Web Scraping Protection

https://campus.barracuda.com/doc/48660515/

Web scraping involves copying large amounts of data from a website or application using automated tools, often for commercial advantages that are to the detriment of the organization that owns the web application. Typically, the motivation of the attacker is to undercut competition, steal leads, hijack marketing campaigns, and appropriate data via the web application. Examples include theft of intellectual property from digital publishers, scraping products and pricing information from e-commerce sites, and stealing listings on real estate, auto dealers, and travel sites.

There are a variety of automated tools, products, and services available for web scraping that can extract data and metadata from the web applications as well as from web-based APIs. Advanced tools can even automatically navigate to pages behind forms by automatically filling them in.

Their navigation and extraction features makes scrapers very similar to search engines that also intend to index the whole site. Unlike search engines that drive prospects to businesses, scrapers intend to take away business from the sites they are scraping. This makes it important for a security solutions to be able to distinguish between genuine search engines and web scrapers, even when some scrapers fake their identity as search engines.

To prevent your web applications from being scraped, configure the web scraping policy on the Barracuda Web Application Firewall.

Configuring Web Scraping Policy

The web scraping policy provides the following settings:

Allowed Bots

Create a list of search engine bots that you want to allow access to your web application by providing the **User Agent** and **Host** value pair. For example: **User Agent**: googlebot and **Host**: *.google.com.

When a client identifies itself as a search engine via the **User Agent** field, the system performs a reverse DNS lookup (rDNS) on the source IP address, which yields the true domain associated with the IP address. If this domain does not match the **Host** value configured above, then the client is classified as a fake bot and web scraping policies are applied on the request. If the configured Host value matches the rDNS domain value, then the request is exempted from further web scraping validation.

Honey Traps

To trap the web scraping tools, configure the following:

Insert Hidden Links in Response

When enabled, the Barracuda Web Application Firewall embeds a hidden link in the response. The embedded link does not get displayed on the browser, so a human browsing the web pages through a common browser should never see and click the hidden link. Therefore, any request that attempts to access the hidden link is identified as an automated bot or scraper.

> This feature requires the response to be formatted as an HTML document with opening and closing <html> </html> tags. Also, hidden links require **Insert Javascript in Response** to be enabled in the web scraping policy.

Insert Disallowed URLs in Robots.txt

Typically, every website includes a "/robots.txt" file that provides access instructions such as the user agents that are allowed to access the site, and the web pages that are allowed/disallowed to be accessed by bots.

Example:

User-agent: *

Disallow: /researchtools/abc/

Here, ***User-agent*** : Asterisk (*) is a wildcard character and indicates that this website can be accessed by all bots, and ***Disallow*** : /researchtools/abc/ indicates that the bots are not allowed to access the /researchtools/abc/ page on the website.

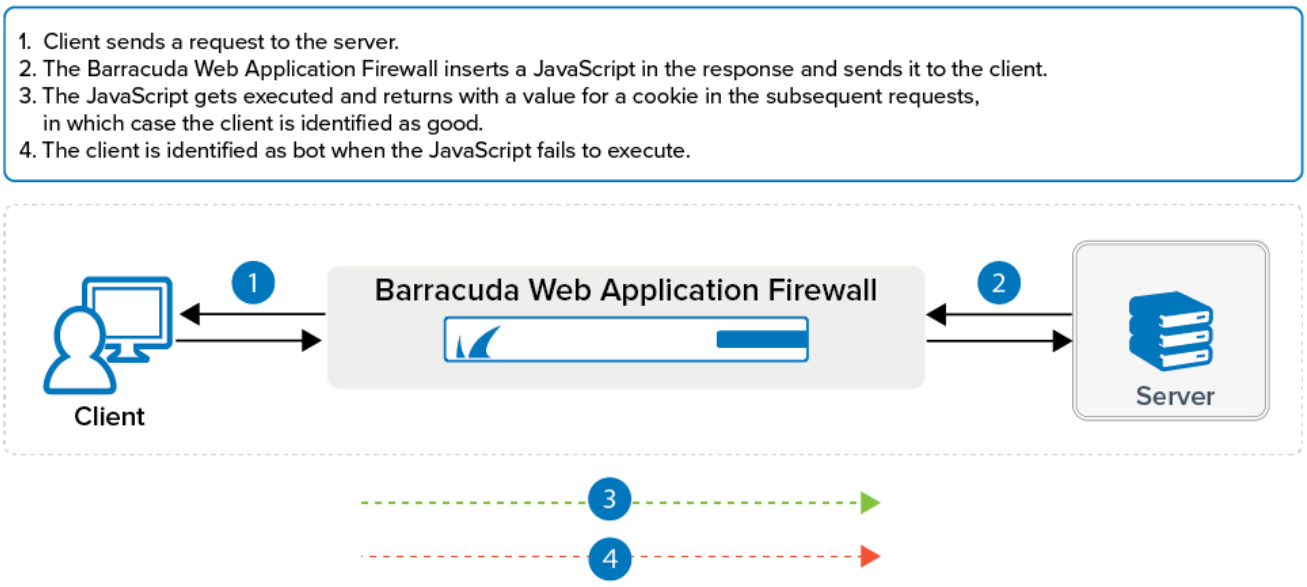
When **Insert Disallowed URLs in Robots.txt** is set to *Yes*, the Barracuda Web Application Firewall inserts an encrypted URL into the robots.txt file under **Disallow**. Any bot that tries to access the encrypted URL is identified as a bad bot, and the corresponding action is taken as configured on the **SECURITY POLICIES** > **Action Policy** page.

Bot Detection

To detect bad bots, configure the following:

Insert JavaScript in Response

When enabled, the Barracuda Web Application Firewall inserts a JavaScript in the response. If the request is from a client (web browser), the JavaScript gets executed and returns with a value for the cookie. If the JavaScript fails to execute, then the client is marked as a bot after the specified **JavaScript Failure Threshold** value. The **JavaScript Failure Threshold** can be configured under **Advanced** in the **ADVANCED > System Configuration** page. The default action configured for **Web Scraping Bots** in the **SECURITY POLICIES > Action Policy** page is to challenge the client with the CAPTCHA image. The client will not be allowed to access any further resource until the CAPTCHA is answered. You can modify the **Follow Up Action** (if required) for the **Web Scraping Bots** attack in the **SECURITY POLICIES > Action Policy** page.



Insert Delay in Robots.txt

You can slow down the requests from a bot to a web application by setting the delay time (in seconds) between subsequent requests, so that server resources are not consumed and are accessible for legitimate traffic.

When **Insert Delay in Robots.txt** is set to *Yes*, the Barracuda Web Application Firewall automatically inserts “crawl-delay” in the robots.txt file with the specified **Delay Time**. All good bots should honor the delay time specified in the robots.txt file while accessing the web application. If not, it is identified as a bad bot and the corresponding action is taken as configured on the **SECURITY**

POLICIES > **Action Policy** page.

Blocked Categories

The Barracuda Web Application Firewall is integrated with an external database that allows you to classify clients based on their IP addresses and user agents. For a web scraping policy, when one or more block-listed category is selected from the available list, all traffic matching the binding Bot Mitigation policy is validated against this external database. If the originating traffic is reported to be from any of the selected category, the request is blocked. Also, Web Firewall Logs corresponding to such attacks provide more information on the category the request matched.

> Most of the blocked categories are partially or completely powered by the Barracuda Active Threat Intelligence engine. The advanced intelligence engine is only available with the Advanced Bot Protection (ABP) license. For the list of bot categories, see the Bot Categories article.

Continue with Enforcing a Web Scraping Policy.

Figures