

Version 1.0
April 2019

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

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

Dr. Bernd Grobauer, Siemens CT RDA ITS

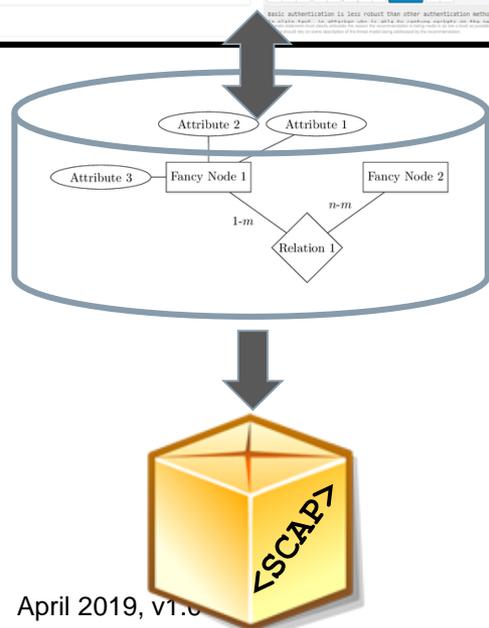
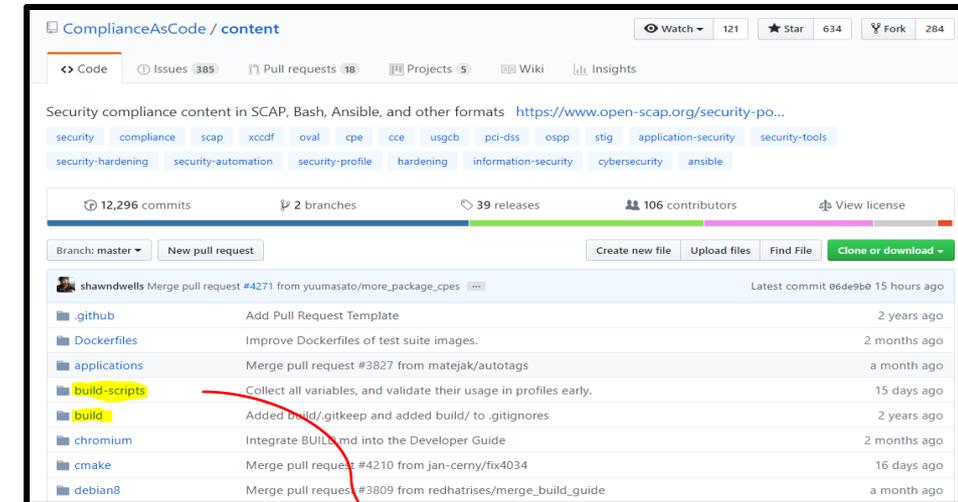
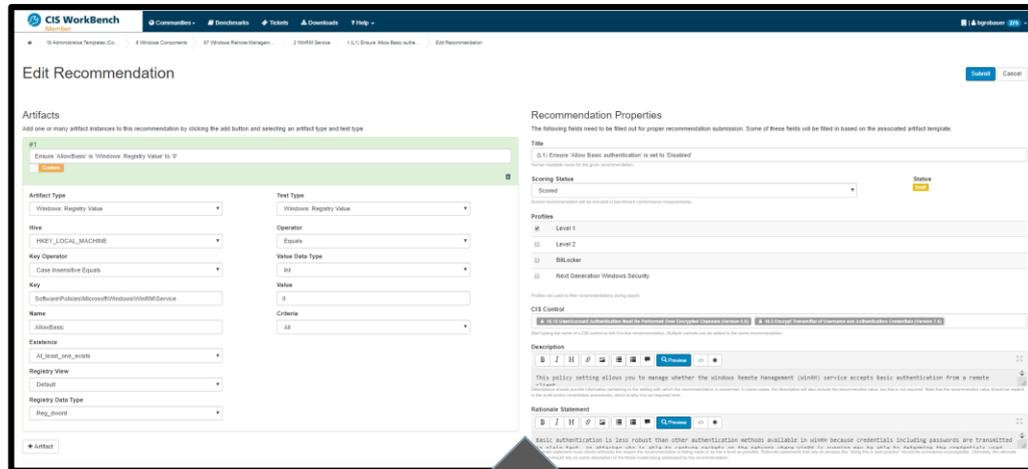


Two (and *only* two) publishers of SCAP content that support community-driven authoring and maintenance

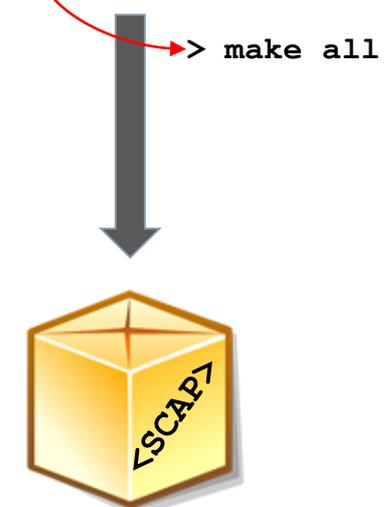


CIS

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 noticeable, 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.

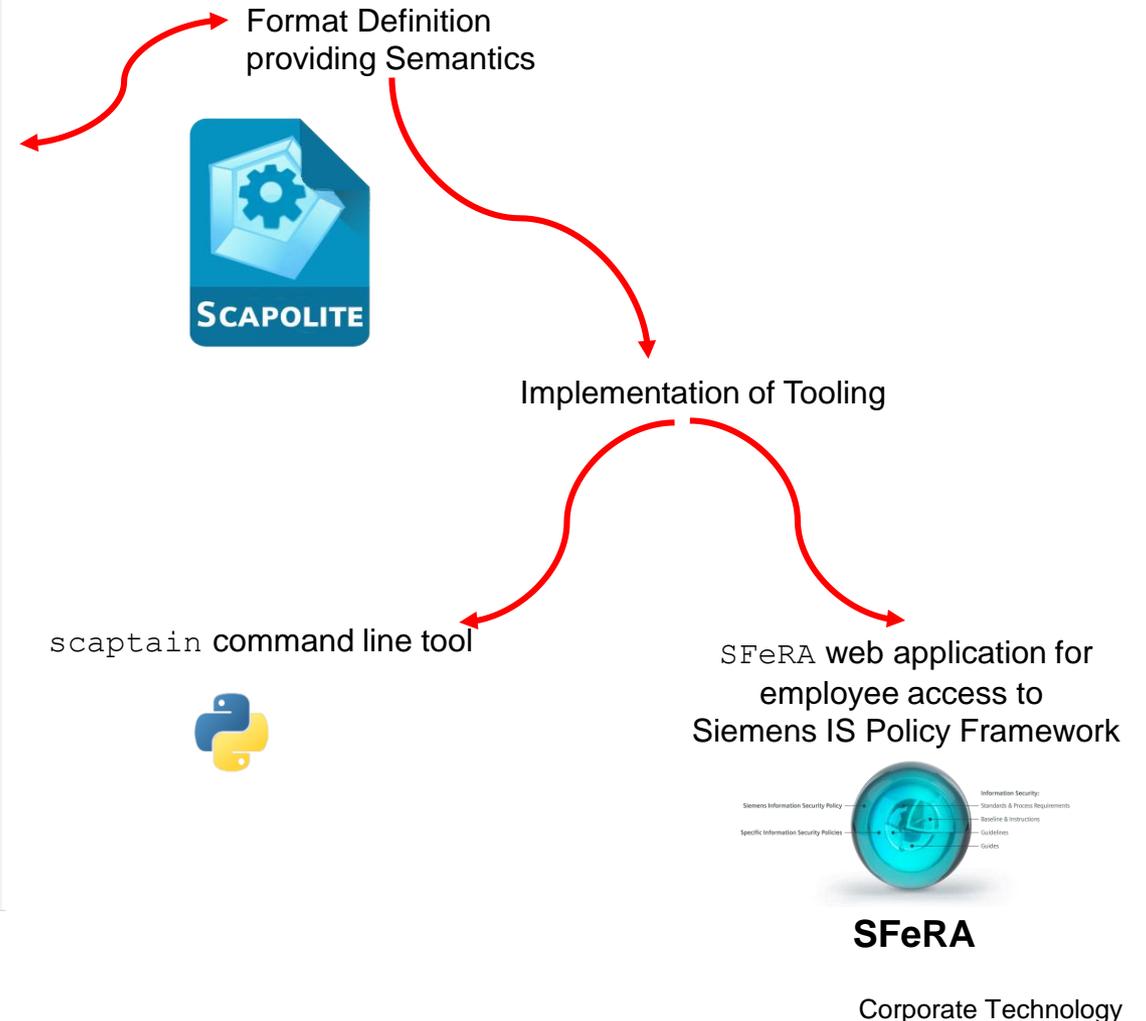
The most promising approach towards writing SCAP standard has been developed by the OpenSCAP project: SCAP content is divided into pieces of simplified XML stored in single files. Collaborative editing is done using git as version control system – SCAP content is then produced by a combination of scripts 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" obviously draws inspiration from the SCAP (SCAP_1_2) standard. Scapolite's content is



Scapolite Rule Example

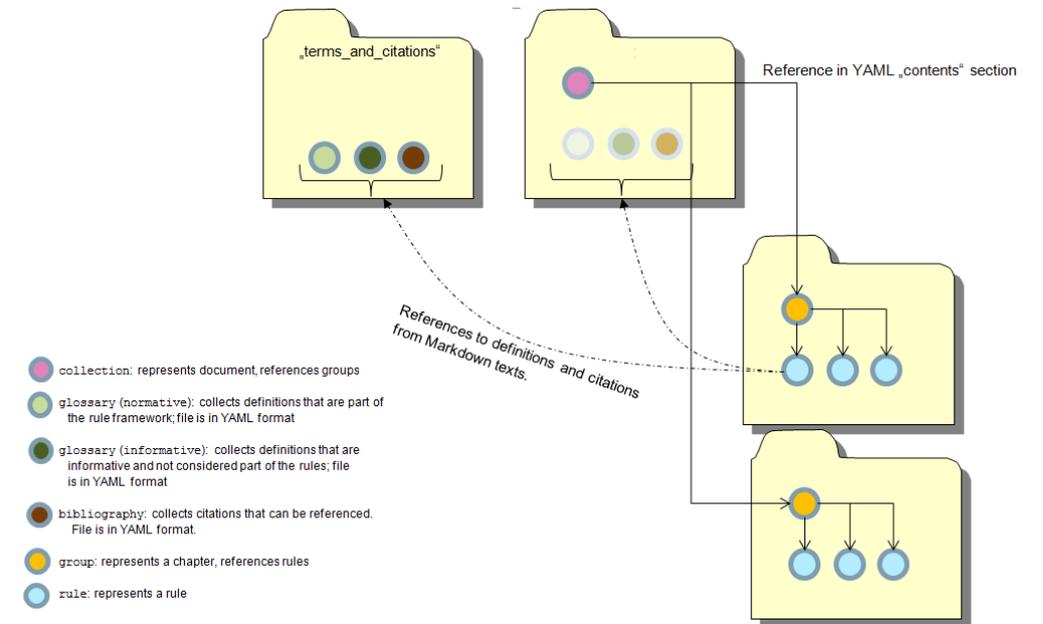
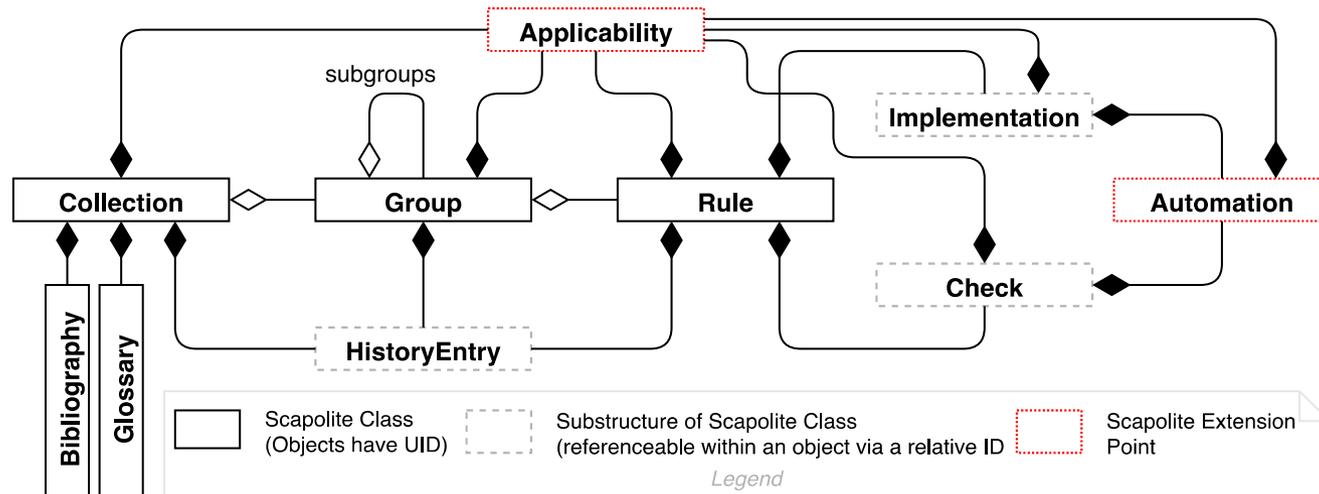
Introductory Example taken from Scapolite Documentation



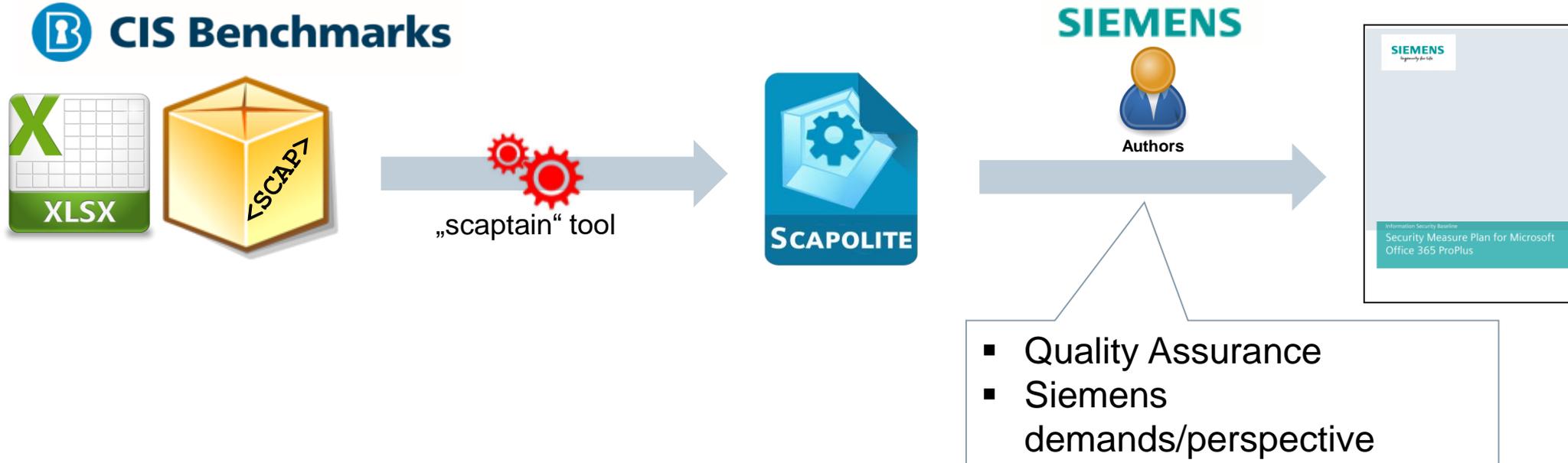
```
scapolite:
  class: rule
  version: '1.0'
  id: C085611074
  id_namespace: com.siemens.cert.scapolite.example_benchmark
  title: An example rule
  rule: Do as I say, not as I do.
  rationale: |
    There are always example of policy/rule makers who do not conform to their own
    rules. Nevertheless, 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.

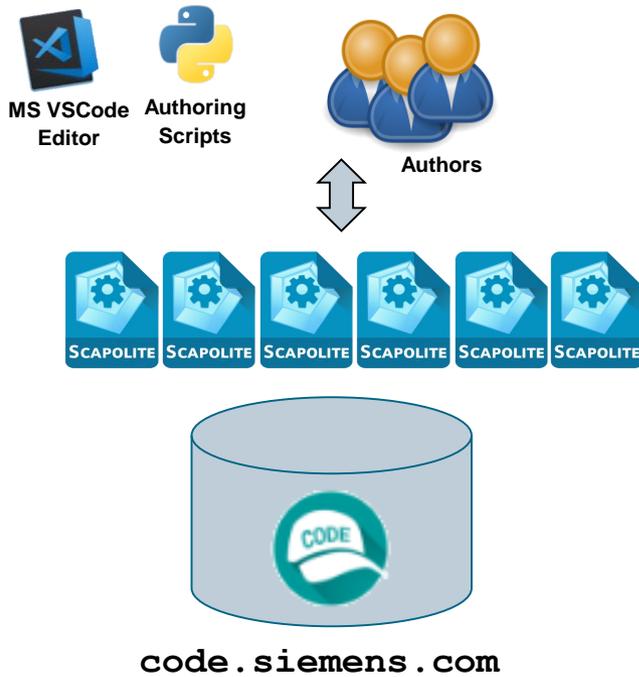
Class Diagram and exemplary file layout



Leveraging externally available benchmarks with Scapolite



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



The screenshot shows the Visual Studio Code interface with a Scapolite rule configuration for 'AccountLockoutThreshold.md'. The left sidebar displays the rule's metadata, including its title, rationale, description, and impact. The main editor shows the rule's implementation in a YAML-like format.

```
eval: true
action: created
description: This rule
internal_comment: ''

/rule
Ensure 'Account lockout thresho

/rationale
Setting an account lockout thres
lockout threshold too low introd

/description
This policy setting determines th
to 0, the account will be never se

/implementations/0/
To establish the recommended c
Computer Configuration\Polic
threshold

Impact:
A locked-out account will not be
If you enforce this setting an att
therefore the Account Lockout D
```

```
90
91
92
93
94
95
96
97
98
99
100 ## /rule
101
102 Ensure 'Account lockout threshold' is set to '10 or fewer invalid logon attempt(s), but not 0'.
103
104 ## /rationale
105
106 Setting an account lockout threshold reduces the likelihood that an
107 online password brute force attack will be successful. Setting the
108 account lockout threshold too low introduces risk of increased
109 accidental lockouts and/or a malicious actor intentionally locking out
110 accounts.
111
112 ## /description
113
114 This policy setting determines the number of failed logon attempts
115 before the account is locked.
116 If the value for this policy setting is configured to 0, the account will be never set to
117 lockout.
118
119 ## /implementations/0/description
120
121 To establish the recommended configuration via GP, set the following UI
122 path to '5 or fewer invalid login attempt(s), but not 0':
123
124 'Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Account
125 Lockout Policy\Account lockout threshold'
126
127 **Impact:**
128
129 A locked-out account will not be
130 usable until it is reset by an administrator or until the account
131 lockout duration expires.
132
133 If you enforce this setting an attacker could cause a denial of service
134 condition by deliberately generating failed logons for multiple user
```

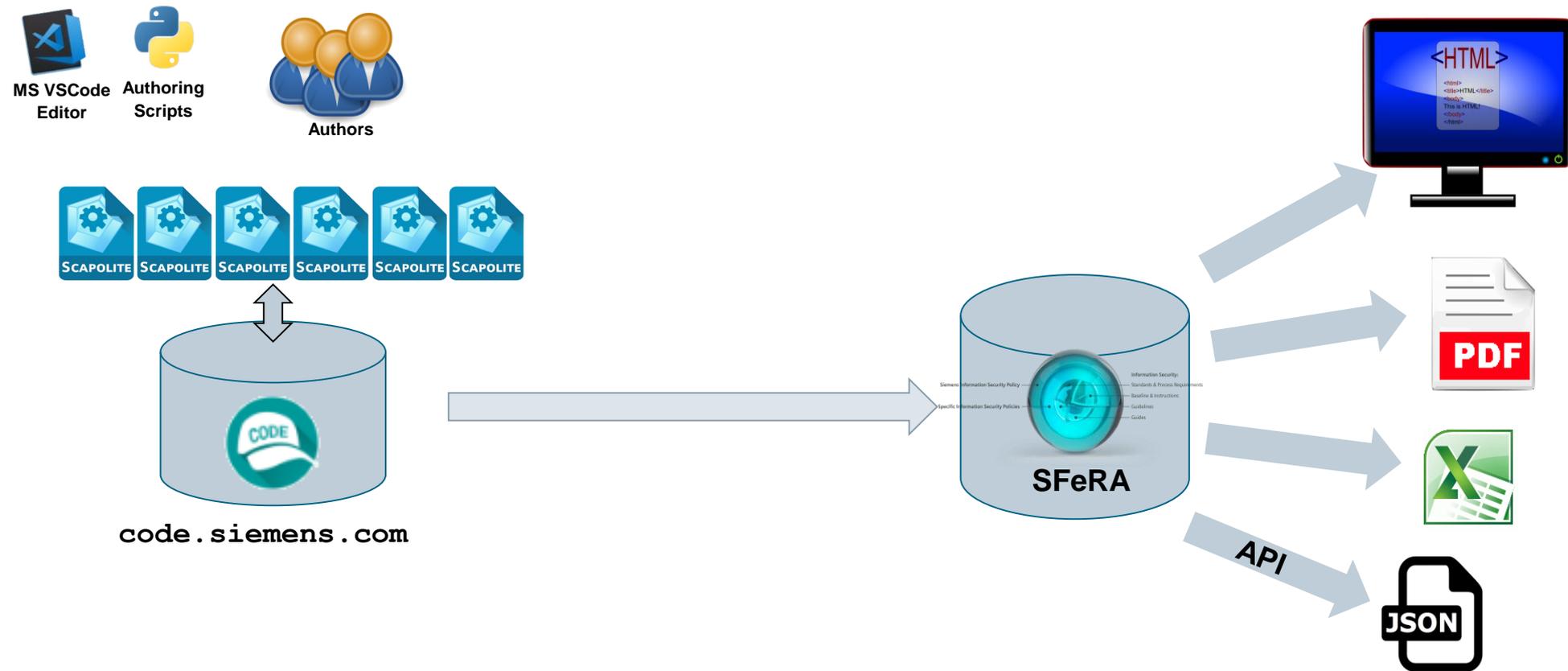
Example: Editing a Scapolite document

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 ## /rationale
53 Basic authentication is less robust than other authentication methods
54 available in WinRM because credentials including passwords are
55 transmitted in plain text. An attacker who is able to capture packets on
56 the network where WinRM is running may be able to determine the
57 credentials used for accessing remote hosts via WinRM.
58 ## /description
59 This policy setting allows you to manage whether the Windows Remote
60 Management (WinRM) service accepts Basic authentication from a remote
61 client.
62 ## /implementations/0/description
63 To establish the recommended configuration via GP, set the following UI
64 path to 'Disabled':
65 `Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service\Allow Basic authentication`
66 **Impact:**
67 The WinRM service will not accept Basic authentication from a remote client. \
68 This is the default configuration.
```

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

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
 - Password Policy**
 - 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 [EVAL until 2019-07-31] [C:1 2 3] [I:1 2 3] [A:1 2 3]

Asset Manager

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.

10 - 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 complexity 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

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 x Asset Manager x Security Measure Plan for Microsoft Windows Server 2016 x Confidentiality - Level 1 x Integrity - Level 1 x Availability - Level 1 x Clear all

Access and query Scapolite objects

Title Access Scapolite objects by identifier

Request GET | BASE URL | REVISION /object/{id_namespace}<id>

Description Access the representation of an object in the Scapolite standard.

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!

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: <ul style="list-style-type: none"> json yaml

Sample GET | BASE URL | REVISION /object/com.siemens.seg.policy_framework.rule:BL112-4711?format=yaml

yields

```

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
    
```

urity

[C:1 2 3] [I:1 2 3] [A:1 2 3]

f the Specific Information Security Policy: Acces ... more

urity

[C:1 2 3] [I:1 2 3] [A:1 2 3]

requirements of the Specific Information Securi ... more

urity

[C:1 2 3] [I:1 2 3] [A:1 2 3]

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



The screenshot displays the GitLab CI/CD interface for a project named 'Office2016'. The top navigation bar includes 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. A search bar and user profile are on the right. The left sidebar shows navigation options: Project, Repository, Issues (0), Merge Requests (0), CI / CD, Pipelines, Jobs, Schedules, Charts, Operations, and Settings.

The main content area shows the pipeline details for 'Merging development into master'. It indicates that the pipeline #2570170 was triggered by Bernd Grobauer and has passed. Below this, it shows '6 jobs from master' and a 'latest' tag. A commit hash 'f589c9d2' is also visible.

The pipeline job graph consists of four stages: Setup, Gpo_generation, Build, and Teardown. Each stage contains one or more jobs, all of which are marked as successful with green checkmarks:

- Setup**: setup
- Gpo_generation**: gpo_automations
- Build**: Overview_xlsx, Powershell_Scri..., Qualys_Policies
- Teardown**: cleanup

- 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)