8/17/2020

Joe Sain

The MITRE Corporation

SCAP Content Authoring Tool High-Level Requirements Document (v0.6)

Table of Contents

[1 Introduction 1](#_Toc48578586)

[1.1 Scope 1](#_Toc48578587)

[2 Use Cases 1](#_Toc48578588)

[2.1.1 Response 1](#_Toc48578589)

[2.1.2 SCAP Content Development 1](#_Toc48578590)

[2.1.3 Number of SCAP Developers 2](#_Toc48578591)

[2.1.4 Content Authoring Difficulty 2](#_Toc48578592)

[2.1.5 Automation 2](#_Toc48578593)

[2.1.6 Customizations to Existing SCAP 2](#_Toc48578594)

[2.1.7 Storing Created Content 2](#_Toc48578595)

[2.1.8 SCAP Content Sources 3](#_Toc48578596)

[2.1.9 SCAP Content Development Tools 3](#_Toc48578597)

[2.1.10 Willingness to Change Tools 3](#_Toc48578598)

[3 Features and Capabilities 3](#_Toc48578599)

[3.1 High-Level Feature Ranking 3](#_Toc48578600)

[3.1.1 Tooling that Makes SCAP Experts More Efficient 4](#_Toc48578601)

[3.1.2 Automation-Friendly Tooling 4](#_Toc48578602)

[3.1.3 Support for the Latest SCAP Component Specification Versions 4](#_Toc48578603)

[3.1.4 Creating XCCDF Benchmarks and Corresponding OVAL Checks 5](#_Toc48578604)

[3.1.5 Creating XCCDF Benchmarks Using existing OVAL Checks 5](#_Toc48578605)

[3.1.6 Creating Individual OVAL Definitions and OVAL Definition Files (No XCCDF Benchmark) 5](#_Toc48578606)

[3.1.7 Support for Legacy SCAP Component Specification Versions 6](#_Toc48578607)

[3.1.8 Simplified Content Creation for Novice Authors 6](#_Toc48578608)

[3.2 Additional Capabilities 6](#_Toc48578609)

[3.2.1 Ability to Specify Common Actions and Have the Tool Generate Content 6](#_Toc48578610)

[3.2.2 Difference Tracking Between Content Sources 7](#_Toc48578611)

[3.2.3 Support for Version/Revision Control in your Tools for Content 7](#_Toc48578612)

[3.2.4 Ability to Define Macros and Libraries that can be used in future Content 7](#_Toc48578613)

[3.2.5 Ability to Compose and Split Source Data Stream Collections 7](#_Toc48578614)

[3.3 Other Feature and Capability Responses 8](#_Toc48578615)

[Appendix A Use Case Responses A-1](#_Toc48578616)

# 

# Introduction

This document describes the use case information derived from the SCAP Content Authoring Tool user community survey. This tool is intended for use by SCAP authors of all skill levels, from Content Authors with little-to-no SCAP experience, to Content Developers who are SCAP experts. The tool will provide quick and easy authoring capabilities for Content Authors while providing powerful authoring capabilities for Content Developers. The tool will support the use of plugins written by OVAL/SCAP experts to assist in automating customer-specific requirements while streamlining the development of SCAP content.

## Scope

This document is intended to provide the requirements and use case data that developers will need to build the SCAP Content Authoring tool.

# Use Cases

### Response

The SCAP v2 team polled the SCAP community regarding the planned development of the SCAP authoring tool. There were eleven responding organizations for this survey:

* Altex-Soft
* Borelli Security Software
* Canonical
* Center for Internet Security
* DISA FSO
* DISA
* NIST SCAP Validation (G2)
* Joval
* McAfee
* Red Hat
* Siemens

The full responses to the use case questions are provided in Appendix A.

### SCAP Content Development

Organizations are developing a variety of SCAP content, with XCCDF and OVAL being the most common. Vulnerability content and compliance program content are also developed by a majority of the responding organizations.

Respondents indicated that they develop SCAP content for many platforms, with Windows, Linux variants, and MacOS being most common. Other platforms covered include Solaris, AIX, and Cisco IOS.

### Number of SCAP Developers

SCAP authoring teams in responding organizations are relatively small in most cases, averaging between 2 to 4 people. One larger organization had up to 15 authors.

Most SCAP authors created all types of SCAP content rather than focusing on a single content type.

### Content Authoring Difficulty

One of the key drivers for the development of the SCAP Authoring Tool is the complexity involved in writing SCAP content. Accordingly, responding organizations indicated that OVAL and XML pose the greatest challenges, with getting content to run on multiple OVAL engines being the top response. The general complexity of OVAL development was listed as a challenge as well.

XML was also noted as difficult to deal with in general, with learning what XML attributes to populate noted as particularly challenging.

Respondents also noted that XCCDF cannot be authored and maintained in an acceptable way without an authoring tool.

In addition to SCAP-specific content challenges, supporting new platforms in SCAP was noted as a challenge. Getting consistent and reliable results using multiple tools across OSes and products is also a problem for SCAP authors.

### Automation

Respondents were asked about automation and what capabilities would be most beneficial to automate. Automated creation of simple, common elements and support for managing complex structures were the most frequent answers. Others noted that support for macros that could be defined for common actions would be beneficial.

### Customizations to Existing SCAP

Respondents were asked about customizations that they performed to existing SCAP content or use of 3rd party content. Answers were more varied. A number of respondents indicated that they do not use any 3rd party content, opting to develop all of their SCAP natively. Others noted that they correct errors in publicly available OVAL content, tailor and customize content to customer specifications, edit benchmarks, and rewrite checks.

### Storing Created Content

Respondents were asked how their created SCAP content is stored. The majority indicated that they store their content in a database, but several noted that their content is stored in directories in a file system, uploaded to GitHub or some other source control mechanism, or loaded into scanning tools. This points to the need to be able to output SCAP code in different formats depending on user requirements. Respondents felt that the format to store and deliver SCAP content should be standardized.

### SCAP Content Sources

Respondents were asked from which sources they collect SCAP content. The majority indicated that the CIS OVAL repository is their data source.

### SCAP Content Development Tools

Respondents were asked about the tools that are used to create their SCAP content. The majority indicated that they have built their own in-house tool to meet their business objectives. The next most popular answer was that SCAP content was developed manually.

### Willingness to Change Tools

Respondents were asked how willing they would be to change the tools that they currently use to develop SCAP content if a general-purpose solution was to be developed. Answers here were mixed, with most indicating that it would depend on the features and limitations of the developed tool.

Key challenges to changing SCAP development tools included ongoing internal tool development and programming language lock-in, indicating that they have spent considerable time and effort developing their in-house tools.

# Features and Capabilities

This section provides a weighted score for the list of high-level features and capabilities that were presented to survey respondents. The purpose of the scoring is to provide a prioritized list for tool feature development.

## High-Level Feature Ranking

Respondents were asked to rank a list of high-level features and capabilities that are under consideration for tool development. Answers were divided into three categories, Very Important, Somewhat Important, and Not Important. This document provides a weighted number for each feature:

* Very Important = 4 points
* Somewhat Important = 2 points
* Not Important = 0 points

Each high-level feature description is provided below with the number of responses in each category and the weighted score.

### Tooling that Makes SCAP Experts More Efficient

This question gets to the heart of the SCAP authoring issue that we are attempting to solve: would an authoring tool benefit the SCAP community?

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 10 | 40 |
| Somewhat Important | 1 | 2 |
| Not Important | 0 | 0 |

**Weighted Score: 42**

### Automation-Friendly Tooling

This question probes into development requirements: is it important for the authoring tool to use tooling methods (APIs, code libraries, etc.) that provide a simple, stable mechanism for generating SCAP content elements?

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 10 | 40 |
| Somewhat Important | 1 | 2 |
| Not Important | 0 | 0 |

**Weighted Score: 42**

### Support for the Latest SCAP Component Specification Versions

This question asks about the importance of the tool supporting the latest SCAP component specification versions. It also implies that the tool will be able to be updated to comply with future versions.

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 10 | 40 |
| Somewhat Important | 1 | 2 |
| Not Important | 0 | 0 |

**Weighted Score: 42**

### Creating XCCDF Benchmarks and Corresponding OVAL Checks

This question and the question listed in section 3.1.5 are meant to capture XCCDF output requirements: when creating XCCDF benchmarks, is it important to also create corresponding OVAL checks?

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 7 | 28 |
| Somewhat Important | 3 | 6 |
| Not Important | 1 | 0 |

**Weighted Score: 34**

### Creating XCCDF Benchmarks Using existing OVAL Checks

This question and the question listed in section 3.1.4 above are meant to capture XCCDF output requirements: when creating XCCDF benchmarks, is it important to use existing OVAL checks?

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 6 | 24 |
| Somewhat Important | 3 | 6 |
| Not Important | 2 | 0 |

**Weighted Score: 30**

### Creating Individual OVAL Definitions and OVAL Definition Files (No XCCDF Benchmark)

This question asks about the importance of creating OVAL content without XCCDF benchmarks.

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 6 | 24 |
| Somewhat Important | 2 | 4 |
| Not Important | 3 | 0 |

**Weighted Score: 28**

### Support for Legacy SCAP Component Specification Versions

This question asks about the importance of supporting SCAP component specification versions that are more than a few years old.

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 5 | 20 |
| Somewhat Important | 2 | 4 |
| Not Important | 4 | 0 |

**Weighted Score: 24**

### Simplified Content Creation for Novice Authors

This question asks about the importance of developing a tool that includes a simplified version of SCAP content creation that would enable authors with little-to-no SCAP knowledge to generate SCAP.

|  |  |  |
| --- | --- | --- |
| **Category** | **Responses** | **Score** |
| Very Important | 4 | 16 |
| Somewhat Important | 3 | 6 |
| Not Important | 4 | 0 |

**Weighted Score: 22**

## Additional Capabilities

The following additional capabilities were presented to the respondents as yes or no options. They were asked if these capabilities should be included in a general purpose SCAP authoring tool.

### Ability to Specify Common Actions and Have the Tool Generate Content

This feature would enable the author to specify common actions (e.g., “check registry key,” “check file presence”) and have the tool venerate content. The author would not be required to understand the underlying OVAL/XCCDF language structures; this would be handled by the tool.

|  |  |
| --- | --- |
| **Category** | **Responses** |
| Important | 8 |
| Not Important | 3 |

### Difference Tracking Between Content Sources

This feature would include checks for consistency when dealing with multiple content sources.

|  |  |
| --- | --- |
| **Category** | **Responses** |
| Important | 7 |
| Not Important | 4 |

### Support for Version/Revision Control in your Tools for Content

This feature would incorporate a content author’s internal tool version and revision control tracking into the SCAP content being generated.

|  |  |
| --- | --- |
| **Category** | **Responses** |
| Important | 6 |
| Not Important | 5 |

### Ability to Define Macros and Libraries that can be used in future Content

This feature would facilitate reuse of content development automation elements for future content.

|  |  |
| --- | --- |
| **Category** | **Responses** |
| Important | 5 |
| Not Important | 6 |

### Ability to Compose and Split Source Data Stream Collections

This feature would give the author the ability to split data streams to meet the requirements of individual customers.

|  |  |
| --- | --- |
| **Category** | **Responses** |
| Important | 5 |
| Not Important | 6 |

## Other Feature and Capability Responses

The following features and capabilities were offered by respondents as being important and that should be incorporated into the development requirements:

* Standardization of the SCAP repository:
  + Storage.
  + Interfaces.
  + Version control.
  + Proof of concept.
  + Public repository.
  + Adoption.
  + Full support of each SCAP component specification.
* Maintenance and support.
  + How long will the SCAP authoring tools be maintained and supported?
* Identifier management for control mappings.
* Integration capabilities into existing tools.
* Ability to allow different levels of authoring:
  + *Policy Level:* Given a path to group policy, construct the appropriate SCAP content.
  + *Mid-Level:* "Check registry key," or "check file presence."
  + Technical Level: Allow users to construct OVAL directly.
* Allow for the ability to abstract SCAP technical details from the author.
* Schema authoring: The ability to use the tool to create new OVAL constructs.
* *Advice*: Develop an SCAP content management environment, not a general purpose SCAP authoring tool.

###### Use Case Responses

1. What type of SCAP content are you authoring and for what operating systems or applications?

|  |  |
| --- | --- |
| XCCDF | 11 |
| OVAL | 8 |
| Vulnerability Content | 8 |
| Compliance Program Content | 8 |

Operating systems supported started with Windows releases to various flavors of Linux, MacOS, AIX, Solaris, and Cisco IOS.

1. Approximately how many SCAP content authors are there in your organization?

|  |  |
| --- | --- |
| Authors create all types of SCAP content | 4 to 15 (avg: 9.5) |
| OVAL | 2 to 3 |
| XCCDF | 2 to 4 |

1. What specific areas of content authoring are the most difficult for your authors?

|  |  |
| --- | --- |
| Getting content to run on multiple OVAL engines | 5 |
| Learning what XML attributes to populate | 4 |
| Writing XML that validates | 3 |
| It is difficult to know what OVAL schemas to use | 3 |

**Other answers included:**

* OVAL is incredibly complex, does not scale well, and is difficult to write
* XCCDF cannot be authored and maintained in an acceptable way without an authoring tool.
* Figuring out how to use OVAL tests and constructs accurately.
* Dealing with XML in general is difficult.
* New platform coverage in SCAP formats.
* Getting consistent and reliable results using multiple tools across OSes and products.

1. What automation would assist you in the authoring of the content that you are developing?

|  |  |
| --- | --- |
| Automate creation of simple, common elements | 6 |
| Support for managing complex structures | 6 |
| Support for macros that can be defined for common actions | 4 |
| Support for automated filing and tracking of versions to simplify reuse | 3 |
| Example templates | 3 |
| The ability to compose and split source data stream collections | 2 |

**Other answers included:**

* We developed our internal .net software for SCAP content creating.
* Assistance in the creation of new OVAL schema documentation and proposal to the OVAL community.
* Ability to wrap OVAL around something like a bash or PowerShell script.
* Ability to create and test OVAL test types.
* We created our own tool for creating SCAP content.
* A standard that is easier for people to pick up and use.

1. What customizations to existing SCAP content or 3rd party SCAP content do you perform?

* We correct errors in CIS OVAL content.
* We built our tool from scratch as there was no good starting point.
* CIS is a content creator. Our authoring platform allows for the creation of CIS SCAP content and allows our SecureSuite members to tailor that content as well.
* None. DISA authors all of their content.
* Tailoring and customizing SCAP content.
* Editing benchmarks and creating new benchmarks by fusing existing OVAL checks together for things they want to track.
* We rewrite checks to improve them.
* Tailoring capabilities are available in the product. Can be our content or others. For the most part, McAfee does not use external SCAP content. All internal components are done in house.
* We base internal baselines on externally available SCAP content.

1. How do you store the content you create?

|  |  |
| --- | --- |
| **Database** | **6** |
| **Directories in a file system** | **4** |
| **Content is loaded into scanning tools** | **4** |
| **GitHub/other source control mechanism** | **4** |

**Other answers included:**

* We write directly to the DISA checklists but will probably back up a step to produce raw XCCDF output as well, to facilitate testing with NIST.
* Company Git Repos and servers.
* Will implement a GitHub-like solution soon.
* IMO, the format to store and deliver SCAP content must be standardized. Storing the content in file format is good, but not optimal.
* For development – done inside content management system. For runtime – database driven, loaded into the product. SCAP package set update 1300 released June 12.

1. From which external SCAP sources do you collect content?

|  |  |
| --- | --- |
| **CIS OVAL repository** | **5** |
| **Red Hat** | **2** |
| **IASE/DISA** | **2** |
| **Novell (SuSE)** | **1** |
| **NIST NCP** | **1** |
| **McAfee** | **1** |
| **Canonical** | **1** |

1. What tools do you use to create SCAP content? What enhancements, if any, would you like to see with your current tools?

|  |  |
| --- | --- |
| **Developed In-House Tool** | **5** |
| **Manual Creation – Oxygen XML Editor/Scripts** | **3** |
| **Developed .NET application** | **2** |
| **McAfee Policy Editor Content Creator** | **1** |

1. How willing would you be to change the tools that you are currently using if a general-purpose SCAP authoring solution was to be developed?

|  |  |
| --- | --- |
| **It Depends** | **4** |
| **Not Very Interested** | **2** |
| **Willing** | **2** |
| **Very Willing** | **2** |
| **Not Interested** | **1** |

1. Are there any challenges your organization would face when adopting a new SCAP authoring solution?

|  |  |
| --- | --- |
| **Ongoing Tool Development** | **6** |
| **Programming Language Lock-in** | **4** |
| **Operating System Lock-in** | **1** |
| **Browser-based vs. Compiled Code** | **1** |

**Other Answers included:**

* We would prefer the ability to use editor of choice on a Linux-based system.
* A new SCAP authoring solution would need to meet a very high threshold/standard in order to be adopted internally. We currently support multiple teams involved in both Benchmark and SCAP content creation.
* No serious challenges, provided that the tool fully implements the current specifications.
* Vendor lock-in with custom content.
* We would need confidence in the quality and completeness of the solution and/or ability to improve and extend it.

1. Would your organization be willing to provide occasional feedback during the development of an authoring solution?

|  |  |
| --- | --- |
| **Yes** | **8** |
| **Possibly** | **1** |
| **Need More Info** | **1** |
| **Not Sure** | **1** |

1. Would your organization consider contributing software development resources toward the development of an SCAP authoring solution (architects, developers, beta testers)?

|  |  |
| --- | --- |
| **Possibly** | **4** |
| **Beta Testers** | **3** |
| **Yes** | **2** |
| **Probably Not** | **1** |
| **Need More Info** | **1** |