Engineer

CVE

den offentlige listen som gir hver kjent sårbarhet én ID

Hva det er

CVE, forkortelse for Common Vulnerabilities and Exposures, er den offentlige katalogen over offentlig kjente sårbarheter innen IT-sikkerhet, med én CVE-post for hver. CVE-programmet beskriver oppdraget sitt som å identifisere, definere og katalogisere disse sårbarhetene, slik at to personer eller to verktøy kan vise til en feil og vite at de snakker om den samme. Ordet brukes om programmet, om listen og løst om en enkelt post, som når noen sier at et bibliotek har to åpne CVE-er.

Hver post har en CVE-ID. Formatet er prefikset `CVE`, et firesifret årstall og et løpenummer på fire eller flere sifre, slik at `CVE-1999-0067`, `CVE-2019-12345` og `CVE-2021-7654321` alle er gyldige; programmet setter ingen øvre grense for hvor mange sifre løpenummeret kan ha. Årstallet er året ID-en ble reservert, eller året sårbarheten ble offentliggjort. Programmet understreker selv at årstallet ikke sier noe om når sårbarheten ble oppdaget.

ID-ene deles ikke ut fra ett sentralt kontor. CVE Numbering Authorities, kalt CNA-er, er leverandører, forskere, åpen kildekode-prosjekter, beredskapsteam for IT-sikkerhet, operatører av nettjenester, tilbydere av dusørprogrammer for sårbarheter og konsortier som programmet har gitt lov til å tildele ID-er og publisere poster, hver innenfor sitt eget område. Hver eneste CVE-post er lagt til av en CNA, det følger verken gebyr eller kontrakt med å bli en, og siden 2016 har programmet aktivt jobbet for å få med flere.

En post går gjennom tre tilstander. Den starter som *Reserved* når en CNA har tatt en ID som de involverte bruker til å koordinere det tidlige arbeidet med sårbarheten før den offentliggjøres. Den blir *Published* når CNA-en har fylt inn

minst ID-en, en kort beskrivelse, de berørte produktene og versjonene og en offentlig referanse. Skal ID-en ikke lenger brukes, blir den *Rejected*, og posten blir stående på listen slik at alle som har ID-en, kan se at den er ugyldig.

Ideen stammer fra 1999. To ingeniører fra MITRE, David E. Mann og Steven M. Christey, la fram konseptet i et notat (*white paper*) på en workshop ved Purdue University i januar 1999, og listen ble offentlig i september samme år med 321 poster.

Bruk begrepet i en beslutning

Bruk «Hva er CVE?» 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

En sårbarhetsrapport er bare nyttig hvis leseren kan avgjøre om den handler om det samme problemet som en annen rapport. Det spørsmålet avgjør en CVE-ID: to sikkerhetsvarsler, et funn fra en skanner og en sak i et saksystem som bærer samme ID, handler om den samme feilen. Mange store leverandører av operativsystemer skriver CVE-ID-er i sikkerhetsvarslene sine, slik at alle har ID-en fra det øyeblikket et problem blir kjent, og sikkerhetsprodukter og -tjenester over hele verden bygger CVE-poster inn i det de rapporterer.

Andre tjenester bygger på listen i stedet for ved siden av den. Den amerikanske National Vulnerability Database, som NIST lanserte i 2005, er et eget program som bygger videre på CVE-postene og legger sin egen analyse til hver av dem. CISAs katalog over Known Exploited Vulnerabilities (KEV) samler feil som angripere er sett å utnytte, og hver oppføring i den er identifisert ved og lenker til en CVE-post.

Det viktigste å vite om CVE er hva det ikke er. Det er ikke en score for alvorlighetsgrad: det er Common Vulnerability Scoring System, CVSS, en egen standard som forvaltes av FIRST, en amerikansk ideell organisasjon som har som formål å hjelpe beredskapsteam for IT-sikkerhet over hele verden. CVSS versjon 4.0 vurderer en sårbarhet fra 0 til 10 ut fra fire grupper av vurderingskriterier,

Base, *Threat*, *Environmental* og *Supplemental*, og oversetter tallet til *None*, *Low*, *Medium*, *High* og *Critical*, der Critical går fra 9,0 til 10,0. CVE Record Format lar en CNA legge en CVSS-score inn i posten sin, men scoren hører til CVSS og ikke til ID-en: den samme feilen kan ha ulike scorer fra ulike utgivere, eller ingen.

Det er heller ikke en klasse av svakheter. Common Weakness Enumeration, CWE, er en liste over svakheter utviklet av fellesskapet: forhold i programvare, fastvare, maskinvare eller en tjeneste som under visse omstendigheter kan åpne for en sårbarhet. CVE-programmet beskriver CWE-er som rotårsakene bak CVE-poster. En CVE navngir én feil i ett produkt; en CWE navngir typen feil som ligger bak. SQL-injeksjon er for eksempel svakheten CWE-89, og hvert produkt som viser seg å ha gjort akkurat den feilen, får sin egen CVE-ID.

Slik ser det ut i praksis

Shellshock, feilen i Bash fra 2014, viser alle tre systemene side om side. cve.org viser postene sine i CVE Record Format, et JSON-format, og toppen av denne posten ser slik ut (felter utelatt og flyttet rundt på):


```
{
  "dataType": "CVE_RECORD",
  "dataVersion": "5.1",
  "cveMetadata": {
    "cveId": "CVE-2014-6271",
    "assignerShortName": "debian",
    "state": "PUBLISHED",
    "dateReserved": "2014-09-09T00:00:00.000Z",
    "datePublished": "2014-09-24T18:00:00.000Z",
    "dateUpdated": "2025-10-22T00:05:36.342Z"
  }
}
```


Debian reserverte som CNA ID-en den 9. september 2014 og publiserte posten femten dager senere. Beskrivelsen sier at Bash til og med versjon 4.3 behandler tekst som står etter en funksjonsdefinisjon i verdien til en miljøvariabel, og at en angriper utenfra derfor kan kjøre vilkårlig kode gjennom et tilrettelagt miljø. Posten noterer også at den første rettelsen var feil, og at den delen som fortsatt sto åpen etterpå, fikk sin egen ID, CVE-2014-7169. Den som bare oppdaterte én gang og stoppet der, var fortsatt sårbar.

Posten inneholder deretter containere fra tre ulike kilder. CNA-containeren har Debians beskrivelse og referanser. CVE Program Container legger til egne referanser. Den tredje kommer fra en Authorized Data Publisher, har tittelen CISA ADP Vulnrichment og legger til det CVE-ID-en alene ikke forteller:

```
{
  "cvssV3_1": {
    "version": "3.1",
    "vectorString": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H",
    "baseScore": 9.8,
    "baseSeverity": "CRITICAL"
  }
}
```

Vektorstrengen viser resonnementet bak 9,8: et angrep over nettverket, med lav kompleksitet, uten krav om rettigheter og uten at en bruker trenger å gjøre noe, med stor innvirkning på konfidensialitet, integritet og tilgjengelighet. Den samme containeren knytter feilen til CWE-78, svakhetsklassen for injeksjon av kommandoer til operativsystemet, og noterer at CISA la den til i KEV-katalogen den 28. januar 2022. Én post rommer altså en ID fra CVE-programmet, en score fra en CVSS-versjon som FIRST utgir, og en svakhetsklasse fra CWE, hver fra sin egen kilde og hver som svar på sitt eget spørsmål.

I det daglige betyr CVE-ID-er mindre som lesestoff enn som nøkler. Sikkerhetssiden vår beskriver rutinen: hva hvert system er satt sammen av, blir notert, og listen sammenlignes med publiserte sikkerhetsvarsler, og ingen går ut fra at den er trygg bare fordi den har fungert hittil. CVE-ID-en er det som gjør at et funn fra den sjekken kan spores tilbake til sikkerhetsvarselet og fram til oppdateringen som tetter hullet.

Hvor det går galt

Å lese scoren som risikoen. Når ingen oppgir verdier for Threat og Environmental, fyller CVSS 4.0-spesifikasjonen dem med standardverdier som antar det verste, og den påpeker at NVD og produktvedlikeholdere som regel bare publiserer Base-scoren. En 9,8 på en komponent som ikke kan nås utenfra, er ikke en 9,8 for deg. Merk scorene slik spesifikasjonen ber om, CVSS-B for bare Base og CVSS-BTE når verdier for Threat og Environmental er brukt, og legg inn Environmental-verdiene for dine egne systemer før du prioriterer arbeidet.

Å blande sammen CVE, CVSS og CWE. «En CVE på 9,8» blander en ID med en score, og «CVE-en for SQL-injeksjon» blander en feil med en klasse av feil. Hold de tre adskilt i saker og rapporter: ID-en fra CVE, scoren med versjonen sin fra CVSS og rotårsaken fra CWE.

Å oppgi et tall uten vektor. FIRST krever at alle som publiserer CVSS-data, oppgir både scoren og vektorstrengen, slik at andre kan se hvordan scoren ble til. Et nakent tall kan ikke etterprøves, sammenlignes på tvers av CVSS-versjoner eller tilpasses dine forhold. Ta vare på hele vektoren, inkludert prefikset `CVSS:3.1/` eller `CVSS:4.0/` som oppgir versjonen.

Å lese årstallet som oppdagelsesdatoen. Årstallet i ID-en er året ID-en ble reservert eller feilen ble offentlig. En ID fra 2014 kan beskrive kode som ble utgitt på 1990-tallet, og en feil som blir funnet i desember, kan få neste års nummer. Bruk postens felter `dateReserved` og `datePublished` til datoer, aldri ID-en.

Å regne med fire sifre. Løpenummeret har fire eller flere sifre uten øvre grense, så et mønster som `CVE-\d{4}-\d{4}` kutter lengre ID-er uten forvarsel. Match `^CVE-[0-9]{4}-[0-9]{4,}$` i stedet, og behandle ID-en som en tekststreng og ikke som et tall, slik at innledende nuller blir bevart.

Å behandle en avvist post som borte. En ID i tilstanden Rejected blir med vilje stående på listen for å fortelle brukerne at den er ugyldig. Verktøy som overser den eller teller den som et åpent funn, villeder i hver sin retning. Les feltet `state`, og håndter `REJECTED` eksplisitt.

Å regne posten som ferdig. Shellshock-posten ble sist endret i oktober 2025, elleve år etter at den ble publisert, og KEV-opplysningen kom til i januar 2022. CVSS-spesifikasjonen beskriver Threat-gruppen sin som de egenskapene ved en sårbarhet som endrer seg over tid. Les postene bak funn du har akseptert i stedet for å rette, på nytt med faste mellomrom.

Å stoppe ved den første rettelsen. Shellshock fikk en ekstra ID fordi den første rettelsen lot en del av feilen stå åpen. Gå gjennom beskrivelsen og referansene i hver post etter oppfølgende ID-er, og lukk et funn først når alle ID-er som nevner komponenten, er løst.

Å ta fravær av en CVE som fravær av en sårbarhet. CVE dekker sårbarheter som er offentliggjort og har fått en ID av en CNA innenfor dens område. En feil i din egen kode har ingen CVE før noen rapporterer og offentliggjør den. Å skanne avhengigheter mot listen dekker det andre har funnet; din egen kode trenger sin egen statiske analyse, gjennomgang og testing.

Les mer

- CVE Program: Overview — programmets oppdrag, CVE-posten og partnerne som tildeler og publiserer postene.
- CVE Program: Process — ID-formatet, minimumsinnholdet i en post og postens tre tilstander.
- CVE Program: Related Efforts — hvordan NVD, CVSS, KEV-katalogen og CWE henger sammen med CVE, og hvor de skiller seg fra hverandre.
- Common Vulnerability Scoring System v4.0: Specification Document — FIRSTs grupper av vurderingskriterier, betegnelsene for scorer og den kvalitative skalaen for alvorlighetsgrad.
- CVE Record Format — JSON-definisjonen av hvordan en post er bygd opp, i den versjonen CVE Services bruker.
- CVE Program: FAQs (archive)

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 CVE?» 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/cve/