Engineer

Користувач

людина чи процес, якому дозволено користуватися системою

Що це таке

Користувач — це той, або те, від чийого імені діє система. У звичайних технічних текстах це слово виконує три різні ролі, і більшість плутанини з користувачами виникає тоді, коли ці ролі зливаються в одну.

Перше значення — людина: хтось, хто відкриває продукт і хоче щось зробити. Дизайнери, підтримка й менеджери продукту зазвичай мають на увазі саме її, і всередині системи вона не має жодної сталої форми. Друге — обліковий запис: запис, який система зберігає, щоб упізнати когось наступного разу. Зазвичай це ім’я для входу, внутрішній ідентифікатор, набір прав і статус на кшталт «активний» чи «призупинений». Більшість коду застосунків має на увазі саме його. Третє — користувач операційної системи: в Unix-подібних системах це запис у базі користувачів з іменем і числовим ідентифікатором. Запущені процеси мають облікові дані з ідентифікаторами користувачів, які ядро використовує для перевірки прав доступу.

Ці три значення рідко збігаються один до одного. Людина може мати облікові записи в багатьох сервісах, і настанови NIST щодо цифрової ідентичності прямо кажуть, що одна людина може мати кілька цифрових ідентичностей, кожна з яких унікальна лише в межах свого сервісу. Одним обліковим записом можуть користуватися кілька людей, і це майже завжди помилка. А дуже багато користувачів не мають за собою жодної людини: вебсервер, база даних, завдання резервного копіювання і скрипт розгортання працюють кожен від імені власного користувача ОС і часто мають облікові записи ще й в інших системах. Тому визначення NIST говорить про особу або системний процес, що діє від імені особи.

На Unix-хості різницю між іменем і числом легко побачити:

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

Ядро працює з числом. POSIX визначає базу користувачів, у якій для кожного користувача зберігаються ім’я, числовий ідентифікатор, початковий ідентифікатор групи, початковий робочий каталог і програма, яку слід запустити першою. Власника файлу записано числом, а ім’я шукають лише тоді, коли щось треба показати людині. До того ж процес має більше ніж один ідентифікатор користувача: реальний, що каже, хто його запустив, і ефективний, що каже, чиїми правами він користується саме зараз.

Облікові записи в застосунках зазвичай роблять такий самий поділ. Модель даних SCIM, якою передають користувачів з однієї системи в іншу, вимагає `userName`, який люди бачать і яким входять, унікальний у межах сервісу й нечутливий до регістру, а також окремий `id`, який призначає сам сервіс, який лишається сталим і ніколи не передається іншому. Поле `active` відрізняє призупинений обліковий запис від робочого.

Застосуйте поняття до рішення

Застосуйте «Що таке користувач?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Ноутбук, принтер, вхід у систему і поштова скринька — швидко, зрозумілою мовою і без зверхності.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Заснувала відділ технічної підтримки — 10,000 клієнтів обслуговує відділ підтримки, створений із нуля

Заснувала відділ технічної підтримки Зв'язатися з нами →

Чому це важливо

Кожне рішення про доступ — це рішення про користувача, тож туманне уявлення про те, хто такий користувач, стає туманним рішенням. Питання «чи може ця людина бачити цей рахунок?» у коді звучить як «чи має цей обліковий запис у цьому контексті право, яке покриває цей запис?», і кожен крок такого перекладу — місце, де людина, про яку йшлося, і фактичний обліковий запис можуть розійтися.

Межа між людиною та обліковим записом — це ще й місце, де живе відповідальність. Журнали фіксують облікові записи. Коли кожен обліковий запис належить рівно одній людині, рядок журналу каже, хто що зробив.

Коли обліковий запис спільний, журнал каже, який секрет використали, і нічого про те, хто його ввів, а щоб забрати доступ в однієї людини, секрет доводиться змінювати для всіх.

Користувачі операційної системи — давно усталений засіб забезпечення найменших привілеїв. Служба, що працює від імені власного непривілейованого користувача, обмежена правами цього користувача, а не лише власністю на файли. Вона все ще може змінювати спільні ресурси, доступні для запису, надсилати мережеві запити й використовувати доступні їй облікові дані. Окремі користувачі для служб зменшують шкоду від компрометації, але не забезпечують повної ізоляції. Та сама служба, запущена від імені суперкористувача, може поставити під загрозу всю машину. Окремий користувач для кожної програми — невелика зміна конфігурації, яка звужує доступні права, коли щось іде не так.

І нарешті, люди й облікові записи живуть різний час. Люди змінюють імена, адреси електронної пошти й роботодавців; облікові записи призупиняють, об’єднують і видаляють. Система, яка вважає ідентичністю облікового запису атрибут, що може змінитися, наприклад адресу пошти, одного дня причепить історію однієї людини до іншої.

Як це виглядає на практиці

На морській платформі фахівці керують організаціями, як-от компаніями й академіями, що не мають власних облікових записів для входу. Сторінка формулює правило прямо: обліковим записом є сама людина, а організація — це те, від імені чого вона діє. Завданням було перемикання контексту, яке дає користувачеві діяти від імені будь-якої організації, якою він керує. Яка організація активна, зберігає localStorage браузера, тож перемикання миттєве, а серверний cookie дублює цей стан, щоб сторінки, відрендерені на сервері, показували той самий контекст. Жодне з цих значень не вирішує питання доступу. Авторизацію повторно перевіряють на сервері під час кожного запиту, тож той, хто змінить збережене значення, змінить лише те, що показує його власний екран.

На власному хості компанії релізи викочує користувач ОС майже без прав. Коли налаштовували контейнерне середовище, розгортання відокремили від підготовки хоста: root один раз налаштовує шлях розгортання, а далі непривілейований користувач перерозгортає застосунок, запускаючи один

фіксований скрипт через вузьке правило, яке дозволяє саме цю команду й жодних аргументів. Звичайний реліз збирає образ і перезапускає службу, і root-оболонка для цього не потрібна.

Знати, хто ваші користувачі, — теж звіт. Одна з семи ролей звітності лише для читання переносить звіт про доступ із живого хоста в Markdown у репозиторії: облікові записи, межі sudo, відбитки ключів, користувачі git-сервера і ролі бази даних поруч одне з одним. Правило для всіх семи звітів — не друкувати жодного числа, якого прогін не виміряв. Разом звіти знайшли цілком реальні речі, серед них два мертві засоби безпеки й незадекларований порт у режимі прослуховування.

Де виникають проблеми

Видалений користувач повертається кимось іншим. Обліковий запис видаляють, згодом його ідентифікатор видають знову, і новачок успадковує все, що досі вказує на старе значення: рядки в таблиці, місце в списку, файли на диску. На Unix-хості файли видаленого користувача зберігають числового власника, і наступний користувач, створений із цим числом, стає їхнім власником. Давайте обліковим записам ідентифікатори, які ніколи не використовуються повторно, як SCIM вимагає від свого `id`, і вирішуйте долю даних користувача, що йде, ще до його видалення.

Обліковим записом стає адреса електронної пошти. Вона схожа на природний ключ, доки хтось не змінить адресу, не втратить доступ до старої або роботодавець не передасть адресу новому працівникові. Тоді історію однієї людини прив’язано до іншої, або одна людина опиняється з двома обліковими записами. Тримайте ключем сталий внутрішній ідентифікатор, а адресу вважайте атрибутом, що може змінитися.

Усі входять під одним обліковим записом. Спільний адміністраторський вхід швидко налаштувати і неможливо перевірити. Коли хтось іде, його доступ припиняється лише тоді, коли секрет змінюють для всіх, а журнал називає роль замість людини. Один обліковий запис на людину, а спільні повноваження — через групи, ролі чи вузько обмежене правило sudo: так чесними лишаються і журнал, і звільнення.

Що можна користувачеві, вирішує браузер. Роль, збережена в localStorage, cookie чи прихованому полі форми, — це лише пропозиція від клієнта, і змінити її може будь-хто. Якщо сервер їй довіряє, така зміна стає атакою.

Клієнт може пам’ятати контекст для зручності; права сервер має перевіряти сам, під час кожного запиту.

Служба працює від root, бо так було простіше. Усе працює, але вразливість цієї служби, яку можна експлуатувати, може поставити під загрозу весь хост. Дайте кожній службі власного непривілейованого користувача, а додаткові права видавайте вузько, на одну названу команду, а не широко.

Що почитати далі

- Глосарій NIST CSRC: user — визначення з публікацій NIST з безпеки, кожне з документом, звідки воно походить.
- POSIX.1-2024 Base Definitions: Definitions — формальне значення бази користувачів, ідентифікатора користувача та імені користувача в Unix-подібних системах.
- RFC 7643: SCIM Core Schema — ресурс User для перенесення облікових записів між сервісами, з полями `userName`, `id` і `active`.
- NIST SP 800-63-4: Digital Identity Guidelines — як одна людина може мати кілька цифрових ідентичностей і що означають ролі applicant, subscriber і claimant.
- POSIX.1-2024: File Access Permissions — доступ за класами прав власника, групи та інших користувачів, альтернативні механізми контролю доступу й належні привілеї; сама власність не визначає меж доступу.

Словник

Обирайте за фактичними умовами

Опишіть пристрій або сервіс, спостережувані симптоми та дані, які слід зберегти. Ми можемо визначити найбезпечніший корисний діагностичний крок.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке користувач?» стосується вашої організації.
Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/user/