Common Configuration Scoring System (CCSS)

Karen Scarfone, NIST

Acknowledgements

- Content based on draft 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

Disclaimer: Certain commercial equipment or materials are identified in this presentation in order to adequately specify and describe the use of CCSS. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Agenda

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

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
- Future work

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 three metrics each:
  - Exploitability: Access Vector, Access Complexity, Authentication
  - Impact: Confidentiality, Integrity, Availability

Active and Passive Exploitation

- Active exploitation of vulnerabilities that permit unauthorized actions to occur
  - Attacker gains access to sensitive file through overly permissive file privileges
- Passive exploitation of vulnerabilities that prevent authorized actions
  - Authorized system service cannot run
  - Audit log records not generated for security events

Access Vector (AV)

- For active exploitation, measures how remote an attacker can be to exploit a vulnerability
- 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 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

Access Complexity (AC)

- For active 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 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

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

Exploitability Base Metrics

- 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

Base Metrics

- Confidentiality Impact (C), Integrity Impact (I), Availability Impact (A)
  - None, Partial, Complete
- Access Vector (AV)
  - Local, Adjacent Network, Network
- Access Complexity (AC)
  - High, Medium, Low
- Authentication (Au)
  - Multiple, Single, None

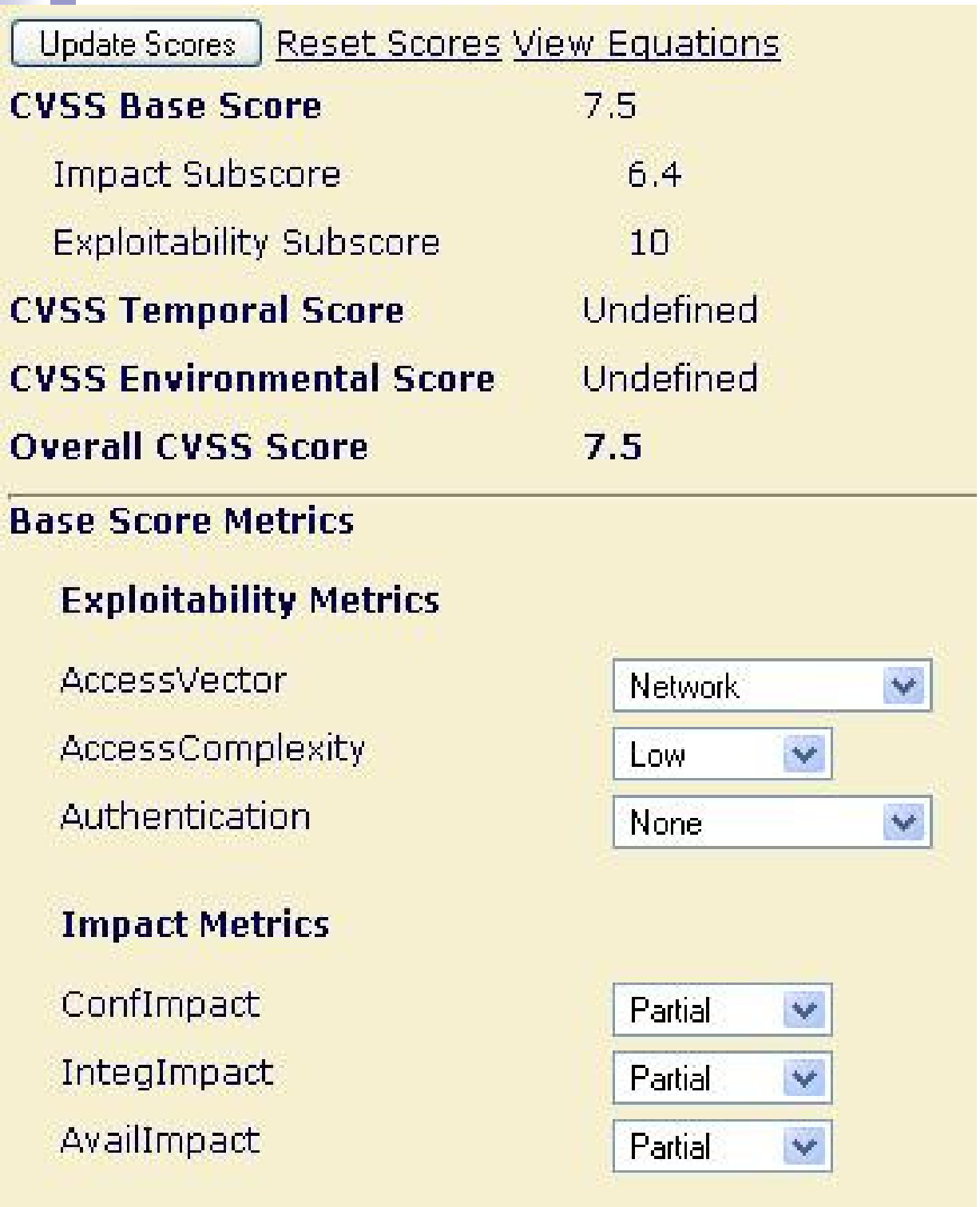
Base Scoring

- 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])
- Sample vector:
  (AV:N/AC:L/Au:N/C:P/I:P/A:P)
- Identical to 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
- Future work

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…
  - 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 set to "None" because they are unaffected.
  - Base score 4.9, vector AV:L/AC:L/Au:N/C:N/I:N/A:C

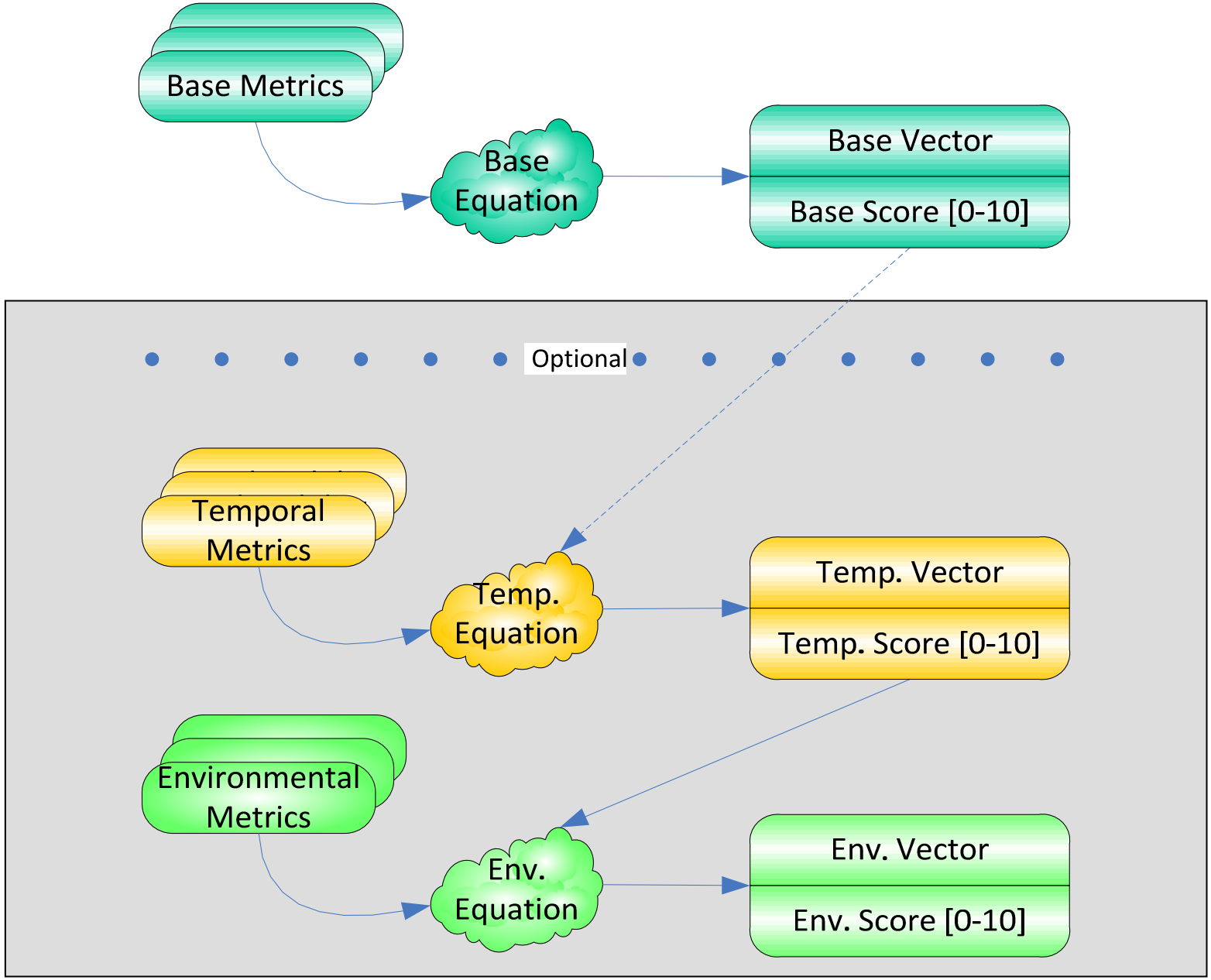
Example (cont.)

- For the case where users should have the right but do not…
  - Since the vulnerability is exploitable only to a user locally logged into the host, the Access Vector is "Local".
  - Access Complexity is "Low" because no action is needed (passive exploitation).
  - 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 set to "None" because they are unaffected.
  - Base score 2.1, vector AV:L/AC:L/Au:N/C:N/I:N/A:P[28]

Agenda

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

CVSS Metrics and Scores



Current State of CCSS

- Draft specification for base metrics and formulas
- Not started on temporal or environmental metrics
- Initial assumption that temporal metrics may not be applicable to CCSS
  - From CVSS: availability of exploit code, availability of remediation, confidence in vulnerability reports
- Environmental metrics work to be done in conjunction with review of CVSS metrics

CVSS Environmental Metrics

- Qualities of a vulnerability specific to a particular IT environment
- Collateral Damage Potential
- Target Distribution
- Security Requirements
  - Confidentiality requirement
  - Integrity requirement
  - Availability requirement

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

Target Distribution (TD)

- Measures the proportion of vulnerable systems in an environment
- None (N): No target systems exist, or targets are highly specialized and exist only in a laboratory setting (0%)
- Low (L): Targets exist on a small scale (1-25%)
- Medium (M): Targets exist on a medium scale (26-75%)
- High (H): Targets exist on a considerable scale (76-100%)
- Not Defined (ND): No value assigned—skip this metric in calculating the score

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 base metric (C, I, A)
- 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

How Scores Could Be Used

- Three primary uses envisioned
  - Compare relative severity of options
  - Inputs to risk assessment tools/methodologies
  - Awareness of the security implications of security configuration choices
- Concerns about over-reliance on scores
  - Do not reflect the full likelihood of attack (such as a popular product being targeted more often than a rarely used product)
  - Do not take into account whether deployed security controls may prevent exploits
  - May be errors in scoring

Agenda

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

Future Work

- Refining CCSS base metric specification based on public feedback
- Creating environmental metrics (and temporal, if needed)
- Testing CCSS on additional vulnerabilities
  - Already tested an earlier version on around 200 CCEs
- Creating a CCSS-like specification for use with other types of software vulnerabilities
- Mapping relationships between all vulnerabilities, configuration settings, security controls, etc. for risk assessment modeling purposes

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

Current State of CMSS

- Draft specification co-authored by Elizabeth Van Ruitenbeek and Karen Scarfone
- Uses CVSS/CCSS base metrics
- Defines mostly new temporal and environmental metrics and formulas
  - All specific to exploitability or impact, so there are subscores for base, temporal, and environmental
- Nearly ready for limited expert review

Links

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

Questions?

- karen.scarfone@nist.gov