Engineer

Passord

∗ en hemmelig streng som beviser hvem som logger inn

Hva det er

Et passord er en tegnstreng en person har i hodet, eller i en passordbehandler, og oppgir for å vise at en konto er deres. Definisjonene i ordlisten til NIST er enige om kjernen: bokstaver, tall og andre symboler som brukes til å autentisere en identitet eller bekrefte at tilgang er autorisert, og i enkelte dokumenter også til å avlede kryptografiske nøkler. Den gjeldende retningslinjen for digital identitet, SP 800-63-4, kaller det faktoren «noe du vet» og opplyser at den første utgaven brukte navnet memorized secret, en hemmelighet man husker. En passfrase, satt sammen av en rekke ord, og en PIN-kode, som bare består av sifre, er begge passord i videre forstand.

Den samme strengen kan gjøre to jobber som ligner hverandre. Et passord som brukes til å logge inn, sendes til en verifikator (verifier), som tar vare på nok til å kjenne det igjen og aldri bør ta vare på selve passordet. Et passord som låser opp noe lokalt, blir værende på enheten. SP 800-63B kaller den typen en aktiveringshemmelighet (activation secret), og det vanlige eksempelet er PIN-koden som låser opp en flerfaktorautentikator: den brukes på enheten for å komme til hemmeligheten autentikatoren holder på.

Vanskeligheten er at et passord er en delt hemmelighet. Brukeren kjenner det, tjenesten må kjenne det igjen, og alle som får vite det, kan bruke det nøyaktig slik brukeren ville gjort. Ingenting i strengen knytter den til nettstedet som ba om den, og derfor sier SP 800-63B rett ut at passord ikke er motstandsdyktige mot phishing. Det meste av ingeniørarbeidet rundt passord handler om å begrense hva den delte kunnskapen koster når den lekker.

Bruk begrepet i en beslutning

Bruk «Hva er et passord?» 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

Et passord er ofte det eneste som vokter en konto. To typer gjetting betyr noe, og det er ulike ting som holder hver av dem tilbake.

Gjetting på nett (online) betyr å prøve passord mot det aktive innloggingsskjemaet, og der er det tjenesten som bestemmer tempoet. SP 800-63B setter et tak på 100 mislykkede forsøk på rad med én autentikator på én konto, og krever at nye passord sjekkes mot en sperreliste med vanlige, forventede og lekkede verdier, så de valgene en angriper ville prøve først, aldri blir godtatt.

Gjetting offline begynner når en kopi av de lagrede passorddataene har forlatt huset. Da finnes det verken innloggingsskjema eller grense. Retningslinjen påpeker at angripere kan regne ut mange milliarder hasher i sekundet offline, og at passord som skal tåle dette, må være flere størrelsesordener mer komplekse enn passord som bare møter et innloggingsskjema. Derfor er det lagringen som avgjør utfallet. SP 800-63B krever at passord saltes og hashes med en passordhashingmetode (password hashing scheme), som tar passordet, et salt og en kostnadsfaktor og finnes for å gjøre hvert gjett dyrt. Saltet må være på minst 32 bit og velges slik at saltverdier sjelden kolliderer, noe som hindrer at to kontoer med samme passord får samme lagrede hash. Kostnadsfaktoren bør være så høy som verifikatoren har råd til, og økes over tid. Retningslinjen anbefaler også å registrere hvilken metode og hvilken kostnad som ga hver hash, så systemet senere kan gå over til en sterkere.

RFC 9106 beskriver Argon2, en minnekrevende (memory-hard) funksjon laget for akkurat denne jobben, og anbefaler varianten Argon2id: ett gjennomløp over 2 GiB minne som standard i alle miljøer, eller tre gjennomløp over 64 MiB der det er lite minne. RFC-en sier at den første innstillingen maksimerer kostnadene for en angriper med spesialbygd maskinvare for brute force, og det er den

egenskapen som teller når en hashfil har lekket. OWASP Password Storage Cheat Sheet gir lavere minimumsverdier for praktisk bruk og alternativer for systemer som ikke kan bruke Argon2id, og sier rett ut regelen som gjelder under dem alle. Raske hashfunksjoner som SHA-256 egner seg ikke, fordi de lar en angriper gjette svært mange ganger på kort tid, og passord hashes i stedet for å krypteres, fordi kryptering går begge veier og originalen kan hentes frem igjen.

Slik ser det ut i praksis

Før et passordhåndteringssystem for hele selskapet ble innført, ble påloggingsinformasjon lagret og delt på den måten hver enkelt foretrakk. Siden beskriver hvordan det ser ut i hverdagen: sjelden et dramatisk sikkerhetsbrudd, oftere et passord i en chatmelding, en delt innlogging ingen roterer, og en langsom opphopning av små eksponeringer. Systemet ga organisasjonen én konsekvent måte å lagre og dele påloggingsinformasjon på, og hovedpoenget på siden er at det ikke var valgfritt per person, fordi en passordbehandler som bare halve teamet bruker, hjelper knapt når risikoen ligger i halvdelen som lot være. Da påloggingsinformasjonen først lå på ett forvaltet sted, ble tilgangskontrollen enklere og mer konsekvent. SP 800-63B peker samme vei fra tjenestens side: verifikatorer må tillate passordbehandlere og autoutfylling, bør tillate innliming, og retningslinjen nevner en passordbehandler som måten å holde et eget passord for hver tjeneste på.

Da en server nummer to ble klargjort fra kode, var det den første maskinens historikk som ga argumentet. Den var opprettet for hånd og fikk brannmuren sin etterpå, noe som etterlater et vindu der en fersk vert med en standard passordpolicy kan nås fra internett. Løsningen er rekkefølgen: den nye maskinen blir født bak brannmuren sin, fordi brannmur-playet kjører før server-playet.

Ved en innlogging hos en stor e-postleverandør fortalte applikasjonen brukeren at passordet var feil. Det var det ikke. Den leverandøren krever et appspesifikt passord for tredjepartsklienter, så én konto hadde to typer passord, og en melding som bare kjente til den ene, sendte brukeren tilbake for å skrive inn noe som aldri kunne fungere. Den omskrevne meldingen nevner leverandøren, sier at den krever et appspesifikt passord, og sier hvor man lager et, og en test sjekker at leverandørens navn og typen påloggingsinformasjon står i meldingen.

Hvor det går galt

Ett passord, mange kontoer. Når det samme passordet beskytter flere tjenester, åpner et innbrudd hos den svakeste resten. SP 800-63B kaller dette password stuffing: et passord tatt fra ett nettsted brukes mot den samme personens kontoer på andre nettsteder. Brukere forsvarer seg med et eget passord per tjeneste, som i praksis betyr en passordbehandler, og tjenester ved å sjekke nye passord mot verdier fra tidligere lekkasjer.

Regler som gjør passordene svakere. Krav til sammensetning føles strenge og gir forutsigbare resultater. Retningslinjens eget eksempel er en bruker som ville ha valgt «password», men velger «Password1» når det kreves en stor bokstav og et siffer, og «Password1» med et utropstegn bakerst når det også kreves et symbol. Den utelukker regler for sammensetning, planlagte passordbytter, hint som kan leses før innlogging og sikkerhetsspørsmål som navnet på det første kjæledyret, og krever fortsatt et bytte så snart det finnes tegn på at passordet er kompromittert.

En rask hash, eller ett og samme salt for alle radene. En passordtabell som er hashet med en generell funksjon, eller saltet med den samme verdien hele veien, gir en angriper billige gjett offline som kan prøves mot alle kontoene på én gang. Systemer som arver gamle hasher, kan hashe hvert passord på nytt med den nye metoden neste gang brukeren logger inn, slik OWASP-veiledningen beskriver, og den lagrede referansen til metoden viser hvilke rader som fortsatt står på den gamle.

Passordet blir kuttet, eller innliming blir sperret. SP 800-63B krever at hele det innsendte passordet verifiseres, aldri en avkortet del, og vil at innliming skal være tillatt, så passordbehandlere virker der autoutfylling ikke gjør det. Noen hashfunksjoner har sin egen grense: OWASP-veiledningen påpeker at bcrypt i de fleste implementasjoner tar imot høyst 72 byte inndata, og ber tjenester som bruker den, håndheve det som maksimum.

Passordet havner der det aldri var ment å være. Et passord limt inn i en chat, skrevet inn i en sak eller skrevet ut i en logg er en kopi utenfor hver kontroll det ekte lagringsstedet har. Å bytte det er den eneste måten å ta tilbake den kopien på, og derfor betyr ett forvaltet sted mer enn noen regel om hvordan passordet ser ut.

Les mer

- Password, in the NIST CSRC glossary — definisjoner fra FIPS 140-2, SP 800-63-4 og flere, med passfrase som et spesialtilfelle.
- NIST SP 800-63B-4 — de gjeldende reglene for å velge, sjekke og lagre passord, med begrunnelsene i vedlegg A.
- RFC 9106 — Argon2, med de to anbefalte innstillingene for Argon2id til passordhashing.
- OWASP Password Storage Cheat Sheet — praktiske parametere for Argon2id, scrypt, bcrypt og PBKDF2, og hvordan gamle hasher oppgraderes.

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 et passord?» 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/password/