# Directory Fuzzing

Now that we understand the concept of Web Fuzzing and know our wordlist, we should be ready to start using `ffuf` to find website directories.

## Ffuf

`Ffuf` is pre-installed on your PwnBox instance. If you want to use it on your own machine, you can either use "`apt install ffuf -y`" or download it and use it from its GitHub Repo. As a new user of this tool, we will start by issuing the `ffuf -h` command to see how the tools can be used:

```
MichaelLuka@htb[/htb]$ ffuf -h

HTTP OPTIONS:
  -H                Header `"Name: Value"`, separated by colon. Multiple -H flags are accepted.
  -X                HTTP method to use (default: GET)
  -b                Cookie data `"NAME1=VALUE1; NAME2=VALUE2"` for copy as curl functionality.
  -d                POST data
  -recursion        Scan recursively. Only FUZZ keyword is supported, and URL (-u) has to end in it. (default: false)
  -recursion-depth Maximum recursion depth. (default: 0)
  -u                Target URL
...SNIP...

MATCHER OPTIONS:
  -mc               Match HTTP status codes, or "all" for everything. (default: 200,204,301,302,307,401,403)
  -ms               Match HTTP response size
...SNIP...

FILTER OPTIONS:
  -fc               Filter HTTP status codes from response. Comma separated list of codes and ranges
  -fs               Filter HTTP response size. Comma separated list of sizes and ranges
...SNIP...

INPUT OPTIONS:
...SNIP...
  -w                Wordlist file path and (optional) keyword separated by colon. eg. '/path/to/wordlist:KEYWORD'

OUTPUT OPTIONS:
  -o                Write output to file
...SNIP...

EXAMPLE USAGE:
  Fuzz file paths from wordlist.txt, match all responses but filter out those with content-size 42.
  Colored, verbose output.
    ffuf -w wordlist.txt -u https://example.org/FUZZ -mc all -fs 42 -c -v
...SNIP...
```

As we can see, the `help` output is quite large, so we only kept the options that may become relevant for us in this module.

## Directory Fuzzing

As we can see from the example above, the main two options are `-w` for wordlists and `-u` for the URL. We can assign a keyword to a wordlist to refer to it where we want to fuzz. For example, we can pick our wordlist and assign the keyword `FUZZ` to it by adding `:FUZZ` after it:

```
MichaelLuka@htb[/htb]$ ffuf -w /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ
```

Next, as we want to be fuzzing for web directories, we can place the `FUZZ` keyword where the directory would be within our URL, with:

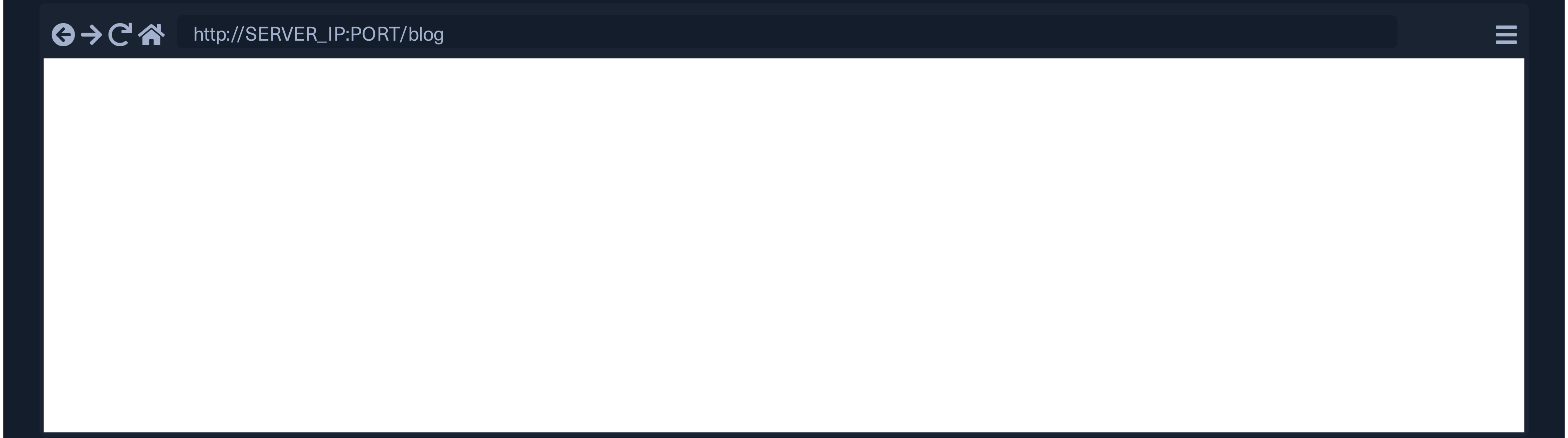
```
MichaelLuka@htb[/htb]$ ffuf -w <SNIP> -u http://SERVER_IP:PORT/FUZZ
```

Now, let's start our target in the question below and run our final command on it:

```
[/htb]$ ffuf -w /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://SERVER_IP:PORT/FUZZ


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

it
_________________________________

      : GET
      : http://SERVER_IP:PORT/FUZZ
      : FUZZ: /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt
rects : false
      : false
      : 10
      : 40
      : Response status: 200,204,301,302,307,401,403
_________________________________


         [Status: 301, Size: 326, Words: 20, Lines: 10]
7651/87651] :: Job [1/1] :: 9739 req/sec :: Duration: [0:00:09] :: Errors: 0 ::
```

We see that `ffuf` tested for almost 90k URLs in less than 10 seconds. This speed may vary depending on your internet speed and ping if you used `ffuf` on your machine, but it should still be extremely fast.

We can even make it go faster if we are in a hurry by increasing the number of threads to 200, for example, with `-t 200`, but this is not recommended, especially when used on a remote site, as it may disrupt it, and cause a `Denial of Service`, or bring down your internet connection in severe cases. We do get a couple of hits, and we can visit one of them to verify that it exists:



We get an empty page, indicating that the directory does not have a dedicated page, but also shows that we do not have access to it, as we do not get an HTTP code `404 Not Found` or `403 Access Denied`. In the next section, we will look for pages under this directory to see whether it is really empty or has hidden files and pages.

Start Instance

1 / 1 spawns left

Waiting to start...

## Questions

Cheat Sheet

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

Target: Click here to spawn the target system!

+ 0 In addition to the directory we found above, there is another directory that can be found. What is it?

Submit your answer here...

Submit

Hint

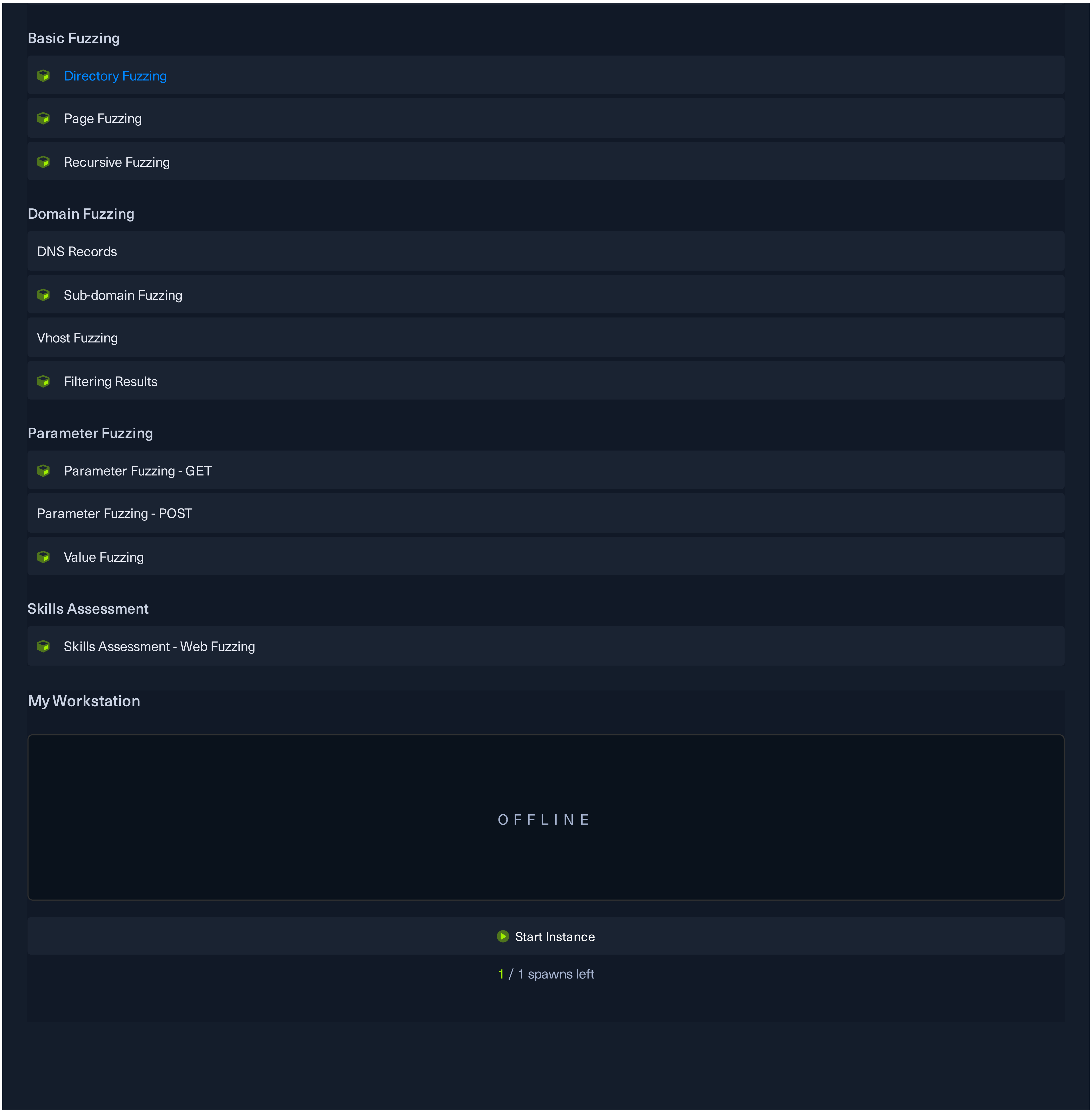
← Previous Next →

Cheat Sheet

? Go to Questions

Table of Contents

Introduction



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