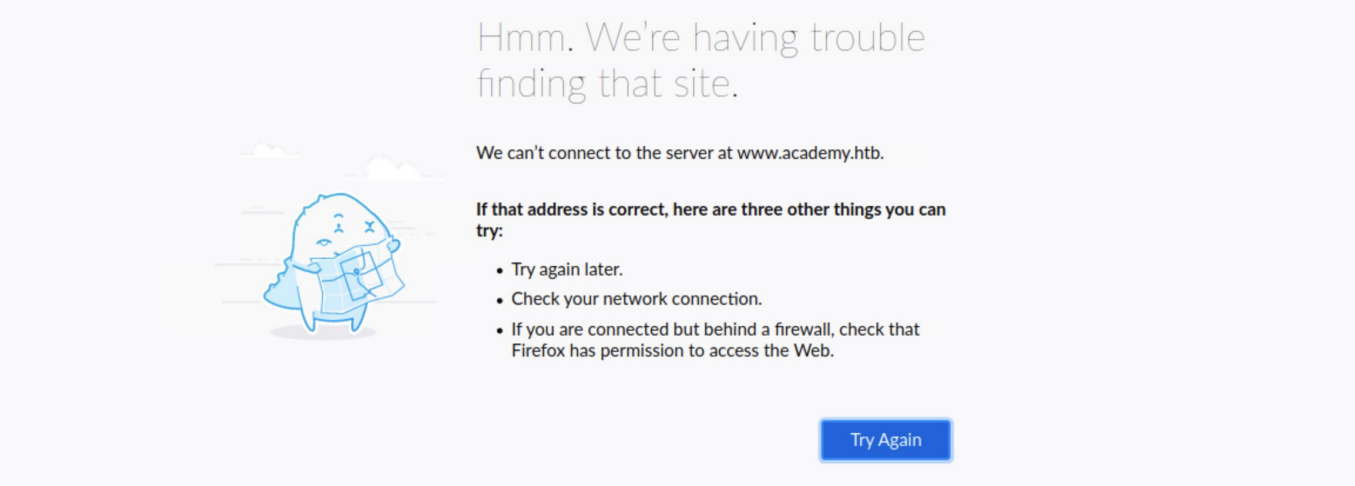
Domain Fuzzing

DNS Records

Once we accessed the page under `/blog`, we got a message saying `Admin panel moved to academy.htb`. If we visit the website in our browser, we get `can't connect to the server at www.academy.htb`:



This is because the exercises we do are not public websites that can be accessed by anyone but local websites within HTB. Browsers only understand how to go to IPs, and if we provide them with a URL, they try to map the URL to an IP by looking into the local `/etc/hosts` file and the public DNS `Domain Name System`. If the URL is not in either, it would not know how to connect to it.

If we visit the IP directly, the browser goes to that IP directly and knows how to connect to it. But in this case, we tell it to go to `academy.htb`, so it looks into the local `/etc/hosts` file and doesn't find any mention of it. It asks the public DNS about it (such as Google's DNS `8.8.8.8`) and does not find any mention of it, since it is not a public website, and eventually fails to connect. So, to connect to `academy.htb`, we would have to add it to our `/etc/hosts` file. We can achieve that with the following command:

mayala@htb[/htb] `$ sudo sh -c 'echo "SERVER_IP academy.htb" >> /etc/hosts'`

Now we can visit the website (don't forget to add the PORT in the URL) and see that we can reach the website:

However, we get the same website we got when we visit the IP directly, so `academy.htb` is the same domain we have been testing so far. We can verify that by visiting `/blog/index.php`, and see that we can access the page.

When we run our tests on this IP, we did not find anything about `admin` or panels, even when we did a full `recursive` scan on our target. `So, in this case, we start looking for sub-domains under '*.academy.htb' and see if we find anything, which is what we will attempt in the next section.`

Sub-domain Fuzzing

In this section, we will learn how to use `ffuf` to identify sub-domains (i.e., `*.website.com`) for any website.

Sub-domains

A sub-domain is any website underlying another domain. For example, `https://photos.google.com` is the `photos` sub-domain of `google.com`.

In this case, we are simply checking different websites to see if they exist by checking if they have a public DNS record that would redirect us to a working server IP. So, let's run a scan and see if we get any hits. Before we can start our scan, we need two things:

- A `wordlist`
- A `target`

Luckily for us, in the `SecLists` repo, there is a specific section for sub-domain wordlists, consisting of common words usually used for sub-domains. We can find it in `/opt/useful/SecLists/Discovery/DNS/`. In our case, we would be using a shorter

wordlist, which is `subdomains-top1million-5000.txt`. If we want to extend our scan, we can pick a larger list.

As for our target, we will use `inlanefreight.com` as our target and run our scan on it. Let us use `ffuf` and place the `FUZZ` keyword in the place of sub-domains, and see if we get any hits:

```
mayala@htb[/htb] $ ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u https://FUZZ.inlanefreight.com/ /'___\ /'___\ /'___\ /\ \__/ /\ \__/ __ __ /\ \__/ \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\ \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/ \ \_\ \ \_\ \ \____/ \ \_\ \/_/ \/_/ \/___/ \/_/ v1.1.0-git _____ :: Method : GET :: URL : https://FUZZ.inlanefreight.com/ :: Wordlist : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt :: Follow redirects : false :: Calibration : false :: Timeout : 10 :: Threads : 40 :: Matcher : Response status: 200,204,301,302,307,401,403,405,500 _____ [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 381ms] * FUZZ: support [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 385ms] * FUZZ: ns3 [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 402ms] * FUZZ: blog [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 180ms] * FUZZ: my [Status: 200, Size: 22266, Words: 2903, Lines: 316, Duration: 589ms] * FUZZ: www <...SNIP...>
```

We see that we do get a few hits back. Now, we can try running the same thing on `academy.htb` and see if we get any hits back:

```
mayala@htb[/htb] $ ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://FUZZ.academy.htb/ /'___\ /'___\ /'___\ /\ \__/ /\ \__/ __ __ /\ \__/ \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\ \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/ \ \_\ \ \_\ \ \____/ \ \_\ \/_/ \/_/ \/___/ \/_/ v1.1.0-git _____ :: Method : GET :: URL : https://FUZZ.academy.htb/ :: Wordlist : FUZZ: /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt :: Follow redirects : false :: Calibration : false :: Timeout : 10 :: Threads : 40 :: Matcher : Response status: 200,204,301,302,307,401,403 _____ :: Progress: [4997/4997] :: Job [1/1] :: 131 req/sec :: Duration: [0:00:38] :: Errors: 4997 ::
```

We see that we do not get any hits back. Does this mean that there are no sub-domain under `academy.htb`? - No.

This means that there are no `public` sub-domains under `academy.htb`, as it does not have a public DNS record, as previously mentioned. Even though we did add `academy.htb` to

our `/etc/hosts` file, we only added the main domain, so when `ffuf` is looking for other sub-domains, it will not find them in `/etc/hosts`, and will ask the public DNS, which obviously will not have them.

Vhost Fuzzing

As we saw in the previous section, we were able to fuzz public sub-domains using public DNS records. However, when it came to fuzzing sub-domains that do not have a public DNS record or sub-domains under websites that are not public, we could not use the same method. In this section, we will learn how to do that with `Vhost Fuzzing`.

Vhosts vs. Sub-domains

The key difference between VHosts and sub-domains is that a VHost is basically a 'sub-domain' served on the same server and has the same IP, such that a single IP could be serving two or more different websites.

`VHosts may or may not have public DNS records.`

In many cases, many websites would actually have sub-domains that are not public and will not publish them in public DNS records, and hence if we visit them in a browser, we would fail to connect, as the public DNS would not know their IP. Once again, if we use the `sub-domain fuzzing`, we would only be able to identify public sub-domains but will not identify any sub-domains that are not public.

This is where we utilize `VHosts Fuzzing` on an IP we already have. We will run a scan and test for scans on the same IP, and then we will be able to identify both public and non-public sub-domains and VHosts.

Vhosts Fuzzing

To scan for VHosts, without manually adding the entire wordlist to our `/etc/hosts`, we will be fuzzing HTTP headers, specifically the `Host:` header. To do that, we can use the `-H` flag to specify a header and will use the `FUZZ` keyword within it, as follows:

mayala@htb[/htb] `$ ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://academy.htb:PORT/ -H 'Host: FUZZ.academy.htb' /'___\ /'___\ /'___\ /\ __/ /\ __/ __ __ /\ __/ \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\ \ \ _/ \ \ _/\ \ _\ \ \ \ _/ \ _\ \ _\ \ ____/ \ _\ \/_/ \/_/ \/___/ \/_/ v1.1.0-git`

```
_____ :: Method : GET :: URL : http://academy.htb:PORT/ :: Wordlist : FUZZ: /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt :: Header : Host: FUZZ :: Follow redirects : false :: Calibration : false :: Timeout : 10 :: Threads : 40 :: Matcher : Response status: 200,204,301,302,307,401,403 _____ mail2 [Status: 200, Size: 900, Words: 423, Lines: 56] dns2 [Status: 200, Size: 900, Words: 423, Lines: 56] ns3 [Status: 200, Size: 900, Words: 423, Lines: 56] dns1 [Status: 200, Size: 900, Words: 423, Lines: 56] lists [Status: 200, Size: 900, Words: 423, Lines: 56] webmail [Status: 200, Size: 900, Words: 423, Lines: 56] static [Status: 200, Size: 900, Words: 423, Lines: 56] web [Status: 200, Size: 900, Words: 423, Lines: 56] www1 [Status: 200, Size: 900, Words: 423, Lines: 56] <...SNIP...>
```

We see that all words in the wordlist are returning `200 OK`! This is expected, as we are simply changing the header while visiting `http://academy.htb:PORT/`. So, we know that we will always get `200 OK`. However, if the VHost does exist and we send a correct one in the header, we should get a different response size, as in that case, we would be getting the page from that VHosts, which is likely to show a different page.

Filtering Results

So far, we have not been using any filtering to our `ffuf`, and the results are automatically filtered by default by their HTTP code, which filters out code `404 NOT FOUND`, and keeps the rest. However, as we saw in our previous run of `ffuf`, we can get many responses with code `200`. So, in this case, we will have to filter the results based on another factor, which we will learn in this section.

Filtering

`Ffuf` provides the option to match or filter out a specific HTTP code, response size, or amount of words. We can see that with `ffuf -h`:

mayala@htb[/htb] `$ ffuf -h ...SNIP... MATCHER OPTIONS: -mc Match HTTP status codes, or "all" for everything. (default: 200,204,301,302,307,401,403) -ml Match amount of lines in response -mr Match regexp -ms Match HTTP response size -mw Match amount of words in response FILTER OPTIONS: -fc Filter HTTP status codes from response. Comma separated list of codes and ranges -fl Filter by amount of lines in response. Comma separated list of line counts and ranges -fr Filter regexp -fs Filter HTTP response size. Comma separated list of sizes and ranges -`

```
fw Filter by amount of words in response. Comma separated list of word counts and ranges <...SNIP...>
```

In this case, we cannot use matching, as we don't know what the response size from other VHosts would be. We know the response size of the incorrect results, which, as seen from the test above, is `900`, and we can filter it out with `-fs 900`. Now, let's repeat the same previous command, add the above flag, and see what we get:

```
mayala@htb[/htb] $ ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://academy.htb:PORT/ -H 'Host: FUZZ.academy.htb' -fs 900 /'___\ /'___\ /'___\ /\ \__/ /\ \__/ __ __ /\ \__/ \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\ \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/ \ \_\ \ \_\ \ \____/ \ \_\ \/_/ \/_/ \/___/ \/_/ v1.1.0-git
_____ :: Method : GET :: URL : http://academy.htb:PORT/ :: Wordlist : FUZZ: /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt :: Header : Host: FUZZ.academy.htb :: Follow redirects : false :: Calibration : false :: Timeout : 10 :: Threads : 40 :: Matcher : Response status: 200,204,301,302,307,401,403 :: Filter : Response size: 900
_____ <...SNIP...> admin [Status: 200, Size: 0, Words: 1, Lines: 1] :: Progress: [4997/4997] :: Job [1/1] :: 1249 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

We can verify that by visiting the page, and seeing if we can connect to it:

Note 1: Don't forget to add "admin.academy.htb" to "/etc/hosts".

Note 2: If your exercise has been restarted, ensure you still have the correct port when visiting the website.

We see that we can access the page, but we get an empty page, unlike what we got with `academy.htb`, therefore confirming this is indeed a different VHost. We can even visit `https://admin.academy.htb:PORT/blog/index.php`, and we will see that we would get a `404 PAGE NOT FOUND`, confirming that we are now indeed on a different VHost.

Try running a recursive scan on `admin.academy.htb`, and see what pages you can identify.