Lessons Learned from a Decade of SCAP

Charles Schmidt

This presentation was developed using contract funds under Basic Contract No. W56KGU-18-D-0004.

The view, opinions, and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.
Overview

- **SCAP = the Security Content Automation Protocol**
  - First published in 2006 by NIST (SP800-126)
  - “a suite of specifications that standardize the format and nomenclature by which software flaw and security configuration information is communicated”
  - SCAP describes how a set of more targeted standards should work together

- **Replace tedious, manual enterprise audits with open automation**
  - Standardize automatable instructions for assessment and results

- **Status today – mixed result**
  - Saw significant adoption, but still often considered a USG niche product

- **What happened?**
Why do I get to talk about this?

- **The MITRE Corporation**
  - MITRE operates Federally Funded Research and Development Centers
  - Non-profit, operating in the public interest

- **MITRE did not create SCAP, but created several of the standards it uses**
  - MITRE has played a lead role in the SCAP community on behalf of the USG

- **Charles Schmidt – Principal Engineer at The MITRE Corp**
  - Lead editor/author of several specifications
  - Currently lead the effort to develop SCAP v2
# What is SCAP?

<table>
<thead>
<tr>
<th>Standard</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVAL</td>
<td>Language for machine-testable device state assertions</td>
</tr>
<tr>
<td>XCCDF</td>
<td>Language for machine and human readable benchmarks</td>
</tr>
<tr>
<td>OCIL</td>
<td>Language for user/operator assessment surveys</td>
</tr>
<tr>
<td>CVE</td>
<td>Names for computer vulnerabilities</td>
</tr>
<tr>
<td>CCE</td>
<td>Names for computer configuration elements</td>
</tr>
<tr>
<td>CPE</td>
<td>Scheme for software/hardware names</td>
</tr>
<tr>
<td>SWID tags</td>
<td>Structure for identifying software products</td>
</tr>
<tr>
<td>CVSS</td>
<td>Metric for scoring vulnerability severity</td>
</tr>
<tr>
<td>CCSS</td>
<td>Metric for scoring configuration control importance</td>
</tr>
<tr>
<td>SCAP Data Streams</td>
<td>Structure for bundling SCAP content for transmission</td>
</tr>
<tr>
<td>SCAP</td>
<td>How all of the above work together</td>
</tr>
</tbody>
</table>
SCAP History

1999 - CVE released – Standard name for vulnerabilities
2002 - OVAL released – Language to standardize when a vulnerability is present (among other things)
2004 - XCCDF released – Language to create security benchmarks by combining checks
2006 – SCAP 1.0 released – Standard for combining these and other standards to automate enterprise assessment
2007– OMB issues memo mandating use of SCAP validated tools for US agencies
2011 (Feb) – SCAP 1.1 released – Integrate more standards
2011 (Sept) – SCAP 1.2 released – Integrate more standards
2018 – SCAP 1.3 released – Integrate more standards
SCAP Current Status

- Depends on who you ask...

- OMB still mandates SCAP validated tools
- Some vendors (e.g., Red Hat) continue to release SCAP content/tools
- There are companies built around providing SCAP services

- But...

- Getting complaints about cost of content and vendor lock-in
- SCAP is commonly perceived as “US government niche standard”
- Internet search for SCAP and the top hits are from 2016
SCAP Challenges, Gaps, and Problems

- Many issues of varying severity – this talk focuses on four

1. **Data normalization ≠ tool interoperability**
   - SCAP focused on standardized languages and formats

2. **Challenging (i.e., expensive) content development and management**
   - Lots of big XML files

3. **SCAP remains focused on monolithic audit**
   - Assessments are automated and relatively fast, but still done quarterly/monthly

4. **Community remains scattered and isolated**
   - Resources are scattered and often out of date – disincentive to join
Data Normalization Challenge

- Agents on endpoints for data collection
- Servers to initiate and orchestrate enterprise collection
- Databases for assessment results
- Analytic and visualization tools

- All these roles use SCAP content, but SCAP never specified how content moves between
- Vendors vertically integrated their products
  - Their agents, their servers, their databases, their displays
  - You still got vendor lock-in
Content Development

- SCAP was supposed to be a democratizing force
  - Product vendors would build content for their software
  - Researchers would release checks with their findings
  - Enterprises could build their own ad-hoc checks
  - Content could be reused and extended as needed

- All of these happened after a fashion, but there was no “critical mass”
- Many groups needed their own custom extensions to content
- Content authoring required language knowledge + subject expertise
  - Usually had to outsource to get that combination
- Reuse is a pain – piecing together chunks from large XML files
Monolithic Audits

- SCAP grew out of the pain and inefficiency of manual audits
  - SCAP made audits easier and faster, but never broke with the audit paradigm

- Assessments continue to be events
  - Monthly/quarterly/weekly
  - After patch Tuesday or other vulnerability release

- This isn’t management – management needs to be real time
Adoption and Advertising

- The Standards Developer’s Fallacy

  GOOD IDEAS SPEAK FOR THEMSELVES. A GREAT DESIGN WILL RISE ABOVE THE OTHERS AND GET ADOPTED.

  - Corollary 1: Advertising is cheating and only necessary for inferior designs

- SCAP hosted events for security tool developers to provide feedback and help with revisions
  - Events to get users interested in the standard were largely neglected

- Today we don’t really know who the users are & new users don’t know where to look for help
What are we doing about this?

- New community effort (kicked off about a year ago)
- Recently held our second in-person workshop
- Build on the existing uses and capabilities of SCAP (languages, etc.)
  - Not losing any of our use cases
- Develop responses to the gaps and issues noted earlier
Standardized Interfaces

- Define standardized interfaces for key SCAP roles

- Vendors increase the power of their services by integrating with others
  - Add new collectors, new analytics, etc.

- Vertical integration is fine, but add hooks to plug-in other tools
Content Authoring Enhancements

- **Work on ways to treat content authoring more like coding**
  - Code reuse/management is a mature practice – we are trying to figure out how to emulate it

- **Better tooling**
  - In SCAP v1, tools (when they existed) were glorified XML editors
  - Developing design patterns for authoring tools that are abstracted above the raw language
Move to continuous monitoring of enterprises

- **SCAP will still support periodic audits, but move beyond that paradigm**

- **Support event-based alerting**
  - If a key control on an endpoint changes, it sends an alert without waiting to be audited

- **Support encoding automated periodicity**
  - Check network ports every minute; check password policy daily; check patch status every boot
  - Allow enterprises to balance operational overhead with need to regular security oversight
Outreach

- SCAP v1 was government led
  - USG led most editing, published most specs, and often set the agenda
- USG kicked off SCAP v2, but has worked to have it be community owned from day 1

- We recognize the need to have active community support
  - A landing page for resources that keep up with the state of SCAP and adoption
  - If we want to attract users, resources can’t look like they were designed by engineers
- Hard work to get people to realize that this is not only in-scope but a critical factor in a successful program
Next steps

- These are all ongoing efforts
  - We continue to have regular teleconferences with sub-teams focused on all these topics
  - Just held our second face-to-face workshop in September

- NIST has a landing page for the effort
    - Or just search for “SCAP v2”
  - This will not be the final community resource page – want something unaffiliated with NIST and USG (and not constrained by NIST release cycles)

- Tell people, join the community, give us feedback...
Questions?
MITRE’s mission-driven teams are dedicated to solving problems for a safer world. Through our federally funded R&D centers and public-private partnerships, we work across government to tackle challenges to the safety, stability, and well-being of our nation.

Learn more www.mitre.org
Backup
SCAP Acronyms (since you cared)

- OVAL – Open Vulnerability and Assessment Language
- XCCDF – eXtensible Configuration Checklist Description Format
- OCIL – Open Checklist Interactive Language
- CVE – Common Vulnerabilities and Exposures
- CCE – Common Configuration Enumeration
- CPE – Common Platform Enumeration
- SWID tags – SoftWare IDentification tags
- CVSS – Common Vulnerability Scoring System
- CCSS – Common Configuration Scoring System
- SCAP Data Streams – SCAP Data Streams
- SCAP – Security Content Automation Protocol

© 2019 The MITRE Corporation. All rights reserved.