Web Application Penetration Testing by using WFUZZ

To install wfuzz, we use git:

```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/xmendez/wfuzz.git
Cloning into 'wfuzz' ...
remote: Enumerating objects: 9355, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 9355 (delta 17), reused 18 (delta 6), pack-reused 9315
Receiving objects: 100% (9355/9355), 7.04 MiB | 6.13 MiB/s, done.
Resolving deltas: 100% (6119/6119), done.

┌──(root㉿kali)-[~]
└─# cd wfuzz

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

┌──(root㉿kali)-[~/wfuzz]
└─#
```

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

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz --help
********************************************************
* 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 w
ith the values of the specified payload.
        FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be th
e first request performed and could be used as a base for filtering.


Options:
        -h/--help                 : This help
        --help                    : Advanced help
        --filter-help             : Filter language specification
        --version                 : Wfuzz version details
        -e <type>                 : List of available encoders/payloads/iterators/p
rinters/scripts

        --recipe <filename>       : Reads options from a recipe. Repeat for various
 recipes.
        --dump-recipe <filename>  : Prints current options as a recipe
        --oF <filename>           : Saves fuzz results to a file. These can be cons
umed later using the wfuzz payload.

        -c                        : Output with colors
        -v                        : Verbose information.
        -f filename,printer       : Store results in the output file using the spec
ified printer (raw printer if omitted).
        -o printer                : Show results using the specified printer.
        --interact                : (beta) If selected,all key presses are captured
```

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:

```
┌──(root㉿kali)-[~/wfuzz]
└─# ./wfpayload -z range,0-15
0
5
15
14
2
13
11
12
10
9
4
7
8
6
3
1
```

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

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

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.

```
┌──(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:

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

Available payloads:

  Name            | Summary
 ------------------------------------------------------------------------------
  ipnet           | Returns list of IP addresses of a network.
  file            | Returns each word from a file.
  names           | Returns possible usernames by mixing the given words, separated by -, using know
                  | n typical constructions.
  dirwalk         | Returns filename's recursively from a local directory.
  shodanp         | Returns URLs of a given Shodan API search (needs api key).
  buffer_overflow | Returns a string using the following pattern A * given number.
  list            | Returns each element of the given word list separated by -.
  burpstate       | Returns fuzz results from a Burp state.
  burplog         | Returns fuzz results from a Burp log.
  hexrand         | Returns random hex numbers from the given range.
  permutation     | Returns permutations of the given charset and length.
  bing            | Returns URL results of a given bing API search (needs api key).
  wfuzzp          | Returns fuzz results' URL from a previous stored wfuzz session.
  range           | Returns each number of the given range.
  burpitem        | This payload loads request/response from items saved from Burpsuite.
  stdin           | Returns each item read from stdin.
  hexrange        | Returns each hex number of the given hex range.
  guitab          | This payload reads requests from a tab in the GUI
  iprange         | Returns list of IP addresses of a given IP range.
  autorize        | Returns fuzz results' from autorize.
```

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

```
┌──(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.

```
┌──(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: 3

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

000000001:   XXX        0 L      0 W        0 Ch        "192.168.76.1! Pycurl error 52: Empty reply from server"
000000002:   XXX        0 L      0 W        0 Ch        "192.168.76.2! Pycurl error 52: Empty reply from server"
000000003:   XXX        0 L      0 W        0 Ch        "192.168.76.254! Pycurl error 52: Empty reply from server"

Total time: 11.29441
Processed Requests: 3
Filtered Requests: 0
Requests/sec.: 0.265618
```

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:

```
┌──(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
=====================================================================

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

000000047:   404        7 L      11 W       153 Ch      "adminsql"
000000001:   404        7 L      11 W       153 Ch      "@"
000000031:   404        7 L      11 W       153 Ch      "action"
000000015:   404        7 L      11 W       153 Ch      "2001"
000000049:   404        7 L      11 W       153 Ch      "adsl"
000000007:   404        7 L      11 W       153 Ch      "10"
000000050:   404        7 L      11 W       153 Ch      "agent"
000000048:   404        7 L      11 W       153 Ch      "admon"
000000046:   404        7 L      11 W       153 Ch      "admin_logon"
000000003:   404        7 L      11 W       153 Ch      "01"
000000045:   404        7 L      11 W       153 Ch      "adminlogon"
000000043:   404        7 L      11 W       153 Ch      "adminlogin"
000000044:   404        7 L      11 W       153 Ch      "admin_login"
000000042:   404        7 L      11 W       153 Ch      "administrator"
000000041:   404        7 L      11 W       153 Ch      "Administration"
000000038:   404        7 L      11 W       153 Ch      "Admin"
000000040:   404        7 L      11 W       153 Ch      "administration"
000000039:   404        7 L      11 W       153 Ch      "administrat"
000000037:   404        7 L      11 W       153 Ch      "admin_"
000000036:   404        7 L      11 W       153 Ch      "_admin"
000000035:   301        7 L      11 W       169 Ch      "admin"
000000034:   404        7 L      11 W       153 Ch      "adm"
000000033:   404        7 L      11 W       153 Ch      "active"
000000032:   404        7 L      11 W       153 Ch      "actions"
000000030:   404        7 L      11 W       153 Ch      "accounting"
000000028:   404        7 L      11 W       153 Ch      "accessgranted"
000000029:   404        7 L      11 W       153 Ch      "account"
000000027:   404        7 L      11 W       153 Ch      "access"
000000026:   404        7 L      11 W       153 Ch      "academic"
000000025:   404        7 L      11 W       153 Ch      "about"
```

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

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w 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: 0
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 0
```

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.

```
┌──(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: 30.28085
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 31.40598
```

```
┌──(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

```
┌──(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
```

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:

```
┌──(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: 30.22916
Processed Requests: 951
Filtered Requests: 947
Requests/sec.: 31.45968
```

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

```
┌──(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: 30.32421
Processed Requests: 951
Filtered Requests: 946
Requests/sec.: 31.36107
```

Double fuzzing

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

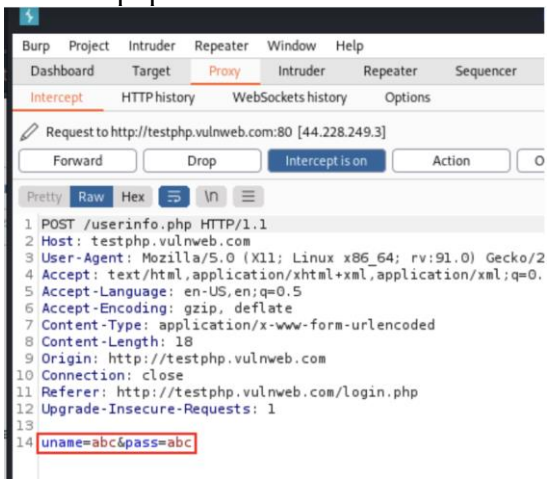
FUZ2Z - 2nd parameter
FUZ3Z - 3rd parameter
FUZ4Z - 4th parameter

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.

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -w wordlist/general/common.txt -w wordlist/general/common.txt --hc 404 http://testphp.vulnweb.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
=====================================================================

000000074:   404        7 L      11 W       153 Ch      "@ - asp"
000000073:   404        7 L      11 W       153 Ch      "@ - arrow"
000000072:   404        7 L      11 W       153 Ch      "@ - archives"
000000071:   404        7 L      11 W       153 Ch      "@ - archive"
000000070:   404        7 L      11 W       153 Ch      "@ - apps"
000000069:   404        7 L      11 W       153 Ch      "@ - apply"
```

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



The same can be implemented using wfuzz like so:

```
┌──(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        119 L    449 W      6014 Ch     "test - test"

Total time: 11.57909
Processed Requests: 52
Filtered Requests: 51
Requests/sec.: 4.490851
```

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:

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

********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://testphp.vulnweb.com/userinfo.php
Total requests: 1

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

000000001:   200        119 L    449 W      6014 Ch     "test - test"

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

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

```
┌──(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: 0
Processed Requests: 951
Filtered Requests: 946
Requests/sec.: 0
```

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

```
┌──(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
=====================================================================

000000046:   200        109 L    388 W      4958 Ch     "admin_logon"
000000007:   200        109 L    388 W      4958 Ch     "10"
000000015:   200        109 L    388 W      4958 Ch     "2001"
000000003:   200        109 L    388 W      4958 Ch     "01"
000000050:   200        109 L    388 W      4958 Ch     "agent"
000000001:   200        109 L    388 W      4958 Ch     "@"
000000047:   200        109 L    388 W      4958 Ch     "adminsql"
000000048:   200        109 L    388 W      4958 Ch     "admon"
000000031:   200        109 L    388 W      4958 Ch     "action"
000000049:   200        109 L    388 W      4958 Ch     "adsl"
000000045:   200        109 L    388 W      4958 Ch     "adminlogon"
000000043:   200        109 L    388 W      4958 Ch     "adminlogin"
000000042:   200        109 L    388 W      4958 Ch     "administrator"
000000041:   200        109 L    388 W      4958 Ch     "Administration"
000000044:   200        109 L    388 W      4958 Ch     "admin_login"
000000040:   200        109 L    388 W      4958 Ch     "administration"
000000039:   200        109 L    388 W      4958 Ch     "administrat"
000000038:   200        109 L    388 W      4958 Ch     "Admin"
```

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

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z file,wordlist/general/common.txt -H "X-Forwarded-By: 127.0.0.1" -H "User-Agent: Firefox" http://t
estphp.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
=====================================================================

000000050:   404        7 L      11 W       153 Ch      "agent"
000000031:   404        7 L      11 W       153 Ch      "action"
000000001:   404        7 L      11 W       153 Ch      "@"
000000046:   404        7 L      11 W       153 Ch      "admin_logon"
000000015:   404        7 L      11 W       153 Ch      "2001"
000000049:   404        7 L      11 W       153 Ch      "adsl"
000000048:   404        7 L      11 W       153 Ch      "admon"
000000047:   404        7 L      11 W       153 Ch      "adminsql"
000000003:   404        7 L      11 W       153 Ch      "01"
000000007:   404        7 L      11 W       153 Ch      "10"
000000045:   404        7 L      11 W       153 Ch      "adminlogon"
000000044:   404        7 L      11 W       153 Ch      "admin_login"
000000043:   404        7 L      11 W       153 Ch      "adminlogin"
000000042:   404        7 L      11 W       153 Ch      "administrator"
000000041:   404        7 L      11 W       153 Ch      "Administration"
000000040:   404        7 L      11 W       153 Ch      "administration"
000000038:   404        7 L      11 W       153 Ch      "Admin"
000000037:   404        7 L      11 W       153 Ch      "admin_"
000000039:   404        7 L      11 W       153 Ch      "administrat"
000000036:   404        7 L      11 W       153 Ch      "_admin"
000000035:   301        7 L      11 W       169 Ch      "admin"
000000034:   404        7 L      11 W       153 Ch      "adm"
000000033:   404        7 L      11 W       153 Ch      "active"
000000030:   404        7 L      11 W       153 Ch      "accounting"
000000032:   404        7 L      11 W       153 Ch      "actions"
000000029:   404        7 L      11 W       153 Ch      "account"
000000028:   404        7 L      11 W       153 Ch      "accessgranted"
```

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

```
┌──(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: 9

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

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

Total time: 10.44685
Processed Requests: 9
Filtered Requests: 6
Requests/sec.: 0.861503
```

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

```
┌──(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
=====================================================================

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

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 SOCKS proxy is active on port 8080 and the request intercepted in the burp intercept as you can see.

```
┌──(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
=====================================================================


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

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

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z list,nonvalid-httpwatch --basic FUZZ:FUZZ https://www.httpwatch.com/httpgallery/authentication/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
=====================================================================

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

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

```
┌──(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
=====================================================================

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

Total time: 10.43172
Processed Requests: 3
Filtered Requests: 0
Requests/sec.: 0.287584
```

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

```
┌──(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": "admin_logon", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http:
//testphp.vulnweb.com/admin_logon", "words": 11}, {"chars": 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, "code": 404, "payload": "adminsql", "lines": 7, "location":
"", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/adminsql", "words": 11}, {"chars": 153, "code": 404, "pay
load": "10", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/10", "words": 11}, {
"chars": 153, "code": 404, "payload": "action", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testp
hp.vulnweb.com/action", "words": 11}, {"chars": 153, "code": 404, "payload": "2001", "lines": 7, "location": "", "method": "GET", "post_data": [], "server":
 "nginx/1.19.0", "url": "http://testphp.vulnweb.com/2001", "words": 11}, {"chars": 153, "code": 404, "payload": "agent", "lines": 7, "location": "", "method
": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/agent", "words": 11}, {"chars": 153, "code": 404, "payload": "adsl",
 "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/adsl", "words": 11}, {"chars": 1
53, "code": 404, "payload": "admon", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.
com/admon", "words": 11}, {"chars": 153, "code": 404, "payload": "0", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0"
, "url": "http://testphp.vulnweb.com/0", "words": 11}, {"chars": 153, "code": 404, "payload": "adminlogon", "lines": 7, "location": "", "method": "GET", "po
st_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/adminlogon", "words": 11}, {"chars": 153, "code": 404, "payload": "admin_login",
"lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/admin_login", "words": 11}, {"cha
rs": 153, "code": 404, "payload": "adminlogin", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testp
hp.vulnweb.com/adminlogin", "words": 11}, {"chars": 153, "code": 404, "payload": "administrator", "lines": 7, "location": "", "method": "GET", "post_data":
[], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/administrator", "words": 11}, {"chars": 153, "code": 404, "payload": "Administration", "lin
es": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/Administration", "words": 11}, {"char
s": 153, "code": 404, "payload": "administration", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://te
stphp.vulnweb.com/administration", "words": 11}, {"chars": 153, "code": 404, "payload": "Admin", "lines": 7, "location": "", "method": "GET", "post_data": [
], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/Admin", "words": 11}, {"chars": 153, "code": 404, "payload": "admin_", "lines": 7, "location
": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/admin_", "words": 11}, {"chars": 153, "code": 404, "pa
yload": "_admin", "lines": 7, "location": "", "method": "GET", "post_data": [], "server": "nginx/1.19.0", "url": "http://testphp.vulnweb.com/_admin", "words
```

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.

```
┌──(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 %%dd%dd escape
  url_safe, url | double_urlencode   | Applies a double encode to special characters in string using the %25xx escape.
                |                    | Letters, digits, and the characters '_.-' are never quoted.
  url           | first_nibble_hex   | Replaces ALL characters in string using the %%dd? escape
  default       | hexlify            | Every byte of data is converted into the corresponding 2-digit hex representatio
                |                    | n.
  html          | html_decimal       | Replaces ALL characters in string using the &#dd; escape
  html          | html_escape        | Convert the characters &<>" in string to HTML-safe sequences.
  html          | html_hexadecimal   | Replaces ALL characters in string using the &#xx; escape
  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 capitals letters
  url           | second_nibble_hex  | Replaces ALL characters in string using the %?%dd escape
  hashes        | sha1               | Applies a sha1 hash to the given string
  hashes        | sha256             | Applies a sha256 hash to the given string
  hashes        | sha512             | Applies a sha512 hash to the given string
  url           | uri_double_hex     | Encodes ALL charachers using the %25xx escape.
  url           | uri_hex            | Encodes ALL charachers using the %xx escape.
  url           | uri_triple_hex     | Encodes ALL charachers using the %25%xx%xx escape.
  url           | uri_unicode        | Replaces ALL characters in string using the %u00xx escape
  url_safe, url | urlencode          | Replace special characters in string using the %xx escape. Letters, digits, and
                |                    | the characters '_.-' are never quoted.
  url           | utf8               | Replaces ALL characters in string using the \u00xx escape
  url           | utf8_binary        | Replaces ALL characters in string using the \uxx escape
```

```
┌──(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
=====================================================================

000000014:   404        7 L      11 W       153 Ch      "08f90c1a417155361a5c4b8d297e0d78"
000000009:   404        7 L      11 W       153 Ch      "a9b7ba70783b617e9998dc4dd82eb3c5"
000000003:   404        7 L      11 W       153 Ch      "96a3be3cf272e017046d1b2674a52bd3"
000000001:   404        7 L      11 W       153 Ch      "518ed29525738cebdac49c49e60ea9d3"
000000010:   404        7 L      11 W       153 Ch      "202cb962ac59075b964b07152d234b70"
000000012:   404        7 L      11 W       153 Ch      "98f13708210194c475687be6106a3b84"
000000013:   404        7 L      11 W       153 Ch      "3644a684f98ea8fe223c713b77189a77"
000000007:   404        7 L      11 W       153 Ch      "d3d9446802a44259755d38e6d163e820"
000000015:   404        7 L      11 W       153 Ch      "d0fb963ff976f9c37fc81fe03c21ea7b"
000000011:   404        7 L      11 W       153 Ch      "c81e728d9d4c2f636f067f89cc14862c"
000000005:   404        7 L      11 W       153 Ch      "e45ee7ce7e88149af8dd32b27f9512ce"
000000008:   404        7 L      11 W       153 Ch      "f899139df5e1059396431415e770c6dd"
000000006:   404        7 L      11 W       153 Ch      "c4ca4238a0b923820dcc509a6f75849b"
000000004:   404        7 L      11 W       153 Ch      "a2ef406e2c2351e0b9e80029c909242d"
000000002:   404        7 L      11 W       153 Ch      "b4b147bc522828731f1a016bfa72c073"
000000016:   404        7 L      11 W       153 Ch      "4ba29b9f9e5732ed33761840f4ba6c53"
000000022:   404        7 L      11 W       153 Ch      "4124bc0a9335c27f086f24ba207a4912"
000000018:   404        7 L      11 W       153 Ch      "b8b4b727d6f5d1b61fff7be687f7970f"
000000021:   404        7 L      11 W       153 Ch      "0cc175b9c0f1b6a831c399e269772661"
000000023:   404        7 L      11 W       153 Ch      "47bce5c74f589f4867dbd57e9ca9f808"
000000017:   404        7 L      11 W       153 Ch      "a591024321c5e2bdbd23ed35f0574dde"
000000020:   404        7 L      11 W       153 Ch      "eccbc87e4b5ce2fe28308fd9f2a7baf3"
000000019:   404        7 L      11 W       153 Ch      "d47268e9db2e9aa3827bba3afb7ff94a"
000000024:   404        7 L      11 W       153 Ch      "900150983cd24fb0d6963f7d28e17f72"
000000032:   404        7 L      11 W       153 Ch      "ebb67a4271abe715344471b0f16321f6"
000000030:   404        7 L      11 W       153 Ch      "d4c143f004d88b7286e6f999dea9d0d7"
000000029:   404        7 L      11 W       153 Ch      "e268443e43d93dab7ebef303bbe9642f"
000000026:   404        7 L      11 W       153 Ch      "8da6f5e5e803dafe72cabfdd8adb476f"
000000031:   404        7 L      11 W       153 Ch      "418c5509e2171d55b0aee5c2ea4442b5"
```

Storing and restoring fuzz from recipes

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

```
┌──(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
```

Filtering results

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

```
┌──(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
```

To view raw responses of the payload sent and the complete HTTP request made, you can use “--efield r” option

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z range --zD 0-1 -u http://testphp.vulnweb.com/artists.php?artist=FUZZ --efield url --efield 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/artists.php?artist=0 | 4735"
000000002:   200        123 L    547 W      6251 Ch     "1 | http://testphp.vulnweb.com/artists.php?artist=1 | 6251"

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

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:

```
┌──(root㉿kali)-[~/wfuzz]
└─# wfuzz -z range,0-10 --filter "c=200 and l>97" 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
=====================================================================

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

Total time: 10.65527
Processed Requests: 11
Filtered Requests: 0
Requests/sec.: 1.032352
```

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.

```
┌──(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
=====================================================================

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

```
┌──(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
=====================================================================

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

Total time: 10.64809
Processed Requests: 11
Filtered Requests: 0
Requests/sec.: 1.033048
```

One such example of this filteration from a previously saved session is as follows where we find an SQL injection vulnerability by utilizing a Python 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.

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

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

Total time: 10.44083
Processed Requests: 5
Filtered Requests: 0
Requests/sec.: 0.478888

┌──(root㉿kali)-[~/wfuzz]
└─# 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                    Redirect                  Payload
=====================================================================

000000003:   10.431s     200        106 L    379 W      4853 Ch    nginx/1.19.0                                        "http://testphp.vulnweb.com/artists.php?artist=3'"

 |_  Error identified Warning: mysql_fetch_array()
 |_  New server HTTP response header nginx/1.19.0
000000004:   10.435s     200        106 L    379 W      4853 Ch    nginx/1.19.0                                        "http://testphp.vulnweb.com/artists.php?artist=4'"

000000002:   10.436s     200        106 L    379 W      4853 Ch    nginx/1.19.0                                        "http://testphp.vulnweb.com/artists.php?artist=2'"

000000001:   10.438s     200        106 L    379 W      4853 Ch    nginx/1.19.0                                        "http://testphp.vulnweb.com/artists.php?artist=1'"

000000005:   10.436s     200        106 L    379 W      4853 Ch    nginx/1.19.0                                        "http://testphp.vulnweb.com/artists.php?artist=5'"
```

As you can see, request number 4 throws an SQL error which indicates SQL injection.

Conclusion

Wfuzz is a versatile tool that can perform more than just directory enumeration. It’s a fast scanner which is easy to use and coded in python for portability.