Engineer

Benutzer

Person oder für sie handelnder Prozess mit Systemzugang

Was es ist

Ein Benutzer ist jemand oder etwas, für den beziehungsweise das ein System handelt. In gewöhnlichen technischen Texten erfüllt das Wort drei verschiedene Aufgaben. Die meiste Verwirrung um Benutzer entsteht, wenn diese Aufgaben ineinander übergehen.

Die erste Bedeutung ist die Person: jemand, der ein Produkt öffnet und etwas erledigen möchte. Designer, Supportmitarbeiter und Produktmanager meinen meist diese Bedeutung; innerhalb des Systems hat sie überhaupt keine feste Form. Die zweite ist das Konto: der Datensatz, den ein System speichert, um jemanden wiederzuerkennen, meist mit einem Anmeldenamen, einer internen Kennung, einem Satz Berechtigungen und einem Status wie aktiv oder gesperrt. Der meiste Anwendungscode meint diese Bedeutung. Die dritte ist der Betriebssystembenutzer: auf Unix-ähnlichen Systemen ein Eintrag in der Benutzerdatenbank mit einem Benutzernamen und einer numerischen Benutzerkennung. Laufende Prozesse führen Benutzerkennungen mit, die der Kernel zur Prüfung der Zugriffsrechte verwendet.

Die drei passen selten eins zu eins zusammen. Eine Person kann Konten bei vielen Diensten haben. Die NIST-Richtlinien zur digitalen Identität sagen ausdrücklich, dass eine Person mehrere digitale Identitäten besitzen kann, von denen jede nur innerhalb ihres eigenen Dienstes eindeutig ist. Ein Konto kann von mehreren Personen geteilt werden, was fast immer ein Fehler ist. Und sehr viele Benutzer haben überhaupt keine Person hinter sich: Webserver, Datenbank, Backupjob und Bereitstellungsskript laufen jeweils unter einem eigenen Betriebssystembenutzer und haben oft zusätzlich Konten in anderen Systemen. Deshalb lautet die NIST-Definition „eine Person oder ein (System-)Prozess, der im Auftrag einer Person handelt“.

Auf einem Unix-Host lässt sich die Trennung zwischen Name und Nummer leicht erkennen:

```
$ id ana
uid=1001(ana) gid=1001(ana) groups=1001(ana)
```

Der Kernel arbeitet mit der Nummer. POSIX definiert eine Benutzerdatenbank, die für jeden Benutzer den Benutzernamen, die numerische Benutzerkennung, eine anfängliche Gruppenkennung, ein anfängliches Arbeitsverzeichnis und ein zuerst auszuführendes Programm enthält. Dateieigentum wird als Nummer gespeichert; der Name wird erst nachgeschlagen, wenn einer Person etwas angezeigt wird. Ein Prozess führt außerdem mehr als eine Benutzerkennung: eine reale, die angibt, wer ihn gestartet hat, und eine effektive, die angibt, wessen Berechtigungen er gerade verwendet.

Anwendungskonten trennen Namen und Kennung meist genauso. Der SCIM-Standard, mit dem Benutzer aus einem System in einem anderen eingerichtet werden, verlangt einen für Menschen sichtbaren `userName`, mit dem sie sich anmelden. Er ist innerhalb des Dienstes eindeutig und wird ohne Beachtung der Groß- und Kleinschreibung verglichen. Dazu kommt eine gesonderte `id`, die der Dienst vergibt, die stabil bleibt und nie neu zugeteilt wird. Ein `active`-Flag unterscheidet ein gesperrtes Konto von einem nutzbaren.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Benutzer?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Laptop, Drucker, Anmeldung und Postfach werden schnell und verständlich in Ordnung gebracht, ohne dass sich jemand unkundig fühlen muss.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Eine Abteilung für technischen Support aufgebaut — 10,000 Kunden betreut von einer neu aufgebauten Supportabteilung

Eine Abteilung für technischen Support aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Jede Zugriffsentscheidung ist eine Entscheidung über einen Benutzer. Ein unklarer Begriff davon, wer der Benutzer ist, führt deshalb zu einer unklaren Entscheidung. Aus der Frage „Darf diese Person diese Rechnung sehen?“ wird im

Code „Hat dieses Konto in diesem Kontext eine Berechtigung für diesen Datensatz?“. Jeder Schritt dieser Übersetzung ist eine Stelle, an der die beabsichtigte Person und das tatsächliche Konto auseinanderfallen können.

An der Grenze zwischen Person und Konto liegt auch die Verantwortlichkeit. Protokolle erfassen Konten. Gehört jedes Konto genau einer Person, sagt eine Protokollzeile, wer etwas getan hat. Bei einem geteilten Konto sagt das Protokoll, welches Geheimnis verwendet wurde, aber nichts darüber, wer es eingegeben hat. Einer Person den Zugriff zu entziehen bedeutet dann, das gemeinsame Geheimnis für alle zu ändern.

Betriebssystembenutzer sind ein seit Langem etabliertes Mittel für das Prinzip der geringsten Rechte. Ein Dienst unter einem eigenen unprivilegierten Benutzer wird durch dessen Berechtigungen begrenzt, nicht allein durch Dateieigentum. Er kann weiterhin gemeinsam genutzte beschreibbare Ressourcen ändern, Netzwerkanfragen stellen und ihm zugängliche Zugangsdaten verwenden. Gesonderte Dienstbenutzer verringern den Schaden einer Kompromittierung; sie bieten keine vollständige Isolation. Derselbe Dienst unter dem Superuser kann den ganzen Rechner gefährden. Jedem Programm einen eigenen Benutzer zu geben ist eine kleine Konfigurationsänderung, die die verfügbaren Rechte begrenzt, wenn etwas schiefgeht.

Schließlich haben Personen und Konten unterschiedliche Lebenszyklen. Menschen ändern ihren Namen, ihre E-Mail-Adresse und ihren Arbeitgeber; Konten werden gesperrt, zusammengeführt und gelöscht. Ein System, das ein veränderliches Merkmal wie eine E-Mail-Adresse als Kontenidentität verwendet, wird eines Tages den Verlauf einer Person einer anderen zuordnen.

Wie es in der Praxis aussieht

Auf einer maritimen Plattform verwalten Fachkräfte Organisationen wie Unternehmen und Akademien, die keine eigenen Anmeldungen haben. Die Seite nennt die Regel ausdrücklich: Die Person ist das Konto, die Organisation die Einheit, in deren Namen sie handelt. Die Arbeit bestand in einem Kontextwechsel, mit dem ein Benutzer für jede von ihm verwaltete Organisation handeln kann. Der lokale Speicher des Browsers hält fest, welche Organisation aktiv ist; der Wechsel erfolgt deshalb sofort. Ein serverseitiges Cookie spiegelt den Wert, damit serverseitig gerenderte Seiten denselben Kontext anzeigen.

Nichts davon entscheidet über den Zugriff. Die Autorisierung wird bei jeder Anfrage erneut auf dem Server geprüft. Wer den gespeicherten Wert bearbeitet, ändert deshalb nur die Anzeige auf dem eigenen Bildschirm.

Auf dem firmeneigenen Host werden Releases durch einen Betriebssystembenutzer mit fast keinen Rechten ausgeliefert. Als die Containerplattform eingerichtet wurde, wurden Bereitstellung und Provisionierung getrennt: root bereitet den Bereitstellungspfad einmal vor. Danach stellt ein unprivilegierter Benutzer erneut bereit, indem er ein festgelegtes Skript über eine eng gefasste Regel aufruft, die genau diesen Befehl ohne Argumente erlaubt. Ein gewöhnliches Release baut das Image und startet den Dienst neu. Niemand öffnet dafür eine root-Shell.

Wer die Benutzer sind, lässt sich ebenfalls in einem Bericht festhalten. Eine von sieben schreibgeschützten Berichtsrollen schreibt vom laufenden Host einen Zugriffsbericht als Markdown ins Repository: Konten, sudo-Umfang, Schlüsselfingerabdrücke, Benutzer der Git-Forge und Datenbankrollen stehen nebeneinander. Für alle sieben Berichte gilt dieselbe Regel: Eine Zahl erscheint nur, wenn der Lauf sie gemessen hat. Eine Lücke bleibt leer, statt durch eine plausible Schätzung gefüllt zu werden, die jemand glauben würde. Zusammen haben die Berichte tatsächliche Probleme gefunden, darunter einen nicht deklarierten lauschenden Port und zwei unwirksame Sicherheitskontrollen.

Wo es schiefgeht

Ein gelöschter Benutzer kehrt als jemand anderes zurück. Ein Konto wird entfernt, seine Kennung später erneut vergeben, und der neue Inhaber erbt alles, was noch auf den alten Wert zeigt: Tabellenzeilen, einen Platz auf einer Liste, Dateien auf der Festplatte. Auf einem Unix-Host behalten die Dateien eines entfernten Benutzers dessen numerischen Eigentümer. Der nächste Benutzer mit dieser Nummer besitzt sie. Konten sollten Kennungen erhalten, die nie wiederverwendet werden, wie SCIM es für seine `id` verlangt. Was mit den Daten eines ausscheidenden Benutzers geschieht, muss vor dessen Entfernung feststehen.

Eine E-Mail-Adresse dient als Konto. Sie wirkt wie ein natürlicher Schlüssel, bis jemand die Adresse ändert, den Zugriff auf eine alte verliert oder ein Arbeitgeber eine Adresse an einen neuen Mitarbeiter weitergibt. Dann wird der

Verlauf einer Person einer anderen zugeordnet, oder eine Person hat plötzlich zwei Konten. Eine stabile interne Kennung sollte als Schlüssel dienen; die Adresse ist ein veränderliches Merkmal.

Alle melden sich mit demselben Konto an. Ein gemeinsamer Administratorzugang ist schnell eingerichtet und unmöglich zu auditieren. Verlässt jemand das Team, endet dessen Zugriff erst, wenn das Geheimnis für alle geändert wird. Das Protokoll nennt eine Rolle statt einer Person. Ein Konto je Person und gemeinsam nutzbare Befugnisse über Gruppen, Rollen oder eine eng gefasste sudo-Regel halten sowohl die Protokolle als auch den Entzug von Zugängen nachvollziehbar.

Der Browser entscheidet, was der Benutzer darf. Eine im lokalen Speicher, in einem Cookie oder einem verborgenen Formularfeld gespeicherte Rolle ist ein Vorschlag des Clients, den jeder bearbeiten kann. Vertraut der Server darauf, wird die Bearbeitung zum Angriff. Der Client darf sich den Kontext der Bequemlichkeit halber merken; die Berechtigungen muss der Server selbst bei jeder Anfrage prüfen.

Ein Dienst läuft als root, weil es einfacher war. Alles funktioniert, doch eine ausnutzbare Schwachstelle in diesem Dienst kann den ganzen Host gefährden. Jeder Dienst sollte einen eigenen unprivilegierten Benutzer bekommen. Zusätzliche Rechte sind eng für einen bestimmten Befehl zu vergeben, nicht pauschal.

Weiterführende Lektüre

- NIST-CSRC-Glossar: user — die Definitionen aus den NIST-Sicherheitspublikationen, jeweils mit dem Ursprungsdokument.
- POSIX.1-2024 Base Definitions: Definitions — die formale Bedeutung von Benutzerdatenbank, Benutzerkennung und Benutzername auf Unix-ähnlichen Systemen.
- RFC 7643: SCIM Core Schema — die Ressource User zum Übertragen von Konten zwischen Diensten, mit `userName`, `id` und `active`.
- NIST SP 800-63-4: Digital Identity Guidelines — wie eine Person mehrere digitale Identitäten haben kann und welche Rollen Antragsteller (applicant), registrierte Nutzer (subscriber) und Personen, deren Identität geprüft werden soll (claimant), erfüllen.

- POSIX.1-2024: File Access Permissions — Zugriff über die Berechtigungsklassen Eigentümer, Gruppe und andere, alternative Zugriffskontrollen und angemessene Privilegien; Eigentum allein begrenzt den Zugriff nicht.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie das Gerät oder den Dienst, die beobachteten Symptome und die Daten, die erhalten bleiben müssen. Wir können den sichersten nützlichen Diagnoseschritt bestimmen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Benutzer?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/user/