Engineer

Autentisering

bekreftelse på identiteten som oppgis i en forespørsel

Hva det er

Autentisering er sjekken et system gjør før det godtar en oppgitt identitet: noen sier «jeg er denne kontoen», og systemet ber om bevis som bare den som har kontoen, skal kunne legge fram. Utviklere forkorter det ofte til authn, og grensesnitt kaller det samme øyeblikket å logge inn eller logge på.

Identifisering er selve påstanden, f.eks. at du skriver inn et brukernavn. Autentisering er beviset for påstanden. Autorisasjon er den separate beslutningen som tas etterpå, om hva den nå kjente identiteten har lov til å gjøre. Selv HTTP visker ut grensen: RFC 9110 kaller statuskode 401 «Unauthorized», men definerer den som en forespørsel som mangler gyldig påloggingsinformasjon, og pålegger serveren å sende et WWW-Authenticate-hode som beskriver hvordan den skal oppgis. Koden for «vi vet hvem du er, og svaret er fortsatt nei» er 403 Forbidden.

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

Bevisene finnes i ulike typer, som regel kalt faktorer. Noe du vet, er et passord eller en PIN-kode. Noe du har, er en telefon, en fysisk sikkerhetsnøkkel eller en fil med en privat nøkkel. Noe du er, er et fingeravtrykk eller et ansikt. Flerfaktorautentisering kombinerer to ulike typer, slik at en tyv som får tak i den ene, fortsatt mangler den andre. To passord er fortsatt bare én faktor.

Mye av det involverer ikke noe menneske i det hele tatt. Tjenester viser fram API-tokener til hverandre, og SSH viser den maskinlike varianten tydelig: RFC 4252 gjør «publickey» til den eneste metoden hver implementasjon må støtte, og klienten beviser at den har en privat nøkkel ved å signere med den, så nøkkelen

krysser aldri nettverket. Enkel pålogging (single sign-on) flytter sjekken et annet sted. Med OpenID Connect autentiserer en identitetsleverandør brukeren og gir applikasjonen et signert ID-token, så applikasjonen ser aldri noe passord. Passnøkler, bygd på W3Cs Web Authentication API, holder en privat nøkkel på enheten, avgrenset til ett nettsted, og signerer serverens utfordring sammen med opprinnelsen nettleseren faktisk snakker med, så et domene som bare ligner, mottar ingenting det kan spille av på nytt.

Bruk begrepet i en beslutning

Bruk «Hva er autentisering?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Kontoer, passord og tilganger ordnet slik at folk har nøyaktig den tilgangen jobben krever, og de som slutter, mister den samme dag.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Skrev 1 000 RBAC-regler for å sikre Linux-servere — 1,000 tilgangsregler skrevet for å tette et sikkerhetshull

Skrev 1 000 RBAC-regler for å sikre Linux-servere Ta kontakt →

Hvorfor det er viktig

Alle andre kontroller forutsetter at denne fungerte. Tillatelser, revisjonslogger, hastighetsgrenser og fakturering er alle knyttet til en identitet, og hvis sjekken som fastsatte identiteten, er svak, blir hver av dem trofast brukt på feil part. En logglinje som sier at en administrator slettet en tabell, registrerer hvilken konto som gjorde det, og bare god autentisering gjør det om til hvem.

Den praktiske beslutningen er hvor sterk sjekken må være for hver dør, og NIST SP 800-63B gir den tydeligste skalaen å bestemme ut fra. Det første sikkerhetsnivået, AAL1, godtar én enkelt faktor. AAL2 krever to faktorer, og en verifikator på det nivået må tilby minst ett alternativ som er motstandsdyktig mot phishing. AAL3 krever en autentikator som er motstandsdyktig mot phishing, og der den private nøkkelen ikke kan eksporteres. Retningslinjen er rett på sak om ett populært valg: koder som skrives inn for hånd, enten de sendes på SMS eller vises i en app, er ikke motstandsdyktige mot phishing, fordi ingenting knytter sifrene som skrives inn, til nettstedet som ba om dem.

Revisjon 4 av samme dokument, som erstattet revisjon 3 den 1. august 2025, avgjorde også flere diskusjoner om passord. Et passord som brukes som eneste faktor, må være minst 15 tegn langt, og et som bare brukes inne i en flerfaktorinnlogging, minst 8. Verifikatorer kan ikke stille andre krav til sammensetning og kan ikke kreve jevnlige bytter, men må tvinge fram et bytte

når det finnes tegn på at passordet er kompromittert. Nye passord sjekkes mot en sperreliste over vanlige og lekkede passord, brukeren får vite hvorfor et valg ble avvist, og formen som lagres, er saltet og hashet med en metode som har en kostnadsfaktor. Antall mislykkede forsøk på rad mot én autentikator er begrenset til 100. For økter på AAL2 anbefaler den å spørre på nytt minst hver 24. time, og etter en time uten aktivitet.

Vår anbefaling følger av dette. Der brukerne allerede har en konto hos en kompetent identitetsleverandør, bør du la leverandøren gjøre sjekken og godta det signerte svaret, fordi et passord applikasjonen aldri lagrer, er et passord den aldri kan lekke. Der applikasjonen må ta vare på passord, bør du følge 800-63B ordrett og ikke etter hukommelsen, og sette en metode som er motstandsdyktig mot phishing, foran alt som har med administrasjon å gjøre.

Slik ser det ut i praksis

Da selskapet flyttet kildekoden sin over på sin egen git-server, var autentisering det første punktet på listen over hva det betyr å drive en forge. Forgen som ble valgt, er SSH-først og har ikke noe administrativt webgrensesnitt, så veien inn er en nøkkel. Hvert innloggingsforsøk må bære en: forgen avviser anonyme forsøk og forsøk uten nøkkel, og hvert deklarert repo er merket som privat. Påloggingen skjer mot en SSH-lytter som bare er bundet til loopback-grensesnittet, port 23231, og den som kommer utenfra, går via en hoppvert, slik at brannmurens deklarerte flate holder seg like stor.

Den samme bekymringen ser annerledes ut i en hel organisasjon. I en tidligere rolle som infrastrukturingeniør gikk arbeidet ut på å samle identiteter og tilganger på tvers av M365, 1Password, Red Hat SSO og OKTA SSO, der tillatelser var blitt gitt ad hoc, over tid, av forskjellige personer. Minste privilegium ble håndhevet, og onboarding og offboarding ble standardisert slik at tilganger ble trukket tilbake raskt i stedet for å bli hengende igjen etter at noen hadde gått videre. Risikoen for uautorisert tilgang falt kraftig, og tilgangene ble mulige å revidere: teamet kunne endelig svare på «hvem kan nå dette, og hvorfor».

Autentisering har også et ansikt, og brukerne bedømmer den etter feilmeldingene. Ved innloggingen i en applikasjon mot en stor e-postleverandør sa meldingen at passordet var feil. Passordet var riktig; leverandøren krever et appspesifikt passord for tredjepartsklienter og avviser kontopassordet uansett hvor nøye det skrives inn. Meldingen ble skrevet om til å nevne leverandøren, si at den krever et appspesifikt passord for andre klienter, og si hvor du lager et. En

test fremkaller nå den avvisningen og sjekker at meldingen inneholder leverandørens navn og typen påloggingsinformasjon som kreves, så den gamle formuleringen kan ikke komme tilbake uten at noen merker det.

Hvor det går galt

- **Meldingen gir brukeren skylden for en årsak som ligger et annet sted.** En låst konto, et utløpt passord, en klokke som har glidd ut av takt med en kodegenerator og en leverandør som krever en annen type påloggingsinformasjon, kommer alle fram som «feil passord», og brukeren skriver på nytt noe som aldri kommer til å virke. Nevn årsaken brukeren kan gjøre noe med, men hold svaret likt for en konto som finnes og en som ikke finnes, så skjemaet ikke gir noen en liste over gyldige brukernavn.
- **En phishing-side videresender koden i sanntid.** Brukeren skriver inn et passord og en sekssifret kode på en kopi av den ekte siden, som sender begge videre til det ekte nettstedet innen de 30 sekundene RFC 6238 anbefaler som standard tidssteg. Engangskoder stopper stjålne og gjenbrukte passord, men gjør lite mot en videresending i sanntid. Nøkler som er bundet til en opprinnelse, som passnøkler, signerer bare for ett nettsted, og det er derfor NIST regner dem som motstandsdyktige mot phishing og innskrevne koder ikke.
- **Påloggingsinformasjon lever lenger enn folkene som hadde den.** Offboarding stenger kontoen i katalogtjenesten, mens en SSH-nøkkel i en `authorized_keys` -fil, et personlig tilgangstoken og et delt administratorpassord fortsetter å virke. Hver enkelt trenger en eier, et sted der den er registrert, og en dato da noen ser på den igjen.
- **Økten er gyldig lenge etter innloggingen.** En informasjonskapsel eller et token som varer i flere måneder, er et passord med lang lunte. Utlogging må avslutte økten på serveren, ikke bare slette informasjonskapselen i én nettleser, og et selvstendig token som serveren aldri sjekker opp mot noe, kan brukes til det utløper, uansett hva som skjer med kontoen.
- **Gjenoppretting av kontoen er den svakeste inngangen.** En sterk innlogging ved siden av en lenke for tilbakestilling sendt til en e-postadresse gjør postkassen til den egentlige påloggingsinformasjonen. Gjenoppretting fortjener den samme granskingen som inngangsdøra, fordi en angriper velger den som er svakest.

Les mer

- NIST SP 800-63B-4 — den gjeldende teksten: sikkerhetsnivåene, minstekravet på 15 tegn for passord og hva som regnes som motstandsdyktig mot phishing.

- NIST SP 800-63B, revision 3 — retningslinjen for autentisering fra 2017, som fortsatt ligger ute og er merket som erstattet av revisjon 4 fra 1. august 2025.
- RFC 9110, section 11: HTTP Authentication — utfordring og svar bak 401, WWW-Authenticate og Authorization-hodet.
- RFC 4252: The SSH Authentication Protocol — innlogging med offentlig nøkkel, passord og vertsbasert innlogging slik SSH definerer dem.
- RFC 6238: TOTP — hvordan autentiseringsapper avleder de korte kodene sine fra en delt hemmelighet og klokka.
- Web Authentication, Level 3 — W3C-API-et bak passnøkler, og påloggingsinformasjon med offentlig nøkkel avgrenset til én opprinnelse.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er autentisering?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/authentication/