

# Agenda

z/OSMF Guild Meeting 33 - October 16th, 2024

## Security Configuration Assistant (SCA) Overview and Customer Experience

**Beate Kawelke**

Lead Architect Mainframe, T-Systems

**David Stein**

T-Systems

**Hiren Shah**

z/OSMF Architect, IBM



Discover | Learn | Share

# NOTICE



This is a public conversation and is NOT protected by any non-disclosure agreement.

Please keep all topics publicly available.

If you would like more information about joining sponsor user programs please reach out to the steering committee.

Includes planned future deliverables

# z/OSMF Guild Team



**Fiona King - [Fiona.King@ibm.com](mailto:Fiona.King@ibm.com)**

z/OSMF Product Management



**Rolando Perez - [rolandop@us.ibm.com](mailto:rolandop@us.ibm.com)**

z/OS Next & z/OSMF Design Lead



**Hiren Shah - [hiren@us.ibm.com](mailto:hiren@us.ibm.com)**

STSM, z/OSMF Chief Architect

Please reach out to us  
with any questions or  
concerns.

# Guild Session 33 Website

<https://ibm.biz/zOSMFGuild33>

- Agenda
- Recordings
- Discussion Forum
- Presentation
- Poll
- Additional Resources

z/OSMF Community Guild - Meeting No. 33

## Security Configuration Assistant Overview and Customer Experience

Join us to discover how the z/OSMF Security Configuration Assistant (SCA) simplifies security management on z/OS. This session will provide an overview of SCA and explore real-world insights from IBM Z customer T Systems, showcasing their experience using SCA for their security configuration.

Wednesday October 16, 2024  
10:00 to 11:00 AM EST

<https://ibm.biz/zOSMFGuild33>



**Featured Speakers**



Beate Kawelke  
Lead Architect Mainframe, T Systems



David Stein  
T Systems



Hiren Shah  
STSM & z/OSMF Architect IBM

Statements regarding IBM future direction and intent are subject to change or withdrawal and represent goals and objectives only.

## What's Next!

- Capacity Provisioning
- Building your own z/OSMF Plugin
- Telemetry
- Zowe and z/OSMF

[ibm.biz/zOSMFGuildHome](https://ibm.biz/zOSMFGuildHome)  
Guild Home Page for access to all materials presented during the Guild

[ibm.biz/zOSMFCommunity](https://ibm.biz/zOSMFCommunity)  
Join our Community page for updates and new content



## Problem Statements:

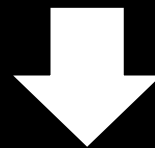
Many z/OS functions are protected by security configuration. However, sometimes security configuration is not free:

Have you ever searched several sources just to figure out what security requirements are required for some function to work?

Have you spent significant effort to investigate why a function doesn't work or stop working only to discover it's because of a missing or broken security setup?

Have you ever had to learn about different security related commands and all the flexibility (even though you are not security administrator) in order to make some function in your sandbox system to work?

Have you had to learn about differences between different security products?

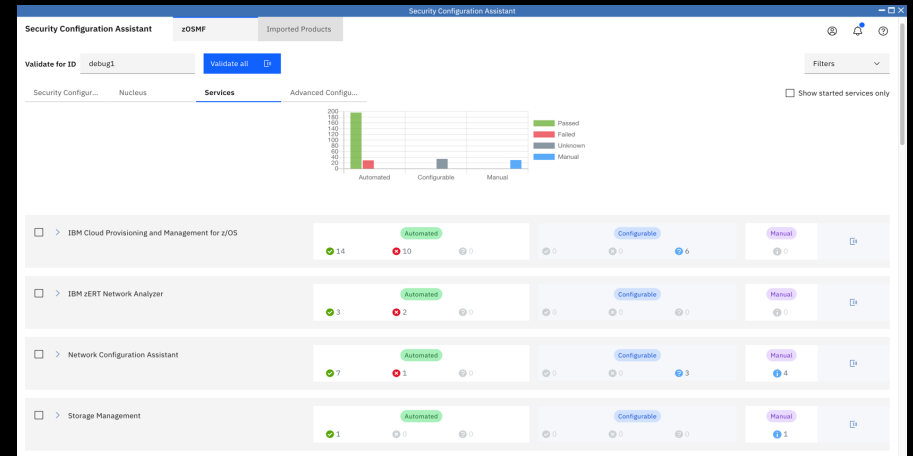


Efficiency + Learning curve

# What is Security Configuration Assistant (SCA)

SCA is a z/OSMF service (plugin) intends to simplify security configuration (starts with SAF resource based configuration) for z/OS functions.

- Allows people to describe security requirements in **JSON format** (easy to create and understand)
- The security requirements can be organized based on flexible needs (**by solution, product, function, etc.**)
- Provides **GUI** to display security requirements based on the organization in JSON file.
- Be able to **automatically validate** security requirements for target user or group. Validation can be done by individual resource, selected resources, selected product or validating all.
- **SAF based** and provides consistent experience for all security products.
- Supports **variables** in the security resource profiles so that runtime values can be leveraged.
- Supports **filtering** by different validation result.



Resources for Network Configuration Assistant base Functions	Description	Class	Who needs the access	Required Access	Validated ID	Validation Result	Action
IZUP3SF.Izu.ConfigurationAssistant.IzuUsers	Allow the user to connect to the Network Configuration A...	EBRROLE	IZUSER IZUADMIN	READ	DEBUG1	Passed	[i]
IZUP3SF.ZOSMF.CONFIGURATION_ASSISTANT.CDNF CONFIGURATION_ASSISTANT	Allow the user to access the Network Configuration Asses...	ZNFAPLA	IZUSER IZUADMIN	READ	DEBUG1	Passed	[i]

Resources for Network support for IBM Cloud Provisioning and Management for z/OS	Description	Class	Who needs the access	Required Access	Validated ID	Validation Result	Action
MVS.VARY.TOPIC.OBEVFILE	Allows the Network Configuration Assistant task to issue ...	OPERCMDS	IZUSV1	CONTROL	IZUSV1	Passed	[i]
MVS.MCSOPER.ZCDPLM*	Allows the Network Configuration Assistant task to issue ...	OPERCMDS	IZUSV1	READ	IZUSV1	Passed	[i]
MVS.DDSPLAY.XCF	Allows the Network Configuration Assistant task to issue ...	OPERCMDS	IZUSV1	READ	IZUSV1	Passed	[i]

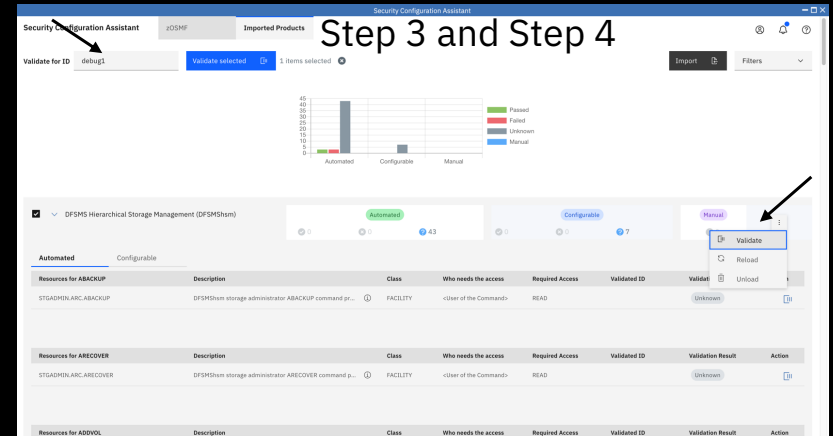
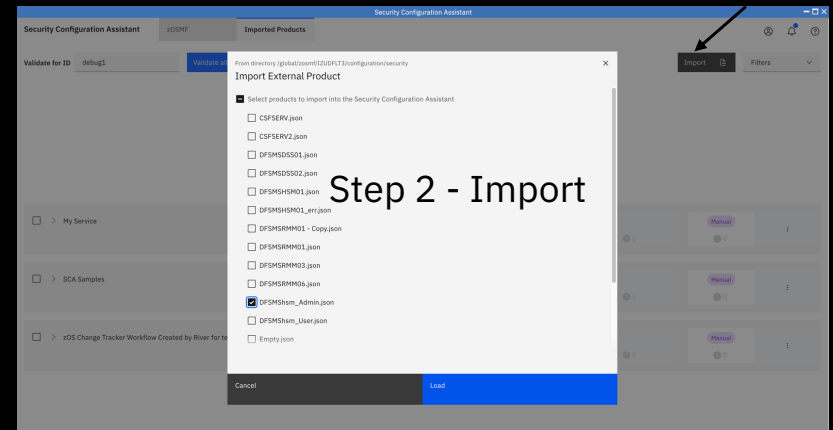
# How to use SCA

SCA was initially built to help with z/OSMF security configuration only. Later on, we open SCA to external products by support importing JSON files. The typical steps are

1. Prepare the JSON file (a.k.a. Security Descriptor file). The JSON file might be from software vendor or created by customer themselves.
2. Import the Security Descriptor file to SCA\*
3. Specify target user id or group id
4. Validate

```
{
  "ServiceId": "5655S280100",
  "ServiceName": "z/OSMF ISPF",
  "MetaValidationItemVersion": 1.01,
  "Vendor": "IBM",
  "SecurityValidationItems": [
    {
      "ItemId": "5655S280100I0001000",
      "ItemType": "PROGRAMMABLE",
      "ItemCategory": "z/OSMF ISPF functions",
      "ResourceProfile": "IZUDLFT.ZOSMF.ISPF.ISPF",
      "ResourceClass": "ZMFAPLA",
      "WhoNeedsAccess": "<user or your group name>",
      "LevelOfAccessRequired": "READ",
      "ItemDescription": "Allow the user to access the ISPF task."
    }
  ]
}
```

Step 1 - A sample json



\* The file needs to be uploaded to <z/OSMF data directory>/configuration/security in advance.



# Authority required to use SCA

The target Persona of SCA is Security Administrator OR trusted System Programmer (e.g., in sandbox systems). Specifically, there are below layers of protection for SCA:

- **For users who need to use SCA**

Must have READ access to below resources:

<SAF-prefix>.ZOSMF.CONFIGURATION.SECURITY\_ASSISTANT CLASS(ZMFAPLA)

<SAF-prefix>.IzuManagementFacilitySecurityConfigurationAssistant.izuUsers CLASS(EJBROLE)

<SAF-prefix> is "IZUDFLT" by default.

- **For z/OSMF server started task user ID ("IZUSVR" by default)**

Must have READ access to below resources:

BBG.SECCLASS.<class-name-to-be-validated> CLASS(SERVER)

<class-name-to-be-validated> is the SAF class name whose resources that you would like SCA be able to validate against. E.g., BBG.SECCLASS.SERVAUTH allows SCA to validate resources in SAF class "SERVAUTH". You can leverage generic profile to avoid defining multiple resources.

- **For target users who you want to validate with SCA**

Must have READ access to below resources (can be done through connecting the user to a group who have below access):

<SAF-prefix> CLASS(APPL)

<SAF-prefix> is IZUDFLT by default

# z/OSMF Security Configuration Assistant

## Customer Showcase

T-Systems International GmbH | SNS-Team | 16.10.2024

**T Systems** Let's power  
higher performance

# Agenda

- 01 Goal
- 02 Approach
- 03 Demo
- 04 Conclusion



# Goal

## Support Security Team

Support the Security team with the setup and maintenance of the security definitions for our software products:

- Scheduling / joblog management (T-Systems AJM)
- Software provisioning (T-Systems AppChange and PSI)
- Other products will follow



Replace reading the documentation or adapting the sample JCL with an interactive GUI

Provide a JSON schema and JSON files (one per product):

- Create security configuration file based on schema and deploy it with our products
- Replace reading / manual adaption by interactive GUI (create / check / modify security settings)

# Goal

## Support Security Team

Support the Security team with the setup and maintenance of the security definitions for our software products:

- Scheduling / joblog management (T-Systems AJM)
- Software provisioning (T-Systems AppChange and PSI)
- Other products will follow



Replace reading the documentation or adapting the sample JCL with an interactive GUI

Provide a JSON schema and JSON files (one per product):

- Create security configuration file based on schema and deploy it with our product
- Replace reading / manual adaption by interactive GUI (create / check / modify security settings)

# Approach



## Gather Information

Gather information through

- IBM manuals/documentation
- GitHub post about SCA
- IBM Community blog posts

# Approach



## Gather Information

Gather information through

- IBM manuals/documentation
- GitHub post about SCA
- IBM Community blog posts



## JSON Schema & JSON File

Create a JSON Schema & File

- Create JSON Schema
- Using JSON Schema in VS-Code to create JSON File

# Approach



## Gather Information

Gather information through

- IBM manuals/documentation
- GitHub post about SCA
- IBM Community blog posts



## JSON Schema & JSON File

Creating a JSON Schema & File

- Create JSON Schema
- Using JSON Schema in VS-Code to create JSON File



## Upload and Test JSON File

Upload File in z/OSMF SCA

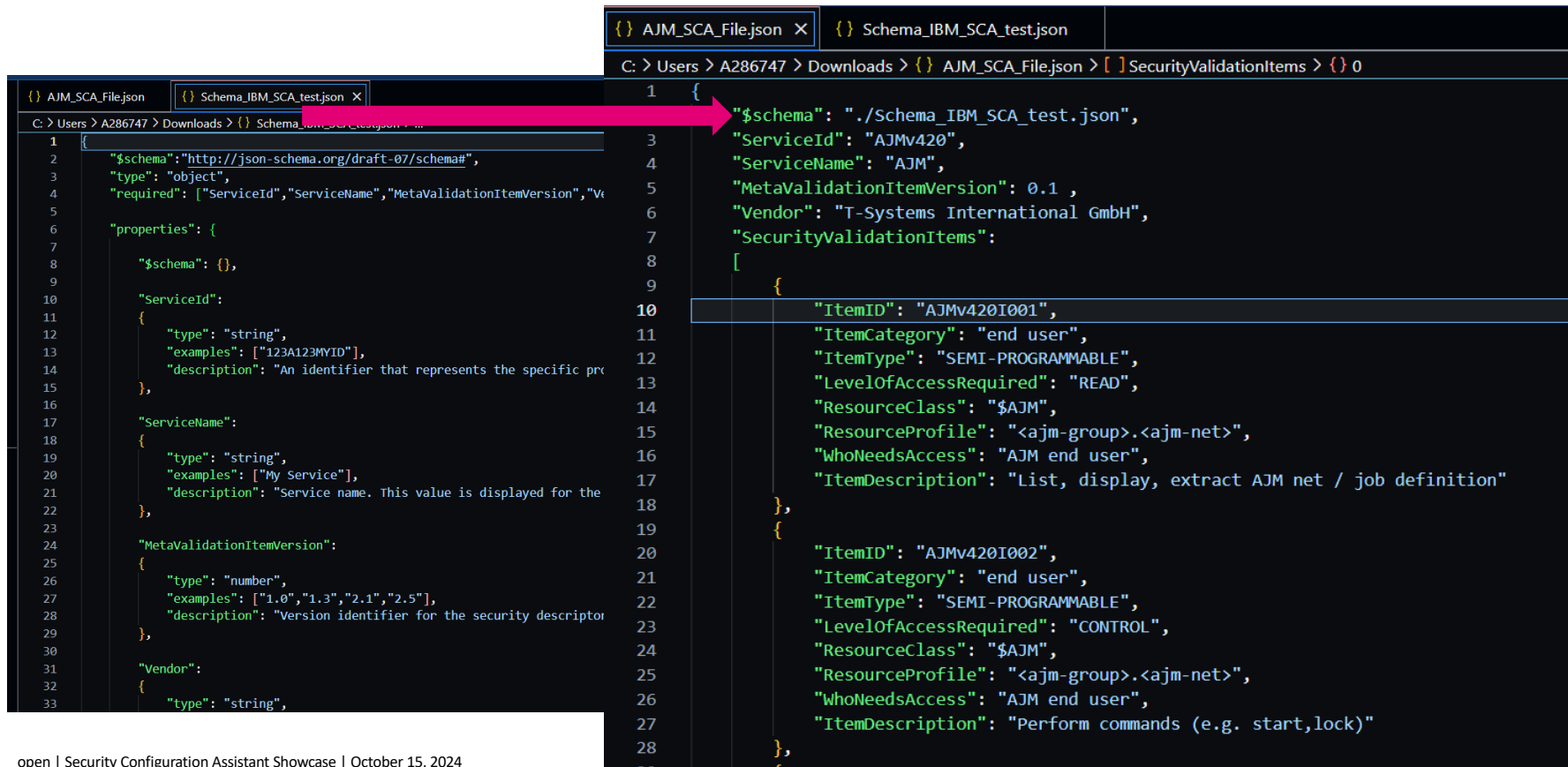
- Upload JSON File to the SCA
- Check what each parameter does
- Check additional features (variables, RACF commands)



DEMO

**T Systems** Let's power  
higher performance

# SCA - Create the JSON file

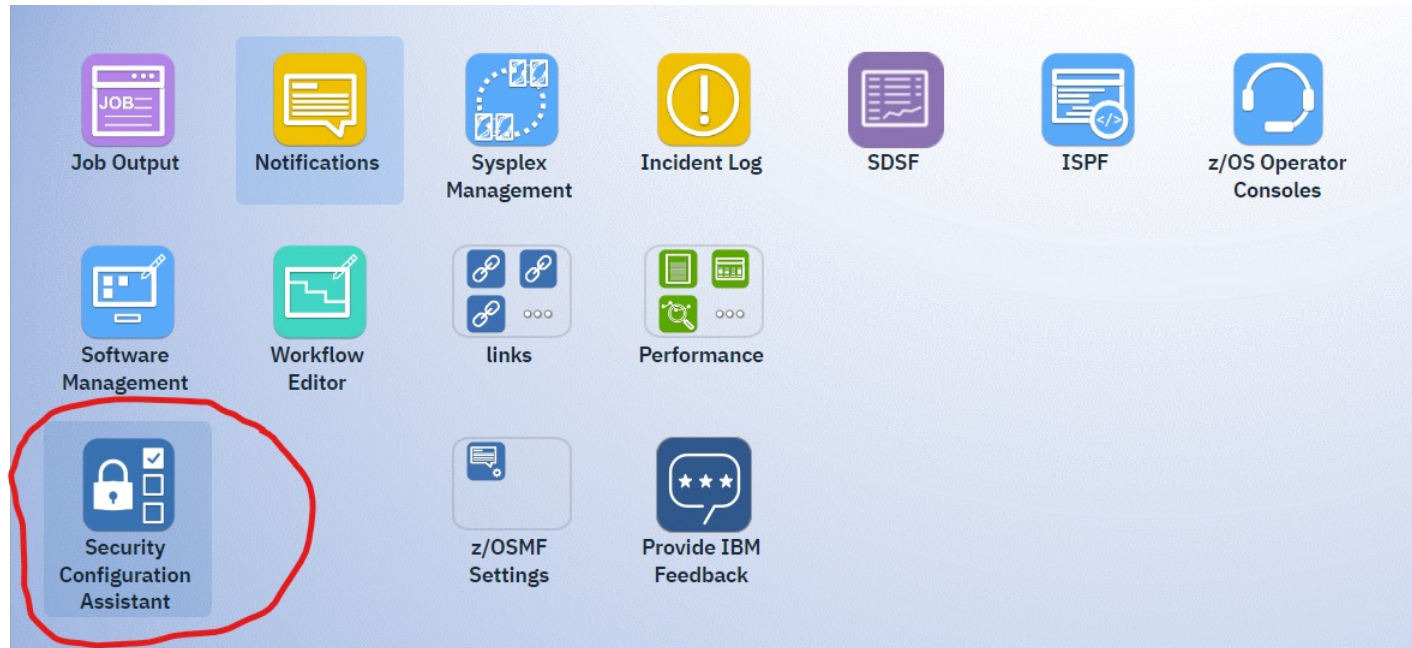


```
{} AJM_SCA_File.json X {} Schema_IBM_SCA_test.json
C: > Users > A286747 > Downloads > {} AJM_SCA_File.json > [ ] SecurityValidationItems > {} 0

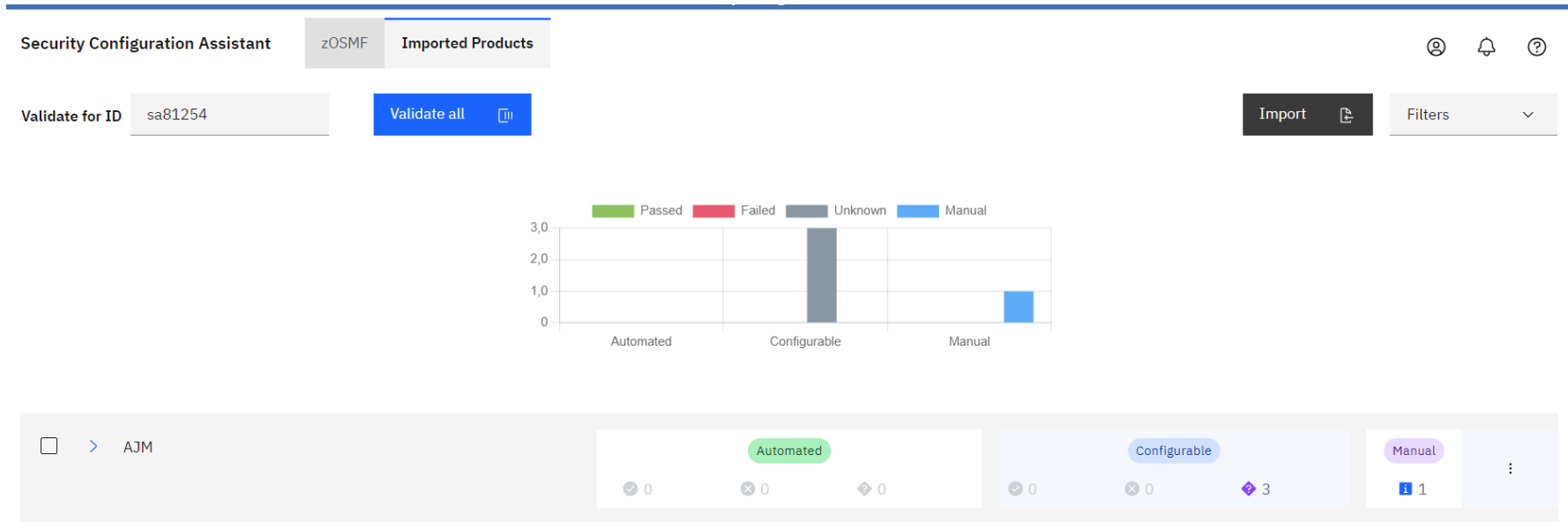
{} AJM_SCA_File.json {} Schema_IBM_SCA_test.json X
C: > Users > A286747 > Downloads > {} Schema_IBM_SCA_test.json > [ ] SecurityValidationItems > {} 0
1 {
2   "$schema": "http://json-schema.org/draft-07/schema#",
3   "type": "object",
4   "required": ["ServiceId", "ServiceName", "MetaValidationItemVersion", "Vendor"],
5   "properties": {
6     "$schema": {},
7     "ServiceId":
8     {
9       "type": "string",
10      "examples": ["123A123MYID"],
11      "description": "An identifier that represents the specific process or service.",
12    },
13    "ServiceName":
14    {
15      "type": "string",
16      "examples": ["My Service"],
17      "description": "Service name. This value is displayed for the user.",
18    },
19    "MetaValidationItemVersion":
20    {
21      "type": "number",
22      "examples": ["1.0", "1.3", "2.1", "2.5"],
23      "description": "Version identifier for the security descriptor.",
24    },
25    "Vendor":
26    {
27      "type": "string",
28    }
29  }
30 }

1 {
2   "$schema": "../Schema_IBM_SCA_test.json",
3   "ServiceId": "AJMv420",
4   "ServiceName": "AJM",
5   "MetaValidationItemVersion": 0.1,
6   "Vendor": "T-Systems International GmbH",
7   "SecurityValidationItems":
8   [
9     {
10    "ItemID": "AJMv420I001",
11    "ItemCategory": "end user",
12    "ItemType": "SEMI-PROGRAMMABLE",
13    "LevelOfAccessRequired": "READ",
14    "ResourceClass": "$AJM",
15    "ResourceProfile": "<ajm-group>.<ajm-net>",
16    "WhoNeedsAccess": "AJM end user",
17    "ItemDescription": "List, display, extract AJM net / job definition"
18  },
19  {
20    "ItemID": "AJMv420I002",
21    "ItemCategory": "end user",
22    "ItemType": "SEMI-PROGRAMMABLE",
23    "LevelOfAccessRequired": "CONTROL",
24    "ResourceClass": "$AJM",
25    "ResourceProfile": "<ajm-group>.<ajm-net>",
26    "WhoNeedsAccess": "AJM end user",
27    "ItemDescription": "Perform commands (e.g. start,lock)"
28  }
29  ]
30 }
```

# SCA - start SCA



# SCA: Import and list imported products



# SCA: Display security settings with variables

☐ AJM

Automated 0 0 0

Configurable 0 0 3

Manual 1

Configurable Manual

Resources for end user	Description	Class	Who needs the access	Required Access	Validated ID	Validation Result	Action
ABC.XYZ	List, display, extract AJM net / job definition	\$AJM	AJM end user	READ			⋮
ABC.XYZ	Perform commands (e.g. start,lock)	\$AJM	AJM end user	CONTROL			⋮
#SUB.<ajm-group>.<ajm-net>.<ajm-job>	Use as value in the "USER=" parameter of the JOB card	\$AJM	AJM end user	READ		⚠	⊕

# SCA: Add variables to "semi-automated"

**Add variable values for profile <ajm-group>.<ajm-net>** ✕

Add a value for the profile variables displayed. Note that security profiles containing variables without a value cannot be validated.

ajm-group ⓘ









ajm-net ⓘ

Apply this value to all the resource names that contain same variables ⓘ

Cancel OK

# SCA: Check authorization for a user

Resources for end user	Description	Class	Who needs the access	Required Access	Validated ID	Validation Result	Action
ABC.XYZ	List, display, extract AJM net / job definition	\$AJM	AJM end user	READ	 SA81254	 Passed	⋮
TEST.T	List, display, extract AJM net / job definition	\$AJM	AJM end user	READ	 SA81254	 Passed	⋮
ABC.XYZ	Perform commands (e.g. start,lock)	\$AJM	AJM end user	CONTROL	 SA81254	 Passed	⋮
#SUB.<ajm-group>.<ajm-net>.<ajm-job>	Use as value in the "USER=" parameter of the JOB card	\$AJM	AJM end user	READ			

# SCA: Generate RACF commands to fix security

### Security update for fixing ABC.XYZ

**Command**

Review the generated commands, then click Submit to run the commands with user ID SA81254.

```
# $AJM (ABC.XYZ)
RDEFINE $AJM (ABC.XYZ)
PERMIT ABC.XYZ CLASS($AJM) ID (TSOC) ACCESS (READ)
```

Cancel Submit



# Security prerequisites

- ▶ User authorization to access the z/OSMF SCA plugin: **IZUDFLT.ZOSMF.CONFIGURATION.SECURITY\_ASSISTANT**
- ▶ z/OSMF STC authorization to check security settings: **BBG.SECCLASS.classname**
- ▶ RACF rights to fix security settings, e.g. to add a user to a profile

# Conclusion



Conclusion

Create / Upload

Display

Replace variables

- ▶ Well-documented layout of JSON file, yet some details unclear (e.g. maximum length of some values)
- ▶ Upload and activation: super easy
- ▶ Display needs some understanding “what goes where”
- ▶ Variables replacement:
  - Useful, but only variables in same position are replaced - why?
  - No “ . “ allowed in values – cannot specify e.g. multiple qualifiers in DSNs
  - Looks as if it was designed with RACF class DATASET in mind – other classes have different options on length / values

# Conclusion



Conclusion

Check authorization

Change

Remove?

- ▶ Checking a user's authorization works, also checking an entire RACF group (tooltip would be helpful)
- ▶ Adapt RACF definitions only for user and fully qualified profile  
Idea: Provide option to add a group (instead of user) to a generic profile (“best match”), do not create a discrete profile
- ▶ New feature idea: Remove authorization

The slide features a background with a red and white geometric pattern on the left and a blurred image of a white keyboard on the right. The text is white and positioned on the red background.

# Thank you for listening

Contact:

Beate Kawelke ([beate.kawelke@t-system.com](mailto:beate.kawelke@t-system.com))

David Stein ([david.stein@t-systems.com](mailto:david.stein@t-systems.com))

# Links

- Short videos on SCA: [https://mediacenter.ibm.com/media/1\\_vi437nry](https://mediacenter.ibm.com/media/1_vi437nry)
- IBM z/OS Documentation: <https://www.ibm.com/docs/en/zos/3.1.0?topic=zmfcg-creating-security-descriptor-files-security-configuration-assistant-task>
- IBM Community: <https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/river-jia/2021/07/25/zosmf-security-configuration-assistant?communityKey=1ca674e5-aada-4194-a16e-059cafe7b807>
- GitHub JSON example files : <https://github.com/IBM/IBM-Z-zOS/blob/main/zOSMF/Zosmf-SCA/README.md>
  
- Ideas:
  - Improve general handling of variables: [Link](#)
  - Improve deletion of variables: [Link](#)
  - Add more options to "fixing": [Link](#)
  - New feature to remove authorizations: [Link](#)

# SCA REST API

SCA also provides capability of programmatic security validation via REST API

- Easy to drive  
REST API is lightweight web service and can be driven by most programming languages either locally or remotely.
- Flexible source of security requirements  
The security requirements REST API accepts can be written directly in HTTP request body or in a standalone USS file.
- Simplified JSON format  
The JSON format that REST API accepts is further simplified.
- Enables use cases like
  - Regular security health check
  - Security trouble shooting
  - Security auditing to avoid over permission

```
POST /zosmf/config/security/v1/validate?userid=<userid>
Request body:
{
  "resourceItems": [
    {
      "resourceProfile": "IRR.DIGTCERT.LISTRING",
      "resourceClass": "FACILITY",
      "access": "READ"
    },
    {
      "resourceProfile": "CEA.SIGNAL.ENF83",
      "resourceClass": "SERVAUTH",
      "access": "READ"
    }
  ]
}
```

Request example

```
{
  "resourceItems": [
    {
      "resourceProfile": "IRR.DIGTCERT.LISTRING",
      "resourceClass": "FACILITY",
      "access": "READ",
      "action": "validate",
      "validatedId": "izusvr3",
      "status": "Passed"
    },
    {
      "resourceProfile": "CEA.SIGNAL.ENF83",
      "resourceClass": "SERVAUTH",
      "access": "READ",
      "action": "validate",
      "validatedId": "izusvr3",
      "status": "Passed"
    }
  ]
}
```

Result example

# Unleash z/OSMF capability to Ansible

- IBM z/OSMF Ansible collection “ibm\_zosmf”, intends to provide simple and consistent experience for Ansible users to drive z/OSMF REST APIs for z/OS operations and automation.
    - Part of Red Hat Ansible Certified Content for IBM Z.
    - Available in both Ansible Galaxy and Red Hat Ansible Automation Hub:
      - [https://cloud.redhat.com/ansible/automation-hub/repo/published/ibm/ibm\\_zosmf](https://cloud.redhat.com/ansible/automation-hub/repo/published/ibm/ibm_zosmf)
      - [https://galaxy.ansible.com/ibm/ibm\\_zosmf](https://galaxy.ansible.com/ibm/ibm_zosmf)
    - Documentation: [https://ibm.github.io/z\\_ansible\\_collections\\_doc/ibm\\_zosmf/docs/ansible\\_content.html](https://ibm.github.io/z_ansible_collections_doc/ibm_zosmf/docs/ansible_content.html)
  - What’s available today via z/OSMF Ansible collection “ibm\_zosmf”
- “ibm\_zosmf” collection drives z/OSMF REST APIs start with:
- **Workflow operations (version 1.0 and above)**
    - Drive a z/OSMF workflow to complete, Delete a workflow instance, Query workflow status, etc.
  - **Provision and Manage z/OS software instances** via Cloud Provisioning and Management for z/OS (version 1.0 and above)
    - Provision or deprovision a z/OS middleware/software instance, start or stop the software instance, etc.
  - **Security validation based on SCA (version 1.1)**
  - **Security fix/provision based on SCA (version 1.2)**
  - **Software query based on z/OSMF Software Management (version 1.3)**
  - **Run zMSC services (version 1.4)**

# Driving z/OS security validation with Ansible playbook

```
1 hosts: zos_systems
2 connection: local
3 collections:
4   - ibm.ibm_zosmf
5 gather_facts: false
6 tasks:
7   # *****
8   # Authenticate with z/OSMF server
9   # *****
10  - zmf_authenticate:
11    zmf_host: "{{ zmf_host }}"
12    zmf_port: "{{ zmf_port }}"
13    zmf_user: "{{ zmf_user }}"
14    zmf_password: "{{ zmf_password }}"
15    register: result_auth
16  # *****
17  # Perform security validation with zmf_sca module
18  # *****
19  - name: Run security validation and expect all requirements are satisfied
20    zmf_sca:
21      zmf_credential: "{{ result_auth }}"
22      location: "{{location}}" # The location of the security requirement file: local or remote
23      path_of_security_requirements: "{{path}}" # Path of the security requirement file
24      target_userid: "{{target_userid}}" # Target user/group id to be validated
25  -----
```

## Playbook



```
1 {
2   "version": "1",
3   "resourceItems": [{
4     "ResourceProfile" : "IZUP01SF.ZOSMF.CONFIGURATION.SECURITY_ASSISTANT",
5     "ResourceClass" : "ZMFAPLA",
6     "LevelOfAccessRequired" : "READ"
7   },
8   {
9     "ResourceProfile" : "IZUP01SF.ZOSMF.CONSOLES.ZOSOPER",
10    "ResourceClass" : "ZMFAPLA",
11    "LevelOfAccessRequired" : "READ"
12  }]
13 }
```

## Security Descriptor file

By default, zmf\_sca expects all requirements are satisfied

## Automation result

```
[test@shjehian49 playbooks]$ ansible-playbook -i inventory.yml security_validation.yml
PLAY [zos_systems] *****
TASK [zmf_authenticate] *****
ok: [zos_host3]
ok: [zos_host2]
ok: [zos_host1]

TASK [Run security validation and expect all requirements are satisfied] *****
fatal: [zos_host3]: FAILED! => {"changed": false, "msg": "Security validation does not match with expected result.", "resourceItems": [{"access": "UPDATE", "action": "validate", "resourceClass": "ZMFAPLA", "resourceProfile": "IZUDFLT.ZOSMF2", "status": "Failed", "validatedId": "ibuser"}, {"access": "READ", "action": "validate", "resourceClass": "ZMFAPLA", "resourceProfile": "IZUDFLT.ZOSMF2", "status": "Failed", "validatedId": "ibuser"}]}
ok: [zos_host1]
ok: [zos_host2]

TASK [debug] *****
ok: [zos_host1] => {"result": {"changed": false, "failed": false}}
ok: [zos_host2] => {"result": {"changed": false, "failed": false}}

PLAY RECAP *****
zos_host1      : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
zos_host2      : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
zos_host3      : ok=1  changed=0  unreachable=0  failed=1  skipped=0  rescued=0  ignored=0
```





**Thank You!**

