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 one. 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:

```
MichaelLuka@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`:

```
MichaelLuka@htb[/htb]$ ffuf -w /opt/useful/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ -u http://SERVER_IP:PORT


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

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://SERVER_IP:PORT/blog/indexFUZZ
 :: Wordlist         : FUZZ: /opt/useful/SecLists/Discovery/Web-Content/web-extensions.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 5
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

.php                    [Status: 200, 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 ::
```

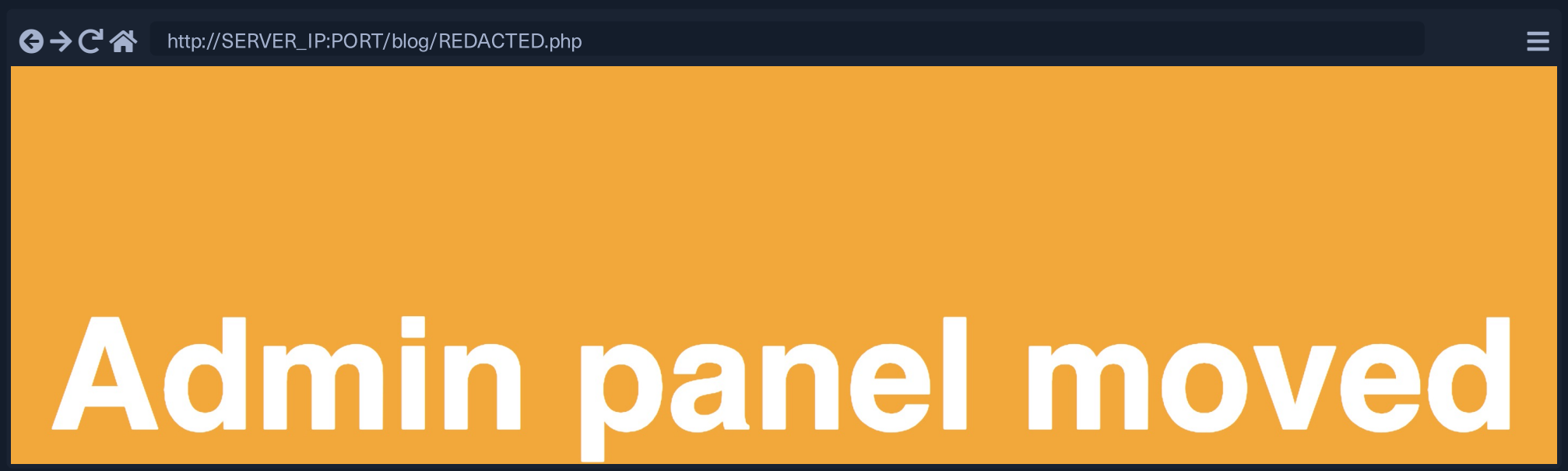
We do get a couple of hits, but only `.php` gives us a response with code `200`. Great! We now know that this website runs on `PHP` to start fuzzing for `PHP` files.

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:

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


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

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://SERVER_IP:PORT/blog/FUZZ.php
 :: Wordlist         : FUZZ: /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

index                   [Status: 200, Size: 0, Words: 1, Lines: 1]
REDACTED                [Status: 200, Size: 465, Words: 42, Lines: 15]
:: Progress: [87651/87651] :: Job [1/1] :: 5843 req/sec :: Duration: [0:00:15] :: Errors: 0 ::
```

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:



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!

+ 1 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?

Submit your answer here...

Submit

Hint

Previous Next

Cheat Sheet

Go to Questions

Table of Contents

My Workstation

OFFLINE

Start Instance

1 / 1 spawns left