FFUF

ffuf- Fuzz Faster U Fool

To get the help menu

```
$ ffuf -h
Fuzz Faster U Fool - v1.3.0-dev

HTTP OPTIONS:
  -H                    Header `"Name: Value"`, separated by colon. Multiple -H
flags are accepted.
  -X                    HTTP method to use
  -b                    Cookie data `"NAME1=VALUE1; NAME2=VALUE2"` for copy as
curl functionality.
  -d                    POST data
  -ignore-body          Do not fetch the response content. (default: false)
  -r                    Follow redirects (default: false)
  -recursion            Scan recursively. Only FUZZ keyword is supported, and URL
(-u) has to end in it. (default: false)
  -recursion-depth      Maximum recursion depth. (default: 0)
  -recursion-strategy Recursion strategy: "default" for a redirect based, and
"greedy" to recurse on all matches (default: default)
  -replay-proxy         Replay matched requests using this proxy.
  -timeout              HTTP request timeout in seconds. (default: 10)
  -u                    Target URL
  -x                    Proxy URL (SOCKS5 or HTTP). For example: http://
127.0.0.1:8080 or socks5://127.0.0.1:8080

SNIP
```

Basic usage-

```
$ ffuf -u http://domainname.TLD/FUZZ -w "path to wordlist here"
```

-u - for specifying URL
FUZZ(keyword) -   query to fuzz for
-w   - path to wordlist to be used
-c - to colorize the output

For filtering and matching :-

```
$ ffuf -h
...
MATCHER OPTIONS:
  -mc                   Match HTTP status codes, or "all" for everything.
(default: 200,204,301,302,307,401,403,405)
  -ml                   Match amount of lines in response
  -mr                   Match regexp
  -ms                   Match HTTP response size
  -mw                   Match amount of words in response

FILTER OPTIONS:
  -fc                   Filter HTTP status codes from response. Comma separated
list of codes and ranges
```

```
  -fl                 Filter by amount of lines in response. Comma separated
list of line counts and ranges
  -fr                 Filter regexp
  -fs                 Filter HTTP response size. Comma separated list of sizes
and ranges
  -fw                 Filter by amount of words in response. Comma separated
list of word counts and ranges
...
```

For piping the values to “-w” flag i.e. read a wordlist from stdout.

Example- To use integers as wordlist for fuzzing the value of parameter “id”.

```
$ ruby -e '(0..255).each{|i| puts i}' | ffuf -u 'http://MACHINE_IP/sqli-labs/
Less-1/?id=FUZZ' -c -w - -fw 33
```

```
$ ruby -e 'puts (0..255).to_a' | ffuf -u 'http://MACHINE_IP/sqli-labs/Less-1/?
id=FUZZ' -c -w - -fw 33
```

```
$ for i in {0..255}; do echo $i; done | ffuf -u 'http://MACHINE_IP/sqli-labs/
Less-1/?id=FUZZ' -c -w - -fw 33
```

```
$ cook '[0-255]' | ffuf -u 'http://MACHINE_IP/sqli-labs/Less-1/?id=FUZZ' -c -w
- -fw 33
```

```
$ seq 0 255 | ffuf -u 'http://MACHINE_IP/sqli-labs/Less-1/?id=FUZZ' -c -w - -fw
33
```

The above methods can be used to pass wordlists as sdtin to -w flag.

We can proxify traffic, by sending traffic through a web proxy(HTTP or socks5)

Example-

```
$ ffuf -u http://MACHINE_IP/ -c -w /usr/share/seclists/Discovery/Web-Content/
common.txt -x http://127.0.0.1:8080
```