Engineer

Härdning

att ta bort det som en angripare skulle kunna använda

Vad det är

Härdning är arbetet med att konfigurera ett system så att det exponerar och tillåter bara det som dess uppgift kräver: färre tjänster som körs, färre portar som lyssnar, färre privilegier för varje process och strängare standardvärden på allt som blir kvar. Mjukvaran är densamma före och efter. Det som ändras är hur mycket av den en inkräktare kan nå och hur långt ett enda fotfäste räcker. Du ser det också kallas systemhärdning, serverhärdning, härdning av operativsystemet eller SSH-härdning, beroende på vilket lager som stramas åt.

Ett typiskt härdningspass på en Linux-server rör flera lager samtidigt. Fjärrinloggning tar emot nycklar och avvisar lösenord. Root-kontot kan inte logga in direkt. Paket som ingen använder tas bort i stället för att bara stoppas. Varje tjänst som körs länge får en egen användare utan privilegier, en skrivskyddad vy av filsystemet och en lista över de systemanrop den får göra. Brandväggen släpper in den handfull portar som värddatorn finns till för att betjäna. Stegen är var för sig helt vardagliga; disciplinen ligger i att göra alla, på varje värddator, och se till att de förblir sanna.

Det hjälper att placera härdning bredvid de metoder som den ofta förväxlas med. Patchning rättar en specifik brist i koden när en rättning finns. Övervakning och intrångsdetektering märker en attack medan den pågår. En brandvägg avgör vem som får nå en port. Härdning avgör vad programmet bakom porten får göra när det väl har nåtts, och om porten över huvud taget ska vara öppen. Publicerade baslinjer som CIS Benchmarks gör om det här till checklistor per operativsystem och produkt, vilket är användbart som golv och missvisande som mållinje.

Använd begreppet i ett beslut

Använd ”Vad är härdning?” 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

Det verkliga beslutet gäller var en begränsad säkerhetsbudget ska användas, och härdning är ovanlig genom att den lönar sig mot sårbarheter som hittas först nästa år. En patch skyddar mot en känd sårbarhet. En härdad tjänst begränsar vad en kompromitterad process kan göra. Exempelvis är förbud mot att skriva till körbart minne, öppna råa socketar och komma åt hemkataloger utformade för att begränsa följderna av en bugg som tillåter fjärrkörning av kod. Patchning täpper till hål; härdning sänker vad varje hål är värt. En seriös drift behöver båda, och det är härdningen som köper tid under dagarna mellan att en sårbarhet offentliggörs och att rättningen kommer på plats.

Kostnaden är mest ingenjörstid och risken att något går sönder, snarare än licenser. Varje begränsning är ett vad om att tjänsten aldrig behöver det som tas bort, och en del av de vaden förloras: ett jobb för säkerhetskopiering som körs en gång per natt, en loggrotation som skickar en signal, ett insticksprogram som anropar skalet. Därför brukar härdning som görs för hand på en server i drift stanna av. Den som gör det bränner sig en gång på en begränsning som slog sönder något klockan tre på natten och slutar strama åt. Görs samma arbete som kod, med begränsningarna granskade, versionshanterade och tillämpade likadant vid varje ombyggnad, blir det billigt att upprepa och säkert att backa.

För en köpare är de praktiska frågorna när det ska göras och hur man vet att det blev gjort. Svaret på den första är vid byggtiden, som en del av den automatisering som skapar värddatorn, eftersom en checklista som tillämpas en gång på en maskin i drift glider isär inom några månader. Svaret på den andra är belägg från det körande systemet snarare än från konfigurationsrepot. Revisorer frågar allt oftare efter båda: en dokumenterad baslinje och bevis för att produktionen stämmer med den.

Så ser det ut i praktiken

På företagets egen produktionsvärd är SSH den enda interaktiva vägen in, och dess konfiguration skrivs av automatisering. I september 2026 läste en åtkomstrapport tillbaka de inställningar som daemonen faktiskt tillämpade och fann två av de avsedda direktiven åsidosatta: operativsystemets standardvärden lästes in före företagets drop-in-fil, och för de inställningarna vinner det första värdet. Ett filnamnsprefix som sorteras tidigare rättade ordningen. Den bestående förändringen var validering i tre steg av 24 SSH-direktiv vid varje konvergenskörning: kandidatfilen syntaxkontrolleras innan den installeras, den sammansatta konfigurationen kontrolleras efteråt, och daemonens egen utdata med de gällande värdena läses tillbaka och kontrolleras mot alla 24. Ytterligare två direktiv utelämnades medvetet, eftersom daemonen inte längre implementerar dem.

Samma värddator kör 13 enhetsfiler som har installerats av dess egen automatisering, och de hade körts med de privilegier som tjänstehanteraren ger som standard. Varje enhetsmall fick ett privilegietak, ett filter för systemanrop, begränsade adressfamiljer och minne som får vara skrivbart eller körbart men aldrig båda. Mätt med tjänstehanterarens egen poängsättning flyttade sandlådearbetet en tjänst för dödmansgrepp från 9,6 till 1,5, den internetvända forgen för git från 8,3 till 1,5 och de två enheterna för säkerhetskopiering från 9,6 till 2,3 respektive 2,5. När de skrevs visade det sig att ett direktiv var felstavat i tre mallar; tjänstehanteraren loggar en okänd nyckel och startar enheten ändå. Varje enhet går nu igenom tjänstehanterarens egen verifierare under konvergenskörningen.

En härdningsgranskning i augusti 2026 frågade om skydden mot brute force-attacker via SSH fungerade, och läste det körande systemet för att ta reda på det. Alla tre skydden hade varit verkningslösa sedan värddatorn byggdes: jailen för spärrning av intrångsförsök bevakade port 22 medan daemonen lyssnade på 1986, en hastighetsbegränsning i brandväggen låg under en bredare regel som matchade först, och spärråtgärden anropade en binärfil för paketfiltrering som paketsystemet bara rekommenderar och som saknades. Två regler kom ut av granskningen och styr fortfarande repot: en säkerhetsåtgärd får en kontroll och inte en kommentar, och en brandvägg verifieras utifrån reglernas position och inte utifrån att de finns.

Var det går fel

Inställningen står i filen och tjänsten ignorerar den. Många daemoner läser flera filer och behåller det första eller det sista värde de möter för varje nyckel. Ett härdat direktiv i en drop-in-fil kan förlora mot ett standardvärde från leverantören som läses in tidigare. Fråga programmet vad det faktiskt kommer att göra, för SSH med `sshd -T`, och jämför den utdatan med de avsedda värdena vid varje körning.

Ett stavfel gör en åtgärd till en kommentar. Tjänstehanterare och många daemoner loggar en okänd nyckel och fortsätter. Enheten startar, dashboarden är grön, och begränsningen saknas. Kör parserns egen verifierare som en del av driftsättningen, så att ett okänt direktiv stoppar utrullningen.

```
# What the SSH daemon will do once every Include is resolved
sshd -T | grep -E '^(passwordauthentication|permitrootlogin) '
# How much a unit's sandbox still exposes, scored 0.0 to 10.0
systemd-analyze security --no-pager example.service
# Unknown directives and other unit-file errors
systemd-analyze verify /etc/systemd/system/example.service
```

Ett skydd vaktar fel dörr. En port flyttas och spärregeln fortsätter att bevaka den gamla. En hastighetsbegränsning läggs till under en bred tillåtande regel som redan har matchat. Åtgärden rapporterar att den mår bra eftersom den körs, vilket inte säger något om huruvida den någonsin slår till. Testa varje åtgärd utifrån dess effekt: utlös den med avsikt utifrån och bekräfta att paketet kastas.

Sandlådan får tjänsten att gå sönder på natten. Ett filter för systemanrop eller en skrivskyddad sökväg slår sönder den sällsynta kodvägen, t.ex. den veckovisa exporten eller skriptet som säkerhetskopieringen anropar, och felet visar sig timmar efter ändringen, vid körning snarare än vid parsning. Strama åt i etapper, kör varje schemalagt jobb efter varje etapp och håll ögonen på journalen efter processer som filtret har dödat.

Poängen i en benchmark blir målet. En värddator kan klara varje rad i en generisk checklista medan dess enda internetvända applikation körs som root med skrivåtkomst till sin egen kod. Börja i det som går att nå från nätverket och det som varje nåbar process skulle kunna röra, och använd sedan benchmarken för att fånga det som granskningen missade.

Läs mer

- Hardening, in the NIST glossary — den definition som NIST använder, med publikationen den är hämtad från.
- sshd_config(5) — varje direktiv för SSH-servern, och regeln att det första värde som erhålls för ett nyckelord är det som används.
- systemd.exec(5) — direktiven för att sätta tjänster i en sandlåda: filter för systemanrop, adressfamiljer och minne som är antingen skrivbart eller körbart.
- systemd-analyze(1) — verbet `security` och dess exponeringspoäng, och verbet `verify` som fångar okända direktiv.
- CIS Benchmarks — konsensusbaserade baslinjer för konfiguration av operativsystem, molnplattformar och servermjukvara.

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 härdning?” 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/hardening/