

Practical OpenSCAP - Part 2: GUI (graphical user interface) and containers

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 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 similar skillsets.

Attendees, during this session, will...:

- Develop a foundational knowledge around the **Security Content Automation Protocol**
- 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.
- Use atomic to scan and remediate container images
- Learn how to output scan results for DISA STIG Viewer

Before you begin...

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

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    1.2.16-6.el7  rhel-7        61 k
  scap-security-guide          noarch    0.1.36-7.el7  rhel-7        2.6 M
  Scap-workbench               x86_64    1.1.6-1.el7   rhel-7        1.8 M
Installing for dependencies:
  openscap-containers          noarch    1.2.16-6.el7  rhel-7        27 k
  openscap-utils               x86_64    1.2.16-6.el7  rhel-7        27 k

Transaction Summary
=====
=====
Install  3 Packages (+2 Dependent packages)

Total download size: 4.5 M
Installed size: 63 M
Is this ok [y/d/N]:
```

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 load content from SCAP Security Guide select **RHEL7** and click **Load Content**.





3. Change selected Profile to **Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)**
4. Make sure **Remediate** is unchecked.
5. Perform a quick out of the box scan by clicking the **Scan** button located on the bottom right.
6. **SCAP Workbench** needs privileges to scan. It will ask for privileges to run as root, type your password and continue.
7. Once completed, click **Show Report** and take a few minutes to review the **OpenSCAP Evaluation Report**.

```
[lab-user@serverX ~]$ scap-workbench
04:12:01 | info      | SCAP Workbench 1.1.6, compiled with Qt
4.8.7, using OpenSCAP 1.2.16
04:12:04 | info      | Opened file
'/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml'.
04:12:15 | info      | Querying capabilities...
04:12:15 | info      | Creating temporary files...
04:12:15 | info      | Starting the oscap process...
04:12:15 | info      | Processing...
04:12:32 | warning   | Remote resources might be necessary for this
profile to work properly. Please select "Fetch remote resources"
for complete scan
04:12:32 | warning   | Skipping
https://www.redhat.com/security/data/oval/com.redhat.rhsa-RHEL7.xml
.bz2 file which is referenced from XCCDF content
04:12:32 | info      | The oscap tool has finished. Reading
results...
04:12:32 | info      | Processing has been finished!
START /bin/firefox "/tmp/qt_temp.jW6007.html"
```

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 profile ID as **xccdf_org.ssgproject.content_profile_rht-ccp_customized** 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
    ☐ Verify User Who Owns gshadow File
...
```

9. 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
...
```

10. Click **OK** located on the bottom right.
11. SCAP Workbench will now load the base **Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)** profile along with your unsaved tailoring changes.
12. Click **Scan** and run a new evaluation. This should be 100% pass.
13. Once completed, click **Show Report** and take a few minutes to review the **OpenSCAP Evaluation Report**.
14. 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 All** → **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 2018**
 - License: **GPLv2+**
4. Click **OK** save the RPM in your home folder.
5. Verify the contents of the RPM by opening a terminal by clicking **Applications** → **Utilities** → **Terminal**

```
[lab-user@serverX ~]$ sudo rpm -qp1
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.

```
[lab-user@serverX ~]$ sudo ls -l /etc/gshadow  
----- . 1 root root 750 Mar  5 13:46 /etc/gshadow  
[lab-user@serverX ~]$ sudo chmod o+r /etc/gshadow  
[lab-user@serverX ~]$ sudo 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 **Remediate** to enable it and perform 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**.

(optional) SCAP Workbench: Profile based remediation

1. Switch back to SCAP Workbench
2. Click back to SCAP Workbench and click the **Clear** button.
3. On the bottom left, click **Generate remediation role** → **bash**
4. Change file name to **my-tailored-remediation-script.sh**.
5. Click **Save** to save remediation role in your home folder.
6. Check that remediation script contains fixes for all 12 rules selected

```

[lab-user@serverX ~]$ vim my-tailored-remediation-script.sh
#####
##
#
# Bash remediation role for profile
xccdf_org.ssgproject.content_profile_rht-ccp_customized
# Profile Title:  Red Hat Corporate Profile for Certified Cloud
Providers (RH CCP) [CUSTOMIZED]
# Profile Description:
# This is a *draft* SCAP profile for Red Hat Certified Cloud
Providers.

...

# This script is generated from an OpenSCAP profile without
preliminary evaluation.
# It attempts to fix every selected rule, even if the system is
already compliant.
#
# How to apply this remediation role:
# $ sudo ./remediation-role.sh
#
#####
##

#####
##
# BEGIN fix (1 / 12) for
'xccdf_org.ssgproject.content_rule_userowner_shadow_file'
#####
##
(>&2 echo "Remediating rule 1/12:
'xccdf_org.ssgproject.content_rule_userowner_shadow_file'")
chown root /etc/shadow
# END fix for
'xccdf_org.ssgproject.content_rule_userowner_shadow_file'
...

```

OpenSCAP Container LAB

Containers are a big part of today's systems and they also require compliance assurance. The recommended way to scan containers is via atomic, it provides an interface for scanners to plugin and easily scan containers. OpenSCAP provides a scanner for Atomic that can scan images for vulnerabilities and configuration compliance.

OpenSCAP Container LAB: installation

On Red Hat Enterprise Linux 7 Atomic and docker are 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 Atomic and docker packages.

```
[root@serverX ~]# yum install atomic docker
...
Dependencies Resolved

=====
Package
  Arch    Version                      Repository                      Size
=====
Installing:
  atomic  x86_64  1:1.22.1-1.gitd36c015.el7    rhel-7-server-extras-rpms    990 k
  docker  x86_64  2:1.13.1-58.git87f2fab.el7   rhel-7-server-extras-rpms    16 M

Transaction Summary
=====
Install 2 Packages

Total download size: 17 M
Installed size: 60 M
Is this ok [y/d/N]:
```


1. Make sure that docker service is running
2. For convenience, enable docker so the service starts automatically when the system boots up.

```
[root@serverX ~]# systemctl start docker
[root@serverX ~]# systemctl enable docker
Created symlink from
/etc/systemd/system/multi-user.target.wants/docker.service to
/usr/lib/systemd/system/docker.service.
```

3. Install the OpenSCAP scanner image in Atomic

```
[root@serverX ~]# atomic install
registry.access.redhat.com/rhel7/openscap
Pulling registry.access.redhat.com/rhel7/openscap:latest ...
...

Installing the configuration file 'openscap' into /etc/atomic.d/.
You can now use this scanner with atomic scan with the --scanner
openscap command-line option. You can also set 'openscap' as the
default scanner in /etc/atomic.conf. To list the scanners you have
configured for your system, use 'atomic scan --list'.

Copying the remediation script 'remediate.py' into
/etc/atomic.d/scripts/. You can now remediate images with atomic
scan using --remediate command-line option.

Saving current config.ini as
config.ini.2018-04-10-08:33:41.atomic_save
Updating config.ini with latest configuration
Installation complete. You can customize /etc/oscaped/config.ini as
needed.
```

4. Pull the image you would like to scan, in this lab we will use rhel7 image

```
[root@serverX ~]# docker pull registry.access.redhat.com/rhel7
Using default tag: latest
Trying to pull repository registry.access.redhat.com/rhel7/openscap
...
```

```
Digest:
sha256:5562b9f50a96de2e1c05d3c969c736f8cdaa9368293600b7445ea8ff5c47
74a3
Status: Downloaded newer image for
registry.access.redhat.com/rhel7/openscap:latest
```

5. Let's check the images we have installed so far

```
[root@serverX ~]# # docker images
REPOSITORY                                TAG
IMAGE ID                                SIZE
registry.access.redhat.com/rhel7/openscap  latest
1454bdcf6f1e                             422 MB
registry.access.redhat.com/rhel7          latest
fd1ba0b398a8                             196 MB
```

OpenSCAP Container LAB: Configuration scan

With the scanner image and target image installed. We can get to know better the scanner and scan our image.

1. To list available scanners issue command **atomic scan --list**.

```
[lab-user@serverX ~]$ atomic scan --list
Scanner: openscap
Image Name: registry.access.redhat.com/rhel7/openscap
Scan type: cve *
Description: Performs a CVE scan based on Red Hat released CVE
OVAL. !WARNING! This CVE is built into container image and it might
be out-of-date. Change config.ini to configure the scanner to fetch
latest CVE data

Scan type: standards_compliance
Description: !DEPRECATED! Performs scan with Standard Profile,
as present in SCAP Security Guide shipped in Red Hat Enterprise
Linux

Scan type: configuration_compliance
Description: Performs a configuration compliance scan
according to selected profile from SCAP Security Guide shipped in
Red Hat Enterprise Linux.
```

* denotes defaults

Note in the Image Name, after the registry URL, in bold is the name of scanner.

OpenSCAP scanner provides two types of scan, **cve** and **configuration_compliance**.

Scan type **cve** is the default, and it checks for known vulnerabilities. Scan type **configuration_compliance** checks for configurations according to a profile.

2. Let's check what is the image version and what is bundled in the scanner. Issue command **atomic help rhel7/openscap**.

```
[root@serverX ~]# atomic help rhel7/openscap
```

```
Image version: 7.5.0
```

Description:

This image can be used by 'atomic scan' to perform two types of scans: Scanning for vulnerabilities using RHEL CVE feeds, which are already part of the image, informs you about installed applications that have known security issues. Scanning for configuration compliance can confirm that the scanned system complies to a given security profile. In cases that it does not comply, you can try to fix the failing rules by passing '--remediate' as an atomic scan argument.

OpenSCAP packages bundled in the image:

openscap-scanner-1.2.16-6.el7.x86_64

openscap-containers-1.2.16-6.el7.noarch

openscap-utils-1.2.16-6.el7.x86_64

openscap-1.2.16-6.el7.x86_64

openscap-daemon-0.1.10-1.el7.noarch

scap-security-guide-0.1.36-7.el7.noarch

Version of image is 7.5.0. Note that everything used to perform the scans is bundled within the image. The OpenSCAP used is version 1.2.16, policies used for configuration compliance scans are sourced from SSG 0.1.36, and CVE feed used for vulnerability scan is the latest at image build time. To check when image was built, issue command **docker images**.

3. Scan the RHEL7 image for compliance against DISA STIG policy. Note that we need to use **configuration_compliance** scan type. (Note that the atomic command supports basic bash completion.)

```
[root@serverX ~]# atomic scan \
  --scan_type configuration_compliance \
  --scanner_args \
  "profile=stig-rhel7-disa,report" \
  rhel7
...
rhel7 (fd1ba0b398a8)

The following issues were found:

    Ensure Software Patches Installed
    Severity: Important
    XCCDF result: notchecked

    Ensure YUM Removes Previous Package Versions
    Severity: Low
    XCCDF result: fail

    Ensure gpgcheck Enabled for Local Packages
    Severity: Important
    XCCDF result: fail
...

Files associated with this scan are in
/var/lib/atomic/openscap/2018-04-10-01-51-08-271553.
```

6. Outputs of the scan will be placed in mentioned directory under a directory named with the container image ID. Take some time to go through the report.

```
[user@serverX ~]$ firefox
/var/lib/atomic/openscap/2018-04-10-01-51-08-271553/fd1ba0b398a82d56
900bb798c8b099fbe3166bc49e2c5e947f7973cd38ff1a90/report.html
```

(optional) OpenSCAP Container LAB: Remediation

The process of remediating a container will generate a **new image** for you, the original image will stay the same.

Currently the OpenSCAP scanner needs the host machine to be subscribed to be able to remediate your containers.

1. The command to remediate container images is very similar to the command issued to scan it. Just add the option **--remediate** to the scan command issued previously.

```
[root@serverX ~]# atomic scan \
  --remediate \
  --scan_type configuration_compliance \
  --scanner_args \
  "profile=stig-rhel7-disa,report" \
  rhel7
...
rhel7 (fd1ba0b398a8)

The following issues were found:

    Ensure Software Patches Installed
    Severity: Important
    XCCDF result: notchecked

    Ensure YUM Removes Previous Package Versions
    Severity: Low
    XCCDF result: fail

    Ensure gpgcheck Enabled for Local Packages
    Severity: Important
    XCCDF result: fail
...

Remediating target
fd1ba0b398a82d56900bb798c8b099fbe3166bc49e2c5e947f7973cd38ff1a90.
Step 1/3 : FROM
fd1ba0b398a82d56900bb798c8b099fbe3166bc49e2c5e947f7973cd38ff1a90
---> 33a3ad89f9ab
Step 2/3 : COPY fix.sh /
---> 5097e2d450df
Step 3/3 : RUN chmod +x /fix.sh; /fix.sh ;
...

Successfully built 70d0c0411021
Successfully built remediated image 70d0c0411021 from
fd1ba0b398a82d56900bb798c8b099fbe3166bc49e2c5e947f7973cd38ff1a90.
```

```
Files associated with this scan are in
/var/lib/atomic/openscap/2018-04-10-02-11-30-585301.
```

2. Now, scan the newly built container image and check the report.

```
[root@serverX ~]# atomic scan \
  --scan_type configuration_compliance \
  --scanner_args \
  "profile=stig-rhel7-disa,report" \
  70d0c0411021

70d0c0411021 (70d0c0411021)
...

The following issues were found:

    Ensure Software Patches Installed
    Severity: Important
    XCCDF result: notchecked

    Ensure gpgcheck Enabled for Repository Metadata
    Severity: Important
    XCCDF result: fail
...

Files associated with this scan are in
/var/lib/atomic/openscap/2018-04-10-02-31-40-395185.

[user@serverX ~]$ firefox
/var/lib/atomic/openscap/2018-04-10-02-31-40-395185/70d0c041102165b
34e9c1bf502f3ef1eb2a8dd69b55329396e2f83a81f7949d6/report.html
```

(optional) OpenSCAP Container LAB: Vulnerability Scan

The OpenSCAP scanner can also scan container images for known vulnerabilities. To scan for known vulnerabilities just issue following command **atomic scan rhel7**.

```
[root@serverX ~]# atomic scan rhel7
...
rhel7 (fd1ba0b398a8)
```

rhel7 passed the scan

Files associated with this scan are in

/var/lib/atomic/openscap/2018-04-10-02-45-58-288970.

(optional) DISA STIG Viewer

The Defense Information Systems Agency (DISA) has developed a tool for the Department of Defense (DoD) to provide an easy GUI for reading, search and sorting of STIG content. The tool also provides additional functionalities, like creation of a checklist for input and review of results.

STIG Viewer expects that IDs of results are the same as in STIG content, but SSG uses a different naming scheme for its content. Therefore, results of a scan performed with SSG content cannot be directly loaded into the STIG Viewer.

To generate results compatible with STIG Viewer, option **--stig-viewer** must be used with oscap command.

Let's do a scan with RHEL7 STIG profile and generate results for STIG Viewer:

```
[root@serverX ~]# oscap xccdf eval \  
  --profile xccdf_org.ssgproject.content_profile_stig-rhel7-disa \  
  --stig-viewer stig-viewer-results.xml \  
  --results-arf arf.xml \  
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

Notice that options **--results-arf** and **--results** can be used in conjunction with **--stig-viewer** option. This way you can obtain the ARF results and results for STIG Viewer in one scan. After the scan finishes, there will be stig-viewer-results.xml with result IDs aligned with STIG content, and arf.xml with result IDs aligned with SSG.

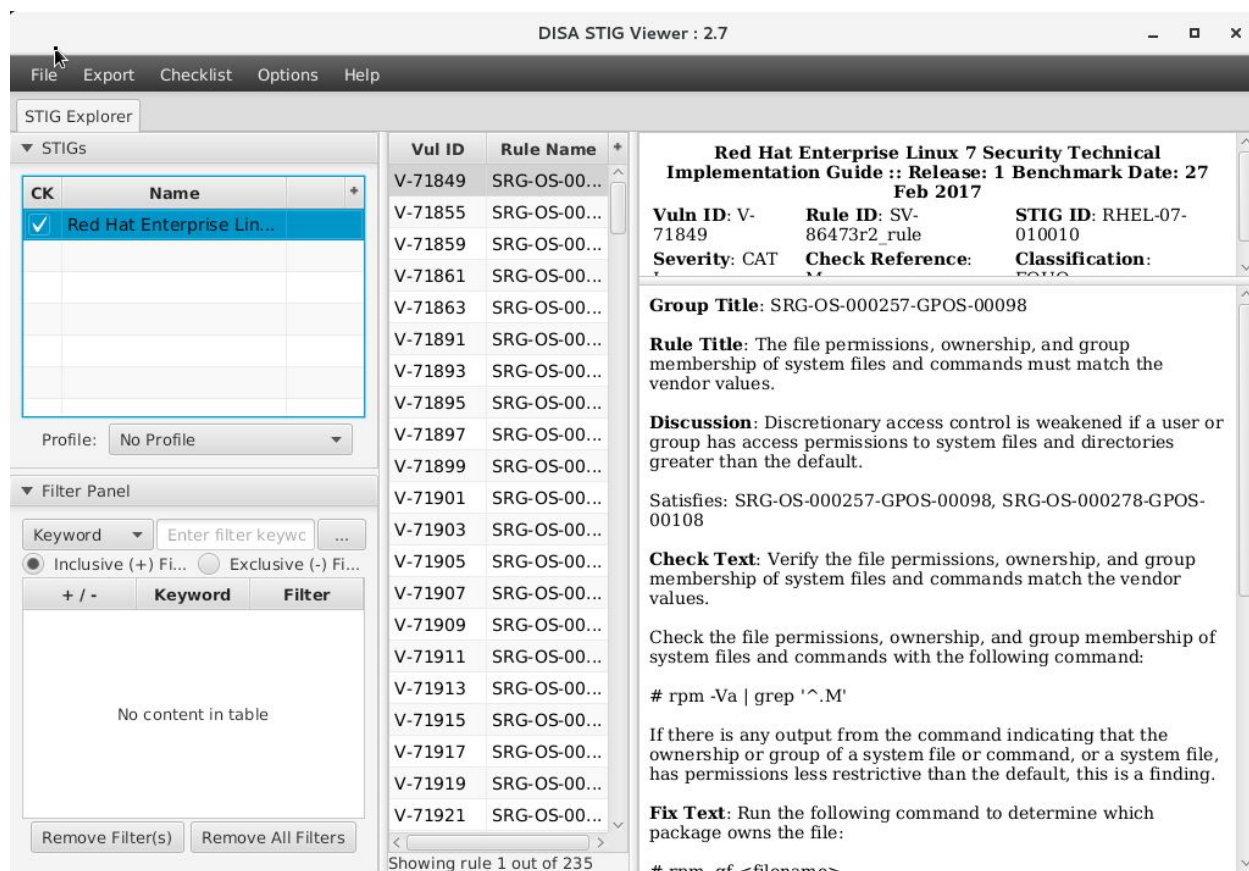
Launch DISA STIG Viewer.

```
[lab-user@serverX ~]$ cd DISA-STIG-Viewer  
[lab-user@serverX DISA-STIG-Viewer]$ java -jar STIGViewer-2.7.jar
```

Load the STIG Content. From STIGViewer, click on **File** → **Import STIG** → **Select file** **"disa-stig-rhel7-v1r1-xccdf-manual.xml"**.

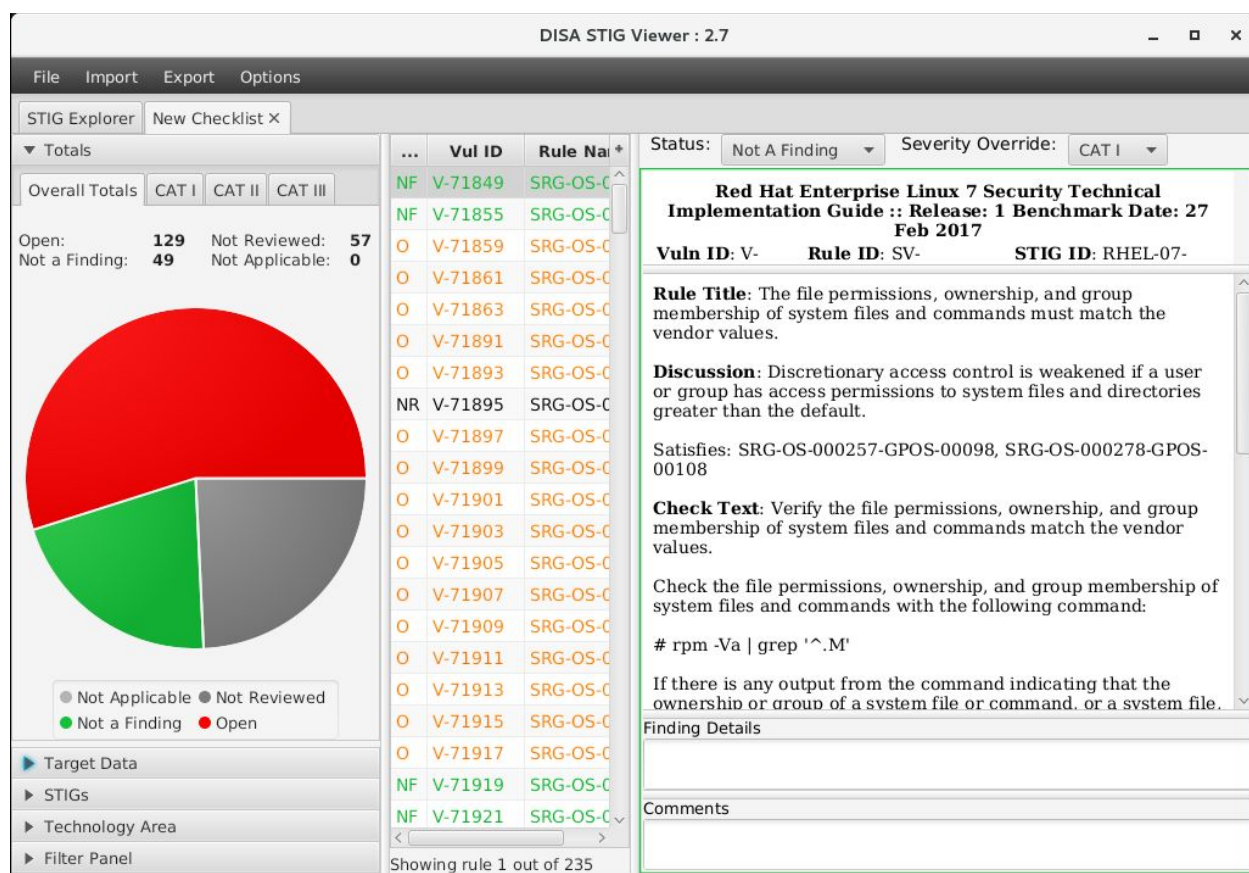
The right panel shows a list of STIG's loaded and filtering tools, central panel presents the list of rules, and right panel their content.

Take some time to navigate through content and get acquainted with the tool.



Now, let's create a checklist from RHEL7 STIG content and input the results of scan of our machine with DISA STIG Profile.

- Select the "Red Hat Enterprise Linux 7" STIG on the left panel
- Click **Checklist** → **Create Checklist - Check Marked STIG(s)**. A new checklist will be created from the selected content.
- Click **Import** → **XCCDF Results File** and select **stig-viewer-results.xml** file in home directory.



The pie chart in **Totals** panel shows the rules and their results, following is a mapping from STIGViewer result to XCCDF result.

- Open → fail
- Not a Finding → pass
- Not Reviewed → notchecked
- Not Applicable → notapplicable

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