Engineer

Hærdning

at fjerne det, en angriber kunne bruge

Hvad det er

Hærdning er arbejdet med at konfigurere et system, så det kun åbner for og tillader det, opgaven kræver: færre kørende tjenester, færre lyttende porte, færre rettigheder til hver proces og strammere standardværdier på alt det, der bliver tilbage. Programmerne er de samme før og efter. Forskellen ligger i, hvor meget en ubuden gæst kan nå, og hvor langt et enkelt fodfæste rækker. På dansk hører man både hærdning, systemhærdning og serverhærdning, og mange it-folk bruger blot det engelske hardening, ofte med lag foran: OS-hardening eller SSH-hardening.

En typisk hærdning af en Linux-server rører flere lag på én gang. Fjernlogin tager imod nøgler og afviser adgangskoder. Root-kontoen kan ikke logge direkte ind. Pakker, som ingen bruger, bliver afinstalleret i stedet for bare at blive stoppet. Hver tjeneste får sin egen bruger uden privilegier, et skrivebeskyttet syn på filsystemet og en liste over de systemkald, den må foretage. Firewallen lukker kun de få porte op, som værten er sat i verden for at betjene. Intet af det er eksotisk. Disciplinen ligger i at gøre det hele, på hver vært, og blive ved med at holde det sandt.

Hærdning bliver tit blandet sammen med beslægtede discipliner. Patching retter en bestemt fejl i koden, når der findes en rettelse. Overvågning og indtrængningsdetektion opdager et angreb, mens det står på. En firewall afgør, hvem der må nå en port. Hærdningen afgør, hvad programmet bag porten må gøre, når nogen først er nået frem, og om porten overhovedet skal være åben. Offentlige baselines som CIS Benchmarks gør det til tjeklister pr. styresystem og produkt. De er nyttige som bundlinje og vildledende som mållinje.

Brug begrebet i en beslutning

Brug »Hvad er hærdning?« 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

Det egentlige valg handler om, hvor et begrænset sikkerhedsbudget gør mest gavn, og hærdning har den sjældne egenskab, at den virker mod sårbarheder, der først bliver opdaget næste år. En patch beskytter mod én kendt sårbarhed. En hærdet tjeneste begrænser, hvad en kompromitteret proces kan gøre. For eksempel er forbud mod at skrive til eksekverbar hukommelse, åbne rå sockets og tilgå hjemmemapper udformet til at begrænse virkningen af en fejl, der giver fjernkørsel af kode. Patching lukker huller, hærdning gør hvert hul mindre værd. Begge dele hører med, og det er hærdningen, der køber tid i dagene mellem en offentliggørelse og rettelsens ankomst.

Prisen er primært ingeniørtid og risikoen for at ødelægge noget, sjældent licenser. Hver begrænsning er et væddemål om, at tjenesten aldrig får brug for det, der bliver taget fra den, og nogle væddemål tabes: et backupjob, der kører én gang i døgnet, en logrotation, der sender et signal, et plugin, der kalder en ekstern kommando. Derfor går manuel hærdning af en kørende server ofte i stå. Den, der står for den, bliver brændt én gang af en stramning, der væltede noget klokken tre om natten, og holder op med at stramme. Lagt i kode, gennemset, versioneret og anvendt ens ved hver genopbygning bliver det samme arbejde billigt at gentage og trygt at rulle tilbage.

For en køber er de praktiske spørgsmål, hvornår det skal ske, og hvordan man ved, at det er sket. Svaret på det første er ved opbygningen, som en del af den automatisering, der skaber værten, for en tjekliste, der én gang er kørt på en kørende maskine, glider inden for få måneder. Svaret på det andet er bevis fra det kørende system frem for fra konfigurationsrepositoryet. Revisorer spørger i stigende grad efter begge dele: en dokumenteret baseline og belæg for, at produktionen følger den.

Sådan ser det ud i praksis

På virksomhedens egen produktionsvært er SSH den eneste interaktive vej ind, og konfigurationen skrives af automatisering. I september 2026 læste en adgangsrapport dæmonens effektive indstillinger tilbage og viste, at to af de tilsigtede direktiver var overskrevet: styresystemets standarder blev indlæst før virksomhedens drop-in-fil, og for de indstillinger vinder den første værdi. Et filnavnspræfiks, der sorterer tidligere, rettede rækkefølgen. Den varige ændring blev validering i tre trin af 24 SSH-direktiver ved hver konvergering: kandidatfilen syntakstjekkes før installationen, den samlede konfiguration tjekkes bagefter, og dæmonens eget output med de effektive værdier læses tilbage og holdes op mod alle 24. To yderligere direktiver er bevidst udeladt, fordi dæmonen ikke længere implementerer dem.

Den samme vært kører 13 unit-filer, som dens egen automatisering installerer, og de kørte med servicemanagerens standardrettigheder. Hver unit-skabelon fik et privilegieloft, et systemkaldsfilter, begrænsede adressefamilier og hukommelse, der må være skrivbar eller eksekverbar, men aldrig begge dele. Målt med supervisorens egen score flyttede sandkassearbejdet en dead man's switch-tjeneste fra 9,6 til 1,5, den internetvendte git-forge fra 8,3 til 1,5 og de to backup-units fra 9,6 til 2,3 og 2,5. Undervejs viste det sig, at ét direktiv var stavet forkert i tre skabeloner. Supervisoren logger en ukendt nøgle og starter unitten alligevel. Nu går hver unit gennem supervisorens egen verifikator under konvergeringen.

En hærdningsgennemgang i august 2026 spurgte, om SSH-beskyttelserne mod brute force virkede, og læste det kørende system for at finde svaret. Alle tre beskyttelser havde været uvirksomme, siden værten blev bygget: jailen til blokering holdt øje med port 22, mens dæmonen lyttede på 1986, en hastighedsbegrænsning i firewallen lå under en bredere regel, der matchede først, og blokeringshandlingen kaldte et pakkefilterprogram, som pakkesystemet kun anbefaler, og som manglede på værten. Gennemgangen gav to regler, der stadig styrer repositoryet: en sikkerhedskontrol får en assertion frem for en kommentar, og en firewall efterprøves på reglernes placering frem for på deres tilstedeværelse.

Hvor det går galt

Indstillingen står i filen, og tjenesten overser den. Mange dæmoner læser flere filer og beholder den første eller den sidste værdi, de møder for hver nøgle. Et hærdet direktiv i en drop-in-fil kan tabe til en leverandørstandard, der indlæses

tidligere. Spørg programmet, hvad det faktisk vil gøre, for SSH med `sshd -T`, og sammenlign svaret med de tilsigtede værdier ved hver kørsel.

En stavefejl gør en kontrol til en kommentar. Servicemanagere og mange dæmoner logger en ukendt nøgle og fortsætter. Unitten starter, dashboardet er grønt, og begrænsningen mangler. Lad parserens egen verifikator køre som en del af udrulningen, så et ukendt direktiv stopper den.

```
# Hvad SSH-dæmonen gør, når alle Include-filer er læst
sshd -T | grep -E '^(passwordauthentication|permitrootlogin) '
# Hvor meget en units sandkasse stadig udstiller, fra 0,0 til 10,0
systemd-analyze security --no-pager example.service
# Ukendte direktiver og andre fejl i unit-filen
systemd-analyze verify /etc/systemd/system/example.service
```

Beskyttelsen vogter den forkerte dør. En port flyttes, og blokeringsreglen bliver ved med at holde øje med den gamle. En hastighedsbegrænsning lægges under en bred tilladelsesregel, der allerede har matchet. Kontrollen melder sund, fordi den kører, og det siger intet om, hvorvidt den nogensinde slår til. Afprøv hver kontrol på dens virkning: udløs den med vilje udefra, og bekræft, at datapakken bliver smidt væk.

Sandkassen vælter tjenesten om natten. Et systemkaldsfilter eller en skrivebeskyttet sti rammer den sjældne kodevej, for eksempel den ugentlige eksport eller det script, backuppen kalder, og fejlen viser sig timer efter ændringen, ved kørsel og ikke ved parsning. Stram i etaper, kør hvert planlagt job efter hver etape, og hold øje med journalen efter processer, som filteret har dræbt.

Benchmarkscoren bliver målet. En vært kan bestå hver linje i en generisk tjekliste, mens dens eneste internetvendte applikation kører som root med skriveadgang til sin egen kode. Begynd med det, der kan nås fra netværket, og med det, hver af de processer kan røre ved, og brug derefter benchmarken til at fange det, gennemgangen overså.

Læs mere

- Hardening, in the NIST glossary — NIST's definition af begrebet med den publikation, den stammer fra.

- sshd_config(5) — alle SSH-serverens direktiver og reglen om, at den første værdi for et nøgleord er den, der bruges.
- systemd.exec(5) — direktiverne til sandkasser for tjenester: systemkaldsfiltre, adressefamilier og skriv-eller-kør-hukommelse.
- systemd-analyze(1) — `security` -verbets eksponeringsscore og `verify` -verbet, der fanger ukendte direktiver.
- CIS Benchmarks — konsensusbaserede konfigurationsbaselines for styresystemer, cloudplatforme og serversoftware.

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 hærdning?« 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/hardening/