Standards in reporting Software Flaws: SCAP, CVE and CWE

Robin A. Gandhi, Ph.D.
University of Nebraska at Omaha (UNO)
College of Information Science and Technology (IS&T)
School of Interdisciplinary Informatics (Si2)
Nebraska University Center on Information Assurance (NUCIA)
Who am I?

- **Job**
  - Assistant Professor of Information Assurance at IS&T since Fall 2008

- **Research highlights**
  - Regulatory Requirements driven Risk Assessment
    - Using the semantic web to bridge the gap from high-level regulations to low-level technical evidence (Domain: SCADA)
  - Software Assurance in the Development Lifecycle
    - Building semantic templates for the most egregious software flaws
  - Cyber attack modeling and forecasting (CyCast)
    - Exploring disturbances in the human network to predict cyber attacks

- **Teaching**
  - Software Assurance (seniors/grad) New!
  - Foundations of Information Assurance (seniors/grad)
  - Introduction to Information Assurance (Freshmen) New!
  - Introduction to Computer Science II (Freshmen/Sophomore)
A two part talk

• SCAP
  – What is it?
  – What does it do?
  – What will it take to realize its potential?
  – What do I need to do to start preparing for it?

• How can we better understand vulnerabilities
  – Research on semantic templates built from CWE and CVE enumerations
The Burning Issue

• It has been said that we have long known how to build secure systems
  – We simply don’t act on what we know

• For a fielded system the details are “enormous” to assess the security posture
  – Rich abstractions supported by automation is key to manage the complexity of current systems
  – If we are in a constant battle, then let’s get efficient about it
What is SCAP

• Pronounced S-CAP
• Security Content Automation Protocol
  – NIST 800-126
    • Technical specification
  – NIST 800-117
    • Guide for adoption
  – NISTIR 7511 rev2
    • Requirements for achieving SCAP validation
    • Demonstration of SCAP capabilities
• This presentation borrows heavily from these documents
Motivation for SCAP

• The number and variety of systems to secure
• The need to respond quickly to new threats
• Compliance often becomes a paperwork exercise
• Lack of standard expression of security content
  – Duplication across standards and baselines
  – Lack of interoperability among tools
Clearing SCAP Myths

• No, NIST and FFRDCs are not attempting to regulate the entire security industry
  – It is really a community effort that wants your participation to grow and mature

• The managed data streams do not limit personal/proprietary innovation
  – Community repositories can be enriched with locally developed content and contributed back to the public
SCAP v1.0

• SCAP has two major elements:
  – **Components:** 6 open specifications that standardize the **format** and **nomenclature** by which **security software** communicates information about **software flaws** and **security configurations**.
  – **Content:** Software flaw and security configuration standardized **reference data**
SCAP v1.0 Components

• Expression and Checking Languages
  1. Express what is to be evaluated and how to report results
     • eXtensible Configuration Checklist Description Format (XCCDF); NSA and NIST
  2. Check the corresponding low level system states
     • Open Vulnerability Assessment Language (OVAL); MITRE
SCAP v1.0 Components

• **Enumerations**
  ③ Common Platform Enumeration (CPE); MITRE
  ④ Common Configuration Enumeration (CCE); MITRE
  ⑤ Common Vulnerabilities and Exposures (CVE); MITRE

• **Vulnerability measurement and scoring**
  ⑥ Common Vulnerability Scoring System (CVSS); Forum of Incident Response and Security Team (FIRST)
SCAP v1.0 Content

- Provided by the National Vulnerability Database (NVD)
- Managed by NIST and sponsored by DHS

National Vulnerability Database Version 2.2

NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables:

- National Checklist Program (automatable security configurations using OVAL)
- SCAP (program and protocol that NVD supports)
- SCAP Compatible Tools
- SCAP Data Feeds (CVE, CCE, CPE, CVSS, XCCDF, OVAL)
- Product Dictionary (CPE)
- Impact Metrics (CVSS)
- Common Weakness Enumeration (CWE)
SCAP component specifications interoperation

- A checklist uses **XCCDF** to describe what to evaluate
  - **OVAL** to perform the tests on the target system
  - **CPE** to identify platforms for which the checklist is valid and on which the tests will run
  - **CCE** to identify security configuration settings to be addressed or assessed in the checklist
  - **CVE** to refer to known vulnerabilities
- **CVSS** to rank the vulnerabilities
XCCDF

- A XCCDF document consists of **Rules** to be evaluated
  - **Profiles** can be used to bundle rules for particular types of systems
  - **Groups** allow multiple rules to be enabled or disabled at once
  - **Values** allow user-defined values for certain rules
XCCDF Sample

<!--  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  -->
<!--  ~~~ File Permissions Group ~~~  -->
<!--  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  -->

- <Group id="file_permissions_group">
  <title>File Permission Settings</title>
  <description>This group checks the permissions of specified files.</description>
  - <Rule id="regedit.exePermissions" selected="false" weight="10.0">
    <title>regedit.exe Permissions</title>
    <description>Failure to properly configure ACL file and directory permissions, allows the possibility of unauthorized and anonymous modification to the operating system and installed applications.</description>
    <ident system="http://cce.mitre.org">CCE-2175-8</ident>
    <ident system="cce.mitre.org/version/4">CCE-795</ident>
    - <check system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
      <check-content-ref href="example-winxp-oval.xml" name="oval:gov.nist.fdcc.xp:def:146"/>
    </check>
  </Rule>
</Group>
Open Vulnerability Assessment Language (OVAL)

- For SCAP, OVAL is commonly used to check the presence of vulnerabilities and insecure configurations
  - A set of instructions used to check for a security problem, is known as a Definition
Oval Definitions

• **Vulnerability Definitions**
  – Is a specific vulnerability present?

• **Patch Definitions**
  – is a particular patch appropriate for a system?

• **Inventory Definitions**
  – is a specific piece of software installed on the system?

• **Compliance Definitions**
  – Do conditions exist on a system necessary for compliance with a specific policy or configuration statement?
How OVAL works?

1. Configuration policy
   Government agencies such as NSA and NIST develop “Best Practices” policy for system security.

2. Security advisories
   CERT-CC, US-CERT, and other organizations publish security advisories that warn of current threats and system vulnerabilities.

Source: http://oval.mitre.org/oval/about/images/how_oval_works.pdf
How OVAL works?

2. Definitions are generated

Specific machine configuration details from Advisory and Policy documents are extracted and encoded as an OVAL Definition.

Source: http://oval.mitre.org/oval/about/images/how_oval_works.pdf
Security advisories
CERT-CC, US-CERT, and other organizations publish security advisories that warn of current threats and system vulnerabilities.

A commercial vulnerability scanner can read OVAL Definitions and use them to gather configuration information to generate a system characteristics file.

OVAL System Characteristics

Data collected from computers
OVAL Definitions are structured to indicate what configuration information needs to be collected from an individual system.

Source: http://oval.mitre.org/oval/about/images/how_oval_works.pdf
OVAL Analysis

Current state → Vulnerable state

Analysis
The OVAL Definitions from Step 2, and the System Characteristics from Step 3 are compared to determine if the current system state is vulnerable or not vulnerable.

Source:
http://oval.mitre.org/oval/about/images/how_oval_works.pdf
How OVAL works?

1. From Step 2, gather characteristics needed to analyze your system.

2. And from the analysis files, a vulnerability report is created. The OVAL Result file contains OVAL results.

Source: http://oval.mitre.org/oval/about/images/how_oval_works.pdf
Definition

- <definition id="oval:gov.nist.fdcc.xp:def:146" version="1" class="compliance">
  - <metadata>
    <title>Administrators and System User Have Full Access to the SYSTEMROOT/regedit.exe File</title>
    <affected family="windows">
      <platform>Microsoft Windows XP</platform>
    </affected>
    <reference source="http://cce.mitre.org" ref_id="CCE-2175-8" />
    <reference source="cce.mitre.org/version/4" ref_id="CCE-795" />
    <description>The Administrators group and the System user should have full access to the SYSTEMROOT/regedit.exe file and all other users should have no file access privileges</description>
  </metadata>
  - <criteria>
    <extend_definition comment="Microsoft Windows XP is installed"
      definition_ref="oval:gov.nist.fdcc.xp:def:2" />
    <criteria operator="AND">
      <criterion comment="The Administrators group is granted full access to the file regedit.exe by users not part of the Administrators group or the System user"
        test ref="oval:gov.nist.fdcc.xp:tst:249" />
      <criterion test ref="oval:gov.nist.fdcc.xp:tst:250" />
    </criteria>
  </criteria>
</definition>
Tests

  comment="The System user is granted full access to the file regedit.exe"
  check_existence="any_exist" check="all">
  <object object_ref="oval:gov.nist.fdcc.xp:obj:156" />
  <state state_ref="oval:gov.nist.fdcc.xp:ste:51" />
</fileeffectiverights53_test>
Object

  `<path datatype="string" var_ref="oval:gov.nist.fdcc.xp:var:4" />
  `<filename>`reqedit.exe`</filename>
  `<trustee_sid>`S-1-5-18`</trustee_sid>
  `</fileeffectiverights53_object>`
State

  comment="specified account is granted full control">
  <standard_delete datatype="boolean">1</standard_delete>
  <standard_read_control datatype="boolean">1</standard_read_control>
  <standard_write_dac datatype="boolean">1</standard_write_dac>
  <standard_write_owner datatype="boolean">1</standard_write_owner>
  <standard_synchronize datatype="boolean">1</standard_synchronize>
  <file_read_data datatype="boolean">1</file_read_data>
  <file_write_data datatype="boolean">1</file_write_data>
  <file_append_data datatype="boolean">1</file_append_data>
  <file_write_ea datatype="boolean">1</file_write_ea>
  <file_execute datatype="boolean">1</file_execute>
  <file_delete_child datatype="boolean">1</file_delete_child>
  <file_read_attributes datatype="boolean">1</file_read_attributes>
  <file_write_attributes datatype="boolean">1</file_write_attributes>
</file_effectiveRights53_state>
Common Platform Enumeration (CPE)

• CPE is a naming format and dictionary of hardware, operating systems, and applications
  – Based upon the generic syntax for Uniform Resource Identifiers (URI)
  – CPE includes
    • A formal name format
    • A method for checking names against a system
    • A description format for binding text and tests to a name
CPE Name Structure

cpe:/ {part} : {vendor} : {product} : {version} : {update} : {edition} : {language}

cpe:/a:acme:wizbang:1.0:update2:pro:en-us

cpe:/o:microsoft:windows_xp:::pro
Standard Configurations

• For a large infrastructure, the lack of a standard configuration on each node often leads to a administration nightmare

• Deployment of new software applications is difficult and unpredictable on different configurations

• Vulnerability and patch management can be significantly difficult without a common baseline
Standard Configurations

• Mandated baselines, or minimum configuration of all systems in a critical infrastructure
  – DISA gold disk
  – Federal Desktop Core Configuration (FDCC)
  – DoD Security Technical Implementation Guides (STIGS)
  – NSA Security Guides
  – NIST SP 800-68: Guidance for Securing Microsoft Windows XP Systems for IT Professional
  – Center for Internet Security (CIS) baselines
Common Configurations Enumeration (CCE)

• CCE a nomenclature and dictionary of security software configurations
  – CCE identifiers link natural language, prose-based configuration guidance documents and machine-readable or executable capabilities such as configuration audit tools

• Does not introduce new entries but maintains traceability to different standard configurations
# CCE entries for IE7

<table>
<thead>
<tr>
<th>CCE ID</th>
<th>CCE Description</th>
<th>CCE Parameters</th>
<th>CCE Technical Mechanisms</th>
<th>Old v4 CCE ID</th>
<th>FDCC IE7 XCCDF (fdcc-accepted-content-20080110/fdcc-ie7-xccdf.xml)</th>
<th>FDCC IE7 OVAL (fdcc-accepted-content-20080110/fdcc-ie7-oval.xml)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE-4017-0</td>
<td>The &quot;Security Zones: Use Only Machine Settings&quot; setting should be configured correctly.</td>
<td>(1) enabled/disabled</td>
<td>HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Use_HKLM_onlyLocal Internet Options: GPO Settings \Computer Configuration \User Configuration \Network \Internet Explorer, Registry Keys: [HKLM]HKCU\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Security \HKLM_only</td>
<td>CCE-5</td>
<td>use_only_machine_settings_local_computer</td>
<td>oval.gov.nist.fdcc.ie7:def.1277</td>
</tr>
<tr>
<td>CCE-3924-8</td>
<td>Internet Explorer Processes (Restrict ActiveX Install)</td>
<td>(1) enabled/disabled</td>
<td>HKLM\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL(Reserved), HKLM\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL\explorer.exe, HKLM\Software\Policies\Local Internet Options: GPO Settings \Computer Configuration \User Configuration \Network \Internet Explorer\Internet Control Panel\Security Features\Restrict ActiveX Install, Registry Keys: [HKLM]HKCU\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL(Reserved), [HKLM I HKCU\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL\explorer.exe, [HKLM I HKCU\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL\explorer.exe, [HKLM I HKCU\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_RESTRICT_ACTIVEXINSTALL\explorer.exe</td>
<td>CCE-119</td>
<td>IEProcesses_RestrictActiveXInstall\LocalComputer</td>
<td>oval.gov.nist.fdcc.ie7:def.668</td>
</tr>
</tbody>
</table>
Common Vulnerabilities and Exposures (CVE)

• One name for a vulnerability or exposure
• A dictionary rather than a database
• Common language to share tool reports and vulnerability information among different entities

– TOTAL CVEs: 42100 and counting..
– On average ~ 15 to 20 added every day
Trying to capture what went wrong....
CVE-1999-0067
CGI phf program allows remote command execution through shell metacharacters.

Source: http://cve.mitre.org/
Common Vulnerability Scoring System (CVSS)

• Determining the severity of a vulnerability can be a highly subjective process

• Common Vulnerability Scoring System (CVSS) provides an open specification for measuring the relative severity of software vulnerabilities
  – Quantitative model
  – Repeatable measurement
  – Transparency of vulnerability characteristics that influence the computed scores
CVSS Calculator

- **Base**: the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.
- **Temporal**: Characteristics of a vulnerability that change over time but not among user environments
- **Environmental**: Characteristics of a vulnerability that are relevant and unique to a particular user’s environment

Source: [http://nvd.nist.gov/cvss.cfm](http://nvd.nist.gov/cvss.cfm)
CVSS computations

- CVSS Calculator
- Equations for the computations

http://nvd.nist.gov/cvssv2.htm
SCAP Usage Scenarios

• Automating checks for known vulnerabilities
• Automating the verification of security configuration settings
• Generating reports that link low-level settings to high-level requirements
• Vulnerability tracking and prioritization
• Scoring and Measurement
• Many others… (malware detection, remediation, etc..)
Implications for software vendors

• Register and use standardized identifiers

• Make the state of security settings available through APIs
  – Be very very careful!

• Develop security software with SCAP validation requirements in mind
Possible SCAP Roles

- Checklist Author (XCCDF)
- Definition Author (OVAL)
- Data Source Maintainer (XCCDF, OVAL, CVE, CCE, CPE)
  - Vulnerability, Patch, Compliance, Inventory enumerations
- Dispatcher (CVSS)
  - Prioritization of tasks based on a uniform vulnerability measuring instrument
- Assessor (Tool Execution and Reporting)
What about People and Process?

• We have automated technology assessment
  – ~60% of NIST 800-53 controls cannot be automated
    Source: http://nvd.nist.gov/scap/docs/SCAP-webpp-10182006.ppt
  – What about people and process?

• SCAP 2.0 has OCIL in the works
  – The Open Checklist Interactive Language (OCIL)
    • Expressing a set of questions to be presented to a user
    • Corresponding procedures to interpret responses to those questions
    • http://scap.nist.gov/specifications/ocil/
Should I pay attention to SCAP?

• The U.S. Federal Government, in cooperation with academia and private industry, is adopting SCAP and encourages its use in support of security automation activities and initiatives.

• ....successfully manage systems in accordance with risk management frameworks such as NIST Special Publication 800-53; Department of Defense (DoD) Instruction 8500.2; and the Payment Card Industry (PCI) framework.

Source: NIST 800-126
Common Attack Pattern Enumeration and Classification (CAPEC)

- A shared indexing standard for common attacks patterns used in exploits or malware

- Attack patterns
  - Capture and communicate an attackers perspective
    - Common vocabulary to express attack vectors
  - List of common methods to exploit vulnerabilities
  - A “destructive” way of thinking
    - Know your enemy. Defense alone is not enough.

- [http://capec.mitre.org/](http://capec.mitre.org/)
Malware Attribute Enumeration and Characterization (MAEC)

- A standardized language for encoding and communicating high-fidelity information about malware based upon attributes such as behaviors, artifacts, and attack patterns
- Eliminate the ambiguity and inaccuracy that currently exists in malware descriptions and by reducing reliance on signatures

- [http://maec.mitre.org/](http://maec.mitre.org/)
# MAEC example

## Conficker.B Self-defense

<table>
<thead>
<tr>
<th>Low-level Observable</th>
<th>Mid-level Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify Registry Key. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BITS, Start = &quot;4&quot;</td>
<td>Disables Background Intelligent Transfer Service (BITS)</td>
</tr>
<tr>
<td>Modify Registry Key. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ERSvc, Start = &quot;4&quot;</td>
<td>Disable Windows Error Reporting Service</td>
</tr>
<tr>
<td>Modify Registry Key. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wscsvc, Start = &quot;4&quot;</td>
<td>Disables Windows Security Center Service</td>
</tr>
<tr>
<td>Modify Registry Key. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced, Hidden = &quot;2&quot;</td>
<td>Hide Hidden Files Even After Changing Setting in Folder Options</td>
</tr>
<tr>
<td>Modify Registry Key. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\hidden\SI-HOWALL CheckedValue = &quot;0&quot;</td>
<td>Deletes All System Restore Points</td>
</tr>
<tr>
<td>Call Library Function: srcclient.dll, ResetSR()</td>
<td></td>
</tr>
</tbody>
</table>
Putting it all together

- Asset
  - CCE
  - CPE
  - OVAL
  - XCCDF
- Threat
  - CAPEC
  - MAEC
- Vulnerability
  - CVE
  - CVSS
- Risk
  - CVE
  - CVSS
Thank you for your Attention