



Common Vulnerability
Scoring System (CVSS)
& Vulnerability
Disclosure Framework
(VDF)

Gaus <gaus@cisco.com>

“Ownership” of CVSS

- **CVSS is now owned by FIRST**
- **Special Interest Group is being formed**
- **The purpose is to improve CVSS**



NATIONAL INFRASTRUCTURE ADVISORY Council

- **Thirty CEOs (or equivalent) who advise the President of the United States regarding the security of information systems affecting the critical infrastructure**
- **Issue “recommendations” that, once accepted, have authority only over the executive branch of US government, but can be adopted by others in US and elsewhere**
- **NIAC members sponsor working groups of subject-matter experts to develop drafts**



Vulnerability Disclosure Framework

- Provides broad structure with alternatives for improving handling and communication of vulnerabilities and associated information
- Does not proscribe policy! Makes reader aware of possible paths and consequences
- Does proscribe improvements to sharing info
 - Defines roles: Discoverer, Researcher, Coordinator, Vendor, Consumer
 - “Slash Security” page simplifies reporting
- Identifies other challenges



Why Scoring Was Left Out Of The VDF

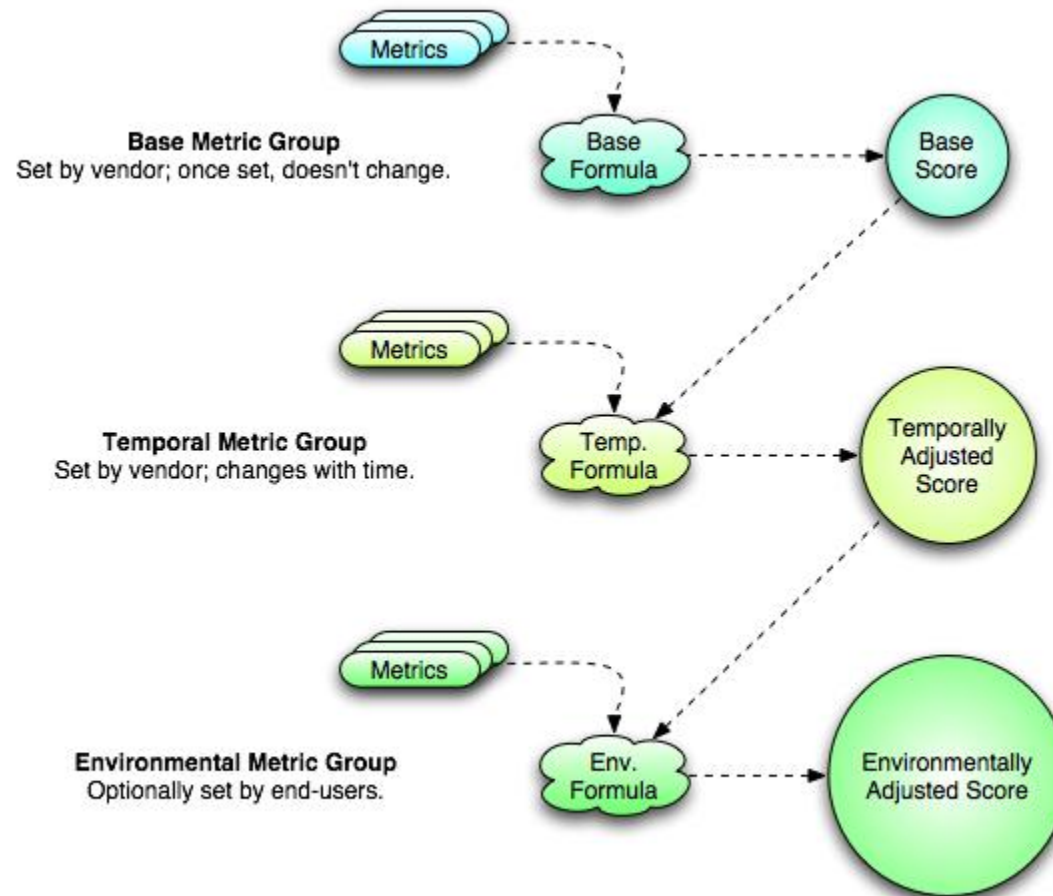
- Originally intended to provide uniform scoring function so vulnerabilities could be compared and prioritized consistently
- (6 different methods) X (9 different experts) X (12 different vulnerabilities) = 648 answers!
- Conclusion: Existing scoring methods were hopelessly subjective, can't be compared
- Scoring was reassigned as a follow-on task
- Same working group produced the Common Vulnerability Scoring System



CVSS At A Glance

- **Scoring is based on a variety of metrics**
- **Grouped into three broad categories**
 - Base: immutable features of core vulnerability
 - Temporal: evolve over lifetime of vulnerability
 - Environmental: how vulnerability affects a specific installation
- **Scoring can be limited to any or all of above**
- **Metrics are defined to ensure consistency**

CVSS Process Diagram



Working Group Results

- **Draft CVSS reviewed for implementation by 10 NIAC members' IT organizations**
- **Achieved 70% to 80% commonality!**
- **Deviations mostly attributed to ambiguity in documentation and textual descriptions**
- **Project continues to evolve, futures include**
 - Development of multi-platform scoring tools
 - Feedback and next generation versions

Things that are not addressed by CVSS

- Potential threats
- Combined vulnerabilities
- Global exposure scoring

Base Score

- Expected to be set by vendor or originator
- Represents *innate characteristics* of the vuln
- Has the largest effect on the final score
- Once set, not expected to change
- Computed from “the big three” of
 - Confidentiality
 - Integrity
 - Availability
- Indicates general severity



Temporal Score

- **Modifies the Base Score**
- **Represents changes over time**
- **Introduces mitigating factors that typically reduce the final score of a vulnerability**
- **Expected to be re-evaluated periodically**
- **Indicates urgency at any point in time**
- **Expected to be set by vendor or coordinators**



Questions related to Temporal Score

- **Would you like to be notified when it changes?**
- **How would you like to be notified? Mail? RSS? Any other method?**

Environmental Score

- **Modifies combined Base+Temporal Score**
- **Represents vulnerability in an installation**
- **Addresses deployment and configuration**
- **Produces the Final Score**
- **Can only be defined by consumer or possibly coordinator**
- **Might be defined by vendor with complete knowledge of all deployments**
- **Indicates overall priority**



Metrics in a Base Score

- **Access Vector: local or remote exploit**
- **Access Complexity: difficulty of exploit**
- **Authentication: need to be logged in?**
- **Confidentiality Impact**
- **Integrity Impact**
- **Availability Impact**
- **Impact Bias: which of the previous three is more important if more than one is used?**
- **Round to 1 digit in 10**

Access Vector

- **Exploitable locally or remotely?**
- **Local Access: attacker must have physical or authenticated login access to the target**
- **NOTE: “remote login” is not “remote access”**
- **For example, a vuln in “passwd” is probably “local”, but a vuln in SSH exploitable via the net without authentication is “remote”**

Access Complexity

- **How difficult is it to stage this attack?**
- **High: one or more other conditions required**
- **Low: no special additional requirements**
- **For example, a buffer overflow in a service needs only the target and a malicious packet, versus an e-mail vuln that requires receiving a message and then clicking on it**

Authentication

- Does the attacker have to be authenticated?
- NOTE: not the same as the Access Vector
- Apply Authentication only after the attacker has logged in per the Access Vector in cases where Local Access is already required

Confidentiality Impact

- **As usual, describes unauthorized disclosure**
- **None: should be self-evident**
- **Partial: “considerable” amount of disclosure but the attacker has no control over what can be taken, or the attack is otherwise limited**
- **Complete: all information is revealed**

Integrity Impact

- **Guaranteed veracity of information**
- **None: also self-evident**
- **Partial: attacker does not control what can be modified or scope of modifications is limited**
- **Complete: total loss of system integrity**

Availability Impact

- **Accessibility of services, typically a DoS**
- **None: still self-evident**
- **Partial: degraded service**
- **Complete: total shutdown**

Impact Bias

- **Confidentiality, Integrity, and Availability are separately more important than the others for specific types of systems**
- **For example, a vulnerability affecting the confidentiality of an encrypting file system is far more severe than if it affected availability**
- **Impact Bias metric provides emphasis**
- **Determined once, but calculated and included after each of the 3 previous metrics**

Metrics in a Temporal Score

- **Exploitability:** Is brilliance required or can anybody succeed with this vulnerability?
- **Remediation Level:** What can be done now to mitigate this vulnerability?
- **Report Confidence:** How well can a specific report be trusted?
- **Round to 1 digit of the product of this result and the previously calculated Base Score**

Exploitability

- **Unproven:** Theoretical, no written PoC code
- **Proof of Concept:** Nonfunctional PoC written
- **Functional:** PoC works for most situations
- **High:** Available PoC code works in all cases

Remediation Level

- **Official Fix:** Vendor has provided a solution
- **Temporary Fix:** Vendor has temporary patch
- **Workaround:** In lieu of vendor's solution
- **Unavailable:** Solution is impossible to apply

Report Confidence

- **How accurate are the statements about the existence of the vulnerability and solutions?**
- **Unconfirmed: Rumors or conflicting reports**
- **Uncorroborated: Several unofficial reports**
- **Confirmed: Vendor reports on own product**



Metrics in an Environmental Score

- **Collateral Damage Potential: what is the second-order impact on assets?**
- **Target Distribution: how many systems are vulnerable in this particular environment?**
- **This metric can reduce score to zero if the consumer is not running the vulnerable code**
- **Round to 1 digit of product of this score with previous scores brought forward**
- **This the FINAL SCORE for the vulnerability**



Collateral Damage Potential

- **Measures other tangible & intangible losses**
- **None: No damage to other systems**
- **Low: Light damage to other systems**
- **Medium: Significant collateral damage**
- **High: Catastrophic collateral losses**



Target Distribution

- **None:** Almost none, maybe in a laboratory
- **Low:** Small-scale targets exist, <15%
- **Medium:** Significant at-risk systems, <50%
- **High:** Large-scale risk of vulnerable systems

Implementing CVSS Version 1.0

- **Currently only an Excel spreadsheet**
- **Plans for various implementations**
 - PalmOS with Windows/Unix conduit
 - Web-based form
 - Database backend? RSS feed?
- **Solutions should include CVSS version to allow for later improvements without conflict**

Comments and Questions

- **Final report at**

- <http://first.org/cvss/cvss-dhs-12-02-04.pdf>

- **What's missing or "wrong"?**

- Not enough vendor orientation

- No specification for timestamps, transport

- Interface with OVAL, CVE, other descriptors

- Defending simplicity revisited

- **Other issues?**