Virtual Room #1

Hosted By: **Nikita Wootten**, Computer Scientist, *NIST*

(OSCAL Webpage)

Disclaimer: Portions of the event may be recorded, and audience Q&A or comments may be captured. The recorded event may be edited and rebroadcasted or otherwise made publicly available by NIST. By attending this event, you acknowledge and consent to having your conversation recorded.

oscal2022@nist.gov
conferences@nist.gov

OSCAL "Deep Diff“

- a model-agnostic OSCAL tool and the concept behind it -

National Institute of Standards and Technology
U.S. Department of Commerce

ITL/CSD/OSCAL Team

3rd OSCAL Workshop

The Problem: Large Documents are Difficult to Digest

Authors

How do I track changes that my team has made between revisions of a document?

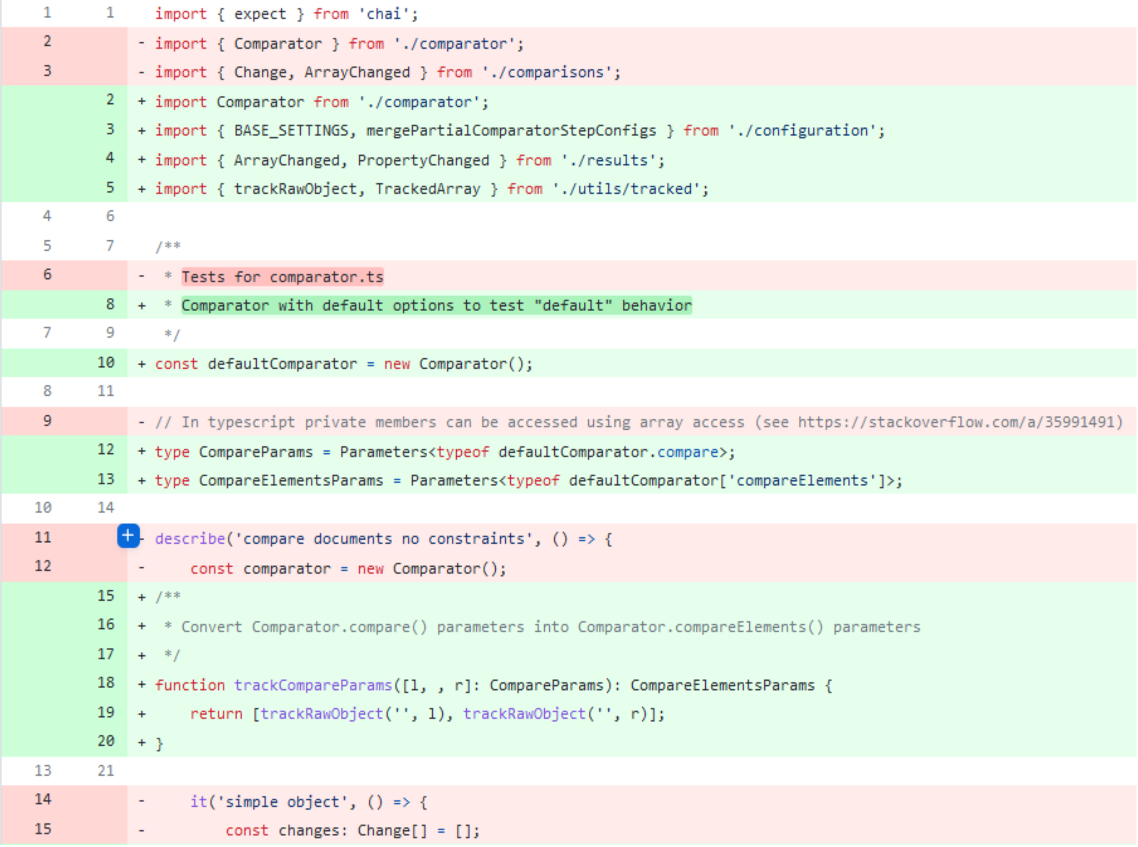
Catalog Consumers

How can I produce a checklist of controls with relevant changes when a new revision of a control catalog comes out?

Developers

How can I track when certain types of changes to a document happens, and make decisions based on those change-lists (such as in a CI/CD pipeline)?

The Solution: A “Diff” Tool for OSCAL Documents

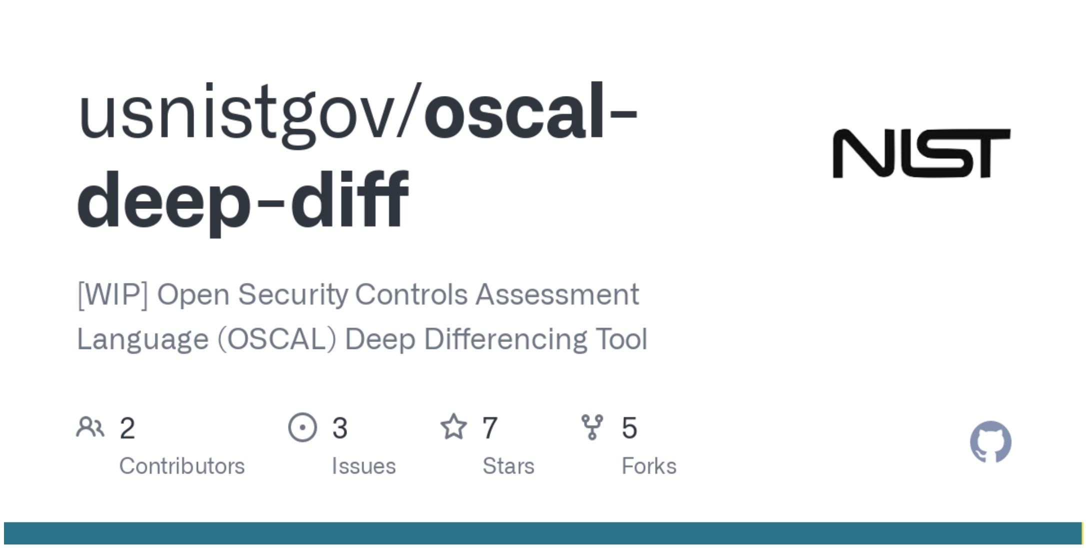


```
  import { expect } from 'chai';
- import { Comparator } from './comparator';
- import { Change, ArrayChanged } from './comparisons';
+ import Comparator from './comparator';
+ import { BASE_SETTINGS, mergePartialComparatorStepConfigs } from './configuration';
+ import { ArrayChanged, PropertyChanged } from './results';
+ import { trackRawObject, TrackedArray } from './utils/tracked';

  /**
-  * Tests for comparator.ts
+  * Comparator with default options to test "default" behavior
   */
+ const defaultComparator = new Comparator();

- // In typescript private members can be accessed using array access (see https://stackoverflow.com/a/35991491)
+ type CompareParams = Parameters<typeof defaultComparator.compare>;
+ type CompareElementsParams = Parameters<typeof defaultComparator['compareElements']>;

- describe('compare documents no constraints', () => {
-     const comparator = new Comparator();
+ /**
+  * Convert Comparator.compare() parameters into Comparator.compareElements() parameters
+  */
+ function trackCompareParams([l, , r]: CompareParams): CompareElementsParams {
+     return [trackRawObject('', l), trackRawObject('', r)];
+ }

-     it('simple object', () => {
-         const changes: Change[] = [];
```

GitHub’s diff view, an example of a diff tool used daily by developers

- A tool that can generate a comparison between two OSCAL documents
- Configurable enough to be applied in multiple scenarios
- Must be able to generate output documents that are easy to digest and share
- Portable and extendable so that it can be integrated into other tools (such as web applications)

OSCAL Deep Diff Introduction



usnistgov/**oscal-deep-diff**

[WIP] Open Security Controls Assessment Language (OSCAL) Deep Differencing Tool

2 Contributors

3 Issues

7 Stars

5 Forks

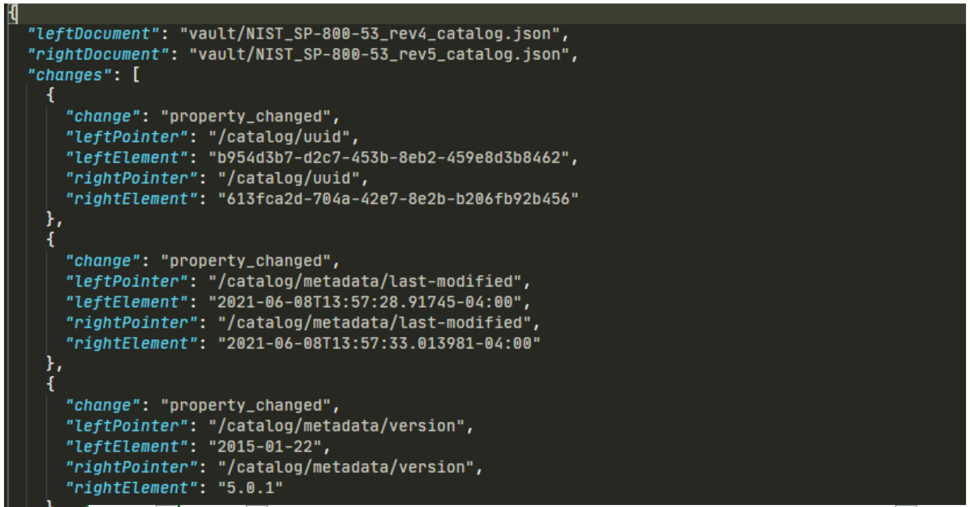
OSCAL-deep-diff GitHub card

- **An open-source JavaScript/TypeScript CLI application and library that can be used to compare arbitrary JSON documents**
- **Does not rely on a schema to compare objects, can be configured to compare documents in a reproducible manner**
- **Generates outputs in multiple formats including easy-to-distribute Excel spreadsheets**
- **Can be integrated into other tools, including web and desktop applications**

Scenario: Comparing two SSPs

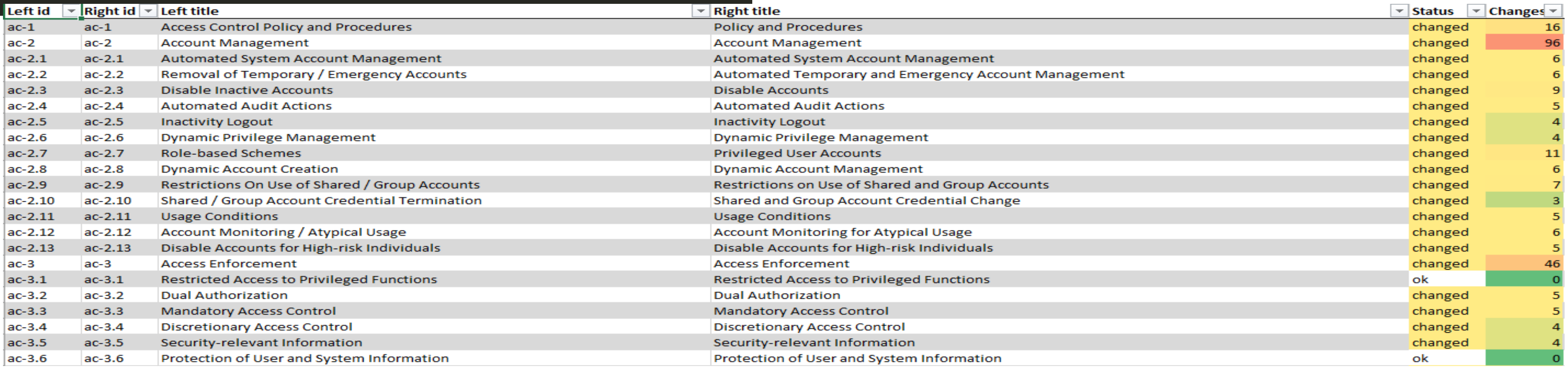
National Institute of
Standards and Technology
U.S. Department of Commerce

Output Format



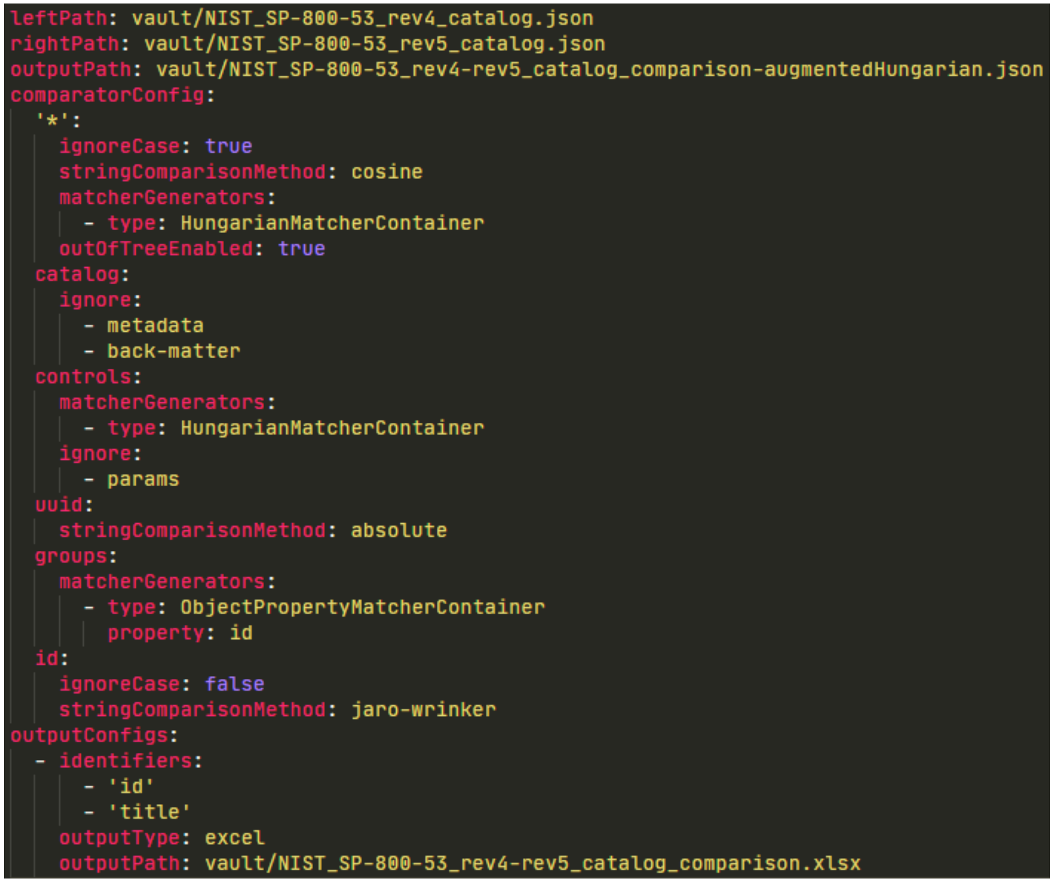
```
{
  "leftDocument": "vault/NIST_SP-800-53_rev4_catalog.json",
  "rightDocument": "vault/NIST_SP-800-53_rev5_catalog.json",
  "changes": [
    {
      "change": "property_changed",
      "leftPointer": "/catalog/uuid",
      "leftElement": "b954d3b7-d2c7-453b-8eb2-459e8d3b8462",
      "rightPointer": "/catalog/uuid",
      "rightElement": "613fca2d-704a-42e7-8e2b-b206fb92b456"
    },
    {
      "change": "property_changed",
      "leftPointer": "/catalog/metadata/last-modified",
      "leftElement": "2021-06-08T13:57:28.91745-04:00",
      "rightPointer": "/catalog/metadata/last-modified",
      "rightElement": "2021-06-08T13:57:33.013981-04:00"
    },
    {
      "change": "property_changed",
      "leftPointer": "/catalog/metadata/version",
      "leftElement": "2015-01-22",
      "rightPointer": "/catalog/metadata/version",
      "rightElement": "5.0.1"
    },
```

- **By default, OSCAL deep diff produces a JSON document listing the differences between the two documents**
  - Valid change types are “property_left_only”, “property_right_only”, “property_changed”, and “array_changed”.
  - Each “array_changed” type has a sub-list of changes for each matched pair of items.
- **The raw JSON document can be used to produce friendlier output documents**
  - Excel output collects all of one object type (like controls) and displays them in an Excel document.
  - The tool can be extended to produce other comparison views (such as a web-application or pdf report)



Left id	Right id	Left title	Right title	Status	Changes
ac-1	ac-1	Access Control Policy and Procedures	Policy and Procedures	changed	16
ac-2	ac-2	Account Management	Account Management	changed	96
ac-2.1	ac-2.1	Automated System Account Management	Automated System Account Management	changed	6
ac-2.2	ac-2.2	Removal of Temporary / Emergency Accounts	Automated Temporary and Emergency Account Management	changed	6
ac-2.3	ac-2.3	Disable Inactive Accounts	Disable Accounts	changed	9
ac-2.4	ac-2.4	Automated Audit Actions	Automated Audit Actions	changed	5
ac-2.5	ac-2.5	Inactivity Logout	Inactivity Logout	changed	4
ac-2.6	ac-2.6	Dynamic Privilege Management	Dynamic Privilege Management	changed	4
ac-2.7	ac-2.7	Role-based Schemes	Privileged User Accounts	changed	11
ac-2.8	ac-2.8	Dynamic Account Creation	Dynamic Account Management	changed	6
ac-2.9	ac-2.9	Restrictions On Use of Shared / Group Accounts	Restrictions on Use of Shared and Group Accounts	changed	7
ac-2.10	ac-2.10	Shared / Group Account Credential Termination	Shared and Group Account Credential Change	changed	3
ac-2.11	ac-2.11	Usage Conditions	Usage Conditions	changed	5
ac-2.12	ac-2.12	Account Monitoring / Atypical Usage	Account Monitoring for Atypical Usage	changed	6
ac-2.13	ac-2.13	Disable Accounts for High-risk Individuals	Disable Accounts for High-risk Individuals	changed	5
ac-3	ac-3	Access Enforcement	Access Enforcement	changed	46
ac-3.1	ac-3.1	Restricted Access to Privileged Functions	Restricted Access to Privileged Functions	ok	0
ac-3.2	ac-3.2	Dual Authorization	Dual Authorization	changed	5
ac-3.3	ac-3.3	Mandatory Access Control	Mandatory Access Control	changed	5
ac-3.4	ac-3.4	Discretionary Access Control	Discretionary Access Control	changed	4
ac-3.5	ac-3.5	Security-relevant Information	Security-relevant Information	changed	4
ac-3.6	ac-3.6	Protection of User and System Information	Protection of User and System Information	ok	0

Configurability



```
leftPath: vault/NIST_SP-800-53_rev4_catalog.json
rightPath: vault/NIST_SP-800-53_rev5_catalog.json
outputPath: vault/NIST_SP-800-53_rev4-rev5_catalog_comparison-augmentedHungarian.json
comparatorConfig:
  '*':
    ignoreCase: true
    stringComparisonMethod: cosine
    matcherGenerators:
      - type: HungarianMatcherContainer
    outOfTreeEnabled: true
  catalog:
    ignore:
      - metadata
      - back-matter
  controls:
    matcherGenerators:
      - type: HungarianMatcherContainer
    ignore:
      - params
  uuid:
    stringComparisonMethod: absolute
  groups:
    matcherGenerators:
      - type: ObjectPropertyMatcherContainer
        property: id
  id:
    ignoreCase: false
    stringComparisonMethod: jaro-wrinker
outputConfigs:
  - identifiers:
      - 'id'
      - 'title'
    outputType: excel
    outputPath: vault/NIST_SP-800-53_rev4-rev5_catalog_comparison.xlsx
```

An example configuration file for comparing control catalogs

The tool can be configured to change the behavior of the comparison:

- **Ignore objects that are irrelevant to the comparison**
- **Change the way properties are compared (select a string similarity algorithm, ignore case, etc.)**
- **Swap out the algorithms used to "match" array items to each other**

…as well as the output format:

- **Change which objects will be collected for the comparison**
- **Choose which metadata should be displayed in the output document**
- **Output to JSON, Excel, etc.**

This is all configured via a YAML file

Scenario: Comparing Component Definitions

National Institute of Standards and Technology
U.S. Department of Commerce

Scenario: Comparing SP 800-53 Revisions

National Institute of Standards and Technology
U.S. Department of Commerce

Shortcomings

- **Speed of comparisons**
  - Array comparison algorithms are computationally expensive.
  - For example, depending on the settings used, comparisons between SP 800-53 revisions can take upwards of 10 minutes.
- **Comparison behavior tuning**
  - Getting the tool fit a particular comparison scenario may require tweaking.
  - This can be solved with community support and examples.
- **Comparison results**
  - Some scenarios are not supported yet, such as object demotion/promotion. (ex. A control becoming an enhancement)

Call to Action

If this tool is exciting or potentially useful to you:

- **Please provide feedback, report bugs, and suggest improvements!**
- **Feel free to submit issues, PRs, and discussions to https://github.com/usnistgov/oscal-deep-diff**

Please note: The version of OSCAL Deep Diff shown here is still experimental, see https://github.com/usnistgov/oscal-deep-diff/pull/34

Questions?