Page 4 / Page Fuzzing

Page Fuzzing

We now understand the basic use of ffuf through the utilization of wordlists and keywords. Next, we will learn how to locate pages.

Note: We can spawn the same target from the previous section for this section's examples as well.

Extension Fuzzing

In the previous section, we found that we had access to /blog, but the directory returned an empty page, and we cannot manually locate any links or pages. So, we will once again utilize web fuzzing to see if the directory contains any hidden pages. However, before we start, we must find out what types of pages the website uses, like .html, .aspx, .php, or something else.

One common way to identify that is by finding the server type through the HTTP response headers and guessing the extension. For example, if the server is apache, then it may be .php, or if it was IIS, then it could be .asp or .aspx, and so on. This method is not very practical, though. So, we will again utilize ffuf to fuzz the extension, similar to how we fuzzed for directories. Instead of placing the FUZZ keyword where the directory name would be, we would place it where the extension would be .FUZZ, and use a wordlist for common extensions. We can utilize the following wordlist in SecLists for extensions:

```
• • •
MisaelMacias@htb[/htb]$ ffuf -w /opt/useful/seclists/Discovery/Web-Content/web-extensions.txt:FUZZ <SNIP>
```

Before we start fuzzing, we must specify which file that extension would be at the end of! We can always use two wordlists and have a unique keyword for each, and then do FUZZ_1.FUZZ_2 to fuzz for both. However, there is one file we can always find in most websites, which is index.*, so we will use it as our file and fuzz extensions on it.

Note: The wordlist we chose already contains a dot (.), so we will not have to add the dot after "index" in our fuzzing.

Now, we can rerun our command, carefully placing our FUZZ keyword where the extension would be after index:

```
http://SERVER_IP:PORT/blog/indexFUZZ
    Wordlist :
Follow redirects :
Calibration :
                                   FUZZ: /opt/useful/seclists/Discovery/Web-Content/web-extensions.txt false false
     Timeout
     Threads
.php [Status: 280, Size: 0, Words: 1, Lines: 1]
.phps [Status: 403, Size: 283, Words: 20, Lines: 10]
:: Progress: [39/39] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

We do get a couple of hits, but only .php gives us a response with code 200. Great! We now know that this website runs on PHP to start fuzzing

Page Fuzzing

We will now use the same concept of keywords we've been using with ffuf, use .php as the extension, place our FUZZ keyword where the filename should be, and use the same wordlist we used for fuzzing directories:

```
• • •
                                                                    Page Fuzzing
 MisaelMacias@htb[/htb]$ ffuf -w /opt/useful/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http:/
                             . Ge:
http://SERVER_IP:PORT/blog/FUZZ.php
: FUZZ: /opt/useful/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
      Follow redirects : false
Calibration : false
Timeout : 10
```

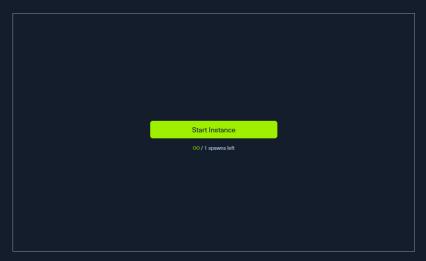




We get a couple of hits; both have an HTTP code 200, meaning we can access them. index.php has a size of 0, indicating that it is an empty page, while the other does not, which means that it has content. We can visit any of these pages to verify this:







Waiting to start...

