



**ManTech**  
International Corporation®

*Leading the Convergence of National Security and Technology™*

# The Promise and Reality of SCAP Implementation

August 2011

# Objectives of the Presentation

- Influence SCAP content development practices, including the revision of current specifications and the development of new specifications
- Provide future implementers of SCAP tools with some insights to the challenges inherent in SCAP adoption and activities to make those challenges more manageable.



- Outline the objectives of SCAP Implementation
- Discuss the experiences of some of those responsible for implementing SCAP tools
- Identify some of the challenges to SCAP adoption and implementation
- Enumerate ways to overcome those challenges and provide a demonstration of those methods
- Discuss prototype improvements that should close the gap between the promise and reality of SCAP implementation



# Objectives of SCAP Implementation

---

- Achieve continuous monitoring
- Integrate vulnerability monitoring and configuration audits into the SOC services
- Reduce the cost of C&A activities



# Implementation Experiences

- SCAP capabilities were not a consideration in the selection of a scanning tools →
  - Multiple tools to perform vulnerability and configuration audits for different target platforms
- Tool was purchased to conduct vulnerability assessments →
  - Although the tool is capable, they are not conducting configuration audit scans with the tool
  - Did not receive any additional training on how to leverage the configuration audit capabilities of the tool
- Selection and Implementation Team included SOC and IA personnel only →
  - Users are limited to SOC and IA personnel, system owners and administrators are not involved in the evaluation of vulnerability finding
  - Limited impact on security service related business processes



- Multiple Tools
  - Select an SCAP tool that provides the broadest selection of target platforms for both vulnerability and configuration audit assessments
  - Where SCAP content is not available for a given platform, use the availability of vendor proprietary configuration assessments as a selection criteria,
  - Determine if the data store for scanning tool can incorporate data from other scanning tools



- Limited Implementation
  - Develop CONOPS for tool implementation that includes both vulnerability and configuration audit management
  - Leverage existing / publically available SCAP configuration content
  - Leverage vendor provided proprietary configuration audit assessment files
  - Ensure that you understand how the vendor product distinguishes between vulnerability scans and configuration audit assessments



- Limited User Base and Limited Impact on Security Service Related Business Processes
  - Integrated Planning Team, consisting of all IT system stakeholders, should participate in the vetting (if not the development) of the CONOPS for tool implementation
  - Ensure that the CONOPS addresses the major objectives of SCAP implementation
  - If CONOPS includes the development of organization specific content, then provision has to be made for access to
    - SMEs for each of the target platforms
    - Personnel familiar with each of the SCAP component specifications
  - Develop a training plan for the users based on their responsibilities within the security service business processes

# What flags SCAP implementation?

- Availability of SCAP compliant checklists,
- SCAP compliance does not imply interoperability,
- The non-technical organizational readiness to leverage a SCAP compliant tool, and
- Gaps in SCAP components.



# Problems with Available Checklists

- Not many SCAP checklists
- Even fewer really good checklists- Many 70% Solutions
- Hard to determining which checklists to use
- Checklist Management



# Problems with Available Checklists: Number of SCAP Checklists

- Simple View
  - NVD Website: 178 checklists available on the NVD website- 91 prose only, 57 non-SCAP automated content, 24 should work in SCAP validated tools, only nine are classified as “will work with SCAP validated tool,”
  - NIST/USGCB Content has four Windows checklists and one Red Hat Linux desktop checklist



# Problems with Available Checklists

## Number of SCAP Checklists

---

- More Complex View
  - Lots of overlap of the automated content
  - Duplication of effort, less evident improvements of existing content
  - No PCI DSS or HIPPA Security Rule specific checklists
- Many organizations have configuration guides, but they are not automating them. Some attempt to use what is available, in any form.

# Problems with Available Checklists

## Not Many Really Good Checklists

- Incomplete information and obscure references
  - The rule elements of the Windows 2003 Member Server Security Technical Implementation Guide lack CCE references and the links to the DoD 8500.2 are buried within the description elements of the rules, but as data (e.g. <code>&lt;IAControls&gt;ECSC-1&lt;/IAControls&gt;</code>)
  - As of 7/14/2011 none of the checklist available on the usgcb.nist.gov site contained references to 800-53 controls.
    - However, several of the FDCC checklist contain references to ISO/IEC 17799, NIST 800-26, GAO FISCAM, DoD 8500.2, and DCID 6/3 high-level security requirements guides (e.g. fdcc-ie-7,
  - Benchmarks omitted from checklists when automated checks not available, without identifying gaps
- Need a process of evaluating the relationship between SCAP content, the prose checklist associated with the SCAP content, and any high-level security requirements



# Problems with Available Checklists

## Determining Which Checklists to Use and How

- Checklists use varied methods to manage the evaluation of HLSR controls:
  - Which HLSR controls are addressed and how they are managed
  - How the evaluation criteria are expressed
- Each of these has advantages and disadvantages, but more importantly, it is necessary to understand the specific implications on the checklist structures on the SCAP tool that is used to run the checklist.

# Problems with Available Checklists

## Determining Which Checklists to Use and How

- Checklists use varied methods to manage the evaluation of HLSR controls:
  - Which HLSR controls are addressed and how they are managed
    - Some checklist contain only those rules for specific operational environment
    - Some checklist contain rules for all of the HLSR controls for each of the operational environments
      - No profiles or groups are used to manage the evaluation of the controls
      - A single profile is used to manage the evaluation of the controls
      - Separate profiles for each operational environment
        - » With only those HLSR controls that for operational environment listed and enabled
        - » With all the HLSR controls listed, but only those relevant to an operational environment enabled
  - How the evaluation criteria are expressed
    - Some rely on the evaluation criteria to be statically defined in the OVAL document
    - Some benchmark rules will pass the evaluation criteria to the OVAL definition as a parameter
      - Defined within the benchmark rule
      - Defined as a variable within a profile linked to the benchmark rule



# Problems with Available Checklists

## Determining Which Checklists to Use and How

- Checklists use varied methods to manage the evaluation of HLSR controls:
  - Which HLSR controls are addressed and how they are managed
  - How the evaluation criteria are expressed
- Each of these has advantages and disadvantages, but more importantly, it is necessary to understand the specific implications on the checklist structures on the SCAP tool that is used to run the checklist.
  - How does the SCAP tool take the benchmark inputs to create the necessary assessment tools?
  - What capabilities does the SCAP tool provide to update the evaluation of the targeted systems?
  - How are the profile and group information reflected in the output provided by the SCAP tool? What are the differences between the tools



# Problems with Available Checklists

## Determining Which Checklists to Use and How

- If interested in FISMA compliance, then choose a USGCB checklist, if one exist
- If interested in DoD 8500.2 compliance, then choose a checklist based off of the DISA STIGs.
- What if you are interested in both?
- What if you are not interested in either, but in PCI DSS or HIPPA Security Rules or an organization specific security requirements?



# Problems with Available Checklists

- Not many SCAP checklists
- Even fewer really good checklists- Many 70% Solutions
- Hard to determining which checklists to use



# Use and/or Improve Existing Content

- Use Vendor Content
  - Numbers
  - Vendors will ensure that the checklist they provide will run with their tool
  - Push you vendors to provide checklists in accordance with ....
  - Users need to have clear understanding to which operational environment the vendor provided checklist is mapped and the impact of changing the parameters of that checklists.
  - Disadvantages
    - Limited or no SCAP compliance, proprietary checks, but more important proprietary output missing critical SCAP component information
    - Tied to a vendor proprietary solution
    - Tied to the vendors production schedule



# Use and/or Improve Existing Content

- Augment, Revise, or Develop Checklists
  - Tools used to augment, revise, or develop checklists
    - Benchmark Editors
    - XML editors
    - SCAP Tool vendor capabilities to augment the checklists
  - What content?
    - Organization specific information
    - Missing checks- even those that can not be automated
    - Missing reference information
- Disadvantage of content management

# Use and/or Improve Existing Content

- Develop CONOPS that takes into account the availability of checklists and specifically notes which checklist will be used for various configuration audits and the necessary business processes to achieve the attainable objectives.
- Update the relevant configuration guides to reflect how the configuration audits will be conducted, which checks will be automated and those that will have to be manually assessed.
- Be attentive to the impact of changes to assessment tools and adoption of new checklists on the historical data



# Problems with Available Checklists

## Configuration Management of Checklists

- Ability to modify checklists
  - With changes to the SCAP components
    - Support for new types of checks, deprecation of component features
  - With changes to platforms
    - Support for new features, such as PowerShell
  - With changes to high-level security requirements
- What happens when you customize a vendor provided checklist and the vendor updates the benchmarks or assessments?
- This challenge is exacerbated by the tight coupling and specificity of all of the components

- Ability to move information from one SCAP compliant tool to another
  - Validation of use cases for information exchange between SCAP compliant scanners
- SCAP compliant
  - Which SCAP- version 1.0 or 1.1
  - Which platforms- recognize the limitations of various SCAP tools, only work with specific CPEs or only SCAP validated for specific CPEs



- Organizations that have implemented SCAP validated tools or other vulnerability and configuration audit tools need to consider the following:
  - Tool training to leverage all of the capabilities of their purchased products
  - Revision of organizational structure to leverage the tool
    - Addition of personnel with competencies with the SCAP components
    - Reduction of the number of personnel to manually perform configuration audits
    - Augmenting network operations support staff/system administrators to handle the increased awareness of vulnerabilities and system mis-configurations
  - Platform SMEs with the necessary knowledge to evaluate assessment definitions to ensure that they are valid with respect to the to desired benchmark evaluation.
  - Changes to vulnerability and configuration compliance business processes
  - Deciding if existing checklists, and which ones, are necessary and sufficient to ensure secure configuration of a platform or if organization specific checklists have to be developed.

# Gaps in SCAP Components

- What does SCAP specifications promise and what are the gaps in the specifications that undermine the achievement of the promises
  - Ability to create traceability from system high level security requirements to benchmark rules
  - Ability to reconcile multiple high level requirements for a particular system
- Options
  - Work within the languages as they exist today
  - Exploit the extensibility of the language, but lose interoperability
    - The best way to make improvements to the SCAP components is to extend the standards with a pilot/prototype and then return to the community with the proof of the benefits



- Using transforms to augment benchmark documents
  - USGCB benchmark documents were augmented with CCE information
    - From human readable documents /spreadsheets
    - Existing SCAP benchmarks that contained the same HLSR control IDs

# Closing the Gap

- Develop High Level Security Requirement Languages, which structure the operational environments of high level security documents and their security requirements
- Define the high-level security requirements applicable to a system and its components
- More granular assignment of HLSR controls to rules
- Develop complete benchmark and assessment documents
- Reduce duplication of effort and increase the reuse of benchmarks and assessment components



# High-Level Security Requirements Language

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="DCID 6-3-
working.xsl"?>
<!--<?xml-stylesheet type="text/xsl" href="DCID 6-
3v5.1.xsl"?-->
<dcid6-3reqs>
<levels_of_concern>
<level_of_concern pl="pl1" name="protection level 1">
<level_of_concern pl="pl2" name="protection level 2">
<level_of_concern integrity="basic" name="integrity -
basic">
<level_of_concern integrity="medium" name="integrity -
medium">
<level_of_concern availability="basic" name="availability -
basic">
<level_of_concern availability="medium"
name="availability - medium">
</levels_of_concern>
<secreq title="[Access1]" type="feature">
<family name="Access"/>
<pl pl1="required" pl2="required" pl3="required"
pl4="required" pl5="required"/>
</secreq>
<secreq title="[Access2]" type='feature'>
<family name="Access"/>
<pl pl1="not required" pl2="required" pl3="required"
pl4="required" pl5="required"/>
</secreq>
```

```
<secreq title="[Avail]" type="feature">
<family/>
<avail ab="required" am="required" ah="required"/>
</secreq>
<secreq title="[Backup1]" type="feature">
<family name="Backup"/>
<integrity ib="required" im="not required" ih="not
required"/>
<avail ab="required" am="not required" ah="not
required"/>
</secreq>
<secreq title="[Power1]" type="feature">
<family name="Power"></family>
<avail ab="not required" am="required" ah="required"/>
</secreq>
<secreq title="[Backup2]" type="feature">
<family name="Backup"/>
<integrity ib="not required" im="required" ih="required"/>
</secreq>
<secreq title="[CM1]" type="feature">
<family name="Configuration Management"/>
<integrity ib="required" im="required" ih="required"/>
</secreq>
```



# Extend the Asset Information Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<xal:AddressDetails
xmlns:core="http://scap.nist.gov/schema/reporting-
core/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:sch="http://purl.oclc.org/dsdl/schematron"
xmlns:ai="http://scap.nist.gov/schema/asset-
identification/1.1"
xmlns:xal="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xmlns:xnl="urn:oasis:names:tc:ciq:xsdschema:xNL:2.0"
xmlns:cpe-name="http://cpe.mitre.org/naming/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:e-ai="http://mantech.com/schema/extended-
asset-identification/1.0"
xmlns:dcid63="http://mantech.com/schema/high-level-
security/dcid6-3/1.0"
</xal:AddressDetails>
<it-asset id="Internal-DC">
  <dcid6-3:pl pl="p11"/>
  <dcid6-3:avail ab="true"/>
  <dcid6-3:intg ib="true"/>
  <nist800-53:minsec cl="low"/>
  <computing-device>

  <cpe>cpe:/o:microsoft:windows_server_2003::sp2</cpe
  >
  <hostname>internal-lab-dc</hostname>
</computing-device>
</it-asset>
```

```
<it-asset id="External-DC">
  <dcid6-3:pl pl="p12"/>
  <dcid6-3:avail am="true"/>
  <dcid6-3:intg im="true"/>
  <nist800-53:minsec cl="moderate"/>
  <computing-device>
  <cpe>cpe:/o:microsoft:windows_server_2003::sp2</cpe
  >
  <hostname>external-lab-dc</hostname>
</computing-device>
</it-asset>
<it-asset id="Internal-rhs">
  <dcid6-3:pl pl="p11"/>
  <dcid6-3:avail ab="true"/>
  <dcid6-3:intg ib="true"/>
  <nist800-53:minsec cl="low"/>
  <computing-device>

  <cpe>cpe:/o:redhat:enterprise_linux:5.4::server_x64</cp
  e>
  <hostname>Internal-lab-rhs</hostname>
</computing-device>
</it-asset>
```



# Providing Traceability

	DCID 6/3 Controls	Internal-DC and Internal-RHS PL1/Availability-Basic/Integrity-Basic	External-DC PL2/Availability-Med./Integrity-Med.
Confidentiality Protection Level Controls	[Access1]	Required	Required
	[Access2]	Not Required	Required
Availability Controls	[Avail]	Required	Required
	[Backup1]	Required	Not Required
	[Power1]	Not Required	Required
Integrity Controls	[Backup1]	Required	Not Required
	[Backup2]	Not Required	Required
	[CM1]	Required	Required

The creation of a high level security requirements language and the extension of the asset information specification enables us to define the operational environment for a system and its components and them to create a security requirements traceability matrix based on the requirements of that operational environment.

