Open Security Controls Assessment Language Catalog Authoring Tool

a.k.a.

OSCAL

CAT

Dmitry Cousin,
CSD/NIST

Sep 7, 2022
**OSCAL Models**

**CATALOG MODEL**

- **Profile (Control Baseline)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import (Catalog or Profile)

- **Profile (Control Baseline)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import (Catalog or Profile)

**PROFILE MODEL**

- **System Security Plan (SSP)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Profile URI pointing to a Profile

- **System Security Plan (SSP)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Profile URI pointing to a Profile

**POA&M MODEL**

- **Assessment Plan (AP)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import (Catalog or Profile)

- **Assessment Plan (AP)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import (Catalog or Profile)

**ASSESSMENT PLAN MODEL**

- **Plan of Action and Milestones (POA&M)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import SSP URI pointing to an SSP

- **Plan of Action and Milestones (POA&M)**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import SSP URI pointing to an SSP

**COMPONENT MODEL**

- **Component Definition**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Component Definition URI pointing to an OSCAL SSP

- **Component Definition**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Component Definition URI pointing to an OSCAL SSP

**ASSESSMENT RESULTS MODEL**

- **Evidence Attachments**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Evidence Attachments Evidence Attachments: Recent Activities, Incident Notes, Screen Shots, Photos, Tool Reports, New Outage, Penetration Test Report

- **Evidence Attachments**
  - Metadata: Title, Version, Date, Document Labels, Revision History, Prepared By/For Roles, People, Teams, Locations
  - Import Evidence Attachments Evidence Attachments: Recent Activities, Incident Notes, Screen Shots, Photos, Tool Reports, New Outage, Penetration Test Report

---

The import arrow identifies what OSCAL content is linked as a result of the import statement. Imported content referenced, not copied.
What is OSCAL CAT?

CAT is a proof of concept for an “easy to use” multiplatform OSCAL editing tool for Catalogs and Profiles expressed in JSON.

CAT is meant to:
- Generate OSCAL Catalogs (after resolution) and OSCAL Profiles
- Import & Modify OSCAL Catalogs (“tailor controls”)
- Generate OSCAL Profiles
- Be connection and platform-independent
- TBD: Resolve Profiles into Catalogs

Source will become open repository on GitHub.

CAT is currently not
- Yet production-ready
- Yet working with multiple Catalogs
- Yet ready to add new Controls
- Yet allow to manage Back-Matter
- Yet able to do assisted markdown editing
- Yet completely fool-proof or user-safe
OSCAL CAT Technical Details

**Development Toolkit Details:**
- CAT is based on the Ionic 6 (6.19.1) – Angular 13 (13.2.2) multi-platform library.
- CAT code mainly consists of TypeScript/HTML/CSS-SCSS/JSON
- CAT is also currently using:
  - QuickType to generate TypeScript Interfaces from OSCAL JSON-Schema
  - AJV to validate downloaded entities’ compliance to corresponding JSON-Schema
  - Shell-scripts (MAC) for some occasional and day-to-day tasks
- CAT currently uses Ionic Storage for persistence. Ionic Storage can decide to use IndexedDB or local-storage.
- In some environments ionic can even use SQLite with a driver & a native build

**Application Packaging Options:**
- CAT can be packaged as a desktop app using Ionic-Electron
- CAT can be served as a locally-served or web-hosted application in a browser (demonstrated)
- CAT can be compiled for iOS/Android as a native app

**Application Source:**
- CAT source will become open, when the project is ready
- CAT source is currently stored in a private repository on GitHub.
**OSCAL CAT Technical Details**

**Authoring Mode – Update Cat-Files Rev4 (New):**

- **Start with a Known OSCAL Catalog**
  - Catalog 800-53 Revision 4
  - Baselines
    - Low
    - Moderate
    - High

- **Continue with the Previously Saved Work**
  - Profile Draft Based on 800-53.Rev-4 [e875f6d-fda9-4741-bd86-87e8169172]
  - Profile Draft Based on 800-53.Rev-5 [e904e64a-3602-44d5-b7cd-df0ab010af0]
Authoring Mode – Update Cat-Files Rev5 (New):

Catalog 800-53 Revision 5

Baselines:
- Low
- Moderate
- High
- Privacy

Continue with the Previously Saved Work:
- Profile Draft Based on 800-53.Rev-4 [e87d568d-fde9-4741-bd6b-8f7e61691172]
- Profile Draft Based on 800-53.Rev-5 [e904e64a-9e62-44d6-b7dd-df0d0ab00a00]
OSCAL CAT Technical Details

Meta-Data Edit (Validation):

Title: My best shot at picking important controls

Last Modified Date: Sep 7, 2022 @ 02:14 (GMT-04:00)

Catalog Version: VERSION-1

Parties:
- Adam
- Eve
- Lucy
- Bad.Org
- Good.Org
- Creator.Org

Roles:
- Content Expert [prepared-by]
- Content Sponsor [prepared-for]
- Content Approver [content-approver]
- Creator [creator]
Select Controls – Highlight Baseline (New):
OSCAL CAT Technical Details

Select Controls – Pre-Check Baseline (New):
OSCAL CAT Technical Details

Select Controls – Cut-Out Baseline (New):

Choose Cut-Out Baseline...
- Low
- Moderate
- High
- Privacy
OSCAL CAT Technical Details

Select Controls – Pre-Check Baseline (Activated):
Modify Controls – Only Selected Controls:

Selected Controls

Modify Selected Controls

You can modify selected controls to tailor them to your profile.

- 800-53 Rev5 Catalog
  - Access Control (AC)
    - Access Control Policy and Procedures (AC-1)
    - Account Management (AC-2)
      - AC-2 Enhancement: Automated System Account Management (AC-2.1)
      - AC-2 Enhancement: Removal of Temporary / Emergency Accounts (AC-2.2)
      - AC-2 Enhancement: Disable Inactive Accounts (AC-2.3)
      - AC-2 Enhancement: Automated Audit Actions (AC-2.4)
      - AC-2 Enhancement: Inactivity Logout (AC-2.5)
      - AC-2 Enhancement: Dynamic Privilege Management (AC-2.6)
Download Profile/Settings – Unfinished(Left)-Build Profile(Right)

OSCAL Catalog Authoring Tool (CAT)

Catalog Parts:
- Authoring Mode
- Meta Info
- All Controls
- Selected Controls
- Regroup Controls

Catalog Files:
- Download File
- Upload File

Configuration:
- Settings

Save your current profile locally (when ready or to keep it), or save workspace until the next time.

Workspace represents the collection of the objects and their states.

The objects reflect selections, actions, and transitions that modified catalogue or baseline, while you were working with the application. The

Final file is a sample of the profile you had tailored in the application

The final file will contain JSON representation of an OSCAL Catalog or OSCAL Profile. The file will have all your changes.
Parametral Settings – Moved to UI from Code (New) In-Progress Big Time

- Offer to Update Catalogs, Profiles, and Schemas in: 0.0009 hours
- Extend Functionality in Meta for Demo: False
- Clear OSCAL-CAT storage (!!!This will wipe out all settings and previously persisted work!!!)

SAVE EXPIRATION INTERVAL
SAVE DEMO FLAG
CLEAR STATE STORE
Originally Envisioned Features:

- Choose Catalog
- Edit metadata information *(common across all models)*
- Make profile: select controls (import), modify, regroup
- Use “future JSON tool(s)” to resolve profile

```javascript
export interface Catalog {
  backMatter?: BackMatter;
  controls?: Control[];
  groups?: ControlGroup[];
  metadata: PublicationMetadata;
  params?: Parameter[];
}

export interface Profile {
  backMatter?: BackMatter;
  imports: ImportResource[];
  merge?: MergeControls;
  metadata: PublicationMetadata;
  modify?: ModifyControls;
  uuid: string;
}
```
Planned Features & Enhancements:

Ongoing:

- Fix Bugs
- Make the application faster (or at least make it feel faster)
- Optimize browser memory footprint, where possible
- Enhance usability and add user-safety protections.

Metadata, Controls, Prose, and Parameter Selection and Editing:

- Find acceptable plugin for markdown editing and rendering
- Regroup Controls Page: Add drag-n-drop for UI simplification
- Add possibility to start with an empty Cat and build it up from scratch
- Add profile resolution using either Saxon and JSON->XML conversion or do it in JSON
- Add possibility for user to upload their own catalogs with auto-validation by the schema
- Add possibility for user to upload own profiles with auto-validation by the schema
- Add possibility for user to upload own profiles, validate them and resolve, resulting in catalog
Questions?

Contact us at: oscal@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

Thank you!