https://youtu.be/5bj1pFmyyBA

You have to define here complexity of vulnerability. Use
https://hacktify.in/bugbounty/index.html

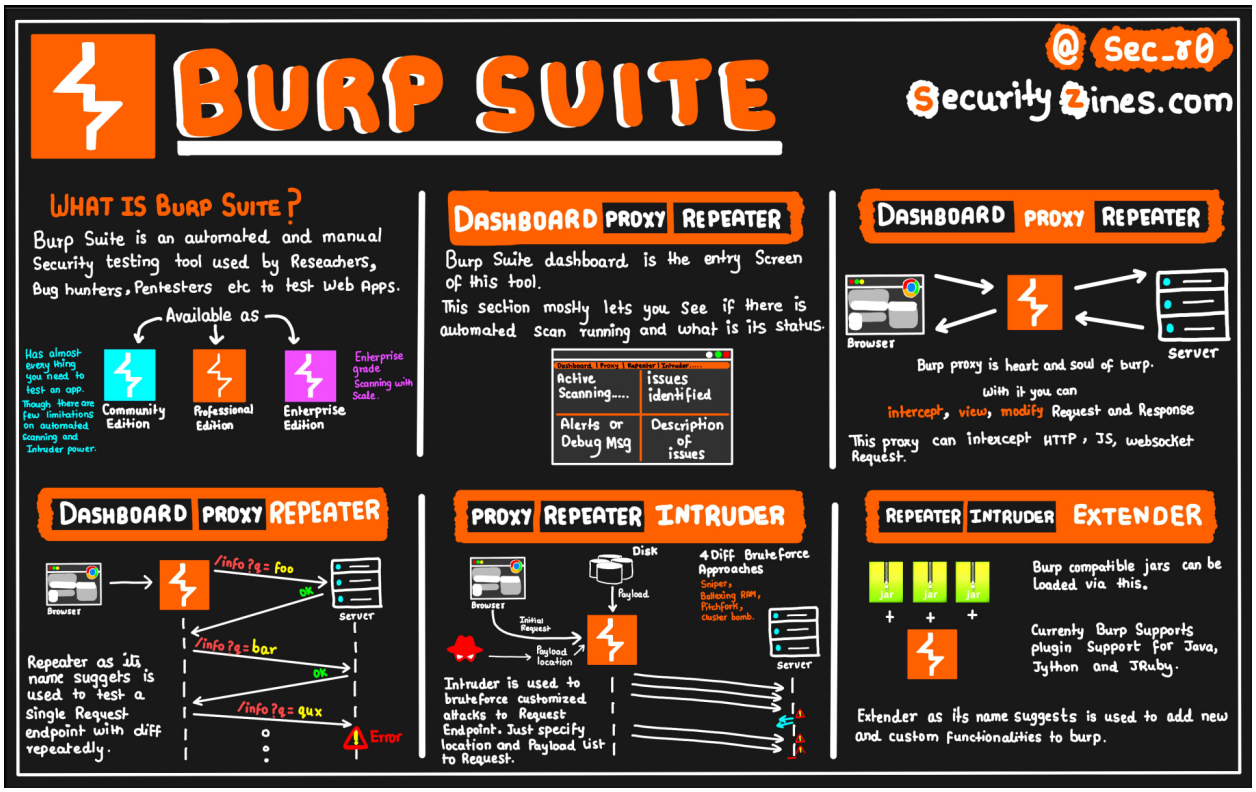
check out this https://exploit.linuxsec.org/xss-payloads-list/

Encode your payload using this link and use the encoded payload on the search box
https://meyerweb.com/eric/tools/dencoder/

https://meyerweb.com/eric/tools/dencoder/ paste your html code here and than copy output and paste it in lab

https://youtu.be/CxaCriqQfN4 Here watch this it's a bit lengthy process

https://www.youtube.com/watch?v=MGHnpbNZUXE Watch this you'll get it



Also, for the stored injection challenge, this may be of use.
https://www.hackingarticles.in/comprehensive-guide-on-html-injection/

Bug Bounty Hunting - Basic XSS Part 2 - YouTube

https://youtu.be/MGHnpbNZUXE Watch this video to understand

I think loading into iframe OR that Spinning dude is enough for PoC as we can display layers of pages @Atharva Varule Do you agree ? - for manual check this https://portswigger.net/web-security/clickjacking

you can refer this https://www.hackingarticles.in/comprehensive-guide-on-htm-injection/

This resource helped me solve the Stored HTML Inject challenge.
https://youtu.be/MGHnpbNZUXE

In Google Search

Html injection acunetix

Clickjacking acunetix

You see the risk

Try this https://htmlcheatsheet.com/ for html injection payloads

only tips i can say for
HTML injection:
1- use what you have learned in Burpsuit
2- see what is proper encode if your input encode and not allow you to type special char
3- see source page, in my opinion, these two tips should be more than enough

Sure. For this lab you'll be looking for `Stored HTML Injection`.

First you'll want to register a user and the log in to the application.

My methodology:
1. Enumerate all input fields.
2. Craft Payloads and Send them.
3. Refresh the page to see if your payload works.

Then refer this video
https://youtu.be/MGHnpbNZUXE

Html is the backbone of the webpage while javascript is used for programming the backend. Html is used with tags and just defining the structure of the webpage while javascript controls what happens when certain events take place and to sum it up, do all the backend work and control logic of application. Both are different! These labs are for html injection not xss

I'm referring to the `Clickjacking.pdf`
https://drive.google.com/file/d/11LfcIbGQA1LJ-VQaJzYcN_10Ie7k-31d/view

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Template%20Injection/README.md

for html lab2 https://youtu.be/Ig_Et0MSocs

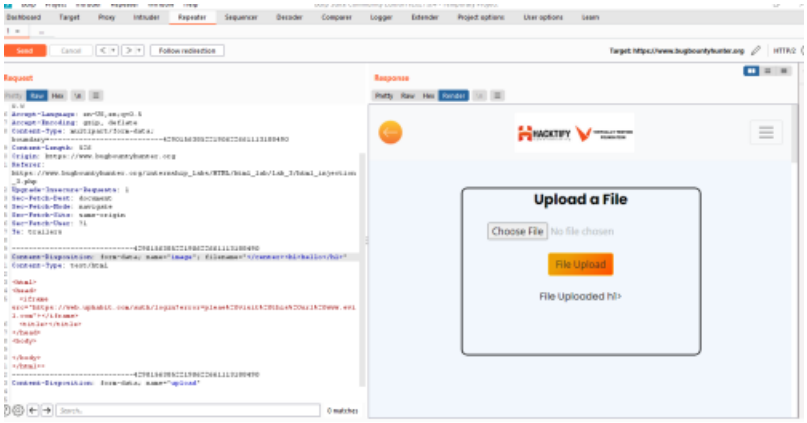
how did you escape</table> I'm not able to do it
try from here
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_headings_left

in encoding lab try to encode tag borders such as "<",">","/" and other as same find encoded form of <>/ each character and you are good to go

https://www.w3schools.com/html/html_urlencode.asp you can encode your payload here

encoded payload try this
%3c%68%31%3e%6c%61%62%20%73%6f%6c%76%65%3c%2f%68%31%3e

i tried many payload such as h1 and img src but fail also tried "> </h1> and but fail please guid me



Use This payload as file name:A
B
C
D
E

For solving clickjacking labs you can refer to this pdf : https://drive.google.com/file/d/11LfcIbGQA1LJ-VQaJzYcN_10Ie7k-31d/view you can use hacktify clickjacking tool.

Can some one say me how to solve lab 2 ? HTML injection

Here's the explanation of our payload working in Lab 2. Hope @Anirudh Nair doesn't mind, so many mates confused.

We can use "/> to break
out normally it would be like

```
<input type="text" value="your input"
```

but we make it

```
<input type="text" value="  "/> <h1>OurPayload</h1>
```

Hope you got working

For Lab6 all of who are in stuck here , they encode their payload/html code here and put in the search box: https://www.urlencoder.io/

May be we can use this for html lab-3
https://null-byte.wonderhowto.com/how-to/bypass-file-upload-restrictions-using-burp-suite-0164148/

https://hacktify.in/clickjacking/

Try this link and see if you can understand it
https://www.youtube.com/watch?v=sLdxy1u2pdo

https://brutelogic.com.br/blog/the-7-main-xss-cases-everyone-should-know/

Tip for lab 1: directly inject the HTML tags in the input field and you will win \n Tip for lab 2: View the source code and try to balance the payload and you will win! \n Tip for lab3: Upload file with a name containing HTML tags and you will win \n Tip for lab4: Upload a file containing simple HTML tags and you will easily win \n Tip for lab5: Read the https://www.hackingarticles.in/comprehensive-guide-on-html-injection/ and you will get easy win \n Tip for lab 6: Read the different web application encoding techniques and encode the payload and inject into the input field and you will easily win! \n All the best

Week 2

VTF internship Week 2 Technical Session - YouTube

Tips for solving

HTML Injection Labs!

Lab1:-
HTML's Are Easy!
Just try normal HTML Injection Payloads.

Lab2:-
Let Me Store Them!
There are filters in the input field , try bypassing them with "> in front of the payload.
Explore all the features of the given website in order to find where the vulnerability is!

Lab3:-
File Names are Also Vulnerable!
So the payload should be the filename of the file. In windows or in linux you cant change the filename with ending tags , so create a payload with SINGLETON TAGS. If you are trying in Burp Open the Burp , Off the Intercept , Go to the lab link , choose the file , on the intercept , change the file name to the payload(using singleton tags or without end tags) , forward the request and off the intercept and view the browser

Lab4:-

File Content and HTML Injection a Perfect Pair!
Create a html file and add the payload in it and upload

Lab5:-
Injecting HTML Using URL
Insert your payload in the url itself ie. /lab_5/html_injection_5.php/?{Insert the Payload Here}

Lab6: Encode IT!
Url Encode your payload!

lab 3 --> ><h1>payload<h1>

TIP:

For the filename challenge, you'll run into an issue with the "/" character. These are restricted in Linux environment. However, in HTML you don't necessarily have to use the closing tags. For example with the "<a>" tag, you don't have to close it with "". It will still render properly in browser even if you don't use the closing tag. So, you could name your file "<a>[place-any-filename-here]".txt and it would still work. I hope this helps.

1. in lab 6 u need to just encode the tags html encoding

Tips for Solving Clickjacking Labs

Lab1:- Let's Hijack
The webpage is a non sensitive one , so follow the clickjacking guide given to craft the exploit and open the html file , clickjacking vulnerability is executed.

Lab2: Re-Hijack!
The webpage is a sensitive one , so go to this url in order to craft the payload (https://hacktify.in/clickjacking/) and load the Lab's url. Now drag the Email , Password

and Submit field according to the login page and View (The option is Near to Load) the Url. Now enter the given credentials and submit. The credentials are captured the vulnerability is exploited.

Lab2: Re-Hijack!
The webpage is a sensitive one , so go to this url in order to craft the payload (https://hacktify.in/clickjacking/) and load the Lab's url. Now drag the Email , Password and Submit field according to the login page and View (The option is Near to Load) the Url. Now enter the given credentials and submit. The credentials are captured the vulnerability is exploited.

Week 3

VTF internship Week 3 Technical Session - YouTube

Tips for Solving XSS Labs

Lab1:- Normal XSS Payload
Lab2:- Balance the XSS Payload by viewing the source code!
Lab3:- Attach the payload at the end of the email. For example , test@gmail.com">payloadhere
Lab4:- Use any other payload other than <script> For example , <img src>
Lab5:- Use any other payload other than <script> For example , <img src>
Lab6:- Use any other payload other than <script> For example , <img src>
Lab7:- URL Encode the payload , (Note:- Avoid using bracket in the payload)
Lab8:- Filename should be the XSS Payload

If You are trying it in Burp Suite

Open the Burp , Off the Intercept , Go to the lab link , choose the file , on the intercept , change the file name to the payload(Use <img src>, and use prompt(1)) , forward the request and off the intercept and view the browser -> It should help I guess , I didnt do by using this method

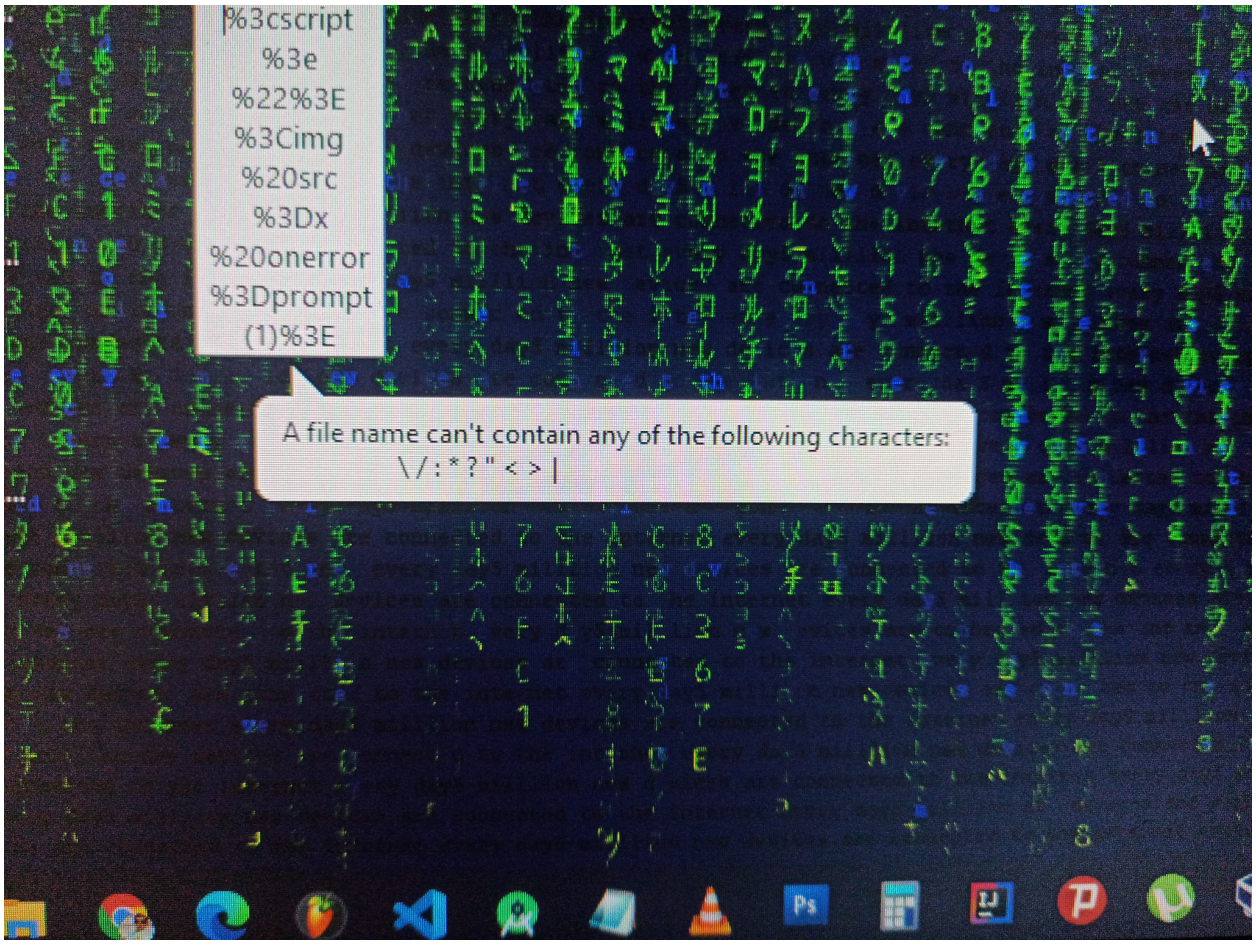
Lab9:- File Content should be the XSS Payload
Lab10:- Same as HTML Injection Lab , In The Update Section , XSS is vulnerable
Lab11:- Add the parameter and inject the payload. (?paramname={Payloadhere})

(I haven't tried using Burp Suite)
Lab9:- File Content should be the XSS Payload

Lab10:- Same as HTML Injection Lab, In The Update Section , XSS is vulnerable
Lab11:- Add the parameter and inject the payload. (?paramname={Payloadhere})

used below paylod still braces are filtered
%3c%2f%62%3e%3c%2f%68%32%3e%3c%2f%63%65%6e%74%65%72%3e%3c%2f%64%69%76%3e%3c%2f%63%65%6e%74%65%72%3e%3c%73%63%72%69%70%74%3e%61%6c%65%72%74%28%31%29%3c%2f%73%63%72%69%70%74%3e

https://www.urlencoder.org/ Use this website to encode the payload



So i found something malicious in lab 10 . hope everyone knows this issue if not then now you do we are able to do account take over here by inserting anyone's email and pass for example test@test.com i tried to register and login and when i entered the

creds after registering all the fields was covered with payloads while i have not entered a single payload there.

If you think you did everything right including balancing tags, yet it still prints your payload out then change your payload. https://github.com/payloadbox/xss-payload-list this is a list of payloads no need to micromanage. Change them as often as you have tried everything

If write ups are not allowed! Check this out https://youtu.be/OAS3ku6gFWs

any hint for DOM xss ??
https://youtu.be/18vThL59L-Y Watch this

try the basic payload with balancing(">) and then store the first name and last name with the payload for lab 10

yep....https://brutelogic.com.br/blog/dom-based-xss-the-3-sinks/ (for help) for lan 11

https://brutelogic.com.br/blog/dom-based-xss-the-3-sinks/

Tips for Solving XSS Labs

Lab1:- Normal XSS Payload
Lab2:- Balance the XSS Payload by viewing the source code!
Lab3:- Attach the payload at the end of the email. For example , test@gmail.com">payloadhere
Lab4:- Use any other payload other than <script> For example , <img src>
Lab5:- Use any other payload other than <script> For example , <img src>
Lab6:- Use any other payload other than <script> For example , <img src>
Lab7:- URL Encode the payload , (Note:- Avoid using bracket in the payload)
Lab8:- Filename should be the XSS Payload (I havent tried using Burp Suite)
Lab9:- File Content should be the XSS Payload
Lab10:- Same as HTML Injection Lab , In The Update Section , XSS is vulnerable
Lab11:- Add the parameter and inject the payload. (?paramname={Payloadhere})

Use <img src> , prompt(1) for the payload as filename , if you are trying in Burp Suite . It works!! For lab 8 its another way…
Mistake while solving lab 8 ….All this while I was balancing the payload. That was the mistake.

anyone know where to get sample png imge with exifdata?
https://github.com/ianare/exif-samples

Week 4

VTF internship Week 4 Technical Session - YouTube

I found online - PNG does not support embedding of EXIF information. When you convert from JPEG to PNG the information is lost. here is the link.
-**https://stackoverflow.com/questions/9542359/does-png-contain-exif-data-like-jpg**

https://portswigger.net/support/using-burp-to-test-for-open-redirections

As spidering is deprecated in the newer versions of BurpSuite Community, you could rely on **https://github.com/jaeles-project/gospider** or **https://github.com/hakluke/hakrawler**.

You don't need one, but you could refer these
https://github.com/jaeles-project/gospider or
https://github.com/hakluke/hakrawler.

Lab 1, As one above said "do intercept", capture response, change 302 to 200 OK. Forward. You didn't redirected to login page but here it opened open_redirect_1.php Dont know what's the aim of this lab.

https://book.hacktricks.xyz/pentesting-web/open-redirect

for lab 2: you've to add the extra header which can redirect you to another site
in lab 2 we need to do changes direct in the url right?
You have to add host header injection for lab 2

https://infosecwriteups.com/identifying-escalating-http-host-header-injection-attacks-7586d0ff2c67 Refer this for lab 2 Who ever is having doubt

https://youtu.be/Oylu2qoGpcg Refer this If you don't know about it

Hey everyone! I've found a lab for open redirection vulnerability which might help you Here's the link:
https://application.security/free-application-security-training/owasp-top-10-insecure-url-redirect Go ahead, it's completely free and also no need to create account over there The website also contains many other labs of all OWASP Top 10 vulnerabilities. Check it out here: https://application.security/free/owasp-top-10

For
lab 1 :
1. if you're redirecting to login page it's intended.

2. if you're getting 403 and cloudflare than if it got working in an hour or so then all are normal if it did not such screenshot will give you the required marks.

lab 2 : you've to add the extra header which can redirect you to another site.
lab 3: play with url parameter.
lab 4: try to insert your payload after `.php`
lab 5 : try to create a svg file and insert your payload in that file and upload it.

Try this one it worked for me

```
<script>
if (top.location.host = "bugbountyhunter.org") {
    window.location.href='http://evil.com';
 }
 </script>
```

lab 6: try to add the same url parameter and add your payload.
lab 7: same as lab 6 but now focus on the lab title :)
lab 8: Use the IP

Lab2 SubLab 5 svg paylaod not working. I have solved this lab before (got PoC SS too) with .txt file having simple HTML and JS paylaods; but now even that is not working anymore.. any help appreciated!

https://youtu.be/Oylu2qoGpcg
They are
There is an attack called response manipulation

https://coolsymbol.com/number-symbols.html

https://hackerone.com/reports/601287 read this for Lab2...it will help you out.

1. try to go with svg file content in the link and create a new txt file and add the codes.after completing the codes,rename the extension to filename,svg and try to upload them

https://book.hacktricks.xyz/pentesting-web/open-redirect#common-injection-parameters

while sending request from burp repeater getting 302 response ...but after in browser getting no redirection

1. check the open redirect PDF document provided for this week (Available in the weekly technical skills guide link). It has all the Server response codes. This should help you understand what the server is trying to communicate through that code. 2. Check the various HTTP response headers in the server responses. The value of one of them in particular should be changing to your injected payloads. 3. Also, in burp repeater, you should get a pop-up button that says "Follow Redirection". So, if you click on follow, and you're landing on your injected payload site, then I believe you've successfully exploited the open redirect vulnerability. Hope this helps!

https://blog.yeswehack.com/yeswerhackers/file-upload-attacks-part-2/

Please Note:

<<<<<< Lab 1 Hint >>>>>>>
1. Capture the request of Start Lab in Burp
2. Do intercept the response of the request.
3. Change 302 Found to 200 Ok.
4. Scroll down, Change href="../../index.php" and change to href="https://google.com/"
5. The page is redirected to "Welcome to OpenRedirect Lab"
6. Move the cursor to Hactify and VTF logo. (below left corner of the browser you will see https://google.com/)
7. Click on the logo, it will get redirected to google.com.
8. Lab succesfully completed.

it's not the correct solution it's just response manipulation. by this you can hack any website

Bro that not how a vulnerability works You are intentionally redirecting yourself to the malicious site You can't ask your victim to go intercept the response and change the location header to the malicious site

In response you can add <script>alert (1)</script>, and there was xss again, this works on Google too

Oh ok, I had looked at this report https://hackerone.com/reports/104087

Did you use this link to find out if the image you had uploaded has metadata?
http://exif.regex.info/exif.cgi

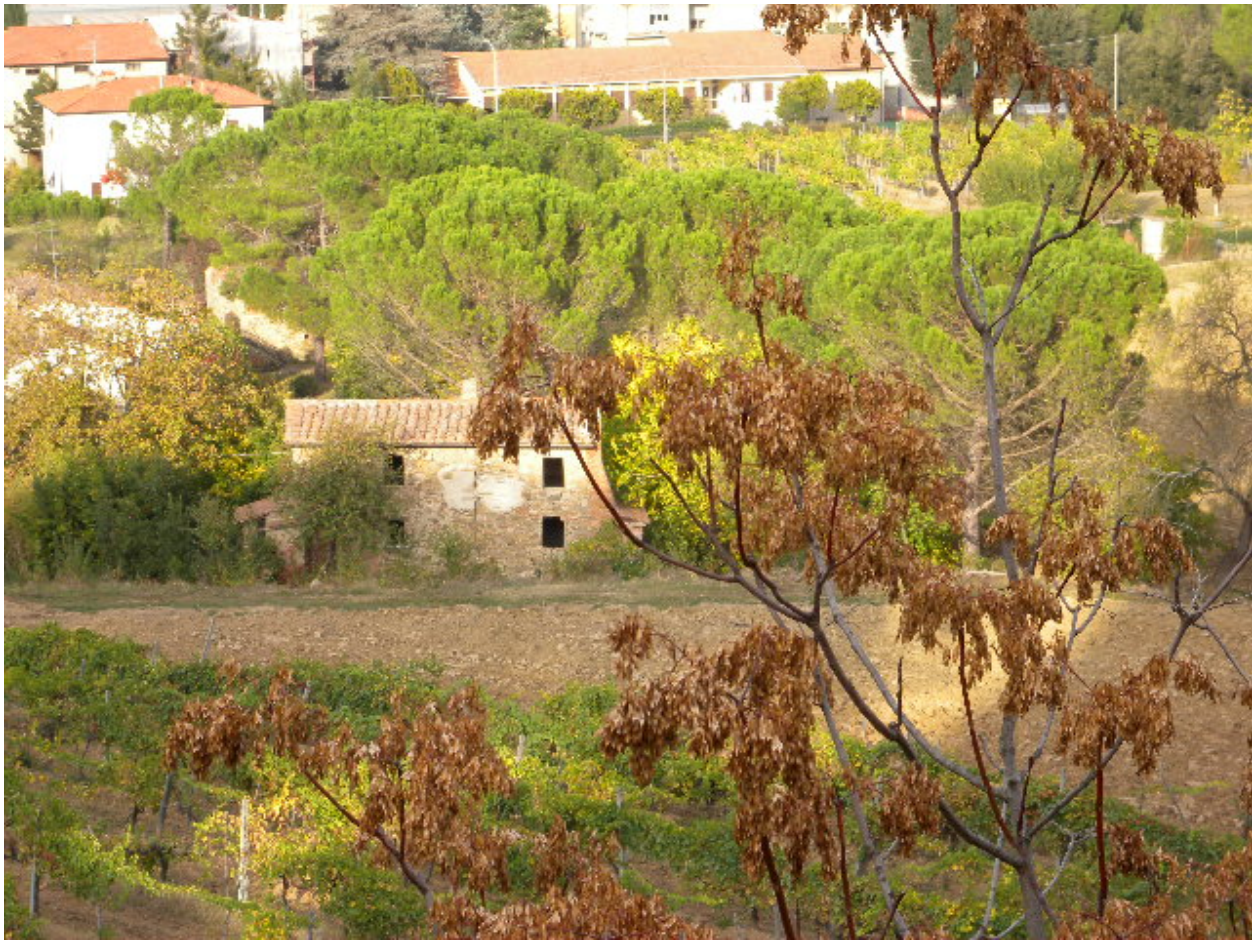
lab 1: intercept the request -> change the host -> you'll either get 403 or it'll get redirected. you've to take screenshot of that
lab 2 : you've to add the extra header(starts with X-) which can redirect you to another site.
lab 3: play with url parameter.
lab 4: try to insert your payload after dashboard`.php` and use ///
lab 5 : try to create a svg,html,txt etc file and insert your payload in that file and upload it.
lab 6: try to add the same url parameter and add your payload. by
lab 7: same as lab 6 but now focus on the lab title :)

lab 8: Use the IP

if you're not able to understand you can refer this article for understanding open redirect :
https://book.hacktricks.xyz/pentesting-web/open-redirect

or lab 5, just rename your file with .html extension and upload it capture the request in burp suite sent it to repeater and in repeater again send it so that you'll have the request and its response side by side take the screenshot and your PoC is ready

Host header injection attack
https://infosecwriteups.com/identifying-escalating-http-host-header-injection-attacks-7586d0ff2c67 Refer this for lab 2



https://youtu.be/Va_TvyBjtKA Refer this its helpful

https://github.com/ianare/exif-samples/blob/master/jpg/gps/DSCN0010.jpg use this image it is working for exif lab

lab 1: intercept the request -> change the host -> you'll either get 403 or it'll get redirected. you've to take screenshot of that
lab 2 : you've to add the extra header(starts with X-) which can redirect you to another site.
lab 3: play with url parameter.
lab 4: try to insert your payload after dashboard.php and use ///

lab 5 : try to create a svg,html,txt etc file and insert your payload in that file and upload it.
lab 6: try to add the same url parameter and add your payload. by
lab 7: same as lab 6 but now focus on the lab title :)
lab 8: Use the IP

https://book.hacktricks.xyz/pentesting-web/open-redirect#open-redirect

first upload the image in exif lab -> after uploading copy image link -> use
http://exif.regex.info/exif.cgi

For demonstration of this vulnerability in real life solve this guided lab. you'll get a taste of it Lab -
https://application.security/free-application-security-training/owasp-top-10-insecure-url-redirect

Are these style digit symbols? https://www.i2symbol.com/symbols/style-digits yes

Any file uploaded will not be reflected unless it's renamed to `redirect.*`.
File Content at default is :
<script> if(top.location.host="bugbountyhunter.org"){
window.location.href='https://virtuallytesting.com/'; } </script>

This is for lab 5

lab 7 and lab 8 both are different. In
lab 7 : it's same as lab 6 that means you've to use 2 url parameters but you've to use ip.

lab 8: either you can use style digit symbols (google it) or you can directly put the ip in url parameter.

Okayy here you go then

Tips for solving Open Redirect Labs

Lab1: Check the pinned message (Just change the host , you should be redirected or a 403)
Lab2 : Host Header Injection Attack , so use `X-.....`
Lab3: After clicking login , you will see the parameter named ?url= insert the payload there
Lab4: Insert the payload right after `.php` and add /// So the payload looks like , `.php///{payload}`
Lab5 : Check the pinned message
Lab6: After clicking login , Copy the url parameter with the value and paste it one more time , now this time attach the payload to the second value , Example: `&url=value&url=value{payloadhere}`
Lab7: Similar to Lab6 , for the payload use the subdomain
Lab8 : Similar to Lab3 , for the payload use IP Address of the payload's site

Check pinned hints or refer this
https://infosecwriteups.com/identifying-escalating-http-host-header-injection-attacks-7586d0ff2c67

Week 5

it's divided into 2 steps:
First : verification if cors is present or not
second : exploitation using the html script

yes, lab 2 is CORS with Null origin

Unable to solve
Lab 1.

Steps performed so far:-
1) In the header of request, changed entry of origin to Origin: https://www.attacker.com/

2) Intercepted response in proxy tab
2) Clicked Forward and In response, Changed 302 to 200 OK and click Forward
3) After this point, login does not happen upon clicking Forward multiple times. Not sure what to do next.

Also, "Access-Control-Allow-Credentials: true" is coming only in Repeater but not in proxy in response

Access-Control-Allow-Origin: attacker.com
Access-Control-Allow-Credentials: true
If these 2 comes then
step 1 completed that you have found vul. Now
step 2 is try to use payload and see if it works

https://drive.google.com/file/d/1ysaZkr4yh9U5rNJA0YawoZlVMX1OaSBz/view?usp=sharing Check this & correct me please !!

Hello, this writeup made some concepts clear to me and will help you guys in solving your lab too. Give it a read
https://infosecwriteups.com/stealing-user-details-by-exploiting-cors-c5ee86ebe7fb

Everyone the hint for each and every lab is the name of lab itself and even if you don't get anything from it then the next hint is the rules given in the lab before we click on start lab. These rules have actually shown what steps to be taken to verify that CORS vulnerability exists. Even after these 2 hints you can't find vulnerability then I would recommend from materials those clicks are given below:
https://book.hacktricks.xyz/pentesting-web/cors-bypass
https://infosecwriteups.com/think-outside-the-scope-advanced-cors-exploitation-techniques-dad019c68397
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/CORS%20Misconfiguration

The hints before lab start are the best still if you are confused then can check this links out
https://infosecwriteups.com/think-outside-the-scope-advanced-cors-exploitation-techniq

ues-dad019c68397
https://book.hacktricks.xyz/pentesting-web/cors-bypass#regexp-bypasses

https://drive.google.com/file/d/1ysaZkr4yh9U5rNJA0YawoZlVMX1OaSBz/view —> please have a look and confirm !!
sir when u are running script no need to use burp and intercept it. Just run without it. It will surely work else things are good

NOTE: We can use both Burp Repeater's response. & Exploit's RESPONSE as POC. Because we turned ON proxy for Burp's response. But PROXY should be OFF to run EXPLOIT code as we are running it in local host.

Everyone the hint for each and every lab is the name of lab itself and even if you don't get anything from it then the next hint is the rules given in the lab before we click on start lab.
These rules have actually shown what steps to be taken to verify that CORS vulnerability exists. Even after these 2 hints you can't find vulnerability then I would recommend from materials those clicks are given below:
https://book.hacktricks.xyz/pentesting-web/cors-bypass
https://infosecwriteups.com/think-outside-the-scope-advanced-cors-exploitation-techniques-dad019c68397
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/CORS%20Misconfiguration Initially I would recommend with just going for verification part that yes lab is vulnerable to CORS and then after completing all labs you can try to exploit with payloads.

https://www.kumaratuljaiswal.in/2021/07/exploitation-of-cors-prefix-suffix-match.html just read this article you will get to know

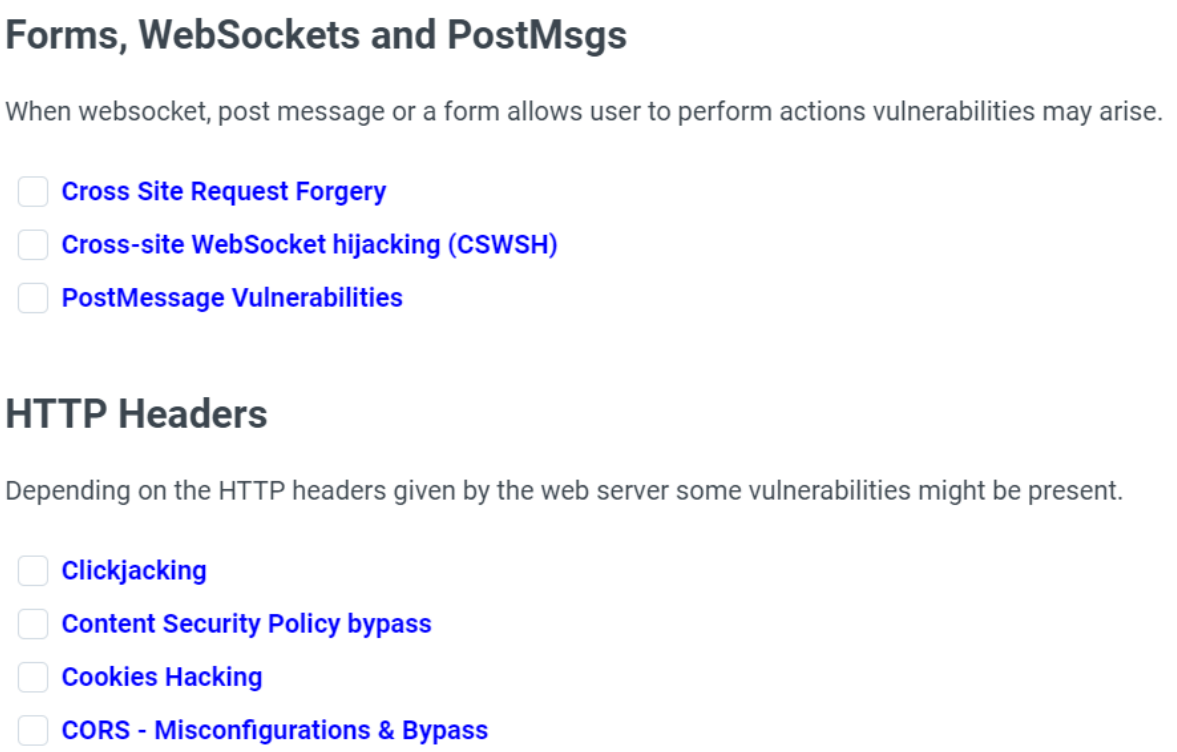
```
curl https://vulnerable_website.com -I -H Origin:attacker.com

-I : Returns only the Headers of the response
-H : Sends a custom header along with the request
Origin: This is the custom header
```

Week 6

I recommend this playlist for csrf :
https://youtube.com/playlist?list=PLuyTk2_mYISKNFqao_NBzYOWvJFqhVmXN

Try this https://security.love/CSRF-PoC-Genorator/

Below is a great source to get the knowledge from
https://book.hacktricks.xyz/pentesting-web/csrf-cross-site-request-forgery



https://youtu.be/sSm16nXoXmI

1. use this to add poc creator to community version of burp suite

csrf poc is available at https://hacktify.in/bugbounty/

And heres right methodology, according to me.
1) Register with "attacker@gmail.com" & "attacker" password.

2)Register with "victim@gmail.com &" victim" password.
2)Login to "attacker" account.
3)Click "change password".
4) New password "password" Confirm password "password". Intercept ON, Submit.
5)Capture that request.
6)Engagement tools> Generate CSRF POC.
7) Save in CSRF.html.
8)Open chrome. Login to bugbountyhunter.org.
9) Open CSRF.html in Chrome browser. Click "submit request".
10)Now the "victim" users password is changed to "password".And "attacker" can login with "victim@gmail.com" & "password" creds.

^^^^Aim of all labs : At the end, Login with victim mail & your own POC script generated password.

https://hackerone.com/reports/174228

we just have to rm -rf the values that are csrf associated like the name and value ... like in linux we delete the directory by rm -rf 'dirname'

In lab6 token means CSRF Token
Now only you need to focus on the lab name.. The **rm -rf** means? You have to find this!

https://github.com/payloadbox/xss-payload-list

https://github.com/s0wr0b1ndef/WebHacking101/blob/master/xss-reflected-steal-cookie.md you will find payloads here

in lab3 --
1)I generate CSRF POC for both attacker and victim
2)then i replaced the victims csrf token with attacker one.
3)Then I ran victim csrf poc in browser. Victim'sPassword successfully changed

you can refer this : https://github.com/merttasci/csrf-poc-generator

Can anyone tell me if we want to find a vulnerability or hack an application Which tool are we supposed to use How is the process ?

First of all you've to do enumeration of the web application like trying different input fields , finding directories , ports etc after that try to find Vulnerabilities keep in mind. Enumeration is the key.

Eveyone those who are having doubts on how to create CSRF Poc then following are the ways:

1. Use Burpsuit Pro

2. Use a extender of CSRF Poc in burpsuit community edition For referance see this video https://www.youtube.com/watch?v=sSm16nXoXmI

3. Use a online CSRF Poc generator Link for it is
https://security.love/CSRF-PoC-Genorator/

4. Code yourself

Remember when you are using the 1,2,3 ways then you may need to edit few things and make it actually workable.(Hint is the action link and few inputs depending on the lab)

Like if you are logged in as victim, you need to login as attacker at the same time, but in a different tab/incognito mode. So, you'll have 2 sessions running at the same time. So, when the attacker will send the exploit to the victim with his/her own token, the victim's password will change!! So, that's your vulnerability and hence the lab name hint for lab 3

For Lab5:
Grab CSRF Token of Victim with XSS and exploit CSRF
For XSS, grab payload from:
https://github.com/swisskyrepo/PayloadsAllTheThings
Get your free Server for testing, alternative to BurpCollaborator:
https://beeceptor.com/
Not Required for Lab but understanding:
Normal XSS case: you just have to alert() and you see cookie
Remote case: If you want someone else(victim)'s cookie to be fetched.. you'll use a payload to get Victim's cookie delievered to You(Attacker)(Your Server). People

often use BurpCollaborator which is a feature of Pro version. Free Alternatives like Beeceptor can be used. (edited)

lab 6 i.e. rm -rf tokens does that mean it is telling us to remove the csrf token from our PoC? because in linux, rm -rf means removing something (file/folder)

use the attcker's token for lab 3

perform xss and obtain info about cookies and use them to exploit csrf for lab 5

change post to get or vise versa as per your poc and try. For lab 4

Lab1
Go with a simple CSRF PoC Payload generated by **Burp Pro CSRF PoC Generator** or **csrf-poc-generator burp community plugin** or **simply code it yourself** !! Whatever suits you the best

Lab2
changed the token value to some random number and it got executed properly
For lab 2, if i change the token with random number and then execute it. Is it fine
yes

Lab3
Like if you are logged in as victim, you need to login as attacker at the same time, but in a different tab/incognito mode. So, you'll have 2 sessions running at the same time. So, when the attacker will send the exploit to the victim with his/her own token, the victim's password will change!! So, that's your vulnerability and hence the lab name

Or

in lab3 --
1)I generate CSRF POC for both attacker and victim
2)then i replaced the victims csrf token with attacker one.
3)Then I ran victim csrf poc in browser. Victim'sPassword successfully changed

Final way
In lab 3 we have to change the csrf token value from victim to attacker!? Yes

Lab 4
This is basically something at the back-end fault that we aren't able to do that ... What other thing we can interpret from this can be …
No. I think you got the lab name misinterpreted. You're getting confused between GET & GET method. Basically that is to just confuse you. It actually means that you GET the CSRF token reflected on the webpage when you try to go for the POST method, without intercepting it. And when you find the vulnerability, you can go for your CSRF PoC, which requires to have a POST method !! So, that's the whole story !! Let me know if you still carry any queries !!
It's all about finding the vulnerability ... Whatever may be the way ... This is how I think it to be because it didn't worked by changing the request method from POST to GET !!

Lab 5
Hmmm see when you enter your xss payload the url looks something like this
Https://bugbounty.orh/lab/?name=payloadtogettokenonAttackercontroledserver
You can send him this link, if he is logged in his account then we can get his token
Hope this explains it

Lab 6
What the lab name you think means? Ever worked with linux? What does rm -rf command does? Now, try to use that in terms of CSRF & find the vulnerability ... Once, you captured it, then go for the exploit !!

Adding Salt to Hashing: A Better Way to Store Passwords (auth0.com)

Week 7 & 8

https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost

https://www.pythonforbeginners.com/modules-in-python/how-to-use-simplehttpserver
https://spoofing.medium.com/how-to-make-a-simple-http-server-using-python-ea35f0b741a4

https://www.youtube.com/watch?v=ih5R_c16bKc Resource for SSRF

https://youtu.be/NBCrlRqX2AY Video on ssrf by hacktify

use this link as a reference and take the lab name as a hint:
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Request%20Forgery

Lab 5 Tip: Using the cURL command below we'll notice that 127.0.0.1 is blacklisted.

```
curl
https://www.bugbountyhunter.org/internship_labs/HTML/ssrf_lab/lab_5/lab_5.php\?url\=http://127.0.0.1/ -s | html2text
```

These are blacklisted as well

```
http://127.0.0.1
http://localhost
https://127.0.0.1
https://localhost
```

(edited)

Use the Decimal values for this lab... Find/search here :
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Request%20Forgery

Is anyone running gdb debugger in linux , I have to exploit buffer overflow in c++ program .My laptop stopped working , Can anyone help??

Like:

brainpan
vulnserver
FreeFloatFTP
Savant
Minishare
WarFtp
SimpleWebServer
SLmail
Pcman

https://github.com/roottusk/vapi
vAPI is a Vulnerable Interface in a Lab like environment that mimics the scenarios from OWASP API Top 10 and helps the user understand and exploit the vulnerabilities according to OWASP API Top 10 2019. The lab is divided into 10 exercises that sequentially demonstrate the vulnerabilities and give a flag if exploited successfully.

It might be useful for Developers as well as Penetration Testers to understand the type of vulnerabilities in APIs. The lab is divided into 10 exercises that sequentially demonstrate the vulnerabilities and give a flag if exploited successfully.

Everyone you can practice IDOR here (Task 6/ Day 1)
https://tryhackme.com/room/adventofcyber3. This is a very simple lab with good amount of details and materials

https://bit.ly/3ltzfi7 Have a look on that guys to learn about ssrf in detail

read ssrf lab walkthroughs lab 7
https://vtfoundation.notion.site/Week-7-e4856ae60332461181ed5235fc6e3942

Week 9

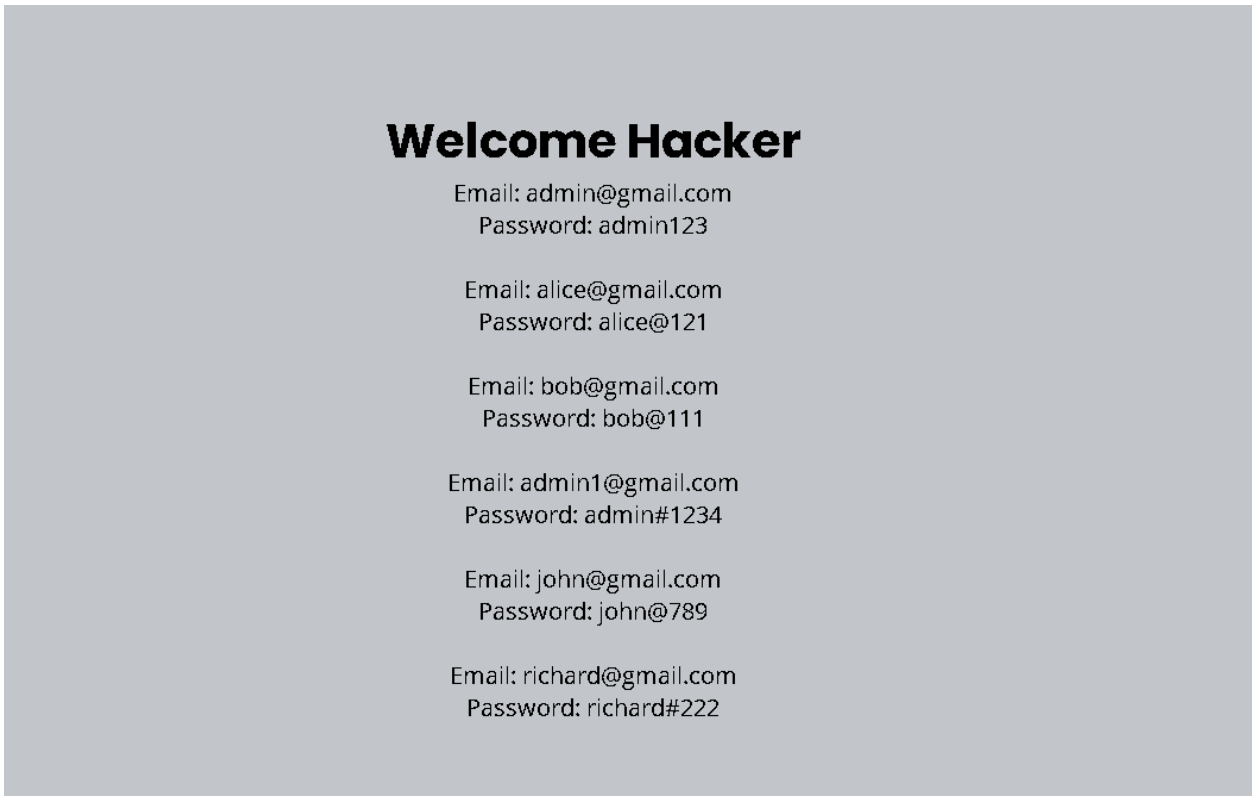
playlist for learning sql injection
https://youtube.com/playlist?list=PLuyTk2_mYlSLaZC4fVqDuW_hOk0dd5rlf

https://infosecwriteups.com/exploiting-error-based-sql-injections-bypassing-restrictions-ed099623cd94 go through this for this weeks lab

https://book.hacktricks.xyz/pentesting-web/sql-injection

Give a check at this
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%20Injection

For lab 2 pop



https://github.com/payloadbox/sql-injection-payload-list refer payload from this , try balancing from error shown

Mates, Lab6 was very messy. I tried lots of payload from https://github.com/swisskyrepo/PayloadsAllTheThings and one worked, also you have to provide like `a vaid mail address + payload`

Lab 7 use simple payload with comments

20211211_Comprehensive_new__https_www_bugbountyhunter_org_internship_labs_HTML_ctf_index_php_url_http_3A_2F_2Flocalhost_3A443.pdf (amazonaws.com)

there i am giving some hints for labs which i have solved yet

lab 1: use your payload with or and double quotation
lab 2: use the id with numerical value in the URL with some payload which will reflect all users credentials
lab 3: same as lab 2 but only need to do some changes in your payload
lab 4: now you have to use some tricky payload which return true in both input field
lab 5: as it name explaining where you have to use your payload, refer to pdf shared of sql injection.
lab 6: try to balance your payload what you have used, use “)or(“1”)=(** these kind of payload.
lab 7: use some payload and after your payload use some comment types which are used in sql for commenting for payloads of sql injection refer to this
lab8:use the payload in the user-agent header
similarly for lab9 and lab10,use the payload according to heading of the lab
,for lab9 it might be the referer header and
for lab 10 it might be the cookies/phpseesid header
Lab 11 and 12 hint
https://owasp.org/www-community/attacks/SQL_Injection_Bypassing_WAF

For lab11:
https://www.ptsecurity.com/upload/corporate/ww-en/download/PT-devteev-CC-WAF-ENG.pdf

1. More hint in Lab11: what does the web server do when adding another parameter?

https://github.com/payloadbox/sql-injection-payload-list

Here's the hint for Lab-8. Append this in User-agent: And find difference And let me know if got any leads. ")-- " & ')-- '

Lab-9 admin@gmail.com admin123

For lab6 ")or(" 1")=(** find the wildcard characters, it might be single quotes or double quotes or combination of both)

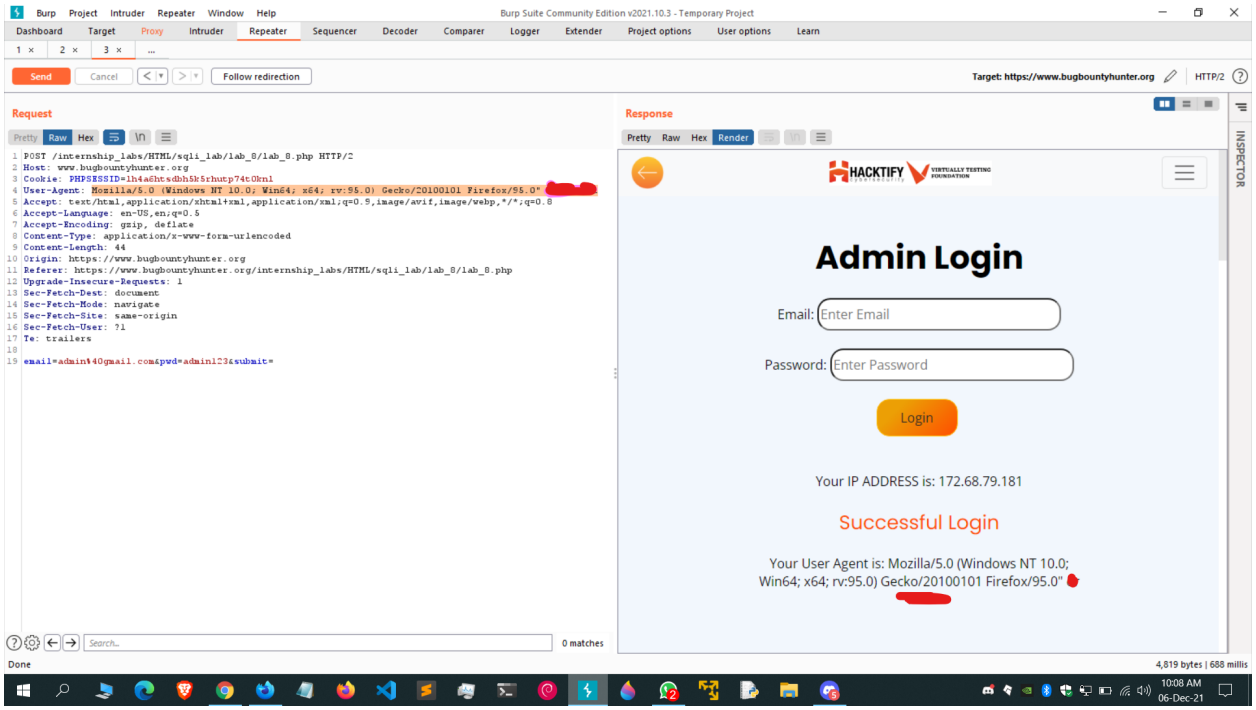
LAB1 HINT Dont close the payload Here the SYNTAX, replace £ Just " OR ££=£

working on remaining labs

Guys have look on that link for ease in solving this weeks assignment
https://www.hacksplaining.com/ they have great resources to learn web application pentesting . and its free . they have good range of tutorials and study materials

`lab 1`: **normal sql query ' or 1=1 #**
`lab 2`: **same as lab 1 but use --+ for commenting**
`lab 3`: **same as lab 2 but use + instead of space**
`lab 4`: **use union based SQLI (it worked for me). use # for commenting. supply the payload in both the fields**
`lab 5`: **same as lab 3 but use "=" instead of booleans like 1=1**
`lab 6`: **brackets are used (). use payloads using "(double quotes) and ()**
`lab 7`: **same as lab 6 but insert the payload using burpsuite in the username and password parameters. Use Repeater.**
`lab 8`: **for lab 8. login using previous recieved creds of the accounts. the u will get the user-agent. copy it and paste in the User-Agent header and supply payload and the end of it. Use AND instead of**

OR in the payload. this is the poc
this is the poc. only the malicious payload is hidden so to not reveal the answer
Step 1: Log in using credentials u recieved by solving previous labs
Step 2: After successful login u will get User-Agent message in the site
Step3: Copy it and paste it in Repeater. Make sure to capture the Request by just simple refreshing
Step4: Paste the copied User-Agent in the Repeater and at the end of it try giving SQL payloads.
Step5: Upon successful injection, your User-Agent with the payload will be visible in the screen
This is the complete procedure for lab 8. The POC will somewhat look like this



`lab 9`:same as lab 8 u will be getting refferer. follow the same steps as done in lab 8. whereas add the referrer in the Referer header only. not in User-Agent

`lab 10`: you will be getting delete your cookies page

`lab 11`: use time based SQLi. use OWASP blog (it worked for me)

`lab 12`: same as 11. but modify the payload

I am not supposed to reveal answers but since u are not able to do thats why i am doing it. for 6 use: `admin@gmail.com") OR ('')=('')#` for lab 8: login first with admin@gmail.com and admin123. then u will get User agent msg on the site. copy it and paste in burp repeater in the User agent section. then add payload `" or "1"="1` These steps worked for me. hope it will work for u too

Also u can—-- **") or ("x")=("x** use this for lab 6

For people
who are solving
lab8 By appending " AND in User-agent. It's not true. Even if I use "Anything, It gets reflected.
SQLI is not that is reflecting something which is manipulating in burp.

Instead
-Either you should login without credentials.

-Or Make Server to dump & display all data on browser.
Please correct me in this if I'm wrong....
The lead I got for lab 8 is
Using below in User-agent
') - - '

For LAB 11.
There are 3 reflecting parameters.
Tips to know it.
Try one by one after?id=
& find difference
'UNION SELECT sleep(5)-- -
' UNION SELECT 1,sleep(5)-- -
'UNION SELECT 1,2,sleep(5)-- -

There won't be any reflection in *TIME BASED SQL INJECTION* in real world.
Information can be retrieved like this
{ If first character of password = 'a' then sleep(5) else sleep(0) }

.

Please check lab 12 of portswigger labs.

Special hint for lab 11 and 12
https://owasp.org/www-community/attacks/SQL_Injection_Bypassing_WAF

TOP 25 SQL INJUCTION PARAMETERS FOR BUGHUNTERS

1. ?id={payload}
2. ?page={payload}
3. ?dir={payload}
4. ?search={payload}
5. ?category={payload}
6. ?class={payload}
7. ?file={payload}
8. ?url={payload}
9. ?news={payload}
10. ?item={payload}
11. ?menu={payload}
12. ?name={payload}
13. ?name={payload}
14. ?ref={payload}
15. ?title={payload}
16. ?view={payload}
17. ?topic={payload}
18. ?thread={payload}
19. ?type={payload}
20. ?date={payload}
21. ?form={payload}
22. ?join={payload}
23. ?main={payload}
24. ?nav={payload}
25. ?region={payload}

CHECK OUT

www.hackerbro.in

Lab 12

```
?id=<add_random_no._here>'<here add payload here>--+
```

https://github.com/payloadbox/sql-injection-payload-list

I was able to display all the users using the list from Generic SQL Injection Payloads - there are three payloads from this list that will produce all the emails and passwords. https://github.com/payloadbox/sql-injection-payload-list

Tips for solving labs 10,11,12

Initially remember you will get your required poc from burpsuit response

Lab 10: Enter admin as username and password and then you will get a page with many details and a button at its bottom "Delete your cookie". So now capture this page on burpsuit and then play with the cookie header and specifically username field and as you give right payload you will be able to get username,database name,version,... any such things depending on what you asked from database in the payload.

Lab 11 & 12: Capture the main page lab_11.php and lab_12.php in burpsuit and then try to explore id field values and you will surely get username,database name,version,... any such things depending on what you asked from database in the payload.

And last hint is try with union based payload in all these labs

This is a list of payload I used for each

lab 1 - " or 1=1# (In PWD)
2 - 1' OR 1=1 %23
3 - 1" OR 1=1 %23
4 - admin'/ or 1=1#
5 - ?id=1' or "=" %23 | ?id=1' AND (SELECT 4216 FROM (SELECT(SLEEP(5)))Jlfk) AND 'Sulg'='Sulg | ?id=1' union select 1,database(),user(),4 %23
6 - ") or 1=1 # (In PWD)
7 - admin' or 1=1 #
8 - ' or '1'='1 (In User-Agent)
9 - ' or '1'='1 (In Referer)

10 - ' or 1=1 (In PHPSESSID)
11 - 'UNION SELECT 1,2,sleep(5)-- -
12 - ?id=')%20or%20sleep(5)%3d'-- -

Week 10 CTF

: MySQL server version for the right syntax to use near '1=1 --") LIMIT 0,1' at line 1

how to interpret this error?
Ans:
`'"""") LIMIT 0,1'` Over here the outside `'`s are the part of query. The inside set of `"`s represent our input. And the second `"` denotes our input. The `) LIMIT 0,1` is the part of the query.

https://charmed-azimuth-370.notion.site/Week-10-Final-WebApp-4f98cdcd9eb347fdbcd13a9ce07b1aa3