# Introduction to Gobuster

Welcome to the Gobuster portion of this room! This part of the room is aimed at complete beginners to enumeration and penetration testing. By completing this portion, you will have learned:

* How to install Gobuster on Kali Linux
* How to use the "dir" mode to enumerate directories and several of its most useful options
* How to use the "dns" mode to enumerate domains/subdomains and several of its most useful option
* Where to go for help

At the end of this section, you will have the opportunity to practice what you have learned by using Gobuster on another room, [*Blog*](https://tryhackme.com/room/blog)*.* This room utilizes what's called a Content Management System (CMS) in order to make things easier for the user. These typically have large and varied directory structures...perfect for directory enumeration with Gobuster!

With the introduction out of the way, let's get started!

# What is Gobuster?

As the name implies, Gobuster is written in [Go](https://golang.org/). Go is an open-source, low-level language (much like C or Rust) developed by a team at Google and other contributors. If you'd like to learn more about Go, visit the website linked above.

### Installing Gobuster on Kali Linux

Luckily, installing Gobuster on Kali Linux does not require any installation of Go and does not carry with it a complicated install process. This means no building from source or running any other complicated commands. Ready?

sudo apt install gobuster

Done.

# Useful Global Flags

There are some useful Global flags that can be used as well. I've included them in the table below. You can review these in the main documentation as well - [here](https://github.com/OJ/gobuster).

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -t | --threads | Number of concurrent threads (default 10) |
| -v | --verbose | Verbose output |
| -z | --no-progress | Don't display progress |
| -q | --quiet | Don't print the banner and other noise |
| -o | --output | Output file to write results to |

I will typically change the number of threads to 64 to increase the speed of my scans. If you don't change the number of threads, Gobuster can be a little slow.

﻿**"dir" Mode**

Dirbuster has a "dir" mode that allows the user to enumerate website directories. This is useful when you are performing a penetration test and would like to see what the directory structure of a website is. Often, directory structures of websites and web-apps follow a certain convention, making them susceptible to brute-forcing using wordlists. At the end of this room, you'll run Gobuster on [*Blog*](https://tryhackme.com/room/blog) which uses WordPress, a very common*Content Management System* (CMS). WordPress uses a very specific directory structure for its websites.

Gobuster is powerful because it not only allows you to scan the website, but it will return the status codes as well. This will immediately let you know if you as an outside user can request that directory or not. Additional functionality of Gobuster is that it lets you search for files as well with the addition of a simple flag!

**Using "dir" Mode**

To use "dir" mode, you start by typing gobuster dir. This isn't the full command, but just the start. This tells Gobuster that you want to perform a directory search, instead of one of its other methods (which we'll get to). It has to be written like this or else Gobuster will complain. After that, you will need to add the URL and wordlist using the -u and -w options, respectively. Like so:

gobuster dir -u http://10.10.10.10 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

**Note**: The URL is going to be the base path where Gobuster starts looking from. So the URL above is using the root web directory. For example, in a typical Apache installation on Linux, this is*/var/www/html*. So if you have a "**products**" directory and you want to enumerate that directory, you'd set the URL as *http://10.10.10.10/****products***. You can also think of this like *http://example.com/path/to/folder*. Also notice that I specified the protocol of HTTP. This is important and required.

This is a very common, simple, and straightforward command for Gobuster. This is typically what I will run when doing capture the flag style rooms on TryHackMe. However, there are some other helpful flags that can be useful in certain scenarios

**Other Useful Flags**

These flags are useful in certain scenarios.  Note that these are not all of the flag options, but some of the more common ones that you'll use in penetration tests and in capture the flag events. If you'd like the full list, you can see that [here](https://github.com/OJ/gobuster#dir-mode-options).

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -c | --cookies | Cookies to use for requests |
| -x | --extensions | File extension(s) to search for |
| -H | --headers | Specify HTTP headers, -H 'Header1: val1' -H 'Header2: val2' |
| -k | --no-tls-validation | Skip TLS certificate verification |
| -n | --no-status | Don't print status codes |
| -P | --password | Password for Basic Auth |
| -s | --status-codes | Positive status codes |
| -b | --status-codes-blacklist | Negative status codes |
| -U | --username | Username for Basic Auth |

A very common use of Gobuster's "dir" mode is the ability to use it's -x or--extensions flag to search for the contents of directories that you have already enumerated by providing a list of file extensions. File extensions are generally representative of the data they may contain. For example, **.conf** or **.config** files usually contain configurations for the application - including sensitive info such as database credentials.

A few other files that you may wish to search for are .txt files or other web application pages such as **.html** or **.php**. Let's assemble a command that would allow us to search the "myfolder" directory on a webserver for the following three files:

1. html

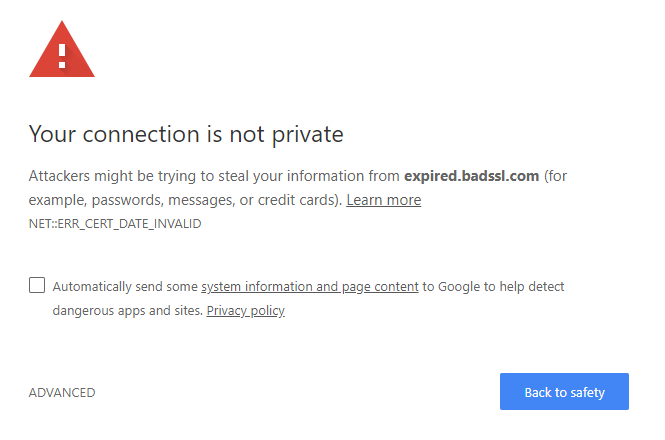
2. js

3. css

gobuster dir -u http://10.10.252.123/myfolder -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x.html,.css,.js

**The -k Flag**

The -k flag is special because it has an important use during penetration tests and captures the flag events. In a capture the flag room on TryHackMe for example, if HTTPS is enabled, you will most likely encounter an invalid cert error like the one below

  
In instances like this, if you try to run Gobuster against this without the-k flag, it won't return anything and will most likely error out with something gross and will leave you sad. Don't worry though, easy fix! Just add the -k flag to your scan and it will bypass this invalid certification and continue scanning and deliver the goods!

**Note:** This flag can be used with "dir" mode and "vhost" modes

**"dns" Mode**

The next mode we'll focus on is the "dns" mode. This allows Gobuster to brute-force subdomains. During a penetration test (or capture the flag), it's important to check sub-domains of your target's top domain. Just because something is patched in the regular domain, does not mean it is patched in the sub-domain. There may be a vulnerability for you to exploit in one of these sub-domains. For example, if State Farm owns statefarm.com and mobile.statefarm.com, there may be a hole in mobile.statefarm.com that is not present in statefarm.com. This is why it is important to search for subdomains too!

**Using "dns" Mode**

To use "dns" mode, you start by typing gobuster dns. Just like "dir" mode, this isn't the full command, but just the start. This tells Gobuster that you want to perform a sub-domain brute-force, instead of one of one of the other methods as previously mentioned. It has to be written like this or else Gobuster will complain. After that, you will need to add the domain and wordlist using the -d and -w options, respectively. Like so:

gobuster dns -d mydomain.thm -w /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt

This tells Gobuster to do a sub-domain scan on the domain "mydomain.thm". If there are any sub-domains available, Gobuster will find them and report them to you in the terminal.

**Other Useful Flags**

-d and -w are the main flags that you'll need for *most* of your scans. But there are a few others that are worth mentioning that we can go over. They are in the table below.

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -c | --show-cname | Show CNAME Records (cannot be used with '-i' option) |
| -i | --show-ips | Show IP Addresses |
| -r | --resolver | Use custom DNS server (format server.com or server.com:port) |

There aren't many additional flags to be used with this mode, but these are the main useful ones that you may use from time to time. If you'd like to see the full list of flags that can be used with this mode, check out the [documentation](https://github.com/OJ/gobuster#dns-mode-help)

**"vhost" Mode**

The last and final mode we'll focus on is the "vhost" mode. This allows Gobuster to brute-force virtual hosts. Virtual hosts are different websites on the same machine. In some instances, they can appear to look like sub-domains, but don't be deceived! Virtual Hosts are IP based and are running on the same server. This is not usually apparent to the end-user. On an engagement, it may be worthwhile to just run Gobuster in this mode to see if it comes up with anything. You never know, it might just find something! While participating in rooms on TryHackMe, virtual hosts would be a good way to hide a completely different website if nothing turned up on your main port 80/443 scan.

**Using "vhost" Mode**

To use "vhost" mode, you start by typing gobuster vhost. Just like the other modes, this isn't the full command, but just the start. This tells Gobuster that you want to perform a virtual host brute-force, instead of one of the other methods as previously mentioned. It has to be written like this or else Gobuster will complain. After that, you will need to add the domain and wordlist using the -u and -w options, respectively. Like so:

gobuster vhost -u http://example.com -w /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt

This will tell Gobuster to do a virtual host scan [http://example.com](http://example.com/) using the selected wordlist.

**Other Useful Flags**

A lot of the same flags that are useful for "dir" mode actually still apply to virtual host mode. Please check out the "dir" mode section for these and take a look at the [official documentation](https://github.com/OJ/gobuster#vhost-mode-options) for the full list. There's really too many that are similar to put them back here.

**Useful Wordlists**

﻿There are many useful wordlists to use for each mode. These may or may not come in handy later on during the VM portion of the room! I'll go over some of the ones that are on Kali by default as well as a short section on SecLists.

**Kali Linux Default Lists**

Below you will find a useful list of wordlists that are installed on Kali Linux by default. This is as of the latest version at the time of writing which is **2020.3**. Anything with a wildcard (\*) character indicates there's more than one list that matches. Keep in mind, a lot of these can be interchanged between modes. For example, "dir" mode wordlists (such as ones from the dirbuster directory) will contain words like "admin", "index", "about", "events", etc. A lot of these could be subdomains as well. Give them a try with the different modes!

* /usr/share/wordlists/dirbuster/directory-list-2.3-\*.txt
* /usr/share/wordlists/dirbuster/directory-list-1.0.txt
* /usr/share/wordlists/dirb/big.txt
* /usr/share/wordlists/dirb/common.txt
* /usr/share/wordlists/dirb/small.txt
* /usr/share/wordlists/dirb/extensions\_common.txt - Useful for when fuzzing for files!

**Non-Standard Lists**

In addition to the above, Daniel Miessler has created an amazing GitHub repo called [SecLists](https://github.com/danielmiessler/SecLists). It compiles many different lists used for many different things. The best part is, it's in apt! You can sudo apt install seclists and get the entire repo! We won't dive into any other lists as there are many. However, between what's installed by default on Kali and the SecLists repo, I doubt you'll need anything else.

**Useful Wordlists**

﻿There are many useful wordlists to use for each mode. These may or may not come in handy later on during the VM portion of the room! I'll go over some of the ones that are on Kali by default as well as a short section on SecLists.

**Kali Linux Default Lists**

Below you will find a useful list of wordlists that are installed on Kali Linux by default. This is as of the latest version at the time of writing which is **2020.3**. Anything with a wildcard (\*) character indicates there's more than one list that matches. Keep in mind, a lot of these can be interchanged between modes. For example, "dir" mode wordlists (such as ones from the dirbuster directory) will contain words like "admin", "index", "about", "events", etc. A lot of these could be subdomains as well. Give them a try with the different modes!

* /usr/share/wordlists/dirbuster/directory-list-2.3-\*.txt
* /usr/share/wordlists/dirbuster/directory-list-1.0.txt
* /usr/share/wordlists/dirb/big.txt
* /usr/share/wordlists/dirb/common.txt
* /usr/share/wordlists/dirb/small.txt
* /usr/share/wordlists/dirb/extensions\_common.txt - Useful for when fuzzing for files!

**Non-Standard Lists**

In addition to the above, Daniel Miessler has created an amazing GitHub repo called [SecLists](https://github.com/danielmiessler/SecLists). It compiles many different lists used for many different things. The best part is, it's in apt! You can sudo apt install seclists and get the entire repo! We won't dive into any other lists as there are many. However, between what's installed by default on Kali and the SecLists repo, I doubt you'll need anything else.