

Value Fuzzing

After fuzzing a working parameter, we now have to fuzz the correct value that would return the **flag** content we need. This section will discuss fuzzing for parameter values, which should be fairly similar to fuzzing for parameters, once we develop our wordlist.

Custom Wordlist

When it comes to fuzzing parameter values, we may not always find a pre-made wordlist that would work for us, as each parameter would expect a certain type of value.

For some parameters, like usernames, we can find a pre-made wordlist for potential usernames, or we may create our own based on users that may potentially be using the website. For such cases, we can look for various wordlists under the **seclists** directory and try to find one that may contain values matching the parameter we are targeting. In other cases, like custom parameters, we may have to develop our own wordlist. In this case, we can guess that the **id** parameter can accept a number input of some sort. These ids can be in a custom format, or can be sequential, like from 1-1000 or 1-1000000, and so on. We'll start with a wordlist containing all numbers from 1-1000.

There are many ways to create this wordlist, from manually typing the IDs in a file, or scripting it using Bash or Python. The simplest way is to use the following command in Bash that writes all numbers from 1-1000 to a file:

```
MichaelLuka@htb[/htb]$ for i in $(seq 1 1000); do echo $i >> ids.txt; done
```

Once we run our command, we should have our wordlist ready:

```
MichaelLuka@htb[/htb]$ cat ids.txt

1
2
3
4
5
6
<...SNIP...>
```

Now we can move on to fuzzing for values.

Value Fuzzing

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:

```
MichaelLuka@htb[/htb]$ ffuf -w ids.txt:FUZZ -u http://admin.academy.htb:PORT/admin/admin.php -X POST -d 'id=FUZZ' -H 'Cont

      /\_/\  /\_/\  /\_/\
     /\_/\  /\_/\  /\_/\
    /\_/\  /\_/\  /\_/\
   /\_/\  /\_/\  /\_/\
  /\_/\  /\_/\  /\_/\
 /\_/\  /\_/\  /\_/\
/_/\    _/\    _/\    _/\

v1.0.2

-----

:: Method      : POST
:: URL         : http://admin.academy.htb:30794/admin/admin.php
:: Header      : Content-Type: application/x-www-form-urlencoded
:: Data        : id=FUZZ
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403
:: Filter      : Response size: xxx

-----

<...SNIP...> [Status: xxx, Size: xxx, Words: xxx, Lines: xxx]
```

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.

Start Instance

1 / 1 spawns left


Waiting to start...

Questions

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

Cheat Sheet

Target: [Click here to spawn the target system!](#)

+ 1

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?

Submit your answer here...

 Submit

 Hint

 Previous

Next 

 Cheat Sheet


 Go to Questions

Table of Contents

Introduction


Introduction

✓


Web Fuzzing

✓


Basic Fuzzing

 Directory Fuzzing

✓

 Page Fuzzing

✓


 Recursive Fuzzing

✓

Domain Fuzzing

DNS Records


✓

 Sub-domain Fuzzing

✓


Vhost Fuzzing

✓

 Filtering Results

✓

Parameter Fuzzing

 Parameter Fuzzing - GET


✓

Parameter Fuzzing - POST

✓


 Value Fuzzing

Skills Assessment

 Skills Assessment - Web Fuzzing

My Workstation

OFFLINE

 Start Instance

1 / 1 spawns left