

SCAP - UN ENSEMBLE DE STANDARDS POUR LA GESTION DE LA SÉCURITÉ



OpenSCAP

PHILIPPE THIERRY



Vendredi 2 janvier 2017



Introduction

- Ce cours décrit l'état de l'art des outils de maintien en condition de sécurité ainsi que les principes généraux et le workflow des standard SCAP.
- Dans le cadre de ce cours, les outils restent orientés pour les cibles GNU/Linux dans l'optique des travaux pratiques mais le sujet du maintien en condition de sécurité est global à l'ensemble d'une infrastructure (appliances propriétaires, postes bureautiques, etc.). Les principes derrière SCAP sont conçus pour supporter l'ensemble des équipements logiciels ou appliances du marché.

Sommaire

1 MCS

2 SCAP

3 Utiliser SCAP

4 Dans les arcanes du standard

5 Annexes

Sommaire

1 MCS

- Les principes de base
- Quelques exemples d'outils
- Les limites de cet état de l'art
- La genèse de SCAP

2 SCAP

3 Utiliser SCAP

4 Dans les arcanes du standard

5 Annexes

Maintenir ses équipements à jour

Suivi des mises à jour et des correctifs

- Principe de base de tout système d'information
- Pas toujours aisé à mettre en application
 - Tests de non-regression des patchs fonctionnels et de sécurité
 - Accès aux dépôts de correctifs pas toujours direct
 - Parfois complexe dans un système d'information hétérogène
 - Imprimantes, appliances, postes bureautiques, serveurs...
- Est souvent dénoté sous l'intitulé "patch management"

Appliquer les bonnes pratiques de configuration

- Historiquement pas toujours présentes, aujourd'hui de plus en plus souvent fournies par les fournisseurs de logiciels
- Peuvent nécessiter une refonte de l'usage des logiciels (utilisation laxiste du compte administrateur, accès direct aux API internes sans filtrage amont, etc.)
- Initialement peu lisibles dans les guides de sécurité car de trop haut niveau (e.g. NIST SP-800-53), les bonnes pratiques de configurations sont apparues au fur et à mesure du temps avec les STIGs ou sous forme de CCE (Common Configuration Enumeration) et dans les divers autres guides techniques (e.g. ANSSI DAT-NT-28, ANSSI DAT-NT-007 pour GNU/Linux)

Définir une politique de sécurité d'un système d'information

- Cette politique, nommée PSSI, est sous la responsabilité du DSSI de l'entreprise, et rédigée en comité avec des experts en sécurité des systèmes d'information
- Cette politique n'est valable que si elle est applicable et contrôlable dans le temps
- Une PSSI peut contenir des éléments techniques (durcissement des configurations par exemple) mais contient également souvent des éléments organisationnels (gestion des accès aux salles machines, au bâtiments, gestion des horaires d'accès au(x) site(s), etc.
- Une PSSI dépend des moyens que la société accepte de mettre dans la sécurité de son système d'information (au sens large), mais aussi de la réglementation en vigueur. En France, il existe plusieurs réglementation imposant un durcissement minimum à certains acteurs (typiquement les OIV), comme la PSSIE [p. 45] ou la PPST [p. 45]

Recherche de vulnérabilité

debsecan

- Outil de recherche de vulnérabilité sur un équipement donné
- Spécifique à la distribution Debian. A été porté sous Ubuntu mais sans supporter ces dernières comme cibles
- S'appuie sur les DSA (Debian Security Announces) de security.debian.org pour remonter les vulnérabilités présentes dans l'équipement
- Assure une traçabilité aux CVEs

```
# debsecan --only-fixed
```

```
[...]
```

```
CVE-2016-3710 qemu-system-common (fixed, high urgency)
```

```
CVE-2016-3712 qemu-system-common (fixed, low urgency)
```

```
CVE-2016-7966 libakonadi-kmime4 (fixed, remotely exploitable, high urgency)
```

```
CVE-2016-6232 libkmediaplayer4 (fixed, remotely exploitable, medium urgency)
```

```
CVE-2017-6410 libkmediaplayer4 (fixed, remotely exploitable, medium urgency)
```

```
CVE-2017-8422 libkmediaplayer4 (fixed, high urgency)
```

```
[...]
```


Bastille

- Bastille est un vieil outil de durcissement de configuration, également capable de faire de l'audit et du reporting
- Basé sur une interface interactive pour le durcissement des configurations, il permet à un administrateur peu habitué à la sécurisation d'un OS UNIX ou GNU/Linux d'appliquer les bonnes pratiques de base
- Bien que toujours supporté, Bastille est aujourd'hui en perte de vitesse face à des outils comme Lynis ou OpenSCAP pour l'audit et la remédiation

Quelques exemples d'outils

lynis

- Lynis est un outil en ligne de commandes bien abouti, permettant d'étudier l'état des configuration et des versions des logiciels du système d'exploitation afin de les confronter à une politique de sécurité donnée, pouvant être personnalisée
- Le rapport de Lynis est de bonne facture et donne un bon aperçu du niveau de durcissement de l'OS et des manquements présents
- Lynis supporte un grand nombre de tests, ce qui en fait un outil riche et efficace, avec une synthèse très convenable

Quelques exemples d'outils

```
$ sudo lynis audit system
```

```
[...]
```

```
[+] Security frameworks
```

```
-----
- Checking presence AppArmor                                [ FOUND ]
  apparmor filesystem is not mounted.
  - Checking AppArmor status                                [ UNKNOWN ]
- Checking presence SELinux                                  [ FOUND ]
  - Checking SELinux status                                [ DISABLED ]
- Checking presence grsecurity                               [ NOT FOUND ]
- Checking for implemented MAC frame                         [ NONE ]
[...]
```

```
Lynis security scan details:
```

```
Hardening index : 51 [##### ]
```

```
Tests performed : 207
```

```
Plugins enabled : 1
```

Un écosystème désordonné

- Bien que les outils de patch management et de suivi de la sécurité d'un équipement existent, l'écosystème de gestion de la sécurité est longtemps resté siloté entre chaque industriel (Microsoft, IBM, etc.), sans véritable définition de standards unifiés permettant de gérer de manière homogène un système d'information
- Les outils de validation des politiques de sécurité ont également longtemps implémenté leurs propres implémentations pour valider les politiques standard, comme le PCI-DSS ou les exigences NIST. Ces implémentations, incluant souvent une validation auprès de l'organisme éditeur de la politique, étant vue (à juste titre) comme une valeur ajoutée du logiciel, imposant de fait aux industriels une fidélité à un éditeur pour simplifier la gestion de la sécurité de leur parc

Un écosystème désordonné

- L'existence de structures verticales multiples (en générale par vendeur) et incompatibles entre elles rend le gestion de la sécurité d'un parc quasiment impossible à valider efficacement, à la fois en terme de coût et en terme de garantie
- L'indépendances des échelles, des outils et des protocoles implique une mise en place difficile du plan de gestion de la sécurité, et un surcoût évident au niveau du SoC (Security Operation Center) du système d'information, écartant de fait ce service des petits acteurs

Comment corriger le tir ?

- Séparer l'implémentation des sondes du formalisme des tests, permettant d'enrichir chaque partie aisément
- Formaliser l'identification des équipements logiciels, matériels
- Formaliser une vulnérabilité et son écosystème (risque, complexité, cible, etc.)
- Définir un ensemble de langages pour définir, sans prendre d'hypothèse sur le logiciel ou matériel cible :
 - une fiche de test
 - une politique de sécurité
 - une mécanique de correction de la faiblesse de sécurité (remédiation)
- Faire en sorte de faire entrer l'ensemble des constructeurs et éditeurs dans le nouvel écosystème

Petit rappel des organismes parties prenantes

- **DoD** - Department of Defense (US)
Ministère de la défense américain. Demandeur de la solution.
- **NSA** - National Security Agency (US) Agence de sécurité américaine garante de l'efficacité sécuritaire de la solution.
- **DoHS** - Department of Homeland Security (US)
Ministère de l'intérieur, co-demandeur.
- **DISA** - Defense Information System Agency (US)
Agence en charge de la sécurité des systèmes d'informations de défense, en charge de la rédaction de politiques de sécurité pour les SI de défense américains.
- **NIST** - National Institute for Standards and Technologies (US)
Organisme normatif américain, en charge de rédiger et de mettre à jour les standards américain dans le domaine des technologies de l'information et de la communication.

Petit rappel des organismes parties prenantes

- **DoD** - Department of Defense (US)
Ministère de la défense américain. Demandeur de la solution.
- **NSA** - National Security Agency (US) Agence de sécurité américaine garante de l'efficacité sécuritaire de la solution.
- **DoHS** - Department of Homeland Security (US)
Ministère de l'intérieur, co-demandeur.
- **DISA** - Defense Information System Agency (US)
Agence en charge de la sécurité des systèmes d'informations de défense, en charge de la rédaction de politiques de sécurité pour les SI de défense américains.
- **NIST** - National Institute for Standards and Technologies (US)
Organisme normatif américain, en charge de rédiger et de mettre à jour les standards américain dans le domaine des technologies de l'information et de la communication.

Petit rappel des organismes parties prenantes

- **DoD** - Department of Defense (US)
Ministère de la défense américain. Demandeur de la solution.
- **NSA** - National Security Agency (US) Agence de sécurité américaine garante de l'efficacité sécuritaire de la solution.
- **DoHS** - Department of Homeland Security (US)
Ministère de l'intérieur, co-demandeur.
- **DISA** - Defense Information System Agency (US)
Agence en charge de la sécurité des systèmes d'informations de défense, en charge de la rédaction de politiques de sécurité pour les SI de défense américains.
- **NIST** - National Institute for Standards and Technologies (US)
Organisme normatif américain, en charge de rédiger et de mettre à jour les standards américain dans le domaine des technologies de l'information et de la communication.

Petit rappel des organismes parties prenantes

- **DoD** - Department of Defense (US)
Ministère de la défense américain. Demandeur de la solution.
- **NSA** - National Security Agency (US) Agence de sécurité américaine garante de l'efficacité sécuritaire de la solution.
- **DoHS** - Department of Homeland Security (US)
Ministère de l'intérieur, co-demandeur.
- **DISA** - Defense Information System Agency (US)
Agence en charge de la sécurité des systèmes d'informations de défense, en charge de la rédaction de politiques de sécurité pour les SI de défense américains.
- **NIST** - National Institute for Standards and Technologies (US)
Organisme normatif américain, en charge de rédiger et de mettre à jour les standards américain dans le domaine des technologies de l'information et de la communication.

Petit rappel des organismes parties prenantes

- **DoD** - Department of Defense (US)
Ministère de la défense américain. Demandeur de la solution.
- **NSA** - National Security Agency (US) Agence de sécurité américaine garante de l'efficacité sécuritaire de la solution.
- **DoHS** - Department of Homeland Security (US)
Ministère de l'intérieur, co-demandeur.
- **DISA** - Defense Information System Agency (US)
Agence en charge de la sécurité des systèmes d'informations de défense, en charge de la rédaction de politiques de sécurité pour les SI de défense américains.
- **NIST** - National Institute for Standards and Technologies (US)
Organisme normatif américain, en charge de rédiger et de mettre à jour les standards américain dans le domaine des technologies de l'information et de la communication.

La genèse de SCAP

- La création de SCAP débute au début des années 2000, sur une demande du DoD pour unifier les mécanismes de gestion de la sécurité des systèmes d'information autour d'un ensemble de standards
- Le but est de casser les technologies en silos de suivi de la sécurité pour mettre en application une solution transverse, applicable à l'ensemble d'un système d'information, allant des appliances réseau aux postes de travail, en passant par les serveurs et les équipements embarqués
- La création du standard est poussée par le DoD, la NSA, le DoHS et le DISA (Agence pour les systèmes d'informations de Défense). le standard est porté par le NIST (National Institute for Standard & Technology)

La genèse de SCAP

- La création de SCAP débute au début des années 2000, sur une demande du DoD pour unifier les mécanismes de gestion de la sécurité des systèmes d'information autour d'un ensemble de standards
- Le NIST définit alors un ensemble de standard :
 - Les *CVE* (Common Vulnerability Exposure), formalisant une vulnérabilité
 - Les *CVSS* (Common Vulnerability Scoring System), associant une pondération à une vulnérabilité
 - Les *CWE* (Common Weakness Enumeration), formalisent des famille de faiblesses par type (défaut de configuration, mauvais nettoyage des entrées logicielles, etc.)
 - Les *CPE* (Common Platform Enumeration), formalisant une identification des équipements
 - *XCCDF* (eXtensible Checklist Configuration Description Format), formalisme XML permettant de définir une checklist de sécurité et de générer des rapports de conformité
 - *OVAL* (Open Vulnerability and Assessment Language), formalisme XML définissant une fiche de test

La genèse de SCAP

- La création de SCAP débute au début des années 2000, sur une demande du DoD pour unifier les mécanismes de gestion de la sécurité des systèmes d'information autour d'un ensemble de standards
- Les premières présentations (publiques!) de SCAP datent de 2008
- La version 1.0 finalisée du standard, publiée par le NIST, date du 27 juillet 2010
- Les formalismes XCCDF, OVAL possèdent leur propre cycle de vie. Oval en est à la version 5.11, XCCDF à la version 1.2

Sommaire

1 MCS

2 SCAP

- Un écosystème de standards
- Un workflow complet pour le MCS
- Etat de l'art des implémentations

3 Utiliser SCAP

4 Dans les arcanes du standard

5 Annexes

Un écosystème de standards

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - **Son impact en terme de confidentialité**
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - **Son impact en terme d'intégrité**
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référénçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - **Son impact en terme de disponibilité**
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référénçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - **Le niveau de gain**
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référénçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - **Le type de vulnérabilité**
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - **L'identifiant de faiblesse associé (CWE)**
 - Un identifiant unique la référçant, sous la forme CVE-<année>-<incrément>

Les CVEs

- Une CVE est une formalisation d'une vulnérabilité.
- Une CVE est composé d'un ensemble de propriétés
 - Son score CVSS (défini plus loin), conséquence des autres propriétés
 - Son impact en terme de confidentialité
 - Son impact en terme d'intégrité
 - Son impact en terme de disponibilité
 - Sa complexité technique de mise en oeuvre
 - Le niveau d'accès nécessaire à sa mise en oeuvre
 - Le niveau de gain
 - Le type de vulnérabilité
 - L'identifiant de faiblesse associé (CWE)
 - Un identifiant unique la référençant, sous la forme CVE-<année>-<incrément>

Un écosystème de standards

Vulnerability Details : [CVE-2016-8437](#)

Improper input validation in Access Control APIs. Access control API may return memory range checking incorrectly. Product: Android. Versions: Kernel 3.18. And

Publish Date : 2017-01-12 Last Update Date : 2017-01-17

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#)

▼ [Scroll To](#)

▼ [Comments](#)

▼ [External Links](#)

[Search Twitter](#) [Search YouTube](#) [Search Google](#)

- CVSS Scores & Vulnerability Types

CVSS Score

10.0

Confidentiality Impact

Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact

Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact

Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity

Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication

Not required (Authentication is not required to exploit the vulnerability.)

Gained Access

None

Vulnerability Type(s)

CWE ID

[20](#)

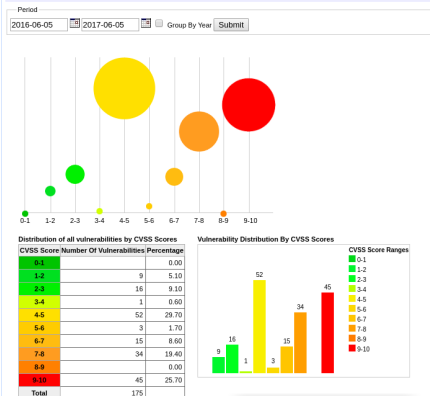
CVE-2016-8437 (<http://www.cvedetails.com/cve/CVE-2016-8437/>)

Le CVSS

- Permet de déterminer le niveau de risque associé à une CVE
- est basé sur une echelle de 1 (très faible) à 10 (très important)
- Le calcul du CVSS est basé sur :
 - l'impact de l'exploitation de la vulnérabilité en terme de confidentialité, d'intégrité et de disponibilité de la cible
 - La complexité technique à l'exploitation de la faille (niveau technique nécessaire pour l'attaquant)
 - Les hypothèses nécessaires à l'attaque (accès local, distant, nombre de séquences d'authentification préalables sur la cible)
- On associe souvent à l'échelle de CVSS l'échelle risque statistique associé à la vulnérabilité. Elle permet de plus de fournir un tableau synthétique du risque moyen par vendeur et/ou composant basé sur l'historique des CVE associées à ce dernier

Un écosystème de standards

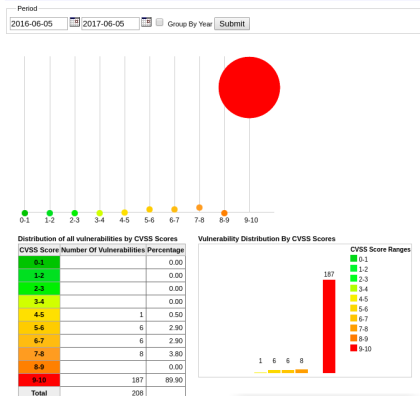
CVSS Scores For [Microsoft Windows 7](#) Between 2016-06-05 and 2017-06-05



Synthèse des impacts des vulnérabilités de Ms. Windows 7
 (http://www.cvedetails.com/cvss-score-charts.php?product_id=17153&fromform=1)

Un écosystème de standards

CVSS Scores For [Adobe Flash Player](#) Between 2016-06-05 and 2017-06-05



Synthèse des impacts des vulnérabilités d'Adobe Flash Player
 (https://www.cvedetails.com/cvss-score-charts.php?product_id=6761&fromform=1)

Les CWE

- Le but des CWE est de définir des familles de faiblesses (à différencier d'une vulnérabilité, une faiblesse n'impliquant pas nécessairement une vulnérabilité), pouvant s'appliquer à tout type de cible logicielle ou matérielle.
- L'usage des CWE permet de fournir un indicateur de qualimétrie de sécurité de la cible selon l'état des CWE associé et selon les faiblesses détectées
- Les CWE sont organisées sous forme d'une structure hiérarchique de faiblesses
- Il existe des centaines de CWE, classées et numérotées, comme par e.g. :
 - CWE 20 : Improper Input Validation. Typiquement une validation incorrecte d'un flux d'entrée non maîtrisée.
 - CWE 494 : Download of code without integrity check. Exemple de faiblesse liée à la mise à jour logicielle non sécurisée d'une appliance
- A chaque identifiant de CWE est indiqué l'impact probable d'une CVE basée sur celle-ci, l'effet probable en terme d'accessibilité, confidentialité et disponibilité, et les manières de s'en prémunir

Les CPE

- Qui dit formalisation généralisée à l'ensemble des équipements et vendeurs dit définition d'un formalisme pour identifier de manière unique une plateforme. C'est le rôle du dictionnaire CPE
- le format CPE (v2.3) définit une structure de donnée hiérarchique, séparée par des ' : '. Voici sa osyntaxe :
 - cpe : chaine fixe
 - cpe-dict-release : numéro de version CPE, aujourd'hui 2.3
 - part : type d'équipement ((a)pplication, (o)s, ou (h)ardware)
 - vendor : non du vendeur
 - productname : nom du produit
 - product-version : version du produit
 - update-version : numéro de mise à jour de la version (si existant)
 - edition-version : édition interne à la numérotation de mise à jour (si existant)
 - language-name : langue cible (si existant)
- Il est possible de rédiger ses propres fiches CPE, au format OVAL, sans déclarer son produit auprès du MITRE.

Les CPE

- Qui dit formalisation généralisée à l'ensemble des équipements et vendeurs dit définition d'un formalisme pour identifier de manière unique une plateforme. C'est le rôle du dictionnaire CPE
- le format CPE (v2.3) définit une structure de donnée hiérarchique, séparée par des ' : '. Voici sa osyntaxe :
 - Pour Microsoft I.E, on retrouve la structure suivante :
`cpe :2.3 :a :microsoft :internet_explorer :8.0.6001 :beta :* :* :* :* :* *`
 - Pour un système d'exploitation (part passe de 'a' à 'o', on aura, par exemple pour Debian :
`cpe :2.3 :o :debian :debian_linux :8.0 :* :* :* :* :* *`
 - Pour un équipement matériel, hiérarchisé dans la partie 'hardware', on aura :
`cpe :2.3 :h :cisco :12000_router :- :* :* :* :* :* *`
- Il est possible de rédiger ses propres fiches CPE, au format OVAL, sans déclarer son produit auprès du MITRE.

Le formalisme OVAL

Une fiche de test

- Un langage verbeux, au format XML
- Utilisant correctement l'écosystème XML (XMLSchema, XSD, etc.)
- Définissant la finalité du test, pas le moyen (laissé à l'implémentation de la sonde)

Le formalisme XCCDF

Langage de génération des STIGs et des checklists

- Permet de rédiger des checklists (typiquement pour auditer un équipement par rapport à une politique de sécurité)
- Génère dans le même temps le guide de sécurité associé
- est un formalisme XML
- Associe un ensemble de fiches OVAL à une politique de sécurité, assurant ainsi une traçabilité des tests

XCCDF est riche et complexe

- Il possède des mécanismes de contrôle intégrés (macros if)
- Permet d'intégrer des mécanismes d'interactivité via le formalisme OCIL (Open Checklist Interactive Language)
- Il supporte la gestion de variables pouvant être raffinées selon les profils associé au guide de sécurité

Petite synthèse

Standards for Enabling Automation in Information Security

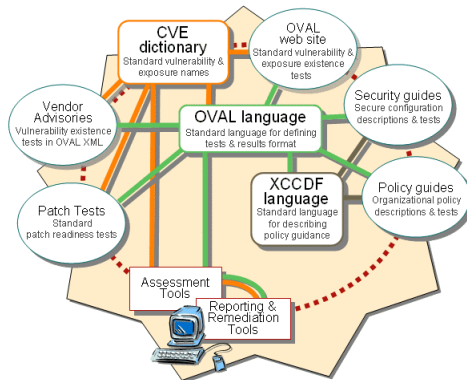


Schéma de synthèse du standard

(https://cve.mitre.org/docs/docs-2005/transformational_standards.html)

Solutions professionnelles

- SCAP est aujourd'hui supporté par diverses outils et fournisseurs de logiciels de sécurité
 - Nessus Manager, Ms Configuration Manager 2007, SAINT Security Suite, Qualys...
- Bien que SCAP soit basé sur un ensemble de standards, les fiches OVAL ou les STIGs ne sont pas toujours interoperables
 - Certains éditeurs, comme CIS, enrichissent le langage
 - La compréhension de la norme peut varier selon les implémentations

Solutions Open-Source

- A ce jour, le projet Open-Source le plus abouti est Open-SCAP, initié par le NIST
- Les politiques de sécurité sont portées par le projet SCAP-Security-Guide

Limitations des solutions Open-Source

- Très orientées GNU/Linux, le support des *BSD est de Windows en est à ses balbutiement
- Un long travail d'enrichissement des sondes est nécessaire

Sommaire

1 MCS

2 SCAP

3 Utiliser SCAP

- OpenSCAP et SCAP-Security-guide
- Tester la conformité d'une cible à une politique de sécurité
- Appliquer des scripts de remédiation
- Gérer la sécurité d'une infrastructure

4 Dans les arcanes du standard

5 Annexes

OpenSCAP

- Implémentation en C
- Supporte un ensemble de sonde correspondant au diverses abstraction OVAL
- Chaque sonde gère une tâche particulière
 - Parsing d'un fichier
 - Intérogation d'un gestionnaire de services
 - Intérogation d'un gestionnaire de paquets
- Gère la génération de rapports aux formats HTML, XML et ARF
- Sait lancer des scripts de remédiation si fournis en entrée
- Sait traiter en entrée des données au formats OVAL ou XCCDF principalement

SCAP-Security-Guide

- Contient les politiques au format OVAL et XCCDF
- Se base sur un moteur de génération écrit en Python
- S'appuie sur OpenSCAP pour construire les STIGs et les benchmarks associés
- Supporte divers STIGs (PCI-DSS, Notes techniques ANSSI...)
- Supporte divers cibles GNU/Linux
- Supporte divers cibles applicatives (Chrome, JBoss, JRE...)
- Supporte les scripts de remédiation dans divers format

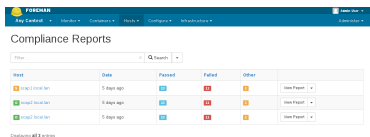
Etat des lieux du projet

- Toutes les distributions et applicatifs ne sont pas supportés de manière égale
- Un gros effort est en cours sur le support des scripts de remédiation
- La complétudes des STIGs n'est pas toujours complète

Tester la conformité d'une cible à une politique de sécurité

Appliquer des scripts de remédiation

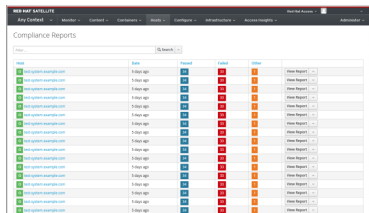
Centraliser les rapports, gérer le SI



Host	Date	Passed	Failed	Other	
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report

Displaying all 3 entries

Intégration d'OpenSCAP à Foreman
https://www.theforeman.org/plugins/foreman_openscap/0.4/)



Host	Date	Passed	Failed	Other	
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report
img2 local lan	5 days ago	100%	0%	0%	View Report

Intégration d'OpenSCAP à Red-Hat Satellite (https://access.redhat.com/documentation/en-us/red_hat_satellite/6.2/html-single/host_configuration_guide/)

Centraliser les rapports, gérer le SI

Interface du daemon OpenSCAP :

```
# oscapd-cli task
```

```
-----+-----+-----+-----+-----+
ID | Title          | Target                      | Modified                | Enabled
-----+-----+-----+-----+-----+
1  | hebdo-t12      | my-debian-laptop           | 2017-04-13 12:44:38    | enabled
```

```
# oscapd-cli result 1
```

```
Results of Task "hebdo-t12", ID = 1
```

```
-----+-----+-----+
ID | Timestamp          | Status
-----+-----+-----+
1  | 2017-04-13 15:51:34 | Non-Compliant
2  | 2017-04-20 11:51:29 | Non-Compliant
3  | 2017-04-27 15:51:41 | Compliant
```

```
# oscapd-cli result 1 2 arf > exported-arf-result.xml
```

```
# oscapd-cli result 1 2 report > exported-report.html
```

Sommaire

1 MCS

2 SCAP

3 Utiliser SCAP

4 Dans les arcanes du standard

- OVAL : rédiger une fiche de test
- XCCDF : Ecrire une politique de sécurité dédiée
- Les profils et les politiques de sécurité

5 Annexes

Formalisme de base

- Une fiche OVAL est une définition, à laquelle on associe un ou plusieurs tests
- Une définition peut être de type :
 - compliance (conformité à une exigence)
 - inventory (détection d'un composant - au sens du standard CPE)
 - patch (détection de la présence d'un patch)
 - vulnerability (détection d'une vulnérabilité)
 - miscellaneous (tout le reste)
- une fiche OVAL possède un identifiant unique (dénnoté *id*), une description, un titre et une ou plusieurs cible(s)

Critères de validation de la fiche

- Une fiche OVAL est vraie lorsque :
 - la cible correspond à celle définie dans la fiche OVAL
 - l'ensemble des critères nécessaires et suffisants à sa validation ont été joués avec un résultat positif

Les critères OVAL

- Les critères utilisent une logique booléenne récursive
- On peut utiliser la mécanique booléenne pour construire des fiches complexes
 - Appliquant des tests différents selon la plateforme cible
 - Se comportant différemment selon l'architecture matérielle
 - etc.
- Les critères permettent de hiérarchiser les fiches, une fiche pouvant enrichir (et donc dépendre) d'une autre
- Le sur-usage de la logique booléenne des critères peut cependant rendre la fiche illisible

Les sélecteurs OVAL

- Des politiques de sécurités de plus haut niveau (comme celles du NIST) peuvent impliquer plusieurs STIGs
- Les STIGs peuvent impliquer des niveaux de durcissement variables selon le profil
- Afin de ne pas multiplier les fiches OVAL, le langage permet l'usage de variables, qu'on dénote sélecteur

Ecrire une fiche avec des sélecteurs

- Un sélecteur est une variable dans un test
- La variable est désignée par son nom, qui devra être résolu au moment de la construction de la checklist XCCDF
- Une fiche OVAL paramétrée impose, pour chaque profil qui l'utilise, de spécifier le sélecteur parmi les choix possibles
- Une fiche OVAL paramétrée impose, dans la checklist XCCDF, de lister exhaustivement l'ensemble des possible pour le sélecteur

Exemple simple de fiche OVAL

```

1 <def-group>
2   <definition class="compliance" id="partition_for_tmp" version="1">
3     <metadata>
4       <title>Ensure /tmp Located On Separate Partition</title>
5       <affected family="unix">
6         <platform>Debian 8</platform>
7       </affected>
8       <description>
9         The /tmp directory is a world-writable directory used for
10        temporary file storage. Verify that it has its own partition or logical
11        volume.
12      </description>
13    </metadata>
14    <criteria>
15      <criterion test_ref="test_tmp_partition" comment="/tmp on own partition" />
16    </criteria>
17  </definition>
18  <linux:partition_test check="all" check_existence="all_exist" id="test_tmp_partition" version
19    = "1" comment="/tmp on own partition">
20    <linux:object object_ref="object_own_tmp_partition" />
21  </linux:partition_test>
22  <linux:partition_object id="object_own_tmp_partition" version="1">
23    <linux:mount_point>/tmp</linux:mount_point>
24  </linux:partition_object>
25 </def-group>

```

OVAl : rédiger une fiche de test

Exemple fiche OVAL avec critères avancés

```

1 <def-group>
2   <definition class="compliance" id="accounts_password_pam_retry" version="1">
3     <metadata>
4       <title>Set Password retry Requirements</title>
5       <affected family="unix">
6         <platform>multi_platform_fedora</platform>
7         <platform>multi_platform_rhel</platform>
8         <platform>multi_platform_rhel-osp</platform>
9       </affected>
10      <description>The password retry should meet minimum requirements</description>
11    </metadata>
12    <criteria operator="OR" comment="Conditions for retry are satisfied">
13      <criteria operator="AND" comment="system with pam_cracklib configured">
14        <criteria operator="OR" comment="OSes with cracklib primarily used">
15          <extend_definition comment="RHEL6 OS installed" definition_ref="installed_OS_is_rhel6"
16            />
17          <extend_definition comment="CentOS6 OS installed" definition_ref="
18            installed_OS_is_centos6" />
19        </criteria>
20        <criteria comment="rhel6 pam_cracklib" test_ref="test_password_pam_cracklib_retry" />
21      </criteria>
22      <criteria operator="AND" comment="system with pam_pwquality configured">
23        <criteria operator="OR" comment="OSes with pwquality primarily used">
24          <extend_definition comment="RHEL7 OS installed" definition_ref="installed_OS_is_rhel7"
25            />
26          <extend_definition comment="CentOS7 OS installed" definition_ref="
27            installed_OS_is_centos7" />
28          <extend_definition comment="Fedora OS installed" definition_ref="
29            installed_OS_is_fedora" />
30        </criteria>
31        <criteria comment="pam_pwquality" test_ref="test_password_pam_pwquality_retry" />
32      </criteria>
33    </criteria>
34  </definition>

```

OVAl : rédiger une fiche de test

Exemple de fiche OVAl avec sélecteur

```

1 <def-group>
2   <definition class="compliance" id="accounts_password_pam_minclass" version="1">
3     [...]
4   </definition>
5   <ind:textfilecontent54_test check="all"
6     comment="check the configuration of /etc/security/pwquality.conf"
7     id="test_password_pam_pwquality_minclass" version="1">
8     <ind:object object_ref="obj_password_pam_pwquality_minclass" />
9     <ind:state state_ref="state_password_pam_pwquality_minclass" />
10  </ind:textfilecontent54_test>
11
12  <ind:textfilecontent54_object id="obj_password_pam_pwquality_minclass"
13    version="1">
14    <ind:filepath>/etc/security/pwquality.conf</ind:filepath>
15    <ind:pattern operation="pattern match">~minclass[\s]*=[\s]*(-?\d+)(?:[\s]|\$)</ind:pattern>
16    <ind:instance datatype="int" operation="less than or equal">1</ind:instance>
17  </ind:textfilecontent54_object>
18
19  <ind:textfilecontent54_state id="state_password_pam_pwquality_minclass"
20    version="1">
21    <ind:instance datatype="int">1</ind:instance>
22    <ind:subexpression datatype="int" operation="greater than or equal"
23      var_ref="var_password_pam_minclass" />
24  </ind:textfilecontent54_state>
25
26  <external_variable comment="External variable for pam_pwquality minclass"
27    datatype="int" id="var_password_pam_minclass" version="1" />
28 </def-group>

```

XCCDF : Ecrire une politique de sécurité dédiée

Principe d'une checklist XCCDF

- Une checklist XCCDF permet de générer autant de STIGs que de profils définit dans la checklist
- une checklist au format XCCDF permet de valider la conformité d'une cible à un STIG donné

XCCDF : Ecrire une politique de sécurité dédiée

une checklist XCCDF

- A pour but de générer des STIGs et les checklist associées
- Inclut autant de profils que nécessaire, à partir du moment où l'ensemble des fiches et règles XCCDF les définissant sont présents
- Est composée de groupes, structurant les guides
- Chaque groupe contient une ou plusieurs règles
- Chaque règle se définit par
 - Un titre
 - Une description
 - Un rationnel
 - Une traçabilité à une ou plusieurs politiques, sous forme de références
 - Un identifiant de configuration CCE associé (s'il existe)
 - Une fiche OVAL décrivant la méthode de validation de la règle
- On peut également y trouver des références OCIL au besoin

Exemple de checklist XCCDF

```

1 <Group id="accounts-pam">
2 <title>Protect Accounts by Configuring PAM</title>
3 <description>Group introduction text</description>
4 <warning category="general">Potential group-wide warning infos
5 <weblink-macro link="http://url/toward/external-content#subchapter"/>
6 </warning>
7 [...]
8 <Value id="var_password_pam_minclass" type="number" operator="equals" interactive="0">
9 <title>minclass</title>
10 <description>Minimum number of categories of characters that must exist in a password</
    description>
11 <value selector="">3</value>
12 <value selector="1">1</value>
13 [...]
14 </Value>
15
16 <Rule id="accounts_password_pam_minclass" severity="medium" prodtype="rhel7">
17 <title>Set Password Strength Minimum Different Categories</title>
18 <description>The rule description...
19 </description>
20 <ocil clause="minclass is not found or not set equal to or greater than the required value">
21 OCIL informational when clause is true
22 </ocil>
23 <rationale>
24 Rule rationale explanation.
25 </rationale>
26 <ident prodtype="rhel7" cce="CCE-27115-5" />
27 <oval id="accounts_password_pam_minclass" value="var_password_pam_minclass"/>
28 <ref prodtype="rhel7" stigid="010170" />
29 <ref nist="IA-5" disa="195" ossrg="SRG-OS-000072-GPOS-00040" />
30 </Rule>

```

Définition d'un profil

Un profil correspond à la formalisation XCCDF d'un STIG.

Elle correspond à :

- une suite de règles XCCDF
- un ensemble de sélecteurs, nécessaires pour spécialiser les règles

Ainsi :

- le profil DAT-NT-28 niveau élevé est un sur-ensemble du profil DAT-NT-28 niveau restreint
- le profil PCI-DSS spécifie des sélecteurs pour diverses règles XCCDF comme la taille et la complexité des mots de passes

Exemple de profil

```

1 <Profile id="pci-dss" xmlns="http://checklists.nist.gov/xccdf/1.1">
2 <title>PCI-DSS v3 Control Baseline for Red Hat Enterprise Linux 7</title>
3 <description>This is a *draft* profile for PCI-DSS v3.</description>
4
5 <refine-value idref="var_password_pam_unix_remember" selector="4" />
6 <refine-value idref="var_account_disable_post_pw_expiration" selector="90" />
7 <refine-value idref="var_accounts_passwords_pam_faillock_deny" selector="6" />
8 <refine-value idref="var_accounts_passwords_pam_faillock_unlock_time" selector="1800" />
9 <refine-value idref="sshd_idle_timeout_value" selector="15_minutes" />
10 <refine-value idref="var_password_pam_minlen" selector="7" />
11 <refine-value idref="var_password_pam_minclass" selector="2" />
12 <refine-value idref="var_accounts_maximum_age_login_defs" selector="90" />
13 <refine-value idref="var_auditd_num_logs" selector="5"/>
14
15 <select idref="service_auditd_enabled" selected="true"/>
16 <select idref="bootloader_audit_argument" selected="true"/>
17 <select idref="auditd_data_retention_num_logs" selected="true"/>
18 <select idref="auditd_data_retention_max_log_file" selected="true"/>
19 <select idref="auditd_data_retention_max_log_file_action" selected="true"/>
20 <select idref="auditd_data_retention_space_left_action" selected="true"/>
21 <select idref="auditd_data_retention_admin_space_left_action" selected="true"/>
22 <select idref="auditd_data_retention_action_mail_acct" selected="true"/>
23 <select idref="auditd_audispd_syslog_plugin_activated" selected="true"/>
24 [...]
25 <select idref="file_group_owner_grub2_cfg" selected="true"/>
26 <select idref="package_libreswan_installed" selected="true"/>
27
28 </Profile>

```

Dimensionnement

Dans les faits, la rédaction d'un STIG au format SCAP demande du temps.

- STIG PCI-DSS pour RHEL7 : profil de 94 règles, 10 sélecteurs
- STIG DISA pour RHEL7 : profil de 200 règles, 37 sélecteurs
- La complétude des checklists restent soumis à validation humaine
- Plus de 900 fiches OVAL pour RHEL7, plus de 250 pour Debian, etc.

Charge de travail

Il ne faut pas sous-estimer la charge associée à la rédaction et au maintien d'un écosystème SCAP

- Les fiches OVAL sont globalement très nombreuses et évoluent avec les guides
- La rédaction des guides restera toujours un travail complexe, impliquant plusieurs expertises

Automatisation

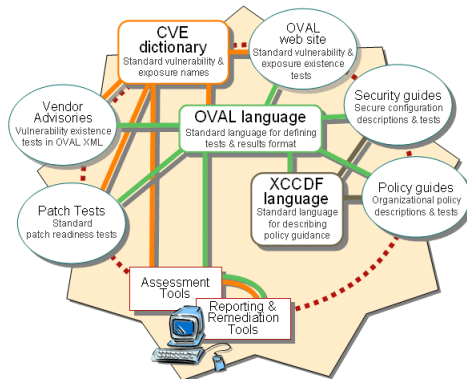
- La rédaction des checklist XCCDF peut difficilement être automatisé
- La génération de fiches OVAL est parfaitement automatisable
 - Tests très redondants (présence d'une partition, droits d'un fichier/dossier)
 - Peu de contenu descriptif (pas de rationale, etc.)
- La grande majorité des fiches OVAL du SCAP-security-guide sont générées à partir de templates

Répartition des rôles

- Les fiches OVAL définissant un patch ou une vulnérabilité sont souvent à la charge de l'éditeur
- Elles sont souvent générées automatiquement en même temps que la CVE, ou à partir d'elle

Le workflow SCAP... reprise

Standards for Enabling Automation in Information Security



Synthèse du workflow SCAP

(https://cve.mitre.org/docs/docs-2005/transformational_standards.html)

Les profils et les politiques de sécurité



Sommaire

1 MCS

2 SCAP

3 Utiliser SCAP

4 Dans les arcanes du standard

5 Annexes

- Acronymes

- Table des références

Acronymes

Acronyme	Signification
ARF	Asset Reporting Format
CCE	Common Configuration Enumeration
CPE	Common Platform Enumeration
CVE	Common Vulnerability Disclosure
CCSS	Common Configuration Scoring System
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
OCIL	Open Checklist Interactive Language
OS	Operating System
OVAL	Open Vulnerability and Assessment Language
STIG	Security Technical Implementation Guide
TMSAD	Trust Model for Security Automation Data
XCCDF	Extensible Configuration Checklist Description Format

Sites web utiles

- Cvedetails : <https://www.cvedetails.com>
- PSSIE (Politique de Sécurité des Système d'Information de l'Etat)
https://www.ssi.gouv.fr/uploads/2014/11/pssie_anssi.pdf
- La PPST (Protection du Potentiel Scientifique et Technique)
<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000026140136&dateTexte=20170605>
- Base des CWE du MITRE (<http://cwe.mitre.org>)
- <https://scap.nist.gov/revision/1.2/index.html>
Spécifications de la dernière version finale du standard SCAP (1.2)