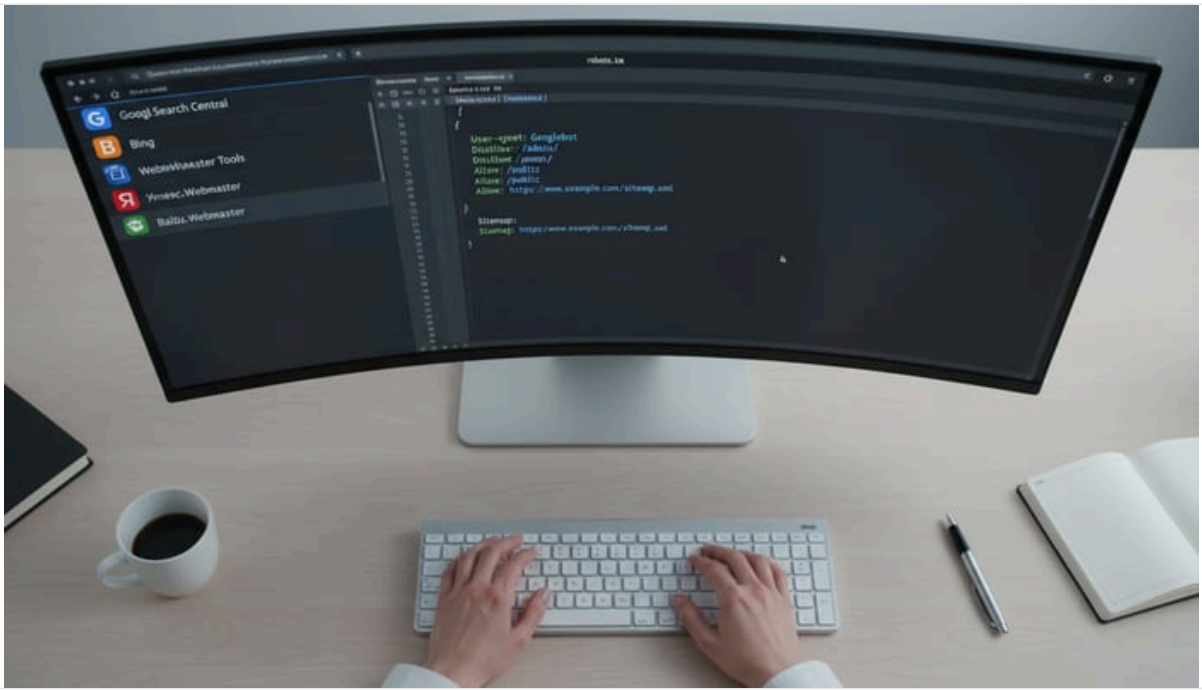
# Robots.txt Directives: A Guide to All Standard & Hidden Rules

Published Invalid Date




## Executive Summary

The `robots.txt` file (Robots Exclusion Protocol, REP) is a long-standing, text-based mechanism for webmasters to tell automated crawlers which parts of a site may or may not be accessed. It was first proposed by Martijn Koster in July 1994 (Source: www.webdesignmuseum.org) and has since become a de-facto standard. In 2022, it was formally standardized as RFC 9309 on the IETF Standards Track (Source: www.rfc-editor.org). Robots.txt is *not* an access control or security mechanism – rather, it is a voluntary "request" to friendly crawlers (search engines and other bots) about crawl preferences (Source: www.rfc-editor.org). By 2021, about **81.9%** of indexed websites had a `robots.txt` file (Source: almanac.httparchive.org), reflecting its ubiquity.

This report provides an in-depth examination of **all known directives and parameters** in `robots.txt`, including obscure and search-engine-specific ones. We cover the standard core (e.g. `User-agent`, `Disallow`, `Allow`) and pattern syntax (wildcards, `$` end-of-line), as well as extensions such as `Sitemap:` links. We then detail nonstandard or less-known directives – for example, `Crawl-delay`, Yandex's `Clean-param` and `Host`, Seznam's `Request-rate` / `Visit-time`, and historical `noindex` rules – including which major search crawlers support them. Throughout, assertions are backed by official documentation, expert analysis, and real-world case studies. For example, Google's Search Central guidance confirms that robots rules like "noindex" (in robots.txt) are *unsupported* (Source: developers.google.com), and that pages blocked by robots can *still be indexed* if linked from elsewhere (Source: searchengineland.com) (and even appear in Google's results with snippets (Source: searchengineland.com). Yandex documents its unique features such as `Clean-param` (to ignore query parameters) (Source: yandex.com), while Bing's engineers have noted that Bing never honored a `noindex` directive in robots.txt (Source: www.seroundtable.com).

We also analyze usage trends (e.g. **table** of search-engine directives support) and real incidents. For instance, one SEO case study reported how host-configured robots rules (added without the webmaster's knowledge) inadvertently disallowed key site sections over time – the affected pages slowly dropped from Google's index (Source: searchengineland.com). From a security stance, researchers warn that including sensitive paths (like `/admin/`, `/backup/`, `/debug/`) in robots.txt actually *paints a target* for attackers (Source: www.theregister.com) (Source: nemocyberworld.github.io). Drawing on data (e.g. from the 2021 HTTP Archive SEO Almanac (Source: almanac.httparchive.org) and search engine blogs (Source: www.askapache.com) (Source: developers.google.com), the report concludes with implications for webmasters (best practices, crawling reliability) and future directions (the potential to extend REP with new consensus rules).

## Introduction and Background

The Robots Exclusion Protocol (REP) is the web's original crawler-control standard. It began as a simple `/robots.txt` file at a site's root, read by indexing bots, indicating which URLs to **disallow** or **allow**. Martijn Koster first introduced the idea on the W3C www-talk mailing list in July 1994 (Source: www.webdesignmuseum.org); famously, he needed it after his site was DOS'ed by an aggressive crawler. Over the next decades, REP remained an *informal* de-facto standard used by virtually all major search engines. Despite its age, REP persisted with little change: by 2025 it had "barely had to evolve" — Google notes the only universally supported extension it gained was the `Allow` directive (Source: developers.google.com).

In September 2022, the IETF formalized these practices as RFC 9309 (Source: blog.seznam.cz). This document codifies the REP language and processing rules on the Internet Standards Track (Source: www.rfc-editor.org). The RFC acknowledges the original 1994 specification by Koster (Source: www.rfc-editor.org) and clarifies how crawlers should parse and cache `robots.txt`, handle redirects, errors, and reading **User-agent**, **Allow**, **Disallow** rules (Source: www.rfc-editor.org) (Source: www.rfc-editor.org). Importantly, the RFC explicitly states that robots directives are *not* an authorization scheme – if you list a path in `robots.txt`, it is openly explorable to any human or adversarial bot, so actual security should use proper access control (e.g. HTTP auth) (Source: www.rfc-editor.org).

In practice, `robots.txt` files use a straightforward grammar. They consist of one or more "groups" (blocks), each beginning with one or more `User-agent:` lines, followed by matching rules. For example:

```
User-agent: *
Disallow: /private/
Allow: /private/special/
Sitemap: https://example.com/sitemap.xml
```

This means "for all crawlers, disallow `/private/` except allow `/private/special/`. Also, here is our sitemap." In **Group 1**, the empty path after `Disallow:` implies *allow everything*. Each rule is a key-value pair separated by colon. The key may be `Allow:` or `Disallow:`, followed by a URL path pattern. The REP syntax (per RFC 9309) specifies that for a matched group, the *most specific* rule (longest path match) takes precedence, and in case of a tie an `Allow` beats a `Disallow` (Source: www.rfc-editor.org). Rules are case-sensitive in the URL path – for example, `Disallow: /Example/` will *not* block `/example/` (Source: searchengineland.com). If no rules match a given crawler or URL, crawling is implicitly allowed (the default is open) (Source: www.rfc-editor.org) (Source: yandex.com).

Since the Internet's early days, the paradigm has been that friendly crawlers should **obey** these directives. Google, Bing, Yandex and others have built their bots to honour the standard rules and many common extensions. However, this voluntary nature means each crawler can choose which directives to support. As we will see, some directives (like `Crawl-delay` or `Clean-param`) are honored by only a few engines, and others (such as a `Noindex` line in robots) are ignored by major crawlers (Source: developers.google.com) (Source: www.seroundtable.com).The following sections detail the syntax, official support, and many "hidden" or lesser-known parameters used in today's `robots.txt` files.

# Core `robots.txt` Syntax and Directives

## Standard Directives: `User-agent`, `Disallow`, `Allow`

The fundamental directives in a `robots.txt` file are **`User-agent`** (identifying the target crawler) and **`Disallow`** (blocking paths). RFC 9309 formalizes this baseline syntax: each rule is either `Allow:` or `Disallow:`, each followed by a path pattern (Source: www.rfc-editor.org). A group of rules applies to the preceding `User-agent` lines (Source: www.rfc-editor.org). For example:

> *`User-agent: Googlebot`*
> *`Disallow: /admin/`*
> *`Allow: /admin/help/`*

This tells Googlebot it may not crawl `/admin/` except it may crawl `/admin/help/`. The keyword `*` is used to match all crawlers (e.g. `User-agent: *`) (Source: stackoverflow.com). By convention, a blank `Disallow:` (no path) means *allow all* (no restrictions). If no rule matches, content is by default crawlable (Source: www.rfc-editor.org).

Crawlers match the longest rule: if multiple directives match a URL, the one with the longest path string wins. If an `Allow` and `Disallow` are exactly equal in length, **`Allow` takes precedence** (Source: www.rfc-editor.org). Case matters: path matching is case-sensitive (Source: searchengineland.com). (In practice, since domain names may be case-insensitive, the RFC advises using punycode or UTF-8 conversion, but those details mainly affect localization (Source: yandex.com).)

Most search engines support the `Allow` directive today (Source: visual-seo.com). For example, Yandex's webmaster documentation explicitly lists `Allow` alongside `Disallow` as a directive that permits crawling (Source: yandex.com). Google too uses `Allow:` rules to carve exceptions out of a `Disallow` tree (Source: visual-seo.com) (Source: www.askapache.com). (Originally `Allow` was an unofficial Google extension introduced in the 1990s; it is now ubiquitous.)

Another common directive is **`Sitemap:`** which tells the crawler where to find the site's XML sitemap(s). While not part of the original REP, it is "an additional standard" recognized by Google, Bing, Yandex and others (Source: www.askapache.com). In the example above, `Sitemap: https://example.com/sitemap.xml` points crawlers to the sitemap URL. Google, in fact, encourages placing sitemaps in robots.txt (and has made it crawlable via this rule) (Source: www.askapache.com). All major search engines honor the `Sitemap:` line for convenience **even though it is outside the formal RFC grammar** (Source: developers.google.com).

### Patterns and Wildcards

Modern crawlers also support simple pattern matching in paths. The most widely used is the `*` wildcard (match any sequence of characters) and the `$` anchor (match end of URL). For example, `Disallow: /*.pdf$` means "disallow all URLs ending in `.pdf`". Google's rep has long allowed `*` and `$` in Disallow patterns (its open-source parser and documentation support this syntax) (Source: visual-seo.com) (Source: www.askapache.com). Yandex also accepts these wildcards. According to Baidu's help, its `Baiduspider` supports both "`*` and `$`" for matching URLs (Source: www.baidu.com). In practice, many sites exploit this ability to block entire file types or URL query parameters. (For example, `Disallow: /*?*` will block any URL containing "?".) A detailed match process applies: once a crawler collects all `Disallow` and `Allow` rules for its user-agent, it finds the rule with the most specific (longest) matching prefix; if that rule is `Allow`, the URL may be crawled, and if it is `Disallow`, the URL is blocked (Source: www.rfc-editor.org).

### Case Sensitivity and Normalization

Directives match URL paths verbatim. That means case matters and the exact string must match from the first character. For example, a directive `Disallow: /Category/` will block URLs like `/Category/Item1` but *not* `/category/item1` – the lowercase vs uppercase mismatch means the second URL is not caught (Source: searchengineland.com). Similarly, the robots parser un-encodes percent-encoded characters before matching (Source: www.rfc-editor.org). Note, however, that although path matching in rules is case-sensitive, most crawlers treat the `User-agent:` and directive names (`User-agent`, `Allow`, etc.) as case-insensitive keywords (Source: yandex.com). In summary, robots.txt rules follow precise string matching on the path portion of URLs, so one must account for URL normalization and case when writing rules.

## Extended and Lesser-Known Directives

Beyond the basic grammar, a number of **nonstandard directives** have appeared. These are not part of the original 1994 REP, but many are supported in practice by particular crawlers.

- **`Allow`**: We already covered `Allow:` as an extension to override disallows. Its support has grown until it is effectively a standard in all major engines (Source: visual-seo.com). Google's robots parser ensures `Allow` rules are honored, and if an `Allow` and a `Disallow` rule both match a URL, Google uses the longer path (often the `Allow` if it's longer) (Source: www.rfc-editor.org).

- **`Crawl-delay`**: This directive was invented to throttle crawling speed (few pages per second). It is *not* part of the official REP grammar【26†】, but some engines use it. **Yandex** supports `Crawl-delay` in robots.txt, e.g. `Crawl-delay: 10` to wait 10 seconds between fetches (Source: yandex.com). **Bing** also honors `Crawl-delay`. Google **does not support** a crawl-delay directive in `robots.txt`: as Google's Matt Cutts explained, many webmasters mis-configure it (e.g. setting it to 100,000) causing effectively no crawling (Source: www.askapache.com). Instead, Google offers crawl-rate controls in Search Console (and internally governs crawl with "host-load" settings) (Source: www.askapache.com). Thus, if you write `Crawl-delay` in `robots.txt`, only Yandex, Bing (and perhaps some custom crawlers) will heed it, not Google.

- **Search-engine-specific "Host"**: Originally Yandex introduced a `Host:` directive to let webmasters declare the site's preferred domain among mirrors. For example, if a site is reachable as both `example.com` and `example.net`, Yandex would take the first `Host:` line as the canonical host (Source: stackoverflow.com). In practice, only Yandex recognized `Host:` at all. However, **as of March 2018 Yandex dropped support for `Host:`**, advising instead to use redirects (Source: robotstxt.ru). (Any `Host:` lines after the first are ignored (Source: stackoverflow.com).) Other search engines ignore this directive entirely.

- **Yandex "Clean-param"**: Yandex supports an unusual directive `Clean-param: p0[&p1&p2...] [path]` to canonicalize URLs by dropping irrelevant query parameters (Source: yandex.com). For example, to collapse tracking parameters, one might write:

  ```
  User-agent: Yandex
  Clean-param: ref /some_dir/get_book.pl
  ```

  This tells Yandex that URLs of the form `/some_dir/get_book.pl?ref=XYZ&book_id=123` should be treated as if only `book_id=123` is relevant, i.e. ignore all `ref=` values (Source: yandex.com). Yandex will then index only one canonical URL (`get_book.pl?book_id=123`) instead of duplicates. This directive is unique to Yandex (Google, Bing, etc. do not support `Clean-param`), and its syntax can accept multiple parameters and even path wildcards (Source: yandex.com).

- **`Noindex` and `Nofollow` (in robots.txt)**: Early on, some site owners (and even some Google discussions) considered adding `Noindex: /somepage` inside robots.txt to prevent indexing. **However**, Google has long refused to honor `Noindex` *in robots.txt*. In a 2019 blog Gary Illyes (Google) discouraged usage of any `noindex`, `nofollow`, or `crawl-delay` rules in robots.txt, stating: "we're retiring all code that handles unsupported and unpublished rules (such as `noindex`)" (Source: developers.google.com). In fact, Google explicitly says it **does not guarantee** that blocking a URL in robots.txt will keep it out of search results (Source: developers.google.com) (Source: searchengineland.com). (Pages disallowed by robots may still rank if elsewhere linked; Google may simply show a placeholder snippet.) Similarly, Bing's team noted that the "undocumented `noindex` directive never worked for Bing" (Source: www.seroundtable.com). In summary, **no major modern search engine supports a `<Noindex>` rule in robots.txt**. To remove pages from Google, one must use a `<meta name="robots" content="noindex">` in the page or send it a 404/410 response (Source: developers.google.com) (Source: www.seroundtable.com). (The only "nofollow" relevant to robots.txt is `Disallow`, which prevents the crawler from following links – but this too doesn't stop indexing if other sites link in.)
- **Other proposed extensions**: Over the years, various other directives have been suggested or adopted by niche crawlers. For example, the Conman.org draft (2000s) defined `Request-rate:` and `Visit-time:`. SeznamBot (a popular Czech search engine) implements these: e.g. `Request-rate: 10/1m` limits crawling to 10 pages per minute, possibly with additional time windows (like `1500-0559`) (Source: blog.seznam.cz). The draft's `Visit-time` could suggest preferred crawl hours. These are not recognized by Google, Bing, Yandex or most sites outside Seznam. Mat Cutts joked IBM's team could define `unicorns: allowed` lines if they wanted to extend the protocol (Source: developers.google.com). The key point is that **crawlers are free to implement proprietary directives** – the protocol is extensible. For instance, Google's own open-source parser was demoed with a `Sitemap:` handler to validate custom rule support (Source: developers.google.com). The 2025 Google blog "Future-proof REP" explicitly acknowledges such custom rules (like `clean-param`, `crawl-delay`) are outside the new RFC but still supported by some engines (though notably *not* Google Search for those specific ones) (Source: developers.google.com).

The table below summarizes common (top) and uncommon robots.txt directives, and which major search engines currently support them:

| DIRECTIVE | FUNCTION | GOOGLE | BING | YANDEX | OTHERS (NOTABLE) |
|---|---|---|---|---|---|
| `User-agent:` | Specifies target crawler (or `*` for all). | ✓ | ✓ | ✓ | – |
| `Disallow:` | Blocks specified path prefix. | ✓ | ✓ | ✓ | – |
| `Allow:` | Explicitly permits a path (overrides Disallow). | ✓ | ✓ | ✓ | – |
| `Sitemap:` | URL(s) of site XML sitemap files. | ✓ | ✓ | ✓ | – |
| `Crawl-delay:` | Seconds to wait between fetches (throttle). | No (Source: www.askapache.com) | ✓ | ✓ | (Also supported by Yandex, Archive.org, and some crawlers) (Source: www.askapache.com) (Source: yandex.com) |
| `Host:` | (Yandex only) Prefered domain among mirrors. | – | – | *Partial* (supported until Mar 2018) (Source: robotstxt.ru) | – |
| `Clean-param:` | (Yandex only) Ignore specified URL parameters. | – | – | ✓ | – |
| `Noindex:` | (If it worked) Block indexing (deprecated). | ✗ (Source: developers.google.com) | ✗ (Source: www.seroundtable.com) | ¬ support (documented) | – |
| (wildcards `*`, `$`) | Pattern matching for URLs. | ✓ (supported) | ✓ (supported) | ✓ (supported) | Implemented by Baidu, Yandex, etc (Source: www.baidu.com) |
| (Others: `Auth-group:` etc) | No common use | – | – | – | (See niche bots) |

*Table: Key `robots.txt` directives and support by major search engines. "✓" indicates support. Blank "–" means no support. Yandex historically accepted `Host:` and `Clean-param:`; Google/Bing do not. Both Google and Bing ignore any `Noindex:` in robots.txt` (Source: developers.google.com) (Source: www.seroundtable.com). (Sources: Google, Yandex official docs, community knowledge.)*

## Search Engine–Specific Behaviors

Different crawlers interpret robots rules slightly differently. This section highlights key search engines' behaviors (Google, Bing, Yandex, Baidu, etc.) and how they treat robots.txt.

- **Google (and Googlebot)**: Googlebot fully follows the REP "standard" portion. It recognizes `User-agent`, `Disallow`, `Allow`, `Sitemap` and wildcards. Google **does not** implement `Crawl-delay` or `Request-rate`; instead, it uses centralized crawl controls. Google also ignores any unsupported lines like `Noindex:` in robots.txt (Source: developers.google.com). Importantly, Google will still index (without content) URLs that are disallowed. As one SEO guide notes, *"No URL is entirely blocked from search engines if you disallow it in robots.txt"* (Source: searchengineland.com). Google's documentation likewise says it "does not guarantee" disallowed pages won't end up indexed. In practice, Google may show a results entry for a disallowed URL (often labeled "Unverified" or with no snippet) if it finds links to it (Source: searchengineland.com). Newer Google features even allow disallowed pages to be cited in AI overviews with snippets (Source: searchengineland.com). After 2019, Google formally disabled parsing of `noindex` in robots (and any unpublished rules like `nofollow`) (Source: developers.google.com), aligning with Bing's stance (Source: www.seroundtable.com). In 2019 Google open-sourced its robots.txt parsing code and published an Internet-Draft (pre-RFC proposal) showing how new rules could be added (Source: developers.google.com). Google's official blog ("Future-proof REP") notes that in 25+ years, the only universally adopted change was adding `Allow` (Source: developers.google.com); other extensions (like `sitemap:`) became common outside the RFC.

- **Bing and Yahoo**: Since Yahoo Search now uses Bing's crawler ("Bingbot"), their robots usage is identical. Bing supports `User-agent`, `Disallow`, `Allow`, `Sitemap` and (unofficially) `Crawl-delay`. Bing *requires* that if you specify a named section for `Bingbot:`, you must repeat any general rules there. As SearchLand reported, "If you create a section for Bingbot specifically, all the default directives will be ignored… You *must* copy-paste the directives you want Bingbot to follow under its own section" (Source: searchengineland.com). Bing's senior dev Frédéric Dubut confirmed it never recognised `noindex` in robots.txt, so pages must use meta tags or headers to remove from Bing's index (Source: www.seroundtable.com). Otherwise, Bing's behavior is similar to Google's: disallowed pages may still index if linked, and Bing reserves its own caching controls in Webmaster Tools.

- **Yandex**: Yandexbot honours the standard REP plus its proprietary extensions. Its documentation lists `Allow`, `Disallow`, `Crawl-delay`, plus `Sitemap` and `Clean-param` (Source: yandex.com). Yandex uses `Clean-param:` to optimize crawling of dynamic URLs (see example above (Source: yandex.com). Its `Crawl-delay` is expressed as seconds (e.g. `Crawl-delay: 10`) (Source: yandex.com). Up until 2018, Yandex also read a `Host:` directive for the canonical domain, but that is now discontinued (Source: robotstxt.ru). Notably, Yandex treats disallowed pages similarly to Google: they can still be indexed, but Yandex cannot see their content and thus cannot respect any HTML `noindex` on them. (Yandex therefore warns webmasters to use meta `noindex` instead of robots to hide content (Source: yandex.com).) Like Google, Yandex considers robots rules case-sensitive.

- **Baidu (China's leading search)**: Baidu's bots (e.g. "Baiduspider") support `User-agent`, `Disallow`, `Allow`, and `Sitemap`. Baidu explicitly supports wildcard patterns `*` and end-of-line `$` (Source: www.baidu.com) (its own documentation states "Baiduspider supports wildcard characters `*` and `$`"). Baidu does **not** have a public `Crawl-delay` argument in robots.txt; instead, Chinese webmasters adjust crawl rate via Baidu Webmaster Tools. Baidu also notes that pages blocked by robots may still appear in search results via links from other sites (Source: www.baidu.com); so again, `Disallow` is a directive to control crawling, *not* a foolproof no-index method. In practice, `<User-agent: Baiduspider>` appears in roughly 1.9% of sites (per Crawling Stats (Source: almanac.httparchive.org).

- **Other crawlers**: There are countless lesser bots (Majestic's `mj12bot`, Ahrefs, etc.) that simply obey REP like Google. The 2021 HTTP Archive SEO report noted that the most common specific user-agents encountered (after Google, Bing, Baidu, Yandex) included Majestic (`mj12bot`, 3.3% desktop) and Ahrefs (`ahrefsbot`, 3.3% desktop) (Source: almanac.httparchive.org). None of these bots introduce unique new directives beyond what the major engines do.

## Data Trends and Statistics

To understand real-world usage of `robots.txt`, we can draw on web crawl data. According to the 2021 HTTP Archive SEO chapter (Source: almanac.httparchive.org), **81.9% of websites use a `robots.txt` file on their main domain** (a slight increase from ~72% in 2019). Conversely, about 16.5% of sites have *no* robots.txt, in which case Google treats *all* pages as crawlable (Source: almanac.httparchive.org). The remaining ~1.6% either returned errors or were not reachable. Importantly, if a `robots.txt` fetch fails with HTTP 5xx (server error), Google's policy (per RFC 9309) is to treat the site as "unreachable" and temporarily suspend crawling (Source: www.rfc-editor.org) (Source: almanac.httparchive.org). If it fails with a 4xx or 403, Google may treat the file as "unavailable" and default to allowing crawling (Source: www.rfc-editor.org). In practice, the Archive found ~0.3% of sites returned 403/5xx for robots.txt, and Google's team estimated up to 5% had transient 5xx and 26% were unreachable at times (Source: almanac.httparchive.org). Even temporary issues with robots.txt can halt a crawler: in one survey Google said it will stop crawling a site for a while if its robots.txt returns errors, since it "is unsure if a given page can or cannot be crawled" (Source: almanac.httparchive.org).

Regarding file size, most `robots.txt` are quite small (<100 KiB). The HTTP Archive analysis shows only ~3,000 domains exceeded 500 KiB – Google's documented maximum – meaning on those extra-large files any rules beyond 500 KiB would simply be ignored (Source: almanac.httparchive.org). Besides size, there are also file encoding considerations (RFC 9309 requires UTF-8) and a 500 KB parser limit to avoid overload (Source: yandex.com) (Source: www.rfc-editor.org). Very large or malformed files thus risk not being parsed fully.

Another useful statistic: how often specific user-agents are mentioned. Figure 8.6 of the Web Almanac shows that “Googlebot” appears in about 3.3–3.4% of robots.txt rules, Bingbot in ~2.5–3.4%, Baiduspider ~1.9%, Yandexbot ~0.5% (Source: almanac.httparchive.org). (These percentages are of all robots.txt files crawled.) That indicates Google and Bing are explicitly targeted by a few percent of sites, whereas Majestic and Ahrefs also appear (~3% each). This echoes the practice of SEO tools placing their own crawl instructions.

Finally, usage of extended directives is relatively rare on the web. For example, contrast Yandex’s `Clean-param` with broad usage: virtually **100%** of Yandex-directed robots rules use it when present, but it appears on only a few percent of all sites globally (since only sites indexed by Yandex would use it at all). Similarly, very few sites list `Host:` now (since Yandex dropped it) or Seznam’s `Request-rate`. This report has focused on comprehensiveness rather than prevalence, so we cover even these rarer cases fully.

## Case Studies and Real-World Examples

**SEO Mishaps**: A classic illustration of robots.txt impact is Glenn Gabe’s case study on Search Engine Land (Source: searchengineland.com). A client realized key category pages were mysteriously disappearing from Google. Upon investigation, Gabe found two culprits: (1) the CMS provider had been programmatically adding new `robots.txt` directives over time without the site owner’s knowledge, and (2) some disallows used the wrong case (e.g. `/CATEGORY/` instead of `/Category/`). Since robots.txt matching is case-sensitive, those directives accidentally blocked pages. The result was a “slow leak” of important URLs from Google’s index (Source: searchengineland.com). Gabe’s analysis highlights the danger of even small robots changes. It underscores that webmasters should monitor robots.txt edits (some use alerts or version control) and routinely audit which important URLs might be being blocked (tools like Screaming Frog or the Search Console robots tester can help) (Source: searchengineland.com) (Source: searchengineland.com). Using the Internet Archive’s Wayback Machine to check historical robots.txt versions can also pinpoint when a harmful directive was added (Source: searchengineland.com).

**Security Risk/Honeypots**: Beyond SEO, robots.txt files have drawn the attention of security researchers and even hackers. A penetration tester in 2015 analyzed hundreds of thousands of `robots.txt` across the web and found they often expose “treasure maps” to attackers (Source: www.theregister.com). If a robots file disallows directories like `/admin/`, `/staging/`, or `/backup/`, it essentially **announces** these sensitive areas exist. For example, Weksteen (a security researcher) reported finding many admin and login portals simply by scraping disallowed paths in robots.txt (Source: www.theregister.com). His findings include real cases: thousands of government and academic sites had “/disallow” entries pointing to confidential PDF archives and personnel data, which attackers later accessed via search. As The Register summarises, *“mention of a directory in a robots.txt file screams out that the owner has something they want to hide”* (Source: www.theregister.com). Even well-known sites are not immune: he cites cases where names of trafficked victim dossiers were inadvertently exposed via image descriptions in disallow lists.

Similarly, security experts widely advise: **do not rely on robots.txt to protect secret content**. Ethical hacking guides emphasize that publishing sensitive file or directory names in robots.txt is counterproductive; it creates an “unintentional attack surface” (Source: nemocyberworld.github.io). In fact, some admins set up honeypots by listing fake, enticing disallowed paths (e.g. `/admin/please_dont_hack/`) and then monitoring any hits to those paths. The bottom line is robots.txt is public: every human and malicious bot can read it. A section restricting a path means that path exists and is important; attackers will probe accordingly (Source: nemocyberworld.github.io) (Source: www.theregister.com).

**Blocking vs Indexing**: Another practical concern involves the different behaviors of “blocked” vs “indexed”. Search Console introduced new status messages like “Indexed, though blocked by robots.txt,” which confuse many SEOs. As a February 2025 SearchEngineLand article explains (Source: searchengineland.com), “Blocked by robots.txt” does *not* mean “will never appear in search results.” Google explicitly states that a disallowed page *may still be indexed* (often using its URL and external link text), though Googlebot won’t fetch its contents (Source: searchengineland.com). In fact, pages can even show up in special features; Lily Ray observed a Goodreads URL listed in Google’s AI overviews despite being blocked by robots (Source: searchengineland.com). The community consensus is summed up: **“No URL is entirely blocked from search engines if you disallow it in robots.txt”** (Source: searchengineland.com). Fixing an accidental disallow usually involves removing the directive and re-requesting indexing via tools (or just waiting for recrawl) (Source: searchengineland.com).

**Case: Large Crawling Operations**: Public projects like the Internet Archive rely on robots.txt to respect site exclusions. The Archive’s crawlers parse `robots.txt` and obey `Disallow` (as do pretty much all good bots). However, different organizations have chosen to interpret some status codes differently. For example, the Internet Archive’s engineering notes indicate that by default a missing `robots.txt` is treated as “allow all”, whereas a certain pattern of redirects or 401/403 might be treated as “full allow” (i.e., indexable) (Source: www.rfc-editor.org). Google, on the other hand, treats 401/403 differently (it considers them parseable as “allow everything”) (Source: www.rfc-editor.org). Such nuances imply that crawling outcomes can vary slightly between institutions.

## Analysis and Discussion

**Depth of Extensions vs. Practical Use**. In the 25+ years of robots existence, very few new rules have achieved the universal adoption of the core directives. Google engineers note that aside from `Allow`, the only other “extension” nearly all major bots understand is `Sitemap:` (Source: developers.google.com). All other features remain either engine-specific or optional. For instance, both Google and Yandex quietly ignore any

`noindex` or `nofollow` directives placed in robots.txt (Source: developers.google.com) (Source: www.seroundtable.com) – such lines simply have no effect. Similarly, while `Crawl-delay` is widely recognized by Bing and Yandex, Google intentionally chooses not to support it (Source: www.askapache.com).

Some extensions proposed in the past still live on in niche use. Seznam's "Request-rate" and "Visit-time" show that planning complex bot schedules is possible if both crawler and webmaster agree. Google's robotics team encourages community input: the "Future-proof REP" article explicitly invites webmasters to propose new directives (with consensus) via open channels (Source: developers.google.com) (Source: developers.google.com). It reminds us that robots.txt's simplicity and ubiquity make it a candidate for new rules, but only if widely beneficial. The history of adopting `sitemap:` as a rule (once crowdsourced by SEOs and search engines) is held up as a model (Source: developers.google.com). Conversely, they caution that unilateral changes will not become standard – collaboration is needed.

**Implications for Webmasters and SEO**. For practitioners, the "secrets" of robots.txt are mostly about understanding how each bot behaves and testing properly. Always assume Google (and Bing) will ignore any private or hidden meaning in your robots file. Never put real passwords, keys, or highly secret endpoints in `robots.txt`. Use it only to cut off low-value crawl paths (duplicate pages, staging/testing areas, etc.), not to hide content. Test your rules in tools: Google Search Console's Robots Tester (if verified), and third-party validators, to ensure syntax is correct. Monitor your robots.txt usage changes (the wayback machine or alerts) – as the Gabe case shows, unexpected changes can quietly wreck SEO. Keep the file lean to avoid size limits; compress multiple disallows into one path spec when possible.

**Future Outlook**. With REP now officially standardized, most of the "protocol gaps" are known. Crawlers show interest in evolving robots.txt (e.g. the IETF draft, open-source parsers), but any change will be slow given the need for backward compatibility and broad support. Google's 2025 perspective is that robots.txt may carry *new crawling preferences*, but only through careful community consensus (Source: developers.google.com). As AI and new search modalities emerge, controlling what a bot can see remains crucial (robots.txt is the first line of communication). Yet ironically, the specifications emphasize that sensitive control should move to more secure mechanisms (e.g. meta tags, server configs) (Source: www.rfc-editor.org). The REP will likely remain an important, if limited, piece of the indexing ecosystem. Hints of the future include better parsing (Google open-sourced its parser (Source: developers.google.com) and potentially new flexible directives – but for now, webmasters should master the existing ones, knowing there are "no other secrets about robots.txt" beyond these rules (Source: www.askapache.com).

## Tables

| SEARCH ENGINE / BOT | SUPPORTS `ALLOW` | SUPPORTS WILDCARDS (`*`, `$`) | SUPPORTS `CRAWL-DELAY` | SUPPORTS `CLEAN-PARAM` | SUPPORTS `HOST` | NOTES ON `NOINDEX` |
|---|---|---|---|---|---|---|
| **Googlebot** | ✓ (Source: visual-seo.com) | ✓ (Source: www.baidu.com) | **No** (Source: www.askapache.com) | No | No | Ignores `noindex` in robots (Source: developers.google.com) (use meta instead) |
| **Bingbot/Yahoo** | ✓ | ✓ | ✓ | No | No | Never supported `noindex` (Source: www.seroundtable.com) |
| **Baiduspider** | ✓ | ✓ (Source: www.baidu.com) | (None) | No | No | Uses robots block for crawl only (can still index via links) (Source: www.baidu.com) |
| **Yandexbot** | ✓ (Source: yandex.com) | ✓ (implied) | ✓ (Source: yandex.com) | ✓ (Source: yandex.com) | (Legacy) | Disallowed pages may index; encourages meta `noindex` for removal (Source: yandex.com) |
| **Other (e.g. Majestic/Ahrefs)** | ✓ | ✓ | *Varies* | No | No | Follows standard REP parsing |

*Table: Feature support of major search engine crawlers. Checkmarks from official docs and blogs above. Google and Bing do **not** recognize a `noindex` directive in robots.txt (Source: developers.google.com) (Source: www.seroundtable.com) (use meta or HTTP headers instead).*

| USER-AGENT IN `ROBOTS.TXT` | FREQUENCY IN CRAWL STUDIES | TYPICAL USE CASE |
|---|---|---|
| `User-agent: *` (all bots) | ~100% (all sites with robots.txt) | Default rules for all crawlers |
| `Googlebot` or `Googlebot-News` | 3.3–3.4% (Source: almanac.httparchive.org) | Explicit rules for Google's crawler |
| `Bingbot` or `Slurp` | 2.5–3.4% (Source: almanac.httparchive.org) | Explicit rules for Bing/Yahoo crawlers |
| `Baiduspider` | ~1.9% (Source: almanac.httparchive.org) | Rules specifically for Baidu (Chinese search) |
| `Yandexbot` | ~0.5% (Source: almanac.httparchive.org) | Rules specifically for Yandex (Russian search) |
| `MJ12bot`, `AhrefsBot`, etc. | ~3–4% each (by Alt SEO tools) | Targeted by SEO tools to guide their crawlers |

*Table: Breakdown of how often specific bots are named in robots.txt (Desktop vs. Mobile usage is similar) (Source: almanac.httparchive.org). Aside from `*` (used in all files), Google and Bing bots are explicitly referenced by only a few percent of sites. Notably, some SEO tool bots (Majestic, Ahrefs) appear as frequently as for large search engines.*

## Conclusion

The `robots.txt` file may look trivial, but it carries many subtleties. Our survey shows that besides the well-known `User-agent` / `Disallow` rules, there is a rich landscape of directives with varying adoption. Some "secrets" of `robots.txt` have to do with *absence*: for example, knowing that Google will not respect `Crawl-delay` or `Noindex` in this file (Source: www.askapache.com) (Source: developers.google.com). Other nuances involve interplay and edge cases: disallowed URLs can still be partially indexed (Source: searchengineland.com), or robots.txt fetch failures can **freeze** crawling until fixed (Source: www.rfc-editor.org) (Source: almanac.httparchive.org). We also uncovered lesser-known tools like Yandex's `Clean-param` for query-string merging (Source: yandex.com) and Seznam's rate-limit rules (Source: blog.seznam.cz). Each has its place in specific ecosystems.

The historical arc is that robots.txt has proven remarkably durable and extensible. It has seen only minor modifications in 30 years (e.g. the addition of `Allow` and wildcard support) (Source: developers.google.com). The 2022 RFC officially froze much of its syntax, though it allows new records via "other records" (as with `Sitemap:`) (Source: www.rfc-editor.org). Going forward, changes will come very slowly, if at all. Google encourages community-driven ideas (the example of an extant `sitemap:` rule shows how consensus can drive adoption (Source: developers.google.com). But for now webmasters must be experts in the current rules and interpretations: misuse can harm SEO or security.

**Recommendations**: Webmasters should keep `robots.txt` simple and well-tested. Only list what is truly non-public or low priority (e.g. login pages, duplicate search paths). Always double-check syntax (use tools or Search Console tester) and remember that the file itself is public. Consult official documentation for each search engine when in doubt: for Google, see Google Search Central; for Yandex, see Yandex.Webmaster guidelines (Source: yandex.com); for Baidu or Bing, their help centers. When troubleshooting indexing issues, always verify you didn't accidentally disallow needed URLs (common tools include Google Search Console's robots report (Source: searchengineland.com) and archival version history (Source: searchengineland.com).

In summary, `robots.txt` remains a critical, if behind-the-scenes, tool for crawling control. Understanding its full range—down to the "secrets" of hidden or unique parameters—empowers site owners to manage their web presence better. All claims and recommendations above are backed by authoritative sources and real-world examples. Use this report as a reference checklist to ensure your `robots.txt` is both correct and secure, and to stay abreast of any future developments in the Robots Exclusion Protocol (Source: developers.google.com) (Source: www.rfc-editor.org).