Engineer

Автентифікація

підтвердження того, хто саме надсилає запит

Що це таке

Автентифікація — це перевірка, яку система проводить, перш ніж прийняти заявлену ідентичність: хтось каже «я — цей обліковий запис», а система просить доказ, який міг би надати лише власник цього облікового запису. Розробники часто скорочують її до authn, а в інтерфейсах той самий момент називають входом або логіном.

Ідентифікація — це саме твердження, наприклад введене ім'я користувача. Автентифікація — доказ цього твердження. Авторизація — окреме рішення, ухвалене потім: що дозволено вже відомій особі. Навіть HTTP плутає ці слова: RFC 9110 називає код стану 401 «Unauthorized», але визначає його як запит без чинних облікових даних для автентифікації і зобов'язує сервер надіслати заголовок WWW-Authenticate з описом того, як їх надати. Відповідь «ми знаємо, хто ви, і однаково ні» має інший код — 403 Forbidden.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="api"
```

Докази бувають різних видів, їх називають факторами. Те, що ви знаєте, — це пароль або PIN-код. Те, що ви маєте, — телефон, апаратний ключ безпеки чи файл із приватним ключем. Те, чим ви є, — відбиток пальця чи обличчя, які розпізнає біометричний сенсор. Багатофакторна автентифікація поєднує два різні види, тож зловмисник, який здобув один, однаково не має другого. Два паролі — це все одно один фактор.

Багато що в автентифікації взагалі обходиться без людей. Сервіси пред'являють одне одному API-токени, застосунки входять до своїх баз даних, а SSH добре показує машинний варіант: RFC 4252 робить «publickey»

єдиним методом, який мусить підтримувати кожна реалізація, і клієнт доводить, що володіє приватним ключем, підписуючи ним дані, тож сам ключ мережею не передається. Єдиний вхід (single sign-on) переносить перевірку деінде. В OpenID Connect постачальник ідентичності автентифікує користувача й передає застосунку підписаний ID-токен, тож застосунок пароля взагалі не бачить. Ключі доступу (passkeys), побудовані на W3C Web Authentication API, зберігають приватний ключ на пристрої, прив’язаний до одного сайту, і підписують виклик сервера разом із походженням (origin), з яким браузер насправді спілкується, тож сайт-двійник отримує підпис, який деінде не спрацює.

Застосуйте поняття до рішення

Застосуйте «Що таке автентифікація?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Облікові записи, паролі й права налаштовані так, щоб доступ відповідав роботі, а звільнені втрачали його того ж дня.

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. Написала 1 000 правил RBAC для Linux-серверів — 1,000 правил доступу написано, щоб усунути вразливість

Написала 1 000 правил RBAC для Linux-серверів Зв'язатися з нами →

Чому це важливо

Кожен інший контроль виходить з того, що цей спрацював. Дозволи, журнали аудиту, обмеження частоти запитів і білінг прив’язані до особи, і якщо перевірка, яка цю особу встановила, слабка, кожен із них сумлінно діє щодо не тієї сторони. Рядок журналу про те, що адміністратор видалив таблицю, фіксує, який обліковий запис це зробив, і лише добра автентифікація перетворює це на відповідь, хто саме.

Практичне рішення — наскільки сильною має бути перевірка біля кожних дверей, і NIST SP 800-63B дає для цього найчіткішу шкалу. Перший рівень гарантії, AAL1, задовольняється одним фактором. AAL2 вимагає двох факторів, і верифікатор на цьому рівні мусить пропонувати принаймні один варіант, стійкий до фішингу. AAL3 вимагає стійкого до фішингу автентифікатора, чий приватний ключ неможливо експортувати. Щодо одного популярного вибору настанова говорить прямо: коди, які вводять вручну, хоч надіслані в SMS, хоч показані в застосунку, не стійкі до фішингу, бо введені цифри ніяк не прив’язані до сайту, що їх запросив, і зловмисник може переслати їх далі, поки вони ще чинні.

Редакція 4 того самого документа, яка 1 серпня 2025 року замінила редакцію 3, також розв'язала кілька давніх суперечок про паролі. Пароль, що є єдиним фактором, мусить мати щонайменше 15 символів, а пароль, який використовують лише в межах багатофакторного входу, — щонайменше 8. Верифікатори не можуть встановлювати інших правил щодо складу пароля чи вимагати періодичної зміни, але мусять вимагати зміну, коли є ознаки компрометації. Нові паролі звіряють зі списком поширених паролів і тих, що потрапили у витоки, користувачеві пояснюють, чому вибір відхилено, а збережену форму солять і хешують схемою з параметром вартості. Невдалих спроб поспіль для одного автентифікатора може бути щонайбільше 100. Для сесій на рівні AAL2 документ рекомендує повторну автентифікацію щонайменше раз на 24 години та після години бездіяльності.

Наша рекомендація випливає з цього. Якщо користувачі вже мають обліковий запис у надійного постачальника ідентичності, нехай перевірку виконує він, а застосунок приймає його підписану відповідь: пароль, якого застосунок не зберігає, він і не зможе розкрити. Якщо ж застосунок мусить зберігати паролі сам, варто виконувати 800-63B буквально, а не з пам'яті, і ставити метод, стійкий до фішингу, перед усім, що стосується адміністрування.

Як це виглядає на практиці

Коли компанія перенесла свій вихідний код на власний git-сервер, автентифікація стала першим пунктом у переліку того, що означає тримати власну git-форджу. Обрана платформа працює насамперед через SSH і не має вебпанелі адміністрування, тож увійти можна ключем. Вона приватна за замовчуванням: анонімний доступ відхиляється, доступ без ключа відхиляється, а кожен оголошений репозиторій позначено як приватний. Ключ перевіряє слухач SSH, прив'язаний лише до інтерфейсу зворотної петлі на порту 23231, тож запит на вхід із-поза машини доходить до нього тільки через проміжний хост, а оголошена поверхня брандмауера лишається того самого розміру.

Та сама турбота виглядає інакше в масштабі цілої організації. Раніше, на інфраструктурній посаді, робота полягала в консолідації ідентичностей і доступу в M365, 1Password, Red Hat SSO та OKTA SSO, де дозволи надавалися ситуативно, з часом і різними людьми. Було впроваджено принцип найменших привілеїв, а онбординг і офбординг стандартизовано, тож доступ надавали й вчасно відкликали, а не залишали після того, як людина

пішла далі. Ризик несанкціонованого доступу різко знизився, а доступ став придатним для аудиту: нарешті можна було відповісти на запитання «хто має доступ до цього і чому».

Автентифікація має й обличчя, і користувачі судять про неї з повідомлень про помилки. Під час входу одного застосунку до великого поштового провайдера повідомлення казало, що пароль хибний. Пароль був правильний: провайдер вимагає окремого пароля застосунку для сторонніх клієнтів і відхиляє пароль облікового запису, хоч як ретельно його набрано. Повідомлення переписали так, щоб воно називало провайдера, пояснювало, що для інших клієнтів той вимагає пароля застосунку, і казало, де його створити. Тепер тест відтворює саме цю відмову й перевіряє, що повідомлення містить назву провайдера і потрібний тип облікових даних, тож старе формулювання непомітно не повернеться.

Де виникають проблеми

- **Повідомлення звинувачує користувача в причині, яка лежить деінде.** Заблокований обліковий запис, прострочений пароль, годинник, що розійшовся з генератором кодів, і провайдер, який вимагає інших облікових даних, — усе це приходить як «неправильний пароль», і користувач знову набирає те, що ніколи не спрацює. Варто називати причину, з якою користувач може щось зробити, але відповідати однаково для облікового запису, що існує, і для неіснуючого, щоб форма не видавала списку дійсних імен користувачів.
- **Фішингова сторінка пересилає код у реальному часі.** Користувач вводить пароль і шестизначний код на копії справжньої сторінки, а вона передає обидва справжньому сайту в межах 30 секунд, які RFC 6238 рекомендує як типовий крок часу. Одноразові коди зупиняють викрадені й повторно використані паролі, але мало допомагають проти живого пересилання. Ключі, прив’язані до походження, як-от ключі доступу, підписують лише для одного сайту, тому NIST вважає їх стійкими до фішингу, а введені вручну коди — ні.
- **Облікові дані переживають людей, яким вони належали.** Офбординг закриває обліковий запис у каталозі, а SSH-ключ у файлі `authorized_keys`, персональний токен доступу та спільний пароль адміністратора працюють і далі. Кожні облікові дані потребують власника, місця, де їх зафіксовано, і дати, коли хтось перегляне їх знову.

- **Сесія лишається чинною довго після входу.** Cookie чи токен, що діє місяцями, — це пароль із довгим запалом. Вихід має завершувати сесію на сервері, а не лише видаляти cookie в одному браузері, а самодостатній токен, який сервер повторно не перевіряє, можна використовувати до кінця його строку, хоч би що сталося з обліковим записом.
- **Відновлення облікового запису — найслабший вхід.** Надійний вхід поруч із посиланням для скидання, яке надсилають на електронну пошту, робить поштову скриньку справжніми обліковими даними. Відновлення заслуговує на ту саму ретельність, що й головні двері, бо зловмисник обере слабше з двох.

Що почитати далі

- NIST SP 800-63B-4 — чинний текст: рівні гарантії, мінімум у 15 символів для пароля і те, що вважається стійким до фішингу.
- NIST SP 800-63B, revision 3 — настанова з автентифікації 2017 року, яка лишається в мережі з позначкою, що з 1 серпня 2025 року її замінила редакція 4.
- RFC 9110, section 11: HTTP Authentication — виклик і відповідь, що стоять за кодом 401, WWW-Authenticate і заголовком Authorization.
- RFC 4252: The SSH Authentication Protocol — вхід за відкритим ключем, паролем і на основі хоста, як їх визначає SSH.
- RFC 6238: TOTP — як застосунки-автентифікатори виводять короткі коди зі спільного секрету й годинника.
- Web Authentication, Level 3 — API W3C, на якому працюють ключі доступу, з обліковими даними відкритого ключа, прив’язаними до походження.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке автентифікація?» 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/authentication/