`gobuster`

`gobuster` is a tool used for **brute-forcing URIs** (directories and files) on websites and DNS subdomains (a subdomain is a subdivision of a domain, like `subdomain.example.com` and is typically used to achieve more granular control over different parts of a website or different services).

**Brute-forcing URIs** refers to the process of systematically and exhaustively trying a large number of directory or file names on a web server to discover hidden or unlisted resources. This technique involves using a wordlist that contains many potential directory or file names and testing each one by appending it to the base URL to see if it results in a valid resource.

For example, if your wordlist contains the words:

"admin"

"resources"

"data"

and you searched the domain `example.com`, the following URIs would be returned if they existed:

`example.com/admin`

`example.com/resources`

`example.com/data`

Recursive searching of subdirectories is not done by default; however this can be enabled (more on this later).

However, you can search for recursive directories (and files) explicitly, for example by including an entry like the following in the wordlist:

"wp-admin/about.php"

A flag can also be set to search for subdomains rather than URI paths, but not both simultaneously.

Wordlist

A wordlist should be a plain text file, where each line contains a single entry, such as a directory name, file name, or subdomain name that you want to test for existence on the target server.

## `gobuster` Command Examples

**Directory Brute-force:**

gobuster dir -u http://example.com -w /path/to/wordlist.txt -t 50

**`dir`** specifies directory mode.

**`-u`** specifies the target URI.

**`-w`** specifies the wordlist to use.

**`-t`** sets the number of concurrent threads to 50.

**DNS Subdomain Enumeration:**

gobuster dns -d example.com -w /path/to/wordlist.txt -t 50

**`dns`** specified DNS mode.

**`-d example.com`** specifies the target domain.

**`-w`** specifies the wordlist to use.

**`-t`** sets the number of concurrent threads to 50.

**Explanation of Concurrency Threads**

* **Concurrency**

`gobuster` uses multiple threads to perform parallel processing, allowing it to send multiple HTTP requests at the same time. This makes the brute-forcing process faster compared to sending requests one at a time.

* **Client-Side Setting**

The number of threads is a client-side setting, meaning it affects how `gobuster` runs on your local machine, not the server. It determines how many worker threads `gobuster` will use to handle requests concurrently.

***Concurrency and Parallelism***

Concurrency is different from parallelism. **Parallelism** involves multiple CPU cores working simultaneously to execute multiple tasks at once. **Concurrency** refers to the ability of a system to handle multiple tasks in a way that they can make progress without waiting for each other to finish completely.

In terms of `gobuster`, not using concurrency would mean sending an HTTP request, and not sending the next one until a response had been received to the first one. With concurrency, `gobuster` would keep sending more (up to the maximum specified number of threads) requests while it waited for an HTTP response, essentially multitasking (switching between tasks) very quickly.

`gobuster` does use parallelism, but this is due to default CPU scheduling rather than an explicit command from `gobuster`.

* **Optimal Number of Threads**

The optimal number of threads to use depends on factors such the number of CPU cores you have available, your RAM capacity, and network bandwidth. This may require some trial and error, while using system and network monitoring tools, as well as monitoring the response times and error rates from the target server.

There is also a risk of overloading the web server. Some servers may have rate-limiting or security measures that could block your IP if too many requests are sent in a short period.

## Virtual Hosts (VHosts)

A VHost is a configuration feature used in web servers to host multiple websites or services on a single physical server. Each virtual host can have its own domain name, content, and configuration, even though they share the same IP address.

`gobuster` can also search for virtual hosts on a target web server.

## Amazon S3 Buckets

`gobuster` can also be used for enumerating Amazon S3 buckets.