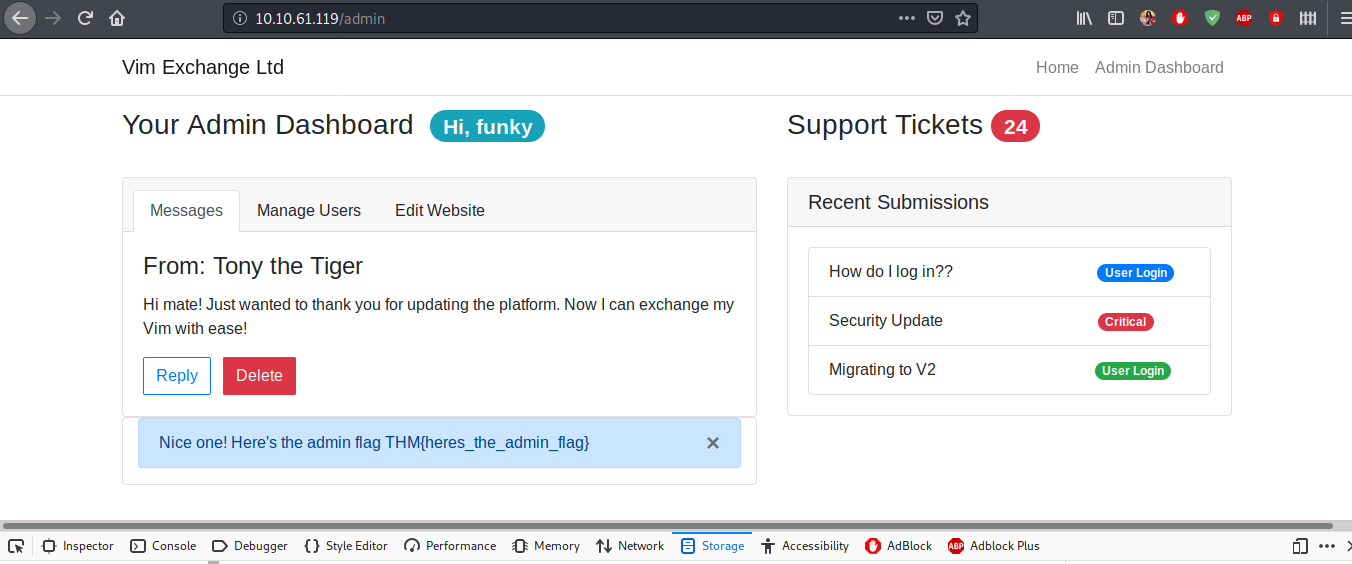
[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day08/cyberchef.png?raw=true)

THM{good\_old\_base64\_huh}

1. 2nd flag (admin dashboard)

For this, I followed the instructions the author provided and obtained the second flag.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day08/cookie2.png?raw=true)

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day08/flag.png?raw=true)

THM{heres\_the\_admin\_flag}

[**Insecure Deserialization - Remote Code Execution**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#insecure-deserialization---remote-code-execution)

This part was mostly following the instructions provided by the author of the room.

1. flag.txt

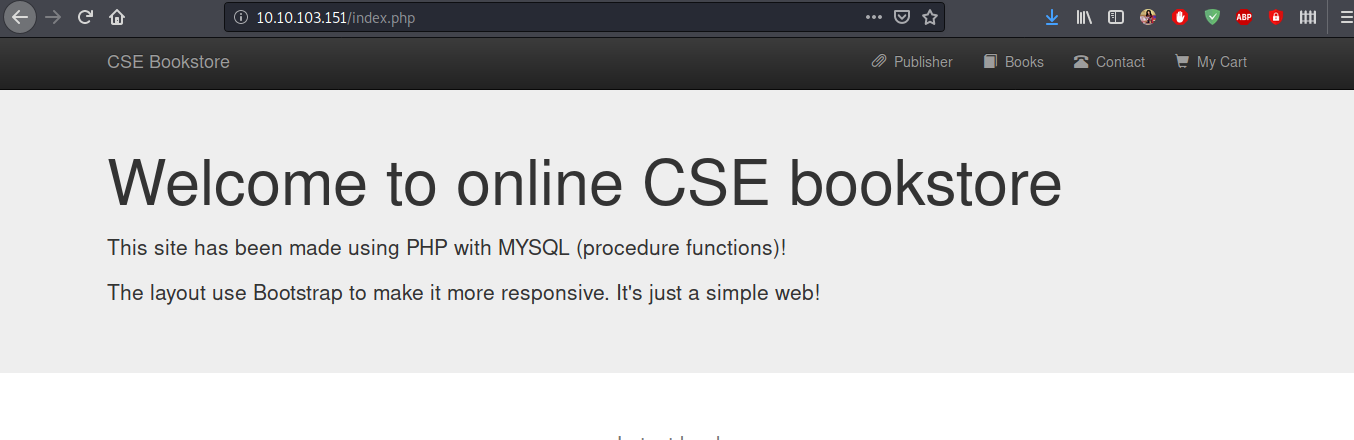
4a69a7ff9fd68

[**Topic 9: Components with Known Vulnerabilities**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-9-components-with-known-vulnerabilities)

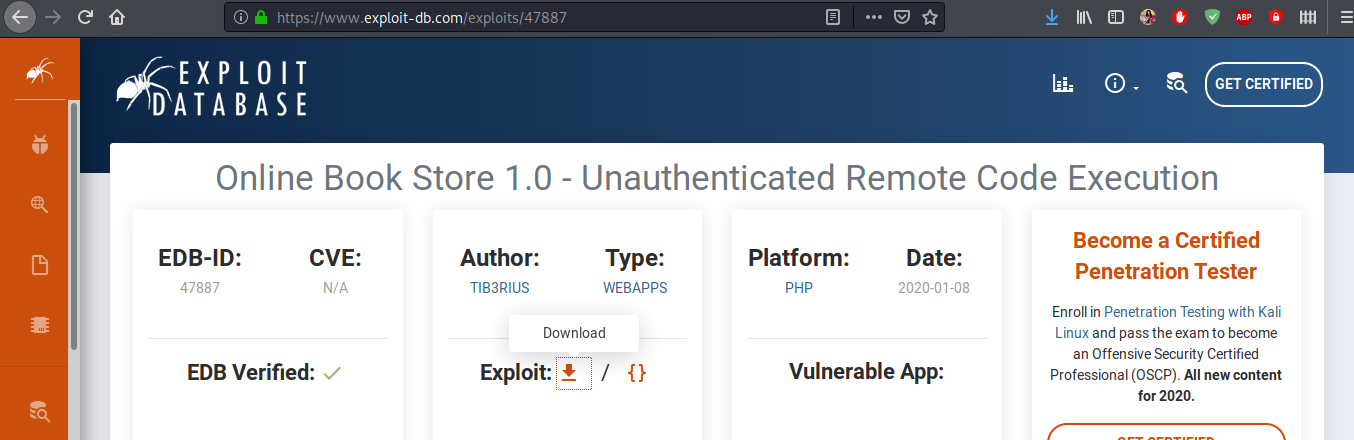
Machine IP: 10.10.103.151

1. How many characters are in /etc/passwd (use wc -c /etc/passwd to get the answer)

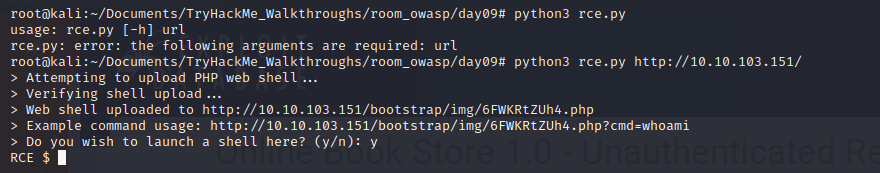
Using the browser to go to the main page of the web app, this is what is displayed:

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day09/index.png?raw=true)

I did a google search of **CSE bookstore exploit** and found the following [exploit-db](https://www.exploit-db.com/exploits/47887) page.

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day09/exploit-db.png?raw=true)

After downloading the exploit script, I tried running it, using the website's url as parameter, since that was what the script asked. After it ran successfully, the script gave me the option to launch a reverse shell in the command line. From this point, I easily found the number of characters are in */etc/passwd*

[](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/screenshots/day09/shell.png?raw=true)

1611

[**Topic 10: Insufficient Logging & Monitoring**](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/writeups/OWASP.md#day-10-insufficient-logging--monitoring)

Simply analyze the given [log file](https://github.com/pamhrituc/TryHackMe_Writeups/blob/master/room_owasp/day10/login-logs.txt) to answer the following two questions.

1. What IP address is the attacker using?

49.99.13.16

1. What kind of attack is being carried out?

Brute Force

The lab is Complete

