Engineer

CVE

the public list giving each disclosed vulnerability one ID

What it is

CVE, short for Common Vulnerabilities and Exposures, is the public catalogue of publicly disclosed cybersecurity vulnerabilities, with one CVE Record for each. The CVE Program describes its mission as identifying, defining and cataloguing those vulnerabilities, so that two people or two tools can refer to a flaw and know they mean the same one. The word is used for the programme, for the list, and loosely for a single entry, as in “that library has two open CVEs”.

Every record carries a CVE ID. The format is the prefix `CVE`, a four-digit year and a sequence number of four or more digits, so `CVE-1999-0067`, `CVE-2019-12345` and `CVE-2021-7654321` are all valid; the programme sets no upper limit on the length of the sequence number. The year is the year the ID was reserved or the year the vulnerability was made public. In the programme’s own words, “the year portion is not used to indicate when the vulnerability was discovered”.

IDs are not handed out by one central desk. CVE Numbering Authorities, or CNAs, are vendors, researchers, open-source projects, emergency response teams, online service operators, bug bounty providers and consortia that the programme has authorised to assign IDs and publish records, each within its own scope. Every CVE Record is added by a CNA, there is no fee or contract to sign to become one, and since 2016 the programme has been actively adding more of them.

A record moves through three states. It starts as Reserved, when a CNA has taken an ID that the parties involved use to coordinate early work on the vulnerability ahead of disclosure. It becomes Published once the CNA has filled in at least the ID, a short description, the affected products and versions, and a public reference. If the ID should no longer be used, it becomes Rejected, and the record stays on the list so that anyone holding the ID can see that it is void.

The idea dates from 1999. Two MITRE engineers, David E. Mann and Steven M. Christey, presented the concept as a white paper at a workshop at Purdue University in January 1999, and the list went public in September 1999 with 321 records.

Use the concept in a decision

Put “What is CVE?” to work in a real decision. We can help relate the concept to the system you are building or buying. Accounts, passwords and permissions arranged so people hold exactly the access their job needs, and leavers lose it that day.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Wrote 1,000 RBAC rules to secure Linux servers — 1,000 access rules written to close a security gap

Wrote 1,000 RBAC rules to secure Linux servers Get in touch →

Why it matters

A vulnerability report is only useful if the people who read it can tell whether it is the same problem as the one in another report. A CVE ID settles that: two advisories, a scanner finding and a ticket that carry the same ID are about the same flaw. Numerous major operating system vendors put CVE IDs into their advisories so that everyone has the ID from the moment a problem is announced, and security products and services around the world build CVE Records into what they report.

Other services are built on the list rather than beside it. The US National Vulnerability Database, launched by NIST in 2005, is a separate programme that builds on CVE Records and adds its own analysis to each one. CISA’s Known Exploited Vulnerabilities (KEV) catalogue lists flaws that attackers have been seen to use. Each entry names its CVE ID and links to the corresponding record.

The most useful thing to know about CVE is what it is not. Severity is measured separately by CVSS, the Common Vulnerability Scoring System, a standard run by FIRST, a US-based non-profit whose mission is to help computer security response teams around the world. CVSS version 4.0 rates a vulnerability from 0 to 10 using the Base, Threat, Environmental and Supplemental groups of metrics, and maps the number onto None, Low, Medium, High and Critical, with Critical running from 9.0 to 10.0. The CVE Record Format lets a CNA include a CVSS score in its record, but the score belongs to CVSS, not to the ID: one flaw can carry different scores from different publishers, or none.

Nor is it a weakness class. The Common Weakness Enumeration, CWE, is a community-developed list of weaknesses: conditions in software, firmware, hardware or a service that can, in the right circumstances, let a vulnerability in. The CVE Program describes CWEs as the root causes of CVE Records. A CVE names one flaw in one product; a CWE names the type of error behind it. SQL injection, for instance, is the weakness CWE-89, and every product found to have made that mistake gets a CVE ID of its own.

What it looks like in practice

Shellshock, the 2014 flaw in Bash, shows all three systems side by side. cve.org displays its records in the CVE Record Format, a JSON format, and the top of this one reads as follows (fields trimmed and reordered):


```
{
  "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, acting as CNA, reserved the ID on 9 September 2014 and published the record fifteen days later. Its description says that Bash up to version 4.3 processes text that trails a function definition in the value of an environment variable, which lets a remote attacker run arbitrary code through a crafted environment. It also records that the first fix was incorrect and that the part still open after it received an ID of its own, CVE-2014-7169. Teams that patched once and stopped were still exposed.

The record then holds containers from three different sources. The CNA container holds Debian’s description and references. The CVE Program Container adds references of its own. A third, from an Authorized Data Publisher and titled CISA ADP Vulnrichment, adds the material that the CVE ID alone does not carry:

```
{
  "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"
  }
}
```

The vector string spells out the reasoning behind 9.8: an attack over the network, of low complexity, needing no privileges and no user interaction, with high impact on confidentiality, integrity and availability. The same container maps the flaw to CWE-78, the weakness class for operating-system command injection, and records that on 28 January 2022 CISA added it to the KEV catalogue. One record therefore holds an ID from the CVE Program, a score from a CVSS version that FIRST publishes, and a weakness class from CWE, each from a different source and each answering a different question.

Day to day, CVE IDs matter less as reading matter than as keys. Our security page describes the routine: the parts each system is built from are listed, and the list is compared with published advisories instead of being trusted because it worked before. The CVE ID is what lets a finding from that check be traced back to the advisory, and forward to the upgrade that closes it.

Where it goes wrong

Reading the score as the risk. When nobody supplies Threat or Environmental values, the CVSS 4.0 specification fills them with defaults that assume the worst, and it notes that NVD and product maintainers usually publish only the Base score. A 9.8 on a component that is unreachable from outside is not a 9.8 for you. Label scores the way the specification asks, CVSS-B for Base only and CVSS-BTE when Threat and Environmental values were applied, and add the Environmental values for your own systems before ranking the work.

Mixing up CVE, CVSS and CWE. “A CVE of 9.8” mixes an identifier with a score, and “the SQL injection CVE” mixes a flaw with a class of flaws. Keep the three apart in tickets and reports: the ID from CVE, the score with its version from CVSS, and the root cause from CWE.

Quoting a number without its vector. FIRST requires anyone who publishes CVSS data to publish its vector alongside the number, making the underlying choices available for inspection. A bare number cannot be checked, compared across CVSS versions or adjusted for your setting. Store the full vector, including the `CVSS:3.1/` or `CVSS:4.0/` prefix that names the version.

Reading the year as the discovery date. The year in the ID is when the ID was reserved or the flaw went public. A 2014 ID can describe code shipped in the 1990s, and a flaw found in December may carry the next year's number. Use the record's `dateReserved` and `datePublished` fields for dates, never the ID.

Assuming four digits. The sequence number has four or more digits with no upper bound, so a pattern such as `CVE-\d{4}-\d{4}` silently cuts longer IDs short. Match `^CVE-[0-9]{4}-[0-9]{4,}$` instead, and treat the ID as a string, not a number, so that leading zeros survive.

Treating a Rejected record as missing. A Rejected ID stays on the list on purpose, to tell users it is void. Tooling that ignores it, or that counts it as an open finding, misleads in either direction. Read the `state` field and handle `REJECTED` explicitly.

Treating the record as finished. The Shellshock record was last changed in October 2025, eleven years after it was published, and its KEV entry arrived in January 2022. The CVSS specification uses its Threat group to account for changes in a vulnerability's circumstances over time. Re-read the records behind any finding you have accepted rather than fixed, and do so on a schedule.

Stopping at the first fix. Shellshock needed a second ID because the first patch left part of the flaw open. Read each record's description and references for follow-up IDs, and close a finding only when every ID that names the component is resolved.

Taking no CVE to mean no vulnerability. CVE covers vulnerabilities that have been publicly disclosed and given an ID by a CNA within its scope. A flaw in your own code has no CVE until someone reports and publishes it. Scanning dependencies against the list covers what others have found; your own code needs its own static analysis, review and testing.

Further reading

- CVE Program: Overview — the mission, the CVE Record, and the partners who assign and publish them.
- CVE Program: Process — the ID format, the minimum content of a record, and its three states.
- CVE Program: Related Efforts — how NVD, CVSS, the KEV catalogue and CWE relate to CVE, and where they differ.
- Common Vulnerability Scoring System v4.0: Specification Document — FIRST's metric groups, score labels and qualitative severity scale.
- CVE Record Format — the JSON definition of how a record is structured, in the version that CVE Services uses.
- CVE Program: FAQs (archive)

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is CVE?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/cve/