Capacity Building with ITU-T Cybersecurity Standards

2013/12/5

Youki Kadobayashi, Rapporteur, ITU-T Q.4/17

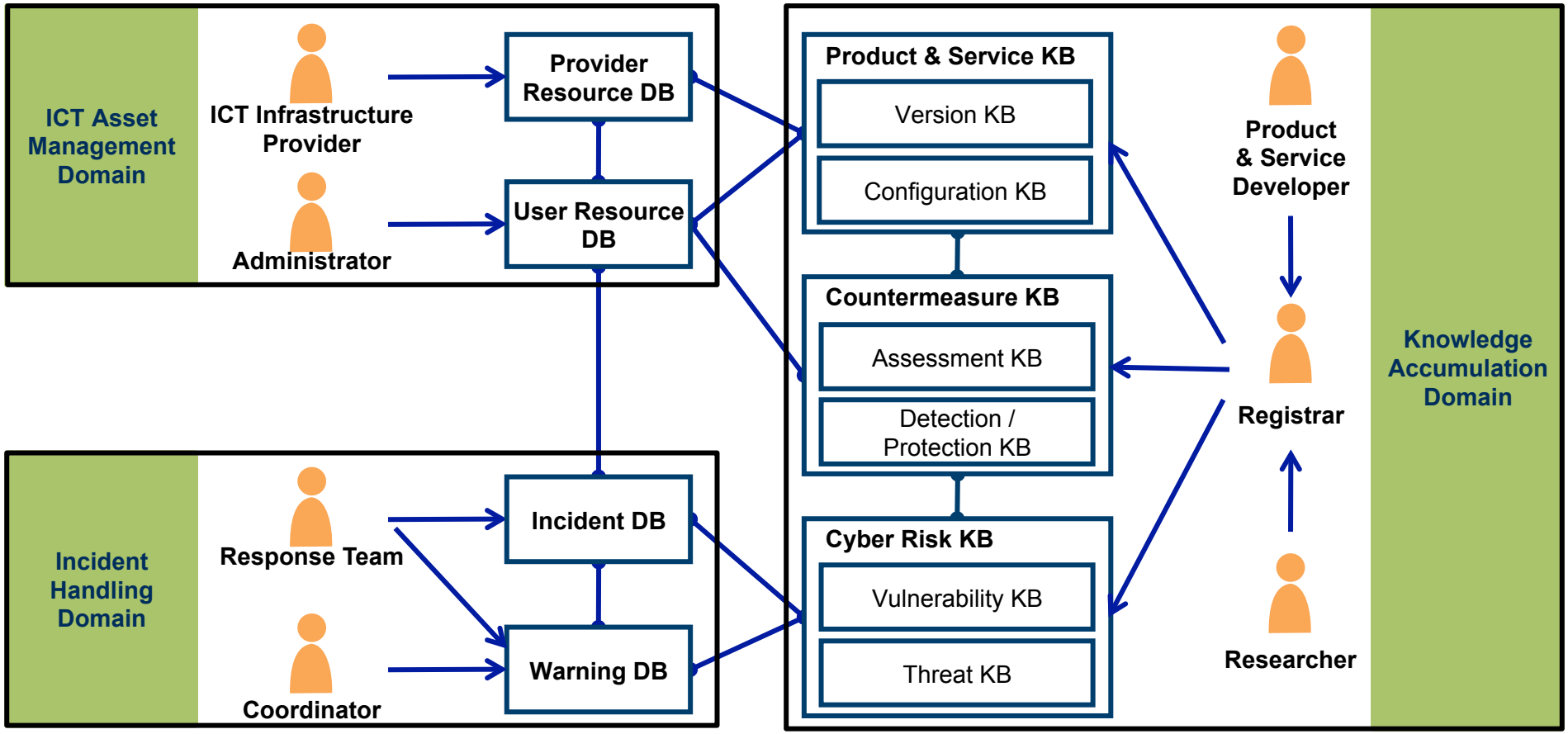
Capacity building with ITU-T cybersecurity standards

- Cybersecurity comprises of process-oriented cycle
  - e.g., Assess – Detect – Mitigate – Analyze – Prevent

- Existing process-oriented standards, as well as checklist standards, should be complemented with detailed knowledge-base of cybersecurity, because:
  - Cyber-risks are highly volatile
  - Chain reactions are typical; difficult to estimate the risk without considering technical detail
  - You'll need to communicate the detail
- ITU-T provides knowledge-base standards

Cybersecurity knowledge base: An ontology for cybersecurity information

Source: ITU-T Recommendation X.1500 Appendix II



Knowledge base of vulnerabilities

- CVE: Common Vulnerability Enumeration
  - a structured means to exchange information on security vulnerabilities and exposures and provides a common identifier for publicly-known problems.
  - http://cve.mitre.org/
  - Standardized as ITU-T X.1520
  - National databases:
    - NIST NVD
    - Japan JVN
  - R. Martin, “Managing Vulnerabilities in Networked Systems”, IEEE Computer, 34(11), Nov 2001.

CVE schema

Name	Description
Overview	Human-readable description
Impact	CVSS scoring
References	Advisories, solutions, tools
Vulnerable software and versions	Enumerations of CPE ID (Common Platform Enumeration)
Vulnerability type	Reference to CWE

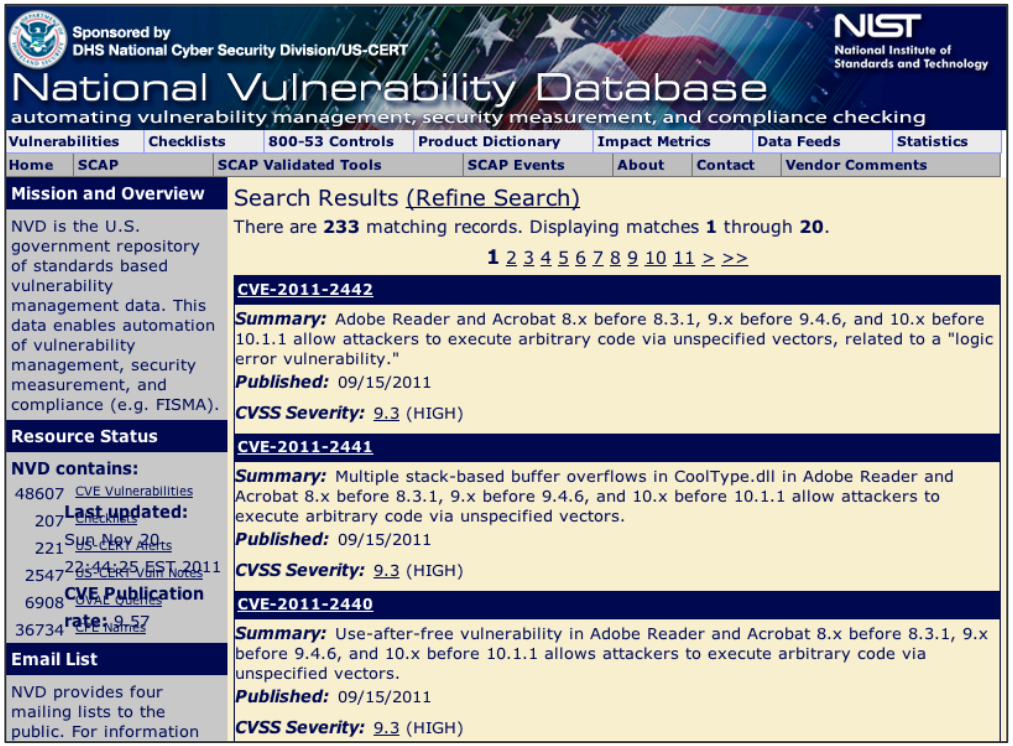
CPE: common naming of IT assets

- CPE: Common Platform Enumeration
  - a structured method of describing and identifying classes of applications, operating systems, and hardware devices present among an enterprise's computing assets.
  - URI for IT assets, primarily software
  - Standardized as ITU-T X.1528

  - `cpe:/o:microsoft:windows_2003`
  - `cpe:/a:adobe:reader:8.1`

Ongoing Proliferation of CVE

- CVE-compatible products and services
  - 27 countries, 157 organizations, 286 products



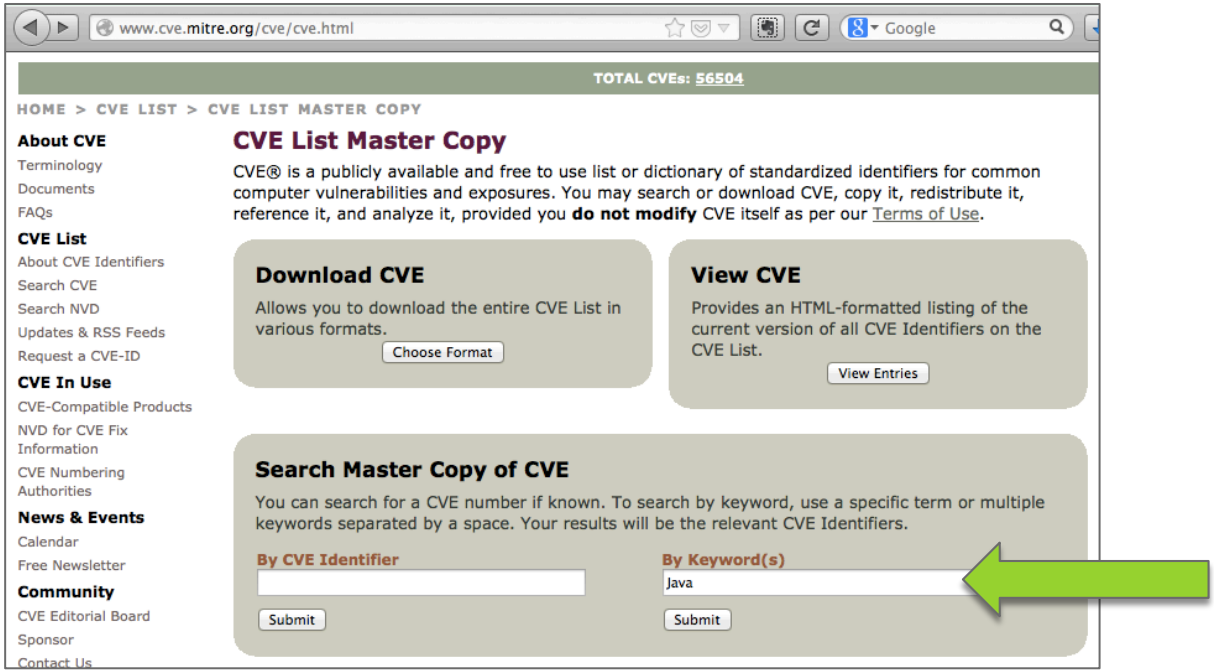
U.S. NIST NVD

Japan IPA JVN

A hands-on example: explore CVE

(estimated time for this hands-on: 15 min.)

- Pick a particular application that you use daily, and search cve.mitre.org with its name
- Create a spreadsheet, listing matching vulnerabilities
- Tip: find a software with lots of vulns, for interesting study



Taxonomy of vulnerabilities

- CWE: Common Weakness Enumeration
  - Group same kind of vulnerabilities into a weakness, and give it a distinct number
  - Provides common names for publicly known problems in the commercial or open source software
  - Intended for security tools and services that can find weaknesses in source code and operational systems
  - Helps better understand and manage software weaknesses related to architecture and design

  - http://cwe.mitre.org/
  - Standardized as ITU-T X.1524

CWE schema

a more taxonomical approach to vulnerability

Name	Description
Description	
Time of introduction	Architecture, design, implementation …
Applicable platforms	languages
Common consequences	Scope and effect
Demonstrative examples	Code example etc.
Potential mitigations	Possible measures in design, implementation, operation..
Taxonomy mappings	Other taxonomies

Reference

R. A. Martin, “Being Explicit About Security Weaknesses”, Crosstalk, Mar 2007.

CWE top 25

- Prioritized list of dangerous software errors
  - Intended to minimize software vulnerability

- cwe.mitre.org/top25/

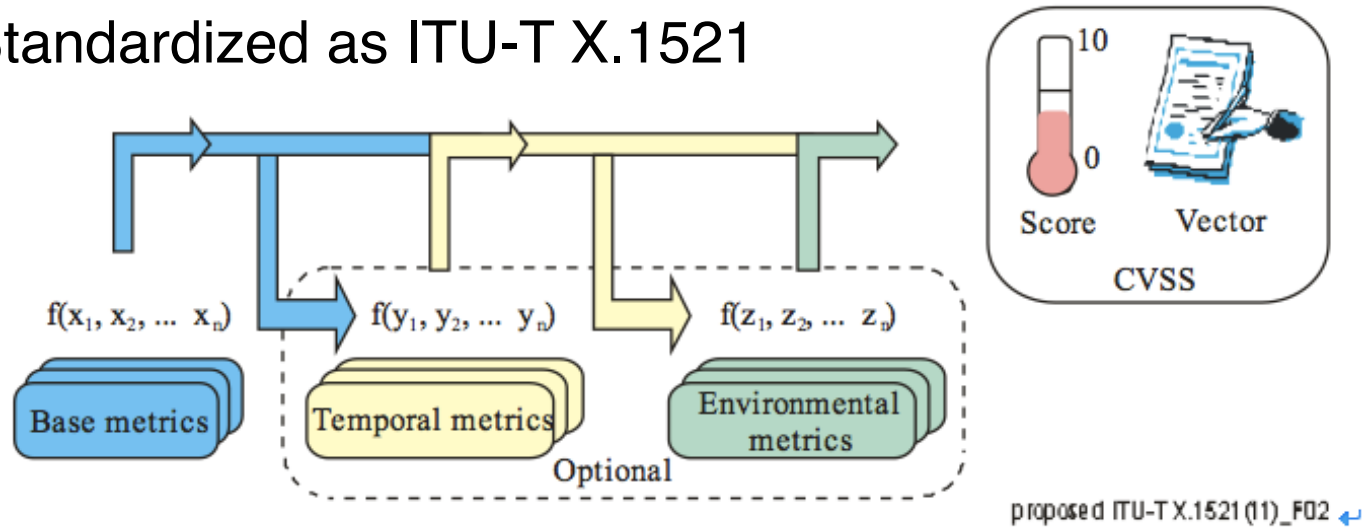
A hands-on example: explore CWE

(estimated time for this hands-on: 30 min.)

- On the previous spreadsheet which you created, analyze the trend of vulnerability by time, type, etc.
- What kind of insight can you draw from the analysis?
- You may contrast it with another software (of similar type, different language, etc.)

Quantification of vulnerabilities

- CVSS: common vulnerability scoring system
  - Base metrics: constant over time and across user environments
  - Temporal metrics: reflects vulnerability landscape
  - Environmental metrics: reflects user environments
  - http://www.first.org/cvss/
  - Standardized as ITU-T X.1521



Metrics in the CVSS v2

Base metrics

Name	Description
Access Vector	How vulnerability is exploited: Local (L), Adjacent network (A), Network (N)
Access Complexity	Complexity of attack required to exploit the vulnerability
Authentication	Number of times attackers must authenticate to exploit vuln
Confidentiality impact	Impact to confidentiality if exploited
Integrity impact	Impact to integrity if exploited
Availability impact	Impact to availability if exploited

Metrics in the CVSS v2

Temporal metrics

Name	Description
Exploitability	Current state of exploit techniques and code availability
Remediation level	Availability of official fix / temporal fix / workaround
Report confidence	Degree of confidence in the existence of vulnerability

Metrics in the CVSS v2

Environmental metrics

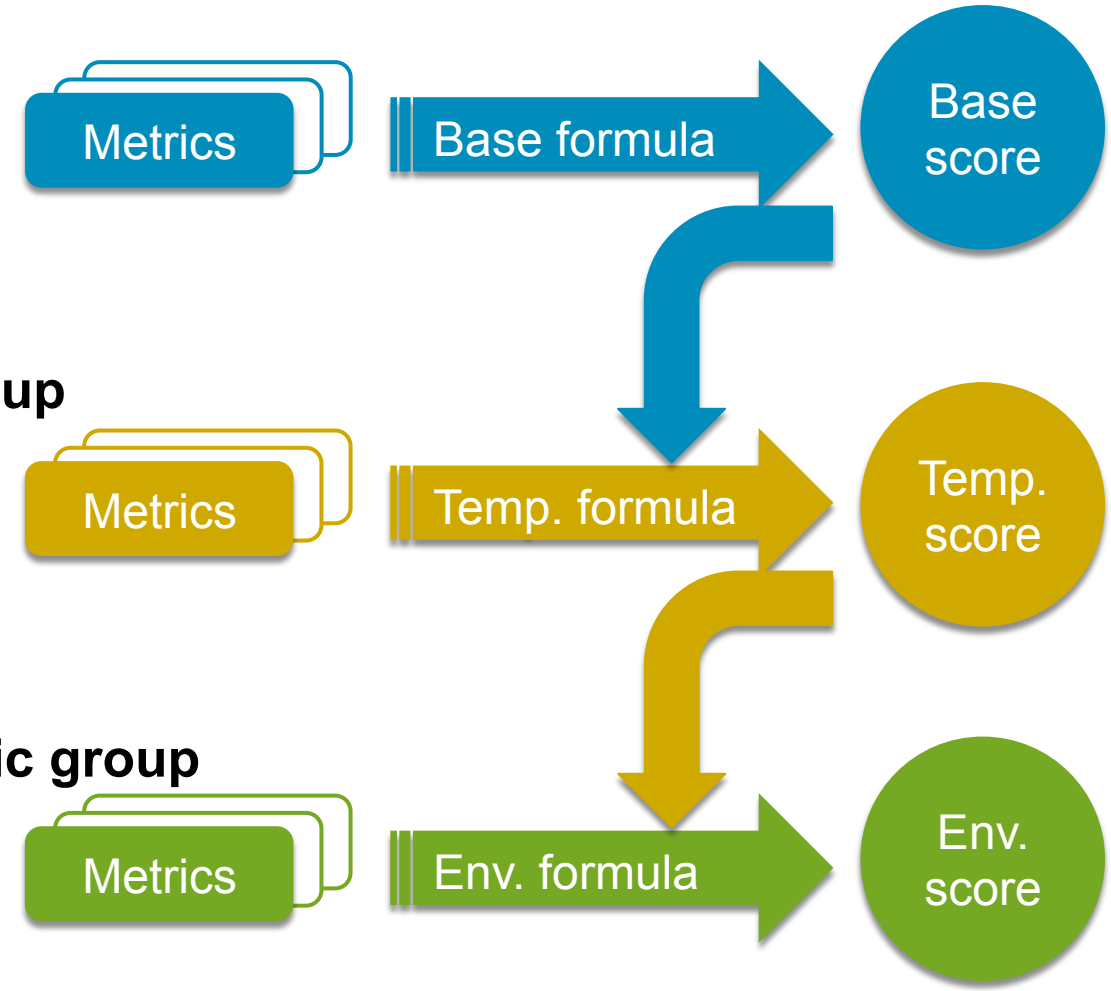
Name	Description
Collateral damage potential	Potential for loss of life, physical assets, productivity or revenue
Target distribution	The proportion of vulnerable systems
Security requirements	User requirements for confidentiality, integrity, availability

Derivation of CVSS v2 Score

Base metric group
Computed by vendor and coordinator;
Represents severity

Temporal metric group
Computed by vendor and coordinator;
changes over time;
represents urgency

Environmental metric group
Computed by user;
changes over time;
represents priority



A hands-on example: use CVSS

(estimated time for this hands-on: 20 min.)

- On the previous spreadsheet which you created, use CVSS to prioritize mitigation of particular set of vulnerabilities over others.
- Which vulnerabilities are considered most important?
- If you have multiple vulnerabilities with same CVSS score, propose a tie-breaking rule.
- Create a top 10 list of vulnerabilities

Knowledge base of attack patterns

- CAPEC: Common Attack Pattern Enumeration and Classification
  - Dictionary of attack patterns, solutions & mitigations
  - Facilitates communication of incidents, issues, as well as validation techniques and mitigation strategies
  - http://capec.mitre.org/
  - Standardized as ITU-T X.1544

CAPEC schema (partial)

Name	Description
Attack Pattern ID	Unique integer identifier
Attack Pattern Name	
Description	
Summary	
Attack Execution Flow	
Related Weakness	CWE ID
Related Vulnerability	CVE ID
Methods of Attack	
References	Further information
Solutions and Mitigations	
Severity	
…	…

A hands-on example: use CAPEC

(estimated time for this hands-on: 30 min.)

- On the previous spreadsheet which you created, associate CAPEC ID with top 10 vulnerabilities
- Create one-page executive summary, which describes impact of those vulnerabilities and persuades your customers to upgrade
- Send resulting document and spreadsheet to us via e-mail

Checklists

- OVAL: Open Vulnerability and Assessment Language
  - A standard for assessment and reporting of machine state of computer systems. OVAL includes a language to encode system details, and an assortment of content repositories held throughout the community.
  - http://oval.mitre.org/
  - Standardized as ITU-T X.1526

OVAL rule for detecting vulnerability

example: rule for detecting CVE-2011-2462

The system is vulnerable

- IF : Any one of the following are true
  - IF : All of the following are true Adobe Acrobat library is less than or equal to 8.2.4 and is greater than or equal to 8.0

    Prerequisites (Extended Definitions)

    Adobe Acrobat 8 Series is installed oval:org.mitre.oval:def:6452

    - IF : Adobe Acrobat library is less than or equal to 8.2.4
      - Windows : File Test : Adobe Acrobat library is less than or equal to 8.2.4

        At least one of the objects listed below must exist on the system (Existence check)

        Windows : File

        [[value of ${{windows:registry_object:HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Adobe Acrobat\8.0\Installer : Path}}]]Acrobat\acrobat.dll

        product_version [entity_check=all] less than or equal **8.2.4.268** (datatype=version)

        State matching the file version less than or equal to 8.2.4.268 windows : file_state
    - IF : Determine if the version of Adobe Acrobat is greater than or equal to 8.0
      - Windows : Registry Test : Determine if the version of Adobe Acrobat is greater than or

Using standards for continuous monitoring of the state of cybersecurity

ICT Asset
- CPE

Vuln.
- CVE
- CWE

Impact
- OVAL
- CVSS

Major ITU-T standards for cybersecurity

Definitions, knowledge base standards

- X.1205, Overview of Cybersecurity
- X.1251, A framework for user control of digital identity
- X.1252, Baseline identity management terms and definitions
- X.1254, Entity authentication assurance framework
- X.1500, Overview of cybersecurity information exchange
- X.1520, Common vulnerabilities and exposures
- X.1521, Common vulnerability scoring system
- X.1524, Common weakness enumeration
- X.1528, Common platform enumeration
- X.1544, Common attack pattern enumeration and classification

Summary

- ITU-T cybersecurity standards provide critical instruments to deal with rapidly changing and diversifying cybersecurity phenomena
- Enumeration standards provides effective means of communication across businesses, government agencies as well as communities
- Cyber-risks are highly volatile and manifests through unexpected combination of components, that requires careful examination of technical risks through knowledge-base standards