**How To Harden a server via OpenSCAP**

**Table of Contents**

[Introduction 2](#__RefHeading___Toc1050_239467253)

[WARNING! 2](#__RefHeading___Toc1664_1632227426)

[OS 2](#__RefHeading___Toc1666_1632227426)

[Definitions 2](#__RefHeading___Toc1054_239467253)

[Security content in various formats 3](#__RefHeading___Toc1668_1632227426)

[Why use SCAP? 3](#__RefHeading___Toc1670_1632227426)

[SCAP targets 4](#__RefHeading___Toc1672_1632227426)

[SCAP Components 4](#__RefHeading___Toc1056_239467253)

[SCAP CONTENT 4](#__RefHeading___Toc1058_239467253)

[SCAP SCANNERS 4](#__RefHeading___Toc1060_239467253)

[Profile types 5](#__RefHeading___Toc1062_239467253)

[ISM profiles 5](#__RefHeading___Toc1064_239467253)

[Scanning remote machines 6](#__RefHeading___Toc1072_239467253)

[Virtual Machine Alternatives 6](#__RefHeading___Toc1074_239467253)

[VM 6](#__RefHeading___Toc1076_239467253)

[Containers 7](#__RefHeading___Toc1078_239467253)

[Installation 7](#__RefHeading___Toc1080_239467253)

[Ansible server 7](#__RefHeading___Toc1082_239467253)

[Yum 7](#__RefHeading___Toc1084_239467253)

[Compliance As Code 7](#__RefHeading___Toc1086_239467253)

[OpenSCAP ssh tool 7](#__RefHeading___Toc1088_239467253)

[OpenSCAP workbench via source 7](#__RefHeading___Toc1090_239467253)

[CIS Baseline 8](#__RefHeading___Toc1092_239467253)

[Target server 8](#__RefHeading___Toc1094_239467253)

[Build the latest SCAP content / playbooks 8](#__RefHeading___Toc1096_239467253)

[Running OpenSCAP 9](#__RefHeading___Toc1098_239467253)

[Setup passwordless ssh 9](#__RefHeading___Toc1100_239467253)

[Scan 9](#__RefHeading___Toc1102_239467253)

[Directly on target 9](#__RefHeading___Toc1104_239467253)

[From Ansible server 9](#__RefHeading___Toc1106_239467253)

[Creating a tailoring file 12](#__RefHeading___Toc1108_239467253)

[Generate html report: 12](#__RefHeading___Toc1112_239467253)

[Create hardening scripts 12](#__RefHeading___Toc1114_239467253)

[Kickstart 13](#__RefHeading___Toc1116_239467253)

[Ansible 13](#__RefHeading___Toc1190_239467253)

[bash 14](#__RefHeading___Toc1192_239467253)

[Using hardening scripts 15](#__RefHeading___Toc1194_239467253)

[Kick-start 15](#__RefHeading___Toc1116_2394672531)

[Ansible 15](#__RefHeading___Toc1118_239467253)

[bash 16](#__RefHeading___Toc1120_239467253)

WIP

# Introduction

## **WARNING**!

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. There are elements of these guides that may make parts of, or the entire server inaccessible or inoperable!

## OS

OpenSCAP is designed to be implemented on many Linuxes, including, Red Hat Enterprise Linux (RHEL), CentOS, Fedora, Ubuntu, Debian, ... -- as well as products including – Firefox, Chromium, JRE, …

The following document is tailored for implementation on RHEL. Please feel free to go to <https://github.com/ComplianceAsCode/content> for more information

## Definitions

* OpenSCAP is sometimes referred to as OSCAP. It means Open Security Content Automation Protocol.
* Hardening, in this context, is the process of securing a system by reducing its [surface of vulnerability](https://en.wikipedia.org/wiki/Attack_surface), which is larger when a system performs more functions; in principle a single-function system is more secure than a multipurpose one. Reducing available ways of attack typically includes changing default passwords, the removal of unnecessary software, unnecessary usernames or logins, and the disabling or removal of unnecessary services.

There are various methods of hardening Linux systems. This may involve, among other measures, applying a patch to the kernel, closing open network ports; and setting up intrusion detection systems, firewall and other intrusion prevention systems.

## **Security content in various formats**

[](https://github.com/ComplianceAsCode/content/blob/master/docs/readme_images/nist_logo.svg)Drawing 1: National Institute of Standards and Technology

[](https://github.com/ComplianceAsCode/content/blob/master/docs/readme_images/ansible_logo.svg)Drawing 2: Ansible Scripts

[](https://github.com/ComplianceAsCode/content/blob/master/docs/readme_images/bash_logo.png)Drawing 3: Bash Scripts

"SCAP content" refers to documents in the XCCDF, OVAL and Source DataStream formats. These documents can be presented in different forms and by different organizations to meet their security automation and technical implementation needs. For general use, we recommend Source DataStreams because they contain all the data you need to evaluate and put machines into compliance. The datastreams are part of our release ZIP archives.

"Ansible content" refers to Ansible playbooks generated from security profiles. These can be used both in check-mode to evaluate compliance, as well as run-mode to put machines into compliance. We publish these on Ansible Galaxy as well as in release ZIP archives.

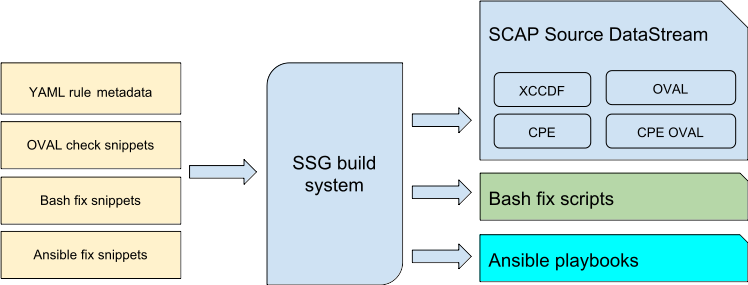
"Bash fix files" refers to Bash scripts generated from security profiles. These are meant to be run on machines to put them into compliance. We recommend using other formats but understand that for some deployment scenarios bash is the only option.

### Why use SCAP?

Multiple organizations are able to efficiently develop and utilise security content. By taking advantage of the powerful build system, as much redundancy as possible is avoided.

The build system combines the easy-to-edit YAML rule files with OVAL checks, Ansible task snippets, Bash fixes, and other files. Templating is provided at every step to avoid boilerplate. Security identifiers (CCE, NIST ID, STIG, ...) appear in all of our output formats but are all sourced from the YAML rule files.

You choose the specific security content format that best suits your organization's needs.

[](https://github.com/ComplianceAsCode/content/blob/master/docs/readme_images/build_schema.svg)

This uses OpenControl-inspired YAML rule format for input. Write once and generate security content in XCCDF, Ansible, and others.

## SCAP targets

Our security content can be used to scan bare-metal machines, virtual machines, virtual machine images (qcow2 and others), containers (including Docker), and container images.

We use platform checks to detect whether we should or should not evaluate some of the rules. For example: separate partition checks make perfect sense on bare-metal machines but go against recommended practices on containers.

## SCAP Components

SCAP content and SCAP scanners are the two main aspects of the Security Content Automation Protocol.

### SCAP CONTENT

SCAP content modules are freely available content developed by the National Institute of Standards and Technologies ([NIST](https://csrc.nist.gov/Projects)) and its industry partners. The content modules are made from "secure" configurations that are agreed to by NIST and its SCAP partners.

### SCAP SCANNERS

An SCAP scanner is a tool that compares a target computer or [a](https://www.lifewire.com/what-is-a-web-application-3486637)pplication's configuration and/or patch level against that of the SCAP content baseline.

The tool will note any deviations and produce a report. Some SCAP scanners also have the ability to correct the target computer and bring it into compliance with the standard baseline.

### Profile types

There are a few profiles types for RHEL and CentOS for hardening. Below are a list of those for RHEL 7 and 8:

1. No hardening – a.k.a. base installation
2. RHEL 7

* [C2S for Red Hat Enterprise Linux 7](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-C2S.html)
* [Criminal Justice Information Services (CJIS) Security Policy](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-cjis.html)
* [Unclassified Information in Non-federal Information Systems and Organizations (NIST 800-171)](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-nist-800-171-cui.html)
* [PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 7](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-pci-dss.html)
* [Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-rht-ccp.html)
* [DISA STIG for Red Hat Enterprise Linux 7](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-stig.html)
* [VPP – Protection Profile for Virtualization v. 1.0 for Red Hat Enterprise Linux Hypervisor (RHELH)](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-rhelh-vpp.html)
* [Standard System Security Profile](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-standard.html)
* [United States Government Configuration Baseline (USGCB / STIG) – DRAFT](http://static.open-scap.org/ssg-guides/ssg-rhel7-guide-ospp.html)

1. RHEL 8

* [Criminal Justice Information Services (CJIS) Security Policy](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-cjis.html)
* [Health Insurance Portability and Accountability Act (HIPAA)](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-hipaa.html)
* [Unclassified Information in Non-federal Information Systems and Organizations (NIST 800-171)](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-cui.html)
* [PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-pci-dss.html)
* [Red Hat Corporate Profile for Certified Cloud Providers (RH CCP)](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-rht-ccp.html)
* [Standard System Security Profile for Red Hat Enterprise Linux 8](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-standard.html)
* [Protection Profile for General Purpose Operating Systems](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-ospp.html)
* [Australian Cyber Security Centre (ACSC) Information Security Manual (ISM) Official](http://static.open-scap.org/ssg-guides/ssg-rhel8-guide-ism_o.html)

**NOTE:** These are not in any order of hardening strength, but should be examined on a case by case basis, to see whether they are suited to <Customer>

### ISM profiles

At the time of writing, only ISM\_O (ACSC Information Security Manual (ISM) Official) profile is the only ISM profile included in the standard OpenSCAP repositories.

There are plans to have a locally produced ISM profiles. They will fall into one of the three following categories:

* ISM\_P for the PROTECTED systems
* ISM\_S for the SECRET systems
* ISM\_TS for the TOP SECRET systems

These profiles should be considered a work in progress and may remain unsupported by Red Hat or the ComplianceAsCode community.

**NOTES:**

1. Each of the ISM protocols, above utilise a fourth profile, ISM\_O, which contains all of the rules that are common to all the three above profiles
2. The higher classification profile DO NOT include the lesser classification profiles. i.e. ISM\_TS does not utilise ISM\_S, which does not utilise ISM\_P

### **Scanning remote machines**

The **oscap-ssh** is a simple tool for scanning remote machines with OpenSCAP over network and collecting results.

The tool uses SSH connection to copy the SCAP content to a remote machine, then it runs an evaluation of the target system and downloads the results back. *The remote machine needs to have OpenSCAP installed.*

The tool can evaluate source DataStreams and OVAL files. Usage of the tool mimics usage and options of oscap tool.

It is recommended to use this remote operations tool so that all the results and reports are centralised.

### Virtual Machine Alternatives

#### VM

OpenSCAP provides a simple tool to evaluate virtual machines called **oscap-vm**.

Usage of the tool mimics usage and options of oscap tool.

**NB:** The tool can scan given virtual machine directly from the virtualisation host. e.g. RHEL host with VM via KVM.

#### Containers

OpenSCAP provides a simple tool to evaluate virtual machines called **oscap-docker**, this utility also uses offline scanning, so it doesn’t install anything in the guest, doesn’t require OpenSCAP installed in the guest and it doesn’t create or change anything in the guest’s filesystem. Usage of the tool mimics usage and options of oscap tool.

**NOTE:** The filesystem of the container is mounted to a directory on the host. The mounted filesystem is read-only. OpenSCAP then assess the container from the host. Therefore no agent is installed in the container and container is not touched or changed in any way.

# Installation

## Ansible server

- if not done via the kickstart of this server

### Yum

* ssh ${ansible\_server}
* sudo yum install -y ansible git openscap scap-security-guide scap-workbench

### Compliance As Code

* cd /${PATH\_TO\_OPENSCAP\_CONTENT} #e.g. /home/ahamilto/Ansible/
* git clone https://github.com/ComplianceAsCode/content.git

OR/

* cd content; git pull

### OpenSCAP ssh tool

* cd /usr/share/openscap
* git clone https://github.com/mpreisler/oscap-ssh.git

OR/

* cd /usr/share/openscap/; git pull

### OpenSCAP workbench via source

The following builds the source code, rather than using yum:

* wget -O ${PATH\_TO\_OPENSCAP\_CONTENT}/OpenSCAP\_WorkBench/ https://github.com/OpenSCAP/scap-workbench/releases
* cd ${PATH\_TO\_OPENSCAP\_CONTENT}/OpenSCAP\_WorkBench/
* tar jxf scap-workbench-${ver}.tar.bz2
* cd scap-workbench-${ver}; less README.md =>
* mkdir build; cd build; \
* cmake -DOPENSCAP\_LIBRARIES:PATH=/usr/lib64/libopenscap.so \

-DOPENSCAP\_INCLUDE\_DIRS:PATH=/usr/include/openscap/ \

-DOPENSCAP\_VERSION:STRING="1.2.1" ../

* make install

### CIS Baseline

* cd /${PATH\_TO\_OPENSCAP\_CONTENT}
* git clone <https://github.com/ggatward/Ansible-RHEL8-CIS-Benchmarks.git>

**NOTES:**

1. Role for RHEL: see <https://martin.preisler.me/2013/11/xccdf-tailoring/>

## Target server

- if not done via the kickstart

* ssh target\_server
* sudo yum install -y git openscap scap-security-guide
* cd /${PATH\_TO\_OPENSCAP\_CONTENT}

[e.g.] /usr/share/openscap/

* git clone https://github.com/ComplianceAsCode/content.git

# Build the latest SCAP content / playbooks

- directly from the project

* cd /usr/share/scap-security-guide
* mkdir build
* podman run -e SCAP\_PRODUCT='rhel8' -v ./build:/build:z \

quay.io/smileyfritz/cacfactory:latest

Error: relabel failed "/usr/share/scap-security-guide/build": relabeling content in /usr is not allowed => Needs investigating

**NOTE:** The latest SCAP datastream files and Ansible playbooks will then be available in the ‘build’ directory

# Running OpenSCAP

- over system => Produce hardening

## Setup passwordless ssh

* [Only once on server] ssh-keygen -t rsa
* [On server] ssh-copy-id username@remotemachine
* [Possibly on client] chmod 755 ~/.ssh/authorized\_keys
* [Possibly on client] chmod 700 ~/.ssh/id\_\*

## Scan

**NOTE:** OVAL XML, does not offer remediations fixes like is found in XCCDF XML

### Directly on target

- MUST have OpenSCAP installed [via `sudo yum install -y openscap scap-security-guide]

* ssh ${target\_server}
* cd /${PATH\_TO\_RESULTS\_FILES}

e.g. /var/tmp/openscap\_results

* oscap xccdf eval --results goldsvr\_${ServerType}\_${HardeningType}.xml \

--profile ${PROFILE} \

--fetch-remote-resources \

${ContentXML}

### From Ansible server

- This is the preferred method so that all the results and reports are centralised.

- The remote machine needs to have OpenSCAP installed

[via `sudo yum install -y openscap scap-security-guide]

* ssh ${ansible\_server}
* cd *${PATH\_TO\_RESULTS\_FILES}*/${TargetHostFQDN}

#e.g. /var/tmp/openscap\_results/${TargetHostFQDN}

* oscap-ssh ${SID}@${TargetHostFQDN} 22 xccdf eval \

--results goldsvr\_${ServerType}\_${HardeningType}.xml \

--results-arf goldsvr\_${ServerType}\_${HardeningType}-arf.xml \

--report goldsvr\_${ServerType}\_${HardeningType}.html \

--fetch-remote-resources \

--profile xccdf\_org.ssgproject.content\_profile\_e8 \

[--tailoring-file /usr/share/openscap/schemas/xccdf/${TailoringFile}.xml \]

/usr/share/xml/scap/ssg/content/${ContentXML}

**NOTES:** For the above commands;

1. ${ContentXML} available are:

[obtained by `ls /usr/share/xml/scap/ssg/content/`]:

ssg-firefox-cpe-dictionary.xml

ssg-firefox-cpe-oval.xml

ssg-firefox-ds-1.2.xml

ssg-firefox-ds.xml

ssg-firefox-ocil.xml

ssg-firefox-oval.xml

ssg-firefox-xccdf.xml

ssg-jre-cpe-dictionary.xml

ssg-jre-cpe-oval.xml

ssg-jre-ds-1.2.xml

ssg-jre-ds.xml

ssg-jre-ocil.xml

ssg-jre-oval.xml

ssg-jre-xccdf.xml

ssg-rhel6-cpe-dictionary.xml

ssg-rhel6-cpe-oval.xml

ssg-rhel6-ds-1.2.xml

ssg-rhel6-ds.xml

ssg-rhel6-ocil.xml

ssg-rhel6-oval.xml

ssg-rhel6-xccdf.xml

ssg-rhel7-cpe-dictionary.xml

ssg-rhel7-cpe-oval.xml

ssg-rhel7-ds-1.2.xml

ssg-rhel7-ds.xml

ssg-rhel7-ocil.xml

ssg-rhel7-oval.xml

ssg-rhel7-xccdf.xml

ssg-rhel8-cpe-dictionary.xml

ssg-rhel8-cpe-oval.xml

ssg-rhel8-ds-1.2.xml

ssg-rhel8-ds.xml

ssg-rhel8-ocil.xml

ssg-rhel8-oval.xml

ssg-rhel8-xccdf.xml

1. ${Profile} [Id’s] available are [obtained by `oscap info "/tmp/tmp.??????????/input.xml"`]:

...

Title: Protection Profile for General Purpose Operating Systems

Id: xccdf\_org.ssgproject.content\_profile\_ospp

Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_pci-dss

Title: [DRAFT] DISA STIG for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_stig

Title: Australian Cyber Security Centre (ACSC) Essential Eight

Id: xccdf\_org.ssgproject.content\_profile\_e8

...

1. ${TailoringFile}.xml is a customisation of a specific type of profile

- See [How To Tailor a Profile by Martin Preisler](https://martin.preisler.me/2013/11/xccdf-tailoring/)

**NB:** ${SID} *MUST* have sudo access

## Creating a tailoring file

- This is not normally needed, but if you need to customise your hardening, it can be done here.

- a.k.a. Standard/Gold file

* ssh -X ${ansible\_server}
* /usr/local/bin/scap-workbench
* Open goldsvr\_${ServerType}\_${HardeningType}.xml [e.g. goldsvr\_CIS.xml]
* Click "Customize" -> this will warn that you are creating a "New Profile" but in fact you are about to create a tailoring file.
* Enter the name for the tailoring file ${TailoringFile}.xml
* (De)Select the rules you wish to (not) execute
* Press "OK"

**NOTES:**

1. The auditd hardening assumes that [for ASD] ace\_audit is installed
2. See <https://github.com/OpenSCAP/scap-workbench/releases/download/1.1.0/user_manual.html>
3. There is NO interim save

## Generate html report:

- This is only to be used if the html report is not already generated in execution of the [Scan](#_toc215) section above.

* ssh ${ansible\_server}
* cd /${PATH\_TO\_RESULTS\_FILES} #e.g. /var/tmp/openscap\_results
* sudo oscap xccdf generate report --output goldsvr\_${ServerType}\_${HardeningType}.html \

goldsvr\_${ServerType}\_${HardeningType}.xml

# Create hardening scripts

**WARNING:** It is advisable to examine ${TargetHost}.xml before the following are run, with a view of removing unwanted elements! There may be resultant procedures of these scripts that may make parts of, or the entire server inaccessible or inoperable! Ansible and bash scripts do **not** have ***automatic*** rollbacks in the case of a

## Kickstart

* Change kickstart [\*.cfg] template to include

e.g.

...

# Create primary system partitions (required for installs)

...

# Harden installation with Essential Eight profile

# For more details and configuration options see

# <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/index>

# addon-org\_fedora\_oscap\_kickstart-commands-for-addons-supplied-with-the-rhel-installation-program

**%addon org\_fedora\_oscap**

**content-type = scap-security-guide**

**profile = xccdf\_org.ssgproject.content\_profile\_e8**

**%end**

# Packages selection (%packages section is required)

%packages

…

**NOTE:** A list of possible oscap kick-start profiles are via:

`grep content\_profile /usr/share/scap-security-guide/kickstart/\*`

## Ansible

* ssh ${ansible\_server}
* cd /${PATH\_TO\_RESULTS\_FILES} #e.g. /var/tmp/openscap\_results
* sudo oscap xccdf generate fix --profile xccdf\_org.ssgproject.content\_profile\_e8 \

--fetch-remote-resources \

[--tailoring-file /usr/share/openscap/schemas/xccdf/${TailoringFile}.xml \] \

--fix-type **ansible** ${TargetHost}.xml > ${TargetHost}.**yml**

**NOTES:**

1. There is a bug in the oscap, so “--output ${TargetHost}.bash” doesn’t produce any output, so need “> ${TargetHost}.yml” at end.
2. ${Profile} [Id’s] available are [obtained by `oscap info "/tmp/tmp.??????????/input.xml"`]:

...

Title: Protection Profile for General Purpose Operating Systems

Id: xccdf\_org.ssgproject.content\_profile\_ospp

Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_pci-dss

Title: [DRAFT] DISA STIG for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_stig

Title: Australian Cyber Security Centre (ACSC) Essential Eight

Id: xccdf\_org.ssgproject.content\_profile\_e8

...

## bash

* ssh ${ansible\_server}
* cd /${PATH\_TO\_RESULTS\_FILES} #e.g. /var/tmp/openscap\_results
* sudo oscap xccdf generate fix --profile xccdf\_org.ssgproject.content\_profile\_e8 \

--fetch-remote-resources \

[--tailoring-file /usr/share/openscap/schemas/xccdf/${TailoringFile}.xml \] \

--fix-type **bash** ${TargetHost}.xml > ${TargetHost}.**bash**

**NOTES:**

1. There is a bug in the oscap, so “--output ${TargetHost}.bash” doesn’t produce any output, so need “> ${TargetHost}.bash” at end.
2. ${Profile} [Id’s] available are [obtained by `oscap info "/tmp/tmp.??????????/input.xml"`]:

...

Title: Protection Profile for General Purpose Operating Systems

Id: xccdf\_org.ssgproject.content\_profile\_ospp

Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_pci-dss

Title: [DRAFT] DISA STIG for Red Hat Enterprise Linux 8

Id: xccdf\_org.ssgproject.content\_profile\_stig

Title: Australian Cyber Security Centre (ACSC) Essential Eight

Id: xccdf\_org.ssgproject.content\_profile\_e8

...

# Using hardening scripts

**WARNING:** Take a snapshot / backup of the server prior to hardening in case a rollback is necessary. Ansible and bash scripts do **not** have ***automatic*** rollbacks in the case of a failure!

## Kick-start

* Kick-start the server

**NOTE:** This only provides a “one off” hardening. It is recommended that the server is regularly checked via Ansible or bash!

## Ansible

* ssh ${ansible\_server}
* PATH\_TO\_DESIRED\_SCRIPT = /usr/share/scap-security-guide/ansible/

OR/

* PATH\_TO\_DESIRED\_SCRIPT = *${PATH\_TO\_RESULTS\_FILES}*

OR/

* PATH\_TO\_DESIRED\_SCRIPT = *${PATH\_TO\_GOLD\_RULE}*
* ansible-playbook -i "${TargetHost1},...,${TargetHostn}," -bkK \

/${PATH\_TO\_DESIRED\_SCRIPT}/${OSCAP\_SCRIPT}.yml

**NBs:**

1. The Ansible ***SHOULD*** be run regularly to ensure that the server is still compliant!
2. ***MUST***:
   * put the "," [incl. The parentheses] after the last hostname in the above command
   * ssh to ALL ${TargetHost}s prior to running Ansible to establish entries in known\_hosts

**NOTE:** Each of the Ansible Playbooks contain instructions on how to deploy them. In the top comments.

## bash

* PATH\_TO\_DESIRED\_SCRIPT = /usr/share/scap-security-guide/ansible/
* OR/
* PATH\_TO\_DESIRED\_SCRIPT = *${PATH\_TO\_RESULTS\_FILES}*
* OR/
* PATH\_TO\_DESIRED\_SCRIPT = *${PATH\_TO\_GOLD\_RULE}*
* bash -x ${PATH\_TO\_DESIRED\_SCRIPT}/${OSCAP\_SCRIPT}

**NOTE:** Each of the bash scripts contain instructions on how to deploy them. In the top comments.