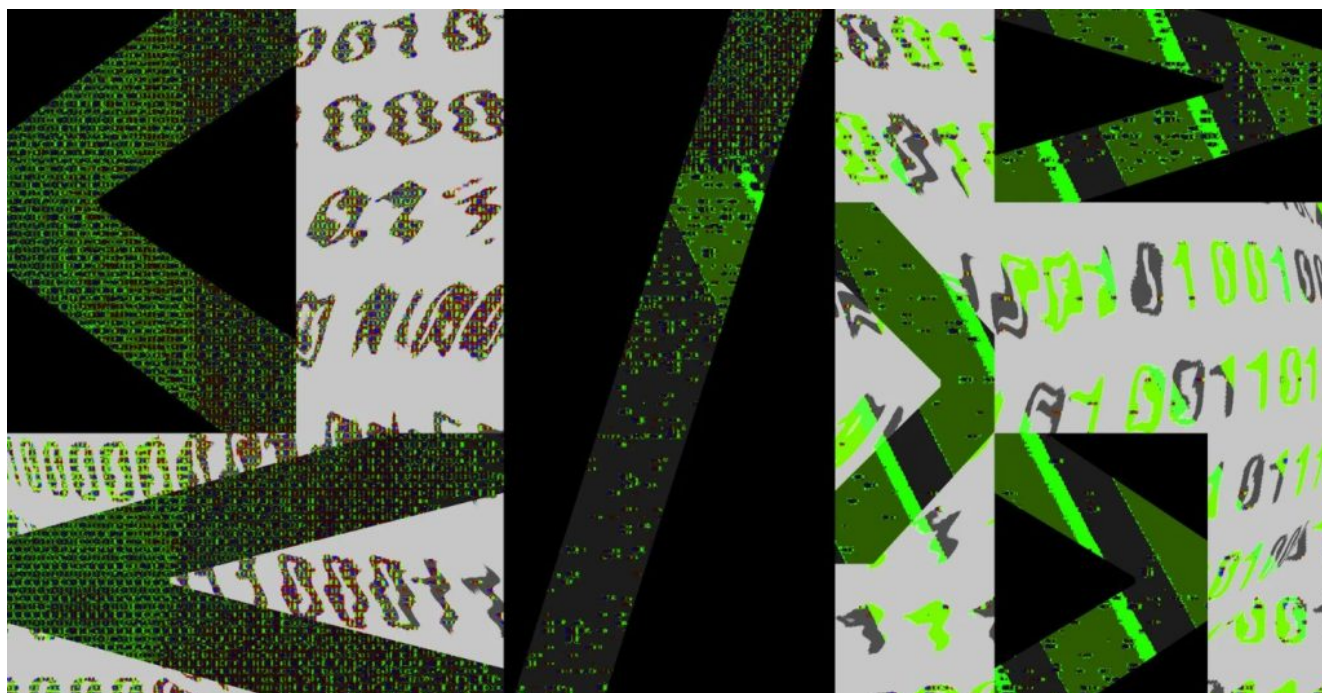


Il vibe coding è il nuovo open source, e non è un complimento



Così come probabilmente non coltivate e macinate il grano per fare la farina, la maggior parte degli **sviluppatori di** software non scrive da zero ogni riga di codice di un nuovo progetto. Questo perché farlo rallenterebbe sensibilmente il processo e rischierebbe di creare problemi aggiuntivi. Per mettere a punto i vari componenti software di base, invece, gli sviluppatori attingono a librerie preesistenti, che spesso sono [open source](#).

Per quanto sia efficiente, questo approccio può però creare **vulnerabilità**. Anche il cosiddetto [vibe coding](#), ovvero il ricorso all'[intelligenza artificiale](#) per programmare, è sempre più spesso impiegato in modo simile, con lo scopo di creare rapidamente codice che gli sviluppatori possono poi semplicemente adattare. I ricercatori di sicurezza avvertono tuttavia che la pratica sta rendendo la **sicurezza della catena di fornitura** dei software ancora **più complicata e pericolosa**.

I rischi del vibe coding

*“Stiamo raggiungendo il punto in cui l’intelligenza artificiale sta arrivando alla fine del suo periodo di grazia in materia di sicurezza”, afferma Alex Zenla, direttore tecnico dell’azienda di sicurezza cloud Edera. **“E l’intelligenza artificiale è il peggior nemico della [sicurezza](#) se si pensa alla generazione di codice che non è sicuro. Se l’AI viene addestrata in parte su software vecchi, vulnerabili o di bassa qualità disponibili sul mercato, tutte le [vulnerabilità](#) esistenti possono ripresentarsi ed essere introdotte di nuovo, per non parlare di nuovi problemi”.***

Oltre ad assorbire dati di formazione potenzialmente insicuri, il vibe coding produce bozze di codice che potrebbero non tenere conto del contesto e di aspetti importanti di un determinato prodotto o servizio. In altre parole, anche se un’azienda addestra un modello locale sul [codice sorgente](#) di un progetto e su una descrizione in [linguaggio naturale](#) degli obiettivi, **il processo di produzione si affida ancora alla capacità dei revisori umani** di individuare ogni possibile difetto o incongruenza nel codice originariamente generato dall’intelligenza artificiale.

*“Nell’era del vibe coding le squadre di tecnici devono pensare al ciclo di vita dello sviluppo”, afferma Eran Kinsbruner, ricercatore presso la società di sicurezza Checkmarx. **“Se chiedete allo stesso [llm](#) di scrivere il vostro codice sorgente specifico, ogni singola volta produrrà un risultato leggermente diverso [...]** Questo introduce un’ulteriore complicazione al di là dell’open source”.*

In un [sondaggio](#) di Checkmarx rivolto a responsabili della sicurezza e dello sviluppo, un terzo degli intervistati ha dichiarato che nel 2024 **oltre il 60% del codice della loro azienda sarà generato dall’AI. Ma solo il 18% ha affermato che la società per cui lavora dispone di strumenti approvati per**

il vibe coding.

Questione di trasparenza

I **progetti open source** possono essere intrinsecamente insicuri, obsoleti o vulnerabili agli attacchi. Ma rischiano anche di essere incorporati nel codebase senza l'opportuna trasparenza o la documentazione adeguata. I ricercatori sottolineano che alcuni dei fondamentali meccanismi di controllo e responsabilità che sono sempre esistiti nell'open source sono mancanti o gravemente frammentati quando lo sviluppo è guidato dall'AI.

"Il codice creato dall'intelligenza artificiale non è molto trasparente", afferma Dan Fernandez, responsabile dei prodotti AI di Edera. *"Nei repository come Github almeno si possono vedere cose come le [richieste di pull](#) e i [messaggi di commit](#) che permettono di capire chi ha fatto cosa al codice, e c'è un modo per risalire a chi ha contribuito. Ma con il codice AI non c'è la stessa responsabilità [...] E le righe di codice provenienti da un umano potrebbero essere parte del problema"*.

Per quanto il vibe coding possa sembrare un metodo a basso costo per creare applicazioni e strumenti essenziali altrimenti inaccessibili a gruppi con poche risorse – come le piccole imprese o le popolazioni vulnerabili – Zenla sottolinea che questa facilità d'uso comporta il **pericolo di creare una falla di sicurezza** nelle situazioni più sensibili e a rischio. *"Si parla molto di utilizzare l'intelligenza artificiale per aiutare le popolazioni vulnerabili, perché richiede meno sforzi per arrivare a qualcosa di utilizzabile"*, spiega il direttore tecnico di Edera. *"Penso che questi strumenti possano aiutare le persone con meno mezzi, ma credo anche che le implicazioni sulla sicurezza del vibe coding avranno un impatto sproporzionato proprio sulle persone che meno possono permetterselo"*.

Anche nelle aziende, che si sobbarcano in gran parte il rischio finanziario d'impresa, le ricadute personali di una vulnerabilità diffusa introdotta come conseguenza del vibe coding rischiano di avere un **impatto pesante**.

“Il fatto è che il materiale generato dall'intelligenza artificiale sta già iniziando a vedersi nei codebase”, afferma Jake Williams, ex hacker della National security agency degli [Stati Uniti](#) e attuale vicepresidente della ricerca e sviluppo di Hunter Strategy. ***“Possiamo imparare dai progressi della sicurezza della catena di fornitura del software open source. Oppure non farlo, e lasciare che la situazione diventi terribile”.***

Questo articolo è apparso originariamente [su Wired US](#).