Security Content Automation Introduction to Day 2

J Todd Wittbold

The MITRE Corporation

IA Content vs IA Tools

- IA Content
  - Knowledge about vulnerabilities, threats, misconfigurations, best practices, etc
  - STIGS, Benchmarks, IAVA, US-Cert alerts
- IA Tools
  - Vulnerability scanners, IDS, Patch management systems, AV products, configuration management systems

IA Content vs IA Tools

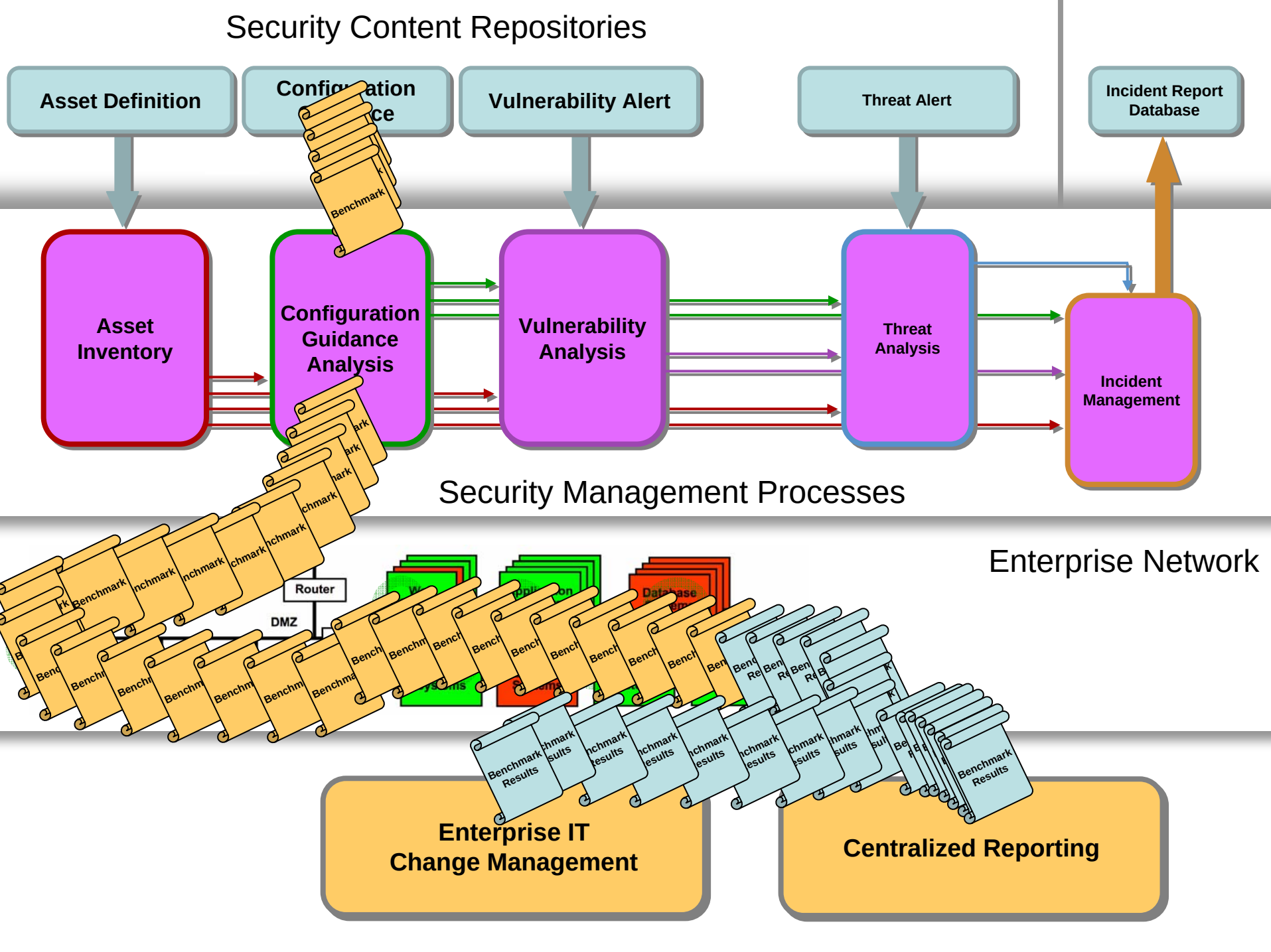
- Today:
  - Each IA tool vendor maintains large repositories of proprietary IA content
  - Naming conventions and testing semantics are specific to the product
  - Analysis results and reporting formats are specific to the product
- Tomorrow:
  - Standardized specifications of the most significant IA content types (e.g. NIST XP Config Guidance)
  - Consistent naming, testing, results reporting

Benefits of Decoupling IA Content from IA Tools

- Consistency, transparency, and concreteness in the specification and measurement of IA requirements
- Consistency in the communication of IA information between tool categories (e.g. vuln assessment to patch management, asset inventory to vuln assessment)
- Organizational subcomponents can make autonomous tool investments and still achieve global integrated reporting

Benefits of Decoupling IA Content from IA Tools (2)

- Policy writers have concrete foundations for expressing requirements for technical controls
- IA tool vendors can import (vs create) government supplied IA content
- Software vendors and government agencies have improved technical collaboration on secure configuration guidance for vendor products


Security Content Repositories
Asset Definition
Configuration Guidance
Vulnerability Alert
Threat Alert
Incident Report Database
Benchmark
Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
Threat Analysis
Incident Management
Security Management Processes
Enterprise Network
Router
DMZ
Database
Benchmark Results
Enterprise IT Change Management
Centralized Reporting

How To Decouple IA Content from IA Tools

- Identify the basic entities that IA Content needs to reference
  - Vulnerabilities, configuration settings, etc
- Provide a machine-readable language for making assertions about the basic IA entities (XCCDF/OVAL)
- Express IA requirements as documents in the XCCDF/OVAL language

The Pieces

- Enumerations (CVE, CCE, UPPN)
  - Catalog the fundamental entities in IA business
    - Software packages, vulnerabilities, misconfigurations
- Languages (XCCDF, OVAL)
  - Support the creation of machine-readable assertions about those entities
- Content (STIGS, Benchmarks, Checklists)
  - Packages of assertions supporting a specific application
    - Vuln assessment, config guidance, asset inventory
- Tools
  - Interpret IA content in context of enterprise network

Enumerated Entities

- **Vulnerabilities**
  - **CVE-2006-4838**
  - 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
- **Configuration Settings**
  - **CCE-W2K-178**

    Definition: The "restrict guest access to application log" policy should be set correctly.

    Technical Mechanism:

    (1)HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\RestrictGuestAccess

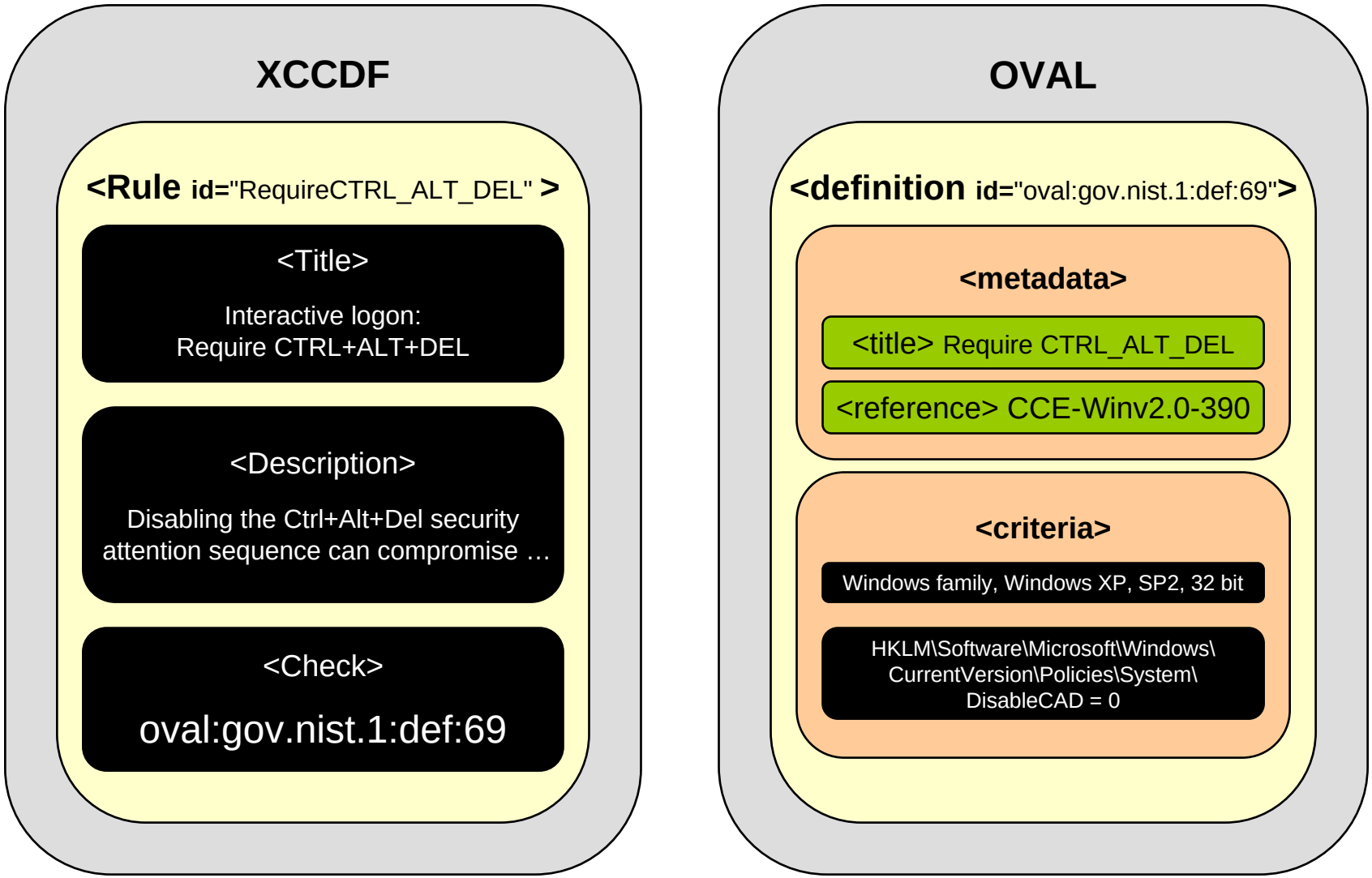
    (2) defined by Group Policy

    Parameters enabled/disabled
- **Software Packages**
  - `uppn-291` `uppn//microsoft.com:windows:xp:professional:sp2`

Languages

- OVAL
  - XML language framework for assertions about software configuration state
- XCCDF
  - XML language framework for packaging and documenting checklist requirements and results
- Checklist item ~ OVAL assertion about s/w configuration parameter(s)

XCCDF-OVAL Connection



Vulnerability Content Example

- OVAL vulnerability definition binds a CVE id to an OVAL test

OVAL-ID: oval:org.mitre.oval:def:399 **Date:** 2006-09-08

Status:	DRAFT
Class:	vulnerability
Ref-ID:	CVE-2006-3590
Schema Version:	5
Platform(s):	Microsoft Windows 2000

Description:

Unspecified vulnerability in mso.dll, as used by Microsoft PowerPoint 2000 through 2003, allows remote user-complicit attackers to execute arbitrary commands via a crafted PPT file, which causes a "memory corruption error," and exploited by Trojan.PPDropper.B. NOTE: As of 20060714, due to the vagueness of the initial disclosure, it is uncertain whether this is related to CVE-2006-1540 or CVE-2006-3493. Other PowerPoint issues were disclosed in the same time frame, including CVE-2006-3655, CVE-2006-3656, and CVE-2006-3660.

Definition Synopsis:

- PowerPoint 2000
  - AND the version of Mso9.dll is less than 9.0.0.8948
- OR PowerPoint 2002
  - AND the version of Mso.dll is less than 10.0.6811.0
- OR PowerPoint 2003
  - AND the version of Mso.dll is less than 11.0.8036.0

Stakeholder Questions (1)

- Government IA Policy Authors
  - How can portions of my IA Policy benefit from NIST SCAP Content?
- Government IA Teams
  - How can I use NIST SCAP enabled products in my enterprise?
- IA Product vendors
  - How can I use NIST SCAP content into my product?
  - How can my product better interoperate with other SCAP enabled products?

Stakeholder Questions (2)

- Software Product Publishers
  - What to propose as baseline security configuration settings for each product?
  - How to precisely specify which of my products are affected by a vulnerability?
  - How to precisely determine when my product is installed and which version of that product is installed?