Hacker tools: Gobuster – the all-in-one tool for you

BY ANNA HAMMOND · JULY 5, 2021 · LAST UPDATED ON MARCH 6, 2025

Summer is at our doorstep, the weather is getting better and the Intigriti team is ready to help you once again. This week, we will go over Gobuster, a well-known tool amongst researchers for mainly brute-forcing directories. But that's not all the tool can do. It has multiple options what makes it a perfect all-in-one tool.

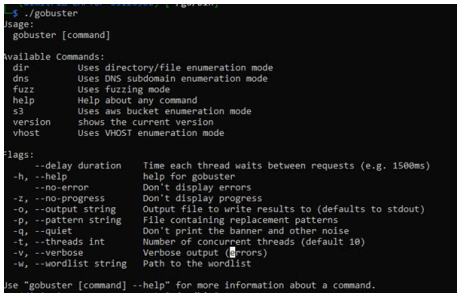
Like the name indicates, the tool is written in Go. Gobuster is a brute force scanner that can discover hidden directories, subdomains, and virtual hosts. It is an extremely fast tool so make sure you set the correct settings to align with the program you are hunting on.

Gobuster can be found on Github: https://github.com/OJ/gobuster and is still actively being maintained by OJ Reeves.

The installation

Let us start by installing Gobuster with Go. Make sure you have installed Golang **1.16** or above. I assume you have Golang installed, if this is not the case, check one of our previous articles.

```
go install github.com/OJ/gobuster/v3@latest
```



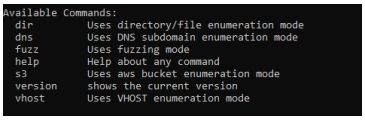
Installing wordlists:

GoBuster is a brute force tool, and brute forcers need wordlists. Let's download some common lists we can use. The most famous one is SecLists. It has wordlists organized for different purposes.

```
wget https://github.com/danielmiessler/SecLists
```

The Basics

GoBuster has a couple of modules and each module has its own flags. We will go over them and discuss the most interesting parts.



There are global flags you can use with each module, and then each module has its own specific flags. The standard ones are self-explanatory. The most important ones are the (**-w**) and (**–delay**) flags. If we want to see more for a specific module you can use the (**-h**) flag in combination with the module (*./gobuster dir -h*)

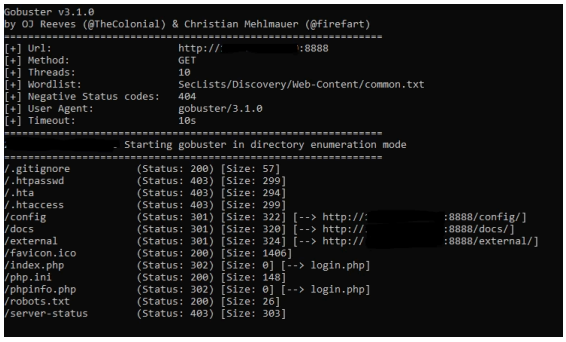
```
Global Flags:
--delay duration    Time each thread waits between requests (e.g. 1500ms)
-h, --help             help for gobuster
    --no-error         Don't display errors
-z, --no-progress      Don't display progress
-o, --output string    Output file to write results
-p, --pattern string   File containing replacement patterns
-q, --quiet            Don't print the banner and other noise
-t, --threads int      Number of concurrent threads (default 10)
-v, --verbose          Verbose output (errors)
-w, --wordlist string  Path to the wordlist
```

Dir Module:

The dir module is the original module from GoBuster, it brute forces directories to discover hidden folders and files.

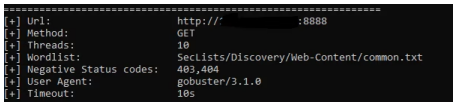
In order to run *gobuster dir* in the most basic way is by providing an URL (**-u**) and a wordlist (**-w**). Wordlists can also be piped into gobuster by providing a **–** on the **-w** flag.

```
./gobuster dir -u <URL> -w common.txt
cat common.txt | gobuster dir -u <URL> -w -
```



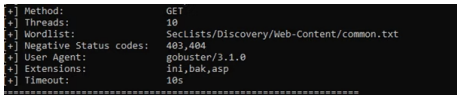
As you can see we have lots of 403 status codes. To exclude those codes, use the (**-b**) flag.

```
./gobuster dir -u <URL> -w common.txt -b 404,403
```



Gobuster dir can look for well-known backup files (**-d**) and add an extension to the wordlist to discover files (**-x**).

```
./gobuster dir -u <URL> -w common.txt -d
./gobuster dir -u <URL> -w common.txt -x .ini,.asp,.bak
```



These are the most interesting flags to be used for the dir module of Gobuster. To tweak a bit more you can make use of the global thread (**-t**) flag, and the delay (**–delay**) flag. Check all options by providing ***dir -h***.

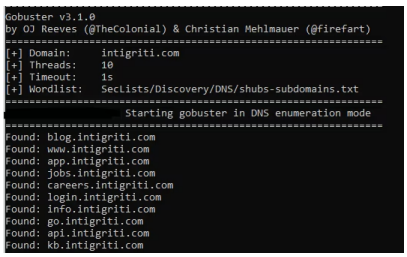
```
./gobuster dir -h
```

DNS Module:

With the DNS module, we can brute force for subdomains. Gobuster only does the discovery of subdomains by brute-forcing them. Unlike previous tools, we discussed that use external resources to discover subdomains.

This module has limited flags, for a basic run, you need a base domain (**-d**) and a wordlist (**-w**).

```
./gobuster dns -d <DOMAIN> -w shubs-subdomains.txt
```



Some programs require you to use specific DNS servers when running your scans. Gobuster DNS can use custom DNS servers by providing the (**-r**) flag.

```
./gobuster dns -r 8.8.4.4 -d <DOMAIN> -w shubs-subdomains.txt
```

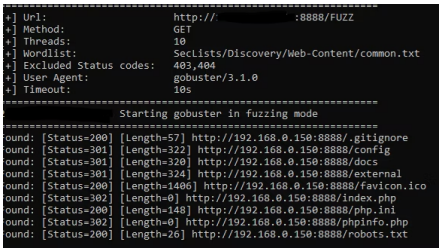
These are the most important options for the DNS module. Keep in mind that the global flags are also available (**–delay**) and (**-t** threads).

Fuzz Module:

Gobuster also has a fuzz module that can fuzz for parameters. For a dedicated fuzzing tool check out FFuF, we discussed this in one of our previous articles.

The fuzzing module has the same options as the dir module with the difference that we need to put the keyword FUZZ where we want to inject our wordlist.

```
./gobuster fuzz -u <URL>/FUZZ -w common.txt
```



That's all to it for this module. For all options run ***gobuster fuzz -h***

Vhost Module:

Another module from Gobuster is one to discover vhosts. Like all the other modules, this is done by brute-forcing, and we need to give at least two parameters. The URL (**-u**) and the wordlist (**-w**) parameter. Again, this is a limited module.

```
./gobuster vhost -u <URL> -w vhosts.txt
```

Conclusion

Gobuster is a useful tool for directory and file discovery. With version 3, there are some new modules implemented and give a nice extension. The most useful is the dir and dns modules as the others are still limited in options. Gobuster is again a super fast brute forcer that needs to be handled with care. Make sure you check the program details before using tools like this. We hope you enjoyed this article.