Sicherheitskonfigurationsrichtlinien effizient verwalten und umsetzen: Der Scapolite-Ansatz

Patrick Stöckle
patrick.stoeckle@tum.de
Lst. f. Software und Systems Engineering
Technische Universität München (TUM)

Bernd Grobauer
bernd.grobauer@siemens.com
T CST
Siemens AG

Alexander Pretschner
alexander.pretschner@tum.de
Lst. f. Software und Systems Engineering
Technische Universität München (TUM)

München, 03.02.2022
29. DFN-Konferenz „Sicherheit in vernetzten Systemen”

Systemhärtung mit Sicherheitskonfigurationsrichtlinien

Unsere Motivation

Grundproblem

Standardeinstellungen von Software sind nicht sicher

- Funktionen und Komfort meistens wichtiger als Sicherheit
- Interessenskonflikte
  - *Hersteller will wissen, was wir tun, um Produkte zu verbessern ⇒ Problematisch*

Lösung

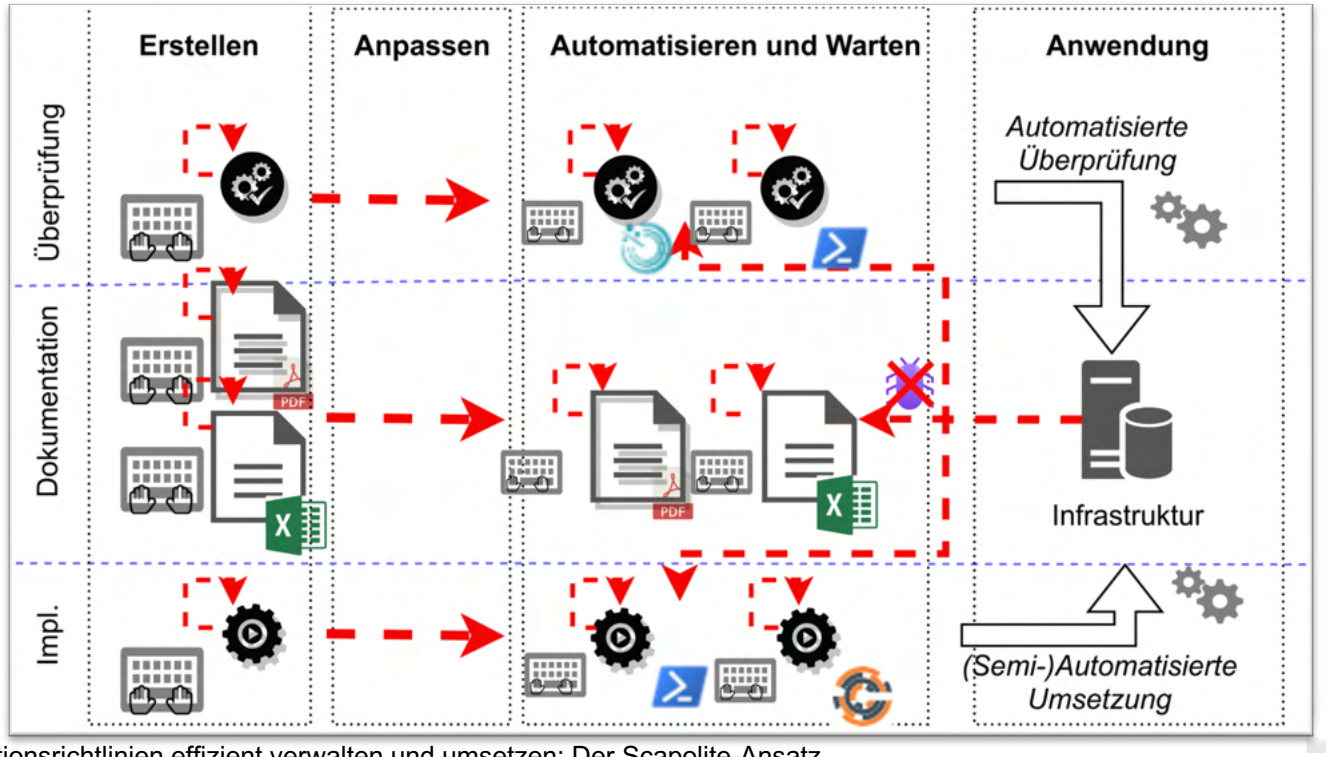
Wir müssen Geräte härten!

Z.B. mit Richtlinien von

- Center for Internet Security (CIS)
- Defense Information Systems Agency (DISA)
- Bundesamt für Sicherheit in der Informationstechnik (BSI)

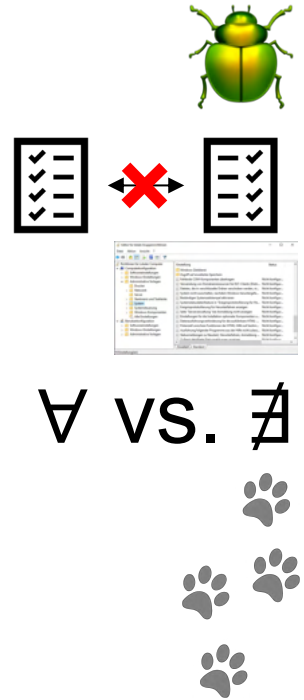
Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Status Quo


Erstellen
Anpassen
Automatisieren und Warten
Anwendung
Überprüfung
Dokumentation
Impl.
Automatisierte Überprüfung
Infrastruktur
(Semi-)Automatisierte Umsetzung

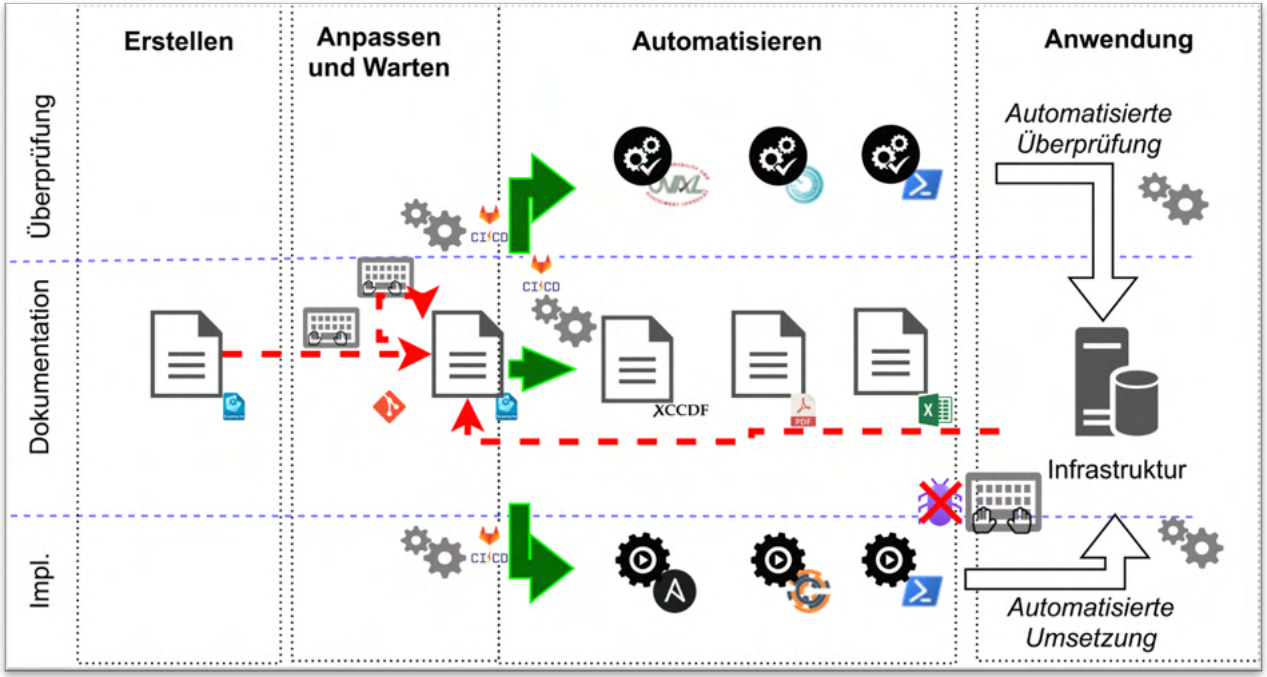
Probleme

a) Fehler beim Anpassen der Richtlinien
   Eine Einstellung soll konfiguriert werden, die es nicht gibt
b) Inkonsistenzen zwischen den Dokumenten
   Check fordert *Aktiviert,* Umsetzung setzt *Deaktiviert*
c) Regeln nur über grafische Nutzeroberfläche umsetzbar
   Wir können viele Windows Regeln nur über Gruppenrichtlinien-Editor umsetzen
d) Ganz oder gar nicht
   Wir können ein Richtlinienbackup einspielen, aber nicht einzelne Regeln.
e) Nachvollziehbarkeit von Änderungen
   Schlechtes Changelog ⟹ Wir vergleichen aktualisierte Richtlinie manuell mit der vorherigen Version ⟹ Zeitaufwändig



Scapolite-Richtlinie verfügbar unter https://go.tum.de/796398

Lösung



Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Lösung

Automatisierte Überprüfung der Regeln auf Semantische Korrektheit

- Wir überprüfen geänderte Regeln automatisiert auf ihre semantische Korrektheit
- Existiert die Regel? Ist der Wert legal? In was müssen wir den Wert übersetzen?

Regel-zentrierte Verwaltung

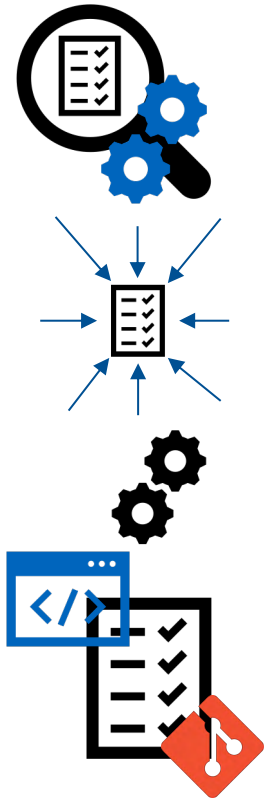
- Jede Regel in einem Dokument
- Im Scapolite-Format ist alle Information an einem Ort

Fokus auf Automatisierung

- Wir können alle Regeln per Kommandozeile umsetzen und überprüfen
- Wir können sowohl Regellisten als auch einzelne Regeln umsetzen

Richtlinien-as-Code

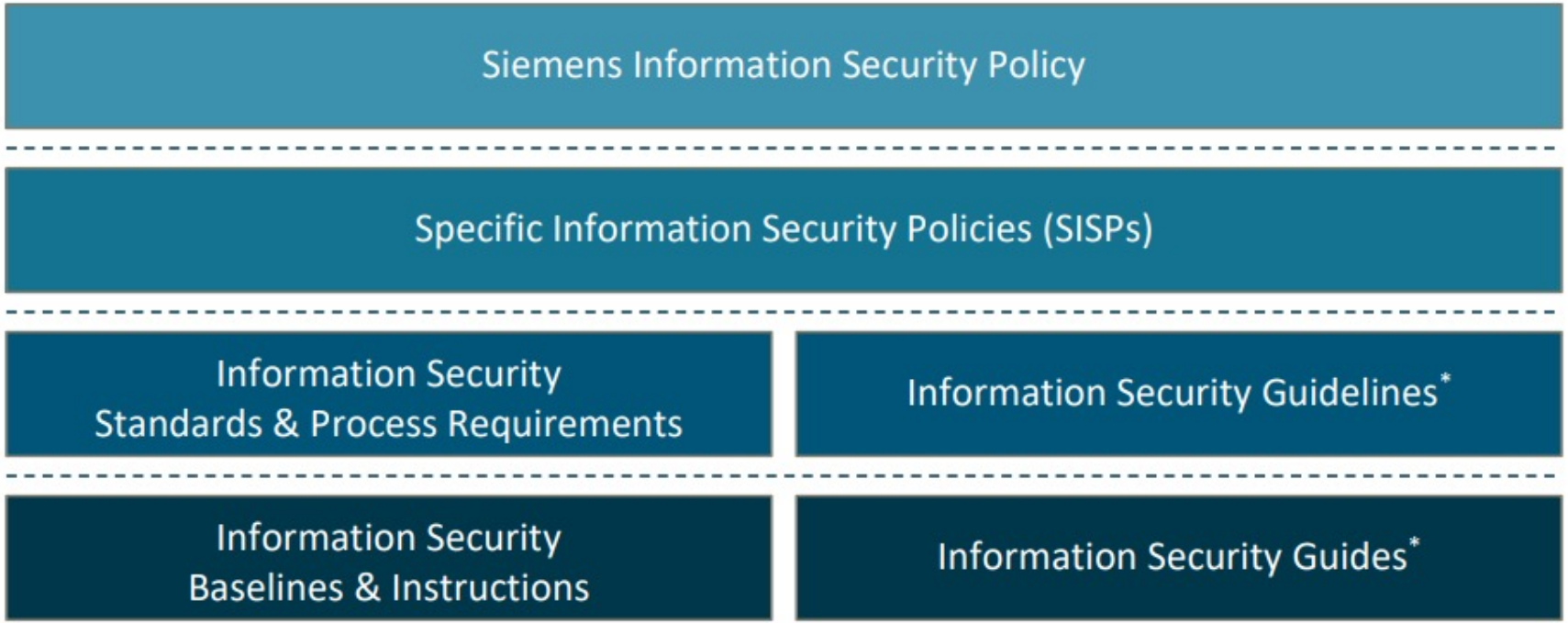
- Scapolite-Format (Markdown) sehr gut für Versionsverwaltung
- Zweige (Branches) bilden verschiedene Versionen des Systems/der Richtlinie ab
- CI läuft erzeugt Artefakte für jeden Branch



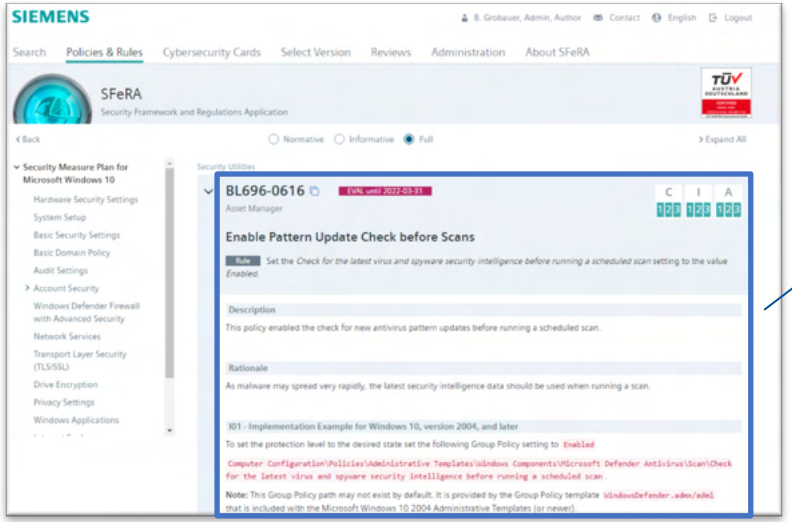
Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

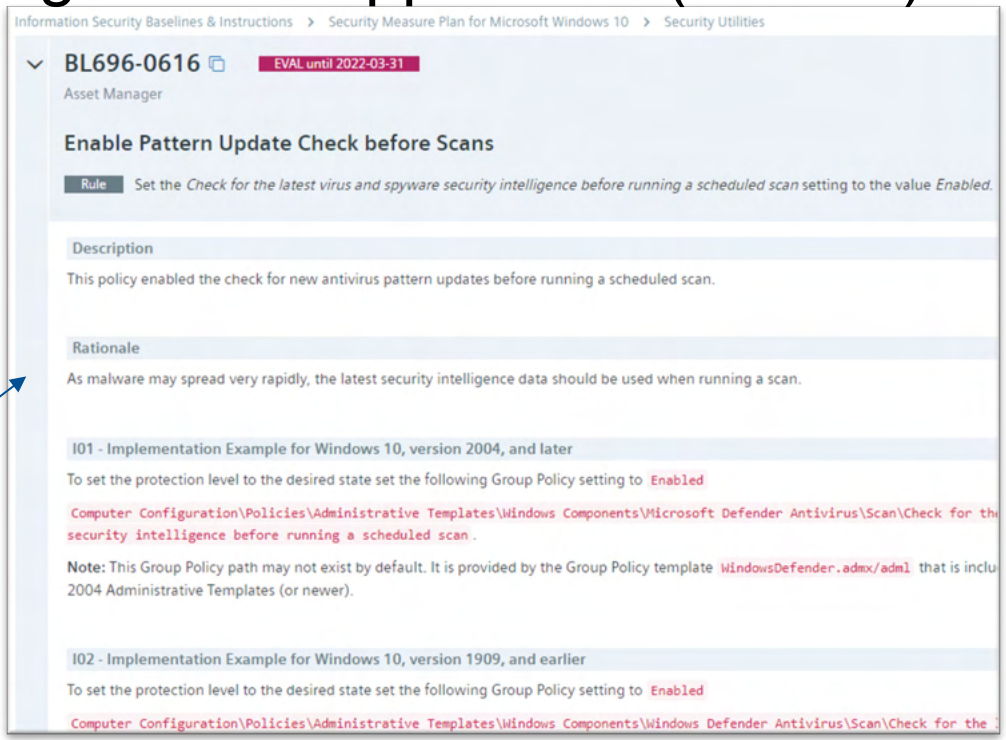
Fallstudie@Siemens

Siemens Security Information Policy Framework



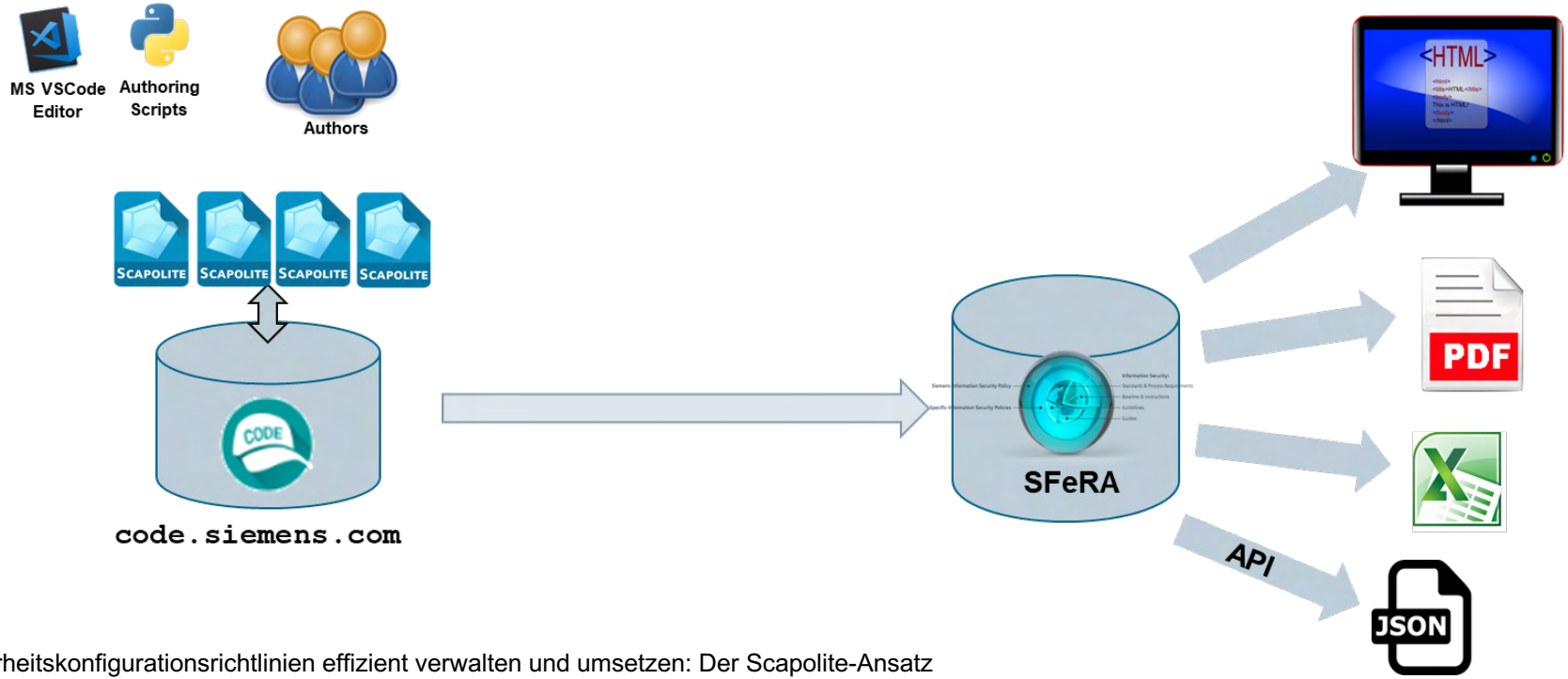
Security Framework and Regulations Application (SFeRA)





Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

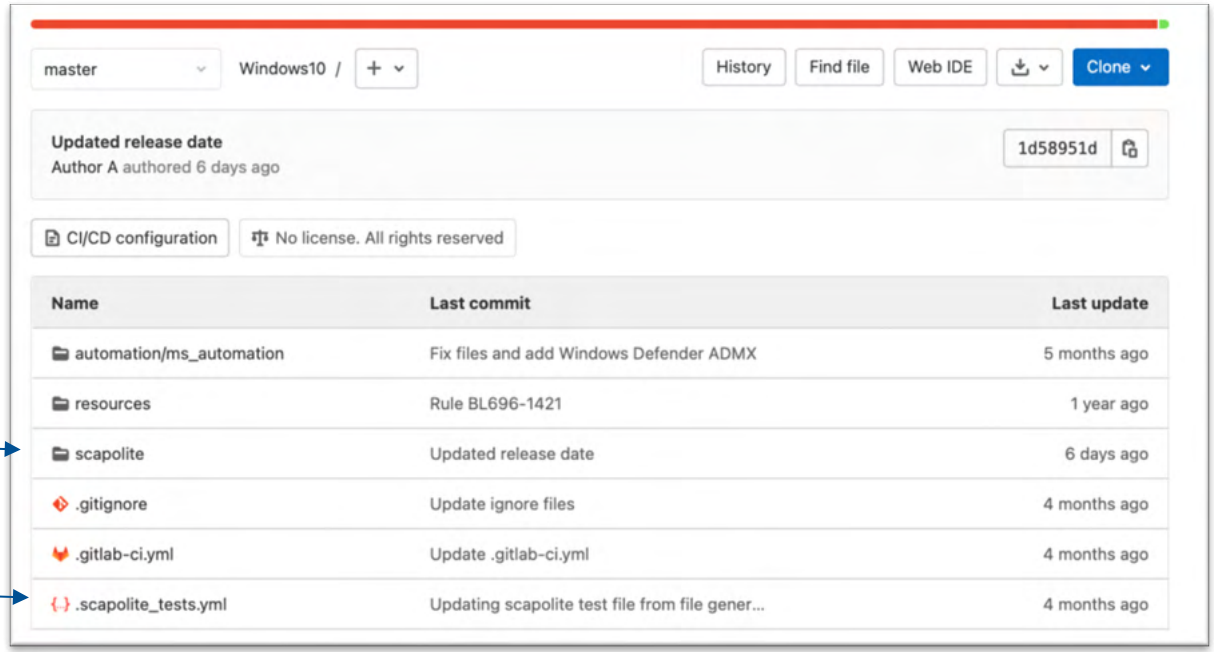
SFeRA Content Development: Scapolite in git



Sicherheitskonfigurationsrichtlinien effizient verwalten und umsetzen: Der Scapolite-Ansatz
Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Windows 10 Security Baseline

Auf *code.siemens.com*
(Siemens GitLab-Instanz)



Scapolite Quellen →

Test Spezifikation →

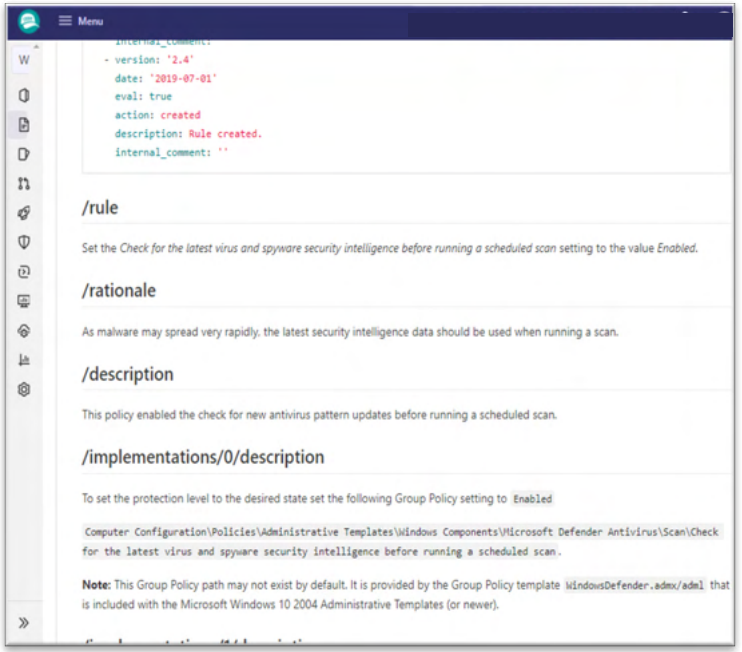
Scapolite Beispiel-Regel: Markdown mit YAML-Preambel

```yaml
---
scapolite:
  class: rule
  version: '1.0'
id: C0815-1074
id_namespace: com.siemens.scapolite.example_benchmark
applicability:
 - system: com.siemens.cert.acp
   c: '123'
   i: '123'
   a: '123'
 - system: com.siemens.cert.scapolite.target_audience
   roles:
     - asset_manager
title: An example rule
rule: Do as I say, not as I do.
rationale: <see below>
implementations:
  - relative_id: '01'
    title: Just do it yourself
    description: <see below>
  - relative_id: '02'
    title: Get people to do it
    description: <see below (second_implementation_description)>
history:
 - version: '1.0'
   eval: true
   action: created
   description: 'Created example rule.'
---
```

```markdown
## /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/0/description

Carry out the following steps:

- Do this
- Do that

## second_implementation_description

Carry out the following steps:

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

Eine "echte" Regel aus der Windows 10 Baseline

```yaml
---
scapolite:
  class: rule
  version: '0.51'
id: BL696-0616
id_namespace:
  ↪  com.siemens.seg.policy_framework.rule
title: Enable Pattern Update Check before Scans
rule: <see below>
rationale: <see below>
description: <see below>
applicability:
- system: siemens.acp
  c: '123'
  i: '123'
  a: '123'
- system: siemens.scapolite.target_audience
  roles:
  - asset_manager
implementations:
- relative_id: '01'
  description: <see below>
  automations:
  - system: org.scapolite.implementation.win_gpo
    ui_path: Computer
      ↪  Configuration\Policies\Administrative
      ↪  Templates\Windows Components\Windows
      ↪  Defender Antivirus\Scan\Check for the
      ↪  latest virus and spyware security
      ↪  intelligence before running a
      ↪  scheduled scan
    value: Enabled
    verification_status: Checked.
...
history:
- version: '2.5'
  date: '2020-03-12'
  action: revised
  hide: true
  description: Added automation information to
    ↪  Yaml
  internal_comment: ''
...
---
## /rule
Set the _Check for the latest virus and spyware
  ↪  security intelligence before running a
  ↪  scheduled scan_ setting to the value
  ↪  _Enabled_.
## /rationale
As malware may spread very rapidly, the latest
  ↪  security intelligence data should be used
  ↪  when running a scan.
## /description
This policy enabled the check for new antivirus
  ↪  pattern updates before running a
  ↪  scheduled scan.
## /implementations/0/description
To set the protection level to the desired state
  ↪  set the following Group Policy setting to
  ↪  `Enabled`
`Computer Configuration\Policies\Administrative
  ↪  Templates\Windows Components\Windows
  ↪  Defender Antivirus\Scan\Check for the
  ↪  latest virus and spyware security
  ↪  intelligence before running a scheduled
  ↪  scan`.
```



Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Artikel verfügbar unter https://go.tum.de/796398

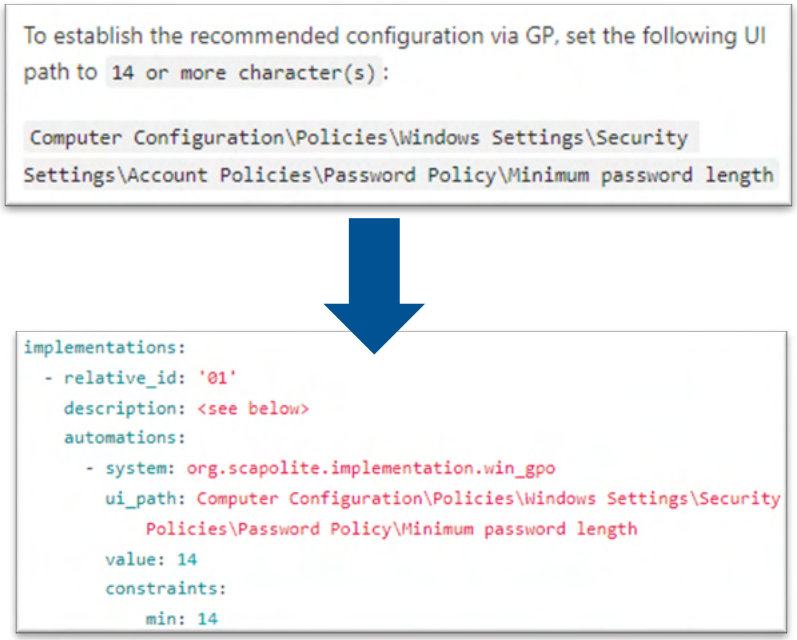
Exkurs: Effiziente Erzeugung von maschinenlesbaren Spezifikationen für Windows Security Baselines

Problem

- CIS/DISA Richtlinien enthalten nur maschinenlesbare Informationen für die Überprüfung
- Einstellung definiert in GPO-Policy (Policy-Pfad und Wert)

Lösung

- CIS/DISA beschreiben Einstellungen sehr schematisch ⇒ mit Natural-Language-Processing können wir eine Vorlage erstellen, die in ca. 80% der Fälle schon richtig ist
- Mittlerweile auch zulässige Wertintervalle erkannt und maschinenlesbar für die Überprüfung spezifiziert



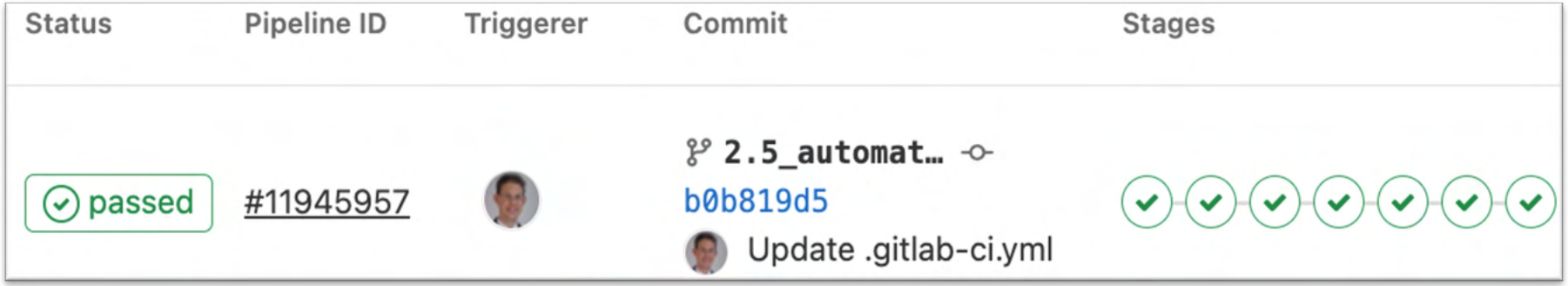
Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

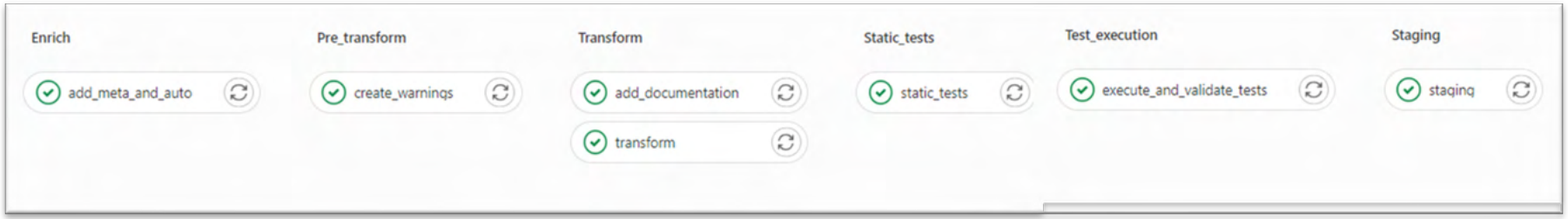
Letztendliches Ziel: Generierung von Automatisierungen


```
implementations:
- relative_id: '01'
  description: <see below>
  automations:
    - system: org.scapolite.implementation.win_gpo
      ui_path: Computer
      Configuration\Policies\Administrative
      Templates\Windows Components\Microsoft
      Defender Antivirus\Scan\Check for the latest
      virus and spyware security intelligence
      before running a scheduled scan
      value: Enabled
```

```
"BL696-0616": {
    "action": "DWORD:1",
    "config": "Computer",
    "path": "Software\\Policies\\Microsoft\\Windows Defender\\Scan",
    "rule_name": "BL696-0616",
    "rule_type": "pol",
    "title": "Enable Pattern Update Check before Scans",
    "type": "DWORD",
    "value": 1,
    "value_name": "CheckForSignaturesBeforeRunningScan",
    "acp": "C:123|I:123|A:123"
},
```

Pipelines erzeugen & testen Skripte/Artefakte





Pipeline befüllt ein Artefakt-Repository mit Resultaten

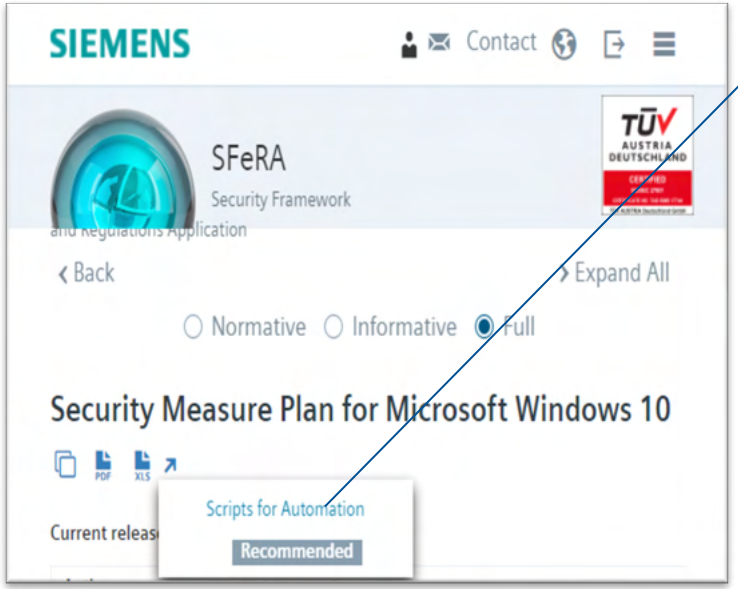
Script Framework + Documentation

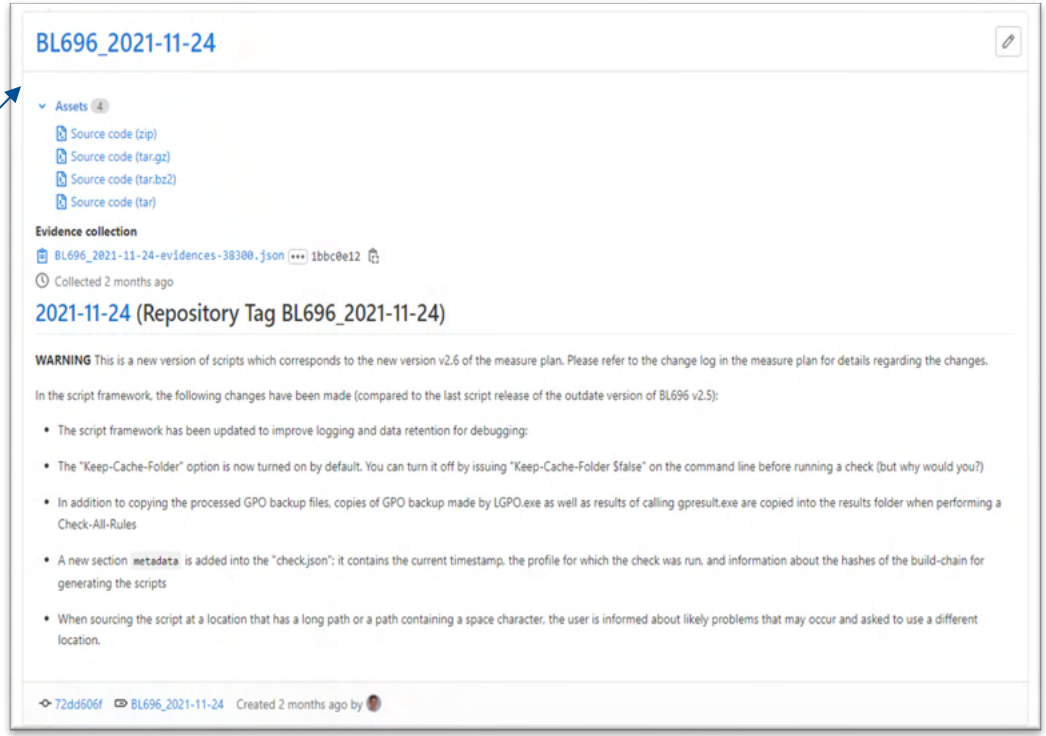
```
BL696_automation.xlsx
CHANGES.md
README.md
README.pdf
SFERA_JSON_Export_doc.md
SFERA_JSON_Export_doc.pdf

automation
    automation_script_library.ps1
    CHANGES.md
    custom_scripts.ps1
    name_to_sid.json
    README.md
    sfera_automation.json
    sfera_automation.ps1
    WARNING.txt
```

Test-Results and Test-Logs

```
.updated_scapolite_tests.yaml
README.txt
test_specification.yaml

outputSferaAutomationPipeline
    runner_logfile.log
    runner_log_file.log
    ws19-acp-333-ms
        000-initial-powershell-check
            check.json
            check.log

            validation
                000-initial_powershell_check.yaml

        010-initial-powershell-check-gpresult
            check.json
            check.log

            validation
                010-initial_powershell_check_gpresult.yaml
```

Bereitstellung als Release via code.siemens.com





Anwendung der Skripte

Umsetzung der Richtlinie

```
> Apply-All-Rules -profile acp111 .\sfera_automation.json
```

```
PS C:\Users\IEUser\Desktop\BL696\automation> Apply-All-Rules -profile acp111 .\sfera_automation.json
WARNING:
WARNING
=======

Applying the security settings may lead to problems in certain setups.
For example, settings may disable certain ways of connecting to a system,
remove authorizations required for certain operations, etc.

Either (1) apply the settings first in a dedicated setup for testing,
(2) acquaint yourself with rules and exclude possibly troublesome rules,
or (3) apply rules one-by-one rather than in bulk.

In case you cannot apply a rule temporarily or have been granted
an exception to the rule, you can supply a blacklist file to "Apply-All"
(see the script documentation).
Please read the warning above carefully. Continue? y/N: _
```

```
executing c:/lgpo/lgpo.exe

 Create folders and files for all rules...
    Time since start: 17 s, 91 % of creations completed!
    [ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo        ]

WARNING: Rule BL696-0786 has no Check script...
```


Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Anwendung der Skripte

Überprüfung der Richtlinien

```
# Compliant checks: 424  / 490
Compliant check ratio: 86.53%
# Non-compliant checks: 9  / 490
# Unknown checks: 57  / 490
# Empty checks: 49
# Blacklist rules: 0
# of automations which are disruptiv                    necking the real value: 0
# of automations which are disruptive and were marked as unknown without checking the real value: 0
# Compliant rules: 379  / 445
Compliant rule ratio: 85.17%
# Non-compliant rules: 9  / 445
# Unknown rules: 57  / 445
The results of the check process have been stored as JSON in the .\check.json
        compliant_rules_ratio       = 0.851685393258427
        length_all_rules            = 445
        length_compliant_rules      = 379
        length_non_compliant_rules  = 9
        length_unknown_rules        = 57
```

Hoher Automatisierungsgrad
⇒ unerlässlich für effiziente und effektive Umsetzung der Sicherheitsrichtlinien

Anwendung der Skripte im Systemtest

Rücknahme von Einstellungen

```
Revert-Rule .\sfera_automation.json BL6960-0801
```

Problematische Regeln können per Blacklist von der Umsetzung ausgeschlossen werden:

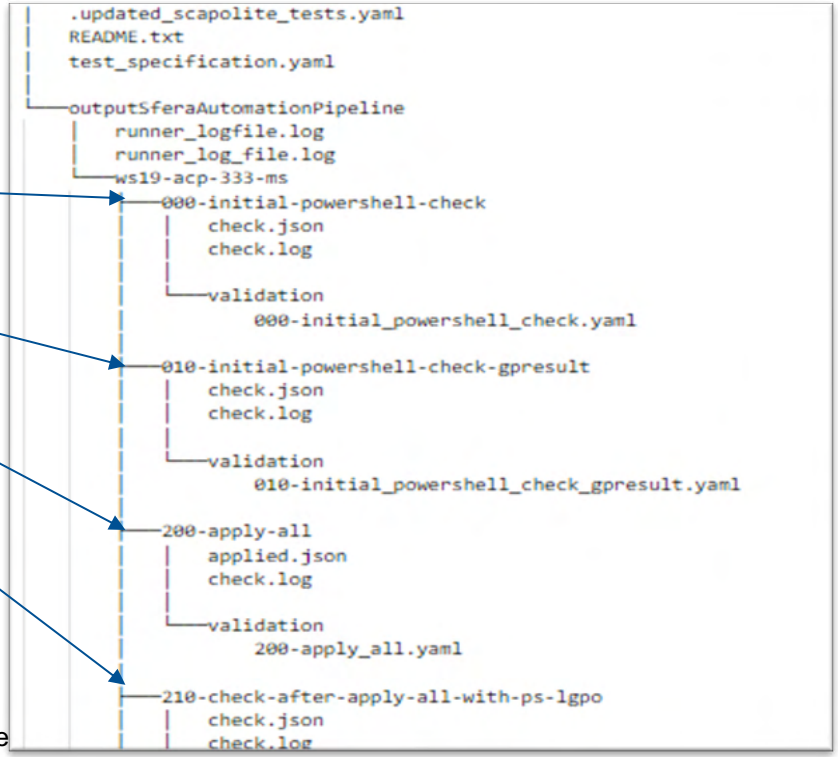
```
[
    {"rule_id": "BL696-0801",
     "comment": "Removed. Removal justified by Exemption E01 of BL969-0801: this is a stand-alone system!"}
]
```

Ausführlicher unter https://go.tum.de/917600

Ein Schritt zurück: Automatisierte Tests als Teil der Pipeline

Testschritte

- **Statische Überprüfung:** Anzahl der Automatisierungen
- **Check**-Skript auf neuerstelltem Image durchführen
  - `lgpo.exe`-basierter Test
  - `gpresult.exe`-basierter Test
- **Implementierungs**-Skript ausführen
- **Check** *nach Implementierung* ausführen
  - `lgpo.exe`-basierter Test
  - `gpresult.exe`-basierter Test
- **Check mit anderem Mechanismus** (z.B. CIS-CAT) durchführen und Ergebnisse vergleichen
- **Skript zum Zurücksetzen** der Regeln durchführen
- **Check** *nach Zurücksetzen* ausführen
  - `lgpo.exe`-basierter Test
  - `gpresult.exe`-basierter Test



Test-Spezifikation

Test-Spezifikationsfile legt Tests fest und dokumentiert erwartete Ergebnisse

```
static:
- id: validate_json_file
  type: examine_sfera_automation_json
  validations:
  - sub_type: count
    expected:
      completely_empty: 48
      no_check_script: 5
      no_impl_script: 11
  - sub_type: by_id
    expected:
      completely_empty: [BL696-0011, BL696-0118, BL696-0126, BL696-0133, BL696-0161, BL696-0167, BL696-0201, BL696-0221, BL696-0248, BL696-0249, BL696-0260,
      no_check_script: [BL696-0406, BL696-0468, BL696-0786, BL696-0830, BL696-1626]
      no_impl_script: [BL696-0031, BL696-0124, BL696-0172, BL696-0227, BL696-0326, BL696-0362, BL696-0476, BL696-0780, BL696-0856, BL696-0938, BL696-0988]
      same_setting:
        pol: [BL696-0487, BL696-3689]
        audit: []
        inf: []
```


Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Test-Spezifikation

```yaml
- id: check-after-apply-all-with-ps
  type: siemens_ps_scripts
  sub_type: check_all
  validations:
  - sub_type: count
    expected:
      blacklist_rules: 0
      compliant_checks: 509
      non_compliant_checks: 14
      empty_checks: 47
      unknown_checks: 52
  - sub_type: by_id
    result: blacklist_rules
    check_ids: []
  - sub_type: by_id
    result: compliant_checks
    check_ids: [BL696-0001, BL696-0006, BL696-0016, BL696-0021,
      BL696-1841, BL696-1856_sub_0, BL696-1856_sub_1, BL696-186
```

```yaml
- sub_type: by_id
  result: non_compliant_checks
  # BL696-0031: checks for TPM 2.0 module, which is not pre
  # BL696-0124: rule has no implementation
  # BL696-0172: rule has no implementation
  # BL696-0227: rule has no implementation
  # BL696-0326: rule has no implementation
  # BL696-0562: BLACKLISTED RULE
  # BL696-0856: rule has no implementation
  # BL696-0896: BLACKLISTED RULE
  # BL696-0949: BLACKLISTED RULE
  # BL696-1621: BLACKLISTED RULE
  # BL696-3116: BLACKLISTED RULE
  check_ids: [BL696-0031, BL696-0124, BL696-0172, BL696-022
```


Patrick Stöckle | patrick.stoeckle@tum.de | Technische Universität München (TUM)
Bernd Grobauer | Bernd.Grobauer@siemens.com | Siemens AG

Wo Tests uns gerettet haben … und wo nicht

- **Fehler eines Baseline-Autors macht einen Test *kaputt***
  - Syntax-Fehler
  - Falsch spezifizierte Policy
- **Fehler in Framework zur Generierung von Automatisierungen führt zu nicht intendierten Änderungen.**
  - Beispiel: Irrtümliche Gleichsetzung von `nicht konfiguriert` und `disabled` aufgrund missverständlicher Dokumentation in ADMX-Templates
- **Laufzeitfehler in Skript-Framework**
- **Diskrepanzen zwischen eigenem Check und zweitem Check-Mechanismus aufgrund von Fehlspezifikation durch den Autor**

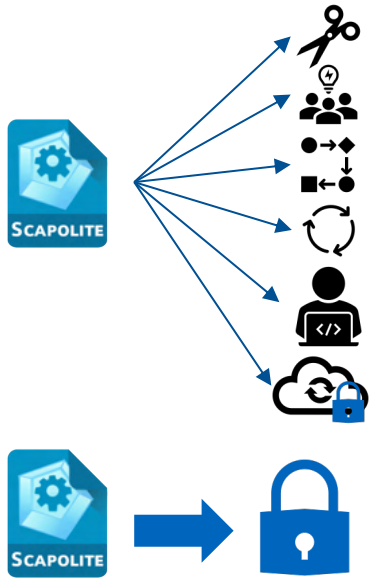
- **Fehlspezifikation** von Implementierung und Test in Fällen, in denen wir keinen zweiten, unabhängigen Check-Mechanismus haben.

Schlussfolgerung

Der Scapolite-Ansatz ermöglicht uns

- Anpassung von externen Richtlinien (z.B. von CIS) für unsere spezifischen Belange
- Kooperation eines Autoren-Teams bei Erstellung/Anpassung und Pflege von Richtlinien
- Generierung von Automatisierungen für Überprüfung und Umsetzung
- Hochautomatisierte Tests über CI/CD-Pipelines
- Bereitstellung von Skripten/Artefakten für die Automatisierte Überprüfung/Umsetzung

⟹ Grundlage für die Effektive & Effiziente Systemhärtung



Sicherheitskonfigurationsrichtlinien effizient verwalten und umsetzen: Der Scapolite-Ansatz

Patrick Stöckle
patrick.stoeckle@tum.de
Lst. f. Software und Systems Engineering
Technische Universität München (TUM)

Bernd Grobauer
bernd.grobauer@siemens.com
T CST
Siemens AG

Alexander Pretschner
alexander.pretschner@tum.de
Lst. f. Software und Systems Engineering
Technische Universität München (TUM)

München, 03.02.2022
29. DFN-Konferenz „Sicherheit in vernetzten Systemen”