

GoSpider & Hakrawler

- Web Crawler Step-by-Step Execution Manual



Prepared By: Moez Javed

Operating System: Kali Linux

Date: July 24, 2025



Go Spider

1. Install Go Language

Install Go, the language required to build GoSpider.

```
sudo apt update
```

```
sudo apt install golang -y
```

```
(kali@kali)~$ sudo apt update
(sudo) password for kali:
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Err:1 http://http.kali.org/kali kali-rolling InRelease
Temporary failure resolving 'http.kali.org'
1712 packages can be upgraded. Run 'apt list --upgradable' to see them.
Warning: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease Temporary failure resolving 'http.kali.org'
Warning: Some index files failed to download. They have been ignored, or old ones used instead.
The following packages were automatically installed and are no longer required:
icu-devtools libicu-dev libpython3.12-dev python3.12 python3.12-dev python3.12-minimal python3.12-venv ruby3.1 ruby3.1-dev ruby3.1-doc
Use 'sudo apt autoremove' to remove them.

Installing:
golang

Installing dependencies:
golang-1.24 golang-1.24-doc golang-1.24-go golang-1.24-src golang-doc golang-go golang-src

Suggested packages:
bzip2 | bzip2 mercurial

Summary:
Upgrading: 0, Installing: 8, Removing: 0, Not Upgrading: 1712
Download size: 50.0 MB
Space needed: 259 MB / 57.5 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 golang-1.24-doc all 1.24.2-2 [111 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 golang-1.24-src all 1.24.2-2 [21.2 MB]
```

2. Install GoSpider

Install GoSpider via Go.

Made By Moez Javed

go install github.com/jaeles-project/gospider@latest

```
(kali@kali)-[~]
└─$ go version
go version go1.24.2 linux/amd64

(kali@kali)-[~]
└─$ go install github.com/jaeles-project/gospider@latest

go: downloading github.com/jaeles-project/gospider v1.1.6
go: downloading github.com/json-iterator/go v1.1.11
go: downloading github.com/sirupsen/logrus v1.8.1
go: downloading github.com/spf13/cobra v1.1.3
go: downloading github.com/gocolly/colly/v2 v2.1.0
go: downloading github.com/gocolly/colly v1.2.0
go: downloading github.com/oxffaa/gopher-parse-sitemap v0.0.0-20191021113419-005d2eb1def4
go: downloading github.com/mitchellh/go-homedir v1.1.0
go: downloading github.com/x-cray/logrus-prefixed-formatter v0.5.2
go: downloading golang.org/x/net v0.0.0-20210614182718-04defd469f4e
go: downloading golang.org/x/sys v0.0.0-20210423082822-04245dca01da
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/modern-go/concurrent v0.0.0-20180306012644-bacd9c7ef1dd
go: downloading github.com/modern-go/reflect2 v1.0.1
go: downloading github.com/PuerkitoBio/gqhttp v1.5.1
go: downloading github.com/antchfx/htmlquery v1.2.3
go: downloading github.com/antchfx/xmlquery v1.2.4
go: downloading github.com/gobwas/glob v0.2.3
go: downloading github.com/kennygrant/sanitize v1.2.4
go: downloading github.com/saintfish/chardet v0.0.0-20210816061221-3af4cd4741ca
go: downloading github.com/temoto/robotstxt v1.1.1
go: downloading google.golang.org/appengine v1.6.6
go: downloading github.com/mgutz/ansi v0.0.0-20200706080029-d51e80ef957d
go: downloading golang.org/x/crypto v0.0.0-20200622136231-75b288015ac9
```

3. Move GoSpider to System Path (Optional)

Move the binary so it can be run from any location.

```
sudo cp ~/go/bin/gospider /usr/local/bin/
```

[illegible]

4. Basic Crawl Command

Performs a simple crawl on the given website.

Made By Moez Javed

gospider -s https://example.com

[illegible]

5. Set Crawling Depth

Increase crawl depth to 2 levels.

```
gospider -s https://example.com -d 2
```

```
(kali㉿kali)-[~]
$ gospider -s https://cust.edu.pk -d 2
[robots] - https://cust.edu.pk
[url] - [code-200] - https://cust.edu.pk
```

6. Parse JavaScript Files

Enable JavaScript parsing to find hidden endpoints.

```
gospider -s https://example.com --js
```

7. Include robots.txt and sitemap.xml

Parse hidden paths from robots.txt and sitemap.xml.

gospider -s https://example.com --robots --sitemap

8. Set Concurrency and Threads

Faster crawling using concurrent requests and threads.

gospider -s https://example.com -c 10 -t 10

9. Save Output to Folder

Saves results in structured output folder.

```
gospider -s https://example.com -o output_folder
```

10. Crawl Multiple Sites from File

Read multiple target URLs from a file and crawl them.

gospider -S targets.txt -o results

11. Exclude Unwanted File Types

Exclude extensions like images and fonts to reduce noise.

```
gospider -s https://example.com --blacklist woff,css,jpg,png
```

12. Get Raw URLs

Useful for integration with fuzzing or scanning tools.

```
gospider -s https://example.com --raw
```

13. Quiet Mode (Clean Output)

Suppress banners and only output URLs.

```
gospider -s https://example.com --quiet
```

14. Disable Redirects

Stops GoSpider from following HTTP redirects.

```
gospider -s https://example.com --no-redirect
```

15. Use With Burp Suite Proxy

Monitor crawl traffic through Burp Suite.

```
gospider -s https://example.com -p http://127.0.0.1:8080
```

16. Combine With ffuf for Fuzzing

Pass URLs to ffuf for further fuzzing.

```
gospider -s https://example.com --raw --quiet | tee urls.txt
```

```
ffuf -u https://example.com/FUZZ -w urls.txt
```

Use Cases

- Discover hidden files and directories
- Extract endpoints from JavaScript files
- Prepare wordlists for fuzzing
- Combine with scanners like ffuf, sqlmap, nuclei
- Ideal for Bug Bounty and Pentest Recon

Ethical Hacking Reminder

Use GoSpider only on systems you have legal permission to test. Unauthorized scanning is illegal and unethical.

Made By Moezz Javed

Resources

GitHub: <https://github.com/jaeles-project/gospider>

Language: Golang

License: MIT

Hakrawler

Tool Overview

Hakrawler is a fast, simple web crawler designed for reconnaissance and URL discovery. It's particularly useful for bug bounty hunters, penetration testers, and web application auditors.

Installation

go install github.com/hakluke/hakrawler@latest

```
(kali@kali) ~$ go install github.com/hakluke/hakrawler@latest
go: downloading github.com/hakluke/hakrawler v0.0.0-20241221204001-76152559a809
go: downloading github.com/gocolly/colly/v2 v2.1.1-0.20220308084714-a61109486557
go: downloading github.com/PuerkitoBio/goquery v1.8.0
go: downloading github.com/antchfx/htmlquery v1.2.4
go: downloading github.com/antchfx/xmlquery v1.3.9
go: downloading github.com/nlnwa/whatwg-url v0.1.0
go: downloading github.com/temoto/robotstxt v1.1.2
go: downloading golang.org/x/net v0.0.0-20220225172249-27dd8689420f
go: downloading google.golang.org/appengine v1.6.7
go: downloading github.com/antchfx/xpath v1.2.0
go: downloading github.com/golang/groupcache v0.0.0-20210331224755-41bb18bfe9da
go: downloading github.com/andybalholm/cascadia v1.3.1
go: downloading github.com/willf/bitset v1.1.10
go: downloading golang.org/x/text v0.3.7
go: downloading google.golang.org/protobuf v1.27.1
```

✓ Ensure your Go binary path (e.g., \$HOME/go/bin) is added to your system PATH.

Basic Usage

Scan a single URL:

echo https://example.com | hakrawler

```
(kali@kali)-[~]  
$ echo https://cust.edu.pk | hakrawler  
  
https://cust.edu.pk/  
https://cust.edu.pk/  
https://cust.edu.pk/  
https://cust.edu.pk/about/  
https://cust.edu.pk/about-capital/messages/chancellors-message/  
https://cust.edu.pk/about-capital/messages/vice-chancellor-s-message/  
https://cust.edu.pk/my-capital/allpolicy/undergrad-regulations/  
https://cust.edu.pk/my-capital/allpolicy/ms-regulations/  
https://cust.edu.pk/my-capital/allpolicy/phd-regulations/  
https://cust.edu.pk/my-capital/allpolicy/general-policies/  
https://cust.edu.pk/my-capital/allpolicy/students-discipline/  
https://cust.edu.pk/my-capital/allpolicy/university-dress-code/  
https://cust.edu.pk/my-capital/academic-calendar/
```

Most Common Options

- d 3 : Crawl depth level (default is 2)
- w : Show the link where each URL was found
- s : Show source element (e.g., href, script, form)
- subs : Include subdomains in the crawl
- dr : Disable following HTTP redirects
- proxy : Use a proxy (e.g., http://127.0.0.1:8080)
- t 20 : Set number of threads to use (default is 8)
- u : Only output unique URLs
- json : Output results in JSON format
- insecure : Disable TLS verification for HTTPS sites
- h : Add custom headers (e.g., -h "Cookie: a=b;;User-Agent: X")

Advanced Usage Examples

1. Crawl deeply and show source elements:

```
echo https://example.com | hakrawler -d 3 -s
```

2. Crawl with thread tuning and unique URLs:

```
echo https://example.com | hakrawler -d 3 -t 20 -u
```

3. Crawl with subdomain discovery:

```
echo https://example.com | hakrawler -subs
```

4. Crawl via proxy (e.g., Burp Suite):

```
echo https://example.com | hakrawler -proxy http://127.0.0.1:8080
```

5. Crawl with SSL verification disabled:

```
echo https://example.com | hakrawler -insecure
```

Multiple URLs from a File

Step 1: Create a file named urls.txt with the following content:

https://site1.com

https://site2.com

https://cust.edu.pk

Step 2: Run:

```
cat urls.txt | hakrawler -d 2 -subs -s
```

Notes

- Hakrawler does not support the -url flag directly. It accepts input via stdin (standard input).
- Combine Hakrawler with tools like waybackurls, httpx, or gf for a more advanced and automated workflow.
- Always ensure you have authorization before scanning or crawling any domain.