

Forum:[Hack Forums](#)

Thread: [What is and how to use Gobuster: A Comprehensive Guide for Ethical Hacking](#)

JoseMar - January 14, 2025, 1:22 am

What and how to use Gobuster

A Comprehensive Guide for Ethical Hacking

Introduction Gobuster is a powerful, fast, and efficient tool commonly used in ethical hacking and penetration testing to brute force directories, files, DNS subdomains, and virtual hosts. It's written in Go programming language, which makes it exceptionally fast compared to other tools in the same category. In this document, we'll dive into what Gobuster is, how it works, and provide hands-on examples to help you understand its application in your virtual laboratory.

What is Gobuster?

Gobuster is an open-source command-line tool that is primarily used to:
Discover hidden directories and files on web servers.

Enumerate DNS subdomains.

Identify virtual hosts on a server.

Unlike other tools that use recursive methods or rely on parsing, Gobuster performs brute-force attacks based on a provided wordlist, which makes it ideal for speed and efficiency.

Installing Gobuster

Before you start using Gobuster, ensure you have it installed on your system. Here's how to set it up:

For Debian-based systems (Ubuntu/Kali Linux):

```
Code sudo apt update  
sudo apt install gobuster
```

Using Go (if not available in your package manager):

```
Code sudo apt install golang  
go install github.com/OJ/gobuster/v3@latest  
export PATH=$PATH:$(go env GOPATH)/bin
```

Get Help Menu:

```
Code gobuster -h
```

Code Usage:

gobuster [command]

Available commands:

dir Uses directory/file enumeration mode
dns Uses DNS subdomain enumeration mode
fuzz Uses fuzzing mode
help Help about any command
s3 Uses aws bucket enumeration mode
version shows the current version
vhost Uses VHOST enumeration mode

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 to (defaults to stdout)
-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

Gobuster DIR command

The DIR mode is used for finding hidden directories and files.

To find additional flags available to use

Code gobuster dir --help

Code Uses directory/file enumeration mode

Usage:

gobuster dir [flags]

Flags:

-f, --add-slash Append / to each request
-c, --cookies string Cookies to use for the requests
-d, --discover-backup Upon finding a file search for backup files
--exclude-length ints exclude the following content length (completely ignores the status). Supply multiple times to exclude multiple sizes.
-e, --expanded Expanded mode, print full URLs
-x, --extensions string File extension(s) to search for
-r, --follow-redirect Follow redirects
-H, --headers stringArray Specify HTTP headers, -H 'Header1: val1' -H 'Header2: val2'
-h, --help help for dir
--hide-length Hide the length of the body in the output
-m, --method string Use the following HTTP method (default "GET")
-n, --no-status Don't print status codes
-k, --no-tls-validation Skip TLS certificate verification
-P, --password string Password for Basic Auth
--proxy string Proxy to use for requests [http(s)://host:port]
--random-agent Use a random User-Agent string
-s, --status-codes string Positive status codes (will be overwritten with

status-codes-blacklist if set)

-b, --status-codes-blacklist string Negative status codes (will override status-codes if set) (default "404")

--timeout duration HTTP Timeout (default 10s)

-u, --url string The target URL

-a, --useragent string Set the User-Agent string (default "gobuster/3.1.0")

-U, --username string Username for Basic Auth

--wildcard Force continued operation when wildcard found

Global Flags:

--delay duration Time each thread waits between requests (e.g. 1500ms)

--no-error Don't display errors

-z, --no-progress Don't display progress

-o, --output string Output file to write results to (defaults to stdout)

-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

How Does Gobuster Work?

In this short tutorial I present a couple of the many active techniques within the fields of cybersecurity and ethical hacking for discovering sensitive or hidden information within a target system or website. Most security professionals make great use of such techniques in finding vulnerabilities or weaknesses that may, when unattended, be exploited by malicious actors.

Gobuster works by brute-forcing web paths, subdomains, or other resources using a wordlist. It sends requests to a target server and checks the responses to identify valid results. It's particularly effective against servers with improperly configured or unprotected directories and files.

In this tutorial, let's see how to execute a directory and file brute-forcing and enumeration using Gobuster on Linux.

Basic Syntax

The general syntax of Gobuster is:

Code `gobuster <mode> -u <URL/Domain> -w <wordlist> [options]`

<mode>: The operation mode (e.g., dir, dns, vhost).

-u: Specifies the target URL or domain.

-w: Specifies the wordlist to use for brute-forcing.

Modes of Operation

1: Directory/File Brute force (dir)

Used to find hidden directories or files on a web server.

Example:

Code `gobuster dir -u http://example.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt`: Specify file extensions to check (e.g., .php, .html).

Code `gobuster dir -u http://example.com -w wordlist.txt -x php,html,txt -t`: Set the number of concurrent threads (default is 10).

2: DNS Subdomain Bruteforce (dns)

Enumerates DNS subdomains for a given domain.

Example:

Code `gobuster dns -d example.com -w /usr/share/wordlists/dns/subdomains-top1million-5000.txt`

3: VHost Bruteforce (vhost)

Identifies virtual hosts on a target.

Example:

Code `gobuster vhost -u http://example.com -w /usr/share/wordlists/vhosts.txt`

4: S3 Bucket Bruteforce (s3):

Finds AWS S3 buckets.

Code `gobuster s3 -b <bucket-name> -w <wordlist>`

Practical Examples

Example 1: Directory Enumeration

Code `gobuster dir -u http://panel.wallet.dl/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt`

Example 2: DNS Subdomain Enumeration

Target: example.com

Command:

Code `gobuster dns -d example.com -w /usr/share/wordlists/dns/subdomains-top1million-5000.txt`
This exposes subdomains that could be further investigated.

Example 3: Adding File Extensions

Target: <http://IP.com>

Command:

Code `gobuster dir -u http://IP.com -w /usr/share/wordlists/dirb/common.txt -x php,html`
This ensures Gobuster checks for files with .php and .html extensions:

Tips for Using Gobuster

Use optimized wordlists for the task at hand. Popular sources include:

Code `/usr/share/wordlists/dirbuster/`

SecLists: Just search on Google and go to GitHub for Download

Adjust thread count (-t) based on your system's capability and the server's performance.

Consider using proxies or VPNs to avoid being blocked during scanning.

Always get permission before scanning a target that isn't yours!

Testing Gobuster in Your Virtual Lab

Step 1: Set Up a Vulnerable Web Application:

Use tools like DVWA (Damn Vulnerable Web Application) or OWASP Juice Shop in your virtual lab environment.

Ensure your lab has a web server running with directories and subdomains to discover.

Or explore different websites that offer Capture the Flag (CTF) challenges, such as Hack The Box, TryHackMe

Step 2: Run Gobuster Commands

In this example I use this website:

<http://panel.wallet.dl>

Code `gobuster dir -u http://panel.wallet.dl/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt`

Test various Gobuster modes to explore the application and understand its capabilities.

Step 3: Analyze Results

Review which directories, files, or subdomains were discovered.

Use other tools or manual testing to further investigate any sensitive finding

Ethical Considerations

Gobuster is a powerful tool that must be used responsibly. Always be sure you follow this principles:

Obtain explicit permission before scanning any target.

Use it solely for ethical hacking, research, or educational purposes.

Ensure your actions comply with legal and organizational guidelines.

Conclusion

Gobuster is an essential tool for ethical hackers and penetration testers, offering a fast and efficient way to discover hidden resources. With its straightforward syntax and versatility, it's a valuable addition to any security professional's toolkit. By practicing with Gobuster in your virtual lab or websites what offers CTF and exploring its various modes, you'll gain hands-on experience and deepen your understanding of web application security.

Happy hacking ethically, of course!