Säker utveckling i en tid

Av Jesper Olsen, CISO Northern Europe, Palo Alto Networks

Palo Alto Networks är ett amerikanskt cybersäkerhetsföretag grundat 2005 av Nir Zuk, tidigare chefsarkitekt på Check Point. Bolaget utvecklar teknik för nätverkssäkerhet, molnsäkerhet och hotanalys, och är särskilt känt för sina brandväggar och system för avancerad intrångsdetektering.
Företaget är noterat, har cirka 16 000 anställda globalt och verkar inom både företags- och myndighetssektorn. Företagets teknik används även i miljöer med uppkopplade produkter, IoT-system och industriella styrsystem.

Bejaka vibekodning. Men håll hård kontroll på användningen.

Generativ AI förändrar snabbt hur mjukvara utvecklas och påverkar därmed också hur elektronik konstrueras, styrs och vidareutvecklas. I takt med att allt fler produkter och system definieras av kod blir kvalitet, spårbarhet och säkerhet avgörande ingenjörsfrågor. Ett tydligt exempel på detta skifte är framväxten av så kallad *vibe coding*.

Vibe coding innebär att utvecklare, och i ökande grad även användare utan formell programmeringsbakgrund, beskriver önskad funktion i naturligt språk och får tillbaka fungerande kod genererad av ett AI-verktyg. Metoden kan kraftigt öka utvecklingstakten. Samtidigt flyttas fokus ofta mot funktionalitet, medan säkerhet och robusthet hamnar i bakgrunden.

Vad visar forskningen om riskerna?

Många av de mönster vi nu ser har identifierats genom analyser från Unit 42, Palo Alto Networks globala forskningsenhet. Unit 42 bedriver kontinuerlig research kring nya attackmetoder, sårbarheter och hur tekniska trender, som exempelvis generativ AI, påverkar hotlandskapet.

I Unit 42:s färska Global Incident Response Report 2026, baserad på över 750 analyserade cyberincidenter globalt, framgår att AI nu fungerar som en tydlig kraftmultiplikator för angripare. I de snabbaste fallen har tiden från intrång till dataexfiltration minskat från nästan fem timmar till endast 72 minuter.

Ett återkommande resultat i deras analyser är att AI-genererad kod ofta tas i bruk utan relevant skydd. Det kan handla om avsaknad av autentisering, bristande *rate limiting* (begränsning av hur ofta ett API eller en

av AI-genererad kod

funktion får anropas) eller otillräcklig validering av in- och utdata. När sådan kod integreras i system som styr elektronik, industriella processer eller uppkopplade produkter uppstår nya, ofta förbisedda, attackytor.

Rapporten visar även att identitetsrelaterade svagheter spelade en avgörande roll i nära 90 procent av de analyserade intrången, vilket ytterligare förstärker riskerna när AI-genererad kod integreras utan strikt behörighetsstyrning och tydlig identitetskontroll.

Citizen developers breddar attackytan

Utvecklingen förstärks av framväxten av så kallade citizen developers. Med hjälp av AI kan medarbetare utan teknisk expertis skapa applikationer, integrationer och automationer som snabbt blir verksamhetskritiska. För organisationen är detta ofta effektivt. För säkerhetsarbetet innebär det att fler personer påverkar kodbasen utan att nödvändigtvis förstå konsekvenserna av exempelvis prompt-injektion, rättighetsmissbruk eller bristande isolering mellan miljöer.

För säkerhetsansvariga innebär detta ett skifte: attackytan växer inte bara genom nya system, utan även genom själva sättet kod produceras på.

Enligt Unit 42:s rapport omfattade 87 procent av intrången flera attackytor samtidigt – exempelvis identitet, moln, nätverk och endpoints, vilket visar hur komplexiteten ökar när nya AI-genererade komponenter kopplas in i redan sammanlänkade miljöer.

SHIELD – ett ramverk för kontrollerad AI-utveckling

För att hantera dessa risker handlar det inte om att stoppa vibe coding, utan om att styra användningen. SHIELD är ett ramverk framtaget för att integrera säkerhetskontroller direkt i AI-drivna utvecklingsflöden och används redan av organisationer som vill kombinera snabb innovation med robust säkerhet.

SHIELD står för:

- **Separation of Duties** – AI-agenter begränsas till utvecklings- och testmiljöer
- **Human in the Loop** – mänsklig granskning vid säkerhetskritisk kod
- **Input/Output-validering** – analys av både promptar och genererad kod
- **Enforced Security Models** – särskilda modeller för säkerhetskontroller
- **Least Agency** – minsta möjliga behörigheter för AI-agenter
- **Defensive Controls** – scanning av beroenden och begränsad auto-exekvering

I praktiken flyttas säkerheten närmare själva kodskapandet, snarare än att läggas på i efterhand.

När mjukvara definierar elektroniken

Elektronik formas och uppdateras i allt högre grad av mjukvara, ofta med stöd av AI. När kod blir en central del av produktens funktion måste säker utveckling betraktas som en grundläggande ingenjörsdisciplin. *Vibe coding* är här för att stanna. De organisationer som lyckas är de som skalar sina säkerhetskontroller i samma takt som produktiviteten och därmed gör AI-driven utveckling hållbar även på lång sikt. ■