🦹 Google Dorks: A Beginner's Notebook

Author : https://github.com/SagarBiswas-MultiHAT

☠️ PART_1: What Are Google Dorks?

Dorks are advanced search operators that help you find specific information on the web that regular searches might miss. Think of them as "search superpowers" — they let you filter results by domain, file type, URL structure, and more.

☠️ Who Uses Them?

- Researchers looking for academic papers or datasets
- Security professionals testing website vulnerabilities
- Students finding study materials in specific formats
- Journalists digging into public records
- Marketers analyzing competitor strategies

🏳️ The 10 Essential Google Dorks

1. site: — Search Within a Specific Website

What it does: Limits your search to a single domain or website.

Basic Syntax: site:domain.com "search term"

💣 Beginner **Examples**:

A) site:wikipedia.org "artificial intelligence"

→ Shows only Wikipedia pages about AI.

B) site:who.int malaria vaccine

→ Finds WHO pages discussing malaria vaccines.

or

B) site:who.int “malaria vaccine”

☠️ **Tip:** Always put **"malaria vaccine"** in quotes if you want results specifically about the malaria vaccine, rather than pages that mention **malaria** and **vaccine** separately.

C) site:docs.python.org "for loop"

→ Searches only Python's official documentation for "for loop."

⚜️ When to use it:

⊗ You trust a specific source and want results only from there

⊗ You're exploring a large website without good internal search

⊗ You want to see how many pages a site has indexed

---

2. intitle: — Search Words in Page Titles

What it does:
Returns pages where your keyword appears in the HTML title tag (the text you see in browser tabs).

Basic Syntax: intitle:"keyword"

💣 Beginner **Examples**:

A) intitle:"getting started" site:docs.github.com

→ Finds GitHub docs pages with "getting started" in the title.

B) intitle:"annual report" site:tesla.com

→ Locates Tesla's annual report pages.

C) intitle:"investor relations" site:tesla.com

→ Finds pages on Tesla's website with **"investor relations"** in the page title.

✴ **When to use it:**

⊗ You want pages specifically *about* a topic (titles usually indicate main focus)

⊗ You're looking for structured pages like reports, guides, or policies

---

3. inurl: — Search Words in URLs

What it does: Finds pages with specific text in their web address.

Basic Syntax: inurl:keyword

💣 Beginner **Examples**:

A) inurl:blog site:openai.com

→ Shows only OpenAI's blog posts.

B) inurl:careers "data scientist" site:microsoft.com

→ Finds Microsoft career pages for data scientists.

C) inurl:press-release site:un.org

→ Locates UN press releases directly.

✴ **When to use it:**

⊗ URLs often reveal page types (blog, careers, docs, product pages)

⊗ Faster than navigating site menus manually

---

4. filetype: — Search by File Format

What it does: Finds specific document types like PDFs, Excel sheets, PowerPoints, or CSVs.

Basic Syntax: filetype:extension "search term"

💣 Beginner **Examples**:

A) filetype:pdf "machine learning cheat sheet"

→ Finds downloadable ML cheat sheets in PDF format.

B) filetype:ppt climate change site:nasa.gov

→ Locates NASA PowerPoint presentations on climate change.

C) filetype:xlsx "budget 2024" site:data.gov

→ Finds public Excel budget files from U.S. government data.

✯ **When to use it:**

⊗ You need downloadable resources (reports, templates, datasets)

⊗ You prefer certain formats for printing, editing, or analysis

---

5. ext: — Search by File Extension

What it does: Similar to filetype:, but works with web file extensions like .php, .asp, or .html.

Basic Syntax: ext:extension keyword

💣 Beginner **Examples**:

A) ext:csv "population data" site:worldbank.org

→ Finds World Bank CSV datasets on population.

B) ext:docx "project proposal template"

→ Locates editable Word document templates.

C) ext:sql site:github.com

→ Finds SQL files hosted on GitHub (useful for developers).

✯ **When to use it:**

⊗ You're looking for raw data or code files

⊗ You need machine-readable formats for automation or analysis

---

6. cache: — View Cached Versions of Pages

What it does:

Shows the **saved copy** of a webpage that **Google last stored** when its crawler (*Googlebot*) visited the site.

Basic Syntax: cache:url.com

💣 Beginner **Examples**:

A) cache:wikipedia.org

→ often returns “did not match any documents”.

B) cache:who.int/news

→ often returns “did not match any documents”.

Why: Google’s public cache links are no longer reliably available; cache: frequently fails even for big sites.

💣 **Wayback Machine (reliable)**

If you want a guaranteed snapshot (use this instead):

A) https://web.archive.org/web/*/https://www.wikipedia.org/
B) https://web.archive.org/web/*/https://www.who.int/news

♠ **Structure:**

```
https://web.archive.org / web / <timestamp-or-*> / <original-URL>
         ↑          ↑                ↑
       host      service        which snapshot
```

✦ **What each piece means**

- ⊗ web.archive.org — the Internet Archive host.
- ⊗ web — the Wayback “web capture” service (i.e., archived web pages).
- ⊗ * — a wildcard meaning “all timestamps” (show every saved snapshot; opens the calendar/list view).
- ⊗ https://www.who.int/news — the original page you want to inspect.

💣 Wayback Machine **Examples** in detail:

- ⊗ Show every snapshot (calendar):
  https://web.archive.org**/web/***/https://www.who.int/news
- ⊗ Show a specific snapshot (timestamp [15 Oct 2021]): YYYYMMDD
  https://web.archive.org**/web/20211015**/https://www.who.int/news
- ⊗ **Save the current page** to the Wayback Machine now:
  https://web.archive.org**/save**/https://www.who.int/news

✯ **When to use it:**

- ⊗ A website is temporarily offline
- ⊗ Content was recently changed or deleted
- ⊗ You want to see what a page looked like before an update

📖 **Note:**

If you see a message like *“Your search did not match any documents”* after using the cache: command, it means **Google doesn’t have a saved (cached) copy** of that webpage.

cache: asks Google for its last saved copy, but that operator often returns **"did not match any documents"** now — use the Wayback Machine (**https://web.archive.org**) for reliable archived snapshots

This can happen if:

- The page is new or not yet crawled.
- The website blocks caching.
- The page was removed from Google's index.

☠️ **Quick Tip — Check if a Page Blocks Caching**

To see whether a webpage prevents caching, open the **page source** and look for:

- **For Google & other search engines:**
  - <meta name="robots" content="noarchive">

→ Tells search engines **not to store or display cached copies** of the page.

- **For the Wayback Machine (web.archive.org):**
  - <meta name="internetarchive" content="noarchive">

→ Acts as a **hint** to Internet Archive **not to archive** this page.

---

7. related: — Find Similar Websites

What it does: Discovers websites similar in topic, structure, or audience to a given site.

Basic Syntax: related:domain.com

💣 Beginner **Examples**:

A) related:wikipedia.org

→ Shows reference sites like Encyclopedia Britannica or Fandom wikis.

B) related:unsplash.com

→ Finds alternatives like Pexels, Pixabay, or Freepik.

C) related:stackoverflow.com

→ Discovers coding Q&A sites like Stack Exchange or Reddit programming forums.

✴️ **When to use it:**

⊗ You want alternatives to a favorite site

⊗ You're researching competitors or similar resources

---

8. info: — Get Info About a Webpage

What it does:
Provides a summary panel with links to the cached version, similar pages, and pages linking to the URL.

Basic Syntax: info:url.com

💣 Beginner **Examples**:

A) info:openai.com

→ Shows Google's stored info about OpenAI's website.

B) info:bbc.com

→ Displays cached link, related sites, and pages referencing BBC.

✴ **When to use it:**

- Quick overview of what Google knows about a domain
- Checking if your own site is properly indexed

---

9. intext: — Search Words in Page Body

What it does: Finds pages with specific words in the main content (not just titles or URLs).

Basic Syntax: intext:"exact phrase"

💣 Beginner **Examples**:

A) intext:"return policy" site:amazon.com

→ Jumps directly to Amazon pages mentioning return policies.

B) intext:"scholarship deadline" site:mit.edu

→ Finds MIT pages with scholarship deadlines in the text.

C) intext:"free trial" site:adobe.com

→ Locates Adobe pages discussing free trials.

✴ **When to use it:**

- You're looking for specific mentions or phrases buried in long pages
- You want content that actually discusses your topic, not just mentions it in passing

---

10. allintitle: — Match Multiple Words in Titles

What it does: Requires *all* specified words to appear in the page title (more restrictive than intitle:).

Basic Syntax: allintitle:word1 word2 word3

💣 **Beginner Examples:**

A) allintitle:python tutorial beginners

→ Finds pages with "Python," "Tutorial," and "Beginners" all in the title.

B) allintitle:project proposal template free

→ Returns pages specifically titled with all four terms.

C) allintitle:cybersecurity best practices 2025

→ Filters for pages with all keywords in the title.

✴ **When to use it:**

⊗ You want highly relevant, focused results

⊗ You're tired of broad matches and need precision

Examples: **Combining Operators for Power Searches**

Google Dorks get *really* powerful when you stack them. Here's how:

💣 Example 1: **Research Papers from a University**

A) site:harvard.edu filetype:pdf "climate change"

→ Finds Harvard-hosted PDFs about climate change.

💣 Example 2: **Job Postings for Remote Roles**

A) inurl:careers intext:"remote" site:microsoft.com
B) inurl:careers intext:"remote" site:.bd

→ Shows career pages mentioning remote work.

C) inurl:career site:grameenphone.com

→ Shows career pages from Grameenphone .

💣 Example 3: **Exclude Unwanted Terms**

A) site:medium.com "machine learning" -tutorial

→ Finds Medium articles on ML but excludes tutorials.

💣 Example 4: **Find Open Datasets**

A) filetype:csv "covid 19" site:data.gov

→ Locates public COVID-19 CSV datasets from U.S. government sources.

💣 Example 5: **Security Audit (Your Own Site Only!)**

A) site:yoursite.com filetype:sql

→ Checks if any SQL files are accidentally indexed (remove them if found).

✦ Try these yourself in Google to build confidence:

💣 Exercise 6: **Find Study Materials**

A) site:mit.edu filetype:pdf "linear algebra"

→ What kind of documents do you find?

💣 Exercise 7: **Discover Competitor Blogs**

B) inurl:blog site:hubspot.com "content marketing"

→ How many relevant blog posts appear?

💣 Exercise 8: **Search for Excel or ZIP data (sometimes CSVs are inside archives)**

C) site:worldbank.org (filetype:xls OR filetype:xlsx OR filetype:zip) population

→ If CSVs aren't indexed, XLS/XLSX or ZIPs might be.

💣 Exercise 9: **Check Your Own Website**

D) site:yourdomain.com intitle:admin

→ Did anything unexpected show up?

💣 Exercise 10: **Research a Topic Deeply**

E) allintitle:cybersecurity best practices 2025

→ How do results differ from a regular search?

📖 **Quick Reference Cheat Sheet**

Operator	Purpose	Example
site:	Search within a domain	site:wikipedia.org AI
intitle:	Match keyword in page title	intitle:"login"
inurl:	Match keyword in URL	inurl:admin
filetype:	Search by file format	filetype:pdf report
ext:	Search by extension	ext:xlsx finance
cache:	View cached version	cache:bbc.com
related:	Find similar sites	related:spotify.com
info:	Get info about a site	info:github.com
intext:	Match in page content	intext:"terms of service"
allintitle:	Require all words in title	allintitle:python tutorial

---

🚩 **PART_2: What Are Some Other Advanced Search Operators?**

Beyond Google Dorks like site: and filetype:, there are several other operators that make searches more precise and powerful.

🩸 **Boolean Operators:** These help combine or exclude terms to refine results.

1. OR (Logical OR)

What it does: Returns results that contain *either* term (or both).

Syntax:

term1 OR term2
term1 | term2

💣 Example:

A) Python OR Java tutorial

→ Finds tutorials on Python, Java, or both.

🌟 **Use case:**

- Searching for synonyms (car OR automobile)
- Exploring multiple topics (blockchain OR cryptocurrency)

📖 **Note:** OR must be in uppercase, or use the pipe symbol | : Python | Java tutorial

---

2. AND (Logical AND)

What it does: Returns results that contain *both* terms.

Syntax: term1 AND term2

💣 Example:

A) machine learning AND healthcare

→ Finds pages discussing both machine learning *and* healthcare.

📖 **Note:** Google assumes AND by default, so these are equivalent:

A) machine learning healthcare
A) machine learning AND healthcare

🌟 **When to use it explicitly:**

⊗ For clarity in complex queries

⊗ When combining with other operators

---

3. - (Minus/Exclusion)

What it does: Excludes results containing a specific term.

Syntax: search term -excluded_term

💣 Example:

A) Python tutorial -video

→ Finds Python tutorials but excludes video results.

B) Jaguar -car

→ Finds results about the animal, not the car brand.

C) site:reddit.com "best laptop" -gaming

→ Reddit threads about laptops, excluding gaming discussions.

☠️ **Tips:**

- No space between -and the excluded term
- Can be used multiple times: site:github.com "wifi deauth" -tutorial -fork

→ Finds GitHub repositories mentioning **"wifi deauth"**, but **excludes** results that contain the words **"tutorial"** or **"fork"** — helping you find more **original projects** instead of duplicates or guides.

🩸 **Phrase and Wildcard Operators** 🔤 – These operators help you control how Google matches words or phrases in your search.

4. " " (Exact Match / Phrase Search)

What it does: Forces Google to search for the exact phrase in the exact order.

💣 Example:

A) "to be or not to be"

→ Finds that exact Shakespeare quote.

A) **Without quotes:** to be or not to be

→ Might return pages with those words in any order or separately.

⚜️ **Use cases:**

⊗ Searching for quotes

⊗ Finding error messages: "error 404 not found"

⊗ Locating specific product names: "iPhone 15 Pro Max"

---

5. * (Wildcard)

What it does: Acts as a placeholder for any word or phrase.

💣 Example:

A) "Python is * programming language"

→ Finds "Python is a powerful programming language," "Python is an easy programming language," etc.

B) "best * for beginners"

→ Finds "best laptop for beginners," "best camera for beginners," etc.

⚜️ **Use cases:**

⊗ Completing partial phrases

⊗ Finding variations of a sentence

⊗ Song lyrics: "I want to * your hand" → Beatles song

🩸 **Numeric Range Operators** 🔢 – These help you **search within a specific number range** (like years, prices, or sizes).

6. .. (Number Range)

What it does: Searches within a numeric range (years, prices, quantities).

Syntax: term min..max

💣 Example:

A) laptop $500..$1000

→ Finds laptops priced between $500 and $1,000.

B) smartphones 2020..2023

→ Finds smartphones released between 2020 and 2023.

C) "best movies" 1990..2000

→ Finds lists of best movies from the 1990s.

⚜ **Use cases:**

⊗ Price filtering

⊗ Date ranges

⊗ Statistical data (population 1000000..5000000)

🩸 **Location-Based Operators** 📍**:** These operators help you **filter Google search results by geographic region** — useful when you want info from a specific country, city, or domain zone.

7. location: (Search by Location)

What it does:
Filters results by geographic location (especially for news).

💣 Example:

A) location:London "tech conference"

→ Finds tech conferences in London.

📖 **Note:** This works better for news and local results. For general searches, include the location in your query:

B) "coffee shops" New York

🩸 **Domain and URL Operators (Recap + Extras)** 🌐 – These operators help you **search within specific websites, domains, or URL structures**, making your queries more targeted and powerful.

8. site: with Subdomains

You already know site:, but you can use it for subdomains:

💣 Example:

A) site:support.google.com "password reset"

→ Searches only the **Support subdomain** of Google for pages mentioning *password reset*.

B) site:*.gov "climate data"

→ Searches all government sites (.gov domains).

C) site:*.microsoft.com "privacy policy"

→ Searches across **all subdomains** of Microsoft's website (e.g., learn.microsoft.com, support.microsoft.com, etc.).

---

9. inanchor: (searches for a single word or phrase in the anchor text)

What it does: Finds pages based on the anchor text of links pointing *to* them.

💣 Example:

A) inanchor:"click here"

→ Finds pages that other sites link to using "click here" as the link text.

B) inanchor:"download report"

→ Finds pages that are linked with "download report" as the clickable text.

C) inanchor:"ethical hacking course"

→ Finds pages that other sites link to using *"ethical hacking course"* — useful for analyzing backlinks or course promotions.

D) inanchor:"data breach news"

→ Finds articles that are commonly linked under that specific anchor phrase.

E) inanchor:"cybersecurity tools" site:github.com

→ Finds **GitHub repositories** that other pages link to using *"cybersecurity tools"* as the anchor text.

✴ **Use Cases:**

⊗ **SEO research:** discover what words others use when linking to your website.

⊗ **OSINT / research:** find pages that are often referenced with a specific keyword (e.g., "leaked database").

⊗ **Content discovery:** locate popular sources based on how they're linked across the web.

---

10. allinanchor: (searches for multiple words appearing together in the anchor text)

What it does:

helps you find pages that other websites link to using **all the given words** in their anchor text — making it powerful for SEO, competitor insights, and open-source investigations.

💣 Example:

A) allinanchor: cybersecurity training free

→ Finds pages that have **all those words** — "cybersecurity", "training", and "free" — somewhere in the anchor text of links pointing to them.

B) allinanchor: bug bounty guide

→ Finds pages that are linked using **all those words** in the anchor text.

✴ **Use Cases:**

- ⊗ **SEO optimization:** Find pages linked with multiple targeted keywords — helpful for understanding how people describe your content.
- ⊗ **Competitor analysis:** Discover which backlinks or anchor phrases competitors are ranking for.
- ⊗ **Content planning:** Identify trending keyword combinations that attract backlinks (e.g., *"free cybersecurity training"*).
- ⊗ **OSINT research:** Locate frequently linked resources or reports containing multiple key terms (e.g., *"leak data breach"*).

🩸 **Date and Freshness Operators** 📅 – These operators help you **filter Google search results by time**, so you can find the **most recent or time-specific content**.

11. before: and after: (Date Filters)

What it does: Filters results by publication date.

Syntax:

search term before:YYYY-MM-DD
search term after:YYYY-MM-DD

💣 Example:

A) "COVID-19 vaccine" after:2021-01-01

→ Finds articles published after January 1, 2021.

B) "Brexit news" before:2020-12-31

→ Finds Brexit news published before 2020.

C) "AI trends" after:2023-01-01 before:2024-01-01

→ Finds AI trends articles from 2023 only.

📖 **Note:** Google also has a "Tools" menu under the search box where you can filter by time (past hour, past 24 hours, past week, past year, custom range).

🚩 Example: **Combining Multiple Operators** – Advanced Search Operators: This is where search becomes truly powerful.

💣 Example 1: **Find Recent Academic Papers**

A) site:arxiv.org "neural networks" filetype:pdf after:2024-01-01

→ Recent neural network papers from arXiv.

💣 Example 2: **Competitor Analysis**

A) site:competitor.com inurl:blog -inurl:author "content marketing"

→ Competitor's blog posts on content marketing, excluding author pages.

💣 Example 3: **Job Hunting**

A) intitle:"software engineer" inurl:careers site:*.com "remote" -internship

→ Remote software engineer jobs, excluding internships.

💣 Example 4: **Fact-Checking**

A) "climate change" site:.edu OR site:.gov filetype:pdf after:2020-01-01

→ Authoritative sources (education or government) on climate change, PDF format, recent.

💣 Example 5: **Security Audit**

A) site:yoursite.com (inurl:admin OR inurl:login OR inurl:wp-admin) -site:www.yoursite.com

→ Checks for admin/login pages on subdomains that shouldn't be public.

✦ Try these searches to master the operators:

💣 Exercise 6: **Find Product Reviews in a Price Range**

A) laptop review 2023..2024 $500..$1000

or

A) “laptop review” 2023..2024 $500..$1000

Always use "laptop review" with quotes if you want **results specifically about laptop reviews**, not random pages mentioning laptops and reviews separately.

💣 Exercise 7: **Exclude Certain Sites**

"data science tutorial" -site:youtube.com -site:udemy.com

💣 Exercise 8: **Find Government Reports on a Topic**

site:.gov "renewable energy" filetype:pdf after:2022-01-01

💣 Exercise 9: **Search for Song Lyrics with a Wildcard**

"I will always * you" lyrics

The asterisk * works as a **wildcard** — it tells Google to fill in any missing word(s) between the quoted phrases.

→ “I will always ___ you” --- can be one or more missing words inside quotes.

💣 Exercise 10: **Combine Everything**

site:github.com "machine learning" (Python OR R) -tutorial

→ Finds **GitHub repositories or projects** related to *machine learning*, written in **Python or R**, but **excludes tutorials**.

🚩 Operator Cheat Sheet

Operator	Purpose	Example
OR , |	Returns results that contain *either* term (or both).	site:github.com "machine learning" (Python OR R) -tutorial
AND	Both terms (default behavior)	machine learning AND ethics
-	Exclude term	recipe -meat
" "	Exact phrase	"artificial intelligence"
*	Wildcard	"best * for data science"
..	Number range	laptop $300..$800
before:	Before date	news before:2023-01-01
after:	After date	article after:2024-01-01
inanchor:	Anchor text of backlinks	inanchor:"learn more"
allinanchor:	All words in anchor text	allinanchor:best python tutorial
around(X)	Words near each other	Apple around(3) innovation

🚩 PART_3: Real-World Applications of Google Dorks

🔐 1. Ethical Hacking & Penetration Testing

How it's used:
Security researchers use Google Dorks to find exposed admin panels, configuration files, or unprotected directories. They report these to website owners to help fix vulnerabilities.

💣 Examples:

How to use: run these queries with site:yourdomain.com (or omit site: to search broadly) **only** for domains you own or have authorization to test.

1. Exposed admin / login panels

A) intitle:"login" inurl:admin

Purpose: locate pages titled “login” that include admin in URL.
Defensive action: ensure admin pages are not publicly indexed; restrict by IP, require MFA, add WAF/CSRF protection, and set X-Robots-Tag: noindex.

B) inurl:/admin/login OR inurl:/administrator

Purpose: common admin endpoints.
Defensive action: move/obscure admin paths, restrict access, add rate limits and strong auth.

---

2. Exposed configuration / credential files

A) filetype:env "DB_PASSWORD" OR "DB_USER"

Purpose: find .env files containing environment variables.
Defensive action: remove such files from web root, rotate secrets, place config outside webroot, and add server rules to block these filetypes.

B) filetype:ini OR filetype:cfg OR filetype:conf "password"

Purpose: config files with credentials.
Defensive action: same as above; enforce least privilege and secret management.

---

3. Backup & directory listings

A) intitle:"index of" inurl:backup

Purpose: public directory listings showing backups.
Defensive action: disable directory listings, move backups to private storage, audit S3/bucket permissions.

B) intitle:"index of" "backup" OR "db_backup" OR ".sql"

Purpose: find DB backup files.
Defensive action: secure backups, enforce encryption at rest and transit.

---

4. Exposed documents (sensitive docs)

A) filetype:pdf OR filetype:xls OR filetype:doc "confidential" OR "sensitive"

Purpose: publicly indexed docs marked confidential.
Defensive action: remove from public web, audit share permissions, enforce DLP.

B) site:yourdomain.com filetype:xlsx intext:"password"

Purpose: detect spreadsheets with password strings.
Defensive action: delete/secure, rotate exposed credentials.

---

5. Cloud storage & buckets indexed

A) inurl:s3.amazonaws.com "your-bucket-name" OR intitle:"Index of /" "s3"

Purpose: find S3 buckets or public cloud storage.
Defensive action: ensure correct bucket ACLs, block public access, enable bucket logging.

B) site:storage.googleapis.com inurl:your-project

Purpose: GCP storage exposure.
Defensive action: apply IAM, remove anonymous access.

---

6. Exposed source code & scripts

A) filetype:php "DB_HOST" OR filetype:py "import psycopg2"

Purpose: find source files that may contain secrets.
Defensive action: remove from public repo, use private repos, secret scanning.

B) inurl:/.git

Purpose: exposed .git repositories.
Defensive action: block access to VCS directories, remove .git from web root.

---

7. Potential SQL Injection / parameterized URLs

A) inurl:"?id=" "php" OR "asp" OR "aspx"

Purpose: pages with query params often targeted by SQLi.
Defensive action: use prepared statements, input validation, WAF rules.

B) inurl:".php?id=" intitle:"error" OR "mysql_fetch"

Purpose: locate pages that produce database error strings.
Defensive action: disable verbose errors in production, sanitize inputs.

---

8. Error messages / stack traces

A) "intext:Warning: mysql" OR "Stack trace" OR "Exception in thread"

Purpose: public error messages revealing software/version/paths.
Defensive action: turn off detailed errors, centralize error logging.

---

9. Exposed cameras / IoT panels (ethical caution)

A) inurl:"/viewer/index.shtml" OR inurl:"/liveview"

Purpose: find IoT device web UIs (often passwordless or default credentials).
Defensive action: change defaults, restrict to VPN, update firmware.

(Only scan devices you own or manage.)

---

10. API keys, tokens, secrets

A) intext:"AKIA" OR intext:"AIza" OR intext:"BEGIN RSA PRIVATE KEY"

Purpose: AWS/Azure/GCP keys and private keys patterns.
Defensive action: revoke exposed keys, rotate, implement secret scanning in CI, use vaults.

B) site:yourdomain.com filetype:json intext:"api_key"

Purpose: Find JSON files on your site that contain exposed API keys.
Defensive action: Remove or restrict those files, rotate the keys, and add secret scanning plus a vault for key storage.

---

11. Mail lists and emails

A) intext:"@yourdomain.com" filetype:xls OR filetype:csv

Purpose: find leaked email lists or contact sheets.
Defensive action: audit data exposure, notify affected users, improve access controls.

---

12. Admin pages by CMS

A) site:yourdomain.com inurl:wp-admin OR inurl:administrator/index.php

Purpose: find CMS admin entry points.
Defensive action: harden CMS, keep plugins/themes updated, use 2FA and IP restrictions.

---

13. Misc — exposed logs

A) filetype:log "GET /" OR "POST /" intext:yourdomain

Purpose: publicly available log files.
Defensive action: never expose logs, sanitize logs, restrict access.

---

14. Aggregator tips / combine operators

A) site:yourdomain.com (filetype:env OR filetype:sql OR filetype:bak OR intitle:"index of")

Purpose: single search to catch multiple exposure types.
Defensive action: run periodically as part of external attack surface monitoring.

📖 **Ethical note:**

Never access or exploit anything you find. Report vulnerabilities responsibly through proper channels (bug bounty programs, security contacts).

🕵️ **2. OSINT (Open-Source Intelligence) Research**

How it's used:
Investigators, journalists, and researchers gather publicly available information without breaching systems.

💣 Example:

A) site:gov.bd filetype:pdf "budget report"

→ Finds public Bangladeshi government budget documents for analysis.

💼 3. Competitive & SEO Analysis

How it's used:
Marketers discover competitor strategies, hidden landing pages, or indexed content.

💣 **Example:**

A) site:competitor.com -inurl:www

(-inurl:word tells Google to **ignore any URL containing that word**; for example, -inurl:www excludes URLs with "www" in them.)

B) site:nytimes.com -inurl:www

→ Reveals subdomains (blog.competitor.com, shop.competitor.com, etc.).

📊 4. Academic & Data Research

How it's used: Students and researchers find datasets, papers, or public records in specific formats.

💣 **Example:**

A) filetype:xlsx "election results" site:eci.gov.in

→ Finds Indian election data in spreadsheet format.

🚩 PART_4: How to Protect Yourself from Google Dorks 🛡️

Google Dorks can expose sensitive information *you* didn't realize was public. Here's how to stay safe:

1. Audit Your Own Website

♠ **Why:**

Before attackers find sensitive data, you can proactively check what Google indexes from your own site using **Google Dorks**. This helps you identify and fix exposed files, admin pages, or logs.

♠ **How to do it:**

1. **Use Google Search Operators (Dorks)**
    - Open Google and enter queries like:
    - site:yoursite.com filetype:pdf
    - site:yoursite.com inurl:admin
    - site:yoursite.com filetype:log
    - site:yoursite.com → limits results to your domain
    - filetype: → searches for specific file types (PDF, log, etc.)
    - inurl: → searches for specific words in the URL (like "admin")

2. **Review Results Carefully**
    - Check if any sensitive files, admin pages, or backups are publicly visible.
3. **Take Immediate Action**
    - Remove unnecessary public files.
    - Protect critical directories using:
        - robots.txt
        - Password protection (.htaccess)
        - Server-level access restrictions
4. **Repeat Regularly**
    - Perform audits periodically (weekly/monthly) to ensure nothing new is exposed.

📖 **Quick note:**

Run Google Dorks on your own site (e.g., site:yoursite.com inurl:admin) to **find exposed sensitive files or pages** and secure them immediately.

---

2. Use robots.txt Properly

♠ **What it is:**

robots.txt is a **text file** you place in the root directory of your website (e.g., https://example.com/robots.txt) that tells search engine crawlers which pages or folders **not to index**.

♠ **How to use it:**

1. **Create a plain text file** named robots.txt.
2. Add rules like this:

```
User-agent: *      # Applies to all crawlers
Disallow: /admin/   # Don't index admin folder
Disallow: /backup/  # Don't index backup folder
Disallow: /config/  # Don't index config folder
```

3. **Upload it to your website's root directory** (the main folder of your site). For example:

   https://yourwebsite.com/robots.txt

4. Search engines will check this file and **skip indexing the disallowed paths**.

♠ **Notes / Tips:**

- User-agent: * applies to **all search engines**.
- You can also block specific crawlers by replacing * with their name (e.g., Googlebot).
- This **doesn't hide the files from direct access**; it only tells crawlers not to index them.

📖 **Quick note:**

Place a robots.txt file in your website's root folder and use Disallow rules to prevent crawlers from indexing sensitive directories.

---

3. Password-Protect Sensitive Directories

♠ **Why:**

Even if a crawler respects robots.txt, anyone can still access those files directly. Password protection **prevents unauthorized users** from opening sensitive areas like admin panels, backups, or configuration files.

♠ **How to do it (Apache example using .htaccess and .htpasswd):**

1. **Create a .htaccess file** in the directory you want to protect (e.g., /admin/) with the following content:

```
AuthType Basic
AuthName "Restricted Area"
AuthUserFile /full/path/to/.htpasswd
Require valid-user
```

   - AuthType Basic → Uses basic authentication.
   - AuthName → The message shown in the login prompt.
   - AuthUserFile → Full server path to the .htpasswd file that stores usernames/passwords.
   - Require valid-user → Only allows users listed in .htpasswd.

2. **Create a .htpasswd file** (outside the web root for security) with username and encrypted password. Example using the command line:

   htpasswd -c /full/path/to/.htpasswd username

   It will prompt you to set a password.

3. **Upload both files** to your server:
   - .htaccess → inside the directory to protect.
   - .htpasswd → outside public web folders (so it cannot be accessed directly).

4. **Test:**
   Visit the protected directory in your browser. You should see a login prompt.

♠ **Server-level alternatives:**

- Nginx: Use auth_basic and auth_basic_user_file in your server config.
- Control panels (cPanel, Plesk) often have **directory protection tools** to do this without manually editing files.

📖 **Quick note:**

Use .htaccess (Apache) or server-level authentication to **lock sensitive directories** — users must enter a username/password to access admin, backup, or config folders.

---

4. Check Google Search Console (GSC)

♠ **Purpose:**

Google Search Console lets you **see which pages of your site are indexed** and gives you tools to **remove sensitive content** from search results.

♠ **How to use it:**

1. **Sign in / set up GSC**
    - Go to https://search.google.com/search-console
    - Add your website property (https://example.com) and **verify ownership** (via DNS, HTML file, or meta tag).
2. **Check indexed pages**
    - In GSC, go to **Coverage → Valid**
    - You'll see all pages Google has indexed.
    - You can search for specific URLs or sensitive folders.
3. **Request removal of a URL**
    - Go to **Removals → New Request**
    - Enter the URL you want temporarily hidden from Google Search.
    - Google will **remove it from search results temporarily** while you fix the content or block it properly.
4. **Optional:**
    - Use **robots.txt** or **password protection** to permanently prevent indexing.
    - Check **Security & Manual Actions** for any exposed or sensitive content warnings.

📖 **Quick note:**

Use Google Search Console to **see what Google has indexed** and submit removal requests for pages or directories that shouldn't be public.

---

5. Keep Software Updated

♠ **Why:**

Outdated software and plugins often have **security vulnerabilities** that hackers can exploit. Regular updates reduce the risk of compromise.

♠ **How to do it:**

1. **Content Management Systems (CMS)**
    - WordPress, Joomla, Drupal, etc.

- Check the **dashboard** for update notifications.
- Apply updates for:
  - CMS core
  - Plugins / extensions
  - Themes

2. **Plugins and Extensions**
   - Remove **unused or abandoned plugins** — they may still have vulnerabilities even if inactive.
   - Update active plugins promptly when updates are released.
3. **Server & Software**
   - Keep your **web server, PHP, MySQL, and frameworks** updated to supported versions.
   - Enable **automatic updates** if possible, especially for minor security patches.
4. **Backups before updates**
   - Always create a **full backup** before updating, so you can restore if anything breaks.

📖 **Quick note:**

Regularly update your CMS, plugins, themes, and server software, and remove unused applications to prevent attackers from exploiting outdated software.

---

6. Enable Two-Factor Authentication (2FA)

♠ **Why:**

Even if attackers get your username and password, 2FA adds a **second layer of security** — usually a temporary code or authentication prompt — making unauthorized access much harder.

♠ **How to enable 2FA:**

1. **For CMS, websites, or online services:**
   - WordPress, Gmail, GitHub, cloud services, and many platforms support 2FA.
   - Go to **Account Settings → Security → Two-Factor Authentication (or 2-Step Verification)**.
2. **Choose a method:**
   - **Authenticator apps** (Google Authenticator, Authy, Microsoft Authenticator) → generates a time-based one-time code.
   - **SMS / phone verification** → sends a code via text message. (Less secure than authenticator apps.)
   - **Hardware keys** (YubiKey, FIDO2) → physical security keys for extra protection.
3. **Enable and test:**
   - Scan the QR code in the app or enter your phone number.

- Try logging in to confirm the second factor works.

4. **Backup codes:**
   - Most services provide backup codes in case you lose access to your 2FA device.
   - Store them **securely offline**.

📖 **Quick note:**

Enable 2FA on your accounts to add a **second layer of protection** — even if your password is compromised, attackers cannot log in without the second factor.

---

7. Monitor Your Digital Footprint

♠ **Why:**

Even if your website is secure, sensitive information can accidentally appear online. Monitoring helps you **catch exposed data early**.

♠ **How to set it up:**

1. **Use Google Alerts**
   - Go to https://www.google.com/alerts
   - Enter search queries to track sensitive content.
2. **Example queries:**
   - site:yoursite.com intitle:admin → Notifies you if your admin page is indexed.
   - "yourcompanyname" filetype:xls password → Notifies you if spreadsheets containing passwords appear online.
   - site:yoursite.com confidential → Tracks accidentally exposed confidential documents.
3. **Customize alerts:**
   - **Frequency:** As-it-happens, daily, or weekly.
   - **Sources:** News, blogs, web, etc.
   - **Email:** Get notifications sent to your inbox.
4. **Action:**
   - If you get an alert about sensitive content, **remove it immediately** from your site or contact the hosting platform.

📖 **Quick note:**

Use Google Alerts with specific queries (e.g., site:yoursite.com intitle:admin) to **monitor your digital footprint** and get notified if sensitive data appears online.

---

8. Educate Your Team

♠ **Why:**

Most security breaches happen because of **human error**. Training your staff reduces the risk of accidental leaks and compromises.

♠ **How to educate your team:**

1. **Regular Training Sessions**
    - Conduct workshops or online training about **data security basics**, including proper file handling and access control.
2. **Password Hygiene**
    - Teach staff to use **strong, unique passwords** for every account.
    - Encourage use of **password managers**.
3. **Recognizing Phishing and Social Engineering**
    - Show examples of **phishing emails, fake login pages, and suspicious links**.
    - Encourage employees to **report suspicious emails or messages** immediately.
4. **Safe File Management**
    - Educate on **not uploading sensitive files** to public folders or sharing them through insecure channels.
    - Use **access-controlled folders** and encrypted storage when needed.
5. **Periodic Assessments**
    - Conduct **simulated phishing tests** or internal audits to reinforce training.

📖 **Quick note:**

Train your team to avoid human errors by teaching proper file handling, strong password use, and how to recognize phishing attempts.

---

9. Use Cloud Storage Securely

♠ **Why:**

Files stored in cloud services can accidentally become public if sharing settings are misconfigured, exposing sensitive information.

♠ **How to do it:**

1. **Check Sharing Settings**
    - Google Drive, Dropbox, OneDrive all have options like:
        - **Private / Only specific people** → safest
        - **Anyone with the link** → only if intentional

        - **Public on the web** → avoid

2. **Review Shared Files Regularly**
    - Periodically audit folders and documents to **remove unnecessary public access**.
3. **Use Folders with Access Control**
    - Group sensitive documents in restricted folders.
    - Assign permissions carefully (view, edit, comment).
4. **Enable Account Security**
    - Enable **2FA** for cloud accounts.
    - Use strong, unique passwords.
5. **Audit Activity**
    - Some services (Google Drive, Dropbox) let you **see who accessed or downloaded** files. Check logs for unusual activity.

📖 **Quick note:**

Keep cloud files private by checking sharing settings, restricting folders, and using 2FA — only share "anyone with the link" when truly needed.

---

10. Regular Backups

♠ **Why:**

If your site or data is attacked (ransomware, accidental deletion, or breach), having **recent backups** ensures you can restore everything **without paying attackers** or losing important information.

♠ **How to do it:**

1. **Automate Backups**
    - Use CMS plugins (WordPress: UpdraftPlus, All-in-One WP Migration) or server scripts to **schedule regular backups**.
    - Decide the frequency: daily, weekly, or monthly based on your site's activity.
2. **Keep Multiple Copies**
    - Store backups in **multiple locations**:
        - External hard drives
        - Offline storage (USB, external disks)
        - Secure cloud storage (encrypted)
3. **Encrypt Backups**
    - Use encryption to prevent unauthorized access if backup files are stolen.

4. **Test Backup Restoration**
    - Occasionally **restore from backup** to make sure files are complete and the process works.
5. **Versioning**
    - Keep **older versions** for rollback in case a recent backup is corrupted or infected.

📖 **Quick note:**

Regularly create **encrypted backups offline and in secure cloud storage** so you can restore your site or data in case of ransomware, deletion, or breach.

🏳️ **PART_5: Legal & Ethical Guidelines AND Final Tips** ⚖️

☑️ **Allowed:**

- Searching publicly indexed information
- Testing your own websites for vulnerabilities
- Reporting security issues responsibly
- Academic research using open data

❌ **Not Allowed:**

- Accessing systems without permission (even if a Dork found them)
- Downloading or distributing private data
- Using Dorks to harm, blackmail, or exploit individuals or organizations
- Ignoring "Authorized Users Only" warnings

📖 **Golden Rule:** *Just because something is findable doesn't mean it's legal to access or use.*

☠️ **Final Tips for Mastering Search Operators**

1. **Start simple, add complexity gradually**: Begin with one operator, then combine as needed.
2. **Use quotes for exact matches**: Especially for error messages, product names, or quotes.
3. **Exclude noise with - :** Remove irrelevant results (ads, forums, certain sites).
4. **Bookmark complex queries**: If you run the same search often, save it.
5. **Experiment and iterate**: Not getting good results? Adjust one operator at a time.
6. **Check Google's official documentation**: Visit Google Search Help for updates.

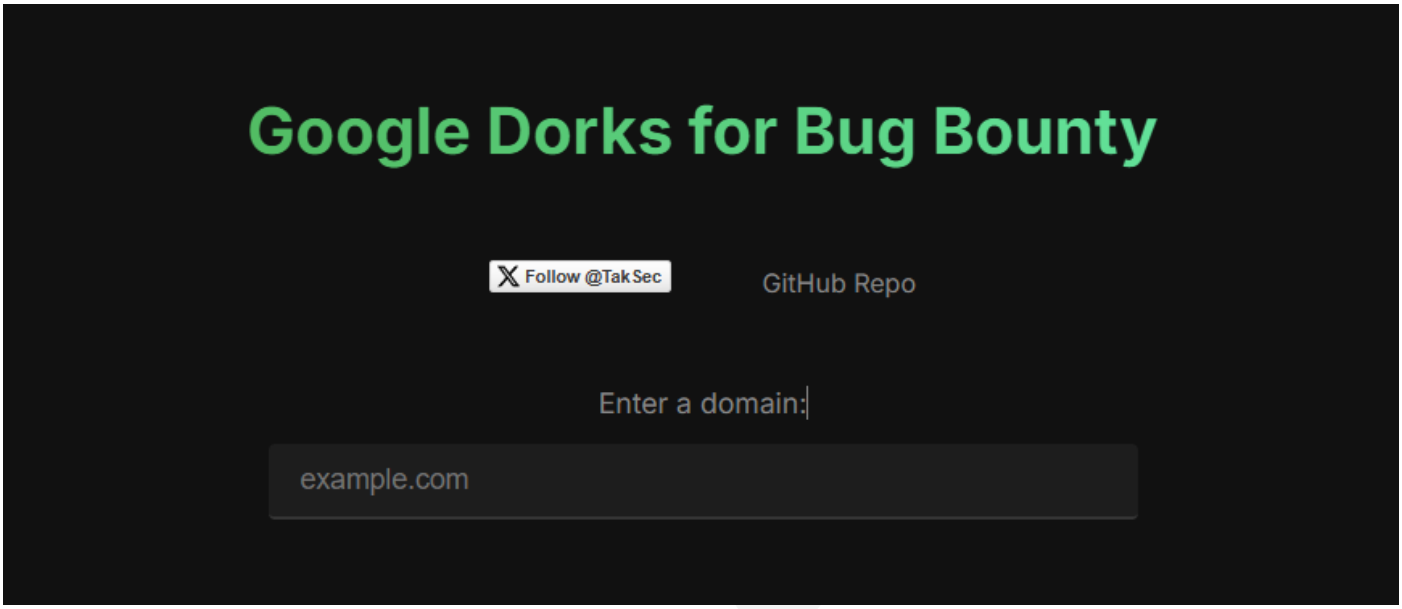
☠️ **But remember:** With great power comes great responsibility. Always respect privacy, follow laws, and use these tools ethically.

🚩 Part_6: Bonus — Google Dorks for Bug Bounty

A useful curated collection of Google Dorks focused on bug-bounty hunting — enter one or more domains to scan and find interesting indexed assets (admin panels, backups, keys, etc.). taksec.github.io

Quick use: site:target.com + a dork from the list to limit results to a scope you own or are authorized to test.
Ethics reminder: Only test targets you own or have written permission to audit.

Link: https://taksec.github.io/google-dorks-bug-bounty/



🚀 **Ready to explore?**
Open Google and start experimenting. You'll be amazed at what you can discover with just a few extra characters in the search bar.

ᕦ(..:: Happy dorking! Permission first, always. — stay ethical ::..)ᕤ

---------- X ----------