Engineer

CVE

den offentliga listan som ger varje känd sårbarhet ett id

Vad det är

CVE, förkortning för Common Vulnerabilities and Exposures, är den offentliga katalogen över offentligt kända sårbarheter inom it-säkerhet, med en CVE-post för var och en. CVE-programmet beskriver sitt uppdrag som att identifiera, definiera och katalogisera dessa sårbarheter, så att två personer eller två verktyg kan hänvisa till ett fel och veta att de menar samma. Ordet används om programmet, om listan och lite slarvigt om en enskild post, som när någon säger att ett bibliotek har två öppna CVE:er.

Varje post har ett CVE-id. Formatet är prefixet `CVE`, ett fyrsiffrigt årtal och ett löpnummer på fyra eller fler siffror, så `CVE-1999-0067`, `CVE-2019-12345` och `CVE-2021-7654321` är alla giltiga; programmet sätter ingen övre gräns för hur många siffror löpnumret får ha. Årtalet är det år då id:t reserverades eller det år då sårbarheten offentliggjordes. Programmet betonar själv att årtalet inte säger när sårbarheten upptäcktes.

Id:na delas inte ut från ett centralt kontor. CVE Numbering Authorities, kallade CNA:er, är leverantörer, forskare, projekt med öppen källkod, beredskapsteam för it-säkerhet, operatörer av onlinetjänster, organisatörer av belöningsprogram för sårbarheter och konsortier som programmet har gett rätt att tilldela id:n och publicera poster, var och en inom sitt eget område. Varje CVE-post har lagts till av en CNA, det kostar ingen avgift och kräver inget avtal att bli en, och sedan 2016 har programmet aktivt arbetat för att få med fler.

En post går igenom tre tillstånd. Den börjar som Reserved när en CNA har tagit ett id som de inblandade använder för att samordna det tidiga arbetet med sårbarheten innan den offentliggörs. Den blir Published när CNA:n har fyllt i

minst id:t, en kort beskrivning, de berörda produkterna och versionerna och en offentlig referens. Om id:t inte längre ska användas blir det Rejected, och posten står kvar på listan så att alla som har id:t kan se att det är ogiltigt.

Idén kommer från 1999. Två ingenjörer på MITRE, David E. Mann och Steven M. Christey, presenterade konceptet i en vitbok på en workshop vid Purdue University i januari 1999, och listan blev offentlig i september samma år med 321 poster.

Använd begreppet i ett beslut

Använd ”Vad är CVE?” 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

En sårbarhetsrapport är bara användbar om läsaren kan avgöra om den handlar om samma problem som en annan rapport. Den frågan avgör ett CVE-id: två säkerhetsbulletiner, ett fynd från en skanner och ett ärende i ett ärendesystem som bär samma id handlar om samma fel. Många stora leverantörer av operativsystem skriver CVE-id:n i sina säkerhetsbulletiner, så att alla har id:t från det ögonblick ett problem blir känt, och säkerhetsprodukter och säkerhetstjänster över hela världen bygger in CVE-poster i det de rapporterar.

Andra tjänster bygger på listan i stället för bredvid den. Den amerikanska National Vulnerability Database, som NIST lanserade 2005, är ett separat program som bygger vidare på CVE-posterna och lägger till sin egen analys för var och en. CISA:s katalog Known Exploited Vulnerabilities (KEV) samlar fel som angripare har setts utnyttja, och varje post i den identifieras av och länkar till en CVE-post.

Det viktigaste att veta om CVE är vad det inte är. Det är inte ett mått på allvarlighetsgrad: det är Common Vulnerability Scoring System, CVSS, en separat standard som förvaltas av FIRST, en amerikansk ideell organisation som har till uppgift att hjälpa beredskapsteam för it-säkerhet över hela världen. CVSS version 4.0 bedömer en sårbarhet från 0 till 10 utifrån fyra grupper av mätvärden, Base,

Threat, Environmental och Supplemental, och översätter talet till None, Low, Medium, High och Critical, där Critical går från 9,0 till 10,0. CVE Record Format låter en CNA lägga in en CVSS-poäng i sin post, men poängen hör till CVSS och inte till id:t: samma fel kan ha olika poäng från olika utgivare, eller ingen alls.

Det är inte heller en klass av svagheter. Common Weakness Enumeration, CWE, är en lista över svagheter som utvecklas av gemenskapen: förhållanden i programvara, inbyggd programvara, hårdvara eller en tjänst som under vissa omständigheter kan öppna för en sårbarhet. CVE-programmet beskriver CWE:er som grundorsakerna bakom CVE-poster. En CVE namnger ett fel i en produkt; en CWE namnger den typ av fel som ligger bakom. SQL-injektion är till exempel svagheten CWE-89, och varje produkt som visar sig ha gjort just det felet får ett eget CVE-id.

Så ser det ut i praktiken

Shellshock, felet i Bash från 2014, visar alla tre systemen sida vid sida. cve.org visar sina poster i CVE Record Format, ett JSON-format, och början av den här posten ser ut så här (fält utelämnade och omflyttade):


```
{
  "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 reserverade som CNA id:t den 9 september 2014 och publicerade posten femton dagar senare. Beskrivningen säger att Bash till och med version 4.3 behandlar text som står efter en funktionsdefinition i värdet av en miljövariabel, och att en angripare utifrån därför kan köra godtycklig kod genom en särskilt utformad miljö. Posten noterar också att den första korrigeringen var felaktig och att den del som fortfarande stod öppen efteråt fick ett eget id, CVE-2014-7169. Den som bara uppdaterade en gång och sedan slutade var fortfarande sårbar.

Posten innehåller sedan containrar från tre olika källor. CNA-containern innehåller Debians beskrivning och referenser. CVE Program Container lägger till egna referenser. Den tredje kommer från en Authorized Data Publisher, har titeln CISA ADP Vulnrichment och lägger till det som CVE-id:t ensamt inte säger:

```
{
  "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"
  }
}
```

Vektorsträngen visar resonemanget bakom 9,8: en attack över nätverket, med låg komplexitet, utan krav på behörighet och utan att en användare behöver göra något, med stor påverkan på konfidentialitet, integritet och tillgänglighet. Samma container kopplar felet till CWE-78, svaghetsklassen för injektion av kommandon till operativsystemet, och noterar att CISA lade till det i KEV-katalogen den 28 januari 2022. En enda post rymmer alltså ett id från CVE-programmet, en poäng från en CVSS-version som FIRST ger ut och en svaghetsklass från CWE, var och en från sin egen källa och var och en som svar på sin egen fråga.

I vardagen betyder CVE-id:n mindre som läsning än som nycklar. Vår säkerhetssida beskriver rutinen: vad varje system är uppbyggt av antecknas, och listan jämförs med publicerade säkerhetsbulletiner; ingen utgår från att den håller bara för att den har fungerat hittills. CVE-id:t är det som gör att ett fynd från den kontrollen kan spåras tillbaka till bulletinen och fram till uppdateringen som täpper till hålet.

Var det går fel

Att läsa poängen som risken. När ingen anger värden för Threat och Environmental fyller CVSS 4.0-specifikationen i dem med standardvärden som antar det värsta, och den påpekar att NVD och produktförvaltare oftast bara publicerar Base-poängen. En 9,8 på en komponent som inte går att nå utifrån är inte en 9,8 för dig. Märk poängen så som specifikationen begär, CVSS-B för enbart Base och CVSS-BTE när värden för Threat och Environmental har använts, och lägg till Environmental-värdena för dina egna system innan du prioriterar arbetet.

Att blanda ihop CVE, CVSS och CWE. ”En CVE på 9,8” blandar ett id med en poäng, och ”CVE:n för SQL-injektion” blandar ett fel med en klass av fel. Håll isär de tre i ärenden och rapporter: id:t från CVE, poängen med sin version från CVSS och grundorsaken från CWE.

Att ange ett tal utan vektor. FIRST kräver att alla som publicerar CVSS-data anger både poängen och vektorsträngen, så att andra kan se hur poängen togs fram. Ett naket tal går inte att kontrollera, jämföra mellan CVSS-versioner eller anpassa till dina förhållanden. Spara hela vektorn, inklusive prefixet `CVSS:3.1/` eller `CVSS:4.0/` som anger versionen.

Att läsa årtalet som upptäcktsdatum. Årtalet i id:t är det år då id:t reserverades eller felet blev offentligt. Ett id från 2014 kan beskriva kod som släpptes på 1990-talet, och ett fel som hittas i december kan få nästa års nummer. Använd postens fält `dateReserved` och `datePublished` för datum, aldrig id:t.

Att utgå från fyra siffror. Löpnumret har fyra eller fler siffror utan övre gräns, så ett mönster som `CVE-\d{4}-\d{4}` kapar längre id:n utan förvarning. Matcha `^CVE-[0-9]{4}-[0-9]{4,}$` i stället, och behandla id:t som en sträng och inte som ett tal, så att inledande nollor bevaras.

Att behandla en avvisad post som försvunnen. Ett id i tillståndet Rejected står avsiktligt kvar på listan för att tala om för användarna att det är ogiltigt. Verktyg som ignorerar det eller räknar det som ett öppet fynd vilseleder åt var sitt håll. Läs fältet `state` och hantera `REJECTED` uttryckligen.

Att se posten som färdig. Shellshock-posten ändrades senast i oktober 2025, elva år efter att den publicerades, och KEV-uppgiften tillkom i januari 2022. CVSS-specifikationen beskriver sin Threat-grupp som de egenskaper hos en sårbarhet som förändras över tid. Läs regelbundet om posterna bakom fynd som du har accepterat i stället för att åtgärda.

Att stanna vid den första korrigeringen. Shellshock fick ett extra id eftersom den första korrigeringen lämnade en del av felet öppen. Gå igenom varje posts beskrivning och referenser efter uppföljande id:n, och stäng ett fynd först när alla id:n som nämner komponenten är åtgärdade.

Att ta frånvaro av en CVE som frånvaro av en sårbarhet. CVE täcker sårbarheter som har offentliggjorts och fått ett id av en CNA inom dess område. Ett fel i din egen kod har ingen CVE förrän någon rapporterar och offentliggör det. Att skanna beroenden mot listan täcker det andra har hittat; din egen kod behöver sin egen statiska analys, granskning och testning.

Läs mer

- CVE Program: Overview — programmets uppdrag, CVE-posten och partnerna som tilldelar och publicerar posterna.
- CVE Program: Process — id-formatet, minimiinnehållet i en post och postens tre tillstånd.
- CVE Program: Related Efforts — hur NVD, CVSS, KEV-katalogen och CWE hänger ihop med CVE, och var de skiljer sig åt.
- Common Vulnerability Scoring System v4.0: Specification Document — FIRST:s grupper av mätvärden, beteckningarna för poäng och den kvalitativa skalan för allvarlighetsgrad.
- CVE Record Format — JSON-definitionen av hur en post är uppbyggd, i den version som CVE Services använder.
- CVE Program: FAQs (archive)

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 CVE?” 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/cve/