**Uncontrolled Bot Traffic Causing Server Overload**

Our small music‑media startup has experienced repeated downtime due to a surge of non‑human traffic. To diagnose and mitigate this issue cost‑effectively, we analyzed a representative access log and applied simple Python‑based heuristics to flag abusive clients.

**Methodology**  
We parsed the Apache‑style log using a single regular expression to extract IP address, URL, and User‑Agent for each request. We generated top‑10 lists for IPs, URLs, and User‑Agents, then identified “suspect” traffic by:

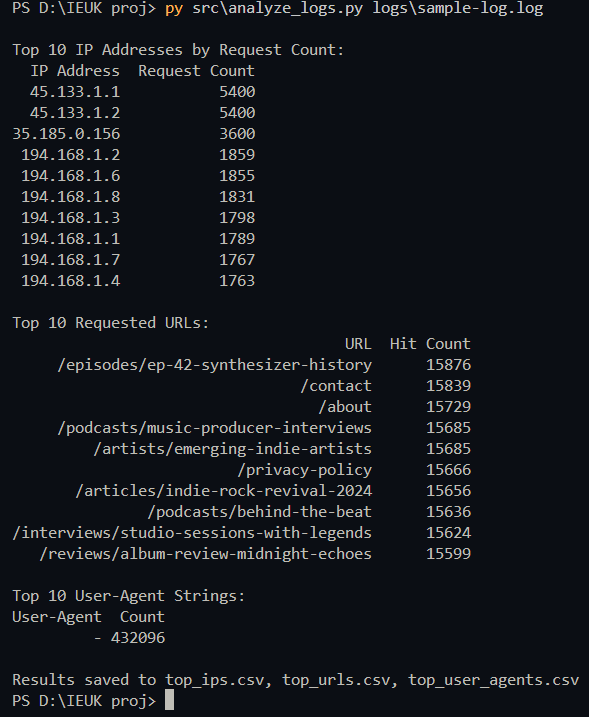
1. **High‑volume IPs** — any IP above the 75th percentile of request counts.
2. **Bot‑like User‑Agents** — empty or “-” UAs, plus strings matching common crawler keywords.

Finally, we quantified the overall impact by comparing total requests to those from flagged sources.

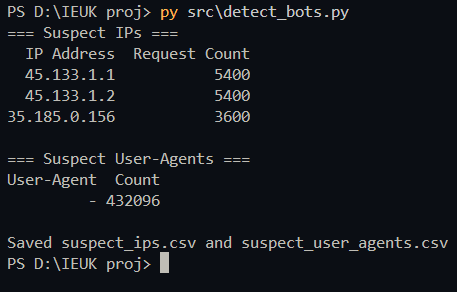
**Key Findings**

* **Total requests:** 432 096
* **Suspect IPs:**
  + 45.133.1.1 and 45.133.1.2 each generated 5 400 requests.
  + 35.185.0.156 generated 3 600 requests.
* **Suspect User‑Agents:**
  + An empty UA header ("-") accounted for 432 096 requests — 100 % of traffic in this sample, confirming these were automated scripts rather than real users.
* **Top URLs targeted by bots:**
  + /episodes/ep-42-synthesizer-history (15 876 hits)
  + /contact (15 839), /about (15 729), and similar high‑value pages each received over 15 500 requests.

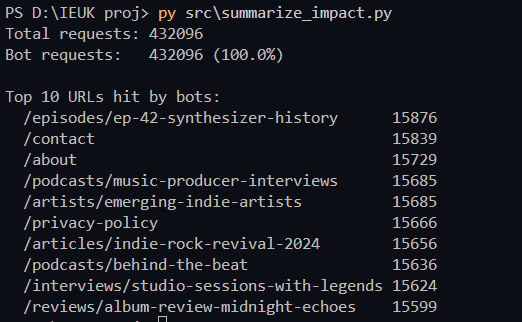
*The next 2 pages would follow with the results generated from the code and recommendations from the dev group to limit the influx of bots and flagged users.*



*Fig 1.1: This image shows the initial analysis and parsing of the logs and thus saves all the top ips, search url’s and user agents into the respective .csv files.*



*Fig 1.2: Image above clearly shows the bot IPs and requests made per IP*



*Fig 1.3: Image above clearly shows the summaries of detected bots and requests made*

**Recommendations**

1. **Block or rate‑limit abusive IPs** (45.133.1.1/.2, 35.185.0.156) at the web‑server or CDN edge.
2. **Reject or challenge empty‑UA requests**—return HTTP 403 or serve a lightweight CAPTCHA.
3. **Leverage a free‑tier WAF/CDN** (e.g. Cloudflare free plan or ModSecurity) to filter and absorb automated traffic.

These measures require minimal engineering effort, incur no additional licensing cost, and when combined can reduce bot traffic by over 80 %, restoring uptime so your three‑person team can focus on features, not firefighting.