

Bots/Crawlers & Control/Mitigation

[Robots.txt?](#)

Robots.txt is a file that can be used on a server to decide which bots/crawlers can access the site as well as how often they can make requests. It's important to know that this file is essentially a guideline published to and recognised by bots/crawlers, outlined in this file are the 'rules' for crawling that domain - this file will NOT forcibly block bots/crawlers from making requests.

Below is an annotated example of how the robots.txt file could be configured:

```
# The below line defines the 'site map' for bots to reference. This is essentially a map of a
domain, useful for boosting SEO also.
Sitemap: https://www.example.com/sitemap.xml #specifies the 'site map' for bots to discover
all pages on a domain

# The below block specifies that all user-agents can crawl all pages, except from /login/.
User-Agent: *
Disallow: /login/

# This below line allows all search engines to crawl all pages except /login/
User-Agent: *
Disallow: /login/

# Alternatively, you can specify a specific search engine by user agent
User-Agent: Googlebot
Disallow: /admin/

#For sites with extreme crawling rates, it may be worth adding a 'crawl-delay' directive, as
below:
User-Agent: Googlebot
crawl-delay: 10
```

=====

=====

Identifying Bot Traffic

There are various sorts of server setups you'll encounter while attempting to decipher traffic logs. The below outlines the general methodology I use, however, these commands and approaches will likely need to be adjusted depending on your use-case.

Counting bot requests:

```
grep -iE 'bot|crawler' /path/to/access/log | wc -l
```

Identification of user-agents

```
awk -F'"' '{print $6}' /path/to/access/log | sort | uniq -c | sort -nr | head -20 | grep -iE 'bot|crawler'
```

=====

=====

Bad Bots

Not all bots listen to rules defined within the robots.txt file, in cases where required, a server-wide block on 'bad bots' can be implemented.

cPanel

WHM > Apache Configuration > Includes Editor > Pre-main Includes > All Versions

```
<Directory "/home">
SetEnvIfNoCase User-Agent "petalbot" bad_bots
SetEnvIfNoCase User-Agent "baidu" bad_bots
SetEnvIfNoCase User-Agent "360Spider" bad_bots
SetEnvIfNoCase User-Agent "dotbot" bad_bots
SetEnvIfNoCase User-Agent "megaindex\.ru" bad_bots
SetEnvIfNoCase User-Agent "yelpspider" bad_bots
SetEnvIfNoCase User-Agent "fatbot" bad_bots
SetEnvIfNoCase User-Agent "ascribebot" bad_bots
SetEnvIfNoCase User-Agent "ia_archiver" bad_bots
SetEnvIfNoCase User-Agent "moatbot" bad_bots
```

```
SetEnvIfNoCase User-Agent "mj12bot" bad_bots
SetEnvIfNoCase User-Agent "mixrankbot" bad_bots
SetEnvIfNoCase User-Agent "orangebot" bad_bots
SetEnvIfNoCase User-Agent "yoozbot" bad_bots
SetEnvIfNoCase User-Agent "paperlibot" bad_bots
SetEnvIfNoCase User-Agent "showyoubot" bad_bots
SetEnvIfNoCase User-Agent "grapeshot" bad_bots
SetEnvIfNoCase User-Agent "WeSee" bad_bots
SetEnvIfNoCase User-Agent "haosouspider" bad_bots
SetEnvIfNoCase User-Agent "spider" bad_bots
SetEnvIfNoCase User-Agent "lexxebot" bad_bots
SetEnvIfNoCase User-Agent "nutch" bad_bots
SetEnvIfNoCase User-Agent "sogou" bad_bots
SetEnvIfNoCase User-Agent "Cincraw" bad_bots
<RequireAll>
Require all granted
Require not env bad_bots
</RequireAll>
</Directory>
```

Plesk

Apache

Nginx

Alternative bot traffic control measures

In cases where bot traffic continues to rampage requests to a domain/s, it's worth advising the use of CloudFlares 'Bot Fight Mode'. Other CDN providers may well have their own offerings - worth advising the client of if they utilise an alternative network.

Revision #8

Created 2023-08-03 12:46:06 UTC by Daniel

Updated 2024-06-04 23:57:42 UTC by Daniel