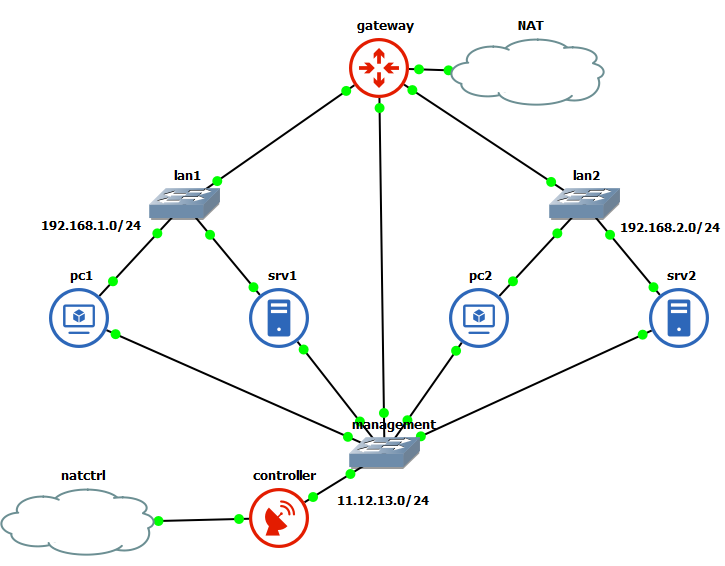
**Mouad Zitouni   
Administration et Sécurité Linux et Windows  
  
LABO 1**  


**Pour la facilité et la lecture on placera ces paramètres en variables. La variable “type” permet de donner une indication sur les fichiers produits.**

target="srv1"

type="$target-std-before"

data\_stream="/usr/share/xml/scap/ssg/content/ssg-centos7-ds-1.2.xml"

profile="xccdf\_org.ssgproject.content\_profile\_standard"

cpe\_dict="/usr/share/openscap/cpe/openscap-cpe-dict.xml"

Un “Data stream” est un fichier qui contient des définitions, des critères de référence, des profils et des règles individuelles. Chaque règle détermine l’applicabilité et les exigences de conformité. Voici la structure d’un “Data stream” :

Data stream

├── xccdf

| ├── benchmark

| ├── profile

| | ├──rule reference

| | └──variable

| ├── rule

| ├── human readable data

| ├── oval reference

├── oval ├── ocil reference

├── ocil ├── cpe reference

└── cpe └── remediation

Un profil est un ensemble de règles basées sur une politique de sécurité telle que OSPP, PCI-DSS et Health

Insurance Portability and Accountability Act (HIPAA). Il permet d’auditer le système de manière automatisée

Pour vérifier la conformité aux normes de sécurité.

--------------------------------------------------------------------------------------------------------------------------------------  
  
**Labo sur SRV2 :**

Paramètres de départ de l’exercice

A partir d’une station de contrôle qui est capable de gérer des cibles sur un réseau de gestion, il est proposé :

d’auditer les politiques de sécurité d’un hôte RedHat nommé srv1.

• cible: machine srv2

• data stream: ssg-centos7-ds-1.2.xml

• profile: xccdf\_org.ssgproject.content\_profile\_standard

**Connection SSH avec une clé RSA 4096**

En considérant que la cible à auditer est srv2, à partir de la station de contrôle on prépare la connexion vers la cible :

**Config SSH pour SRV2 :**

Target2="srv2"

type="$target-std-before"

data\_stream="/usr/share/xml/scap/ssg/content/ssg-centos7-ds-1.2.xml"

profile="xccdf\_org.ssgproject.content\_profile\_standard"

cpe\_dict="/usr/share/openscap/cpe/openscap-cpe-dict.xml"

**1.3.4 Tâche 2.**ssh-keygen -b 4096 -t rsa -f $HOME/.ssh/id\_rsa2 -q -N "" 🡺 (id\_rsa2 est la clé pour le SRV2)

Target2="srv2" 🡺 on change la variable est on l’attribue au SRV2

ssh-copy-id $target2 🡺 on la copie sur la sible (srv2)   
  
ssh $target2 "yum -y install scap-security-guide" 🡺 Installer oscap-scanner sur le SRV2

**1.3.5 Tâche 3.**

version=0.1.64

copr\_repo="http://copr.fedoraproject.org/coprs/openscapmaint/openscap-latest/repo/epel-

7"

curl -L $copr\_repo/openscapmaint-openscap-latest-epel-7.repo \

-o /etc/yum.repos.d/openscapmaint-openscap-latest-epel-7.repo

yum repolist

yum -y install unzip openscap-utils scap-security-guide

/usr/bin/cp -f $cpe\_dict \

$cpe\_dict.dist

/usr/bin/cp -f /usr/share/openscap/cpe/openscap-cpe-oval.xml \

/usr/share/openscap/cpe/openscap-cpe-oval.xml.dist

maint\_url="https://raw.githubusercontent.com/OpenSCAP/openscap/maint-1.3"

curl -L $maint\_url/cpe/openscap-cpe-dict.xml \

-o $cpe\_dict

curl -L $maint\_url/openscap-cpe-oval.xml \

-o /usr/share/openscap/cpe/openscap-cpe-oval.xml

ssg\_url="https://github.com/ComplianceAsCode/content/releases/download"

curl -L $ssg\_url/v${version}/scap-security-guide-${version}-oval-5.10.zip \

-o ./scap-security-guide-${version}-oval-5.10.zip

unzip scap-security-guide-${version}-oval-5.10.zip

/usr/bin/cp -f scap-security-guide-${version}-oval-5.10/\*.xml \

/usr/share/xml/scap/ssg/content/

« Dans cette procédure, on installe les paquets openscap-utils et scap-security-guide à partir

du dépôt de paquets OpenSCAP COPR, les guides de sécurité SSG à jour en format OVAL, un dictionnaire et

un fichier OVAL pour les CPE.

**1.3.6 Tâche 4.**  
Veuillez télécharger localement et décompresser la dernière version du contenu **Scap Security Guide (SSG) v0.1.64** et examiner les fichiers que vous trouvez dans le nouveau dossier :

./

├── ansible

├── bash

├── guides

├── kickstart

├── ssg-centos7-\*.xml

├── ssg-centos8-\*.xml

├── ...

├── ssg-jre-\*.xml

├── ssg-rhel7-\*.xml

├── ssg-rhel8-\*.xml

├── ssg-rhel9-\*.xml

├── ssg-ubuntu1604-\*.xml

├── ssg-ubuntu1804-\*.xml

├── ssg-ubuntu2004-\*.xml

└── tables

Le projet SSG construit des contenus de sécurité dans différents formats : XCCDF, OVAL et Source DataStream.

Ces documents peuvent être présentés sous différentes formes et par différentes organisations pour répondre

à leurs besoins en matière de sécurité, d’automatisation et de mise en oeuvre technique.

Pour une utilisation générale, on recommandera les Source DataStreams car ils contiennent toutes les données

dont on a besoin pour évaluer et mettre les machines en conformité.

“Ansible” fait référence aux playbooks Ansible générés à partir des profils de de sécurité. Ceux-ci peuvent être

utilisés à la fois en mode check-mode pour évaluer la conformité, ainsi qu’en mode exécution pour mettre les

machines en conformité.

“Bash fix files” fait référence aux scripts Bash générés à partir des profils de sécurité. Ils sont destinés à être

exécutés sur les machines pour les mettre en conformité. On recommandera l’utilisation d’autres formats mais

il parfaitement compréhensible que pour certains scénarios de déploiement Bash soit la seule option.

Le système de construction combine les fichiers de règles YAML faciles à éditer avec les vérifications OVAL, des

extraits de tâches Ansible, des corrections Bash et d’autres fichiers. La modélisation est fournie à chaque étapeafin d’éviter les modèles passe-partout. Les identifiants de sécurité (CCE, NIST ID, STIG, …) apparaissent dans

tous nos formats de sortie mais proviennent tous de fichiers de règles YAML.

prodtype: rhel7

title: 'Configure The Number of Allowed Simultaneous Requests'

description: |-

The <tt>MaxKeepAliveRequests</tt> directive should be set and configured to

<sub idref="var\_max\_keepalive\_requests" /> or greater by setting the following

in <tt>/etc/httpd/conf/httpd.conf</tt>:

<pre>MaxKeepAliveRequests <sub idref="var\_max\_keepalive\_requests" /></pre>

rationale: |-

Resource exhaustion can occur when an unlimited number of concurrent requests

are allowed on a web site, facilitating a denial of service attack. Mitigating

this kind of attack will include limiting the number of concurrent HTTP/HTTPS

requests per IP address and may include, where feasible, limiting parameter

values associated with keepalive, (i.e., a parameter used to limit the amount of

time a connection may be inactive).

severity: medium

identifiers:

cce: "80551-5"

La page d’accueil du SSG est https://www.open-scap.org/security-policies/scap-security-guide/.

On trouvera aussi des ressources dignes d’intérêt sur :

• SSG User Manual

• SSG Developer Guide

• Compliance As Code Blog

**1.3.7 Tâche 5.**

Découvrir les binaires oscap :

=>oscap -h

//// AFFICHE LES COMMANDE LIE A L'OUTIL OSCAP

oscap

OpenSCAP command-line tool

Usage: oscap [options] module operation [operation-options-and-arguments]

oscap options:

-h --help - show this help

-q --quiet - quiet mode

-V --version - print info about supported SCAP versions

Commands:

ds - DataStream utilities

oval - Open Vulnerability and Assessment Language

xccdf - eXtensible Configuration Checklist Description Format

cvss - Common Vulnerability Scoring System

cpe - Common Platform Enumeration

cve - Common Vulnerabilities and Exposures

cvrf - Common Vulnerability Reporting Framework

info - info module

==>oscap info -h

Affiche des informations lié au profil SSG

==>oscap xccdf -h

Permet de faire une evaluation dirigé par un fichier XCCDF

oscap -> xccdf

eXtensible Configuration Checklist Description Format

Usage: oscap [options] xccdf command [command-specific-options]

Commands:

eval - Perform evaluation driven by XCCDF file and use OVAL as checking engine

resolve - Resolve an XCCDF document

validate - Validate XCCDF XML content

validate-xml - Validate XCCDF XML content

export-oval-variables - Export XCCDF values as OVAL external-variables document(s)

generate - Convert XCCDF Benchmark to other formats

remediate - Perform remediation driven by XCCDF TestResult file or ARF.

==> oscap xccdf eval -h

oscap -> xccdf -> eval

Perform evaluation driven by XCCDF file and use OVAL as checking engine

Usage: oscap [options] xccdf eval [options] INPUT\_FILE [oval-definitions-files]

==>oscap xccdf generate -h

Genere un benchmark dans un certain format a partir d'un fichier XCCDF

oscap -> xccdf -> generate

Convert XCCDF Benchmark to other formats

Usage: oscap [options] xccdf generate [options] <subcommand> [sub-options] benchmark-file.xml

Generate options:

--profile <profile-id> - Apply profile with given ID to the Benchmark before further processing takes place.

==> oscap xccdf generate fix -h

oscap -> xccdf -> generate -> fix

Generate a fix script from an XCCDF file

Usage: oscap [options] xccdf generate [options] fix [options] xccdf-file.xml

==> oscap xccdf generate guide -h

oscap -> xccdf -> generate -> guide

Generate security guide

Usage: oscap [options] xccdf generate [options] guide [options] xccdf-file.xml

Generate options:

--profile <profile-id> - Apply profile with given ID to the Benchmark before further processing takes place.

Guide Options:

--output <file> - Write the document into file.

--hide-profile-info - Do not output additional information about selected profile.

--benchmark-id <id> - ID of XCCDF Benchmark in some component in the datastream that should be used.

(only applicable for source datastreams)

==> oscap-ssh -h

oscap-ssh -- Tool for running oscap over SSH and collecting results.

Usage:

$ oscap-ssh user@host 22 info INPUT\_CONTENT

$ oscap-ssh user@host 22 xccdf eval [options] INPUT\_CONTENT

**1.3.8 Tâche 6. Afficher des informations sur un Data Stream (DS) SCAP**data\_stream="/usr/share/xml/scap/ssg/content/ssg-centos7-ds-1.2.xml"

oscap info --fetch-remote-resources $data\_stream

La sortie est explicite :

• Document type décrit le format du fichier. Les types courants sont XCCDF, OVAL, “Data stream” source

et “Data stream” résultat.

• Imported est la date à laquelle le fichier a été importé pour être utilisé avec OpenSCAP. Comme OpenSCAP

utilise le système de fichiers local et ne possède pas de format de base de données propriétaire, la

date d’importation est la même que la date de modification du fichier.

• Stream est l’ID du “Data stream”.

• Version est la version de la norme SCAP.

• Checklists liste les listes de contrôle disponibles incorporées dans le “Data stream” que l’on peut utiliser

pour l’attribut de ligne de commande --benchmark-id avec oscap xccdf eval. Les

informations détaillées sont également imprimées pour chaque liste de contrôle.

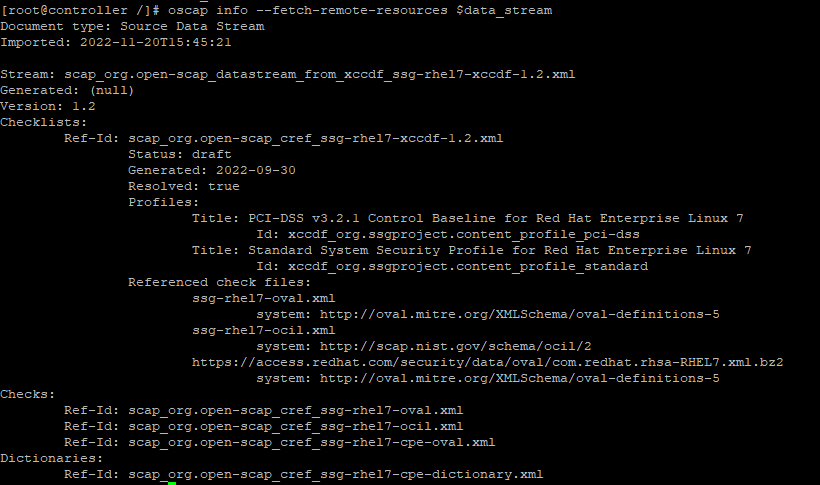
• Status est le statut du Benchmark XCCDF. Les valeurs courantes incluent accepted”, “draft”, “deprecated”

et “incomplete”. Veuillez vous référer à la spécification XCCDF pour plus de détails.

• Generated date est la date à laquelle le fichier a été créé ou généré. Cette date est indiquée pour les

fichiers XCCDF et les listes de contrôle et provient de l’élément XCCDF “Status”.

data\_stream="/usr/share/xml/scap/ssg/content/ssg-centos7-ds-1.2.xml"

oscap info --fetch-remote-resources $data\_stream  
  


**1.3.9 Tâche 7. Scan avec un Data Stream (DS) SCAP :**

L’objectif principal d’OpenSCAP est d’effectuer des analyses de configuration et de vulnérabilité d’un système.

OpenSCAP est capable d’évaluer les “Data stream” sources SCAP, les benchmarks XCCDF et les définitions OVAL et de générer les résultats appropriés.

Ici on propose d’utiliser l’exécutable oscap-ssh qui exécute le scan sur le SRV2 avec toutes les informations appropriées et qui récupère les résultats :

type="$target2-std-before"

profile="xccdf\_org.ssgproject.content\_profile\_standard"

oscap-ssh --sudo root@$target 22 xccdf eval \

--fetch-remote-resource \

--profile $profile \

--results $type-results.xml \

--report $type-report.html \

--oval-results \

--cpe $cpe\_dict \

$data\_stream

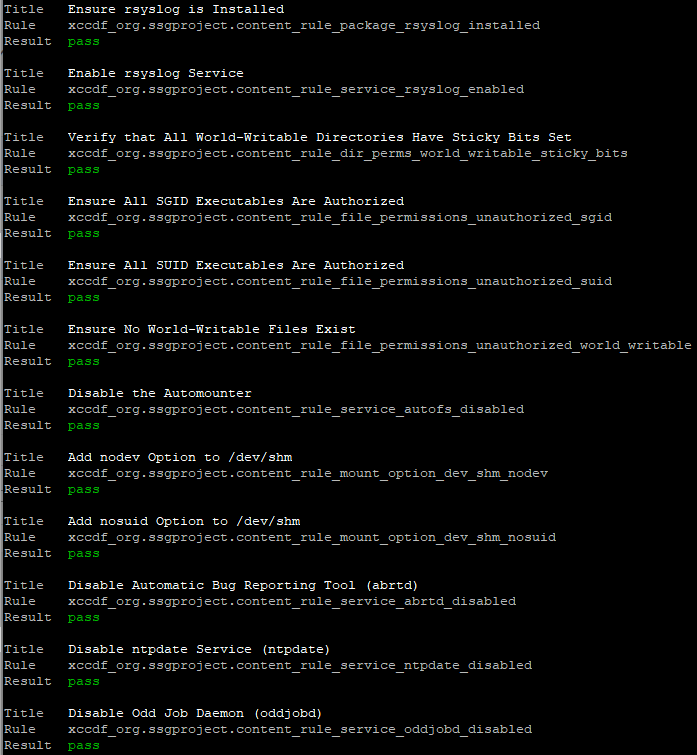
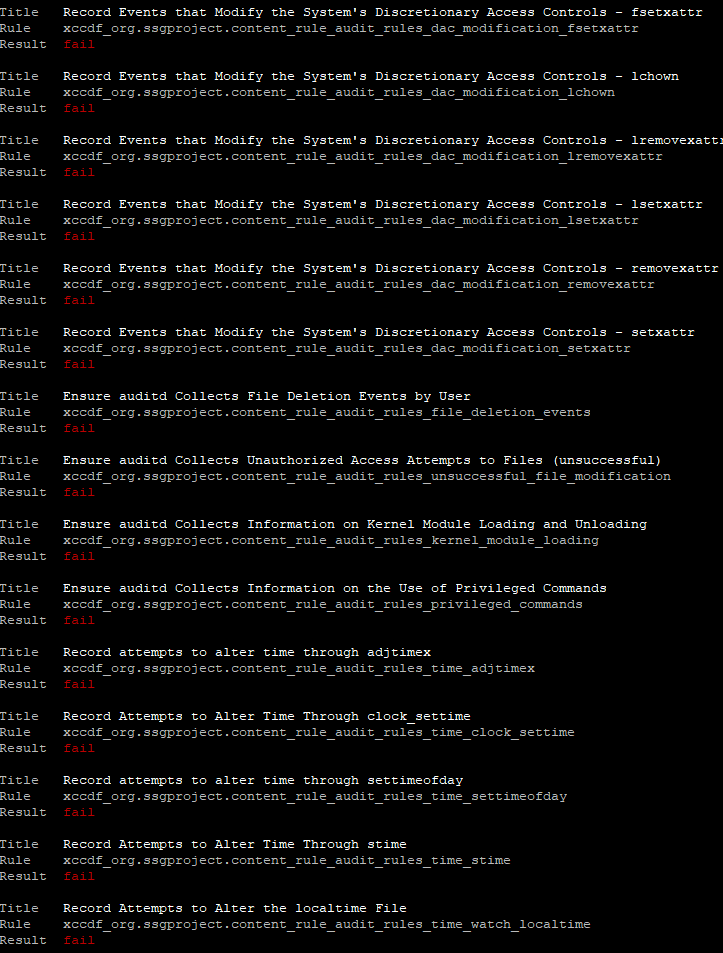
///// Ne pas oublier d'installer le scanner sur la cible :

ssh-keygen -b 4096 -t rsa -f $HOME/.ssh/id\_rsa -q -N ""

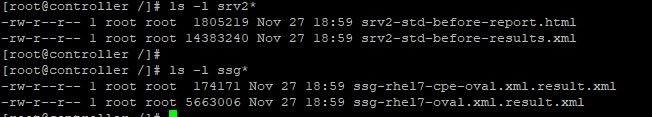
target="srv1"

ssh-copy-id $target

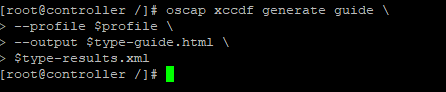
ssh $target "yum -y install scap-security-guide"

Voici les fichier générés :



**1.3.10 Tâche 8. Générer un guide de configuration**



oscap xccdf generate guide \

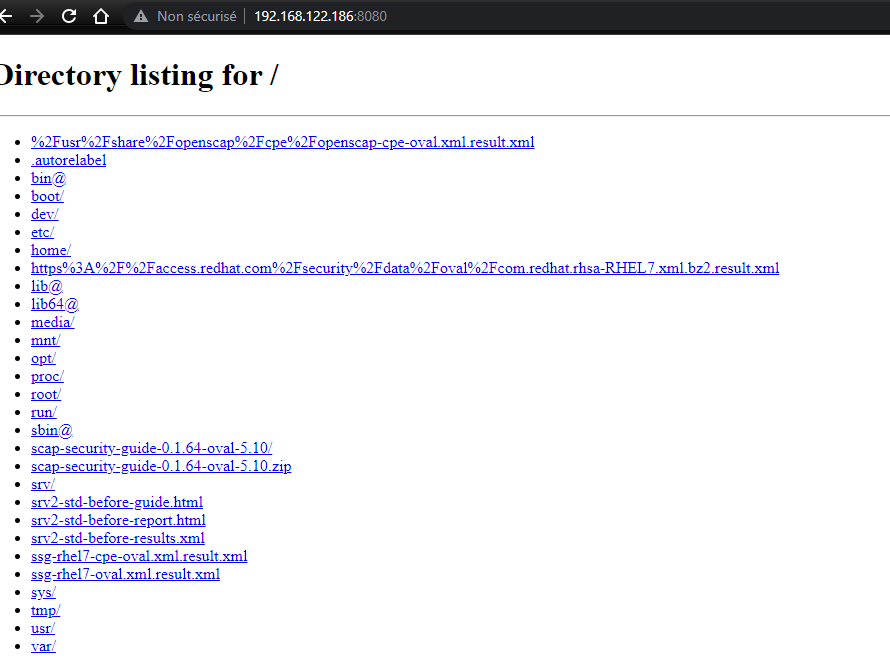
--profile $profile \

--output $type-guide.html \

$type-results.xml

**1.3.11 Tâche 9. Consulter le rapport et le guide de configuration :**

On se connecte sur le contrôleur est on check le rapport « srv2-std-before-report.html » :



**Résultat du test :**   
  
**(Regardez les rapports avant / après remédiation) en annexe.**

**1.3.12. Préparer une remédiation :**OpenSCAP permet de remédier automatiquement aux systèmes qui ont été trouvés dans un état non conforme.

Pour la remédiation du système, les règles dans le contenu SCAP doivent avoir un script de remédiation attaché.

Il existe trois modes de fonctionnement d’oscap en ce qui concerne la remédiation :

1. online : en invoquant directement **oscap xccdf eval --remediate** ;

2. offline : avec **oscap xccdf remediate --results** sur les résultats d’un scan ;

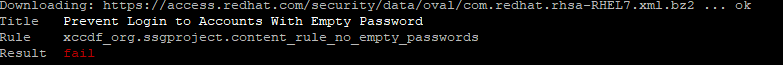
3. et review : en générant un ou plusieurs fixs avec oscap xccdf generate fix sur base des résultats d’un scan.

Il sera probablement indiqué de redémarrer le SRV2 après l’application des correctifs.

Nous allons tenter de remédier à un contrôle “Prevent Login to Accounts With Empty Password” en échec dans cet audit :



Résultat :



**Commandes Firewall :**

firewall-cmd --add-port=8080/tcp

firewall-cmd --reload

Commande pour allumer un serveur HTTP sur le controller (Local)

python3 -m http.server 8080