A Complete Guide to The Common Vulnerability Scoring System (CVSS)

Mike Schiffman, Cisco CIAG

Page last updated: Tue Jun 7 13:58:34 PDT 2005

Abstract

To date, a number of commercial computer security vendors and not-for-profit organizations have developed, promoted, and implemented systems to rank information system vulnerabilities. Unfortunately, there is no cohesion or interoperability among those systems and they are limited in scope as to what they cover. This document proposes an open and universal vulnerability scoring system to address and solve these shortcomings, with the ultimate goal of promoting a common language to discuss vulnerability severity and impact.

Contents

Introduction

The ability to score information system vulnerabilities is extremely important to the professional computing world. It provides the foundation for a standard process for stakeholders to prioritize their actions and respond to the threat vulnerabilities present. Prior to this document several competing, incompatible, and closed vulnerability scoring systems were the only available solutions [1] [2] [3]. This led to a lack of a unified standard in the space and resulted in much confusion when a single vulnerability would be released and would be scored differently among the different systems (sometimes resultant scores would be inversely correlated which made no sense). This document describes The Common Vulnerability Scoring System (CVSS), an open standard for scoring vulnerabilities.

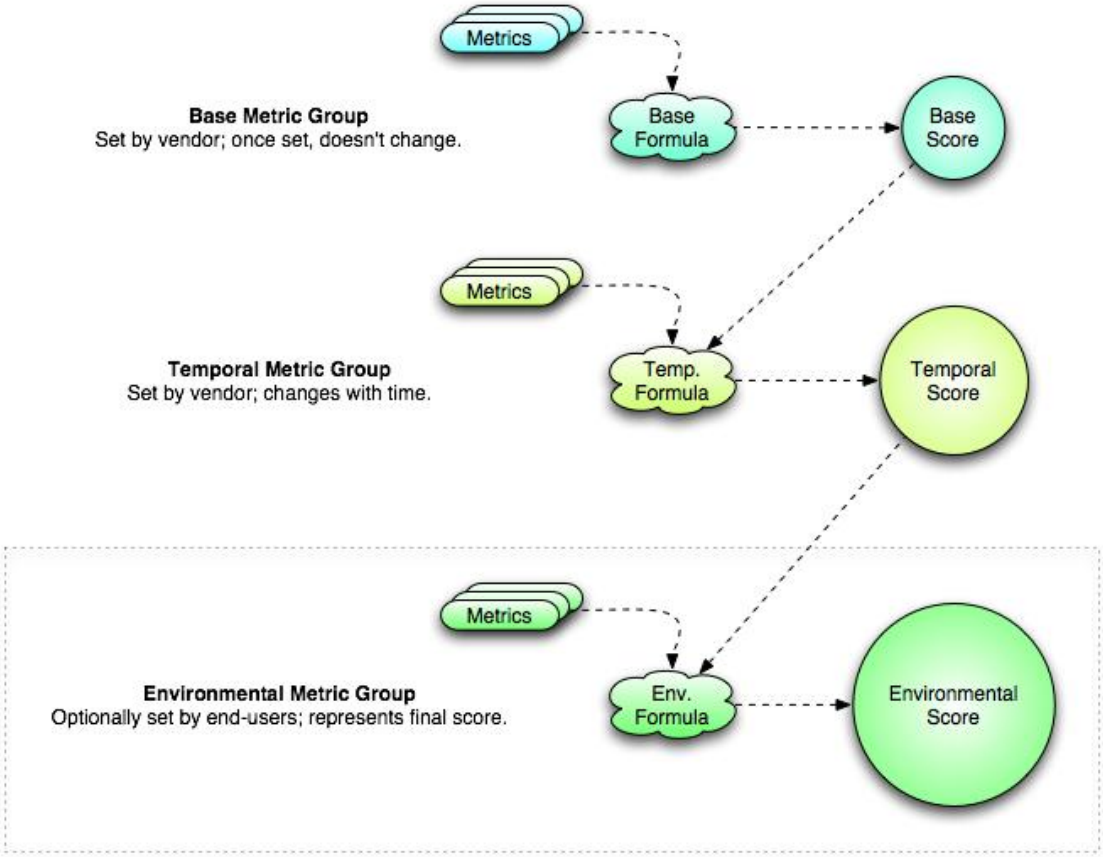
CVSS is designed to rank information system vulnerabilities and provide the end user with a composite score representing the overall severity and risk the vulnerability presents. Using CVSS, security professionals, executives, and end-users will have a common language with which to discuss security vulnerability severity.

Vulnerability Definition

A vulnerability is defined as a bug, flaw, behavior, output, outcome or event within an application, system, device, or service that could lead to an implicit or explicit failure of confidentiality, integrity, or availability.

CVSS Overview

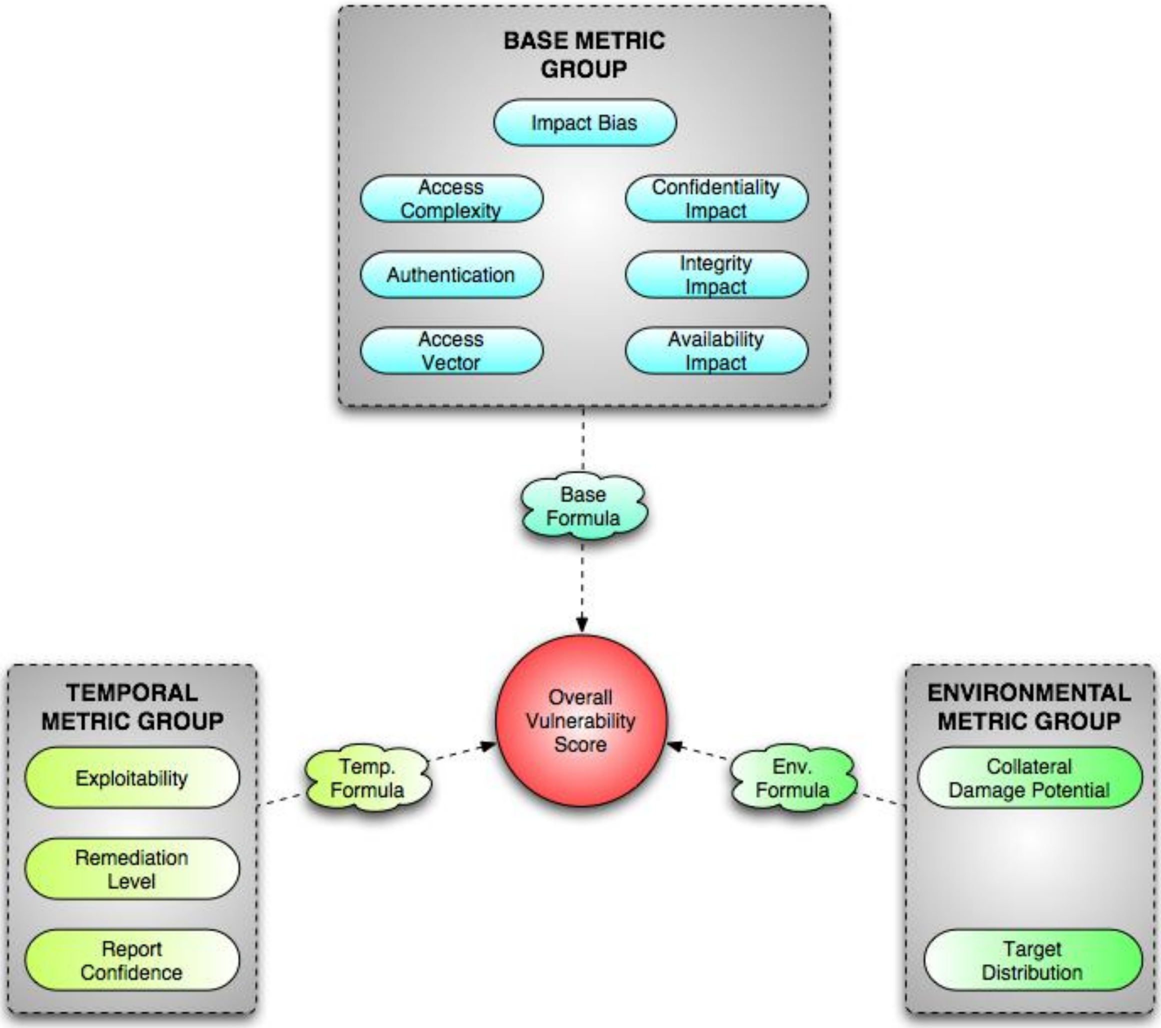
CVSS, as shown below, is structured as a modular system with three distinct groups. Each of these groups clusters together related qualities that capture certain characteristics of a vulnerability. Each of these qualities or "metrics" has a specific way of being measured and each group has a unique formula for combining and weighing each metric. While complex under the hood, CVSS can be implemented to present a very simple interface to users (all numeric translation and formula computation is done behind the scenes to the end-user).


Metrics
Base Metric Group
Set by vendor; once set, doesn't change.
Base Formula
Base Score
Metrics
Temporal Metric Group
Set by vendor; changes with time.
Temp. Formula
Temporal Score
Metrics
Environmental Metric Group
Optionally set by end-users; represents final score.
Env. Formula
Environmental Score

1.0 Vulnerability Metrics

A metric is a constituent component or characteristic of a vulnerability that can be quantitatively or qualitatively measured. These atomic values are clustered together in three separate areas: a base group, a temporal group, and an environmental group. The base group contains all of the qualities that are intrinsic and fundamental to any given vulnerability that do not change over time or in different environments. The temporal group contains the characteristics of a vulnerability that are time-dependent and change as the vulnerability ages. Finally, the environmental group contains the characteristics of vulnerabilities that are tied to implementation and environment. The final adjusted score represents the threat a vulnerability presents at a particular point in time for a specific environmental condition. The metric groups are shown below.

The authors recognize that many other metrics could be included in CVSS. They also realize that no one scoring system will fit everyone's need perfectly. The particular constituent metrics used in CVSS were identified as the best compromise between completeness, ease-of-use and accuracy. They represent the cumulative experience of the authors as well as extensive testing of real-world vulnerabilities in end-user environments.



1.1 Base Metrics

Once discovered, analyzed, and catalogued, assuming the initial information is complete and correct, there are certain aspects of a vulnerability that do not change. These core characteristics will not change over time, nor will they change in different target environments; for all intents and purposes, once set, they are immutable. The base metric group captures these unchanging qualities, which are access to and impact on the target.

The three access metrics capture how a vulnerable information system may be reached. Consideration is given to not only how a target may be accessed in order to exploit the vulnerability, but also whether or not there are mitigating factors that complicate the process.

The three impact metrics measure how a vulnerability will affect a given information system. A widely accepted view of information systems security breaks down the goals of securing a system into three properties: confidentiality, integrity and availability. The impact of a vulnerability on affected systems can be defined as a combination of losses to varying degrees of each of these properties. Vulnerability impact needs to be expressed in terms of the confidentiality, integrity, and availability properties: from negligible to total losses for each of the three properties as well as combinations of losses, for example, the partial loss of integrity and the partial loss of confidentiality due to a vulnerability in a logging mechanism.

1.1.1 Access Vector

This metric measures whether or not the vulnerability is exploitable locally or remotely. A vulnerability exploitable with only local access typically means the attacker must have either physical or authenticated login access to the target system, often either a walk-in scenario or a local account on a target computer system. Remote access typically means the attacker can trigger the vulnerability from across a network, either from across a wireless network or from across the Internet.

A vulnerability that is exploitable remotely is considered to be a higher risk than one that is only exploitable locally, since the complexity of access is lower, which typically increases the pool of would-be attackers. Therefore, if a vulnerability is only exploitable locally, its resulting CVSS score will be lower than if it was exploitable remotely.

When a vulnerability can be exploited both locally and remotely, the remote value should be chosen.

1.1.1.1 Access Vector Scoring Evaluation

Local: The vulnerability is only exploitable locally (i.e., it requires physical access or authenticated login to the target system)

Remote: The vulnerability is exploitable remotely

1.1.2 Access Complexity

This metric measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system. In most cases, once the target system is contacted, exploit of the vulnerability is academic. The traditional example is a simple remotely exploitable buffer overflow in an Internet server program that runs continuously. Once the target system is located, there is no additional complexity in accessing the target - an attacker presumably can exploit at the target at will. Other vulnerabilities require specialized access considerations in order to become exploitable. In other words, once the system is accessed, there may be additional barriers to exploitation. An example in this case would be a vulnerability in an email program that is only exploitable when the user downloads and opens a tainted attachment.

1.1.2.1 Access Complexity Scoring Evaluation

High: Specialized access conditions exist; for example: the system is exploitable during specific windows of time (a race condition), the system is exploitable under specific circumstances (nondefault configurations), or the system is exploitable with victim interaction (vulnerability exploitable only if user opens e-mail)

Low: Specialized access conditions or extenuating circumstances do not exist; the system is always exploitable

1.1.3 Authentication

This metric measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability. The specific type of and mechanism for authentication is not important because it is considered that authentication in any form will add significant complexity to the exploitation process. Additionally, authentication is an either-or consideration. Attackers without valid credentials should not be able to access the target in order to exploit the vulnerability. Therefore, this metric's values are mutually exclusive; only one of them can be true. If authentication of some sort is required, the final CVSS score will be considerably lower than if it were not required.

It is important to note that the Authentication metric is distinct from the Access Vector metric. The requirement for authentication represented by this metric is considered once the system has already been accessed. Specifically, in the case of locally exploitable vulnerabilities, this metric should only be set to "required" if authentication is needed beyond what is required for a user to login to the system (and thus becoming "local"). The Access Vector metric (local or remote) reduces the score if the vulnerability is flagged as locally exploitable, thus taking into consideration the prerequisite authentication.

An example of a locally exploitable vulnerability that requires authentication is one affecting a database engine listening on a Unix domain socket or some other non-network interface. If the user must authenticate as a valid database user to exploit the vulnerability, then this metric should be set to "required" resulting in a lower CVSS score.

1.1.3.1 Authentication Scoring Evaluation

Required: Authentication is required to access and exploit the vulnerability

Not Required: Authentication is not required to access or exploit the vulnerability

1.1.4 Confidentiality Impact

This metric measures the impact on confidentiality of a successful exploit of the vulnerability on the target system. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by or disclosure to unauthorized ones. Confidentiality is usually preserved by a system's information protection mechanisms: cryptography, data compartmentalization, identification and authentication, etc.. Compromise of a system's information protection mechanism can negatively impact confidentiality.

1.1.4.1 Confidentiality Impact Scoring Evaluation

None: No impact on confidentiality.

Partial: There is considerable informational disclosure. Access to critical system files is possible. There is a loss of important information, but the attacker doesn't have control over what is obtainable or the scope of the loss is constrained. For example, a partial confidentiality impact would indicate a vulnerability that divulges bits in an encryption key or password hash information. Or, privileges are altered by one user to gain access to files of another user.

Complete: A total compromise of critical system information. A complete loss of system protection resulting in all critical system files being revealed. The attacker has sovereign control to read all of the system's data (memory, files, etc).

1.1.5 Integrity Impact

This metric measures the impact on integrity a successful exploit of the vulnerability will have on the target system. Integrity refers to the trustworthiness and guaranteed veracity of information. Integrity measures are meant to protect data from unauthorized modification. When the integrity of a system is sound, it is fully proof from unauthorized modification of its contents.

1.1.5.1 Integrity Impact Scoring Evaluation

None: No impact on integrity.

Partial: Considerable breach in integrity. Modification of critical system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is constrained. For example, key system or program files may be overwritten or modified, but at random or in a limited context or scope.

Complete: A total compromise of system integrity. There is a complete loss of system protection resulting in the entire system being compromised. The attacker has sovereign control to modify any system files.

1.1.6 Availability Impact

This metric measures the impact on availability a successful exploit of the vulnerability will have on the target system. Availability refers to the accessibility of information resources. Almost exclusive to this domain are denial-of-service vulnerabilities. Attacks that compromise network bandwidth, processor cycles, disk space, or administrator time all impact the availability of a system.

1.1.6.1 Availability Impact Scoring Evaluation

None: No impact on availability.

Partial: Considerable lag in or interruptions in resource availability. For example, a network-based flood attack that reduces available bandwidth to a web server farm to such an extent that only a small number of connections successfully complete.

Complete: Total shutdown of the affected resource. The attacker can render the resource completely unavailable.

1.1.7 Impact Bias

This metric allows a score to convey greater weighting to one of three impact metrics over the other two. An important consideration of the impact metrics is that the importance of the individual properties they measure can vary among systems. For example, a vulnerability affecting the confidentiality of an encrypted file system is far more severe than one affecting its availability. The Impact Bias metric will have no effect if the three impact metrics are all assigned the same value.

1.1.7.1 Impact Bias Scoring Evaluation

Normal: Confidentiality Impact, Integrity Impact, and Availability Impact are all assigned the same weight.

Confidentiality: Confidentiality impact is assigned greater weight than Integrity Impact or Availability Impact.

Integrity: Integrity Impact is assigned greater weight than Confidentiality Impact or Availability Impact.

Availability: Availability Impact is assigned greater weight than Confidentiality Impact or Integrity Impact.

1.2 Temporal Metrics

During the lifecycle of a vulnerability, certain events may occur which affect the urgency of the threat posed by the vulnerability. Three such factors that the CVSS attempts to capture are: confirmation of the vulnerability or its technical details, the remediation status of the vulnerability and availability of exploit code or exploit techniques. Each of these dynamic factors are important in adjusting the urgency (i.e. priority) of a vulnerability over time.

1.2.1 Exploitability

This metric attempts to measure the current state of exploit technique or code availability and suggests a likelihood of exploitation. It is assumed that there are far more unskilled attackers than there are attackers who are skilled enough to investigate vulnerabilities and create their own functional exploit code.

Public availability of easy-to-use exploit code increases the pool of potential attackers by including those who are unskilled, thereby increasing the urgency of the vulnerability.

Initially, real world exploitation may only be theoretical. Publication of proof of concept code, functional exploit code or sufficient technical details 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 an Internet-based worm or virus. This metric attempts to include these stages in the temporal score.

1.2.1.1 Exploitability Scoring Evaluation

Unproven: No exploit code is yet available or an exploit method is entirely theoretical.

Proof of Concept: Proof of concept exploit code or an attack demonstration that is not practically applicable to deployed systems is available. The code or technique is not functional in all situations and may require substantial hand tuning by a skilled attacker for use against deployed systems.

Functional: Functional exploit code is available. The code works in most situations where the vulnerability is exploitable.

High: Either the vulnerability is exploitable by functional mobile autonomous code or no exploit is required (manual trigger) and the details for the manual technique are widely available. The code works in every situation where the vulnerability is exploitable and/or is actively being delivered via a mobile autonomous agent (a worm or virus).

1.2.2 Remediation Level

The remediation status of a vulnerability is an important factor for prioritization. The typical vulnerability is unpatched when initially published. Workarounds or hotfixes submitted by the vendor or users 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.

1.2.2.1 Remediation Level Scoring Evaluation

Official Fix: A complete vendor solution is available. Either the vendor has issued the final, official patch which eliminates the vulnerability or an upgrade that is not vulnerable is available.

Temporary Fix: There is an official but temporary fix available. This includes instances where the vendor issues a temporary hotfix, tool or official workaround.

Workaround: 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 against the vulnerability. When it is generally accepted that these unofficial fixes are adequate in plugging the hole for the mean time and no official remediation is available, this value can be set.

Unavailable: There is either no solution available or it is impossible to apply.

1.2.3 Report Confidence

This metric measures the degree of confidence in the existence of the vulnerability and the credibility of the known technical details. In many cases, vulnerabilities are initially reported by individual users either directly or indirectly through symptoms that suggest the existence of the vulnerability. The vulnerability may later be corroborated and then confirmed through acknowledgement by the author or vendor of the affected technology. The urgency of a vulnerability is the higher when a vulnerability is known to exist with certainty. This metric also suggests the level of technical knowledge available to would-be attackers.

1.2.3.1 Report Confidence Scoring Evaluation

Unconfirmed: A single unconfirmed source or possibly several conflicting reports. There is little confidence in the validity of the report. For example, a rumor that surfaces from the hacker underground.

Uncorroborated: Multiple non-official sources; possibily including independent security companies or research organizations. At this point there may be conflicting technical details or some other lingering ambiguity.

Confirmed: Vendor or author of the affected technology has acknowledged that the vulnerability exists. This value may also be set when existence of a vulnerability is confirmed with absolute confidence through some other event, such as publication of functional proof of concept exploit code or widespread exploitation.

1.3 Environmental Metrics

Different environments can have an immense bearing on the risk that a vulnerability poses to an organization and its stakeholders. The CVSS environmental metrics group captures characteristics of vulnerabilities that are tied to implementation and environment.

1.3.1 Collateral Damage Potential

This metric measures the potential for a loss in physical equipment, property damage or loss of life or limb.

1.3.1.1 Collateral Damage Potential Scoring Evaluation

None: There is no potential for physical or property damage.

Low: A successful exploit of this vulnerability may result in light physical or property damage or loss. The system itself may be damaged or destroyed.

Medium: A successful exploit of this vulnerability may result in significant physical or property damage or loss.

High: A successful exploit of this vulnerability may result in catastrophic physical or property damage and loss. The range of effect may be over a wide area.

1.3.2 Target Distribution

This metric measures the relative size of the field of target systems susceptible to the vulnerability. It is meant as an environment-specific indicator in order to approximate the percentage of systems within the environment that could be affected by the vulnerability.

1.3.2.1 Target Distribution Scoring Evaluation

None: No target systems exist, or targets are so highly specialized that they only exist in a laboratory setting. Effectively 0% of the environment is at risk.

Low: Targets exist inside the environment, but on a small scale. Between 1% - 15% of the total environment is at risk.

Medium: Targets exist inside the environment, but on a medium scale. Between 16% - 49% of the total environment is at risk.

High: Targets exist inside the environment on a considerable scale. Between 50% - 100% of the total environment is considered at risk.

2.0 Scoring

Scoring is the process of combining the values of each metric from each group into a final composite score that represents the overall risk of a given vulnerability. The CVSS scoring process is broken into three phases, one for each metric group. Scoring begins with the base metric group and then temporal and environmental scores are computed to produce a final score.

Each metric group has a different formula that combines its constituent metrics. The base metric group captures the fundamental constituent qualities of a given vulnerability and therefore provides the foundation for the final score. The temporal and environmental metric groups serve to increase or decrease this base score.

The formulae should operate behind the scenes of the CVSS implementation and be transparent to the end-user.

2.1 Base Metric Scoring

The base score provides the foundation for the overall vulnerability score. The most significant metrics in the scoring process are the three impact metrics. These metrics dictate the overall effect the vulnerability will have on target systems and therefore have the strongest bearing on the final score.

```
Base Metric Formula

AccessVector     = case AccessVector of
                        local:           0.7
                        remote:           1.0

AccessComplexity = case AccessComplexity of
                        high:             0.8
                        low:              1.0

Authentication   = case Authentication of
                        required:         0.6
                        not-required:     1.0

ConfImpact       = case ConfidentialityImpact of
                        none:             0
                        partial:          0.7
                        complete:         1.0

ConfImpactBias   = case ImpactBias of
                        normal:           0.333
                        confidentiality:  0.5
                        integrity:        0.25
                        availability:     0.25

IntegImpact      = case IntegrityImpact of
                        none:             0
                        partial:          0.7
                        complete:         1.0

IntegImpactBias  = case ImpactBias of
                        normal:           0.333
                        confidentiality:  0.25
                        integrity:        0.5
                        availability:     0.25

AvailImpact      = case AvailabilityImpact of
                        none:             0
                        partial:          0.7
                        complete:         1.0

AvailImpactBias  = case ImpactBias of
                        normal:           0.333
                        confidentiality:  0.25
                        integrity:        0.25
                        availability:     0.5

BaseScore = round_to_1_decimal(10 * AccessVector
                                  * AccessComplexity
                                  * Authentication
                                  * ((ConfImpact * ConfImpactBias)
                                  + (IntegImpact * IntegImpactBias)
                                  + (AvailImpact * AvailImpactBias)))
```

2.2 Temporal Metric Scoring

The temporal score adjusts the base score by including factors that may change over time. The temporal score will be less than or equal to the base score; that is, the temporal metrics serve only to reduce the base score by a maximum of 33%. This is shown at the end of the scoring section.

```
Temporal Metric Formula

Exploitability   = case Exploitability of
                        unproven:             0.85
                        proof-of-concept:     0.9
                        functional:           0.95
                        high:                 1.00

RemediationLevel = case RemediationLevel of
                        official-fix:         0.87
                        temporary-fix:        0.90
                        workaround:           0.95
                        unavailable:          1.00

ReportConfidence = case ReportConfidence of
                        unconfirmed:          0.90
                        uncorroborated:       0.95
                        confirmed:            1.00

TemporalScore = round_to_1_decimal(BaseScore * Exploitability
                                             * RemediationLevel
                                             * ReportConfidence)
```

2.3 Enviromental Metric Scoring

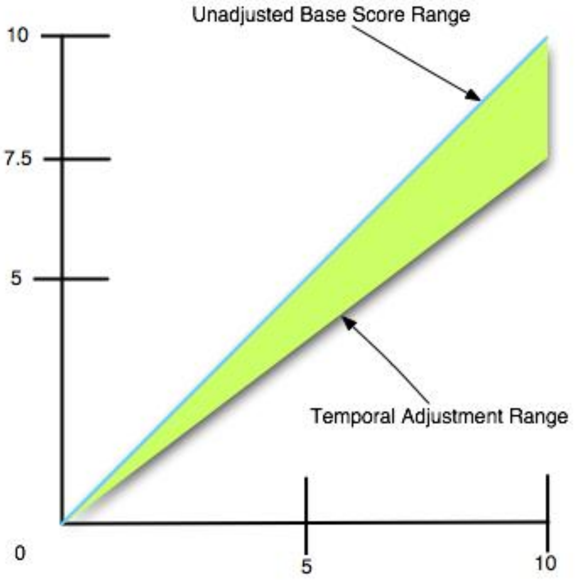
The environmental score adjusts the temporal score to account for aspects of an organization's environment. The environmental score may be higher or lower than the temporal score. While the collateral damage metric allows to increase the resulting score, the metric for target distribution only allows downwards correction, even with the possibility of a resulting environmental score of zero.

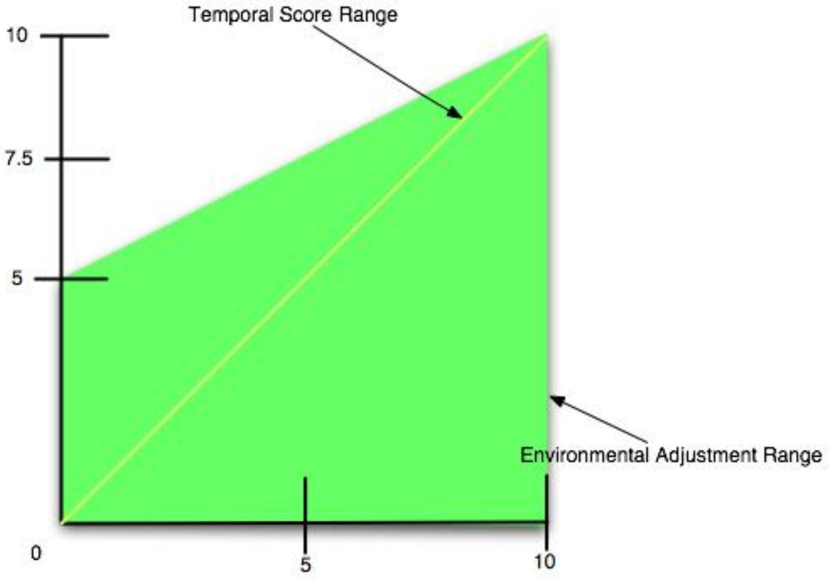
```
Environmental Metric Formula

CollateralDamagePotential = case CollateralDamagePotential of
                                 none:            0
                                 low:             0.1
                                 medium:          0.3
                                 high:            0.5

TargetDistribution        = case TargetDistribution of
                                 none:            0
                                 low:             0.25
                                 medium:          0.75
                                 high:            1.00

EnvironmentalScore = round_to_1_decimal((TemporalScore + ((10 - TemporalScore)
                                         * CollateralDamagePotential))
                                         * TargetDistribution)
```





3.0 Expected and Recommended Usage

Typical implementations of CVSS will begin with security organizations that are normally tasked with investigating and corroborating vulnerabilities as well as notifications to end-users. Often these organizations are security product companies such as vulnerability management, anti-virus and intrusion detection/prevention companies, both commercial and open-source. However, they may also be community forums or mailing lists either privately- or government-maintained.

These organizations, through either research of their own, or collaboration with other information security groups, will score the BASE and TEMPORAL metrics. It is at their discretion as to how they choose to disseminate these scores; CVSS does not attempt to define or dictate this.

The end-user, whether they are an individual, a community group, a privately- or a publicly-held enterprise would then score the ENVIRONMENTAL metrics and arrive at final score suitable to their environment.

3.1 Examples

Apache Chunked-Encoding Memory Corruption Vulnerability (CVE-2002-0392)

In June 2002, a vulnerability was discovered in the means by which the Apache web server handles requests encoded using chunked encoding. The Apache Foundation reported that a successful exploit can lead to denial of service in some cases, and in others, the execution of arbitrary code with the privileges of the web server.

Because the vulnerability can be exploited remotely, the Access Vector is "Remote". The Access Complexity is "Low" because no additional circumstances need to exist for this exploit to be successful; the attacker need only craft a proper exploit message to the apache web listener. No authentication is required to trigger the vulnerability (any Internet user can connect to the web server) and so the Authentication metric is "Not-

Required".

Given that the most likely outcome of a successful attack is denial of service, the Availability Impact is set to "Complete" and the Impact Bias is set to "Availability". In other conditions, however, the attacker would be able to execute code with the permissions of the web user, thereby altering web content and possibly viewing local user or configuration information (including connection settings and passwords to back-end databases). Therefore, Confidentiality and Integrity Impact metrics are set to "Partial". Together, these metrics result in a BASE score of 8.5.

Exploit code is known to exist and therefore Exploitability is set to "Functional". The Apache foundation has released patches for this vulnerability (available to both 1.3 and 2.0) and so Remediation Level is "Official-Fix". Naturally, report confidence is "Confirmed". These metrics adjust the BASE score to give a TEMPORAL score of 7.0.

Depending on the values for Collateral Damage Potential and Target Distribution the ENVIRONMENTAL (final) score could vary between 0.0 ("None", "None") and 8.5 ("High", "High"). The results are summarized below.

```
----------------------------------------------------
BASE METRIC                 EVALUATION         SCORE
----------------------------------------------------
Access Vector               [Remote]          (1.00)
Access Complexity           [Low]             (1.00)
Authentication              [Not-Required]    (1.00)
Confidentiality Impact      [Partial]         (0.70)
Integrity Impact            [Partial]         (0.70)
Availability Impact         [Complete]        (1.00)
Impact Bias                 [Availability]    (0.25)
----------------------------------------------------
BASE FORMULA                              BASE SCORE
----------------------------------------------------
round(10 * 1.0 * 1.0 * 1.0 * (0.7 * 0.25) +
     (0.7 * 0.25) + (1.0 * 0.5)) ==           (8.50)
----------------------------------------------------

----------------------------------------------------
TEMPORAL METRIC             EVALUATION         SCORE
----------------------------------------------------
Exploitability              [Functional]      (0.95)
Remediation Level           [Official-Fix]    (0.90)
Report Confidence           [Confirmed]       (1.00)
----------------------------------------------------
TEMPORAL FORMULA                      TEMPORAL SCORE
----------------------------------------------------
round(8.50 * 0.95 * 0.90 * 1.00) ==           (7.00)
----------------------------------------------------

----------------------------------------------------
ENVIRONMENTAL METRIC        EVALUATION         SCORE
----------------------------------------------------
Collateral Damage Potential [None - High]  {0 - 0.5}
Target Distribution         [None - High]  {0 - 1.0}
----------------------------------------------------
ENVIRONMENTAL FORMULA            ENVIRONMENTAL SCORE
----------------------------------------------------
round((7.0 + ((10 - 7.0) * {0 - 0.5})) *
     {0 - 1.00}) ==                    (0.00 - 8.50)
----------------------------------------------------
```

Microsoft Windows ASN.1 Library Integer Handling Vulnerability (CAN-2003-0818)

In September 2003, a vulnerability was discovered that targets the ASN.1 library of all Microsoft operating systems. Successful exploitation of this vulnerability results in a buffer overflow condition allowing the attacker to execute arbitrary code with administrative (system) privileges.

This is a remotely exploitable vulnerability that does not require authentication, therefore the Access Vector is "Remote" and Authentication is "Not-Required". The Access Complexity is "Low" because no additional access or specialized circumstances need exist for the exploit to be successful. Each of the Impact metrics is set to "Complete" because of the possibility of a complete system compromise. The Impact Bias is "Normal". Together, these metrics result in a maximum BASE score of 10.0.

Known exploits do exist for this vulnerability and so Exploitability is "Functional". In February 2004, Microsoft released patch MS04-007 making the Remediation Level "Official-Fix" and the Report Confidence "Confirmed". These metrics adjust the BASE score to give a TEMPORAL score of 8.3.

Depending on the values for Collateral Damage Potential and Target Distribution the ENVIRONMENTAL (final) score could vary between 0.0 ("None", "None") and 9.2 ("High", "High"). The results are summarized below.

```
----------------------------------------------------
BASE METRIC                 EVALUATION         SCORE
----------------------------------------------------
Access Vector               [Remote]          (1.00)
Access Complexity           [Low]             (1.00)
Authentication              [Not-Required]    (1.00)
Confidentiality Impact      [Complete]        (1.00)
```

```
Integrity Impact            [Complete]        (1.00)
Availability Impact         [Complete]        (1.00)
Impact Bias                 [Normal]         (0.333)
----------------------------------------------------
FORMULA                                   BASE SCORE
----------------------------------------------------
round(10 * 1.0 * 1.0 * 1.0 * (1.0 * 0.333) +
     (1.0 * 0.333) + (1.0 * 0.333)) ==        (10.0)
----------------------------------------------------

----------------------------------------------------
TEMPORAL METRIC             EVALUATION         SCORE
----------------------------------------------------
Exploitability              [Functional]      (0.95)
Remediation Level           [Official-Fix]    (0.90)
Report Confidence           [Confirmed]       (1.00)
----------------------------------------------------
FORMULA                               TEMPORAL SCORE
----------------------------------------------------
round(10.0 * 0.95 * 0.90 * 1.00) ==            (8.3)
----------------------------------------------------

----------------------------------------------------
ENVIRONMENTAL METRIC        EVALUATION         SCORE
----------------------------------------------------
Collateral Damage Potential [None - High]  {0 - 0.5}
Target Distribution         [None - High]  {0 - 1.0}
----------------------------------------------------
FORMULA                          ENVIRONMENTAL SCORE
----------------------------------------------------
round((8.3 + ((10 - 8.3) * {0 - 0.5})) *
     {0 - 1.00}) ==                    (0.00 - 9.20)
----------------------------------------------------
```

Buffer Overflow In NOD32 Antivirus Software (CVE-2003-0062)

NOD32 is an antivirus software application developed by Eset. In February 2003, a buffer overflow vulnerability was discovered in linux and unix versions prior to 1.013 that could allow local users to execute arbitrary code with the privileges of the user executing NOD32. To trigger the buffer overflow, the attacker must wait for (or coax) another user (possibly root) to scan a directory path of excessive length.

Since the vulnerability is exploitable only to a user locally logged into the system, the Access Vector is "Local". The Access Complexity is "High" because this vulnerability is not exploitable at the attacker's whim. There is an additional layer of complexity because the attacker must wait for another user to run the virus scanning software. Authentication is set to "Not-Required" because the attacker does not need to authenticate to any additional system. If an administrative user were to run the virus scan, causing the buffer overflow, then a full system compromise would be possible. Since the most harmful case must be considered, each of the three Impact metrics is set to "Complete" and the Impact Bias is "Normal". Together, these metrics result in a BASE score of 5.6.

Partial exploit code has been released and so the Exploitability metric is set to "Proof-Of-Concept". Eset has released updated software giving a Remediation Level of "Official-Fix" and Report Confidence of "Confirmed". These three metrics adjusts the BASE score to give a TEMPORAL score of 4.4.

Depending on the values for Collateral Damage Potential and Target Distribution the ENVIRONMENTAL (final) score could vary between 0.0 ("None", "None") and 7.2 ("High", "High"). The results are summarized below.

```
----------------------------------------------------
BASE METRIC                 EVALUATION         SCORE
----------------------------------------------------
Access Vector               [Local]           (0.70)
Access Complexity           [High]            (0.80)
Authentication              [Not-Required]    (1.00)
Confidentiality Impact      [Complete]        (1.00)
Integrity Impact            [Complete]        (1.00)
Availability Impact         [Complete]        (1.00)
Impact Bias                 [Normal]         (0.333)
----------------------------------------------------
FORMULA                                   BASE SCORE
----------------------------------------------------
round(10 * 0.7 * 0.8 * 1.0 * (1.0 * 0.333) +
     (1.0 * 0.333) + (1.0 * 0.333)) ==         (5.6)
----------------------------------------------------

----------------------------------------------------
TEMPORAL METRIC             EVALUATION         SCORE
----------------------------------------------------
Exploitability              [Proof-Of-Concept](0.90)
Remediation Level           [Official-Fix]    (0.90)
Report Confidence           [Confirmed]       (1.00)
----------------------------------------------------
```

```
FORMULA                               TEMPORAL SCORE
----------------------------------------------------
round(5.6 * 0.90 * 0.90 * 1.00) ==             (4.4)
----------------------------------------------------


----------------------------------------------------
ENVIRONMENTAL METRIC        EVALUATION         SCORE
----------------------------------------------------
Collateral Damage Potential [None - High]  {0 - 0.5}
Target Distribution         [None - High]  {0 - 1.0}
----------------------------------------------------
FORMULA                          ENVIRONMENTAL SCORE
----------------------------------------------------
round((4.4 + ((10 - 4.4) * {0 - 0.5})) *
     {0 - 1.00}) ==                    (0.00 - 7.20)
----------------------------------------------------
```

4.0 Future Considerations

The authors of CVSS recognize the difficulties with scoring vulnerabilities and assessing their risk. They realize that other scoring systems exist, both commercial [1], [2], [5] and non-commercial [3], [4]. While they are each equally valid, they consider a contrasting and fuzzy set of factors used to determine the final score.

CVSS differs by offering an open framework where anyone (and everyone) can use the same model to rank vulnerabilities in a consistent fashion while at the same time allowing for personalization within each user environment.

CVSS provides this by identifying and separately scoring the natural groupings of a vulnerability that combine to determine its overall risk. It offers a common language with which computer application and system vendors as well as end-users can consistently and openly score vulnerabilities. As CVSS matures, these metrics may expand or adjust making it even more accurate, flexible and representative of modern vulnerabilities and their risks.

5.0 References

[1] Microsoft Threat Scoring System

[2] Symantec Threat Scoring System

[3] CERT Vulnerability Scoring

[4] SANS Critical Vulnerability Analysis Scale Ratings

[5] Qualys Vulnerability Knowledgebase

EOF