Gobuster

Used for reconnaissance,

- **Enumerate web directories**
- **Subdomains**
- **Virtual hosts**

It is an **open source offensive tool** written in Golang. Uses brute force tactics using wordlists.

Enumeration:
It is the act of listing all the available resources, whether they are accessible or not. Eg. Gobuster.

Brute force:
Act of trying every possible combination until a match is found. Using a wordlist.

Gobuster:
Included by default for distributions such as kali linux.

FLAGS:

Short Flag	Long Flag	Desc
-t	**--threads**	**No. of threads per scan.**
-w	**--wordlist**	**Wordlist used.**
	--delay	**Amount of time to wait.**
	--debug	**Helps troubleshoot.**
-o	**--output**	**Writes enumeration.**
-u	**--url**	**Url specified.**
-C	**--cookies**	**Cookies to pass each request.**
-X	**--extentions**	**Specifies which file extension.**
-H	**--headers**	**Entire header to pass which request.**
-k	**--no-tls-validation**	**Skips process that checks HTTPS certificate.**
-n	**--no-status**	**Don’t want to see status codes for each response.**
-P	**password**	**Handy for when user is identified.**
-s	**--status-codes**	**Configure which status codes wanted to display.**
-b	**--status-codes-blacklist**	**Configure which status codes not wanted to display.**
-U	**--username**	**Handy when you have obtained credentials.**
-r	**--followredirect**	**Follow redirect of received response.**

dir: Directory
DNS: Subdomain enumeration
Vhost: Virtual host.

How do use dir mode:

```
gobuster dir -u “https://www.example.thm” -w /path/to./wordlist
```

How to enumerate dir mode:

```
gobuster dir -u “https://www.example.thm” -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -r
```

How do we skip TLS verification?
--no-tls-validation

I ran this code:

```
gobuster dir -u “http://10.10.221.234” -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -r
```

I found a file called: /secret

I then continued to enumerate this file in particular:

-X -> To specify the file in which I wish to enumerate.

```
gobuster dir -u "http://10.10.221.234" -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x .js
```

We are only specifying the .js [json] extension.

```
gobuster dir -u "http://10.10.221.234/secret/content" -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x .js
```

```
gobuster dir -u "http://10.10.221.234/secret/content/flag.js" -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x .js
```

I then typed into the browser the following directory we managed to enumerate:
http://10.10.221.234/secret/content/flag.js
Flag: THM{ReconWasASuccess}

Use case: Subdomain Enumeration

For subdomain enumeration we need to introduce a couple new keywords:
-d flag = Enumerates dns subdomains.

```
gobuster dns -d 10.10.221.234 -w
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
```

The above command found that 4 threads were open.

Use case: Virtual host enumeration

```
gobuster vhost -u "http://10.10.221.234" -w
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
```

But, we need to know how many have a status code of 200. So we can filter that in!

```
gobuster vhost -u "http://10.10.221.234" -w
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt --append-domain
--exclude-length 250-320
```

Found **4 with status 200 codes.**

Conclusion:

I really enjoyed this room because it uses wordlists and enumeration techniques using gobuster to gain access to unauthorized or hidden areas of a domain that would otherwise be hidden to the ordinary user. But using gobuster we can reveal this information, especially useful in the reconnaissance stage where using enumeration, we can save this knowledge for later on potentially setting up a back door, or a way into the system during the practical offensive stages.