FedRAMP

Developer Data Bites

November 10, 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:

- Awareness of FedRAMP's proposed end state for Automated Continuous Monitoring
- Feedback on the proposed end state
- Productive Discussion around OSCAL for FedRAMP

Agenda:

- Welcome
- Pre-Submitted Q&A
- Automated ConMon Vision
- Open Forum
- Next Steps & Closing

Data Bites Guiding Principles

- Keep the discussion respectful
- Speak from your own experience
- Focus on ideas
- Be curious, seek understanding
- Challenge through questions
- Keep it technical

Pre-Submitted Questions

Pre-Submitted Questions

Question 1: Should we generate a new UUID with each new iteration of an SSP?

UUIDs in OSCAL are intended to uniquely identify information and link information between various OSCAL documents.   That being said, they are intended to be **consistently** used to represent the same concept over multiple major and minor revisions of the same document; they should only be changed if the underlying identified subject has changed in a **significant** way.

(See question 2 for additional information)

Pre-Submitted Questions

Question 2: Should we be persisting UUIDs for any of the other objects (controls, components, assets, etc.)?

The UUIDs should not change for controls, components, assets, etc). If controls, components, assets, etc. are added in a subsequent release of the document they would get new UUIDs. It is important to note that the downstream artifacts from the SSP (SAP, SAR, POAM) import from the SSP or preceding artifact. If you change the UUIDs in the SSP then the downstream artifacts must have matching UUIDs.

Automated ConMon Vision

ConMon Vision

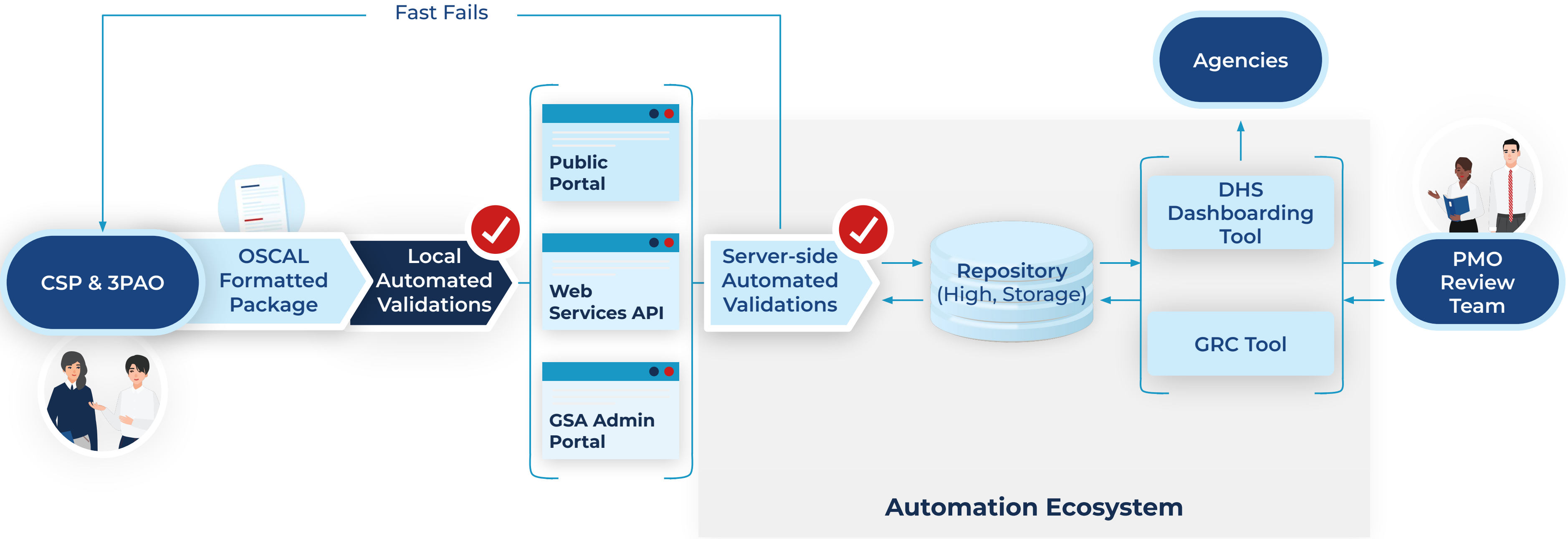
*What are the **opportunities** for the future of ConMon?*

Future-State Opportunities

The PMO aims to establish and deploy a ConMon ecosystem consisting of the following:

- Standardized data model leveraging OSCAL
- Automated data collection via **Web Services API**
- Automated validations of ConMon deliverables to identify issues at the gateway
- Dashboarding (i.e. DHS CDM) that models risk against capabilities and sub-capabilities
- Clearly defined defect checks for each sub-capability that translate into actionable assessment plans for CSPs.

Future State

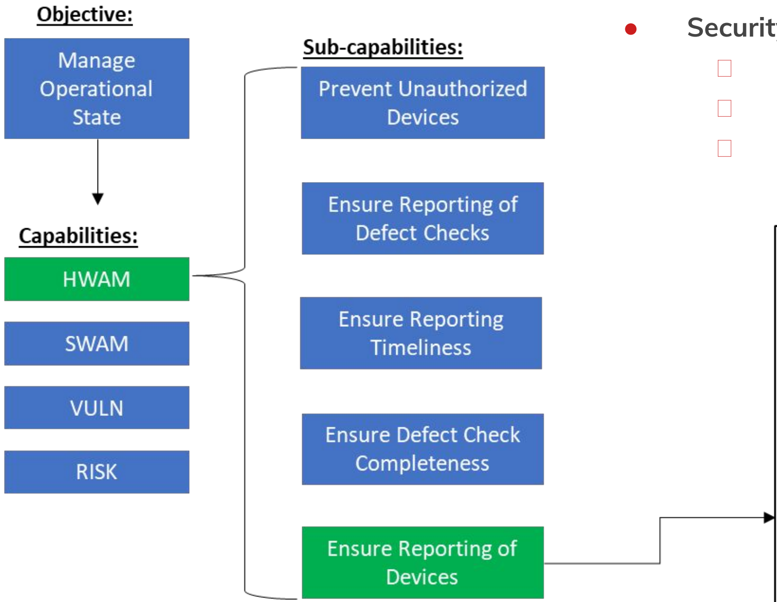


Terminology

- **Security capability** - A set of mutually reinforcing security controls implemented by technical, physical, and procedural means. They are typically defined to achieve a common information security-related purpose.
- **Sub-Capability** - A capability that supports the achievement of a larger capability. Each defined capability is decomposed into the set of sub-capabilities that are necessary and sufficient to support the larger capability.
- **Control Item** - All or part of a SP 800-53 security control requirement, expressed as a statement for implementation and assessment. Both controls and control enhancements are treated as control items. Controls and control enhancements are further subdivided if multiple security requirements within the control or control enhancement in SP 800- 53 are in listed format: a, b, c, etc.
- **Assessment Objective** - A set of determination statements that expresses the desired outcome for the assessment of a security control, privacy control, or control enhancement.
- **Defect Check:** A defect check is a way to assess determination statements. It has the following additional properties.
  - Is stated as a test (wherever appropriate)
  - Can be automated
  - Explicitly defines a desired state specification that is then compared to the corresponding actual state to determine the test result
  - Provides information that may help determine the degree of control effectiveness/level of risk that is acceptable
  - Suggests risk response options
  - Assesses a corresponding sub-capability.

Relationship: Security Capabilities & Controls

- **Security Capabilities:** define mission-oriented security objectives
  - □ 16 capabilities divided into sub-capabilities
  - □ Defect checks can be written for each sub-capability
  - □ Defect checks provide artifacts to demonstrate compliance with control requirements and support the automation of assessment objectives



Objective:

Manage Operational State

Capabilities:

HWAM

SWAM

VULN

RISK

Sub-capabilities:

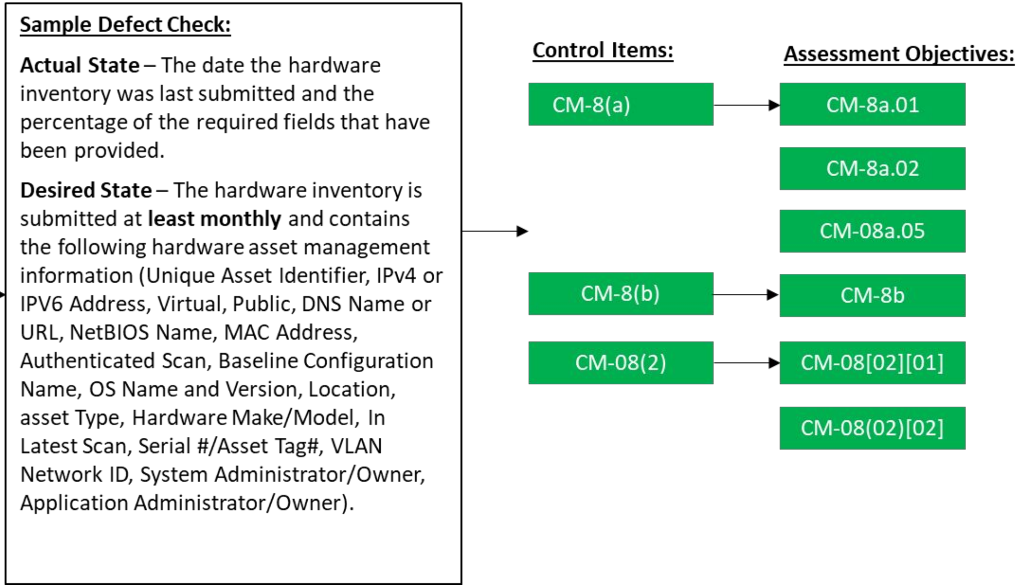
Prevent Unauthorized Devices

Ensure Reporting of Defect Checks

Ensure Reporting Timeliness

Ensure Defect Check Completeness

Ensure Reporting of Devices



Sample Defect Check:

Actual State – The date the hardware inventory was last submitted and the percentage of the required fields that have been provided.

Desired State – The hardware inventory is submitted at **least monthly** and contains the following hardware asset management information (Unique Asset Identifier, IPv4 or IPV6 Address, Virtual, Public, DNS Name or URL, NetBIOS Name, MAC Address, Authenticated Scan, Baseline Configuration Name, OS Name and Version, Location, asset Type, Hardware Make/Model, In Latest Scan, Serial #/Asset Tag#, VLAN Network ID, System Administrator/Owner, Application Administrator/Owner).

Control Items:

CM-8(a)

CM-8(b)

CM-08(2)

Assessment Objectives:

CM-8a.01

CM-8a.02

CM-08a.05

CM-8b

CM-08[02][01]

CM-08(02)[02]

Sub-Capability: Ensure reporting of Devices

Sub-capability Purpose: Ensure that individual devices are regularly reported in the actual state inventory to prevent defects associated with other capabilities from going undetected.

Defect Check:

Actual State - The date the hardware inventory was last submitted and the percentage of the required fields that have been provided.

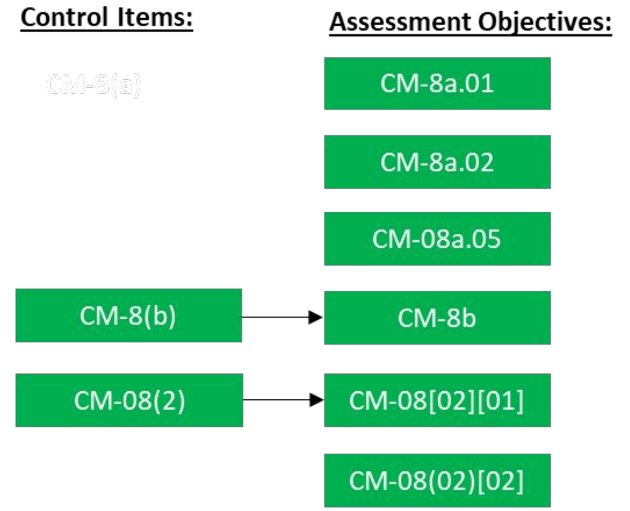
Desired State - The hardware inventory is submitted at **least monthly** and contains the following hardware asset management information (Unique Asset Identifier, IPv4 or IPV6 Address, Virtual, Public, DNS Name or URL, NetBIOS Name, MAC Address, Authenticated Scan, Baseline Configuration Name, OS Name and Version, Location, asset Type, Hardware Make/Model, In Latest Scan, Serial #/Asset Tag#, VLAN Network ID, System Administrator/Owner, Application Administrator/Owner).

Defect - A defect occurs when any of the following is observed:

The hardware inventory has not been submitted within the last 30 days

The hardware inventory is submitted but it is not complete (i.e., completion percentage is less than the defined threshold).

Control Items and Assessment Objectives Supported



Control Items:

Assessment Objectives:

CM-8(a)

CM-8a.01

CM-8a.02

CM-08a.05

CM-8(b) → CM-8b

CM-08(2) → CM-08[02][01]

CM-08(02)[02]

Sub-Capability: Ensure reporting of Devices

CSP Activity Description:
Provide an OSCAL compliant SSP containing a complete, up-to-date hardware inventory, ensuring the following is observed:

- The hardware inventory has been updated within the last 30 days
- The hardware inventory submitted contains all the required asset management information:
    - Unique Asset Identifier, IPv4 or IPV6 Address, Virtual, Public, DNS Name or URL, NetBIOS Name, MAC Address, Authenticated Scan, Baseline Configuration Name, OS Name and Version, Location, asset Type, Hardware Make/Model, In Latest Scan, Serial #/Asset Tag#, VLAN Network ID, System Administrator/Owner, Application Administrator/Owner

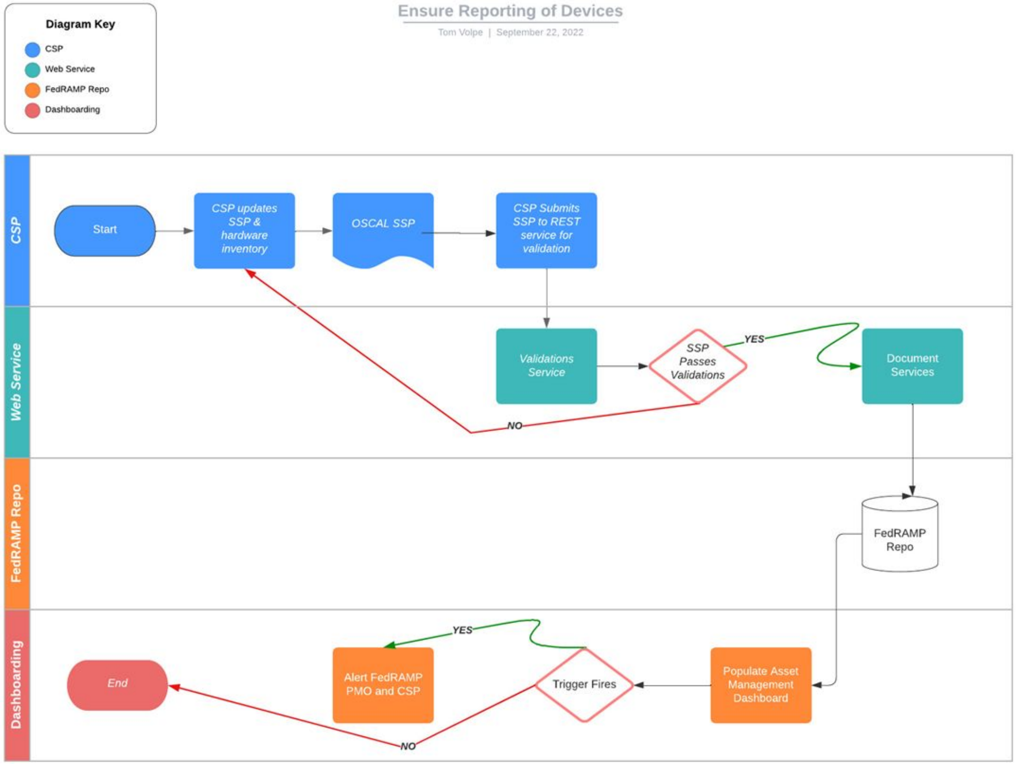


Diagram Key
CSP
Web Service
FedRAMP Repo
Dashboarding
Ensure Reporting of Devices
Tom Volpe | September 22, 2022
CSP
Web Service
FedRAMP Repo
Dashboarding
Start
CSP updates SSP & hardware inventory
OSCAL SSP
CSP Submits SSP to REST service for validation
Validations Service
SSP Passes Validations
YES
NO
Document Services
FedRAMP Repo
Populate Asset Management Dashboard
Trigger Fires
YES
NO
Alert FedRAMP PMO and CSP
End

OSCAL Inventory Demo

Open Forum

Thank you

Save the Date! Our next Developer Data Bites virtual meeting will be on **Thursday, December 8, 2022 at 12p EST.**

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