atsec information security
Tel: +86-10-82893001
Fax: +86-10-82890017
www.atsec.com

SCAP 标准简介

atsec 张力 2010 年 12 月

关键词：SCAP、OVAL、CPE、CVE、CCE、CVSS

2010 年 11 月 7 日至 16 日，IETF79 研讨会在北京召开，全球范围内数以千计的科研工作者参加了这次盛会，其中一个很重要的议题就是讨论将安全内容自动化协议（SCAP： Security Content Automation Protocol）引入 IETF 标准化，使之成为真正意义上的全球标准。

1 SCAP 的产生背景

由于计算机通信技术的飞速发展，美国联邦政府强烈的意识到由于计算机系统配置问题而暴漏出越来越多的安全漏洞，为此，2007 年美国联邦预算管理办公室（OMB：Office of Management and Budget）提出要求所有的政府部门开始试行联邦桌面核心配置计划（FDCC：Federal Desktop Core Configuration）。2008 年则开始强制执行。FDCC 最初是由美国国家标准与技术研究所联合 OMB、DHS（Department of Homeland Security）、NSA（National Security Agency）以及 Microsoft 共同开发，用于美国空军 Windows XP 的公共安全配置，2008 年 6 月发布第一个版本 FDCC1.0，在本文发稿前最新的版本为 2009 年 8 月发布的 1.2 版本。

FDCC 定义了针对 Windows XP 与 Vista 操作系统的公共配置准则。为了检测 FDCC 的合规性，NIST 开发了安全内容自动化协议 SCAP，以标准化的方式表达与使用安全数据，以及进行安全问题评估。最初主要针对于 Windows 操作系统开发检查列表（checklist），后来逐步扩展到 UNIX 系统，web 浏览器，反病毒软件以及防火墙等。

SCAP 提供了一种自动、标准化的方法来维护企业系统的安全，如实现安全配置基线，验证当前的补丁程序，进行系统安全配置设置的持续性监测，检查系统的折衷标记（sign of compromise），以及能在任意设定时刻给出系统的安全状态。SCAP 的提出主要源于如下几个方面的原因：

- 大量的以及多样的系统需要保护。

  大多数组织有需要保护的一些系统，针对每个系统有众多的应用需要保护。一般一个企业内部装有多种操作系统及上千种的应用软件，每个系统或应用都有自己的补丁机制及安全配置管理。相同的软件在不同主机上的保护机制也可能会有些许的不同。一个单独的主机针对它的操作系统与应用也会有上千种安全配置设置。所有这些因素使得决定每个系统上需要什么样的安全变化，快速、正确、一致地实现这些变化，以及验证安全配置更为复杂。

- 快速响应新的威胁。

  一些组织经常需要重新配置软件或安装补丁以消除新发现地或成为攻击者目标的脆弱性。在 2009 年，有多达 5,700 个软件缺陷被加入到美国国家脆弱性数据库（NVD：National Vulnerability Database）。

- 缺乏互操作性。

  大多数系统安全工具采用私有格式、命名法、测量方法与内容，如补丁管理与脆弱性管理软件。例如，如果脆弱性扫描器没有采用标准化的弱点命名法，安全管理人员将不清楚多个扫描器的扫描结果报告是否指向相同的脆弱性。这种互操作性的缺乏将导致安全评估的不一致性，延迟制作决定以及做出及时的修正。

2 SCAP 协议框架

SCAP 包含两个主要元素。首先，它是一个协议，一组标准化格式与术语的开放规范，通过它软件安全产品可以互通软件缺陷与安全配置信息，每一个规范也被称作一个 SCAP 组件；其次，SCAP 包括软件缺陷与安全配置标准化的参考数据，也被称作 SCAP 内容。

下表列出了目前的 SCAP 1.0 协议组件，这些组件按类型分为：列举（Enumerations）组，为安全与产品相关的信息定义了标准表述符与目录；脆弱性测量与评分（Vulnerability Measurement and Scoring）组，用于测量脆弱性特征以及根据这些特征给予评分；表述与检查语言（Expression and Checking Languages）组，运用 XML Schema 说明检查列表，产生检查列表报告，以及说明用于检查列表的低级测试过程。

Table 1 **SCAP 1.0 协议组件**

SCAP 组件	描述	维护组织
列举		
通用配置列举（CCE：Common Configuration Enumeration）	定义了与安全相关的系统配置项的标准标识符与目录	MITRE Corporation
通用平台列举（CPE：Common Platform Enumeration ）	定义了平台及版本大多标准名称与目录	MITRE Corporation
通用脆弱性与漏洞列举（CVE：Common Vulnerabilities and Exposures)	定义了与软件缺陷相关的标准标识符与目录	MITRE Corporation
脆弱性测量与评分		
通用脆弱性评分系统（CVSS：Common Vulnerability Scoring System)	定义了脆弱性对系统影响的评分	Forum of Incident Response and Security Teams (FIRST)
表述与检查语言		
扩展配置检查列表描述格式（XCCDF：Extensible Configuration Checklist Description Format)	定义了检查列表与检查报告的XML描述格式	National Security Agency (NSA) and NIST
开放脆弱性评估描述语言（OVAL：Open Vulnerability and Assessment Language)	定义了用于检查列表的低级测试过程	MITRE Corporation

关于SCAP content，存在多个资源，例如，NVD拥有CPE与CVE标识项，MITRE公司拥有OVAL数据库，并且维护CCE标识项的列表。每一个SCAP组件提供唯一的功能，可以被独立地运用，但是联合的运用能带来更大的好处，例如，用XCCDF格式依据CPE表达CCE的能力形成了用于SCAP表达检查列表的基本元素，换句话说，SCAP表达的检查列表用一种标准化的语言（XCCDF）来表达所讨论的平台是什么（CPE），访问什么安全设置（CCE）。运用SCAP表达检查列表可以很容易的帮助组织实现对系统的安全控制，进行安全监测，自动化高级安全需求的合规性报告。

总之检查列表用SCCDF来描述评估什么，用OVAL在目标系统做相应的测试，用CPE标识检查列表是有效的以及运行测试的平台，用CCE标识在检查列表中被访问或被评估的安全配置设置，用CVE参考已知的脆弱性，CVSS用于分级这些脆弱性。

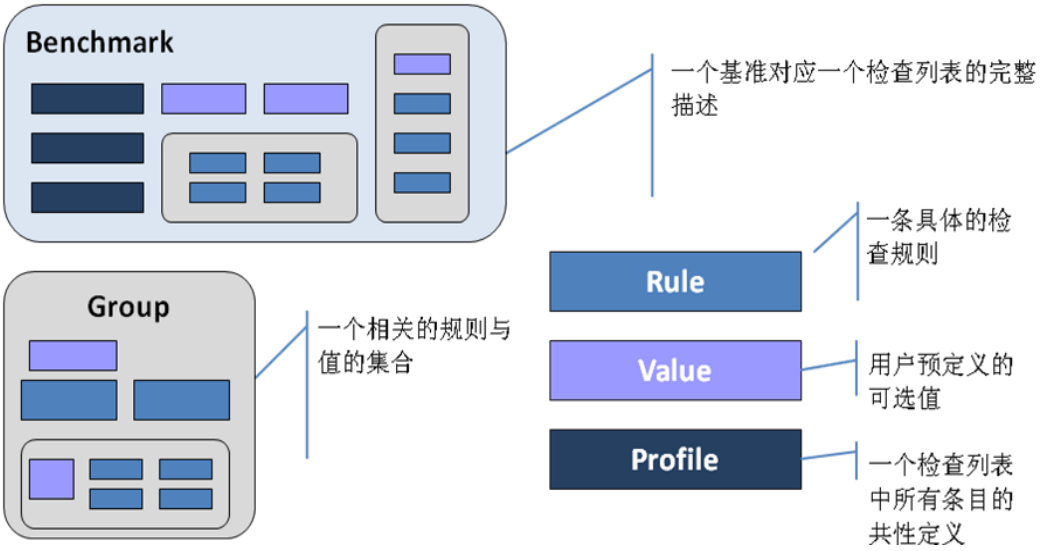
2.1 XCCDF

XCCDF 基本数据模型由以下四个主要的对象数据类型组成：

- 基准（Benchmark）。可以简单理解为检查列表，一个 XCCDF 只能拥有一个基准对象，它相当于一个容器，包含条目及其它的对象。

- 条目（Item）。条目是一个基准对象中的命名要素，每一条目有一个唯一的参考 ID。条目包含以下三种类型：
  a) 组（Group）。这种类型条目可以包含其它条目，即一个组可以包含其它的组以及规则与值。一个组可以被选择，也可以不被选择。
  b) 规则（Rule）。这种类型条目定义具体的检查规则，可以拥有检查参考与评分权重。一个规则可以被选择，也可以不配选择。
  c) 值（Value）。这种类型条目是一命名的数据值，可以理解为允许用户为每种规则预定义的可选值。
- 轮廓（Profile）。轮廓定义了对规则、组与值对象的共性收集。
- 测试结果（TestResult）。测试结果对象记录单独的设备或系统的合规性测试结果。

下图显示了这几种数据对象类型的关系。



XCCDF 的目的是提供一种统一的方式描述安全检查列表与检查列表的评估结果。XCCDF 文档由一个或多个 XCCDF 规则组成。每一个 XCCDF 规则是系统中一个技术检查点的高级定义。一个规则没有直接说明怎样做一个检查，而是间接的通过 OVAL 的定义文件来说明相应的检查方法。

2.2 OVAL

OVAL 用于表达标准化的、机器可读的规则，这些规则能用于评估系统的状态。在 SCAP 框架下，OVAL 通常用于决定存在的缺陷与不安全配置。一套用于检查安全问题的指令(例如：一不正确的最小口令长度设置)被称作一个 OVAL 定义（Definition），包含一个或多个 OVAL 定义的文件称为一个 OVAL 定义文件。

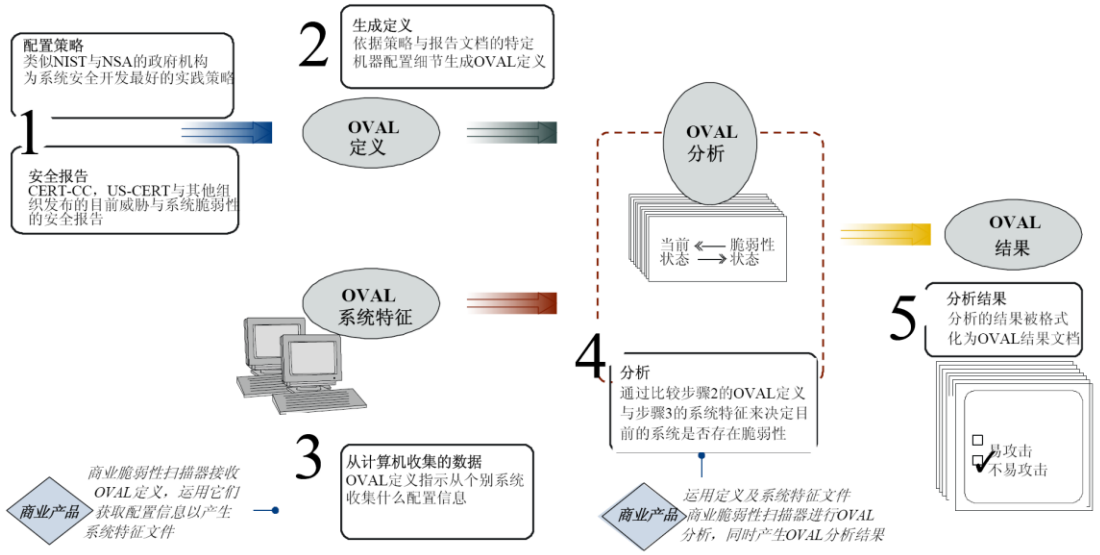
有四种类型的 OVAL 定义：

- 脆弱性定义：定义了针对特定的脆弱性的出现，在计算机上必须存在的条件。

- 补丁定义：定义条件以决定是否一个特定的补丁适合一个系统。
- 详细清单定义：定义条件以决定是否软件的特定部分被安装在系统上。
- 合规性定义：定义条件以决定是否计算机已符合特定的策略或配置描述。

OVAL 语言借助 XML 语法描述，包含三种核心的 Schema，即 OVAL 定义 Schema，对以上四种类型的 OVAL 定义进行描述；OVAL 系统特征 Schema，根据系统的实际特征进行相应的编码描述；OVAL 结果 Schema，是对分析结果的详细编码描述。

下图说明了怎样运用 OVAL 三种核心 Schema 描述与分析一个实际系统脆弱性过程。



2.3 CPE，CVE 与 CCE

这部分描述来了SCAP1.0中的三个列举规范：CPE 2.2，CCE 5与CVE。SCAP列举基本上都包含一个ID，一个相关的描述或定义，与一个支持的参考列表。下面对每一个规范以及它们之间的相互依赖关系给予说明。

CPE 2.2 是对操作系统、硬件与软件的标准命名规范，它使得不同的组织可以共享相同的名字，以及针对特定平台获得相同的解决方案。一个单独的 CPE 命名语法如下：

cpe:/{part}:{vendor}:{product}:{version}:{update}:{edition}:{language}

例如，“cpe:/o:redhat:enterprise_linux:2.1::es ”是指 RedHat 企业服务器版 2.1，“o”指示这个 CPE 描述一个操作系统，在这个例子中，“edition”域是空白，表示这个 CPE 应用于 RedHat 企业版服务器 2.1 的所有版本。CPE 在 SCAP 中应用有以下几种方式：

- XCCDF： 在一个 XCCDF 检查列表中，CPE 名字能用于标识一个 XCCDF 对象（benchmark，profile，group 或 rule）应用的硬件或软件平台。
- CCE：CCE 通过关联 CPE 用以说明配置缺陷发生的平台。
- CVE：CVE 被关联到 CPE 描述的一个或多个平台。这种映射关系被保存在 NVD 中。

CCE 5 用于标准化安全配置问题与条目。每一种安全相关的配置问题被分配一个唯一的 ID 以便于快速、精确地发现多个信息源与工具之间配置数据的相互关系。MITRE 公司维护与出版

CCE 列表，每个 CCE 条目包括 5 种属性：一个唯一的 ID，一个配置问题的描述，CCE 的逻辑参数，相关的技术机制，与附加信息源的参考。下面是一个用于 Windows XP 中 CCE 条目的例子。

CCE ID: CCE-3108-8
Definition: The correct service permissions for the Telnet service should be assigned.
Parameters: (1) set of accounts (2) list of permissions
Technical Mechanisms: (1) set via Security Templates (2) defined by Group Policy
References: Listed at http://cce.mitre.org/lists/cce_list.html

在一个 XCCDF 检查列表中，CCE 用于说明哪些安全配置设置应被检查，OVAL 运用 CCE 条目也是同样的目的。

CVE 用于标准化公共已知的软件缺陷。这种通用的命名机制允许多个组织间共享数据，有效地集成服务于工具。例如，一个纠错工具可以运用几个扫描器与检测传感器的 CVE 信息来制定一种集成的风险消除解决方案。CVE 的好处表现在：

- 公共已知软件缺陷的全面的列表
- 一个全球唯一的名字来标识每一个脆弱性
- 讨论脆弱性特性与风险的基础
- 是用户集成脆弱性信息到不同工具与服务的方式

每个CVE条目由一个唯一名字，一个简短描述与参考组成。下面是一CVE条目的例子。

CVE ID: CVE-2000-0001
Description: RealMedia server allows remote attackers to cause a denial of service via a long ramgen request.
References: BUGTRAQ: 19991222 RealMedia Server 5.0 Crasher (rmscrash.c)
BID: 888. URL:http://www.securityfocus.com/bid/888

2.4 CVSS

CVSS2.0提供了一种可重复的方法以评估和表达软件缺陷的风险。运用这种共享的评分模型可以很容易的比较脆弱性的严重程度。CVSS提供了以下三种尺度用以权重脆弱性的评分：

- 基本因素，脆弱性的内在因素决定的基本分数。
- 时间因素，捕捉随时间而改变的外部因素，考虑时间因素来调整基本分数。
- 环境因素，它标识了一个组织操作环境的上下文中这种脆弱性的严重程度。

CVSS可以帮助组织理解多种脆弱性的相对重要性，从而使他们能有效地评估、优化与消除脆弱性。因为每周会公开发布数百个脆弱性问题，因此找到一种容易的方法标识哪些脆弱性会对系统带来重要的影响是非常重要的。NVD分析专家为所有的CVE条目计算与发布CVSS基本评分，但是每个组织将根据他们特定的时间因素与环境因素来裁剪基本评分以得到实际的脆弱性评分。

3 标准的评估认证

NIST 已经建立了 SCAP 产品认证体系与 SCAP 实验室委任体系，这些体系一起确保 SCAP 产品的测试与验证过程。SCAP 测评实验室需要经过美国国家实验室自愿认可计划（NVLAP）的授权。实验室一经授权，实验室就可以根据 NISTIR （National Institute of Standards and

Technology Interagency Report） 7511 中的 DTR (Derived Test Requirements)的描述进行 SCAP 产品测试。产品经测试后，测评实验室将发布测试报告（包括特定产品的需求列表，被需要的开发商文档，由实验室做的详细的测试总结）给 SCAP 产品认证体系，产品认证体系专家将审阅测试报告，然后发布产品认证。

一个产品可以被认证符合六个 SCAP 组件规范的一个或多个，或单独的符合一个特定的 SCAP 能力。所谓 SCAP 能力不是指产品类型 ，而是产品运用 SCAP 的方式，如认证的脆弱性扫描器、认证的配置扫描器、入侵检测与预防、漏洞修补、错误配置修补、设备管理、脆弱性数据库等等。随着 SCAP 被应用到更多类型的安全工具，SCAP 能力列表将随着时间而不断发展。SCAP 认证体系保证一个产品符合一套 SCAP 能力与/或一个或多个 SCAP 组件规范。

如果没有重新认证的话，SCAP 产品的认证有效期仅为一年，这保证 SCAP 产品始终与 SCAP 技术发展同步，持续的结合 SCAP 的参考数据为基础，使用最新的与改进的方法对产品进行重新测试。

4 SCAP 展望

截至此文章发稿前，已有30个厂商获得了扫描与审计产品的SCAP认证，正式通过SCAP认证的厂商可以参考链接：http://nvd.nist.gov/scapproducts.cfm。我们不难发现越来越多的厂商，组织与社团加入到SCAP的研究与发展中，它已经成为业界事实上的标准，目前正式发布的版本为SCAP 1.0（包括XCCDF 1.1.4，OVAL 5.3与5.4，CPE 2.2，CCE 5，CVE与CVSS 2.0），SCAP1.1目前处于修订中，并且SCAP已经正式提交IETF讨论组，相信不久的将来即会成为正式的全球标准。

5 参考文献

[1] SCAP Homepage. http://scap.nist.gov

[2] The Extensible Configuration Checklist Description Format (XCCDF) Version 1.1.4. https://datatracker.ietf.org/doc/draft-waltermire-scap-xccdf/

[3] Naming Conventions for Vulnerabilities and Configurations. https://datatracker.ietf.org/doc/draft-landfield-scap-naming/

[4] Open Vulnerability and Assessment Language (OVAL). http://oval.mitre.org/

[5] Common Configuration Enumeration (CCE). http://cce.mitre.org/

[6] Common Platform Enumeration (CPE). http://cpe.mitre.org/

[7] Common Vulnerabilities and Exposures (CVE). http://cve.mitre.org/

[8] Common Vulnerability Scoring System (CVSS). http://www.first.org/cvss/

[9] National Voluntary Laboratory Accreditation Program (NVLAP). http://ts.nist.gov/standards/accreditation/index.cfm

[10] SCAP Validated Tools. http://nvd.nist.gov/scapproducts.cfm

[11] National Checklist Program. http://checklists.nist.gov

[12] National Vulnerability Database. http://nvd.nist.gov