Engineer

Autentificering

bekræftelse af den påståede identitet bag en anmodning

Hvad det er

Autentificering er den kontrol, et system udfører, før det godtager en identitet, som nogen gør krav på: der står »jeg er denne konto«, og systemet beder om noget, som kun kontoens indehaver burde kunne fremlægge. Udviklere forkorter det ofte til authn, og i brugerfladen hedder det samme øjeblik at logge ind.

Identifikation er selve påstanden, som når et brugernavn tastes ind. Autentificering er beviset for påstanden. Autorisation er den særskilte beslutning bagefter om, hvad den nu kendte identitet må. Selv HTTP blander dem: RFC 9110 kalder statuskode 401 »Unauthorized«, men definerer den som en anmodning uden gyldige autentificeringsoplysninger og kræver, at serveren sender en WWW-Authenticate-header, der beskriver, hvordan de skal leveres. Svaret »vi ved godt, hvem du er, og svaret er stadig nej« hedder 403 Forbidden.

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

Beviserne findes i forskellige slags, som kaldes faktorer. Noget, du ved, er en adgangskode eller en pinkode. Noget, du har, er en telefon, en hardwarenøgle eller en fil med en privat nøgle. Noget, du er, er et fingeraftryk eller et ansigt. Multifaktorautentificering kombinerer to forskellige slags, så en tyv, der får fat i den ene, stadig mangler den anden. To adgangskoder er stadig én faktor.

Meget af autentificeringen foregår helt uden mennesker. Tjenester viser API-tokens til hinanden, og SSH viser maskinudgaven tydeligt: RFC 4252 gør »publickey« til den eneste metode, enhver implementering skal understøtte, og klienten beviser, at den har den private nøgle, ved at underskrive med den, så selve nøglen aldrig sendes over netværket. Single sign-on flytter kontrollen et

andet sted hen. Med OpenID Connect autentificerer en identitetsudbyder brugeren og giver programmet et signeret ID-token, så programmet aldrig ser en adgangskode. Passkeys, der bygger på W3C's Web Authentication API, holder en privat nøgle på enheden, bundet til ét websted, og underskriver serverens udfordring sammen med den oprindelse (origin), browseren faktisk taler med, så et forvekslingsdomæne får intet, det kan afspille igen.

Brug begrebet i en beslutning

Brug »Hvad er autentificering?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Konti, adgangskoder og rettigheder sat op, så folk har præcis den adgang, jobbet kræver, og fratrådte mister den samme dag.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har skrevet 1.000 RBAC-regler til Linux-servere — 1,000 adgangsregler skrevet for at lukke et sikkerhedshul

Har skrevet 1.000 RBAC-regler til Linux-servere Kontakt os →

Hvorfor det betyder noget

Alle andre kontroller går ud fra, at denne lykkedes. Rettigheder, revisionslogge, hastighedsgrænser og fakturering knytter sig alle til en identitet, og er kontrollen, der fastlagde identiteten, svag, rammer hver af dem trofast den forkerte. En loglinje om, at en administrator slettede en tabel, viser, hvilken konto det var, og kun god autentificering gør det til, hvem det var.

Den praktiske beslutning er, hvor stærk kontrollen skal være ved hver dør, og NIST SP 800-63B giver den klareste skala. Det første sikringsniveau, AAL1, nøjes med én faktor. AAL2 kræver to faktorer, og en verifikator på det niveau skal tilbyde mindst én phishing-resistent mulighed. AAL3 kræver en phishing-resistent autentifikator, hvis private nøgle ikke kan eksporteres. Om et populært valg er retningslinjen ligefrem: koder, der tastes ind i hånden, hvad enten de kommer som sms eller vises i en app, er ikke phishing-resistente, fordi de indtastede cifre ikke er bundet til det websted, der bad om dem.

Revision 4 af samme dokument, som afløste revision 3 den 1. august 2025, afgjorde også flere stridspunkter om adgangskoder. En adgangskode, der bruges som eneste faktor, skal være mindst 15 tegn lang, og en, der kun bruges inden for et login med flere faktorer, mindst 8. Verifikatorer må ikke stille andre krav til sammensætningen og må ikke kræve periodiske skift, men skal kræve et skift, når der er tegn på kompromittering. Nye adgangskoder tjekkes mod en blokeringsliste over almindelige og lækkede koder, brugeren får at vide, hvorfor

et valg blev afvist, og den gemte form saltes og hashes med en metode, der har en omkostningsfaktor. Fortløbende mislykkede forsøg mod én autentifikator begrænses til 100. For sessioner på AAL2 anbefaler dokumentet at spørge igen mindst hver 24. time og efter en times inaktivitet.

Vores anbefaling følger af det. Hvor brugerne allerede har en konto hos en kompetent identitetsudbyder, så lad udbyderen foretage kontrollen og tag imod dens signerede svar, for en adgangskode, programmet aldrig gemmer, kan det heller aldrig lække. Hvor programmet selv skal opbevare adgangskoder, så følg 800-63B til punkt og prikke frem for efter hukommelsen, og sæt en phishing-resistent metode foran alt, der har med administration at gøre.

Sådan ser det ud i praksis

Da virksomheden flyttede sin kildekode over på sin egen git-server, stod autentificering først på listen over, hvad det vil sige at drive en forge. Den valgte forge er bygget til SSH først og har intet webadministrationspanel, så vejen ind går gennem en nøgle. En klient, der ikke oplyser, hvem den er, bliver vist bort, og det bliver en klient uden nøgle også, mens hvert erklæret repository står som privat. Autentificeringen begynder allerede ved netværket: forgens SSH-port, 23231, lytter kun på loopback, så en klient udefra skal først ind over en jump host, og firewallen får ikke en åben port mere at erklære.

Det samme hensyn ser anderledes ud, når det gælder en hel organisation. I en tidligere rolle som infrastrukturingeniør bestod arbejdet i at konsolidere identiteter og adgang på tværs af M365, 1Password, Red Hat SSO og OKTA SSO, hvor rettigheder var givet ad hoc, over tid, af forskellige folk. Least privilege blev håndhævet, og on- og offboarding blev standardiseret, så adgang blev tilbagekaldt hurtigt i stedet for at hænge ved, efter nogen var gået videre. Risikoen for uautoriseret adgang faldt markant, og adgangen blev sporbar: man kunne endelig svare på »hvem kan nå det her, og hvorfor«.

Autentificering har også et ansigt, og brugerne bedømmer den på dens fejlbeskeder. Ved et programs login mod en stor mailudbyder sagde beskeden, at adgangskoden var forkert. Adgangskoden var rigtig; udbyderen kræver en programspecifik adgangskode til tredjepartsklienter og afviser kontoens egen adgangskode, uanset hvor omhyggeligt den tastes. Beskeden blev skrevet om, så den nævner udbyderen, siger, at den kræver en programspecifik adgangskode til andre klienter, og fortæller, hvor man opretter en. En test fremkalder nu netop

den afvisning og kontrollerer, at beskeden indeholder udbyderens navn og den krævede type adgangsoplysninger, så den gamle formulering ikke kan vende tilbage ubemærket.

Hvor det går galt

- **Beskeden giver brugeren skylden for en årsag et andet sted.** En låst konto, en udløbet adgangskode, et ur, der er gledet ud af takt med en kodegenerator, og en udbyder, der kræver en anden slags adgangsoplysninger, ender alle som »forkert adgangskode«, og brugeren taster igen noget, der aldrig vil virke. Nævn den årsag, brugeren kan handle på, men giv samme svar for en konto, der findes, og en, der ikke gør, så formularen ikke udleverer en liste over gyldige brugernavne.
- **En phishingside videresender koden i realtid.** Brugeren taster adgangskode og en sekscifret kode ind på en kopi af den rigtige side, som sender begge videre til det ægte websted inden for de 30 sekunder, RFC 6238 anbefaler som standardtidstrin. Engangskoder stopper stjålne og genbrugte adgangskoder, men hjælper kun lidt mod en direkte videresendelse. Nøgler, der er bundet til en oprindelse, som passkeys, underskriver kun for ét websted, og derfor regner NIST dem for phishing-resistente og indtastede koder ikke.
- **Adgangsoplysninger overlever de mennesker, der havde dem.** Offboarding lukker kontoen i kataloget, mens en SSH-nøgle i en `authorized_keys` -fil, et personligt adgangstoken og en delt administratoradgangskode bliver ved med at virke. Hvert sæt adgangsoplysninger skal have en ejer, et sted, hvor det er registreret, og en dato, hvor nogen ser på den igen.
- **Sessionen gælder længe efter login.** En cookie eller et token, der holder i månedsvis, er en adgangskode med lang lunte. At logge ud skal afslutte sessionen på serveren, ikke kun slette cookien i én browser, og et selvstændigt token, som serveren aldrig slår op igen, kan bruges, til det udløber, uanset hvad der sker med kontoen.
- **Kontogendannelse er den svageste indgang.** Et stærkt login ved siden af et nulstillingslink, der sendes til en e-mailadresse, gør postkassen til den egentlige adgangsoplysninger. Gendannelsen fortjener samme granskning som hoveddøren, for en angriber vælger den, der er svagest.

Læs mere

- NIST SP 800-63B-4 — den gældende tekst: sikringsniveauer, minimum på 15 tegn og hvad der tæller som phishing-resistent.
- NIST SP 800-63B, revision 3 — retningslinjen fra 2017, stadig online og markeret som afløst af revision 4 fra 1. august 2025.

- RFC 9110, section 11: HTTP Authentication — udfordring og svar bag 401, WWW-Authenticate og Authorization-headeren.
- RFC 4252: The SSH Authentication Protocol — login med offentlig nøgle, adgangskode og værtsbaseret login, som SSH definerer dem.
- RFC 6238: TOTP — hvordan autentificeringsapps udleder deres korte koder af en delt hemmelighed og uret.
- Web Authentication, Level 3 — W3C-API'et bag passkeys og dets offentlige nøgler, der er bundet til én oprindelse.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er autentificering?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/authentication/