Editors: Ramaswamy Chandramouli, mouli@nist.gov
Tim Grance, tim.grance@nist.gov
Rick Kuhn, kuhn@nist.gov
Susan Landau, susan.landau@sun.com

Common Vulnerability Scoring System

Peter Mell and Karen Scarfone
US National Institute of Standards and Technology

Sasha Romanosky
Carnegie Mellon University

Historically, vendors have used their own methods for scoring software vulnerabilities, usually without detailing their criteria or processes. This creates a major problem for users, particularly those who manage disparate IT systems and applications. For example, should they first address a vulnerability with a severity of "5" or one with a severity of "high"?

The Common Vulnerability Scoring System (CVSS) is a public initiative designed to address this issue by presenting a framework for assessing and quantifying the impact of software vulnerabilities. Organizations currently generating CVSS scores include Cisco, US National Institute of Standards and Technology (through the US National Vulnerability Database; NVD), Qualys, Oracle, and Tenable Network Security. CVSS offers the following benefits:

- *Standardized vulnerability scores*. CVSS is application-neutral, enabling an organization to score all of its IT vulnerabilities using the same scoring framework. This allows them to leverage a single vulnerability management policy to govern how quickly each vulnerability is validated and remediated.
- *Contextual scoring*. CVSS scores represent the actual risk a given vulnerability poses, helping them prioritize remediation efforts.
- *Open framework*. CVSS provides full details regarding the parameters used to generate each score. This helps organizations understand both the reasoning behind, and the differences among, vulnerabilities scores.

The goal is for CVSS to facilitate the generation of consistent scores that accurately represent the impact of vulnerabilities.

Introduction

CVSS was conceived by the US National Infrastructure Assurance Council (NIAC), a group of industry leaders that provides the US Department of Homeland Security with security recommendations for critical US infrastructures. Published in 2005 as a first-generation open scoring system,[1] CVSS is currently under the custodial care of the international Forum for Incident Response and Security Teams (FIRST; www.first.org/cvss/), which manages the official mailing list and documentation. Many individuals and organizations have formed a special interest group (SIG) to promote and refine the framework. A current list of adopters is available at www.first.org/cvss/eadopters.html.

Several vulnerability scoring systems exist. Examples include US-CERT (www.kb.cert.org/vuls/html/fieldhelp#metric), the SANS Institute's Critical Vulnerability Analysis Scale (www.sans.org/newsletters/cva/), and the Microsoft Security Response Center Severity Rating System (www.microsoft.com/technet/security/bulletin/rating.mspx). As with these scoring systems, CVSS has several restrictions. For example, it doesn't provide a method for aggregating individual scores across systems or departments. By itself, it's also unsuitable for managing IT risk because it doesn't consider mitigation strategies such as firewalls and access control. Further, CVSS is neither a score repository (Bugtraq), nor a vulnerability database (Open Source Vulnerability Database), nor a vulnerability classification system (Common Vulnerabilities and Exposures; CVE).

CVSS scores are composites derived from the following three categories of metrics:

- *Base*. This group represents the properties of a vulnerability that don't change over time, such as access complexity, access vector, degree to which the vulnerability compromises the confidentiality, integrity, and availability of the system, and requirement for authentication to the system.
- *Temporal*. This group measures the properties of a vulnerability that do change over time, such as the existence of an official patch or functional exploit code.
- *Environmental*. This group measures the properties of a vulnerability that are representative of users' IT environments, such as prevalence of affected systems and potential for loss.

We further describe these metric groups in the next section.

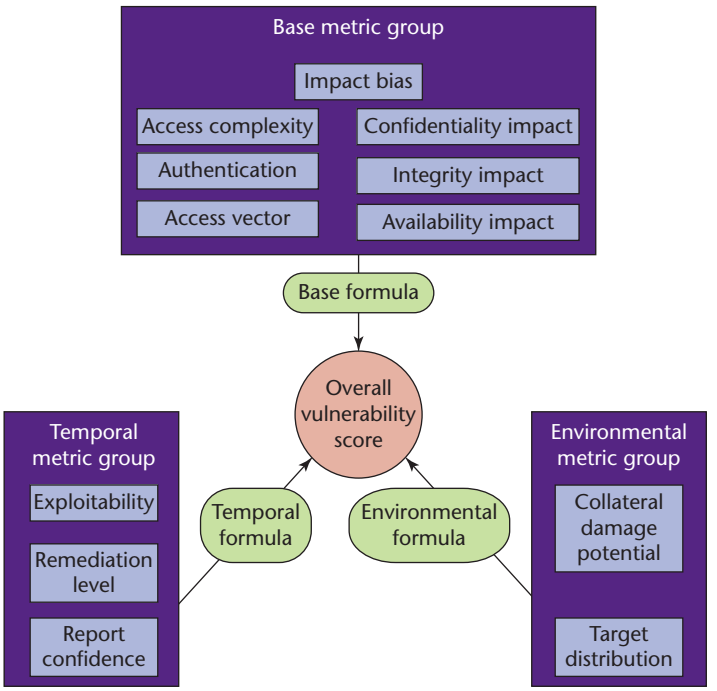

Figure 1. Common Vulnerability Scoring System (CVSS) metric groups. The overall score is determined by generating a base score and modifying it through the temporal and environmental formulas.

CVSS metrics

As Figure 1 illustrates, CVSS uses simple metrics and formulas to create composite scores, derived from base, temporal, and environmental groups. You can find complete details regarding the algorithms at www.first.org/cvss/cvss-guide.html.

The base metrics represent the vulnerability's immutable characteristics (properties that are constant over time and across systems). They produce a score within the range of 0.0 to 10.0:

- *Access vector*. Can an attacker exploit the vulnerability remotely or locally?
- *Authentication*. Must an attacker authenticate to the operating system, application, or service after gaining access to the target to exploit the vulnerability? Scoring options are `required` or `not-required`.
- *Access complexity*. How difficult is it to exploit the vulnerability (for example, does it require action by the user, such as clicking on a malicious URL?) Options are `high` or `low`.
- *Confidentiality impact*. What is the potential impact of unauthorized access to the system's data? Options are `none`, `partial`, or `complete`.
- *Integrity impact*. What is the potential impact of unauthorized modification or destruction of the system's files or data? Options are `none`, `partial`, or `complete`.
- *Availability impact*. What is the potential impact if the system or data is unavailable? Again, the scoring options are `none`, `partial`, or `complete`.
- *Impact bias*. To which property (if any) should the score convey a greater weighting: `confidentiality`, `integrity`, or `availability`? For example, confidentiality might be the most important for encryption software, and so the scoring analyst would assign a `confidentiality` bias.

The temporal metrics represent the vulnerability's properties that might change over time. They modify the base score, lowering it by as much as one-third:

- *Exploitability*. What is the current state (or ease) of the vulnerability's exploitability? Options are `unproven`, `proof-of-concept`, `functional`, or `high`.
- *Remediation level*. What level of mitigating controls currently exists for the vulnerability? Scoring options are `official-fix`, `temporary-fix`, `workaround`, or `unavailable`.
- *Report confidence*. How credible are the vulnerability details? Options are `unconfirmed`, `uncorroborated`, or `confirmed`.

The environmental metrics modify this result to generate a final score, ranging from 0.0 (no affected systems) to 10.0 (most systems affected with a high risk of catastrophic damage). This score is most important because it provides the context for vulnerabilities within the organization:

- *Collateral damage potential*. What is the degree of loss to information, systems, or people? Options are `none`, `low`, `medium`, or `high`.
- *Target distribution*. What percentage of systems could the vulnerability affect? Options are `none`, `low`, `medium`, or `high`.

Having three scores for each vulnerability is efficient and flexible because all organizations can use the same base score for a vulnerability, yet each organization can customize the score for its own environment.

Assessing CVSS score generation

In an effort to validate real-world CVSS scoring, NVD analysts calculated CVSS base scores for 1,291 vulnerabilities listed in the CVE vulnerability dictionary. Although 101 values are possible (0.0 to 10.0, in

increments of 0.1), the experimental data produced only 25 distinct scores. Indeed, 79 percent of the vulnerabilities had one of two base scores:

- 2.3 typically reflected a locally exploitable vulnerability with user-level impact (40.3 percent of the vulnerabilities).
- 7.0 typically reflected a remotely exploitable vulnerability with user-level impact (38.9 percent of the vulnerabilities).

The seven most commonly assigned scores covered 94 percent of the vulnerabilities. In addition, we analyzed the base equation itself and found that most of the scores were in the lower half of the range. This is largely because of the equation's multiplicative nature—it can generate only 68 of the 101 possible base score values, and only 23 of those values are in the upper half of the range.

Moreover, the score distribution is highly bimodal. We expected a much more uniform distribution than for 79 percent of the vulnerabilities to map to just two scores. A primary reason for the distribution is that multiple sets of metric inputs can generate the same scores. For example, a remotely exploitable vulnerability that provides user-level access would produce the same score as a locally exploitable vulnerability that provides complete control of the same target (assuming all other metrics are set to the same values). In most environments, however, the former should have a higher score because it necessitates a faster response than the latter.

Proposed changes to the base metrics (described later) might add greater diversity to scores by increasing the range of values for individual metrics. However, this won't eliminate the highly bimodal distribution. Another concern is that users might mistakenly believe that the scoring is more accurate than it truly is, based on the scoring range's granularity. The CVSS SIG is reviewing the metrics and equations and considering what granularity level would be most useful to users.

The CVSS SIG has also observed discrepancies among analysts scoring particular types of vulnerability. To address these issues, the SIG created a separate task force and identified several problem areas and solutions. To date, the SIG has addressed the following issues through clarifications to the documentation and proposed modifications to the metrics.

Base metric: Access vector

This metric didn't differentiate between remote and local network (subnet) vulnerabilities. For example, attacks launched from hosts on the target's subnet received the same access vector score as attacks launched from hosts on the Internet.

To capture the relative difficulty of exploitation, the SIG proposed creating three access vector values: remote, local-network accessible, and local.

Base metric: Authentication

This metric didn't differentiate between requiring one or multiple successful authentication steps. For example, a vulnerability that could be exploited only after authenticating to an operating system and the application running on it received the same score as a vulnerability that could be exploited after simply authenticating to the operating system.

To more accurately reflect the difference in authentication sophistication, the SIG proposed three authentication metric values: none, single, or multiple authentications required.

Documentation: Target privileges assumptions

Varying assumptions from scoring analysts resulted in scoring inconsistencies. For example, some analysts assumed that IT assets would be secured properly according to best practices, thus limiting the impact of exploitation; others assumed that targets would use weaker, default security configurations (end users running email clients or Web browsers with administrative-level privileges, for example), thus causing a greater impact.

To ensure greater scoring consistency, the SIG updated the documentation to specify that scores should reflect typical (most common) software deployments and configurations. If privileges aren't clear, the scorer should assume the default configuration.

Documentation: Scoring indirect vs. direct impact

Attackers exploit many cross-site scripting vulnerabilities by luring users to click on Web links containing malicious code. Some analysts scored such vulnerabilities based on the impact to the server because that's where the malicious data is located, whereas others scored them based on the impact to end users, which is generally more severe but also much harder to quantify. These conflicting approaches caused significant scoring discrepancies for vulnerabilities that differed in terms of their direct and indirect impact.

The SIG responded by updating the documentation to specify that vulnerabilities should be scored with respect to their impact to the vulnerable target or target service only. For

A vulnerability-scoring system that's accurate is wonderful; a scoring system that's accurate and usable is even better.

cross-site scripting vulnerabilities, this typically means a partial impact on integrity and no impact on either confidentiality or availability.

ity-scoring analysts in large enterprises. Environmental scoring thus represents a possible barrier to entry for CVSS.

CVSS is significant because it can eliminate duplicate efforts in IT vulnerability scoring and allow organizations to make more informed decisions when managing vulnerabilities.

Increasing adoption

The CVSS SIG's primary effort has been to create a scoring system that generates consistent and representative vulnerability scores—that is, scores that properly reflect the severity and impact of a given vulnerability to any user's computing environment. However, CVSS has numerous challenges.

User participation in environmental scoring

Most end users will access and use CVSS scores indirectly—through vulnerability-scanning tools, for example. Although such tools can determine base scores and possibly temporal scores, they probably won't be able to generate accurate environmental scores without user participation. A scanning tool with full visibility into a user's network could potentially deduce target distribution, but it couldn't infer a vulnerability's damage potential. A proper solution therefore needs to allow users to score environmental metrics (collateral damage potential and target distribution), although we recognize that they might often face difficulties in collecting this information. Disaster recovery and business continuity departments often prioritize assets according to their value to the organization. This might provide the necessary data, but political and procedural difficulties could make obtaining that information difficult for vulnerabil-

End-user adoption

CVSS must be compatible with an organization's vulnerability management processes. Rather than a technical challenge of metrics or scoring algorithms, the concern is that the CVSS framework must satisfy a business need and make a valuable contribution to vulnerability management practices. Altering existing processes, procedures, and technologies could require considerable effort. For CVSS to make business sense, its payoff must be greater to an organization than the integration costs.

Vendor adoption

For CVSS to succeed, software and service vendors must identify the business case (just as with end users). Demand will likely be the greatest driver: vendors will offer CVSS scores when customers request them. Vendors might also choose to support CVSS as a way to differentiate themselves from their competition. Authoritative sources such as NVD make CVSS scores freely available to the public. (NVD currently offers XML feeds of CVSS base scores for all vulnerabilities.) These sources are particularly useful for vendors that don't wish to develop proprietary scoring mechanisms or perform their own scoring.

Security vs. usability

A vulnerability-scoring system that's accurate is wonderful; a scoring system that's accurate and usable is even better. For CVSS to become successful, the scoring mechanism must be clear, quick, and understandable. Complex scoring metrics might increase score accuracy and resolution, but if this comes at the expense of usability, analysts might score incorrectly, or end users might avoid incorporating temporal and environmental factors.

Scoring for IT misconfigurations

CVSS was developed to generate scores for vulnerabilities in the context of software flaws. Yet, many organizations are also interested in scores for security misconfigurations, such as applications configured to permit blank passwords or operating systems that allow any user to modify critical system files. Such misconfigurations can be exploited, so it should be possible to use a similar or identical scoring system for both types of problems. The CVSS specification and its accompanying documentation don't currently address misconfigurations, but this opportunity certainly exists. CVSS would be even more valuable to users if it let them compare and prioritize both software flaws and misconfigurations on their systems.

CVSS is an emerging standard, but for it to succeed, we must ensure that it generates consistent scores, that they accurately represent the impact of the vulnerabilities assessed, and that users can easily provide temporal and environmental data. We can achieve these goals by refining the CVSS metrics and equations, maintaining clear documentation, and clearly defining what CVSS is and is not.

We believe CVSS is significant because it can eliminate duplicate efforts in IT vulnerability scoring and let organizations make more informed decisions when managing vulnerabilities. We've observed scoring discrepancies and responded by improving the documentation and proposing better metrics. We've

seen strong industry adoption and are encouraged by the increasing participation, yet we continue to improve both the mechanics and usability of CVSS for the benefit of the information security community. ▢

Acknowledgments

We thank the CVSS SIG members, the SIG scoring analysts, and those who continue to use and implement CVSS.

Reference

1. M. Schiffman et al., *Common Vulnerability Scoring System*, tech. report, US Nat'l Infrastructure Advisory Council, 2004; www.dhs.gov/interweb/assetlibrary/NIAC_CVSS_FinalRpt_12-2-04.pdf.

Peter Mell *is a senior computer scientist in the Computer Security Division at the US National Institute of Standards and Technology (NIST). He is the creator and project manager of the National Vulnerability Database. For this work, he was awarded the 2006 Federal 100 award and the US Commerce Department's bronze medal. He also wrote the NIST Special Publications on patching, malware, intrusion detection, and the Common Vulnerabilities and Exposures (CVE) standard. Mell has an MS in computer science from the University of California, Davis. Contact him at mell@nist.gov.*

Sasha Romanosky *is a PhD student at Carnegie Mellon University researching the economics of information security. He has worked with Internet and security technologies for more than 10 years, predominantly within the financial and e-commerce industries. Romanosky has a BS in electrical engineering from the University of Calgary. He is coauthor of* J2EE Design Patterns Applied *(WROX Press, 2002), developer of the FoxTor tool for anonymous Web browsing, and co-developer of the Common Vulnerability Scoring System (CVSS). Contact him at sromanos@cmu.edu.*

Karen Scarfone *is a computer scientist in the Computer Security Division at NIST. She has coauthored NIST publications on incident response, intrusion detection, malware, and security configuration checklists, and has contributed to several books, including* Inside Network Perimeter Security *(Sams, 2005) and* Intrusion Signatures and Analysis *(Sams, 2001). Scarfone has an MS in computer science from the University of Idaho. Contact her at karen.kent@nist.gov.*