The Common Vulnerability Scoring System (CVSS)v2

Gavin Reid

[For Public Release]

Agenda

- Introduction and overview of CVSS
- Why CVSS?
- Internals
- Scoring
- Roadmap
- Closing comments and questions

Overview

- Common Vulnerability Scoring System (CVSS)
- A universal language to convey vulnerability severity and help determine urgency and priority of response
- Solves problem of multiple, incompatible scoring systems in use today
- Initially a NIAC project
  - Subgroup of the global Vulnerability Disclosure Framework WG
  - Now under the custodial care of FIRST-SIG
- Open
- Usable, understandable, and dissectible by anyone
- In v2 now (June 20th 2007)

A joint NIAC effort

- Cisco
- Symantec
- Qualys
- eBay
- DHS/MITRE
- CERT/CC
- Microsoft
- ISS

Adopters

Amazon
ArcSight
Assuria Auditor
Big Fix
Cisco
(MySDN) CSC
eBay
E-Secure-IT
HP
IBM
Intellishield
IPA (Japan)
ISS
McAfee
MySDN
nCircle
netVigilance
Tenable Network Security
netForensics
NIST
Npower
Oracle
Redseal
RWE
Skype
Symantec
Skybox Security
ThreatGuard, Inc.
Qualys
US-CERT
webMethods

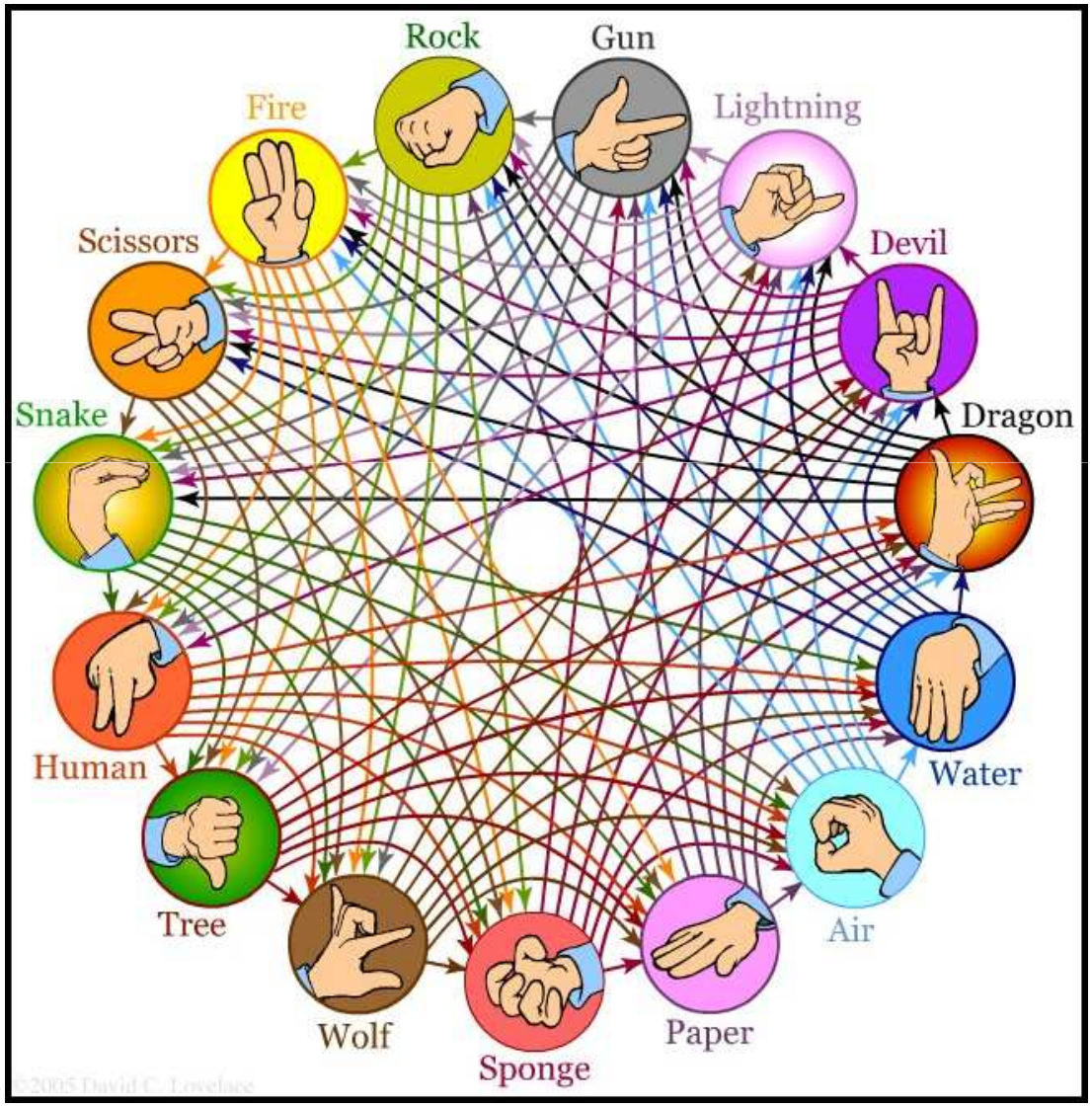
Scope Constraints

- CVSS is not:
  - Threat scoring system (The DHS color warning system)
  - Vulnerability database (Symantec's bugtraq)
  - Real-time attack scoring system (Symantec's ARIS)
  - Overall risk management program

Why CVSS?

- Different Organizations
  - Vendors (response)
  - Coordinators (notification, coordination)
  - Reporters (research, discovery)
  - Users (mitigation)
- All have different roles, motivations, priorities, resources, etc
- We need a common way to communicate!
- Set an industry example on alert disclosure

How do we score now?



Vendor Scoring: Microsoft

Rating	Definition
Critical	A vulnerability whose exploitation could allow the propagation of an Internet worm without user action.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users data, or of the integrity or availability of processing resources.
Moderate	Exploitability is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
Low	A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

Coordinator Scoring: CERT/CC

The metric value is a number between 0 and 180 that assigns an approximate severity to the vulnerability. This number considers several factors, including:

- Q1 Is information about the vulnerability widely available or known?
- Q2 Is the vulnerability being exploited in the incidents reported?
- Q3 Is the Internet Infrastructure at risk because of this vulnerability?
- Q4 How many systems on the Internet are at risk from this vulnerability?
- Q5 What is the impact of exploiting the vulnerability?
- Q6 How easy is it to exploit the vulnerability?
- Q7 What are the preconditions required to exploit the vulnerability?

```
3 * (Q1 + Q2 + Q3) * (Q4 * Q5 * Q6 * Q7) / (20^4)
```

Researcher Scoring: Secunia

Rating	Definition
Extremely Critical	Typically used for remotely exploitable vulnerabilities, which can lead to system compromise. Successful exploitation does not normally require any interaction and exploits are in the wild.
Highly Critical	As Above, no known exploits
Moderately Critical	As Above, but DoS only or requiring user interaction
Less Critical	XSS, privilege escalation, sensitive data exposure
Not Critical	Very limited privilege escalation, locally exploitable DoS, non-sensitive data exposure

And the User…?

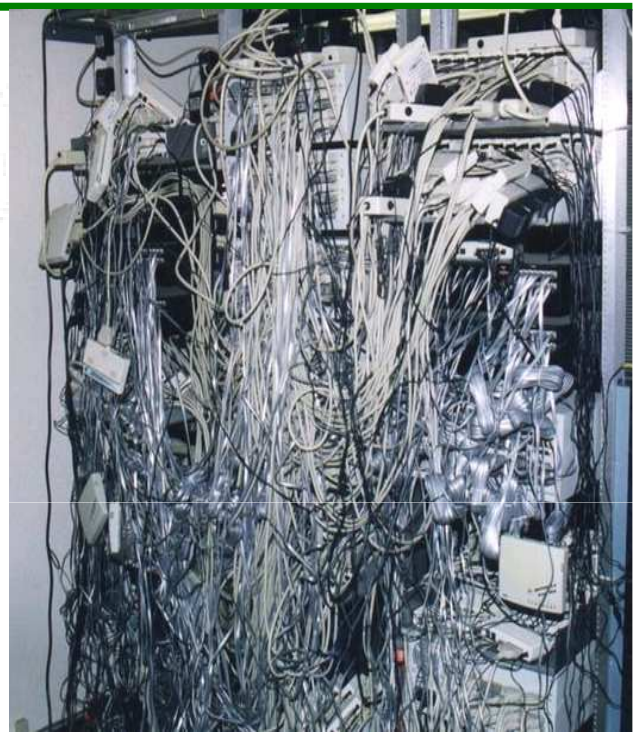
- Microsoft says “Important”
- CERT says “47.31”
- Secunia says “Less Critical”

- User says “Huh?”

The Busy Security Operations Guy

2000-2005

Year	2000	2001	2002	2003	2004	1Q,2005
Vulnerabilities	1,090	2,437	4,129	3,784	3,780	1,220



What does it mean to have 4,129 vulnerabilities reported in 2002?

- Read the descriptions
  - 4,129 vulnerabilities * 15 minutes = 129 days
- Affected by 10% of the vulnerabilities?
- Install patches on one system
  - 413 vulnerabilities * 1 hour = 52 days
- Reading reports and patching a single system costs 129 + 52 = 181 days
- Which vulnerability should I patch first?
- Remote root in DNS? Web server?
- Desktop systems? DoS affecting routing infrastructure?

Scoring Discrepancy Chart

TOP STORY CHART

A LOOK AT RECENT VULNERABILITY RATINGS

CRN

Each organization that rates security flaws in vendors' products uses its own rating scale (depicted numerically in the chart below) and often differs from other groups on the severity of these vulnerabilities. For companies that use these ratings to develop a proactive security posture, it can be difficult to sift through the conflicting threat information to determine how—or if—a particular vulnerability will affect their network. Following are ratings of recent high-profile security vulnerabilities from several organizations that regularly publish threat analysis information.

Vulnerability (CVE Number)	Symantec*	National Vulnerability Database CVSS	eEye	Secunia	Internet Security Systems	FrSIRT	McAfee
Symantec Client Security and Symantec AntiVirus Elevation of Privilege (CVE-2006-2630)	9.4/10 (aggregate)	7/10	High (3/3)	Moderately critical (3/5)	High (3/3)	Critical (4/4)	Did not rate
Cisco Wireless Access Point Web Interface Authorization Bypass (CVE-2006-3291)	9.8/10 (aggregate)	7/10	Did not rate	Less critical (2/5)	Medium (2/3)	Moderate (2/4)	Did not rate
Cisco Internet Key Exchange Denial Of Service Vulnerability (CVE-2006-3906)	6/10 (aggregate)	2.3/10	Did not rate	Did not rate	Low (1/3)	Did not rate	Did not rate
Cisco Secure ACS Session Management Security Issue (CVE-2006-3226)	9.4/10 (aggregate)	7/10	Did not rate	Less critical (2/5)	Medium (2/3)	Low (1/4)	Did not rate
Symantec Backup Exec Multiple Heap Overflow Vulnerabilities (CVE-2006-4128)	8.8/10 (aggregate)	4.2/10	Did not rate	Moderately critical (3/5)	High (3/3)	Critical (4/4)	Did not rate
IBM Informix Dynamic Server Multiple Vulnerabilities (multiple CVE entries)	9.8/10 (aggregate)	4.5/10 (aggregate)	Did not rate	Moderately critical (3/5)	High (3/3)	High (3/4)	Did not rate
Apple Xsan Path Name Buffer Overflow Vulnerability (CVE-2006-3506)	9.4/10 (aggregate)	4.9/10	Did not rate	Less critical (2/5)	Did not rate	Moderate (2/4)	Did not rate
McAfee SecurityCenter Vulnerability (CVE-2006-3961)	7.8/10 (aggregate)	7/10	High (3/3)	Highly critical (4/5)	High (3/3)	Critical (4/4)	Medium (2/3)

Note: CVE = Common Vulnerabilities and Exposures (A list of standardized names for vulnerabilities and other information security exposures funded by the U.S. Department of Homeland Security);
FrSIRT = French Security Incident Response Team
*Symantec scores are presented in aggregate of three separate DeepSight Threat Management System ratings: Urgency, Impact and Severity

FIRST™ Improving Security Together

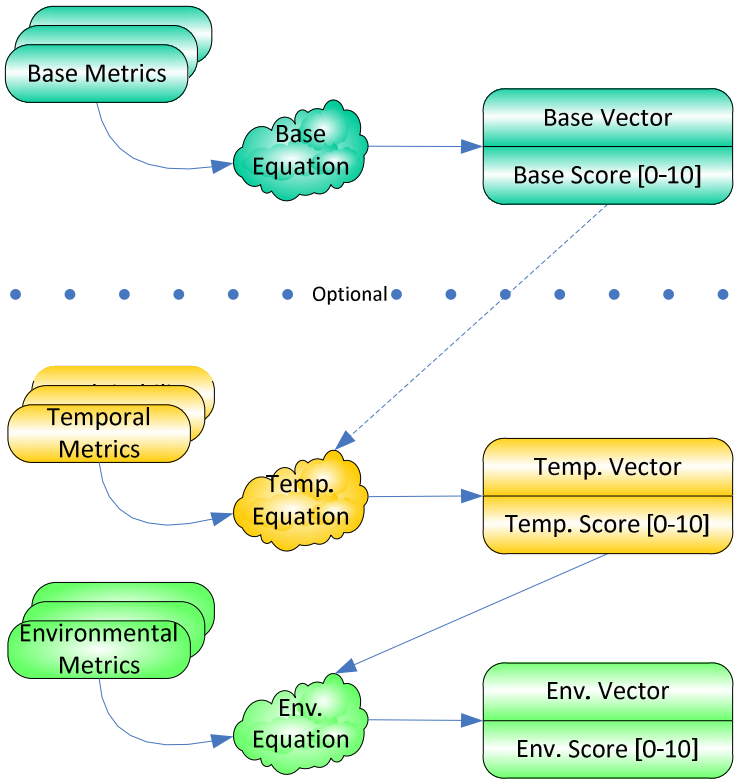
How does CVSS work?

- Metrics and formulas yield a score
- That's all!

Metrics

- A constituent component or characteristic of a vulnerability that can be quantitatively or qualitatively measured
- Make up the bulk of CVSS
- Three distinct groups
  - Base Metrics (Good)
  - Temporal Metrics (Better)
  - Environmental Metrics (Best)
- Designed with OBJECTIVITY in mind

CVSS (Metrics View)



Base Metric Group

- Most fundamental qualities of a vulnerability
- Do not change; “Immutable”
- Intrinsic attributes of a vulnerability
- 7 Base metrics

Base Metrics

Access Vector (AV)

- Measures whether a vulnerability is exploitable locally or remotely
- Local (L): The vulnerability is only exploitable locally
- Adjacent Network (A): The vulnerability must be staged from either the broadcast or collision domain of the vulnerable software
- Network (N): The vulnerability is exploitable remotely (and possibly locally as well) An example of a network attack is an RPC buffer overflow.

Base Metrics

Access Complexity (AC)

- Measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system
- High (H) : Specialized access conditions exist. For example:
- In most configurations, the attacking party must already have elevated privileges or spoof additional systems in addition to the attacking system (e.g., DNS).
- The attack depends on social engineering methods that would be easily detected by knowledgeable people. For example, the victim must perform several suspicious or atypical actions.
- The vulnerable configuration is seen very rarely in practice. - If a race condition exists, the window is very narrow.
- Medium (M) : The access conditions are somewhat specialized; the following are examples:
- The attacking party is limited to a group of systems or users at some level of authorization
- The affected configuration is non-default, and is not commonly configured
- The attack requires a small amount of social engineering that might occasionally fool cautious
- Low (L) : Specialized access conditions or extenuating circumstances do not exist. The following are examples:
- The affected product typically requires access to a wide range of systems and users, possibly anonymous and untrusted (e.g., Internet-facing web or mail server).
- The attack can be performed manually and requires little skill or additional information gathering. Used default configuration

Base Metrics

Authentication (Au)

- Measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability
- Multiple (M) Exploiting the vulnerability requires that the attacker authenticate two or more times, even if the same credentials are used each time. An example is an attacker authenticating to an operating system in addition to providing credentials to access an application hosted on that system.
- Single (S) The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).
- None (N) Authentication is not required to exploit the vulnerability.

Base Metrics

Confidentiality Impact (C)

- Measures the impact on confidentiality of a successful exploit of the vulnerability on the target system
- None (N): No impact on confidentiality
- Partial (P): There is considerable informational disclosure
- Complete (C) : A total compromise of critical system information

Base Metrics

Integrity Impact (I)

- Measures the impact on Integrity of a successful exploit of the vulnerability on the target system
- None (N): No impact on integrity
- Partial (P): Considerable breach in integrity
- Complete (C) : A total compromise of system integrity

Base Metrics

Availability Impact (A)

- Measures the impact on Availability of a successful exploit of the vulnerability on the target system
- None (N) : No impact on availability
- Partial (P) : Considerable lag in or interruptions in resource availability
- Complete (C) : Total shutdown of the affected resource

Temporal Metric Group

- Time dependent qualities of a vulnerability
- 3 Temporal metrics

Temporal Metrics

Exploitability (E)

- Measures how complex the process is to exploit the vulnerability in the target system once it has been accessed
- Unproven (U): No exploit code is yet available
- Proof of Concept (POC): Proof of concept exploit code is available
- Functional (F) : Functional exploit code is available
- High (H): Exploitable by functional mobile autonomous code or no exploit required (manual trigger)
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

Temporal Metrics

Remediation Level (RL)

- Measures the level of solution available
- Official Fix (OF): Complete vendor solution available
- Temporary Fix (TF): There is an official temporary fix available
- Workaround (W) : There is an unofficial non-vendor solution available
- Unavailable (U): There is either no solution available or it is impossible to apply
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric

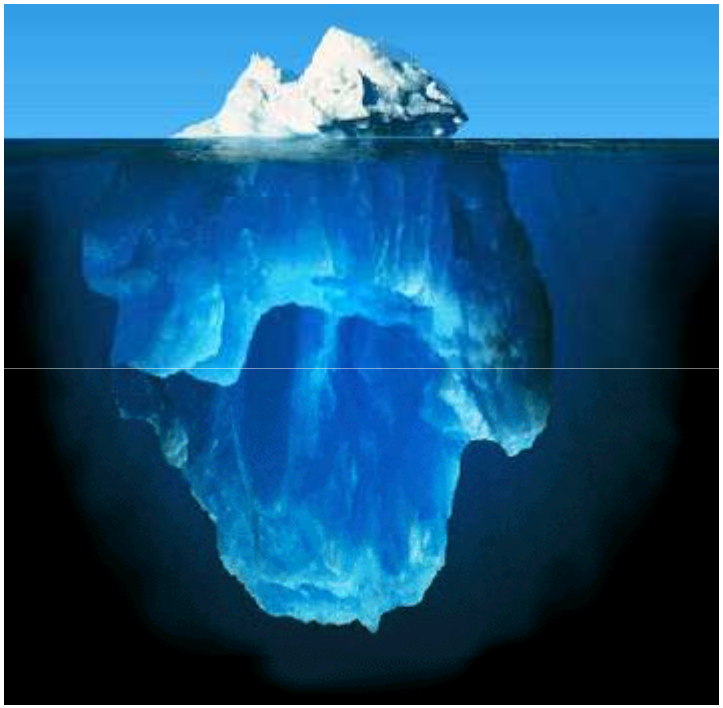
Temporal Metrics

Report Confidence (RL)

- Measures the degree of confidence in the existence of the vulnerability and the credibility of its report
- Unconfirmed (UC): A single unconfirmed source or possibly several conflicting reports
- Uncorroborated (UR): Multiple non-official sources; possibly including independent security companies or research organizations
- Confirmed (C): Vendor has reported/confirmed a problem with its own product
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric

Environmental Metric Group

- Implementation and environment specific qualities of a vulnerability
- 3 Environmental metrics



Environmental Metrics

Collateral Damage Potential (CDP)

- This metric measures the potential for loss of life or physical assets through damage or theft of property or equipment.
- None (N): There is no potential for physical assets, productivity or revenue damage
- Low (L): A successful exploit of this vulnerability may result in slight loss of revenue or productivity to the organization
- Low-Medium (LM): A successful exploit of this vulnerability may result in moderate loss of revenue or productivity to the organization.
- Medium-High (MH): A successful exploit of this vulnerability may result in significant loss of revenue or productivity
- High (H):A successful exploit of this vulnerability may result in catastrophic loss of revenue or productivity.
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric

Environmental Metrics

Target Distribution (TD)

- Measures the relative size of the field of target systems susceptible to the vulnerability
- None (N) : No target systems exist, or targets are so highly specialized that they only exist in a laboratory setting (0%)
- Low (L): Targets exist inside the environment, but on a small scale (1% - 15%)
- Medium (M): Targets exist inside the environment, but on a medium scale (16% - 49%)
- High (H) : Targets exist inside the environment on a considerable scale (50% - 100%)
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric

Environmental Metrics

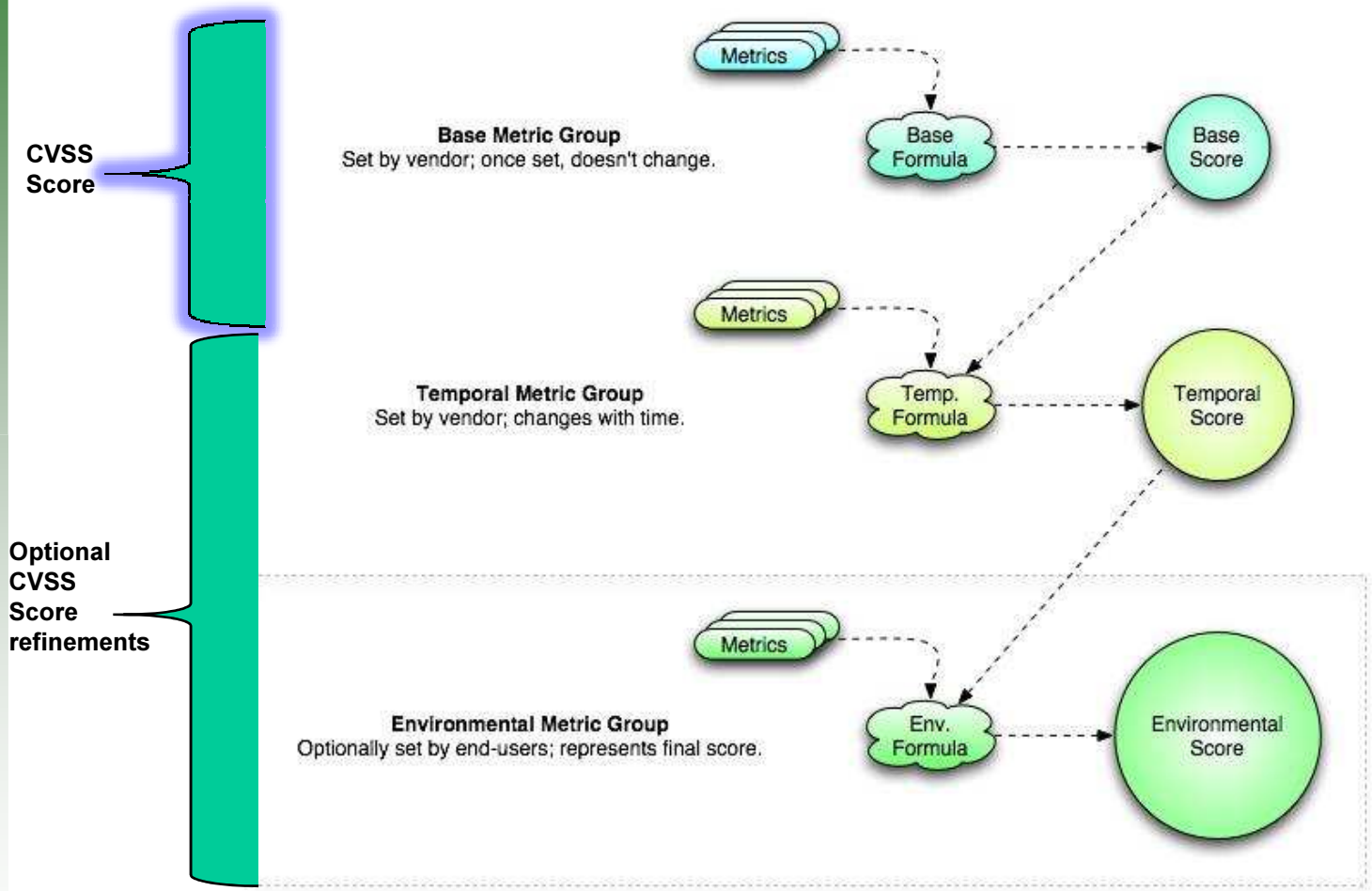
Impact Requirement (IR) based of FIPS 199

- This metric enables the analyst to customize the CVSS score depending on the criticality of the affected IT asset.
- Low (L): Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization
- Medium (M): Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization
- High (H): Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization
- Not Defined (ND): Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric

Scoring and Formulas

- The process of combining metric values
- Base score is the “foundation” and stands alone as the CVSS representation of a vulnerability attributes
  - Modified by Temporal and Environmental metrics
- Base and Temporal scores computed by vendors and coordinators with the intent of being published
- Environmental score optionally computed by end-user / organization

CVSS (Scoring View)



Base Scoring

- Computed by vendors and coordinators
- Combines innate characteristics of the vulnerability
- The base score has the largest bearing on the final score
  - Computed primarily from the Impact Metrics
- Represents vulnerability severity

Temporal Scoring

- Computed by vendors and coordinators
- Modifies the Base Score
- Allows for the introduction of mitigating factors to reduce the score of a vulnerability
- Designed to be re-evaluated at specific intervals as a vulnerability ages
- Represents urgency at specific points in time

Environmental Scoring

- Computed by end users
- Adjusts combined Base-Temporal score
- Should be considered the FINAL score
- Represents a snapshot in time, tailored an environment
- User organizations will use this to prioritize responses within their own environments

Format for publishing Vectors

- Every application or service that uses the Common Vulnerability Scoring System (CVSS) should provide not only the CVSS score - but also a vector describing the components from which the score was calculated.
- This allows end-users to validate score while providing a common set of vulnerability attributes to be disclosed

CVSS Base Vectors

CVSS vectors containing only base metrics take the following form:

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

http://nvd.nist.gov/cvss.cfm?vectorinfov2

Vector definitions

- Example 1: (AV:L/AC:H/Au:N/C:N/I:P/A:C)
  Example 2: (AV:A/AC:L/Au:M/C:C/I:N/A:P)

  Metric: AV = AccessVector (Related exploit range)
  Possible Values: L = Local access, A = Adjacent network, N = Network

  Metric: AC = AccessComplexity (Required attack complexity)
  Possible Values: H = High, M = Medium, L = Low

  Metric: Au = Authentication (Level of authentication needed to exploit)
  Possible Values: M= Requires multiple instances, S= Requires single instance, N= None required

  Metric: C = ConfImpact (Confidentiality impact)
  Possible Values: N = None, P = Partial, C = Complete

  Metric: I = IntegImpact (Integrity impact)
  Possible Values: N = None, P = Partial, C = Complete

  Metric: A = AvailImpact (Availability impact)
  Possible Values: N = None, P = Partial, C = Complete

Example - **CVE-2003-0062**

- Consider CVE-2003-0062: Buffer Overflow in NOD32 Antivirus. 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.

Example - **CVE-2003-0062**

- 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 “None” 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”. Together, these metrics produce a base score of 6.2.
- The base vector for this vulnerability is therefore: AV:L/AC:H/Au:N/C:C/I:C/A:C.

Example - **CVE-2003-0062**

- Partial exploit code has been released, 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 adjust the base score to give a temporal score of 4.9.
- Assuming that confidentiality, integrity, and availability are roughly equally important for the targeted systems, and depending on the values for Collateral Damage Potential and Target Distribution, the environmental score could vary between 0.0 (“None”, “None”) and 7.5 (“High”, “High”). \

application

- **So what does a CVSS Environmental Score of 7.5 for CVE-2003-0062 mean to me?**
  - Your response to 7.5 may be different than mine based on constituency
  - Consistent universal scoring of Base and Temporal categories provides relative severity
  - So far…

0-3	No impact – wait for SP
4-5	Next Patch Cycle
6-7	Within 7 days
7-10	Firedrill

- **Any scoring / normalization of this many variables is going to be a gross generalization**
  - Some subjectivity in evaluating metrics
  - Formulas encode pre-defined values
  - Some things are missed

applications

The Common Vulnerability Scoring System (CVSS) and Its Applicability to US Federal Agency Systems

- NIST IR 7435 is published as final. CVSS provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. This publication defines and describes the CVSS standard, provides advice on performing scoring, and discusses how Federal agencies can incorporate Federal Information Processing Standards (FIPS) 199 impact ratings into their CVSS scores to generate scores that are specifically tailored to particular Federal agency environments.
- For complete article see:
- http://csrc.nist.gov/publications/nistir/ir7435/NISTIR-7435.pdf

applications

CVSS and the Payment Card Industry (PCI)

- In order for private-sector firms to process credit cards, they need to comply with the Payment Card Industry Data Security Standards (PCI DSS). Effective June 2007, the PCI governing body is requiring firms use CVSS in order to determine how vulnerable are their IT systems. The PCI DSS is available:
- https://www.pcisecuritystandards.org/pdfs/pci_dss_technical_and_operational_requirements_for_appnroved_scanning_vendors_ASVs_v1-1.pdf

applications

- Generally, to be considered compliant, a component must not contain any vulnerability that has been assigned a CVSS base score equal to or higher than 4.0

The following exceptions or clarifications apply:

- A component must be considered non-compliant if the installed SSL version is limited to Version 2.0, or older. SSL must be a more recent version than 2.0.
- Vulnerabilities or mis-configurations that may lead to DoS should not be taken into consideration

Final comments

- The authors recognize that many other metrics could have been included in CVSS. We also realize that no one scoring system will fit everyone's needs perfectly.
- The particular metrics used in CVSS were identified as the best compromise between completeness, ease-of-use and accuracy. They represent the cumulative experience of the CVSS Special Interest Group members as well as extensive testing of real-world vulnerabilities in end-user environments.
- As CVSS matures, these metrics may expand or adjust, making the scoring even more accurate, flexible and representative of modern vulnerabilities and their risks.

Links

- FIRST: http://www.first.org/cvss
- NIST: http://nvd.nist.gov/cvss.cfm?showall and http://nvd.nist.gov/cvss.cfm?calculator.
- Cisco MySDN: http://tools.cisco.com/MySDN/Intelligence/home.x
- Xforce http://xforce.iss.net/xforce/alerts/alerts
- FIPS 199 http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

questions?

Additional slides - Changes

- Add additional granularity into AccessComplexity. We would change Add an additional "medium" field to help better describe the complexity needed to use a particular exploit.
- Move the “impact bias” metric from the base scoring to the environmental scoring. In its new form within the environmental section, it would enable end users to declare which CIA attribute is most important to them in the context of a particular vulnerability.
- Explicitly add to the CVSS standard that vulnerabilities that give root level access should be CIA of all ‘complete’ and vulnerabilities that give user level access to the OS or general access to an application should be CIA of ‘partial’.

Additional slides - Changes

- Reword the access complexity definition to make it clear that the metric is dealing with how easy it is to execute an attack once exploit code exists as opposed to measuring how hard it is to write exploit code.
- Add a new option to the access vector metric for vulnerabilities that are accessible only over a local network and rename existing metrics appropriately. The new option is called “Local network accessible” (sometimes referred to as “adjacent” in the CVSS SIG email list discussions).
- Change the collateral damage metric so it may also measure economic loss of productivity or revenue.

Additional slides - Changes

- Modification of Authentication Metrics to the following: Requires no authentication: Requires a single instance of authentication: Requires multiple instances of authentication
- In order to make scoring consistent and to focus scoring on the software that is directly vulnerable, the CVSS documentation should be updated to reflect that vulnerabilities should always be scored with respect to the impact on the vulnerable service.
- To make scoring more consistent, the CVSS documentation should be updated to indicate that vulnerabilities should be scored based on the privileges that are most often used (sometimes referred to as “most probably”) for the application. This does not necessarily reflect the best practice for the application, especially for client applications, which are often run with root-level privileges. If it is not clear what privileges are most often used for an application, analysts should assume the default configuration

Additional slides - Changes

- For multi vector vulnerabilities : To make scoring more consistent, the CVSS documentation should be updated to indicate that analysts should generate a score for each approach to exploitation and then assign the vulnerability the highest of the scores. If the highest score is shared by multiple approaches, then analysts should compare those approaches and select the one that is most likely to be used.

Additional slides - Changes

- NIST statisticians in conjunction the SIG went over the CVSS formulas to identify and fix some of the issues with the existing formulas for example:
  - lack of diversity of scores,
  - multiplicative equations issues,
  - high score scarcity problems
  - many to one scoring issues.

Summary

- CVSS is a way to talk about vulnerability severity
- New
- Open
- Simple
- Objective
- Comprehensive