# <https://app.hackthebox.com/profile/1321956>

# Machine: BroScience Type: Medium

1)Ping to machine IP address

Text

Description automatically generated

TTL = 63. The particular machine is running on Linux

2)Conduct a Nmap scan

-Sv determine the version of the service running on port  
-V verbosity level  
-SC Scan with default NSE script

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Analyse

Text

Description automatically generated

3)I found a domain <https://broscience.htb>

The link will direct us to a body building page.

Graphical user interface, application, website

Description automatically generated

When I clicked at the latest entry, “Bench Press”. I noticed that the post is created by Administrator. Hence, there will be account’s username is “administrator”.

Graphical user interface, application, website

Description automatically generated

4)Using “dirsearch”, I’m able to have an overall look on the directories that lies on the website. “dirseach” will brute force directories and files in the webservers.

Text

Description automatically generated

5)We explore the “includes” directory. It contains .php files.

Graphical user interface, text, application

Description automatically generated

All the files are empty when clicked except for img.php.

An error is shown “ Error: Missing ‘path’ when img.php was clicked. To know the content of the other files, I need to export them using the curl method.

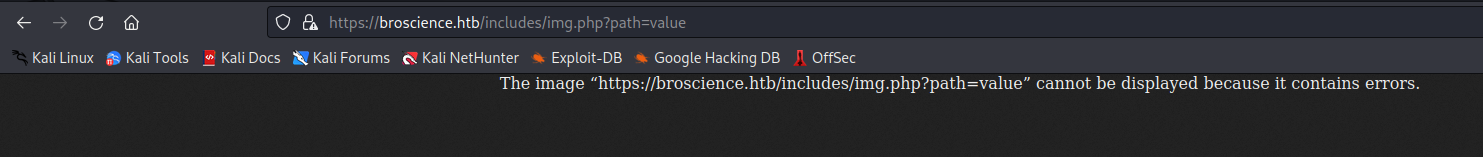
Graphical user interface, text, application, email

Description automatically generated

6)We add in argument to the path .



An error appeared.



Since the file path require a certain type of arguments, I shall test for Directory path traversal vulnerability. Directory path traversal is web security vulnerability that allows the attacker to read arbitrary files on the server running in the web application. Arbitrary files include application code and data, credential for backend system and sensitive operating files. Attacker can also write and modify the arbitrary files.

Images are loaded in the following format. <img src=”loadImage?filename = cart.png>.  
This display the image on the webpage. The image files themselves are stored in the following location, /var/www/images. Appending the requested filename to this base directory and uses a filesystem API will render the image. The application did not deter against file path traversal attack. Hence an attacker can request for an arbitrary such as the passwd file by using the below filepath https://shoppingWEbsite.com/loadImage?filename= ../../../etc/paswd.

By typing “../” the attacker move up to a parent directory by one. Therefore, the current position of the file path is /etc/passwd.

With this, I would exploit the passwd file using the mentioned technique.

<https://broscience.htb/includes/img.php?path=../../../../../> . An error appeared.

Error: Attack detected.

A screenshot of a computer

Description automatically generated

7)I decide to use URL encoding

https://broscience.htb/includes/img.php?path=...%2f..%2f..%2f..%2f..%2fetc..%2fpasswd

Graphical user interface, website

Description automatically generated

The reason I encode (“/”) to (‘%2f’) is to manipulate input parameters sent to the server. However the server will treat the encoded characters as a literal forward slash instead of a directory separator.

I further URL encode using a decoder in burpsuite. The double encoding allows me to bypass security controls that rely on blocking the forward slash character (“/”).

Graphical user interface, text, email

Description automatically generated

https://broscience.htb/includes/img.php?path=..5..%25%32%66..%25%32%66..%25%32%66..%25%32%66etc..%25%32%66passwd

I received another error.



8)Instead of writing the file path on the URL, I wrote it in the terminal using the “curl” method.   
The below is content of passwd file.

Text

Description automatically generated

9)By exploiting the directory path traversal vulnerability, I will be able to fetch files from the server.

I exported the source code of php files found in the “include”. I need to export them using the curl method.

For instance:

curl -k <https://broscience.htb/includes/img.php?path=..%25%32%66..%25%32%66..%25%32%66..%25%32%66..%25%32%66var%25%32%66www%25%32%66html%25%32%66includes%25%32%66utils.php> > utils.php

Repeat the process till all the files has been exported to our own machine. This would allow us to have a copy of the source code of each php file for easy reference.

10) I enumerated img.php and retrieved the source code of the file.

Text

Description automatically generated

Analyse the source code.

Text

Description automatically generated

11)I tried enumerate further by creating an account.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing graphical user interface

Description automatically generated

In search for the activation link,

12)In the register.php , I noticed that the activation link is made up of an url link and an activation code.

Text

Description automatically generatedText

Description automatically generated

However, I am still missing the activation code.

Graphical user interface

Description automatically generated

13)Explore utils.php

Text

Description automatically generated

Text

Description automatically generated

14)The cropped image above is the function that would generate the activation code when it is being called .

a variable $chars is a string of char contains uppercase and lowercase of the alphabets. It also contains the digit from 0 to 9. Srand is used to seed the random number created by a rand() function. For the time function will take the number from the converted value from the unix epoch. Hence srand will be seeding with a different value each time it is called. A for loop will be iterating 32 times. Each time it will called the function rand() with arguments 0 and the $char length . This will a random number between the 2 arguments and is used to select a character from the $char string and this character will be appended to the end of the activation code till the loop ends.

15)Deliberately creating an activation code to be appended to the activation link.

Extract the function generate\_activation\_code() and copy it editor and save it as activation.php. Copy the Epochj time value in the bracket srand();

Text

Description automatically generated

16)In the register.php, right click “inspect”. Choose register.php under file. Under the header extract the date from the response header.

Graphical user interface, text

Description automatically generated

17Place the date on the Epoch converter.

Graphical user interface, text, website

Description automatically generated

18)

Text

Description automatically generated

19)In the terminal call the activation.php to generate the activation code.

Text

Description automatically generated  
Text

Description automatically generated

20)Key in the code together with the URL

Graphical user interface, text, application

Description automatically generated

21)Account activated.

Graphical user interface, text, application

Description automatically generated

22)Login.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

23)Click on my profile . My user id is 6.

Graphical user interface, text, application

Description automatically generated

Parameter tampering . By changing the userid to 1 . I can see other users profile attributes. In this case, I can see the administrator’s account .

Graphical user interface, text

Description automatically generated

24)Enumerate the swap theme. It is indicated with the yellow highlight. I can see this feature after I logged on.



The function of the swap theme button is Change background color to grey.

A picture containing text, monitor, screenshot, screen

Description automatically generated

25)Lets look at swap\_theme.php

Text

Description automatically generated

Lets look back at util.php

Utils.php

Shape

Description automatically generated with low confidence

In the get\_theme\_(),Here we can take advantage of cookie deactivation so that we can inject a new cookie and call for a reverse shell to gain access.

A screenshot of a computer

Description automatically generated with medium confidence

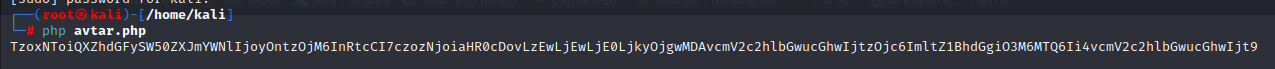
26)The imgPath here refers to the path of the image to store in the server. By using reverse shell, I’m able to do a shell.

27)Copy the codes from both Avatar and AvatarInterface classes and save it as avtar.php . Add in the filepath to the revshell.php.

Text

Description automatically generated

28)Launch payload.



Top of Form

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedBottom of Form

29)Copy the payload value to replace the value in “user-prefs”

30)Set up http server

Text

Description automatically generated

31) Http server gets notified. Hence, I am able to call for revershell.php

Text

Description automatically generated

32)Revshell.php   
Change the necessary IP and listener port to 9001.

Text

Description automatically generated

33)Set up a netcat listener.

Text

Description automatically generated

34)Call revshell.php



35)Received a shell

Text

Description automatically generated

36)Ls and changing to bash

Text

Description automatically generated

37)Enumerating database.

Viewing the database attributes

A screenshot of a computer screen

Description automatically generated with medium confidence

38)Login in to database by entering localhost , port number , database name , database user and database password

Text

Description automatically generated

Google postgresql commands

Graphical user interface, application

Description automatically generated

39)Display tables in database Broscience.

Text

Description automatically generated

40)Enumerating the users

A screenshot of a computer

Description automatically generated with medium confidence  
  
41)Copy the MD5 hashes and include NaCl at the back and save it as hash.

Graphical user interface, text

Description automatically generated

42)Cracking the hash against rockyou.txt. wordlist.

Graphical user interface, text

Description automatically generated

Text

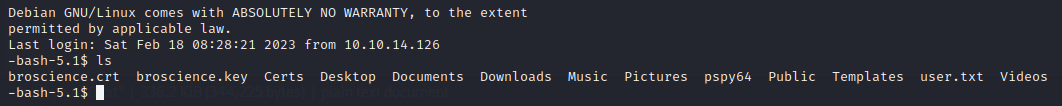
Description automatically generated

43)We found Bill password. Login as Bill

Text

Description automatically generated

44)ls



45)User.txt

Text

Description automatically generated

User.txt

68cf5883b7b91c471e81b781cdd48015

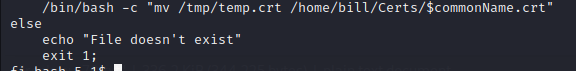
46)Reading the contents in renew.cert.sh

Text

Description automatically generated

Text

Description automatically generated



The above script checks if the SSL certificate has expired.

If it has expired , it will extract the important information from the certificate such as country,state,locality,organizationUnit and common name and generates a new self-signed certificate if necessary and stores the certificate in specific directory. Once the process is completed, the cert will moves to /tmp/temp.crt to $commonName.crt. With this try to derive a shell

Text

Description automatically generated

47)The $(chmod u+s /bin/bash) allow us to modify the permissions on /bin/bash and making it suid.

Text

Description automatically generated

48)Root.txt

f51bb2d1e1433b500174731dcd16799a

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

A screenshot of a video game

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated

# References:

<https://book.hacktricks.xyz/pentesting-web/nosql-injection>

<https://gtfobins.github.io/gtfobins/docker/#shell>

<https://gtfobins.github.io/>

<https://www.techrepublic.com/article/how-to-quickly-give-users-sudo-privileges-in-linux/#:~:text=To%20use%20this%20tool%2C%20you,%2Fsudoers%20file%20for%20editing).&text=Save%20and%20close%20the%20file,full%20range%20of%20sudo%20privileges>.

<https://book.hacktricks.xyz/pentesting-web/nosql-injection>

<https://www.imperva.com/learn/application-security/nosql-injection/#:~:text=The%20main%20differences%20between%20NoSQL,don't%20use%20standardized%20languages>.

<https://github.com/danielmiessler/SecLists>

<https://portswigger.net/web-security/file-path-traversal>

<https://github.com/pentestmonkey/php-reverse-shell>