Engineer

Herding

å fjerne det en angriper kunne ha brukt

Hva det er

Herding er arbeidet med å konfigurere et system slik at det bare eksponerer og tillater det jobben krever: færre tjenester som kjører, færre porter som lytter, færre privilegier for hver prosess og strengere standardverdier for alt som blir igjen. Programvaren er den samme før og etter. Det som endrer seg, er hvor mye av den en inntrenger kan nå, og hvor langt ett enkelt fotfeste rekker. Du vil også se det kalt systemherding, serverherding, OS-herding eller SSH-herding, alt etter hvilket lag som strammes inn.

En typisk herdingsrunde på en Linux-server berører flere lag på en gang. Fjerninnlogging godtar nøkler og avviser passord. Root-kontoen er stengt for direkte innlogging. Pakker som ingen bruker, blir fjernet og ikke bare stoppet. Hver tjeneste som kjører over lang tid, får sin egen bruker uten privilegier, en skrivebeskyttet visning av filsystemet og en liste over systemkallene den får gjøre. Brannmuren slipper inn den håndfullen porter verten finnes for å betjene. Hvert av disse stegene er helt vanlig; disiplinen ligger i å gjøre alle sammen, på hver vert, og holde dem sanne over tid.

Det hjelper å plassere herding ved siden av praksisene den ofte forveksles med. Patching retter en bestemt feil i koden når det finnes en retting. Overvåking og inntrengningsdeteksjon legger merke til et angrep mens det pågår. En brannmur avgjør hvem som får nå en port. Herding avgjør hva programmet bak den porten får lov til å gjøre når noen har nådd det, og om porten i det hele tatt skal være åpen. Publiserte grunnlinjer som CIS Benchmarks gjør dette om til sjekklister per operativsystem og produkt, og de er nyttige som et gulv og villedende som en målstrek.

Bruk begrepet i en beslutning

Bruk «Hva er herding?» 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

Den egentlige avgjørelsen er hvor et begrenset sikkerhetsbudsjett skal brukes, og herding skiller seg ut ved at den lønner seg mot sårbarheter som først blir funnet neste år. En patch beskytter mot én kjent sårbarhet. En herdet tjeneste begrenser hva en kompromittert prosess kan gjøre. For eksempel skal forbud mot å skrive til kjørbart minne, åpne rå sockets og få tilgang til hjemmekataloger begrense virkningen av en feil som åpner for fjernkjøring av kode. Patching tetter hull; herding senker verdien av ethvert hull. En seriøs drift trenger begge deler, og det er herdingen som kjøper tid i dagene mellom at en sårbarhet blir offentlig kjent, og at rettingen er på plass.

Kostnaden er mest ingeniørtid og risikoen for at noe går i stykker, og lite i lisenser. Hver begrensning er et veddemål om at tjenesten aldri trenger det som blir tatt fra den, og noen av de veddemålene taper: en sikkerhetskopijobb som kjører én gang hver natt, en loggrotasjon som sender et signal, et programtillegg som starter et skall. Derfor har herding som gjøres for hånd på en server i drift, en tendens til å stoppe opp. Den som gjør jobben, blir brent én gang av en begrensning som ødela noe klokka tre om natten, og slutter å stramme inn. Gjort som kode, med begrensningene gjennomgått, versjonert og brukt helt likt ved hver ombygging, blir det samme arbeidet billig å gjenta og trygt å rulle tilbake.

For en kjøper er de praktiske spørsmålene når det skal gjøres, og hvordan man vet at det ble gjort. Svaret på det første er ved byggetid, som en del av automatiseringen som oppretter verten, fordi en sjekkliste som brukes én gang på en maskin i drift, glir ut i løpet av måneder. Svaret på det andre er bevis fra det kjørende systemet og ikke fra konfigurasjonsrepoet. Revisorer ber stadig oftere om begge deler: en dokumentert grunnlinje og bevis for at produksjonen samsvarer med den.

Slik ser det ut i praksis

På selskapets egen produksjonsvert er SSH den eneste interaktive veien inn, og konfigurasjonen skrives av automatisering. I september 2026 leste en tilgangsrapport daemonens effektive innstillinger tilbake og fant at to av de tiltenkte direktivene var overstyrt: operativsystemets standardverdier ble lastet før selskapets drop-in-fil, og for de innstillingene er det den første verdien som vinner. Et filnavnprefiks som sorteres tidligere, rettet rekkefølgen. Endringen som varte, var validering i tre trinn av 24 SSH-direktiver ved hver kjøring: kandidatfilen syntakssjekkes før den installeres, den sammensatte konfigurasjonen sjekkes etterpå, og daemonens egne effektive verdier leses tilbake og kontrolleres mot alle 24. To direktiver til ble utelatt med vilje, fordi daemonen ikke lenger implementerer dem.

Den samme verten kjører 13 enhetsfiler som er installert av dens egen automatisering, og de hadde kjørt med de privilegiene tjenestestyreren gir som standard. Hver enhetsmal fikk et privilegietak, et filter for systemkall, begrensede adressefamilier og minne som kan være skrivbart eller kjørbart, men aldri begge deler. Målt med tjenestestyrerens egen poengsum flyttet sandkassearbeidet en tjeneste for dødmannsknapp fra 9,6 til 1,5, den internetteksponerte git-forgen fra 8,3 til 1,5 og de to sikkerhetskopienhetene fra 9,6 til 2,3 og 2,5. Mens malene ble skrevet, viste det seg at ett direktiv var feilstavet i tre av dem; tjenestestyreren logger en nøkkel den ikke kjenner igjen, og starter enheten likevel. Hver enhet går nå gjennom tjenestestyrerens eget verifiseringsverktøy under kjøringen.

En herdingsgjennomgang i august 2026 spurte om vernet mot brute force-angrep på SSH virket, og leste det kjørende systemet for å finne ut av det. Alle tre beskyttelsene hadde vært virkningsløse siden verten ble bygd: jailen for utestenging overvåket port 22 mens daemonen lyttet på 1986, en hastighetsgrense i brannmuren lå under en bredere regel som slo til først, og utestengingshandlingen kalte et pakkefilterprogram som pakkesystemet bare anbefaler, og som manglet på verten. To regler kom ut av den gjennomgangen og styrer fortsatt repoet: et sikkerhetstiltak får en kontroll og ikke en kommentar, og en brannmur verifiseres ut fra hvor reglene står, og ikke ut fra at de finnes.

Hvor det går galt

Innstillingen står i filen, og tjenesten overser den. Mange daemoner leser flere filer og beholder den første eller den siste verdien de møter for hver nøkkel. Et herdet direktiv i en drop-in-fil kan tape mot en standardverdi fra leverandøren som lastes tidligere. Spør programmet hva det faktisk kommer til å gjøre, for SSH med `sshd -T`, og sammenlign utdataene med de tiltenkte verdiene ved hver kjøring.

En skrivefeil gjør et sikkerhetstiltak om til en kommentar. Tjenestestyrere og mange daemoner logger en ukjent nøkkel og fortsetter. Enheten starter, dashbordet er grønt, og begrensningen er borte. Kjør parserens eget verifiseringsverktøy som en del av produksjonssettingen, slik at et direktiv den ikke kjenner igjen, stopper utrullingen.

```
# 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
```

Et vern passer feil dør. En port flyttes, og utestengingsregelen fortsetter å overvåke den gamle. En hastighetsgrense legges inn under en bred tillatelsesregel som allerede har slått til. Kontrollen melder at den er frisk fordi den kjører, og det sier ingenting om den noen gang slår til. Test hver kontroll ut fra virkningen: utløs den med vilje utenfra, og bekreft at pakken blir forkastet.

Sandkassen knekker tjenesten om natten. Et filter for systemkall eller en skrivebeskyttet sti knekker den sjeldne kodestien, som den ukentlige eksporten eller kroken for sikkerhetskopiering, og feilen viser seg timer etter endringen, ved kjøretid og ikke ved parsing. Stram inn i trinn, kjør hver planlagte jobb etter hvert trinn, og følg med i journalen etter prosesser som filteret har drept.

Resultatet på sjekklisten blir selve målet. En vert kan bestå hver linje i en generell sjekkliste mens den ene internetteksponerte applikasjonen kjører som root med skrivetilgang til sin egen kode. Begynn med det som kan nås fra nettverket, og med hva hver prosess som kan nås, kunne rørt, og bruk så sjekklisten til å fange det den gjennomgangen gikk glipp av.

Les mer

- Hardening, in the NIST glossary — definisjonen NIST bruker, med publikasjonen den er hentet fra.
- sshd_config(5) — hvert direktiv for SSH-serveren, og regelen om at den første verdien som hentes for et nøkkelord, er den som brukes.
- systemd.exec(5) — direktivene for å sette tjenester i sandkasse: filtre for systemkall, adressefamilier og minne som er enten skrivbart eller kjørbart.
- systemd-analyze(1) — eksponeringspoengsummen fra `security` og `verify`, som fanger direktiver ingen kjenner igjen.
- CIS Benchmarks — konfigurasjonsgrunnlinjer laget ved konsensus, for operativsystemer, skyplattformer og serverprogramvare.

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 herding?» 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/hardening/