Wie man seine IT vor Spectre und Meltdown schützt

Gespensterjagd

Kurt Garloff

Zum Jahreswechsel machten gravierende Sicherheitslücken bei vielen Systemverantwortlichen die über die Festtage getankte Erholung zunichte. Ein Wrap-up für Administratoren.

Noch nicht ganz aus dem Weihnachtsurlaub zurück, war es für viele IT-Mitarbeiter vorbei mit der Erholung: Ein paar Tage vor dem geplanten Datum wurde am 3. Januar eine neue Klasse von Sicherheitslücken publik. Sicherheitsforscher der TU Graz, von Googles Project Zero und andere hatten diese 2017 entdeckt und im Juni vergangenen Jahres die betroffenen Prozessorhersteller informiert: vorneweg Intel, aber auch ARM, AMD, IBM und andere.

Vereinfacht gesagt liegt das Problem darin, dass die meisten auf Performance optimierten modernen CPUs Maschinenbefehle im Voraus spekulativ ausführen. Das bringt einen hohen Leistungsgewinn, denn die CPU ist im Vergleich zu den anderen Komponenten sehr schnell. Benötigt sie etwa ein Datum aus dem Hauptspeicher, das nicht im Cache liegt, wartet sie schon einmal über 200 Taktzyklen. Oder sie verbringt ihre Zeit mit Sinnvollerem, etwa einem anderen Hyperthread oder eben dem spekulativen Ausführen von Befehlen, die eigentlich später dran wären. Das passiert versteckt im Prozessor, das Programm soll davon nichts merken. Meistens geht das gut, und kommt das Datum endlich aus dem Speicher, sind schon Dutzende weitere Schritte fertig und können offiziell sichtbar gemacht werden. Diese Vorausberechnung heißt Out-of-Order-Architektur (OoO).

Natürlich verspekuliert sich der Prozessor hin und wieder. Dann macht er alles wieder rückgängig: Er verwirft die Ergebnisse und versetzt alle Register wieder in den Zustand, als hätte die Fehlspekulation nie stattgefunden. Das wäre kein Sicherheitsproblem, hätten die CPU-Designer nicht übersehen, dass das messbare Seiteneffekte hat. So landet Speicher im Cache – und der wird nicht aufgeräumt. An die Daten im Cache kommt man nicht direkt ran, sehr wohl lässt sich aber an der Geschwindigkeit des Zugriffs leicht ablesen, ob eine Adresse im Cache ist oder nicht. Hängt diese von spekulativ genutzten Daten ab, kann man auf Letztere rückschließen.

Die Fehlspekulationen haben also Nebeneffekte, die nicht sauber aufgeräumt werden. Fatal dabei: Es können auch Daten sichtbar werden, auf die der Code gar nicht hätte zugreifen dürfen. Auch die Berechtigungsprüfungen der CPUs werden nämlich erst mal übersprungen, denn wenn dies unrechtmäßig war, wird ja aufgeräumt – nur nicht vollständig.

Spectre und Meltdown

Die Sicherheitsforscher haben sich drei Angriffsszenarien ausgedacht und sie erforscht. Zwei davon hören auf den Namen Spectre (Gespenst), eines auf den Namen Meltdown (Kernschmelze). Die Grafik zeigt potenzielle Angriffswege.

Die Spectre-Variante 1 (Spectre-1) erlaubt das Umgehen klassischer Längenchecks. Während die CPU für die Längenprüfung noch auf ein Datum wartet, liest sie im Hintergrund spekulativ Daten jenseits der Feldgrenze. Werden diese jetzt noch zu einer Adressberechnung genutzt, landen Adressen im Cache, abhängig von Daten, die nicht hätten gelesen werden sollen. Dieses Szenario kommt relativ häufig vor – um es auszunutzen, muss der Angreifer aber Code ausführen können, der die Speichergeschwindigkeit der fraglichen Adressen misst.

Die Forscher haben am Beispiel von JavaScript gezeigt, dass die Lücke ausnutzbar ist. Der Interpreter macht eigentlich alles richtig und verbietet dem Programm, Daten jenseits seiner Grenzen zu lesen; aber durch den CPU-Fehler kommt es an geschützte Daten: die vom Browser selbst und die von anderen Browserfenstern. Hoffentlich keine mit einem wichtigen Passwort!

Ausnutzbar dürfte die Lücke in Interpretern und JIT-Umgebungen sein. Gefahr besteht, sobald potenziell nicht vertrauenswürdiger Code in vermeintlich geschützter Umgebung läuft. Das Lesen von Daten beschränkt sich immerhin auf die im Adressraum des betroffenen Prozesses – oder eben des Kernels, wenn ein solcher Interpreter dort existiert. In modernen Linux-Kernels ist dies mit eBPF der Fall – auch hier konnten die Forscher den CPU-Bug erfolgreich ausnutzen.

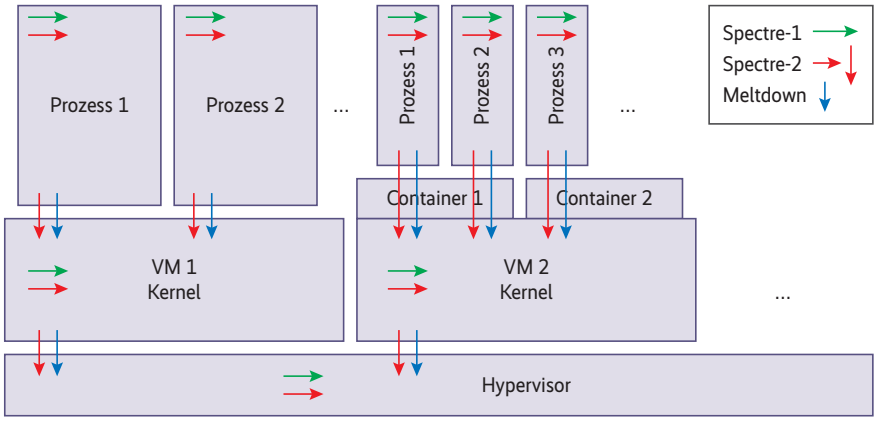
Schlimmer noch kommt es mit der zweiten Variante, Spectre-2. Hier erfolgt der Angriff, indem man den Sprungadressenzwischenspeicher (BTB, Branch Target Buffer) manipuliert. Durch geschickte Sprünge schiebt man der CPU einen kontrollierbaren Wert in den BTB. Beim nächsten Sprung springt die CPU erst mal spekulativ an diese Adresse. Natürlich bemerkt sie etwas später den Irrtum, räumt aber wieder nicht richtig auf. Der Angriff ist schwierig, da er viel Detailwissen über

die Eigenschaften des BTB erfordert. Dafür hat er dramatischere Auswirkungen: So kann ein unprivilegiertes Programm mit normalen Nutzerrechten den BTB manipulieren und anschließend per SysCall den Betriebssystemkern aufrufen. Die Kernel-Funktion springt dann spekulativ an eine vom Angreifer injizierte Adresse und er kann die Cache-Effekte nutzen, ein paar Bits Kernel-Speicher auszulesen. Mit derselben Methode kann eine Gast-VM den Hypervisor austricksen und dessen Speicher lesen.

Beim dritten Angriff, Meltdown, greift das Schadprogramm direkt auf eigentlich geschützte Daten zu. Natürlich untersagt der Prozessor das. Aber bis er es merkt, sind bei geschickter Programmierung viele Zyklen vergangen, in denen er munter spekulativ weiterrechnet und datenabhängige Cache-Spuren hinterlässt. Gravierend dabei: Meltdown trickst den normalerweise durch die Hardware erzwungenen Speicherschutz aus. Ein Prozess kann auf den Kernel-Speicher zugreifen und somit auch auf allen Speicher anderer Prozesse unter der Kontrolle desselben Kernels. Dafür ist der Angriff vergleichsweise einfach durchzuführen.

Grundsätzlich wäre vom Gastsystem aus per Meltdown auch der Hypervisor angreifbar. Für Xen im paravirtualisierten Modus trifft das zu, bei voller Virtualisierung mit Hardwareunterstützung (VT-x, Xen-HVM-Modus, KVM) jedoch nicht: Hier hat jedes Gastsystems eigene Adressbereiche und immerhin greift die CPU dann nicht über EPT spekulativ auf den Hypervisor-Speicher zu.

Die Tabelle „Gefahrenpotenzial“ auf Seite 64 zeigt, dass fast alle modernen Out-of-Order-CPUs von mindestens einer der Lücken betroffen sind: alle aktuelleren Intel-x86- (außer Atom vor Silvermont 2013), die ARM-Big-Core- (Cortex-A8, A9, A15, A57, A72, A73, A75) und AMD-Prozessoren. Die Sicherheitsforscher hatten AMD-CPUs nicht erfolgreich mit Spectre-2 angegriffen, weshalb AMD seine CPUs zunächst als sicher einstufte, was man mittlerweile zurückziehen musste. Immerhin sind AMD-CPUs wohl immun gegen Meltdown, auch sind nicht alle ARM-Big-Cores betroffen.



CPU-Fehler eröffnen Angriffsszenarien bis zur untersten Systemebene.

Eine Abschätzung der Risiken

Am Anfang steht eine Risikoanalyse. In allen drei Szenarien kann ein Angreifer unberechtigt Speicher auslesen und so an schutzwürdige Daten kommen: Passwörter, Schlüssel, Kundendaten et cetera. Dazu muss er Code auf der angegriffenen Maschine ausführen können. Diese Einschränkung ist aber nicht so groß wie vielleicht erhofft: Code wird ja oft ohne große Überprüfung in (vermeintlich) geschützten Umgebungen ausgeführt, etwa der JavaScript-Umgebung im Browser.

Laufen auf einem System VMs mit verschiedenen Eigentümern, ist sicherlich die Abschottung vordringlichste Aufgabe. Spectre-2 ist hier ein großes Risiko. Wer Container sauber separieren muss oder es mit potenziell bösartigen Prozessen zu tun hat, muss sich zusätzlich um Meltdown kümmern, zumal sich dieser CPU-Fehler vergleichsweise einfach ausnutzen lässt.

Wer sich auf geschützte Laufzeitumgebungen verlässt, tut gut daran, Spectre-1 ebenfalls intensiv zu beobachten und Patches an den Interpretern/JITs einzuspielen. Während einige Kernel-Patches gegen Spectre-1 im Umlauf sind, haben Analysen an Xen und KVM bislang keine per Spectre-1 ausnutzbaren Codefragmente gefunden. Auf Virenscanner sollte man nicht hoffen – die Angriffe hinterlassen keine Spuren (es wird ja nichts verändert) und sind nur schwer zu erkennen.

Für die drei bekannten Angriffe sind Workarounds in Software möglich. Allerdings sind diese aufwendig, da sie für das Umgehen der CPU-Fehler umfangreiche Umbauten erfordern.

Impfen von Umgebungen ...

Gegen Spectre-1 empfiehlt Intel, nach sicherheitskritischen Grenzabfragen eine *lfence*-Instruktion einzufügen. Diese vermeidet Spekulation, solange die Abfrage nicht geklärt ist. Der Compiler könnte diesen Befehl großzügig verteilen – allerdings würde dies die Codeausführung deutlich bremsen. Insofern muss man die Notwendigkeit durch Reviews oder Codeanalysetools feststellen. Intel hat wohl Partnern entsprechende Profile für das Tool *coverity* zur Verfügung gestellt.

Der Linux-Kernel hat von Herstellern wie SUSE und Red Hat aufgrund dieser Analysen fünf Dutzend Patches erhalten. Insbesondere dichteten die Entwickler den eBPF-Code gegen die CPU-Lücken ab. Die Empfehlung, *sysctl kernel.unprivileged_bpf_disable 1* zu setzen, würde der Autor aber erst mal aufrechterhalten.

Weitere Software-Patches dürften nach und nach kommen. Erste Verbesserungen für die JavaScript-Engines sind bei Google Chrome und Firefox bereits erfolgt. Beide empfehlen, die Site beziehungsweise Firstparty-Isolation zu nutzen und damit verschiedene Browsertabs in eigene Prozesse auszulagern und somit sicherer zu trennen. Mozilla will weiterhin die Messgenauigkeit von Zeitmessungen verringern – dies erschwert die Messung, ob Adressen im Cache sind oder nicht.

Eine Verteidigung gegen Meltdown wurde in Linux seit vielen Monaten vorbereitet. Die Kernel-Verantwortlichen ent-

iX-TRACT

- Über die neu entdeckten Prozessordesignfehler lassen sich geheime Daten ausspähen und dabei Sicherheitsbarrieren überwinden.
- Da noch keine neuen CPUs in Sicht sind, müssen Nutzer die Probleme mit nicht immer ausgereifter Software umgehen – auf Kosten der Leistung.
- Dennoch ist es in den meisten Umgebungen unvermeidlich, diese Workarounds schleunigst einzuspielen; Infrastrukturbetreiber tun gut daran, sich auf das kurzfristige Einspielen weiterer Patches einzustellen.

Gefahrenpotenzial

Szenario	CVE-2017	Kurzbeschreibung	Angriffsziele	Komplexität	betroffene CPUs
Spectre-1	-5753	Umgehung von Grenzen	Interpreter	mittel	alle OoO
Spectre-2	-5715	Sprungadressen-manipulation	Interpreter, Kernel, Hypervisor	hoch	OoO-Intel, AMD, ARM-Big-Cores
Meltdown	-5754	unberechtigter Datenzugriff	Kernel	mittel	OoO-Intel, ARM

wickelten ein Patchset der TU Graz namens KAISER weiter und machten es fit für die Integration in den Kernel. Unter dem Namen KPTI (Kernel Page Table Isolation) wollten sie damit ursprünglich Angriffe gegen die Kernel-Adressverwürfelung (kASLR) abwehren. Der ungewöhnliche Zeitpunkt der Aufnahme der Änderungen löste bei Kernel-Kennern einige Fragen aus – als dann noch Ankündigungen einiger namhafter amerikanischer Cloud-Provider über groß angelegte Patch-Installationen folgten, kamen Sicherheitsforscher der wahren Ursache gefährlich nahe. Dies führte letztlich dazu, dass die Lücken ein paar Tage vor dem geplanten Datum veröffentlicht wurden.

Aufgrund des Vorlaufs sind die KPTI-Patches gegen Meltdown gut ausgereift. Neben den in 4.15-rc6 und 4.14.11/12 eingeflossenen gab es auf älteren KAISER-Varianten beruhende Backports für die Langzeit-Kernel 4.9 und 4.4. Der KPTI-Einsatz führt aber zu mehr Overhead bei Systemaufrufen und Interrupts.

Am schwierigsten gestaltet sich das Umschiffen von Spectre-2. Intel stellt für die meisten CPUs der letzten Jahre mittlerweile Microcode-Updates bereit. Diese „CPU-Firmware“ lässt sich über das BIOS/UEFI oder das Betriebssystem aktualisieren – unter Linux per Hotplug-Mechanismus *echo 1 > /sys/devices/sytem/cpu/microcode/reload* oder altmodisch per *microcode_ctl –u*. Mit dem neuen Microcode lässt sich über zwei maschinenspezifische Register die Spekulation indirekter Sprünge kontrollieren. Die relativ überschaubaren Änderungen an Kernel (IBRS) und Hypervisor können damit nach derzeitigem Kenntnisstand Spectre-2 verhindern; leider ist dieses Vorgehen aber mit Leistungseinbußen verbunden, die diejenigen von KPTI noch mal deutlich in den Schatten stellen.

Intel kämpft auf einigen CPUs noch mit der Stabilität der Microcode-Updates. Weiter gibt es andere Architekturen (etwa ARM) und ältere betroffene Prozessoren. Für diese arbeiten Google-Entwickler derzeit an einer Alternative zum Umgehen des Problems: Die realisiert indirekte Sprünge über Schreiboperationen auf den Stapelspeicher und ein *return* – dabei ist der BTB nicht so leicht auszutricksen. Diese sogenannten *retpoline*-Patches funktionieren auf vielen CPU-Typen und häufig sogar schneller als der Ansatz über den Microcode – allerdings nicht auf CPUs der Skylake-Generation.

Red Hat und SUSE wussten im Vorfeld Bescheid und haben direkt am 4. Januar Kernel- und Microcode-Updates bereitgestellt; beide haben viele Patches gegen Spectre-1, nutzen IBRS plus den Microcode (soweit verfügbar) gegen Spectre-2 und KPTI gegen Meltdown.

Auch Microsoft und Apple waren im Vorfeld informiert. Microsoft hat ebenfalls am 4.1. ein Update ausgeliefert. Allerdings wird das nur gefunden, wenn Virenscanner über einen Registry-Eintrag ihre Kompatibilität mit der Änderung erklären. Und selbst dann müssen Anwender die Patches mit speziellen Registry-Einträgen aktivieren. macOS erhielt bereits Anfang Dezember Patches gegen Meltdown. Google hat nach eigener Aussage Patches gegen Meltdown und Spectre im Android-Sicherheitsupdate 2018-01-05 integriert.

Wer eine der verwundbaren CPUs einsetzt, hat eigentlich nicht viele Alternativen zum Einspielen der Patches. Nur wer sich ganz sicher ist, dass auf einem System niemals nicht vertrauenswürdiger Code ausgeführt wird, kann erwägen, das Risiko erst mal in Kauf zu nehmen.

Zum Glück sind die Patches gegen Spectre-1 meist unproblematisch. Der Autor kennt weder Berichte, wo sie Probleme verursachten, noch Benchmarks mit signifikanten Leistungseinbußen. Schwieriger wird das bei den zum Meltdown-Schutz angewandten KPTI/KAISER-Patches.

… und Nebenwirkungen einiger Patches

Der Linux-Kernel wurde sorgfältig darauf optimiert, bei einem Systemaufruf (und bei Interrupts) den Adressraum nicht umschalten zu müssen – nicht nur, weil das Umschalten ein paar Hundert Zyklen Zeit kostet, sondern auch weil der Cache für die Speicherverwaltung (TLB) dabei verloren geht, was noch mal viele Hundert Zyklen Zeitverlust bedeutet. Letzteres ist immerhin mit aktuellen CPUs (Intel seit Westmere, nicht bei Atoms) vermeidbar: Sie unterstützen Adressraum-IDs (Process Context ID – PCID – genannt), sodass sich TLB-Einträge einem Adressraum zuordnen lassen und somit beim Wechsel nicht automatisch hinfällig werden.

Mit den KPTI-Patches gegen Meltdown steigt der Overhead für einen Systemaufruf von rund 250 CPU Taktzyklen auf circa 1000 – wenn denn PCID funktioniert und aktiv ist, sonst doppelt so viele. Manche Virtualisierungsumgebungen sind so konfiguriert, dass sie PCID nicht an die Gäste weiterreichen. Volle PCID-Unterstützung nutzt Linux erst seit dem Kernel 4.14; die Backports für 4.9 und 4.4 (und die von SUSE und Red Hat auf 3.12 und 3.10) beherrschen aber genug PCID, um beim Systemaufruf keinen TLB-Verlust zu erleiden, und verhindern damit auch die doppelten Kosten.

Die Leistungseinbußen hängen stark von der Anwendung ab. Für reale Anwendungen hat Phoronix Rückgänge bis zu 30 % gemessen (siehe ix.de/ix1802062). Das Komprimieren von Videomaterial mag gar keinen messbaren Unterschied vorweisen, während ein Programm, das viele kleine Netzwerkpakete verschickt, eher auf der 30-Prozent-Seite liegt. Im Durchschnitt kann man für KPTI von Leistungsverlusten unter 10 % ausgehen.

Problematischer sind die Spectre-2-Patches. Die neuen Register erlauben eine feine Steuerung der indirekten Sprungvorhersage. In den Kernel-Updates von Red Hat sind die Parameter *ibrs* und *ibpb* direkt zur Laufzeit änderbar – der Red-Hat-3.10er-Kernel eignet sich daher gut für Messungen, auch wenn er etwas älter ist.

Der grundlegendste Schutz – zu verhindern, dass ein Prozess nach einem Kontextwechsel den vom vorherigen Prozess möglicherweise manipulierten BTB bekommt – lässt sich per IBPB einstellen. Die Kosten dafür sind zum Glück gering und waren mit dem Benchmark des Autors nicht sichtbar.

Anders ist das beim vollständigen Schutz des Kernels gegen BTB-Manipulationen (IBRS). Auf dem gemessenen System (Intel Xeon E5-2667 v4 Bare Metal) beträgt der Overhead für Systemaufrufe mit dem Red-Hat-Kernel ohne die Sicherheitsfeatures 250 Zyklen, 1000 mit KPTI, 1010 mit KPTI + IBPB und 3470 mit KPTI + IBPB + IBRS = 1.

Mit vollem Spectre-2-Schutz erhöht sich der Leistungsverlust also noch mal deutlich im Vergleich zum reinem Meltdown-Schutz. Derzeit liegen nicht genug

Messungen für die Auswirkungen auf reale Anwendungsfälle vor – aber es ist zu befürchten, dass der volle Schutz gegen Spectre-2 per IBRS durchschnittlich Leistungsverluste weit über 10 % bewirkt.

Das trifft leider alle – nur Nutzer mit entsprechendem Expertenwissen können natürlich eine Risikobewertung vornehmen und gegebenenfalls nur *kpti* und *ibpb* einschalten sowie mit *ibrs=0* leben und nur ein Viertel bis ein Drittel des Leistungsverlustes erleiden.

Intel hat Performancewerte für Desktop-Systeme (Windows) mit SYSMark ermittelt und dort einen Leistungsverlust von knapp 10 % gemessen – wie weit der Schutz gegen Spectre-2 dort aktiv war, ist leider nicht ganz klar (siehe ix.de/ix1802062).

Ein Blick in die Glaskugel

Bezüglich Spectre-2 ist die Situation im Fluss, noch sind nicht alle Microcode-Updates in stabiler Fassung verfügbar. Auch hat die Linux-Kernel-Community die dazu passenden Änderungen (IBRS-Patches) noch nicht angenommen. Da weder IBRS noch die *retpolines* schon überall funktionieren und die Leistungsverluste hoch sind, sind hier noch einige Verbesserungen zu erwarten. Patches für sicherere Interpreter und JIT-Umgebungen dürften nach und nach kommen; das zeitnahe Einspielen ist sicherlich ratsam.

Auch sind die Angriffsmöglichkeiten gegen CPUs über spekulative Ausführung vermutlich noch nicht ausgeschöpft. Insofern sind weitere Angriffsszenarien jenseits der drei bekannten zu erwarten. Mit etwas Pech benötigen diese wieder neue Umbauten in Software, die weitere Patch-Orgien und Leistungsverluste nach sich ziehen werden.

Den Siegeszug von Infrastrukturautomatisierung wird das nicht dauerhaft bremsen; zu groß sind die Vorteile der softwaregesteuerten Infrastruktur in Cloud- und Containerumgebungen. Aber vielleicht erfahren dedizierte Umgebungen in Public Clouds stärkere Nachfrage und Firmen betonen den privaten Teil in einer hybriden Cloud-Strategie stärker als zuvor geplant.

Eine grundlegende Lösung für die Probleme wäre natürlich eine neue CPU-Generation, die mit spekulativer Ausführung etwas vorsichtiger ist (und beispielsweise vorab gelesene Daten als riskant markiert und erst dann spekulativ weiter benutzt, wenn der zugehörige Zugriffscheck erfolgreich war) und bei Fehlspekulation auch sorgfältiger wieder aufräumt sowie Caches stärker trennt.

Sobald solche CPUs verfügbar sind, wird wohl ein beschleunigter Austausch stattfinden. Angesichts der Vorlaufzeiten in der Prozessorentwicklung ist aber nicht damit zu rechnen, dass dies noch in größerem Stil im Jahr 2018 geschieht.

Fazit

Die Designfehler der modernen Out-of-Order-CPUs sind ein kleines Erdbeben in der IT-Industrie. Obwohl einige Firmen über ein halbes Jahr Vorlauf hatten, standen Anfang Januar nicht für alle Probleme stabile Lösungen parat – und die kommen derzeit in Form von Workarounds in Software, die mit signifikanten Leistungsverlusten verbunden sind.

Für IT-Umgebungen, die nicht vertrauenswürdigen Nutzern ausgesetzt sind, gibt es keine Wahl, als schnellstmöglich die verfügbaren Patches einzuspielen. Dass damit Leistungseinbußen und schlimmstenfalls sogar temporär Instabilitäten zu befürchten sind, muss man dort in Kauf nehmen. Mit weiteren Patches ist zu rechnen – sowohl Verbesserungen der derzeitigen Umbauten als auch Umgehungen für noch zu entdeckende Angriffsvarianten.

Nur für gut geschützte Systeme, an die niemand herankommt, können IT-Verantwortliche kurzfristig eine andere Abwägung treffen, wenn sich das dortige Sicherheitsmodell nicht so sehr auf die nun beschädigten Schutzmechanismen innerhalb der Server stützt.

In jedem Fall ist davon auszugehen, dass die CPU-Fehler die IT-Industrie weiter auf Trab halten werden. Deutlich im Vorteil sind Nutzer mit einem kontinuierlich funktionierenden Softwareentwicklungs- und -installationsprozess, der es ihnen erlaubt, Patches schnell zu testen und auszurollen.

Dieselbe Überlegung gilt auch für Infrastrukturbetreiber. Während die drei größten amerikanischen Cloud-Anbieter den Luxus einer Vorabinformation bekamen, mussten Betreiber eigener Infrastruktur sowie die kleineren Anbieter Anfang Januar zeigen, wie gut sie darauf vorbereitet sind, ohne Vorwarnung auf solche Szenarien schnell und effektiv zu reagieren. Nutzern sei geraten, dies genau zu beobachten und Partner, Lieferanten und interne Einheiten so auszusuchen, dass man auf weitere böse Überraschungen gut reagieren kann. (avr@ix.de)

Kurt Garloff

ist seit Langem in der Linux- und der OpenStack-Community aktiv und ist als Chefarchitekt für die Open Telekom Cloud verantwortlich, eine von T-Systems angebotene und betriebene Public Cloud auf OpenStack-Basis.

Alle Links: www.ix.de/ix1802062