

Version 1.0  
April 2019

# Siemens' experiences with 'Scapolite', a YAML+Markdown- based alternative to XCCDF

(presented @NIST SCAP v2 Workshop April 30<sup>th</sup> to May 2<sup>nd</sup> 2019)

Dr. Bernd Grobauer, Siemens CT RDA ITS

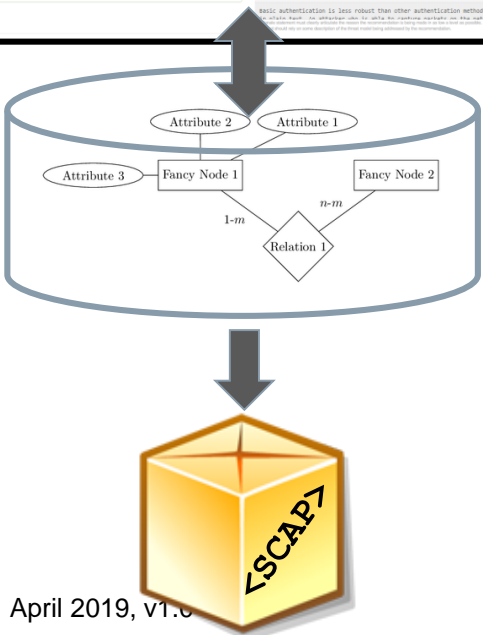
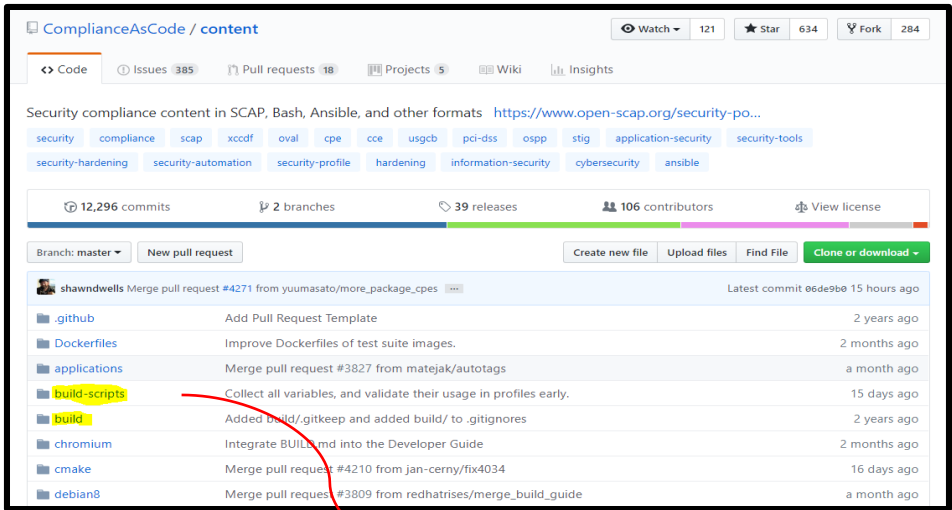
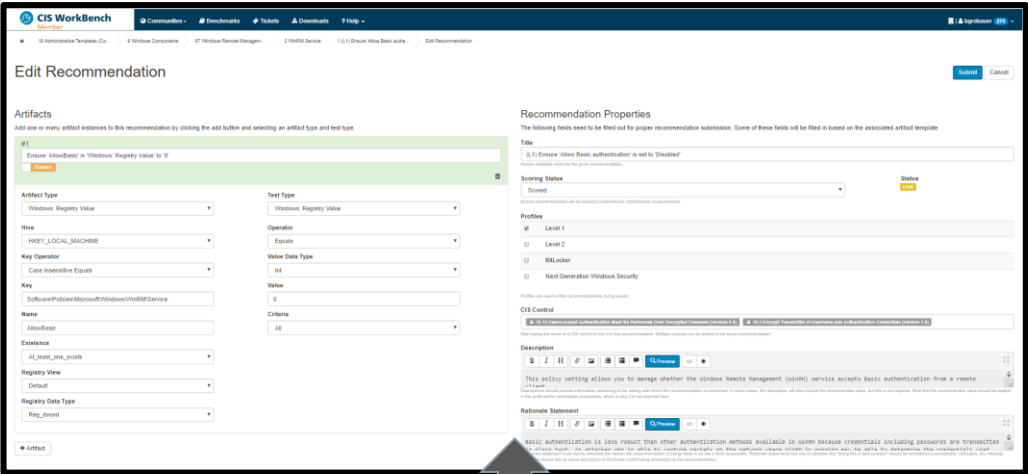


# Two (and *only* two) publishers of SCAP content that support community-driven authoring and maintenance



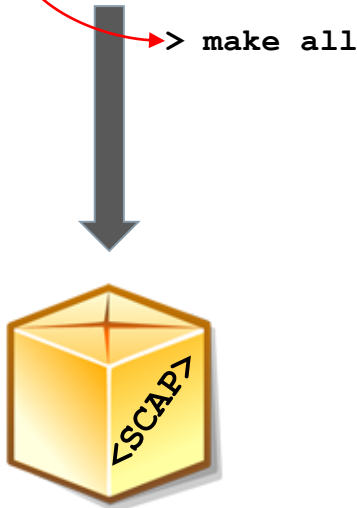
CIS

OpenSCAP



Both approaches have in common:

- Found it impossible to author and maintain SCAP as „SCAP proper“
- Chose to work with an internal“ representation of data is **not** SCAP:
  - CIS: database schema with semantics implicit in code of workbench application
  - OpenScap: mixture of file formats and file-system layout, with semantics implicit in code of build process



# Siemens' approach: Inspired by OpenSCAP (THANKS!!!), but separating content and build process via explicit semantics

Siemens CT RTC ITS

B. Grobauer  
Siemens AG  
June 23, 2017

## Table of Contents

- 1. Introduction
  - 1.1. Terminology
- 2. Scapolite Definition: Introduction
  - 2.1. Formats
    - 2.1.1. Pure YAML representation
    - 2.1.2. Combination of YAML and Markdown
  - 2.2. File-based representation
  - 2.3. Identifiers and inclusion by reference
  - 2.4. Qualified Identifiers, local identifiers, and references in Markdown
- 3. Scapolite's YAML conventions
  - 3.1. Order
  - 3.2. Scalars
  - 3.3. Attributed Text Fields
  - 3.4. The Scapolite preamble
- 4. The Scapolite object classes
  - 4.1. Rule
    - 4.1.1. Exemption
    - 4.1.2. Implementation and Check
    - 4.1.3. Crossref
  - 4.2. Collection
  - 4.3. Group
  - 4.4. Bibliography
  - 4.5. Glossary
  - 4.6. Value
  - 4.7. Profile
- 5. Structures used by several classes
  - 5.1. Applicability
    - 5.1.1. Scapolite extension mechanism for applicability
    - 5.1.2. Hierarchical resolution of applicability information
    - 5.1.3. Examples of applicability extensions

## Scapolite: YAML- and Markdown-based Authoring and Manipulation of IS Rules and Benchmarks

draft-grobauer-scapolite-latest

### Abstract

The SCAP standard for communicating machine-readable security benchmarks [SCAP\_1\_2] has been around since 2009. There has been significant but by no means overwhelming take-up of SCAP. What is most noticable, is the huge disparity between authors and consumers of SCAP. Authoring of SCAP content is done mostly by a few organizations, mainly governmental as well as the occasional vendor and non-for-profit organization. But even though there are many more organizations that author and maintain IT security rules, almost no organization uses SCAP for this purpose. This is, because authoring SCAP content is extremely complicated and cumbersome. Tools are hard to come by and have mostly proven inadequate for all but the most basic usage. SCAP's XML also does not lend itself for direct authoring on text-file basis, no matter whether a special XML editor is used, or not.

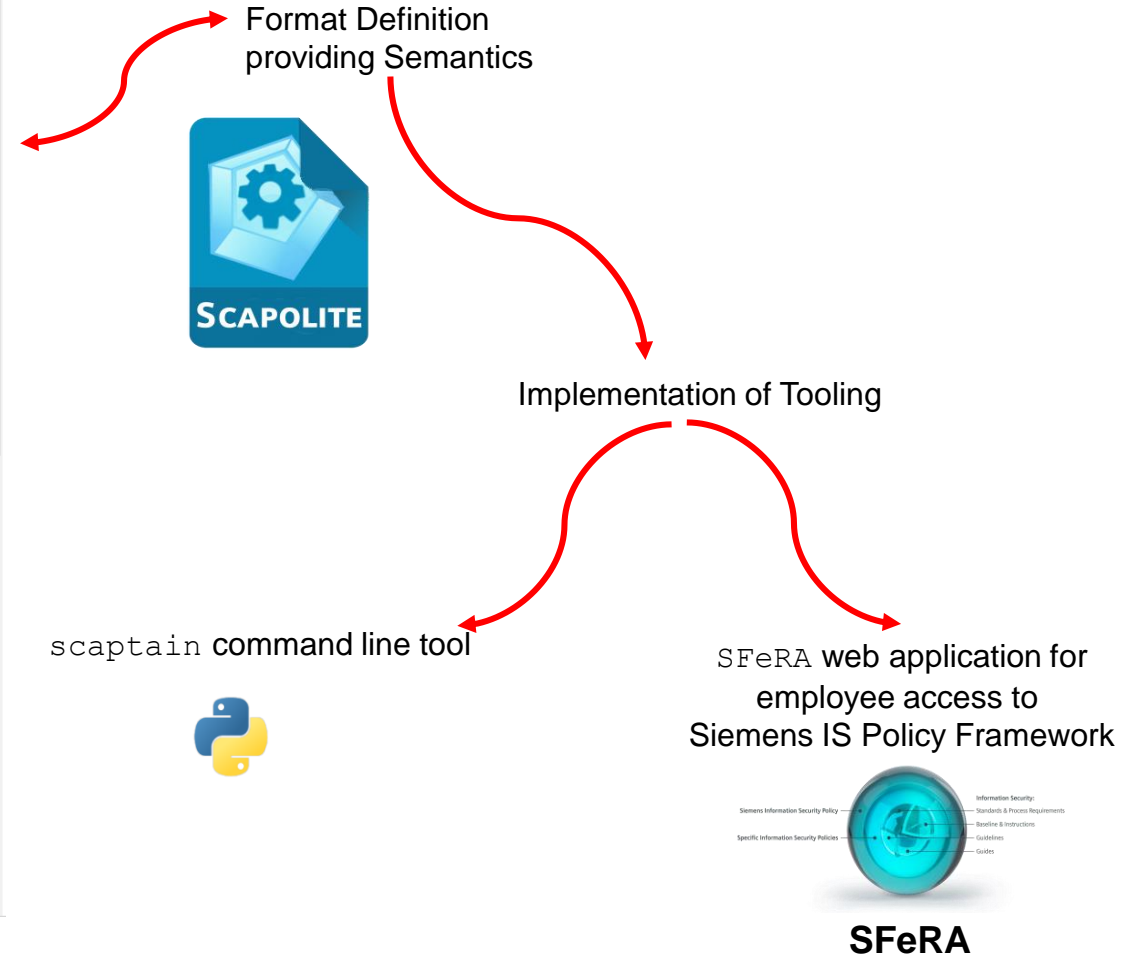
The most promising approach towards writing SCAP standard has been developed by the OpenSCAP project: SCAP content is divided into pieces of simplified XML stored in single files. Collaborative editing is done using git as version control system – SCAP content is then produced by a combination of scripts and tools that collect the XML pieces and then transform and combine them into a SCAP data stream.

Scapolite takes inspiration from OpenSCAP's approach, but goes one step further: rule collections and rules are specified in a combination of YAML and Markdown, thus putting content into a form that can easily be edited directly with a text editor, but at the same time is machine readable and thus can be read by tools for transformation into SCAP or other relevant formats.

(c) 2017 Siemens CT RDA ITS

### 1. Introduction

The name "Scapolite" is derived from inspiration from the SCAP (SCAP\_1\_2) standard. Scapolite is a tool to



# Scapolite Rule Example

## Introductory Example taken from Scapolite Documentation



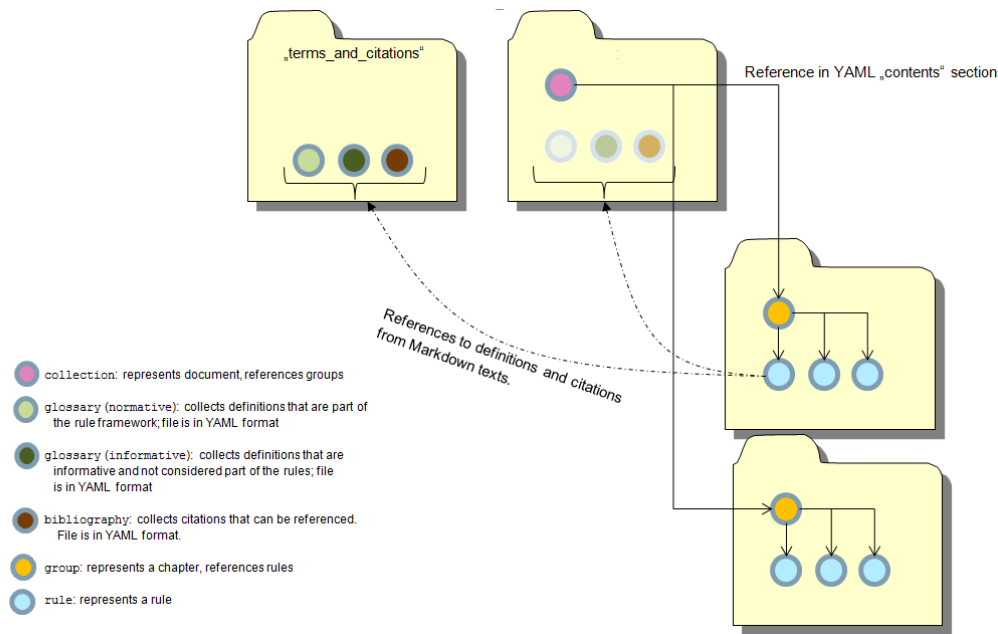
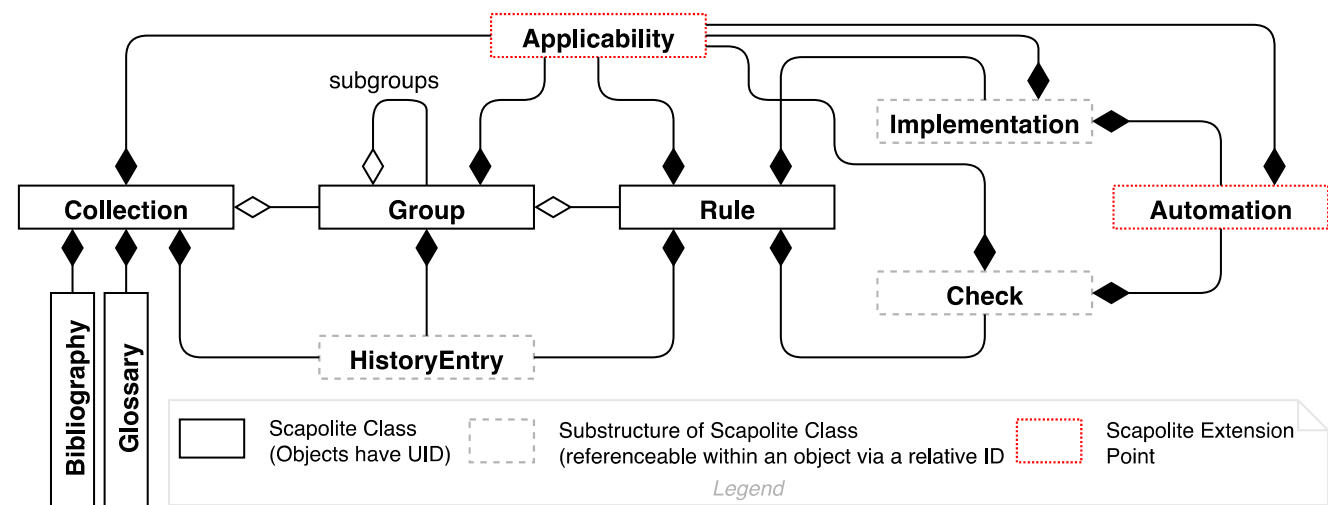
```
scapolite:
  class: rule
  version: '1.0'
  id: C085611074
  id_namespace: com.siemens.cert.scapolite.example_benchmark
  title: An example rule
  rule: Do as I say, not as I do.
  rationale: |
    There are always example of policy/rule makers who do not conform to their own
    rules. Nevertheless, many of their rules are sensible and **MUST** be obeyed.
  implementations:
    - relative_id: '01'
      title: Just do it yourself
      description: |
        Carry out the following steps:

        - Do this
        - Do that
    - relative_id: '02'
      title: Get people to do it
      description: |
        Carry out the following steps:

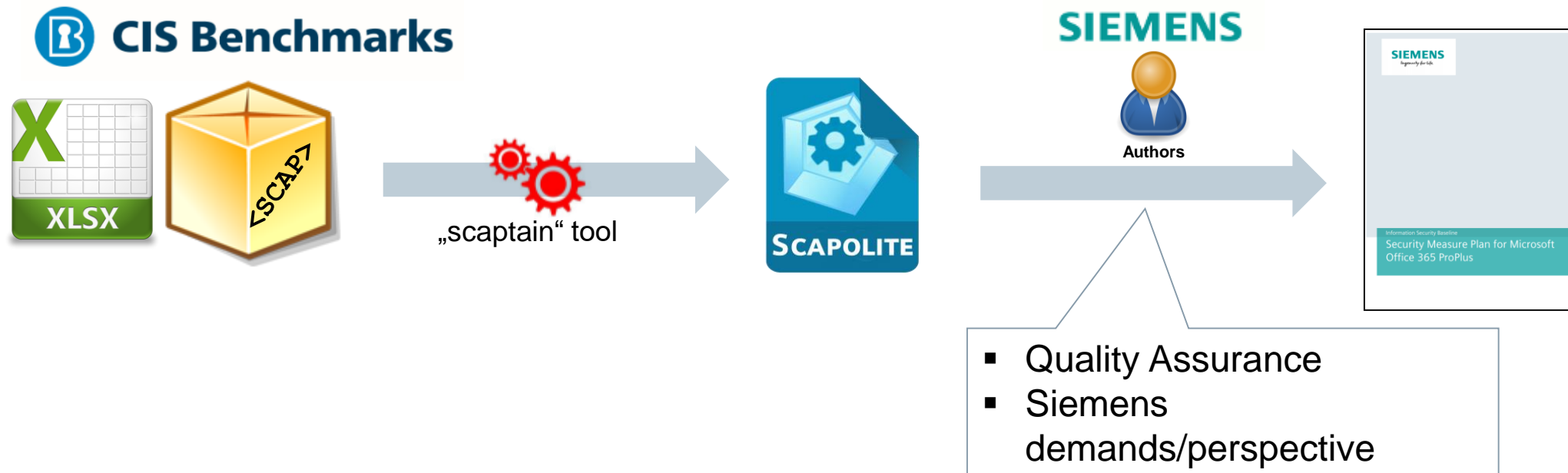
        - Check whether people are doing it
        - If not: **shout** at them
        - Repeat
```

Substructures with machine-readable automations of an implementation can be added to an implementation or a check.

# Class Diagram and exemplary file layout



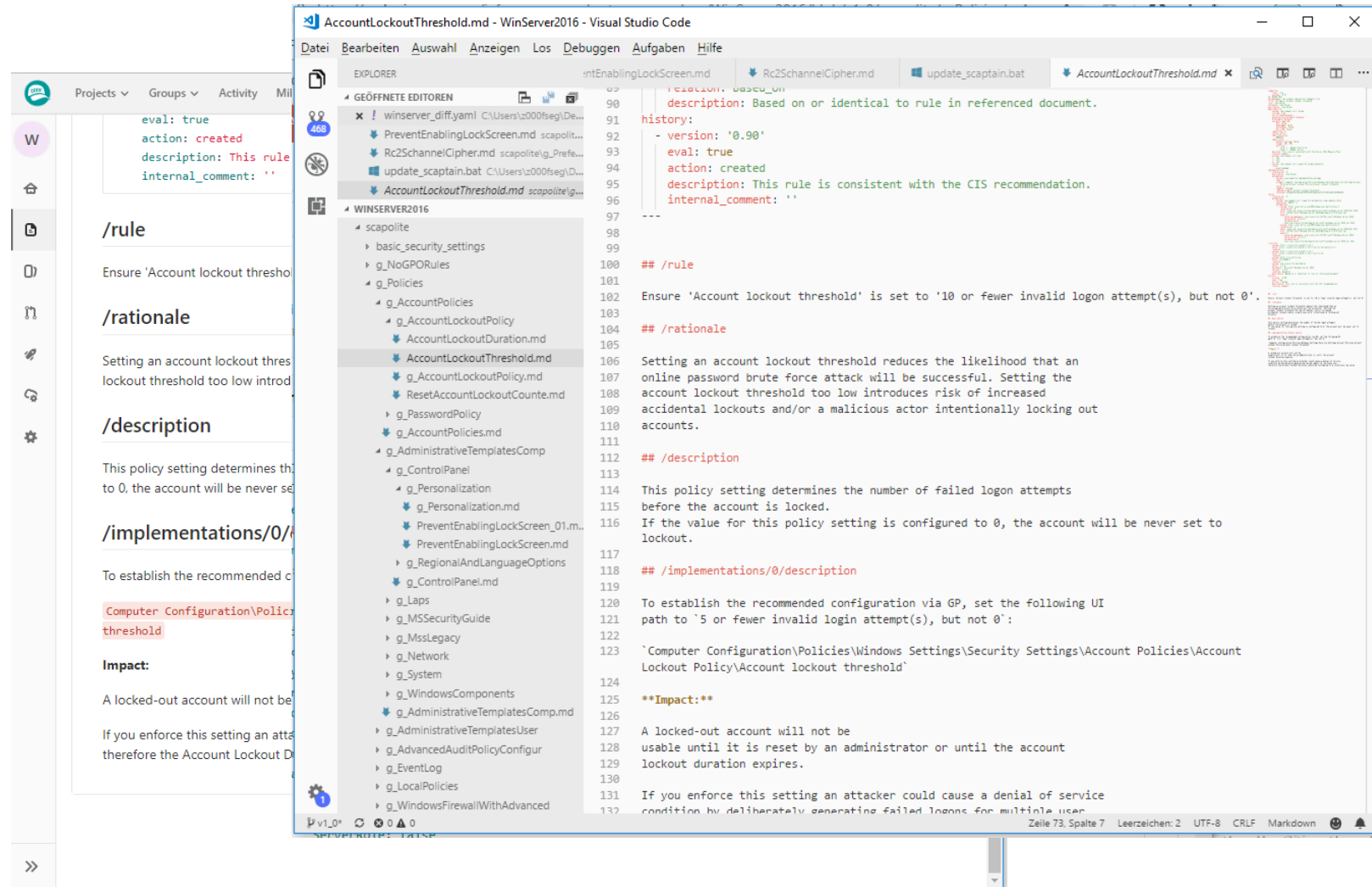
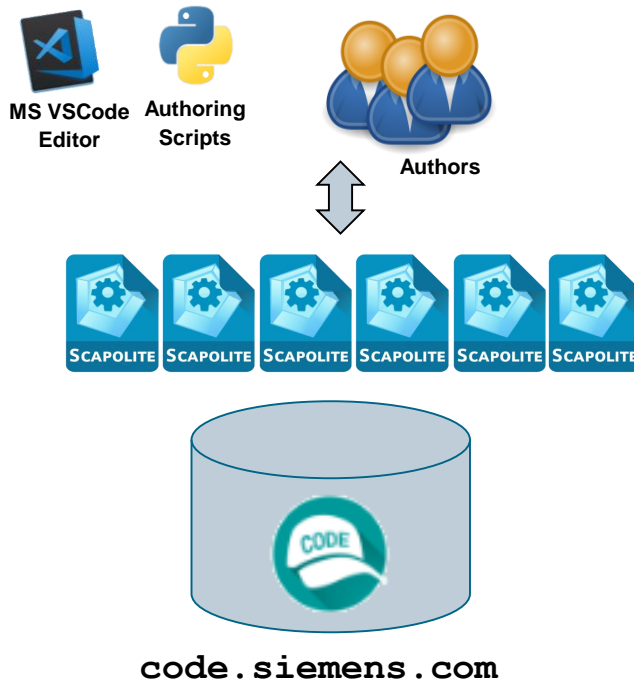
# Leveraging externally available benchmarks with Scapolite





# Scapolite: Taking a leaf out of the book of managing code

**SIEMENS**  
*Ingenuity for life*



# Example: Editing a Scapolite document

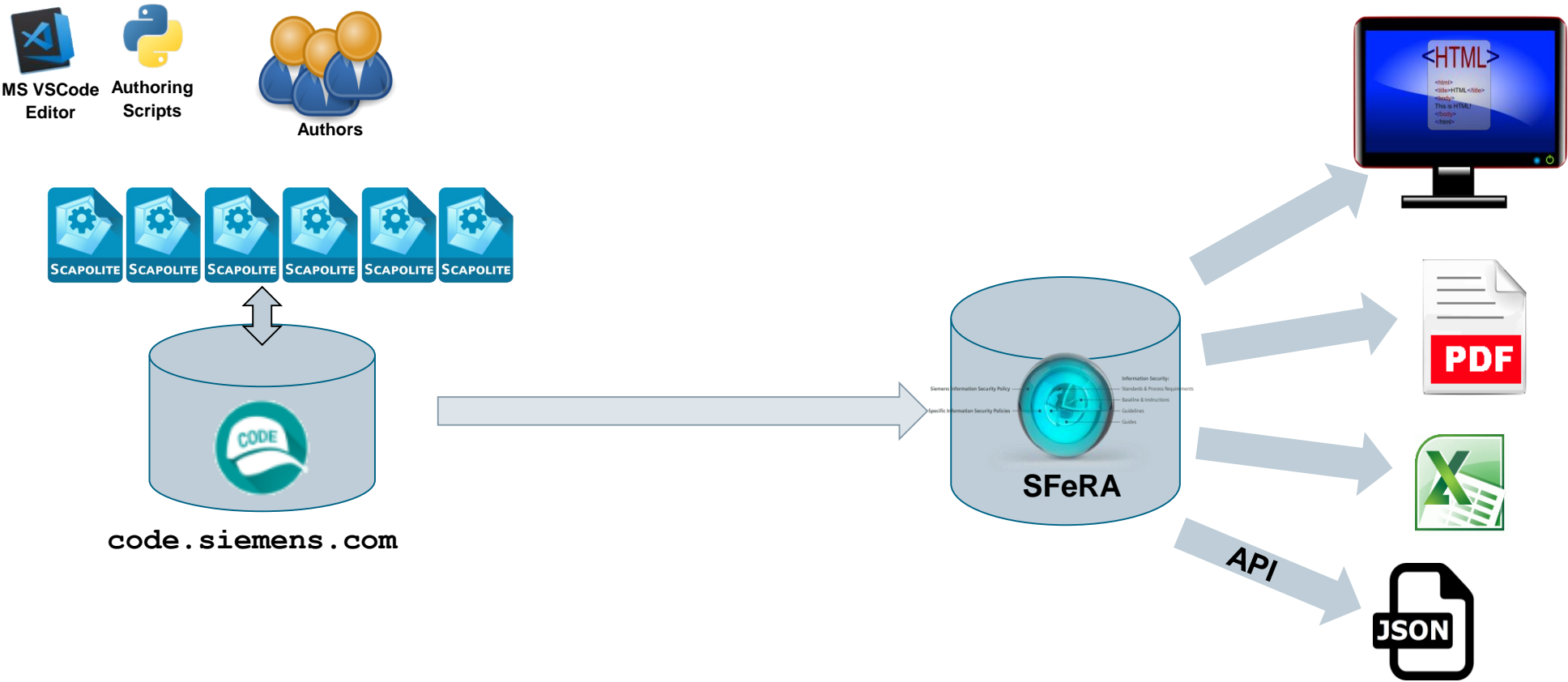
Factoring out Markdown content from YAML fields allows use of editors' /gitlab's preview feature and avoids problems with „double indentation“ of YAML & Markdown



```
20 c: '123'
21 i: '123'
22 a: '123'
23 - system: com.siemens.cert.scapolite.target_audience
24 roles:
25   - asset_manager
26 implementations:
27   - relative_id: '0'
28     description: <see below>
29     automations:
30       - system: org.scapolite.implementation.win_gpo
31         ui_path: Computer Configuration\Policies\Administrative Templates\Windows
32           Components\Windows Remote Management (WinRM)\WinRM Service\Allow Basic
33             authentication
34         value: Disabled
35         checksum: sha224:e84f74b3de46742661c1edb7b5ef772296876022696b8da39fc1c5a2
36         verification_status: Checked.
37 crossrefs:
38   - system: org.cisecurity.benchmarks
39     idref: 18.9.86.2.1
40     benchmark: Microsoft Windows Server 2016
41     version: 1.0.0
42     relation: based_on
43     description: Based on or identical to rule in referenced document.
44 history:
45   - version: '1.0'
46     eval: true
47     action: created
48     description: Rule created.
49 ---
50 ## /rule
51 Ensure 'Allow Basic authentication' is set to 'Disabled'.
52
53 ## /rationale
54 Basic authentication is less robust than other authentication methods
55 available in WinRM because credentials including passwords are
56 transmitted in plain text. An attacker who is able to capture packets on
57 the network where WinRM is running may be able to determine the
58 credentials used for accessing remote hosts via WinRM.
59
60 ## /description
61 This policy setting allows you to manage whether the Windows Remote
62 Management (WinRM) service accepts Basic authentication from a remote
63 client.
64
65 ## /implementations/0/description
66 To establish the recommended configuration via GP, set the following UI
67 path to 'Disabled':
68
69 'Computer Configuration\Policies\Administrative Templates\Windows Components\Windows
70 Remote Management (WinRM)\WinRM Service\Allow Basic authentication'
71
72 **Impact:**
73 The WinRM service will not accept Basic authentication from a remote client. \
74
```



# SFeRA: From plain text to many formats



SIEMENS

Ingenuity for life

B. Grobauer, Admin, Author

Contact

English

Logout

Search

Policies & Rules

Select Version

Reviews

Administration

About SFeRA

4

7

SFeRA

Security Framework and Regulations Application

Back

Normative

Informative

Full

Expand All

Security Measure Plan for Microsoft Windows Server 2016

Hardware Security

System Setup

Group Policies

Account Policies

Account Lockout Policy

Local Policies

Event Log

Windows Firewall With Advanced Security

Advanced Audit Policy Configuration

Administrative Templates (Computer)

Administrative Templates (User)

Group Policy Preferences (GPP)

Rules that are not based on Group Policies (GPOs) or Registry Settings (GPP)

Group Policies

Account Policies

Password Policy

BL968-3756

Asset Manager

C

I

A

123

123

123

Configure 'Password must meet complexity requirements'

Rule

Ensure 'Password must meet complexity requirements' is set to 'Enabled'.

Description

This policy setting checks all new passwords to ensure that they meet basic requirements for strong passwords.

When this policy is enabled, passwords must meet the following minimum requirements:

Not contain the user's account name or parts of the user's full name that exceed two consecutive characters

Contain characters from three of the following four categories:

English uppercase characters (A through Z)

English lowercase characters (a through z)

Base 10 digits (0 through 9)

Non-alphabetic characters (for example, !, \$, #, %)

Rationale

Passwords that contain only alphanumeric characters are extremely easy to discover with several publicly available tools.

IO - Implementation Example

To establish the recommended configuration via GP, set the following UI path to **Enabled** :

Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies>Password Policy>Password must meet com requirements

Impact:

None - This rule has been enabled for a long time.

Crossreferences

Based on CIS Benchmark 'Microsoft Windows Server 2016' (v1.0.0): 1.1.5

Version History

1.0 (2019-03-15): created (EVAL until 2019-07-31)

This rule is consistent with the CIS recommendation.

SIEMENS

Ingenuity for life

B. Grobauer, Admin, Author

Contact

English

Logout

Search

Policies & Rules

Select Version

Reviews

Administration

About SFeRA

4

7

SFeRA

Security Framework and Regulations Application

Download Result List

Search Term

Target Audiences

ACP Level

Documents

Document Type

ISO 27001 Domain

Only rules under evaluation

Search

Your choice:

password

Asset Manager

Security Measure Plan for Microsoft Windows Server 2016

Confidentiality - Level 1

Integrity - Level 1

Availability - Level 1

Clear all

Expand All

Access and query Scapolite objects

Title

Access Scapolite objects by identifier

Request

GET

BASE URL

REVISION

/object/{id\_namespace}:{id}

Description

Note

Scapolite objects may contain other Scapolite objects, e.g., a Scapolite collection may contain groups and rules; groups may contain sub-groups and rules. When authoring a Scapolite object, the author may chose to inline contained objects rather than to reference these objects.

API access to an object will always result in a representation that references other objects rather than inlining these objects!

Parameters

Name	Type	Datatype	Status	Default	Description
id_namespace	URL component	string (with syntax as specified in Scapolite standard)	REQUIRED	n/a	The id_namespace of a Scapolite object
id	URL component	string (with syntax as specified in Scapolite standard)	REQUIRED	n/a	The identifier of a Scapolite object
format	query parameter	list of values	OPTIONAL	json	Governs the format in which the result is returned. The following values are supported: <div><div>• json</div><div>• yaml</div></div>

Sample

GET

BASE URL

REVISION

/object/com.siemens.seg.policy\_framework.rule:BL112-4711?format=yaml

yields

Collapse source

```
1 ---
2 scapolite:
3   class: rule
4   version: '1.00'
5   id: BL112-4711
6   id_namespace: com.siemens.seg.policy_framework.rule
7   title: rule title
8   rule: rule text
9   rationale: rule rationale
10  description: rule description
11  applicability:
12    - system: com.siemens.cert.scapolite.target_audience
```

© Siemens AG 2019

Page 10

April 2019, v1.0

Corporate Technology

# Using Gitlab's CI features for DevOps-inspired approach towards maintaining security baselines



The screenshot displays the GitLab web interface for a project named 'Office2016'. The left sidebar contains navigation links for Project, Repository, Issues, Merge Requests, CI / CD (highlighted), Pipelines, Jobs, Schedules, Charts, Operations, and Settings. The main content area shows the 'Pipelines' view for the 'Office2016' project. A specific pipeline, #2570170, is highlighted, indicating it has 'passed'. The pipeline was triggered by Bernd Grobauer. The title of the pipeline is 'Merging development into master'. Below the title, it shows '6 jobs from master' and a 'latest' tag. The pipeline ID is 'f589c9d2'. The pipeline jobs are visualized in a flowchart:

- Setup**: setup (status: passed)
- Gpo\_generation**: gpo\_automations (status: passed)
- Build**:
  - Overview\_xlsx (status: passed)
  - Powershell\_Scri... (status: passed)
  - Qualys\_Policies (status: passed)
- Teardown**: cleanup (status: passed)

- There is huge demand for machine-readable security baselines, yet it seems that most organizations merely consume SCAP content by one of the „big three“ (IASE, CIS, OpenSCAP) rather than producing their own SCAP content
- One probable reason: authoring and maintaining content in „SCAP proper“ is almost impossible
- Proposed solution: SCAP v2 must define standard formats that truly allow a „security-as-code“ approach
- Internal usage of Scapolite for all new IS Policies published within Siemens in the past 1.5 years shows that Scapolite is a format that supports the „security-as-code“ approach

(If there is interest, it might be possible for us to publish Scapolite (e.g., as IETF RfC) and (2) provide central parts of scaptain as open-source code.)

- SCAP v2 should also try to put more focus on automating also the implementation rather only the check (more about this topic in tomorrow's talk by Patrick Stöckle)