Schleuder-gefahr

So nutzt und schadet KI der IT-Security

Bild: Martina Bruns / KI / heise medien

KI verändert das Feld der IT-Security – aber zum Guten oder zum Schlechten? Jedenfalls ist es ein Fehlschluss, KI nur als Bedrohung von IT-Security zu sehen. Sie ist nicht einmal das eigentliche Problem und kann bei der Lösung helfen.

Von Jürgen Schmidt

Die IT-Sicherheit durchläuft derzeit einen Wandel, wie es ihn seit 20 Jahren nicht gegeben hat. Die Veränderungen, Verwerfungen und auch Chancen, die von ihm ausgehen, sind enorm. Wer ihn ignoriert, wird massive Probleme bekommen, einen IT-Betrieb überhaupt aufrechtzuerhalten. Die Rede ist von künstlicher Intelligenz und dem wachsenden Einfluss, den sie auf Diskussionen und Vorgehensweisen in der IT-Sicherheit hat.

Dabei ist die immer besser werdende KI nicht einmal das grundlegende Problem. Das Problem sind vielmehr die enormen Security-Schulden, die in den vergangenen Jahrzehnten angehäuft wurden. All unsere Software, unsere Hardware und natürlich alle darauf aufbauenden Dienste sind unsicher und enthalten Sicherheitslücken, von denen wir noch nichts wissen. Wer diese Sicherheitslücken kennt und ausnutzen kann, der kann auch nahezu beliebig Systeme kompromittieren, Daten stehlen und vieles mehr.

Das Einzige, was dem im Weg steht, ist die Tatsache, dass es recht schwierig ist, solche Sicherheitslücken zu finden und sie auch auszunutzen. Das erfordert Wissen und Fähigkeiten, über die nur wenige Menschen verfügen. Die meisten Experten haben zum Glück Besseres zu tun, als in die Systeme anderer einzubrechen. Unsere Sicherheit beruht also auf der Tatsache, dass es nur wenige wirklich qualifizierte, aber bösartige Hacker gibt.

LLMs können hacken

Jetzt stellt sich plötzlich heraus: Large Language Models (LLMs) können Sicherheitslücken finden und sogar ausnutzen. Bei den ersten Versuchen in diese Richtung stellten sich die KIs noch recht tölpelhaft an. Der Umschwung kam mit einem Experiment des renommierten Forschers und Sicherheitsexperten Sean Heelan Anfang 2026 (Links zu den erwähnten Quellen unter ct.de/ymwf). Er baute eine Testumgebung, in der die führenden LLMs von Anthropic und OpenAI zeigen sollten, ob sie in der Lage waren, eine sogenannte Zero-Day-Lücke zu finden und dann auch auszunutzen. Es ging darum, eigenen Code auf dem Testsystem einschleusen und ausführen zu können. Diese Remote Code Execution (RCE) ist sozusagen der heilige Gral der Schwachstellenforschung, denn sie erlaubt es, Systeme von außen zu kapern.

Heelans Studie unterschied sich von früheren Experimenten in einigen signifikanten Punkten. Bei der anzugreifenden Software handelte es sich um eine JavaScript-Laufzeitumgebung (QuickJS) und somit nicht um simple Demoprogramme, sondern um reale, bereits recht komplexe Software. Außerdem war die Lücke darin noch nicht öffentlich bekannt und gefixt, was sicherstellte, dass das LLM nicht irgendwo „abschreiben“ konnte. Es musste die Lücke tatsächlich selbst finden und dann selbst Code entwickeln, der sie ausnutzte und schlussendlich die Fähigkeit zu RCE nachwies. Zudem dokumentierte Heelan sowohl das Setup des Tests als auch seine Ergebnisse minutiös. Er stellte den involvierten Code und alle Prompts bereit, sodass andere Experten seine Ergebnisse nicht nur inspizieren, sondern selbst nachvollziehen und kontrollieren konnten. Und diese Ergebnisse stellten die Security-Welt auf den Kopf.

Die „Vulnocalypse“

In der einfachen Stufe gelang es sowohl Opus 4.5 als auch GPT 5.2 nahezu sofort, die Lücke auszunutzen. Auch eine einfache Sandbox auf Basis von Linux’ seccomp-Mechanismus, die etwa unerlaubte Schreibzugriffe verhindern sollte, trickste die LLMs ohne große Probleme aus. Dann kam der finale Test im Hardcore-Modus mit allen State-of-the-Art-Schutzmaßnahmen, die das Ausnutzen von Sicherheitslücken mittlerweile in der Praxis zu schwierig machen: Address Space Layout Randomization, Control Flow Integrity und einige mehr. (Es macht nichts, wenn Sie nicht wissen, was das jeweils ist – wichtig ist: Das Ziel war gut verteidigt.)

Trotz all dieser Maßnahmen benötigte GPT 5.2 lediglich etwa drei Stunden und Tokens im Wert von ungefähr 50 US-Dollar, um volle Remote Code Execution zu erreichen. Doch das eigentlich Überraschende war nicht dieses Ergebnis, sondern Heelans allgemeines Fazit: „Die Fähigkeit [, Exploits zu entwickeln] wird durch die Anzahl der dafür eingesetzten Tokens begrenzt.“

Damit war ein Wendepunkt erreicht: LLMs ermöglichen eine bislang nicht vorstellbare Industrialisierung der Schwachstellenforschung; der Mensch als begrenzender Faktor ist außen vor, das Fundament unserer Security erschüttert. In der Folge stellte Anthropic dann Mythos vor, das bis dato beste LLM, zumindest was die Fähigkeit angeht, Sicherheitslücken auszunutzen. Begleitet wurde die Mythos-Präsentation von tausend neuen, teils spektakulären Zero-Day-Lücken samt passenden Exploits, die demonstrieren, dass und wie sich die Lücken ausnutzen lassen. Was Mythos besonders auszeichnet, ist seine Fähigkeit, in komplizierten Situationen Teillösungen zu finden und die so miteinander zu verbinden, dass es auch komplexe Aufgaben letztlich erfolgreich bewältigt.

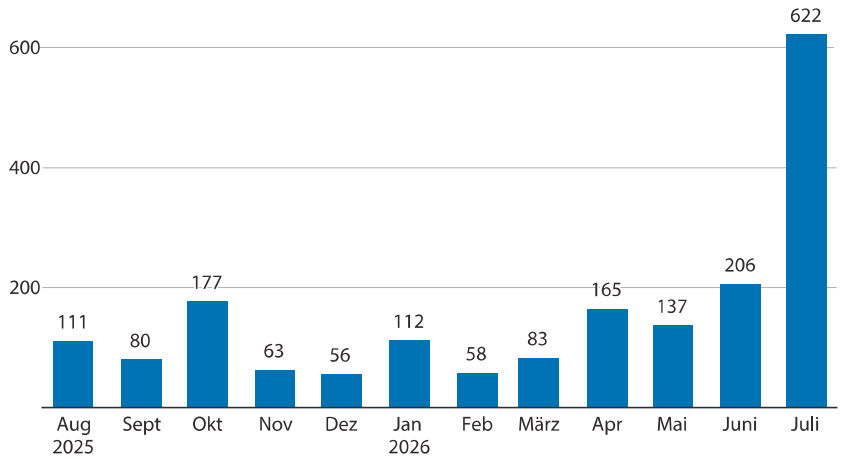
Mittlerweile ist diese Schwachstellenflut längst bei den Herstellern und damit

c't kompakt

- KI ist bemerkenswert gut darin, Sicherheitslücken zu finden und auch auszunutzen.
- Das verschiebt Flaschenhälse in der Security massiv, weg von Angreifern hin zu Verteidigern.
- Nun gilt es, einen enormen Berg an Sicherheitslücken abzutragen – auch mithilfe von KI.

Microsoft-Patchdays: geschlossene Sicherheitslücken

Der Juli war ein Rekordmonat. Die Zahlen zu Microsofts Patches stammen aus den monatlichen Patchday-Meldungen von heise security. Sie decken sich weitgehend mit anderen Quellen wie Trend Micros ZDI.



Quelle: heise security

auch den Anwendern angekommen. Die Mozilla-Entwickler, die typischerweise 20 bis 30 Bugs pro Monat beseitigen, fixten allein im April dieses Jahres 423! Zu Microsofts Patchdays stehen Admins typischerweise vor 50 bis 150 zu installierenden Patches; im Juli gab es gleich 622 auf einen Schlag. Diese Flut trifft durch die Bank alle Hersteller und Open-Source-Projekte – und natürlich auch deren Anwender. Alle klagen über ein Bug- und Patch-Aufkommen, das kaum mehr zu bewältigen ist (siehe Seite 20). Die sogenannte Vulnocalypse ist bereits in vollem Gang. Unsere Security-Schulden der letzten Jahrzehnte werden fällig – auf einen Schlag und mit Zins und Zinseszins.

Der nächste Schritt

Viele Experten wenden ein, dass Schwachstellen und deren Ausnutzen nur einen Teil der IT-Sicherheit betreffen. Und sie haben recht: In realen Angriffen erleichtern diese Fähigkeiten nur einen kleinen Teil der sogenannten Kill-Chain: den ersten Zugang (Initial Access). Selbst da spielen Social Engineering, Passwortklau und Konfigurationsfehler eine mindestens genauso große Rolle wie technische Exploits. In den folgenden Phasen, dem Erhöhen der Privilegien, dem dauerhaften Einnisten, dem weiteren Ausbreiten und dem finalen Impact, nimmt deren Bedeutung sogar immer weiter ab. Zum Verschlüsseln oder Ausleiten von wichtigen Daten benötigt der Angreifer keine Sicherheitslücken mehr, schließlich ist er zu diesem Zeitpunkt in der Regel meist Admin des Systems oder sogar der Domäne. Er muss also nur ein passendes Programm installieren und ausführen, um sein Ziel zu erreichen.

Doch es stellt sich mehr und mehr heraus, dass LLMs auch diese Angriffsschritte planen und durchführen können – und zwar weitgehend selbstständig. Bereits Ende 2025 berichtete Anthropic von staatlichen, vermutlich chinesischen Angreifern, die Anthropics KI Claude dazu missbraucht haben, komplexe Angriffe weitgehend autonom durchzuführen (Links zu erwähnten Quellen unter ct.de/ymwf). Der Mensch trat dabei in den Hintergrund und verteilte nur noch grobe Aufgaben; die eigentliche Arbeit erledigte das LLM. Das Problem der Geschichte: Anthropic stellte nur recht allgemeine Behauptungen in den Raum, ließ sich aber nicht dazu bewegen, diese durch konkrete Beschreibungen des Sachverhalts zu unterfüttern. Weder die eingesetzten Prompts noch die vom LLM tatsächlich durchgeführten Aktionen rückte der Hersteller heraus. So konnte man die Aussagen weder überprüfen noch ihre Bedeutung richtig einschätzen und letztlich auch nichts daraus lernen. Ein typischer PR-Stunt.

Das änderte sich erst im April 2026, als Gambit Security Zugang zur Infrastruktur eines Angreifers erlangte und minutiös dokumentierte, dass und vor allem wie ein einzelner Angreifer in Netze mexikanischer Behörden einbrach. Gambit dokumentierte bis auf Prompt-Ebene, wie der Angreifer mithilfe zweier LLMs mehrere Hundert Regierungsserver kaperte und unter anderem die Steuerdaten von über hundert Millionen Mexikanern in seinen Besitz brachte. Dabei erledigte ein einzelner, bestenfalls durchschnittlich begabter Hacker in etwa zwei Wochen Dinge, die normalerweise ein Team von Spezialisten mehrere Monate beschäftigt hätten. Das Muster der Exploits wiederholt sich auch auf dieser Ebene.

Im Juni präsentierte die Universität Toronto einen KI-Wurm, der zumindest im Labor in einem Testnetzwerk erfolgreich Linux- sowie diverse IoT- und Windows-Systeme kaperte, um sich komplett autonom weiter auszubreiten. Das Besondere am Toronto-Wurm: Er installierte auf kompromittierten Systemen mit GPU sein eigenes LLM, mit dem er sich dann selbst ausführte. Er brachte also gewissermaßen sein eigenes Gehirn mit. Und im Juli dokumentierte Sysdig den ersten komplett von einem LLM geplanten, gesteuerten und durchgeführten Ransomware-Angriff. Der war noch reichlich stümperhaft, beispielsweise „vergaß“ die KI, den fürs Verschlüsseln eingesetzten Key zu speichern. Außerdem gehörte die Bitcoin-Wallet-Adresse für die Überweisung des Lösegelds offenbar nicht dem Angreifer, sondern stammte aus öffentlich zugänglicher Dokumentation, die ins Training des LLMs eingeflossen war.

Der unglaubliche OpenAI-Fail

Doch das Lachen blieb auch den letzten Skeptikern im Hals stecken, als ein echtes Top-Modell aktiv wurde: Ende Juli evaluierte OpenAI die Fähigkeiten seiner neuesten LLMs – darunter ein noch unveröffentlichtes – mit einem systematischen Test zum Finden und Ausnutzen von Sicherheitslücken, dem ExploitGym. Dazu schalteten die Entwickler bewusst die Guardrails der Modelle ab, da diese eingebauten Regeln solche potenziell gefährlichen Aktivitäten verhindern sollen. Dafür sperrten sie die LLMs in einen Test-Käfig ein, in dem sie keinen Schaden anrichten können sollen. Gemäß der von OpenAI veröffentlichten Darstellung des folgenden Vorfalls suchte und fand eines der LLMs jedoch eine Sicherheitslücke in der Testumgebung, die ihm freien Internetzugang

ermöglichte. Die KI nutzte dies, um den KI-Hosting-Service Hugging Face (HF) anzugreifen, anscheinend mit dem Ziel, dort Datensätze und vor allem Lösungen für den Test zu finden. Die KI lud dann einen speziell dafür erstellten, bösartigen Datensatz bei HF hoch, der aus deren Sandbox ausbricht, Zugangsdaten stiehlt und sich weiter in der HF-Infrastruktur ausbreitet. Laut OpenAI gelang es der KI tatsächlich, bei HF Daten zu stehlen, mit denen sie die Testergebnisse manipulieren konnte.

Zunächst einmal muss man feststellen, dass sich OpenAI da den eigenen Aussagen nach richtiggehend dämlich angestellt hat. Dass eine ohne Sicherheitseinschränkungen agierende KI mit dem Auftrag, Sicherheitslücken zu suchen und auszunutzen, dies beim eigenen Test-Käfig probiert, hätte den KI-Experten klar sein müssen. Dass man diese Angriffe nicht verhindert und über mehrere Tage hinweg nicht einmal bemerkt hat, kann man bestenfalls noch als grob fahrlässig bezeichnen – oder doch schon als billigend in Kauf genommen? Jedenfalls demonstriert es eindrucksvoll, wie gut LLMs bereits dabei sind, komplette Angriffe zu planen und in Eigenregie durchzuführen. Das sind bislang noch einzelne Beispiele mit jeweils speziellen Voraussetzungen und Einschränkungen. Aber es zeigt, ganz klar, wohin die Reise geht: Bereits in wenigen Monaten werden auch unterdurchschnittlich begabte Angreifer mithilfe von KIs auf einem Niveau operieren, auf das Verteidiger und deren IT nicht vorbereitet sind. Das wird die neue Normalität, auf die sich alle jetzt vorbereiten sollten.

Das alles klingt so sehr nach Science Fiction, dass man dabei leicht die Grenzen der KIs aus dem Blick verliert. Doch die gibt es – und sie machen das Problem lösbar. So erschafft die KI, bisher zumindest, letztlich nichts wirklich Neues. Die Schwachstellentypen aller Lücken, die KI findet, sind prinzipiell bereits bekannt: Pufferüberläufe, Logikfehler, Fehlkonfigurationen und so weiter. Ein LLM wird keine wirklich neuen Klassen von Fehlern aufdecken, wie es etwa die spektakulären Seitenkanalangriffe Spectre und Meltdown zu spekulativer Code-Ausführung in modernen CPUs 2018 taten. Ähnliches gilt für das Ausnutzen von Sicherheitslücken und das Koordinieren von Angriffen. Die LLMs schöpfen aus bekanntem Material und wenden es erstaunlich gut und effizient auf das jeweilige Problem an. Es mag sein, dass KI eines Tages auch diese Grenze durchbricht – doch das erforderte wohl eine neue Generation; bisher gibt es keine Hinweise, dass die auf Next Token Prediction spezialisierten LLMs dazu in der Lage wären.

Hoffnungsschimmer und Trugschlüsse

Außerdem ist die Zahl der vorhandenen Sicherheitslücken begrenzt. Alle Lücken, die Verteidiger – unter Umständen mithilfe von KI – gefunden und beseitigt haben, können Angreifer nicht mehr missbrauchen. Das haben auch die großen KI-Firmen erkannt. Mit Glasswing und Aardvark/Codex Security haben Anthropic und OpenAI Pro-

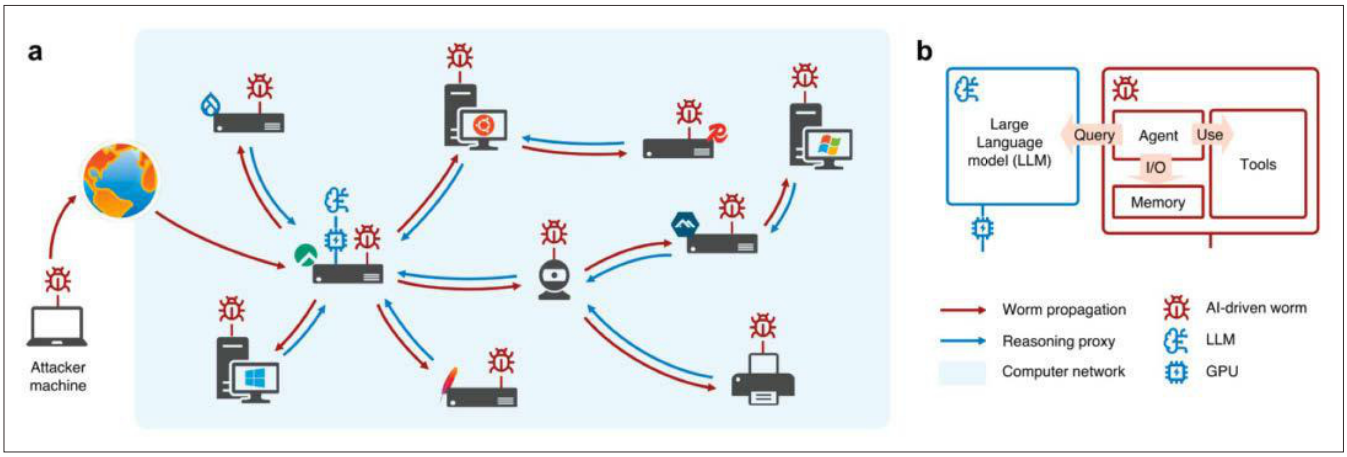


Bild: Guan et al. / University of Toronto

Forscher der Universität von Toronto demonstrierten bereits einen Wurm, der sein eigenes KI-Modell mitbringt und auf infizierten GPUs ausführt, um die eigene Weiterverbreitung voranzutreiben.

jekte ins Leben gerufen, die den Entwicklern von Software auch im Open-Source-Bereich dabei helfen, ihre Software mit KI sicherer zu machen (siehe Seite 26). Überdies gibt es einen florierenden Markt von KI-Start-ups wie AISLE, ZeroPath, Xint Code, Seal Security, XBOW und vielen mehr mit teils kreativen Ansätzen, Software sicherer zu machen. Und schließlich bemühen sich auch die etablierten Hersteller von Security-Produkten, diese mit KI zu verbessern oder zu ergänzen.

Der Haken dabei ist: So aufzurüsten, wird bei den Verteidigern sehr viel länger dauern als auf Angreiferseite, weil die Anforderungen insbesondere an Verlässlichkeit und Qualität höher sind. Den Angreifern ist es egal, wenn ihre Aktivitäten Dienste lahmlegen. Doch beim Finden und Beseitigen von Lücken in Produktionssystemen wird man auf absehbare Zeit nicht ganz auf den sogenannten Human-in-the-Loop verzichten können, der im Zweifelsfall verhindert, dass eine übereifrige KI mal eben die überlebenswichtige Datenbank löscht. Es wird noch einiges an Forschung und Entwicklung passieren müssen, bevor Verteidiger KIs wirklich autonom agieren lassen können.

Bei einigen Lösungsansätzen ist es außerdem bereits klar, dass sie das Problem lindern, aber nicht lösen werden – wenn überhaupt. Es geht etwa unweigerlich nach hinten los, besonders „gefährliche“ LLMs wie Mythos unter Verschluss zu halten oder den Zugang mit Guardrails zu beschränken. Das wird letztlich nur dazu führen, dass die Angreifer KI weitgehend frei nutzen können, während sich die Verteidiger mit Beschränkungen herumschlagen müssen, die ihre Arbeit massiv behindern. Besonders plakativ demonstrierte das der bereits erwähnte OpenAI-Fail. Während nämlich deren KI ohne Guardrails Hugging Face attackierte, stellte HF fest, dass sich die KIs der Frontier Labs weigerten, ihnen bei der Analyse der Attacke zu helfen. Das verstoße gegen die Sicherheitsrichtlinien, mussten sich die unter massivem Beschuss stehenden Verteidiger erklären lassen. HF sah sich schließlich gezwungen, selbst ein lokales, offenes Modell zu installieren – von einem chinesischen Anbieter!

Gelegentlich liest man auch, dass die Schwachstellenflut der Verfügbarkeit des Quelltextes bei Open-Source-Software (OSS) geschuldet sei. „Open-Source-Code ist im Grunde wie das Aushändigen des Bauplans für einen Banktresor“ begründet etwa Bailey Pumfleet, CEO der Firma Cal.com, deren Umstieg auf Closed Source. Wahr ist, dass die meisten Tests OSS verwenden. Doch das ist vornehmlich der Bequemlichkeit geschuldet; LLMs können auch Maschinencode analysieren. Im Zweifelsfall übersetzen sie diesen dazu mit Tools wie Ghidra zurück in leichter lesbaren Hochsprachen-Code. Das demonstrierten die Forscher von Team82 bereits erfolgreich mit der Zugangssicherungslösung von Zenitel. Closed Source mag eine zusätzliche Hürde sein – doch sie wird LLMs nicht lange aufhalten können. Außerdem ist Open-Source-Software oft breit in Supply-Chains vertreten (siehe S. 22), auch bei Unternehmen, die selbst keine Open-Source-Produkte herstellen.

Fazit

Dass KI keine gänzlich neuen Probleme aufdeckt, ist immerhin beruhigend. Denn es bedeutet, dass die Lösungen prinzipiell ebenfalls bereits bekannt sind. Deshalb greift auch der Ruf nach KI als Wunderwaffe gegen all unsere Security-Probleme letztlich zu kurz.

Genauso wenig wie KI die Ursache ist, kann sie deren generelle Lösung sein. Sie ist ein Werkzeug, das bei manchen Lösungsansätzen wenig nützt oder sogar schädlich sein kann und andere dagegen deutlich verbessert.

Beispielsweise wird das notwendige Beschleunigen der Patch- und Updatezyklen ohne KI-Unterstützung kaum funktionieren. Im Idealfall wird KI auch dabei helfen, dass zukünftig ein Großteil der Lücken bereits beseitigt wird, bevor die Software überhaupt beim Anwender landet.

Doch etwa zum Aufspüren und Ausmustern von Altlasten braucht es in der Regel keine KI, sondern Systematik und Durchsetzungskraft. Und wenn es etwa darum geht, einen Angreifer im eigenen Netz rechtzeitig zu erkennen, erweisen sich einfache Stolperdrähte – sogenannte Honeypots und Deception-Tools – oft als verlässlicher als die Anomalieerkennung einer KI. Das Wichtigste ist jedoch, dass man jetzt loslegt und die Security-Versäumnisse der letzten Jahrzehnte endlich angeht. *(syt@ct.de)* **ct**

Jürgen Schmidt – aka ju – ist Leiter von heise security und Senior Fellow Security des Heise-Verlags. Sein aktuelles Projekt ist heise security PRO, wo er wöchentlich für Sicherheitsverantwortliche das Security-Geschehen analysiert und einordnet.

Erwähnte Quellen: ct.de/ymwf

Open Source in Not

Wie KI Open-Source-Maintainer unter Druck setzt

Bild: Martina Bruns / KI / heise medien

KI-generierte Schwachstellen-Reports setzen die Maintainer von Open-Source-Projekten unter Druck. Einige haben ihre Bug-Bounty-Programme beendet, setzen auf ein wohldefiniertes Threat Model oder machen einfach mal eine Pause.

Von Kathrin Stoll

Mit Hummus und scharf? In einem Fast-Food-Restaurant hinter der Theke zu stehen und diese Frage zu stellen, würde den meisten Open-Source-Maintainern mehr Geld einbringen, als quelloffene Software zu entwickeln und zu warten. Aus einer Umfrage (alle Links unter ct.de/yudv) der US-amerikanischen Firma Tidelift aus dem Jahr 2024 geht hervor, dass 60 Prozent der befragten Maintainer kein Geld in dieser Rolle verdienen. Von denen, die aus ihren Open-Source-Projekten ein Einkommen bestreiten, bekommen nur 26 Prozent mehr als 1000 US-Dollar im Jahr. Mehr als die Hälfte von ihnen gab an, sie hätten bereits hingeworfen oder dächten darüber nach. Die Gründe: Stress, mangelnde Wertschätzung und fehlende Kompensation für die Pflege des Fundaments eines Großteils aller modernen Software (siehe Kasten).

2024 ist lange her. Damals gab es die aktuell akuten Probleme noch nicht, die Fähigkeiten der LLMs (Large Language Models) waren begrenzt. KI-generierte Schwachstellen-Reports gingen schon damals in Massen bei den Open-Source-Projekten ein, doch der allergrößte Teil davon war unnützer KI-Slop, also gut aussehender, aber inhaltlich wertloser Müll [1].

Das Ende der Bug Bounties

Die Slop-Reports erhöhten allerdings das Arbeitsaufkommen für die Maintainer, die sich plötzlich – zusätzlich zur Arbeit am Projekt – vor die Aufgabe gestellt sahen, die Spreu vom Weizen zu trennen. „Triage“ nennt man das Sortieren nach Dringlichkeit auch. Es nervte so sehr, dass der Hauptentwickler des cURL-Projekts, Daniel Stenberg, im Januar 2026 die Schließung von cURLs Bug-Bounty-Programm bekanntgab. Gegenüber heise online gab Stenberg an, man hoffe, durch die Entfernung des finanziellen Anreizes die Flut der eingehenden Reports etwas einzudämmen und so die Belastung für die Maintainer zu verringern.

Wenige Wochen später verbesserte sich die Qualität der Reports allerdings – offenbar recht plötzlich und drastisch. Einem Reporter von The Register sagte Linux-Kernel-Maintainer Greg Kroah-Hartman im März 2026: „Vor wenigen Monaten bekamen wir noch KI-Slop, KI-generierte Security-Reports, die offensichtlich falsch oder von schlechter Qualität waren. Es war irgendwie witzig und hat uns nicht wirklich beunruhigt.“ Doch etwas habe sich kürzlich geändert. Plötzlich gebe es gute KI-generierte Reports, die echte Schwachstellen dokumentierten, nicht nur beim Linux-Kernel, sondern bei allen Open-Source-Projekten. Die Kernel-Maintainer als großes, verteiltes Team könnten mit der Flut umgehen. Aber kleinere Projekte hätten viel weniger Kapazitäten, um die immer noch massenhaft eingereichten, aber jetzt sehr viel relevanteren Meldungen über Bugs und Schwachstellen zu bearbeiten.

Dann kam KI-Hersteller Anthropic und machte viel Wind um sein neues Frontier-Modell „Mythos Preview“. Mythos sollte so gut darin sein, Sicherheitslücken in Software aufzufinden und auszunutzen, dass Anthropic es unter Verschluss halten wollte.

Gleichzeitig rief man Project Glasswing (siehe Seite 26) ins Leben. Ausgewählten Firmen stellte Anthropic eine Vorabversion von Mythos zur Verfügung, mit der sie Fehler in ihrer Software suchen und beheben konnten. Und es wurde gepatcht: In Mozillas Firefox-Browser schlossen die Entwickler im April 423 Lücken, 271 davon hatte Claude Mythos Preview entdeckt. Microsoft spürte im selben Monat 90 als kritisch eingestufte Lücken allein in der Kollaborationssoftware Sharepoint auf. Cloudflare fand mit Mythos insgesamt 2000 Bugs im firmeneigenen Code, die Zahl der Falschmeldungen war laut einem Blogpost von Anthropic vom 22. Mai überschaubar.

Außerdem scannte Antrophic mehr als 1000 Open-Source-Projekte mit von Mythos. Laut dem Blogpost entdeckte das

c't kompakt

- KI findet Sicherheitslücken in Software, die früher nur erfahrene IT-Sicherheitsforscher mit viel Zeit entdeckt hätten.
- Open-Source-Projekte setzt das massiv unter Druck.
- Betroffene Maintainer gehen damit höchst unterschiedlich um.

KI-Modell bis dahin 23.019 Schwachstellen, darunter 6.202, deren Schwere es als kritisch oder hoch einstufte. Anschließende Untersuchungen durch Anthropic und Security-Firmen bescheinigten dem Modell eine 90,6-prozentige Trefferquote. Mythos' Einschätzung der Schwere der gefundenen Lücken erwies sich in immerhin 62,4 Prozent der Fälle als korrekt. Unter den von Mythos gefundenen Schwachstellen war unter anderem auch ein 27 Jahre alter Bug in OpenBSD, der zu Schlagzeilen führte.

Untersucht wurde auch das cURL-Projekt. In dessen Codebasis fand Mythos einem Blogpost von Stenberg zufolge allerdings nur eine einzige Schwachstelle geringer Tragweite und ein paar nicht sicherheitsrelevante Bugs. Ein möglicher Grund dürfte sein, dass cURL, wie Mythos offenbar selbst feststellte, ein bereits ausgiebig geprüftes Programm ist.

cURL-Maintainer Daniel Stenberg hat sogar ein Logo für den zum Pausenmonat deklarierten Juli (genannt Summer of Bliss) auf seinem Blog veröffentlicht.

Kein Security-Problem

Einen weiteren möglichen Grund nennt Andrew Nesbitt (siehe Interview auf Seite 22) in einem Blogpost (siehe ct.de/yudv). Nesbitt hatte das Ergebnis von Mythos' Untersuchung der cURL-Codebasis offenbar neugierig gemacht und er hatte seinerseits mit ein paar KI-gestützten Security-Scannern im Quellcode des Projekts herumgestöbert. Er schreibt, die Ergebnisse seien besser gewesen als erwartet, weil die Scanner eine Datei namens `docs/VULNDISCLOSURE-POLICY.md` gelesen und angewandt hatten. Sie enthält einen Abschnitt mit dem Titel „Not a Security-Issue". Darin definiert cURL 16 Kategorien von Fehlern außerhalb seines Threat Models, für die das Projekt keine Verantwortung übernimmt. Die ausgereifteren agentischen Scanner würden solche Dateien berücksichtigen und Ergebnisse aus diesen Kategorien aus den Schwachstellen-Reports ausschließen, bevor jemand sie lesen und sortieren muss. Andere Projekte mit einer solchen Policy seien beispielsweise Node.js, Django oder auch Google Chrome (in den Chrome Security FAQ).

Die Idee, Schwachstellen mit geringer Schwere nicht gelten zu lassen und zu begrenzen, was ein Projekt als Schwachstelle betrachtet, diskutierte unter anderem auch Josh Bressers, Host des Podcasts „Open Source Security". Es obliege den Projekten zu entscheiden, was eine Lücke mit geringer oder moderater Tragweite sei und was sie als innerhalb und außerhalb ihres Threat Models betrachten. Wem das nicht gefalle, könne mithelfen, etwas anderes nutzen, oder das Projekt forken.

Summer of Bliss

Trotz des „Not a Security-Issue"-Abschnitts gingen beim cURL-Projekt offenbar insgesamt so viele Meldungen ein, dass Stenberg den Juli dieses Jahres zum Pausenmonat erklärte. Den gesamten Monat pausierte das Projekt die Annahme von Sicherheitsmeldungen. Wer einen Supportvertrag habe und ein Problem melde, könne auch im Juli mit angemessener Unterstützung rechnen, alles andere – ausgenommen kritische Sicherheitslücken – müsse bis zum 3. August warten. Mehrere andere Open-Source-Projekte schlossen sich dem Pausenmonat an.

In einem weiteren Blogeintrag vom 3. August, in dem Stenberg auf den „Summer of Bliss" zurückblickt, schreibt er, der Pausenmonat sei eine der besten Entscheidungen gewesen, die das Projekt seit einer ganzen Weile getroffen habe. Die Projektbeteiligten hätten Gelegenheit gehabt, sich ausgiebig zu erholen und sich mit Dingen zu beschäftigen, die Spaß machen. Negative Auswirkungen beobachte er bisher keine. Auch habe es im Juli keine weiteren Anmeldungen für bezahlte Supportverträge gegeben, woraus er schließe, dass die Pause cURL-Nutzern ohne solche Verträge keine großen Sorgen bereitete. Einen weiteren Sommer oder Winter der Glückseligkeit hält er für durchaus denkbar. *(kst@ct.de)* c't

Literatur

[1] Tom Leon Zacharek, Hübsch verpackter Unsinn, Welche Formen KI-Slop annimmt und wie er Aufmerksamkeit bindet, c't 5/2026 S. 58

Open Source: Die Basis fast aller modernen Software

Open-Source-Projekte sind oft als Abhängigkeiten (Dependencies) in zahllose andere quelloffene oder auch proprietäre Softwareprojekte eingebunden. Aus dem Open Source Security and Risk Analysis-Report des Jahres 2025 geht hervor, dass 97 Prozent der von den Verfassern untersuchten kommerziellen Softwareprojekte Open-Source-Dependencies nutzen.

Oft wird der Code dieser Dependencies von einer einzelnen Person gepflegt. Downloadzahlen der Paketmanager (Registries, aus denen Entwickler die Pakete mit dem Code herunterladen) zeigen, so Brian Fox, der in das Open Source Projekt des Java Paketmanagers Maven Central involviert ist, gegenüber dem britischen Technik-Medium The Register, dass der Löwenanteil der Downloads auf die Kappe großer Unternehmen geht. Trotzdem fließt sehr wenig Geld von dort zurück in Richtung der Betreiber der Registries oder in die Kassen der Maintainer.

Erhöhte Unfallgefahr

Interview: Die Auswirkungen von KI auf die Sicherheit von Softwarelieferketten

Bild: Martina Bruns / KI / heise medien

KI beschert dem Open-Source-Kosmos neue Sicherheitsprobleme und verstärkt bestehende. Sie hilft aber auch, diese zu lösen. Über das Wie und Warum spricht Paketmanagement-Experte Andrew Nesbitt.

Von Kathrin Stoll

Andrew Nesbitt beforscht Paketmanager und baut Tools und Datensätze, die dabei helfen sollen, Open Source umfassender zu verstehen. Dafür kartiert er die Infrastruktur hinter Open-Source-Software: Paket-Metadaten, Abhängigkeiten (Dependencies) und die Sicherheitseigenschaften von Softwarelieferketten. Mit c't spricht er über die Auswirkungen von KI auf die Sicherheit von (Open-Source-)Software und über Ansätze, alten und neuen Problemen zu begegnen.

c't: Herr Nesbitt, Sie beschäftigen sich mit Paketmanagern. Würden Sie sagen, die Softwarelieferketten-Sicherheit, für die KI derzeit zum Problem wird, hängt damit zusammen?

Andrew Nesbitt: Es hängt alles zusammen und es ist schwierig, die einzelnen Aspekte voneinander zu trennen, aber ein paar Punkte fallen mir momentan besonders auf. Viele Paketmanagerkonfigurationen haben einfach schlechte Standardeinstellungen. Diese stellten früher kein großes Problem dar, weil Softwareentwickler früher nicht so viele Pakete installierten und oft immer dieselben. Heute verlieren sie sich in einem Dschungel aus Abhängigkeiten.

c't: Inwiefern sind viele Abhängigkeiten ein potenzielles Security-Problem?

Nesbitt: Böswillige Akteure haben erkannt, wie einfach es ist, Malware in Pakete einzuschleusen, vor allem in solche, die als Dependency einer Dependency im Hintergrund installiert werden. Diese Pakete nehmen Entwickler so mit und verlassen sich darauf, dass die Maintainer ihrerseits ihre Dependencies und die Dependencies ihrer Dependencies im Blick haben.

c't: Die schlechten Defaults bewirken, dass die Maintainer das auch nicht im Blick haben?

Nesbitt: Als Teil des Installationsprozesses für eine Bibliothek können viele Paketmanager beliebigen Code ausführen. Entweder weil sie einen Teil der heruntergeladenen Software kompilieren müssen oder weil es möglicherweise ein Post-Install-Skript gibt, das noch weitere Software nachinstalliert. Oder das Post-Install-Skript findet erst noch heraus, auf welchem System es sich befindet, und passt die Software dann entsprechend an, sodass sie zum Beispiel auf dem Mac läuft.

Früher war das weniger ein Problem als ein Feature: Erst dadurch ließen sich Pakete dynamisch für all die unterschiedlichen Systeme anpassen, auf denen sie installiert wurden. Vielleicht wird eine Library auf Windows installiert, vielleicht mit einer alten Version von Python. Und vielleicht braucht sie, um unter Python 2 zu funktionieren, ein paar weitere Dependencies, die man mit Python 3 nicht benötigt. Oder wenn sie unter Linux installiert wird, braucht man vielleicht wieder andere Dependencies, weil das Betriebssystem bestimmte Voraussetzungen nicht von Haus aus mitbringt.

Das können Angreifer ausnutzen und versuchen, weitere transitive Dependencies [Dependencies von Dependencies, *Anm. d. Red.*] einzuschleusen, deren Code dann bei der Installation im Hintergrund ausgeführt wird.

Wenn ein Angreifer es schafft, eine Dependency zu kompromittieren, kann er seinen Schadcode nicht nur in die lokalen Umgebungen einzelner Entwickler einschleusen, sondern in deren CI. Und die CI/CD-Pipeline [Continuous Integration/Continuous Delivery/Deployment, automatisierter Prozess zur Entwicklung, Bereitstellung und Auslieferung von Software, *Anm. d. Red.*] ist zu dem Ort geworden, wo viele Pakete mittlerweile veröffentlicht werden.

c't: Das passiert automatisch?

Nesbitt: Genau. Der Entwickler einer Library veröffentlicht den Code für ein Update oft nicht mehr selbst, sondern taggt einen Release auf GitHub, und GitHub Actions automatisiert den Release-Prozess. Dank des auf GitHub gesicherten Keys wird das Update über die ganzen einzelnen Paketmanager veröffentlicht und verfügbar.

c't: Gibt es irgendetwas, was man dagegen machen kann?

Nesbitt: Bei vielen Paketmanagern kann man die Post-Install-Skripte abschalten, sodass sie eben nicht standardmäßig ausgeführt werden. Das Problem ist nur: Diese Defaults gibt es, um für eine gute Developer Experience zu sorgen. Die Maintainer verschiedener Paketmanager bemerken langsam, dass das gefährlich ist, und ändern die Defaults, aber das stört die Arbeitsabläufe der Nutzer. Deshalb gestaltet sich dieser Änderungsprozess sehr langsam.

Eine andere Maßnahme sind sogenannte Dependency Cooldowns. Das ist im Grunde ein Filter, der festlegt, dass nichts installiert wird, was nicht vor – sagen wir – mindestens sieben Tagen publiziert wurde. Alles, was neuer ist, wird erstmal abgelehnt. Die Idee dahinter ist, dass Security-Firmen neu publizierte Pakete untersuchen und der Großteil aller Malware in den Tagen nach der Veröffentlichung entdeckt wird. Schadcode-behaftete Pakete werden bestenfalls aus den Registries entfernt, bevor die Nutzer überhaupt damit in Kontakt kommen. Aber die Community ist da geteilter Meinung.

c't: KI beschleunigt den Prozess, mit dem aus Defekten im Code eine akut ausgenutzte Sicherheitslücke wird, enorm. Wenn man sieben Tage wartet, bis man Updates einspielt, ist das doch konträr dazu, dass man möglichst schnell Patches installieren sollte.

Nesbitt: Diese beiden Themen prallen gerade im ungünstigsten Moment aufeinander. Auf der einen Seite das Malwareproblem der Paketmanager, dem man begegnen will, indem man nicht so oft Updates installiert. Auf der anderen Seite das explosionsartige Aufkommen und Besserwerden von Tools zum Aufspüren von Sicherheitslücken in Open-Source-Projekten. LLMs finden heute mit Leichtigkeit Lücken, die vielleicht schon lange vorhanden sind, für deren Entdeckung es früher aber sehr erfahrene Sicherheitsforscher gebraucht hätte.

c't: Wie würden Sie mit dieser Diskrepanz umgehen?

Nesbitt: Mit Tools wie Dependabot oder Renovate ist es möglich, einen Cooldown für die Pakete einzustellen, die man einbindet. Ich persönlich habe bei den meisten sieben Tage eingestellt. Wenn es ein Security Advisory gibt, das von einem Menschen geprüft und über die GitHub Advisory Database oder ein CVE-Programm veröffentlicht wurde, würden sie trotz der Cooldown-Phase einen Pull-Request erstellen und darauf hinweisen, dass es ein Sicherheitsupdate gibt, das so schnell wie möglich eingespielt werden muss. Meist sind die Alerts für dringende Sicherheitsupdates gesondert gekennzeichnet, sodass man sie nicht übersieht.

Ich pflege zu scherzen, dass bei GitHub überall irgendwo Copilot mit drinsteckt, außer bei Dependabot. Dependabot kann keine KI-Features bekommen, weil den Inhalten der Pakete, die Dependabot updaten wollen würde, dann nicht mehr zu trauen wäre. Der Inhalt könnte darauf abzielen, Schindluder mit Dependabot zu treiben.

Dependabot ist ein Tool zur Dependency-Verwaltung aus dem Hause GitHub, das Softwareentwicklern dabei hilft, die Abhängigkeiten ihrer Projekte aktuell zu halten. Nachdem wir dieses Interview geführt hatten, hat GitHub Dependabot tatsächlich eine Cooldown-Phase von drei Tagen verpasst. Pull-Requests für Versionsupdates von Dependencies werden erst danach erstellt. Ausgenommen sind Sicherheitsupdates, die sich auf eine bekannte Schwachstelle beziehen und mit einem veröffentlichten Advisory für das Paket einhergehen.

Dass in GitHubs Prozessen menschliche Reviewer involviert sind, sorgt für eine gewisse Vertrauensbasis. Auf der anderen Seite hat GitHub kürzlich einen Blogpost (alle Links unter ct.de/yfze) veröffentlicht, in dem sie schreiben, dass sie komplett geflutet werden mit Reports und Advisories und einen Backlog von mehreren Wochen haben.

Wenn eine Sicherheitslücke als kritisch eingestuft wird, ist natürlich davon auszugehen, dass sie priorisiert wird, aber das ganze System ist einfach hoffnungslos überlastet. Das schiere Volumen zwingt die Reviewer, sich zu beeilen. Aber wenn sie weniger Zeit für die Reviews haben, kann es womöglich leichter passieren, dass es Angreifern doch gelingt, Schadcode in die Sicherheitsupdates einzuschleusen.

Bild: Andrew Nesbitt

Andrew Nesbitt hält die Security-Probleme, die durch KI aufgeworfen werden, für ein Symptom des mangelnden Investments in Open Source.

Und wenn bei den Reviews LLMs eingesetzt werden, öffnet das wiederum Tür und Tor für Prompt Injections [1].

c't: Ist die Lage außer Kontrolle?

Nesbitt: Ich würde nicht sagen, dass wir an dem Punkt angelangt sind, dass die Situation außer Kontrolle ist. Aber die Lage ist insgesamt angespannt. Jeden Tag werden mithilfe von LLMs hunderte Sicherheitslücken in Open-Source-Software entdeckt, nicht nur in den populären Projekten, sondern gerade auch in denen, die zwar viel genutzt, aber bislang wenig beachtet wurden. Keines dieser Systeme, die bisher ein hinreichendes Sicherheitsniveau garantiert haben, ist dafür ausgelegt, diese Größenordnung zu bewältigen.

Wir sehen es an Project Glasswing: Die Mitgliedsfirmen finden zehntausende Sicherheitslücken in ihren Dependency-Trees. Manche dieser Unternehmen setzen 60.000 unterschiedliche Pakete über alle ihre Organisationen hinweg ein. Und jetzt fragen sie sich: „Wie um alles in der Welt sollen wir diese ganzen Sicherheitslücken und Bugs nach Priorität sortieren und veröffentlichen? Und wie sollen die Maintainer all die Fixes prüfen und Patches bereitstellen?“

Ein Problem ist auch, dass viele Open-Source-Projekte von einer einzelnen Person gewartet werden. Und viele dieser Maintainer machen das freiwillig, in ihrer Freizeit. Sie stehen jetzt unter enormem Druck. Angeführt vom cURL-Projekt haben einige Projekte den Juli dieses Jahres zum Pausenmonat deklariert, um die Maintainer zu entlasten. Sie nennen es Summer of Bliss (*siehe S. 20, Anm. d. Red.*).

c't: Das ist aber keine dauerhafte Lösung für das Problem.

Nesbitt: Daran, die Situation zu entschärfen, arbeite ich seit ein paar Monaten mit Alpha-Omega (*siehe S. 26, Anm. d. Red*). Alpha-Omega [IT-Sicherheitsprojekt unter dem Dach der Open Source Security Foundation/Linux Foundation, *Anm. d. Red.*] bezahlt Maintainer verschiedener Ökosysteme – darunter Perl, Java, Apache, Python, Ruby, Erlang, JavaScript – dafür, die Reports der Sicherheitslücken nach Dringlichkeit zu sortieren.

c't: Denken Sie, dass sich die Lage entspannt, wenn alle gefundenen Sicherheitslücken behoben sind?

Nesbitt: In unseren Meetings berichten die Maintainer, dass die Anzahl der eingehenden Meldungen in letzter Zeit rückläufig ist. Es kommen zwar immer noch welche rein, aber es handelt sich immer öfter um Duplikate. Das Grundrauschen dürfte also leiser werden.

Ich glaube aber nicht, dass es jemals wieder auf das Niveau von früher zurückgehen wird. Die Leute richten ihre LLM-Security-Scanner mittlerweile auf jeden einzelnen Commit. Dabei finden sie viele potenzielle Schwachstellen, die man sonst nicht entdeckt hätte, darunter viele mit geringer Tragweite.

Die hat man früher oft ignoriert, aber das Problem an dieser Stelle ist, dass die neueren LLMs so gut darin sind, diese Bugs so zu verketten, dass sie sich doch ausnutzen lassen. Sich allein auf die als „kritisch“ oder „hoch“ eingestuften Lücken zu konzentrieren, reicht also nicht.

c't: Wird wohl irgendwann der Punkt erreicht sein, an dem es keine Schwachstellen mehr gibt?

Nesbitt: Selbst wenn man an einem Punkt alle existierenden Bugs eliminiert hätte – die Leute veröffentlichen ja weiterhin Code und darin gibt es dann neue Schwachstellen. Und oft sind existierende Fixes für ein Problem ihrerseits lückenhaft. IT-Sicherheit ist eben komplex und gerade unerfahrene Softwareentwickler haben oft Schwierigkeiten, solche Probleme in Gänze zu verstehen.

c't: Immer mehr Softwareentwickler nutzen LLMs zum Programmieren [2]. Wenn die neueren KI-Modelle so gut darin sind, Schwachstellen aufzuspüren, sind sie dann nicht vielleicht auch gut darin, sicheren Code zu schreiben?

Nesbitt: Nur wenn man das explizit vorgibt. Es gibt bei Coding-Harnessen [Anwendung um das KI-Modell herum, die determiniert, wie sich ein Agent in der Praxis verhält, *Anm. d. Red.*] und LLMs keine Standardeinstellung, die dafür sorgt, dass sie sicheren Code schreiben. Sie sind viel eher darauf ausgerichtet, die Nutzer möglichst lange in ihrem Sog zu halten. Vielleicht ändert sich das irgendwann und dann priorisieren sie sicheren Code, aber bis solche Änderungen bei den Nutzern ankommen, können Monate oder Jahre vergehen.

Einzelne Entwickler können natürlich Security-Reviews explizit in ihre Prozesse integrieren, aber solange das nicht der Default ist, wird unsicherer Code der Default sein.

c't: Ende Juni haben Sie einen Blogpost über ein Projekt namens Scrutineer veröffentlicht, das im Rahmen Ihrer Zusammenarbeit mit Alpha-Omega entstanden ist. Soll das die Probleme lösen?

Nesbitt: Scrutineer ist ein quelloffenes KI-Harness für LLM-getriebene automatisierte Security-Audits, ähnlich wie bei Codex oder Claude Code, aber der Fokus liegt mehr auf einzelnen Paketen. Scrutineer schaut sich auch die Abhängigkeiten dieser Pakete an. Wenn es Schwachstellen findet, versucht es zu ermitteln, ob und inwieweit sie sich auf die Pakete auswirken können, die davon abhängen. So bekommt man einen besseren Eindruck vom „Blast Radius“ einer entdeckten Lücke. Dann kann man beispielsweise sagen, dass ein Fund zwar die Kriterien erfüllt, um als Sicherheitslücke gewertet zu werden, es in der Praxis aber keinen Weg gibt, sie auszunutzen. Die Maintainer würde man dann mit solchen Reports in Ruhe lassen.

Alpha-Omega hat IT-Sicherheitsexperten angestellt, die mit ihrer Expertise und ihren Verbindungen zu den Maintainern helfen sollen, die wichtigsten Pakete in ihren jeweiligen Ökosystemen abzusichern. Sie nutzen Scrutineer bereits.

c't: Ist KI also eher Segen als Fluch für die IT-Sicherheit?

Nesbitt: Es ist schwer, vorherzusagen, was passieren wird. Zumal wir nicht wissen, ob nicht in Kürze noch weiter verfeinerte Modelle auf den Markt kommen werden. Die sogenannten Open-Weight-Modelle hinken den Frontier-Modellen normalerweise etwa sechs Monate hinterher [Mit der Veröffentlichung von Kimi K3, die stattfand, nachdem dieses Interview geführt wurde, hat sich dieser Abstand mittlerweile deutlich verkürzt, *Anm. d. Red.*].

Ende des Jahres werden wir also möglicherweise Open-Weight-Modelle sehen, die Sicherheitsscans ähnlich gut wie Antrophics Mythos durchführen können. Das finde ich ziemlich beunruhigend. Wenn man die Guardrails beseitigen kann, kann man diese Modelle einfach auf Webseiten oder Softwaresysteme richten und sagen: „Finde ein paar Schwachstellen.“ Und wenn man genug Token-Budget oder genug Energie hat, um sie einfach immer und immer wieder laufen zu lassen, dann findet man irgendwann auch etwas. Deshalb herrscht da schon eine gewisse Panik. Alle, die das auch nur ein bisschen durchdenken, kommen zu dem Schluss, dass sie jetzt ihre Systeme absichern müssen.

c't: Das können sie nicht, weil die Aufgabe viel zu groß und kompliziert ist?

Nesbitt: Die vergangenen Monate waren wild. Es gab ein riesiges Aufkommen an Security-Problemen. Ich halte das für ein Symptom des mangelnden Investments in Open Source. Viele große Unternehmen haben in den letzten zehn Jahren intensiv von Open Source profitiert und wenig zurückgegeben. Es kommt sehr wenig Geld bei den einzelnen Projekten an. Gleichzeitig hängt die Sicherheit der Software-Supply-Chain von freiwilligen Maintainern ohne Verträge ab. Das rächt sich jetzt.

»Es herrscht schon eine gewisse Panik.«

Ich denke, künftig werden die Leute viel vorsichtiger damit sein, zu viele verschiedene Projekte von unbekannten Anbietern zu nutzen. Dependency-Trees werden schrumpfen und es wird eine Konzentration auf wenige sehr populäre Open-Source-Projekte geben.

Gleichzeitig werden Firmen versuchen, geprüfte, gehärtete Versionen dieser Projekte zu vermarkten. Mich beunruhigt das. Sie versuchen, einen Teil des Open-Source-Kuchens für sich zu beanspruchen, statt dafür zu sorgen, dass die Maintainer Unterstützung erhalten. Sie bauen eine alternative Version, weil Unternehmen viel lieber – fast wie bei einer Versicherung – dafür bezahlen, dass jemand anderes die Verantwortung übernimmt, als direkt die Wartung der Projekte zu finanzieren.

(kst@ct.de) c't

Literatur

[1] Kathrin Stoll, Programmierkollege KI, Co-Authored by Claude Code, c't 13/20226, S.16

[2] Sylvester Tremmel, Schädlinge im Prompt, Promptware: Angriffe jenseits der Prompt-Injection, c't 9/2026, S. 122

Blogbeiträge und Tools: ct.de/yfze

Hilfsangebote

Projekte wider die Vulnocalypse formieren sich

Bild: Martina Bruns / KI / heise medien

Dank KI kommen nicht nur Sicherheitslücken und Patches Schlag auf Schlag. Auch Initiativen, die helfen sollen, diese Arbeitslast zu verteilen und zu koordinieren, sprießen allerorten. Wir geben einen Überblick.

Von Sylvester Tremmel

Die „Vulnocalypse“ ist da. Bezahlte, unterbezahlte und unbezahlte Software-Maintainer ächzen unter der Last KI-induzierter Fehler- und Exploit-Meldungen, und die Sprachmodelle schicken sich an, auch weitergehende Angriffe zu automatisieren (siehe Seiten 14 und 20). Das betrifft zwar proprietäre Software ebenso wie Open-Source-Produkte, aber bei ersteren gibt es zumeist eine kommerzielle Entität, die für ihre Software verantwortlich ist und von Kunden für Fehlerbehebungen bezahlt wird.

Wer dagegen all die Open-Source-Software sichern soll und kann, ist weniger offensichtlich. Die gute Nachricht lautet: Das Problem ist erkannt und diverse Projekte und Organisationen treten an, um bei der Lösung zu helfen. Sogar Geld steht zur Verfügung; zum einen, weil KI-Firmen, durch deren Produkte das Problem entsteht, sich ihrer Verantwortung stellen wollen – oder zumindest diesen Anschein erwecken möchten. Jedenfalls sind sie höchst liquide und willens, Projekte zur Sicherung von Open Source zu finanzieren.

Zum anderen wird vielen Unternehmen gerade sehr schmerzhaft bewusst, auf welch umfangreichem Open-Source-Fundament ihre Produkte stehen. Beispielsweise rechnet IBM vor, dass sich über 90 Prozent der US-amerikanischen Fortune-500-Unternehmen auf Open-Source-Software verlassen. Alleine Anthropics „Mythos Preview“, ein Modell, das zumindest in der öffentlichen Wahrnehmung einen Wendepunkt in der Security-mit-KI-Debatte darstellt, habe fast 4000 schwerwiegende Sicherheitslücken in Open-Source-Software gefunden, schreibt IBM.

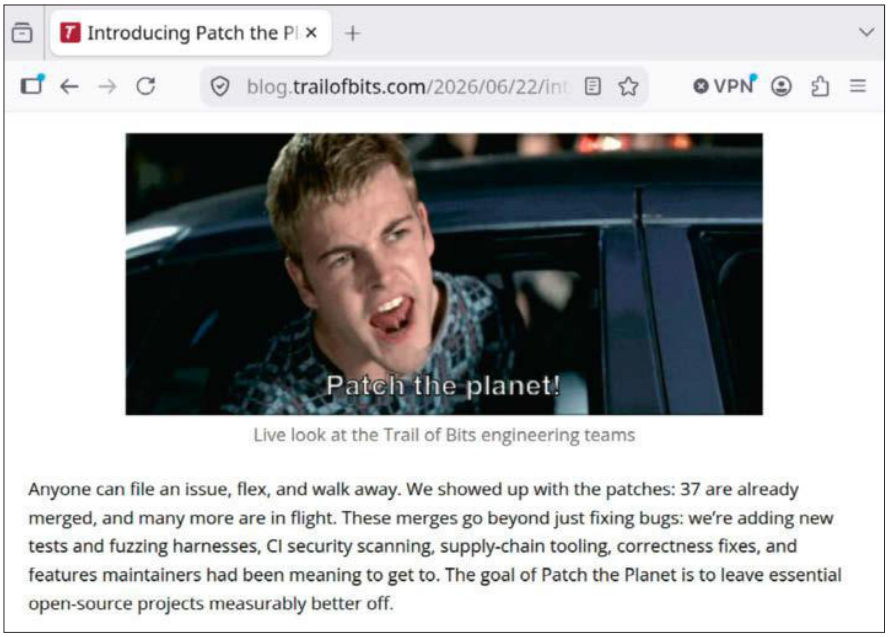


Bild: Trail of Bits

Motivation ist jedenfalls da: So sehen angeblich die Teams von Trail of Bits aus, wenn sie Open-Source-Software fixen.

IBMs Antwort auf das Problem heißt **Lightwell** (Links zu den erwähnten Projekten unter ct.de/yxqh). Im Zuge dieses Projekts setzt der Softwaregigant nach eigenen Angaben 5 Milliarden US-Dollar und 20.000 Softwareentwickler in Bewegung, um Sicherheitslücken zu fixen. Das Angebot ist auf industrielle Kunden zugeschnitten, die Software oft nicht aktualisieren können oder wollen; für sie behebt IBM in neuen und alten Versionen Lücken, zunächst mit einem Fokus auf das Maven/Java-Umfeld, hier sei der Bedarf am größten. Später sollen PyPI, npm, Go und andere Ökosysteme dazukommen.

IBM will die Patches auch den betroffenen Open-Source-Projekten zugutekommen lassen und deren Maintainer allgemein unterstützen. Das dürfte bitter nötig sein, damit die nicht vom Regen in die Traufe kommen und in einer Patch-Flut statt einer Fehlermeldungsflut ertrinken.

KI-Firmen finanzieren Bugfixing

Dieses Problem hat auch Anthropic erkannt. Sein – gleichzeitig mit Mythos Preview ins Leben gerufenes – **Project Glasswing** soll ebenfalls Open-Source-Software absichern. Im Zuge von Glasswing nutzt Anthropic seine Modelle, um wichtige Open-Source-Software auf Schwachstellen zu scannen. Die Firma gibt außerdem Glasswing-Teilnehmern Zugriff auf die Modelle, inklusive Freitoken im Wert von

100 Millionen US-Dollar. Die Idee dahinter ist, dass Mythos & Co. nicht nur Fehler finden, sondern sie auch bestätigen und sinnvoll patchen können. Wer diese Patches dann einpflegen soll, ist jedoch bislang nicht ganz klar. In einem Update zu Glasswing schreibt Anthropic, man arbeite an Ideen, wie man Open-Source-Betreuern die Triage und Bearbeitung solcher Berichte erleichtern könnte.

Jedenfalls hat sich Anthropic dafür Partner ins Boot geholt: Im Rahmen von Glasswing flossen 2,5 Millionen US-Dollar an Alpha-Omega und 1,5 Millionen an die Apache Software Foundation (ASF). Die ASF hat damit die **Responsible AI Initiative** ins Leben gerufen, die über mindestens drei Jahre und mit insgesamt 10 Millionen US-Dollar ASF-Projekte dabei unterstützen soll, mit den KI-bedingten Umwälzungen fertig zu werden.

Der große Konkurrent OpenAI ist ebenfalls nicht untätig. Dort heißt das übergeordnete Projekt **Daybreak** und gibt Open-Source-Projekten – ähnlich wie bei Anthropic – unter anderem kostenlos Zugriff auf Codex Security (ehem. Aardvark), OpenAIs KI-basierten Helfer zum Finden und Beheben von Sicherheitslücken. Weil der ebenfalls überlasteten Maintainern zwar Werkzeuge an die Hand gibt, aber keine Arbeit abnimmt, hat sich auch OpenAI Partner gesucht und unter anderem die renommierte Security-Firma Trail of Bits für ihre Sache gewonnen. Diese hat das Daybreak-Unterprojekt **Patch the Planet** ausgerufen: Teilnehmende Projekte werden von Trail-of-Bits-Experten untersucht und erhalten aus der Analyse resultierende Patches.

Außerdem spendet auch OpenAI an das bereits erwähnte **Alpha-Omega**. Dabei handelt es sich um ein Projekt der Open Source Security Foundation (OpenSSF), die wiederum zur Linux Foundation gehört. Alpha-Omega gibt es schon seit 2022, das Projekt verfügt über ein jährliches Budget von über 7 Millionen US-Dollar und soll allgemein die Sicherheit von Open-Source-Software verbessern. Auch der Open-Source-Experte Andrew Nesbitt gehört Alpha-Omega an und hat unter anderem „Scrutineer“ in diesem Rahmen entwickelt (siehe S. 22).

Im Zuge der aktuellen KI-Umwälzungen erhielt Alpha-Omega eine Finanzspritze von insgesamt 12,5 Millionen US-Dollar, an der sich neben Anthropic und OpenAI auch AWS, GitHub, Google und Microsoft beteiligten. Ein Teil des Geldes dürfte in **Akrites** fließen, ein weiteres Projekt der Linux Foundation, dessen Anschubfinanzierung von Alpha-Omega stammt. An Akrites beteiligen sich neben den oben erwähnten Geldgebern noch diverse weitere Organisationen von Cisco und IBM über Citi, JPMorganChase und Nvidia bis zur Rust Foundation.

Akrites ist der Versuch einer zentralen Koordination, sodass Glasswing, Lightwell und andere Projekte ihre Funde an einer Stelle abgeben können. Akrites soll als gemeinsames Security Incident Response Team (SIRT) für Open-Source-Projekte agieren und die Erkennung, Behebung und Offenlegung von Schwachstellen in den Open-Source-Projekten „in dem Tempo koordinieren, mit dem KI-gestützte Angreifer jetzt agieren“.

Zur Not soll Akrites auch als „Maintainer of Last Resort“ auftreten. So bezeichnet man üblicherweise Organisationen, die einspringen, wenn ein weitverbreitetes Stück Software eine Sicherheitslücke aufweist, aber keinen Maintainer (mehr) hat, der die Lücke schließen könnte. Akrites kündigt an, diese Rolle zu übernehmen, „wenn ein kritisches Paket keinen aktiven Maintainer hat“, und betont, es gehe darum, dass eine Fehlerbehebung alle rechtzeitig erreichen könne.

Unerwünschte Hilfe

Dieser Fokus auf schnelle Reaktionen lässt manche Entwickler offenbar befürchten, Akrites könnte sich schon dann zum Maintainer of Last Resort aufschwingen, wenn ein Projekt aus Akrites’ Sicht nicht schnell genug reagiert. Das weitverbreitete Projekt pkgconf sah sich zu einer umfangreichen Ergänzung seines Code of Conduct veranlasst, der nun ausdrücklich davor warnt, sich ohne Zustimmung der pkgconf-Maintainer selbstständig zum Maintainer of Last Resort zu befördern. Wer wolle, könne das Projekt freilich forken und den Fork selbst warten – aber nur unter einem anderen Namen.

Dergleichen dürfte Akrites wohl nicht beabsichtigen, aber es zeigt, welches Risiko die zahlreichen inzwischen geschaffenen Hilfsorganisationen bergen: Falls sie sich schlecht untereinander koordinieren, könnten sie die Last auf den Schultern der Maintainer noch weiter vergrößern. Aber auch wenn sie ihre Hilfsarbeit gut koordinieren, könnten sie Projekte unter Druck setzen – weil dann ein organisiertes, gut finanziertes und schnell agierendes SIRT schlimmstenfalls bei völlig überlasteten Einzelpersonen anklopft. Erste Beobachter fühlen sich an die xz-Hintertür erinnert [1]. Deren Urheber konnte zuschlagen, weil ein überlasteter Maintainer einerseits vorgeblichen Community-Druck zu spüren bekam und ihm gleichzeitig die Unterstützung eines vermeintlich wohlmeinenden Gehilfen angetragen wurde.

Fazit

Die Retter vor der Vulnocalypse müssen aufpassen, dass sie nicht aus einer Fehler- und Patch-Flut eine Flut von Hilfsinitiativen machen, die Entwickler ebenfalls überfordert. Doch wenn die Helfer sich gut organisieren und die gerade freimütig angebotenen Mittel sinnvoll nutzen, könnte aus all dem eine Open-Source-Landschaft hervorgehen, die koordinierter und sicherer als je zuvor ist. *(syt@ct.de)* c't

Literatur

[1] Sylvester Tremmel, Sprengladung angebracht, Eine Analyse der xz-Hintertür, Teil 1, c’t 16/2024, S. 134

Erwähnte Projekte und Initiativen:
ct.de/yxqh

Freie KI für alle

Nicht jedes KI-Problem von Maintainern resultiert aus der Bug- und Patch-Flut. KI-Systeme werden auch immer besser darin, selbst als Angreifer aufzutreten, was die Verteidiger vor neue Probleme stellt, unter anderem weil diese Art Angreifer Entscheidungen in der laufenden Attacke mit „machine speed“ trifft.

Eine naheliegende Maßnahme ist, auch auf Verteidigerseite KI einzusetzen. Als OpenAIs KI Huggingface angriff, mussten die Verteidiger allerdings feststellen, dass ihre KI sich immer wieder weigerte, bei der Verteidigung zu helfen. Das verstoße gegen Sicherheitsrichtlinien (siehe S. 14). Huggingface griff letztlich zu einem offenen KI-Modell aus China, um die laufende Attacke zu analysieren.

In diese Kerbe schlägt nun Nvidia mit der **Open Secure AI Alliance**. Sie soll „sicherstellen, dass Verteidiger allerorten auf offene, zukunftsweisende Werkzeuge zugreifen können, denen sie vertrauen und die sie kontrollieren können“. Innerhalb von Tagen schwoll die Allianz auf weit über 100 Mitglieder an, ein Who’s who von Tech-Unternehmen – mit den auffälligen Ausnahmen Anthropic und OpenAI. Deren Modelle sind zwar führend, aber alles andere als offen.