Common Weakness Scoring System (CWSS)

Steve Christey

February 28, 2011

cwss@mitre.org

http://cwe.mitre.org/cwss

Outline

- Poll
- Problem Statement
- Past/Present Efforts
- CVSS
- Vignettes
- CWSS 0.2 Metric Groups and Factors
- Scoring Examples
- Future Work and Contributions

Poll

- How many people want to influence the development of CWSS to suit their needs?
- How many people are hoping that CWSS can be used to overcome limitations of CVSS?

The Problem

- **In the process of discovering new vulnerabilities, automated and human analysis will find weaknesses**
  - Everyone scores weaknesses differently
- **Not all reported weaknesses necessarily indicate a vulnerability**
- **Hundreds or thousands of weaknesses could be reported for a single software package**
- **Weaknesses can be treated as an entire class of problem to eradicate, independent of any specific bug in a specific software product**
- **Weakness prioritization may vary according to a variety of contexts and threat environments**

Beginnings

- **CWSS Kickoff Meeting – Oct 24, 2008**
- **Briefing to SwA Working Groups – July 2010**
- **Start with CVSS**
  - Try to address some of CVSS' limitations
  - Examine other metrics
- **Environment / Context is critical**
  - Business/mission priorities, how SW is deployed, …
- **Ideally supports tools and humans**
- **Must be stable/predictable even when there is limited information**

- **Public white paper, December 2010**

CWSS Kickoff Meeting

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **October 24, 2008**
  - ~12 organizations represented
  - Primarily tool vendors and consultants
- **Main topics**
  - What weakness-scoring approaches are currently used?
  - How much should CWSS borrow from previous efforts like CVSS?
  - How closely should CWSS and CWE be affiliated?
  - Who are the stakeholders?
  - What are some core elements to capture in CWSS?
- **Main conclusions**
  - CWSS is independent of CVSS
  - Complete information is rarely available
  - Multiple stakeholders are involved
  - Distinct usage models
  - Line between “Weakness” and “Vulnerability” is fuzzy

Related Scoring Efforts (from 2008 kickoff)

- **Veracode – Confidentiality, Integrity, Availability analysis of CWE weaknesses**
  - Independence from analysis method
- **Cigital – feasibility study of CVSS**
  - Some attributes like “Target Distribution” didn’t fit well
  - Added more granularity to some attributes
  - Recommended polynomial scoring
  - Important to model the distinction between likelihood and impact

Related Scoring Efforts (Continued)

- **Cenzic – HARM (Hailstorm Application Risk Metric)**
  - Quantitative score focused on black box analysis of web applications
  - Goal: scalable focus on remediation
  - 4 impact areas: browser, session, web app, server
  - Benefit: “easily consumable”
- **CERT/SEI – scoring of C Secure Coding Rules**
  - FMECA (ISO standard) metric: Severity, Likelihood (of leading to vuln), Remediation Cost

2009 CWE Top 25 - Informal Criteria

- Attack frequency
- Consequences
- Weakness prevalence
- Ease of detection
- Language/platform independence when possible
- Others considered: remediation cost, amount of public knowledge, likelihood of future increase

2010 SANS/CWE Top 25

- **Real-world, raw data is still very difficult to find**
- **Prioritized items based on "Prevalence" and "Importance" (4 values each)**
- **25 participating organizations evaluated 41 nominee CWE entries**
  - Developers, researchers, educators
- **Focus profiles allowed alternate ranking**
  - E.g. educational emphasis, importance to software consumers

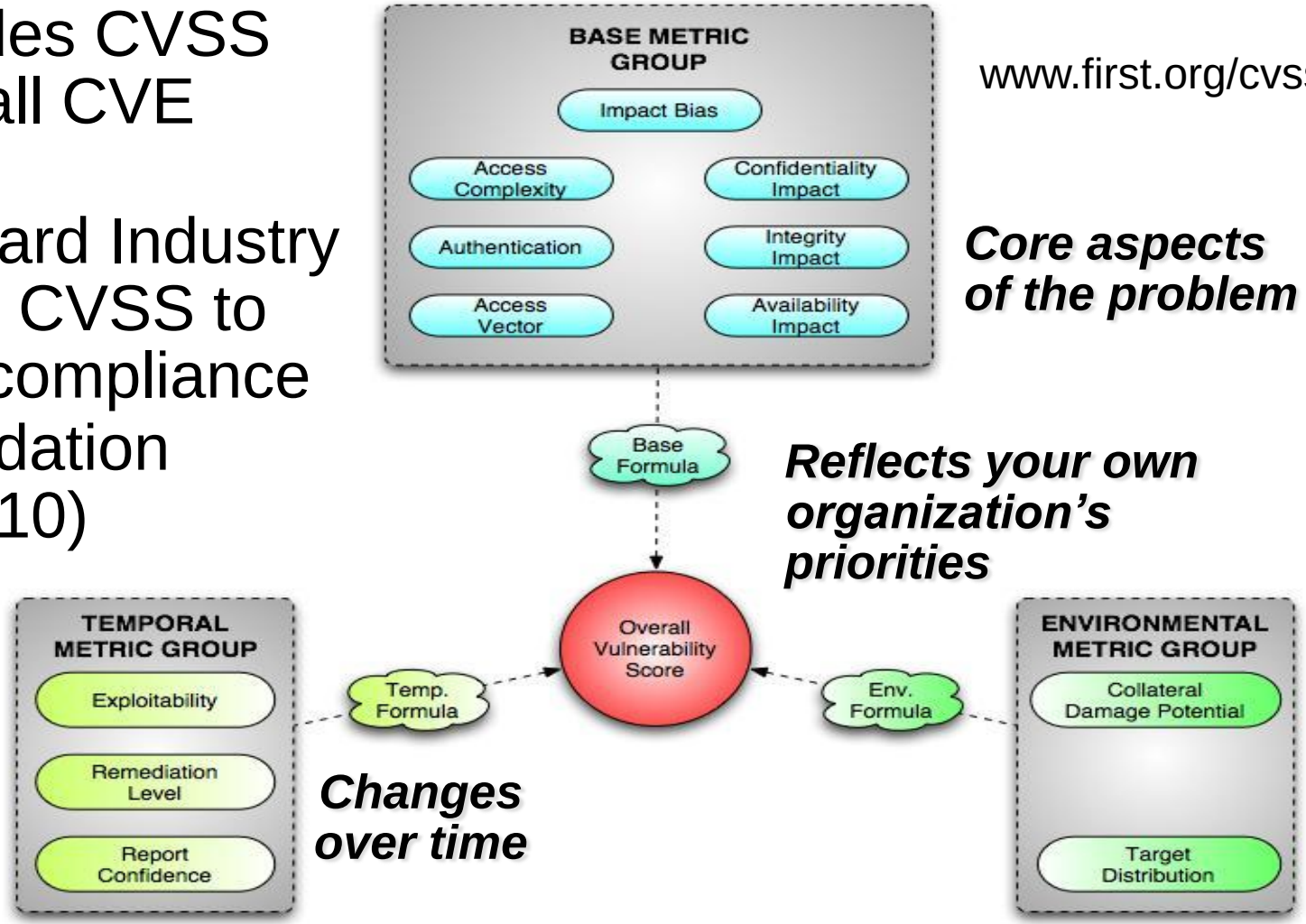
http://cwe.mitre.org/top25/#AppendixC

2010 OWASP Top Ten - Factors

- **Ease of Exploit**
- **Prevalence**
- **Detectability**
- **Technical Impact**

Common Vulnerability Scoring System

- NVD provides CVSS scores for all CVE identifiers
- Payment Card Industry (PCI) using CVSS to determine compliance
- Formal validation program (~10)



www.first.org/cvss/

Core aspects of the problem

Reflects your own organization's priorities

Changes over time

Some CVSS Strengths

- Relative Simplicity
- Repeatability (within some epsilon)
- Efficient representation (vector)
- Widely adopted
- “Good enough” for non-expert admins

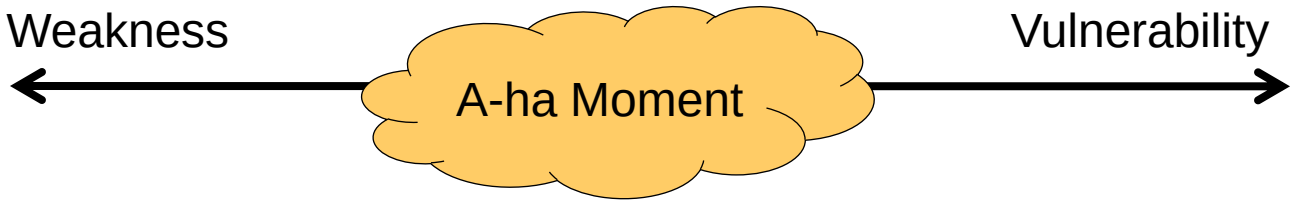
Why Not CVSS?

- **Focuses on impact to system**
  - The “Oracle” problem: even with an entire DB compromise, can’t exceed 7.0 score because it’s not running with admin privileges
- **Requires good documentation**
- **Not granular enough for expert consumers**
  - E.g. confidentiality/integrity/availability
- **Doesn’t handle insufficient information well**
  - The “Missing Oracle” problem: published vulnerabilities rarely have complete information, especially from vendors who don’t like to publish details
- **Temporal/Environmental aspects not well-tested**

CVSS assumes that you’ve already proven a vulnerability exists

Weakness vs. Vulnerability

- **Weakness: a mistake in software's design, implementation, etc. that *might* result in a vulnerability:**
  - *if* an attacker can reach the affected code
  - *if* an attacker can manipulate input to create unexpected results
  - *if* those results violate the intended security policy
- **A weakness may lead to a:**
  - Bug (unintended functionality where the only "victim" is the "attacker")
  - Feature (intended functionality)
  - Vulnerability (unintended functionality where the attacker is not the victim)
- **CWSS is often intended for use BEFORE it is clear whether the weakness contributes to a vulnerability**



Some Potential Stakeholders for CWSS

- **Software developers/programmers**
  - "We'll concentrate on what we can afford to fix, or what our worst problems are"
- **Software project managers**
- **SW acquirers**
  - Adaptation of PCI DSS: "The purchased software shall not have any outstanding weaknesses greater than CWSS score 7.0, as determined by methods X and Y."
- **Code analysis vendors – tools and services**
- **Vulnerability researchers**
- **Secure development advocates**
- **CIO's and CSO's**
- **System administrators**
- **Application users**

Developer Use Case: Example

HS SEDI
Homeland Security Systems Engineering and Development Institute

- ***I ran a tool that gave me hundreds of results. Which issues should I address, based on:***
  - My company's goals
  - Compliance requirements
  - Customer expectations
  - Best current practices
  - Amount of time before next release
  - My company's willingness to accept the risk of not fixing

Design Requirements

- **Account for incomplete information**
- **Scalable and, where possible, automatable**
- **Flexible**
- **Integrate (or at least indirectly support) environmental/business/mission considerations**
- **Support for multiple scoring modes**
  - General vs. targeted
- **Stakeholder needs must be well-understood**
- **Avoid unnecessary complexity**

CWSS Support for Multiple Scoring Modes

- **Targeted: score a weakness based on its occurrence within a specific software package**
  - How to score weakness X in line 1234 of vuln.c?
  - Won't always have complete information
  - Operational environment, business impact are important
- **General: score weaknesses based on their general occurrence in software**
  - In general, how bad are buffer overflows versus memory leaks?
  - Won't always be correct for a specific instance
  - "Top 25," and other lists
  - But even "buffer overflow" risk varies widely, e.g. OS-level overflow protection mechanisms
- **Vignette-oriented scoring: Consider priorities of a particular community or product type**

Vignettes

- Scenarios that occur within a particular business domain
- Identify one or more technology archetypes, linked together to form a business function
- Business Value Context (BVC): Describes importance of Confidentiality/Integrity/Availability in light of business needs
- Technical Impact Scorecard: links the technical results of weakness exploitation with the business impact

http://cwe.mitre.org/cwss/index.html#model

http://cwe.mitre.org/cwss/vignettes.html

Business Domains – a Sample

HS SEDI
Homeland Security Systems Engineering and Development Institute

Domain	Description
E-Commerce	The use of the Internet or other computer networks for the sale of products and services, typically using the WWW.
Finance	Financial industry.
Health Care IT	Medical encoding and billing, Critical or emergency care, medical devices - "implantable" or "partially embedded" in humans, as well as usage in clinic or hospital environments ("patient care" devices.)
Smart Grid	An electricity network through a large region, using digital technology for monitoring or control.
Telecommuting & Teleworking	Support for employees to have remote access to internal business networks and capabilities.
eVoting	Electronic voting systems, as used within state-run elections, shareholder meetings, etc.

Archetypes – a Sample

- Often used in different domains
- Linked together to address a particular area of functionality
  - Database
  - General-purpose OS
  - Web browser
  - Web server
  - Programmable Logic Controller
  - Smartphone
  - Virtualized OS
  - Service-Oriented Architecture

Business Value Context (BVC)

- Identifies critical assets and security concerns
- Links Technical Impacts (derived from CWE weaknesses) with business implications
- More fine-grained model than CIA

- **Modify memory**
- **Read memory**
- **Modify files or directories**
- **Read files or directories**
- **Modify application data**
- **Read application data**
- **DoS: crash / exit / restart**
- **DoS: amplification**
- **DoS: instability**
- **DoS: resource consumption (CPU)**
- **DoS: resource consumption (memory)**
- **DoS: resource consumption (other)**
- **Execute unauthorized code or commands**
- **Gain privileges / assume identity**
- **Bypass protection mechanism**
- **Hide activities**

Vignette: Web-based Retail Provider

- **Business Domain: E-Commerce**
- **Internet-facing, E-commerce provider of retail goods or services**
- **Data-centric - PII, credit card numbers, order history**
- **Archetypes: Database, Web client/server, General-purpose OS**
- **Business Value Context (BVC):**
  - Confidentiality essential from a financial PII perspective
  - Identity PII usually less important.
  - PCI compliance a factor.
  - Security incidents might have organizational impacts including financial loss, legal liability, compliance/regulatory concerns, and reputation/brand damage.

Vignette: Smart Meters (Smart Grid)

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **Business Domain: Smart Grid**
- **Meter that records electrical consumption and communicates this information to the supplier on a regular basis.**
- **Archetypes: Web Applications, Real-Time Embedded System, Process Control System, End-point Computing Device**
- **Business Value Context (BVC):**
  - Confidentiality of customer energy usage statistics is important - could be used for marketing or illegal purposes. For example, hourly usage statistics could be useful for monitoring activities.
  - Integrity of metering data is important because of the financial impact on stakeholders (consumers manipulating energy costs).
  - Availability typically is not needed for real-time; other avenues exist (e.g. site visit) if communications are disrupted.

More Vignette Examples

HS SEDI
Homeland Security Systems Engineering and Development Institute

Domain	Vignette	Description
Finance	Financial Trading	Financial trading system supporting high-volume, high-speed transactions. N-tier distributed, using J2EE and supporting frameworks.
Health Care IT	Medical Billing	Medical encoding and billing. Data used includes Electronic Health Records (EHR), financial management, interactions with insurance companies.
Health Care IT	Human Medical Devices	Medical devices - "implantable" or "partially embedded" in humans. Includes items such as pacemakers and automatic drug delivery.
eVoting	State Election DRE	State Election Administration using DRE (Direct Recording Election) machines.
Human Resources	Employee Compensation	Product for managing employee salary and bonuses. PII includes salary, financial transaction (e.g. for direct deposit), social security number, home address, etc.

http://cwe.mitre.org/cwss/vignettes.html

Technical Impact Scorecard Example: Web-based Retail Provider

Technical Impact	Subscore	Business Value Impact
Hide activities	3	Inability to identify source of attack; Cannot obtain sufficient evidence for criminal prosecution.
DoS: resource consumption (CPU)	3	Customers experience delays in reaching site; delays in order placement and resulting financial loss.
Modify application data	8	Modify or delete customer order status and pricing, contact information, inventory tracking, customer credit card numbers, cryptographic keys and passwords (hopefully encrypted).
Read application data	5	Read customer credit card numbers, contact information, order status, cryptographic keys and passwords (hopefully encrypted). Read application configuration.

These subscores are demonstrative.

Technical Impact Subscore Comparison across Vignettes

Technical Impact	Base Subscore (Max)	Base Subscore (Ave)	Financial Trading	Human Resources	Retail WWW	SCADA / Chemical	Social Network
Modify files or directories	8	7.80	8	8	8	7	8
DoS: crash	10	7.20	10	6	4	7	9
Read application data	8	6.20	8	7	8	4	4
Hide activities	3	3.00	**3**	**3**	**3**	**3**	**3**

These subscores are demonstrative, and subject to change based on community feedback.

Technical Impacts for Individual CWE Entries - Example

- **Retail WWW Vignette**
- **CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**
  - Read application data (subscore: 8)
  - Bypass protection mechanism (subscore: 7)
  - Modify application data (subscore: 8)
  - *Maximum impact score: 8*
- **CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')**
  - Execute unauthorized code or commands (subscore: 10)
  - DoS: crash / exit / restart (subscore: 4)
  - *Maximum impact score: 10*

CWSS 0.2 Metric Groups

HS SEDI
Homeland Security Systems Engineering and Development Institute

Base Finding Group

- Impact
- Finding Confidence
- Remediation Cost
- Prevalence

All factors support "Unknown" and "Not Applicable"

Attack Surface Group

- Universality
- Access Vector
- Required Privilege Level
- Authentication Strength
- Authentication Instances

Exploitability Group

- Likelihood of Discovery
- Likelihood of Exploit
- Level of Interaction
- Internal Control Effectiveness
- External Control Effectiveness

Comparison to CVSS:
http://cwe.mitre.org/cwss/index.html#Comparison

CWSS 0.2 Scoring

HS SEDI
Homeland Security Systems Engineering and Development Institute

Impact * Prevalence * Attackability * Confidence * RemediationCost

*** OR ***

Base Finding * Attack Surface * Exploitability

- **Prevalence is 1.0 in "targeted" scoring for weaknesses found in a specific application**

Common Scoring Adjustments

- **An initial weakness finding might have "Unknown" as the value for many factors**
  - E.g. authentication strength, required privilege level
  - Human analysis may update the values for all findings in a single component/executable
- **Individual factors might be informed by Business Value Context**
  - Customers don't directly care about remediation cost
  - High assurance and report finding confidence
  - Insider threat concerns and required privilege level
- **Scores for weaknesses may vary across vignettes or business domains**
  - But why would you compare the "weakness surface" of a SCADA system with that of a mobile phone app?

CWSS 0.2 Factors

- **“Unknown”/“Not Applicable” values supported everywhere**
- **Impact (I)**
- **Finding Confidence (FC)**
  - Proven True, Proven Locally True, Proven False
- **Remediation Cost (RC)**
  - Systemic, Localized, Minimal
- **Universality (UN)**
  - All, Moderate, Rare, Potentially Reachable
- **Access Vector (AV)**
  - Remote, Local, Network-adjacent, Physical
- **Required Privilege Level (RP)**
  - None, Guest, User, Partially Privileged User, Administrator

CWSS 0.2 Factors - Continued

- **Authentication Strength (AS)**
  - High, Medium, Low, None
- **Authentication Instances (AI)**
  - None, One, Multiple
- **Likelihood of Discovery (DI)**
  - High, Medium, Low
- **Likelihood of Exploit (EX)**
  - High, Medium, Low
- **Level of Interaction (IN)**
  - Automated, Limited, Moderate, Opportunistic, High
- **Internal Control Effectiveness (IC)**
  - None, Limited, Moderate, Complete
- **External Control Effectiveness (EC)**
  - None, Limited, Moderate, Complete

Finding Confidence (FC)

- **The confidence that the reported issue:**
  1. is a weakness, and
  2. can be triggered or utilized by an attacker.

Value	Code	Weight	Notes
Proven True	T	1.0	the weakness is reachable by the attacker.
Proven Locally True	LT	0.8	the weakness occurs within an individual function or component whose design relies on safe invocation of that function, but attacker reachability to that function is unknown or not present. For example, a utility function constructs a DB query without encoding its inputs, but it is only called with constant strings.
Proven False	F	0.0	the finding is erroneous (i.e. the finding is a false positive and there is no weakness), and/or there is no possible attacker role.

This factor could be continuous instead of discrete.

Remediation Cost (RC)

- **The cost (and/or amount of effort) to remediate the weakness so that it no longer poses a security risk to the software.**

Value	Code	Weight	Notes
Systemic	Sys	0.1	the remediation likely requires modifications to design or architecture
Localized	Loc	0.5	the remediation may require a number of modifications, but these are localized to a small number of components or source files
Minimal	Min	1.0	the remediation can be applied by modifying a relatively small number of lines of code
Not Applicable	NA	1.0	Software customers, and/or high assurance environments, might not care about remediation cost.

This factor could be continuous instead of discrete.

Prevalence (P)

- **The cost (and/or amount of effort) to remediate the weakness so that it no longer poses a security risk to the software.**

Value	Code	Weight	Notes
Widespread	W	1.0	Seen throughout software
High	H	0.9	
Common	C	0.5	
Limited	L	0.2	
Not Applicable	NA	1.0	With targeted scoring, prevalence is typically “NA” because the number of findings of CWSS would be accounted for in aggregate scores.

This factor could be continuous instead of discrete.

Universality (UN)

- **Whether the weakness is present in all deployable instances of the software, or if it is limited to a subset of platforms and/or configurations.**

Value	Code	Weight	Notes
All	All	1.0	Present in all platforms or configurations
Moderate	Mod	0.9	Present in common platforms or configurations
Rare	Rare	0.5	Only present in rare platforms or configurations
Potentially Reachable	*Pot*	*0.2*	*Potentially reachable, but all code paths are currently safe, and/or the weakness is in dead code*
Not Applicable	NA	1.0	

Access Vector (AV)

- **Whether the weakness is present in all deployable instances of the software, or if it is limited to a subset of platforms and/or configurations.**

Value	Code	Weight	Notes
Network	N	1.0	weakness is bound to the network stack and the attacker does not require local network access or local access (“remotely exploitable”)
Local	L	0.5	The attacker must have an interactive, local (shell) account that interfaces directly with the underlying operating system.
Adjacent Network	A	0.9	An attacker must have access to either the broadcast or collision domain of the vulnerable software. Examples of local networks include local IP subnet, Bluetooth, IEEE 802.11, and local Ethernet segment.
Physical	P	0.2	The attacker must have physical access to the system that the software runs on, or otherwise able to interact with the system via interfaces such as USB, CD, keyboard, mouse, etc.

Required Privilege Level (RP)

- **The level of privileges required for an entity to reach the code/functionality that contains the weakness.**

Value	Code	Weight	Notes
None	N	1.0	No privileges are required. For example, a web-based search engine may not require any privileges for an entity to enter a search term and view results.
Guest	G	0.9	The entity has limited or "guest" privileges that can significantly restrict allowed activities; the entity might be able to register or create a new account without any special requirements or proof of identity. Example: blog comments.
Regular User	RU	0.7	The entity is a regular user who has no special privileges.
Partially-Privileged User	P	0.4	The entity is a valid user with some special privileges, but not equivalent to an administrator. For example: backup privileges.
Admin	NA	0.1	The entity has administrator, root, SYSTEM, or equivalent privileges that imply full control.

Authentication Strength (AS)

- **The strength of the authentication routine that protects the code/functionality that contains the weakness.**

Value	Code	Weight	Notes
High	H	0.7	The weakness requires authentication using the strongest methods available to tie the entity to a human identity, such as biometrics or one-time passwords.
Medium	M	0.8	The weakness requires authentication using strong methods that can establish identity but may be subject to spoofing, such as the use of certificates from untrusted authorities.
Low	L	0.9	The weakness requires minimal authentication, such as a password.
None	N	1.0	The weakness does not require any authentication at all.

Authentication Instances (AI)

- **The number of distinct instances of authentication that an entity must perform to reach the weakness.**

Value	Code	Weight	Notes
None	H	1.0	
Single	M	0.8	
Multiple	L	0.5	
Not Applicable	NA	1.0	This might not be applicable in a BVC with high assurance requirements.

Likelihood of Discovery (DI)

- **The likelihood that an attacker can discover the weakness.**

Value	Code	Weight	Notes
High	H	1.0	
Medium	M	0.6	
Low	L	0.2	
Not Applicable	NA	1.0	This might not be applicable in a BVC with high assurance requirements.

This factor could be continuous instead of discrete.

Likelihood of Exploit (EX)

- **The likelihood that, if the weakness is discovered, an attacker with the required privileges/authentication/access would exploit it.**

Value	Code	Weight	Notes
High	H	1.0	
Medium	M	0.6	
Low	L	0.2	
None	N	0.0	Not clear when this value should be used, if ever.
Not Applicable	NA	1.0	This might not be applicable in a BVC with high assurance requirements.

This factor could be continuous instead of discrete.

Level of Interaction (IN)

- **The actions that are required by the human victim(s) to enable a successful attack to take place.**

Value	Code	Weight	Notes
Automated	Aut	1.0	No human interaction is required.
Limited / Typical	Ltd	0.9	The attacker must convince the user to perform an action that is common or regarded as "normal" within typical product operation. For example, clicking on a link in a web page, or previewing the body of an email, is common behavior.
Moderate	L	0.8	The attacker must convince the user to perform an action that might appear suspicious to a cautious, knowledgeable user. For example: the user has to accept a warning that suggests the attacker's payload might contain dangerous content.
Opportunistic	N	0.3	The attacker cannot directly control or influence the victim, and can only passively capitalize on mistakes or actions of others.
High	High	0.1	A large amount of social engineering is required, possibly including ignorance or negligence on the part of the victim.
No Interaction	NI	0.0	There is no interaction possible, not even opportunistically; this finding is a “bug” that does not lead to a vulnerability.

Internal Control Effectiveness (IC)

- **The ability of an explicitly-built control, protection mechanism, or mitigation to render the weakness unable to be exploited by an attacker.**

Value	Code	Weight	Notes
None	N	1.0	No controls exist.
Limited	L	0.9	There are simplistic methods or accidental restrictions that might prevent a casual attacker from exploiting the issue.
Moderate	M	0.7	The protection mechanism is commonly used but has known limitations that might be bypassed with some effort by a knowledgeable attacker. For example, the use of HTML entity encoding to prevent XSS attacks may be bypassed when the output is placed into another context such as a Cascading Style Sheet or HTML tag attribute.
Defense-in-Depth	D	0.5	The control does not specifically protect against exploitation of the weakness, but reduces the impact when a successful attack is launched.
Best-Available	B	0.3	The control follows best current practices, although it may have some limitations that can be overcome by a skilled, determined attacker, possibly requiring the presence of other weaknesses. Example: the double-submit method for CSRF protection.
Complete	C	0.0	The control is completely effective against the weakness, i.e., there is no bug or vulnerability, and no adverse consequence of exploiting the issue.

External Control Effectiveness (EC)

- **The ability of a software-external control, protection mechanism, or mitigation to render the weakness unable to be exploited by an attacker (e.g. ASLR, WAF).**

Value	Code	Weight	Notes
None	N	1.0	No controls exist.
Limited	L	0.9	There are simplistic methods or accidental restrictions that might prevent a casual attacker from exploiting the issue.
Moderate	M	0.7	The protection mechanism is commonly used but has known limitations that might be bypassed with some effort by a knowledgeable attacker.
Defense-in-Depth	D	0.5	The control does not specifically protect against exploitation of the weakness, but reduces the impact when a successful attack is launched. Example: ASLR.
Best-Available	B	0.3	The control follows best current practices, although it may have some limitations that can be overcome by a skilled, determined attacker, possibly requiring the presence of other weaknesses.
Complete	C	0.0	The control is completely effective against the weakness, i.e., there is no bug or vulnerability, and no adverse consequence of exploiting the issue. Example: sandbox environment to prevent path traversal.

Example Scoring - 1

HS SEDI
Homeland Security Systems Engineering and Development Institute

Base Finding Group	Attack Surface Group	Exploitability Group
•I:0.8 •FC:T -> 1.0 •RC:Min -> 1.0 •P:NA -> 1.0	•UN:All -> 1.0 •AV:N -> 1.0 •RP:G -> 0.9 •AS:N -> 1.0 •AI:N -> 1.0	•DI:H -> 1.0 •EX:H -> 1.0 •IN:Ltd -> 0.9 •IC:N -> 1.0 •EC:N -> 1.0
0.8	**0.9**	**0.9**

100 x 0.8 x 0.9 x 0.9 = 64.8

Impact = 8/10, Required Privileges = Guest,
Interaction = Limited; prevalence not applicable

Example Scoring - 2

HS SEDI
Homeland Security Systems Engineering and Development Institute

Base Finding Group	Attack Surface Group	Exploitability Group
•I:1.0 •FC:T -> 1.0 •RC:Loc -> 0.5 •P:NA -> 1.0	•UN:All -> 1.0 •AV:P -> 0.2 •RP:N -> 1.0 •AS:N -> 1.0 •AI:N -> 1.0	•DI:H -> 1.0 •EX:M -> 0.6 •IN:Aut -> 1.0 •IC:N -> 1.0 •EC:N -> 1.0
0.5	**0.2**	**0.6**

100 x 0.5 x 0.2 x 0.6 = 6.0

Impact = 10/10, Remediation Cost = Localized, Access Vector = Physical, Exploit Likelihood = Medium,.

Example Scoring - 3

HS SEDI
Homeland Security Systems Engineering and Development Institute

Base Finding Group	Attack Surface Group	Exploitability Group
•I:0.9 •FC:T -> 1.0 •RC:NA -> 1.0 •P:NA -> 1.0	•UN:All -> 1.0 •AV:N -> 1.0 •RP:P -> 0.4 •AS:L -> 0.9 •AI:S -> 0.8	•DI:H -> 1.0 •EX:H -> 1.0 •IN:Aut -> 1.0 •IC:L -> 0.9 •EC:N -> 1.0
0.9	**0.288**	**0.9**

100 x 0.9 x 0.288 x 0.9 = 23.328

Impact = 9/10, Remediation Cost = Not Applicable, Partially Privileged User, Low Authentication Strength, Limited Internal Controls

CWSS Vector Examples

HS SEDI
Homeland Security Systems Engineering and Development Institute

Ex. 1: (I:Q,0.8/FC:T,1.0/RC:Min,1.0/P:NA,1.0/UN:All,1.0/AV:N,1.0/RP:G,0.9/AS:N,1.0/AI:N,1.0/DI:H,1.0/EX:H,1.0/IN:Ltd,0.9/IC:N,1.0/EC:N,1.0)

Ex. 2: (I:Q,1.0/FC:T,1.0/RC:Loc,0.5/P:NA,1.0/UN:All,1.0/AV:P,0.2/RP:N,1.0/AS:N,1.0/AI:N,1.0/DI:H,1.0/EX:M,0.6/IN:Aut,1.0/IC:N,1.0/EC:N,1.0)

Weights are important for "portability"

Ex. 3: (I:Q,0.9/FC:T,1.0/RC:NA,1.0/P:NA,1.0/UN:All,1.0/AV:N,1.0/RP:P,0.4/AS:L,0.9/AI:S,0.8/DI:H,1.0/EX:H,1.0/IN:Aut,1.0/IC:L,0.9/EC:N,1.0)

Limitations of the Current Approach

- **All factors are given equal emphasis in the final equation**
- **One or two low-scoring factors can significantly reduce the overall score**
  - This was an issue for CVSSv1
  - Currently-proposed weights don't have to be so drastic
    - Ranges from 0.2 to 1.0 could be narrowed from 0.8 to 1.0
    - If all scores are 0.8, minimum score is ~4.3 out of 100
- **Factors are not entirely independent**
  - Higher-impact issues (I) are more likely to be exploited (EX)
  - Attackers will focus on discovering (DI) higher-impact issues (I)
  - Authentication strength (AS) implies at least one authentication instance (AI) and may impact Discoverability (DI)
- **Factors such as discoverability and likelihood of exploit may be CWE-specific and/or vignette-specific, and difficult to quantify**

Aggregated Scoring

- **For a software package, how to combine all reported weaknesses to get an overall score?**
  - Individual score of the worst weakness
  - Combined score of all weaknesses
  - Account for size of code?
  - Normalize results from 0 to 100
- **“Weakness Surface”**
- **One step closer to the Software Facts Label**

Security Facts

Type: Web Application — Oct 25, 2010

OWASP Top 10 2010

A1-Injection	●
A2-Cross Site Scripting (XSS)	◒
A3-Authentication	○
A4-Object References	●
A5-Cross Site Request Forgery	●
A6-Security Configuration	●
A7-Cryptographic Storage	●
A8-URL Access Control	●
A9-Transport Layer Protection	◒
A10-Redirects and Forwards	○

Custom Code Modules

Name	Languge	Size (LOC)
Reports	Java	19000
Midtier	Java	135000
UI	JSP	215105
Engine	Java	512013
Database	SQL	65000

Libraries

Name	Languge	H
Struts 2.1.0	Java	●
Log4j 1.9.1	Java	○
XOM 1.2	Java	○

Platform Components

Name	Languge	H
WebSphere	Java	●

Interfaces and Connections

Name	Protocol	D	S
MPayment	SOAP	↔	●
DB2	JDBC	↔	○
Filenet	FTP	→	○

Sensitive Data

Name	C	I	A
Medical Imagery	●	●	●
Statements	●	●	○

Application Security Program

Key Practice Area	M
M1-Strategy and Metrics	◒
M2-Policy and Compliance	◒
M3-Education and Guidance	◒
M4-Threat Assessment	○
M5-Security Requirements	○
M6-Secure Architecture	○
M7-Design Analysis	○
M8-Code Review	○
M9-Security Testing	○
M10-Vulnerability Mgmt	○
M11-Environment Hardening	○
M12-Operational Enablement	○

Security Contact: security@owasp.org

OWASP T10

OWASP OpenSAMM

http://www.aspectsecurity.com/SecurityFacts/

From Jeff Williams, Aspect Security

Considerations for CWSS beyond 0.2

- **Technical impact model is limited**
  - "code execution" applies to XSS, SQL injection, OS command injection, buffer overflows…
- **Factors for scores might change regularly**
  - Prevalence may change
  - Vignettes may change
  - Technical impacts may change
  - CWE will change
  - "Versioning" for CWSS when factors change
  - Is this manageable when sharing CWSS scores?
- **How to score design-level/systemic issues?**
  - Remediation Cost = Systemic gets 0.1 weight
- **Should users be allowed to modify the weights as well?**
- **Should more factors be quantitative?**
- **Should the skew towards low scores be addressed?**

Recent Activities

- **http://cwe.mitre.org/cwss**
- **White paper published**
  - Vignettes available for review
- **Community recruitment underway**
  - Working with SANS
  - Talked with several software security capability vendors
  - Software security tool vendors
  - Developers
  - End users / consumers
  - Vignette-oriented experts (e.g. SCADA)
  - CVSS SIG
- **Associated CWE content changes**

Future Activities

- **Get broader feedback on white paper and vignettes**
  - Experts needed for vignettes
  - Might need narrower focus
  - Investigate/document existing scoring methods
- **Continue building community**
- **Enhance targeted scoring methods in CWSS 0.3**
- **Adapt CWSS to 2011 CWE/SANS Top 25**
- **Improve CWE "Technical Impacts" model**
- **Explore aggregated scoring**
- **Use vignette-oriented scoring in Pocket Guide**

- **Refine the message!**

cwss@mitre.org

Prevalence Estimates per CWE

HS SEDI
Homeland Security Systems Engineering and Development Institute

CWE	Name	Prevalence (1-10)
CWE-79	XSS	9.46
CWE-89	SQL injection	7.43
CWE-120	Classic overflow	6.04
CWE-352	Cross-site Request Forgery	7.75
CWE-285	Insufficient Authorization	6.04
…	…	…

- Prevalence data is rarely available at this level of detail
- Borrow data from 2010 CWE Top 25 votes
- Normalize 1-4 scores to 1-10 range

http://cwe.mitre.org/cwss/vignettes.html#votesum