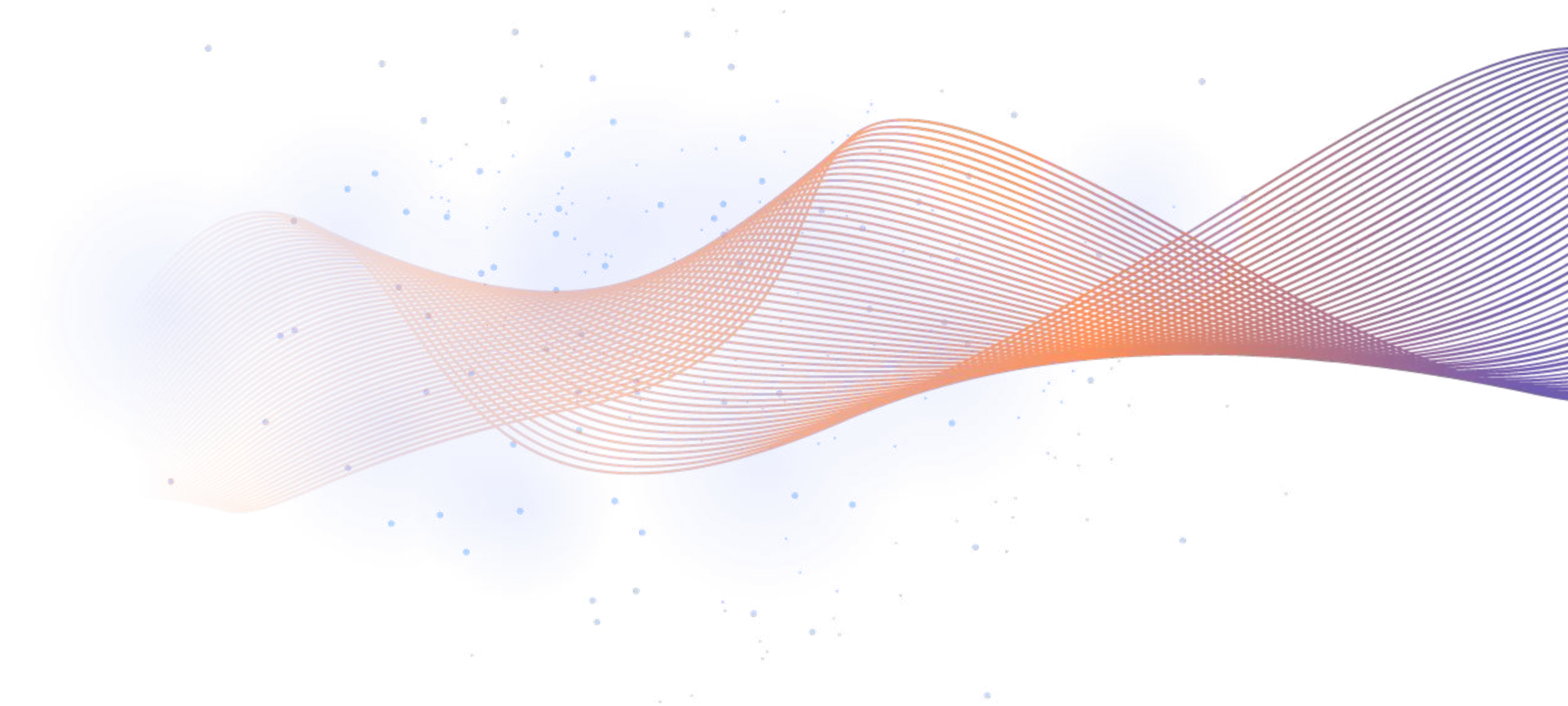
tetrate

CVE®

Common Vulnerabilities and Exposures Explained



What is Common Vulnerabilities and Exposures (CVE) ?

Businesses today face an unprecedented level of risk from cybersecurity attacks and data breaches, often resulting in substantial financial damages. These threats stem from vulnerabilities and exposures within computer systems, making it imperative for organizations to understand and address these risks.

A CVE, or Common Vulnerabilities and Exposures, is a standardized identifier assigned to a known security vulnerability in software, including open source software. The CVE system is maintained by the MITRE Corporation and provides a way to uniquely identify and track vulnerabilities across different information security databases and tools.

A vulnerability can be described as a flaw or weakness in computer systems or software that unintentionally grants unauthorized access to users. Exploiting vulnerabilities allows attackers to execute destructive actions, such as installing malware or gaining unauthorized access to sensitive information.

On the other hand, an exposure refers to a misconfiguration that provides attackers access to a computer system or its stored data. For instance, a loosely secured cloud storage system may permit unauthorized access to sensitive data, or an open network port on a server can be exploited by command and control malware.

It's crucial to recognize exposures as vulnerabilities actively targeted and exploited by attackers.

Given the complex nature of the applications we use daily and their frequent development cycles, vulnerabilities and exposures are common. Vendors respond swiftly by developing and releasing patches as soon as vulnerabilities are identified. Recognizing the need for effective communication about newly discovered vulnerabilities, the industry employs Common Vulnerabilities and Exposures (CVEs). This international framework maintains an updated registry of all known computer security vulnerabilities and exposures.

Some reasons why software vendors choose to manage CVEs includes:

- Security Best Practices: Following security best practices is essential for building trust with users and maintaining a positive reputation. Actively addressing and disclosing security vulnerabilities demonstrates a commitment to security.
- Transparency and Accountability: Managing CVEs in a transparent manner helps vendors communicate openly with their user base. It fosters accountability and allows users to make informed decisions about their use of the software.
- Coordination with Security Community: CVE management facilitates coordination with the broader security community, including vulnerability researchers, other software vendors, and security organizations. This collaborative approach helps ensure that vulnerabilities are addressed promptly and effectively.

- Customer Trust: Users, especially in enterprise and critical infrastructure settings, often require assurance that the software they use is actively maintained and secure. Managing CVEs contributes to building and maintaining trust with customers.
- Regulatory Compliance: In certain industries and regions, regulatory frameworks may require software vendors to address and disclose security vulnerabilities promptly. Adhering to CVE management practices helps vendors meet these compliance requirements.
- Risk Mitigation: Identifying and addressing security vulnerabilities promptly helps mitigate the risk of exploitation by malicious actors. Proactive management of CVEs can prevent or minimize potential security incidents.

While there may not be a legal requirement specifically mandating CVE management, some industries or contracts, especially those dealing with sensitive information or critical infrastructure, may have specific security and disclosure requirements that effectively necessitate CVE management. Open source projects, being transparent and community-driven, often make use of CVE identifiers to track and communicate security issues, allowing users and developers to stay informed about potential risks and take appropriate actions to secure their systems.

An Overview of CVEs: Core Concepts

According to Deloitte's report, most companies spend about 10% of their annual IT budget on cybersecurity. This is a significant amount of money for medium to large organizations. One way to reduce this cost is to maintain an updated database of known vulnerabilities.

A Central Database Operated by the MITRE Corporation

CVE is operated by the MITRE Corporation and funded in part by the United States Department of Homeland Security. CVEs have become an indispensable source of information for cybersecurity professionals worldwide. Developers and organizations rely on CVEs to receive critical security updates, stay informed on breached systems and implement preventative measures to thwart malicious attacks.

Although MITRE manages a list of current CVEs, they don't actively search for new application vulnerabilities. That task falls on individuals or organizations in the open-source community who spend their time and effort discovering application flaws and reporting them to the vendors.

CVE Numbering Authorities (CNAs)

MITRE Corporation's other role in the CVE program is to manage the CVE Numbering Authorities (CNAs). CNAs are organizations throughout the world that are also CVE program partners. CNAs are usually part of major corporations - such as Microsoft, Oracle or Apple - and they're essentially a bridge between individuals who find a new vulnerability and the CVE community. They help the discovery process by checking and submitting documents about the vulnerability and publishing the CVE.

CNAs are also in charge of assigning unique IDs to new CVEs. The ID helps you find all the relevant information about a vulnerability or exposure. Different CVE databases worldwide use these IDs to add more detailed information about the CVE, including:

- Severity
- Affected software systems
- The steps to follow to patch the vulnerability and contain the damage

Created to Share and to Inform

CVEs were created to share information and technical details about these vulnerabilities with the wider community, informing everyone about the latest threat landscape and instructing them on how to defend against the ever-emerging risks.

Such vulnerabilities can come from unmaintained open-source software and commercial applications alike. Software developers often use third-party libraries and dependencies to develop applications.

Examples of CVEs

A classic example of a CVE is the recent Log4j vulnerability report (CVE-2021-44228). It contains detailed information about a vulnerability of the popular Java logging framework, Apache Log4j. Many service providers, like AWS, Cloudflare and Twitter, were affected by this vulnerability. Another example is the ProxyShell vulnerability. It affected Microsoft Exchange servers by allowing remote code execution. It was announced via CVE-2021-34473, CVE-2021-34523 and CVE-2021-31207.

> *"The greatest benefit of a centralized CVE reporting system is that it can alert everyone at the same time about vulnerable applications or libraries."*

Integrating CVE Scanning into CI/CD

One of the best use cases of CVEs is to include them in your application development and deployment pipeline. As mentioned before, applications may often use packages or libraries with vulnerabilities. By including the CVE scanning capability in the CI/CD pipeline, developers can automate security testing, stop the code from merging into the code repository and receive alerts.

What Qualifies as a CVE?

Before a CVE can be accepted and published, it must meet a specific set of criteria. Fulfilling the requirements helps separate and distinguish between bugs and vulnerabilities. As a CNA, you don't want to go through countless CVE reports only to find out most of them are bugs that developers can fix by changing a few lines of code. The following are some of the criteria for qualifying as a CVE.

- **Repairability:** The vulnerability should be repairable. This means that the end user should be able to resolve the problem by installing a software update that includes a patch for that vulnerability.
- **Provability:** The entity filing the report must be able to prove the alleged vulnerability through documentation or present the software vendor's confirmation.
- **Limited Effect:** The vulnerability should affect only a single piece of software or codebase. If the vulnerability affects multiple products, it should be split into different CVEs.

It's worth noting that even if a vulnerability fulfills all of these criteria, it may not be published immediately. Delaying publication can give the vendor time to develop a patch. This allows the vendor to announce the fix at the same time the CVE is made public.

Many companies also offer bug bounty programs to encourage the community to discover and report vulnerabilities. Bug bounty hunters are experienced cybersecurity professionals who perform extensive tests to identify threats and exposures in an application.

What Is the Common Vulnerability Scoring System (CVSS)?

CVEs can be of different types with varying levels of severity and needing varying degrees of attention. Since CVEs are being published all the time, the Common Vulnerability Scoring System (CVSS) helps determine how to prioritize CVEs.

CVSS is a numbering system for assigning priority and severity levels to CVEs. It works by assigning a number between 0.0 and 10.0 to a CVE, indicating its severity. A vulnerability with a CVSS score between 9.0 and 10.0 is considered critical and needs immediate action.

CVSS helps companies plan risk management and response strategies and prioritize their patching cycles. Many security advisories release lists of CVEs ordered by the CVSS scores, with more severe vulnerabilities at the top of the list.

How Does CVE Work?

When a vulnerability is discovered, it must go through a standardized CVE lifecycle before publication.

1. **Discover:** The process starts with a person or organization discovering a vulnerability.
2. **Report:** The person or entity that discovered the vulnerability files a report with a CVE program partner.
3. **Request:** The CVE partner (CNA) issues an ID for the vulnerability.
4. **Reserve:** The ID is reserved for that particular vulnerability and is used in the early-stage assessment of the CVE and all related communications between different parties.
5. **Submit:** The CVE partner assesses the vulnerability's submitted documents, which should include all information needed to prove the presence of the vulnerability or exposure, the root cause, the type of threat and the impact.
6. **Publish:** After all the documented details are verified, the CNA publishes the CVE, making it public.

Challenges of Staying Informed on the Latest CVEs

CVEs offer valuable information about the latest computer system vulnerabilities. However, there are too many to sort through every week, and not all CVEs may apply to your organization's landscape. These announcements are only effective if you can stay updated.

The United States Computer Emergency Readiness Team (US-CERT) is one of the organizations providing public security advisory services on CVEs. They send a weekly newsletter containing the latest CVEs ordered by severity. If you're trying to stay informed, another option is to subscribe to the RSS feed of the CERT/CC Vulnerability Notes Database to receive real-time notifications.

Vulnerability management platforms can increase the effectiveness of your organization's IT security initiatives. These platforms constantly assess, research, and report active vulnerabilities within your infrastructure and applications. The graphical dashboards help you get a bird's-eye view of the current threat posture, track vulnerabilities and receive step-by-step patching information.

While there is no universal legal requirement that mandates software vendors to manage CVEs (Common Vulnerabilities and Exposures), it is considered a best practice, and many responsible software vendors voluntarily adhere to it. The management of CVEs is crucial for ensuring the security of software products and the protection of users' systems and data.

Publicly Available Vulnerability Databases

- **Common Vulnerabilities and Exposures (CVE)** - CVE is a database operated by MITRE. It is a dictionary that provides publicly disclosed cybersecurity vulnerabilities and exposures. CVE entries comprise an identification number, a description, and at least one public reference. CVE list does not include any severity rating such as CVSS score.
- **National Vulnerability Database (NVD)** - NVD is a database maintained by the U.S. government. The National Institute of Standards and Technology (NIST) NVD team analyzes the new CVE in the CVE dictionary and assigns severity ratings such as Common Vulnerability Scoring System (CVSS) score as High/Medium/Low to the CVE.
- **Vulnerability Notes Database** - This database is operated by the CERT division of the Carnegie Mellon software engineering institute.
- **Exploit Database** - This database is operated by Offensive security.
- **Vulnerability Lab** - This is an open-source database.
- **VulDB** - this is also an open source vulnerability database with over 140k entries.

All these databases are used to share computer flaws detected in the open source software and/or dependencies.

About Tetrate

Rooted in open source, Tetrate was founded to solve the application networking and security challenges created by modern computing so enterprises can innovate with speed and safety in hybrid and multi-cloud environments. As applications evolve into collections of decentralized microservices, monitoring and managing the network communications and security among those myriad services becomes challenging. This is why some of the largest financial institutions, governments and other enterprises rely on Tetrate to deliver modern application networking and security on a foundation of Zero Trust.

Find out more at www.tetrate.io.

Tetrate Academy

If you are new to service mesh and Kubernetes security, we offer free online courses at Tetrate Academy that will quickly get you up and running with Istio and Envoy. Our courses are expertly curated, hands-on training experiences from the co-creators of open source Istio and Envoy. Private training for enterprise customers available upon request.

Learn more at academy.tetrate.io

Get Started with TIS

If you're looking for a fast way to get to production with Istio, check out Tetrate Istio Distribution (TID), Tetrate's hardened, fully upstream Istio distribution, with FIPS-verified builds and support available. It's a great way to get started with Istio knowing you have a trusted distribution to begin with, an expert team supporting you, and also have the option to get to FIPS compliance quickly if you need to.

Get Started with TSB

As you add more apps to the mesh, you'll need a unified way to manage those deployments and to coordinate the mandates of the different teams involved. That's where Tetrate Service Bridge comes in. Learn more about how Tetrate Service Bridge makes service mesh more secure, manageable, and resilient here, or contact us for a quick demo

Additional Resources

For more information about the service mesh and Tetrate's solutions, visit https://tetrate.io/resources/