FedRAMP

Developer Data Bites

October 27, 2022

GSA

info@fedramp.gov

fedramp.gov

Introduction

Purpose: To create space for dialogue between developers who use OSCAL and the FedRAMP® automation team.

Outcomes:

- Understanding of the OSCAL tools FedRAMP currently uses
- Awareness of NIST's OSCAL CLI tool and other NIST resources

Agenda:

- Welcome
- Pre-Submitted Q&A
- FedRAMP Tools Review
- NIST OSCAL Tools Demo
- Open Forum
- Next Steps & Closing

Data Bites Guiding Principles

Keep the discussion respectful

Be curious, seek understanding

Speak from your own experience

Challenge through questions

Focus on ideas

Keep it technical

Pre-Submitted Questions

Pre-Submitted Questions

Question 1: LI-SaaS Control Implementation (Rev 4): Are users supposed to fill out the parent control or all sub statements? As the parent statement does not contain any details, how would they address this?

If the parent statement does not contain any requirement details then users do not need to fill out the parent control. However, any parent control that exists without a requirement has sub-requirements thus requiring sub-statements. Sub statements should be included in this case.

Additionally: A rev 4 Issue exists such that high, moderate, and low have "response-point" prop at control part level, whereas LI-SaaS profile and CIS WB only specifies it at the parent control level.

Pre-Submitted Questions

Question 2: What scenario requires fedramp-li-saas profile to be used?

The FedRAMP LI-SaaS profile is for Low-Impact SaaS applications that do not store personal identifiable information (PII) beyond that generally required for login capability (i.e. username, password, and email address). See https://www.fedramp.gov/understanding-baselines-and-impact-levels/ for more information

FedRAMP Tools Review

FedRAMP Tools Review

What tools does FedRAMP use for OSCAL Artifact Validation and Conversion?

OSCAL Artifact Validation/Conversion Steps:

1. **Validate File Format** in JSON, XML or YAML
   - □ Any valid schema aware editor (VS Code (free), XML Notepad, OxygenXML). Plenty of Open Source tools available.
   - □ XML musts: root elements, element closing tags, case sensitive tags, properly nested elements, and quoted attributes
2. **Validate file's NIST OSCAL Schema** against Version 1.0.x of OSCAL.
   - □ We use the tools available off of the NIST OSCAL website and NIST Github. You can also use VS Code or OxygenXML to do this!
3. **Convert file to XML** and **validate** using FedRAMP Schematron validations for FedRAMP specific extensions to NIST Core OSCAL.
   - □ Conversion: https://github.com/usnistgov/oscal-cli/
   - □ Validation: https://github.com/GSA/fedramp-automation/src/validations

NIST OSCAL CLI Tool

NIST OSCAL Tools

*The NIST OSCAL Command Line Tool enables users working with OSCAL in XML, JSON, and YAML to **simplify** basic operations.*

What is NIST's OSCAL Command Line Interface (CLI)?

The OSCAL CLI is a **Java** command line tool that performs common operations on Open Security Controls Assessment Language (OSCAL) and Metaschema content such as:

- Content Validation
- Conversion
- Profile resolution
- Schema generation

CLI Tool Demo

Open Forum

Thank you

Save the Date! Our next Developer Data Bites virtual meeting will be on **Thursday, November 10, 2022 at 12p EDT.**

Submit questions and future discussion topics to OSCAL@fedramp.gov

Learn more at fedramp.gov

@FEDRAMP

How to Submit Issues with FedRAMP

Ensuring your outstanding issues or questions are received:

Issues can be submitted in several ways:

- **Preferred:** Open an issue on fedramp-automation github so that it will benefit the NIST/FedRAMP community. https://github.com/GSA/fedramp-automation/issues

- **Alternate:** Email us at oscal@fedramp.gov

OSCAL Resources

NIST:

https://pages.nist.gov/OSCAL/

Learning Resources: https://pages.nist.gov/OSCAL/learn/

Current release: https://github.com/usnistgov/OSCAL/releases

Development version: https://github.com/usnistgov/OSCAL/tree/develop

Content repo: https://github.com/usnistgov/oscal-content

FedRAMP:

Current repo: https://github.com/GSA/fedramp-automation

Current issues: https://github.com/GSA/fedramp-automation/issues

Validations work: https://github.com/18F/fedramp-automation/tree/master/src/validations

Web based validation tool:

https://federalist-2372d2fd-fc94-42fe-bcc7-a8af4f664a51.app.cloud.gov/site/18f/fedramp-automation/#/documents/system-security-plan