

# ATTACKING WEB APPLICATIONS WITH FFUF



## What is Fuzzing?

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

## What is FFuF?

**FFuF (Fuzz Faster u Fool)** is a fast web fuzzer written in Go that allows typical directory discovery, virtual host discovery (without DNS records) and GET and POST parameter fuzzing.

## Directory Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u  
http://target.com/FUZZ -s
```

## Extension Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ -u  
http://target.com/blog/indexFUZZ
```

**Note:** The wordlist we chose already contains a dot (.), so we will not have to add the dot after "index" in our fuzzing.

## Page Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u  
http://target.com/blog/FUZZ.php
```

## Recursive Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u  
http://target.com/FUZZ -recursion -recursion-depth 1 -e .php -v
```

## Subdomain Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u  
https://FUZZ.target.com/
```

## Vhost Fuzzing

```
ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u  
http://target.com/ -H 'Host: FUZZ.target.com'
```

**Note:** To add DNS record → `sudo sh -c 'echo "SERVER_IP URL" >> /etc/hosts'`

**Note:** We know that we will always get "200 OK". However, if the VHost does exist and we send a correct one in the header, we should get a "different response size", as in that case, we would be getting the page from that VHosts, which is likely to show a different page.

# ATTACKING WEB APPLICATIONS WITH FFUF

## Filtering Results

```
ffuf -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt:FUZZ -u http://target.com/ -H 'Host: FUZZ.target.com' -fs xxx
```

## Parameter Fuzzing – GET

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.target.com/admin/admin.php?FUZZ=test_value -fs xxx
```

## Parameter Fuzzing – POST

```
ffuf -w /opt/useful/SecLists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ -u http://admin.target.com/admin/admin.php -X POST -d 'FUZZ=test_value' -H 'Content-Type: application/x-www-form-urlencoded' -fs xxx
```

**Note:** In PHP, POST data "**content-type**" can only accept "**application/x-www-form-urlencoded**". So, we can set that in ffuf with "**-H 'Content-Type: application/x-www-form-urlencoded'**".

## Value Fuzzing

```
ffuf -w ids.txt:FUZZ -u http://admin.target.com/admin/admin.php -X POST -d 'id=FUZZ' -H 'Content-Type: application/x-www-form-urlencoded' -fs xxx
```

**Note:** The simplest way is to use the following command in Bash that writes all numbers from 1-1000 to a file → **for i in \$(seq 1 1000); do echo \$i >> ids.txt; done**

## Parameters

- w** → Wordlist file path
- u** → Target URL
- s** → Do not print additional information (silent mode)
- recursion** → Scan recursively
- recursion-depth** → Maximum recursion depth
- e** → Comma separated list of extensions
- v** → Verbose output
- fs** → Filter HTTP response size
- X** → HTTP method to use
- H** → Header
- d** → POST data

