Sector Skills – Engineering

Jaden Toon

GitHub Link - https://github.com/jadentoon/ieuk-task-2025

Disclaimer – To run the code, you will need Python installed with the Pandas module installed through pip.

Findings:

Analysis of the server logs revealed that the top two IP addresses by request count, '45.133.1.1' and '45.133.1.2', each submitted 5,400 requests within a single hour (15:00–15:59). This volume is highly atypical of normal human behavior, strongly indicating bot activity.

Further, their associated user agents—curl/7.68.0, sqlmap/1.6.12, and python-requests/2.28.1—are commonly used in automated scripts, not standard browsers. This supports the conclusion that these IPs are bots likely using tools such as Python's requests module, sqlmap, or similar.

Geographically, the top three nations by request volume are the UK, US, and Sweden (SE). However, the top nations by error code counts (status codes 400–599) are Russia (RU), UK, and Canada (CA). The Russian IPs '45.133.1.1' and '45.133.1.2' account for most of RU's traffic, reinforcing their bot-like nature.

Assumptions:

The analysis shows that requests that are non-standard user-agents such as 'curl' and 'python-requests' are generated by automated scripts rather than legitimate users. Whereas logs with more legitimate user-agents such as 'Mozilla/5.0' suggest that these are legitimate users, but spoofing is possible, for example, in Python, being able to add a user agent with Mozilla/5.0 to pretend to be a real user.

In addition, IP addresses trying access API endpoints such as '/api/v5/users/admin', are likely to indicate bot activity as a legitimate user would be accessing front-end parts of the website such as '/contact' and '/about', whereas bots would skip the front-end entirely and try to access the back end like the API directly. These sort of requests (especially if they are frequent) should be blocked for potential abuse.

Proposed Solutions:

One solution is implementing IP-based rate limiting, for example, allowing a max of 100 requests per hour so that the service is bombarded with requests. This can be done by using tools like NGINX rate limiting module or Cloudflare's free tier which has built in rate limiting so requests coming from an IP address like '45.133.1.1' where they sent 5400 requests in an hour would be blocked.

Another solution is blocking known bad user agents, for example, 'curl/', 'sqlmap/' and 'python-requests/', which are likely to be bots and are non-browser agents that real users would never

use. So, we could deny access at the web server level, returning a 403 or 404 error for any request that have these dodgy user agents. Then, we can log them separately for auditing purposes.