Engineer Mjukvara

Byggde hastighetsbegränsning som stänger vid fel

Byggde skydd mot missbruk som stänger vid fel — 22 hastighetsbegränsare i Redis, Cloudflare Turnstile, idempotenta anrop och ett ursprungslås — så att plattformen avvisar botar och anropsfloder i stället för att lita på sina anropare.

2025

22

hastighetsbegränsare som stänger vid fel

Situation. En publik katalog över företag och yrkespersoner är en måltavla från den dag den lanseras. Skrapbotar vill åt datan, spamkonton vill åt räckvidden, och en endpoint som kostar plattformen riktiga pengar att leverera — sökning, export, allt som berör ett externt API — är värd att missbruka helt enkelt för att den är gratis att anropa. Inget av det är illvilja riktad mot just den här plattformen; det är vardagsmat på det öppna internet.

Uppgift. De kostsamma anropsvägar som går att missbruka behövde gränser som håller under tryck, inklusive det tryck som uppstår när anropsbegränsarens eget beroende inte är tillgängligt.

Handling. Det finns 22 anropsbegränsare, var och en konstruerad för den väg den skyddar i stället för ett enda globalt tak, eftersom ett inloggningsförsök, en sökning och en massexport blir missbruk vid helt olika frekvenser. Tillståndet ligger i Redis, så att en gräns delas mellan instanser i stället för att gälla per process och vara trivial att komma runt. Det viktiga beslutet är vad som händer när Redis inte finns där: anropsbegränsarna stänger vid fel. Trafiken nekas i stället för att vinkas igenom, vilket är det mindre bekväma svaret och det enda försvarbara. Runt dem finns Cloudflare Turnstile på de vägar som är värda att utmana, 351 rader middleware för idempotens så att en skrivning som görs om inte blir två, ett ursprungslås som nekar anrop som inte kommer in genom ytterdörren, och en specialbyggd utmaning på själva katalogen.

Resultat. Missbruk blir dyrt för den som missbrukar och billigt för plattformen, och ett avbrott i anropsbegränsarens egen lagring övergår i nekade anrop i stället för i en öppen dörr. Att stänga vid fel innebär visserligen att ett Redis-problem blir ett problem som användarna ser — vilket accepterades medvetet, eftersom alternativet är att ett Redis-problem blir ett faktureringsproblem.

Använd erfarenheten i ert projekt

Använd ”Byggde hastighetsbegränsning som stänger vid fel” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

Se våra tjänster Hör av er →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Byggde hastighetsbegränsning som stänger vid fel — 22 hastighetsbegränsare som stänger vid fel

Se våra tjänster Hör av er →

Bombarderas du av botar och scrapers?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: Säkerhetsingenjör. Det bygger på Backendutveckling, API:er och integration, Tillförlitlighet och säkerhetskopiering, Nätverk och VPN, Säkerhet och Prestandaoptimering. Och nu när ni är här: vi är ganska bra på Backend- och API-utveckling, Tillförlitlig drift och övervakning, Säkerhet och behörighetshantering och Säkerhet för AI-agenter.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Bombarderas du av botar och scrapers?

Fråga om besluten bakom ”Byggde hastighetsbegränsning som stänger vid fel”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://software.engineer.company/sv/portfolio/built-fail-closed-rate-limiting-and-anti-abuse/