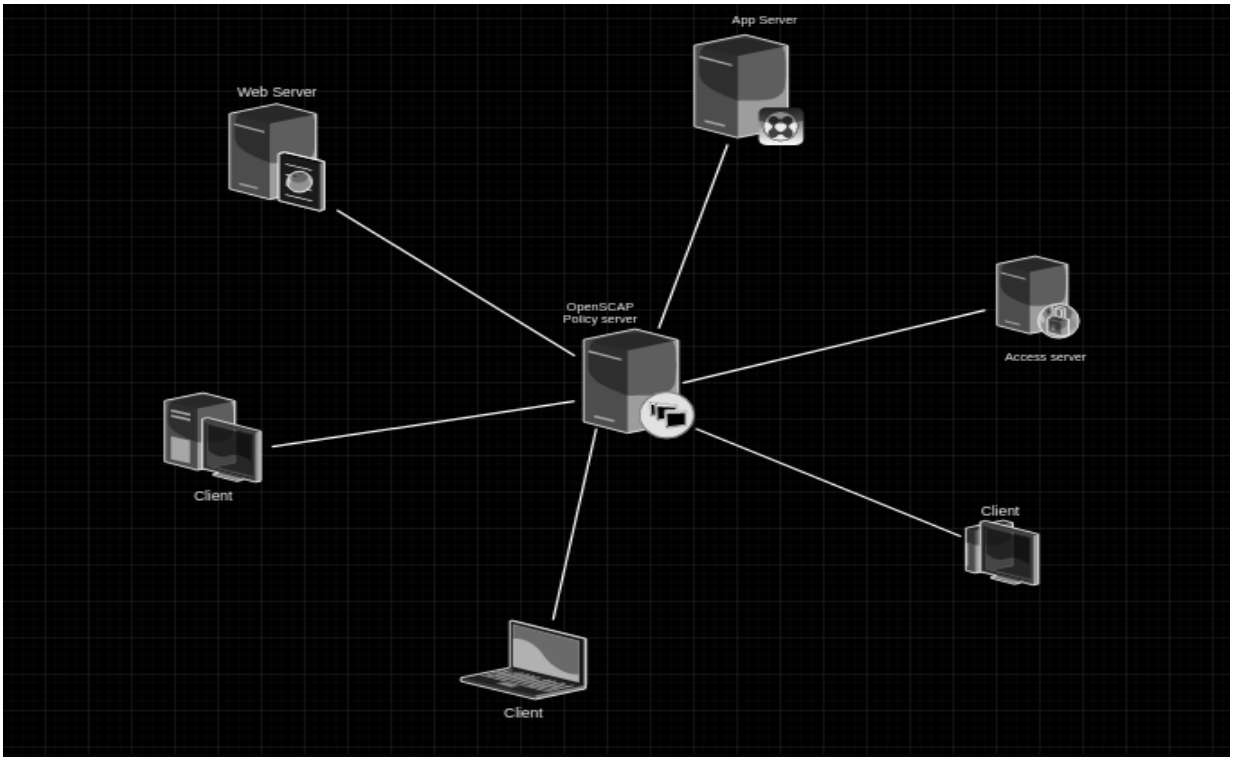
openSCAP
Security Content Automation Protocol



Security Content Automation Protocol (SCAP) is a standard method that provides automated vulnerability management and compliance checks in various hosts. it is maintained by NIST.

SCAP checks Security Compliance by using XCCDF (Extensible content checklist definition format) and OVAL (open vulnerability Automation language) components.

XCCDF: It is an XML file that contains policy structure. it must have a unique name and benchmark ID. it helps to check the vulnerability of hosts.

OVAL: This file contains policy checks. it helps to check whether the host has up-to-date patches or not.these are two components that we would use in our audit checks with openscap in further demo part.

More :

SCAP Components

- ***Languages*** — This group consists of SCAP languages that define standard vocabularies and conventions for expressing compliance policy.

    - The **eXtensible Configuration Checklist Description Format (XCCDF)** — A language designed to express, organize, and manage security guidance.
    - **Open Vulnerability and Assessment Language (OVAL)** — A language developed to perform logical assertion about the state of the scanned system.
    - **Open Checklist Interactive Language (OCIL)** — A language designed to provide a standard way to query users and interpret user responses to the given questions.
    - **Asset Identification (AI)** — A language developed to provide a data model, methods, and guidance for identifying security assets.
    - **Asset Reporting Format (ARF)** — A language designed to express the transport format of information about collected security assets and the relationship between assets and security reports.

- **Enumerations** — This group includes SCAP standards that define naming format and an official list or dictionary of items from certain security-related areas of interest.
    - **Common Configuration Enumeration (CCE)** — An enumeration of security-relevant configuration elements for applications and operating systems.
    - **Common Platform Enumeration (CPE)** — A structured naming scheme used to identify information technology (IT) systems, platforms, and software packages.
    - **Common Vulnerabilities and Exposures (CVE)** — A reference method to a collection of publicly known software vulnerabilities and exposures.
- **Metrics** — This group comprises of frameworks to identify and evaluate security risks.
    - **Common Configuration Scoring System (CCSS)** — A metric system to evaluate security-relevant configuration elements and assign them scores in order to help users to prioritize appropriate response steps.

    - **Common Vulnerability Scoring System (CVSS)** — A metric system to evaluate software vulnerabilities and assign them scores in order to help users prioritize their security risks.

- **Integrity** — An SCAP specification to maintain integrity of SCAP content and scan results.
    - ***Trust Model for Security Automation Data (TMSAD)*** — A set of recommendations explaining usage of existing specification to

represent signatures, hashes, key information, and identity information in context of an XML file within a security automation domain.

Each of the SCAP components has its own XML-based document format and its XML name space. A compliance policy expressed in SCAP can either take a form of a single OVAL definition XML file, data stream file, single zip archive, or a set of separate XML files containing an XCCDF file that represents a policy checklist.

SCAP

Security Content Automation Protocol (SCAP) is a standard method that provides automated vulnerability management and compliance checks in various hosts. it is maintained by NIST.

openscap is the best tool for performing security audits and provides a great way to check systems vulnerability in an easy and automated way.It is a collection of open-source tools for implementing and enforcing security audits that helps system administrators and auditors with assessment, measurement, and enforcement of security baselines.It is a cross-platform open-source tool that provides a wide range of hardening and auditing guidelines for enforcing security.In this demo, we are going to perform security audits to the remote hosts from the policy server using openscap.

Requirements :

- Two Linux having IP's 192.168.1.103, 192.168.1.104 with network connectivity with each other.
- user with sudo rights.
- both servers should have libopenscap8 installed.
- policy server should have oscap-ssh configured.
- policy server should have keyless to other Ubuntu server in order to avoid password prompt.

Now we need to install oscap-ssh which would allow to perform audit on remote hosts.

However, this is a part of openscap project but it's not included in libopenscap8 package. we would need to download it from openscap project repository.

```
wget
https://raw.githubusercontent.com/OpenSCAP/openscap/maint-1.2/utils/oscap-ssh
```

```
sudo su -
chmod 755 oscap-ssh
mv -v oscap-ssh /usr/local/bin
chown root:root /usr/local/bin/oscap-ssh
```

furthermore we need to make keyless by adding the public key to remote host.

take the content of id_rsa.pub from home directory of user of the policy server. then go to the remote host and make authorized_keys file if not exists in home directory of user under .ssh and add whole content of id_rsa.pub of the policy server.

after that give 600 rights to authorized_keys and 700 to .ssh directory, which is placed under home directory of user.

```
chmod 700 /home/gopal/.ssh
chmod 600 /home/gopal/.ssh/authorized_keys
```

Introduction to scap security guide

scap security guide is a security policy that is written in scap document. it covers many areas of system security and provides best practices to check security audits in hosts.

In addition, this guide consists of predefined rules and remediations scripts for target hosts.

scap security guide with openscap can be used together to perform security audits in an automated way.

It implements security guidelines recommended by respected authorities PCI DSS, STIG, and USGCB.

Download this scap security guide from scap project repo and extract it.

```
wget
https://github.com/ComplianceAsCode/content/releases/download/v0.1
.50/scap-security-guide-0.1.50.zip
```

```
unzip scap-security-guide-0.1.50.zip
```

Launching compliance test:

```
oscap-ssh gopal@192.168.1.104 22 xccdf eval --profile
xccdf_org.ssgproject.content_profile_standard --report
~/192.168.1.104.html
~/Downloads/scap-security-guide-0.1.50/ssg-ubuntu1804-ds-1.2.xml
```

oscap-ssh: this is a script that allows us to ssh to the remote host to check the security audit.

xccdf_org.ssgproject.content_profile_standard: this is a xccdf profile id(benchmark id) which is part of xccdf xml file.

And this is the main xccdf file “ssg-ubuntu1804-ds-1.2.xml” which we would use to check the vulnerability of hosts.

however this is in-build security rules which come with scap security guide.

After running the above command, we will get an evaluation report which explicitly displays remote hosts details with pass and fail status of various security rules.

this rule is defined in the XCCDF profile file “ssg-ubuntu1804-ds-1.2.xml” of scap security guide.

Here is a sample openscap evaluation report output.

https://blog.knoldus.com/openscap/

Other Sample :

```
<?xml version="1.0" encoding="UTF-8"?>
<Benchmark xmlns="http://checklists.nist.gov/xccdf/1.1"
id="xccdf_org.opensuse.benchmark">
     <status>draft</status>
     <title>openSUSE Leap Security Benchmark</title>
     <description>This benchmark provides security configuration
guidance for openSUSE Leap systems.</description>
     <version>1.0</version>
     <profile id="xccdf_org.opensuse.profile.default">
     <title>Default Security Profile</title>
     <description>Recommended security configurations for openSUSE
Leap systems.</description>

     <!-- Rule to ensure password complexity -->
     <select idref="xccdf_org.opensuse.rule.password_complexity"
selected="true"/>

     <!-- Rule to ensure SSH root login is disabled -->
     <select
idref="xccdf_org.opensuse.rule.ssh_root_login_disabled"
selected="true"/>

     <!-- Rule to ensure firewall is enabled -->
     <select idref="xccdf_org.opensuse.rule.firewall_enabled"
selected="true"/>

     <!-- Add other rules as needed -->

     </profile>

     <!-- Rule Definitions -->

     <!-- Password Complexity Rule -->
     <Rule id="xccdf_org.opensuse.rule.password_complexity"
selected="true">
     <title>Ensure Password Complexity Requirements</title>
     <description>Password complexity ensures that user passwords
are strong.</description>
```

```
      <check
system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
           <check-content-ref href="opensuse-leap-oval.xml"
name="oval:org.opensuse:def:password_complexity"/>
      </check>
      </Rule>

      <!-- SSH Root Login Disabled Rule -->
      <Rule id="xccdf_org.opensuse.rule.ssh_root_login_disabled"
selected="true">
      <title>Disable SSH Root Login</title>
      <description>Prevent SSH access for the root
user.</description>
      <check
system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
           <check-content-ref href="opensuse-leap-oval.xml"
name="oval:org.opensuse:def:ssh_root_login_disabled"/>
      </check>
      </Rule>

      <!-- Firewall Enabled Rule -->
      <Rule id="xccdf_org.opensuse.rule.firewall_enabled"
selected="true">
      <title>Ensure Firewall is Enabled</title>
      <description>Ensure that the firewall service is running and
enabled on boot.</description>
      <check
system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
           <check-content-ref href="opensuse-leap-oval.xml"
name="oval:org.opensuse:def:firewall_enabled"/>
      </check>
      </Rule>
</Benchmark>
```

Explanation

1. **Benchmark**: The root element that defines the entire security benchmark.
2. **Profile**: The profile section defines which rules to select by default for the given profile.
3. **Rule**: Each `<Rule>` contains a specific security control or policy.
4. **Check**: This section references an OVAL (Open Vulnerability and Assessment Language) definition that contains the actual criteria for checking compliance on the system.

How to Customize and Use

1. Modify the `<Rule>` elements to add more policies specific to your organization.
2. Replace `opensuse-leap-oval.xml` with the path to the actual OVAL definition file for your checks.
3. Save the content to an `.xml` file, such as `opensuse-benchmark.xml`.
4. Run the content through OpenSCAP to evaluate the security policy, using a command such as:

```
oscap xccdf eval --profile default --results results.xml --report
report.html opensuse-benchmark.xml
```

---

ssh with No Password :

To set up SSH passwordless login to a remote host, you can use SSH key authentication. Here's a step-by-step guide:

Step 1: Generate SSH Key Pair (if you don't already have one)

1. Open a terminal on your local machine.
2. Run the following command to generate an SSH key pair:

```
ssh-keygen -t rsa -b 4096
```

Press Enter to accept the default location (`~/.ssh/id_rsa`) and choose whether to set a passphrase. If you want truly passwordless login, leave the passphrase blank.

Step 2: Copy the Public Key to the Remote Host

To copy your public key to the remote host, use the following command:

```
ssh-copy-id username@remote_host
```

Replace `username` with your remote username and `remote_host` with the IP address or hostname of the remote machine.

Alternatively, if `ssh-copy-id` is not available, you can manually copy the key:

1. Display the public key content:

```
cat ~/.ssh/id_rsa.pub
```

2. Copy the output, then SSH into the remote host:

```
ssh username@remote_host
```

3. On the remote host, open the `~/.ssh/authorized_keys` file (create it if it doesn't exist):

```
mkdir -p ~/.ssh
```

```
echo "your_public_key" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Replace `"your_public_key"` with the key you copied from `~/.ssh/id_rsa.pub`.

Step 3: Test Passwordless SSH Login

Log out of the remote host, then try logging back in:

```
ssh username@remote_host
```

If everything was set up correctly, you should be able to log in without being prompted for a password.