***Log Analysis Report on Music Start-Up Website***

**Scenario**

**You've been hired by a small music media start-up. Their successful podcast and newsletter have led to a spike in traffic, initially bringing a welcome increase in subscriptions. However, concerns are growing that some traffic may be non-human, and the servers are becoming overwhelmed. Parts of the website go down every few days due to the sheer volume of traffic. With an engineering team of just three people, this downtime is severely impacting their productivity.**

**Your task is to solve this issue: Using a provided set of logs you must identify the problem and determine the best way to handle the increased traffic.**

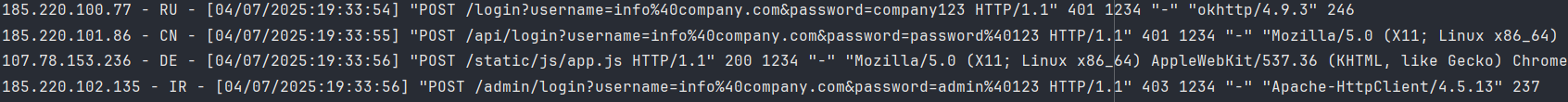
**Identifying the Problem: Traffic Description & Findings**

A rapid traffic spike is stated, successful for the music start-up yet suspicious with concerns that most of the traffic are bots, potentially malicious. The primary problem is highlighted – the website frequently suffers with downtime due to overwhelmed servers from mentioned traffic requests. To add on, the engineering team managing the website is small therefore this issue must be fixed immediately with cheap effectiveness. This documentation covers discovered patterns, through analysing given sample logs and reveals main causes of frequent website downtime along with useful recommendations to prevent or mitigate future issues. Starting with viewing a small sample, specifically the period when the server crashed, key components were split and individually applied through filtration. For example, the IPs were filtered to catch specifics with abnormal behaviour. Some private IP address ranges and even public IP addresses like 192.168.\*.\* and 172.\*.\*.\* are seen modifying valuable backend data such as the news URL and APIs. These private IP addresses could most likely be spoofed or bots, hailing from multiple nations shown by different country codes, attempting brute force which can lead to a distributed denial of service attack. Other concerns include:

* Multiple requests within a short time duration, unusual for a small website. 50+ requests in 2 minutes from [04/07/2025:18:51:37] to [04/07/2025:18:53:37].
* HTTP method PUT applied to access and modify backend data.



* Several attempts to login to admin accounts. HTTP method POST frequently visible, possibly attempting brute force on login page.



A cost-effective solution must deal with the frequency of bots overloading the server and their suspicious behaviour towards backend data, specifically on the APIs and login endpoints. Below, there are common recommendation tailored for the small website.

**Recommendations**

Assumed constraints include limited finances and technical skillset from the tiny management software engineering team to implement long-term improvements to handle frequent traffic however immediate action can be taken. Apply IP address rate limiting to reduce requests per minute to shrink load, enabling servers’ space along with blocking constant IP addresses spamming requests or modifying backend data using PUT method as a safety precaution. To mitigate bot presence, authentication should be applied such as CAPTCHA to the login page and automatic IP address blacklisting if it fails to authenticate a specific number of times.

As the engineering team scales, servers can be scaled to improve website performance while implementing long-term monitoring tools to view logs in real-time against bots.