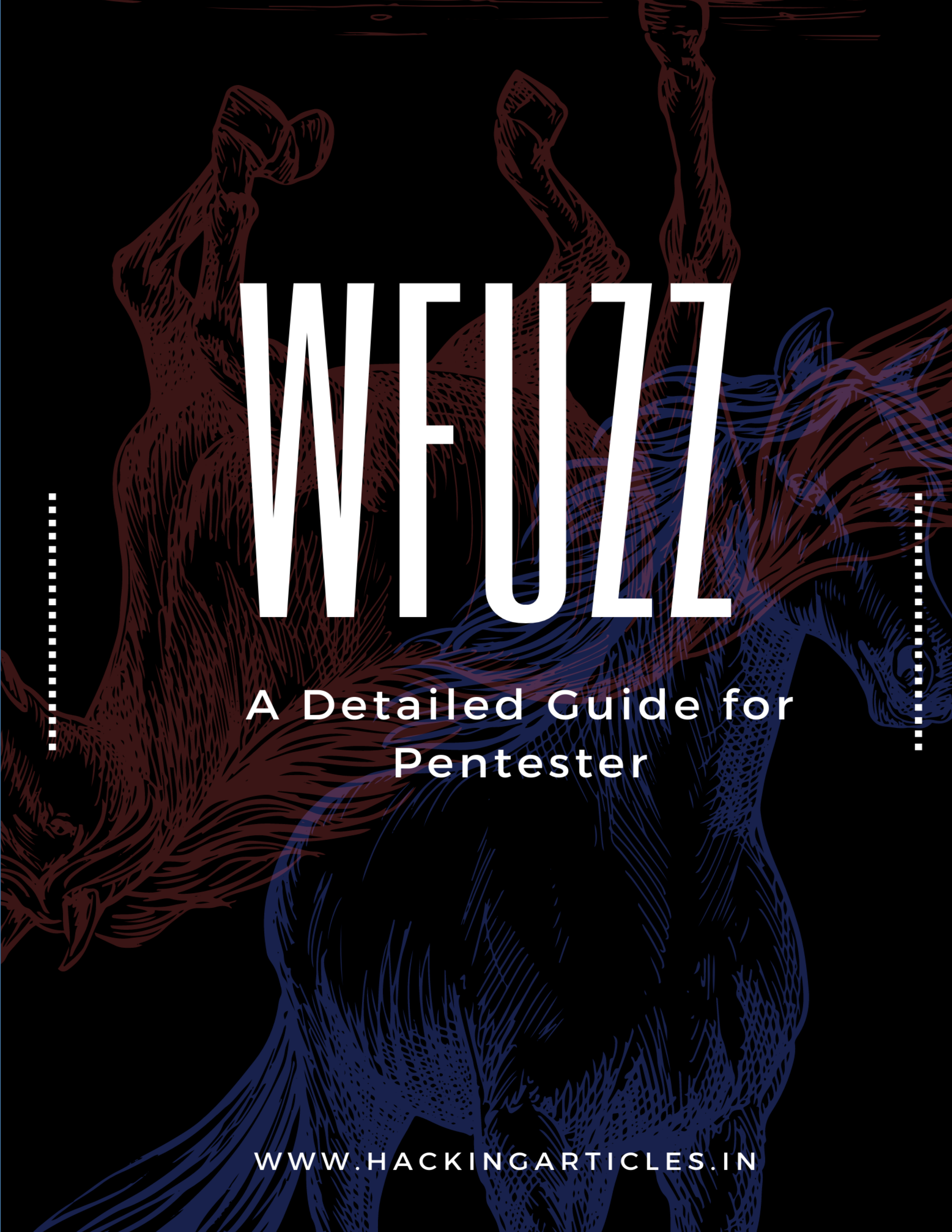


WFUZZ

A Detailed Guide for Pentester

WWW.HACKINGARTICLES.IN

Contents

Introduction

Many tools have been developed that create an HTTP request and allow a user to modify their contents. Fuzzing works the same way. A user can send a similar request multiple times to the server with a certain section of the request changed. When that certain section is replaced by a variable from a list or directory, it is called fuzzing.

In this article, we will learn how we can use wfuzz, which states for "Web Application Fuzzer", which is an interesting open-source web fuzzing tool. Since its release, many people have gravitated towards wfuzz, particularly in the bug bounty scenario. So, let's dive into this learning process.

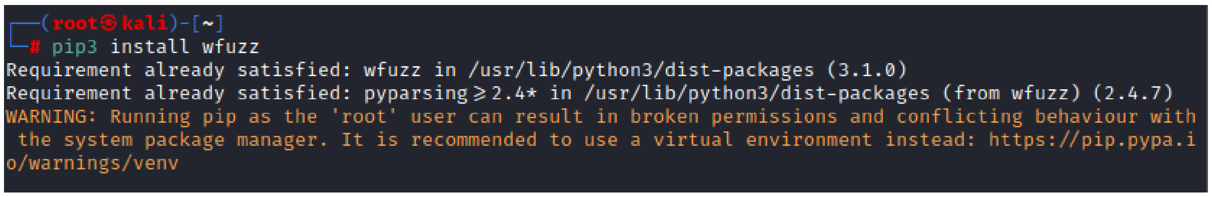
Introduction to Wfuzz

Wfuzz is a python coded application to fuzz web applications with a plethora of options. It offers various filters that allow one to replace a simple web request with a required word by replacing it with the variable "FUZZ."

```
pip3 install wfuzz
```

Setup

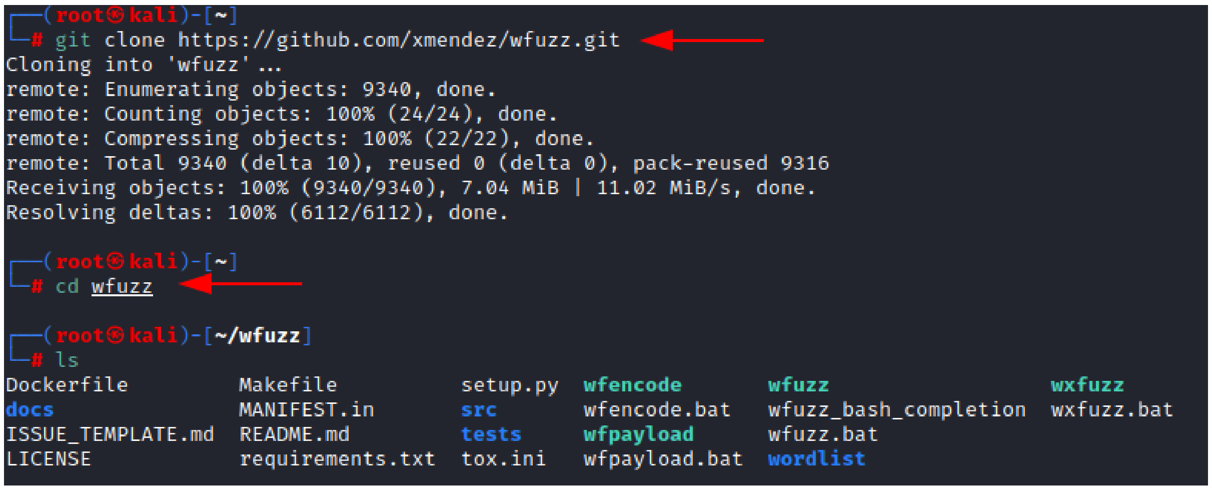
To install wfuzz using pip, we can:



```
┌──(root㉿kali)-[~]
└─# pip3 install wfuzz
Requirement already satisfied: wfuzz in /usr/lib/python3/dist-packages (3.1.0)
Requirement already satisfied: pyparsing≥2.4* in /usr/lib/python3/dist-packages (from wfuzz) (2.4.7)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
 the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.i
o/warnings/venv
```

```
git clone https://github.com/xmendez/wfuzz.git
```

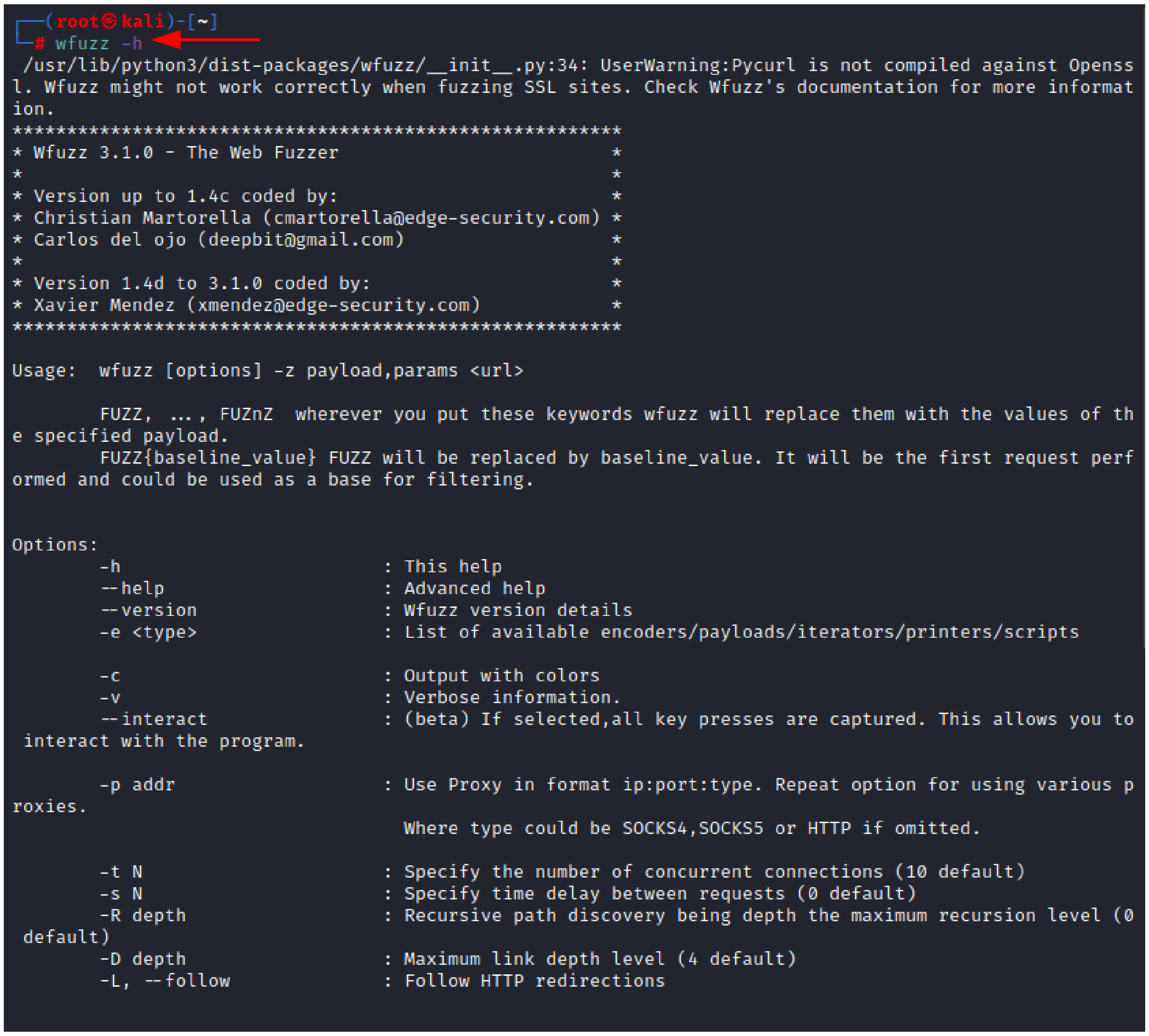
The same could be achieved by installing from the source using git.



```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/xmendez/wfuzz.git
Cloning into 'wfuzz'...
remote: Enumerating objects: 9340, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 9340 (delta 10), reused 0 (delta 0), pack-reused 9316
Receiving objects: 100% (9340/9340), 7.04 MiB | 11.02 MiB/s, done.
Resolving deltas: 100% (6112/6112), done.

┌──(root㉿kali)-[~]
└─# cd wfuzz

┌──(root㉿kali)-[~/wfuzz]
└─# ls
Dockerfile         Makefile          setup.py  wfencode      wfuzz                  wxfuzz
docs               MANIFEST.in       src       wfencode.bat  wfuzz_bash_completion  wxfuzz.bat
ISSUE_TEMPLATE.md  README.md         tests     wfpayload     wfuzz.bat
LICENSE            requirements.txt  tox.ini   wfpayload.bat wordlist
```

The help menu to see all the working options is as follows:

```
wfuzz -h
wfuzz --help
```



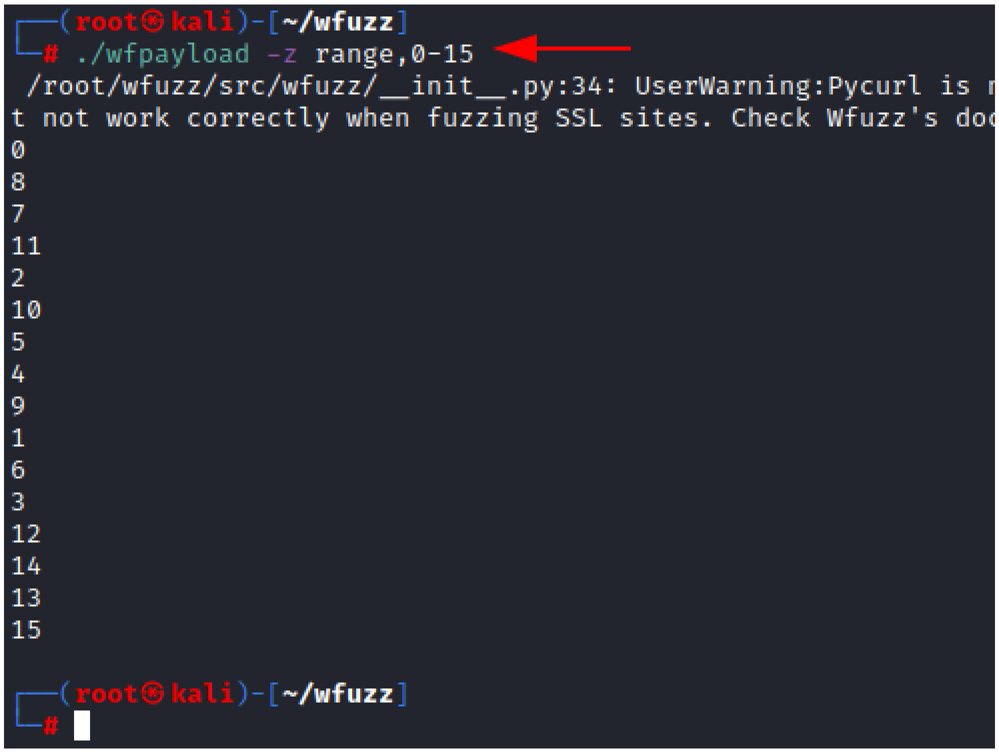
```
┌──(root㉿kali)-[~]
└─# wfuzz -h
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openss
l. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more informat
ion.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
*                                                      *
* Version up to 1.4c coded by:                         *
* Christian Martorella (cmartorella@edge-security.com) *
* Carlos del ojo (deepbit@gmail.com)                   *
*                                                      *
* Version 1.4d to 3.1.0 coded by:                      *
* Xavier Mendez (xmendez@edge-security.com)            *
********************************************************

Usage:  wfuzz [options] -z payload,params <url>

        FUZZ, ..., FUZnZ  wherever you put these keywords wfuzz will replace them with the values of th
e specified payload.
        FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the first request perf
ormed and could be used as a base for filtering.


Options:
        -h                        : This help
        --help                    : Advanced help
        --version                 : Wfuzz version details
        -e <type>                 : List of available encoders/payloads/iterators/printers/scripts

        -c                        : Output with colors
        -v                        : Verbose information.
        --interact                : (beta) If selected,all key presses are captured. This allows you to
 interact with the program.

        -p addr                   : Use Proxy in format ip:port:type. Repeat option for using various p
roxies.
                                    Where type could be SOCKS4,SOCKS5 or HTTP if omitted.

        -t N                      : Specify the number of concurrent connections (10 default)
        -s N                      : Specify time delay between requests (0 default)
        -R depth                  : Recursive path discovery being depth the maximum recursion level (0
 default)
        -D depth                  : Maximum link depth level (4 default)
        -L, --follow              : Follow HTTP redirections
```

You can use a module by using "-z"

Wfpayload and Wfencode

When you install the tool from source, compiled executables called wfpayload and wfencode are available. These are responsible for payload generation and encoding. They can be individually used. For example, command to generate digits from 0 to 15 is as follows:

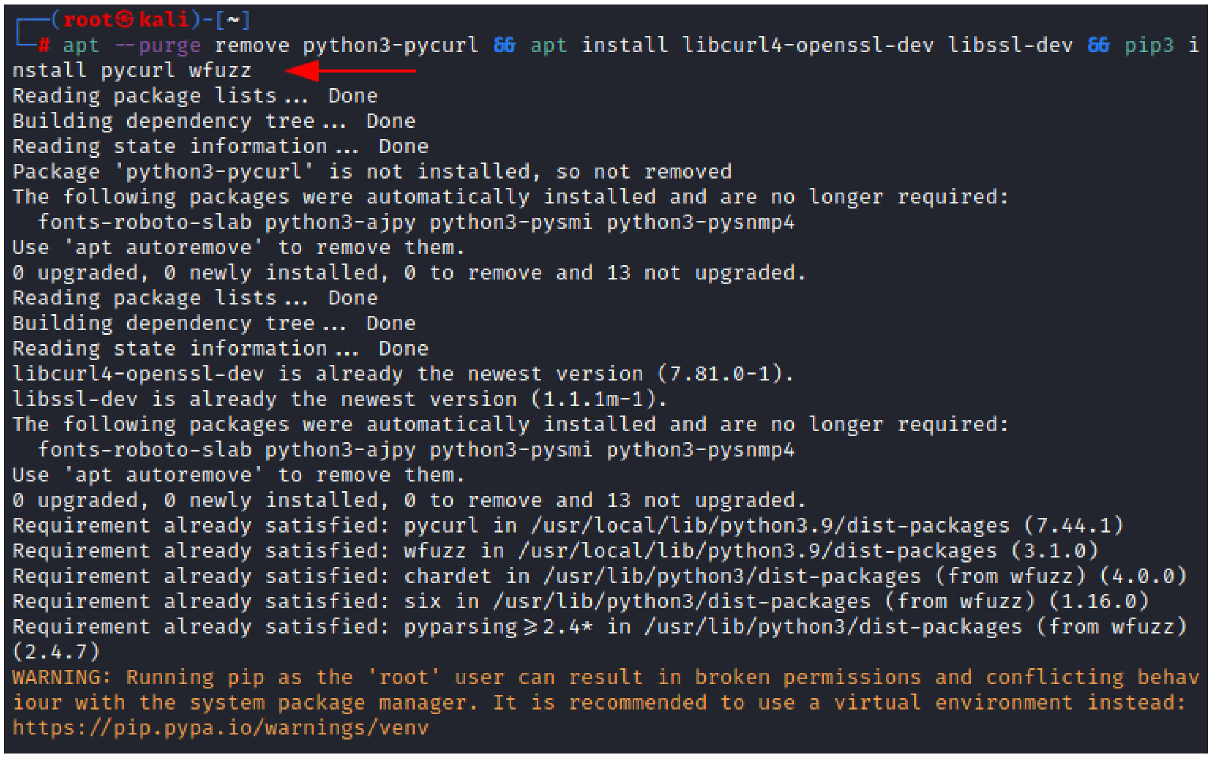
```
./wfpayload -z range,0-15
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# ./wfpayload -z range,0-15
 /root/wfuzz/src/wfuzz/__init__.py:34: UserWarning:Pycurl is n
t not work correctly when fuzzing SSL sites. Check Wfuzz's doc
0
8
7
11
2
10
5
4
9
1
6
3
12
14
13
15

┌──(root㉿kali)-[~/wfuzz]
└─#
```

As you can see, there is a pycurl error. It can go away like so:

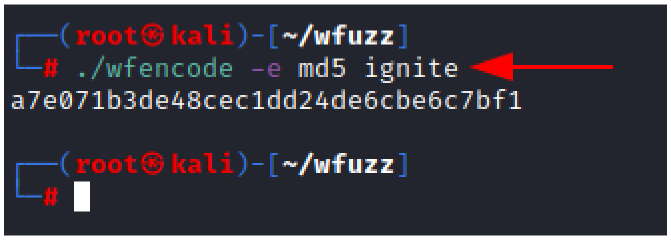
apt --purge remove python3-pycurl && apt install libcurl4-openssl-dev libssl-dev && pip3 i



```
┌──(root㉿kali)-[~]
└─# apt --purge remove python3-pycurl && apt install libcurl4-openssl-dev libssl-dev && pip3 i
nstall pycurl wfuzz
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'python3-pycurl' is not installed, so not removed
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab python3-ajpy python3-pysmi python3-pysnmp4
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libcurl4-openssl-dev is already the newest version (7.81.0-1).
libssl-dev is already the newest version (1.1.1m-1).
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab python3-ajpy python3-pysmi python3-pysnmp4
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
Requirement already satisfied: pycurl in /usr/local/lib/python3.9/dist-packages (7.44.1)
Requirement already satisfied: wfuzz in /usr/local/lib/python3.9/dist-packages (3.1.0)
Requirement already satisfied: chardet in /usr/lib/python3/dist-packages (from wfuzz) (4.0.0)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from wfuzz) (1.16.0)
Requirement already satisfied: pyparsing>=2.4* in /usr/lib/python3/dist-packages (from wfuzz)
(2.4.7)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behav
iour with the system package manager. It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
```

Now, when you run wfencode, which is a module to encode a supplied input using a hash algorithm, there is no pycurl error now.

```
./wfencode -e md5 ignite
```

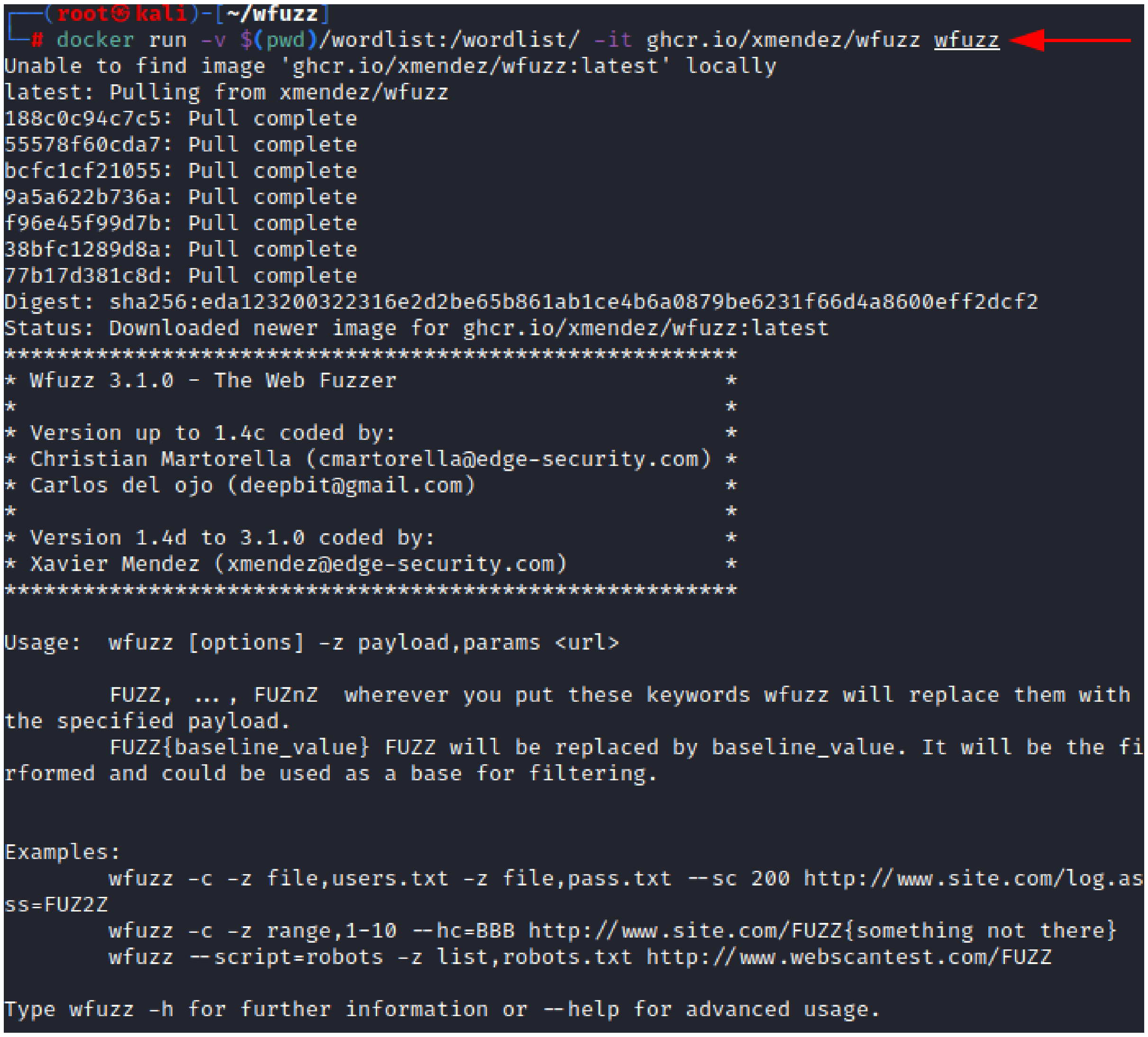


```
┌──(root㉿kali)-[~/wfuzz]
└─# ./wfencode -e md5 ignite
a7e071b3de48cec1dd24de6cbe6c7bf1

┌──(root㉿kali)-[~/wfuzz]
└─#
```

Docker run wfuzz

Wfuzz can also be launched using docker in the following way using the repo ghcr.io. The respective command can be run by replacing the last variable wfuzz.

```
docker run -v $(pwd)/wordlist:/wordlist/ -it ghcr.io/xmendez/wfuzz wfuzz
```

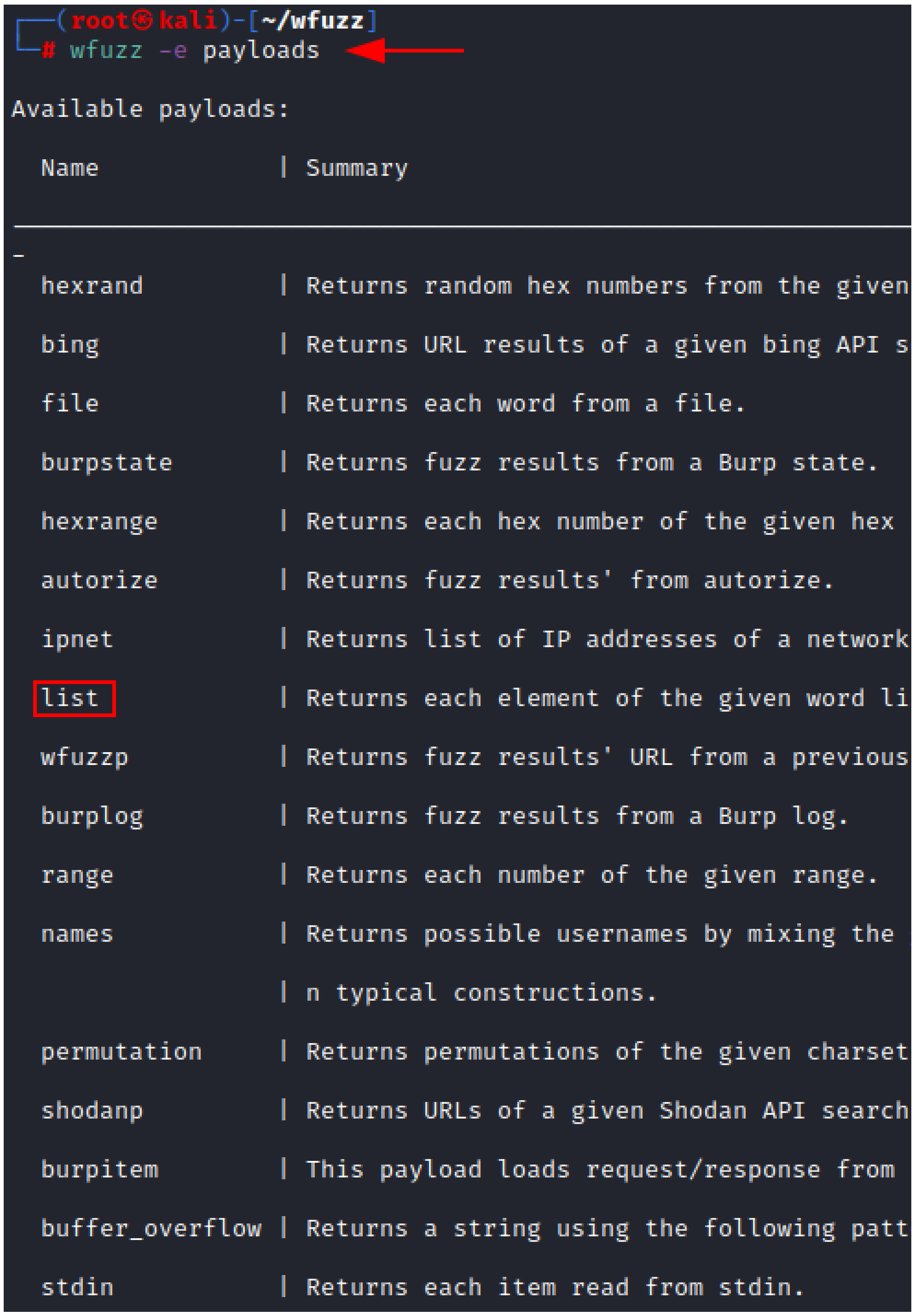


```
┌──(root㉿kali)-[~/wfuzz]
└─# docker run -v $(pwd)/wordlist:/wordlist/ -it ghcr.io/xmendez/wfuzz wfuzz
Unable to find image 'ghcr.io/xmendez/wfuzz:latest' locally
latest: Pulling from xmendez/wfuzz
188c0c94c7c5: Pull complete
55578f60cda7: Pull complete
bcfc1cf21055: Pull complete
9a5a622b736a: Pull complete
f96e45f99d7b: Pull complete
38bfc1289d8a: Pull complete
77b17d381c8d: Pull complete
Digest: sha256:eda123200322316e2d2be65b861ab1ce4b6a0879be6231f66d4a8600eff2dcf2
Status: Downloaded newer image for ghcr.io/xmendez/wfuzz:latest
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
*                                                      *
* Version up to 1.4c coded by:                         *
* Christian Martorella (cmartorella@edge-security.com) *
* Carlos del ojo (deepbit@gmail.com)                   *
*                                                      *
* Version 1.4d to 3.1.0 coded by:                      *
* Xavier Mendez (xmendez@edge-security.com)            *
********************************************************

Usage:  wfuzz [options] -z payload,params <url>

        FUZZ, ..., FUZnZ  wherever you put these keywords wfuzz will replace them with
the specified payload.
        FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the fi
rformed and could be used as a base for filtering.


Examples:
        wfuzz -c -z file,users.txt -z file,pass.txt --sc 200 http://www.site.com/log.as
ss=FUZ2Z
        wfuzz -c -z range,1-10 --hc=BBB http://www.site.com/FUZZ{something not there}
        wfuzz --script=robots -z list,robots.txt http://www.webscantest.com/FUZZ

Type wfuzz -h for further information or --help for advanced usage.
```

Payloads

A payload in Wfuzz is a source of input data. The available payloads can be listed by executing:

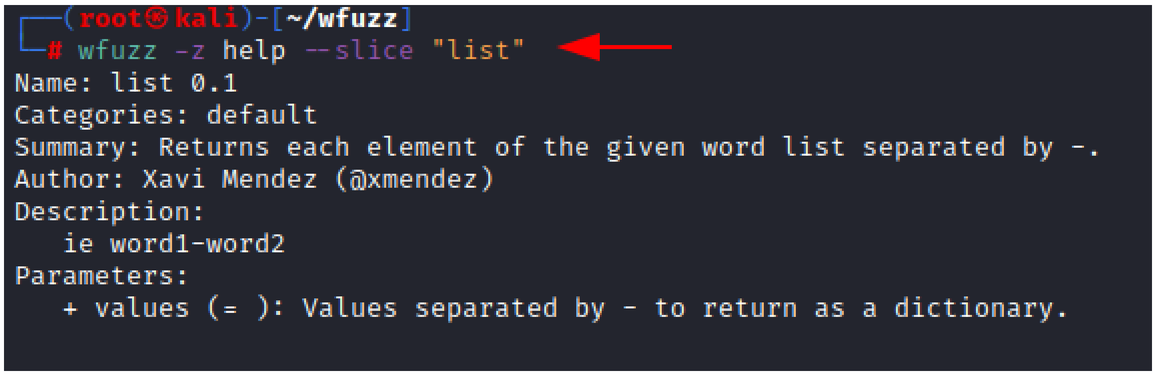
```
wfuzz -e payloads
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -e payloads

Available payloads:

  Name            | Summary

 ──────────────────────────────────────────────────────────────
-
  hexrand         | Returns random hex numbers from the given
  bing            | Returns URL results of a given bing API s
  file            | Returns each word from a file.
  burpstate       | Returns fuzz results from a Burp state.
  hexrange        | Returns each hex number of the given hex
  autorize        | Returns fuzz results' from autorize.
  ipnet           | Returns list of IP addresses of a network
  list            | Returns each element of the given word li
  wfuzzp          | Returns fuzz results' URL from a previous
  burplog         | Returns fuzz results from a Burp log.
  range           | Returns each number of the given range.
  names           | Returns possible usernames by mixing the
                  | n typical constructions.
  permutation     | Returns permutations of the given charset
  shodanp         | Returns URLs of a given Shodan API search
  burpitem        | This payload loads request/response from
  buffer_overflow | Returns a string using the following patt
  stdin           | Returns each item read from stdin.
```

The detailed view can also be looked using the slice filter:

```
wfuzz -z help --slice "list"
```



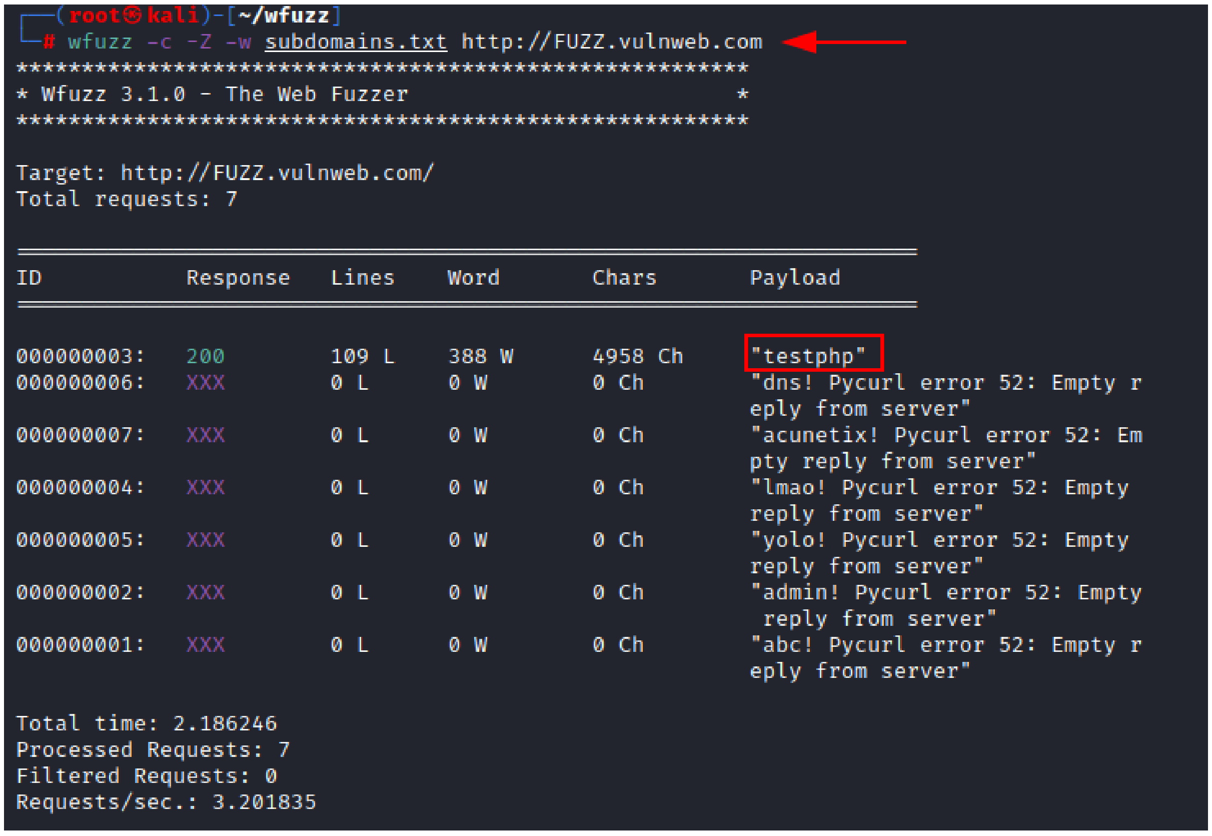
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z help --slice "list"
Name: list 0.1
Categories: default
Summary: Returns each element of the given word list separated by -.
Author: Xavi Mendez (@xmendez)
Description:
   ie word1-word2
Parameters:
   + values (= ): Values separated by - to return as a dictionary.
```

Subdomain Fuzzing

Subdomain discovery is extremely helpful in pentesting scenarios. Often, attackers launch attacks on subdomains rather than main domains and it can be fuzzed like so:

Here, -c color codes the output response codes
-Z specifies a URL to be input in scan mode and ignores any connection error
-w specifies the wordlist use while subdomain bruteforce.

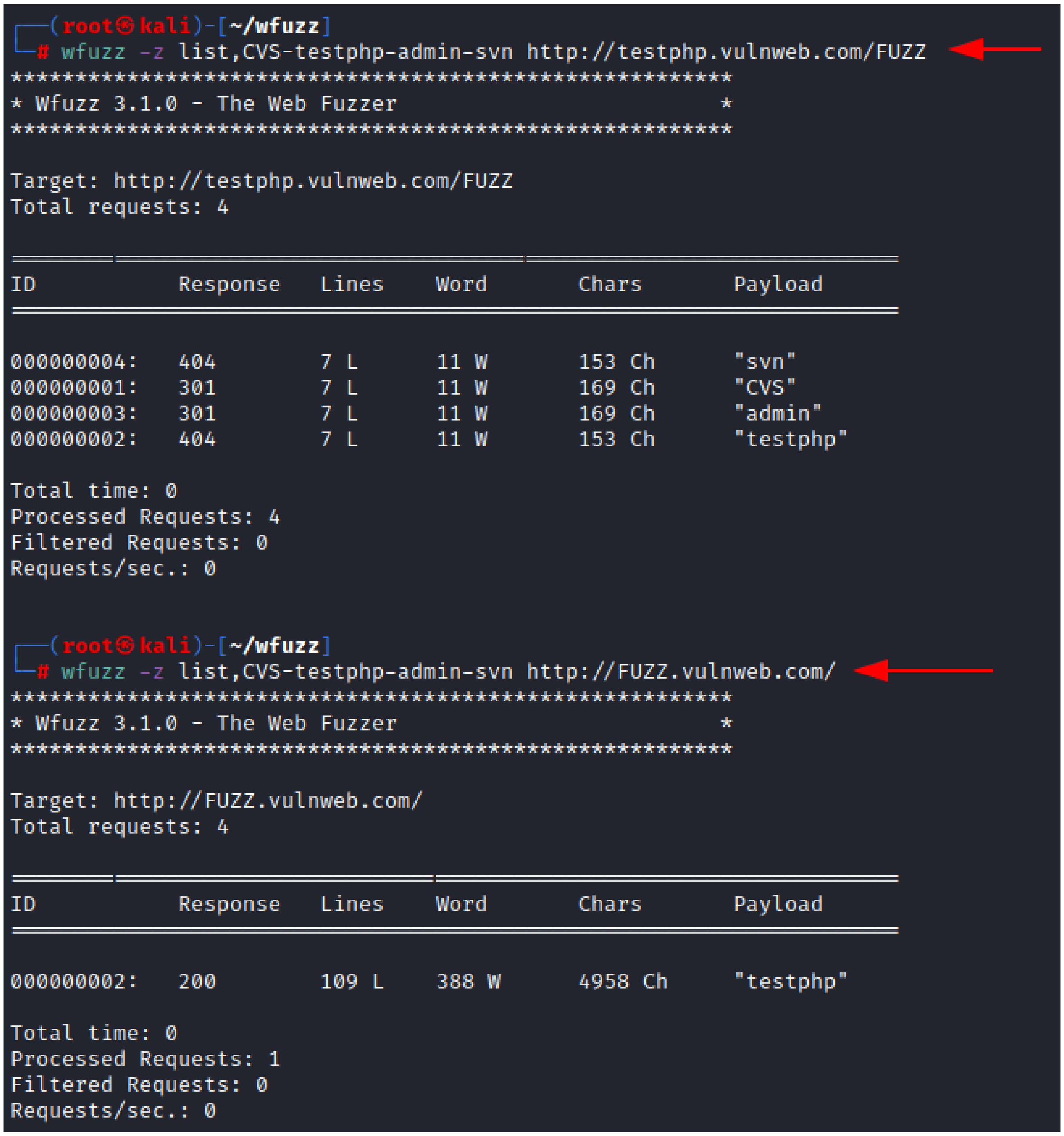
```
wfuzz -c -Z -w subdomains.txt http://FUZZ.vulnweb.com
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -c -Z -w subdomains.txt http://FUZZ.vulnweb.com
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://FUZZ.vulnweb.com/
Total requests: 7

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000003:   200        109 L    388 W      4958 Ch     "testphp"
000000006:   XXX        0 L      0 W        0 Ch        "dns! Pycurl error 52: Empty r
                                                        eply from server"
000000007:   XXX        0 L      0 W        0 Ch        "acunetix! Pycurl error 52: Em
                                                        pty reply from server"
000000004:   XXX        0 L      0 W        0 Ch        "lmao! Pycurl error 52: Empty
                                                        reply from server"
000000005:   XXX        0 L      0 W        0 Ch        "yolo! Pycurl error 52: Empty
                                                        reply from server"
000000002:   XXX        0 L      0 W        0 Ch        "admin! Pycurl error 52: Empty
                                                         reply from server"
000000001:   XXX        0 L      0 W        0 Ch        "abc! Pycurl error 52: Empty r
                                                        eply from server"

Total time: 2.186246
Processed Requests: 7
Filtered Requests: 0
Requests/sec.: 3.201835
```

The same can be achieved by providing the subdomain list inline too. Only, the payload (-z option) should be supplied in with “list” as an input. The list is supplied in the format ITEM1-ITEM2-ITEM3 like so:

```
wfuzz -z list,CVS-testphp-admin-svn http://testphp.vulnweb.com/FUZZ
wfuzz -z list,CVS-testphp-admin-svn http://FUZZ.vulnweb.com/
```



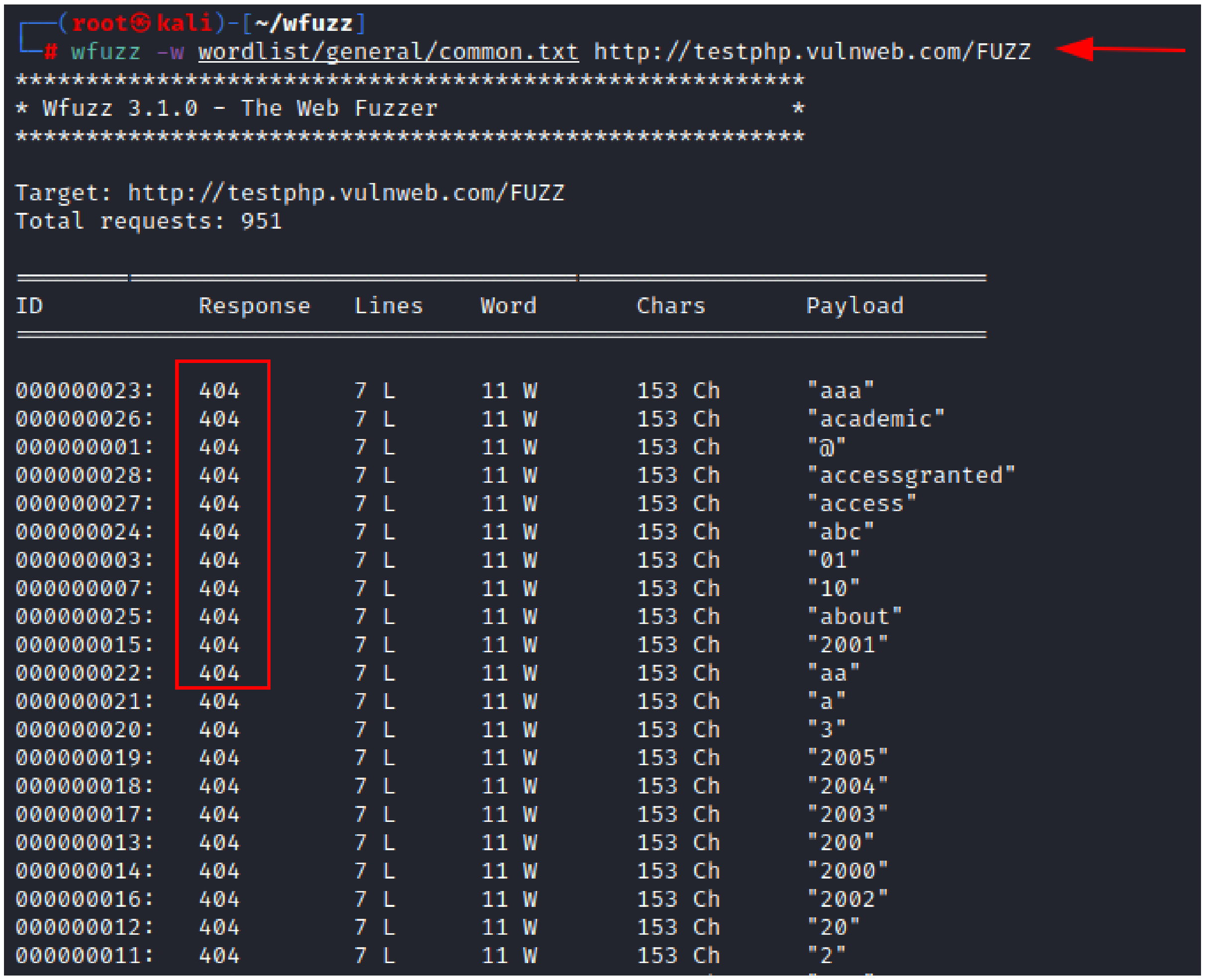
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,CVS-testphp-admin-svn http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 4

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000004:   404        7 L      11 W       153 Ch      "svn"
000000001:   301        7 L      11 W       169 Ch      "CVS"
000000003:   301        7 L      11 W       169 Ch      "admin"
000000002:   404        7 L      11 W       153 Ch      "testphp"

Total time: 0
Processed Requests: 4
Filtered Requests: 0
Requests/sec.: 0


┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,CVS-testphp-admin-svn http://FUZZ.vulnweb.com/
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://FUZZ.vulnweb.com/
Total requests: 4

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000002:   200        109 L    388 W      4958 Ch     "testphp"

Total time: 0
Processed Requests: 1
Filtered Requests: 0
Requests/sec.: 0
```

Directory Fuzzing

Directories can be enumerated using wfuzz just like with gobuster by using a supplied wordlist.

```
wfuzz -w wordlist/general/common.txt http://testphp.vulnweb.com/FUZZ
```

This can be done using a -w flag and input the path of the wordlist:

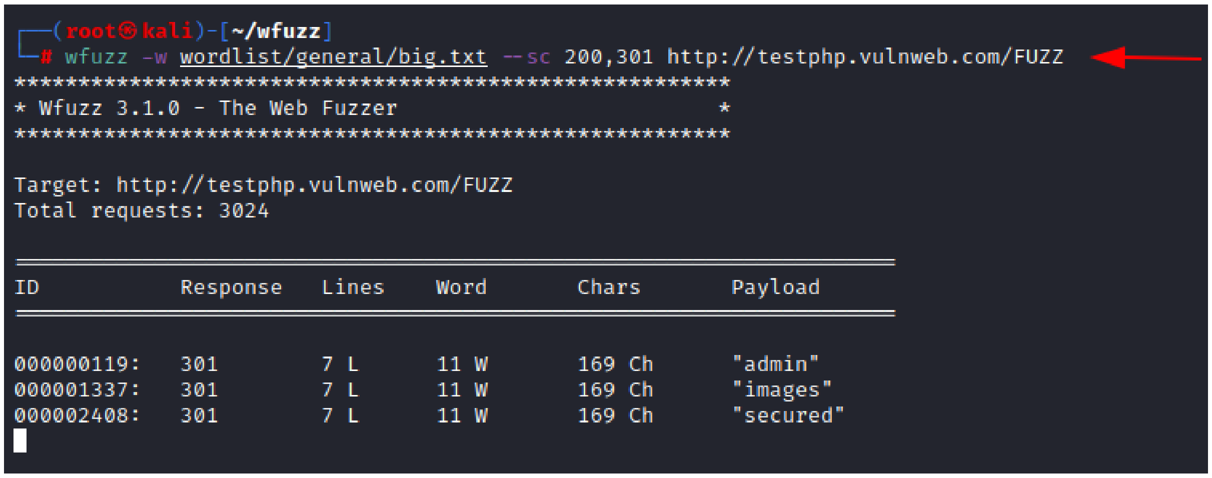


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/common.txt http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000023:   404        7 L      11 W       153 Ch      "aaa"
000000026:   404        7 L      11 W       153 Ch      "academic"
000000001:   404        7 L      11 W       153 Ch      "@"
000000028:   404        7 L      11 W       153 Ch      "accessgranted"
000000027:   404        7 L      11 W       153 Ch      "access"
000000024:   404        7 L      11 W       153 Ch      "abc"
000000003:   404        7 L      11 W       153 Ch      "01"
000000007:   404        7 L      11 W       153 Ch      "10"
000000025:   404        7 L      11 W       153 Ch      "about"
000000015:   404        7 L      11 W       153 Ch      "2001"
000000022:   404        7 L      11 W       153 Ch      "aa"
000000021:   404        7 L      11 W       153 Ch      "a"
000000020:   404        7 L      11 W       153 Ch      "3"
000000019:   404        7 L      11 W       153 Ch      "2005"
000000018:   404        7 L      11 W       153 Ch      "2004"
000000017:   404        7 L      11 W       153 Ch      "2003"
000000013:   404        7 L      11 W       153 Ch      "200"
000000014:   404        7 L      11 W       153 Ch      "2000"
000000016:   404        7 L      11 W       153 Ch      "2002"
000000012:   404        7 L      11 W       153 Ch      "20"
000000011:   404        7 L      11 W       153 Ch      "2"
```

As you can see in the above screenshot, all the results including page not found have been dumped which makes it tedious to go through the results and find pin in a haystack. Therefore, to sort the results out we can see the show code flag (--sc). Other such flags are:

- --hc/sc CODE #Hide/Show by code in response
- --hl/sl NUM #ide/Show by number of lines in response
- --hw/sw NUM #ide/Show by number of words in response
- --hc/sc NUM #ide/Show by number of chars in response

```
wfuzz -w wordlist/general/common.txt --sc 200,301 http://testphp.vulnweb.com/FUZZ
```



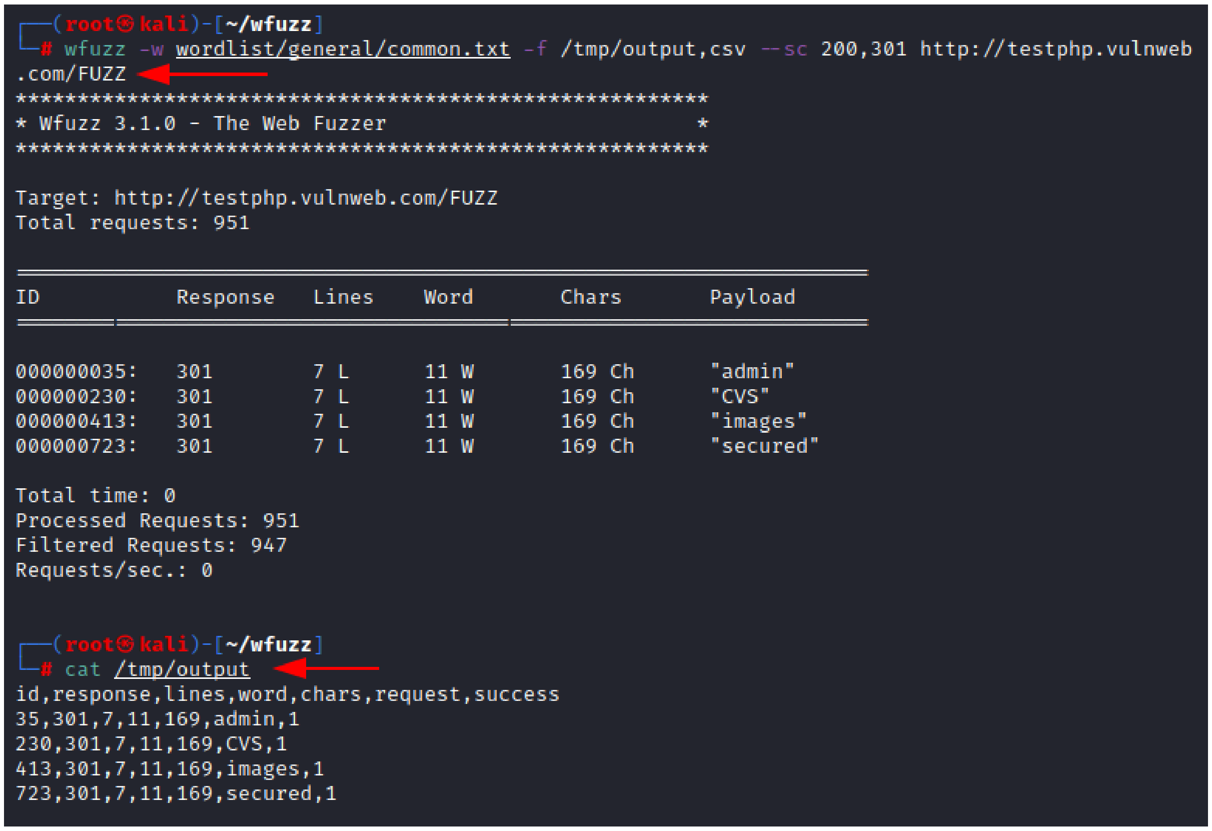
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/big.txt --sc 200,301 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 3024

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000119:   301        7 L      11 W       169 Ch      "admin"
000001337:   301        7 L      11 W       169 Ch      "images"
000002408:   301        7 L      11 W       169 Ch      "secured"
```

Saving fuzzing output

Wfuzz output can also be saved in multiple formats using the -f option.

-f option allows a user to input a file path and specify a printer (which formats the output) after a comma.

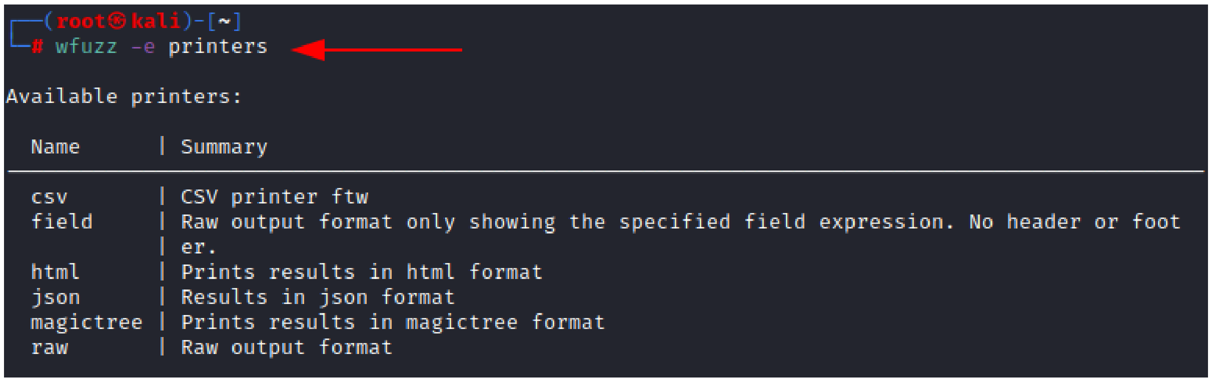
```
wfuzz -w wordlist/general/common.txt -f /tmp/output,csv --sc 200,301
http://testphp.vulnweb.com/FUZZ
cat /tmp/output
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/common.txt -f /tmp/output,csv --sc 200,301 http://testphp.vulnweb
.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000035:   301        7 L      11 W       169 Ch      "admin"
000000230:   301        7 L      11 W       169 Ch      "CVS"
000000413:   301        7 L      11 W       169 Ch      "images"
000000723:   301        7 L      11 W       169 Ch      "secured"

Total time: 0
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 0


┌──(root㉿kali)-[~/wfuzz]
└─# cat /tmp/output
id,response,lines,word,chars,request,success
35,301,7,11,169,admin,1
230,301,7,11,169,CVS,1
413,301,7,11,169,images,1
723,301,7,11,169,secured,1
```

In place of csv, you can specify any one of the printers

wfuzz -e printers



```
┌──(root㉿kali)-[~]
└─# wfuzz -e printers

Available printers:

  Name        | Summary
--------------------------------------------------------------------------------------------------
  csv         | CSV printer ftw
  field       | Raw output format only showing the specified field expression. No header or foot
              | er.
  html        | Prints results in html format
  json        | Results in json format
  magictree   | Prints results in magictree format
  raw         | Raw output format
```

Basic wordlist filters

There are certain sub-arguments that can be preceded by -z or -w filter to play around more with. These filters are:

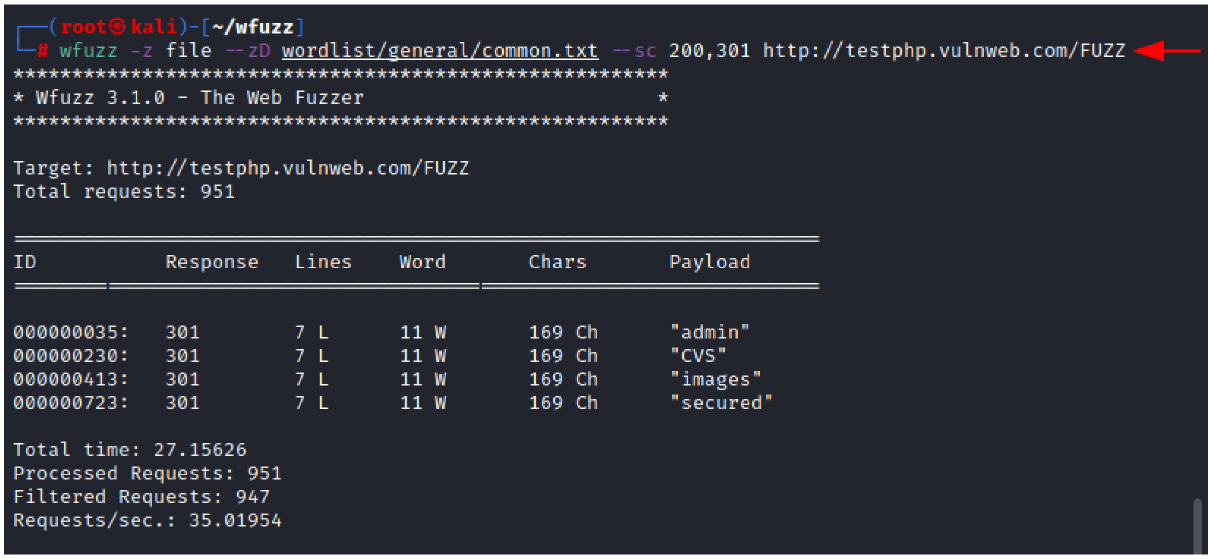
--zP <params>: Arguments for the specified payload

--zD <default>: Default parameter for the specified payload

--zE <encoder>: Encoder for the specified payload

So, to specify a wordlist with the payload, we can do it like so:

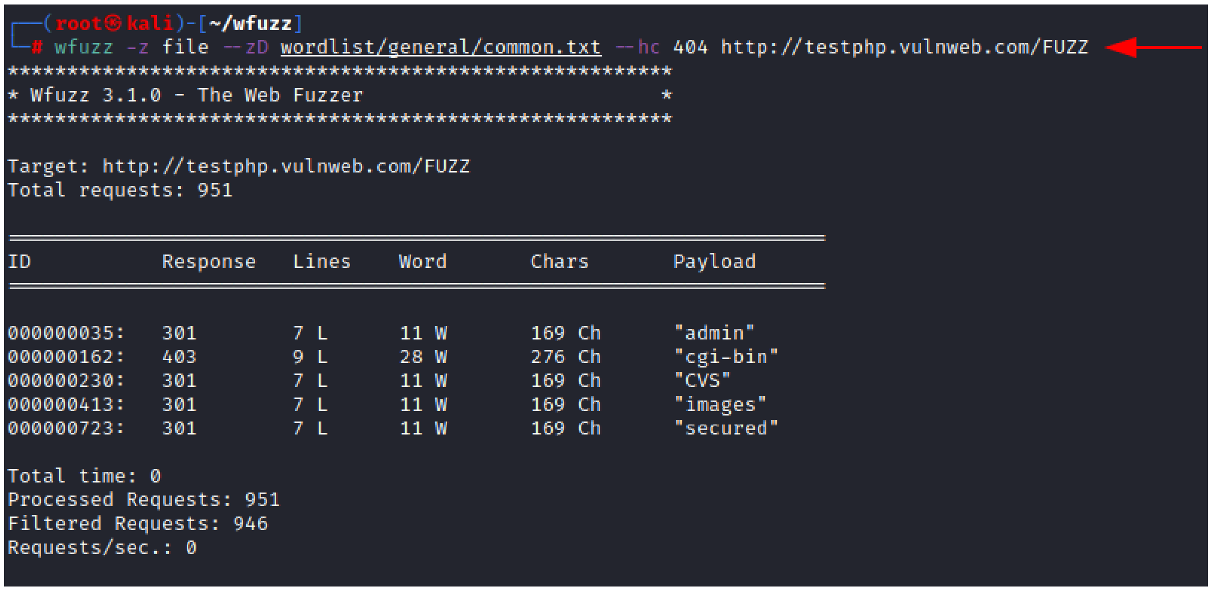
```
wfuzz -z file --zD wordlist/general/common.txt --sc 200,301 http://testphp.vulnweb.com/FUZZ
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file --zD wordlist/general/common.txt --sc 200,301 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000035:   301        7 L      11 W       169 Ch      "admin"
000000230:   301        7 L      11 W       169 Ch      "CVS"
000000413:   301        7 L      11 W       169 Ch      "images"
000000723:   301        7 L      11 W       169 Ch      "secured"

Total time: 27.15626
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 35.01954
```

To hide the HTTP response code 404, the same can be obtained like so:

```
wfuzz -z file --zD wordlist/general/common.txt --hc 404 http://testphp.vulnweb.com/FUZZ
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file --zD wordlist/general/common.txt --hc 404 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000035:   301        7 L      11 W       169 Ch      "admin"
000000162:   403        9 L      28 W       276 Ch      "cgi-bin"
000000230:   301        7 L      11 W       169 Ch      "CVS"
000000413:   301        7 L      11 W       169 Ch      "images"
000000723:   301        7 L      11 W       169 Ch      "secured"

Total time: 0
Processed Requests: 951
Filtered Requests: 946
Requests/sec.: 0
```

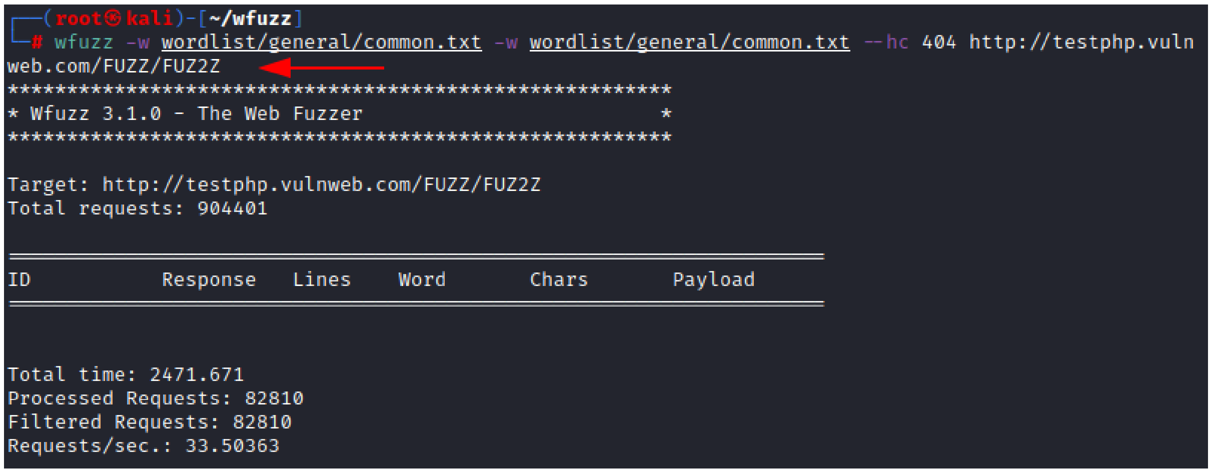
Double fuzzing

Just like a parameter in a payload can be fuzzed using the keyword "FUZZ" multiple fuzzing is also possible by specifying keywords:

- FUZ2Z - 2nd parameter
- FUZ3Z - 3rd parameter
- FUZ4Z - 4th parameter

And each parameter can be allotted its own wordlist. The first "-w" stands for first FUZZ. Second "-w" holds for second FUZ2Z and so on.

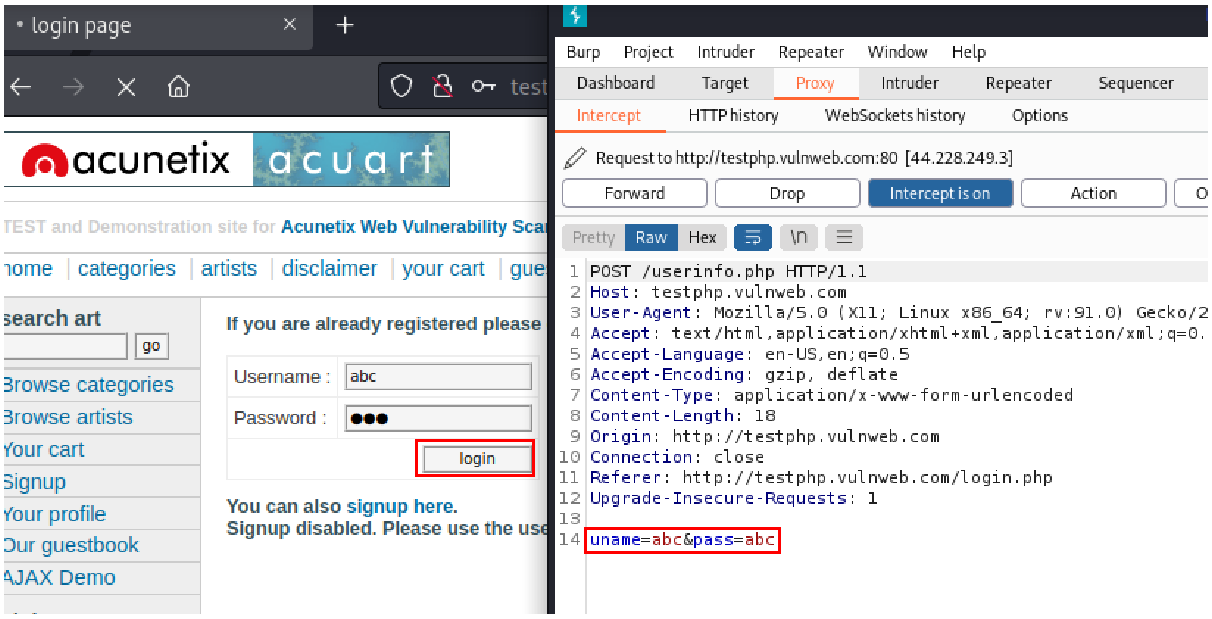
```
wfuzz -w wordlist/general/common.txt -w wordlist/general/common.txt --hc 404
http://testphp.vulnweb.com/FUZZ/FUZ2Z
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/common.txt -w wordlist/general/common.txt --hc 404 http://testphp.vuln
web.com/FUZZ/FUZ2Z
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ/FUZ2Z
Total requests: 904401

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================


Total time: 2471.671
Processed Requests: 82810
Filtered Requests: 82810
Requests/sec.: 33.50363
```

Login bruteforce

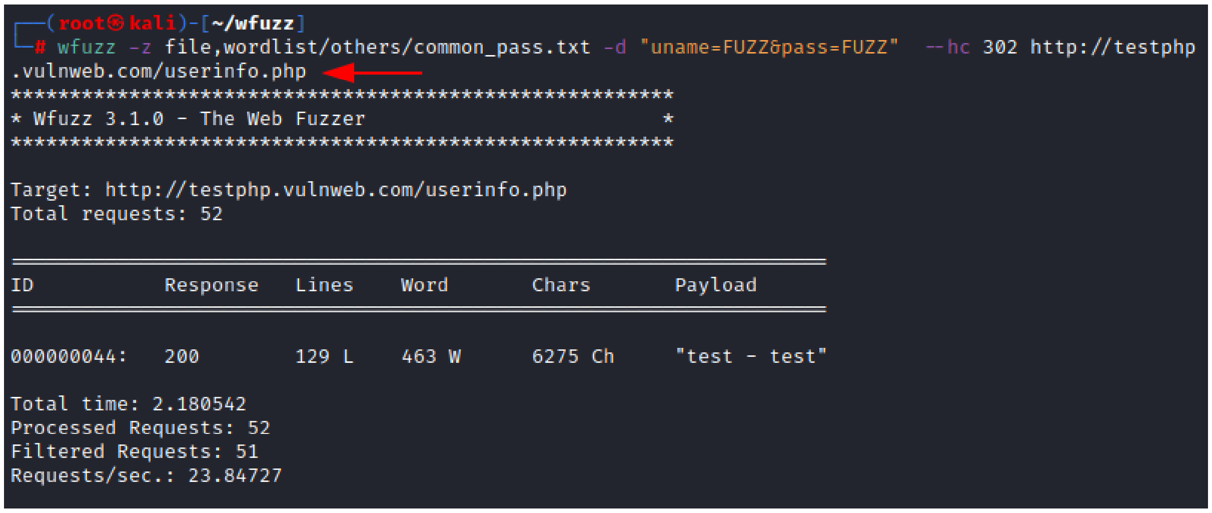
HTTP responses can be brute-forced using wfuzz. For example, testphp's website makes a POST request to the backend and passes "uname" and "pass" as the arguments to a page userinfo.php


login page
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scan
home | categories | artists | disclaimer | your cart | gue
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
If you are already registered please
Username : abc
Password : ●●●
login
You can also signup here.
Signup disabled. Please use the use
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer
Intercept HTTP history WebSockets history Options
Request to http://testphp.vulnweb.com:80 [44.228.249.3]
Forward Drop Intercept is on Action
Pretty Raw Hex

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/2
Accept: text/html,application/xhtml+xml,application/xml;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: http://testphp.vulnweb.com
Connection: close
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1

uname=abc&pass=abc
```

The same can be implemented using wfuzz like so:

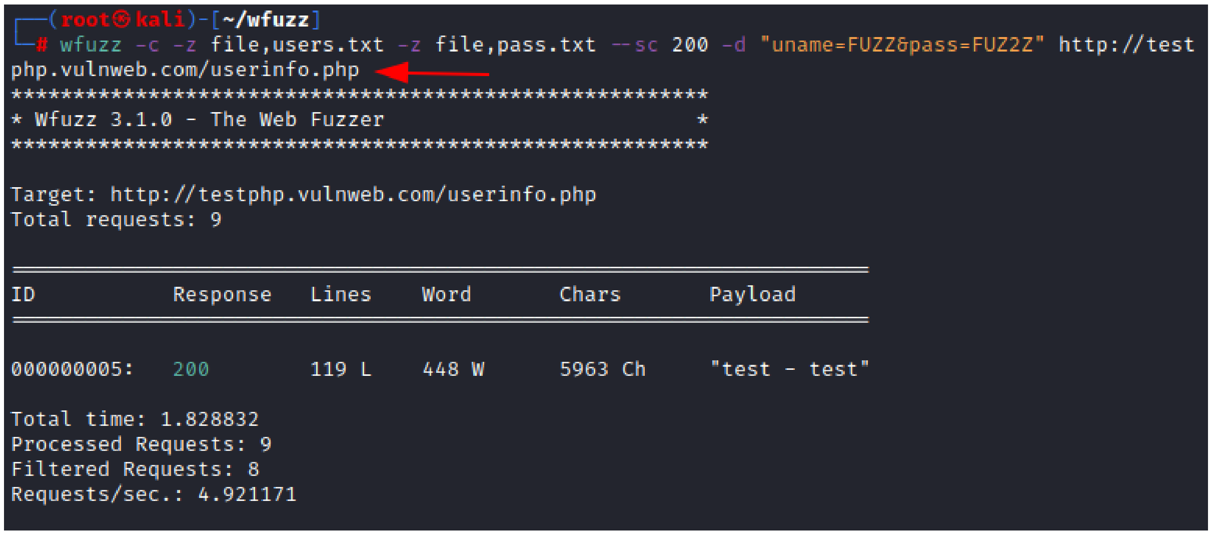
```
wfuzz -z file,wordlist/others/common_pass.txt -d "uname=FUZZ&pass=FUZZ"  --hc 302 http://testphp.vulnweb.com/userinfo.php
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/others/common_pass.txt -d "uname=FUZZ&pass=FUZZ"  --hc 302 http://testphp
.vulnweb.com/userinfo.php
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/userinfo.php
Total requests: 52

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000044:   200        129 L    463 W      6275 Ch     "test - test"

Total time: 2.180542
Processed Requests: 52
Filtered Requests: 51
Requests/sec.: 23.84727
```

As you can see, the correct credentials “test-test” have been found. We used a common file for both username and password. The same can be done by providing different files for both usernames and passwords like so:

```
wfuzz -z file,users.txt -z file,pass.txt --sc 200 -d "uname=FUZZ&pass=FUZ2Z" http://testphp.vulnweb.com/userinfo.php
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -c -z file,users.txt -z file,pass.txt --sc 200 -d "uname=FUZZ&pass=FUZ2Z" http://test
php.vulnweb.com/userinfo.php
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/userinfo.php
Total requests: 9

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000005:   200        119 L    448 W      5963 Ch     "test - test"

Total time: 1.828832
Processed Requests: 9
Filtered Requests: 8
Requests/sec.: 4.921171
```

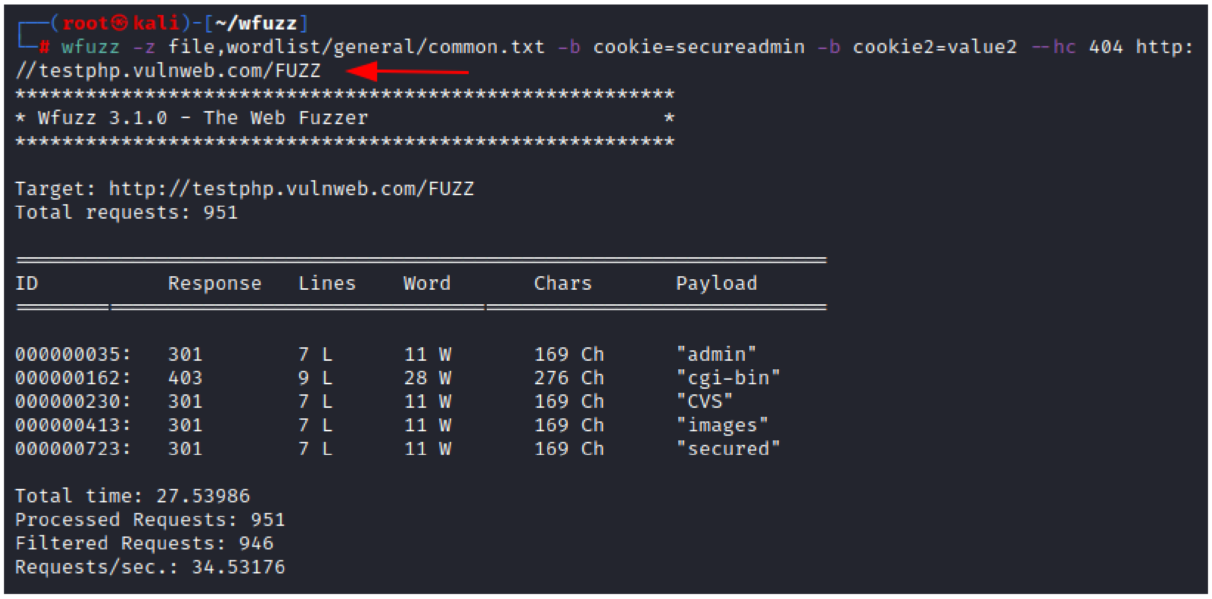
Cookie fuzzing

To send a custom cookie along a request to different fuzzed directories we can use the "-b" plug. This would add a cookie to the sent HTTP request.

Scenario useful:

- Cookie poisoning
- Session hijacking
- Privilege Escalation

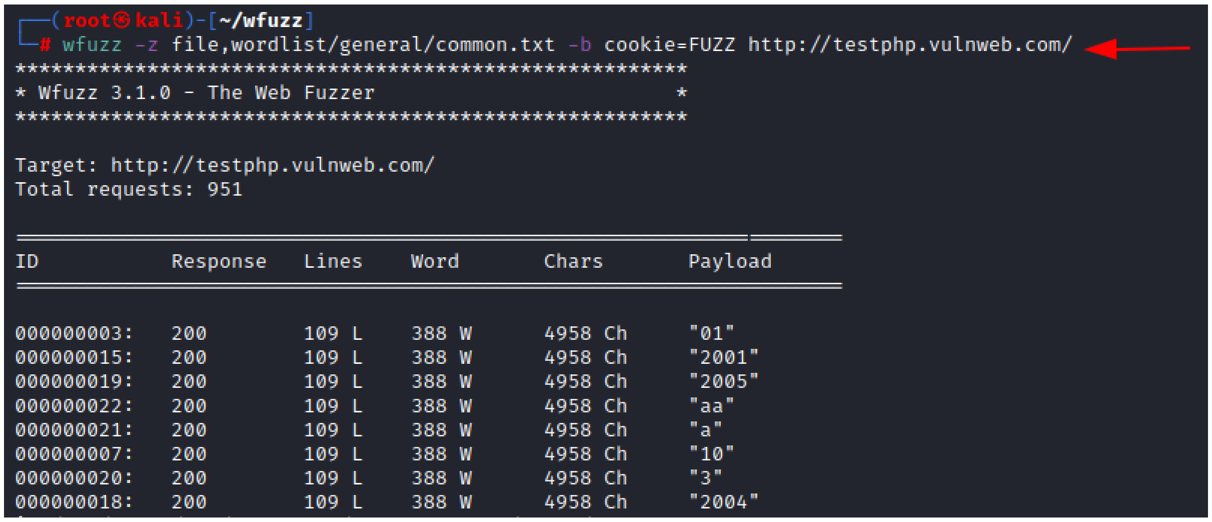
wfuzz -z file,wordlist/general/common.txt -b cookie=secureadmin -b cookie2=value2 --hc 404 http://testphp.vulnweb.com/FUZZ



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -b cookie=secureadmin -b cookie2=value2 --hc 404 http:
//testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000035:   301        7 L      11 W       169 Ch      "admin"
000000162:   403        9 L      28 W       276 Ch      "cgi-bin"
000000230:   301        7 L      11 W       169 Ch      "CVS"
000000413:   301        7 L      11 W       169 Ch      "images"
000000723:   301        7 L      11 W       169 Ch      "secured"

Total time: 27.53986
Processed Requests: 951
Filtered Requests: 946
Requests/sec.: 34.53176
```

In the above scenario, we have added 2 static cookies on multiple directories. Now, we can also fuzz the cookie parameter too like so:

```
wfuzz -z file,wordlist/general/common.txt -b cookie=FUZZ http://testphp.vulnweb.com/
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -b cookie=FUZZ http://testphp.vulnweb.com/
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000003:   200        109 L    388 W      4958 Ch     "01"
000000015:   200        109 L    388 W      4958 Ch     "2001"
000000019:   200        109 L    388 W      4958 Ch     "2005"
000000022:   200        109 L    388 W      4958 Ch     "aa"
000000021:   200        109 L    388 W      4958 Ch     "a"
000000007:   200        109 L    388 W      4958 Ch     "10"
000000020:   200        109 L    388 W      4958 Ch     "3"
000000018:   200        109 L    388 W      4958 Ch     "2004"
```

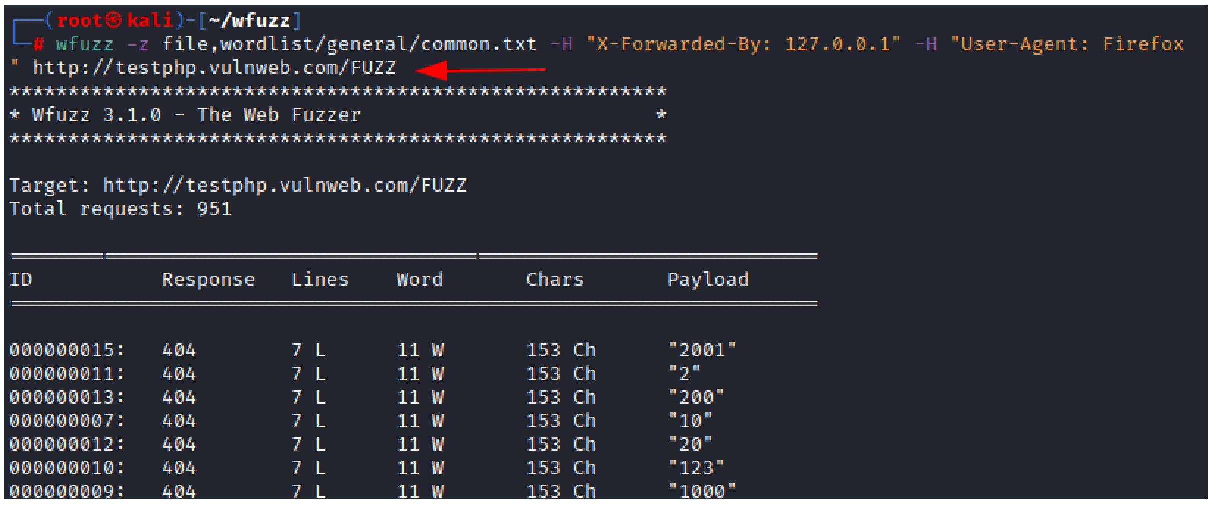
Header fuzzing

HTTP header can be added in a request being sent out by wfuzz. HTTP headers can change the behavior of an entire web page. Custom headers can be fuzzed or injected in an outgoing request

Scenarios useful:

- HTTP Header Injections
- SQL Injections
- Host Header Injections

```
wfuzz -z file,wordlist/general/common.txt -H "X-Forwarded-By: 127.0.0.1" -H "User-Agent: Firefox" http://testphp.vulnweb.com/FUZZ
```



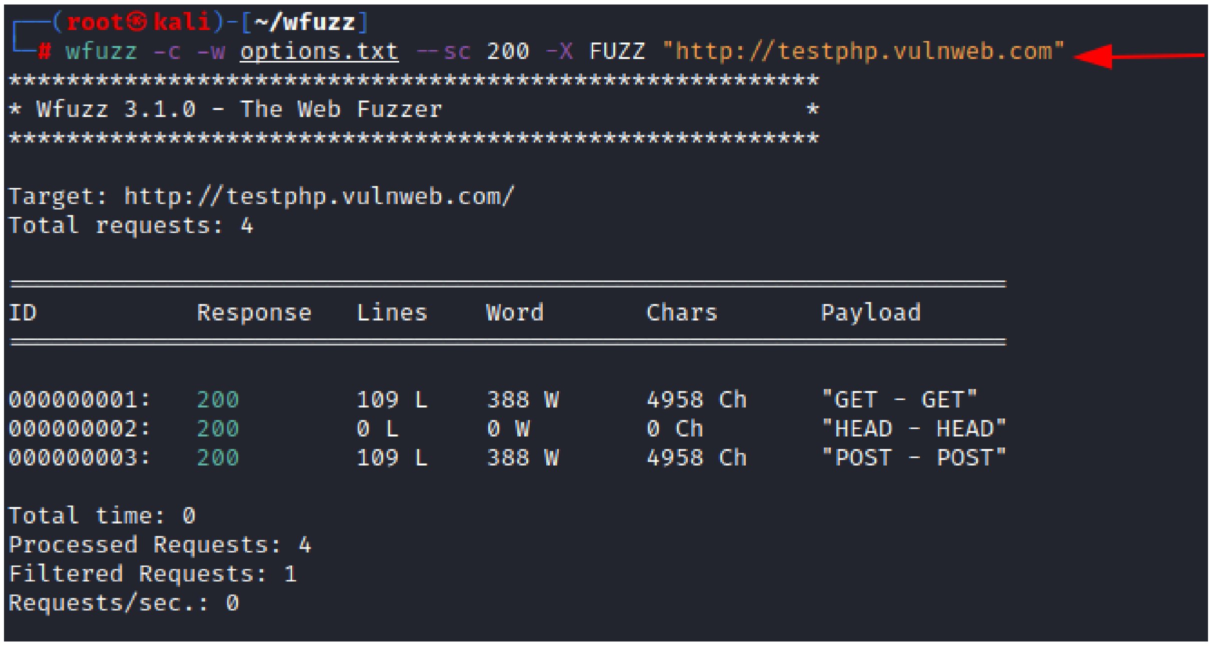
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -H "X-Forwarded-By: 127.0.0.1" -H "User-Agent: Firefox
" http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000015:   404        7 L      11 W       153 Ch      "2001"
000000011:   404        7 L      11 W       153 Ch      "2"
000000013:   404        7 L      11 W       153 Ch      "200"
000000007:   404        7 L      11 W       153 Ch      "10"
000000012:   404        7 L      11 W       153 Ch      "20"
000000010:   404        7 L      11 W       153 Ch      "123"
000000009:   404        7 L      11 W       153 Ch      "1000"
```

HTTP OPTIONS fuzzing

There are various HTTP Request/Options methods available which can be specified by using the "-X" flag. In the following example, We have inserted the following options in a text file called options.txt

- GET
- HEAD
- POST
- PUT
- DELETE
- CONNECT
- OPTIONS
- TRACE
- PATCH

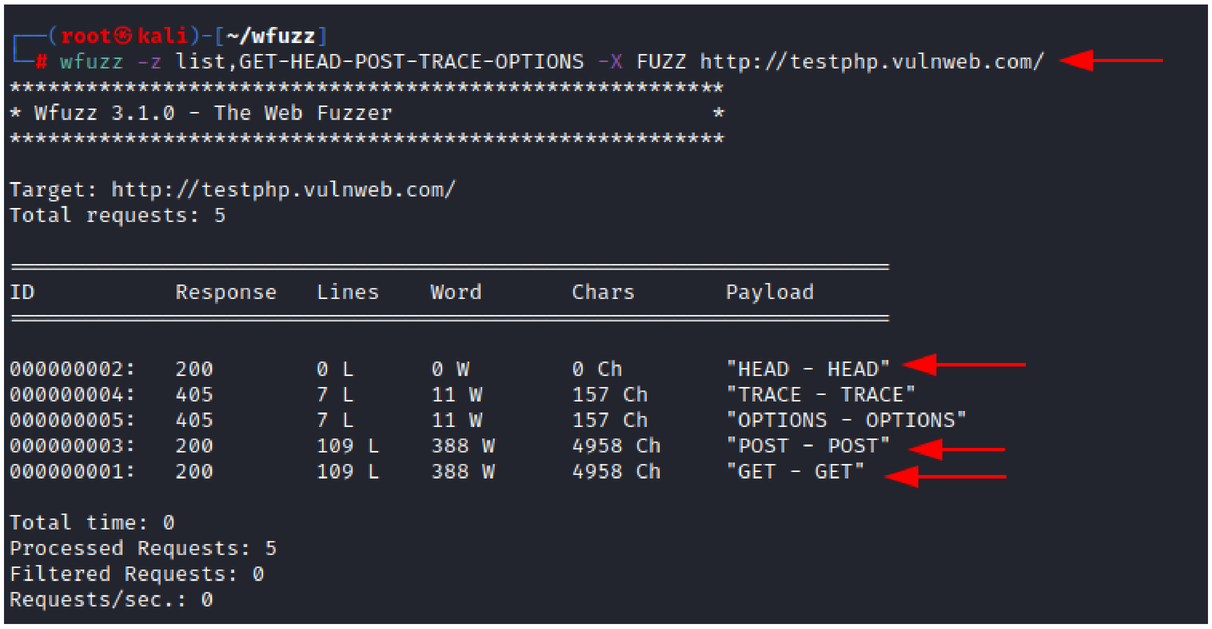
```
wfuzz -c -w options.txt --sc 200 -X FUZZ "http://testphp.vulnweb.com"
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -c -w options.txt --sc 200 -X FUZZ "http://testphp.vulnweb.com"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/
Total requests: 4

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   200        109 L    388 W      4958 Ch     "GET - GET"
000000002:   200        0 L      0 W        0 Ch        "HEAD - HEAD"
000000003:   200        109 L    388 W      4958 Ch     "POST - POST"

Total time: 0
Processed Requests: 4
Filtered Requests: 1
Requests/sec.: 0
```

As you could see, three valid options returned a 200 response code.

```
wfuzz -z list,GET-HEAD-POST-TRACE-OPTIONS -X FUZZ http://testphp.vulnweb.com/
```

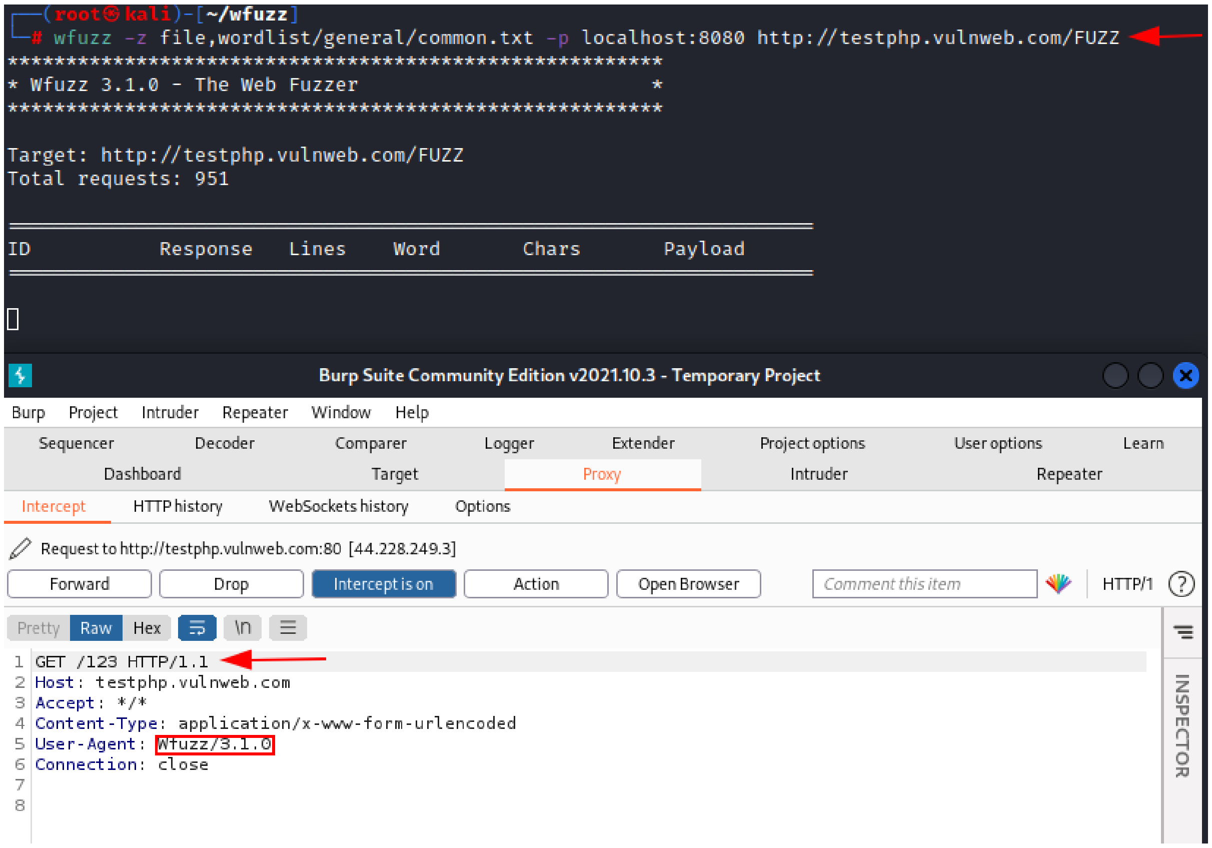


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,GET-HEAD-POST-TRACE-OPTIONS -X FUZZ http://testphp.vulnweb.com/
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/
Total requests: 5

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000002:   200        0 L      0 W        0 Ch        "HEAD - HEAD"
000000004:   405        7 L      11 W       157 Ch      "TRACE - TRACE"
000000005:   405        7 L      11 W       157 Ch      "OPTIONS - OPTIONS"
000000003:   200        109 L    388 W      4958 Ch     "POST - POST"
000000001:   200        109 L    388 W      4958 Ch     "GET - GET"

Total time: 0
Processed Requests: 5
Filtered Requests: 0
Requests/sec.: 0
```

Fuzzing through Proxy

Wfuzz can also route the requests through a proxy. In the following example, a Burp proxy is active on port 8080 and the request intercepted in the burp intercept as you can see.

```
wfuzz -z file,wordlist/general/common.txt -p localhost:8080 http://testphp.vulnweb.com/FUZZ
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -p localhost:8080 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
```

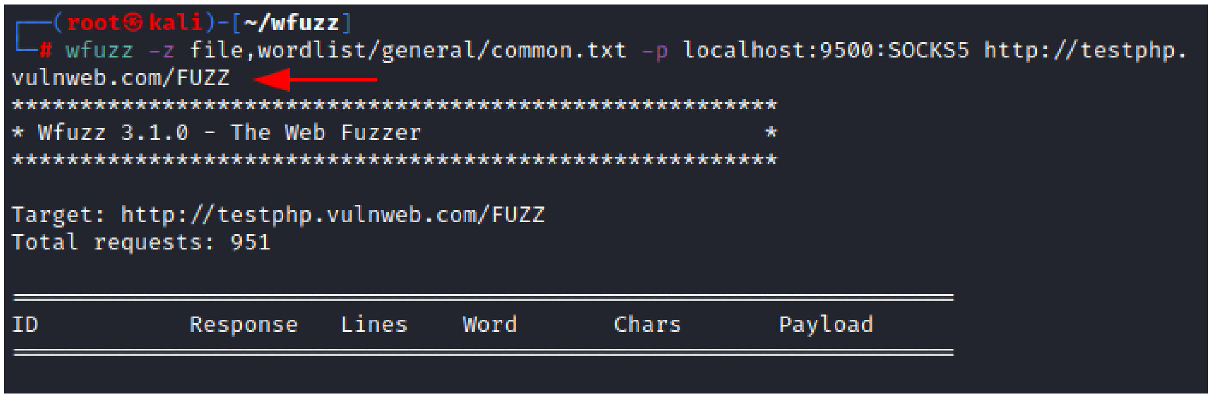
Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer Decoder Comparer Logger Extender Project options User options Learn
Dashboard Target Proxy Intruder Repeater
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Comment this item
HTTP/1
Pretty Raw Hex \n
INSPECTOR

Request to http://testphp.vulnweb.com:80 [44.228.249.3]

```
GET /123 HTTP/1.1
Host: testphp.vulnweb.com
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Wfuzz/3.1.0
Connection: close
```

The same can also be achieved with SOCKS proxy like so:

```
wfuzz -z file,wordlist/general/common.txt -p localhost:9500:SOCKS5
http://testphp.vulnweb.com/FUZZ
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -p localhost:9500:SOCKS5 http://testphp.
vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
```

Authentication fuzz

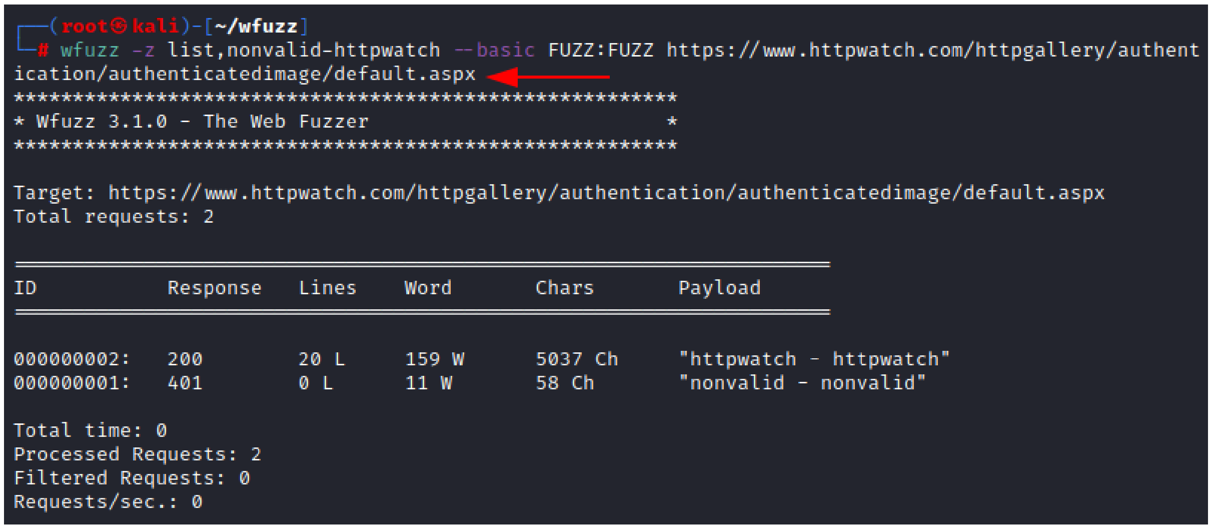
Wfuzz can also set authentication headers and provide means of authentication through HTTP requests.

Flags:

- --basic: provides basic Username and Password auth
- --ntlm: windows auth
- --digest: web server negotiation through digest access

In the following example, I am providing a list inline with two variables and --basic input to bruteforce a website httpwatch.com

```
wfuzz -z list,nonvalid-httpwatch --basic FUZZ:FUZZ
https://www.httpwatch.com/httpgallery/authentication/authenticatedimage/default.aspx
```



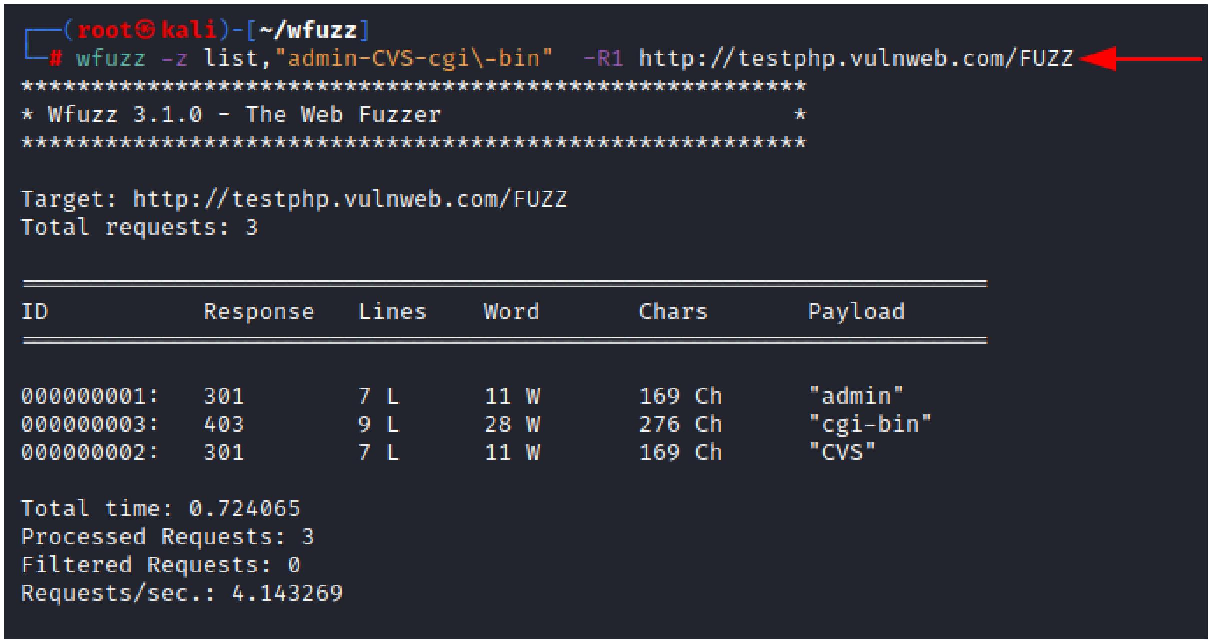
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,nonvalid-httpwatch --basic FUZZ:FUZZ https://www.httpwatch.com/httpgallery/authent
ication/authenticatedimage/default.aspx
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://www.httpwatch.com/httpgallery/authentication/authenticatedimage/default.aspx
Total requests: 2

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000002:   200        20 L     159 W      5037 Ch     "httpwatch - httpwatch"
000000001:   401        0 L      11 W       58 Ch       "nonvalid - nonvalid"

Total time: 0
Processed Requests: 2
Filtered Requests: 0
Requests/sec.: 0
```

Recursive fuzz

-R switch can specify the levels of recursion while fuzzing directories or parameters. Recursion in simple terms means fuzzing at multiple different levels of directories like /dir/dir/dir etc

In the following example, we are recursing at level 1 with a list inline containing 3 directories: admin, CVS and cgi-bin. Note how a directory with - in its name can be supplied inline

```
wfuzz -z list,"admin-CVS-cgi\-bin" -R1 http://testphp.vulnweb.com/FUZZ
```

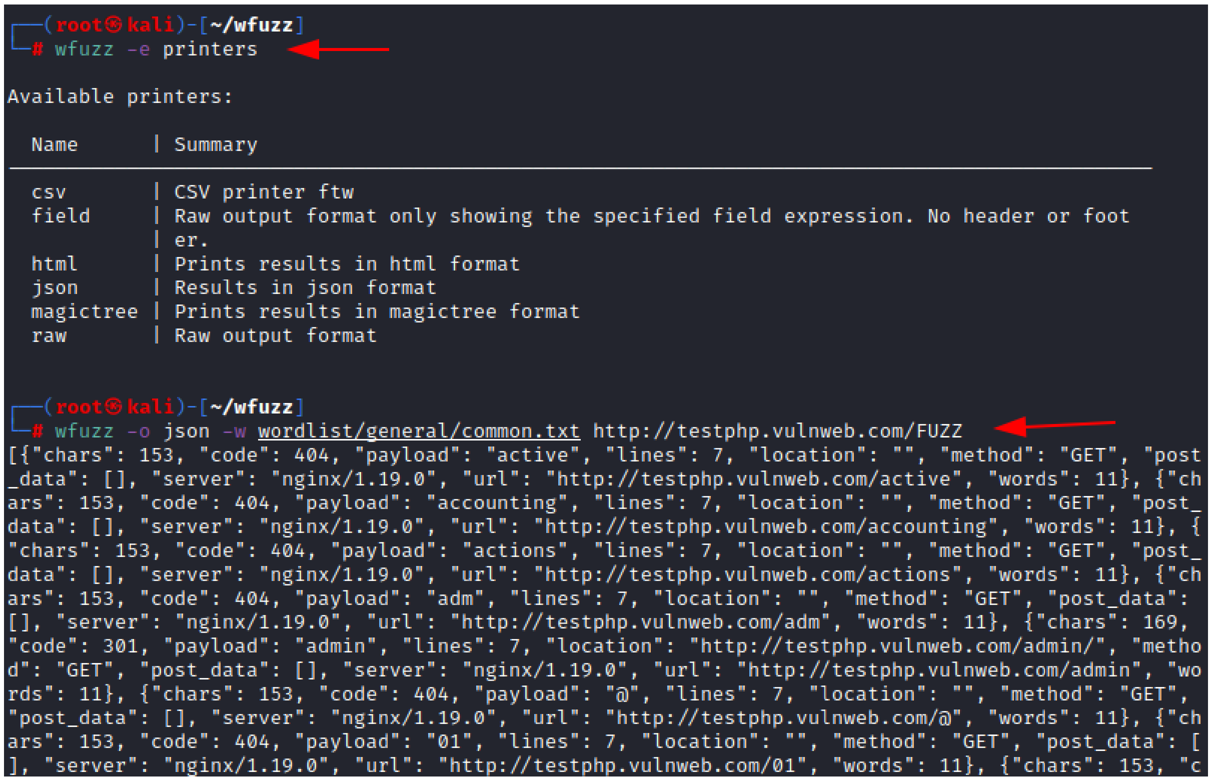


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,"admin-CVS-cgi\-bin"  -R1 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 3

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   301        7 L      11 W       169 Ch      "admin"
000000003:   403        9 L      28 W       276 Ch      "cgi-bin"
000000002:   301        7 L      11 W       169 Ch      "CVS"

Total time: 0.724065
Processed Requests: 3
Filtered Requests: 0
Requests/sec.: 4.143269
```

Printers and output

Printers in wfuzz refers to all the formats a payload's output can be processed as. It can be viewed using -e succeeded by printers argument. Furthermore, "-o" flag can specify the format of the output too

```
wfuzz -e printers
wfuzz -o json -w wordlist/general/common.txt http://testphp.vulnweb.com/FUZZ
```

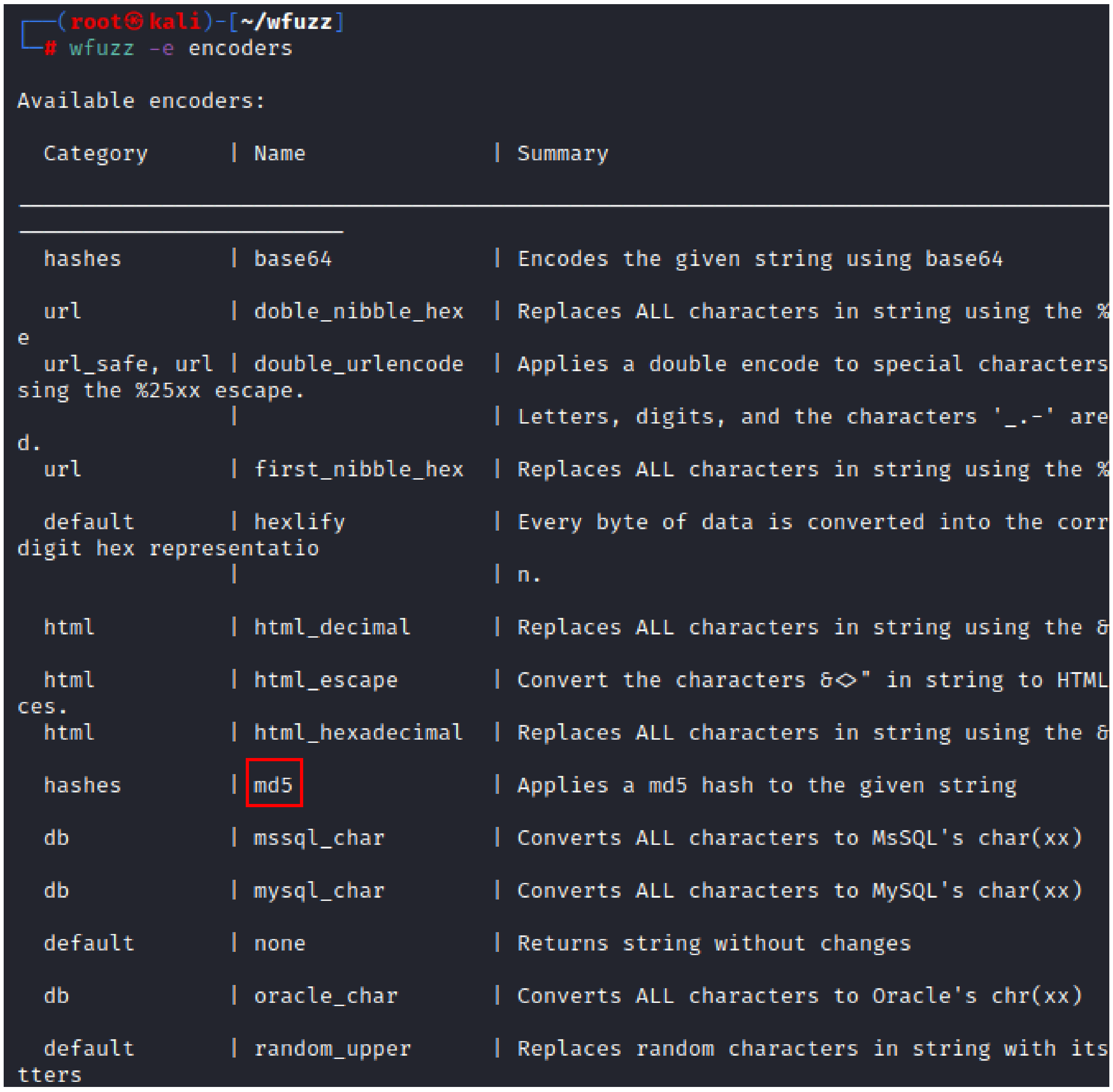


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -e printers

Available printers:

  Name        | Summary
 ─────────────────────────────────────────────────────────────────────────────────────────────────

  csv         | CSV printer ftw
  field       | Raw output format only showing the specified field expression. No header or foot
              | er.
  html        | Prints results in html format
  json        | Results in json format
  magictree   | Prints results in magictree format
  raw         | Raw output format


┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -o json -w wordlist/general/common.txt http://testphp.vulnweb.com/FUZZ
[{"chars": 153, "code": 404, "payload": "active", "lines": 7, "location": "", "method": "GET", "post
_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/active", "words": 11}, {"ch
ars": 153, "code": 404, "payload": "accounting", "lines": 7, "location": "", "method": "GET", "post_
data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/accounting", "words": 11}, {
"chars": 153, "code": 404, "payload": "actions", "lines": 7, "location": "", "method": "GET", "post_
data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/actions", "words": 11}, {"ch
ars": 153, "code": 404, "payload": "adm", "lines": 7, "location": "", "method": "GET", "post_data":
[], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/adm", "words": 11}, {"chars": 169,
"code": 301, "payload": "admin", "lines": 7, "location": "http://testphp.vulnweb.com/admin/", "metho
d": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/admin", "wo
rds": 11}, {"chars": 153, "code": 404, "payload": "@", "lines": 7, "location": "", "method": "GET",
"post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/@", "words": 11}, {"ch
ars": 153, "code": 404, "payload": "01", "lines": 7, "location": "", "method": "GET", "post_data": [
], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/01", "words": 11}, {"chars": 153, "c
```

Encoders

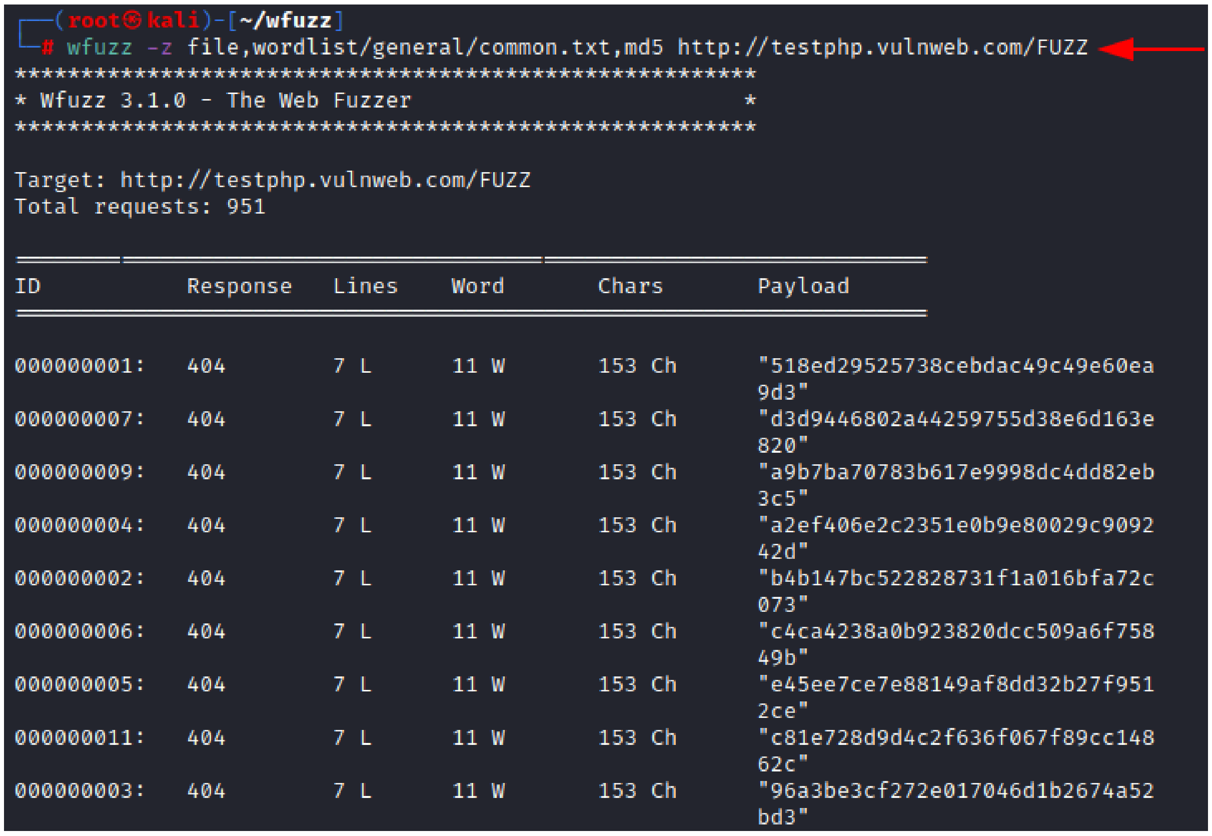
Various encoders are available in wfuzz. One such encoder we saw earlier was md5. Other encoders can be viewed by using “-e” flag with encoders argument.

```
wfuzz -e encoders
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -e encoders

Available encoders:

  Category      | Name              | Summary
────────────────────────────────────────────────────────────────────────────────
─────────────────────────
  hashes        | base64            | Encodes the given string using base64

  url           | doble_nibble_hex  | Replaces ALL characters in string using the %
e
  url_safe, url | double_urlencode  | Applies a double encode to special characters
sing the %25xx escape.
                |                   | Letters, digits, and the characters '_.-' are
d.
  url           | first_nibble_hex  | Replaces ALL characters in string using the %

  default       | hexlify           | Every byte of data is converted into the corr
digit hex representatio
                |                   | n.

  html          | html_decimal      | Replaces ALL characters in string using the &

  html          | html_escape       | Convert the characters &<>" in string to HTML
ces.
  html          | html_hexadecimal  | Replaces ALL characters in string using the &

  hashes        | md5               | Applies a md5 hash to the given string

  db            | mssql_char        | Converts ALL characters to MsSQL's char(xx)

  db            | mysql_char        | Converts ALL characters to MySQL's char(xx)

  default       | none              | Returns string without changes

  db            | oracle_char       | Converts ALL characters to Oracle's chr(xx)

  default       | random_upper      | Replaces random characters in string with its
tters
```

```
wfuzz -z file,wordlist/general/common.txt,md5 http://testphp.vulnweb.com/FUZZ
```

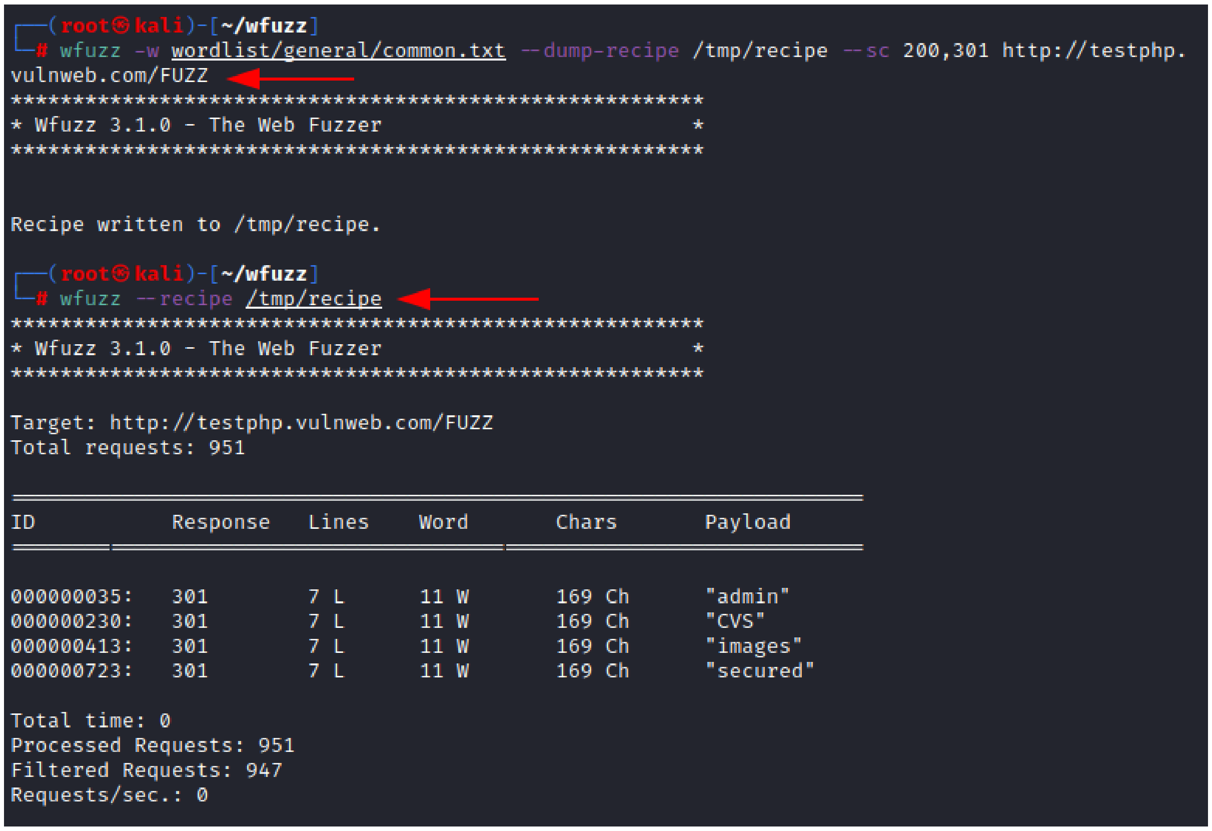


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt,md5 http://testphp.vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   404        7 L      11 W       153 Ch      "518ed29525738cebdac49c49e60ea
                                                        9d3"
000000007:   404        7 L      11 W       153 Ch      "d3d9446802a44259755d38e6d163e
                                                        820"
000000009:   404        7 L      11 W       153 Ch      "a9b7ba70783b617e9998dc4dd82eb
                                                        3c5"
000000004:   404        7 L      11 W       153 Ch      "a2ef406e2c2351e0b9e80029c9092
                                                        42d"
000000002:   404        7 L      11 W       153 Ch      "b4b147bc522828731f1a016bfa72c
                                                        073"
000000006:   404        7 L      11 W       153 Ch      "c4ca4238a0b923820dcc509a6f758
                                                        49b"
000000005:   404        7 L      11 W       153 Ch      "e45ee7ce7e88149af8dd32b27f951
                                                        2ce"
000000011:   404        7 L      11 W       153 Ch      "c81e728d9d4c2f636f067f89cc148
                                                        62c"
000000003:   404        7 L      11 W       153 Ch      "96a3be3cf272e017046d1b2674a52
                                                        bd3"
```

Storing and restoring fuzz from recipes

To make scanning easy, wfuzz can save and restore sessions using the "--dump-recipe" and "--recipe" flag.

```
wfuzz -w wordlist/general/common.txt --dump-recipe /tmp/recipe --sc 200,301
http://testphp.vulnweb.com/FUZZ
wfuzz --recipe /tmp/recipe
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/common.txt --dump-recipe /tmp/recipe --sc 200,301 http://testphp.
vulnweb.com/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************


Recipe written to /tmp/recipe.

┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz --recipe /tmp/recipe
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/FUZZ
Total requests: 951

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000035:   301        7 L      11 W       169 Ch      "admin"
000000230:   301        7 L      11 W       169 Ch      "CVS"
000000413:   301        7 L      11 W       169 Ch      "images"
000000723:   301        7 L      11 W       169 Ch      "secured"

Total time: 0
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 0
```

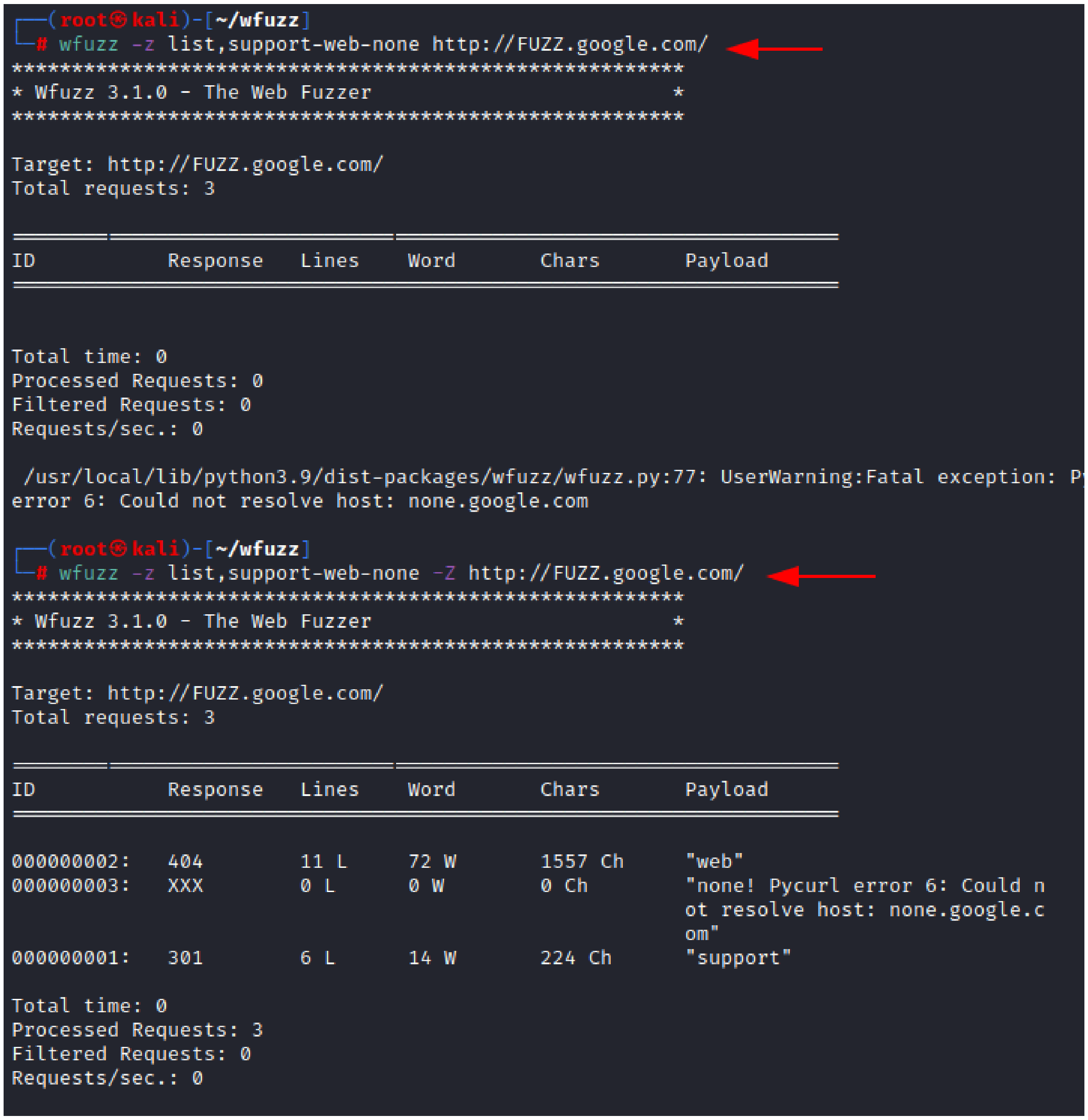
Ignoring exceptions and errors

Often while fuzzing, there are various errors and exceptions that a website can throw. "-Z" option can make wfuzz ignore these errors and exceptions. First, we run a normal subdomain fuzzing routine and then with -Z option:

```
wfuzz -z list,support-web-none http://FUZZ.google.com/
wfuzz -z list,support-web-none -Z http://FUZZ.google.com/
```

As you could see, -Z ignores that error on the bottom. Further, any invalid response can also be hidden like so:

```
wfuzz -z list,support-web-none -Z --hc "XXX" http://FUZZ.google.com/
```

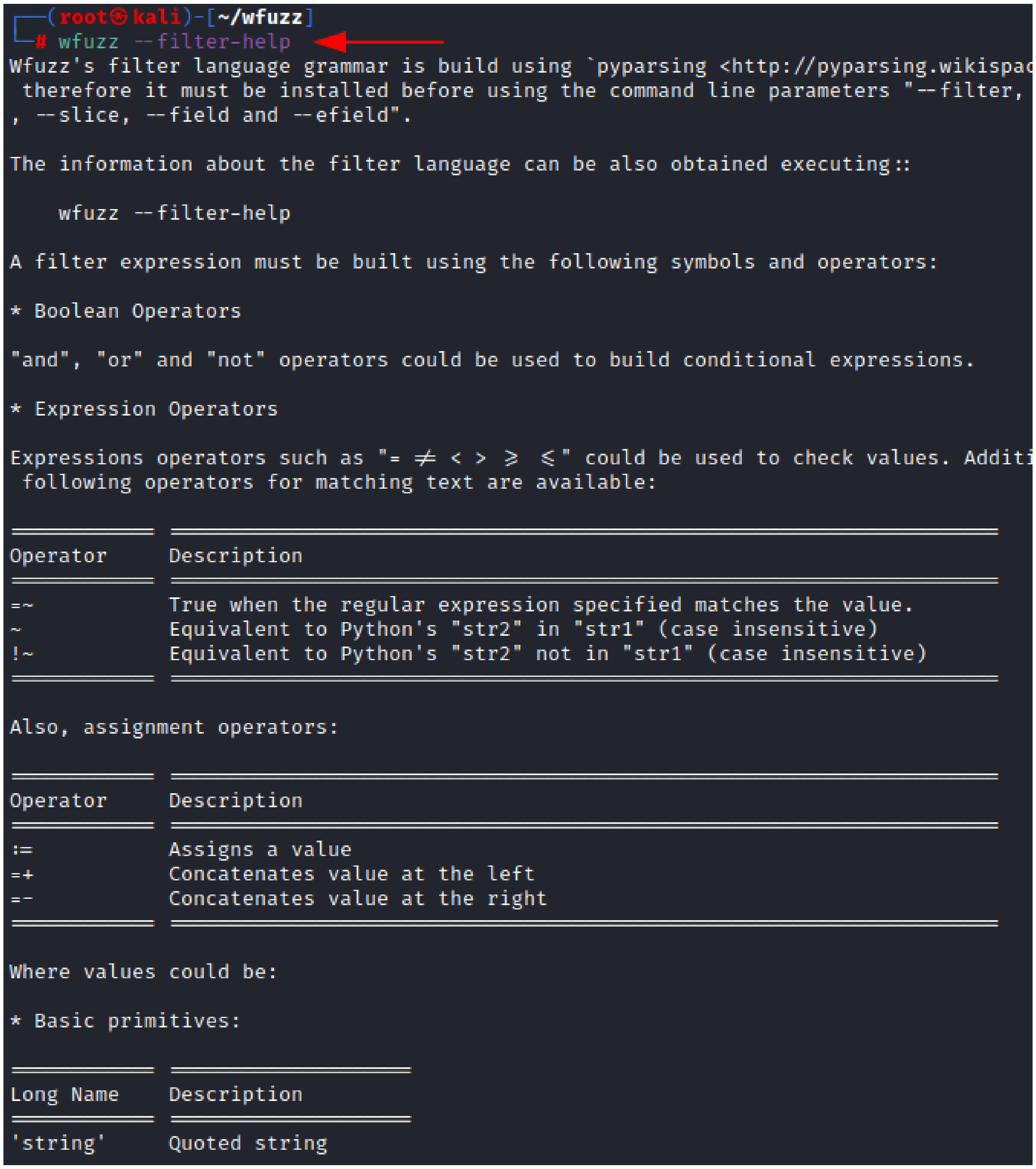


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,support-web-none http://FUZZ.google.com/
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://FUZZ.google.com/
Total requests: 3

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================


Total time: 0
Processed Requests: 0
Filtered Requests: 0
Requests/sec.: 0

 /usr/local/lib/python3.9/dist-packages/wfuzz/wfuzz.py:77: UserWarning:Fatal exception: Py
error 6: Could not resolve host: none.google.com

┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,support-web-none -Z http://FUZZ.google.com/
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://FUZZ.google.com/
Total requests: 3

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000002:   404        11 L     72 W       1557 Ch     "web"
000000003:   XXX        0 L      0 W        0 Ch        "none! Pycurl error 6: Could n
                                                        ot resolve host: none.google.c
                                                        om"
000000001:   301        6 L      14 W       224 Ch      "support"

Total time: 0
Processed Requests: 3
Filtered Requests: 0
Requests/sec.: 0
```

Filtering results

There are many filters available to manipulate a payload or output.

```
wfuzz --filter-help
```

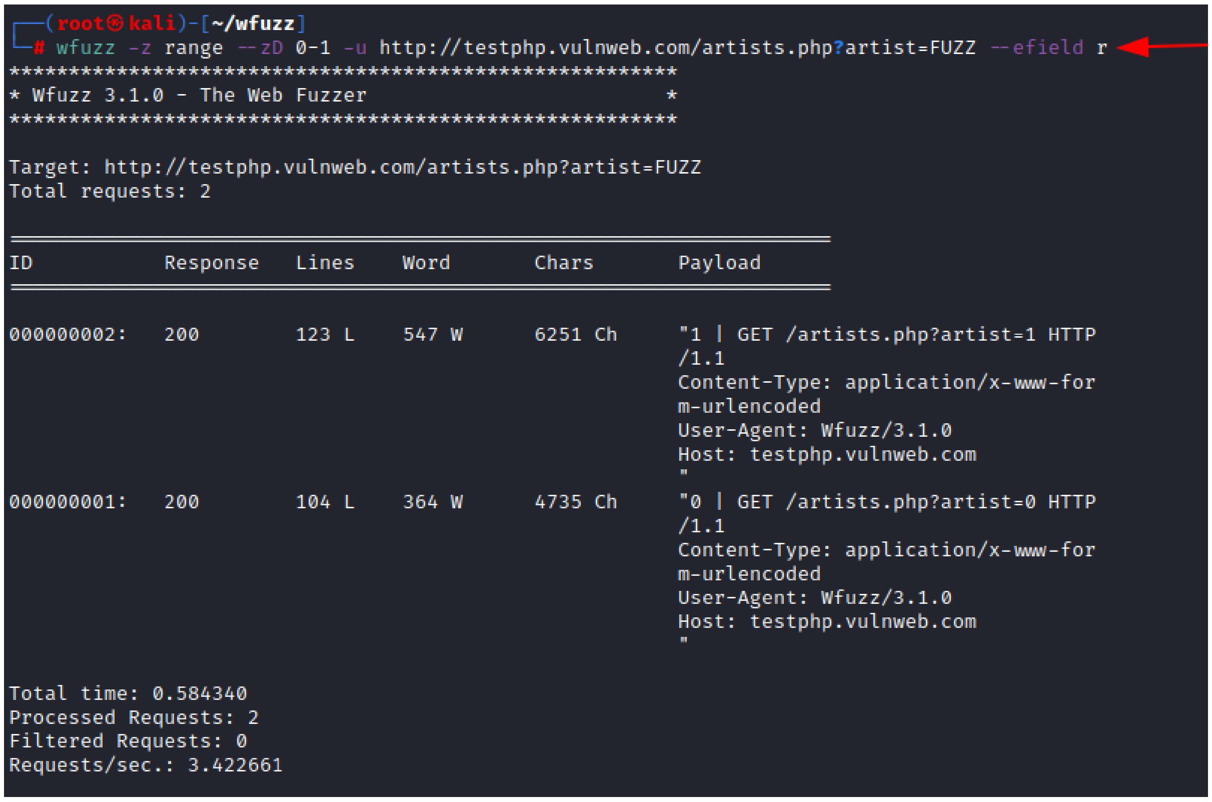


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz --filter-help
Wfuzz's filter language grammar is build using `pyparsing <http://pyparsing.wikispac
 therefore it must be installed before using the command line parameters "--filter,
, --slice, --field and --efield".

The information about the filter language can be also obtained executing::

    wfuzz --filter-help

A filter expression must be built using the following symbols and operators:

* Boolean Operators

"and", "or" and "not" operators could be used to build conditional expressions.

* Expression Operators

Expressions operators such as "= ≠ < > ⩾  ⩽" could be used to check values. Additi
 following operators for matching text are available:

============  ====================================================================
Operator      Description
============  ====================================================================
=~            True when the regular expression specified matches the value.
~             Equivalent to Python's "str2" in "str1" (case insensitive)
!~            Equivalent to Python's "str2" not in "str1" (case insensitive)
============  ====================================================================

Also, assignment operators:

============  ====================================================================
Operator      Description
============  ====================================================================
:=            Assigns a value
=+            Concatenates value at the left
=-            Concatenates value at the right
============  ====================================================================

Where values could be:

* Basic primitives:

============  ====================
Long Name     Description
============  ====================
'string'      Quoted string
```

These can be manipulated using "--filter, --slice, --field and --efield" arguments.

For example, to view raw responses of the payload sent and the complete HTTP request made, you can use "--efield r" option

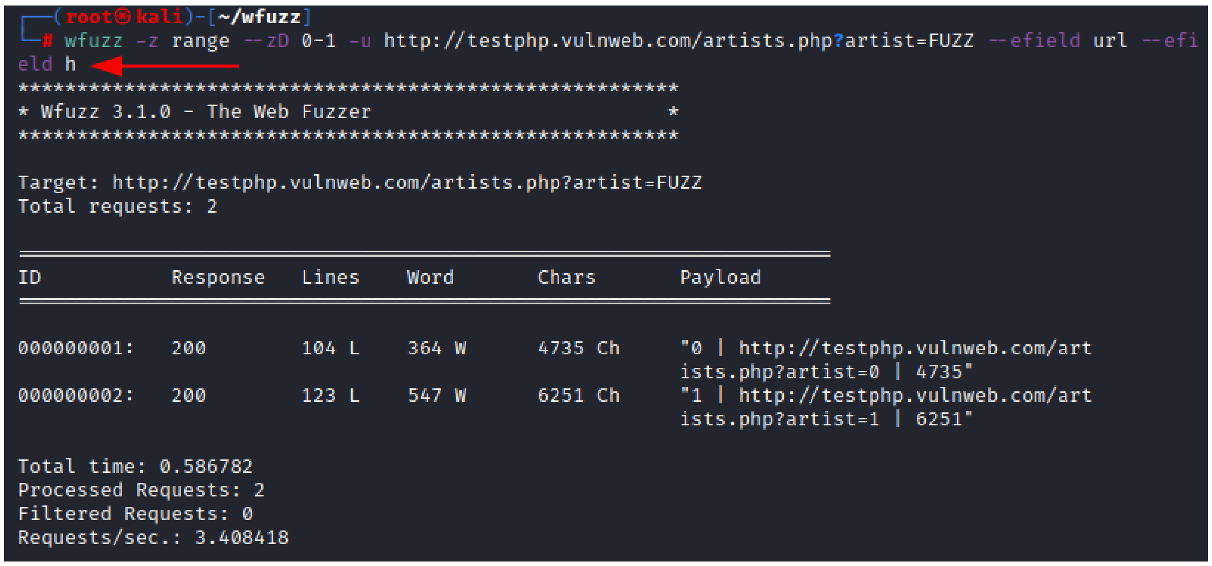
```
wfuzz -z range --zD 0-1 -u http://testphp.vulnweb.com/artists.php?artist=FUZZ --efield r
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z range --zD 0-1 -u http://testphp.vulnweb.com/artists.php?artist=FUZZ --efield r
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/artists.php?artist=FUZZ
Total requests: 2

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000002:   200        123 L    547 W      6251 Ch     "1 | GET /artists.php?artist=1 HTTP
                                                        /1.1
                                                        Content-Type: application/x-www-for
                                                        m-urlencoded
                                                        User-Agent: Wfuzz/3.1.0
                                                        Host: testphp.vulnweb.com
                                                        "
000000001:   200        104 L    364 W      4735 Ch     "0 | GET /artists.php?artist=0 HTTP
                                                        /1.1
                                                        Content-Type: application/x-www-for
                                                        m-urlencoded
                                                        User-Agent: Wfuzz/3.1.0
                                                        Host: testphp.vulnweb.com
                                                        "

Total time: 0.584340
Processed Requests: 2
Filtered Requests: 0
Requests/sec.: 3.422661
```

However, if only the intended URL is needed, one can do it by providing --efield url input.

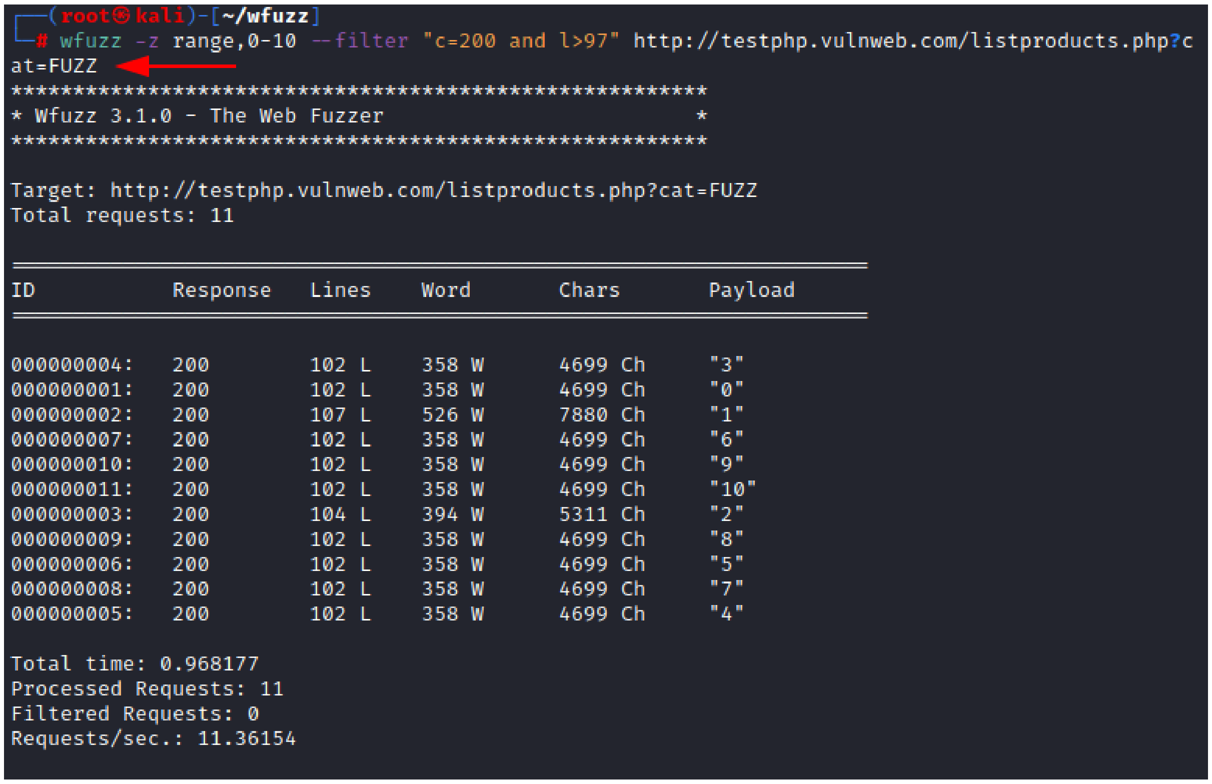
```
wfuzz -z range --zD 0-1 -u http://testphp.vulnweb.com/artists.php?artist=FUZZ --efield url --efield h
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z range --zD 0-1 -u http://testphp.vulnweb.com/artists.php?artist=FUZZ --efield url --efi
eld h
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/artists.php?artist=FUZZ
Total requests: 2

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   200        104 L    364 W      4735 Ch     "0 | http://testphp.vulnweb.com/art
                                                        ists.php?artist=0 | 4735"
000000002:   200        123 L    547 W      6251 Ch     "1 | http://testphp.vulnweb.com/art
                                                        ists.php?artist=1 | 6251"

Total time: 0.586782
Processed Requests: 2
Filtered Requests: 0
Requests/sec.: 3.408418
```

Similarly, to filter out results based on the response code and the length of the page (lines greater than 97), you can do it like:

```
wfuzz -z range,0-10 --filter "c=200 and l>97" http://testphp.vulnweb.com/listproducts.php?cat=FUZZ
```



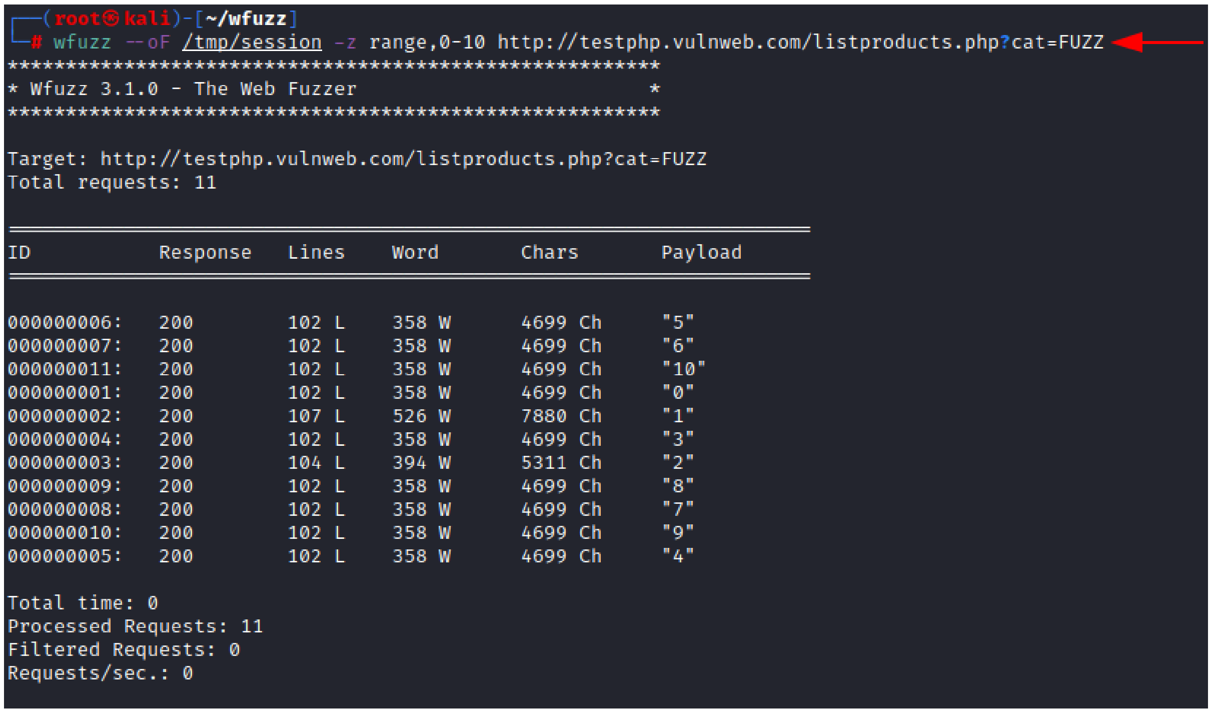
```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z range,0-10 --filter "c=200 and l>97" http://testphp.vulnweb.com/listproducts.php?c
at=FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/listproducts.php?cat=FUZZ
Total requests: 11

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000004:   200        102 L    358 W      4699 Ch     "3"
000000001:   200        102 L    358 W      4699 Ch     "0"
000000002:   200        107 L    526 W      7880 Ch     "1"
000000007:   200        102 L    358 W      4699 Ch     "6"
000000010:   200        102 L    358 W      4699 Ch     "9"
000000011:   200        102 L    358 W      4699 Ch     "10"
000000003:   200        104 L    394 W      5311 Ch     "2"
000000009:   200        102 L    358 W      4699 Ch     "8"
000000006:   200        102 L    358 W      4699 Ch     "5"
000000008:   200        102 L    358 W      4699 Ch     "7"
000000005:   200        102 L    358 W      4699 Ch     "4"

Total time: 0.968177
Processed Requests: 11
Filtered Requests: 0
Requests/sec.: 11.36154
```

A detailed table of all the filters for the payloads can be found here.

Sessions in wfuzz

A session in wfuzz is a temporary file which can be saved and later picked up, re-processed and post-processed. This is helpful in situations where one result saved already needs alterations or an analyst needs to look for something in the results. "--oF" filter can save the session output to a file.

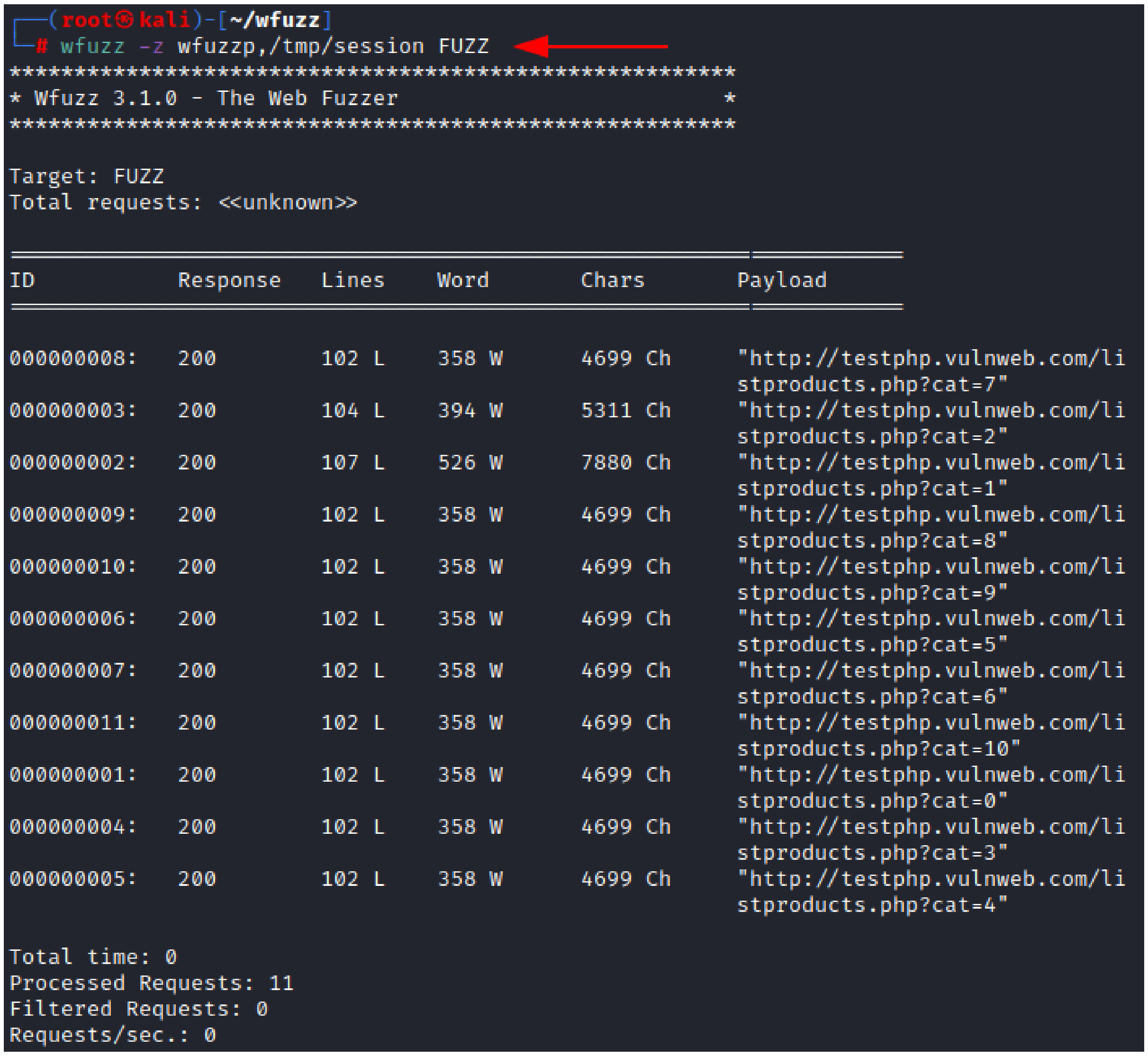
```
wfuzz --oF /tmp/session -z range,0-10 http://testphp.vulnweb.com/listproducts.php?cat=FUZZ
```



```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz --oF /tmp/session -z range,0-10 http://testphp.vulnweb.com/listproducts.php?cat=FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/listproducts.php?cat=FUZZ
Total requests: 11

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000006:   200        102 L    358 W      4699 Ch     "5"
000000007:   200        102 L    358 W      4699 Ch     "6"
000000011:   200        102 L    358 W      4699 Ch     "10"
000000001:   200        102 L    358 W      4699 Ch     "0"
000000002:   200        107 L    526 W      7880 Ch     "1"
000000004:   200        102 L    358 W      4699 Ch     "3"
000000003:   200        104 L    394 W      5311 Ch     "2"
000000009:   200        102 L    358 W      4699 Ch     "8"
000000008:   200        102 L    358 W      4699 Ch     "7"
000000010:   200        102 L    358 W      4699 Ch     "9"
000000005:   200        102 L    358 W      4699 Ch     "4"

Total time: 0
Processed Requests: 11
Filtered Requests: 0
Requests/sec.: 0
```

This session file can now be opened up again and consumed using the “wfuzzp” payload like so:

```
wfuzz -z wfuzzp,/tmp/session FUZZ
```

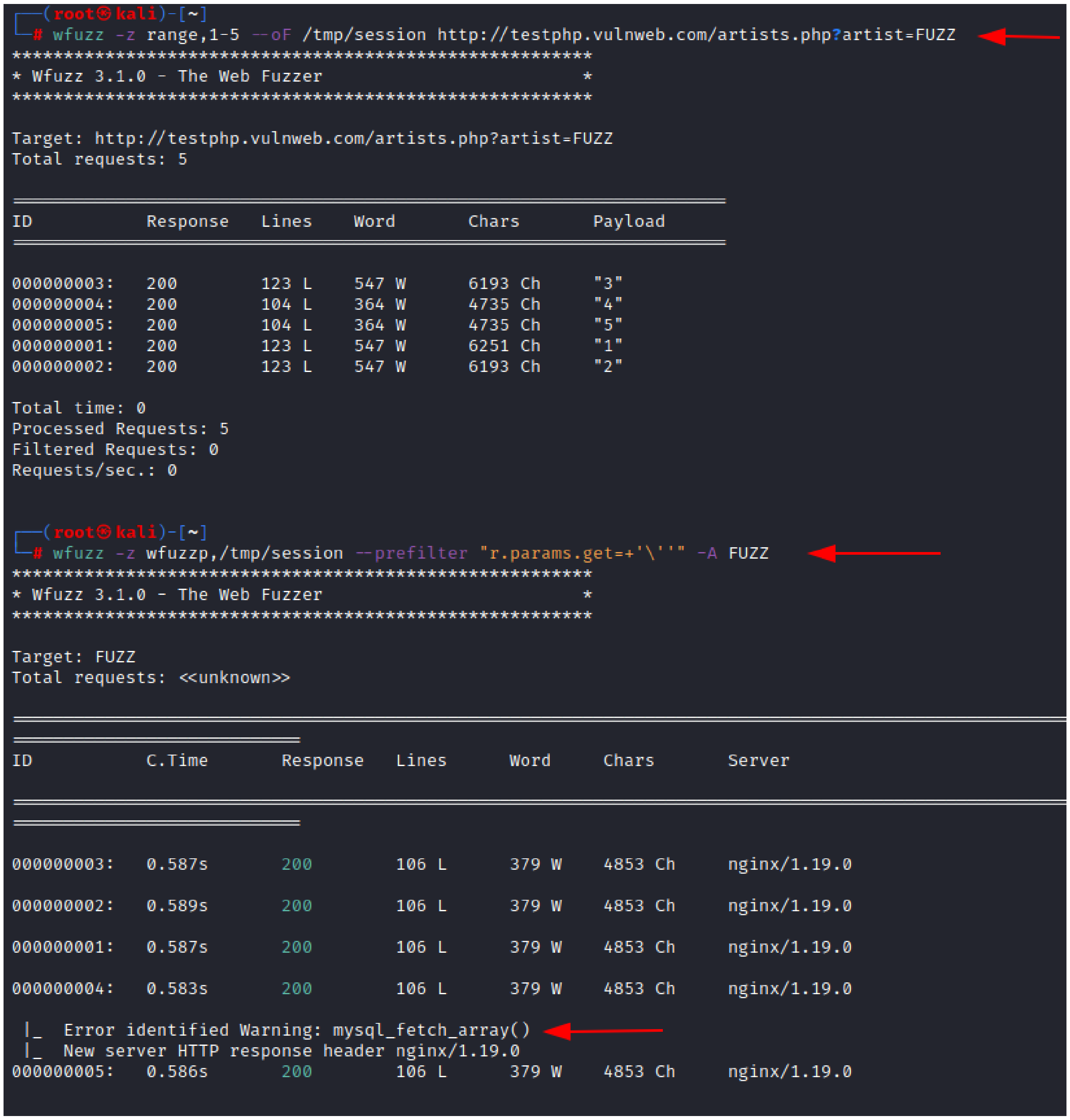


```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z wfuzzp,/tmp/session FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: FUZZ
Total requests: <<unknown>>

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000008:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=7"
000000003:   200        104 L    394 W      5311 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=2"
000000002:   200        107 L    526 W      7880 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=1"
000000009:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=8"
000000010:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=9"
000000006:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=5"
000000007:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=6"
000000011:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=10"
000000001:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=0"
000000004:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=3"
000000005:   200        102 L    358 W      4699 Ch     "http://testphp.vulnweb.com/li
                                                        stproducts.php?cat=4"

Total time: 0
Processed Requests: 11
Filtered Requests: 0
Requests/sec.: 0
```

One such example of this filteration from a previously saved session is as follows where we find an SQL injection vulnerability by utilizing a Pytho regex designed to read responses after a request modifies a parameter by adding apostrophe (‘) and fuzzing again. “-A” displays a verbose output.

The regex r.params.get=+’\’ adds apostrophe (‘) in the get parameter. r stands for raw response.

```
wfuzz -z range,1-5 --oF /tmp/session http://testphp.vulnweb.com/artists.php?artist=FUZZ
wfuzz -z wfuzzp,/tmp/session --prefilter "r.params.get=+'\'" -A FUZZ
```



```
┌──(root㉿kali)-[~]
└─# wfuzz -z range,1-5 --oF /tmp/session http://testphp.vulnweb.com/artists.php?artist=FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/artists.php?artist=FUZZ
Total requests: 5

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000003:   200        123 L    547 W      6193 Ch     "3"
000000004:   200        104 L    364 W      4735 Ch     "4"
000000005:   200        104 L    364 W      4735 Ch     "5"
000000001:   200        123 L    547 W      6251 Ch     "1"
000000002:   200        123 L    547 W      6193 Ch     "2"

Total time: 0
Processed Requests: 5
Filtered Requests: 0
Requests/sec.: 0


┌──(root㉿kali)-[~]
└─# wfuzz -z wfuzzp,/tmp/session --prefilter "r.params.get=+'\''" -A FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: FUZZ
Total requests: <<unknown>>

====================================================================================================
ID           C.Time       Response   Lines      Word     Chars       Server
====================================================================================================

000000003:   0.587s       200        106 L      379 W    4853 Ch     nginx/1.19.0

000000002:   0.589s       200        106 L      379 W    4853 Ch     nginx/1.19.0

000000001:   0.587s       200        106 L      379 W    4853 Ch     nginx/1.19.0

000000004:   0.583s       200        106 L      379 W    4853 Ch     nginx/1.19.0

 |_  Error identified Warning: mysql_fetch_array()
 |_  New server HTTP response header nginx/1.19.0
000000005:   0.586s       200        106 L      379 W    4853 Ch     nginx/1.19.0
```

As you can see, request number 4 throws an SQL error which indicates SQL injection. For more regex operations refer here.

Conclusion

Wfuzz is a versatile tool that can perform more than just directory enumeration and truly help a pentester in his analyses. It's a fast scanner which is easy to use and coded in python for portability. Hope you liked the article. Thanks for reading.

****