Common Vulnerability Scoring System version 4.0 Consumer Implementation Guide

Document Version: 1.0

Introduction

Executive Summary — TL;DR

The purpose of this guide is to help CVSS consumers use the Threat and Environmental metrics to better reflect the severity of vulnerabilities in their own environments. The CVSS SIG strongly recommends bringing Threat and Environmental metrics into the CVSS lifecycle to provide a consistent, transparent way to align prioritization with real-world risk.

We introduce a practical maturity model that progressively layers Threat and Environmental information onto Base scores (which describe system-agnostic, worst-case severity) to produce deployment-specific scores that account for:

- The current threat landscape and observed exploit activity
- More precise assumptions about local impact to IT assets
- Existing mitigations that materially affect exploitability

When applied through the Threat and Environmental metric groups, CVSS incorporates more information to calculate an enriched score. The result is a scoring approach that moves from

generalized, worst-case assumptions toward decisions grounded in your environment. This improves accuracy and makes your choices about priority easier to explain and defend. This guide includes examples to illustrate how to apply these metrics. Note that while the examples in this guide use the latest version of CVSS, version v4.0, the concepts apply to CVSS v3.0 and v3.1 as well.

Key Takeaway

CVSS Base scores provide a common baseline, but used alone, they lack the deployment-specific context that many programs need. By layering in Threat and Environmental metrics, you can move from worst-case severity to a risk estimate that is aligned with how the vulnerability actually manifests in your environment — a much more customized approach. The incentive is improved prioritization and resource allocation.

What is not in This Guide

Prescriptive solutions
CVSS is a standard, but the ways in which it can be implemented and used by organizations is diverse and individual. This guide provides advice about implementing CVSS without prescribing exact usage.

Recommended processes or tooling
Just as processes are unique to organizations, so are tooling solutions. We point out tools and automation where appropriate, but cannot describe an end-to-end vulnerability management program toolset.

Definitions of terms
For explanation of terms within this document, see the CVSS User Guide Glossary of Terms.

Aim and Audiences

Aim
The purpose of this guide is to explain in detail, with numerous examples, how Section 1.2 of the CVSS Specification Document is to be implemented by consumers of CVSS scores downstream of the initial Base CVSS scores that are typically encountered.

The CVSS standard was initially developed by hardware and software vendors who were responding to community need for a uniform method of rating the severity of a vulnerability. Since its initial release, CVSS has been adopted by many other entities who are tasked with managing a steady stream of vulnerabilities and patches. This document refers to them collectively as “consumers” that ingest CVSS scores that are set by vendors.

With the publication of this document, the main CVSS version 4.0 documentation is as follows:

- CVSS v4.0 Specification Document that defines the Common Vulnerability Scoring System (CVSS) v4.0 standard.
- CVSS v4.0 User Guide aimed at Product Security Incident Response Team (PSIRT) analysts and other entities that aim to score a newly identified vulnerability, preferably before public disclosure.
- CVSS v4.0 Consumer Implementation Guide (this document) aimed squarely at the consumers that ingest these scores into threat intelligence products, research reports, vulnerability scan reports, and, ultimately, the front-line folks who manage vulnerabilities across an organization.

Main Audience

Vulnerability managers and analysts, those of you on the front lines of patch management and mitigation, this document is for you. The other consumer groups are important and stand to benefit, but this guide was made with you in mind. Guidance and examples from practitioners from around the globe have been donated to help others mature their approach to vulnerability management. We encourage you to incorporate threat intelligence and leverage your knowledge of deployed mitigations and compensating controls to achieve the most mature CVSS scores possible.

Secondary Audiences

Hardware/software/cloud/service vendors may appreciate this guide as a way to direct their clients toward better understanding and tailoring product-wide vulnerability scores for their given circumstances.

Threat intelligence and vulnerability management vendors will find that this document highlights multiple ways that CVSS v4.0 vector strings can be supplemented to help their clients evaluate vulnerabilities with more context.

Information security auditors and assessors who are assessing organizations against standards such as PCI-DSS, FedRAMP, or others will benefit from understanding how CVSS scores mature beyond the initial Base score. In turn, this helps auditors and clients achieve and document compliance that takes into account the local deployment environment.

Regulatory bodies are encouraged to understand the CVSS lifecycle and consider using the maturity model, in addition to raw scores, as a way to provide more nuanced guidance and improve compliance.

Researchers and academics are also encouraged to look beyond raw CVSS Base scores and identify how maturing CVSS scores improve vulnerability-related decision making.

Regardless of your perspective, if your mission includes using CVSS scores to make decisions related to vulnerability management, this guide aims to improve your understanding of the often overlooked value of extending CVSS scores.

Table of Contents

CVSS Metric Groups and Lifecycle

CVSS v4.0 defines several metric groups that are important to understand. Vendors provide initial assessments using CVSS Base, and then consumers further refine the assessments using the following metrics:

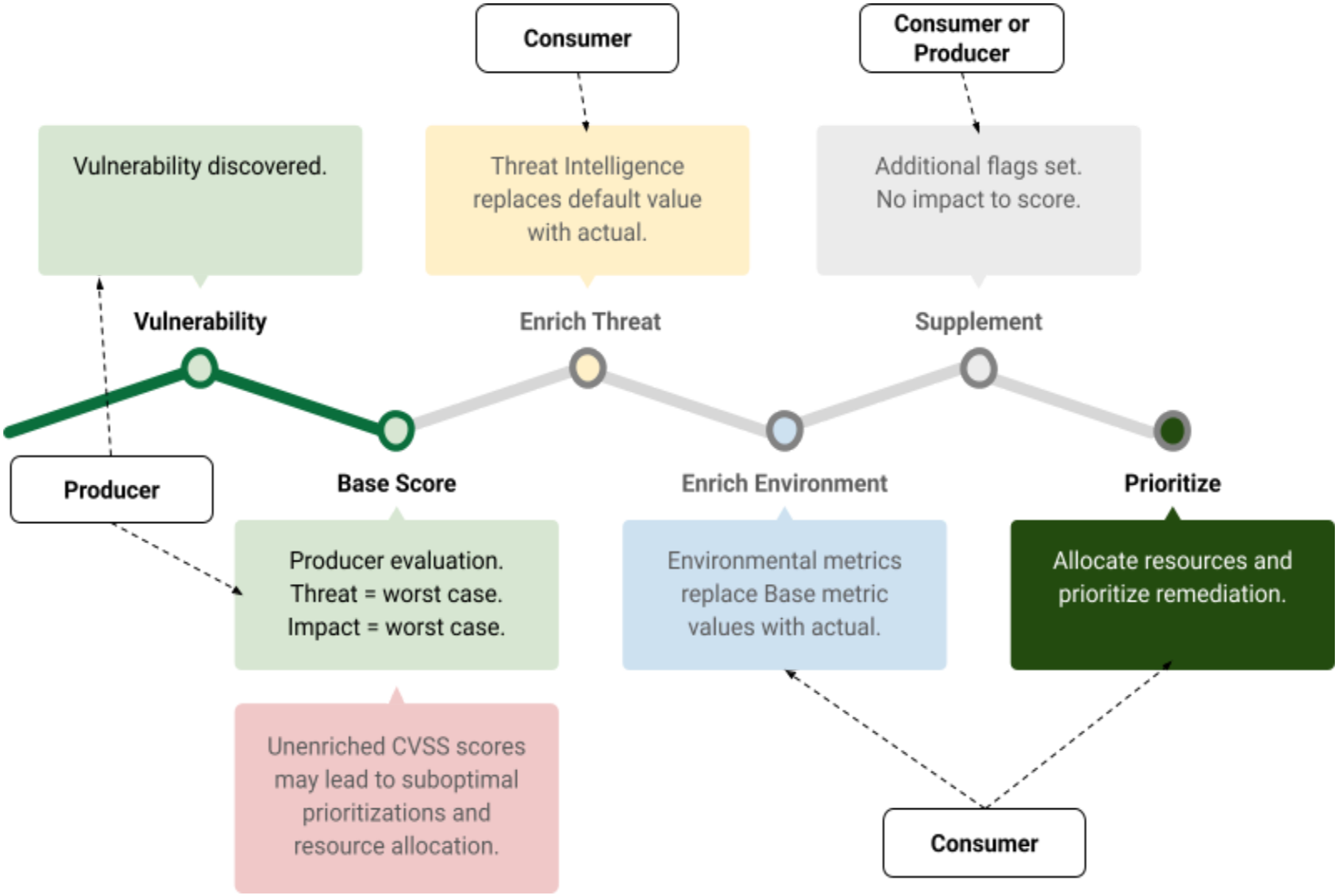
- **Threat:** Contains a single metric that reflects the level of maturity around exploitation tools that a threat actor might use to exploit the vulnerability.
- **Environmental:** Allows local customization, including Security Requirements, to reflect organizational or mission context. This metric category is *explicitly* reserved for use by the consumer of the CVSS score. Environmental assessments represent the expert knowledge of the security and administration teams who manage the products in their organizations, often with mitigations that *should* influence the original Base scores.
- **Supplemental:** Offers optional elements that add context to vector strings.

The *intended* life of a CVSS score begins when a vendor issues a Base score that contains impact and exploitability metrics for a *reasonable worst-case* exploitation scenario in an unmitigated environment. However, the key word here is "reasonable", which does not necessarily mean the absolute worst case in all imaginable scenarios. Instead, it represents a plausible, high-impact scenario that aligns with how assets that contain any given vulnerability is typically deployed and exploited[1]. As such, the consumer is highly encouraged to enhance vendor-provided Base vectors with internal and external threat and business intelligence to create an improved CVSS score that more accurately reflects the severity of the situation in the context of a specific deployment in a specific organization.

Figure 1 depicts the intended lifecycle and describes who is responsible for creating and enriching CVSS scores. Once a vulnerability is discovered, the producer assigns a Base score that intentionally makes worst-case assumptions about the deployment environments. CVSS consumers are responsible for enhancing CVSS scores to reflect actual threat intelligence and their local deployment environment. Bypassing the enrichment stages will produce scores that may lead to suboptimal prioritizations and resource allocation.

[1] https://www.first.org/cvss/specification-document

Figure 1 - CVSS Lifecycle



CVSS Vector Strings

Vector strings encode all the metrics that are related to a CVSS evaluation in a compact form. The term *vector string* is an explicit reference to the metric values that underlie a given CVSS score. CVSS vector strings have thousands of possible permutations but CVSS scores have only 101 steps (0.0 - 10.0 in 0.1 increments). For this reason, many vector strings may yield identical scores. However, as the metrics are enriched, the score will often change to reflect the new permutation. This section offers a brief explanation for those who are unfamiliar with the vector string.

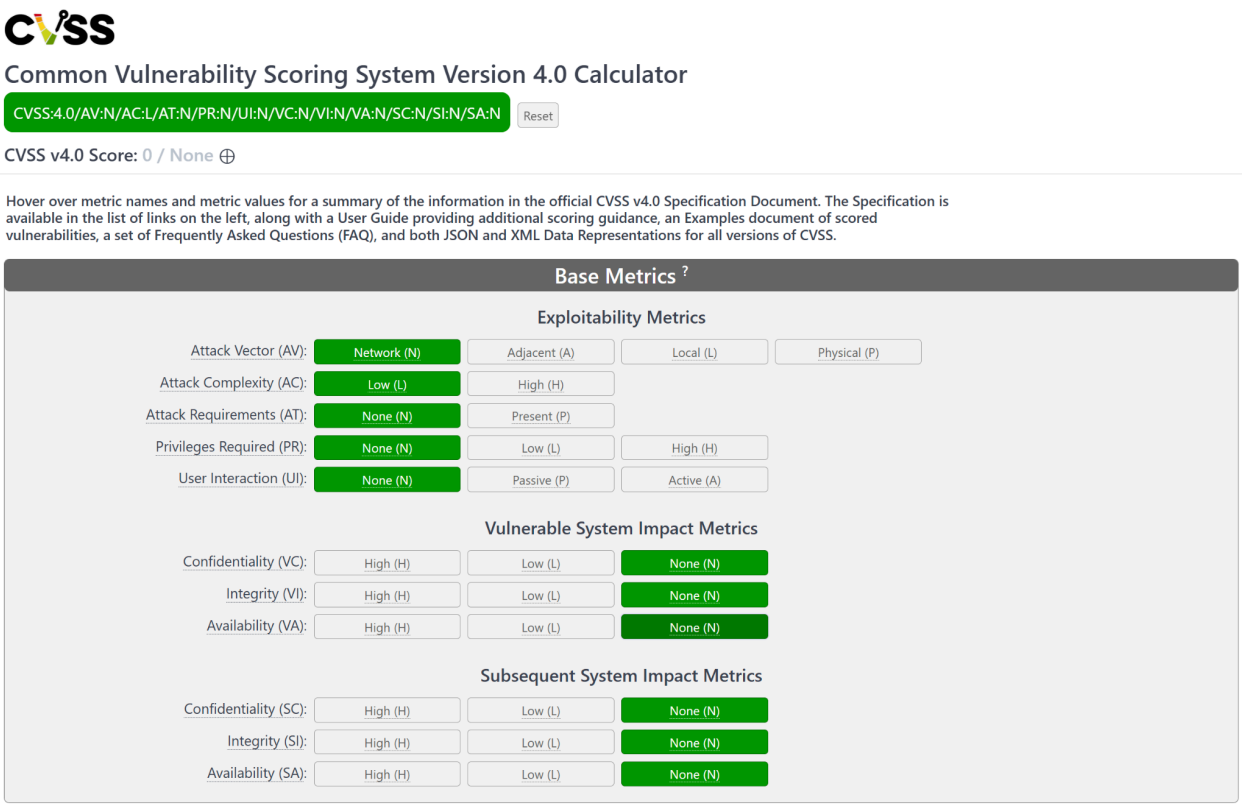
When you open the CVSS v4.0 calculator, you will see a string like the following one at the top in a green bar:

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:N/SC:N/SI:N/SA:N`

This is the vector string. It is a compact, text-based representation of all the choices you might make from the available CVSS metrics to arrive at a CVSS score.The leading part (CVSS:4.0/) identifies the vector string as aligning with the CVSS v4.0 standard. What follows is a list of metric identifiers and metric value pairs separated by a colon. Identifier-value pairs are separated by slashes. The first pair in our example is AV:N, which is the Attack Vector (AV) metric and the chosen value of Network (N). For a full list of the metric names and the values each metric can take, refer to the CVSS v4.0 Specification.

While human readable, vector strings are better suited for machines. Humans will find the visual calculator, hosted by FIRST.org and shown in Figure 2, much easier to use.

Figure 2 - CVSS v4.0 Calculator hosted by FIRST.org



The visual calculator provides buttons with full metric and value names as well as their identifiers. As you make selections in the visual interface, the vector string will update and the CVSS severity score will be recalculated. The important takeaways are:

- Vector strings capture all the metric and value inputs for a given vulnerability.
- Vector strings do not include a severity score, only the metric and value pairs that influence the CVSS severity score.

We refer to the vector string throughout this document as a sort of shorthand for the collection of metrics and values that are *defined* for a given vulnerability. CVSS requires all of the Base metrics to be defined in a valid vector string. However, there are many more metrics that are not required per the standard and they either do not appear in the vector string or their value is set to Undefined (X). One example of an additional CVSS metric is Modified Attack Vector (MAV). You may see vector strings, like the example in Figure 2, where MAV is not present, but you may also encounter a vector string with MAV:X; these are identical in terms of the influence an undefined MAV has on the CVSS severity score.

CVSS Vector String Enrichment

Vector string enrichment is the process of defining metrics beyond CVSS Base so the CVSS score can reflect these important environmental differences.

Metrics that make up the Base group are required and will always be defined. The other metrics are not required but very important when context matters — and context almost always matters. For instance, a piece of vulnerable equipment in a high-security Sensitive Compartmented Information Facility (SCIF) has a different security context than the same piece of equipment being used to provide public WiFi access at a music festival.

Product vendors do not attempt to account for contextual elements because they may not know how customers deploy their products. As a result, they typically present only a CVSS Base score, which is generic enough to cover worst-case scenarios across their entire customer base. Because threats may also change over time, setting Threat and Environmental metrics is left to consumers who are downstream of the product vendor. Each of these metrics are independent and nothing prevents you from defining a couple of Environmental metrics before defining the Threat metric. Define the metrics that make sense and leave the rest.

Furthermore, even within a single organization, vulnerable software that is deployed in one environment versus another environment can have very different contexts when it comes to CVSS scores. The level of differentiation that you choose to implement is a matter of resources and whether or not the added context will result in a significant shift in decision making or posture. The CVSS standard aims to be non-prescriptive, while this guide attempts to show how context is captured when doing so is helpful.

Equally important is that neither the CVSS standard nor this guide draw a hard line between intrinsic context and mitigated context. How an organization interprets a mitigation is left to the consumer, or is perhaps between the consumer and an auditor. The standard is not prescriptive. Here is an example to better clarify:

> *Attack Vector is a required Base metric and Modified Attack Vector is an Environmental metric. A vendor may score a vulnerability with Attack Vector set to Network, which means an attacker can remotely exploit this vulnerability, for which no patch is available. However, the local deployment environment mitigates this by placing the vulnerable device behind a firewall, making it inaccessible to remote attack. One way to encode this local context would be to set Modified Attack Vector to Adjacent in the vector string. When the vulnerability is rescored, the severity will drop to represent the additional context. For compliance and auditing purposes, the choice to set this metric to a different value that more accurately reflects the deployment environment would be documented and perhaps signed off by the appropriate level of management.*

The CVSS standard does not supply a list of mitigation-equivalent value changes like the one we just described. To allow for maximum flexibility, this process is left to the consumer. Likewise, the standard does not prescribe timetables for updating CVSS metrics. Again, this is left to the consumer to define with input, oversight, and documentation that is commensurate with their regulatory environment.

In summary, vector string enrichment is how an organization progresses through the lifecycle and maturity model presented earlier.The next sub-sections will show by example how you might approach using each of the additional metric groups: Threat, Environmental, and Supplemental.

CVSS Threat Metrics

The Threat Metrics group, which encompasses Exploit Maturity, is a critical element of the CVSS scoring system. Data for analysts to implement this updated metric are plentiful. Because in most cases a CVSS score reflects data at a point in time when a Common Vulnerability and Exposures (CVE) identifier is published, in absence of specification of the Threat Metric, the standard calculates the score under the assumption that a malicious actor has a functional exploit. This is the case even though only three to five percent of CVEs have known exploits. This worst-case scenario calculation shows why vector enrichment using the CVSS Threat Metric is important for your organization. Open source threat intelligence from trustworthy platforms allows a consumer to update the Threat Metric, which will lower the numeric score of the CVE if there is no known exploit or only a proof of concept.

The CVSS Threat Metrics group is broken down into four Exploit Maturity levels, as explained in the CVSS Standard.

Deciding which intelligence feeds to trust is up to each organization. There are open source and commercial threat intelligence sources, but a good reference to identify CVEs with known exploits is the free Cybersecurity & Infrastructure Security Agency (CISA) Known Exploited Vulnerabilities (KEV) Catalog. Additional resources for CVSS Threat vector enrichment are Metasploit or the searchsploit tool that allows searches of ExploitDB by CVE. It should be noted that these data sources can be accessed programmatically using simple python scripts.

CVSS Environmental Metrics

The CVSS Environmental Modified Base Metrics group is where vulnerability analysts can start to see the results of evaluating vulnerabilities more closely. The name that the CVSS standard assigned to this metric group derives from the fact that the data that is necessary to populate these metrics must come from the specific environment in which the vulnerable system is deployed. A seasoned vulnerability analyst who is leveraging environmental metrics while assessing incoming vulnerabilities can significantly reduce the patching and mitigation load on an organization. Authors of this document have seen significant efficiency gains when applying these principles in the real world. In the following sections, we will identify how to properly

update Environmental Metrics with specific examples. The CVSS SIG recommends that controls are validated before making adjustments to Environmental Base Metrics.

The v4.0 CVSS Specification document defines Environmental Metrics. Sample data sources for environmental metrics available for most organizations include:

- **External vulnerability scanner:** If a vulnerability is not detected by external scan, it may make sense to assume MAV:A for all AV:N within the scope.
- **Internal monitoring system:** If an asset is included into monitoring, it is likely that it has at least availability requirements (AR) and will probably source some other metrics, too.
- **Data from previous security audits and Crown Jewel Analysis:** These may be useful for setting CR/IR/AR for relevant systems.
- **Endpoint security tools:** Check protection status and adjust MAC accordingly.

Asset Management Systems and Environmental Metrics

Organizations often maintain an asset or configuration management system that might readily help the organization enrich their CVSS data by adding more environmental context.
It is not a roadblock in that a typical configuration management database is far from being 100 percent up to date and accurate, but some data is always better than none.

For example, a configuration management database (CMDB) might include asset information like the use of a proxy-based firewall or the fact that it is accessible only on a secure network using SSH. Another critical piece of information that is often tracked is where a device is located within the network (Topology). Does the vulnerability reside on a router that is exposed to the Internet or is it on an internal web server that is accessible only from the corporate LAN protected by a proxy-based firewall? Integrations with IP Asset Management (IPAM) systems can potentially allow for CVSS Environmental scoring updates in bulk when entire subnets are identified as being protected by a proxy-based firewall or accessible only through secure jump hosts.

In some cases, each asset is also tagged with the relevant confidentiality, integrity, and availability requirements.In the banking sector, this is actually a formal requirement for operational risk management that is mandated by central banks.

Based on the experience of the authors, here is a list of asset attributes that might contribute to significantly changing the score of a vulnerability from the Base score and, therefore, the urgency and nature of additional mitigation measures. We illustrate how things may change in the next few examples.

Table 1 - Asset attributes for modified environment metrics

Data Element	Applicable CVSS Metric

Asset behind a proxy	Modified Attack Vector (MAV)
Asset behind an IPS	Modified Attack Complexity (MAC)
Asset Configuration Baselines	Modified Attack Requirements (MAT)
Asset Privilege Configuration Settings	Modified Privileges Required (MPR)
Application Allow or Deny Listing	Modified User Interaction (MUI)
Asset Encryption Settings	Modified Confidentiality (MVC)
Immutability of the Asset	Modified Integrity (MVI)
Asset Load Balancing Configuration	Modified Availability (MVA)
Data Confidentiality Requirements	Confidentiality Requirements (CR)
Data Integrity Requirements	Integrity Requirements (IR)
Data Availability Requirements	Availability Requirements (AR)

Modified Exploitability Metrics

Modified Attack Vector (MAV)

A modified Attack Vector is one of the most straightforward data points an analyst can find. It illustrates how important it is for analysts to understand the environment because this additional information can change the level of urgency for updating the software. For more information, see CVSS User Guide - Section 3.8 : Vulnerable Systems Protected by a Firewall.

Most systems on a modern network will be protected by a firewall. Exceptions include network devices such as wireless access points, edge routers, or the edge firewalls themselves.

Per CVSS Specification Document Section 2.1 Table 1: Attack Vector, in the case that an environment has a hardened management network or secure access network where systems are accessible in ways other than just terminal emulation, the Modified Attack Vector can be updated from Network (N) to Local (L).

Take, for example, a CVSS vector string expressing a vulnerability in a network device that allows an attacker to gain complete control over a device attached to the network, with high impact to Confidentiality, Integrity, and Availability.

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

Observe the changes to the numeric score when applying modified metrics, as shown here in Table 2:

Table 2 - Modifying the Attack Vector Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Attack Vector	System protected by a firewall	Network to Adjacent	**9.3** -> **8.7**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A**			
Modified Attack Vector	System accessible only from console	Network to Local	**9.3** -> **8.6**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:L**			

While updating the Modified Attack Vector (MAV) from Network (N) to Local (L) may be less common, a clear example of this is an isolated web server that is running on a system that an administrator accesses through Remote Desktop Protocol (RDP). Although the web server is bound to a network protocol stack (TCP), the service is exposed locally on “localhost” or 127.0.0.1 only. In this case, the administrator would be forced to RDP to the console of the web server and access the website “locally”.

Modified Attack Complexity (MAC)

The Modified Attack Complexity (MAC) metric is also simple to update. Take the example of a vulnerability in a web server with known exploits. When that web server is behind an intrusion prevention system (IPS), if the IPS has a validated signature for the vulnerability, the analyst can change the Modified Base Metric of Access Complexity (MAV) from Low (L) to High (H).

Similar protections, such as enabling memory protections on systems that must be overcome before an attacker can accomplish an exploit, or other mitigations that may make exploitation unreliable, relate to Modified Attack Complexity.

Here is an updated CVSS vector string to help you understand the data that is displayed and how it impacts the overall CVSS score, continuing the example from before.

Table 3 - Modified Attack Complexity Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Attack Complexity	System protected by a IPS	Low to High	**9.3** -> **7.7**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H**			

The standard combination of placing internal assets behind a proxy-based firewall with IPS protections has a significant impact on the overall CVSS numerical score.

Modified Attack Requirements (MAT)

The Modified Attack Requirements (MAT) metric may be more complex to update at scale and is dependent on the vulnerability and how it is exploited.

To illustrate how you can deploy a mitigating control we're going to use the example of CVE-2024-23897, a critical vulnerability in Jenkins.

Jenkins commonly makes up part of software build pipelines. These systems are ephemeral, running only during build processes and then either turned off or removed entirely, and are likely accessible only from protected networks or from within limited environments. Where systems are available only during certain windows while the software build process occurs, this timing window creates a requirement that the attacker cannot control.

It is important to understand the difference between Attack Complexity and Attack Requirements. Review the standard definition on Attack Requirements from the CVSS v4.0 Specification Document.

Attack Complexity deals with things like address space layout randomization (ASLR), data execution prevention (DEP), or an IPS. Think carefully about where to apply Modified Attack Complexity (MAC) vs Modified Attack Requirements (MAT) in your analysis.

Next, we will compare how modifying Attack Requirements updated the CVSS numerical score when continuing with the example.

Table 4 - Modified Attack Requirements Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Attack Requirements	Systems are not always available and cannot be exploited at-will.	None to Present	**9.3** -> **7.7**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H/MAT:P**			

Modified Privileges Required (MPR)

Modifying the Privileges Required to block all vulnerabilities may be challenging, but for especially critical vulnerabilities it may be worth it to implement compensating or mitigating

control. One example is a PowerShell vulnerability, CVE-2022-41076 (scored in CVSS version v3.1). In this case, Privileges Required were Low (L) and the attack could allow an attacker to gain elevated privileges. If the low-privileged user could not execute PowerShell through Group Policy, you can update the Modified Privileges Required (MPR) to High (H).

Continuing with our previous vector string, we are able to move our Privileges Required (MPR) from Low (L) to High (H) and the resulting CVSS v4.0 score decreases again.

Table 5 - Modified Privileges Required Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Privileges Required	Restricting access to administrative users	Low to High	**9.3** -> **7.3**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/MAV:A/MAC:H/MAT:P/**MPR:H**			

We want to take a moment here to address some concerns readers may have. Thus far we have changed several Environmental (Modified Base Metrics) and seen some improvements in our CVSS scores but we have not seen a meaningful drop. Now, we want to jump ahead to make sure we keep you interested. Consider now adding threat intelligence that relates to Threat Metrics and you could see a significant change in score. Depending on the source of threat intelligence, the determination of existing threats may differ. In this example, some sources of threat intelligence do indicate a proof-of-concept related to CVE-2022-41076, and the example here reflects that. However, choosing other sources of threat intelligence may change this determination. The choices and sources of threat intelligence are a substantive decision of each organization.

Table 6 - Additional Modified Privileges Required Example

Vector	**Modification justification**	**Metric modified**	**CVSS-B**	**CVSS-BTE**
Exploit Maturity	No reported exploits from available threat intelligence	Not Defined to Unproven	**9.3**	**6.4**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:P**/MAV:A/MAC:H/MAT:P/MPR:H				

A numerical score of 6.4 (medium) moves the needle! We will get to the Threat Metrics shortly. We wanted to show you that there are ways to significantly impact your CVSS using Threat Metrics and Modified Environment Metrics.

Modified User Interaction (MUI)

It may be difficult to modify User Interaction metrics at scale. Some organizations implement *application white* or *allow listing*. This allows organizations to “allow list” using characteristics that are controllable, such as developer certificate or binary signature.

Any new binary that attempts to execute is met with a warning where the user is required to present a justification as to why they need to execute the application. This notification is sent to an administrator, who can approve or deny the execution.

Another example that may allow a vulnerability analyst to move the Modified User Interaction (MUI) from Passive (P) to Active (A) is enforcing requirements for signed PowerShell scripts. Of course, this would apply only to PowerShell vulnerabilities but it is effective. The victim would need to request that the PowerShell script be permitted to execute, thereby increasing the User Interaction to an Active (A) state.

Finally, consider a document-based attack. Some organizations have systems that check and sandbox documents to validate that they are safe. Documents or attachments received through email can be sandboxed and detonated, validating that they are safe. Document-based attacks involving users are typically rated as User Interaction: Passive. In an environment implementing document sandboxing, an attacker must then convince a user to first retrieve a document from a secure filtering store and then view it on the local system, bypassing protections and best practices, thus changing User Interaction from Passive to Modified User Interaction Active (A).

Continuing with our vector string from the above example:

Table 7 - Modified User Interaction Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified User Interaction	System protected document sandboxing	Passive to Active	**9.3** -> **5.4**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/MAV:A/MAC:H/MPR:H/**MUI:A**			
Exploit Maturity	No reported exploits from available threat intelligence	Not Defined to Unproven	**9.3** -> **2.0**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:U/MAV:A/MAC:H/MPR:H/MUI:A**			

These vector string enrichments can have a significant impact on your vulnerability program!

Modified Vulnerable System Impact Metrics

As we move into the Modified Vulnerable System Impacts we need to remember that we are dealing with compensating controls to protect the confidentiality, integrity, and availability requirements of the system itself, not the data that is present on the system. Compensating and mitigating controls may benefit data security as a byproduct, but for the purposes of CVSS scoring that is not their primary function.

Modified Vulnerable System Confidentiality (MVC)

When considering how to implement a control to update the Vulnerable System Impact Metrics to Modify Confidentiality there are some well understood options. For example, consider a database that implements column level encryption with an external key management solution (KMS). It is common in cloud environments for an application to leverage cryptography keys stored in an external system such as AWS Key Management Service, Azure Key Vault, or other hardware security module (HSM) and key management solutions. If the database system itself is compromised an attacker cannot access the encrypted data within the database unless otherwise compromising the key management service and gaining access to the encryption key.

Table 8 - Modified Vulnerable System Confidentiality Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Vulnerable System Confidentiality	Encryption protects data at rest	High to Low (or None)	**9.3** -> **4.6**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H/MPR:H/MUI:A/MVC:L**			
Exploit Maturity	No reported exploit	Not Defined to Unproven	**9.3** -> **0.7**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:U/MAV:A/MAC:H/MPR:H/MUI:A/MVC:N**			

This is another situation where it would be fairly easy to identify systems where the Modified Vulnerable System Confidentiality (MVC) can be updated. If the organization has a policy that all workstations and servers must be encrypted, these changes can be applied in bulk to CVSS scores.

Similarly, protecting applications and endpoints against data loss makes successful exploitation of confidentiality-impacting vulnerabilities more difficult. Strong evidence of mitigating controls is critical and a clear understanding of where those controls live in the attack path from the threat to the vulnerability present in an asset in your environment will allow you to make intelligent decisions on modifying the Attack Complexity Exploitability Metric.

Modified Vulnerable System Integrity (MVI)

Updating the Modified Vulnerable System Integrity (MVI) values could be simple or complex, depending on your environment. In a modern environment with containers, you have the option to run immutable (read-only) containers. If there is a vulnerability within any immutable container, the data within that container cannot be changed or updated because the container itself is immutable.

Systems can be hardened with read-only file systems, similar to some Linux distributions that are released as immutable distributions. This means that the operating system is read-only so you cannot make any changes to the core operating system.

Implementing an immutable container or operating system may allow you to lower the Modified System Integrity (MVI) value to Low (L) or None (N), depending on your comfort level as an analyst.

See the following respective CVSS vectors.

Table 9 - Modified Vulnerable System Integrity Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Vulnerable System Integrity	Read only file system	High to Low (or None)	**9.3** -> **4.4**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H/MPR:H/MUI:A/MVC:L/MVI:L**			
Exploit Maturity	No reported exploit	Not Defined to Unproven	**9.3** -> **0.7**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:U**/MAV:A/MAC:H/MPR:H/MUI:A/MVC:N			

Modified Vulnerable System Availability (MVA)

Updating the Modified Vulnerable System Availability (MVA) values can be fairly straightforward, depending on your application. Many modern application architectures reside behind a load balancer. Network appliances can be deployed in redundant configurations or topologies. Cloud environments commonly auto-scale. The function of these technologies is to provide additional resources as they are needed. Each of these examples mitigates potential loss of availability due to exploitation of a vulnerability.

An analyst updates the Modified System Availability (MVA) metric based on policy. The likely scenario in this case is that the analyst would adjust the value to Low (L) because availability

could still be impacted, unlike an immutable container that is a functionally equivalent compensating control for Modified System Integrity (MVI) value.

Table 10 - Modified Vulnerable System Availability Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Vulnerable System Availability	Redundancy reduces impacts	High to Low (or None)	**9.3** -> **5.3**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H/MPR:H/MUI:A/MVA:L**			
Exploit Maturity	No reported exploit	Not Defined to Unproven	**9.3** -> **1.9**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:U**/MAV:A/MAC:H/MPR:H/MUI:A/MVA:L			

When you start updating all of the CVSS Vulnerable System Impact Metrics, the CVSS numerical scores start to drop quite low. Here is a CVSS vector string where all Vulnerable System Impact Metrics have compensating and mitigating controls that allow the vulnerability analyst to assign Low (L) for Confidentiality (MVC), Integrity (MVI), Availability (MVA). With all previous Environmental updates and Vulnerable System Impacts updated the CVSS numerical score is 1. Here is the CVSS vector string for that vulnerability configuration.

Table 11 - Additional Modified Vulnerable System Availability Example

Vector	**Modification justification**	**Metric modified**	**Score Change**
Modified Vulnerable System C, I, and A	Combined mitigations	High to Low (or None)	**9.3** -> **1.0**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**MAV:A/MAC:H/MPR:H/MUI:A/MVC:L/MVI:L/MVA:L**			
Exploit Maturity	No reported exploit	Not Defined to Unproven	**9.3** -> **0.1**
CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/**E:U/MAV:A/MAC:H/MPR:H/MUI:A/MVC:L/MVI:L/MVA:L**			

CVSS Security Requirements (Environmental)

While Environmental Modified Base Metrics accounts for the effect of compensating or mitigating controls on the system, Environmental Security Requirements allow the vulnerability analyst to enhance the scoring system based on importance of the vulnerable asset.

Environmental Security Requirements are not in the Modified Base Metrics section of the calculator because they do not have a corresponding metric within the CVSS Base Metrics. The Environmental Security Requirements stand alone like the CVSS Supplemental Metrics; however, Environmental Security Requirements do update the CVSS numerical score. The Environmental Security Requirements have three metrics:

1. Confidentiality Requirements (CR)
2. Integrity Requirements (IR)
3. Availability Requirements (AR)

Each of these metrics have four values as documented in the CVSS Specification Document.

You may ask what a practical application of Environmental Security Requirements looks like. Some organizations have cafeterias with digital signage. The systems that run the signage likely have software applications and a database that holds all the information required for the menus. However, that data may have Low (L) Confidentiality Requirement (CR) because the data is public. There may be Low (L) Integrity Requirements (IR) because if the menu is changed there is little to no impact on the organization. Finally the Availability Requirements (AR) may be Low (L) because if the menu is not available, the employees can simply ask the staff about lunch. This is a simple example, but others exist. Most organizations have a file server or service somewhere that holds holiday photos or other such items.

Those without specific knowledge of your environment can not provide data about the value of data on assets in your environment. This is the reason why the CVSS Base scores provided by vulnerability scanners, while correct, provide little value in assessing the impact in regard to the specific environment of an organization. Unless the scanning platform has the context of your environment, the data it gives you will be incomplete at best, or wrong at worst.

A simple process for identifying key systems involves identifying what systems are most important to running the business, and classifying other systems as having reduced security requirements. Assessments on these most important systems would have higher scores, reducing scores on other systems, prioritizing vulnerability management on the most important systems, all other things being equal. Again it is up to the discretion of the vulnerability analyst as aligned with organizational policy, but all this information is required for accurate CVSS vector strings and scores.

CVSS Supplemental Metrics

CVSS Supplemental Metrics are a way for organizations to add additional context based on their organization. For example a healthcare provider may wish to add more information to highlight if a human could be injured as an impact of a vulnerability. It is important to state again that Supplemental Metrics do not update the numerical score but they are represented in CVSS vector strings. This is another reason why vulnerability analysts should not rely only on the numerical score but also use the vector string.

The Supplemental group consists of four metrics that augment impact (Safety, Recovery, Value Density, Response Effort), one that augments exploitability (Automatable), and one that serves a completely different purpose: Provider Urgency.

The impact metrics of the Supplemental group, despite not being used to derive a numeric score, enable implementing a separate handling process for the most important vulnerabilities: those that affect human safety, cause permanent damage or extreme recovery costs, or affect highly valuable information.

The Automatable metric is used to mark vulnerabilities that could potentially enable self-propagating attacks (aka “wormable”). Such attacks can cause important infrastructure damage with limited involvement of the attacker after the launch.

Provider Urgency may be interpreted by consumers to add context to other evaluations from a scoring provider. This metric should not be used as a shortcut for the full implementation of CVSS-BTE. This metric is sometimes used by software supply chain vendors that do not supply their own CVSS vectors for various reasons (Curl, Canonical, Debian are known examples, it also applies to Docker container publishers). CVSS consumers should not use a single metric, including the Provider Urgency value, when making decisions for their environment.

Provider Urgency should be used with caution. The metric as provided by the “packaging vendor” does not assess the severity or impact of the vulnerability itself. Marking the vulnerability as low priority, which translates to Clear value may indicate something else, like, the upstream product is EOL and not supported anymore, thus considered low urgency for the solution provider, but not for the end user.

CVSS Maturity

Enhancing a vector string with additional metrics beyond the Base metric group forms a five-step maturity model. It begins with zero, which indicates no CVSS scores are being used at all. From there, it progresses through four stages of maturity that culminate in a fully expressed vector string.

It might be tempting to interpret the levels as a measure of score accuracy, but that would be incorrect. A CVSS Base score reflects the severity of a vulnerability according to its intrinsic characteristics without specific knowledge of the environment provided by a vendor which tries to take into account the worst case scenarios for deployment across all the environments in which its client base may be using their system. Adding threat intelligence to reach level two introduces an additional severity factor—one that is temporal in nature. The key metric, Exploit Maturity, is something that will change over time and may also be specific to the deployment environment. Similarly extending a Base vector string with Environmental and Supplemental metrics to reach levels three and four simply include information that is relevant only to a specific deployment environment. For these reasons, CVSS maturity is best interpreted as a *level of enrichment*, not a level of accuracy.

Table 12 provides a quick reference to the abbreviations used to denote CVSS score (or vector strings) that have been enhanced with metric values from the four metric groups discussed previously. Base (CVSS-B) is always included because the Base metrics are *required* for all CVSS scores. Then, Table 13 provides a compact illustration of the CVSS Maturity Model. Following that depiction, the section explains the principles behind its creation and then defines each level of maturity in detail.

Table 12 - CVSS Enrichment Abbreviations

Metric Group	Abbreviation	Notes
Base	CVSS-B	A.K.A. CVSS Base
Threat	CVSS-BT	Typically the first step in enrichment
Environmental	CVSS-BE	Environmental enhancement by itself is rare
Threat and Environmental	CVSS-BTE	Very mature combination

CVSS Maturity Model and Vector String Map

Table 13 - The CVSS Maturity Model*

Level	Label	Base	Threat	Env	Supp	Provider	Description	Best for
0	N/A					N/A	No CVSS.	N/A
1	Base (CVSS-B)	X			+	Vendor	CVSS Base, including vector string data that reflects intrinsic vulnerability information.	Vendors, academics, policy makers, etc.
2	Base and Threat (CVSS-BT)	X	X		+	Threat Intelligence	CVSS Base enriched with threat intelligence about exploit availability in an automated fashion.	3rd party threat intelligence, pen-testers, etc.
3	CVSS Base, Threat, Environmental (CVSS-BTE)	X	X	X	+	Consumer	CVSS Base enriched with threat intelligence about exploit availability and complete environmental data in an automated fashion.	Front line entities that deploy potentially vulnerable products and services.

Note: The CVSS Maturity Model also aligns to the Analyze/Prioritize row of the SANS Vulnerability Management Maturity Model.

Principles for CVSS Maturity Levels

Several principles were identified when considering the maturity levels for CVSS. These principles guided the ordering of the CVSS maturity levels. The goal is to make CVSS maturity accessible and relevant to as many organizations as possible, not just those with large budgets. The principles are as follows:

Automatable at Scale

Recommendations favor those that could be executed at scale in an automated way. With a small asset footprint, managing vulnerabilities manually may be possible. However, if you are working in a larger environment with tens or hundreds of thousands of systems, it is impractical to do high-quality vulnerability analysis manually.

Difficulty of Obtaining Data

Recommendations prioritize data that is easily accessible. This is a primary reason that CVSS-BT became maturity level two. There are several high-quality datasets available online or through various downloadable tools. For organizations with larger budgets there are commercially available data sources available as well. All referenced vulnerability intelligence sources are available using programmatic means, which is critical for automating the vulnerability analysis processes.

Cost of Obtaining Data

Likewise, cost has an impact on obtaining data. A vulnerability analyst should not require expensive resources to do basic prioritization. The vulnerability threat intelligence options provided here favors free options. This will help organizations reach maturity with less organizational resistance.

Level of Effort to Generate Data

When moving to CVSS Maturity Level 3 there are few, if any, external sources of information to update Environment Modified Base Metrics. Vulnerability analysts will need to work with the business to identify and apply data that is unique to their environment. Tooling like an asset management or configuration management database are highly recommended, either by creating such a system internally or by procuring a commercial solution.

In addition to procuring asset management systems, there is a non-trivial amount of work required to maintain the system and ensure that the data that the system manages remains complete and trustworthy. Free and open-source tools exist that can assist with network-based discovery scans. The recommendation is to start implementing Environment Modified Base Metrics that are easier to locate data for such as Attack Vector (AV) or Environmental Security Requirements. See the section Asset Management Systems and Environmental Metrics later in

this document for guidance on the kind of data that should be present in an asset management system. A little ingenuity can go a long way when implementing Environment Modified Base Metrics.

Levels of Maturity

With these principles in mind, the next four sections describe each level of the maturity model. Examples are available in the respective Threat, Environmental, and Supplemental sections later in this guide. A key attribute of a maturing CVSS program is that vulnerability evaluation eventually becomes more automated. Also, because nuances in the vector string help analysts make better decisions, maturity toward understanding and using the vector string directly is a sign of maturation. CVSS numeric scores have limits that vector strings can overcome.

CVSS Maturity Level Zero

Level Zero represents an organization not using the CVSS scoring methodology.

As a starting point, open source tools are available to gather minimal vulnerability information. Some tools include NMAP for network-based vulnerability scanning or Wazuh for agent-based scanning. Once a CVE is identified within an environment, the National Vulnerability Database (NVD) or CVE website can be used to ascertain the CVSS score and vector string for most vulnerabilities. The NVD database can be accessed for free through the API. You can request an NVD API key at their NVD website Request an API Key page.

CVSS Maturity Level One

Level One represents an organization that is leveraging only the CVSS Base score without further enhancements.

While a good first step, this level of maturity may lead to suboptimal prioritization and resource allocation. The primary reason is that CVSS Base scores use default values for Threat and Security Requirement metrics that are based on a worst-case deployment environment.

CVSS Maturity Level Two

Level Two represents an organization that is setting CVSS Threat Metrics with reliable threat intelligence sources.

Depending on current sources, such as FIRST/EPSS, Cyentia, and CMU SEI, less than six percent of known vulnerabilities have functioning exploits. Understanding which vulnerabilities in your environment have working exploits can help you make informed decisions on how to prioritize remediation. Without this enrichment, the Base scores assume a working exploit is available.

CVSS Maturity Level Three

Level Three represents an organization that is integrating CVSS Environmental Modified Base Metrics and Environmental Security Requirements metrics in addition to Threat metrics from Level Two.

Using Environmental metrics likely requires greater effort in time and resources than maturing from CVSS Level One to Level Two. One reason is that the organization implementing the CVSS standard must generate the data themselves. To achieve automation maturity, the data must be available in a programmatic fashion to operate at scale. It is recommended that organizations start by looking for opportunities to apply Environmental Modified Base Metrics that are easier to generate data for than those that are specific to individual vulnerabilities. Organizations may also consider focusing data automation efforts on business critical systems and those controls that represent the most efficient returns on investment for reducing patching and remediation cycles to drive better decision making.

The Environmental Security Requirements in CVSS Base scores default to worst-case scenario values. Explicitly setting these metrics provides greater accuracy and enriches both the score and its vector string. Without this adjustment, Base scores assume all impacts are maximally harmful, which can lead to suboptimal prioritization and resource allocation.

CVSS Maturity Plus “+”

Supplemental metrics are available at all levels of maturity above zero. These metrics do not impact the numeric severity score, but we encourage their use to add context and nuance to scores. When supplemental metrics are in use, a plus sign “+” can be added to indicate a Supplementally enriched vector string.

Key Take Away

In summary, the CVSS maturity model represents the milestones CVSS consumers should strive to achieve.

The optimal maturity level will depend on available data and resources and whether or not achieving a higher level of maturity will lead to better vulnerability management. Better decision making should always be the primary measure. With the lifecycle and maturity model as background, we are finally ready to dig into the specifics of how the Threat, Environmental, and Supplemental metric groups are used in practice.

Conclusion

In this guide we have highlighted how CVSS Base scores alone should not be used for customer CVSS scoring and vulnerability program management. CVSS Base scores are a worst-case scenario impact assessment of an instance of a vulnerability and likely do not reflect real-world severity. This implementation guide demonstrates how to enrich and refine CVSS assessments by leveraging metrics beyond the Base group, which may often result in meaningful reductions in CVSS numerical scores in consumer environments.

Similarly, auditors and assessors should use the most mature assessment available to evaluate the severity of vulnerabilities in an environment. The greater the enrichment to the CVSS score, the more accurate the numerical score is for that environment. Each organization and analyst must make a decision on what threat intelligence sources are valuable and applicable for their organization.

It is our hope that this document has provided clear levels of maturity for adopting the CVSS standard and how CVSS consumers can implement these levels of maturity within their organizations. We welcome feedback about this guide and the standard in general at cvss@first.org

Appendix A - Acknowledgements

FIRST and the CVSS SIG recognizes the authors of this guide, listed in alphabetical order by last name:

- Steven Aiello
- Rob Arnold, Acorn Pass
- David Glosser
- Cat Gibbs
- Nick Leali, Cisco
- Art Manion
- Fabio Massacci
- Vivek Nair, Microsoft
- Alex Smirnoff, Glanc, ltd
- José L. Sosa, JOESOS Consulting
- Oren Yulevitch

Thanks to Michele Cohen of Cisco for assistance with the editorial review.

Special thanks to Grace Staley from CAPS, LLC. for extra effort in facilitating this guide and the working group.

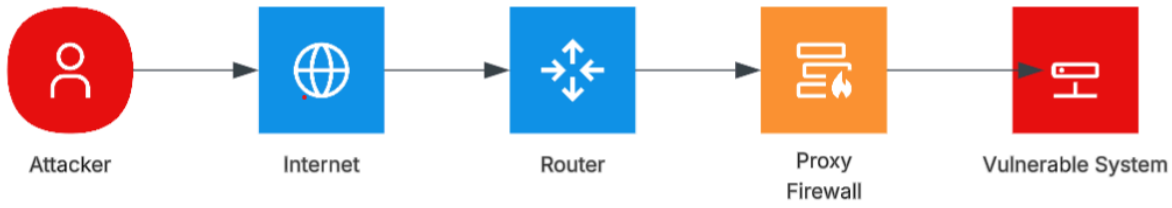
Appendix B - Network Architecture Examples

The Impact of Secure Architecture on CVSS Scores

As you have seen throughout this document, and specifically when dealing with modifying Base CVSS metrics, implementing secure design principles can have a significant impact on the severity of vulnerabilities in a consumer environment. The CVSS SIG wanted to provide architectural examples of Modified Base Metric configurations with pictographic examples. It is important for vulnerability analysts to know that multiple compensating and mitigating controls can be used collectively to reduce the overall severity of the vulnerability system.

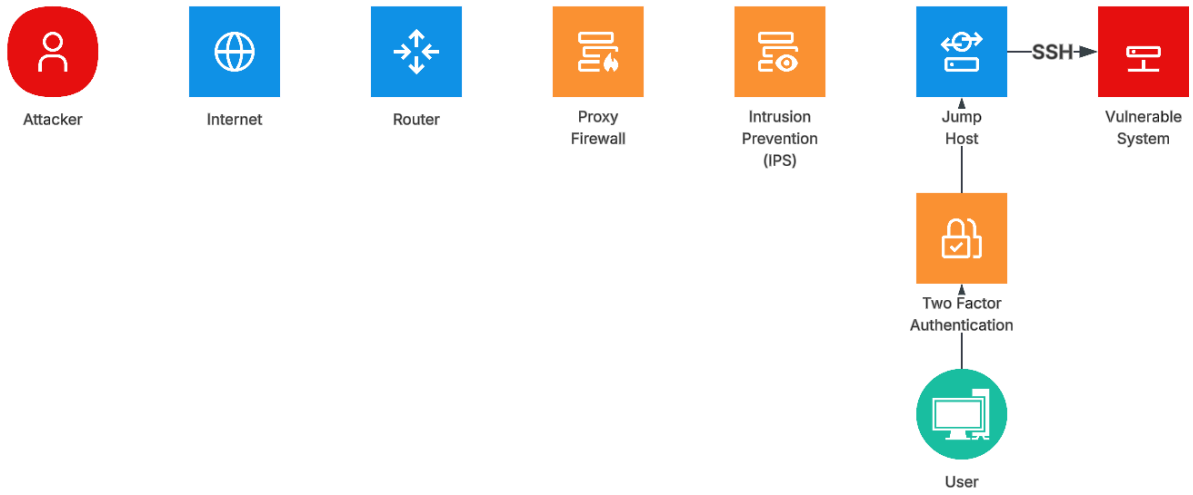
Sample Network Configurations and CVSS Score Impacts

Figure 3 - Architectural Example: Modified Attack Vector (MAV) Adjacent



In this configuration example, the vulnerability analyst has the option to change the Attack Vector (AV) of Network (N) to Modified Attack Vector (MAV) to Adjacent (A).

Figure 4 - Architectural Example: Modified Attack Vector (MAV) Local

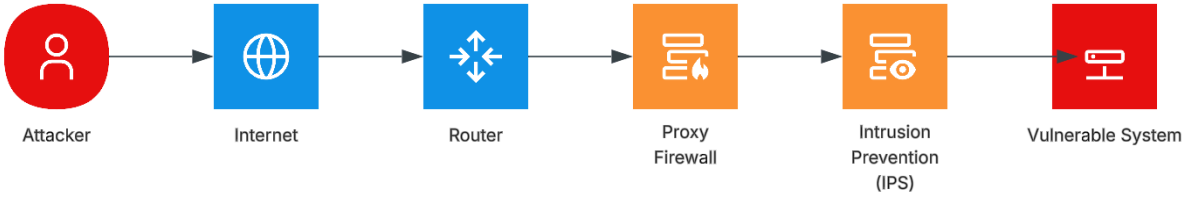


In this configuration example, the vulnerability analyst has the option to change the Attack Vector (AV) of Network (N) to Modified Attack Vector (MAV) to Local (L). The important configuration factor in this example is that the vulnerable system is available only through a keyboard, console, or terminal emulation, as seen in the Attack Vector specification language:

"The attacker exploits the vulnerability by accessing the target system locally (e.g., keyboard, console), or through terminal emulation (e.g., SSH); or"… (RDP)

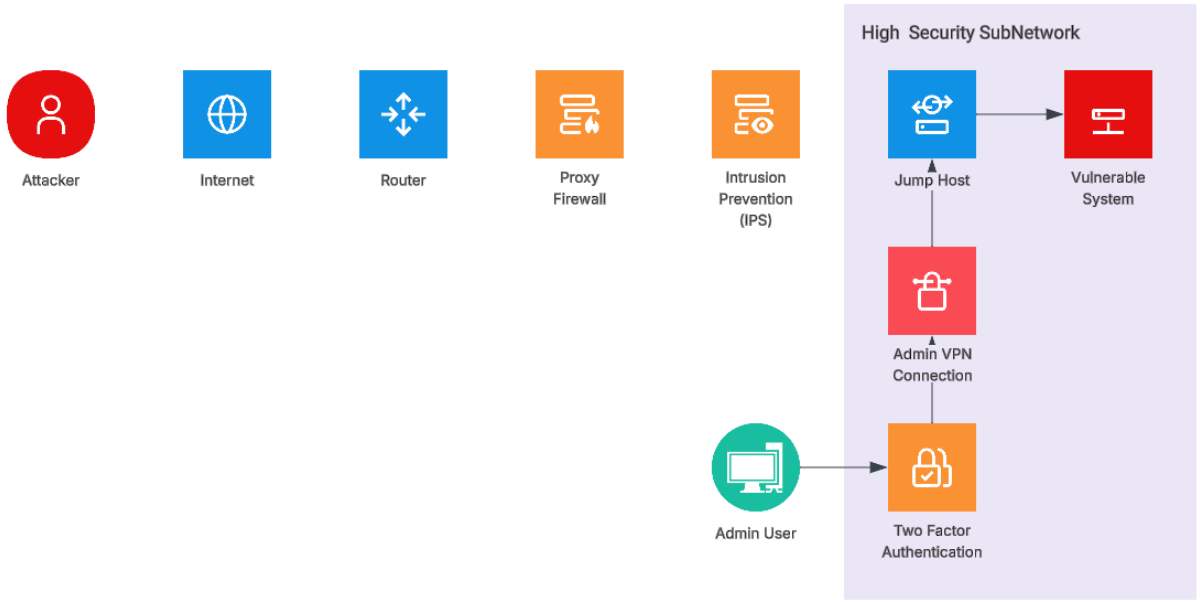
Other "network"-based access does not allow for this CVSS Modified Attack Vector (MAV) update.

Figure 5 - Architectural Example: Modified Attack Complexity (MAC)



In this configuration example, the vulnerability analyst has the option to change Attack Vector (AV) of Network (N) to Modified Attack Vector (MAV) to Adjacent (A). Additionally, because there is an intrusion prevention system (not detection system), the vulnerability analyst has the option to change Modified Attack Complexity (MAC) to high. Note that this requires that the vulnerability analyst verify that the control is in place using such processes as validating a proper IPS signature or through manual testing.

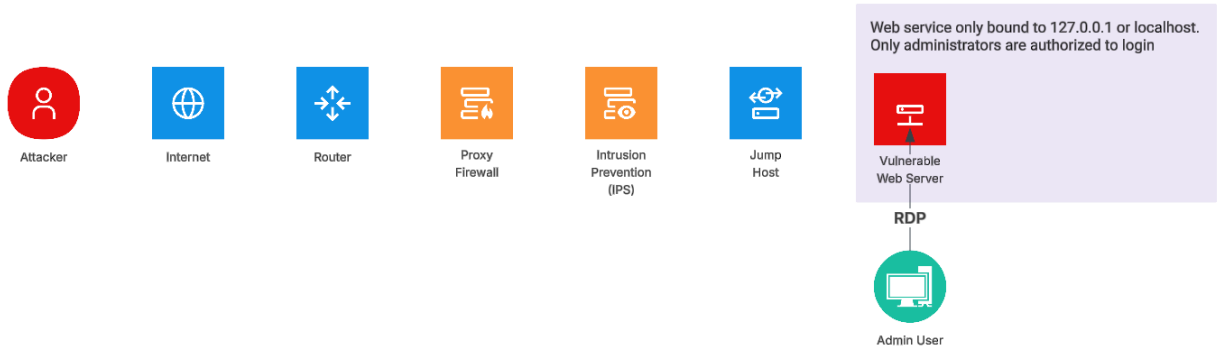
Figure 6 - Architectural Example: Modified Privileges Required (MPR)



In this configuration example, the vulnerability analyst has the option to change from Privileges Required (PR) —None (N) to Modified Privileges Required (MPR) — Low (L) or High (H) based on analysis. If, for example, an exploit on an Apache server usually would require no permissions on the system but the server was isolated in a management network, it may be reasonable to update Modified Privileges Required (MPR) to Low (L). If there are greater controls, such as the web server being placed in a high security network that is available only through administrative permissions and controls like VPN, the analyst may find it reasonable to update Modified Privileges Required (MPR) to High (H). It is up to analysts to use their best judgement. The CVSS v4.0 Specification describes Privileges Required as:

“This metric describes the level of privileges an attacker must possess prior to successfully exploiting the vulnerability. The method by which the attacker obtains privileged credentials prior to the attack (e.g., free trial accounts), is outside the scope of this metric”.

Figure 7 - Combining System and Network Architecture to Reduce Vulnerability Severity



In this second example, there is less overall system configuration required but the permissions required to access the vulnerable system are increased. In this configuration, only administrators are authorized to access the server through a remote access protocol such as RDP or VNC. In addition, the web server is configured to listen only on localhost or 127.0.0.1. In this scenario, it is reasonable to update Modified Privileges Required (MPR) to Low (L) or High (H) based on risk tolerance. Additionally, in this specific example, because the web server is accessible only through RDP, the Modified Attack Vector (MAV) could be updated to Local (L) and Modified Privileges Required can be updated with the desired value.

Version History

Date	Ver	Description
2026-01-06	v1.0	Initial Publication
2026-01-16	v1.01	Formatting and additional clarifications.