C2 LABS
TAKE BACK CONTROL

Cyber Security Controls: Data Portability between vendor tools using NIST OSCAL

J. Travis Howerton
Chief Technology Officer (CTO)
C2 Labs
https://www.c2labs.com
https://atlasity.io

WHO WE ARE

C2 LABS
TAKE BACK CONTROL

Who We Are

About Us

- **Minority-Owned DC small business:** Founded in 2014, with a diverse workforce & strong nationwide commercial and government past performance
- **Mission:** To serve as a security focused and agile digital transformation partner that blends *Art* and *Science* to enable our customers to expand their vision, drive cultural change, and avoid being left behind
- **Certifications and Awards:** PMP, ITIL, CISSP, DAWIA, AWS, Agile SCRUM Masters, Fed100 Award, ACT-IAC Excellence.gov overall award winner (Most innovative project in Government)

Key Cultural Values

I nnovative

D riven

E thical

A gile

S ervice

> “Logic will get you from A to B. Imagination will take you everywhere”
>
> **– Albert Einstein**

Oak Ridge National Laboratory

Entergy

United States Environmental Protection Agency

Department of Energy United States of America

Federal Energy Regulatory Commission

USAID From the American People

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

Juniper Networks

Bio – J. Travis Howerton, C2 Labs Chief Technology Officer

- Travis Howerton is the co-founder and Chief Technology Officer (CTO) of C2 Labs and has previously held positions as the National Nuclear Security Administration CTO, Deputy CIO at Oak Ridge National Laboratory, and the Global Director for Strategic Programs with Bechtel Corporation.
- Howerton holds a B.S. in Organizational Management from Tusculum College, a M.S. in Computer Information Systems from Boston University, and holds multiple certifications to include the CISSP, ITIL, PMP, Scrum Master, Harvard Credential of Readiness, and AWS Certified Developer.
- Howerton is a native of Oak Ridge, TN where he lives with his wife (Beth) and two daughters (Taylor and Sarah Beth).

Abstract

Today, System Security Plans (SSPs) are usually generated in Word or Excel documents using unstructured formats that make them difficult to process in an automated way or to port the information across tools due to the wide variability in formats. In this session, we will discuss how NIST’s OSCAL standard can enable cyber security control data portability, moving cyber security risk and assessment information across different vendor tools using the OSCAL format. By leveraging OSCAL, we will demonstrate:

-The ability to load all OSCAL Catalogs and Baselines of NIST SP 800-53 Rev 4, 5, and FedRAMP Baselines (Low, Moderate, High, and Privacy) into the C2 Labs Atlasity tool

-The ability to load an OSCAL version of an SSP from GovReady (another vendor tool) into Atlasity

-The ability to cross-walk controls from the GovReady SSP against the NIST SP 800-53 FedRAMP Moderate Baseline to load programmatically into Atlasity

-The ability to support leveraged authorizations within Atlasity in support of the OSCAL standard

Hypotheses

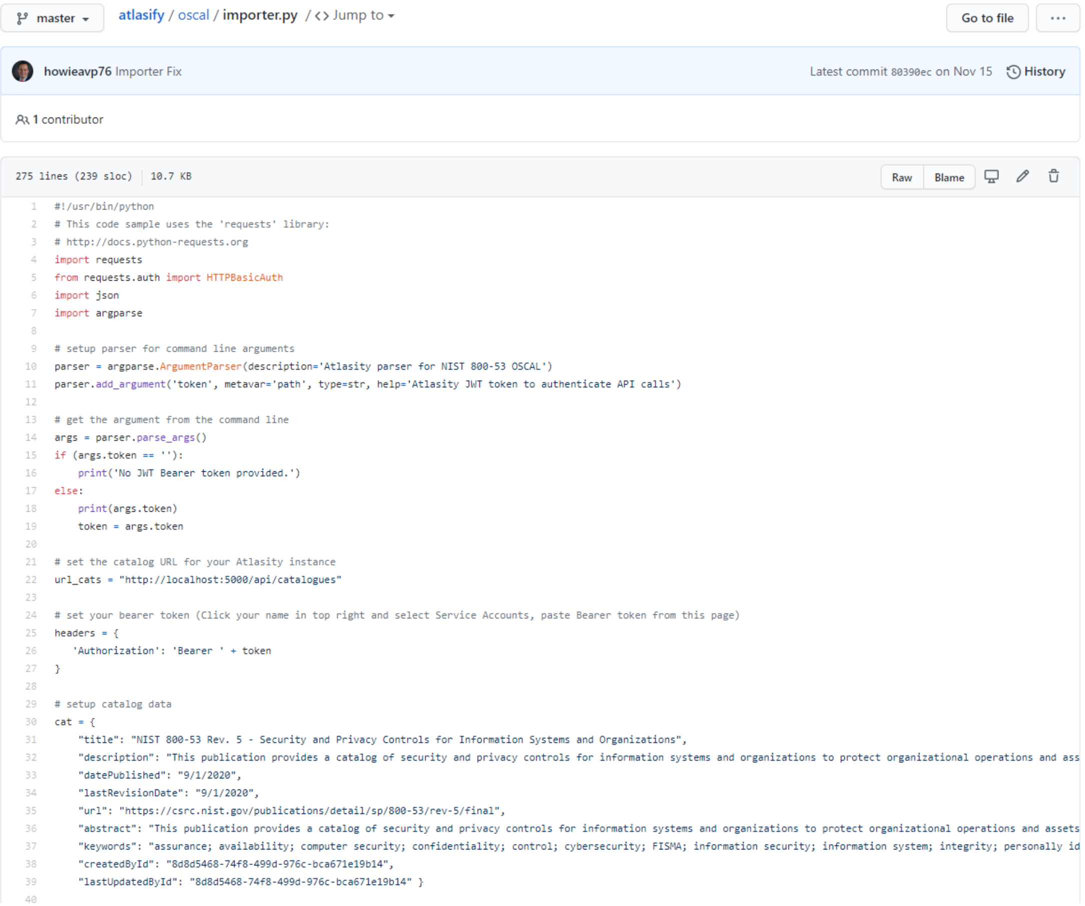
This proposal sought to test two hypotheses of the effectiveness of using OSCAL to programmatically load content. These include:

- OSCAL can be used to efficiently load NIST controls to allow the rapid creation of security plans
  - EXAMPLE BENEFIT: Improved quality and time to value by automating the creation of SSPs

- OSCAL can be used to efficiently transfer SSP content programmatically between tools
  - EXAMPLE BENEFIT: Cloud Service Providers could submit SSP content in an automated manner to perform compliance checks with less manual labor and resulting costs (i.e. to FedRAMP)

Experiment 1 – Loading Catalogs and Creating a SSP

Python Script for Loading (271 lines of code)



master
atlasify / oscal / importer.py / <> Jump to
Go to file
···
howieavp76 Importer Fix
Latest commit 80390ec on Nov 15
History
1 contributor
275 lines (239 sloc) | 10.7 KB
Raw
Blame

```python
#!/usr/bin/python
# This code sample uses the 'requests' library:
# http://docs.python-requests.org
import requests
from requests.auth import HTTPBasicAuth
import json
import argparse

# setup parser for command line arguments
parser = argparse.ArgumentParser(description='Atlasity parser for NIST 800-53 OSCAL')
parser.add_argument('token', metavar='path', type=str, help='Atlasity JWT token to authenticate API calls')

# get the argument from the command line
args = parser.parse_args()
if (args.token == ''):
    print('No JWT Bearer token provided.')
else:
    print(args.token)
    token = args.token

# set the catalog URL for your Atlasity instance
url_cats = "http://localhost:5000/api/catalogues"

# set your bearer token (Click your name in top right and select Service Accounts, paste Bearer token from this page)
headers = {
   'Authorization': 'Bearer ' + token
}

# setup catalog data
cat = {
    "title": "NIST 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations",
    "description": "This publication provides a catalog of security and privacy controls for information systems and organizations to protect organizational operations and ass
    "datePublished": "9/1/2020",
    "lastRevisionDate": "9/1/2020",
    "url": "https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final",
    "abstract": "This publication provides a catalog of security and privacy controls for information systems and organizations to protect organizational operations and assets
    "keywords": "assurance; availability; computer security; confidentiality; control; cybersecurity; FISMA; information security; information system; integrity; personally id
    "createdById": "8d8d5468-74f8-499d-976c-bca671e19b14",
    "lastUpdatedById": "8d8d5468-74f8-499d-976c-bca671e19b14" }

```

- C2 Labs downloaded the latest NIST 800-53 and FedRAMP Baselines from the OSCAL GitHub site
- Developed an open-source Python script to parse the OSCAL baseline JSON files, enriched with other data, and bulk uploaded them as catalogs via REST APIs in Atlasity (also published interim artifacts in JSON)
- Developed many Atlasity profiles based on NIST 800-53 Rev. 4, Rev. 5, and FedRAMP
- Used our SSP wizard to create a security plan template in less than 5 minutes after import
- Source code open-sourced @Atlasify

EVIDENCE ON GITHUB:

https://github.com/C2-Labs/atlasify/tree/master/oscal

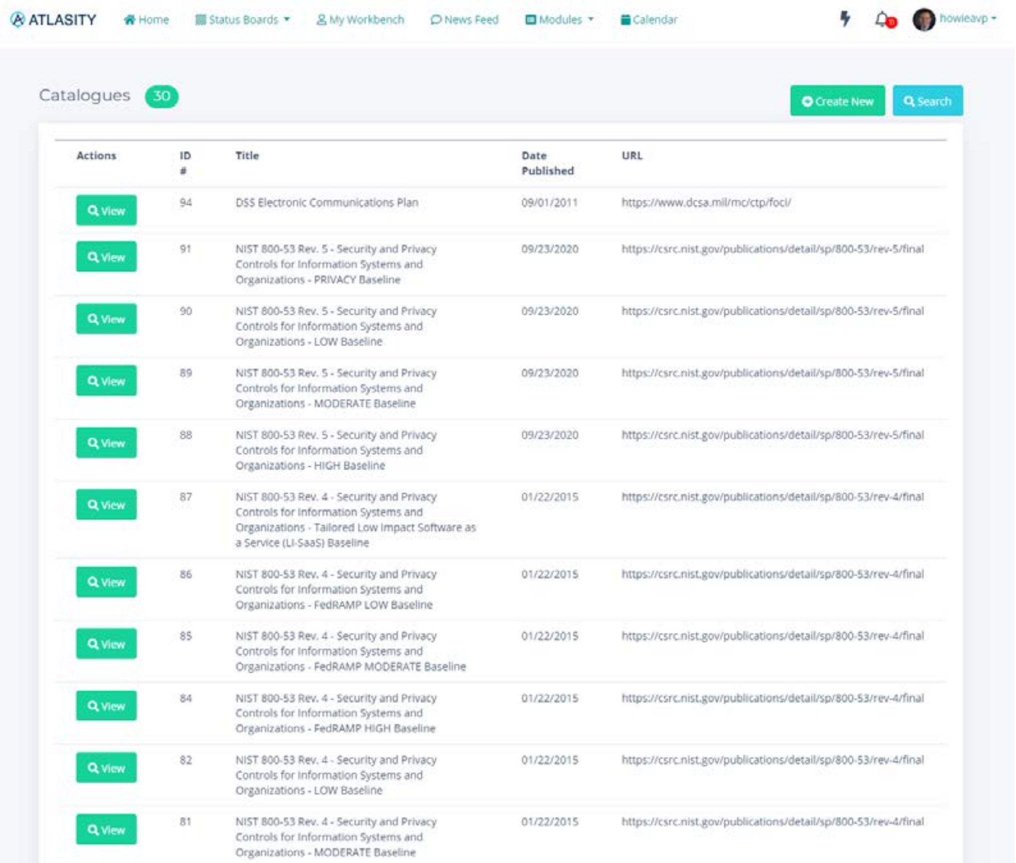
Level of Effort: ~ 15 hours

README: contains detailed process and results info

Experiment 1 – Results

SUCCESS

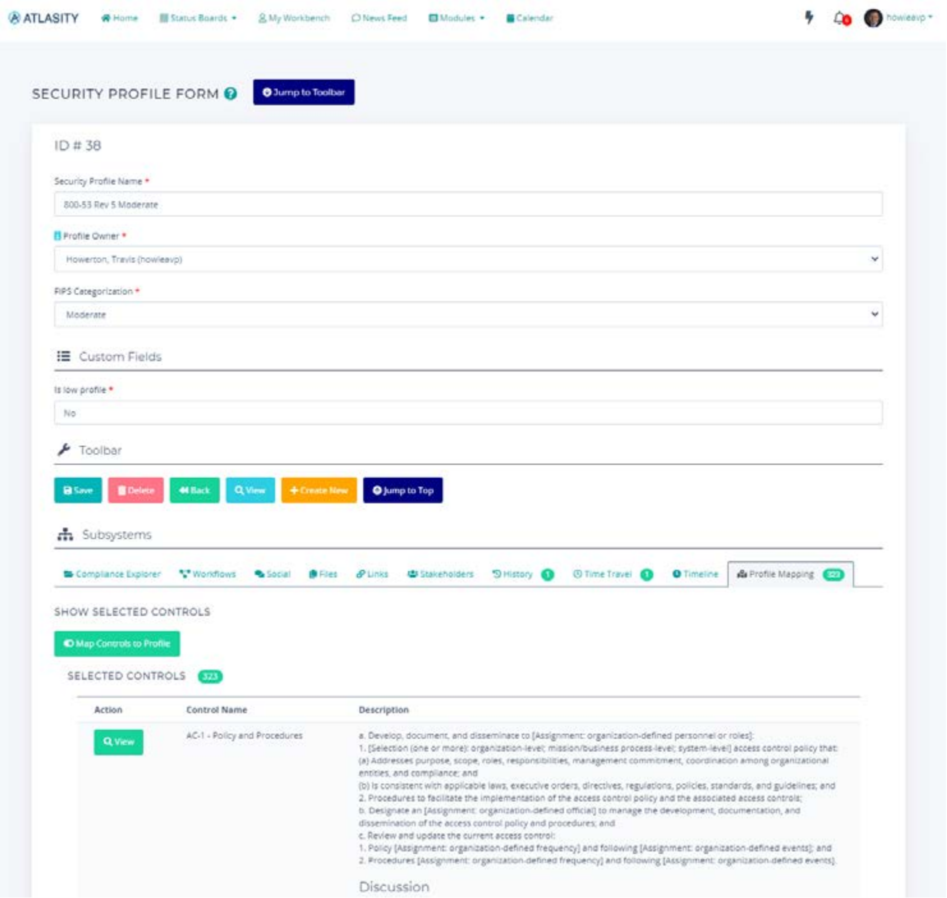
Catalogs Loaded



Catalogues 30

Actions	ID #	Title	Date Published	URL
View	94	DSS Electronic Communications Plan	09/01/2011	https://www.dcsa.mil/mc/ctp/foci/
View	91	NIST 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations - PRIVACY Baseline	09/23/2020	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
View	90	NIST 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations - LOW Baseline	09/23/2020	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
View	89	NIST 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations - MODERATE Baseline	09/23/2020	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
View	88	NIST 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations - HIGH Baseline	09/23/2020	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
View	87	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - Tailored Low Impact Software as a Service (LI-SaaS) Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
View	86	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - FedRAMP LOW Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
View	85	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - FedRAMP MODERATE Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
View	84	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - FedRAMP HIGH Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
View	82	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - LOW Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
View	81	NIST 800-53 Rev. 4 - Security and Privacy Controls for Information Systems and Organizations - MODERATE Baseline	01/22/2015	https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final

Profiles Created



SECURITY PROFILE FORM

ID # 38

Security Profile Name *
800-53 Rev 5 Moderate

Profile Owner *
Howerton, Travis (howieavp)

FIPS Categorization *
Moderate

Custom Fields

Is low profile *
No

Toolbar

Subsystems

SHOW SELECTED CONTROLS

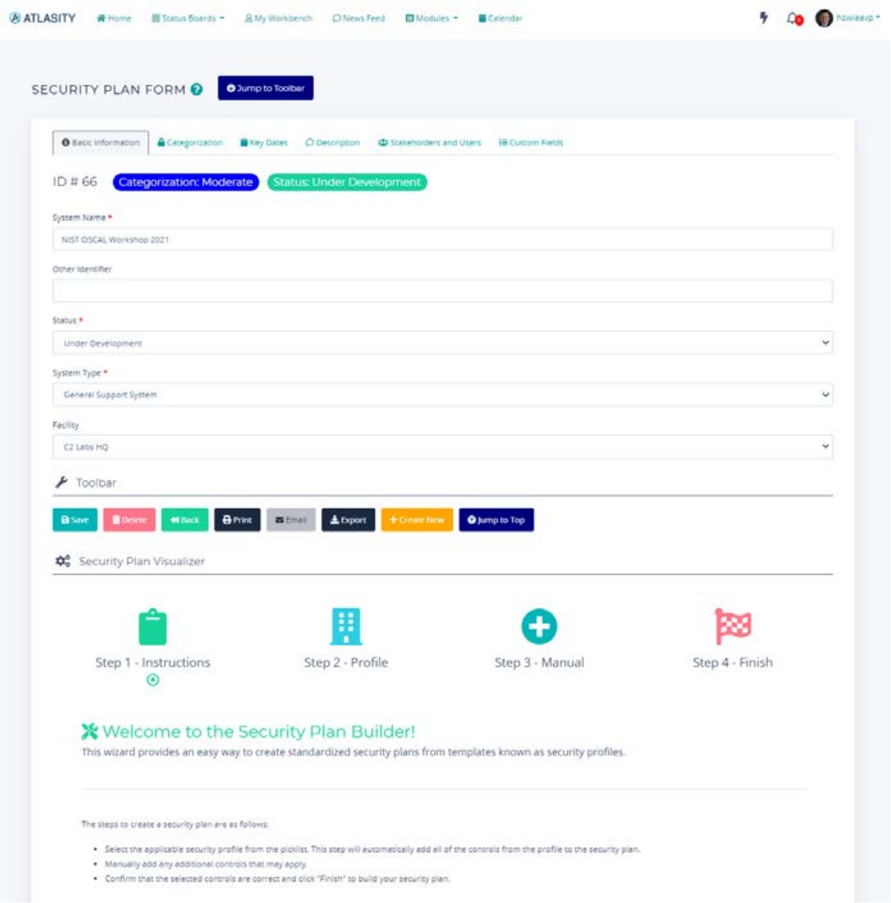
SELECTED CONTROLS 323

Action	Control Name	Description
View	AC-1 - Policy and Procedures	a. Develop, document, and disseminate to [Assignment: organization-defined personnel or roles]: 1. [Selection (one or more): organization-level; mission/business process-level; system-level] access control policy that: (a) Addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and (b) Is consistent with applicable laws, executive orders, directives, regulations, policies, standards, and guidelines; and 2. Procedures to facilitate the implementation of the access control policy and the associated access controls; b. Designate an [Assignment: organization-defined official] to manage the development, documentation, and dissemination of the access control policy and procedures; and c. Review and update the current access control: 1. Policy [Assignment: organization-defined frequency] and following [Assignment: organization-defined events]; and 2. Procedures [Assignment: organization-defined frequency] and following [Assignment: organization-defined events]. Discussion

Experiment 1 – Results

SUCCESS

SSP Creation



SECURITY PLAN FORM

ID # 66 Categorization: Moderate Status: Under Development

System Name *: NIST OSCAL Workshop 2021

Other Identifier:

Status *: Under Development

System Type *: General Support System

Facility: C2 Labs HQ

Toolbar

Security Plan Visualizer

Step 1 - Instructions

Step 2 - Profile

Step 3 - Manual

Step 4 - Finish

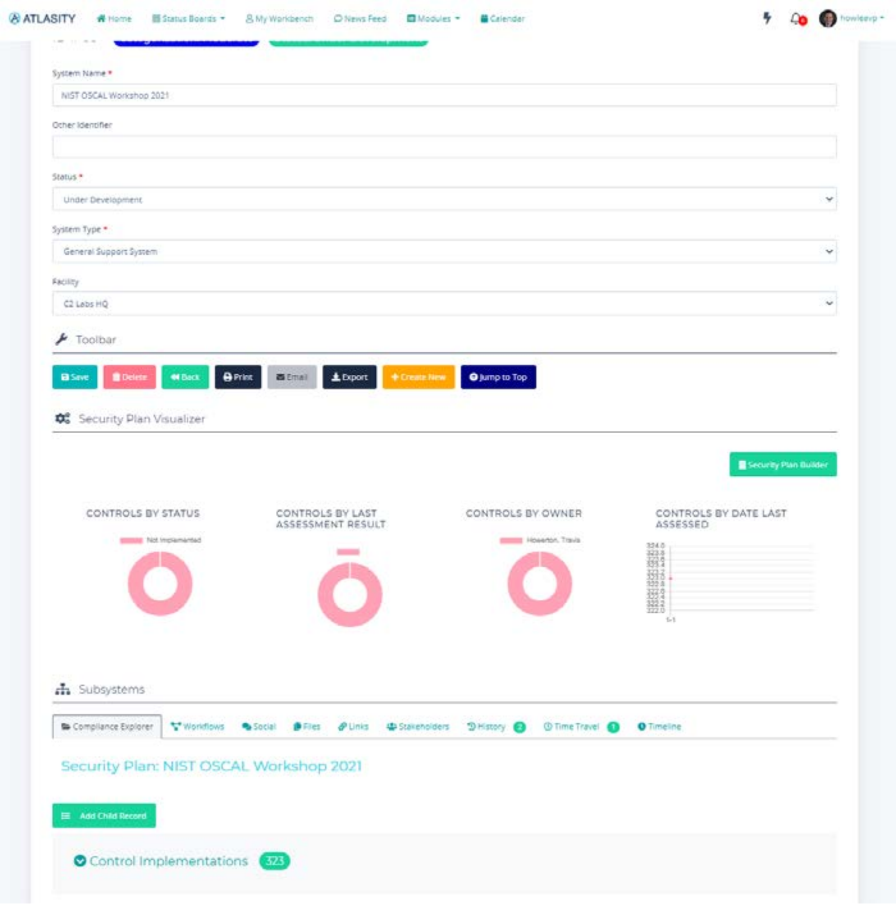
Welcome to the Security Plan Builder!

This wizard provides an easy way to create standardized security plans from templates known as security profiles.

The steps to create a security plan are as follows:

- Select the applicable security profile from the picklist. This step will automatically add all of the controls from the profile to the security plan.
- Manually add any additional controls that may apply.
- Confirm that the selected controls are correct and click "Finish" to build your security plan.

SSP Visualization



System Name *: NIST OSCAL Workshop 2021

Other Identifier:

Status *: Under Development

System Type *: General Support System

Facility: C2 Labs HQ

Toolbar

Security Plan Visualizer

CONTROLS BY STATUS

CONTROLS BY LAST ASSESSMENT RESULT

CONTROLS BY OWNER

CONTROLS BY DATE LAST ASSESSED

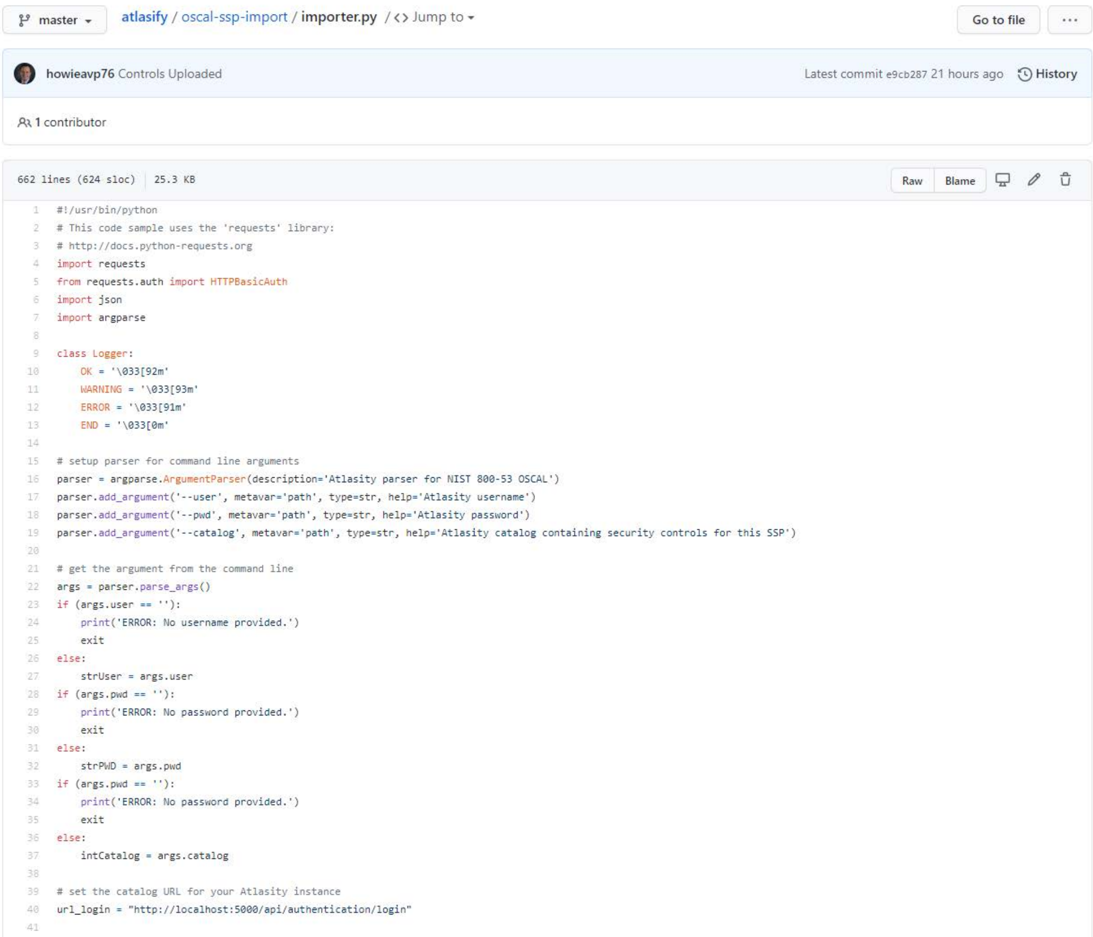
Subsystems

Security Plan: NIST OSCAL Workshop 2021

Control Implementations 323

Experiment 2 – Loading OSCAL SSP from GovReady

Python Script for Loading (662 lines of code)


master
atlasify / oscal-ssp-import / importer.py / <> Jump to
Go to file
···
howieavp76 Controls Uploaded
Latest commit e9cb287 21 hours ago
History
1 contributor
662 lines (624 sloc) | 25.3 KB
Raw
Blame

```python
#!/usr/bin/python
# This code sample uses the 'requests' library:
# http://docs.python-requests.org
import requests
from requests.auth import HTTPBasicAuth
import json
import argparse

class Logger:
    OK = '\033[92m'
    WARNING = '\033[93m'
    ERROR = '\033[91m'
    END = '\033[0m'

# setup parser for command line arguments
parser = argparse.ArgumentParser(description='Atlasity parser for NIST 800-53 OSCAL')
parser.add_argument('--user', metavar='path', type=str, help='Atlasity username')
parser.add_argument('--pwd', metavar='path', type=str, help='Atlasity password')
parser.add_argument('--catalog', metavar='path', type=str, help='Atlasity catalog containing security controls for this SSP')

# get the argument from the command line
args = parser.parse_args()
if (args.user == ''):
    print('ERROR: No username provided.')
    exit
else:
    strUser = args.user
if (args.pwd == ''):
    print('ERROR: No password provided.')
    exit
else:
    strPWD = args.pwd
if (args.pwd == ''):
    print('ERROR: No password provided.')
    exit
else:
    intCatalog = args.catalog

# set the catalog URL for your Atlasity instance
url_login = "http://localhost:5000/api/authentication/login"

```

- The GovReady team provided an example SSP from their tool in OSCAL format for C2 Labs to process
- Developed an open-source Python script to parse the OSCAL SSP JSON file to create the SSP and Control Implementations in Atlasity
- Mapped schema differences between GovReady and Atlasity with OSCAL
- Loaded data programmatically via Atlasity REST APIs
- Source code open-sourced @Atlasify

EVIDENCE ON GITHUB:

https://github.com/C2-Labs/atlasify/tree/master/oscal-ssp-import

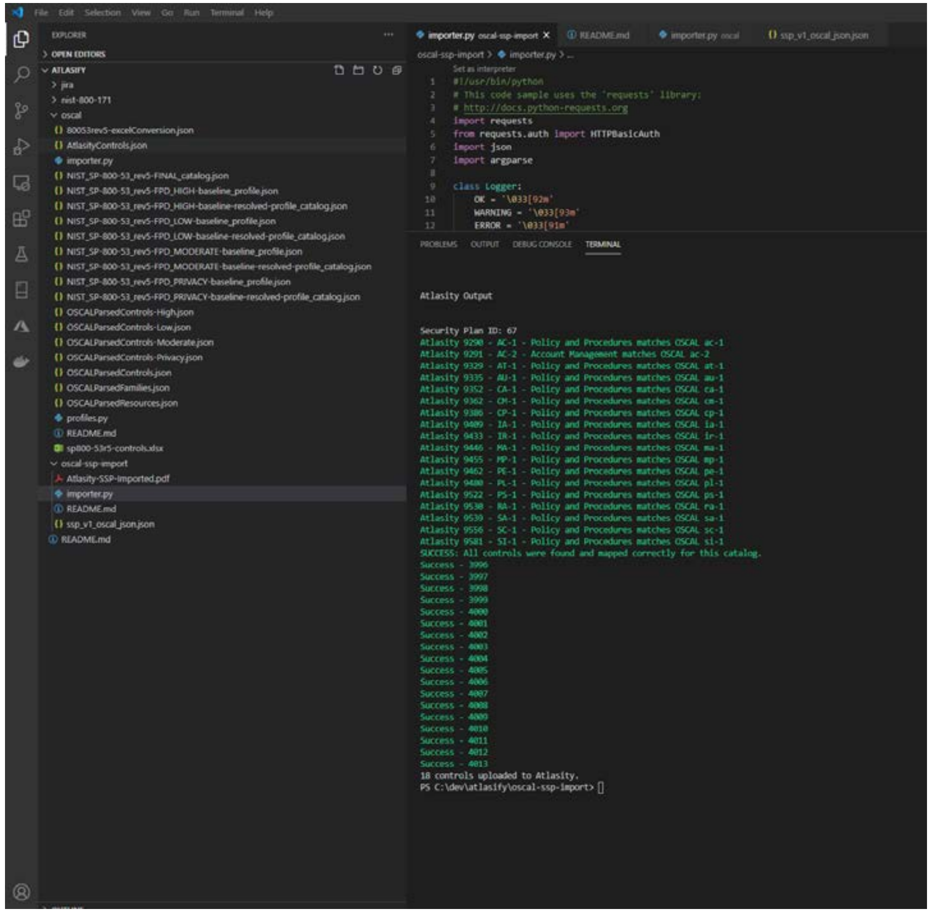
Level of Effort: ~ 30 hours

README: contains detailed process and results info

Experiment 2 – Results

SUCCESS

SSP Loading (Raw Logs)


File
Edit
Selection
View
Go
Run
Terminal
Help
EXPLORER
OPEN EDITORS
ATLASIFY
jira
nist-800-171
oscal
80053rev5-excelConversion.json
AtlasityControls.json
importer.py
NIST_SP-800-53_rev5-FINAL_catalog.json
NIST_SP-800-53_rev5-FPD_HIGH-baseline_profile.json
NIST_SP-800-53_rev5-FPD_HIGH-baseline-resolved-profile_catalog.json
NIST_SP-800-53_rev5-FPD_LOW-baseline_profile.json
NIST_SP-800-53_rev5-FPD_LOW-baseline-resolved-profile_catalog.json
NIST_SP-800-53_rev5-FPD_MODERATE-baseline_profile.json
NIST_SP-800-53_rev5-FPD_MODERATE-baseline-resolved-profile_catalog.json
NIST_SP-800-53_rev5-FPD_PRIVACY-baseline_profile.json
NIST_SP-800-53_rev5-FPD_PRIVACY-baseline-resolved-profile_catalog.json
OSCALParsedControls-High.json
OSCALParsedControls-Low.json
OSCALParsedControls-Moderate.json
OSCALParsedControls-Privacy.json
OSCALParsedControls.json
OSCALParsedFamilies.json
OSCALParsedResources.json
profiles.py
README.md
sp800-53r5-controls.xlsx
oscal-ssp-import
Atlasity-SSP-Imported.pdf
importer.py
README.md
ssp_v1_oscal_json.json
README.md
importer.py oscal-ssp-import
README.md
importer.py oscal
ssp_v1_oscal_json.json
oscal-ssp-import > importer.py > ...
Set as interpreter
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

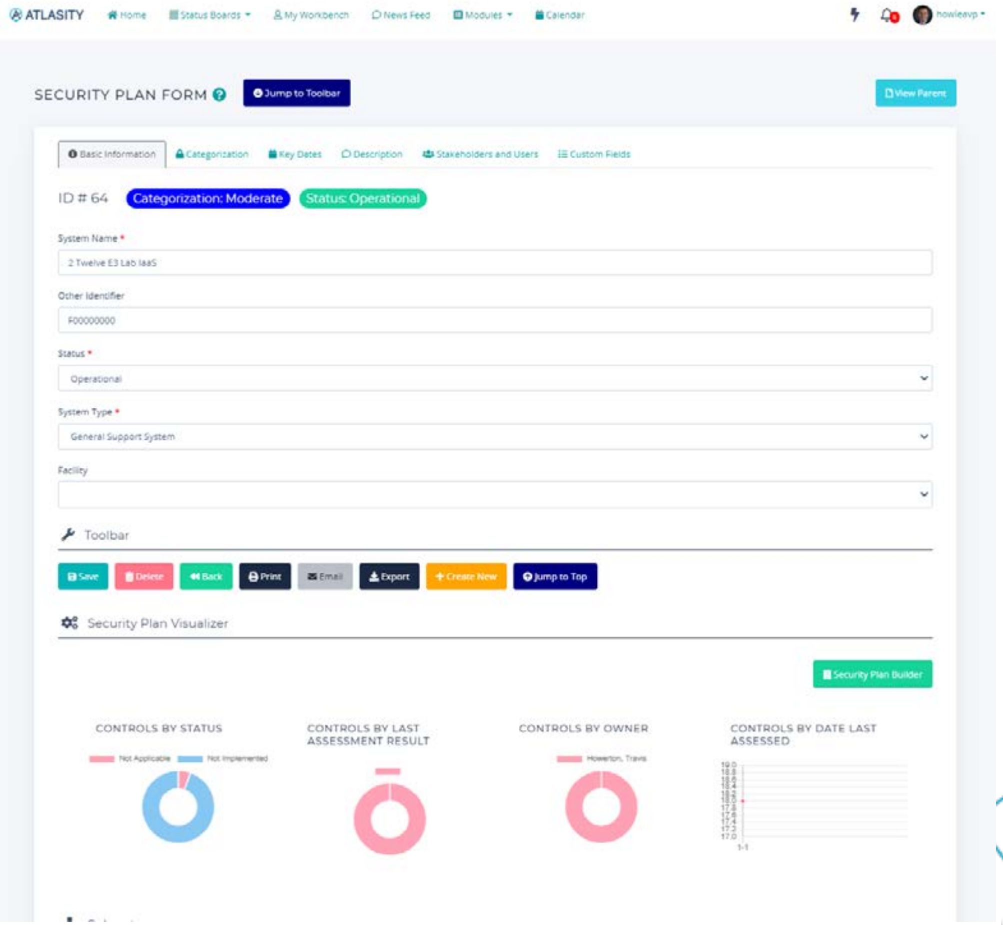
```
#!/usr/bin/python
# This code sample uses the 'requests' library:
# http://docs.python-requests.org
import requests
from requests.auth import HTTPBasicAuth
import json
import argparse

class Logger:
    OK = '\033[92m'
    WARNING = '\033[93m'
    ERROR = '\033[91m'
```

```
Atlasity Output


Security Plan ID: 67
Atlasity 9290 - AC-1 - Policy and Procedures matches OSCAL ac-1
Atlasity 9291 - AC-2 - Account Management matches OSCAL ac-2
Atlasity 9329 - AT-1 - Policy and Procedures matches OSCAL at-1
Atlasity 9335 - AU-1 - Policy and Procedures matches OSCAL au-1
Atlasity 9352 - CA-1 - Policy and Procedures matches OSCAL ca-1
Atlasity 9362 - CM-1 - Policy and Procedures matches OSCAL cm-1
Atlasity 9386 - CP-1 - Policy and Procedures matches OSCAL cp-1
Atlasity 9409 - IA-1 - Policy and Procedures matches OSCAL ia-1
Atlasity 9433 - IR-1 - Policy and Procedures matches OSCAL ir-1
Atlasity 9446 - MA-1 - Policy and Procedures matches OSCAL ma-1
Atlasity 9455 - MP-1 - Policy and Procedures matches OSCAL mp-1
Atlasity 9462 - PE-1 - Policy and Procedures matches OSCAL pe-1
Atlasity 9480 - PL-1 - Policy and Procedures matches OSCAL pl-1
Atlasity 9522 - PS-1 - Policy and Procedures matches OSCAL ps-1
Atlasity 9538 - RA-1 - Policy and Procedures matches OSCAL ra-1
Atlasity 9539 - SA-1 - Policy and Procedures matches OSCAL sa-1
Atlasity 9556 - SC-1 - Policy and Procedures matches OSCAL sc-1
Atlasity 9581 - SI-1 - Policy and Procedures matches OSCAL si-1
SUCCESS: All controls were found and mapped correctly for this catalog.
Success - 3996
Success - 3997
Success - 3998
Success - 3999
Success - 4000
Success - 4001
Success - 4002
Success - 4003
Success - 4004
Success - 4005
Success - 4006
Success - 4007
Success - 4008
Success - 4009
Success - 4010
Success - 4011
Success - 4012
Success - 4013
18 controls uploaded to Atlasity.
PS C:\dev\atlasify\oscal-ssp-import> []
```

SSP in Atlasity



SECURITY PLAN FORM

ID # 64 Categorization: Moderate Status: Operational

System Name *
2 Twelve E3 Lab IaaS

Other Identifier
F00000000

Status *
Operational

System Type *
General Support System

Facility

Toolbar

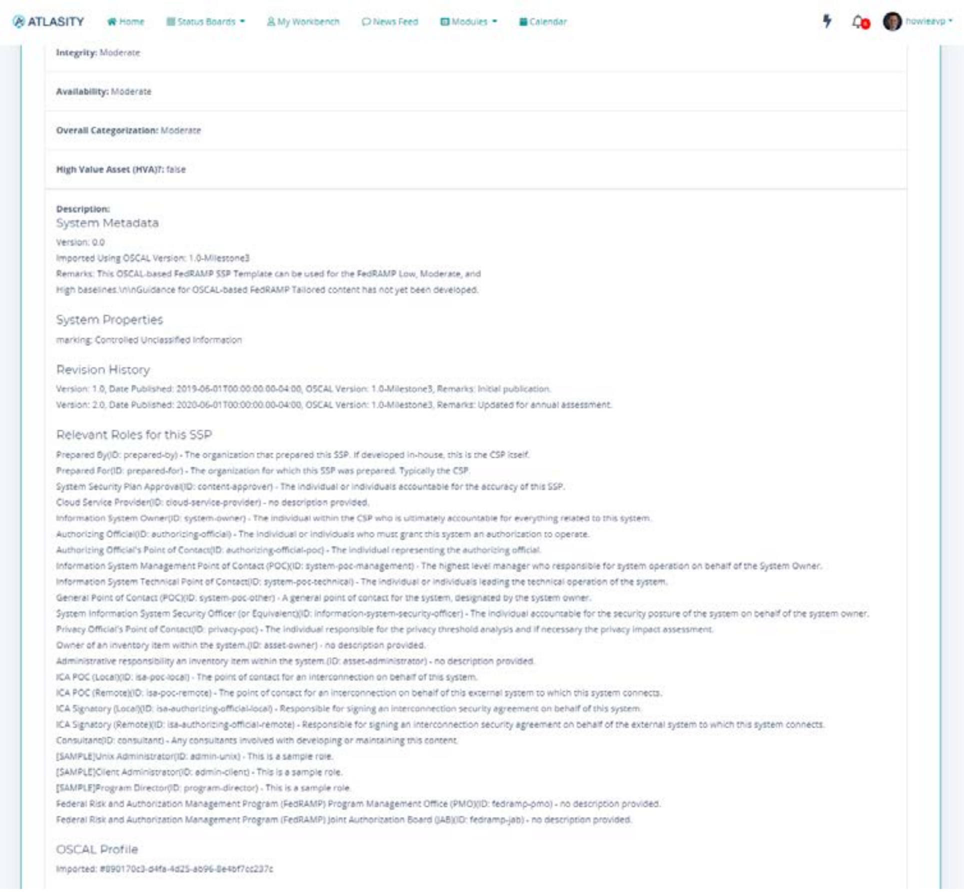
Security Plan Visualizer

© c2labs.com

Experiment 2 – Results MVP 1

SUCCESS

SSP Details



Integrity: Moderate

Availability: Moderate

Overall Categorization: Moderate

High Value Asset (HVA)?: false

Description:

System Metadata

Version: 0.0

Imported Using OSCAL Version: 1.0-Milestone3

Remarks: This OSCAL-based FedRAMP SSP Template can be used for the FedRAMP Low, Moderate, and High baselines.\n\nGuidance for OSCAL-based FedRAMP Tailored content has not yet been developed.

System Properties

marking: Controlled Unclassified Information

Revision History

Version: 1.0, Date Published: 2019-06-01T00:00:00.00-04:00, OSCAL Version: 1.0-Milestone3, Remarks: Initial publication.

Version: 2.0, Date Published: 2020-06-01T00:00:00.00-04:00, OSCAL Version: 1.0-Milestone3, Remarks: Updated for annual assessment.

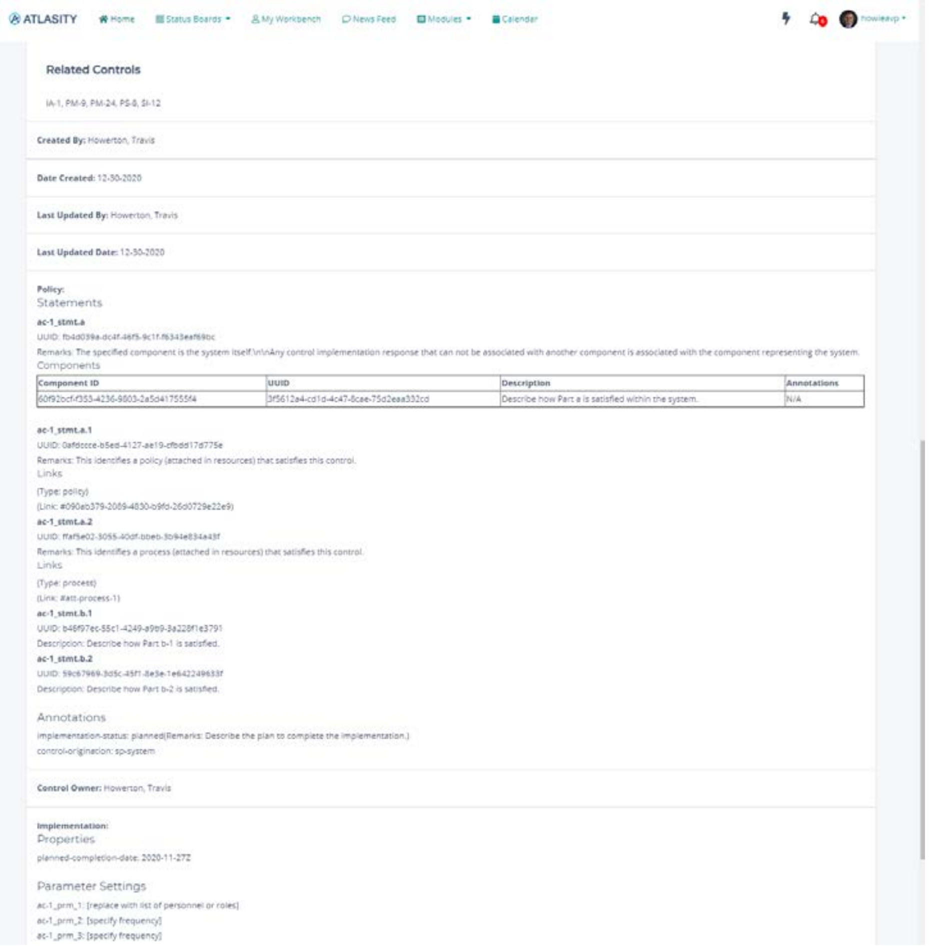
Relevant Roles for this SSP

Prepared By(ID: prepared-by) - The organization that prepared this SSP. If developed in-house, this is the CSP itself.
Prepared For(ID: prepared-for) - The organization for which this SSP was prepared. Typically the CSP.
System Security Plan Approval(ID: content-approver) - The individual or individuals accountable for the accuracy of this SSP.
Cloud Service Provider(ID: cloud-service-provider) - no description provided.
Information System Owner(ID: system-owner) - The individual within the CSP who is ultimately accountable for everything related to this system.
Authorizing Official(ID: authorizing-official) - The individual or individuals who must grant this system an authorization to operate.
Authorizing Official's Point of Contact(ID: authorizing-official-poc) - The individual representing the authorizing official.
Information System Management Point of Contact (POC)(ID: system-poc-management) - The highest level manager who responsible for system operation on behalf of the System Owner.
Information System Technical Point of Contact(ID: system-poc-technical) - The individual or individuals leading the technical operation of the system.
General Point of Contact (POC)(ID: system-poc-other) - A general point of contact for the system, designated by the system owner.
System Information System Security Officer (or Equivalent)(ID: information-system-security-officer) - The individual accountable for the security posture of the system on behalf of the system owner.
Privacy Official's Point of Contact(ID: privacy-poc) - The individual responsible for the privacy threshold analysis and if necessary the privacy impact assessment.
Owner of an inventory item within the system.(ID: asset-owner) - no description provided.
Administrative responsibility an inventory item within the system.(ID: asset-administrator) - no description provided.
ICA POC (Local)(ID: isa-poc-local) - The point of contact for an interconnection on behalf of this system.
ICA POC (Remote)(ID: isa-poc-remote) - The point of contact for an interconnection on behalf of this external system to which this system connects.
ICA Signatory (Local)(ID: isa-authorizing-official-local) - Responsible for signing an interconnection security agreement on behalf of this system.
ICA Signatory (Remote)(ID: isa-authorizing-official-remote) - Responsible for signing an interconnection security agreement on behalf of the external system to which this system connects.
Consultant(ID: consultant) - Any consultants involved with developing or maintaining this content.
[SAMPLE]Unix Administrator(ID: admin-unix) - This is a sample role.
[SAMPLE]Client Administrator(ID: admin-client) - This is a sample role.
[SAMPLE]Program Director(ID: program-director) - This is a sample role.
Federal Risk and Authorization Management Program (FedRAMP) Program Management Office (PMO)(ID: fedramp-pmo) - no description provided.
Federal Risk and Authorization Management Program (FedRAMP) Joint Authorization Board (JAB)(ID: fedramp-jab) - no description provided.

OSCAL Profile

Imported: #890170c3-d4fa-4d25-ab96-8e4bf7cc237c

Control Implementation Details



Related Controls

IA-1, PM-9, PM-24, PS-8, SI-12

Created By: Howerton, Travis

Date Created: 12-30-2020

Last Updated By: Howerton, Travis

Last Updated Date: 12-30-2020

Policy:

Statements

ac-1_stmt.a
UUID: fb4d039a-dc4f-46f5-9c1f-f6343eaf69bc
Remarks: The specified component is the system itself.\n\nAny control implementation response that can not be associated with another component is associated with the component representing the system.

Components

Component ID	UUID	Description	Annotations
60f92bcf-f353-4236-9803-2a5d417555f4	3f5612a4-cd1d-4c47-8cae-75d2eaa332cd	Describe how Part a is satisfied within the system.	N/A

ac-1_stmt.a.1
UUID: 0afdccce-b5ed-4127-ae19-cfbdd17d775e
Remarks: This identifies a policy (attached in resources) that satisfies this control.
Links
(Type: policy)
(Link: #090ab379-2089-4830-b9fd-26d0729e22e9)

ac-1_stmt.a.2
UUID: ffaf5e02-3055-40df-bbeb-3b94e834a43f
Remarks: This identifies a process (attached in resources) that satisfies this control.
Links
(Type: process)
(Link: #att-process-1)

ac-1_stmt.b.1
UUID: b46f97ec-55c1-4249-a9b9-3a228f1e3791
Description: Describe how Part b-1 is satisfied.

ac-1_stmt.b.2
UUID: 59c67969-3d5c-45f1-8e5e-1e642249633f
Description: Describe how Part b-2 is satisfied.

Annotations

implementation-status: planned(Remarks: Describe the plan to complete the implementation.)
control-origination: sp-system

Control Owner: Howerton, Travis

Implementation:

Properties

planned-completion-date: 2020-11-27Z

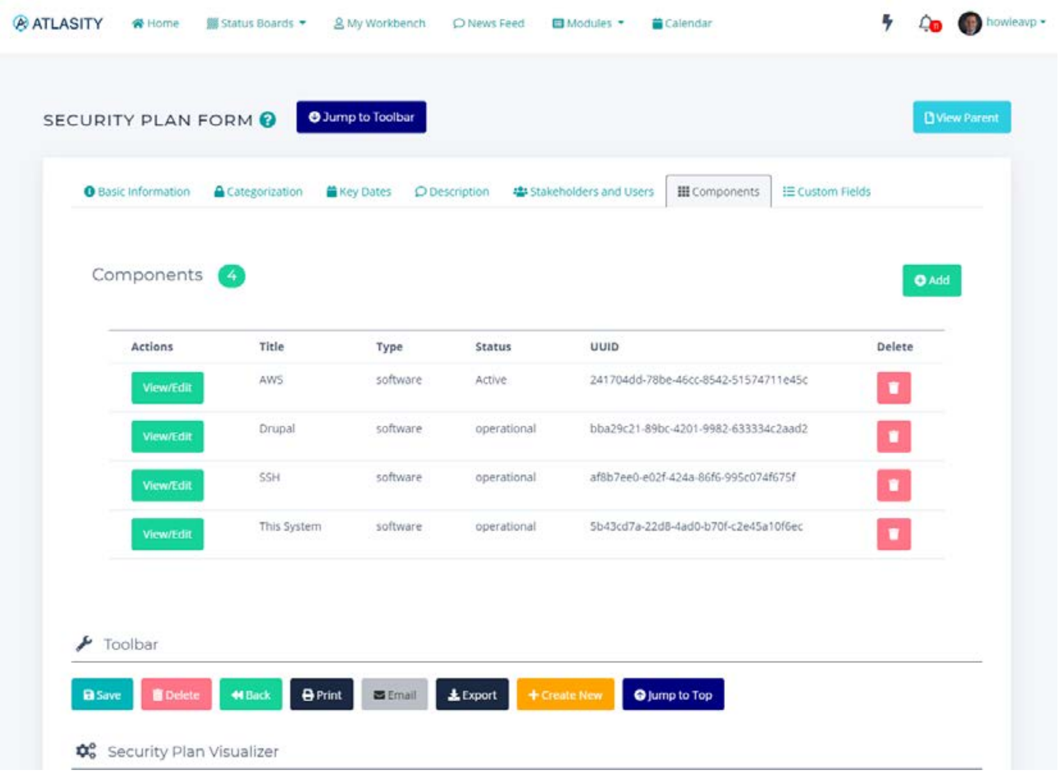
Parameter Settings

ac-1_prm_1: [replace with list of personnel or roles]
ac-1_prm_2: [specify frequency]
ac-1_prm_3: [specify frequency]

Experiment 2 – Results MVP 2

SUCCESS

Components



SECURITY PLAN FORM

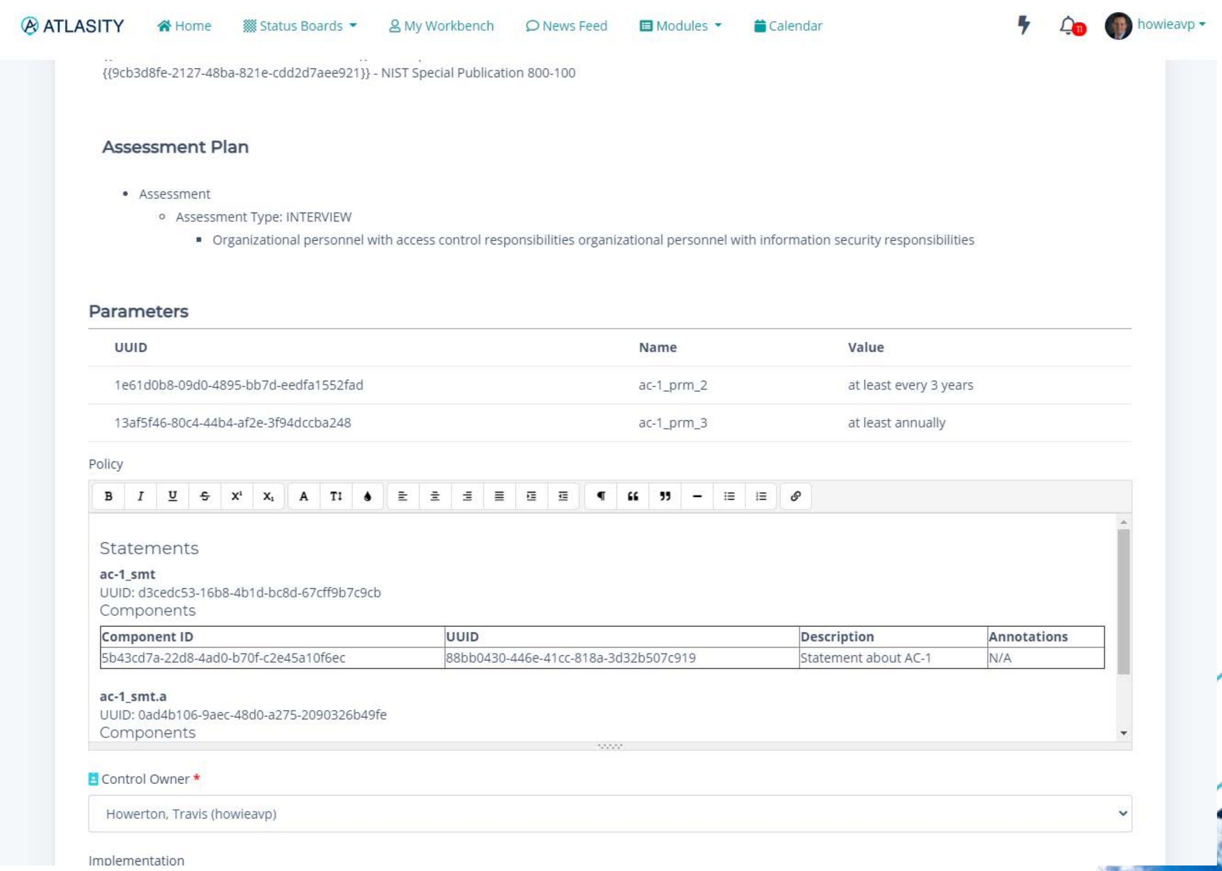
Components 4

Actions	Title	Type	Status	UUID	Delete
View/Edit	AWS	software	Active	241704dd-78be-46cc-8542-51574711e45c	
View/Edit	Drupal	software	operational	bba29c21-89bc-4201-9982-633334c2aad2	
View/Edit	SSH	software	operational	af8b7ee0-e02f-424a-86f6-995c074f675f	
View/Edit	This System	software	operational	5b43cd7a-22d8-4ad0-b70f-c2e45a10f6ec	

Toolbar

Security Plan Visualizer

Parameters



{{9cb3d8fe-2127-48ba-821e-cdd2d7aee921}} - NIST Special Publication 800-100

Assessment Plan

- Assessment
  - Assessment Type: INTERVIEW
    - Organizational personnel with access control responsibilities organizational personnel with information security responsibilities

Parameters

UUID	Name	Value
1e61d0b8-09d0-4895-bb7d-eedfa1552fad	ac-1_prm_2	at least every 3 years
13af5f46-80c4-44b4-af2e-3f94dccba248	ac-1_prm_3	at least annually

Policy

Statements

ac-1_smt

UUID: d3cedc53-16b8-4b1d-bc8d-67cff9b7c9cb

Components

Component ID	UUID	Description	Annotations
5b43cd7a-22d8-4ad0-b70f-c2e45a10f6ec	88bb0430-446e-41cc-818a-3d32b507c919	Statement about AC-1	N/A

ac-1_smt.a

UUID: 0ad4b106-9aec-48d0-a275-2090326b49fe

Components

Control Owner *

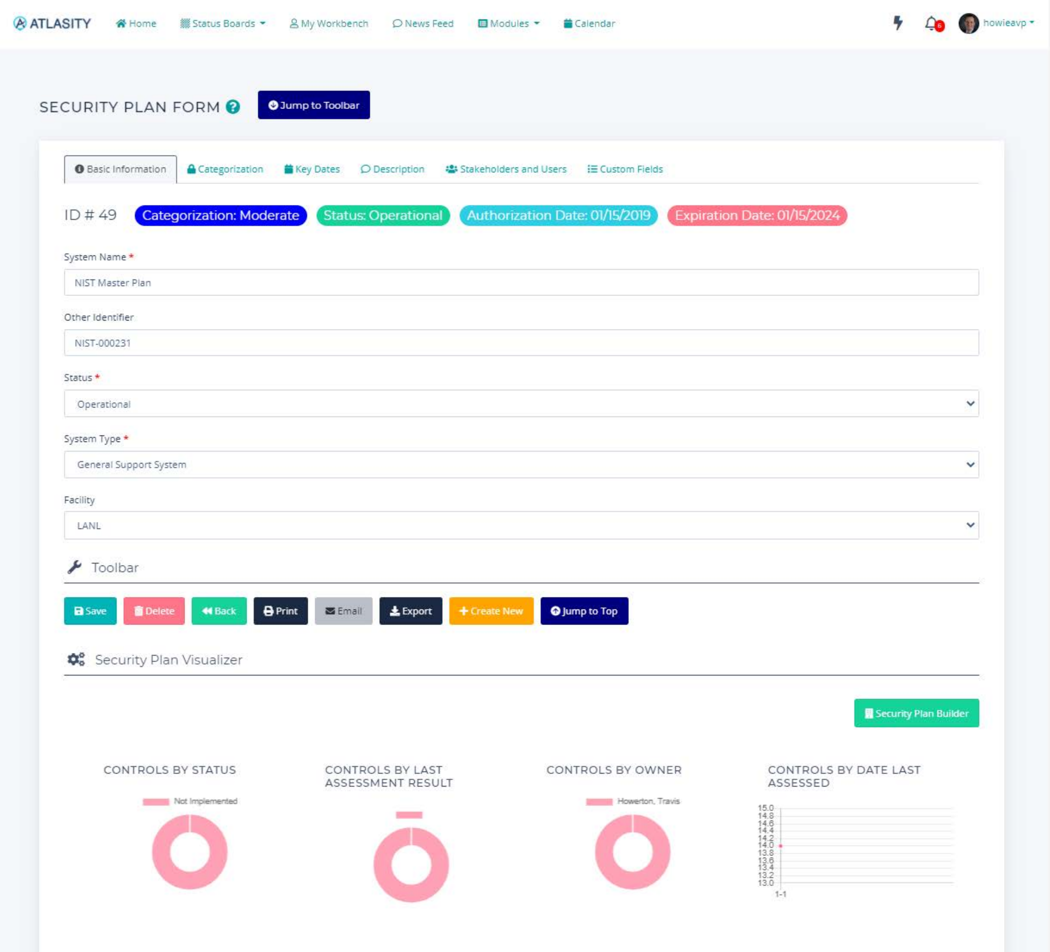
Howerton, Travis (howieavp)

Implementation

Experiment 2 – Leveraged

SUCCESS

Parent Security Plan



ATLASITY Home | Status Boards | My Workbench | News Feed | Modules | Calendar | howieavp

SECURITY PLAN FORM

Jump to Toolbar

Basic Information | Categorization | Key Dates | Description | Stakeholders and Users | Custom Fields

ID # 49 | Categorization: Moderate | Status: Operational | Authorization Date: 01/15/2019 | Expiration Date: 01/15/2024

System Name *	NIST Master Plan
Other Identifier	NIST-000231
Status *	Operational
System Type *	General Support System
Facility	LANL

Toolbar

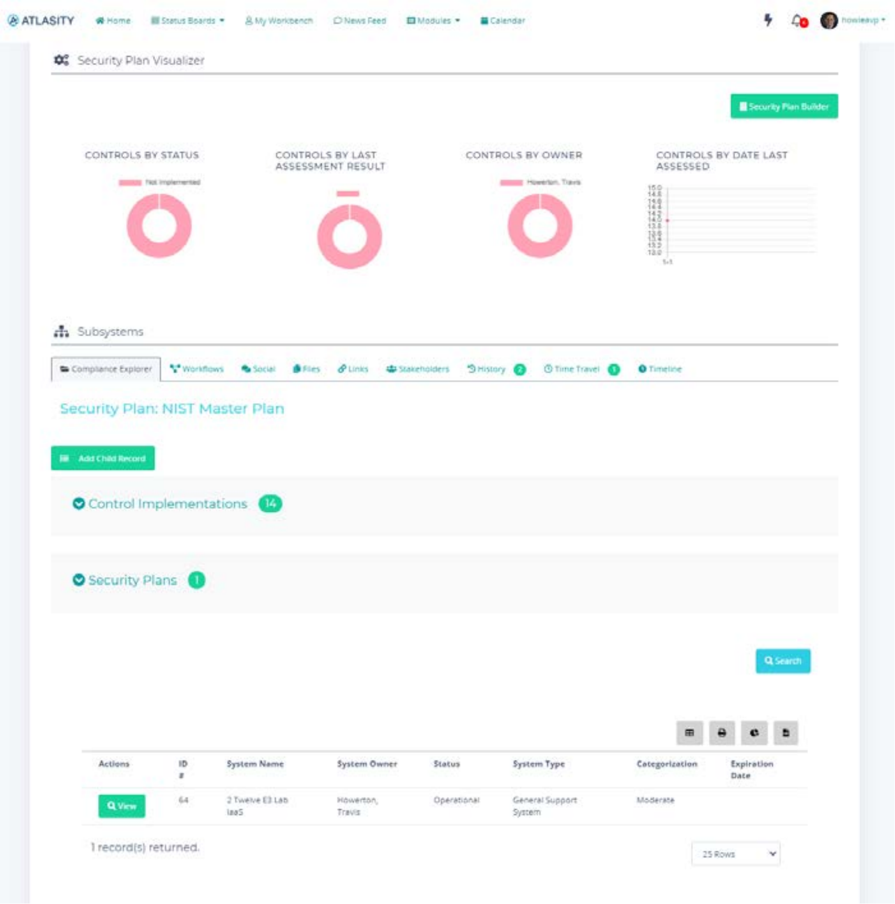
Save | Delete | Back | Print | Email | Export | Create New | Jump to Top

Security Plan Visualizer

Security Plan Builder

CONTROLS BY STATUS | CONTROLS BY LAST ASSESSMENT RESULT | CONTROLS BY OWNER | CONTROLS BY DATE LAST ASSESSED

Leveraged Authorization



ATLASITY Home | Status Boards | My Workbench | News Feed | Modules | Calendar | howieavp

Security Plan Visualizer

Security Plan Builder

CONTROLS BY STATUS | CONTROLS BY LAST ASSESSMENT RESULT | CONTROLS BY OWNER | CONTROLS BY DATE LAST ASSESSED

Subsystems

Compliance Explorer | Workflows | Social | Files | Links | Stakeholders | History 2 | Time Travel 1 | Timeline

Security Plan: NIST Master Plan

Add Child Record

Control Implementations 14

Security Plans 1

Search

Actions	ID #	System Name	System Owner	Status	System Type	Categorization	Expiration Date
View	64	2 Twelve E3 Lab IaaS	Howerton, Travis	Operational	General Support System	Moderate	

1 record(s) returned.

25 Rows

C2 LABS

© c2labs.com

Learnings

- OSCAL standard is extremely robust and can structure huge amounts of content for efficient machine processing
  - Less than 1 week of total programming time to demonstrate these Proof of Concepts with Atlasity and OSCAL
- OSCAL allows for porting data between two separate vendor systems (GovReady and Atlasity)
  - Atlasity can now repeatably import any GovReady SSP via the OSCAL standard using automation
- Atlasity was able to quickly align to OSCAL Release Candidate (RC) nomenclature and constructs to demonstrate feasibility of these use cases providing data on readiness level for other vendors
  - Initial proof point for the broader vendor community
- Demonstrated leveraged authorizations using parent/child security plan relationships (native control inheritance) in Atlasity
  - Important for tiering security plans and for cloud service providers to break out cloud v/s customer responsibilities

Potential Next Steps

- Adding the ability to export SSPs, SAPs, SARs, Catalogs, and Profiles in OSCAL format from Atlasity – using Atlasity Community Edition as a free OSCAL content publishing tool
  - Provides the ability to quickly and easily generate OSCAL content

- Automating Security Assessment Plans (SAPs) and Security Assessment Reports (SARs)
  - Initial integration work has been performed to take automated scan results against DISA STIGS using the MITRE Heimdall tool to automate assessments in Atlasity

- Leveraging the Atlasity Issues module for managing Plans of Actions and Milestones (POAMs)
  - Automating tracking of security deficiencies from automated scanning and CDM tools

- Community feedback and sharing lessons learned with the NIST OSCAL and ATARC teams from this POC for continuous improvement

- Development of PIP or NPM packages for handling OSCAL content
  - Lower barrier of entry for adoption by tool vendors and other developers