DevSecComp(liance)Ops with OSCAL

March 2022

Ray Gauss, Director of Innovation, Easy Dynamics
Who is Easy Dynamics?

Information Sharing
- B2B and B2C Information Sharing
- Consent based and full audit traceability
- Migrations and assessments
- Digital identity and governance
- Knowledge management

Cloud Delivery
- Migration readiness, analysis, & planning
- Enterprise cloud operations
- Software defined networking
- DevSecOps and CI/CD as a service
- FISMA/FedRAMP accreditation

Cyber/Identity/Privacy
- Identity and access management
- Privacy engineering and policy development
- Compliance framework enforcement
- Asset, ISVM, Anti-virus, SOC/NOC Integration
- Risk Management and Fraud Detection

HISTORY OF INNOVATION
- 2020 Public Sector Innovation of the Year
- 2018 Public Sector Innovation of the Year

STANDARDS ALIGNMENT
- NIST SP800-63 / NIST SP800-53
- NIST Privacy Framework
- NIST OSCAL
- Cybersecurity profile for payroll industry

ACCREDITATION EXPERIENCE
- First FISMA High
- Cloud Factory 2 General Support System (GSS) FISMA High
- FedRAMP Accreditation
- Early Adopter OSCAL Submission
Where We Typically Fit in the Assessment Process
(FedRAMP ATO Example)
Challenges We Often See

- **Compliance Framework**
  - **NIST RMF**
  - ISO 27001
  - AICPA SOC

1. **Categorize**
2. **Select**
3. **Implement**
4. **Assess**
5. **Authorize**
6. **Monitor**

- **Build System**
  - Features & Capabilities

- **Security Audit**
  - Evidence and Verification

- **Maintain**
  - Oversight

- **Dev/Eng**

  - **Lacks** Informative Security Guidance
  - **Lacks** Guidance on How to Leverage Shared Services Authorizations
  - **Lacks** Organizational Parameter Guidance
  - **Lacks** Incremental Methods for Reviews and Evidence Collection Mostly Performed During Assessment Phase
  - **Lacks** Ability to Continuously Authorize Only the Changes in System Increments
  - **Lacks** Operating Models that Support Desired Monitoring Rigor

Creates a Significant Burden of Responsibility, Awareness, and Understanding of Broad Cybersecurity Concepts
Traditional Assessment Workflow Has Impediments

- Compliance typically documented after development is complete
- Engineers don’t typically have access to GRC platform

Dev/Eng
Dev Environment

Test Environment

ISSO/ISSM
Inherited SSPs

Assessors
Assessment/Evidence

Authorizing Body Approval

Production Environment
How We’ve Addressed These Challenges in the Past

1. Select Applicable Control Baselines
   - NIST 800-53 High
   - NIST 800-53 Moderate
   - DoD SRG 4
   - DoD SRG 5
   - Customer Specific Controls

2. Load Selected Controls into an Issue Tracking Project as ATO EPICs
   (Each control is a separate task)

3. Project team and ISSO team iteratively and parallel document and approve each control - instead of waiting for the whole control to be documented and reviewed.

4. Revise controls and artifacts as necessary

5. Authorizing Official (AO) Reviews, Approves, and Issues ATO
   - AT0 Letter
   - POAMS

6. GRC Platform performs continuous monitoring of ATO Environment

[Diagram of the process steps]
The Shift-Left Workflow That OSCAL and Tools Enable

- Dev/Eng
- Assessors
- ISSO/ISSM
- SSP
- POA&Ms
- Assessment/Evidence
- Continuous Authorization
- Production Environment
- Authorizing Body Approval

- Cloud Components
- Reusable Components & Profiles
- Agency Components & Profiles

- Informative Leveraged Authorization Guidance
- Informative Security Guidance
- Organizational Parameters Guidance

- Continuous Authorization
How We See This Space Evolving
## What We’ve Been Working On

### Contributions To Date

<table>
<thead>
<tr>
<th>OSCAL React Component Library and Web App</th>
<th>OSCAL REST API (OpenAPI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub Repository</td>
<td>GitHub Repository</td>
</tr>
<tr>
<td>Storybook Docs</td>
<td>GitHub Repository</td>
</tr>
<tr>
<td><a href="https://docs.oscal-react-library.msd.easydynamics.com">https://docs.oscal-react-library.msd.easydynamics.com</a></td>
<td><a href="https://editor.swagger.io/?url=...">https://editor.swagger.io/?url=...</a></td>
</tr>
<tr>
<td>Web App Demo</td>
<td>GitHub Repository</td>
</tr>
</tbody>
</table>

### What’s New!

<table>
<thead>
<tr>
<th>OSCAL REST Service</th>
<th>OSCAL Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>★ Implementation of some of the OSCAL REST API</td>
<td>★ Docker Container with OSCAL REST Service and Web App</td>
</tr>
<tr>
<td>★ Persists against a file store (such as a Git repository)</td>
<td></td>
</tr>
</tbody>
</table>
OSCAL Editor
High-Level Architecture
Developer/Security Engineer Workflow

- Dev/Eng
  - Assessors
  - ISSO/ISSM

- Dev Environment
  - Test Environment
  - Assessment/Evidence
  - SSP
  - PO&Ms

- Feature Task (DoD includes Compliance Docs)

- Dev/Eng

- Product Backlog

- Git Repository
  - Feature Code
  - OSCAL Code

- ISSO/ISSM

- Other Devs/Eng

- Request Changes/Approve

- Dev/Test Environment

- Clone/Pull Repo
  - Incremental Updates to SSP
  - Commit/Push Changes
  - Create Pull Request

- OSCAL Editor

- Reusable Components
Demo
Demo

- Dev/Eng
- Browses OSCAL Content Repository on GitHub
- Copies Clone URL
Demo

- Dev/Eng
  - Clones OSCAL Content GitHub Repo
  - Pulls Docker Container
  - Runs Docker Container Pointing to Local Content Repo
Demo

- Dev/Eng
- Uses OSCAL Editor to Update SSP Title
Demo

- Dev/Eng
  - Creates Git Branch
  - Commits Change
  - Pushes New Branch
Demo

- Dev/Eng
- Creates PR on GitHub
Demo

- ISSO/ISSM
- Review PR on GitHub
What’s Next?

• REST API Improvements
  • With the OSCAL Community’s Help

• React Component Library & Web App Improvements
  • Expanded Edit Capabilities
  • UI/UX Improvements

• OSCAL Standards
  • Registry Specification
    • Component Definitions In Particular

• CI/CD Integration
  • Validation
  • Attachment of Evidence

• IDE Extensions
Questions / More Info

- [https://easydynamics.com/oscal](https://easydynamics.com/oscal)
- [https://github.com/EasyDynamics](https://github.com/EasyDynamics)
  - oscal-react-library
  - oscal-rest
  - oscal-rest-service
  - oscal-editor-deployment