Practical OpenSCAP - part 1: CLI (command-line)

Presenters

- Martin Preisler
- Watson Yuuma Sato

Special thanks to Robin Price II who contributed greatly to this document.

Abstract

OpenSCAP is a family of open source SCAP tools and content that help users create and evaluate standard security checklists for enterprise systems. Natively shipping in Red Hat Enterprise Linux, OpenSCAP provides practical security hardening advice for Red Hat technologies and links to compliance requirements, making deployment activities like certification and accreditation easier.

Audience / Intro / Prerequisites

This lab is geared towards Linux system administrators that have completed the Red Hat Certified System Administrator (RHCSA), the Red Hat Certified Engineer (RHCE) certification or have a similar skillset.

Attendees, during this session, will:

- Develop a foundational knowledge around the Security Content Automation Protocol
- Go hands-on with OpenSCAP from the command-line.
- Understand the OpenSCAP tool and security standards used to generate reports and perform remediation.

Before you begin

You should have a standard base installation of **Red Hat Enterprise Linux 7.5**.

Installing the necessary packages

Install the OpenSCAP scanner and the SCAP Security Guide packages:

```
[root@serverX ~]# yum install openscap scap-security-guide
…
Dependencies Resolved

=========================================================================
=
 Package              Arch      Version          Repository       Size
=========================================================================
=Installing:
 openscap             x86_64    1.2.16-6.el7     rhel             3.8 M
 scap-security-guide  noarch    0.1.36-7.el7     rhel             2.6 M
Installing for dependencies:
 openscap-scanner     x86_64    1.2.16-6.el7     rhel             61 k
 xml-common           noarch    0.6.3-39.el7     rhel             26 k
…
```

The oscap tool does not provide any security policies on its own — you have to obtain the rule sets from a separate package. On Red Hat Enterprise Linux, default policies are provided by SCAP Security Guide (SSG).

(optional) Install Ansible

In case you want to run Ansible playbooks we will generate later you also need Ansible. On Red Hat Enterprise Linux 7 Ansible is shipped via the “extras” channel. Let us enable it first.

```
[root@serverX ~]# subscription-manager repos
--enable=rhel-7-server-extras-rpms
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
```

Now we can proceed to install Ansible.

```
[root@serverX ~]# yum install ansible
…
```

```
==========================================================================
=
 Package     Arch       Version         Repository                   Size
==========================================================================
=
Installing:
 ansible     noarch     2.4.2.0-2.el7  rhel-7-server-extras-rpms     7.6
M
…
```

Basics of OpenSCAP CLI

First let us verify the installation of OpenSCAP by running its version command:

```
[lab-user@serverX ~]$ oscap -V
OpenSCAP command line tool (oscap) 1.2.16
Copyright 2009--2017 Red Hat Inc., Durham, North Carolina.

==== Supported specifications ====
XCCDF Version: 1.2
OVAL Version: 5.11.1
CPE Version: 2.3
CVSS Version: 2.0
CVE Version: 2.0
Asset Identification Version: 1.1
Asset Reporting Format Version: 1.1
CVRF Version: 1.1

==== Capabilities added by auto-loaded plugins ====
No plugins have been auto-loaded...

==== Paths ====
Schema files: /usr/share/openscap/schemas
Default CPE files: /usr/share/openscap/cpe
Probes: /usr/libexec/openscap

==== Inbuilt CPE names ====
Red Hat Enterprise Linux - cpe:/o:redhat:enterprise_linux
Red Hat Enterprise Linux 5 - cpe:/o:redhat:enterprise_linux:5
Red Hat Enterprise Linux 6 - cpe:/o:redhat:enterprise_linux:6
Red Hat Enterprise Linux 7 - cpe:/o:redhat:enterprise_linux:7

…
```

```
==== Supported OVAL objects and associated OpenSCAP probes ====
OVAL family   OVAL object                  OpenSCAP probe
----------    ----------                   ----------
(null)        system_info                  probe_system_info
independent   family                       probe_family
independent   filehash                     probe_filehash
independent   environmentvariable          probe_environmentvariable
independent   textfilecontent54            probe_textfilecontent54
independent   textfilecontent              probe_textfilecontent
independent   variable                     probe_variable
independent   xmlfilecontent               probe_xmlfilecontent
independent   environmentvariable58        probe_environmentvariable58
independent   filehash58                   probe_filehash58
linux         inetlisteningservers         probe_inetlisteningservers
linux         rpminfo                      probe_rpminfo
linux         partition                    probe_partition
linux         iflisteners                  probe_iflisteners
linux         rpmverify                    probe_rpmverify
linux         rpmverifyfile                probe_rpmverifyfile
linux         rpmverifypackage             probe_rpmverifypackage
linux         selinuxboolean               probe_selinuxboolean
linux         selinuxsecuritycontext       probe_selinuxsecuritycontext
linux         systemdunitproperty          probe_systemdunitproperty
linux         systemdunitdependency        probe_systemdunitdependency
unix          file                         probe_file
unix          interface                    probe_interface
unix          password                     probe_password
unix          process                      probe_process
unix          runlevel                     probe_runlevel
unix          shadow                       probe_shadow
unix          uname                        probe_uname
unix          xinetd                       probe_xinetd
unix          sysctl                       probe_sysctl
unix          process58                    probe_process58
unix          fileextendedattribute        probe_fileextendedattribute
unix          routingtable                 probe_routingtable
unix          symlink                      probe_symlink
…
```

The *oscap -V* command is great for reviewing what specifications versions, builtin CPE names (Common Platform Enumeration - essentially standardized platform IDs), supported OVAL objects and associated OpenSCAP probes are installed.

Let us move onto more productive uses of the command-line interface. The OpenSCAP CLI is split into sub-modules and each performs a very specialized task. E.g.: the “info” sub-module can be used to examine SCAP related files.

Locate the SCAP content installed on the system from the SCAP Security Guide package:

```
[root@serverX ~]# rpm -ql scap-security-guide | grep content
/usr/share/xml/scap/ssg/content
/usr/share/xml/scap/ssg/content/ssg-firefox-cpe-dictionary.xml
/usr/share/xml/scap/ssg/content/ssg-firefox-cpe-oval.xml
/usr/share/xml/scap/ssg/content/ssg-firefox-ds.xml
/usr/share/xml/scap/ssg/content/ssg-firefox-ocil.xml
/usr/share/xml/scap/ssg/content/ssg-firefox-oval.xml
/usr/share/xml/scap/ssg/content/ssg-firefox-xccdf.xml
/usr/share/xml/scap/ssg/content/ssg-jre-cpe-dictionary.xml
/usr/share/xml/scap/ssg/content/ssg-jre-cpe-oval.xml
/usr/share/xml/scap/ssg/content/ssg-jre-ds.xml
/usr/share/xml/scap/ssg/content/ssg-jre-ocil.xml
/usr/share/xml/scap/ssg/content/ssg-jre-oval.xml
/usr/share/xml/scap/ssg/content/ssg-jre-xccdf.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-cpe-dictionary.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-cpe-oval.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-ocil.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-oval.xml
/usr/share/xml/scap/ssg/content/ssg-rhel6-xccdf.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-cpe-dictionary.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-cpe-oval.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-ocil.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-oval.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml
```

Let us run the “oscap info” command on the RHEL7 SCAP datastream file - *ssg-rhel7-ds.xml*:

```
[root@serverX ~]# oscap info
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
Document type: Source Data Stream
Imported: 2018-01-08T08:03:07

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-rhel7-xccdf-1.2.xml
Generated: (null)
Version: 1.2
Checklists:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-xccdf-1.2.xml
```

```
                Status: draft
                Generated: 2018-01-08
                Resolved: true
                Profiles:
                        Title: Standard System Security Profile
                                Id: xccdf_org.ssgproject.content_profile_standard
                        Title: PCI-DSS v3 Control Baseline for Red Hat Enterprise Linux 7
                                Id: xccdf_org.ssgproject.content_profile_pci-dss
                        Title: C2S for Red Hat Enterprise Linux 7
                                Id: xccdf_org.ssgproject.content_profile_C2S
                        Title: Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)
                                Id: xccdf_org.ssgproject.content_profile_rht-ccp
                        Title: Common Profile for General-Purpose Systems
                                Id: xccdf_org.ssgproject.content_profile_common
                        Title: DISA STIG for Red Hat Enterprise Linux 7
                                Id: xccdf_org.ssgproject.content_profile_stig-rhel7-disa
                        Title: STIG for Red Hat Virtualization Hypervisor
                                Id: xccdf_org.ssgproject.content_profile_stig-rhevh-upstream
                        Title: United States Government Configuration Baseline (USGCB / STIG) -
DRAFT
                                Id: xccdf_org.ssgproject.content_profile_ospp-rhel7
                        Title: Criminal Justice Information Services (CJIS) Security Policy
                                Id: xccdf_org.ssgproject.content_profile_cjis-rhel7-server
                        Title: Standard Docker Host Security Profile
                                Id: xccdf_org.ssgproject.content_profile_docker-host
                        Title: Unclassified Information in Non-federal Information Systems and
Organizations (NIST 800-171)
                                Id: xccdf_org.ssgproject.content_profile_nist-800-171-cui
                Referenced check files:
                        ssg-rhel7-oval.xml
                                system: http://oval.mitre.org/XMLSchema/oval-definitions-5
                        ssg-rhel7-ocil.xml
                                system: http://scap.nist.gov/schema/ocil/2
                        https://www.redhat.com/security/data/oval/com.redhat.rhsa-RHEL7.xml.bz2
                                system: http://oval.mitre.org/XMLSchema/oval-definitions-5
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-pcidss-xccdf-1.2.xml
                Status: draft
                Generated: 2018-01-08
                Resolved: true
                Profiles:
                        Title: PCI-DSS v3 Control Baseline for Red Hat Enterprise Linux 7
                                Id: xccdf_org.ssgproject.content_profile_pci-dss_centric
                Referenced check files:
                        ssg-rhel7-oval.xml
                                system: http://oval.mitre.org/XMLSchema/oval-definitions-5
                        ssg-rhel7-ocil.xml
                                system: http://scap.nist.gov/schema/ocil/2
                        https://www.redhat.com/security/data/oval/com.redhat.rhsa-RHEL7.xml.bz2
                                system: http://oval.mitre.org/XMLSchema/oval-definitions-5
Checks:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-oval.xml
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-ocil.xml
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-cpe-oval.xml
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-oval.xml000
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-ocil.xml000
Dictionaries:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel7-cpe-dictionary.xml
```

Each SCAP datastream can have multiple profiles which provide policies implemented according to specific security baselines. Every profile can select different rules and use different parameters. Examples of these profiles are PCI-DSS, DISA STIG, USGCB and others. One of the capabilities of oscap is to display information about the SCAP contents within a file. When examining an XCCDF document or a SCAP data stream, generally, the most useful information is about profiles.

Example of a profile is the *Certified Cloud Providers (CCP)*. We will use this profile going forward.

```
Profiles:
…
  Title: Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)
  Id: xccdf_org.ssgproject.content_profile_rht-ccp
…
```

Scanning and reporting

We are now ready to perform our first baseline scan. From the information provided, run an actual scan from the terminal now that we have determined which security policy and profile we want to use.

```
[root@serverX ~]# oscap xccdf eval --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results-arf /tmp/arf.xml
--report /tmp/report.html --oval-results
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

The options can be broken down as follows:

```
# oscap xccdf eval \
 --profile xccdf_org.ssgproject.content_profile_rht-ccp \
 --results-arf /tmp/arf.xml \
 --report /tmp/report.html \
 --oval-results \
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

- xccdf eval
    - The oscap tool calls on the xccdf module.

  - The xccdf module is used with the eval operation which then allows us to perform the evaluation. The XCCDF module will try to load all OVAL Definition files referenced from XCCDF automatically. In addition to being able to load XCCDF file it can also load Source DataStreams.
- --profile PROFILE
  - Select a particular profile from the data stream file (INPUT file) at the end of the command.
- --results-arf FILE
  - This option tells oscap that we want the results stored as an Asset Reporting Format (ARF) in a file called arf.xml. It is recommended to use this option instead of --results when dealing with datastreams. This is because --results will write XCCDF results into the FILE. The ARF file is a more complete results file than plain XCCDF results file.
- --report FILE
  - Write HTML report into FILE. You also have to specify a --results/--results-arf for this feature to work. This is a human-readable report as opposed to the machine readable ARF.
- --oval-results TODO
- /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
  - This is the INPUT_FILE needed to perform the evaluation. Can be XCCDF or Source DataStream. You are strongly encouraged to use Source DataStream instead of plain XCCDF files.

The *ssg-rhel7-ds.xml* file which is a *Source DataStream* with *XCCDF 1.2* built inside. The advantage of *Source DataStream* is that you have everything you need bundled in one file - *XCCDF*, *OVAL(s)*, *CPE(s)*, and it supports digital signatures.

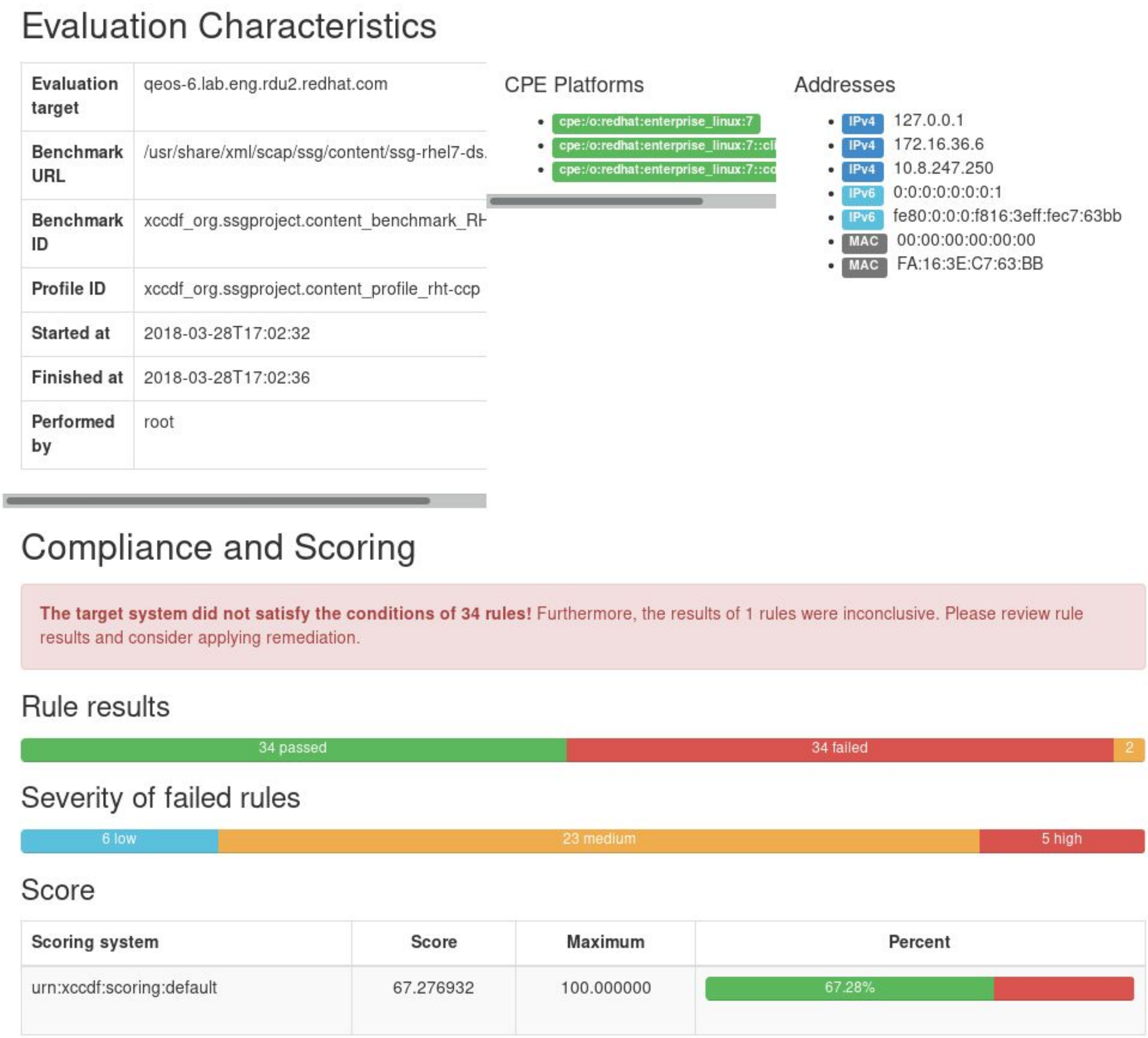
The evaluation process usually takes a few minutes, depending on the number of selected rules. Similarly to *SCAP Workbench* - the GUI frontend we will work with in the second part of the lab, *oscap* will also provide you an overview of results after it’s finished,
and you will find reports saved and available for review in your current working directory.

```
[root@serverX ~]# oscap xccdf eval --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results-arf arf.xml
--report report.html --oval-results
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

Title   Ensure /tmp Located On Separate Partition
Rule    xccdf_org.ssgproject.content_rule_partition_for_tmp
Ident   CCE-27173-4
Result  fail

Title   Ensure /var Located On Separate Partition
Rule    xccdf_org.ssgproject.content_rule_partition_for_var
Ident   CCE-26404-4
```

```
Result  fail

Title   Ensure /var/log Located On Separate Partition
Rule    xccdf_org.ssgproject.content_rule_partition_for_var_log
Ident   CCE-26967-0
Result  fail

Title   Ensure /var/log/audit Located On Separate Partition
Rule    xccdf_org.ssgproject.content_rule_partition_for_var_log_audit
Ident   CCE-26971-2
Result  fail

Title   Ensure Red Hat GPG Key Installed
Rule    xccdf_org.ssgproject.content_rule_ensure_redhat_gpgkey_installed
Ident   CCE-26957-1
Result  fail

Title   Ensure gpgcheck Enabled In Main Yum Configuration
Rule
xccdf_org.ssgproject.content_rule_ensure_gpgcheck_globally_activated
Ident   CCE-26989-4
Result  pass

Title   Ensure gpgcheck Enabled For All Yum Package Repositories
Rule    xccdf_org.ssgproject.content_rule_ensure_gpgcheck_never_disabled
Ident   CCE-26876-3
Result  fail
…
```

Let us now look at the generated HTML report:

```
$ firefox /tmp/report.html
```



Evaluation Characteristics

Evaluation target	qeos-6.lab.eng.rdu2.redhat.com
Benchmark URL	/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.
Benchmark ID	xccdf_org.ssgproject.content_benchmark_RH
Profile ID	xccdf_org.ssgproject.content_profile_rht-ccp
Started at	2018-03-28T17:02:32
Finished at	2018-03-28T17:02:36
Performed by	root

CPE Platforms

- cpe:/o:redhat:enterprise_linux:7
- cpe:/o:redhat:enterprise_linux:7::cli
- cpe:/o:redhat:enterprise_linux:7::co

Addresses

- IPv4 127.0.0.1
- IPv4 172.16.36.6
- IPv4 10.8.247.250
- IPv6 0:0:0:0:0:0:0:1
- IPv6 fe80:0:0:0:f816:3eff:fec7:63bb
- MAC 00:00:00:00:00:00
- MAC FA:16:3E:C7:63:BB

Compliance and Scoring

The target system did not satisfy the conditions of 34 rules! Furthermore, the results of 1 rules were inconclusive. Please review rule results and consider applying remediation.

Rule results

34 passed 34 failed 2

Severity of failed rules

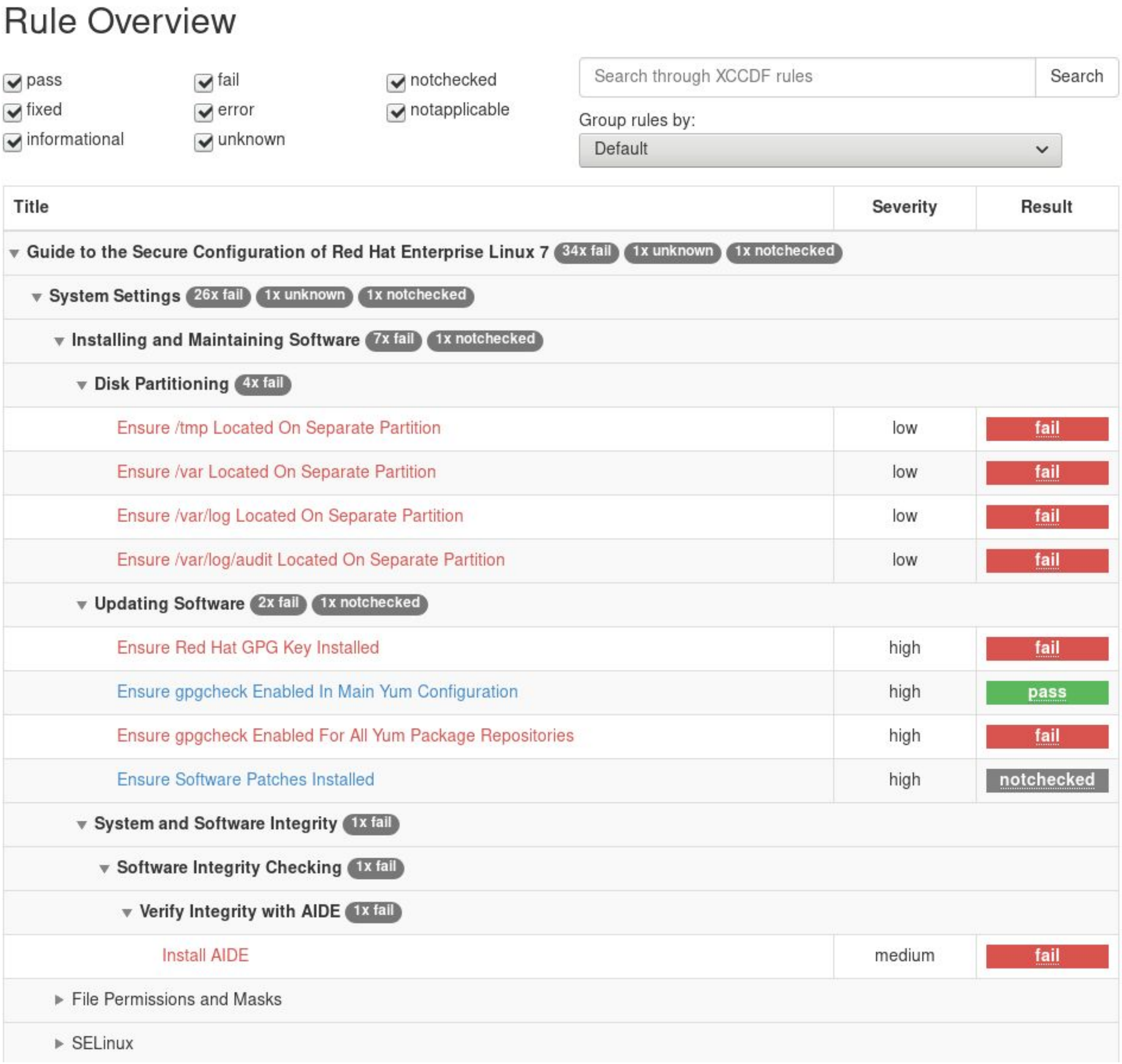
6 low 23 medium 5 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	67.276932	100.000000	67.28%

1. Evaluation Characteristics:
    - **Target Machine:** What server or container was scanned
    - **Benchmark URL:** The content of XCCDF Benchmark is mostly text. This includes titles, descriptions, CPEs, references to CVEs, CCEs, etc. All of this metadata comes together to form a nice checklist.
    - **Addresses:** IPv4, IPv6, addresses assigned to the network. These include Public, Private, and Loopback. The media access control (MAC) address are also displayed.
2. Compliance and Scoring:
    - A red or green banner will be presented with the number of satisfied or not satisfied conditions.
    - The **`Rule result breakdown`** provides a visual on the number of rules passed, failed, and not checked (other).
    - **`Failed rules by severity breakdown`** visual is a convenient way to see how many rules failed based on High, Medium, and Low definitions.

3. Score:
    - Scoring will give points to rules and the XCCDF interpreter will sum the scores of all rules to give a final score to the “compliance” state of the system. This is represented by a table outlining the Scoring system used.
    - XCCDF has four scoring models. Each apply computation of XCCDF scores differently.
        - The Default Model: **`urn:xccdf:scoring:default`**
        - The Flat Model: **`urn:xccdf:scoring:flat`**
        - The Flat Unweighted Model: **`urn:xccdf:scoring:flat-unweighted`**
        - The Absolute Model: **`urn:xccdf:scoring:absolute`**
4. Rule Overview:
    - This section is used to quickly filter out which content you would like to review.



Rule Overview

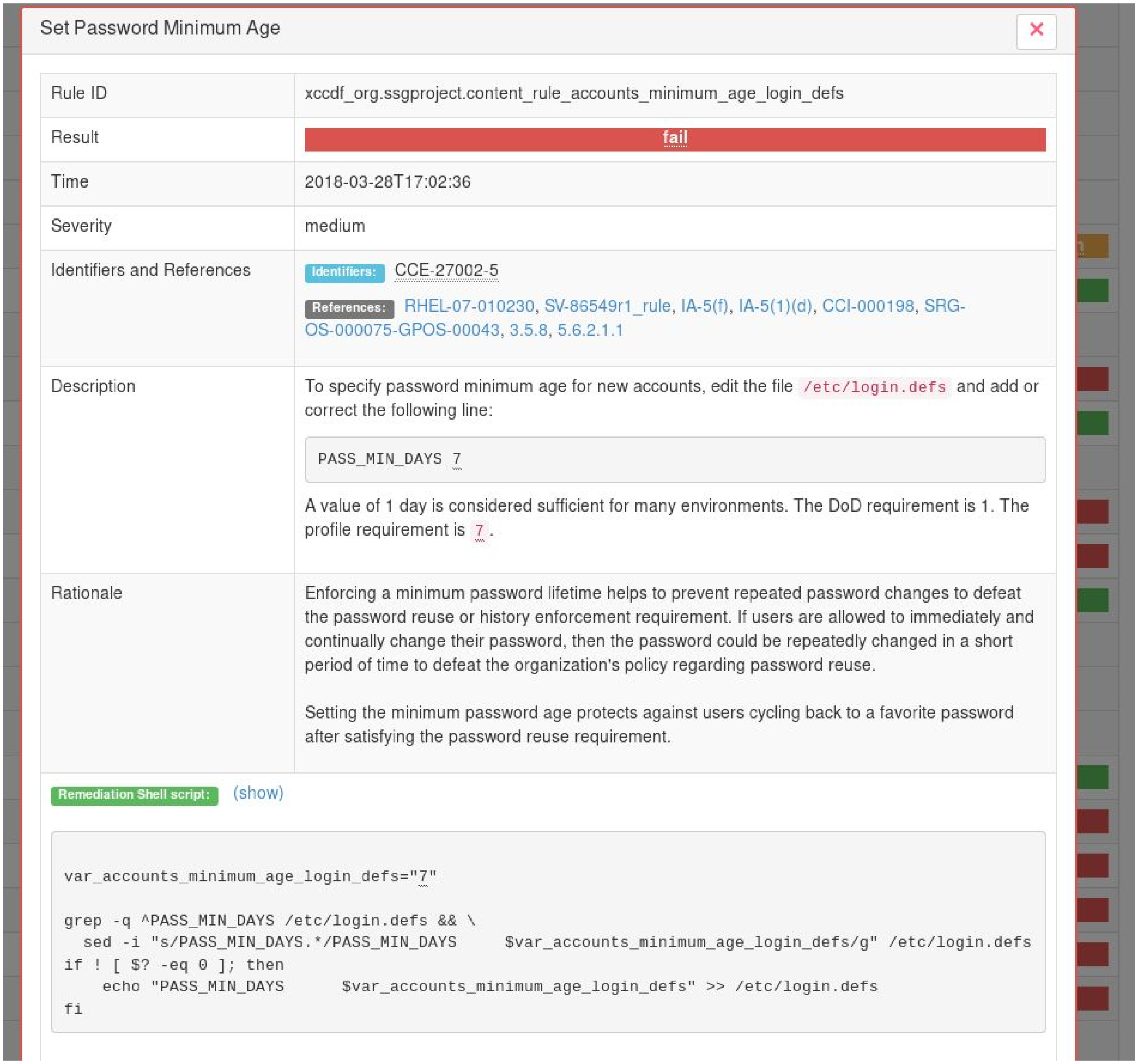
☑ pass ☑ fail ☑ notchecked
☑ fixed ☑ error ☑ notapplicable
☑ informational ☑ unknown

Search through XCCDF rules | Search

Group rules by: Default

Title	Severity	Result
▼ Guide to the Secure Configuration of Red Hat Enterprise Linux 7 34x fail 1x unknown 1x notchecked		
▼ System Settings 26x fail 1x unknown 1x notchecked		
▼ Installing and Maintaining Software 7x fail 1x notchecked		
▼ Disk Partitioning 4x fail		
Ensure /tmp Located On Separate Partition	low	fail
Ensure /var Located On Separate Partition	low	fail
Ensure /var/log Located On Separate Partition	low	fail
Ensure /var/log/audit Located On Separate Partition	low	fail
▼ Updating Software 2x fail 1x notchecked		
Ensure Red Hat GPG Key Installed	high	fail
Ensure gpgcheck Enabled In Main Yum Configuration	high	pass
Ensure gpgcheck Enabled For All Yum Package Repositories	high	fail
Ensure Software Patches Installed	high	notchecked
▼ System and Software Integrity 1x fail		
▼ Software Integrity Checking 1x fail		
▼ Verify Integrity with AIDE 1x fail		
Install AIDE	medium	fail
▶ File Permissions and Masks		
▶ SELinux		

Note: There is a cheatsheet at end of these instructions to help you quickly remember the meaning of each XCCDF Rule result.



Set Password Minimum Age

Rule ID	xccdf_org.ssgproject.content_rule_accounts_minimum_age_login_defs
Result	fail
Time	2018-03-28T17:02:36
Severity	medium
Identifiers and References	Identifiers: CCE-27002-5 References: RHEL-07-010230, SV-86549r1_rule, IA-5(f), IA-5(1)(d), CCI-000198, SRG-OS-000075-GPOS-00043, 3.5.8, 5.6.2.1.1
Description	To specify password minimum age for new accounts, edit the file `/etc/login.defs` and add or correct the following line: `PASS_MIN_DAYS 7` A value of 1 day is considered sufficient for many environments. The DoD requirement is 1. The profile requirement is `7`.
Rationale	Enforcing a minimum password lifetime helps to prevent repeated password changes to defeat the password reuse or history enforcement requirement. If users are allowed to immediately and continually change their password, then the password could be repeatedly changed in a short period of time to defeat the organization's policy regarding password reuse. Setting the minimum password age protects against users cycling back to a favorite password after satisfying the password reuse requirement.

Remediation Shell script: (show)

```
var_accounts_minimum_age_login_defs="7"

grep -q ^PASS_MIN_DAYS /etc/login.defs && \
  sed -i "s/PASS_MIN_DAYS.*/PASS_MIN_DAYS     $var_accounts_minimum_age_login_defs/g" /etc/login.defs
if ! [ $? -eq 0 ]; then
    echo "PASS_MIN_DAYS      $var_accounts_minimum_age_login_defs" >> /etc/login.defs
fi
```

Remediation

After reviewing the HTML report we can see that a lot of rules are failing. To put machine into compliance we have to perform a so-called “remediation” and in most cases also reboot the machine. There are three types of remediation:

Online remediation

In online remediation the fix scripts are executed immediately after evaluation. For each and every rule that has a “fail” result a fix script will be run. Then the rule will be re-evaluated to verify that the machine is now indeed in compliance.

To enable online remediation, use the **--remediate** command-line option:

```
[root@serverX ~]# oscap xccdf eval --remediate --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results
scan-xccdf-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

The output of this command consists of two sections. The first section shows the result of the scan prior to the remediation, and the second section shows the result of the scan after applying the remediation. The second part can contain only fixed and error results. The fixed result indicates that the scan performed after the remediation passed. The error result indicates that even after applying the remediation, the evaluation still does not pass.

(optional) Offline remediation

Offline remediation allows you to postpone fix execution and perform it outside of the oscap evaluation. This allows more flexible workflows. First, the system is only evaluated, and the results are stored in a TestResult element in an XCCDF file:

```
[root@serverX ~]# oscap xccdf eval --profile
xccdf_org.ssgproject.content_profile_rht-ccp --results xccdf-results.xml
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

In this next step, oscap generates the fix scripts based on the results of the scan.

```
[root@serverX ~]# oscap xccdf generate fix \
  --fix-type bash \
  --result-id "" \
  --output my-result-based-remediation.sh \
xccdf-results.xml
```

To generate fixes based on a TestResult XCCDF file, we need to know the result-id of TestResult element from the XCCDF file that we would like to use. Using option **`--result-id “”`** is a trick to pick the first TestResult element from results file.
To get the actual result-ids, you can issue following command:

```
[root@serverX ~]# oscap info xccdf-results.xml | grep “Result ID”
     Result ID: xccdf_org…ssgproject.content_profile_rht-ccp
```

(optional) Profile-based remediation

Another option is to generate a remediation script based on a profile. This will include a fix for every single rule in the profile, regardless or whether it would pass or fail on the target machine. This works well because all fixes are supposed to be idempotent - if you run them repeatedly they won’t do any changes to the machine. Profile-based remediation allows very powerful fix deployment using either bash or ansible.

```
[user@serverX ~]$ oscap xccdf generate fix \
  --fix-type bash \
  --profile xccdf_org.ssgproject.content_profile_rht-ccp \
  --output my-remediation-script.sh \
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

[user@serverX ~]$ vim my-remediation-script.sh
[user@serverX ~]$ chmod +x ./my-remediation-script.sh
[user@serverX ~]$ sudo ./my-remediation-script.sh
```

```
[user@serverX ~]$ oscap xccdf generate fix \
  --fix-type ansible \
  --profile xccdf_org.ssgproject.content_profile_rht-ccp \
  --output my-remediation-playbook.yml \
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

[user@serverX ~]$ vim my-remediation-playbook.yml
[user@serverX ~]$ sudo ansible-playbook -i "localhost," -c local --check
--diff my-remediation-playbook.yml
# once we review the changes and are happy with them
[user@serverX ~]$ sudo ansible-playbook -i "localhost," -c local
my-remediation-playbook.yml
```

(optional) You can customize the generate ansible playbook by changing the variables. Looking at the structure of the generated playbook these will be in the beginning of the file:

```
---
#########################################################################
#
#
# Ansible remediation role for profile rht-ccp
# Profile Title:  Red Hat Corporate Profile for Certified Cloud Providers
(RH CCP)
…
#########################################################################
#

 - hosts: all
   pre_tasks:
     - name: Verify Ansible meets SCAP-Security-Guide version
requirements.
       assert:
         that: "ansible_version.full | version_compare('2.3', '>=')"
         msg: >
           "You must update Ansible to at least version 2.3 to use this
role."

   vars:
      sshd_idle_timeout_value: 300
      sshd_listening_port: 22
      var_selinux_policy_name: targeted
      var_selinux_state: enforcing
      var_password_pam_unix_remember: 5
      var_accounts_passwords_pam_faillock_deny: 5
      var_accounts_passwords_pam_faillock_unlock_time: 604800
      var_accounts_passwords_pam_faillock_fail_interval: 100000000
      var_password_pam_dcredit: -1
      var_password_pam_difok: 3
      var_password_pam_ocredit: -2
      var_password_pam_lcredit: -2
      var_password_pam_ucredit: -2
      var_password_pam_retry: 3
      var_accounts_password_minlen_login_defs: 6
      var_accounts_password_warn_age_login_defs: 7
      var_accounts_minimum_age_login_defs: 7
   tasks:
    - name: Ensure telnet is removed
      package:
        name="{{item}}"
        state=absent
      with_items:
…
```

The advantage of profile-based remediations is their ease of deployment at scale. You can use tools such as Satellite 6 remote execution or Ansible Tower to deploy these playbooks on hundreds of machines at once.

(optional) Vulnerability scanning

The Red Hat Security Response Team provides OVAL definitions for all vulnerabilities (identified by CVE handles) that affect Red Hat Enterprise Linux 3, 4, 5, 6, and 7. This enables users to perform a vulnerability scan and diagnose whether a system is vulnerable or not.

Download the CVE feed content from Red Hat:

```
[root@serverX ~]# wget
https://www.redhat.com/security/data/oval/Red_Hat_Enterprise_Linux_7.xml
```

Run the vulnerability scan on a local machine:

```
[root@serverX ~]# oscap oval eval --results /tmp/results.xml --report
/tmp/report.html Red_Hat_Enterprise_Linux_7.xml

Definition oval:com.redhat.rhsa:def:20180592: false
Definition oval:com.redhat.rhsa:def:20180549: false
Definition oval:com.redhat.rhsa:def:20180527: false
Definition oval:com.redhat.rhsa:def:20180505: false
Definition oval:com.redhat.rhsa:def:20180502: true
Definition oval:com.redhat.rhsa:def:20180483: false
Definition oval:com.redhat.rhsa:def:20180418: false
Definition oval:com.redhat.rhsa:def:20180414: false
…
```

This may take several minutes to complete, depending on how many packages are installed. “true” results show vulnerabilities on the machine, ideally you want to see just “false” results in stdout after running the command.

Open the HTML report for viewing:

```
[lab-user@serverX ~]$ firefox report.html
```

OVAL Definition Results

🟧 ✕ 🟩 ✓ 🟨 Error 🟦 Unknown ⬜ Other

ID	Result	Class	Reference ID	Title
oval:com.redhat.rhsa:def:20180502	true	patch	[RHSA-2018:0502-01], [CVE-2017-16994], [CVE-2017-17712]	RHSA-2018:0502: kernel-alt security and bug fix update (Important)
oval:com.redhat.rhsa:def:20180180	true	patch	[RHSA-2018:0180-02], [CVE-2017-1000405]	RHSA-2018:0180: kernel-alt security and bug fix update (Important)
oval:com.redhat.rhsa:def:20170372	true	patch	[RHSA-2017:0372-01], [CVE-2016-5195], [CVE-2016-7039], [CVE-2016-8666]	RHSA-2017:0372: kernel-aarch64 security and bug fix update (Important)
oval:com.redhat.rhsa:def:20180592	false	patch	[RHSA-2018:0592-01], [CVE-2018-8088]	RHSA-2018:0592: slf4j security update (Important)
oval:com.redhat.rhsa:def:20180549	false	patch	[RHSA-2018:0549-01], [CVE-2018-5146]	RHSA-2018:0549: firefox security update (Critical)
oval:com.redhat.rhsa:def:20180527	false	patch	[RHSA-2018:0527-01], [CVE-2018-5125], [CVE-2018-5127], [CVE-2018-5129], [CVE-2018-5130], [CVE-2018-5131], [CVE-2018-5144], [CVE-2018-5145]	RHSA-2018:0527: firefox security update (Critical)
oval:com.redhat.rhsa:def:20180505	false	patch	[RHSA-2018:0505-01], [CVE-2018-5950]	RHSA-2018:0505: mailman security update (Moderate)
oval:com.redhat.rhsa:def:20180483	false	patch	[RHSA-2018:0483-01], [CVE-2018-5732], [CVE-2018-5733]	RHSA-2018:0483: dhcp security update (Important)
oval:com.redhat.rhsa:def:20180418	false	patch	[RHSA-2018:0418-01], [CVE-2018-6871]	RHSA-2018:0418: libreoffice security update (Moderate)
oval:com.redhat.rhsa:def:20180414	false	patch	[RHSA-2018:0414-01], [CVE-2017-15135], [CVE-2018-1054]	RHSA-2018:0414: 389-ds-base security and bug fix update (Important)
oval:com.redhat.rhsa:def:20180412	false	patch	[RHSA-2018:0412-01], [CVE-2017-7518], [CVE-2017-12188]	RHSA-2018:0412: kernel-rt security and bug fix update (Important)

XCCDF Rule Results cheatsheet

pass – the target system (its particular component) satisfied all the conditions of the XCCDF rule

fail – the target system (its particular component) did not meet certain condition of the XCCDF rule. For simple rules (containing reference just to one OVAL check) this means relevant system property did not meet its expected value, for compound rules at least one OVAL check of the set didn’t succeed. Particular system property should be corrected and scan rerun.

error – the checking engine was not able to complete the rule evaluation due some reason (scanner run with insufficient privileges etc.). Therefore it is not possible to decide if particular system is compliant with the requested policy or not. Reason of the error should be further investigated, corrected, and scan rerun to obtain trustworthy report.

unknown – a problem different from the error was encountered during rule evaluation (the checking engine might have presented the result and was not understood by the tool)

notapplicable – particular rule is not applicable to be tested on this system (system component / property scanned by this rule is not present on this system)

notchecked – relevant XCCDF rule does not have its OVAL counterpart defined (therefore it was not possible to obtain actual system’s property state), or the OVAL check is written in language not recognized / supported by the checking engine, or rule was not checked because it depends on fulfillment of some previous “parent” rule, and this parent rule didn’t evaluate to success earlier

notselected – particular rule is not selected for evaluation in the XCCDF benchmark

informational – the rule was checked, but the obtained data is rather meant to be an information to share, than a comparison of actual system’s property with expected policy value

fixed – previously the rule evaluated to failure, but has been corrected already