# Practical OpenSCAP, Security Standard Compliance and Reporting

# Part 2: GUI (graphical user interface)

## Presenters:

**Robin Price II** and **Martin Preisler**

## Abstract:

OpenSCAP is a family of open source SCAP tools and content that help users create 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 Admistration (RHCSA), the Red Hat Certified Engineer (RHCE) certification or have similar skillsets.

## Attendees, during this session, will...

- Develop a foundational knowledge around the **S**ecurity **C**ontent **A**utomation **P**rotocol
- Go hands-on with OpenSCAP Workbench to generate customized security baselines
- Use OpenSCAP Workbench interface and the SCAP Security Guide content to perform security scans.

# Practical OpenSCAP, Security Standard Compliance and Reporting

# Part 2: GUI (graphical user interface)

## BEFORE YOU BEGIN...

You should have a standard base installation of Red Hat Enterprise Linux 7.3

## OPENSCAP GUI LAB: INSTALLATION

- Install **OpenSCAP, SCAP Workbench,** and the **SCAP Security Guide** packages

```
[root@serverX ~]# yum install openscap-scanner scap-workbench scap-security-guide
...
Dependencies Resolved
================================================================================
Package                    Arch       Version         Repository      Size
================================================================================
Installing:
 openscap-scanner          x86_64     1.2.10-3.el7_3 rhel-dvd          50 k
 scap-security-guide       noarch     0.1.30-5.el7_3 rhel-dvd         1.2 M
 scap-workbench            x86_64     1.1.2-1.el7    rhel-dvd         1.4 M
Installing for dependencies:
 dwz                       x86_64     0.11-3.el7     rhel-dvd          99 k
 libmng                    x86_64     1.0.10-14.el7  rhel-dvd         171 k
 openscap                  x86_64     1.2.10-3.el7_3 rhel-dvd         3.5 M
 openscap-utils            x86_64     1.2.10-3.el7_3 rhel-dvd          35 k
 ...
 redhat-rpm-config         noarch     9.1.0-72.el7   rhel-dvd          78 k
 rpm-build                 x86_64     4.11.3-21.el7  rhel-dvd         145 k
 rpmdevtools               noarch     8.3-5.el7      rhel-dvd          97 k

 Transaction Summary

================================================================================
 Install  3 Packages (+14 Dependent packages)

 Total download size: 24 M
 Installed size: 130 M
 Is this ok [y/d/N]:
```

# Practical OpenSCAP, Security Standard Compliance and Reporting

# Part 2: GUI (graphical user interface)

## OPENSCAP GUI LAB: SCANNING

1. Open **SCAP Workbench** by navigating and clicking on **Applications → System Tools → SCAP Workbench**. Optionally you can open a terminal and run **scap-workbench** from the command-line.
2. You will then be asked to open an **Open Source Datastream or XCCDF** file. You should already open to **/usr/share/xml/scap**. Open the **ssg** folder and the **content** folder. Select **ssg-rhel7-ds.xml** and click **Open**.
3. SCAP Workbench should have now loaded the **Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)**
4. Make sure **Online Remediation** is unchecked.
5. Preform a quick out of the box scan by clicking the **Scan** button located on the bottom right.
6. Once completed, click **Show Report** and take a few minutes to review the **OpenSCAP Evaluation Report**.

```
[root@serverX ~]#  scap-workbench
13:27:39 | info  | SCAP Workbench 1.1.2, compiled with Qt 4.8.5, using OpenSCAP
1.2.10
13:27:58 | info  | Opened file '/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml'.
13:28:06 | info  | Querying capabilities...
13:28:06 | info  | Creating temporary files...
13:28:06 | info  | Starting the oscap process...
13:28:06 | info  | Processing...
13:28:56 | info  | The oscap tool has finished. Reading results...
13:28:56 | info  | Processing has been finished!
START /usr/bin/firefox "/tmp/qt_temp.n29737.html"
```

## OPENSCAP GUI LAB: REPORT OVERVIEW

- Since there are various XCCDF rule's evaluation results possible (except plain *pass* or *fail*) we will briefly document the differences between them.

**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.

# Practical OpenSCAP, Security Standard Compliance and Reporting

## Part 2: GUI (graphical user interface)

**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 (either by a tool capable of automated remediation or by human intervention).

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`**

# Practical OpenSCAP, Security Standard Compliance and Reporting

# Part 2: GUI (graphical user interface)

4. Rule Overview:
   - This section is used to quickly filter out which content you would like to review.

The rest of the report shows all of the rules used during the scan. You can now click on the Title of the rule to get more information regarding the conditions. These include the Rule ID, Identifiers, remediation commands, OVAL details, and a Remediation Script located at the bottom of the page.

## OPENSCAP GUI LAB: TAILORING

**Note:** The following table outlines what each icon and line items Tailoring represents.

```
This clipboard icon represents the the Benchmark being use.
        Type: xccdf:Benchmark
```

```
The folder icon is a Group that can contain multiple groups, rules, and values.
        Type: xccdf:Group
```

```
The paper icon is a Rule with-in the Group.  You can not modify these values.
        Type: xccdf:Rule
```

```
The tools icon is a changeable value.  You are able to modify these values.
        Type: xccdf:Value
```

1. Switch back to the SCAP Workbench window and click the **Clear** button.
2. Click and open the **Customize** button once enabled.
3. A new **Tailor profile** window will popup asking to name the **New profile ID**.
4. Keep the suggested prodile ID as **xccdf_org.ssgproject.content_profile_rht-ccp_tailored** and click **OK.**
5. SCAP Workbench will now display all the rules for the SCAP profile for **Red Hat Certified Cloud Providers**.
6. Click **Deselect All** at the very top.
7. Next, type **gshadow** in the search box and click **Search.**
8. You should now be located under the following rules:

```
...
☑ Verify Permissions on Important Files and Directories
  ☑ Verify Permissions on Files with Local Account Information and Credentials
    ☐ Verify User Who Owns shadow File
    ☐ Verify Group Who Owns shadow File
    ☐ Verify Permissions on shadow File
    ☐ Verify User Who Owns group File
    ☐ Verify Group Who Owns group File
    ☐ Verify Permissions on group File
```

# Practical OpenSCAP, Security Standard Compliance and Reporting

# Part 2: GUI (graphical user interface)

```
    ☐ Verify User Who Owns gshadow File
...
```

- Click and enable all twelve rules.

```
...
☑ Verify Permissions on Important Files and Directories
  ☑ Verify Permissions on Files with Local Account Information and Credentials
    ☑ Verify User Who Owns shadow File
    ☑ Verify Group Who Owns shadow File
    ☑ Verify Permissions on shadow File
    ☑ Verify User Who Owns group File
    ☑ Verify Group Who Owns group File
    ☑ Verify Permissions on group File
    ☑ Verify User Who Owns gshadow File
...
```

9. Click **Confirm tailoring** located on the bottom left.
10. SCAP Workbench will now load the base **Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)** profile along with your unsaved tailoring changes.
11. Click **Scan** and run a new evaluation. This should be 100% pass.
12. Once completed, click **Show Report** and take a few minutes to review the **OpenSCAP Evaluation Report**.
13. Take note the **Profile ID** has been changed.

## OPENSCAP GUI LAB: SAVE TAILORING

1. Switch back to SCAP Workbench
2. At the top, click **File → Save As ... → Save as RPM**
3. Use the following to prep the RPM
    - Package Name: **ssg-rhel7-ds-tailored**
    - Version: **1**
    - Release: **1**
    - Summary: **Customized SCAP content for Summit 2017**
    - License: **GPLv2+**
4. Click **OK** save the RPM in your home folder.

# Part 2: GUI (graphical user interface)

5. Verify the contents of the RPM by opening a terminal by clicking **Applications → Utilities → Terminal**

```
[root@serverX ~]# rpm -qpl ssg-rhel7-ds-tailored-1-1.noarch.rpm
/usr/share/xml/scap/ssg-rhel7-ds-tailored/ssg-rhel7-ds.xml
/usr/share/xml/scap/ssg-rhel7-ds-tailored/tailoring-xccdf.xml
```

You can now use this RPM with Satellite 6 or through your preferred configuration management tool.

## SCAP WORKBENCH: ONLINE REMEDIATION

1. Open a terminal by clicking **Applications → Utilities → Terminal**
2. Change the permissions on **/etc/gshadow** for other users to read.

```
[root@serverX ~]# ls -lah /etc/gshadow
----------. 1 root root 750 Mar  5 13:46 /etc/gshadow
[root@serverX ~]# chmod o+r /etc/gshadow
[root@serverX ~]# ls -lah /etc/gshadow
-------r--. 1 root root 750 Mar  5 13:46 /etc/gshadow
```

3. Click back to SCAP Workbench and click the **Clear** button.
4. Run a new **Scan** and notice the **Verify Permissions on gshadow File** failed.
5. Click **Clear** again.
6. Check **Online Remediation** to enable it and preform a new **Scan**.
7. The rule will still show as fail but notice the additional line at the bottom. The **Verify Permissions on gshadow File** has been labeled as **fixed**.
8. Click back over to the open terminal and verify the file has been successfully changed back.

```
[root@serverX ~]# ls -lah /etc/gshadow
----------. 1 root root 750 Mar  5 13:46 /etc/gshadow
```

9. Once completed, click **Show Report** and take a few minutes to review the **OpenSCAP Evaluation Report.**