**Attacking Web Applications with Ffuf**

**Introduction**

In this module the learner will learn tools and methods to utilize for directory and parameter fuzzing/brute-forcing. In this module, the main focus is built on the [ffuf](https://github.com/ffuf/ffuf) tool for web fuzzing, as it is one of the most common and reliable tools available for web fuzzing.

The learner will go on to understand following topics:

* Fuzzing for directories
* Fuzzing for files and extensions
* Identifying hidden vhosts
* Fuzzing for PHP parameters
* Fuzzing for parameter values

***Activities***

***Web Fuzzing***

Fuzzing refers to a testing technique that sends various types of user input to a certain interface to study how it would react.

**Wordlists**

This refers to a wordlist containing commonly used words for web directories and pages, very similar to a ***Password Dictionary Attack***.

Some of the most commonly used wordlists can be found under the GitHub [SecLists](https://github.com/danielmiessler/SecLists) repository, which categorizes wordlists under various types of fuzzing, even including commonly used passwords, which we'll later utilize for Password Brute Forcing.

The learner can find the entire SecLists repo available under /opt/useful/SecLists. The specific wordlist to be utilized for pages and directory fuzzing is another commonly used wordlist called directory-list-2.3, and it is available in various forms and sizes. The learner can find the one we will be using under:

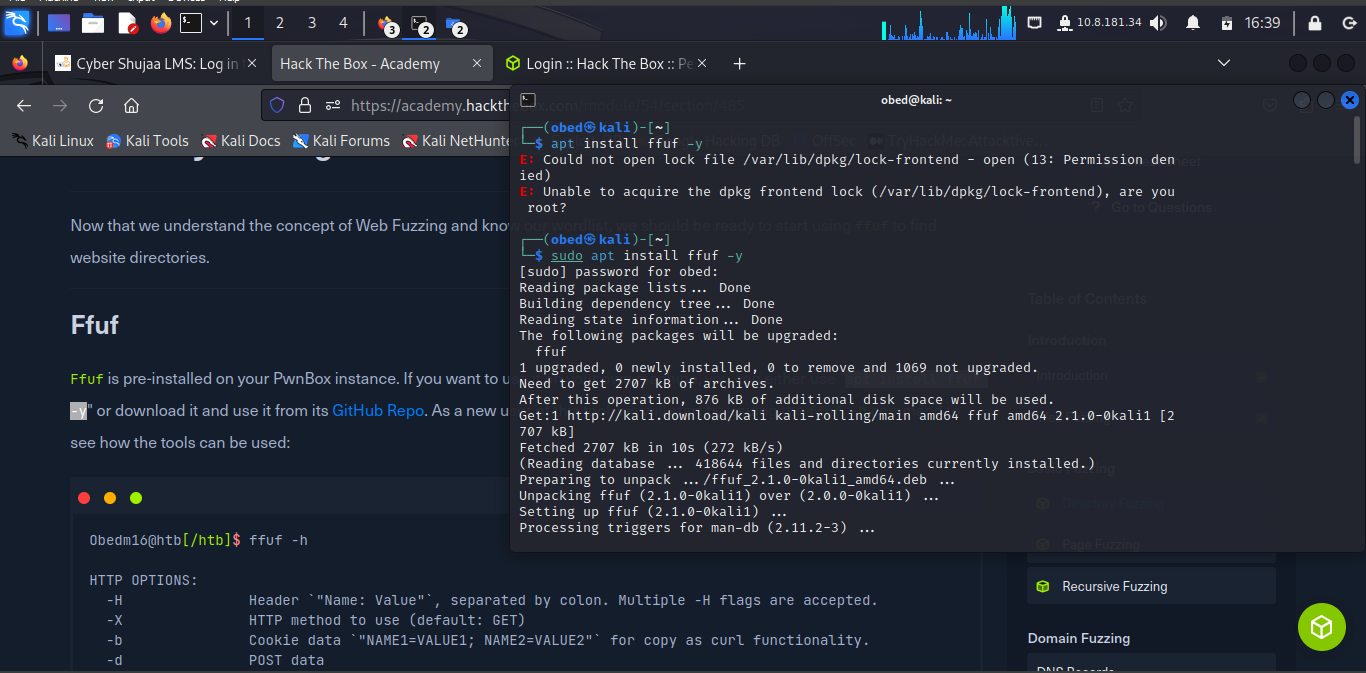
**locate directory-list-2.3-small.txt**

**Directory Fuzzing**

The learner having known a location of the wordlist to be used and the concept of Web fuzzing, they are ready to exercise this activity to find website directories.

**Ffuf**

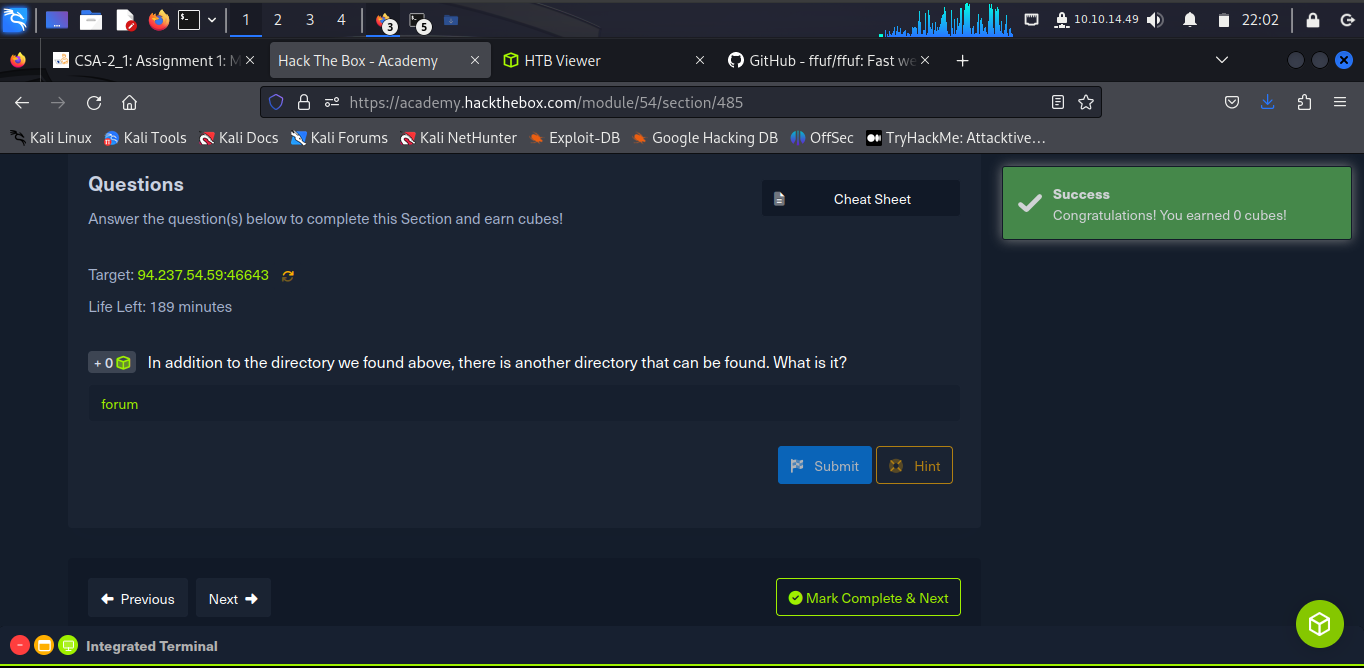
***Ffuf*** is pre-installed on the PwnBox instance. If the learner want to use it on their own machine, they can either use "**sudo apt install ffuf -y**" or download it and use it from its [GitHub Repo](https://github.com/ffuf/ffuf.git). As a new user of this tool, we will start by issuing the **ffuf -h** command to see how the tools can be used:



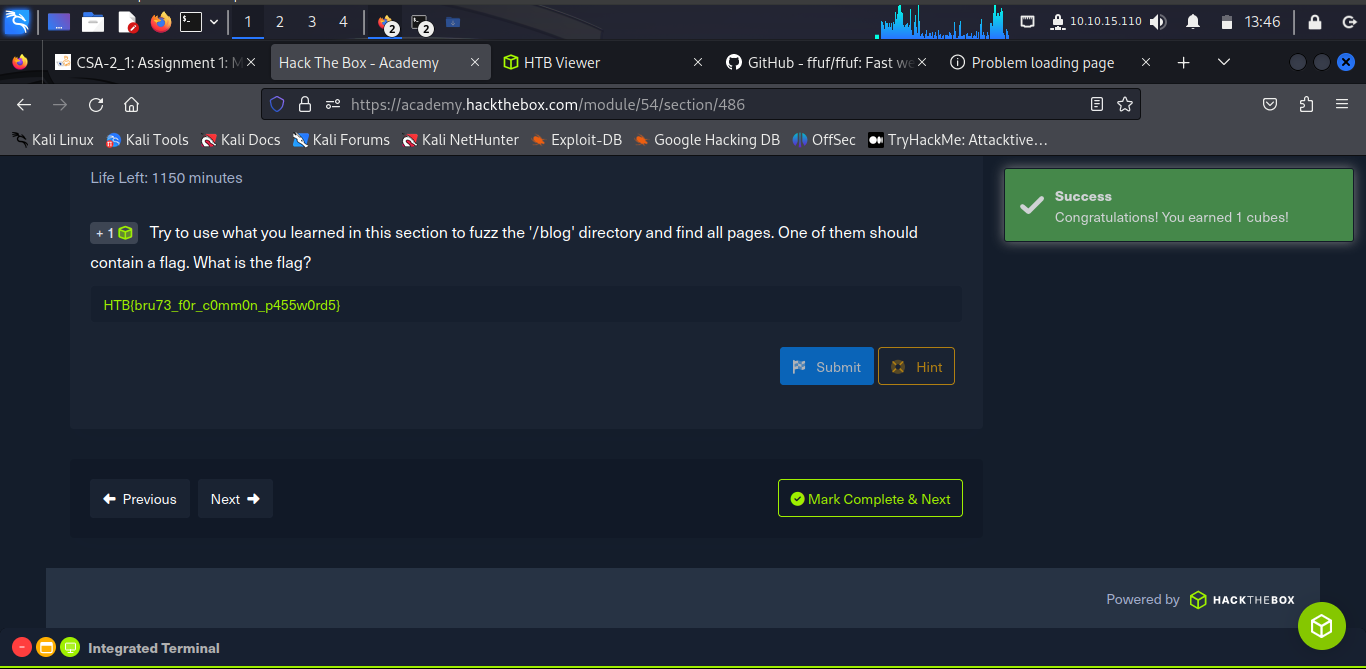
**Directory Fuzzing**

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.

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:



The learner run **94.237.54.59:46643/blog/home.php**

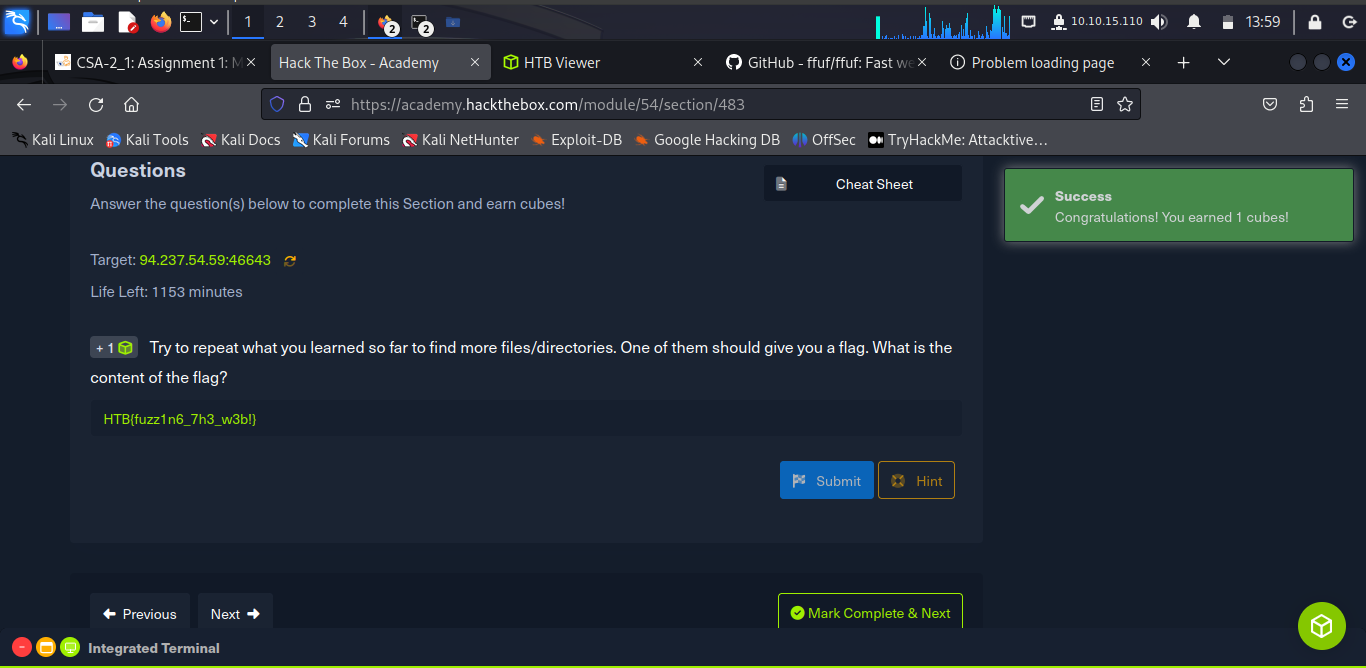


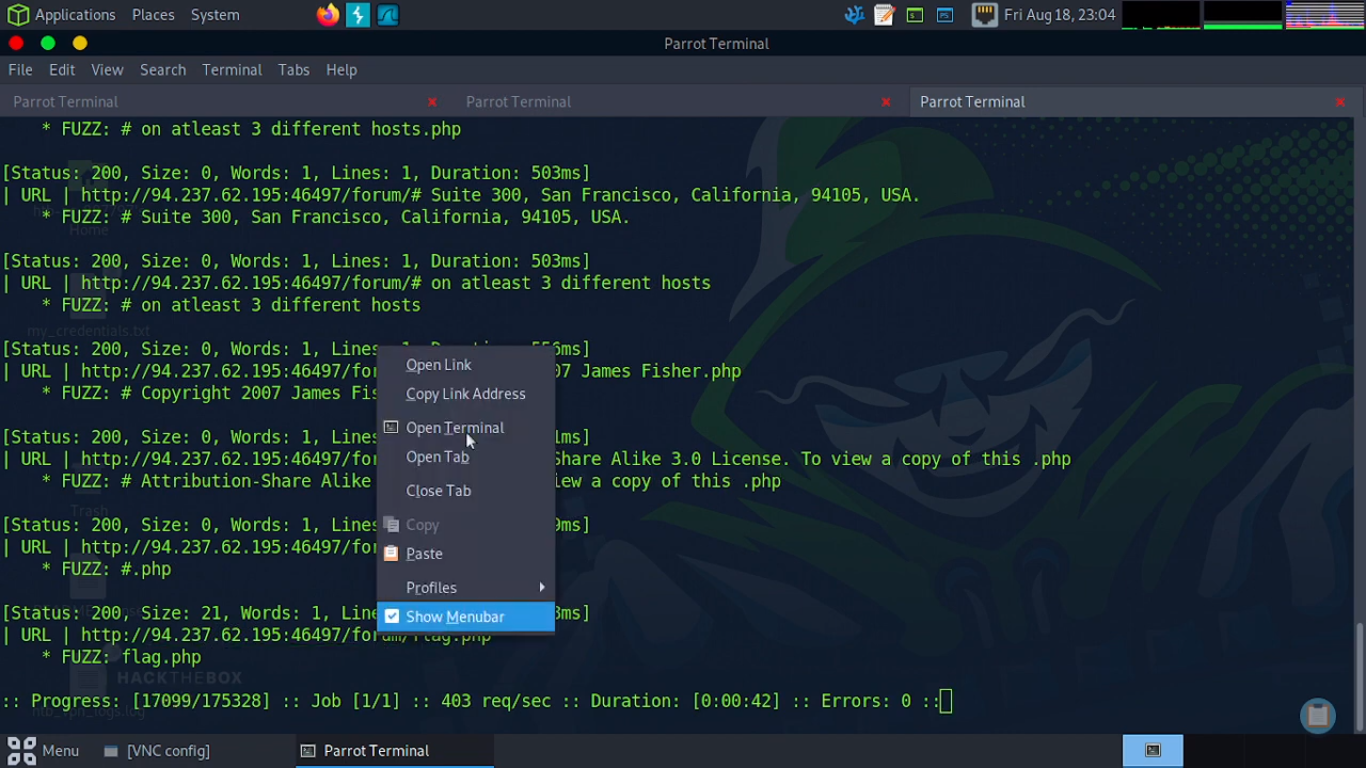


**Recursive Fuzzing**

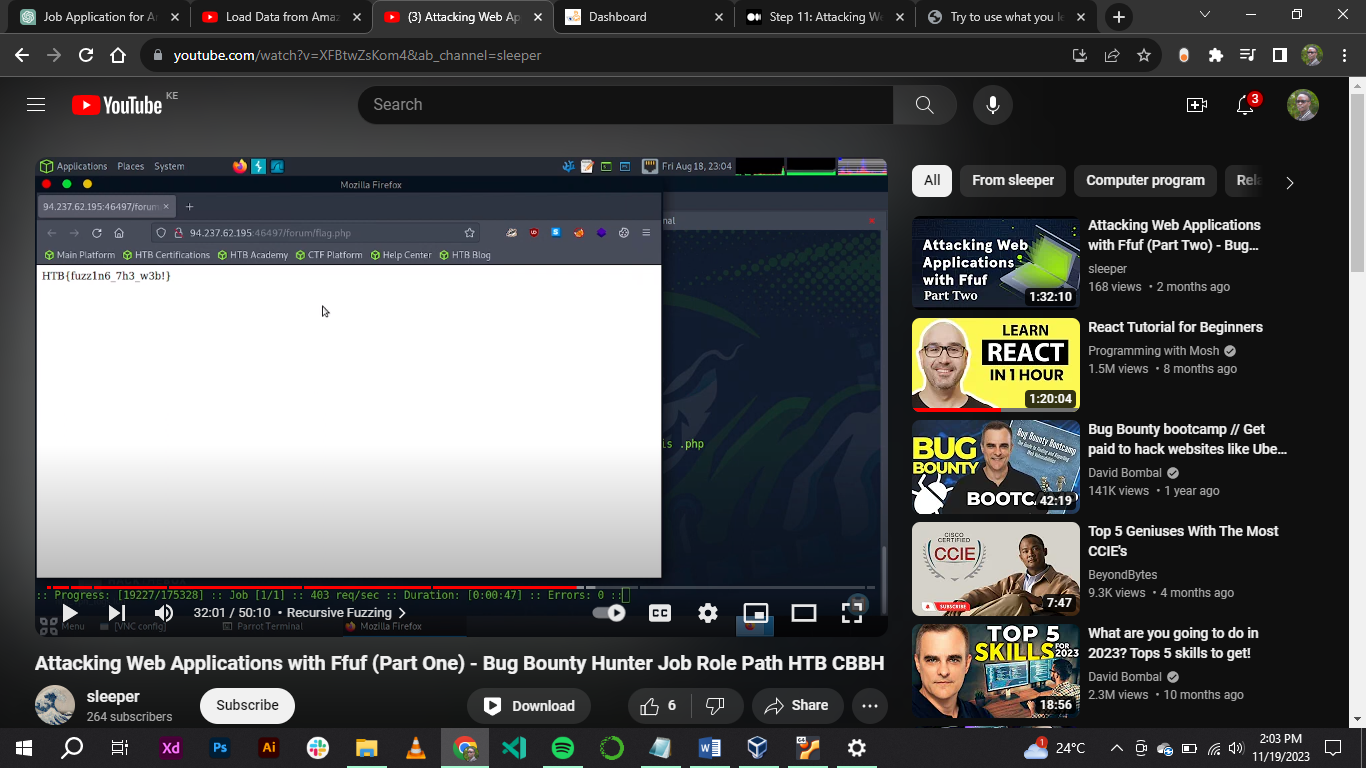
For dozens of directories, each with their own subdirectories and files are automated through **recursive fuzzing**.

It automatically starts another scan under any newly identified directories that may have on their pages until it has fuzzed the main website and all of its subdirectories.





From the above, the learner run **http:94.237.62.195:46497/forum/flag.php**



**DNS Records**

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.

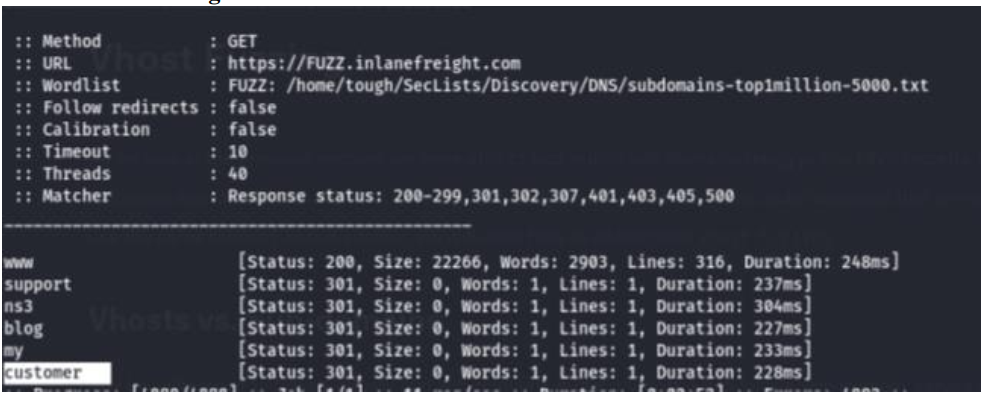
**Sub-domain Fuzzing**

The learner will learn how to use **ffuf** to identify sub-domains e.g., **\*.website.com** for any website.

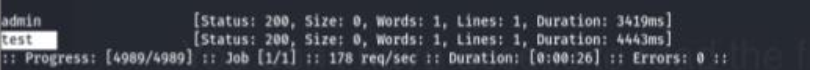
***Sub-domains***

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

Run the **ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://FUZZ.** **inlanefreight.com**

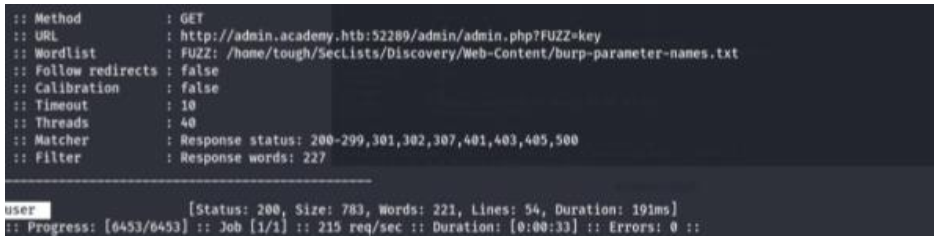


The full subdomain is **customer.inlanefreight.com**



When the learner run a VHost fuzzing scan on 'academy.htb', the other VHosts seen is:

**test.academy.htb**

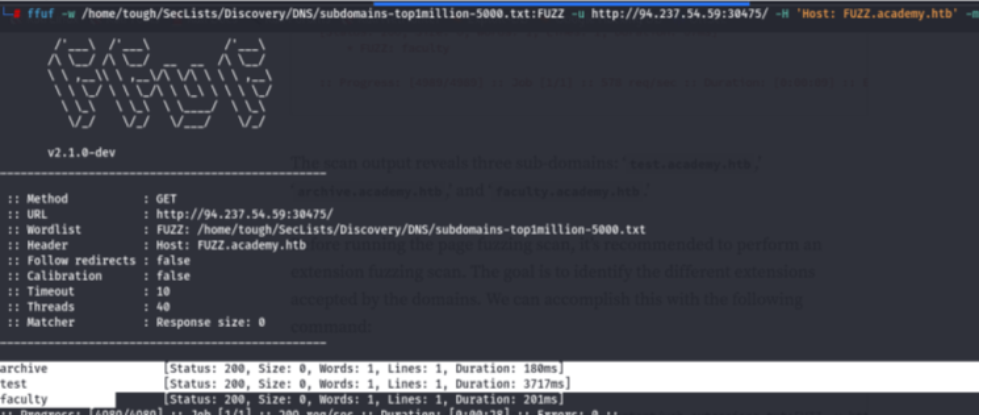


Upon running a parameter fuzzing scan on this page, the parameter accepted by this webpage is **User.**

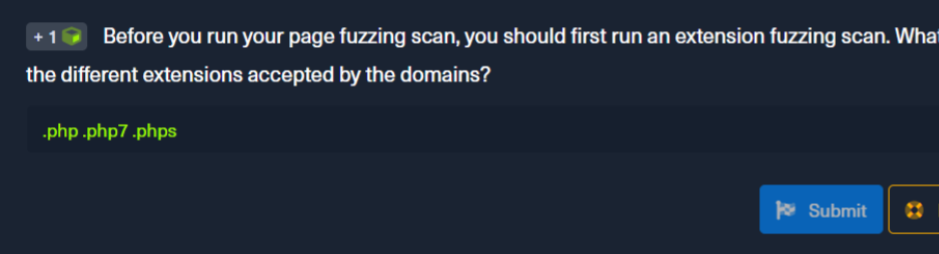


On creating the 'ids.txt' wordlist, identified and the accepted value with a fuzzing scan; and used in a 'POST' request with 'curl' in collecting the flag is:

HTB{p4r4m373r\_fuzz1n6\_15\_k3y!}



Run a sub-domain/vhost fuzzing scan on '\*.academy.htb' for the IP shown above. What are all the sub-domains you can identify? (Only write the sub-domain name) archive faculty test



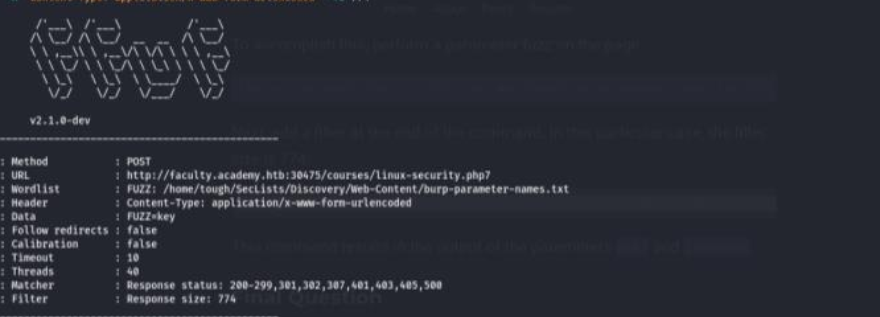
Before you run your page fuzzing scan, you should first run an extension fuzzing scan. What are the different extensions accepted by the domains? .php .php7 .phps



One of the pages you will identify should say 'You don't have access!'. What is the full

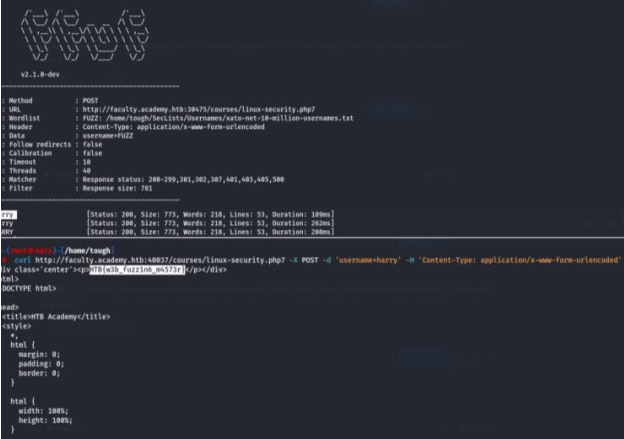
page URL? http://faculty.academy.htb:PORT/courses/linux-security.php7

*customer.inlanefreight.com*



In the page from the previous question, you should be able to find multiple parameters

that are accepted by the page. They are: user username



Try fuzzing the parameters you identified for working values. One of them should return

a flag. The content of the flag is: HTB{w3b\_fuzz1n6\_m4573r}

**Conclusion**

The learning concepts of domain fuzzing and sub domain is a bit challenging, but with time the learner was able penetrate sites and achieved to Attack Web Applications with **ffuf command** and other complex tools (FUZZ)



Completion: <https://academy.hackthebox.com/achievement/978332/54>