

明日はじめる OpenSCAP + SCAP Security Guide

レッドハット株式会社
ソリューションアーキテクト 森若和雄
2017.11.28

「セキュリティポリシー遵守してね」 「はい」

- システムが **5 台**
 - 「ポリシーを遵守していることを確認してください」
→ (1 日 1 台チェック、**1 週間後**) 「遵守できてました」
- システムが **50 台**
 - 「ポリシーを遵守していることを確認してください」
→ (**2.5 ヶ月後**) 「遵守できてました」
- システムが **50 台**
 - 「ポリシーを遵守していることを**毎月**確認してください」
→ .oO(絶対無理だから人を**3 人に増やして……**)
- **コンテナや VM が 200 台**
 - 「ポリシーを遵守していることを**毎月**確認してください」
→ .oO(**10 人に増員して……いやさすがに無理だよな……**)
→ 不正？ 虚偽の報告？

Security Content Automation Protocol

- セキュリティポリシーを遵守しているかチェックする作業を自動化したい！（しないと死ぬから）
- そのために作られた規格群を SCAP と呼びます

- 脆弱性の識別 CVE

- 設定の識別 CCE

- プラットフォームの識別 CPE

- 脆弱性の分類と識別 CWE

- 脆弱性の深刻さのスコア付け CVSS

識別子をつけて
区別できる

- チェック手順の記述言語 OVAL

- チェックリスト記述言語 XCCDF

自動実行できる
チェックリスト

- など

SCAP 策定の背景

2002 年 連邦情報セキュリティマネジメント法

- 米国の政府省庁で、**全情報システムにセキュリティ要求事項を反映することが必須に**

結果として……

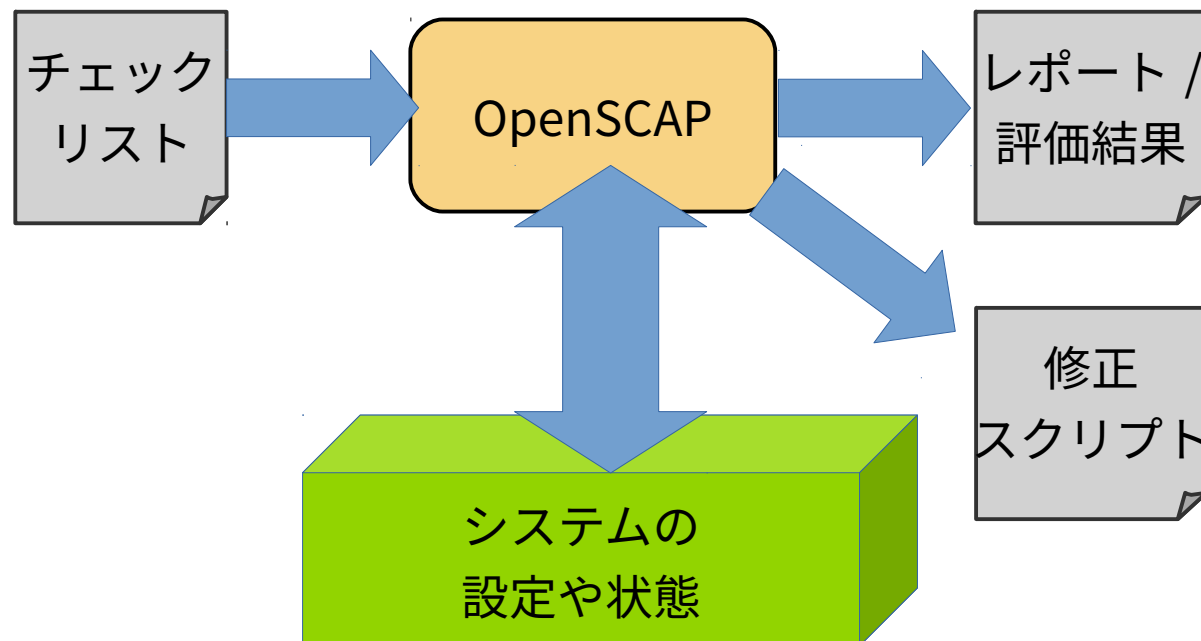
- 現場が疲弊
- ミスや判断の相違によるバラつき
- バラバラのツール群による部分的な自動化
→ 標準化された自動化規格の必要性が高まる

2010 年 NIST が SCAP 1.0 をリリース

OpenSCAP

- SCAP の処理系

- XCCDF で記述されたチェックリストを実行する
- 評価結果を出力,HTML のレポート生成,修正スクリプト生成



OpenSCAP で既知の脆弱性を確認する

- チェックリストのファイルを取得
 - <https://www.redhat.com/security/data/metrics/> から
 - `com.redhat.rhsa-all.xccdf.xml` 「導入 rpm に対応したセキュリティ fix が適用されているか」のチェックリスト (XCCDF 形式)
 - `com.redhat.rhsa-all.xml` 全セキュリティ fix のメタデータと導入有無のチェック方法 (OVAL 形式)
- OpenSCAP で処理
 - `oscap xccdf eval --results results-xccdf.xml --report report-xccdf.html com.redhat.rhsa-all.xccdf.xml`

※ 上記コマンド例はファイルに 2003 年以來の全 RHTSA の情報が含まれているので実行に時間がかかります。RHEL のメジャーバージョン毎や年ごとに分割されたファイルも提供しています。

HTML 形式のレポート

- XCCDF 形式での結果の他に HTML 形式のレポートも生成
- 複数台の結果はまとめられない

Automatically generated XCCDF from OVAL file: com.redhat.rhsa-all.xml

This file has been generated automatically from oval definitions file.

Evaluation Characteristics

Target machine	snake.usersys.redhat.com
Benchmark URL	com.redhat.rhsa-all.xccdf.xml
Started at	2017-11-06T20:44:07
Finished at	2017-11-06T20:44:29
Performed by	kmoriwak

CPE Platforms

Addresses

- IPv4 127.0.0.1
- IPv4 10.64.193.207
- IPv4 192.168.122.1
- IPv6 0:0:0:0:0:0:1
- IPv6 fe80:0:0:0:2da:a206:ae7:4200
- MAC 00:00:00:00:00:00
- MAC D4:3D:7E:DC:C8:AD
- MAC 52:54:00:2D:34:13

Compliance and Scoring

There were no failed or uncertain rules. It seems that no action is necessary.

Rule results

3292 passed

Severity of failed rules

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	100.000000	100.000000	100%

Rule Overview

pass
 fail
 notchecked

fixed
 error
 notapplicable

informational
 unknown

Title	Severity	Result
▶ Automatically generated XCCDF from OVAL file: com.redhat.rhsa-all.xml		

Show all result details

複数台のスキューンハ……？

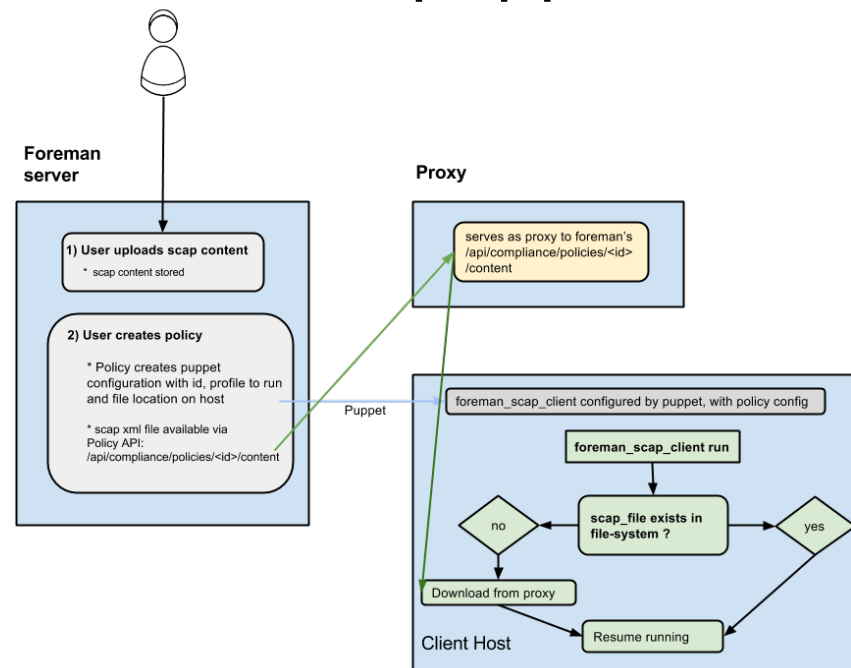
- Spacewalk

<https://spacewalkproject.github.io/>

- Foreman OpenSCAP plugin

https://www.theforeman.org/plugins/foreman_openscap/0.4/

- チェックリスト配布、 puppet での定期実行、レポート表示



脆弱性以外のチェックもやりたい

- 既知のセキュリティ fix があるかどうかだけなら yum security plugin 等でチェックできる
 - OpenSCAP を使わなくていいのでは……？
- それ以外のポリシー遵守のチェックもやりたい
 - /var パーティションは別にする
 - 最小のパスワード長設定
 - 空パスワード禁止
 - ファイルの権限変更を監査ログに記録する
 - ssh での root ログイン禁止
 - suidexec 禁止
 - など

SCAP Security Guide(SSG)

SCAP で記述された Linux システム向けのセキュリティポリシー集

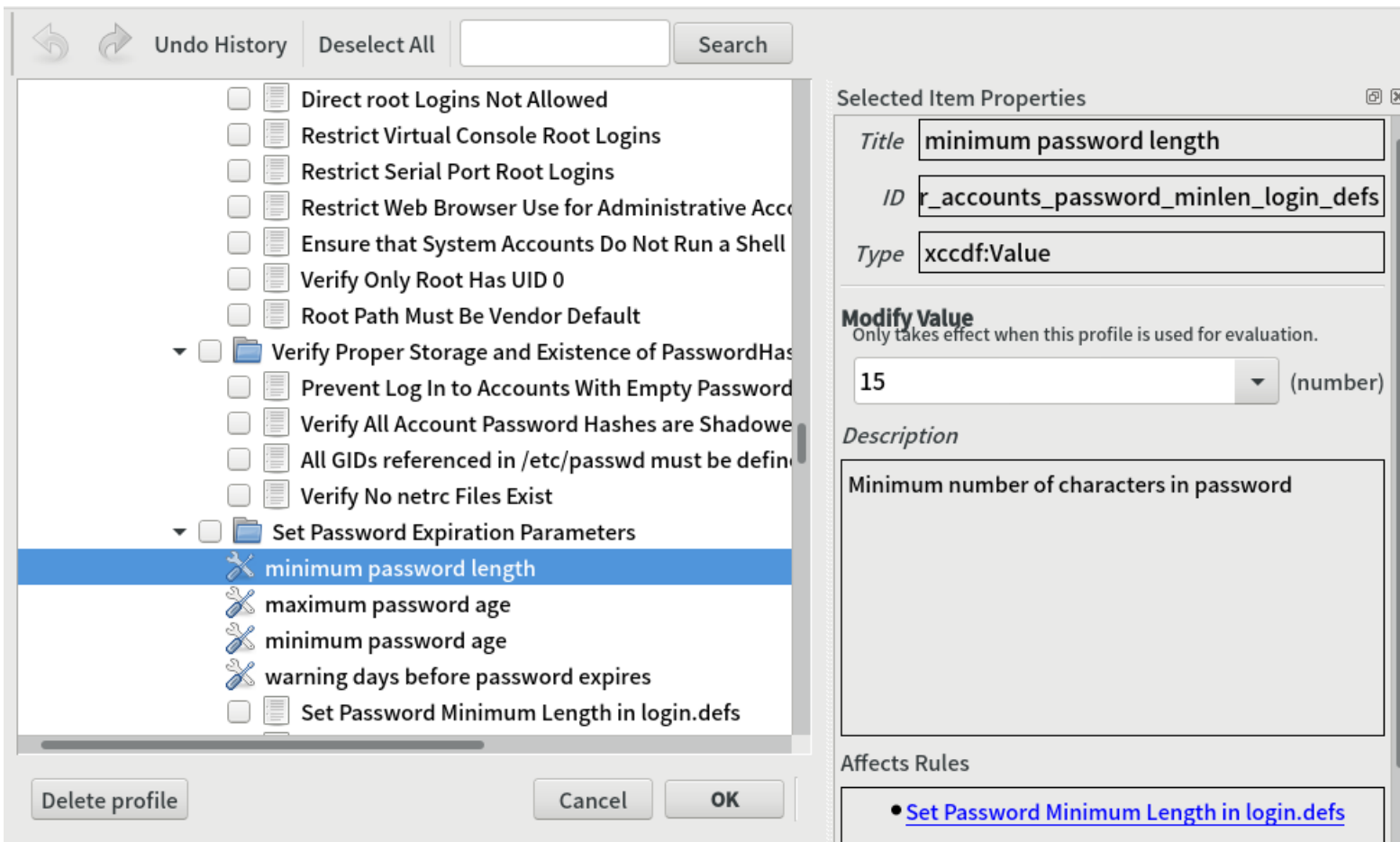
- 主に設定をチェック
 - ソフトウェアの脆弱性は扱わない
- 各種セキュリティ規格用プロファイル同梱
 - DISA STIG, USGCB, PCI-DSS v3, CIS benchmark(に似たもの) etc.
- 対象ソフトウェアは各種ディストリビューション, ブラウザ, JRE など多数

SCAP Security Guide の特徴

- NSA, DISA, NIST と Red Hat を中心とするコミュニティで共同開発
 - SSG 以前
「政府が規格を策定→ベンダが SCAP でチェックリスト作成」 ~3 年
 - SSG 以後
「共同で SCAP でチェックリストを作成」 ~1 年
 - OVAL でのチェック手順について作成ベンダーによる解釈ブレを排除
- 修正スクリプトを生成
 - 一部の問題には、bash または ansible による修正スクリプトを同梱
- カスタマイズして独自プロファイルを作成可
 - チェック項目を取捨選択
 - パスワード長などの項目はパラメータを設定できる

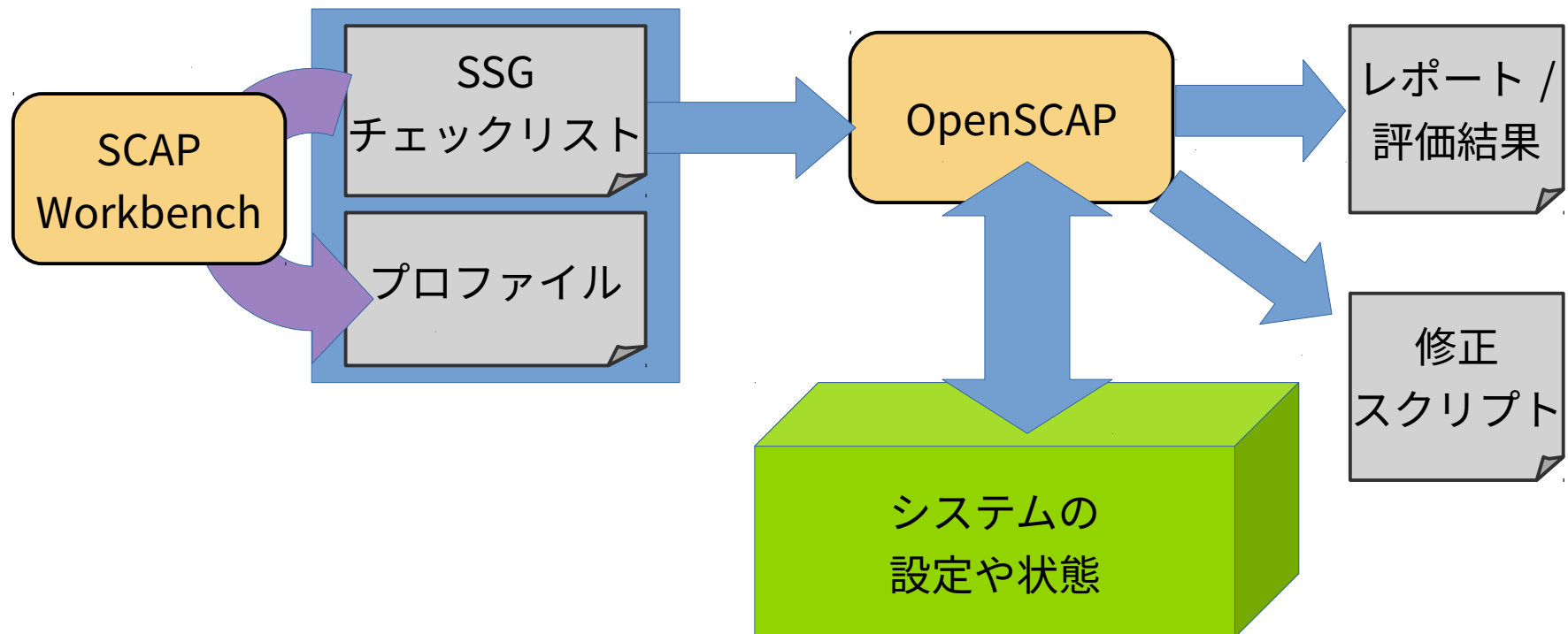
SCAP Workbench

- SSG(XCCDF 文書一般) のプロファイルを作成
 - 既存のチェック項目を GUI で取捨選択、指定



SSG と OpenSCAP でのチェック

- SCAP Workbench で SSG から必要な項目を取捨選択
- 作成したプロファイルを利用して OpenSCAP でチェック



レポート例

チェックをどのように実施したか？

対象システムのバージョン等 (CPE)

各種アドレス

適合状況
不適合ルールの深刻度
重み付きのスコア

各ルール毎の pass/fail

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. The creators of this guidance assume no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Evaluation Characteristics

Target machine	snake.usersys.redhat.com
Benchmark URL	/usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_FEDORA
Profile ID	xccdf_org.ssgproject.content_profile_standard
Started at	2017-11-07T14:54:08
Finished at	2017-11-07T14:59:18
Performed by	kmoriwak

CPE Platforms

- cpe:/o:fedora:project:fedora:28
- cpe:/o:fedora:project:fedora:27
- cpe:/o:fedora:project:fedora:26
- cpe:/o:fedora:project:fedora:25

Addresses

- IPv4 127.0.0.1
- IPv4 10.64.193.207
- IPv4 192.168.122.1
- IPv4 0.0.0.0:0:0:1
- IPv6 fe80::0:0:0:2da:a206:ae7:4200
- MAC 00:00:00:00:00:00
- MAC D4:3D:7E:DC:C8:AD
- MAC 52:54:00:2D:34:13

Compliance and Scoring

The target system did not satisfy the conditions of 4 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	66.666664	100.000000	66.67%

Rule Overview

pass fail notchecked Search through XCCDF rules Search
 fixed error notapplicable
 informational unknown Group rules by: Default

Title	Severity	Result
▼ Guide to the Secure Configuration of Fedora (4x fail)		
▼ System Settings (4x fail)		
▼ Installing and Maintaining Software (2x fail)		
▶ Updating Software		
▼ Software Integrity Checking (2x fail)		
▼ Verify Integrity with RPM (2x fail)		
Verify and Correct File Permissions with RPM	low	fail
Verify File Hashes with RPM	low	fail
▼ File Permissions and Masks (2x fail)		
▼ Verify Permissions on Important Files and Directories (2x fail)		
▼ Verify File Permissions Within Some Important Directories (2x fail)		

レポート例

- 各ルールをクリックすると詳細を表示
 - どのようなチェックか
 - 関係する規格
 - 失敗 or 成功した理由
 - 対策用のスクリプト

Verify File Hashes with RPM

Rule ID	xccdf_org.ssgproject.content_rule_rpm_verify_hashes
Result	fail
Time	2017-11-07T14:58:38
Severity	low
Identifiers and References	References: CM-6(d) , CM-6(3) , SI-7 , 1496
Description	<p>The RPM package management system can check the hashes of installed software packages, including many that are important to system security. Run the following command to list which files on the system have hashes that differ from what is expected by the RPM database:</p> <pre># rpm -Va grep '^..5'</pre> <p>A "c" in the second column indicates that a file is a configuration file, which may appropriately be expected to change. If the file was not expected to change, investigate the cause of the change using audit logs or other means. The package can then be reinstalled to restore the file. Run the following command to determine which package owns the file:</p> <pre># rpm -qf FILENAME</pre> <p>The package can be reinstalled from a dnf repository using the command:</p> <pre>dnf reinstall PACKAGENAME</pre> <p>Alternatively, the package can be reinstalled from trusted media using the command:</p> <pre>rpm -Uvh PACKAGENAME</pre>
Rationale	The hashes of important files like system executables should match the information given by the RPM database. Executables with erroneous hashes could be a sign of nefarious activity on the system.

OVAL details

[verify file md5 hashes](#) **failed** because of these items:

Name	Epoch	Version	Release	Arch	Filepath	Extended name	Size differs	Mode differs	Md5 differs	Dev differs
texlive-kpathsea	6	svn41139	33.fc26.2	noarch	/usr/share/texlive/texmf-dist/web2c/fmtutil.cnf	texlive-kpathsea-6:svn41139-33.fc26.2.noarch	fail	fail	fail	pas:
gnome-themes	(none)	2.32.0	15.fc26	noarch	/usr/share/icons/Crux/icon-theme.cache	gnome-themes-0:2.32.0-15.fc26.noarch	fail	pass	fail	pas:
gnome-themes	(none)	2.32.0	15.fc26	noarch	/usr/share/icons/Mist/icon-theme.cache	gnome-themes-0:2.32.0-15.fc26.noarch	fail	pass	fail	pas:

Remediation Ansible snippet: [\(show\)](#)

Complexity: high

Disruption: medium

```
- name: "Set fact: Package manager reinstall command (dnf)"
  set_fact:
    package_manager_reinstall_cmd: dnf reinstall -y
  when: ansible_distribution == "Fedora"
  tags:
    - rpm_verify_hashes
    - low_severity
```

SSG がない独自のチェックは？

- OVAL でチェック方法を記述するのはしんどい
 - OVAL は XML でチェックを記述する文法
 - 「あるサービスが起動しているか？」だけで 55 行……
 - やりたいことが既存の OVAL リポジトリにないかを探す
 - OVAL のリポジトリ <https://oval.cisecurity.org/repository>
 - 独自のチェック追加は Ansible 等で行う方が生産性が高そう
- XCCDF の基本は「名前」「説明」「OVAL を呼ぶ」くらいなので XCCDF だけ勉強するのはアリ
- 専用エディタを使う
 - VMware Modified Enhanced SCAP Content Editor
 - <https://github.com/vmware/vmware-scap-edit>

まとめ

- SCAP は基本的なセキュリティチェックを自動的に実行することを目的とした規格群です
- OpenSCAP は OSS の SCAP 処理系です
- SCAP Security Guide は SCAP 規格にもとづくチェックリストのひな型を提供しています
- SCAP Workbench はチェックリストのカスタマイズを行いプロファイルを生成します
- 「SSG から必要なところだけ SCAP Workbench で選択して OpenSCAP でチェックする」ことで基本的な確認作業の多くを自動化できます

Appendix

oscap コマンド例

パッケージ導入

```
# yum install openscap scap-security-guide
```

チェックリストの諸元確認

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

“common” プロファイルでのチェック実行

```
# oscap xccdf eval --profile common \
```

```
--results /tmp/results.xml \
```

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

remediate スクリプトの生成

```
# oscap xccdf generate fix --fetch-remote-resources \
```

```
--profile common --output /tmp/remediate.sh \
```

```
/tmp/results.xml
```

読むといいもの

- IPA の SCAP 概説
 - <https://www.ipa.go.jp/security/vuln/SCAP.html>
- RHEL ドキュメント 「 Security Guide 」
 - <http://red.ht/2yb6Zft> (英語)
 - <http://red.ht/2zps2yg> (和訳)
- OpenSCAP プロジェクト
 - <https://www.open-scap.org/>
 - <https://github.org/OpenSCAP/>
 - 今回紹介以外のツール、詳しいドキュメント、各種チュートリアル
- OVAL のリポジトリ <https://oval.cisecurity.org/repository>
- OpenSCAP Scanning in Satellite 6 and CloudForms
 - <http://red.ht/2AwBIFk>
 - Red Hat の管理ツールとの連携について紹介

RHEL での利用

- oscap-anaconda-plugin
 - RHEL や Fedora のインストーラ用プラグイン
 - インストール時にスキャンを実施し、remediation script を実行する
 - 規格に対応した環境のデプロイ作業時間を短縮
- Red Hat Satellite
 - Foreman をベースとする管理スイート
 - OpenSCAP の定期実行、結果蓄積と表示
- 同梱 SSG の対象ソフトウェア
 - RHEL6, RHEL7, JRE, Firefox