# WFuzz

## Overview:

* Web application vulnerability scanner and fuzzer
* Allows arbitrary input to be injected through HTTP requests
  + Allows for brute force
  + Allows attacker to find files
* Written in python

Example:

* Looking for files of interest

./wfuzz-cli.py -c -w <directory for word list> --hc 404 <url/FUZZ>

* -c
  + Outputs with colour
* -w
  + allows for a word list
* --hc
  + Hides 404 response

So what this will do is look for directories on the webserver connected to this URL. What you want to look out for are

* Api
* Index
* Login
* Passwords
* Secrets
* Php files
* Html files
* Txt files

In order to search for words in your word list and see if there are any files with php txt or html extensions you can use “go buster”

gobuster -u <URL >-w <Word list>-t 50 -x php,txt,html,html

* -U specifies you are giving it a URL
* -w is wordlist
* -t is the threads
* -x allows you to specify extensions to search for

If you find an api directory you can search through that by adding on /api to your URL and you may find a php file.

If we visit the url /api/test.php we may bet a parameter not set error

We can use wfuzz to find the GET parameter

wfuzz --hh=24 -c -w <Word list> <URL/api/test.php?FUZZ=anything>

--hh is to set the length of characters in the source code. This is a way of filtering results, like filtering out 404 errors. Otherwise we are flooded with random results. It will tell you the characters if search without filtering initially and then you can filter them

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generatedHere I did a fuzz without the –hh flag and it is just showing me all the words it is searching for as it goes through my word list. It says in the chars tab 24 so if I filter this it will only show me actual results and not every attempt made

Now when we are fuzzing we change the command from

wfuzz --hh=24 -c -w /root/Documents/wordlist.txt <http://docker.hackthebox.eu:30345/api/action.php?FUZZ=test>

to

A screenshot of a cell phone

Description automatically generatedwfuzz --hh=24 -c -w /root/Documents/wordlist.txt <http://docker.hackthebox.eu:30345/api/action.php?reset=FUZZ>

When accessing the page with reset= as the parameter we actually need to find the value for this which is an Account ID. By fuzzing this page we could find some results

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedNow we need to filter –hh=27

We have found a successful account ID to send the get request and capture the flag