Open Security Controls Assessment Language

The Anatomy of OSCAL Models?

OSCAL 101 Series - Lecture #2

Presenters:
Dr. Michaela Iorga
NIST, OSCAL Strategic Director

Robert Sherwood
Principal, Credentive Security
NIST is hosting a series of monthly educational workshops, on the third Tuesday of each month, 11:00-12:00 EST.

Purpose: improve OSCAL adoption by expanding the OSCAL community of interest (COI) through the onboarding of members who have no previous knowledge of OSCAL.


Welcome to the Lecture #2

Agenda

- Brief Review of OSCAL
- The Anatomy of OSCAL models
  - Catalog and Profile Models
  - Rob’s Yellow Bricks Road to the FPKI OSCAL Catalog
OSCAL is a standardized, flexible, open-source language designed to express security controls and their associated implementations and assessment methods in machine-readable formats (XML, JSON, and YAML). OSCAL content can be easily transformed into human-friendly formats.

OSCAL:
- Enables automated traceability
- Provides a standards-based foundation for the next generation GRCs
- Helps improve the risk management posture, consistency, and interoperability.
OSCAL: the Open Security Controls Assessment Language

Providing control-related information in machine-readable formats.

NIST, in collaboration with industry, is developing the Open Security Controls Assessment Language (OSCAL). OSCAL is a set of formats expressed in XML, JSON, and YAML. These formats provide machine-readable representations of control catalogs, control baselines, system security plans, and assessment plans and results.
Model Reference

OSCAL is distributed in a series of releases that represent increments of features and functionality that have been added to OSCAL over time.

This reference provides format documentation for the following OSCAL releases.

- Development Snapshot
- Latest Release (v1.0.4)
- 1.0.0
- 1.0.0-rc2
- 1.0.1
- 1.0.2
- 1.0.3
OSCADC Models’ Outline

Common OSCAL Structure

Complete v1.0.4 JSON Format Outline

The following outline is a representation of the JSON format for the combination of all OSCAL models. For each property, the data type is also provided. The cardinality and data type are also provided for each property where appropriate.

```json
▼ catalog [1]:
  uuid [1]: uuid,
  ▼ metadata [1]: { ... },
    ▼ metadata [1]: { ... },
      ▼ params [0 or 1]: [ ... ],
      ▼ controls [0 or 1]: [ ... ],
      ▼ groups [0 or 1]: [ ... ],
      ▼ back-matter [0 or 1]: { ... },

▼ profile [1]:
  uuid [1]: uuid,
  ▼ metadata [1]: { ... },
    ▼ imports [1]: [ ... ],
    ▼ merge [0 or 1]: { ... },
    ▼ modify [0 or 1]: { ... },
    ▼ back-matter [0 or 1]: { ... },

▼ component-definition [1]:
  uuid [1]: uuid,
  ▼ metadata [1]: { ... },
    ▼ import-component-definitions [0 or 1]: [ ... ],
    ▼ components [0 or 1]: [ ... ],
    ▼ capabilities [0 or 1]: [ ... ],
    ▼ back-matter [0 or 1]: { ... },

Root Element & Root UUID
Body (Model Specific)
Root Element & Root UUID
Body (Model Specific)
Root Element & Root UUID
Body (Model Specific)
```
Common OSCAL Structure

- **Root Element**: The root element of the document indicates the type of content within the body of the file. The name of this element is unique to the specific model.

- **Root UUID**: A RFC 4122 Version 4 Universally Unique Identifier (UUID) that identifies the specific document instance. Changed when the document is modified.

- **Metadata**: Information about the document (i.e., title, last-modified timestamp, OSCAL version). Also used to define roles, parties (people, teams and organizations), and locations referenced in the document.

- **Model-specific Body**: The body is specific to each model.

- **Back Matter**: Used to link to and attach resources, which may contain citations. Used to associate graphics, supporting documentation, etc. with the OSCAL document. A reference entry here can be referenced from within the body of an OSCAL document.
The Metadata Element – Cardinality and Data Type

```
> catalog [1]: {
    uuid [1]: uuid,
    > metadata [1]: {
        title [1]: markup-line,
        published [0 or 1]: dateTime-with-timezone,
        last-modified [1]: dateTime-with-timezone,
        version [1]: string,
        oscal-version [1]: string,
        > revisions [0 or 1]: [ ... ],
        > document-ids [0 or 1]: [ ... ],
        > props [0 or 1]: [ ... ],
        > links [0 or 1]: [ ... ],
        > roles [0 or 1]: [ ... ],
        > locations [0 or 1]: [ ... ],
        > parties [0 or 1]: [ ... ],
        > responsible-parties [0 or 1]: [ ... ],
        remarks [0 or 1]: markup-multiline,
    }
```
```
 ▼ back-matter [0 or 1]: {
   ▼ resources [0 or 1]: [
       An array of resource objects [1 to ∞] {
         uuid [1]: uuid,
         title [0 or 1]: markup-line,
         description [0 or 1]: markup-multiline,
         ▶ props [0 or 1]: [ ... ],
         ▶ document-ids [0 or 1]: [ ... ],
         ▶ citation [0 or 1]: { ... },
         ▶ rlinks [0 or 1]: [ ... ],
         ▶ base64 [0 or 1]: { ... },
         remarks [0 or 1]: markup-multiline,
       }
   }
}
```
OSCAL Controls Layer
OSCAL Catalog Model

Represents a collection of security and privacy controls, which may be used as part of a risk management program.

- **Metadata:** Same for each OSCAL model
- **Parameter:** Provides a global policy variable used by one or more controls
- **Control:** An individual control in the catalog.
  - May contain control-specific parameters, control requirement statements, control objectives, assessment methods, references
  - Controls can have child controls.
- **Group:** Related controls may be grouped. Parameters related to this group may be defined here.
- **Back Matter:** Same for each OSCAL model
Catalog Model v1.0.4 JSON Format Outline

The following outline is a representation of the JSON format for this model. For each property, the name links to the corresponding entry in the JSON Format Reference. The cardinality and data type are also provided for each property where appropriate.

```
  ▼ catalog [1]: {
    ▶ uuid [1]: uuid,
    ▶ metadata [1]: { ... },
    ▶ params [0 or 1]: [ ... ],
    ▶ controls [0 or 1]: [ ... ],
    ▶ groups [0 or 1]: [ ... ],
    ▶ back-matter [0 or 1]: { ... },
  }
```

Model Reference

- Data Types
- Release Notes
- Development Snapshot
OSCAL Profile Model

Used to establish a baseline of controls to be implemented with a system.

- **Metadata:** Same for each OSCAL model

- **Import:** Identifies an OSCAL catalog or other profile to import controls from
  - A control must be imported to be included in a baseline.
  - All parameters and back-matter resources cited by an imported control are also imported.

- **Merge:** Provides directives used to organize controls and to resolve conflicts when the same control is imported multiple times

- **Modify:** Allows tailoring of imported controls, including their parameters, control requirement definitions, references, control objectives, and assessment actions.

- **Back Matter:** Same for each OSCAL model
Profile Model v1.0.4 JSON Format Outline

The following outline is a representation of the JSON format for this model. For each property, the name links to the corresponding entry in the JSON Format Reference. The cardinality and data type are also provided for each property where appropriate.

```json
profile [1]: {
    uuid [1]: uuid,
    ▶ metadata [1]: { ... },
    ▶ imports [1]: [ ... ],
    ▶ merge [0 or 1]: { ... },
    ▶ modify [0 or 1]: { ... },
    ▶ back-matter [0 or 1]: { ... },
}
```

Model Reference

- Data Types
- Release Notes
- Development Snapshot

Latest Release (v1.0.4)
A profile can import controls from:

- A catalog or multiple catalogs
- Another profile or multiple profiles

This allows a baseline to be established by customizing another baseline.
OSCAL Content Validation
https://pages.nist.gov/OSCAL/concepts/validation/

"well-formed" vs "valid" OSCAL content

XML Schema Validators:
https://www.w3.org/XML/Schema#Tools

JSON Schema Validators:
https://json-schema.org/implementations.html#validators
Rob’s Yellow Bricks Road to the FPKI OSCAL Catalog
Thank you!

OSCAL is a community-driven program! Please join us!

Contact us at: oscal@nist.gov
Subscribe to our mailing lists: oscal-dev@list.nist.gov or oscal-updates@list.nist.gov
Chat with us on Gitter: https://gitter.im/usnistgov-OSCAL/Lobby
Collaborate with us on GitHub: https://github.com/usnistgov/OSCAL
Join our COI meetings: https://pages.nist.gov/OSCAL/contribute/#community-meetings

OSCAL Catalog Tutorial: https://pages.nist.gov/OSCAL/learn/tutorials/control/basic-catalog/

https://www.nist.gov/OSCAL
Open Floor Discussion

Ground Rules of Engagement

- Keep the discussion respectful by:
  - using welcoming and inclusive language
  - being respectful of differing viewpoints and experiences
  - gracefully accepting constructive criticism
  - wait for one speaker to finish before speaking

- Speak from your own experience instead of generalizing.

- Do not be afraid to respectfully challenge one another by asking questions focused on ideas not on the company or presenter.

- The final goal is not to always agree but rather gain a deeper understanding.