Common Configuration Scoring System (CCSS)

Karen Scarfone

Scarfone Cybersecurity

Acknowledgements

- Content based on NIST Interagency Report 7502, *The Common Configuration Scoring System (CCSS),* by Karen Scarfone and Peter Mell, NIST
- Some slides based on CVSS presentation by Gavin Reid, Cisco Systems

Agenda

- Overview of CCSS
- Base metrics and scores
- Base example
- Temporal and environmental metrics

Security Configuration Issues

- Settings—options for the security of operating systems and applications
  - Enable or disable encryption of stored passwords
  - Access control list for file privileges
- Uninstalling unneeded software features
- CCE version 5 examples
  - CCE-2519-7 (Vista): “The amount of idle time required before disconnecting a session should be set correctly.”
  - CCE-4191-3 (RHEL 5): “The dhcp client service should be enabled or disabled as appropriate for each interface.”

CCSS Overview

- Common Configuration Scoring System
- A universal way to convey the relative severity of security configuration choices
- A set of metrics and formulas
- Solves problem of incompatible scoring systems
- Open, usable, and understandable by anyone
- Based on CVSS version 2
  - CVSS = software flaw vulnerabilities
  - CCSS = software security configuration issues
- Not a risk assessment solution

Why CCSS?

- Many exploits performed by taking advantage of vulnerabilities other than software flaws
- Dozens or hundreds of security configuration elements in each operating system and many applications
- Understanding security implications of each configuration option allows better risk assessment and sound decision-making
- Metrics and formulas designed to be fully compatible with CVSS metrics and formulas

Why Not Use CVSS Instead?

- Identified two key differences in scoring software flaws and configuration settings
- Software flaws and some settings permit unauthorized actions; other settings prevent authorized actions (insufficient privileges, lack of auditing, etc.)
  - Have two classes of settings in CCSS
- Software flaws are universally bad, but many settings are environment-specific—no “correct” value
  - Often multiple scores possible per setting
  - Both positive and negative security implications

Agenda

- Overview of CCSS
- Base metrics and scores
- Base example
- Temporal and environmental metrics

Base Metric Group

- Most fundamental qualities of a vulnerability, also referred to as a “weakness”
- Does not change; intrinsic and immutable
- Represents general vulnerability severity
- Two subsets of metrics:
  - Exploitability: Access Vector, Authentication, Access Complexity, and Exploitation Method
  - Impact: Confidentiality, Integrity, Availability, and Privilege Level

Exploitation Method (EM)

- Active (A) exploitation of vulnerabilities that permit unauthorized actions to occur
  - Attacker gains access to sensitive file through overly permissive file privileges
- Passive (P) exploitation of vulnerabilities that prevent authorized actions
  - Authorized system service cannot run
  - Audit log records not generated for security events
- EM not used directly in generating CCSS scores

Access Vector (AV)

- For Active (A) exploitation, measures from where the vulnerability can be exploited
- Local (L): The vulnerability is only exploitable locally (physical access or local account)
- Adjacent Network (A): The attacker must have access to either the broadcast or collision domain of the vulnerable software
- Network (N): The vulnerable software is bound to the network stack and the attacker does not need local or adjacent network access to exploit it

Access Vector (AV) (cont.)

- For Passive (P) exploitation, measures from where authorized parties should be able to perform the prevented action
- Local (L): The vulnerability only affects local users, processes, services, etc.
- Adjacent Network (A): The vulnerability affects users or other hosts on the same broadcast or collision domain
- Network (N): The vulnerability affects all users or hosts

Authentication (Au)

- Measures the number of times an attacker must authenticate to a target *once it has been accessed* in order to exploit a vulnerability
- Multiple (M): Exploiting the vulnerability requires that the attacker authenticate two or more times (e.g., first OS, then application), even if the same credentials are used each time
- Single (S): One instance of authentication is required
- None (N): Authentication is not required to exploit the vulnerability

Access Complexity (AC)

- For Active (A) exploitation, measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target host
- High (H): Specialized access conditions exist, such as the attacker already having elevated privileges, or the vulnerability only making it slightly easier for a subsequent attack to succeed
- Medium (M): The access conditions are somewhat specialized, such as only certain hosts or users being able to perform attacks, the affected configuration being uncommon, or some information gathering being required
- Low (L): Generally easy to exploit, such as the affected configuration being the default, and the attack requiring little skill or information gathering

Access Complexity (AC) (cont.)

- For Passive (P) exploitation, always set to Low (L)
- The outcome of the vulnerability, such as not permitting an authorized service to run or not logging security events, has already occurred or is constantly occurring
  - No additional actions are needed to “exploit” it

Exploitability Base Metrics

- Exploitation Method (EM)
  - Active, Passive
- Access Vector (AV)
  - Local, Adjacent Network, Network
- Access Complexity (AC)
  - High, Medium, Low
- Authentication (Au)
  - Multiple, Single, None

Confidentiality Impact (C)

- Measures the impact on confidentiality of a successfully exploited vulnerability
  - Includes both information and resource access
- None (N): No impact on confidentiality
- Partial (P): Considerable informational disclosure, such as access to some files or certain database tables; or considerable (but not total) unauthorized access to the host
- Complete (C) : Total information disclosure; the attacker can read all of the host's data (including files and memory)

Integrity Impact (I)

- Measures the impact to integrity of a successfully exploited vulnerability
- None (N): No impact on integrity
- Partial (P): Modification of some system files or information; or, the vulnerability can be misused to alter the host's security configuration, such as placing malware-infected files on the host
- Complete (C): Total compromise of system integrity; the attacker can modify any data (files, memory, etc.) on the target host

Availability Impact (A)

- Measures the impact to availability of a successfully exploited vulnerability
- None (N): No impact on availability
- Partial (P): Reduced performance or interruptions in resource availability
- Complete (C): Total shutdown of the affected host
- Underlying assumption in all impact metrics of impact to the OS, not just a targeted application or service

Privilege Level (PL)

- The level of unauthorized access that an attacker could gain
  - For example, impersonating a user or gaining full access to an application or OS
  - Root Level (R)
  - User Level (U)
  - Application Level (A)
  - Not Defined (ND)
- Does not directly affect base scores; used by environmental metrics

Base Metrics

- Confidentiality Impact (C), Integrity Impact (I), Availability Impact (A)
  - None, Partial, Complete
- Privilege Level (PL)
  - Root Level, User Level, Application Level, Not Defined
- Exploitation Method (EM)
- Access Vector (AV)
- Access Complexity (AC)
- Authentication (Au)

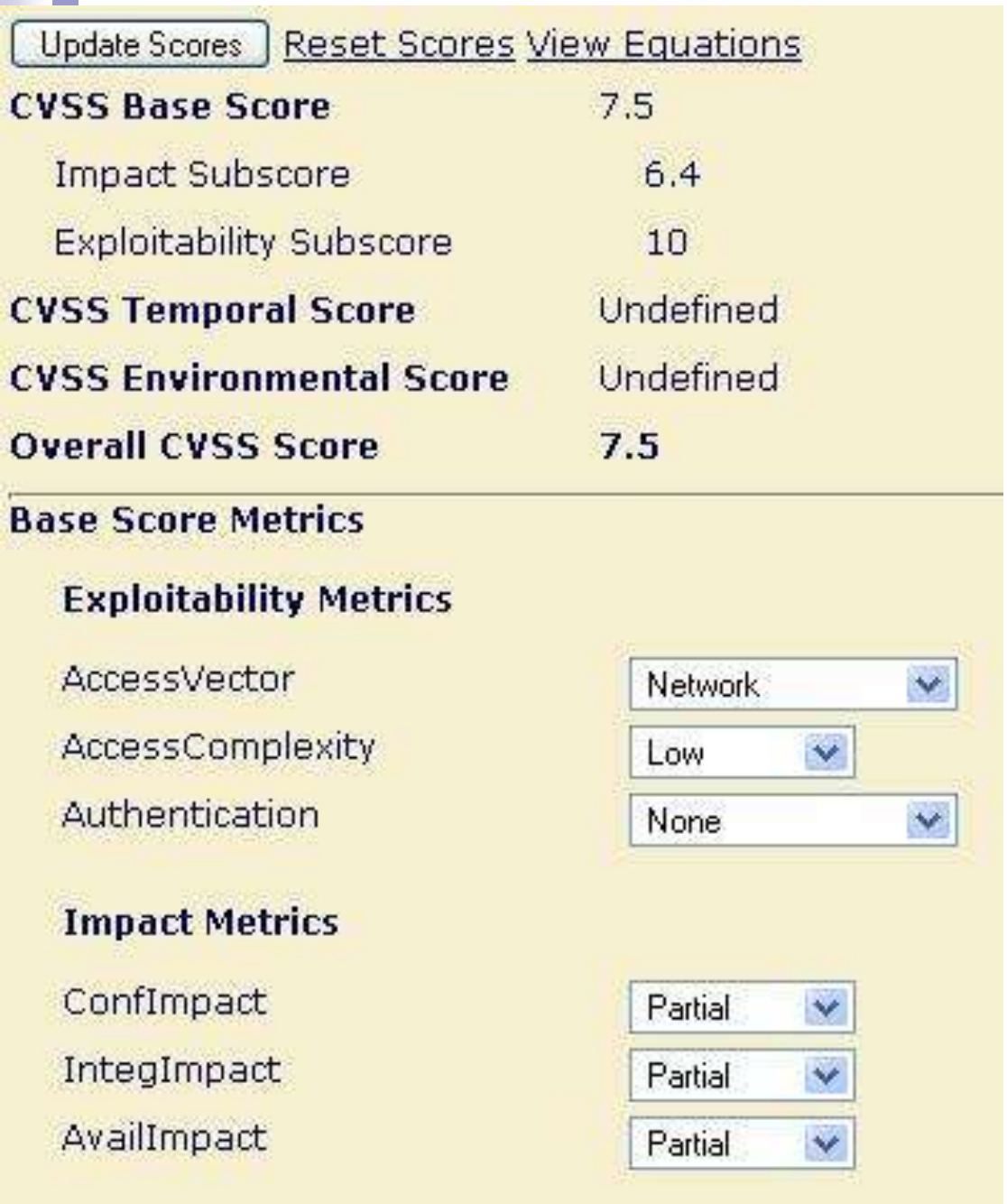
Base Scoring

- To be computed by vendors and coordinators
- Each metric has a number assigned to each possible value
  - AccessComplexity: high = 0.35, medium = 0.61, low = 0.71
  - Integrity: none = 0.0, partial = 0.275, complete = 0.66
- The metrics' values are combined with formulas that give different weights to the base metrics
- Base subscores for impact and exploitability
- The final base score is between 0.0 and 10.0
  - 60% of impact subscore + 40% of exploitability subscore
- All metric values and formulas the same as CVSS's

Base Vector

- A vector is a representation of the values assigned to the CCSS metrics
- Every CCSS score should be accompanied by the corresponding vector, so that people can see the components of the score and validate them
- CCSS base vector has 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]/PL:[R,U,A,ND]/EM:[A,P])
- Sample vector:
  (AV:N/AC:L/Au:N/C:P/I:P/A:P/PL:ND/EM:A)
- Superset of CVSS vector format



NVD CVSS Calculator can be used for CCSS base scores

Multiple Scores Per Vulnerability

- No universally “right” option for many configuration issues
- Some have only a few options, such as enabled/disabled or low/medium/high
  - Consider each combination of desired setting vs. actual setting that has security implications, and generate a score and vector for each
- Some have many options, such as ACLs
  - Consider the common cases independently
  - Example—for timeout, it could be set too high, set too low, or disabled
- Users have to select the appropriate scores and vectors for their environment and situation

Agenda

- Overview of CCSS
- Base metrics and scores
- Base example
- Temporal and environmental metrics

Example - CCE-2366-3

- CCE-2366-3 for Windows XP: “The ‘shut down the system’ user right should be assigned to the correct accounts.”
- Do not know to whom the right has been granted
  - Perhaps granted to some users that should not have it?
  - Perhaps not granted to some users that should have it?

Example (cont.)

- For the case where users should not have the right but do…
  - Exploitation Method is set to "Active" because users have to perform particular actions to take advantage of this.
  - Since the vulnerability is exploitable only to a user locally logged into the host, the Access Vector is "Local".
  - Access Complexity is "Low" because a user could use features built into the OS to exploit the vulnerability.
  - Authentication is set to "None" because no additional authentication is needed after local login.
  - Availability Impact is set to "Complete" because the user can make the entire host unavailable at will.
  - Confidentiality Impact and Integrity Impact are both set to "None" because they are unaffected.
  - Privilege Level is set to "Not Defined".
  - Base score 4.9, vector AV:L/AC:L/Au:N/C:N/I:N/A:C/PL:ND/EM:A

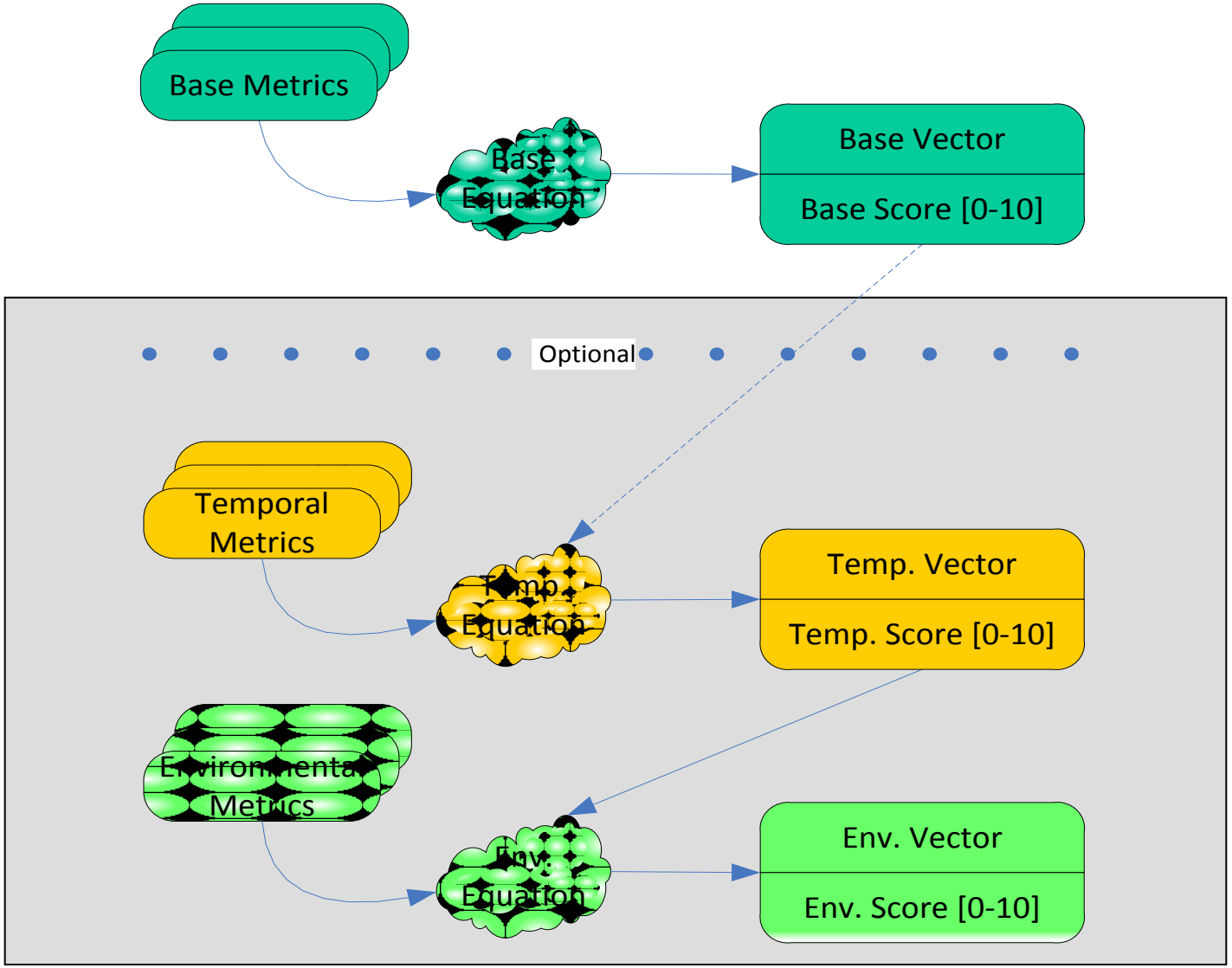
Example (cont.)

- For the case where users should have the right but do not…
  - Exploitation Method is set to "Passive" because users do not have to perform particular actions to be affected.
  - Since the vulnerability is exploitable only to a user locally logged into the host, the Access Vector is "Local".
  - Access Complexity is "Low" because EM is Passive.
  - Authentication is set to "None" because no additional authentication is needed after local login.
  - Availability Impact is set to "Partial" because a needed feature is unavailable to users.
  - Confidentiality Impact and Integrity Impact are both set to "None" because they are unaffected.
  - Privilege Level is set to "Not Defined".
  - Base score 2.1, vector AV:L/AC:L/Au:N/C:N/I:N/A:P/PL:ND/EM:P

Agenda

- Overview of CCSS
- Base metrics and scores
- Base example
- Temporal and environmental metrics

CCSS Metrics and Scores



Temporal: General Exploit Level (GEL)

- Prevalence of attacks against the vulnerability—how often any vulnerable system is likely to come under attack
  - None (N) (no exploits yet observed)
  - Low (L) (rarely observed; months to years)
  - Medium (M) (occasional; days)
  - High (H) (frequent; hours, minutes, or seconds)
  - Not Defined (ND) (skip this metric in calculating the score)

Temporal: General Remediation Level (GRL)

- Availability of remediation measures that can mitigate the vulnerability, other than changing the configuration setting or rendering it useless
  - High (H) (remediations can collectively decrease exploitation by 76-100%)
  - Medium (M) (remediations can collectively decrease exploitation by 26-75%)
  - Low (L) (decrease exploitation by 1-25%)
  - None (N) (remediations not available)
  - Not Defined (ND) (skip this metric in calculating the score)

CCSS Environmental Metrics

- Local Exploit Level
  - Counterpart to General Exploit Level (temporal)
  - Scaling factor applied to Exploitability components of the base metric
- Local Remediation Level
  - Counterpart to General Remediation Level (temporal)
  - Scaling factor applied to Exploitability components of the base metric
- Local Impact
  - Several metrics that adjust the base impact metrics

Local Exploit Level

- Local Vulnerability Prevalence (LVP)
  - Prevalence of vulnerable hosts in a specific environment; approximate % of hosts that could be affected by the vulnerability
  - None (N), Low (L), Medium (M), High (H), Not Defined (ND)
- Perceived Target Value (PTV)
  - Likelihood of attack using the configuration issue in an environment relative to vulnerable hosts in other environments
  - Low (L), Medium (M), High (H), Not Defined (ND)

Local Remediation Level (LRL)

- Level of protection against a vulnerability within the local IT environment; how widespread mitigation is implemented and how effective the mitigation is
  - % decrease in the incidence of exploitation
  - High (H), Medium (M), Low (L), None (N), Not Defined (ND)

Local Impact

- Environment Confidentiality, Integrity, and Availability Impact (EC, EI, EA)
  - Take the place of the corresponding base impact metrics
- Collateral Damage Potential (CDP)
  - Augments the Environment Impact metrics
- Confidentiality, Integrity, Availability Requirements (CR, IR, AR)
  - Used to compute scaling factors that are applied to the Environment Impact metrics

Environment Impact

- Customize score if the privileges in the environment differ significantly from best practices related to the vulnerability
  - For example, allowing users to run with full, administrator-level privileges
- Environment Confidentiality (EC), Environment Integrity (EI), Environment Availability (EA) Impact metrics
  - Include all the same definitions as the base impact metrics (None, Partial, Complete)
  - Each also includes a Not Defined (ND) value, indicating to skip the metric

Collateral Damage Potential (CDP)

- Measures the potential for loss of life or physical assets through damage or theft of property or equipment, and economic loss of productivity or revenue
- None (N): No potential for physical assets, productivity or revenue damage
- Low (L): Slight damage or loss of revenue or productivity
- Low-Medium (LM): Moderate damage or loss
- Medium-High (MH): Significant damage or loss
- High (H): Catastrophic damage or loss
- Not Defined (ND): No value assigned—skip this metric in calculating the score
- Each organization has to define precisely what “slight”, “moderate”, “significant”, and “catastrophic” mean

Security Requirements

- Customize score based on the importance of the targets to the organization in terms of the targets' confidentiality, integrity, and availability
- Confidentiality requirement (CR), integrity requirement (IR), availability requirement (AR): each affects the weight of the corresponding Environment Impact metric
- Effect on the organization or associated individuals:
  - Low (L): Likely to have only a limited adverse effect
  - Medium (M): Likely to have a serious adverse effect
  - High (H): Likely to have a catastrophic adverse effect
  - Not Defined (ND): No value assigned—skip this metric in calculating the score. Default value is Medium.

Links

- NIST Interagency Report 7502 (CCSS)
  http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7502
- NIST NVD CVSS v2 Calculator
  http://nvd.nist.gov/cvss.cfm?calculator&version=2

Questions?

- karen@scarfonecybersecurity.com

Overview of CxSS

- CVSS for software flaw vulnerabilities
- CCSS for security configuration vulnerabilities
- Common Misuse Scoring System (CMSS) for software feature/trust relationship misuse vulnerabilities
- CxSS example—use IM to transfer unwanted files (malware) to the user's host
  - CVSS: Coding flaw in IM client permits such transfers
  - CCSS: IM client is configured to permit such transfers
  - CMSS: Social engineering tricks user into permitting such transfers; user mistakenly accepts transfer request; IM client does not offer a configuration option for restricting transfers