

Quick guide for using the consolidated data streams

For applying the required configuration, the labs may use the individual scripts or commands provided in the `..\validationTestSuites\Consolidated\<Platform>\configurationTools` folder.

Windows systems

1. Ensure that the following pre-requisites are met:

- 1.1 The test systems shall be named TESTMACHINE, TESTMACHINE01, TESTMACHINE02, up to TESTMACHINE99.

- 1.2 Python 3.2 is installed

Windows x86 MSI Installer (<http://www.python.org/ftp/python/3.2.5/python-3.2.5.msi>) or Windows X86-64 MSI Installer (<http://www.python.org/ftp/python/3.2.5/python-3.2.5.amd64.msi>)

- 1.3 The Python extensions for Windows (PyWin32) are installed

From <http://sourceforge.net/projects/pywin32/files/pywin32/>, download `pywin32-218.win32-py3.2.exe` for 32-bit systems or `pywin32-218.win-amd64-py3.2.exe` for 64-bit systems.

- 1.4 The Microsoft Visual C++ 2010 and 2012 runtimes are installed

- 1.5 The Java Runtime Environment must be installed if you are running the SCAP reference implementation.

The latest version of the Java Runtime Environment can be downloaded from: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

- 1.6 Please ensure the system is fully patched.

- 1.7 If the test system is connected to a domain, block the GPO inheritance:

- On the AD server: Start Group Policy Management -> Right Click on your domain -> New Organizational Unit: "OU-NO-GPOs" -> right click on "OU-NO-GPOs" -> Block Inheritance

- Start Active Directory Users and computers -> Drag and drop your test system to "OU-NO-GPOs"
- On the test system: Restart the test system

2. Create c:\validationTestSuites\ folder. Copy the combinedDataStreams_<version>.zip file to the test system and extract it to c:\validationTestSuites\. For the next steps we will assume that the zip file has been extracted to "c:\validationTestSuites\". If you extract the files to a different location substitute that location in the following instructions.

The content of the c:\validationTestSuites\ folder should be as follows:

Parent folder	Folder or File name	Description
..\Windows\	ValidationFiles	SCAP datastream validation results
	Documents	OVAL Test Data and configuration spreadsheet
	configurationTools	Tools for configuring the system
	Windows-datastream.xml	SCAP datastream
	catalog.xml	Expected results
..\tools\	compare.py	Python program used to compare the actual scan results with the expected results

3. Configure the target system(s), either using the provided tools and instructions below or using an alternate method of your choice:

3.1 From a command prompt, change to the folder to
"c:\validationTestSuites\Windows\configurationTools"

3.2 Execute the **config#.bat** file from the folder, where # is the configuration number that you wish to execute. This file must be run with administrator privileges. There are 4 different configurations for Windows XP (config 1 to 4) and 8 configurations for Windows Vista and 7 (config 1 to 8).

For example, to apply the Configuration 1 on Windows 7 64bit, run as Administrator:
"config1.bat Win764bit" or just run config1.bat and select the option for Windows 7.
The following options are available for the config#.bat scripts:
config[1-8].bat {WinXP32bit | WinVista32bit | Win732bit | Win764bit}

Note: The configuration settings for the following tests are not automated:

1. win_cmdlet_test for Configuration 1 – please follow the configuration instructions included in the ..\win_cmdlet_test\README.txt file to install the custom cmdlet.
4. Import the validation test content data stream for Windows (c:\validationTestSuites\Windows\Windows-datastream.xml) into the product following the tool instructions.
5. Perform configuration scans and export the evaluation results following the vendor tool instructions. The result file format may look like:

[arf | xccdf | oval | ocil | sc]-[OS]-[platform]-config[1-8].xml

Where:

arf = ARF results
xccdf = XCCDF results
oval = OVAL results
ocil = OCIL results
sc = systems characteristics
OS = w7, wvista, or wxp
Platform = 32 or 64

For instance, the results file for consolidated data stream on Windows 7 64, configuration 1 may look like:

arf-w7-64-config1.xml

If you're exporting results from an USGCB data stream, replace "config[1-8]" with "exact", "reduced" or "enhanced":

xccdf-win7-64-exact.xml

6. Compare the results produced by the tool with the expected results from the spreadsheets or catalog.xml file. You may use the provide **compare.py** tool or an alternate method. The instructions below apply to the use of compare.py:
 - From a command prompt, change the directory to c:\validationTestSuites\tools or the directory where the compare.py tool is located.
 - Run: **python compare.py {RESULTSFILE} {CATALOGFILE} -i {SuiteID_from_catalog.xml} -o {Output_file}**
 - o The compare.py script expects an ARF file. If you want to use it to parse an XCCDF file you must specify the **-x** switch. If the version of XCCDF is something other than 1.2 use the **-xv {XccdfVersion}** switch specify the version. For additional help or to see all available option run compare.py with the **-h** switch.

For instance, using the catalog.xml file available for the Windows datastream (c:\validationTestSuites\Windows\catalog.xml), run:

```
python compare.py arf-w7-64-config1.xml catalog.xml -i G2.w7_configuration1 -o arf-w7-64-config1.output
```

The script will flag all the differences between the results obtained by the product and the expected results from the catalog.xml and save it to a file (arf-w7-64-config1.output). If there are any differences, please verify all the configuration settings have been successfully applied to the system.

7. If you plan to use the same system for testing another configuration you should create a snapshot and clean up the configuration changes made in step 4. You can do this using the provided **cleanup.bat** tool or an alternate method. Run the cleanup.bat from the "c:\validationTestSuites\Windows\configurationTools" to restore a default configuration to the test system (as much as possible). Please note that the cleanup.bat file attempts to restore the system to an out-of-the-box state, not the state that was present before running the configuration scripts. As such, if you are testing a tool that requires a specific configuration on the system you may need to manually clean up the settings and/or modify the cleanup process to work with the tool.
8. Repeat steps 3 - 7 for each of the 8 configurations.

RHEL systems

1. Ensure that the following pre-requisites are met:

- 1.1 The /tmp must be created as separate partition and not as a folder on the root partition.
- 1.2 The test systems shall be named localhost.localdomain
- 1.3 Python 3.2 is installed

Instruction for installing Python 3.2.5 on RHEL:

```
# yum groupinstall "Development Tools"
# wget http://www.python.org/ftp/python/3.2.5/Python-3.2.5.tar.bz
# bzip2 -cd Python-3.* | tar xvf -
# cd Python-3.2.5
# ./configure
# make
# make install
```

Verify if Python was installed successful:

```
# python3 -V
(it should print out: Python 3.2.5)
```

- 1.4 If using the reference implementations the Java Runtime environment must be installed
 - 1.5 Set SELINUX to permissive mode: edit /etc/selinux/config, replace "SELINUX=enforcing" with "SELINUX=permissive" and reboot the system.
 - 1.6 It is recommended to configure a channel for your system which provides the mysql-server package: Login to <https://rhn.redhat.com> -> Subscriptions -> Registered systems -> Select your system -> Alter Channel Subscriptions -> Select " RHEL Desktop Workstation (v. 5 for 32-bit x86 or x86_64)" -> Change Subscription.
2. Create /validationTestSuites folder directly on the / partition. Copy the combinedDataStreams_<version>.zip file to the /validationTestSuites folder and extracted there. For the next steps we will assume that the zip file has been extracted to "/validationTestSuites". If you extract the files to a different location substitute that location in the following instructions.

The content of the /validationTestSuites folder should be as follows:

Parent folder	Folder or File name	Description
../RHEL/	ValidationFiles	SCAP datastream validation results
	Documents	OVAL Test Data and configuration spreadsheet
	configurationTools	Tools for configuring the system
	RHEL-datastream.xml	SCAP datastream
	catalog.xml	Expected results
	README.txt	Readme file
../tools/	compare.py	Python program used to compare the actual scan results with the expected results

3. Configure the target system(s), either using the provided tools and instructions below or using an alternate method of your choice:

- 3.1 Run the following commands as root:

```
# ]# cd /validationTestSuites/RHEL/configurationTools/  
# dos2unix *.sh  
# dos2unix *.py  
# chmod +x *.sh
```

- 3.2 Apply Configuration 1 to System 1

- 3.2.1 You may skip this step if the system is registered with RHN and has the proper channels configured. If the system is not registered with RHN, manually install the following RPM packages and configure the corresponding services.

Required services:

Service Name	Installed
httpd	yes
mysqld	yes

e.g. # yum install httpd mysql-server

- 3.2.2 Run as root:

```
# source config1.sh
```

3.3 Apply Configuration 2 to System 2

3.3.1 You may skip this step if the system is registered with RHN and has the proper channels configured. If the system is not registered with RHN, manually install the following RPM packages and configure the corresponding services:

Required services:

Service Name	Installed
httpd	yes
mysqld	yes

e.g. # yum install httpd mysql-server

3.3.2 Run as root:

```
# source config2.sh
```

3.4 Apply Configuration 3 to System 3

Run as root:

```
# source config3.sh
```

3.5 Apply Configuration 4 to System 4

3.5.1 You may skip this step if the system is registered with RHN. If the system is not registered with RHN, manually install the following services:

Required services:

Service Name	Installed
httpd	yes
mysqld	no

e.g. # yum install httpd

3.5.1 Run as root:

```
# source config4.sh
```

3.6 Import the consolidated data stream for RHEL (/validationTestSuites/RHEL/RHEL-datastream.xml) into the product

3.7. Perform configuration scans and export the evaluation results following the vendor tool instructions. The result file format may look like:

```
[arf | xccdf | oval | ocil | sc ]-[OS]-[platform]-config[1-4].xml
```

Where:

arf = ARF results

xccdf = XCCDF results

oval = OVAL results

ocil = OCIL results

sc = systems characteristics

OS = rhel5

Platform = 32 or 64

For instance, the results file for consolidated data stream on RHEL 5 64 bit, configuration 1 may look like: arf-rhel5-64-config1.xml

3.8 Compare the results with the expected results from the catalog.xml using the compare.py tool:

- From a command prompt, change the directory to /validationTestSuites/tools or the directory where the compare.py tool is located.
- Run: **python3 compare.py {RESULTSFILE} {CATALOGFILE} -i {SuiteID_from_catalogfile} -o {Output_file}**
- The compare.py script expects an ARF results file by default, if you want to use it to parse and XCCDF results file then you must also specify the -x switch to indicate that the file is XCCDF. If the XCCDF version is not version 1.2 you must

use the `-xv VersionNumber` switch to specify the XCCDF version. For additional help or to see all available options run `compare.py` with the `-h` switch.

For instance, using the `catalog.xml` file available for the RHEL datastream (`/validationTestSuites/RHEL/catalog.xml`), run:

```
python3 compare.py arf-rhel5-64-config1.xml catalog.xml -i G2.configuration1 -o arf-rhel5-64-config1.output
```

The script will flag all the differences between the results obtained by the product and the expected results from the `catalog.xml` and save it to output file (`arf-rhel5-64-config1.output`). If there are any differences, please verify all the configuration settings have been successfully applied to the system.