Engineer

Lösenord

∗ en hemlig sträng som bevisar vem som loggar in

Vad det är

Ett lösenord är en teckensträng som en person har i minnet, eller i en lösenordshanterare, och visar upp för att styrka att ett konto är hens. Definitionerna i ordlistan från NIST är eniga om kärnan: bokstäver, siffror och andra tecken som används för att autentisera en identitet eller verifiera behörighet till åtkomst, och i vissa dokument också för att härleda kryptografiska nycklar. Den gällande riktlinjen för digital identitet, SP 800-63-4, kallar det faktorn ”something you know”, något man vet, och noterar att den första utgåvan använde namnet memorized secret. En lösenfras, som består av en följd av ord, och en PIN-kod, som består av enbart siffror, är båda lösenord i vidare mening.

Samma sträng kan göra två jobb som liknar varandra. Ett lösenord som används för att logga in skickas till en verifierare, som behåller tillräckligt för att känna igen det och aldrig bör behålla själva lösenordet. Ett lösenord som används för att låsa upp något lokalt stannar på enheten. SP 800-63B kallar den sorten en aktiveringshemlighet, med PIN-koden som låser upp en autentiserare för flerfaktorsautentisering som det vanliga exemplet: den används på enheten för att nå den hemlighet som autentiseraren håller.

Svårigheten är att ett lösenord är en delad hemlighet. Användaren kan det, tjänsten måste känna igen det, och den som får reda på det kan använda det precis som användaren skulle. Själva strängen binder det inte till den webbplats som bad om det, och därför slår SP 800-63B fast att lösenord inte står emot nätfiske. Det mesta av ingenjörsarbetet kring lösenord handlar om att begränsa vad den delade kunskapen kostar när den läcker.

Använd begreppet i ett beslut

Använd ”Vad är ett lösenord?” 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

Ofta är ett lösenord det enda som skyddar ett konto. Två sorters gissning spelar roll, och olika saker håller tillbaka var och en av dem.

Gissning online innebär att pröva lösenord mot det aktiva inloggningsformuläret, och där bestämmer tjänsten takten. SP 800-63B begränsar antalet misslyckade försök i följd med en autentiserare mot ett konto till 100, och kräver att nya lösenord kontrolleras mot en spärrlista med vanliga, förväntade och läckta värden, så att de val som en angripare skulle pröva först aldrig godtas.

Gissning offline börjar när en kopia av de lagrade lösenordsdata har lämnat huset. Då finns varken formulär eller gräns. Riktlinjen noterar att angripare kan beräkna många miljarder hashvärden per sekund offline, och att lösenord som ska klara det måste vara flera tiopotenser mer komplexa än sådana som bara möter ett inloggningsformulär. Lagringen avgör alltså utgången. SP 800-63B kräver att lösenord saltas och hashas med en metod för lösenordshashning, som tar lösenordet, ett salt och en kostnadsfaktor och finns till för att göra varje gissning dyr. Saltet måste vara minst 32 bitar och väljas så att saltvärden sällan krockar, vilket hindrar två konton med samma lösenord från att dela samma lagrade hashvärde. Kostnadsfaktorn bör vara så hög som verifieraren har råd med och höjas med tiden. Riktlinjen rekommenderar också att det sparas vilken metod och kostnad som gav varje hashvärde, så att systemet senare kan gå över till en starkare.

RFC 9106 beskriver Argon2, en minneskrävande funktion byggd för just det här jobbet, och rekommenderar varianten Argon2id: en genomgång över 2 GiB minne som standard för alla miljöer, eller tre genomgångar över 64 MiB där minnet är knappt. Enligt RFC:n maximerar den första inställningen kostnaderna för en angripare med specialbyggd hårdvara för brute force, och det är den egenskapen som räknas när en fil med hashvärden väl har läckt. OWASP

Password Storage Cheat Sheet ger lägre minimivärden för praktisk användning och reservlösningar för system som inte kan använda Argon2id, och anger regeln som ligger under dem alla. Snabba hashfunktioner som SHA-256 duger inte, eftersom de låter en angripare göra stora mängder gissningar snabbt, och lösenord hashas i stället för att krypteras, eftersom kryptering går åt båda hållen och originalet kan återskapas ur den.

Så ser det ut i praktiken

Tre beslut om lösenord ligger bakom arbetet nedan: att hela företaget förvarar dem i samma hanterare, att en ny server aldrig går att nå utan sin brandvägg och att ett felmeddelande skiljer på två sorters lösenord.

Innan ett system för lösenordshantering infördes i hela företaget lagrades och delades inloggningsuppgifter på det sätt som var och en föredrog. Sidan beskriver hur det ser ut i vardagen: sällan ett dramatiskt intrång, oftare ett lösenord i ett chattmeddelande, en delad inloggning som ingen byter och en långsam ansamling av små exponeringar. Systemet gav organisationen ett enda konsekvent sätt att lagra och dela inloggningsuppgifter, och sidans huvudpoäng är att det inte var frivilligt för den enskilda, eftersom en lösenordshanterare som bara halva teamet använder knappt hjälper när risken bor i den halva som låter bli. När inloggningsuppgifterna väl fanns på ett enda hanterat ställe blev åtkomstkontrollen enklare och mer konsekvent. SP 800-63B pekar åt samma håll från tjänstens sida: verifierare måste tillåta lösenordshanterare och autofyll, bör tillåta inklistring, och riktlinjen nämner en lösenordshanterare som sättet att ha ett eget lösenord för varje tjänst.

När en andra server provisionerades från kod var det den första maskinens historik som stod för argumentet. Den hade skapats för hand och fått sin brandvägg i efterhand, vilket lämnar ett fönster där en ny värddator med en standardpolicy för lösenord går att nå från internet. Lösningen är ordningsföljden: den nya maskinen föds bakom sin brandvägg, eftersom brandväggens play körs före serverns play.

Vid en inloggning mot en stor e-postleverantör sa applikationen till användaren att lösenordet var fel. Det var det inte. Leverantören kräver ett applikationsspecifikt lösenord för klienter från tredje part, så ett och samma konto hade två sorters lösenord, och ett meddelande som bara kände till den ena skickade användaren till att skriva om något som aldrig kunde fungera. Det

omskrivna meddelandet namnger leverantören, anger att den kräver ett appspecifikt lösenord och säger var man skapar ett, och ett test kontrollerar att leverantörens namn och typen av inloggningsuppgift finns med.

Var det går fel

Ett lösenord, många konton. När samma lösenord skyddar flera tjänster öppnar ett intrång hos den svagaste resten. SP 800-63B kallar det password stuffing: ett lösenord som tagits från en webbplats används mot samma persons konton på andra webbplatser. Användare skyddar sig med ett eget lösenord per tjänst, vilket i praktiken innebär en lösenordshanterare, och tjänster genom att kontrollera nya lösenord mot värden från tidigare intrång.

Regler som gör lösenorden svagare. Krav på sammansättning känns stränga och ger förutsägbara resultat. Riktlinjens eget exempel är en användare som skulle ha valt ”password” och i stället väljer ”Password1” när en versal och en siffra krävs, och ”Password1” med ett utropstecken sist när ett specialtecken också krävs. Den utesluter sammansättningsregler, schemalagda byten, ledtrådar som går att läsa före inloggningen och säkerhetsfrågor som namnet på ens första husdjur, och den kräver fortfarande ett byte så snart det finns tecken på att lösenordet har röjts.

En snabb hashfunktion, eller samma salt på varje rad. En lösenordstabell som hashats med en allmän funktion, eller saltats med samma värde rakt igenom, ger en angripare offline billiga gissningar som kan prövas mot varje konto på en gång. System som ärver gamla hashvärden kan hasha om varje lösenord med den nya metoden vid användarens nästa inloggning, så som OWASP beskriver, och den sparade uppgiften om metod visar vilka rader som fortfarande ligger kvar på den gamla.

Lösenordet kortas av, eller går inte att klistra in. SP 800-63B kräver att hela det inskickade lösenordet verifieras, aldrig en avkortad del, och vill att inklistring tillåts så att lösenordshanterare fungerar där autofyll inte gör det. Vissa hashfunktioner har en egen gräns: OWASP noterar att bcrypt tar högst 72 byte indata i de flesta implementationer och uppmanar tjänster som använder den att upprätthålla det maxvärdet.

Lösenordet hamnar där det aldrig var tänkt att vara. Ett lösenord som klistrats in i en chatt, skrivits in i ett ärende eller skrivits ut i en logg är en kopia utanför varje kontroll som det riktiga förvaret har. Att byta lösenordet är det enda sättet

att ta tillbaka den kopian, och därför betyder ett enda hanterat ställe mer än någon regel om hur lösenordet ser ut.

Läs mer

- Password, in the NIST CSRC glossary — definitioner ur FIPS 140-2, SP 800-63-4 med flera, med lösenfras som ett specialfall.
- NIST SP 800-63B-4 — de gällande reglerna för att välja, kontrollera och lagra lösenord, med skälen i bilaga A.
- RFC 9106 — Argon2, med de två rekommenderade inställningarna för Argon2id vid lösenordshashning.
- OWASP Password Storage Cheat Sheet — praktiska parametrar för Argon2id, scrypt, bcrypt och PBKDF2, och hur gamla hashvärden uppgraderas.

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 ett lösenord?” 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/password/