

Attacking Web application with Ffuf

Here is the link that shows completion of this module

<https://academy.hackthebox.com/achievement/1917469/54>

Ffuf is a fast web fuzzing tool written in GO designed to discover hidden files,directories,parameters,subdomains and virtual hosts.

Ffuf is fast,flexible and scriptable as it sends requests suing wordlists,allows detecting valid responses of HTTP and discovers pages,endpoints,files or input parameters.

Fuzzing is a technique where automated tools sends may requests with different values (from a wordlist) to identify hidden or vulnerable parts of a web application.

To install ffuf use the command **apt install ffuf -y**.

When you run **ffuf -h** there are two main options **-w** for wordlists and **-u** for url. The wordlist for this will be using **/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt**

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target(s): 94.237.51.33:38615

Life Left: 62 minute(s)

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

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://94.237.51.33:38615/FUZZ -m 200,302,404,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103,1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,1116,1117,1118,1119,1120,1121,1122,1123,1124,1125,1126,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,1139,1140,1141,1142,1143,1144,1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,1155,1156,1157,1158,1159,1160,1161,1162,1163,1164,1165,1166,1167,1168,1169,1170,1171,1172,1173,1174,1175,1176,1177,1178,1179,1180,1181,1182,1183,1184,1185,1186,1187,1188,1189,1190,1191,1192,1193,1194,1195,1196,1197,1198,1199,1200,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,1211,1212,1213,1214,1215,1216,1217,1218,1219,1220,1221,1222,1223,1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,1236,1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248,1249,1250,1251,1252,1253,1254,1255,1256,1257,1258,1259,1260,1261,1262,1263,1264,1265,1266,1267,1268,1269,1270,1271,1272,1273,1274,1275,1276,1277,1278,1279,1280,1281,1282,1283,1284,1285,1286,1287,1288,1289,1290,1291,1292,1293,1294,1295,1296,1297,1298,1299,1300,1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321,1322,1323,1324,1325,1326,1327,1328,1329,1330,1331,1332,1333,1334,1335,1336,1337,1338,1339,1340,1341,1342,1343,1344,1345,1346,1347,1348,1349,1350,1351,1352,1353,1354,1355,1356,1357,1358,1359,1360,1361,1362,1363,1364,1365,1366,1367,1368,1369,1370,1371,1372,1373,1374,1375,1376,1377,1378,1379,1380,1381,1382,1383,1384,1385,1386,1387,1388,1389,1390,1391,1392,1393,1394,1395,1396,1397,1398,1399,1400,1401,1402,1403,1404,14
```

The discoverable endpoints here are /forum and /blog.

Page Fuzzing

When we visits the above endpoints returns empty so will utilize web fuzzing to see if the directories contains any hidden pages. Before we start we need to identify what types of pages the website's uses like .html,.aspx,.php or something else.

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/web-extensions.txt:FUZZ -u http://94.237.51.33:38615/blog/indexFUZZ
```

Here we used index as this file can be found in many websites and after this we discovered that the websites run on php as it gave a response of 200.

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -u http://94.237.60.55:56366/blog/FUZZ.php
```

The output of this shows two file home and index but index has a size of 0 indicating its empty and home has a size of 1046.

This suggests that:

□ <http://94.237.60.55:56366/blog/home.php> is a valid and accessible page that is likely not a default/empty page, unlike index.php.

```
home [Status: 403, Size: 280, Words: 20, Lines: 10, Duration: 4269ms]
index [Status: 200, Size: 1046, Words: 438, Lines: 58, Duration: 4268ms]
      [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 4273ms]
      [Status: 403, Size: 280, Words: 20, Lines: 10, Duration: 151ms]
```

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?

When you visit above page you get flag.

Recursive Fuzzing

In above fuzzing we found dozens of directories each with their own subdirectories and files and this was taking time to fuzz, the idea of recursive is to automate this.

When we scan recursively, it automatically starts another scan under any newly identified directories that may have on their pages until it has fuzzed the main website and all of its subdirectories.

In ffuf we can enable recursive scanning with the **-recursion** flag and we can specify the depth with the **-recursion-depth** flag.

If we specify **recursion-depth 1**, it will only fuzz the main directories and their direct subdirectories and we can specify our extension with **-e .php** also we can add **-v** flag to output full urls.

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?

HTB{fuzz1n6_7h3_w3b!}

To get this I ran the following commands

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -u http://94.237.51.33:40263/FUZZ -recursion -recursion-depth 1 -e .php -v
```

```
#ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -u http://94.237.51.33:40263/FUZZ -recursion -recursion-depth 1 -e .php -v
When we visit the above endpoints returns empty so will utilize web fuzzing to see if the
```

I found several files together with their directories and one that found interesting was flag.php.

```
[Status: 200, Size: 21, Words: 1, Lines: 1, Duration: 155ms]
| URL | http://94.237.51.33:40263/forum/flag.php
* FUZZ: flag.php
```

DNS Records

At first when we visited the page under the /blog we got message saying Admin panel moved to academy.htb. This is because the website is not publically accessible by anyone but local websites within HTB.

When you access academy.htb, your browser (or any tool) needs to resolve the domain name to an IP address.

Because academy.htb is not a public domain (like google.com), public DNS servers such as 8.8.8.8 don't know how to resolve it. That's why you must manually tell your system what IP to associate with academy.htb by editing your /etc/hosts file.

```
sudo sh -c 'echo "SERVER_IP academy.htb" >> /etc/hosts'
```

This command adds academy.htb to /etc/hosts, so now, when your browser or a tool like curl tries to access academy.htb, your computer will redirect the request to server_ip instead of asking the internet.

In case of subdomains, we would be simply checking different websites to see if they exist by checking if they have public DNS record that would redirect us to a working server IP using a wordlist and target.

In this case our target will be inlanefreight.com and will be using ffuf and place FUZZ keyword in the place of subdomains.

Quiz:

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

Vhost Fuzzing

The key difference between a Vhost and a subdomain is that vhost is basically a 'subdomain' served on the same server and has the same IP such that a single IP could be serving two or more different websites.

Subdomain fuzzing targets public DNS records (e.g., support.example.com).

VHost fuzzing targets *hidden* subdomains served on the same IP via Host: header manipulation it does **not** require public DNS resolution.

FFUF is a perfect tool for this using the -H flag.

```
#ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://94.237.51.33:58472/ -H 'Host:FUZZ.academy.htb' -s 200 -s 900
```

When this command is run it returns the same result status:200,size 900 meaning the server is giving a default response for unknown hosts. But if one or more VHosts return different response sizes, that's a sign they exist and are configured, those are the hidden VHosts you're trying to discover.

When doing fuzzing of vhost we can provide an option to match or filter out specific HTTP code, response size or amount of words and we can see this through **ffuf -h**.

Answer the question(s) below to complete this Section and earn cubes!

Target(s): 94.237.51.33:58472

Life Left: 57 minute(s)

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

```
#ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://94.237.51.33:58472/ -H 'Host: FUZZ.academy.htb' -fs 986

When doing fuzzing of vhost we can provide option to match or filter out specific HTTP
code ,response size or amount of words and we can see this through ffuf -h.
Answer the question(s) below to complete this Section and earn cubes!
Target(s): 94.237.51.33:58472
Life Left: 57 minute(s)

v2.1.0-dev

:: Method      : GET
:: URL         : http://94.237.51.33:58472/
:: Wordlist     : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
:: Header      : Host: FUZZ.academy.htb
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Filter      : Response size: 986

admin      [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 162ms]
test      [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 162ms]
:: Progress: [4989/4989] :: Job [1/1] :: 253 req/sec :: Duration: [0:00:22] :: Errors: 0 ::
```

We get two admin and test and answer was test.academy.htb

Parameter Fuzzing – GET

This technique is used to discover hidden GET parameters that may trigger interesting behavior like debugging, authentication bypass, or functionality exposure even if they're not shown in the page or source code.

ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.academy.htb:PORT/admin/admin.php?FUZZ=key -fs xxx

Target(s): 94.237.58.50:56850

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

```
#ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.academy.htb:58754/admin/admin.php?FUZZ=key -fs 798
```

```
user      [Status: 200, Size: 783, Words: 221, Lines: 54, Duration: 160ms]
:: Progress: [6453/6453] :: Job [1/1] :: 73 req/sec :: Duration: [0:00:32] :: Errors: 0 ::
```

user

POST

To fuzz the data field with ffuf, we can use the -d flag, as we saw previously in the output of ffuf -h. We also have to add -X POST to send POST requests.

ffuf -w /opt/useful/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.academy.htb:PORT/admin/admin.php -X POST -d 'FUZZ=key' -H 'Content-Type: application/x-www-form-urlencoded' -fs xxx

As we can see this time, we got a couple of hits, the same one we got when fuzzing GET and another parameter, which is id. Let's see what we get if we send a POST request with the id parameter. We can do that with curl, as follows:

```
curl http://admin.academy.htb:PORT/admin/admin.php -X POST -d 'id=key' -H 'Content-Type: application/x-www-form-urlencoded'
```

Value Fuzzing

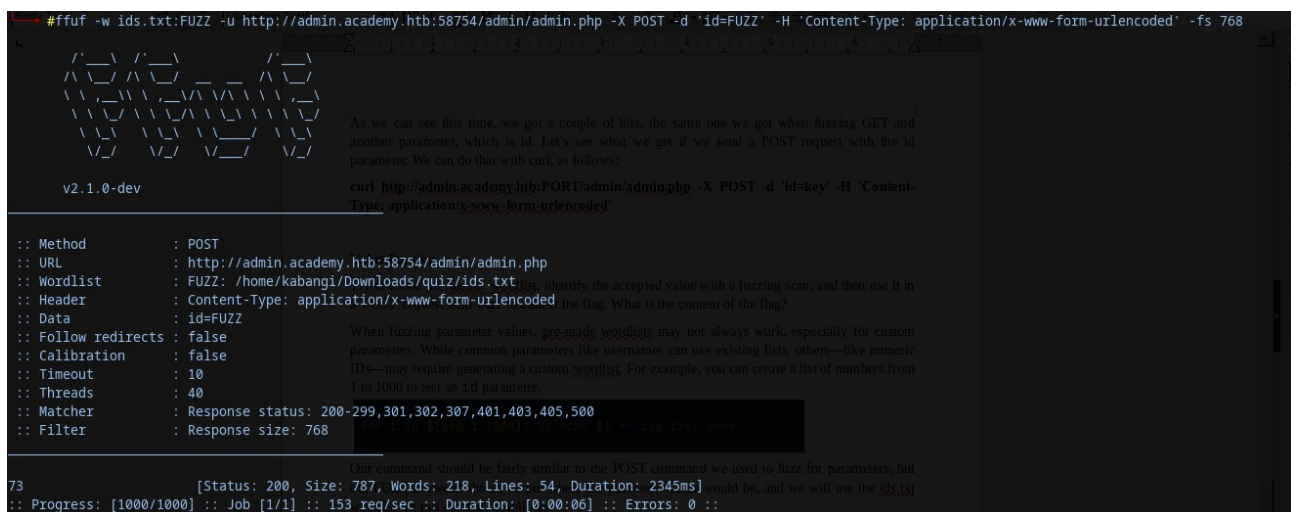
Try to create the 'ids.txt' wordlist, identify the accepted value with a fuzzing scan, and then use it in a 'POST' request with 'curl' to collect the flag. What is the content of the flag?

When fuzzing parameter values, pre-made wordlists may not always work, especially for custom parameters. While common parameters like usernames can use existing lists, others—like numeric IDs—may require generating a custom wordlist. For example, you can create a list of numbers from 1 to 1000 to test an id parameter.

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

Our command should be fairly similar to the POST command we used to fuzz for parameters, but our FUZZ keyword should be put where the parameter value would be, and we will use the ids.txt wordlist we just created, as follows:

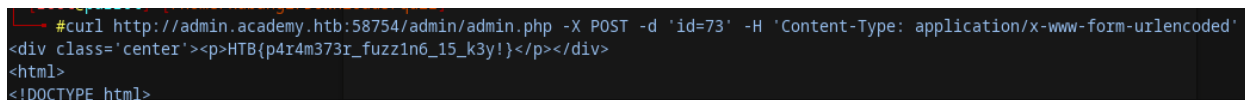
```
#ffuf -w ids.txt:FUZZ -u http://admin.academy.htb:58754/admin/admin.php -X POST -d 'id=FUZZ' -H 'Content-Type: application/x-www-form-urlencoded' -fs 768
```



The screenshot shows the output of the ffuf command. It displays a list of hits, with the first hit being 'id=73'. The output also shows the command used and the response status (200) and size (787). The command is: #ffuf -w ids.txt:FUZZ -u http://admin.academy.htb:58754/admin/admin.php -X POST -d 'id=FUZZ' -H 'Content-Type: application/x-www-form-urlencoded' -fs 768. The response is: [Status: 200, Size: 787, Words: 218, Lines: 54, Duration: 2345ms]. The progress bar shows 1000/1000 jobs completed.

We see that we get a hit right away. We can finally send another POST request using curl, as we did in the previous section, use the id value we just found, and collect the flag.

```
#curl http://admin.academy.htb:58754/admin/admin.php -X POST -d 'id=73' -H 'Content-Type: application/x-www-form-urlencoded'
```



The screenshot shows the output of the curl command. It displays the HTML response, which includes the flag HTB{p4r4m373r_fuzz1n6_15_k3y!}. The command is: #curl http://admin.academy.htb:58754/admin/admin.php -X POST -d 'id=73' -H 'Content-Type: application/x-www-form-urlencoded'. The response is: <div class='center'><p>HTB{p4r4m373r_fuzz1n6_15_k3y!}</p></div><html><!DOCTYPE html>

Skills Assessment – Web Fuzzing

You are given an online academy's IP address but have no further information about their website. As the first step of conducting a Penetration Test, you are expected to locate all pages and domains linked to their IP to enumerate the IP and domains properly.

Finally, you should do some fuzzing on pages you identify to see if any of them has any parameters that can be interacted with. If you do find active parameters, see if you can retrieve any data from them.

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target(s): 94.237.51.146:48510

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)

```
root@kali:~/# fuzzie -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://admin.academy.htb:48510/ -H 'Host:FUZZ.academy.htb' -fs 985

Answer the question(s) below to complete this Section and earn cubes!

v2.1.0-dev
You can identify? (Only write the sub-domain name)

:: Method      : GET
:: URL         : http://admin.academy.htb:48510/
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
:: Header      : Host: FUZZ.academy.htb
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Filter      : Response size: 985

extensions: .css, .js, .php, .png, .txt, .xml, .xsl, .yml, .zip

test [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 162ms]
archive [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 150ms]
faculty [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 168ms]
:: Progress: [4989/4989] :: Job [1/1] :: 111 req/sec :: Duration: [0:00:22] :: Errors: 0 ::
```

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

```
#ffuf -w /usr/share/seclists/Discovery/Web-Content/web-extensions.txt:FUZZ -u http://faculty.academy.htb:48510/indexFUZZ
```

This will be repeated for every identified subdomain above as for this is for test.academy.htb

The identified extensions were .php,.phps,

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

I tried all the subdomains above and the one that was promising was `faculty.academy.htb` and had a directory called `courses`, I ran the following commands and got `http://faculty.academy.htb:31912/courses/linux-security.php7`

```
#ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://faculty.academy.htb:31912/courses/FUZZ -recursion -recursion-depth 1 -e .php,.phps,.php7 -v -fs 287 -c -ic -t 1000
```

```
flag. What is the content of the flag?
[Status: 200, Size: 774, Words: 223, Lines: 53, Duration: 151ms]
| URL | http://faculty.academy.htb:31912/courses/linux-security.php7
* FUZZ: linux-security.php7
```


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

```
#ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://faculty.academy.htb:58912/courses/linux-security.php7 -X POST -d 'FUZZ=key' -H 'Content-Type: application/x-www-form-urlencoded' -fs 774
```

Method : POST
URL : http://faculty.academy.htb:58912/courses/linux-security.php7
Wordlist : FUZZ: /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt
Header : Content-Type: application/x-www-form-urlencoded
Data : FUZZ=key
Follow redirects : false
Calibration : false
Timeout : 10
Threads : 40
Matcher : Response status: 200-299,301,302,307,401,403,405,500
Filter : Response size: 774

user [Status: 200, Size: 780, Words: 223, Lines: 53, Duration: 173ms]
username [Status: 200, Size: 781, Words: 223, Lines: 53, Duration: 168ms]
:: Progress: [6453/6453] :: Job [1/1] :: 245 req/sec :: Duration: [0:00:58] :: Errors: 0 ::

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

```
#ffuf -w /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt:FUZZ -u http://faculty.academy.htb:58912/courses/linux-security.php7 -X POST -d 'username=FUZZ' -H 'Content-Type: application/x-www-form-urlencoded' -fs 781
```

This produced several usernames and replaced username=FUZZ with username=harry and used curl to find the flag

```
#curl http://faculty.academy.htb:58912/courses/linux-security.php7 -X POST -d 'username=harry' -H 'Content-Type: application/x-www-form-urlencoded' -fs 781
```

```
<div class="center"><p>HTB{w3b_fuzz1n6_m4573r}</p></div>
```