CVSS v3 – Preview 1: Base, Temporal, and Environmental Metrics
FIRST.ORG
June 2014

Introduction

In this first preview of the Common Vulnerability Scoring System, version 3, the CVSS Special Interest Group (CVSS-SIG) has provided the descriptions and values for the CVSS v3 Metrics, as well as the vector string notation to represent a vulnerability's CVSS v3 score in an abbreviated format.

Upon release, it is our intention that recipients of this guide would begin to produce CVSS v3 scores alongside whatever other scores they are using today (CVSS v2, or other scores for vulnerabilities). When it is time to release the completed CVSS v3 formula, organizations that have stored scores produced via this CVSS v3 Preview will be able to use the stored scores to generate CVSS v3 numeric scores.

The CVSS-SIG hopes that this preview will give additional lead time to incident response teams, analysts, and those doing vulnerability rating and classification with CVSS or similar systems. Because rating and classification typically is the more time-intensive activity, we encourage teams to start early and produce scores for storage as soon as possible. This will give incident responders and analysts additional time to practice CVSS v3 scoring, and to help ease their transition into CVSS v3.

WHILE THE CVSS-SIG HOPES THAT MANY WILL TAKE ADVANTAGE OF THIS PREVIEW TO HELP THEMSELVES BECOME ACQUAINTED WITH THE STANDARD, WE ASK THAT NOONE USE THIS DOCUMENT TO GIVE OFFICIAL PUBLIC CVSS V3 METRICS OR VECTOR STRINGS TO VULNERABILITIES.

The CVSS-SIG does not want to discourage any public commentary regarding CVSS v3 Preview metrics or vector strings, but we feel the community would be disadvantaged by anyone assigning CVSS v3 metrics in any official, public manner (such as in a product security advisory, as the results of a vulnerability scan, in a vulnerability database, etc.) before the final specification is released.

Seth Hanford
Chair, CVSS-SIG
seth@first.org
@SethHanford

Please submit general comments on this Preview to:
cvss-v3-comments@first.org

Base Metrics

Attack Vector (AV)

This metric reflects the context in which the vulnerability exploitation occurs. The values for this metric are listed in the table below. The more remote an attacker can be to the target, the greater the vulnerability score. The possible values for this metric are listed in Table 1. This rationale is that, in general, the number of potential attackers for a remotely exploitable vulnerability would be much larger than that for an attack requiring local access.

Metric Value	Description
Network (N)	A vulnerability exploitable with network access means the vulnerable component is bound to the network stack and the attacker's path to the vulnerable system is at the network layer. Such a vulnerability is often termed "remotely exploitable". An example of a network attack is an RPC buffer overflow.
Adjacent Network (A)	A vulnerability exploitable with adjacent network access means the vulnerable component is bound to the network stack and the attacker's path to the vulnerable system is at the data link layer. Examples include local IP subnet, Bluetooth, IEEE 802.11, and local Ethernet segment. For instance, a vulnerability in this category would be a bug in application software that processes Ethernet frames.
Local (L)	A vulnerability exploitable with local access means the vulnerable component is not bound to the network stack and the attacker's path to the vulnerable component is via read / write / execute capabilities. If the attacker has the necessary Privileges Required to interact with the vulnerable component, they may be logged in locally; otherwise, they may deliver an exploit to a user and rely on User Interaction. An example of a locally exploitable vulnerability is a flaw in a word processing application when processing a malformed document.
Physical (P)	A vulnerability exploitable with physical access requires the ability to physically touch or manipulate a vulnerable component. Physical interaction may be

	brief (evil maid attack) or persistent. Example of such an attack is cold boot attack [1] which allows an attacker to get access to disk encryption keys after gaining physical access to the system, or peripheral attacks such as Firewire/USB Direct Memory Access attacks.

Table 1

Attack Complexity (AC)

This metric describes the conditions beyond the attacker's control that must occur in order to place the system in a vulnerable state, this also excludes any user interaction requirements. The possible values for this metric are listed in Table 2.

Metric Value	New Description
High (H)	A successful attack depends on conditions outside the attacker's control that may be difficult to circumvent or satisfy. This may require the attacker to gather some knowledge about the specific target. Examples of knowledge to be gathered are: target configuration settings, sequence numbers, shared secrets, etc. These attack conditions are typically unique to individual target environments and may require significant resources to produce unreliable success rates. A resourceful and motivated attacker might circumvent these conditions with routine methods, but represents a non-trivial level of access complexity that may limit the success of tool kit generated attacks.
Low (L)	Specialized access conditions or extenuating circumstances do not exist. An attacker can expect repeatable exploit success against a vulnerable target

Table 2

Privileges Required (PR)

This metric describes the privileges an attacker requires before successfully exploiting the vulnerability, and the potential impact they could inflict on a system after exploiting it. The possible values for this metric are listed in Table 3.

Metric Value	Description
High (H)	The attacker is authenticated with privileges that provide significant control over component resources. With these starting privileges an attacker can cause a Complete impact to one or more of: Confidentiality, Integrity, or Availability. Alternatively, an attacker with High privileges may have the ability to cause a Partial impact to sensitive resources.
Low (L)	The attacker is authenticated with privileges that provide basic, low-impact capabilities. With these starting privileges an attacker is able to cause a Partial impact to one or more

	of: Confidentiality, Integrity, or Availability. Alternatively, an attacker with Low privileges may have the ability to cause an impact only to non-sensitive resources.
None (N)	The attacker is unprivileged or unauthenticated.

Table 3

User Interaction (UI)

This metric captures the requirement for a user (other than the attacker) to participate in the successful exploit of the target information system. The possible values for this metric are listed in Table 4. This new user interaction metric will determine whether or not the vulnerability can be exploited solely at the will of the attacker, or if a user must participate by taking action.

Metric Value	Description
None (N)	The vulnerable system can be exploited without any interaction from any user.
Required (R)	Successful exploitation of this vulnerability requires a user to take one or more actions that may or may not be expected in a scenario involving no exploitation, or a scenario involving content provided by a seemingly trustworthy source.

Table 4

Scope (S)

Components run within a scope that authorizes the actions they can perform and the resources they can access. An example of an authorization scope is the user list and the privileges granted to users of an operating system. A separate authorization scope could be contained within a database application that runs on the operating system. The possible values for this metric are listed in Table 5.
If a successful exploit only impacts resources within the scope of the vulnerable component, then Scope is Unchanged. If a successful exploit impacts resources outside the scope of the vulnerable component, then Scope is Changed.

Exploitability Subscore Metrics (Attack Vector, Attack Complexity, Privileges Required, User Interaction) are measured relative to the vulnerable component's Scope, not to any potentially Changed Scope. Impact Subscore Metrics (Confidentiality, Integrity, Availability) are scored relative to the Scope (Changed or Unchanged).

Metric Value	Description
Unchanged (U)	The attacker attacks and impacts the environment that authorizes actions taken by the vulnerable component. Impact scored relative to the Unchanged Scope.
Changed (C)	The attacker attacks the vulnerable component and has an impact

	to its environment. Because of this impact, there is a direct impact to another scope. Impact scored relative to the Changed Scope.

Table 5

Confidentiality Impact (C)

This metric measures the impact to confidentiality of a successfully exploited vulnerability. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones. The possible values for this metric are listed in Table 6. Increased confidentiality impact increases the vulnerability score.

Metric Value	Description
None (N)	There is no impact to confidentiality within the affected scope.
Low (L)	There is informational disclosure or a bypass of access controls. Access to some restricted information is obtained, but the attacker does not have control over what is obtained, or the scope of the loss is constrained. The information disclosure does not have a direct, serious impact on the affected scope.
High (H)	There is total information disclosure, resulting in all resources in the affected scope being divulged to the attacker. Alternatively, access to only some restricted information is obtained, but the disclosed information presents a direct, serious impact to the affected scope (e.g. the attacker can read the administrator's password, or private keys in memory are disclosed to the attacker).

Table 6

Integrity Impact (I)

This metric measures the impact to integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and guaranteed veracity of information. The possible values for this metric are listed in Table 7. Increased integrity impact increases the vulnerability score.

Metric Value	Description
None (N)	There is no impact to integrity within the affected scope.
Low (L)	Modification of data is possible, but the attacker does not have control over the end result of a modification, or the scope of modification is constrained. The data

	modification does not have a direct, serious impact on the affected scope.
High (H)	There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised. The attacker is able to modify any files on the target system.

Table 7

Availability Impact (A)

This metric measures the impact to availability of a successfully exploited vulnerability. Availability refers to the accessibility of information resources. Attacks that consume network bandwidth, processor cycles, or disk space all impact the availability of a system. The possible values for this metric are listed in Table 8. Increased availability impact increases the vulnerability score.

Metric Value	Description
None (N)	There is no impact to availability within the affected scope.
Low (L)	There is reduced performance or interruptions in resource availability. The attacker does not have the ability to completely deny service to legitimate users, even through repeated exploitation of the vulnerability. The resources in the affected scope are either partially available all of the time, or fully available only some of the time, but the overall there is no direct, serious impact to the affected scope.
High (H)	There is total loss of availability, resulting in the attacker being able to fully deny access to resources in the affected scope; this loss is either sustained (while the attacker continues to deliver the attack) or persistent (the condition persists even after the attack has completed). Alternatively, the attacker has the ability to deny some availability, but the loss of availability presents a direct, serious impact to the affected scope (e.g. the attacker cannot disrupt existing connections, but can prevent new connections; the attacker can repeatedly exploit a vulnerability that, in each instance of a successful attack, leaks a only small amount of memory, but after repeated exploitation causes a service to become completely unavailable).

Table 8

Temporal Metrics

Exploitability (E)

This metric measures the current state of exploit techniques or code availability. Public availability of easy-to-use exploit code increases the number of potential attackers by including those who are unskilled, thereby increasing the severity of the vulnerability.

Initially, real-world exploitation may only be theoretical. Publication of proof of concept code, functional exploit code, or sufficient technical details necessary to exploit the vulnerability may follow. Furthermore, the exploit code available may progress from a proof-of-concept demonstration to exploit code that is successful in exploiting the vulnerability consistently. In severe cases, it may be delivered as the payload of a network-based worm or virus. The possible values for this metric are listed in Table 9. The more easily a vulnerability can be exploited, the higher the vulnerability score.

Metric Value	Description
Unproven (U)	No exploit code is available, or an exploit is entirely theoretical
Proof-of-Concept (P)	Proof-of-concept exploit code or an attack demonstration that is not practical for most systems is available. The code or technique is not functional in all situations and may require substantial modification by a skilled attacker.
Functional (F)	Functional exploit code is available. The code works in most situations where the vulnerability exists.
High (H)	Either the vulnerability is exploitable by functional mobile autonomous code, or no exploit is required (manual trigger) and details are widely available. The code works in every situation, or is actively being delivered via a mobile autonomous agent (such as a worm or virus).
Not Defined (X)	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

Table 9

Remediation Level (RL)

The remediation level of a vulnerability is an important factor for prioritization. The typical vulnerability is unpatched when initially published. Workarounds or hotfixes may offer interim remediation until an official patch or upgrade is issued. Each of these respective stages adjusts the temporal score downwards, reflecting the decreasing urgency as remediation becomes final. The possible values for this

metric are listed in Table 10. The less official and permanent a fix, the higher the vulnerability score is.

Metric Value	Description
Official Fix (O)	A complete vendor solution is available. Either the vendor has issued an official patch, or an upgrade is available.
Temporary Fix (T)	There is an official but temporary fix available. This includes instances where the vendor issues a temporary hotfix, tool, or workaround.
Workaround (W)	There is an unofficial, non-vendor solution available. In some cases, users of the affected technology will create a patch of their own or provide steps to work around or otherwise mitigate the vulnerability.
Unavailable (U)	There is either no solution available or it is impossible to apply.
Not Defined (X)	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

Table 10

Report Confidence (RC)

This metric measures the degree of confidence in the existence of the vulnerability and the credibility of the known technical details. Sometimes, only the existence of vulnerabilities are publicized, but without specific details. For example, an impact may be recognized as undesirable, but the root cause may not be known. The vulnerability may later be corroborated by research which suggests where the vulnerability may lie, though the research may not be certain. Finally, a vulnerability may be confirmed through acknowledgement by the author or vendor of the affected technology. The urgency of a vulnerability is higher when a vulnerability is known to exist with certainty. This metric also suggests the level of technical knowledge available to would-be attackers. The possible values for this metric are listed in Table 11. The more a vulnerability is validated by the vendor or other reputable sources, the higher the score.

Metric Value	Description
Unknown [U]	There are reports of impacts that indicate a vulnerability is present. The reports indicate that the cause of the vulnerability is unknown, or reports may differ on the cause or impacts of the vulnerability. Reporters are uncertain of the true nature of the vulnerability, and there is little confidence in the validity of the reports or whether a static Base Score can be applied given the differences described. An example is a bug

	report which notes that an intermittent but non-reproducible crash occurs, with evidence of memory corruption suggesting that denial of service, or possible more serious impacts, may result.
Reasonable (R)	Significant details are published, but researchers either do not have full confidence in the root cause, or do not have access to source code to fully confirm all of the interactions that may lead to the result. Reasonable confidence exists, however, that the bug is reproducible and at least one impact is able to be verified (Proof-of-concept exploits may provide this). An example is a detailed write-up of research into a vulnerability with an explanation (possibly obfuscated or "left as an exercise to the reader") that gives assurances on how to reproduce the results.
Confirmed (C)	Detailed reports exist, or functional reproduction is possible (functional exploits may provide this). Source code is available to independently verify the assertions of the research, or the author or vendor of the affected code has confirmed the presence of the vulnerability.
Not Defined (X)	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

Table 11

Environmental Metrics

Security Requirements (CR, IR, AR)

These metrics enable the analyst to customize the CVSS score depending on the importance of the affected IT asset to a user's organization, measured in terms of confidentiality, integrity, and availability, That is, if an IT asset supports a business function for which availability is most important, the analyst can assign a greater value to availability, relative to confidentiality and integrity. Each security requirement has three possible values: "low," "medium," or "high."

The full effect on the environmental score is determined by the corresponding base impact metrics. That is, these metrics modify the environmental score by reweighting the (base) confidentiality, integrity, and availability impact metrics. For example, the confidentiality impact (C) metric has increased weight if the confidentiality requirement (CR) is "high." Likewise, the confidentiality impact metric has decreased weight if the confidentiality requirement is "low." The confidentiality impact metric weighting is neutral if the confidentiality requirement is "medium." This same logic is applied to the integrity and availability requirements.

Note that the confidentiality requirement will not affect the environmental score if the (base) confidentiality impact is set to "none." Also, increasing the confidentiality requirement from "medium" to "high" will not change the environmental score when the (base) impact metrics are set to "complete." This is because the impact sub score (part of the base score that calculates impact) is already at a maximum value of 10.

The possible values for the security requirements are listed in Table 12. For brevity, the same table is used for all three metrics. The greater the security requirement, the higher the score (remember that "medium" is considered the default). These metrics will modify the score as much as plus or minus 2.5.

Metric Value	Description
Low (L)	Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
Medium (M)	Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
High (H)	Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or

	individuals associated with the organization (e.g., employees, customers).
Not Defined (X)	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

Table 12

Mitigated Base Metrics

These metrics enable the analyst to adjust the Base metrics according to mitigations that exist within the analyst's environment. That is, if an environment has made general changes for the affected software that differs in a way which would affect its Exploitability, Scope, or Impact, then the environment can reflect this via an appropriately-modified, or "Mitigated", environmental score.

The full effect on the environmental score is determined by the corresponding base metrics. That is, these metrics modify the environmental score by reassigning the (base) metrics values, prior to applying the (environmental) Security Requirements. For example, the default configuration for a vulnerable component may be to run a listening service as "root", for which compromises might grant an attacker Confidentiality, Integrity, and Availability impacts that are all High. Yet, in the analyst's environment, that same listening service might be running with reduced privileges; in that case, the Mitigated Confidentiality, Mitigated Integrity, and Mitigated Availability might each be set to Low.

For brevity, only the names of the Mitigated Base Metrics are mentioned. Each mitigated environmental metric has the same values as its corresponding Base metric, plus a "Not Defined" value.

Note: While the intent of this metric is to define the mitigations in place for a given environment, it is acceptable to use the Mitigated metrics to describe modifications that in fact increase the Base score. For example, the default configuration of a component may be to require High privileges (PR: High) in order to access a particular function, but in the analyst's environment, there may be No privileges required (PR: None). The analyst can set MPR: None to reflect this more serious condition for their environment.

Mitigated Base Metric	**Corresponding Values**
Mitigated Attack Vector (MAV)	The same values as the corresponding Base Metric (see Base Metrics above), as well as "Not Defined" (the default)
Mitigated Attack Complexity (MAC)	
Mitigated Privileges Required (MPR)	
Mitigated User Interaction (MUI)	
Mitigated Scope (MS)	
Mitigated Confidentiality (MC)	
Mitigated Integrity (MI)	
Mitigated Availability (MA)	

Table 13

Vector String

Each metric in the vector consists of the abbreviated metric name, followed by a ":" (colon), then the abbreviated metric value. The vector lists these metrics in a predetermined order, using the "/" (slash) character to separate the metrics. If a temporal or environmental metric is not to be used, it is given a value of "X" (not defined). The base, temporal, and environmental vectors are shown below in Table 14.

Metric Group	Vector
Base	AV:[N,A,L,P]/AC:[L,H]/PR:[N,L,H]/UI:[N,R]/S:[U,C]/C:[H,L,N]/I:[H,L,N]/A:[H,L,N]
Temporal	E:[H,F,P,U,X]/RL:[U,W,T,O,X]/RC:[U,R,C,X]
Environmental	CR:[L,M,H,X]/IR:[L,M,H,X]/AR:[L,M,H,X]/MAV:[N,A,L,P,X]/MAC:[L,H,X]/MPR:[N,L,H,X]/MUI:[N,R,X]/MS:[U,C,X]/MC:[H,L,N,X]/MI:[H,L,N,X]/MA:[H,L,N,X]

Table 14

For example, a vulnerability with base metric values of "Attack Vector: Network, Attack Complexity: Low, Privileges Required: High, User Interaction: None, Scope: Unchanged, Confidentiality: High, Integrity: High, Availability: High" would have the following base vector: **"**AV:N/AC:L/PR:H/UI:N/S:U/C:C/C:I/A:C**"**

Acknowledgements

This preview is the first in a series that will culminate with the complete release of CVSS v3 for public comment. This document would not have been possible without the tireless efforts of the CVSS-SIG, and our many supporters in the community, in our workplaces, and at FIRST. A proper list of acknowledgements and thanks will be produced for the final release of CVSS v3.