Jag ville bara följa upp lite efter eventet. Aligned Intelligence-challengen var faktiskt anledningen till att jag kom till hackathonet från början – problemet med att skilja legitima agenter från skadliga bots är något jag tycker är väldigt intressant. Även om vi bytte spår under själva hackathonet fastnade jag verkligen för området och vill gärna förstå det mer på djupet.

I mitt utforskande fastnade jag inte så mycket för beteendegenerering, utan mer för hur man bygger system som är säkra, förutsägbara och samtidigt inte står i vägen för legitima automatiserade användare. Jag kommer inte från ML/AI-hållet, vilket gör det ännu mer intressant att förstå hur ni som jobbar närmare tekniken tänker kring modeller, identitet och trust.

Inledande insikter från challengen

När vi började spåna kring problemet identifierade vi tre typer av trafik:

- människor
- AI-agenter
- scrapers / webcrawlers

Det som snabbt blev tydligt var att svårigheten inte ligger i vilken kategori en klient tillhör, utan om den är **godartad eller skadlig**. Beteendeanalys räcker inte som första lager, eftersom:

- scrapers kan vara både legitima och skadliga
- AI-agenter kan användas konstruktivt eller destruktivt
- tidiga beteenden ser ofta identiska ut mellan “snälla” och “elaka” agenter
- AI kan operera via DOM, headless browsers, screenshots m.m.
- både AI och scrapers kan imitera mänsklig interaktion

Det här gjorde att jag fokuserade mer på **trust-modell och arkitektur**, snarare än ML i det första steget.

Om autentisering

Det finns många sätt att autentisera maskiner och automatiserad trafik: API-nycklar, JWT/OAuth2 client credentials, HMAC-signering, IP-baserad attestering och device attestation m.m. Alla har sin plats.

Men i just den här problemformen – där frågan handlar om maskinidentitet, provenans och vad som ska betraktas som godartad automatisering – upplevde jag mTLS som den mest robusta baslinjen:

- tvåvägs verifiering
- kryptografiskt stark identitet
- möjlighet att använda egna eller vendor-specifika CA:er

- svårförfalskade credentials
- kan bära både identitet och policy på transportlagret

Det blev därför min grundsten, även om JWT/OAuth och andra metoder förstås kan komplettera på applikationslagret.

Min baseline-modell: bots är risk tills de bevisar legitimitet

I stället för att försöka avgöra “människa eller bot” på första requesten (vilket i praktiken är svårt) landade jag i en enklare och mer robust princip:

1. Första beslutet: mTLS eller public

- Har klienten ett giltigt mTLS-certifikat från betrodd CA?
  → skicka till **trusted lane**
- Saknar klienten certifikat?
  → skicka till **public lane**

Det separerar tydligt:

- verifierad automation
- oidentifierad automation + människor

2. Public-vägen: endast mänsklig trafik + strikt verifierade bottar

I public-läget behandlas all inkommande trafik som okänd tills motsatsen är bevisad.
Den övergripande policyn är:

- **Mänsklig trafik → OK**
- **Icke-mänsklig trafik → Inte OK**

Det enda undantaget gäller bottar som kan verifieras med hög säkerhet, exempelvis Googlebot. För att räknas som legitim krävs:

- korrekt och förväntad **User-Agent**
- att klientens **IP-adress ligger i officiella ranges**
- korrekt **forward + reverse DNS-matchning**

All annan automatiserad trafik betraktas som icke-tillåten i public-vägen.

Eftersom många bottar försöker imitera mänskligt beteende används ML/AI för att identifiera automatiserad trafik som inte öppet kan verifieras. Modellen analyserar nätverkssignaler,

header-konsistens och beteendemönster för att upptäcka bot-liknande beteende.
Detta stöder policyn: **botar hör inte hemma i public-lanen**, och ML/AI används för att hitta dem.

ML/AI-delen – rollen i public-vägen

Eftersom många bottar kan imitera mänsklig interaktion räcker det inte med statiska regler för att avgöra vad som är legitima användare. I public-läget används därför ML/AI som ett **sekundärt verifieringslager**, vars uppgift är att identifiera automatiserad trafik som inte kan verifieras öppet men ändå försöker framstå som människa.

Modellen fungerar som ett riskfilter: den markerar trafik som avviker från normalt mänskligt beteende och därmed bör behandlas som bot.
Det exakta genomförandet eller vilka tekniska signaler som används är inte det viktiga i denna fas – poängen är att ML/AI kompletterar de övriga trustmekanismerna genom att lyfta fram trafik som sannolikt är automatiserad.

Detta stödjer policyn: **botar hör inte hemma i public-lanen**, och ML/AI används för att upptäcka dem när traditionell verifiering inte räcker.

3. All övrig automation måste använda mTLS eller vendor-attestering

- människor → public
- verifierad Googlebot → public
- all annan automation → mTLS eller attestering

Sammanfattat:
Good automation bör kunna identifiera sig.

4. Stealth-agenter via betrodd vendor

För agenter som behöver fungera i stealth-läge tänker jag mig att:

- agenten kan vara anonym utåt
- men trafikens legitimitet intygas av en betrodd vendor (cert, attestering, CA-signering etc.)

5. Reverse proxy som central trust-punkt

En reverse proxy som:

- terminerar mTLS
- avgör lane
- filtrerar och kategoriserar icke-mänsklig trafik
- tillämpar policies och rate limiting

- enrichar requesten med metadata (lane, identitet)

Poängen är att flytta de tidiga besluten om trust och klassificering till ett gemensamt lager nära kanten, och sedan börja med identifiering.

Cloudflare och liknande principer

Du nämnde Cloudflare, vilket gjorde mig ännu mer nyfiken, eftersom deras modell verkar bekräfta mycket av det jag själv intuitivt landade i:

- mTLS / Client Certificates
- device attestation
- lagerbaserad bot-hantering (identitet → risk → beteende)
- adaptive trust
- klassificering baserad på identitet och policy, inte bara beteende

Det liknar tanken att:

- identitet → trust-nivå
- arkitektur → kontroll
- beteendeanalys → verifiering / anomali-detektion
- ML → finjustering

Det fick mig att fundera på om min baseline ligger på ungefär samma nivå som branschens etablerade aktörer använder – inte som färdig lösning, men som startpunkt.

Min fråga till er

Hur rimlig tycker ni att denna baseline-modell är?

- första beslut: mTLS eller public
- i public: endast människor + verifierad Googlebot
- all annan automation → mTLS eller vendor-attestering
- reverse proxy som central trust-punkt
- arkitektur och identitet som grund; beteendeanalys som sekundärt lager

Det område där jag själv har minst erfarenhet är ML-delen – särskilt vilka signaler som faktiskt är mest användbara för att skilja mänskligt beteende från automatiserat. Jag skulle verkligen uppskatta input kring vad som typiskt fungerar bra i sådana modeller.

Skulle ni ha möjlighet att ge lite input på dessa tankar, eller peka mig i rätt riktning om det finns något jag missar? Jag uppskattar verkligen all feedback ni vill dela. /Jakob Hallin