Engineer

Autentisering

bekräftelse av den identitet som anges i en begäran

Vad det är

Autentisering är den kontroll som ett system gör innan det godtar en uppgiven identitet: någon säger ”jag är det här kontot”, och systemet ber om bevis som bara den som innehar kontot borde kunna lägga fram. Utvecklare kortar ofta ordet till authn, och gränssnitt kallar samma ögonblick för att logga in.

Identifiering är själva påståendet, t.ex. när någon skriver ett användarnamn. Autentisering är beviset för påståendet. Auktorisering är det separata beslutet, som fattas efteråt, om vad den nu kända identiteten får göra. Till och med HTTP suddar ut gränsen: RFC 9110 ger statuskod 401 namnet ”Unauthorized” men definierar den som en begäran som saknar giltiga uppgifter för autentisering, och ålägger servern att skicka ett WWW-Authenticate-huvud som beskriver hur de ska lämnas. Koden för ”vi vet vem du är, och svaret är ändå nej” är 403 Forbidden.

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

Bevisen finns i olika slag, som brukar kallas faktorer. Något du vet är ett lösenord eller en pinkod. Något du har är en telefon, en säkerhetsnyckel i hårdvara eller en fil med en privat nyckel. Något du är är ett fingeravtryck eller ett ansikte. Multifaktorautentisering kombinerar två olika slag, så att en tjuv som kommer över det ena fortfarande saknar det andra. Två lösenord är fortfarande en enda faktor.

Mycket av det sker helt utan människor. Tjänster visar upp API-token för varandra, och SSH visar den maskinella varianten tydligt: RFC 4252 gör ”publickey” till den enda metod som varje implementation måste stödja, och

klienten bevisar att den har en privat nyckel genom att signera med den, så nyckeln korsar aldrig nätverket. Single sign-on flyttar kontrollen någon annanstans. Med OpenID Connect autentiserar en identitetsleverantör användaren och ger applikationen en signerad ID-token, så applikationen ser aldrig något lösenord. Passkeys, som bygger på W3C:s Web Authentication API, håller en privat nyckel kvar på enheten, avgränsad till en enda webbplats, och signerar serverns utmaning tillsammans med det ursprung som webbläsaren faktiskt talar med, så en domän som bara liknar den riktiga får ingenting som den kan spela upp igen.

Använd begreppet i ett beslut

Använd ”Vad är autentisering?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Skrev 1 000 RBAC-regler som säkrar Linux-servrar — 1,000 åtkomstregler, skrivna för att täppa till en säkerhetslucka

Skrev 1 000 RBAC-regler som säkrar Linux-servrar Hör av er →

Varför det spelar roll

Varje annan kontroll utgår från att den här fungerade. Behörigheter, granskningsloggar, anropsgränser och fakturering är alla knutna till en identitet, och om kontrollen som fastställde identiteten är svag tillämpas var och en av dem troget på fel part. En loggrad som säger att en administratör raderade en tabell visar vilket konto som gjorde det, och bara god autentisering gör det till en uppgift om vem.

Det praktiska beslutet är hur stark kontrollen måste vara vid varje dörr, och NIST SP 800-63B ger den tydligaste skalan att avgöra det med. Dess första tillitsnivå, AAL1, godtar en enda faktor. AAL2 kräver två faktorer, och en verifierare på den nivån måste erbjuda minst ett nätfiskeresistent alternativ. AAL3 kräver en nätfiskeresistent autentiserare vars privata nyckel inte går att exportera. Riktlinjen är rak om ett populärt val: koder som skrivs in för hand, vare sig de skickas med sms eller visas i en app, är inte nätfiskeresistenta, eftersom ingenting knyter de inskrivna siffrorna till den webbplats som bad om dem.

Revision 4 av samma dokument, som ersatte revision 3 den 1 augusti 2025, avgjorde också flera tvister om lösenord. Ett lösenord som används som enda faktor måste vara minst 15 tecken långt, och ett som bara används inom en

inloggning med flera faktorer minst 8. Verifierare får inte ställa några andra krav på hur lösenordet är sammansatt och får inte kräva byten med jämna mellanrum, men måste tvinga fram ett byte när det finns tecken på att lösenordet har röjts. Nya lösenord kontrolleras mot en spärrlista med vanliga och läckta lösenord, användaren får veta varför ett val avvisades, och den lagrade formen saltas och hashas med en metod som har en kostnadsfaktor. Antalet misslyckade försök i följd mot en och samma autentiserare begränsas till 100. För sessioner på AAL2 rekommenderar riktlinjen att användaren tillfrågas igen minst var 24:e timme, och efter en timme utan aktivitet.

Vår rekommendation följer av det. Där användarna redan har ett konto hos en kompetent identitetsleverantör, låt den leverantören göra kontrollen och godta dess signerade svar, eftersom ett lösenord som applikationen aldrig lagrar är ett lösenord som den aldrig kan läcka. Där applikationen måste ha egna lösenord, följ 800-63B ordagrant i stället för ur minnet, och sätt en nätfiskeresistent metod framför allt som är administrativt.

Så ser det ut i praktiken

När företaget flyttade sin källkod till en egen git-server stod autentisering först på listan över vad det innebär att köra en forge. Den valda forgen är byggd i första hand för SSH och saknar administrativt webbgränssnitt, så vägen in är en nyckel. Varje inloggning måste visa en: anonyma försök avvisas, liksom försök utan nyckel, och varje deklarerat repo är privat som standard. Själva inloggningen sker på port 23231, som bara lyssnar på loopback och som en användare utifrån når via en hoppserver, och därför växer brandväggens deklarerade yta inte.

Samma fråga ser annorlunda ut i en hel organisations skala. I en tidigare roll som infrastrukturingenjör, mellan 2022 och 2024, handlade arbetet om att samla identiteter och åtkomst i M365, 1Password, Red Hat SSO och OKTA SSO, där behörigheter hade delats ut ad hoc, över tid, av olika personer. Principen om minsta privilegium upprätthölls, och onboarding och offboarding standardiserades så att åtkomst återkallades snabbt i stället för att ligga kvar efter att någon hade gått vidare. Risken för obehörig åtkomst sjönk kraftigt och åtkomsten blev granskningsbar: teamet kunde äntligen svara på ”vem kommer åt det här, och varför”.

Autentisering har också ett ansikte, och användare bedömer den efter dess felmeddelanden. Vid en applikations inloggning mot en stor e-postleverantör sa meddelandet att lösenordet var fel. Lösenordet var rätt; leverantören kräver ett

appspecifikt lösenord för klienter från tredje part och avvisar kontots lösenord hur noggrant det än skrivs. Meddelandet skrevs om så att det namnger leverantören, säger att den kräver ett appspecifikt lösenord för andra klienter och anger var man skapar ett. Ett test framkallar nu det avslaget och kontrollerar att meddelandet innehåller leverantörens namn och den typ av inloggningsuppgift som krävs, så att den gamla formuleringen inte kan komma tillbaka obemärkt.

Var det går fel

- **Meddelandet skyller på användaren när orsaken ligger någon annanstans.** Ett låst konto, ett utgånget lösenord, en klocka som har glidit ur takt med en kodgenerator och en leverantör som kräver en annan inloggningsuppgift kommer alla fram som ”fel lösenord”, och användaren skriver om något som aldrig kommer att fungera. Namnge den orsak som användaren kan göra något åt, men håll svaret identiskt för ett konto som finns och ett som inte finns, så att formuläret inte ger någon en lista över giltiga användarnamn.
- **En nätfiskesida skickar vidare koden i realtid.** Användaren skriver ett lösenord och en sexsiffrig kod på en kopia av den riktiga sidan, som vidarebefordrar båda till den äkta webbplatsen inom de 30 sekunder som RFC 6238 rekommenderar som standardvärde för tidssteget. Engångskoder stoppar stulna och återanvända lösenord; mot en vidarebefordran i realtid hjälper de föga. Nycklar som är bundna till ett ursprung, som passkeys, signerar bara för en webbplats, och därför räknar NIST dem som nätfiskeresistenta men inte inskrivna koder.
- **Inloggningsuppgifter lever längre än de människor som hade dem.** Offboarding stänger kontot i katalogen, medan en SSH-nyckel i en `authorized_keys` -fil, en personlig åtkomsttoken och ett delat administratörslösenord fortsätter att fungera. Varje inloggningsuppgift behöver en ägare, en plats där den finns registrerad och ett datum då någon tittar på den igen.
- **Sessionen förblir giltig långt efter inloggningen.** En kaka eller token som varar i månader är ett lösenord med lång stubin. Utloggningen måste avsluta sessionen på servern, inte bara radera kakan i en webbläsare, och en självbärande token som servern aldrig stämmer av mot något förblir användbar tills den löper ut, vad som än händer med kontot.
- **Kontoåterställningen är den svagaste ingången.** En stark inloggning bredvid en återställningslänk som skickas till en e-postadress gör brevlådan till den egentliga inloggningsuppgiften. Återställningen förtjänar samma granskning som ytterdörren, eftersom en angripare väljer den ingång som är svagast.

Läs mer

- NIST SP 800-63B-4 — den aktuella texten: tillitsnivåerna, minimikravet på 15 tecken för lösenord och vad som räknas som nätfiskeresistent.
- NIST SP 800-63B, revision 3 — riktlinjen för autentisering från 2017, som finns kvar online och är märkt som ersatt av revision 4 från den 1 augusti 2025.
- RFC 9110, section 11: HTTP Authentication — utmaningen och svaret bakom 401, WWW-Authenticate och huvudet Authorization.
- RFC 4252: The SSH Authentication Protocol — inloggning med publik nyckel, med lösenord och med värdbaserad metod, så som SSH definierar dem.
- RFC 6238: TOTP — hur autentiseringsappar härleder sina korta koder ur en delad hemlighet och klockan.
- Web Authentication, Level 3 — W3C:s API bakom passkeys och dess inloggningsuppgifter med publik nyckel, avgränsade till ett ursprung.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är autentisering?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/authentication/