NISTIR 7946

CVSS Implementation Guidance

Joshua Franklin
Charles Wergin
Harold Booth

http://dx.doi.org/10.6028/NIST.IR.7946

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 7946

CVSS Implementation Guidance

Joshua Franklin
Harold Booth
Computer Security Division
Information Technology Laboratory

Charles Wergin
CocoaSystems Inc.
Eldersburg, MD

http://dx.doi.org/10.6028/NIST.IR.7946

April 2014

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology Interagency or Internal Report 7946
42 pages (April 2014)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Computer Security Division publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems.

Abstract

This Interagency Report provides guidance to individuals scoring vulnerabilities using the Common Vulnerability Scoring System (CVSS) Version 2.0 scoring metrics. CVSS defines a vulnerability as a bug, flaw, weakness, or exposure of an application, system device, or service that could lead to a failure of confidentiality, integrity, or availability. The guidance in this document is the result of applying the CVSS specification to over 50 000 vulnerabilities scored by analysts at the National Vulnerability Database (NVD). This document is intended to serve as an extension to the CVSS Version 2.0 specification, providing additional guidance for difficult and/or unique scoring situations. To assist vulnerability analysts, common keywords and phrases are identified and accompanied by suggested scores for particular types of software vulnerabilities. The report includes a collection of scored vulnerabilities from the NVD, alongside a justification for the provided score. Finally, this report contains a description of the NVD's vulnerability scoring process.

Acknowledgments

The authors wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content including Melanie Cook, Nelson Hastings, Nicole Keller, Benjamin Long, Celia Paulsen, Victoria Pillitteri, and David Waltermire of NIST; Christopher McCormick and Matthew Storm of Booz Allen Hamilton; Meisam Izadjoo of Exeter Government Services; and Art Manion of the CERT Coordination Center. A special thanks is extended to Peter Mell for all of his work instantiating the National Vulnerability Database.

Audience

This document is intended for those wishing to score vulnerabilities via the CVSS including, but not limited to, vulnerability and risk analysts, software developers, and security professionals. It is assumed readers are familiar with the CVSS v2.0, although a thorough understanding of the specification is not required. The material in this document is technically oriented, and readers should possess a basic understanding of network, software, and system security principles and practices. Readers are encouraged to take advantage of the detailed information and examples provided throughout the text, and learn about the NVD's vulnerability scoring process.

Keywords

Common Vulnerability Scoring System Version 2.0; CVSS v2.0; National Vulnerability Database; NVD; security metrics; vulnerabilities; vulnerability scoring

Trademark Information

All product names are registered trademarks or trademarks of their respective companies.

CVE is a registered trademark and CWE is a trademark of The MITRE Corporation.

Table of Contents

1 Introduction 1

1.1 Purpose and Scope 1

1.2 Document Structure 1

1.3 Document Conventions 2

2 CVSS Overview 3

2.1 Exploring the Base Metrics 4

2.1.1 Access Vector 4

2.1.2 Access Complexity 4

2.1.3 Authentication 5

2.1.4 Confidentiality 6

2.1.5 Integrity 6

2.1.6 Availability 6

2.2 Limitations of the CVSS 7

2.3 Further Guidance and Considerations 8

3 Scoring Practices 10

3.1 Common Keywords, Phrases and Suggested Vectors 10

3.2 Suggested Scoring Templates 11

List of Appendices

Appendix A - NVD Scoring Examples 12
A.1 CVE-2012-5841 – XSS without Authentication 12
A.2 CVE-2012-2360 – XSS with Authentication 13
A.3 CVE-2011-2917 – SQL Injection 14
A.4 CVE-2013-0214 – Cross-site Request Forgery 15
A.5 CVE-2012-0656 – Race Condition 15
A.6 CVE-2012-6530 – Access Complexity Example 1 16
A.7 CVE-2012-3754 – Access Complexity Example 2 16
A.8 CVE-2008-1447 – The Kaminsky Bug 17
A.9 CVE-2011-3389 – Cryptographic Issues 18
A.10 CVE-2012-5533 – Denial of Service: Application 19
A.11 CVE-2011-3918 – Denial of Service: Operating System 19
A.12 CVE-2012-4687 – Poor Key Generation 20
A.13 CVE-2012-2144 – Session Fixation 20
A.14 CVE-2012-5652 – Information Leak 21
A.15 CVE-2011-1007 – Physically Proximate 21
A.16 CVE-2008-1453 – Network Adjacent 22
A.17 CVE-2012-4507 – NULL Pointer Dereference 23
A.18 CVE-2012-4472 – Unrestricted File Upload 23
A.19 CVE-2011-5252 – Open Redirect 24
A.20 CVE-2013-0900 – Use-After-Free 24
A.21 CVE-2013-1763 – Array Index Error 25
A.22 CVE-2012-0204 – Untrusted Search Path 26
A.23 CVE-2013-2292 – Physical Resource Consumption 27
A.24 CVE-2013-0969 – Integrity Complete 27
A.25 CVE-2011-4583 – Unspecified Impact 28
A.26 CVE-2012-5895 – Unknown Impact and Attack Vectors 28

Appendix B - NVD Scoring Methodology 29
B.1 Scoring Overview 29
B.2 Link Availability and Applicability 30
B.3 Link Verification 30
B.4 CWE Identification 30
B.5 Assigning CVSS Metrics 31

Appendix C - Acronyms and Abbreviations..... 32

Appendix D - References..... 33

1 Introduction

The Common Vulnerability Scoring System Version 2.0 (CVSS v2.0) provides an open framework for communicating the characteristics of vulnerabilities [12]. The CVSS v2.0 defines a vulnerability as a bug, flaw, weakness, or exposure of an application, system device, or service that could lead to a failure of confidentiality, integrity, or availability. The CVSS v2.0 model attempts to ensure repeatable and accurate measurement while enabling users to view the underlying vulnerability characteristics used to generate numerical scores. The CVSS v2.0 provides a common measurement system for industries, organizations, and governments requiring accurate and consistent vulnerability exploit and impact scores. Two common uses of the CVSS v2.0 are calculating the severity and prioritization of vulnerability remediation activities.

The National Vulnerability Database (NVD) is the U.S. government repository of standards based vulnerability management data. The NVD collects, analyzes and stores data describing specific computer system vulnerabilities enumerated by the Common Vulnerabilities and Exposure (CVE) dictionary [9] and the NVD supports the CVSS v2.0 specification for all vulnerabilities assigned a CVE identification number. Additionally, the NVD hosts databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics [11]. The NVD data assists automation of vulnerability management, security measurement, and compliance through the publication of machine-readable information.

1.1 Purpose and Scope

This document is intended to assist individuals who wish to score vulnerabilities via the CVSS v2.0. The guidance in this document is the result of applying the CVSS v2.0 specification to over 50 000 vulnerabilities scored by analysts at the National Vulnerability Database (NVD). The CVSS v2.0 comprises of three distinct metric groups - base, temporal, and environmental. While this document does not provide guidance for assessing the temporal and environmental metric groups, end-user organizations should obtain or assign values for all metric groups to fully determine the consequence of a vulnerability. Additionally, this report solely applies to CVSS v2.0 and all other versions are outside the scope of this report, as are other vulnerability scoring systems.

Guidance in this document for applying the CVSS v2.0 base metrics is provided in the following manner:

- Describing the CVSS v2.0 base metrics and providing guidance on implementing these metrics,
- Suggesting values for the CVSS v2.0 base metrics by enumerating common keywords and phrases,
- Providing a robust collection of scored vulnerabilities from the NVD, and
- Describing the process the NVD uses to collect, analyze, and score vulnerability information.

The included guidance demonstrates one manner of determining base scores for vulnerabilities. While much of the NVD's scoring process is discussed, the process of associating products to vulnerabilities is not covered.

1.2 Document Structure

The remainder of this document is organized into the following major sections:

- Section 2 provides an overview of the CVSS v2.0, and
- Section 3 details common keywords, phrases, and suggested scoring templates for performing vulnerability analysis.

The document also contains appendices with supporting material:

- Appendix A provides scored vulnerabilities, with corresponding explanations, from the NVD,
- Appendix B describes the internal process the NVD analysts use to collect, analyze, and assign the CVSS v2.0 base metrics,
- Appendix C defines selected acronyms and abbreviations used in this specification, and
- Appendix D contains a list of references used in the development of this document.

1.3 Document Conventions

The following conventions are used throughout the Interagency Report:

- All references to the CVSS are references to the Common Vulnerability Scoring System Version 2.0,
- Square brackets are used to indicate mutually exclusive elements, such as [High, Low]. In this instance, the element ‘High’ or ‘Low” would be selected from the two provided options, and
- CVEs are referenced throughout the body of the text and each CVE mentioned is discussed in detail within Appendix A except where otherwise noted.

2 CVSS Overview

The CVSS allows users to understand a standardized set of characteristics about vulnerabilities. These characteristics are conveyed in the form of a vector composed of three separate metric groups: base, environmental, and temporal. The base metric group is composed of six metrics: Access Vector (AV), Access Complexity (AC), Authentication (Au), Confidentiality (C), Integrity (I), and Availability (A). The base score, ranging from 0 to 10, is derived from an equation specified within the CVSS. AV, AC, and Au are often referred to as exploit metrics, while C, I, and A are referred to as impact metrics. The following graphic illustrates these concepts:

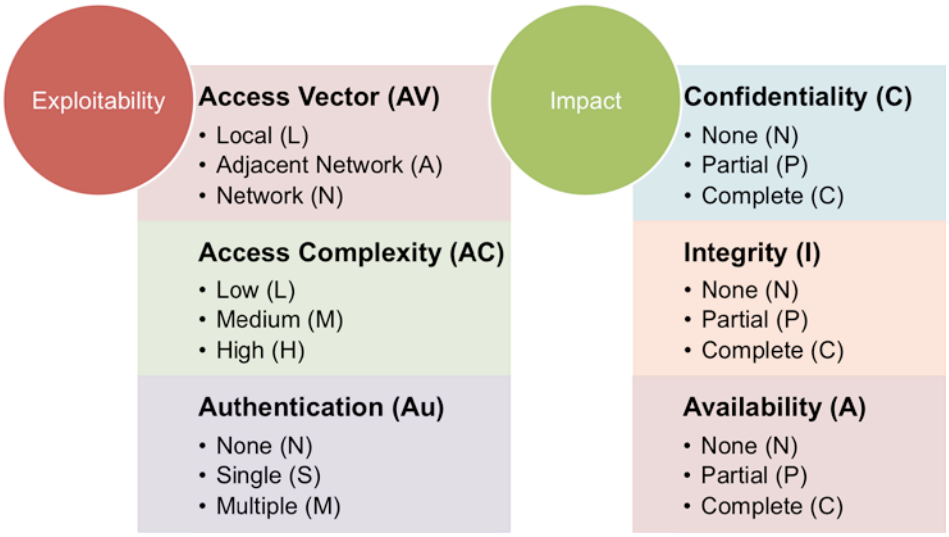

Figure 1 – CVSS Base Metrics

Vectors are expressed via a machine-readable textual representation of the values used to derive the score. This representation consists of the abbreviated metric name in a predetermined order, followed by a colon, and finally, the abbreviated metric value. The forward slash character ("/") is used to separate the metrics and square brackets are used to identify optional elements. A detailed description of the vector template is provided in section 2.1 and the CVSS specification [12]. The vector template syntax for the base score is:

AV:[L,A,N]/AC:[H,M,L]/Au:[M,S,N]/C:[N,P,C]/I:[N,P,C]/A:[N,P,C]

Organizations will typically have software with newly reported vulnerabilities affecting their systems on a daily basis. Vulnerabilities are disclosed in a variety of ways: through vendor advisories, security research reports, vulnerability databases, and bug tracking systems are a few examples. The CVSS specification can assist in comparing different vulnerabilities with each other. Vulnerability analysts are typically the individuals assessing vulnerabilities and assigning values for the various CVSS metrics. The base metric group measures the static qualities of a vulnerability that do not change over time. The temporal metric group measures the qualities of a vulnerability that do change over time, while the environmental metric group measures the characteristics unique and relevant to an individual platform or environment. The temporal metrics are primarily concerned with the availability of exploit code and patches, which often change over time. The environmental metrics are specific to an end-user environment allowing for adjustment based on the specific enterprise and services affected.

CVSS producers may optionally provide values for the temporal or environmental metric groups which provide important context that is not included in the base metric group. For example the Exploitability vector in the Temporal metrics may help to convey current threat information and Target Distribution in the Environmental metrics may help to convey exposure. If a CVSS producer chooses to measure the Temporal metrics it is recommended that the CVSS producer provide the date the Temporal vector values were assigned. If a CVSS producer chooses to measure the Environmental metrics it is recommended that the CVSS producer provide a description of the assumed end-user environment used when generating the Environmental vector values. End-users should update values for the Temporal and Environmental metric groups using more recent and locally relevant information for their organizations.

2.1 Exploring the Base Metrics

Guidance for assessing the six base metrics is provided within the following sections and should be used to compliment the definitions and information provided by the CVSS specification [12]. Limitations of the CVSS specification are discussed in section 2.2, and further considerations and guidance are provided in section 2.3.

2.1.1 Access Vector

The Access Vector metric measures an attacker's ability to successfully exploit a vulnerability based on how remote an attacker can be, from a networking perspective, to an information system. There are three possible values for this metric: Local (AV:L), Adjacent Network (AV:A), and Network (AV:N).

For the Access Vector to receive a value of "Network," a vulnerability must be exploitable without requiring physical (i.e., local) or adjacent network access. Often, AV:N vulnerabilities can be exploited from IP addresses on the Internet. Examples of terms that should trigger a vulnerability analyst to believe a vulnerability is AV:N are *remote, remotely exploitable, or remote attacker*. Appendix A includes a variety of AV:N vulnerabilities such as CVE-2012-5841, CVE-2013-0214, CVE-2012-5652, and CVE-2012-5895.

To receive a value of "Adjacent Network," vulnerabilities must be exploitable solely through a broadcast or collision domain, as in CVE-2008-1453. Examples of terms that should trigger a vulnerability analyst to believe the vulnerability is AV:A are *local network* or *adjacent*. Often the CVE description does not contain sufficient information to determine AV:A and requires reviewing security advisories relating to the vulnerability. Examples of *local networks* include, but are not limited to, wireless networks such as Wi-Fi or Bluetooth, or a connection to a local area network (LAN). Hardware vulnerabilities related to routers and switches are often categorized with an Access Vector of "Adjacent Network."

To receive a value of "Local," a vulnerability must only be exploitable via physical access, proximity to a device, or local shell/terminal access. Examples of terms that should trigger a vulnerability analyst to believe the vulnerability is AV:L are *local, physical access, or physically proximate*. To take advantage of CVE-2011-1007 one must have physical, or near physical access to the USB flash drive. It is important to note that local attacks do not suggest a change in score for the Authentication metric. If a vulnerability description mentions both remote and local access, then the appropriate metric should receive whichever value is more severe, according to the worst-case scenario.

2.1.2 Access Complexity

The Access Complexity metric is a means to convey the level of difficulty required for an attacker to exploit a vulnerability once the target system is identified. The amount of effort is estimated by the number of special or unique conditions required to exploit the vulnerability. Conditions not within the

control of the attacker will lower the overall score of the vulnerability. Access Complexity is evaluated independently; therefore changes in other base metrics are not considered reasons to raise Access Complexity. Access Complexity conditions typically include specialized access, non-default settings, and race conditions. In addition, other items outside the control of the attacker may raise Access Complexity.

One example of increased Access Complexity would be the user interaction required to open an attachment containing a malicious payload. A remote attacker would typically have no direct control over whether a user will open an attachment, therefore increasing the complexity to exploit the vulnerability. There are three possible values for this metric: High (AC:H), Medium (AC:M), and Low (AC:L). The CVSS specification contains examples to assist in determining the appropriate value for Access Complexity [12].

Any time a vulnerability has two or more specialized access conditions it should receive an Access Complexity value of "High." Other reasons include an atypically complex or extremely rare scenario, or a race condition which tightly narrows the window of opportunity for a successful attack. Vulnerabilities requiring expanded privileges or a specialized server configuration are often AC:H. For example, vulnerability CVE-2012-6530 requires non-default settings, such as specific privileges and a precise value for a configuration parameter, and therefore is AC:H.

For Access Complexity to be set to "Medium," a single special condition is required for a vulnerability to be exploited. If a victim is required to interact in some way to unintentionally assist an attacker, it is referred to as victim interaction. Victim interaction is a common property of vulnerabilities receiving an AC value of "Medium," and the NVD uses this concept to enhance CVSS by noting this property within the database. XSS vulnerabilities often rely on some level of victim interaction, and it can be observed in CVE-2012-5841 and CVE-2012-2360.

To receive a value of AC:L, no special conditions must be required for a vulnerability to be exploitable. If a vulnerability is present within default configurations or if it can be exploited with little skill or excessive information gathering, the Access Complexity is likely "Low." For instance, vulnerability CVE-2013-1763 is exploitable without special or unique circumstances, and is therefore AC:L. Vulnerabilities with insufficient information should receive a value of "Low."

2.1.3 Authentication

The Authentication metric measures the access an attacker requires to exploit a vulnerability. As the number of times an attacker must authenticate increases the CVSS base score will decrease. There are three possible values for this metric: Multiple (Au:M), Single (Au:S), and None (Au:N). A value for the Authentication metric is assigned to a vulnerability based upon the number of authentication instances required to exploit the vulnerability.

To receive a value of Au:M, the attacker must be required to successfully authenticate more than once in order to exploit a vulnerability. For instance, the requirement of authenticating to exploit a vulnerability within a restricted area of a web application, an attacker may need to first authenticate to gain access to the web application, and authenticate another time to gain privileged access. If an attacker must only prove their identity a single time, the Authentication metric is set to "Single." Note that this includes authenticating via the command line, a desktop session, or a web interface. Vulnerability CVE-2012-6530 references remote authenticated users; in this case an attacker is required to authenticate to the server (among other considerations) to exploit the vulnerability. Examples of terms that should trigger a vulnerability analyst to believe the vulnerability is AV:S are *authenticated users* or *authenticated attackers*. If authentication is not required to successfully exploit a vulnerability it receives a value of

Au:N. Many vulnerabilities, such as CVE-2012-3754 and CVE-2011-4583, within Appendix A do not require authentication.

2.1.4 Confidentiality

The Confidentiality metric measures the attacker's ability to obtain unauthorized access to information from an application or system. Disclosure of passwords, personal information, or other information used to control, configure or maintain systems are examples of a loss of Confidentiality. There are three possible values for this metric: None (C:N), Partial (C:P), and Complete (C:C).

If no information or data residing on or within a system is exposed due to exploitation, the Confidentiality metric receives a value of "None," as in examples CVE-2008-1447 and CVE-2011-3918. If there is unauthorized information disclosure, but less than complete read access to an entire system, the Confidentiality metric receives a value of "Partial," as in CVE-2012-5652. Finally, if an attacker has complete read access to all files and data on a system, the loss of Confidentiality is considered "Complete" as in CVE-2012-3754.

2.1.5 Integrity

The Integrity metric measures an attacker's ability to manipulate or remove data from a product or system. Altering data in a database, modifying files, changing access control lists, and DNS cache poisoning are all examples of a loss of Integrity. There are three possible values for this metric: None (I:N), Partial (I:P), and Complete (I:C).

I:N is used when vulnerability exploitation cannot manipulate data. For example, the information leak in CVE-2012-5652 only exposes information –modification is not possible. A "Partial" impact to Integrity occurs when exploiting a vulnerability will allow a limited or uncontrolled modification to files or other contents of a system, as in CVE-2012-2144. Additionally, a vulnerability will have a "Partial" impact if modification is confined only to the application context. For the Integrity metric to be I:C, an attacker must be able to arbitrarily modify any system file or other data throughout the system on an as needed basis. CVE-2013-0900 allows for remote code execution, and therefore a "Complete" impact to Integrity. CVE-2013-0969 is an example of a vulnerability with only an impact to Integrity - in this example it is "Complete."

It is important to remember that according to Scoring Tip #10 of the CVSS specification, a "Partial" or "Complete" loss of Integrity may also affect Availability because if data is altered, access to the unmodified data is no longer possible [12].

2.1.6 Availability

The Availability metric measures an attacker's ability to disrupt or prevent access to services or data. Vulnerabilities that impact availability can affect hardware, software, and network resources, such as flooding network bandwidth, consuming large amounts of memory, CPU cycles, or unnecessary power consumption. There are three possible values for this metric: None (A:N), Partial (A:P), and Complete (A:C).

When there are no impacts to the availability of system resources or data, the Availability metric should receive a value of "None." The impact is considered "Partial" if only an application is affected or if there are temporary resource or service interruptions, such as in CVE-2012-5533. Finally, to receive a value of "Complete," access to a resource must no longer be possible, often in the form of freezing all processing, shutting down the resource, or taking the information system offline. Vulnerability CVE-2011-3918

causes a system to enter into a reboot loop causing a "Complete" impact to Availability. Examples of terms and phrases that should trigger a vulnerability analyst to believe the vulnerability is A:C are *system hang* or a reference to a restart after an attack has occurred. CVE-2013-2292 is an example of a vulnerability with only an impact to Availability – in this example it is "Complete."

2.2 Limitations of the CVSS

While the CVSS provides a standardized mechanism to communicate a subset of vulnerability information, the CVSS has some limitations. These limitations include but are not limited to: evaluating relative vulnerability severity based exclusively on the score, only using the CVSS base metrics, and using the CVSS score as the sole means to determine organizational risk.

There are a number of cases where the overall consequence of a vulnerability is greater than the numerical CVSS base score since the CVSS ignores externality of vulnerability impact. The CVSS specification is meant to score the impact to the system containing the vulnerability, not any downstream impact to other systems. A common example is a vulnerability which exists within a web application; the vulnerability is evaluated based on the impact to the web server, impacts to other systems that may navigate to the web application containing the vulnerability are not taken into account. Scoring Tip #2 from the CVSS specification explicitly states that the score should only consider the direct impact to the target host and describes how to score a cross-site scripting vulnerability [12]. The externality of vulnerability impact limitation logically extends to similar type of vulnerabilities like cross-site request forgery (CSRF).

Another example where the CVSS base score discounts the impact of a vulnerability, is when that vulnerability is discovered within a protocol (or common implementations), such as TLS or DNS. CVE-2008-1447, colloquially referred to as the Kaminsky Bug, highlights a past flaw within DNS, and the severity only accounted for impact to the DNS server and not to clients relying on the DNS server [3]. Finally, vulnerabilities affecting cyber-physical and/or industrial control systems, such as CVE-2012-4687, may also require additional scrutiny as these systems directly affect the physical world and misuse of these systems could pose a serious threat to human life and safety. Use of the environmental metrics can provide some remedy for both the DNS and the industrial control systems examples to influence the final score, but perhaps not a comprehensive solution.

A reliance on only the CVSS base metrics without accounting for temporal aspects or environmental specific circumstances of a vulnerability may lead to organizations improperly measuring the severity of a vulnerability. While some environmental specific circumstances are accounted for through the use of the environmental metrics focusing largely on impact, no attempt is made to account in the CVSS for any mitigating factors within the context of an environment that could increase or decrease the ability to exploit a particular vulnerability. End-user organizations may wish to prioritize vulnerability response based on timely threat information, which is measured by the Exploitability vector in the temporal metrics.

Vulnerability assessment via the CVSS can assist in conducting risk assessments, but the CVSS scores should not be the sole factor when determining risk. The CVSS scores do not provide an aggregate score of a complete information system, and one should not sum up the scores to determine a final score for a system. Additionally, the CVSS score represents the impact of an individual vulnerability residing within an information system, and does not account for vulnerability chaining. Vulnerability chaining is the situation where multiple vulnerabilities are used together to perform an attack on a system. While useful as part of a risk management solution, the CVSS scores should not be used as the sole factor in determining risk.

2.3 Further Guidance and Considerations

When performing vulnerability analysis, organizations should determine which information sources they will accept when attempting to research a specific vulnerability. Security researchers, vendors, and governmental entities are categories of information sources that can be leveraged. In the event of a conflict between two sources, a hierarchy should be created to assist in determining which source is more authoritative. Organizations should determine how much effort vulnerability analysts should expend in order to provide values for the CVSS metrics. Vulnerability analysts may not initially have sufficient information to fully assess a given vulnerability and will on occasion be unable to identify an appropriate source containing the desired information. In the event insufficient information is available, vulnerabilities should be scored according to the worst-case scenario. Vulnerability descriptions often state this as *unknown impact vectors* or *unknown attack vectors*. The worst-case scenario for all six base metrics results in the Access Vector set as "Network," Authentication as "None," Access Complexity as "Low," and a value of "Complete" for the Confidentiality, Integrity, and Availability (CIA) triad. The worst-case scenario is represented by the following base vector:

AV:N/AC:L/Au:N/C:C/I:C/A:C

As an example the vulnerability description and available references for CVE-2012-5895 do not provide sufficient information to properly score the vulnerability and is therefore scored according to the worst-case scenario.

Reliably applying CIA impact levels across different classes of information systems and applications can be difficult. The following guidelines may assist in consistently assigning impact values. When considering Confidentiality, Integrity, and Availability at the application level, the resulting score is most likely "Partial" (i.e., CVE-2012-5533). As an example, when a vulnerability in an application renders an application unusable, as long as the underlying system is not compromised, the Availability value is "Partial." When considering vulnerabilities at the hardware or system level, the impact for an affected metric is generally "Complete" (i.e., CVE-2011-3918).

In addition to considering whether a vulnerability affects an application or system, it is also important to recognize that the security architecture of the operating system hosting the application influences impact. Access control and permission models, default settings, and configurations all vary widely from one operating system to the next, which affect vulnerability scores. The following example illustrates this scenario:

Operating System A by default results in applications running within the context of a privileged user with extended access to system information beyond those of a standard user would have. Operating System B by default results in applications running within the context of a process with standard or restricted system access. A vulnerability affecting an application running on Operating System A would result in higher impact scores than the same application running on Operating System B.

Occasionally, vulnerabilities which have been chained together as part of an exploit will be reported and described at the same time and in relation to each other making vulnerability assessment difficult. For instance, the iOS evasi0n jailbreak [15] leverages multiple vulnerabilities including CVE-2013-0977, CVE-2013-0978, CVE-2013-0979, and CVE-2013-0981 (these are not included within Appendix A.) Research is often required to identify and separate indistinctly reported vulnerabilities from each other. Vulnerabilities should be scored independently of each other as mentioned in Scoring Tip #1 [12]. Analysts should not consider the outcome of making a system or application more vulnerable as a reason to raise the score of the original vulnerability.

Finally, end-user organizations may wish to use the CVSS vectors with an alternative scoring or decision-making mechanism. This could help an organization better integrate readily-available CVSS vectors into existing vulnerability response processes.

3 Scoring Practices

Organizations who wish to produce consistent vulnerability scores from different vulnerability analysts should correlate terminology from disparate vulnerability sources with CVSS metrics and values. Creating a mapping from terminology to CVSS metrics and values enables the organization to ensure a repeatable process that can be communicated from those responsible for providing vulnerability assessments to security implementers and system administrators. This is only possible if the vulnerability descriptions use consistent wording and results may vary for sources outside of CVE.

3.1 Common Keywords, Phrases and Suggested Vectors

The following table contains common keywords and phrases typically used within vulnerability descriptions. These common keywords and phrases are commonly used within the description and/or reference links provided by the CVE dictionary entry and often suggest an initial value for a base metric. It is important to remember that these initial values can be influenced by other factors, and therefore analysts should consider all available information before determining a final value.

Table 1 - Common keywords and phrases in vulnerability descriptions

Metric	Common Keywords and Phrases	Suggested Value
Access Vector (AV)	Remote, remotely exploitable, remote attacker	AV:N
	Local network, adjacent network	AV:A
	Physically proximate[1]	AV:[A, L]
	Local, physical access	AV:L
	Context dependent (assume worst-case)	AV:N
	Unknown attack vectors	AV:N/AC:L/Au:N
Access Complexity (AC)	Where a *<configuration setting>* is enabled disabled	AC:M
Authentication (Au)	Authenticated user, authenticated attacker	Au:[S,M]
Confidentiality (C)	Read files, view sensitive information, information leak	C:[P,C]
Integrity (I)	Modify or delete files	I:[P,C]
Availability (A)	System hang, denial of service (DoS), reboot	A:[P,C]
CIA	Execute arbitrary code, execute arbitrary files	C:[P,C]/I:[P,C]/A:[P,C]
	Gain root privileges, gain system privileges, gain user privileges, gain administrator privileges, gain application privileges	C:[P,C]/I:[P,C]/A:[P,C]
	Unknown or unspecified impact	C:[P,C]/I:[P,C]/A:[P,C][2]

[1] Usually AV:L, but in certain cases the term "physically proximate" may be an indicator for AV:A, as in CVE-2008-1453.
[2] Usually "Complete," but where the impact is constrained to the context of the application, CIA would be assessed as "Partial."

3.2 Suggested Scoring Templates

The following scoring templates suggest typical scores for frequently occurring types of vulnerabilities described within the Common Weakness Enumeration (CWE) dictionary [10]. Based on information gathered from the NVD, these are some of the most common scoring scenarios that a vulnerability analyst may encounter. *It is important to consider that these scoring templates do not fit all situations.* Vulnerabilities often have unique characteristics that require deviation from these templates, and for some types of vulnerabilities, only a truncated vector can be supplied. Table 2 lists types of vulnerabilities by their CWE definition in no particular order.

Table 2 - Suggested Scoring Templates

CWE	CWE Name	Suggested Scores
CWE-59	Improper Link Resolution Before File Access ('Link Following')	AC:M
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	C:C/I:C/A:C
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	AC:M, C:N/I:P/A:N
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	C:P/I:P/A:P
CWE-96	Improper Neutralization of Directives in Statically Saved Code ('Static Code Injection')	C:P/I:P/A:P
CWE-129	Improper Validation of Array Index	AC:L
CWE-352	Cross-site Request Forgery (CSRF)	AC:[M,H]/C:P/I:P/A:P
CWE-384	Session Fixation	AC:M/C:[N,P]/I:P/A:[N,P]
CWE-399	Resource Management Errors[3]	A:C
CWE-399	Resource Management Errors[4]	A:[P,C]
CWE-416	Use-after-free	C:[P,C]/I:[P,C]/A:[P,C]
CWE-426	Untrusted Search Path	AC:[M,H]/C:C/I:C/A:C
CWE-434	Unrestricted File Upload	C:[P,C]/I:[P,C]/A:[P,C]
CWE-476	Null Pointer Dereference	AC:[L,M]/C:N/I:N/A:[P,H]
CWE-601	Open Redirect	C:P/I:P/A:N

[3] Affecting the hardware and/or operating system.
[4] Affecting the application.

Appendix A - NVD Scoring Examples

This section showcases a list of example vulnerabilities scored via the CVSS to assist vulnerability analysts in scoring vulnerabilities via the CVSS. The scores are based on information provided by the NVD and includes the CVE ID, CWE ID, CVSS base score, CVSS vector, a description of the vulnerability, and a justification for each CVSS base score.

A.1 CVE-2012-5841 – XSS without Authentication

CVE Description:

Mozilla Firefox before 17.0, Firefox ESR 10.x before 10.0.11, Thunderbird before 17.0, Thunderbird ESR 10.x before 10.0.11, and SeaMonkey before 2.14 implement cross-origin wrappers with a filtering behavior that does not properly restrict write actions, which allows remote attackers to conduct cross-site scripting (XSS) attacks via a crafted web site.

Additional Considerations:

The scoring template for Cross-site Scripting takes into consideration SCORING TIP #2 which states:

"When scoring a vulnerability, consider the direct impact to the target host only. For example, consider a cross-site scripting vulnerability: the impact to a user's system could be much greater than the impact to the target host. However, this is an indirect impact. Cross-site scripting vulnerabilities should be scored with no impact to confidentiality or availability, and partial impact to integrity."

Analysis:

Vector: AV:N/AC:M/Au:N/C:N/I:P/A:N Base Score: 4.3

CWE: CWE-79 - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Metric	Value	Explanation
Access Vector	Network	From keyword "remote attacker"
Access Complexity	Medium	From Table 2 Cross-site Scripting Scoring Template (due to victim interaction)
Authentication	None	Not required
Confidentiality	None	From Table 2 Cross-site Scripting Scoring Template
Integrity	Partial	From Table 2 Cross-site Scripting Scoring Template
Availability	None	From Table 2 Cross-site Scripting Scoring Template

A.2 CVE-2012-2360 – XSS with Authentication

CVE Description:

Cross-site scripting (XSS) vulnerability in the Wiki subsystem in Moodle 2.0.x before 2.0.9, 2.1.x before 2.1.6, and 2.2.x before 2.2.3 allows remote authenticated users to inject arbitrary web script or HTML via a crafted string that is inserted into a page title.

Additional Considerations:

The scoring template for Cross-site Scripting takes into consideration SCORING TIP #2 which states:

"When scoring a vulnerability, consider the direct impact to the target host only. For example, consider a cross-site scripting vulnerability: the impact to a user's system could be much greater than the impact to the target host. However, this is an indirect impact. Cross-site scripting vulnerabilities should be scored with no impact to confidentiality or availability, and partial impact to integrity."

Analysis:

Vector: AV:N/AC:M/Au:S/C:N/I:P/A:N Base Score: 3.5

CWE: CWE-79 - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Metric	Value	Explanation
Access Vector	Network	From keyword "remote…user"
Access Complexity	Medium	From Table 2 Cross-site Scripting Scoring Template (due to victim interaction)
Authentication	Single	From keyword "authenticated"
Confidentiality	None	From Table 2 Cross-site Scripting Scoring Template
Integrity	Partial	From Table 2 Cross-site Scripting Scoring Template
Availability	None	From Table 2 Cross-site Scripting Scoring Template

A.3 CVE-2011-2917 – SQL Injection

CVE Description:

SQL injection vulnerability in administrator/index2.php in Mambo CMS 4.6.5 and earlier allows remote attackers to execute arbitrary SQL commands via the zorder parameter.

Additional Considerations:

The scoring template for SQL Injection takes into consideration SCORING TIP #9 which states:

"Vulnerabilities with a partial or complete loss of integrity can also cause an impact to availability. For example, an attacker who is able to modify records can probably also delete them."

Analysis:

Vector: AV:N/AC:L/Au:N/C:P/I:P/A:P Base Score: 7.5

CWE: CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Metric	Value	Explanation
Access Vector	Network	From keyword "remote attackers"
Access Complexity	Low	No special conditions exist.
Authentication	None	Not required
Confidentiality	Partial	From Table 2 SQL Injection Scoring Template and affects only the application
Integrity	Partial	From Table 2 SQL Injection Scoring Template and affects only the application
Availability	Partial	From Table 2 SQL Injection Scoring Template and affects only the application

A.4 CVE-2013-0214 – Cross-site Request Forgery

CVE *Description:*

Cross-site request forgery (CSRF) vulnerability in the Samba Web Administration Tool (SWAT) in Samba 3.x before 3.5.21, 3.6.x before 3.6.12, and 4.x before 4.0.2 allows remote attackers to hijack the authentication of arbitrary users by leveraging knowledge of a password and composing requests that perform SWAT actions.

Analysis:

Vector: AV:N/AC:H/Au:N/C:P/I:P/A:P Base Score: 5.1

CWE: CWE-352 Cross-site Request Forgery (CSRF)

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	High	From Table 2 Cross-site Request Forgery (CSRF) due to victim interaction plus knowledge of password from vulnerability description
Authentication	None	Not required
Confidentiality	Partial	From Table 2 Cross-site Request Forgery (CSRF) Scoring Template and affects only the application
Integrity	Partial	From Table 2 Cross-site Request Forgery (CSRF) Scoring Template and affects only the application
Availability	Partial	From Table 2 Cross-site Request Forgery (CSRF) Scoring Template and affects only the application

A.5 CVE-2012-0656 – Race Condition

CVE Description:

Race condition in LoginUIFramework in Apple Mac OS X 10.7.x before 10.7.4, when the Guest account is enabled, allows physically proximate attackers to login to arbitrary accounts by entering the account name and no password.

Analysis:

Vector: AV:L/AC:M/Au:N/C:C/I:C/A:C Base Score: 6.2

CWE: CWE-362 – Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')

Metric	Value	Explanation
Access Vector	Local	From keyword “physically proximate attackers”
Access Complexity	Medium	From description “when the Guest account is enabled” (special condition, not enabled by default)
Authentication	None	Not required
Confidentiality	Complete	Worst case scenario if OS admin account accessed
Integrity	Complete	Worst case scenario if OS admin account accessed
Availability	Complete	Worst case scenario if OS admin account accessed

A.6 CVE-2012-6530 – Access Complexity Example 1

CVE Description:

Stack-based buffer overflow in Sysax Multi Server before 5.52, when HTTP is enabled, allows remote authenticated users with the create folder permission to execute arbitrary code via a crafted request.

Analysis:

Vector: AV:N/AC:H/Au:S/C:C/I:C/A:C Base Score: 7.1

CWE: CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer

Metric	Value	Explanation
Access Vector	Network	From keyword "remote…users"
Access Complexity	High	From description and reference link [13], "HTTP is enabled" is not a default parameter and user must have "create folder permission" which is not given by default
Authentication	Single	From keyword "authenticated"
Confidentiality	Complete	From reference link [13], "Sysax Multi Server runs as LOCALSYSTEM by default
Integrity	Complete	From reference link [13], "Sysax Multi Server runs as LOCALSYSTEM by default
Availability	Complete	From reference link [13], "Sysax Multi Server runs as LOCALSYSTEM by default

A.7 CVE-2012-3754 – Access Complexity Example 2

CVE Description:

Use-after-free vulnerability in the Clear method in the ActiveX control in Apple QuickTime before 7.7.3 allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via unspecified vectors.

Analysis:

Vector: AV:N/AC:M/Au:N/C:C/I:C/A:C Base Score: 9.3

CWE: CWE-399 - Resource Management Errors

Metric	Value	Explanation
Access Vector	Network	From keyword "remote attackers"
Access Complexity	Medium	From reference link [6] " by persuading a victim to visit a specially-crafted Web site…" (victim interaction)
Authentication	None	Not required
Confidentiality	Complete	Worst case scenario if victim has elevated privileges
Integrity	Complete	Worst case scenario if victim has elevated privileges
Availability	Complete	Worst case scenario if victim has elevated privileges

A.8 CVE-2008-1447 – The Kaminsky Bug

CVE *Description:*

The DNS protocol, as implemented in (1) BIND 8 and 9 before 9.5.0-P1, 9.4.2-P1, and 9.3.5-P1; (2) Microsoft DNS in Windows 2000 SP4, XP SP2 and SP3, and Server 2003 SP1 and SP2; and other implementations allow remote attackers to spoof DNS traffic via a birthday attack that uses in-bailiwick referrals to conduct cache poisoning against recursive resolvers, related to insufficient randomness of DNS transaction IDs and source ports, aka "DNS Insufficient Socket Entropy Vulnerability" or "the Kaminsky bug."

Analysis:

Vector: AV:N/AC:L/Au:N/C:N/I:P/A:P Base Score: 6.4

CWE: CWE-330 - Use of Insufficiently Random Values

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required.
Confidentiality	None	Not impacted
Integrity	Partial	Exploit allows attacker to control the destination of the victim
Availability	Partial	Exploit allows attacker to control the destination of the victim

A.9 CVE-2011-3389 – Cryptographic Issues

CVE Description:

The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other products, encrypts data by using CBC mode with chained initialization vectors, which allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.

Additional Considerations:

From reference link [4]:

"The code can be injected into the user's browser through JavaScript associated with a malicious advertisement distributed through a Web ad service or an IFRAME in a linkjacked site, ad, or other scripted elements on a webpage.

Using the known text blocks, BEAST can then use information collected to decrypt the target's AES-encrypted requests, including encrypted cookies, and then hijack the no-longer secure connection. That decryption happens slowly, however; BEAST currently needs sessions of at least a half-hour to break cookies using keys over 1,000 characters long."

Analysis:

Vector: AV:N/AC:M/Au:N/C:P/I:N/A:N Base Score: 4.3

CWE: CWE-310 – Cryptographic Issues

Metric	Value	Explanation
Access Vector	Network	One example use of SSL is HTTPS which is often exposed as a remote service
Access Complexity	Medium	Per *Additional Considerations*, an additional vulnerability is required for exploitation, alongside a large number of minimum requests for the attack to be successful.
Authentication	None	Not required
Confidentiality	Partial	From description "obtain plaintext HTTP headers" which should not be possible using SSL
Integrity	None	Not impacted
Availability	None	Not impacted

A.10 CVE-2012-5533 – Denial of Service: Application

CVE Description:

The http_request_split_value function in request.c in lighttpd before 1.4.32 allows remote attackers to cause a denial of service (infinite loop) via a request with a header containing an empty token, as demonstrated using the "Connection: TE,,Keep-Alive" header.

Analysis:

Vector: AV:N/AC:L/Au:N/C:N/I:N/A:P Base Score: 5.0

CWE: CWE-399 - Resource Management Errors

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	None	Not impacted
Integrity	None	Not impacted
Availability	Partial	From Table 2 Resource Management Template and affects only the application

A.11 CVE-2011-3918 – Denial of Service: Operating System

CVE Description:

The Zygote process in Android 4.0.3 and earlier accepts fork requests from processes with arbitrary UIDs, which allows remote attackers to cause a denial of service (reboot loop) via a crafted application.

Analysis:

Vector: AV:N/AC:L/Au:N/C:N/I:N/A:C Base Score: 7.8

CWE: CWE-399 - Resource Management Errors

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	None	Not impacted
Integrity	None	Not impacted
Availability	Complete	From Table 2 Resource Management Template and affects the operating system

A.12 CVE-2012-4687 – Poor Key Generation

CVE Description:

Post Oak AWAM Bluetooth Reader Traffic System does not use a sufficient source of entropy for private keys, which makes it easier for man-in-the-middle attackers to spoof a device by predicting a key value.

Analysis:

Vector: AV:N/AC:H/Au:N/C:C/I:C/A:C Base Score: 7.6

CWE: CWE-310 - Cryptographic Issues

Metric	Value	Explanation
Access Vector	Network	From reference link [8], "this vulnerability can be exploited remotely,"
Access Complexity	High	From the CVSS v2 specification description of High Access Complexity
Authentication	None	Not required
Confidentiality	Complete	From reference link [8], "by impersonating the device, an attacker can obtain the credentials of administrative users"
Integrity	Complete	From reference link [8], "by impersonating the device, an attacker can obtain the credentials of administrative users"
Availability	Complete	From reference link [8], "by impersonating the device, an attacker can obtain the credentials of administrative users"

A.13 CVE-2012-2144 – Session Fixation

CVE Description:

Session fixation vulnerability in OpenStack Dashboard (Horizon) folsom-1 and 2012.1 allows remote attackers to hijack web sessions via the sessionid cookie.

Analysis:

Vector: AV:N/AC:M/Au:N/C:P/I:P/A:P Base Score: 6.8

CWE: CWE-384 - Session Fixation

Metric	Value	Explanation
Access Vector	Network	From keyword "remote attackers"
Access Complexity	Medium	From reference link [7], "hijack web sessions" indicates victim interaction
Authentication	None	Not required
Confidentiality	Partial	Attacker obtains the privileges of the application user
Integrity	Partial	Attacker obtains the privileges of the application user
Availability	Partial	Attacker obtains the privileges of the application user

A.14 CVE-2012-5652 – Information Leak

CVE Description:

Drupal 6.x before 6.27 allows remote attackers to obtain sensitive information about uploaded files via a (1) RSS feed or (2) search result.

Analysis:

Vector: AV:N/AC:L/Au:N/C:P/I:N/A:N Base Score: 5.0

CWE: CWE-200 - Information Exposure

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	Partial	From description “obtain sensitive information about uploaded files” and only affects the application
Integrity	None	Not impacted
Availability	None	Not impacted

A.15 CVE-2011-1007 – Physically Proximate

CVE Description:

Best Practical Solutions RT before 3.8.9 does not perform certain redirect actions upon a login, which allows physically proximate attackers to obtain credentials by resubmitting the login form via the back button of a web browser on an unattended workstation after an RT logout.

Analysis:

Vector: AV:L/AC:L/Au:N/C:P/I:P/A:P Base Score: 4.6

CWE: CWE-310 – Cryptographic Issues

Metric	Value	Explanation
Access Vector	Local	From keyword “physically proximate”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	Partial	Attacker obtains the credentials of the application user
Integrity	Partial	Attacker obtains the credentials of the application user
Availability	Partial	Attacker obtains the credentials of the application user

A.16 CVE-2008-1453 – Network Adjacent

CVE Description:

The Bluetooth stack in Microsoft Windows XP SP2 and SP3, and Vista Gold and SP1, allows physically proximate attackers to execute arbitrary code via a large series of Service Discovery Protocol (SDP) packets.

Additional Considerations:

From reference link [1], the range of the Bluetooth radio in this context is listed as 0 m to 100 m.

Analysis:

Vector: AV:A/AC:L/Au:N/C:C/I:C/A:C Base Score: 8.3

CWE: CWE-20 - Improper Input Validation

Metric	Value	Explanation
Access Vector	Adjacent Network	From keyword "physically proximate" and within Bluetooth range. See *Additional Considerations*.
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	Complete	From reference link [14] "attackers can exploit this issue to execute arbitrary code with SYSTEM-level privileges"
Integrity	Complete	From reference link [14] "attackers can exploit this issue to execute arbitrary code with SYSTEM-level privileges"
Availability	Complete	From reference link [14] "attackers can exploit this issue to execute arbitrary code with SYSTEM-level privileges"

A.17 CVE-2012-4507 – NULL Pointer Dereference

CVE Description:

The strchr function in procmime.c in Claws Mail (aka claws-mail) 3.8.1 allows remote attackers to cause a denial of service (NULL pointer dereference and crash) via a crafted email.

Analysis:

Vector: AV:N/AC:L:Au:N/C:N/I:N/A:P Base Score: 5.0

CWE: CWE-476 - NULL Pointer Dereference

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	None	From Table 2 Null Pointer Dereference Scoring Template. Not impacted
Integrity	None	From Table 2 Null Pointer Dereference Scoring Template. Not impacted
Availability	Partial	From Table 2 Null Pointer Dereference Scoring Template and description “cause a denial of service” of the application

A.18 CVE-2012-4472 – Unrestricted File Upload

CVE Description:

Unrestricted file upload vulnerability in upload.php in the Drag & Drop Gallery module 6.x-1.5 and earlier for Drupal allows remote attackers to execute arbitrary PHP code by uploading a file with an executable extension followed by a safe extension, then accessing it via a direct request to the directory specified by the filedir parameter.

Analysis:

Vector: AV:N/AC:H/Au:N/C:P/I:P/A:P Base Score: 5.1

CWE: CWE-434 - Unrestricted Upload of File with Dangerous Type

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	High	From description uploading a file with an executable extension followed by a safe extension, then accessing it via a direct request to the directory specified by the filedir parameter.
Authentication	None	Not required
Confidentiality	Partial	From Table 2 Unrestricted File Upload Scoring Template and affects only application
Integrity	Partial	From Table 2 Unrestricted File Upload Scoring Template and affects only application
Availability	Partial	From Table 2 Unrestricted File Upload Scoring Template and affects only application

A.19 CVE-2011-5252 – Open Redirect

CVE Description:

Open redirect vulnerability in Users/Account/LogOff in Orchard 1.0.x before 1.0.21, 1.1.x before 1.1.31, 1.2.x before 1.2.42, and 1.3.x before 1.3.10 allows remote attackers to redirect users to arbitrary web sites and conduct phishing attacks via a URL in the ReturnUrl parameter.

Analysis:

Vector: AV:N/AC:M/Au:N/C:P/I:P/A:N Base Score: 5.8

CWE: CWE-601 - URL Redirection to Untrusted Site ('Open Redirect')

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Medium	From description “users to arbitrary web sites and conduct phishing attacks” indicating victim interaction
Authentication	None	Not required
Confidentiality	Partial	From Table 2 Open Redirect Scoring Template
Integrity	Partial	From Table 2 Open Redirect Scoring Template
Availability	None	From Table 2 Open Redirect Scoring Template

A.20 CVE-2013-0900 – Use-After-Free

CVE Description:

Use-after-free vulnerability in Microsoft Internet Explorer 6 through 10 allows remote attackers to execute arbitrary code via a crafted web site that triggers access to a deleted object, aka "Internet Explorer CCaret Use After Free Vulnerability."

Analysis:

Vector: AV:N/AC:M/Au:N/C:C/I:C/A:C Base Score: 9.3

CWE: CWE-416 - Use After Free

Metric	Value	Explanation
Access Vector	Network	From keyword “remote attackers”
Access Complexity	Medium	From description “via a crafted web site” indicating victim interaction
Authentication	None	Not required
Confidentiality	Complete	Worst case scenario if victim has elevated privileges
Integrity	Complete	Worst case scenario if victim has elevated privileges
Availability	Complete	Worst case scenario if victim has elevated privileges

A.21 CVE-2013-1763 – Array Index Error

CVE Description:

Array index error in the __sock_diag_rcv_msg function in net/core/sock_diag.c in the Linux kernel before 3.7.10 allows local users to gain privileges via a large family value in a Netlink message.

Analysis:

Vector: AV:L/AC:L/Au:N/C:C/I:C/A:C Base Score: 7.2

CWE: CWE-129 - Improper Validation of Array Index

Metric	Value	Explanation
Access Vector	Local	From keyword “local users”
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	Complete	From reference link [16] “An unprivileged local user could exploit this flaw to crash the system or run programs as an administrator”
Integrity	Complete	From reference link [16] “An unprivileged local user could exploit this flaw to crash the system or run programs as an administrator”
Availability	Complete	From reference link [16] “An unprivileged local user could exploit this flaw to crash the system or run programs as an administrator”

A.22 CVE-2012-0204 – Untrusted Search Path

CVE Description:

Untrusted search path vulnerability in InfoSphere Import Export Manager 8.1 through 9.1 in InfoSphere Information Server MetaBrokers & Bridges (MBB) in IBM InfoSphere Information Server 8.1, 8.5 before FP3, 8.7, and 9.1 allows local users to gain privileges via a Trojan horse DLL in the current working directory.

Additional Considerations:

There is a conflict between the CVE and vendor descriptions. While it can be reasonably assumed that the vendor has a better understanding of how a vulnerability can be exploited and extremity of the impact, some evidence should be provided. In this case the access vector Network is not explained in depth, but the advisory states "CVSS Base Score: 9.3 / CVSS Vector: (AV:N/AC:M/Au:N/C:C/I:C/A:C)"

Analysis:

Vector: AV:N/AC:M/Au:N/C:C/I:C/A:C Base Score: 9.3

CWE: CWE-426 - Untrusted Search Path

Metric	Value	Explanation
Access Vector	Network	From reference link [5] vendor advisory
Access Complexity	Medium	Requires placement of malicious DLL into current working directory
Authentication	None	Not required
Confidentiality	Complete	Worst case scenario if victim has elevated privileges
Integrity	Complete	Worst case scenario if victim has elevated privileges
Availability	Complete	Worst case scenario if victim has elevated privileges

A.23 CVE-2013-2292 – Physical Resource Consumption

CVE Description:

bitcoind and Bitcoin-Qt 0.8.0 and earlier allow remote attackers to cause a denial of service (electricity consumption) by mining a block to create a nonstandard Bitcoin transaction containing multiple OP_CHECKSIG script opcodes.

Analysis:

Vector: AV:N/AC:L/Au:N/C:N/I:N/A:C Base Score: 7.8

CWE: CWE-399 - Resource Management Errors

Metric	Value	Explanation
Access Vector	Network	From keyword "remote attackers"
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	None	From Table 2 Resource Management Errors Scoring Template Not impacted
Integrity	None	From Table 2 Resource Management Errors Scoring Template Not impacted
Availability	Complete	From Table 2 Resource Management Errors Scoring Template and impacts the device due to increased power consumption.

A.24 CVE-2013-0969 – Integrity Complete

CVE Description:

Login Window in Apple Mac OS X before 10.8.3 does not prevent application launching with the VoiceOver feature, which allows physically proximate attackers to bypass authentication and make arbitrary System Preferences changes via unspecified use of the keyboard.

Analysis:

Vector: AV:L/AC:L/Au:N/C:N/I:C/A:N Base Score: 4.9

CWE: CWE-264 - Permissions, Privileges, and Access Control

Metric	Value	Explanation
Access Vector	Local	From keyword "physically proximate"
Access Complexity	Low	No special conditions exist
Authentication	None	Not required
Confidentiality	None	Not impacted
Integrity	Complete	From description, "…make arbitrary System Preference changes…"
Availability	None	Not impacted

A.25 CVE-2011-4583 – Unspecified Impact

CVE Description:

Moodle 2.0.x before 2.0.6 and 2.1.x before 2.1.3 displays web service tokens associated with (1) disabled services and (2) users who no longer have authorization, which allows remote authenticated users to have an unspecified impact by reading these tokens

Analysis:

Vector: AV:N/AC:L/Au:S/C:P/I:P/A:P Base Score: 6.5

CWE: CWE-264 - Permissions, Privileges, and Access Controls

Metric	Value	Explanation
Access Vector	Network	From keyword “remote…attackers”
Access Complexity	Medium	No special conditions exist
Authentication	None	From keyword “authenticated”
Confidentiality	Partial	From description, “unspecified impact” and affects only application
Integrity	Partial	From description, “unspecified impact” and affects only application
Availability	Partial	From description, “unspecified impact” and affects only application

A.26 CVE-2012-5895 – Unknown Impact and Attack Vectors

CVE Description:

Multiple unspecified vulnerabilities in iRODS before 3.1 have unknown impact and attack vectors.

Additional Considerations:

In cases where available information is too ambiguous to be useful, assume worst case scenario

Analysis:

Vector: AV:N/AC:L/Au:N/C:C/I:C/A:C Base Score: 10.0

CWE: Insufficient information

Metric	Value	Explanation
Access Vector	Network	From description “unknown impact and attack vectors”
Access Complexity	Low	From description “unknown impact and attack vectors”
Authentication	None	From description “unknown impact and attack vectors”
Confidentiality	Complete	From description “unknown impact and attack vectors”
Integrity	Complete	From description “unknown impact and attack vectors”
Availability	Complete	From description “unknown impact and attack vectors”

Appendix B - NVD Scoring Methodology

This appendix describes the process NVD uses to collect, analyze, and score vulnerabilities in accordance with the CVSS. An overview of the CVSS is provided within section 2. Version 2.0 of the CVSS was first established as the vulnerability scoring system used by SCAP in specification version 1.0 [2] and has been used as primary guidance by the NVD since September 2007. Vulnerabilities scored prior to September 2007 used version 1.0 of the CVSS and were approximated to version 2.0's metrics without human analysis and are noted as "incomplete approximation" in the description.

B.1 Scoring Overview

The NVD receives vulnerability information via the CVE dictionary data feeds. This information allows the NVD vulnerability analysts to perform research using links from CVE data feeds, and the analysts' conclusions are captured within a web application developed by the NVD development team.

The CVE dictionary feeds include:

- The unique CVE identifier,
- A description of the vulnerability, and
- Links to websites and other references with information related to the vulnerability.

NVD vulnerability analysts process this information in four distinct steps:

1. **Link Availability and Applicability** - Verify that the links supplied are publically available and are related to the vulnerability,
2. **Link Verification** - Identify if a link contains specific information that directly relates to any of the following:
    - A U.S. government resource,
    - An advisory notice or bulletin,
    - A patch or update for this vulnerability, and
    - Proof of concept or exploit code.
3. **CWE Identification** - Determine if the vulnerability description and/or information available in the reference links can be used to categorize the vulnerability as recognized in the CWE dictionary, and
4. **Assigning CVSS Metrics** - Assign the CVSS base metric values, using previously determined suggested scoring templates when possible to ensure consistent scoring among vulnerability analysts.

Additional guidance for these four steps is provided in the following sections.

B.2 Link Availability and Applicability

It is necessary to verify that the links supplied by the CVE data feed are publically available and are related to the vulnerability under scrutiny. The NVD analysts are presented with all of the references provided from the CVE data feed. Analysts should navigate to each reference link and verify that it resolves to an active web page and that the web page contains information pertinent to the vulnerability being analyzed. If a link is not pertinent to the vulnerability, analysts should 'hide' the link from the published vulnerability on the NVD web site. The vulnerability should be noted for later analysis, as links are dynamic and may be updated in the future, at which time the link can be reactivated.

B.3 Link Verification

The next step is to determine if the reference link contains specific information that directly relates to any of the following:

- A U.S. Government Resource – Indicated by generic top-level domains (gTLD), typically .gov, .mil, although others are included,
- An advisory notice or bulletin – Including vendors of the vulnerable product and well-known security research organizations,
- A patch or update – This must be a downloadable installation package that does not require any user manipulation (e.g., manual code modifications). Workarounds are not considered patches. Typically, links identified as containing patches should resolve to an actual download within three re-directs, and
- Proof of concept or exploit code – This can be actual code or a link to a proof-of-concept.

If reference links can be directly mapped to one of the previous descriptions, it will be indicated on the published web page.

B.4 CWE Identification[5]

Categorizing the type of the software vulnerability is the next step in the vulnerability analysis process. The description and/or information available in reference links can be used to classify the vulnerability according to the CWE dictionary. The NVD uses a subset of the CWE dictionary to determine the type of vulnerability or exposure being used to exploit the CVE. Most commonly, this information is directly available within the CVE description. NVD analysts assign the CWE type available from the subset list. If a CWE is indicated but not available, analysts should use the CWE dictionary to map the vulnerability based on the CWE taxonomy. If the CWE exists, but cannot be mapped directly, the CVE is labeled as CWE-Other. Other options include:

- Design error – This should only be used if it is indicated by the vendor of the vulnerable software.
- Not in CWE – Used to identify a weakness that is not part of the current CWE dictionary.

[5] http://nvd.nist.gov/cwe.cfm#cwes

- Insufficient Information – Many CVEs do not identify a specific vulnerability type.

CWE assignment has a direct impact on CVSS scores, as certain types of vulnerabilities are explicitly scored within examples and Scoring Tips. The NVD has expanded on this notion by developing the suggested scoring templates available within section 3.

B.5 Assigning CVSS Metrics

The final step in the vulnerability assessment process is to assign the CVSS base metrics. This is primarily accomplished via the use of common keywords within CVE descriptions and external research. An initial attempt is made to match the vulnerability to a scoring template such as in Table 2, but if the information within the CVE description is ambiguous or the templates do not apply, analysts should attempt to utilize previously analyzed vulnerabilities available in the NVD data set by way of the public search capabilities on the NVD website. Searching for a keyword or phrase in the description may return an exact match or similar result that can be used as scoring guidance.

If a vendor or third party includes a CVSS score as part of a reference link to a vulnerability, consider the source and whether or not the CVSS guidance is being implemented correctly. Often, when a vendor provides a conflicting score, it is due to the existence of additional information that has not been publically disclosed. While every effort should be made to determine why a vendor-provided score does not conform with an original assessment, the NVD analysts will generally only use publically available information to score a vulnerability.

Appendix C - Acronyms and Abbreviations

Selected terms used in the publication are defined below.

API	Application Programming Interface
CIA	Confidentiality, Integrity, and Availability
CSRF	Cross-site Request Forgery
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DNS	Domain Name System
FIRST	Forum of Incident Response and Security Teams
HW	Hardware
ICS	Industrial Control System
LAN	Local Area Network
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
OS	Operating System
RFC	Request for Comment
SCAP	Security Content Automation Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
SW	Software
XSS	Cross-site Scripting

Appendix D - References

[1] Bluetooth SIG, *Bluetooth Basics: A Look at the Basics of Bluetooth Technology*. [Web page] http://www.bluetooth.com/Pages/Basics.aspx [accessed 3/26/14].

[2] D. Waltemire, S. Quinn, K. Scarfone, and A. Halbardier, *The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2*, NIST SP 800-126 Revision 2, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, September 2011. [Web page] http://csrc.nist.gov/publications/nistpubs/800-126-rev2/SP800-126r2.pdf [accessed 3/26/14].

[3] D. Kaminsky, *It's The End Of The Cache As We Know It*, Presented at Black Ops 2008, Japan, 2008. [Web page] http://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf [accessed 3/26/14].

[4] Gallagher, Sean, "New JavaScript hacking tool can intercept PayPal, other secure sessions," *ArsTechnica*, September 21, 2011. [Web page] http://arstechnica.com/business/2011/09/new-javascript-hacking-tool-can-intercept-paypal-other-secure-sessions/ [accessed 3/26/14].

[5] IBM, *Security Bulletin: Multiple security vulnerabilities in the IBM InfoSphere Information Server Suite*. [Web page] http://www-01.ibm.com/support/docview.wss?uid=swg21623501 [accessed 3/26/14].

[6] IBM Internet Security Systems, *Apple QuickTime Clear() code execution*. [Web page] http://xforce.iss.net/xforce/xfdb/79901 [accessed 3/26/14].

[7] IBM Internet Security Systems, *OpenStack Dashboard session hijacking*. [Web page] http://xforce.iss.net/xforce/xfdb/75423 [accessed 3/26/14].

[8] Industrial Control Systems Cyber Emergency Response Team, *Post Oak Bluetooth Traffic Systems Insufficient Entropy Vulnerability*, Advisory (ICSA-12-335-01), November 30, 2012. [Web page] http://ics-cert.us-cert.gov/advisories/ICSA-12-335-01 [accessed 3/26/14].

[9] MITRE, *Common Vulnerabilities and Exposures*. [Web page] http://cve.mitre.org/ [accessed 3/26/14].

[10] MITRE, *Common Weakness Enumeration*. [Web page] http://cwe.mitre.org/ [accessed 3/26/14].

[11] National Institute of Standards and Technology, *National Vulnerability Database*. [Web page] http://nvd.nist.gov/ [accessed 3/26/14].

[12] P. Mell, K. Scarfone and S. Romanosky, *A Complete Guide to the Common Vulnerability Scoring System Version 2.0 (CVSS)*, Forum of Incident Response and Security Team (FIRST), June 2007. [Web page] http://www.first.org/cvss/cvss-guide.pdf [accessed 3/26/14].

[13] pwnag3. *Sysax Multi Server 5.50 Exploit*, January 17, 2012. [Web page] http://www.pwnag3.com/2012/01/sysax-multi-server-550-exploit.html [accessed 3/26/14].

[14] Security Focus, *Microsoft Windows Bluetooth Stack Remote Code Execution Vulnerability*, June 10, 2008. [Web page] http://www.securityfocus.com/bid/29522/info [accessed 3/26/14].

[15] The iPhone Wiki, *evasi0n*. [Web page] http://theiphonewiki.com/wiki/Evasi0n [accessed 3/26/14].

[16] Ubuntu, *Ubuntu Security Notice USN-1749-1, Linux kernel (Quantal HWE) vulnerability*, February 26, 2013. [Web page] http://www.ubuntu.com/usn/USN-1749-1/ [accessed 3/26/14].