COMPLIANCE AUTOMATION WITH OPENSAP

Robin Price II
Senior Solutions Architect, U.S. Public Sector, Red Hat
rprice@redhat.com

Martin Preisler
Senior Software Engineer, Security Technologies, Red Hat
mpreisle@redhat.com
GOALS OF THIS PRESENTATION

1. What exactly is SCAP?
   - Understand the core components
   - Implementations from Red Hat

2. What tools and content are available today and what's in development?
   - For enumerating known vulnerabilities
   - For assessing configurations
   - For single systems, groups of systems, bare metal, virtual or containerized

3. Understand how to install, scan, and remediate using OpenSCAP
LIVE DEMOS DURING THIS PRESENTATION

1. Assess configuration compliance for your RHEL7 nodes
2. Customize a compliance profile with SCAP Workbench, a GUI tailoring tool for SCAP profiles on Linux/OSX/Windows
3. Vulnerability scanning with RHEL using OpenSCAP
4. Deconstruction of each command for complete understanding
SECURITY AUTOMATION

USE CASES

1. **Configuration Management**
   Does your system configuration settings comply with policy?

2. **Vulnerability Management**
   Detect & prioritize known vulnerabilities (software flaws) on a system, determine whether appropriate patches have been applied.

3. **System Inventory**
   Identify products installed on the system (e.g. hardware, operating system, and applications).

4. **Malware Detection** [evolving space]
   Detect presence of malware on a system, allowing zero day signature building for consumption by SCAP tools.
WHAT IS SCAP?
AUTOMATION LANGUAGE
AN SCAP PRIMER

- **Security Content Automation Protocol**
  - Uses standards from all three of the automation families
    - Language, Enumeration, and Risk Measurement

- Collection of Data Formats defined in XML

- Created to provide a standardized approach to maintaining the security of enterprise systems, such as automatically verifying the presence of patches, checking system security configuration settings, and examining systems for signs of compromise.
We needed standardized formats for automated checklists

Because we wanted:
- Standardized inputs (e.g. a compliance baseline, status query)
- Standardized outputs (compliance reports)

Provides the enterprise liberty with regards to product choices
- Avoids vendor lock-in, enables interoperability
- Federal procurement language *requires* SCAP in some cases (e.g. DHS CDM)
THE COMPONENT STANDARDS OF SCAP INCLUDE:

- **Languages:**
  - **XCCDF**: eXtensible Configuration Checklist Description Format
  - **OVAL**: Open Vulnerability Assessment Language
  - **OCIL**: Open Checklist Interactive Language
  - **ARF**: Asset Reporting Format
SECURITY CONTENT AUTOMATION PROTOCOL COMPONENTS

THE COMPONENT STANDARDS OF SCAP INCLUDE:

- Languages (explained):
  - **XCCDF**: Checklists for evaluating a system based on the criteria defined within security and/or nonsecurity checklists.
  
  - **OVAL**: Designed for performing individual security checks, such as verifying security settings, known vulnerabilities, and reporting the results of each check performed.
  
  - **OCIL**: Checks that collection information from people or from existing data stores.
  
  - **ARF**: Framework for documenting informations related to a variety of assets.
THE COMPONENT STANDARDS OF SCAP INCLUDE:

- Enumerations:
  - **CVE**: Common Vulnerabilities and Exposures
  - **CCE**: Common Configuration Enumeration
  - **CPE**: Common Platform Enumeration
THE COMPONENT STANDARDS OF SCAP INCLUDE:

- Enumerations (explained):
  - **CVE**: Enumeration for software vulnerabilities
  - **CCE**: Enumeration of security-relevant configuration elements for applications and operating systems.
  - **CPE**: A structured naming scheme used to identify information technology systems (hardware), platforms (operating systems), and packages (applications).
THE COMPONENT STANDARDS OF SCAP INCLUDE:

-Enumerations (examples):
  - **CVE**: CVE-2014-0160: Heartbleed bug in OpenSSL
  - **CCE**: CCE-3999-0: Make sure SELinux is enforcing
  - **CPE**: cpe:/o:redhat:enterprise_linux:7
SECURITY CONTENT AUTOMATION PROTOCOL

COMPONENTS

THE COMPONENT STANDARDS OF SCAP INCLUDE:

- Risk Measurement:
  - **CVSS**: Common Vulnerability Scoring System
  - **CCSS**: Common Configuration Scoring System
THE COMPONENT STANDARDS OF SCAP INCLUDE:

- Risk Measurement (explained):
  - **CVSS**: Metrics to assign a score to software vulnerabilities to help users prioritize risk.
  - **CCSS**: Metrics to assign a score to security-relevant configuration elements to help users prioritize responses.
SCAP COMPONENT INTERACTION
<table>
<thead>
<tr>
<th>CHECKLIST LANGUAGE</th>
<th>XCCDF</th>
</tr>
</thead>
</table>

SCAP COMPONENT INTERACTION
<table>
<thead>
<tr>
<th>CHECKLIST LANGUAGE</th>
<th>XCCDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK INSTRUCTIONS</td>
<td>OVAL</td>
</tr>
</tbody>
</table>
## SCAP COMPONENT INTERACTION

<table>
<thead>
<tr>
<th>Checklist Language</th>
<th>XCCDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Instructions</td>
<td>OVAL OCIL</td>
</tr>
<tr>
<td>Enumerations</td>
<td>CCE CPE CVE</td>
</tr>
</tbody>
</table>
## SCAP COMPONENT INTERACTION

<table>
<thead>
<tr>
<th>CHECKLIST LANGUAGE</th>
<th>XCCDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK INSTRUCTIONS</td>
<td>OVAL</td>
</tr>
<tr>
<td>ENUMERATIONS</td>
<td>CCE</td>
</tr>
<tr>
<td>RISK MEASUREMENT</td>
<td>CVSS</td>
</tr>
</tbody>
</table>
## SCAP COMPONENT INTERACTION

<table>
<thead>
<tr>
<th>Category</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checklist Language</td>
<td>XCCDF</td>
</tr>
<tr>
<td>Check Instructions</td>
<td>OVAL, OCIL</td>
</tr>
<tr>
<td>Enumerations</td>
<td>CCE, CPE, CVE</td>
</tr>
<tr>
<td>Risk Measurement</td>
<td>CVSS</td>
</tr>
<tr>
<td>Report &amp; Results</td>
<td>ARF</td>
</tr>
</tbody>
</table>
WHAT IS OPENS CAP?
A framework of libraries and tools to improve the accessibility of SCAP and enhance the usability of the information it represents.

The main goal is to perform configuration and vulnerability scans of a local system by evaluating both XCCDF benchmarks and OVAL definitions and generate the appropriate results.
SECURITY AUTOMATION

COMPONENTS

THE COMPONENT STANDARDS OF OPENS Kap INCLUDE:

- Library:
  - libopenscap provides API to SCAP document processing and evaluation.

- Toolkit:
  - SCAP scanner (oscap) is a command line tool that provides various capabilities:
    - configuration scanner
    - vulnerability scanner
    - SCAP content validation and remediation.
RED HAT SCAP TOOLS

OPENSCAP/SCAP SECURITY GUIDE

**OpenSCAP**: suite of open source tools and libraries for security automation

**OpenSCAP Scanner**: CLI tool for configuration and vulnerability measurements

**SCAP Workbench**: GUI front-end for OpenSCAP with remote scanning and policy modification (tailoring).

**SCAP Security Guide**: Provides pre-built profiles for common configuration requirements, such as DoD STIG, PCI-DSS, CJIS, and the Red Hat Certified Cloud Provider standards.

**SCAP Security Guide Docs**: HTML formatted documents containing security guides generated from XCCDF benchmarks.
SHIPPING PROFILES

SCAP-SECURITY-GUIDE

RHEL 7.2 (aka, today via SCAP Security Guide v0.1.25)

- PCI-DSS
- RHEL7 Vendor STIG

RHEL 7.3 (est. SCAP Security Guide v0.1.30, upstream released now)

- Department of Justice Criminal Justice Information Systems (FBI CJIS)
- CIA's C2S ("inspired from CIS RHEL7")
- Certified Cloud Provider (CCP)
- FISMA Moderate (NIST 800-53 Medium/Medium/Medium)

Upstream / In Progress

- DoD Baseline for Workstations (aka, GNOME3)
- Need customer input for prioritization of OpenShift, OpenStack, JBoss...
OSCAP Anaconda: An add-on for the Anaconda installer that enables administrators to feed security policy into the installation process and ensure that systems are compliant from first boot.

Red Hat Satellite: An on-premise (connected or disconnected) systems life-cycle management tool. Can be an alternative to downloading all of your content from the Red Hat content delivery network and limit the risks of malicious content or access.

Red Hat CloudForms: Manage private clouds, virtual environments, and public cloud security through the full life cycle of systems and apps. This allows other Red Hat products like Red Hat OpenShift Enterprise to scan images(containers) for vulnerabilities and policy compliance.
OPENS CAP
HTTPS://WWW.OPENS CAP.COM
HTTPS://GITHUB.COM/OPENS CAP

&

SCAP SECURITY GUIDE
HTTPS://GITHUB.COM/OPENS CAP/SCAP-SECURITY-GUIDE
DEMONSTRATION

Following slides are supplementals to the live demos.

These should enable you to replicate everything from the live demo.

Send an e-mail if something seems wrong or forgotten.

Contact info included at the end of this deck.
HTML REPORT (1/3)

Evaluation Characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target machine</td>
<td>devbox-rhel7</td>
</tr>
<tr>
<td>Benchmark URL</td>
<td>/usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml</td>
</tr>
<tr>
<td>Profile ID</td>
<td>stig-rhel7-server-upstream</td>
</tr>
<tr>
<td>Started at</td>
<td>2016-06-28T10:44:05</td>
</tr>
<tr>
<td>Finished at</td>
<td>2016-06-28T10:44:18</td>
</tr>
<tr>
<td>Performed by</td>
<td>shawnw</td>
</tr>
</tbody>
</table>

CPE Platforms

- cpe:/redhat:enterprise:linux:7
- cpe:/redhat:enterprise:linux:7:client

Addresses

- IPv4: 127.0.0.1
- IPv4: 10.211.55.3
- IPv4: 192.168.122.1
- IPv4: 0.0.0.0:0.0.0:0
- IPv6: fdb2:2c26:44e4:0:21c:42ff:fe84:3983
- IPv6: fe80:0:0:0:21c:42ff:fe84:3983
- MAC: 00:00:00:00:00:00
- MAC: 00:1C:42:84:39:83
- MAC: 52:54:00:D4:6B:CC

Compliance and Scoring

The target system did not satisfy the conditions of 45 rules! Please review rule results and consider applying remediation.

Rule results

- 11 passed
- 45 failed
- 4 other

Severity of failed rules

- 37 low
- 8 medium

Score

<table>
<thead>
<tr>
<th>Scoring system</th>
<th>Score</th>
<th>Maximum</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:scap:scoring:default</td>
<td>47.500000</td>
<td>100.000000</td>
<td>47.5%</td>
</tr>
</tbody>
</table>
## Guide to the Secure Configuration of Red Hat Enterprise Linux 7

<table>
<thead>
<tr>
<th>Section</th>
<th>Fail</th>
<th>Not checked</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Settings</strong></td>
<td>42</td>
<td>4</td>
</tr>
<tr>
<td><strong>Installing and Maintaining Software</strong></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Disk Partitioning</strong></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Ensure /var/log Located On Separate Partition</td>
<td>low</td>
<td>fail</td>
</tr>
<tr>
<td>Ensure /var/log/audit Located On Separate Partition</td>
<td>low</td>
<td>fail</td>
</tr>
<tr>
<td>Encrypt Partitions</td>
<td>low</td>
<td></td>
</tr>
<tr>
<td><strong>Updating Software</strong></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Ensure Red Hat GPG Key Installed</td>
<td>high</td>
<td>pass</td>
</tr>
<tr>
<td>Ensure gpgcheck Enabled In Main Yum Configuration</td>
<td>high</td>
<td>pass</td>
</tr>
</tbody>
</table>
### HTML REPORT (3/3)

#### Set Password Minimum Age

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>accounts_minimum_age_login_defs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>fail</td>
</tr>
<tr>
<td>Time</td>
<td>2016-06-28T10:44:06</td>
</tr>
<tr>
<td>Severity</td>
<td>medium</td>
</tr>
<tr>
<td>Identifiers and References</td>
<td>CCE-27002-5</td>
</tr>
<tr>
<td></td>
<td>IA-5(f), IA-5(1)(d), 198, 75, Test attestation on 20121026 by DS</td>
</tr>
<tr>
<td>Description</td>
<td>To specify password minimum age for new accounts, edit the file <code>/etc/login.defs</code> and add or correct the following line, replacing DAYS appropriately:</td>
</tr>
</tbody>
</table>

```
PASS_MIN_DAYS DAYS
```

A value of 1 day is considered for sufficient for many environments. The DoD requirement is 1.

<table>
<thead>
<tr>
<th>Rationale</th>
<th>Setting the minimum password age protects against users cycling back to a favorite password after satisfying the password reuse requirement.</th>
</tr>
</thead>
</table>

**Remediation script:**

```bash
var_accounts_minimum_age_login_defs="1"
grep -q "PASS_MIN_DAYS /etc/login.defs & & \n   sed -i "s/PASS_MIN_DAYS.*/PASS_MIN_DAYS $var_accounts_minimum_age_login_defs/g" /etc/login.defs
if ! [ $? -eq 0 ]; then
   echo "PASS_MIN_DAYS $var_accounts_minimum_age_login_defs" >> /etc/login.defs
fi
```
INSTALLING OPENSECAP

To install OpenSCAP scanner and the SCAP Security Guide content:
# yum -y install openscap-scanner scap-security-guide

To install SCAP Workbench, the GUI tailoring tool:
# yum -y install scap-workbench

To install documentation (optional):
# yum -y install scap-security-guide-doc
WHAT'S INCLUDED?

Take a look:

```
# rpm -ql scap-security-guide
```

- `/usr/share/xml/scap/ssg/content/`
  Houses SCAP content for automated testing

- `/usr/share/scap-security-guide/kickstart/
  Sample kickstarts using the Anaconda OpenSCAP plugin

- `/usr/share/doc/scap-security-guide-*/
  - HTML tables that map NIST 800-53 back to configuration checks, forming the base of RTMs
  - HTML editions of configuration baselines, e.g. "Privileged User Guides"
BREAKING DOWN SCAP

**XCCDF**: Human-readable prose guidance, expressed in XML
Found @ `/usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml`

**OVAL**: Machine language for pass/fail unit tests
Found @ `/usr/share/xml/scap/ssg/content/ssg-rhel7-oval.xml`

**SCAP Datastream**: Combines XCCDF and OVAL into one file.
Found @ `/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml`
# oscap info /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

Document type: Source Data Stream
Imported: 2015-10-02T06:17:44

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-rhel7-xccdf-1.2.xml
Generated: (null)
Version: 1.2

Checklists:
Ref-Id: scap_org.open-scap_cref_ssg-rhel7-xccdf-1.2.xml
Status: draft
Generated: 2015-10-02
Resolved: true

Profiles:
- xccdf_org.ssgproject.content_profile_standard
- xccdf_org.ssgproject.content_profile_pci-dss
- xccdf_org.ssgproject.content_profile_rht-ccp
- xccdf_org.ssgproject.content_profile_common
- xccdf_org.ssgproject.content_profile_stig-rhel7-server-upstream

Referenced check files:
- ssg-rhel7-oval.xml

...
SHIPPING PROFILES

# oscap info /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

Document type: Source Data Stream
Imported: 2015-10-02T06:17:44

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-rhel7-xccdf-1.2.xml
Generated: (null)
Version: 1.2
Checklists:
Ref-Id: scap_org.open-scap_cref_ssg-rhel7-xccdf-1.2.xml
  Status: draft
  Generated: 2015-10-02
  Resolved: true
Profiles:
  xccdf_org.ssgproject.content_profile_standard
  xccdf_org.ssgproject.content_profile_pci-dss
  xccdf_org.ssgproject.content_profile_rht-ccp  <-- Choose for demo
  xccdf_org.ssgproject.content_profile_common
  xccdf_org.ssgproject.content_profile_stig-rhel7-server-upstream

Referenced check files:
  ssg-rhel7-oval.xml

...
SINGLE-HOST SCAN

```
# oscap xccdf eval \
    --profile xccdf_org.ssgproject.content_profile_rht-ccp \ 
    --results-arf arf.xml --report report.html \ 
    /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml

...  
Title Ensure /var/log/audit Located On Separate Partition  
Rule partition_for_var_log_audit  
Ident CCE-26971-2  
Result fail

Title Encrypt Partitions  
Rule encrypt_partitions  
Ident CCE-27128-8  
Result notchecked

Title Ensure Red Hat GPG Key Installed  
Rule ensure_redhat_gpgkey_installed  
Ident CCE-26957-1  
Result pass

...```
SINGLE-HOST SCAN

IMPORTANT NOTE:

The `ssg-rhel7-ds.xml` file which is the Source DataStream with XCCDF 1.2 built inside. The advantage of Source DataStream is that you have everything you need bundled in one file - XCCDF, OVAL(s), CPE(s), and it supports digital signatures.

The evaluation process usually takes a few minutes, depending on the number of selected rules. Similarly to SCAP Workbench, oscap will also provide you an overview of results after it’s finished, and you will find reports saved and available for review in your current working directory.
SINGLE-HOST SCAN

SCAN DECONSTRUCTION

```bash
# oscap xccdf eval \
  --profile xccdf_org.ssgproject.content_profile_rht-ccp \
  --results-arf arf.xml --report report.html \
/usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

**xccdf eval**

- The **oscap** tool calls on the **xccdf** module.
- The **xccdf** module is used with the **eval** operation which then allows us to perform the evaluation.
- The XCCDF module will try to load all OVAL Definition files referenced from XCCDF automatically.
- **man oscap** for more module operations.

**--profile** PROFILE

- Select a particular profile from the data stream file (INPUT file) at the end of the command.
SINGLE-HOST SCAN

SCAN DECONSTRUCTION (CONT.)

```bash
# oscap xccdf eval \
   --profile xccdf_org.ssgproject.content_profile_rht-ccp \
   --results-arf arf.xml --report report.html \
   /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```

**--results-arf FILE**

- Tell oscap that we want the results stored as an Asset Reporting Format (ARF) in a file called `arf.xml`.
- It is recommended to use this option instead of `--results` when dealing with datastreams.

**--report FILE**

- Write HTML report into `report.html`

/`usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml`

- This is the INPUT_FILE needed to perform the evaluation.
- Print result of each rule to standard output, including rule title, rule id and security identifier (CVE, CCE).
REMEDIATION

Or scan & fix everything at once (note the --remediate flag):

```bash
# oscap xccdf eval --remediate --profile \n  xccdf_org.ssgproject.content_profile_rht-ccp \n  --results scan-xccdf-results.xml \n  /usr/share/xml/scap/ssg/content/ssg-rhel7-ds.xml
```
CVE SCAN
VULNERABILITY SCANNER

Download content from Red Hat:

# cd /tmp

Run CVE scan:

# oscar xccdf eval --results-arf results.xml --report report.html com.redhat.rhsa-RHEL7.ds.xml

View report

# firefox report.html

- Only detects vulnerabilities in Red Hat packages
  - Not Supported: EPEL, 3rd party vendor repos, non-RPM packages, CentOS
  - Only detects vulnerabilities fixed in Red Hat Security Advisories (RHSA)
SATELLITE 6
Audit Scanning

File Upload
- Title: SCAP Security Guide for RHEL7
- Scap file: ssg-rhel7-ds.xml

Locations
Organizations

Submit
SATELLITE

Define policies

New Compliance Policy

1. Create policy
2. SCAP Content
3. Schedule
4. Locations
5. Organizations
6. Hostgroups

Name: weekly_ssg_scans
Description:
SATELLITE

Define policies

New Compliance Policy
Define policies
## Satellite

See past reports

### Compliance Reports

<table>
<thead>
<tr>
<th>Host</th>
<th>Reported At</th>
<th>Passed</th>
<th>Failed</th>
<th>Other</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[URL]</td>
<td>about 7 hours ago</td>
<td>108</td>
<td>113</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>108</td>
<td>113</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>14</td>
<td>44</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>14</td>
<td>44</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>108</td>
<td>113</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>14</td>
<td>44</td>
<td>3</td>
<td>Delete</td>
</tr>
<tr>
<td>[URL]</td>
<td>4 days ago</td>
<td>108</td>
<td>113</td>
<td>3</td>
<td>Delete</td>
</tr>
</tbody>
</table>
SATELLITE

Browse & filter in the rule result overview

<table>
<thead>
<tr>
<th>Severity</th>
<th>Message</th>
<th>Resource</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Ensure Red Hat GPG Key Installed</td>
<td>xccdf_org.ssgproject.content...</td>
<td>pass</td>
</tr>
<tr>
<td>Low</td>
<td>Record Events that Modify the System's Discretionary Access Controls - setxattr</td>
<td>xccdf_org.ssgproject.content...</td>
<td>fail</td>
</tr>
<tr>
<td>Low</td>
<td>Ensure auditd Collects System Administrator Actions</td>
<td>xccdf_org.ssgproject.content...</td>
<td>fail</td>
</tr>
<tr>
<td>Low</td>
<td>Ensure auditd Collects Information on the Use of Privileged Commands</td>
<td>xccdf_org.ssgproject.content...</td>
<td>fail</td>
</tr>
<tr>
<td>Low</td>
<td>Record Events that Modify the System's Discretionary Access Controls - chown</td>
<td>xccdf_org.ssgproject.content...</td>
<td>fail</td>
</tr>
</tbody>
</table>
SATELLITE

Browse HTML reports on per-system views
CONTACT INFORMATION

Robin Price II
Senior Solutions Architect, U.S. Public Sector, Red Hat
(e) robin@redhat.com
(w) 919-754-4412
https://people.redhat.com/rprice
THANK YOU!