Security Content Automation Protocol (SCAP) Introduction

Andrew Buttner

October 26th, 2009

Goals

- Introduce SCAP to those just coming on board.
- Give insight as to what SCAP is trying to do and what benefits it can provide.
- Provide an in-depth technical look at the specifications that are part of SCAP

Agenda

- What is SCAP
- SCAP Lifecycle
- Introduction to the Specifications
  - Enumerations (CPE, CVE, CCE)
  - Languages (OVAL, XCCDF)
  - Scoring Systems (CVSS)
- SCAP Validation
- FDCC and SCAP

What is SCAP

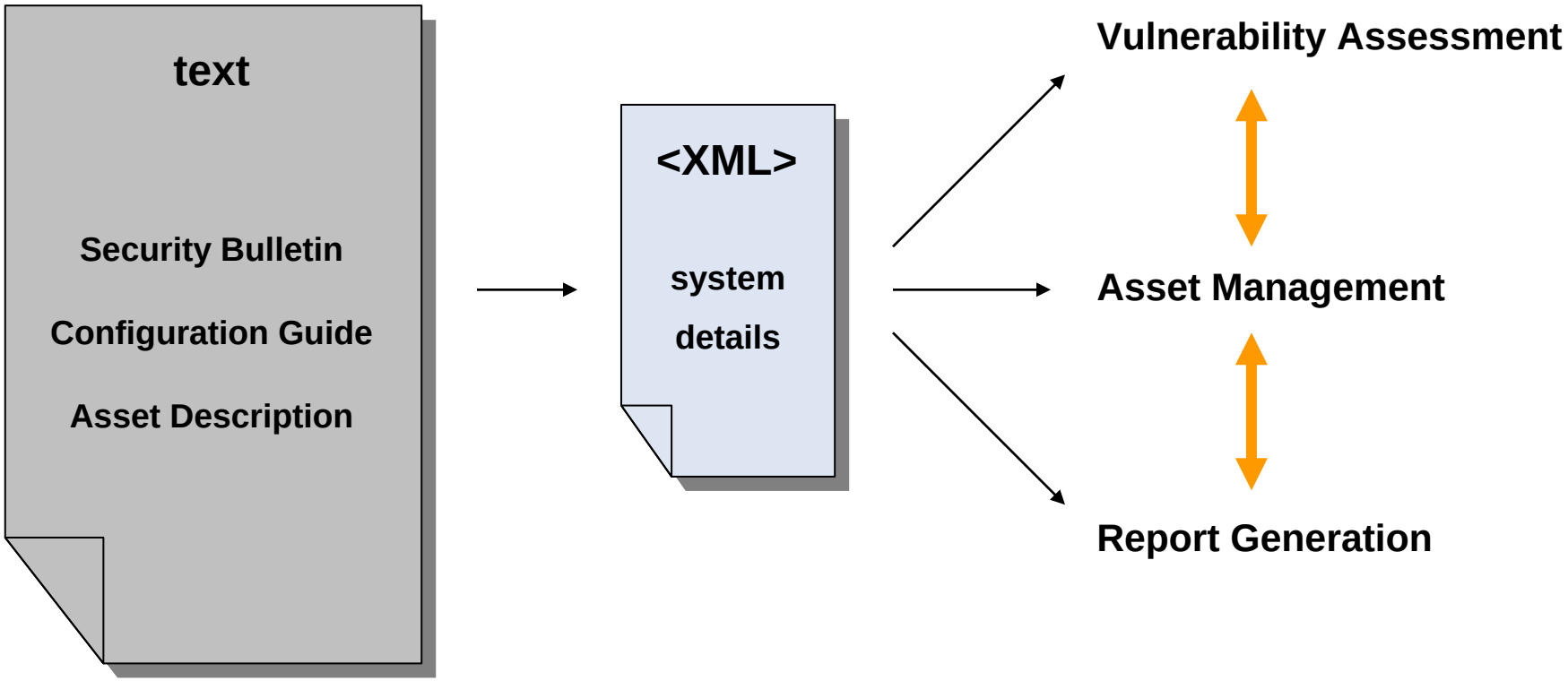
Configuration Checklists

- assist users in configuring IT products
- more protection
  - than the installed out-of-the-box defaults
- greater levels of product security
  - protection from future threats
- peace of mind

Need For Automation

- complex guidance
  - difficult to determine the applicability
- large number of systems
- high number of security-related configuration settings
- verify the security posture regularly
- need to respond quickly to new threats

Process Overview

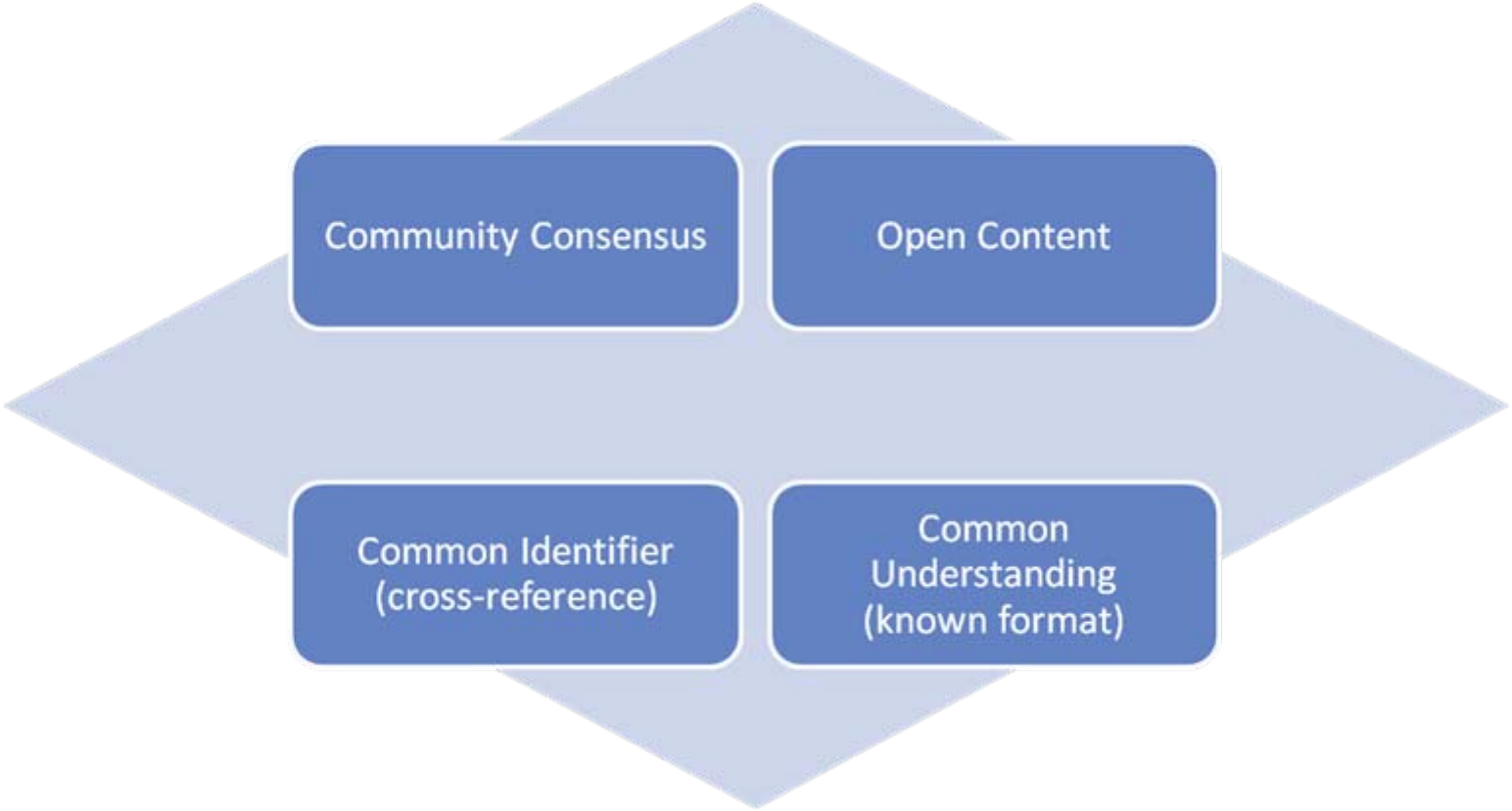


SCAP

- Created to bring together existing specifications and to provide a standardized approach to maintaining the security of enterprise systems.
    - vulnerability and patch management
    - policy compliance evaluation
    - system compromise
- SCAP ...
    - is a suite of individually maintained, open specifications
    - defines how these specificatiosn are combined
    - includes standardized reference data -- SCAP Content

MITRE

Community Involvement



SCAP 1.0 Specifications

- Extensible Configuration Checklist Description Format — XCCDF
- Open Vulnerability and Assessment Language — OVAL®
- Common Platform Enumeration — CPE™
- Common Vulnerabilities and Exposures — CVE®
- Common Configuration Enumeration — CCE™
- Common Vulnerability Scoring System — CVSS

Question --> Standard

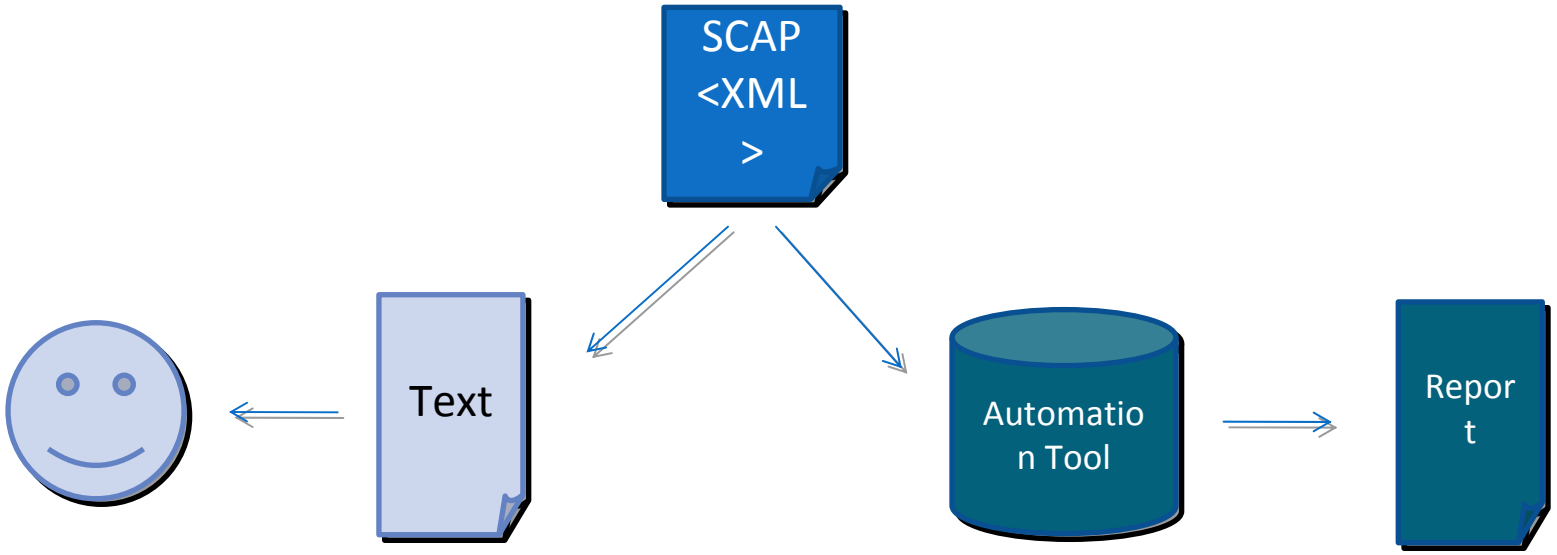
What IT systems do I have in my enterprise?	• CPE
What vulnerabilities do I need to worry about?	• CVE
What vulnerabilities do I need to worry about RIGHT NOW?	• CVSS
How can I configure my systems more securely?	• CCE
How do I define a policy of secure configurations?	• XCCDF
How can I be sure my systems conform to policy?	• OVAL

Remembering the Acronyms

What IT systems do I have in my enterprise?	• CPE (Platforms)
What vulnerabilities do I need to worry about?	• CVE (Vulnerabilities)
What vulnerabilities do I need to worry about RIGHT NOW?	• CVSS (Scoring System)
How can I configure my systems more securely?	• CCE (Configurations)
How do I define a policy of secure configurations?	• XCCDF (Configuration Checklists)
How can I be sure my systems conform to policy?	• OVAL (Assessment Language)

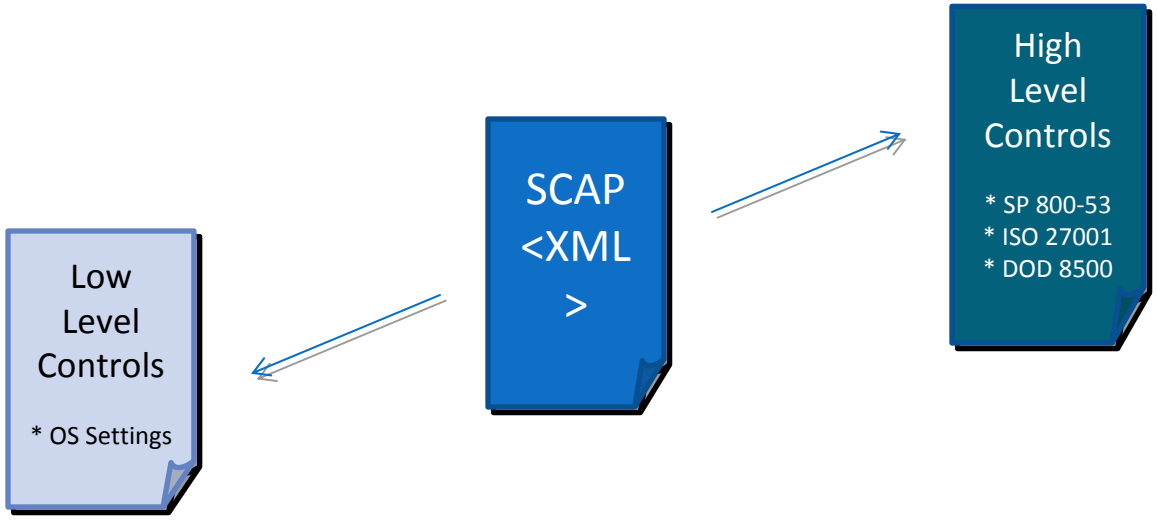
Uses of SCAP - 1

- Security Configuration Verification
  - SCAP enables both human and machine readable security configuration checklists that can be processed by SCAP-validated authenticated configuration scanners.



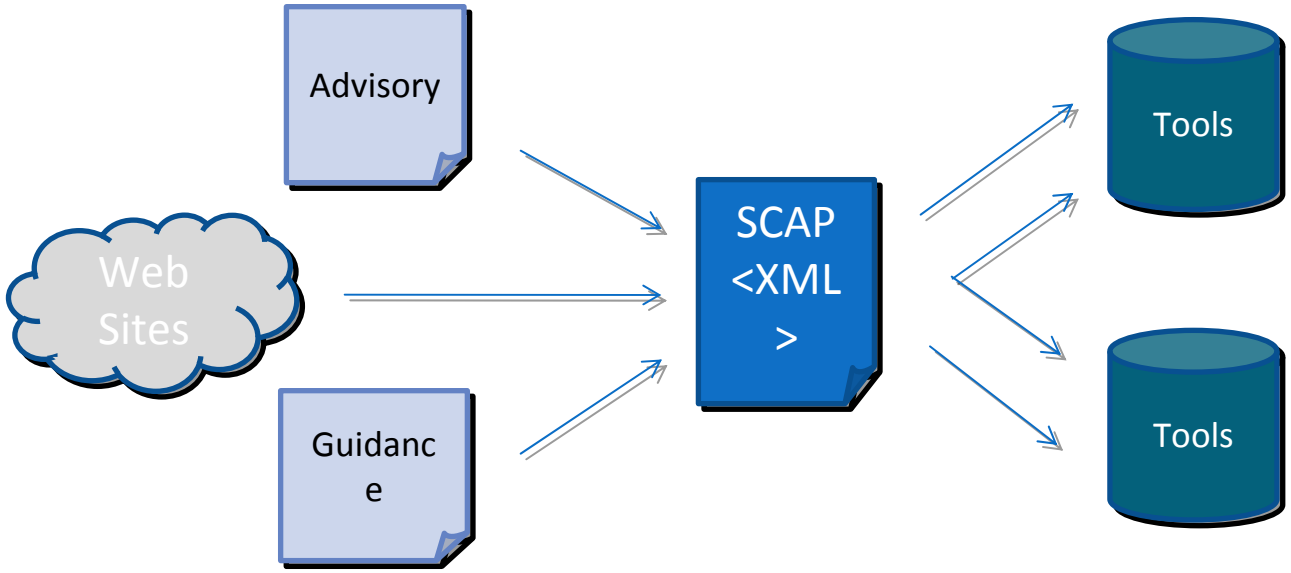
Uses of SCAP - 2

- Requirements Traceability
  - SCAP enables traceability between low-level controls and high-level requirements and can be used to demonstrate that organizations have implemented their security controls in accordance with requirements.



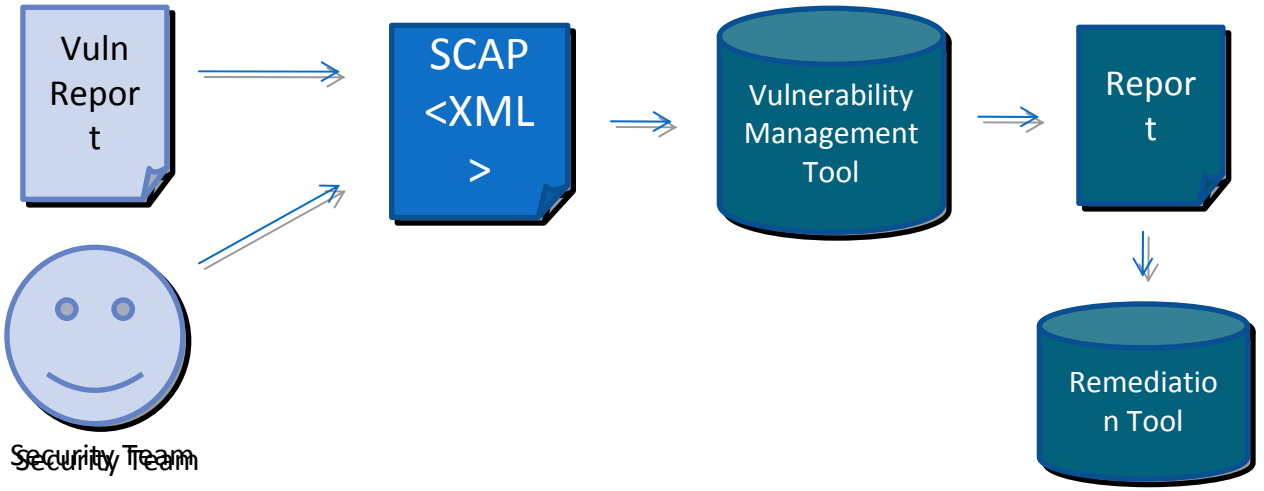
Uses of SCAP - 3

- Standardized Security Enumerations
  - Through the use of standardized enumerations, SCAP makes it easier to use security tools, share information, and issue guidance to address security issues.



Uses of SCAP - 4

- Vulnerability Measurement
  - SCAP enables quantitative and repeatable measurement and scoring of software flaw vulnerabilities across systems allowing organizations to institute consistent and repeatable mitigation policies throughout the enterprise.



Taking Advantage of SCAP

- Software Developer
  - register and use standardized identifiers (CPE)
  - make security settings available through automation
  - develop software with SCAP validation requirements in mind

Taking Advantage of SCAP

- SCAP Content Producers
  - develop security checklists in SCAP format
  - contribute checklists to the National Checklist Program
  - participate in developing OVAL

SCAP Documentation

SP800-117: Adopting and Using Security Content Automation Protocol

SP800-126: Security Content Automation Protocol Specification

SP800-70 Rev 1: DRAFT National Checklist Program for IT Products--Guidelines for Checklist Users and Developers

IR-7511: DRAFT Security Content Automation Protocol (SCAP) Validation Program Test Requirements

IR-7435: The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems

IR-7275 Rev 3: Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.1.4

SP800-117: Adopting and Using Security Content Automation Protocol

- **Purpose and Scope**: SP 800-117 provides an overview of SCAP, focusing on how organizations can use SCAP-enabled tools to enhance their security posture. It also explains how IT product and service vendors can adopt SCAP's capabilities within their offerings.

- **Audience**: Individuals who have responsibilities for maintaining or verifying the security of systems in operational environments. This includes mid-level management, chief information security officers, and technical directors within Federal and state governments and other large organizations; software and hardware vendor product managers, and auditors.

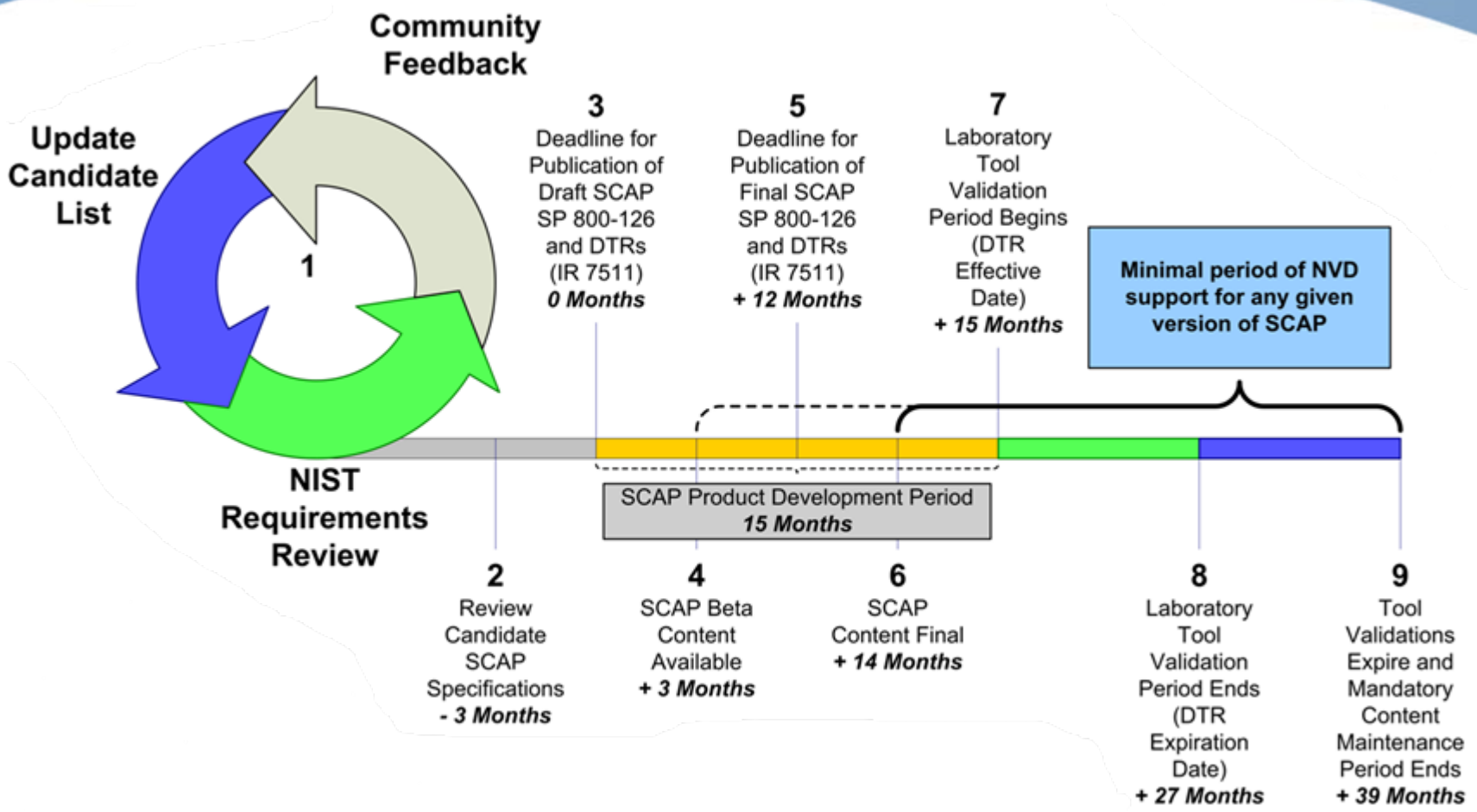
SP800-126: Security Content Automation Protocol Specification

- **Purpose and Scope**: SP800-126 defines the technical specification for SCAP. This document describes the basics of the SCAP component specifications and their interrelationships. It also defines the characteristics of SCAP content, as well as all other requirements for SCAP that are not defined in the individual SCAP component specifications.

- **Audience:** Technically focused individuals who have responsibilities for developing or testing applications or processes that leverage SCAP. This includes security team members, guidance writers, and managers in charge with overseeing the technical delivery of related teams.

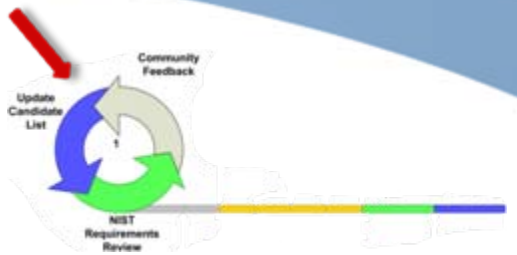
SCAP Lifecycle

SCAP Lifecycle



MITRE

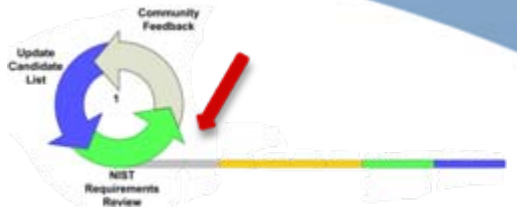
Lifecycle -- Step 1



Feedback Loop

This step allows a specification to mature and demonstrate value in terms of operational use within organizations, community feedback, vendor use and adoption, etc., without imposing a time limit.

Lifecycle -- Step 2



Review Candidates

As specifications evolve, NIST may consider a new or modified specification for SCAP adoption. Periodically, a specification reaches a degree of maturity, adoption, and utility where NIST considers it a potential candidate for SCAP.

Lifecycle -- Step 3

Draft NIST SP 800-126

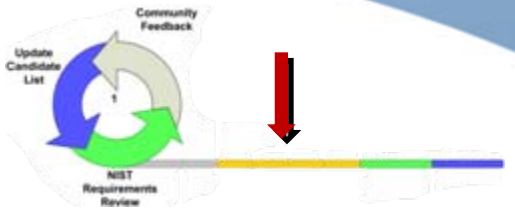
Candidate specifications that are identified as potential SCAP specifications will be included in the Draft NIST SP 800-126. This draft publication serves as the official notice to the community that SCAP will include new or updated specifications. Review of this draft will follow the NIST publication review process.

Lifecycle -- Step 4

SCAP Beta Content

After publishing the draft NIST SP 800-126, sample, beta quality content, for data streams for which NIST is responsible will be produced by NIST for use and testing by the community.

Lifecycle -- Step 5



Final NIST SP 800-126

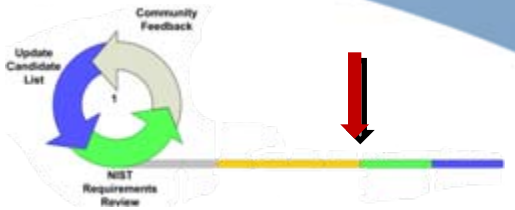
No later than twelve months after the draft NIST SP 800-126 is published, it will become official.

Lifecycle -- Step 6

SCAP Final Content

Related to step 4, content originally published as beta will become final at this time. The community can expect that the content will be released in various maturing versions including several versions of alpha, several versions of beta, and then in a final version at this time.

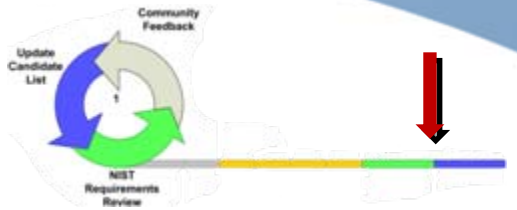
Lifecycle -- Step 7



Product Validation Begins

After the finalization of NIST SP 800-126 accredited laboratories begin testing products using the finalized SP 800-126 and IR 7511 as official references. Products seeking new validations and those seeking re-validations can be tested using the version of NIST SP 800-126 and the related validation documents.

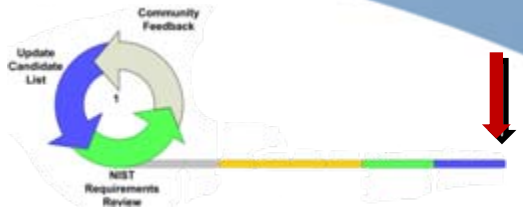
Lifecycle -- Step 8



Product Validation Ends

12 months from the start of step 7, product testing according to the previous versions of NIST SP 800-126 and related validation documentation ends. Future product testing will use the next version of NIST SP 800-126.

Lifecycle -- Step 9



Product Validations Expire

Product validations are valid for one year from the time the validation was originally awarded. As a result of this, NIST will maintain all SCAP content in their control for a minimum period of twelve months from the date of step 8.

SCAP Lifecycle Timing

- SCAP version 1.0
  - currently at step 5 (final 800-126)
  - product validation begins January 2010 (planned)
  - supported through end of 2012 (planned)
- SCAP version 1.1
  - currently at step 3 (draft 800-126)
  - final in September 2010 (planned)
  - product validation begins January 2011 (planned)
  - supported through end of 2013 (planned)
- SCAP version 1.2
  - currently at step 1 (feedback loop)
  - draft available in September 2010 (planned)

Enumerations – CPE, CVE, CCE

Crisis Event: Solar Sunrise (1998)

Lessons From History - Enumerations

- Identifiers gain broader (use-case) acceptance than schema based data objects
  - No existing IDS schema (CIDF or IETF/IDWG)
  - CVEs used in IDS, patch, vuln assessment, malware …
- Enumerations tend to emerge in established domains
  - Common among the sciences and manufacturing sectors
  - Appear AFTER communication patterns form, not BEFORE
- Structured names inherit baggage of lightweight schemas
  - Multiple structured names for same “thing” not uncommon
  - E.g. many color naming schemes, geo-location

Enumerations Defined

- a naming scheme
  - specific entities identified using a common term
- defined set of things
  - seen to be members of the same category
- used by multiple groups
  - communicate with each other
  - coordinate activities
- just enumerate the entities
  - trying to do more leads to many problems related to different use cases

By keeping things simple, we can accomplish a lot.

MITRE

Benefits of Enumerations

- Enable faster, more accurate correlation
  - Standardized identifiers used in:
    - Databases
    - Tools
    - Guidance
- Facilitate information exchange
  - Requirements – what do we need to check for?
  - Reporting – what did we find?
  - Roll-up – how do standard elements map to local needs?
  - Information more easily flows:
    - Across the configuration management lifecycle
    - Through different communities of interest
- Allow increased automation
  - Diverse tools can share input and output

IA Data Without Enumerations

- data correlation is:
  - Mostly manual
  - Key word driven
  - Costly
  - Error prone
  - Pair-wise between data sets
  - Not scalable

- result:
  - Data is locked in proprietary repositories

IA Data With Enumeration

- common identifiers:
  - Community agree upon “tags”
  - Easily added to legacy repositories & tools

- KEY: common identification enables correlation!
  - Faster
  - More accurate
  - Less expensive

Enumerated Entities in SCAP

- CVE - Vulnerabilities
  - CVE-2006-4838

Description: Multiple cross-site scripting (XSS) vulnerabilities in DCP-Portal SE 6.0 allow remote attackers to inject arbitrary web script or HTML via the (1) root_url and (2) dcp_version parameters in (a) admin/inc/footer.inc.php, and the root_url, (3) page_top_name, (4) page_name, and (5) page_options parameters in (b) admin/inc/header.inc.php

- CCE - Configuration Settings
  - CCE-2116-2

Description: The "restrict guest access to application log" policy should be set correctly.
Parameters: enabled/disabled

- CPE - Platforms
  - cpe:/o:microsoft:windows_xp:::pro

Title: Microsoft Windows XP Professional

Common Platform Enumeration (CPE™)

- CPE Name
  - identifies a platform type
    - does not ID a system
  - ideally associated with an OVAL Inventory Definition
- CPE Language
  - used to combine CPE Names to identify complex platform types
- CPE Dictionary
  - collection of known CPE Names

CPE Status

Sponsor	**NSA**
Community Type	**Open Working Group**
Maturity	**- Concepts mature, content in development** **- Version 2.2 released Mar 11, 2009**
Adoption	**- Early stages** **- Used by NVD, FDCC** **- 28 SCAP Validated products**

CPE Name Format

- repeatable format
  - GOAL: 2 people in different rooms will come up with the same name
  - REALITY: common data point that 2 parties can use to represent the same platform related concept

- name is built by using known information
  - 7 (optional) components

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

Prefix Property

- set of platforms identified by a long name should be a subset of the set of platforms identified by a shorter initial portion of that same name
  - called the “prefix property”
  - allows matching to take place

For example, the platforms identified by:

cpe:/o:redhat:enterprise_linux:4

would be a subset of:

cpe:/o:redhat:enterprise_linux

MITRE

Dictionary

- Collection of known CPE Names
  - help users determine which names exists
  - help those creating new names
  - enough information to identify the platform
    - others can build more elaborate repositories based off dictionary

- Hosted by NIST at:

  http://nvd.nist.gov/cpe.cfm

CPE Resources

- Web site: http://cpe.mitre.org

- Mailing list: cpe-discussion-list
  - Open forum for developing the specification
  - registration form
    - http://cpe.mitre.org/registration.html

Common Vulnerabilities and Exposures (CVE®)

- Dictionary of standardized descriptions for vulnerabilities and exposures
  - Over 38,000 entries
- Publicly accessible for review or download from the Internet

ID: CVE-2007-1751

Description: Microsoft Internet Explorer 5.01, 6, and 7 allows remote attackers to execute arbitrary code by causing Internet Explorer to access an uninitialized or deleted object, related to prototype variables and table cells, aka "Uninitialized Memory Corruption Vulnerability."

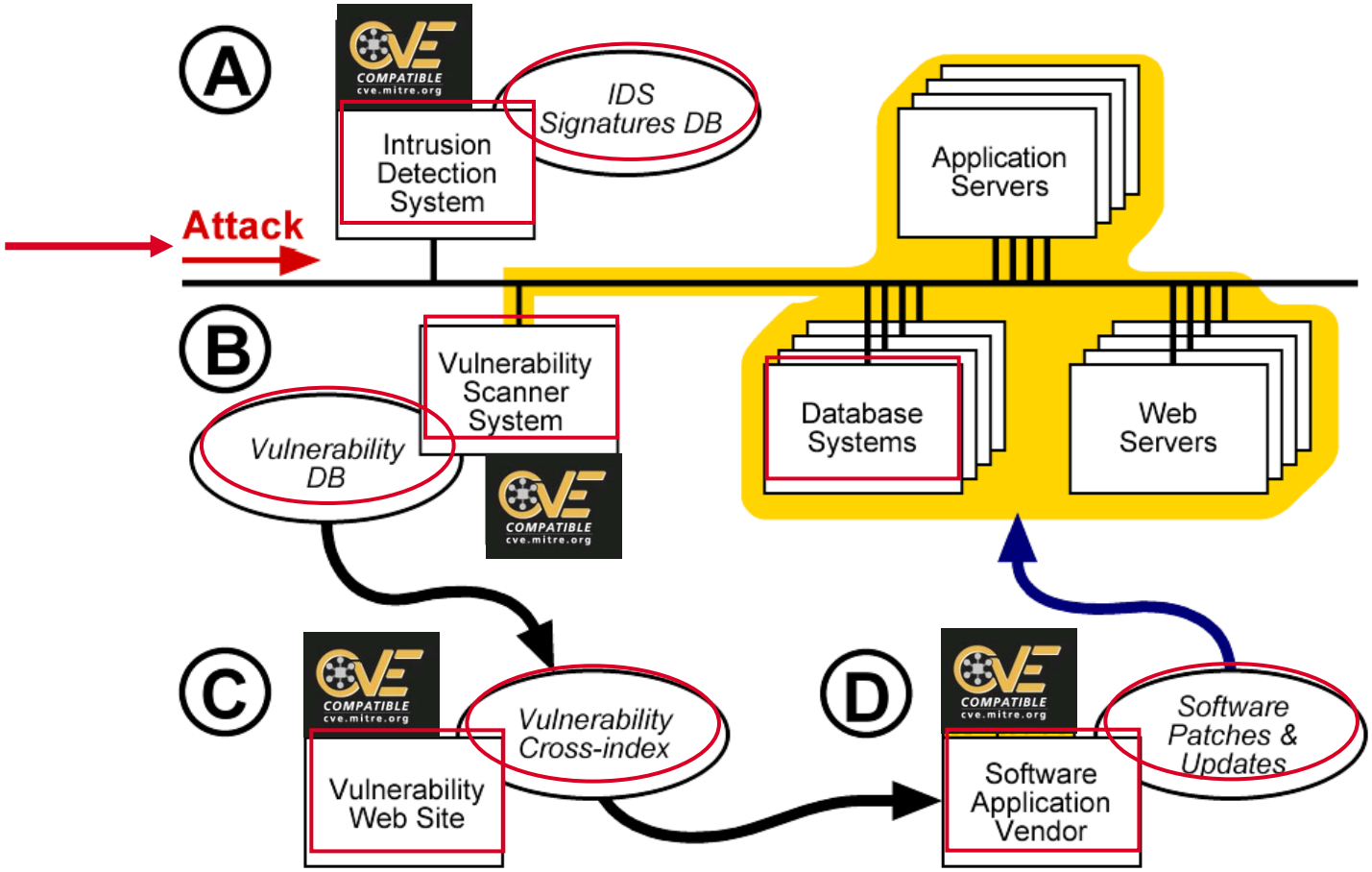
Reference: BUGTRAQ : 20070612 ZDI-07-038 - Microsoft Internet Explorer - Prototype Dereference Code Execution Vulnerability

Reference: MS : MS07-033

CVE Status

Sponsor	**DHS**
Community Type	**Editorial Board** **- Membership by invitation / nomination**
Maturity	**Mature**
Adoption	**Widespread** **- Over 280 products in 27 countries** **- Over 75 officially compatible**

Leveraging CVE compatibility

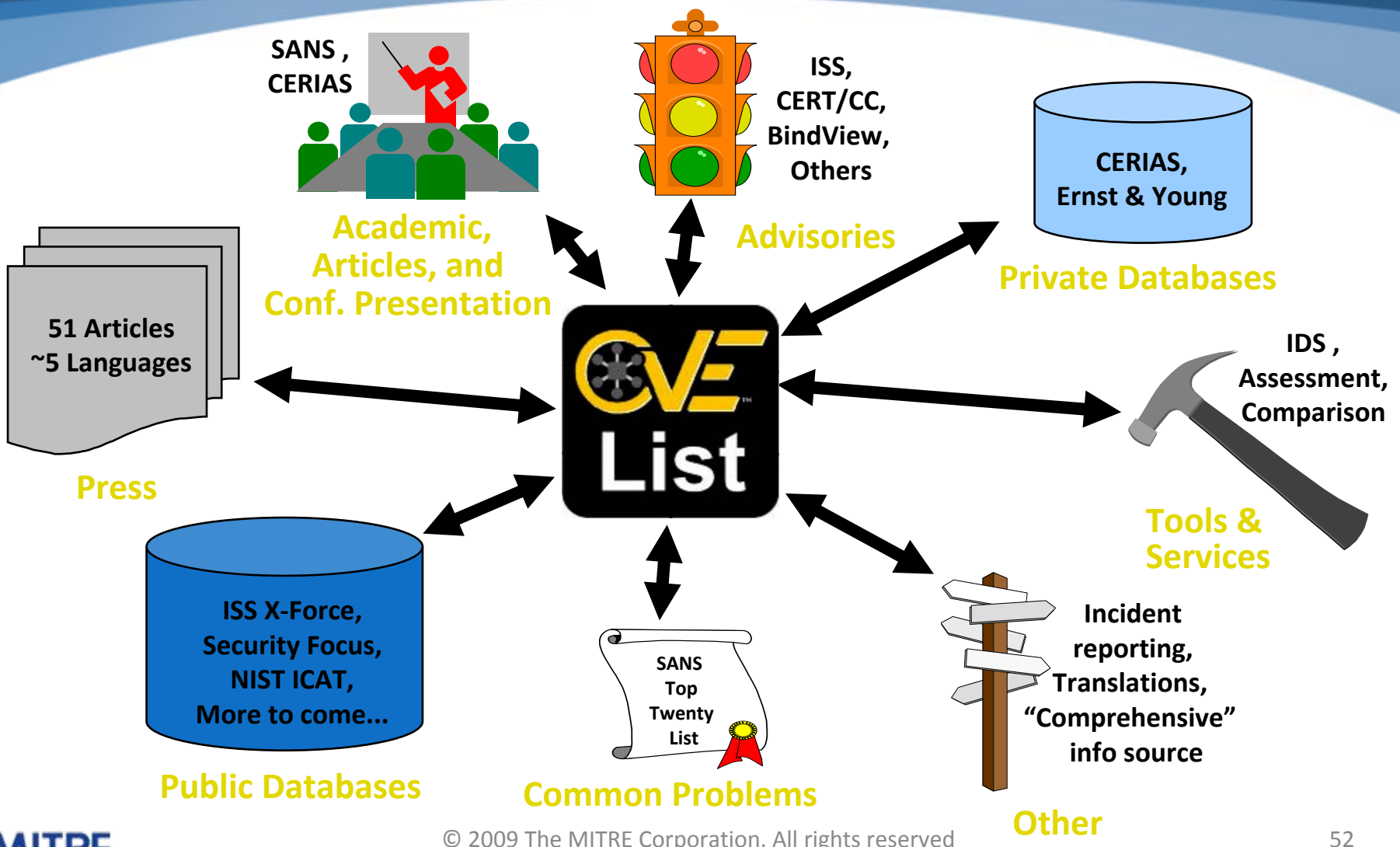


CVE List

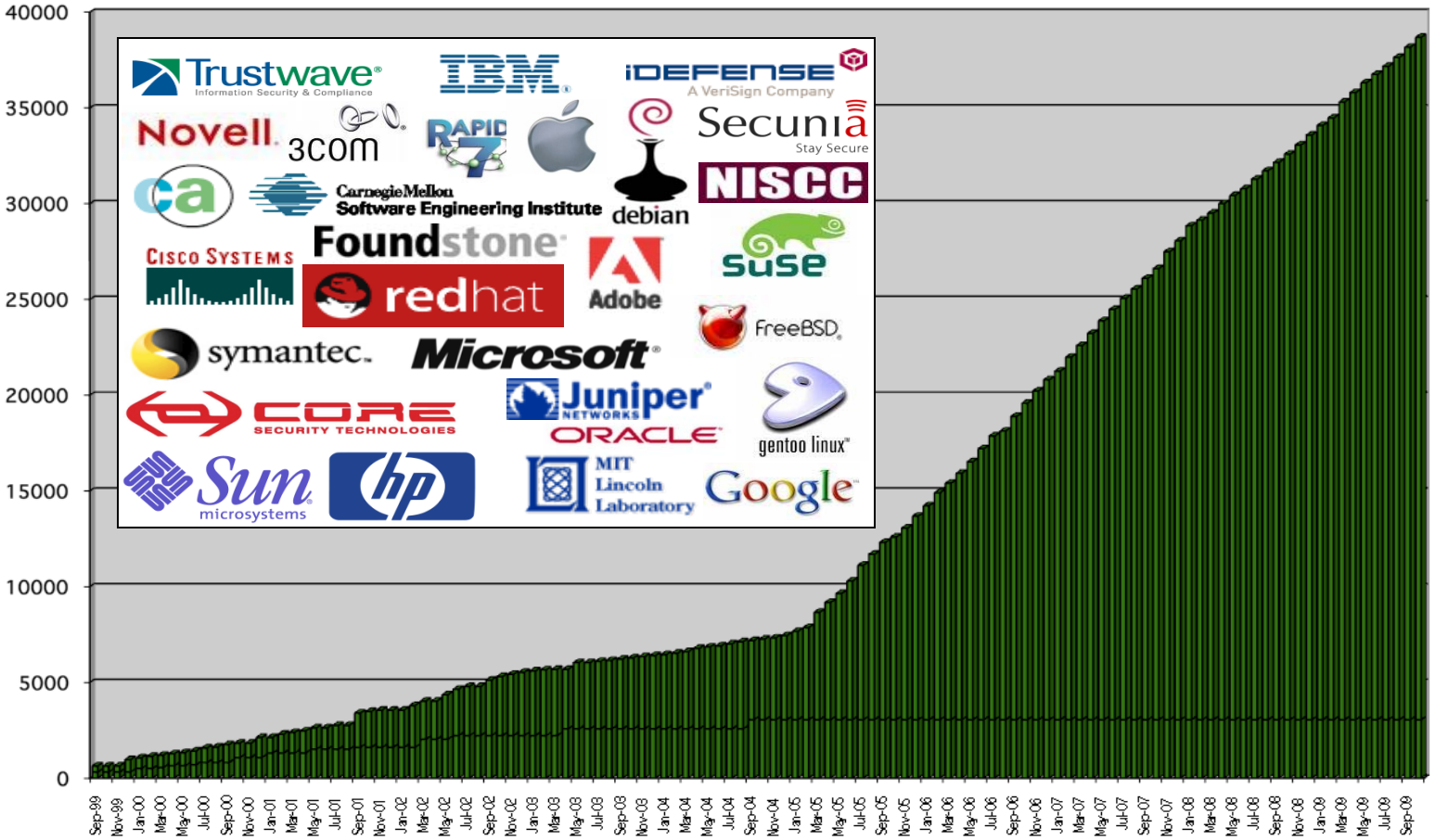
- List of all known CVE identifiers
  - 38,871 (as of October 16, 2009)
  - hosted at http://cve.mitre.org
  - xml feed

- NVD at NIST provide full search capabilities
  - additional metadata
  - hosted at http://nvd.nist.gov

The Center of Many Activities



CVE Numbering Authorities



Common Configuration Enumeration (CCE™)

- Assigns standardized identifiers to configuration issues, allowing comparability and correlation

ID:	CCE-3121-1
Description:	The "restrict guest access to application log" policy should be set correctly.
Technical Mechanisms:	(1)HKLM\SYSTEM\CurrentControlSet\Services\EventLog\Application\RestrictGuestAccess (2) defined by Group Policy
Parameter:	enabled/disabled

MITRE

CCE Status

Sponsor	**NSA**
Community Type	**Open Working Group**
Maturity	**- Concepts mature, content in development** **- Version 5 released Mar 5, 2008**
Adoption	**- Early stages** **- Microsoft security (Office 2007, Server 2008)** **- Primary identifier for FDCC** **- 28 SCAP Validated products**

The Identifier

The use of CCE-IDs as tags provide a bridge between natural language, prose-based configuration guidance documents and machine-readable or executable capabilities such as configuration audit tools.

- last digit is a check digit
- assigned on per platform basis

Descriptions

- a humanly understandable description of the configuration issue
- describes the configuration control
  - but does not assert a recommendation

Technical Mechanisms

- the technical setting that is being identified
  - for any given configuration issue there may be one or more ways to implement the desired result
- specific mechanisms
  - registry keys
  - group policy paths
  - api calls

Parameters

- parameters that would need to be specified in order to implement a CCE on a system
  - describes the possible values or the conceptual range of values
- the human readable notation
  - “enabled” instead of “1”

Enumerations - Creation

- content teams ensure uniqueness
- leverage vendor and community knowledge
- regular updates to official lists
- feedback channel to report issues

Summary

When dealing with information from multiple sources, use of consistent identifiers can

- improve data correlation
- enable interoperability
- foster automation
- and ease the gathering of metrics for use in situation awareness, IT security audits, and regulatory compliance.

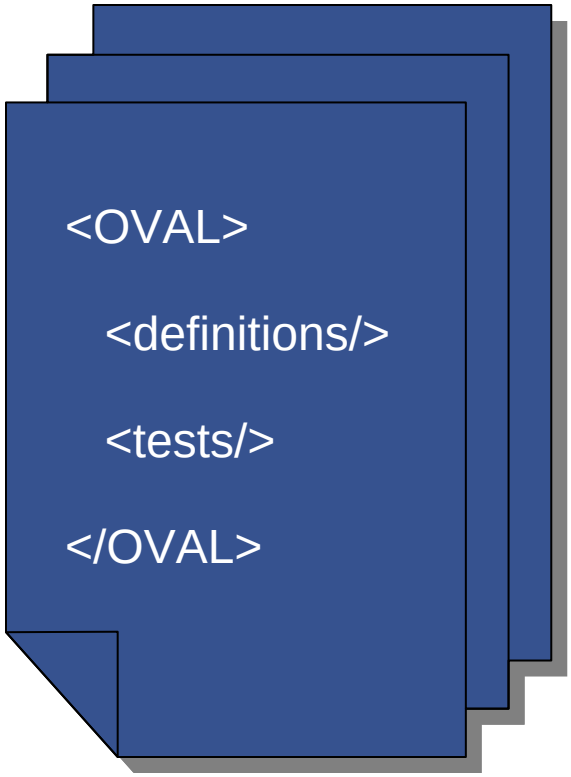
Languages – XCCDF, OVAL

Why Languages

- Use a standardized format to ensure guidance is easily consumed by a broad audience.
  - assessment tools
  - reporting
  - system administrators

Benefits

- machine readable document
  - less errors due to human translation
- immediate response
  - through automation
- interoperability
  - vendor neutral languages
- open to the user



XML

- XML instance document
  - the data being encoded
  - what is passed around by tools
- XML schema
  - defines the structure of an instance document
  - allows a tool to know what to expect

Introduction to OVAL

"Open Vulnerability and Assessment Language"

What is OVAL?

- XML language framework for assertions
- Can describe many different machine states
  - Vulnerable
  - Compliant
  - Installed application

An international, information security, community standard to promote open and publicly available security content, and to standardize the transfer of this information across the entire spectrum of security tools and services.

OVAL Language

- Standardizes the three main steps of the assessment process
  - **Representing** configuration information of systems for testing
    - characteristics of the system
  - **Analyzing** the system for the presence of a specified machine state
    - defining how to check for a state
  - **Reporting** the results of the assessment
    - results

- More than just compliance, can describe many states:
  - Vulnerable
  - Compliant
  - Installed application
  - Patched

http://oval.mitre.org/language

The OVAL Process

1

Security advisories

Vendors and leading security organizations publish security advisories that warn of current threats and system vulnerabilities.

Configuration policy

Government agencies such as NSA and NIST develop "Best Practices" policy for system security.

2

OVAL Definitions

Definitions are generated

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

3

Data collected from computers

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

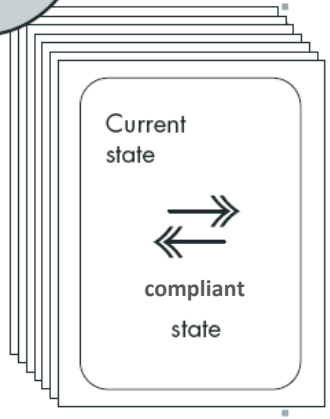
OVAL System Characteristics

OVAL Analysis

4

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.



OVAL Results

5

Analysis results

Results of analysis are formatted as an OVAL Results document.

OVAL Language: Schemas

OVAL Definitions Schema

- Framework for logical assertions about a system

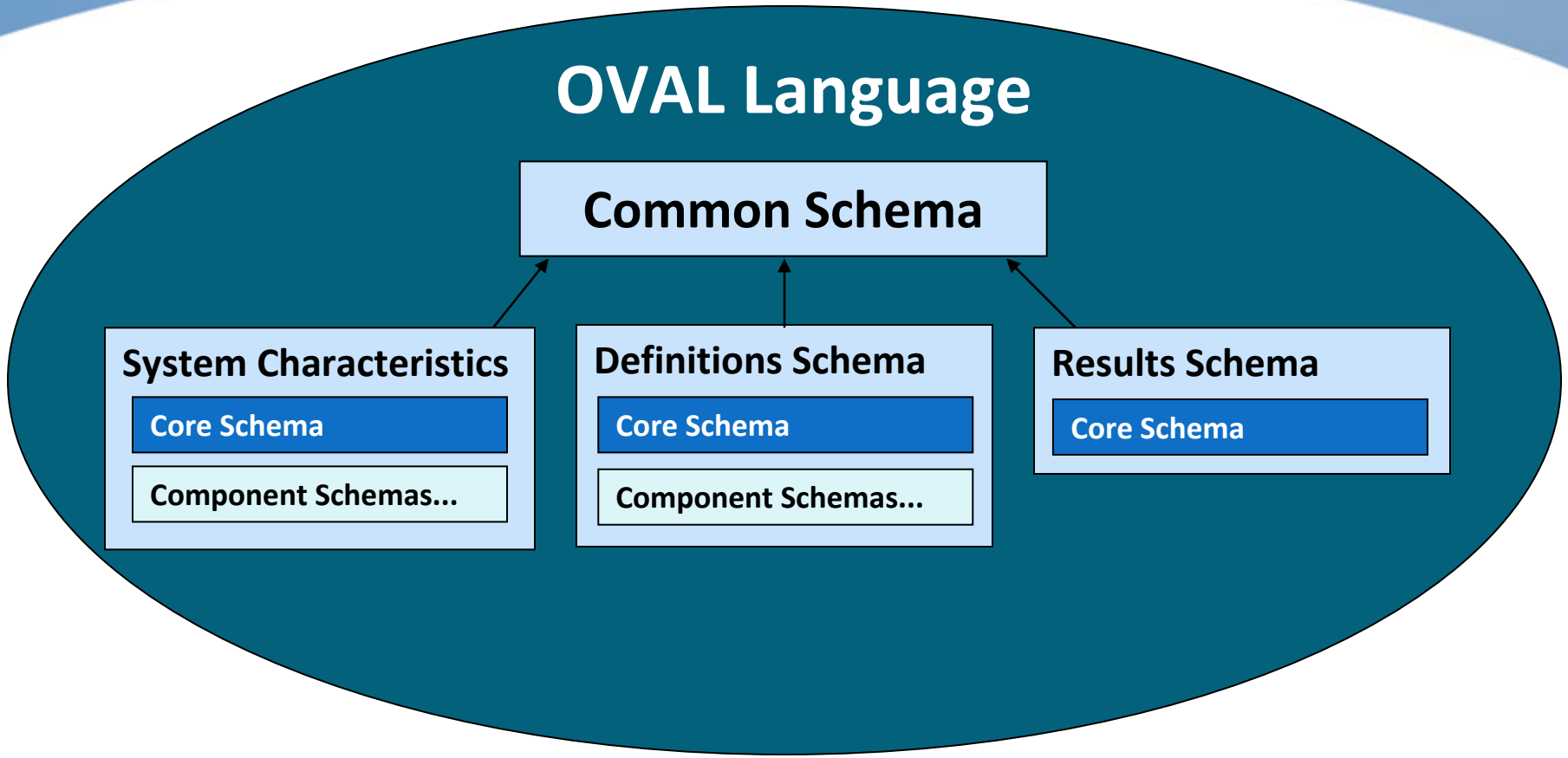
OVAL System Characteristics Schema

- Encoding of the details of a system (database of system info)

OVAL Results Schema

- Encoding of the detailed results of an analysis

Core Schemas Relationships

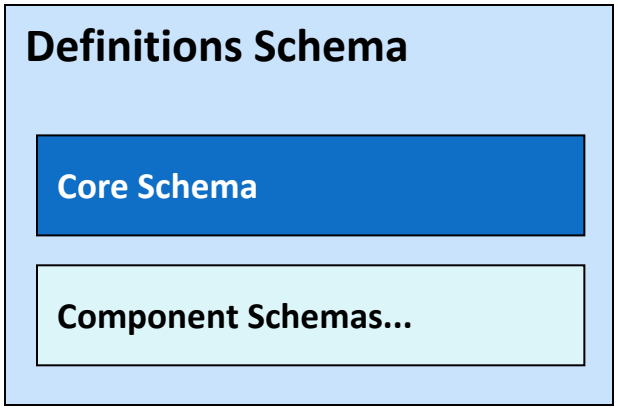


Component Schemas

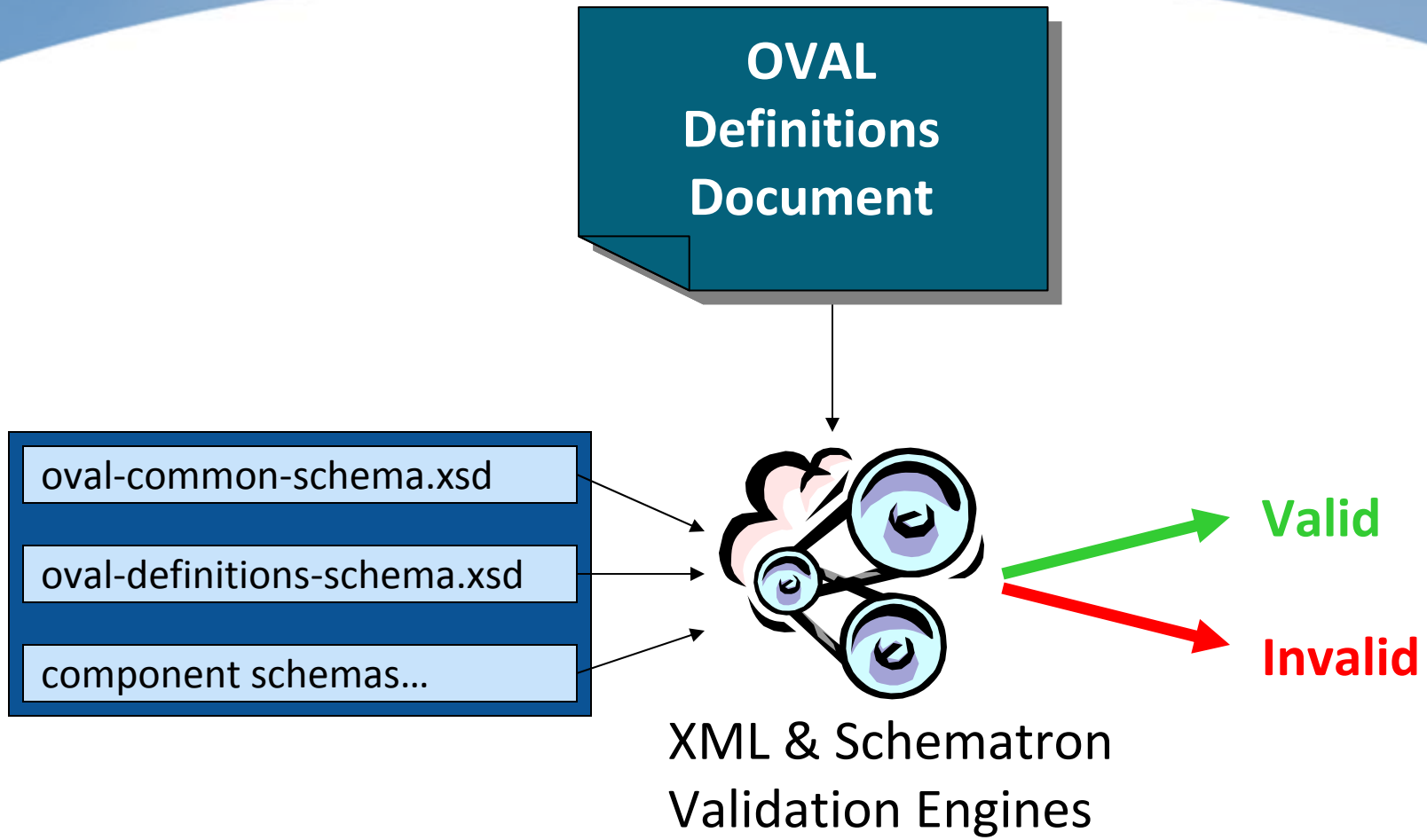
- Apple MacOS
- Cisco IOS
- Micosoft Windows
- Red Hat Enterprise Linux
- Sun Solaris

- Linux
- UNIX

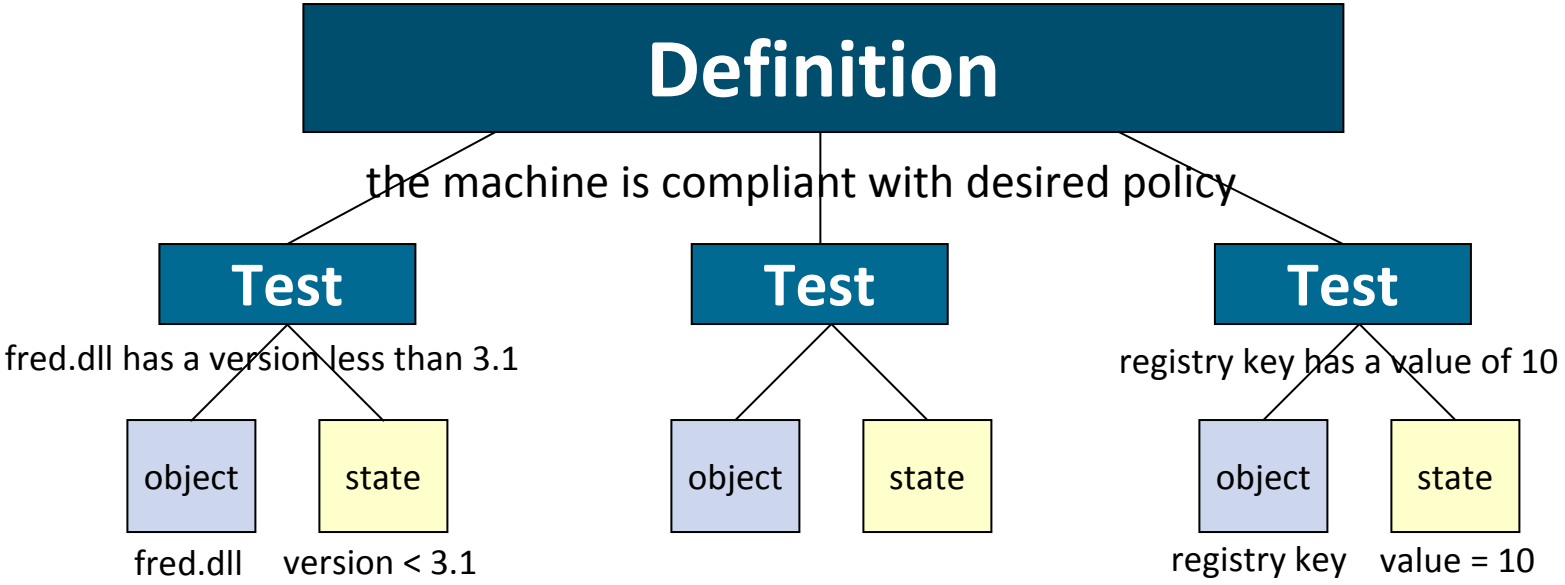
- Microsoft Sharepoint
- Vmware ESX



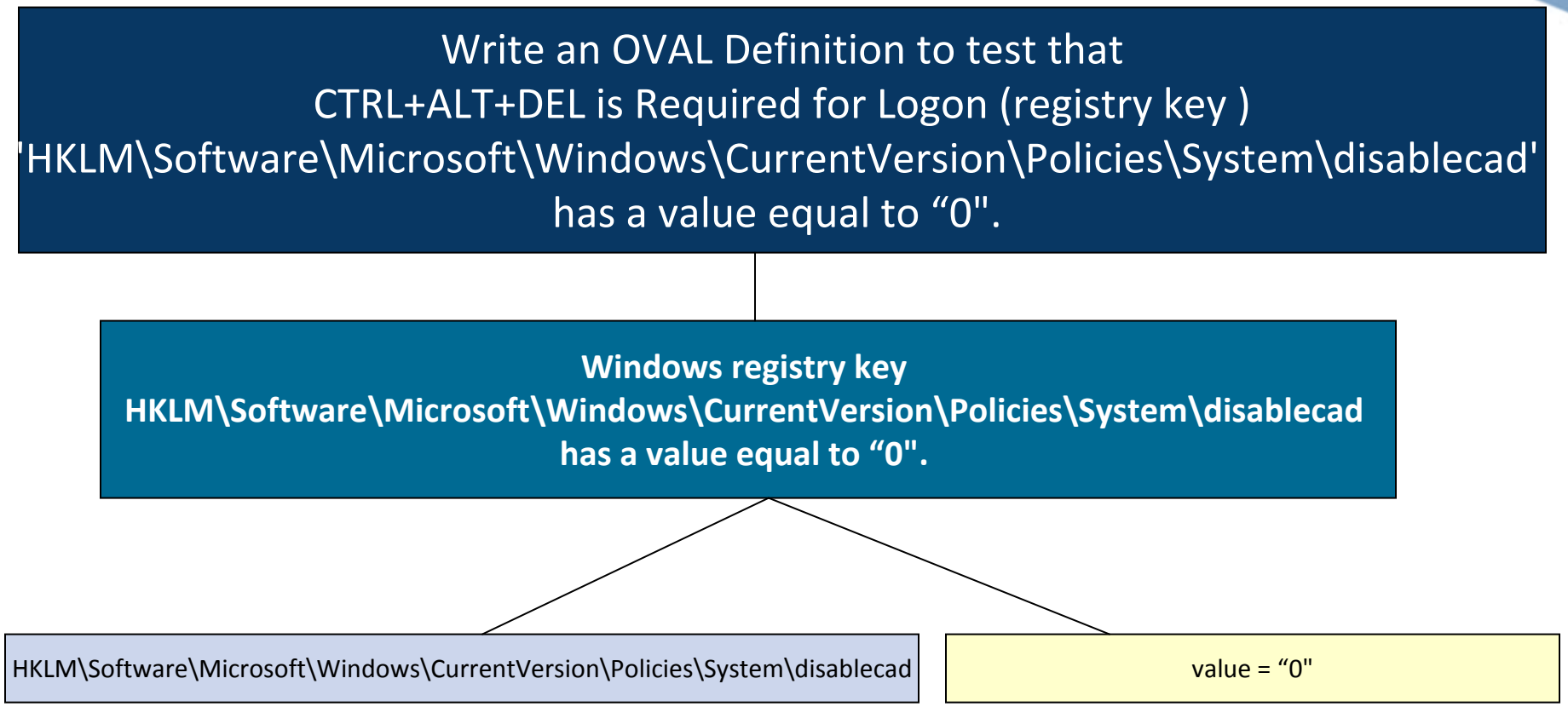
OVAL Document Validation Process



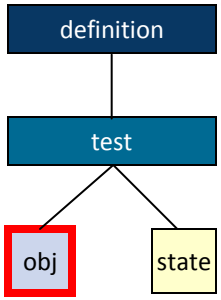
Structure of an OVAL Definition



CTRL+ALT+DEL - OVAL Definition

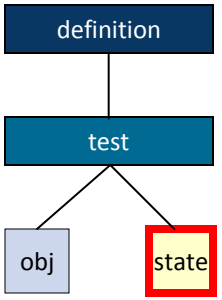


CTRL+ALT+DEL - Registry Object



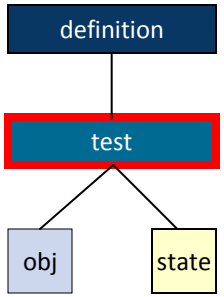

```
<registry_object id="oval:com.example:obj:1">
   <hive>HKEY_LOCAL_MACHINE</hive>
   <key>Software\Microsoft\Windows\CurrentVersion\Policies\System</key>
   <name>disablecad</name>
</registry_object>
```

CTRL+ALT+DEL - Registry State



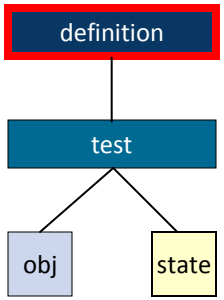

```
<registry_state id="oval:com.example:ste:1">
  <value datatype="int" operation="equals">0</value>
</registry_state>
```

CTRL+ALT+DEL - Registry Test




```
<registry_test id="oval:com.example:tst:1" check="all">
    <object object_ref="oval:com.example:obj:1"/>
    <state state_ref="oval:com.example:ste:1"/>
</registry_test>
```

CTRL+ALT+DEL - OVAL Definition



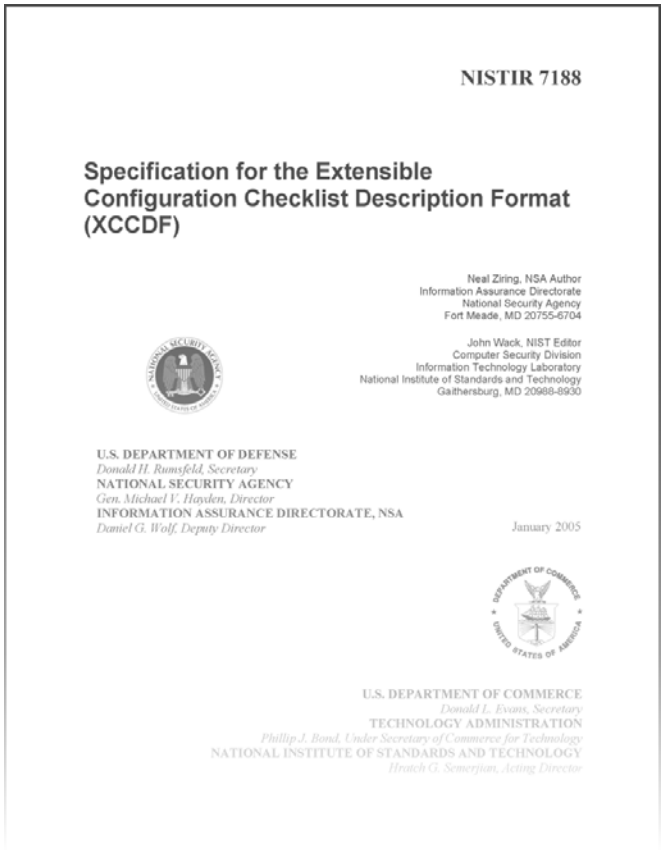

```
<definition id="oval:com.example:def:1">
  <metadata>
    <title>CTRL+ALT+DEL Required for Logon</title>
    <description>
        This definition is used to introduce the OVAL Language to
        individuals interested in writing OVAL Content.
    </description>
  </metadata>
  <criteria>
    <criterion test_ref="oval:com.example:tst:1"
               comment="The registry key is set to require CTRL+ALT+DEL for Logon"/>
  </criteria>
</definition>
```

Hello World - Full XML

```
<oval_definitions …>
 <generator>…</generator>
 <definitions>
  <definition id="oval:org.mitre.oval.tutorial:def:1" version="1" class="miscellaneous">
   <metadata>
    <title>CTRL+ALT+DEL Required for Logon</title>
    <affected family="windows"/>
    <description>This definition is used to introduce the OVAL Language.</description>
   </metadata>
   <criteria>
    <criterion test_ref="oval:org.mitre.oval.tutorial:tst:1 comment="The registry key is set to require CTRL+ALT+DEL for Logon"/>
   </criteria>
  </definition>
 </definitions>
 <tests>
  <registry_test id="oval:org.mitre.oval.tutorial:tst:1" version="1" check=“all" comment="The registry key is set to require CTRL+ALT+DEL for Logon" xmlns="http://oval.mitre.org/XMLSchema/oval-
        definitions-5#windows">
   <object object_ref="oval:org.mitre.oval.tutorial:obj:1"/>
   <state state_ref="oval:org.mitre.oval.tutorial:ste:1"/>
  </registry_test>
 </tests>
 <objects>
  <registry_object id="oval:org.mitre.oval.tutorial:obj:1" version="1" xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5#windows">
   <hive>HKEY_LOCAL_MACHINE</hive>
   <key>Software\Microsoft\Windows\CurrentVersion\Policies\System</key>
   <name>disablecad </name>
  </registry_object>
 </objects>
 <states>
  <registry_state id="oval:org.mitre.oval.tutorial:ste:1" version="1" xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5#windows">
   <value datatype=“int“ operation="equals">0</value>
  </registry_state>
 </states>
</oval_definitions>
```

What is XCCDF

- The Extensible Configuration Checklist Description Format
- An XML specification for expressing security benchmarks and recording assessment results.
- Designed for three purposes:
  - driving system security checking tools
  - generating human-readable documents and reports
  - scoring and tracking compliance



NISTIR 7188

Specification for the Extensible Configuration Checklist Description Format (XCCDF)

Neal Ziring, NSA Author
Information Assurance Directorate
National Security Agency
Fort Meade, MD 20755-6704

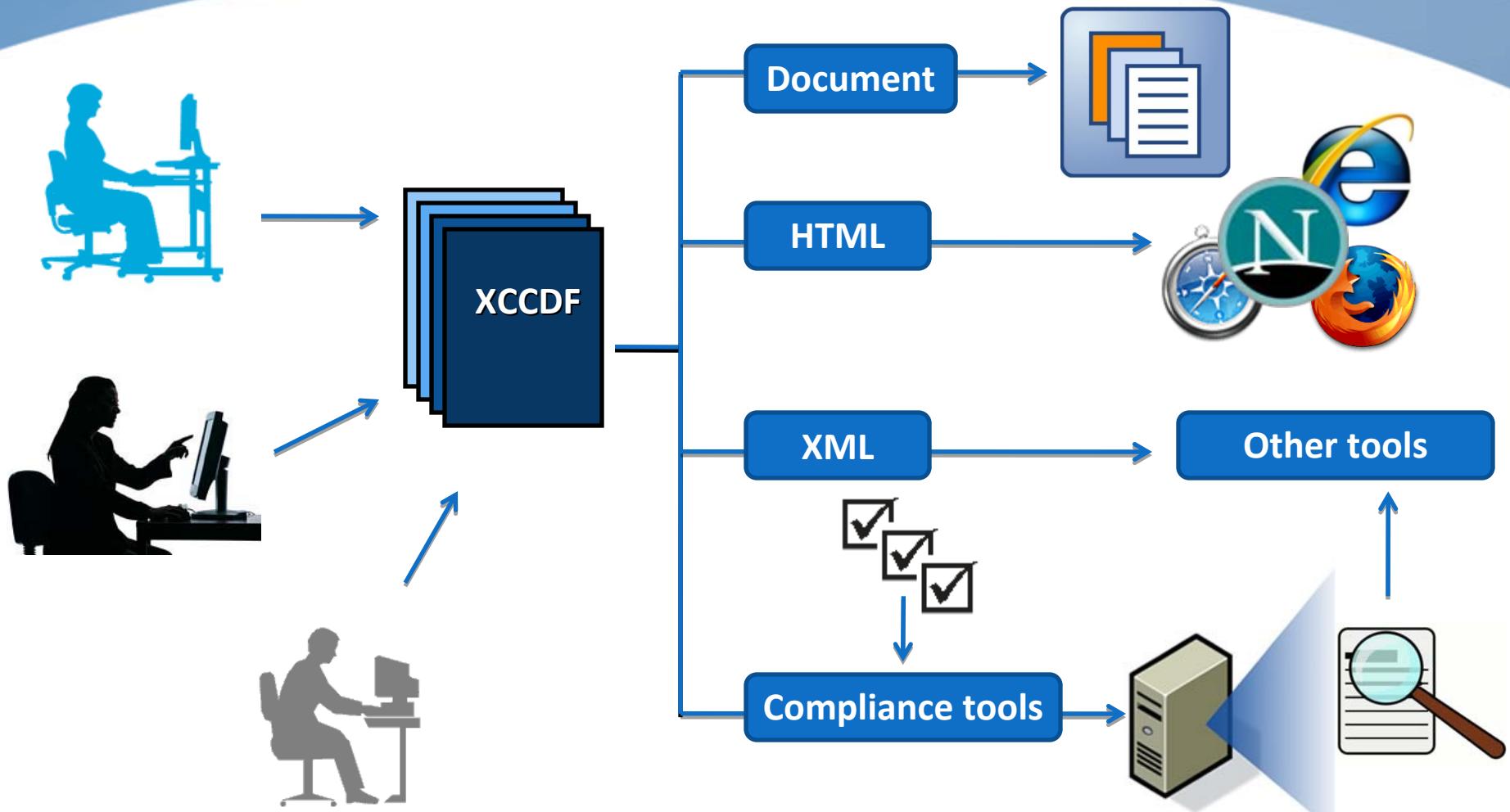
John Wack, NIST Editor
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20988-8930

U.S. DEPARTMENT OF DEFENSE
Donald H. Rumsfeld, Secretary
NATIONAL SECURITY AGENCY
Gen. Michael V. Hayden, Director
INFORMATION ASSURANCE DIRECTORATE, NSA
Daniel G. Wolf, Deputy Director

January 2005

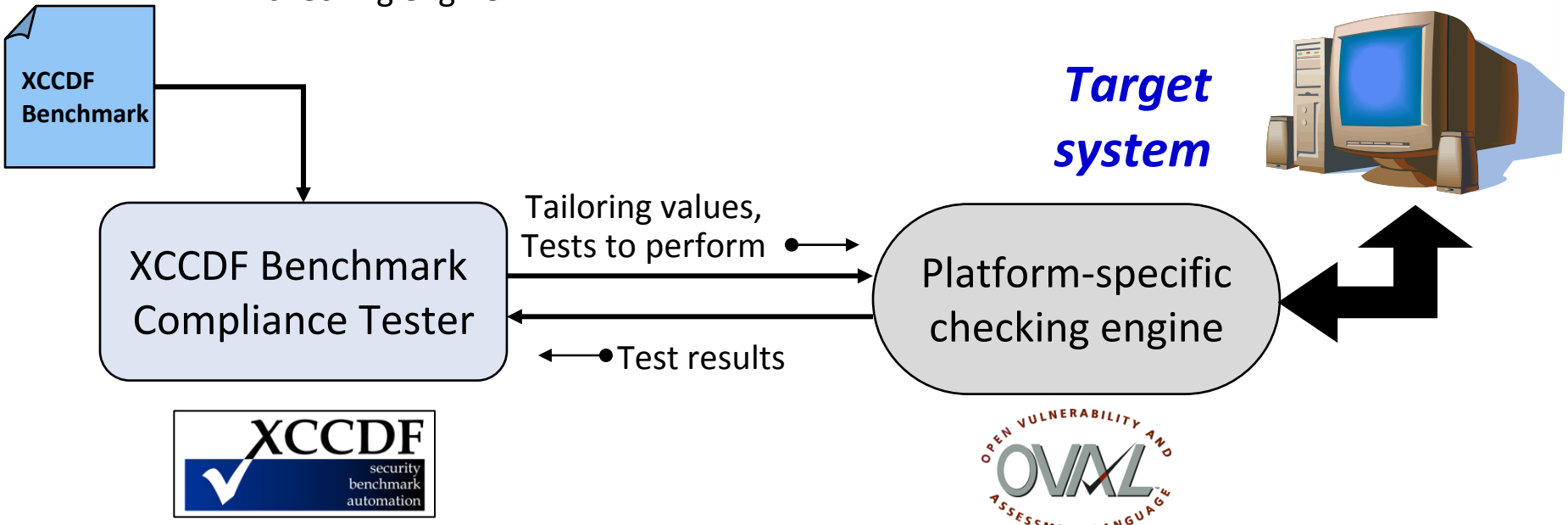
U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary
TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Hratch G. Semerjian, Acting Director

XCCDF Use Cases



XCCDF and Checking Engines

- XCCDF does ***not*** specify platform-specific system rule checking logic.
  - The `Rule/check` element contains information for driving a platform-specific checking engine.



XCCDF and OVAL Interaction

Guidance Structure and Customization

Support guidance tailoring and customization

Collect, structure, and organize guidance

Score and track general compliance

End-System Assessment

Define tests to check compliance

Define system-specific tests of system state

Characterize low-level system state

MITRE

84

XCCDF and OVAL Interaction

Guidance Structure and Customization

End-System Assessment

Support guidance tailoring and customization

Collect, structu

Score and track general compliance

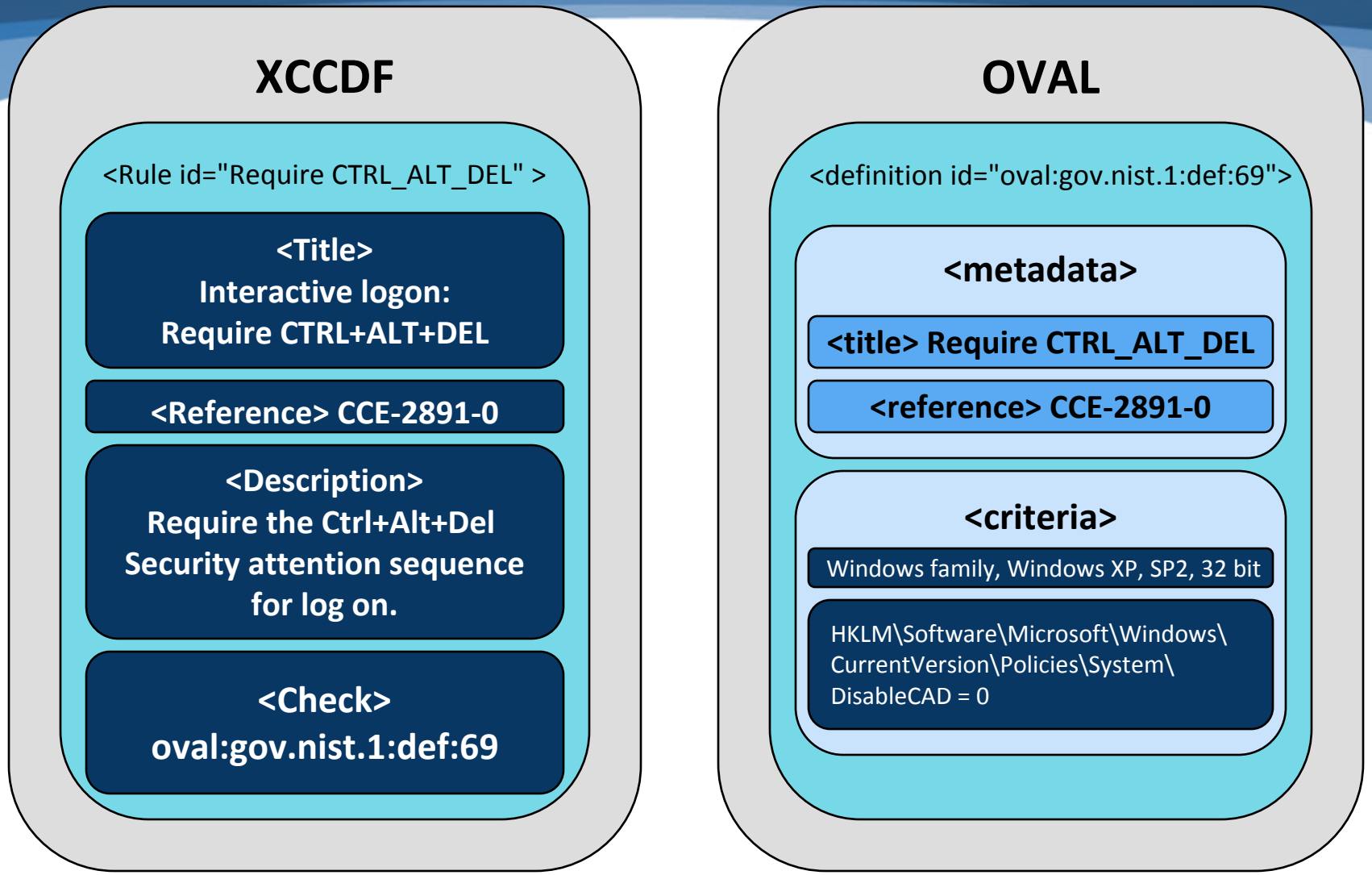
Define tests to che

Define system-spec ate

Characterize low-le

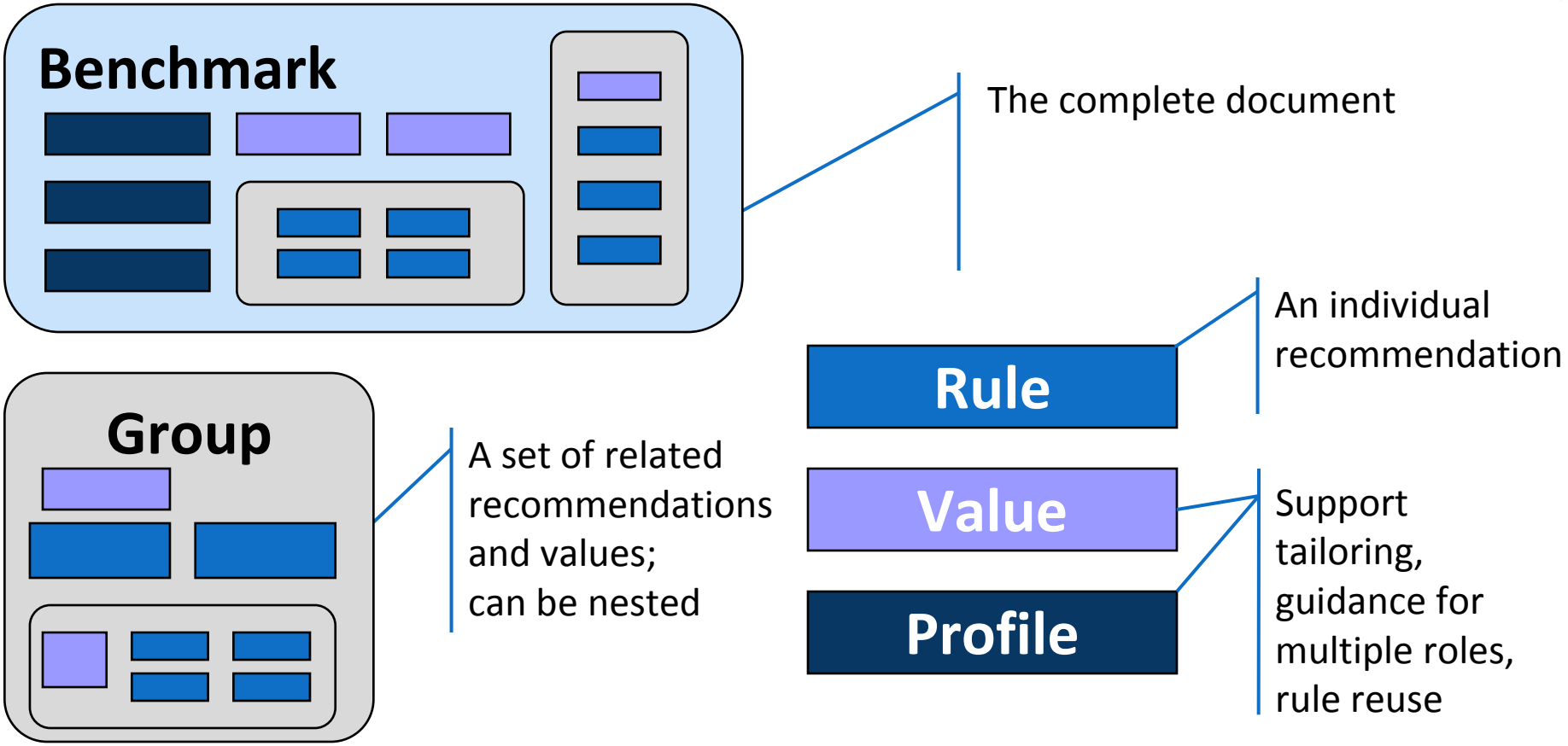
MITRE

XCCDF & OVAL Illustrated



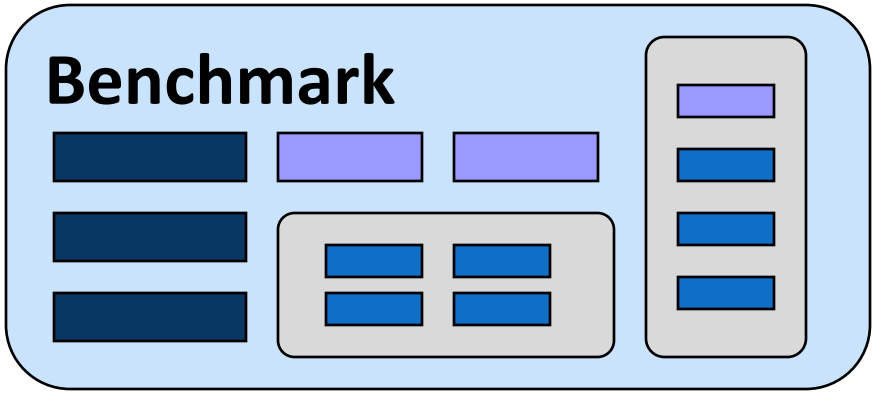
XCCDF Data Model

XCCDF defines the following key object types:



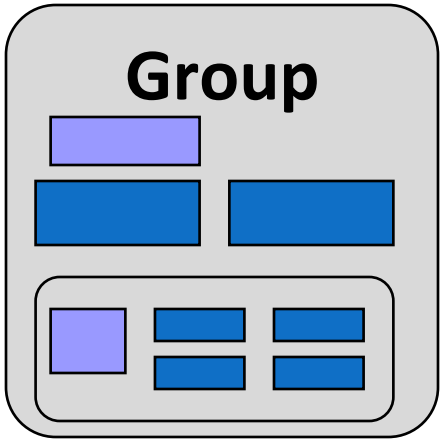
XCCDF Benchmark

```
<Benchmark id="Windows-XP">
   <title>Guidance for Securing Microsoft Windows XP</title>
   <platform idref="cpe:/o:microsoft:windows_xp"/>
   <Profile id="XP-Pro">...</Profile>
   <Group id="Chapter1">
      <Group id="PasswordPolicy">
         <Value>
         <Rule>
      </Group>
      <Group id="AuditPolicy">
          <Rule>
      </Group>
   </Group>
   <Group id="Chapter2">
   </Group>
</Benchmark>
```



XCCDF Group

```
<Group id="account_policies_group">
    <Group id="password_policies">
       <title>Password Policies</title>
       <description>In addition to educating users regarding the
       selection and use of good passwords, it is also important
       to set password parameters so that passwords are
       sufficiently strong...</description>
       <value>...</value>
       <rule>...</rule>
       <rule>...</rule>
    </Group>
</Group>
<Group id="file_permissions_group">
   ...
</Group>
```



XCCDF Rule

```
<Rule id="maximum_password_age" >
    <title>Maximum Password Age</title>
    <description>Set the “Maximum password age” password parameter to 90
   days.</description>
    <reference href="http://cce.mitre.org">CCE-2920-7</reference>
    <rationale>The “Maximum password age” password parameter is set to
         force users to change passwords at regular, defined, intervals…
     </rationale>
    <fixtext>1 - Launch the Local Security Policy editor: Start ->
         All Programs -> Administrative Tools -> Local Security Policy…
     </fixtext>
    <check system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
      <check-export value-id="maximum_password_age_var"
                    export-name="oval:gov.nist.fdcc.xp:var:90"/>
      <check-content-ref href="BDC-XP-oval.xml"
                         name="oval:gov.nist.fdcc.xp:def:17"/>
    </check>
</Rule>
```

Rule

XCCDF Profile

```
<Profile id="federal_desktop_core_configuration">
    <title>Federal Desktop Core Configuration</title>
    <description>This profile represents guidance outlined in
   Federal Desktop Core Configuration settings for Desktop
   systems.</description>
    <!--Password Policy Settings-->
    <select idref="maximum_password_age" selected="true"/>
    <select idref="minimum_password_length" selected="true"/>
    <refine-value idref="maximum_password_age_var"
        selector="5184000_seconds"/>
    <refine-value idref="minimum_password_length_var"
        selector="12_characters"/>
</Profile>
```

Profile

Summary

Standard languages allow for automated exchange of information between different sources.

- saves time
- reduces error
- interoperability
- greater visibility into what is being assessed

Scoring Systems - CVSS

CVSS

- Common Vulnerability Scoring System
  - open and universally standard severity ratings of software vulnerabilities
    - possible extension into configuration issues
  - help organizations appropriately prioritize

  - base scores - represent the innate characteristics of each vulnerability
  - temporal scores - change over time due to external events
  - environmental scores - customized to reflect the impact on your organization

SCAP Validation

SCAP Validation

- Verify that a tool performs an SCAP evaluation correctly
- Run by NIST
  - accreditation of 3rd party labs under NVLAP to perform actual testing
- 28 products currently validated (as of 10/16/2009)

IR-7511 : SCAP Validation Program Test Requirements

- **Purpose and Scope**: IR-7511 Rev 1 describes the requirements that must be met by products to achieve SCAP Validation. Validation is awarded based on a defined set of SCAP capabilities and/or individual SCAP components by independent laboratories that have been accredited for SCAP testing by the NIST National Voluntary Laboratory Accreditation Program.

- **Audience**: Laboratories that are accredited to do SCAP product testing for the program, vendors that are interested in receiving SCAP validation for their products, and government agencies and integrators seeking to deploy SCAP tools in their environments.

Federal Desktop Core Configuration

FDCC

- a baseline
  - set of configuration settings
- OMB mandated
  - March 2007
  - compliant by February 2008
- all Federal agencies
  - general purpose systems
    - desktops and laptops
    - servers, embedded systems are out of scope

What FDCC is Not

- FDCC is not a set of XML documents
- FDCC is not a tool
- FDCC is not a mandate

Platforms

- currently only defined for
  - Windows XP
  - Windows Vista
- includes IE7 and Windows Firewall
- examples
  - password policy
  - user rights
  - logging

Questions?